



Red Hat Enterprise Linux 8

Identity Management 구성 및 관리

Red Hat Enterprise Linux 8에서 Identity Management 구성, 관리 및 유지 관리

Red Hat Enterprise Linux 8 Identity Management 구성 및 관리

Red Hat Enterprise Linux 8에서 Identity Management 구성, 관리 및 유지 관리

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/Configuring_and_managing_Identity_Management.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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

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

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

All other trademarks are the property of their respective owners.

초록

이 문서 컬렉션은 Red Hat Enterprise Linux 8의 ID 관리를 효과적으로 구성, 관리 및 유지 관리하는 방법에 대한 지침을 제공합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체	18
RED HAT 문서에 관한 피드백 제공	19
1장. 명령줄에서 IDENTITY MANAGEMENT에 로그인	20
1.1. KINIT 를 사용하여 IDM에 수동으로 로그인합니다.	20
1.2. 사용자의 활성 KERBEROS 티켓 삭제	21
1.3. KERBEROS 인증을 위한 외부 시스템 구성	21
2장. ID 관리 서비스 보기, 시작 및 중지	23
2.1. IDM 서비스	23
2.1.1. IdM 서버에서 호스팅하는 서비스 목록	23
2.1.2. IdM 클라이언트가 호스팅하는 서비스 목록	24
2.2. IDM 서비스 상태 보기	25
2.3. 전체 ID 관리 서버 시작 및 중지: IPACTL 유틸리티	26
ipactl 명령	26
2.4. 개별 ID 관리 서비스 시작 및 중지: SYSTEMCTL 유틸리티	26
유용한 systemctl 명령	27
2.5. IDM 소프트웨어 버전을 표시하는 방법	28
3장. IDM 명령줄 유틸리티 소개	29
3.1. IPA 명령줄 인터페이스란 무엇입니까?	29
3.2. IPA 도움말이란 무엇입니까?	29
3.3. IPA 도움말 항목 사용	30
3.4. IPA 도움말 명령 사용	30
3.5. IPA 명령 구조	31
3.6. IPA 명령을 사용하여 IDM에 사용자 계정 추가	32
3.7. IPA 명령을 사용하여 IDM에서 사용자 계정 수정	33
3.8. IDM 유틸리티에 값 목록을 제공하는 방법	34
3.9. IDM 유틸리티와 함께 특수 문자를 사용하는 방법	34
4장. 명령줄에서 IDENTITY MANAGEMENT 항목 검색	36
4.1. IDM 항목 나열 개요	36
4.2. 특정 항목의 세부 정보 표시	36
4.3. 검색 크기 및 시간 제한 조정	37
4.3.1. 명령줄에서 검색 크기 및 시간 제한 조정	37
4.3.2. 웹 UI에서 검색 크기 및 시간 제한 조정	38
5장. 웹 브라우저에서 IDM 웹 UI에 액세스	40
5.1. IDM 웹 UI란 무엇입니까	40
5.2. 웹 UI에 액세스하는 데 지원되는 웹 브라우저	40
5.3. 웹 UI에 액세스	40
6장. 웹 UI에서 IDM에 로그인: KERBEROS 티켓 사용	42
6.1. IDENTITY MANAGEMENT의 KERBEROS 인증	42
6.2. KINIT 를 사용하여 IDM에 수동으로 로그인합니다.	42
6.3. KERBEROS 인증을 위한 브라우저 구성	43
6.4. KERBEROS 티켓을 사용하여 웹 UI에 로그인	44
6.5. KERBEROS 인증을 위한 외부 시스템 구성	45
6.6. ACTIVE DIRECTORY 사용자를 위한 웹 UI 로그인	46
7장. 일회성 암호를 사용하여 ID 관리 웹 UI에 로그인	47
7.1. 사전 요구 사항	47

7.2. IDENTITY MANAGEMENT에서 한 번만 암호(OTP) 인증	47
7.3. 웹 UI에서 일회용 암호 활성화	47
7.4. 웹 UI에서 OTP 토큰 추가	48
7.5. 일회성 암호로 웹 UI에 로그인	49
7.6. 웹 UI를 사용하여 OTP 토큰 동기화	50
7.7. 만료된 암호 변경	51
8장. IDM의 SSSD를 사용한 인증 문제 해결	53
8.1. SSSD를 사용하여 IDM 사용자 정보를 검색할 때 데이터 흐름	53
8.2. SSSD를 사용하여 AD 사용자 정보를 검색할 때 데이터 흐름	55
8.3. IDM에서 SSSD를 사용하여 사용자로 인증할 때 데이터 흐름	56
8.4. 인증 문제 범위 줄이기	58
8.5. SSSD 로그 파일 및 로깅 수준	61
8.5.1. SSSD 로그 파일 목적	61
8.5.2. SSSD 로깅 수준	62
8.6. SSSD.CONF 파일에서 SSSD에 대한 세부 로깅 활성화	63
8.7. SSSCTL 명령을 사용하여 SSSD에 대한 세부 로깅 활성화	64
8.8. IDM 서버의 인증 문제 해결을 위해 SSSD 서비스에서 디버깅 로그 수집	64
8.9. IDM 클라이언트의 인증 문제를 해결하기 위해 SSSD 서비스에서 디버깅 로그 수집	66
8.10. SSSD 백엔드에서 클라이언트 요청 추적	68
9장. ANSIBLE 플레이북을 사용하여 글로벌 IDM 설정 구성	70
9.1. ANSIBLE 플레이북을 사용하여 IDM 구성 검색	70
9.2. ANSIBLE 플레이북을 사용하여 IDM CA 갱신 서버 구성	72
9.3. ANSIBLE 플레이북을 사용하여 IDM 사용자에게 대한 기본 셀 구성	74
10장. 명령줄을 사용하여 사용자 계정 관리	77
10.1. 사용자 라이프 사이클	77
10.2. 명령줄을 사용하여 사용자 추가	79
10.3. 명령줄을 사용하여 사용자 활성화	81
10.4. 명령줄을 사용하여 사용자 보존	82
10.5. 명령줄을 사용하여 사용자 삭제	82
10.6. 명령줄을 사용하여 사용자 복원	83
11장. IDM 웹 UI를 사용하여 사용자 계정 관리	86
11.1. 사용자 라이프 사이클	86
11.2. 웹 UI에서 사용자 추가	88
11.3. IDM 웹 UI에서 단계 사용자 활성화	91
11.4. 웹 UI에서 사용자 계정 비활성화	92
11.5. 웹 UI에서 사용자 계정 활성화	94
11.6. IDM 웹 UI에서 활성 사용자 보존	95
11.7. IDM 웹 UI에서 사용자 복원	97
11.8. IDM 웹 UI에서 사용자 삭제	98
12장. ANSIBLE 플레이북을 사용하여 사용자 계정 관리	100
12.1. 사용자 라이프 사이클	100
12.2. ANSIBLE 플레이북을 사용하여 IDM 사용자가 있는지 확인	102
12.3. ANSIBLE PLAYBOOK을 사용하여 여러 IDM 사용자가 있는지 확인	104
12.4. ANSIBLE 플레이북을 사용하여 JSON 파일에서 여러 IDM 사용자가 있는지 확인	106
12.5. ANSIBLE 플레이북을 사용하여 사용자가 없는지 확인	108
13장. IDM CLI에서 사용자 그룹 관리	111
13.1. IDM의 다양한 그룹 유형	111
13.2. 직접 및 간접 그룹 구성원	113
13.3. IDM CLI를 사용하여 사용자 그룹 추가	113

13.4. IDM CLI를 사용하여 사용자 그룹 검색	114
13.5. IDM CLI를 사용하여 사용자 그룹 삭제	115
13.6. IDM CLI를 사용하여 사용자 그룹에 멤버 추가	115
13.7. 사용자 개인 그룹없이 사용자 추가	117
13.7.1. 사용자 개인 그룹이 없는 사용자	117
13.7.2. 개인 그룹이 전역적으로 활성화된 경우 사용자 개인 그룹이 없는 사용자 추가	118
13.7.3. 모든 사용자에게 대해 사용자 개인 그룹 전역 비활성화	118
13.7.4. 사용자 개인 그룹이 전역적으로 비활성화된 경우 사용자 추가	119
13.8. IDM CLI를 사용하여 IDM 사용자 그룹에 멤버 관리자로 사용자 또는 그룹 추가	120
13.9. IDM CLI를 사용하여 그룹 멤버 보기	121
13.10. IDM CLI를 사용하여 사용자 그룹에서 멤버 제거	122
13.11. IDM CLI를 사용하여 IDM 사용자 그룹에서 사용자 또는 그룹을 구성원 관리자로 제거	123
14장. IDM 웹 UI에서 사용자 그룹 관리	125
14.1. IDM의 다양한 그룹 유형	125
14.2. 직접 및 간접 그룹 구성원	127
14.3. IDM 웹 UI를 사용하여 사용자 그룹 추가	127
14.4. IDM 웹 UI를 사용하여 사용자 그룹 삭제	128
14.5. IDM 웹 UI를 사용하여 사용자 그룹에 멤버 추가	129
14.6. 웹 UI를 사용하여 IDM 사용자 그룹에 멤버 관리자로 사용자 또는 그룹 추가	130
14.7. IDM 웹 UI를 사용하여 그룹 구성원 보기	133
14.8. IDM 웹 UI를 사용하여 사용자 그룹에서 멤버 제거	134
14.9. WEB UI를 사용하여 IDM 사용자 그룹에서 멤버 관리자로 사용자 또는 그룹 제거	135
15장. ANSIBLE 플레이북을 사용하여 사용자 그룹 관리	137
15.1. IDM의 다양한 그룹 유형	137
15.2. 직접 및 간접 그룹 구성원	139
15.3. ANSIBLE PLAYBOOK을 사용하여 IDM 그룹 및 그룹 구성원이 있는지 확인	140
15.4. ANSIBLE PLAYBOOK을 사용하여 IDM 사용자 그룹에 멤버 관리자가 있는지 확인	142
15.5. ANSIBLE PLAYBOOK을 사용하여 IDM 사용자 그룹에 멤버 관리자가 없는지 확인합니다.	143
16장. IDM CLI를 사용하여 그룹 멤버십 자동화	146
16.1. 자동 그룹 멤버십의 이점	147
16.2. 자동 구성원 규칙	147
16.3. IDM CLI를 사용하여 자동 구성원 규칙 추가	148
16.4. IDM CLI를 사용하여 자동 구성원 규칙에 조건 추가	149
16.5. IDM CLI를 사용하여 기존 자동 구성원 규칙 보기	151
16.6. IDM CLI를 사용하여 자동 구성원 규칙 삭제	152
16.7. IDM CLI를 사용하여 AUTOMEMBER 규칙에서 조건 제거	153
16.8. IDM CLI를 사용하여 기존 항목에 자동 구성원 규칙 적용	154
16.9. IDM CLI를 사용하여 기본 자동 구성원 그룹 구성	155
17장. IDM 웹 UI를 사용하여 그룹 멤버십 자동화	157
17.1. 자동 그룹 멤버십의 이점	158
17.2. 자동 구성원 규칙	158
17.3. IDM 웹 UI를 사용하여 자동 구성원 규칙 추가	159
17.4. IDM 웹 UI를 사용하여 자동 구성원 규칙에 조건 추가	160
17.5. IDM 웹 UI를 사용하여 기존 자동 구성원 규칙 및 조건 보기	162
17.6. IDM 웹 UI를 사용하여 자동 구성원 규칙 삭제	163
17.7. IDM 웹 UI를 사용하여 AUTOMEMBER 규칙에서 조건 제거	164
17.8. IDM 웹 UI를 사용하여 기존 항목에 자동 구성원 규칙 적용	165
17.8.1. 모든 사용자 또는 호스트에 대한 자동 멤버십 다시 빌드	166
17.8.2. 단일 사용자 또는 호스트에 대한 자동 멤버십 다시 빌드	166
17.9. IDM 웹 UI를 사용하여 기본 사용자 그룹 구성	167

17.10. IDM 웹 UI를 사용하여 기본 호스트 그룹 구성	168
18장. ANSIBLE을 사용하여 IDM의 그룹 멤버십 자동화	170
18.1. IDM 관리를 위한 ANSIBLE 제어 노드 준비	171
18.2. ANSIBLE을 사용하여 IDM 사용자 그룹에 대한 자동 관리 규칙이 있는지 확인합니다.	173
18.3. ANSIBLE을 사용하여 지정된 조건이 IDM 사용자 그룹 AUTOMEMBER 규칙에 있는지 확인합니다.	175
18.4. ANSIBLE을 사용하여 IDM 사용자 그룹 AUTOMEMBER 규칙에서 조건이 없는지 확인합니다.	179
18.5. ANSIBLE을 사용하여 IDM 사용자 그룹에 대한 자동 관리 규칙이 없는지 확인합니다.	182
18.6. ANSIBLE을 사용하여 IDM 호스트 그룹 AUTOMEMBER 규칙에 조건이 있는지 확인합니다.	184
19장. CLI를 사용하여 IDM에서 셀프 서비스 규칙 관리	188
19.1. IDM의 셀프 서비스 액세스 제어	188
19.2. CLI를 사용하여 셀프 서비스 규칙 생성	188
19.3. CLI를 사용하여 셀프 서비스 규칙 편집	189
19.4. CLI를 사용하여 셀프 서비스 규칙 삭제	190
20장. IDM 웹 UI를 사용하여 셀프 서비스 규칙 관리	192
20.1. IDM의 셀프 서비스 액세스 제어	192
20.2. IDM 웹 UI를 사용하여 셀프 서비스 규칙 생성	192
20.3. IDM 웹 UI를 사용하여 셀프 서비스 규칙 편집	195
20.4. IDM 웹 UI를 사용하여 셀프 서비스 규칙 삭제	196
21장. ANSIBLE 플레이북을 사용하여 IDM에서 셀프 서비스 규칙 관리	198
21.1. IDM의 셀프 서비스 액세스 제어	198
21.2. ANSIBLE을 사용하여 셀프 서비스 규칙이 있는지 확인합니다.	199
21.3. ANSIBLE을 사용하여 셀프 서비스 규칙이 없는지 확인합니다.	201
21.4. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 특성이 있는지 확인합니다.	203
21.5. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 속성이 없는지 확인합니다.	205
22장. IDM CLI를 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한을 위임	209
22.1. 위임 규칙	209
22.2. IDM CLI를 사용하여 위임 규칙 생성	209
22.3. IDM CLI를 사용하여 기존 위임 규칙 보기	210
22.4. IDM CLI를 사용하여 위임 규칙 수정	211
22.5. IDM CLI를 사용하여 위임 규칙 삭제	212
23장. IDM WEBUI를 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한을 위임	213
23.1. 위임 규칙	213
23.2. IDM WEBUI를 사용하여 위임 규칙 생성	213
23.3. IDM WEBUI를 사용하여 기존 위임 규칙 보기	215
23.4. IDM WEBUI를 사용하여 위임 규칙 수정	216
23.5. IDM WEBUI를 사용하여 위임 규칙 삭제	218
24장. ANSIBLE 플레이북을 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한을 위임	219
24.1. 위임 규칙	219
24.2. IDM용 ANSIBLE 인벤토리 파일 생성	219
24.3. ANSIBLE을 사용하여 위임 규칙이 있는지 확인합니다.	221
24.4. ANSIBLE을 사용하여 위임 규칙이 없는지 확인합니다.	223
24.5. 위임 규칙에 특정 속성이 있는지 확인하려면 ANSIBLE을 사용합니다.	225
24.6. 위임 규칙에 특정 속성이 없는지 확인하려면 ANSIBLE을 사용합니다.	228
25장. CLI를 사용하여 IDM에서 역할 기반 액세스 제어 관리	231
25.1. IDM의 역할 기반 액세스 제어	231
25.1.1. IdM의 권한	231
25.1.2. 기본 관리 권한	233

25.1.3. IdM의 권한	235
25.1.4. IdM의 역할	236
25.1.5. ID 관리에 사전 정의된 역할	236
25.2. CLI에서 IDM 권한 관리	237
25.3. 기존 권한의 명령 옵션	240
25.4. CLI에서 IDM 권한 관리	240
25.5. 기존 권한에 대한 명령 옵션	241
25.6. CLI에서 IDM 역할 관리	242
25.7. 기존 역할에 대한 명령 옵션	243
26장. IDM 웹 UI를 사용하여 역할 기반 액세스 제어 관리	244
26.1. IDM의 역할 기반 액세스 제어	244
26.1.1. IdM의 권한	244
26.1.2. 기본 관리 권한	246
26.1.3. IdM의 권한	248
26.1.4. IdM의 역할	249
26.1.5. ID 관리에 사전 정의된 역할	249
26.2. IDM 웹 UI에서 권한 관리	250
26.3. IDM WEBUI에서 권한 관리	256
26.4. IDM 웹 UI에서 역할 관리	260
27장. ANSIBLE 플레이북을 사용하여 IDM을 관리하기 위한 환경 준비	266
28장. ANSIBLE 플레이북을 사용하여 IDM에서 역할 기반 액세스 제어 관리	269
28.1. IDM의 권한	270
28.2. 기본 관리 권한	271
28.3. IDM의 권한	274
28.4. IDM의 역할	274
28.5. ID 관리에 사전 정의된 역할	275
28.6. ANSIBLE을 사용하여 권한이 있는 IDM RBAC 역할이 있는지 확인합니다.	275
28.7. ANSIBLE을 사용하여 IDM RBAC 역할이 없는지 확인합니다.	278
28.8. ANSIBLE을 사용하여 사용자 그룹이 IDM RBAC 역할에 할당되었는지 확인합니다.	280
28.9. ANSIBLE을 사용하여 특정 사용자가 IDM RBAC 역할에 할당되지 않도록 합니다.	282
28.10. 서비스가 IDM RBAC 역할의 멤버인지 확인하려면 ANSIBLE을 사용합니다.	285
28.11. 호스트가 IDM RBAC 역할의 멤버인지 확인하려면 ANSIBLE을 사용합니다.	287
28.12. ANSIBLE을 사용하여 호스트 그룹이 IDM RBAC 역할의 멤버인지 확인합니다.	289
29장. ANSIBLE 플레이북을 사용하여 RBAC 권한 관리	292
29.1. ANSIBLE을 사용하여 사용자 지정 IDM RBAC 권한이 있는지 확인	292
29.2. ANSIBLE을 사용하여 멤버 권한이 사용자 지정 IDM RBAC 권한에 있는지 확인	294
29.3. ANSIBLE을 사용하여 IDM RBAC 권한에 권한이 포함되지 않도록 합니다.	297
29.4. ANSIBLE을 사용하여 사용자 지정 IDM RBAC 권한 이름 변경	299
29.5. ANSIBLE을 사용하여 IDM RBAC 권한이 없는지 확인합니다.	301
30장. ANSIBLE 플레이북을 사용하여 IDM에서 RBAC 권한 관리	304
30.1. ANSIBLE을 사용하여 RBAC 권한이 있는지 확인합니다.	304
30.2. ANSIBLE을 사용하여 속성이 있는 RBAC 권한이 있는지 확인합니다.	307
30.3. ANSIBLE을 사용하여 RBAC 권한이 없는지 확인합니다.	310
30.4. ANSIBLE을 사용하여 속성이 IDM RBAC 권한의 멤버인지 확인합니다.	311
30.5. ANSIBLE을 사용하여 속성이 IDM RBAC 권한의 멤버가 아닌지 확인합니다.	313
30.6. ANSIBLE을 사용하여 IDM RBAC 권한 이름 변경	316
30.7. 추가 리소스	317
31장. ID 보기를 사용하여 IDM 클라이언트에서 사용자 속성 값을 재정의	319
31.1. ID 보기	319

31.2. SSSD 성능에 대한 ID 뷰의 잠재적 영향	321
31.3. ID 보기가 재정의할 수 있는 속성	321
31.4. ID 보기 명령에 대한 도움말 가져오기	322
31.5. ID 보기를 사용하여 특정 호스트의 IDM 사용자 로그인 이름 덮어쓰기	323
31.6. IDM ID 보기 수정	326
31.7. IDM 클라이언트에서 IDM 사용자 홈 디렉터리 재정의의 위해 ID 보기 추가	328
31.8. IDM 호스트 그룹에 ID 보기 적용	331
32장. 수동으로 ID 범위 조정	335
32.1. ID 범위	335
32.2. 자동 ID 범위 할당	335
32.3. 서버 설치 중 IDM ID 범위 수동 할당	336
32.4. 새 IDM ID 범위 추가	337
32.5. 신뢰할 수 있는 AD를 제거한 후 ID 범위 제거	339
32.6. 현재 할당된 DNA ID 범위 표시	340
32.7. 자동 DNA ID 범위 확장	340
32.8. 수동 DNA ID 범위 조정	342
32.9. 수동으로 DNA ID 범위 조정	343
33장. ANSIBLE을 사용하여 IDM의 복제 토폴로지 관리	345
33.1. ANSIBLE을 사용하여 IDM에 복제 계약이 있는지 확인	345
33.2. ANSIBLE을 사용하여 여러 IDM 복제본 간에 복제 계약이 있는지 확인	348
33.3. ANSIBLE을 사용하여 두 복제본 간에 복제 계약이 있는지 확인	350
33.4. ANSIBLE을 사용하여 토폴로지 접미사가 IDM에 있는지 확인합니다.	353
33.5. ANSIBLE을 사용하여 IDM 복제본 다시 초기화	355
33.6. ANSIBLE을 사용하여 IDM에 복제 계약이 없음	358
33.7. 추가 리소스	361
34장. 사용자의 외부 프로비저닝을 위한 IDM 구성	362
34.1. 단계 사용자 계정의 자동 활성화를 위한 IDM 계정 준비	363
34.2. IDM 스테이징 사용자 계정의 자동 활성화 구성	365
34.3. LDIF 파일에 정의된 IDM 단계 사용자 추가	368
34.4. LDAPMODIFY를 사용하여 CLI에서 직접 IDM 단계 사용자 추가	370
35장. LDAPMODIFY를 사용하여 IDM 사용자 외부 관리	373
35.1. 외부에서 IDM 사용자 계정을 관리하는 템플릿	373
35.2. 외부에서 IDM 그룹 계정을 관리하는 템플릿	376
35.3. LDAPMODIFY를 사용하여 IDM 사용자 보존	378
36장. IDM CLI에서 호스트 관리	381
36.1. IDM의 호스트	381
36.2. 호스트 등록	382
36.2.1. 호스트 등록에 필요한 사용자 권한	383
IdM LDAP에서 수동으로 호스트 항목을 생성하는 데 필요한 사용자 권한	383
클라이언트를 IdM 도메인에 가입하는 사용자 권한	383
36.2.2. IdM 호스트 및 사용자의 등록 및 인증: 비교	384
36.2.3. IdM 호스트의 대체 인증 옵션	385
36.3. 호스트 작업	386
36.4. IDM LDAP의 호스트 항목	388
36.4.1. 호스트 항목 구성 속성	388
36.5. IDM CLI에서 IDM 호스트 항목 추가	390
36.6. IDM CLI에서 호스트 항목 삭제	390
36.7. ID 관리 클라이언트 재등록	391
36.7.1. IdM의 클라이언트 등록	391

36.7.1.1. 클라이언트를 다시 등록하는 동안 발생하는 사항	391
36.7.2. 사용자 인증 정보를 사용하여 클라이언트 재등록: 대화형 등록	392
36.7.3. 클라이언트 키탭을 사용하여 클라이언트를 다시 등록합니다. 비대화형 재등록	392
36.7.4. 설치 후 Identity Management 클라이언트 테스트	393
36.8. IDENTITY MANAGEMENT 클라이언트 시스템 이름 변경	393
36.8.1. 변경 작업을 위해 IdM 클라이언트 준비	394
36.8.2. ID 관리 클라이언트 설치 제거	395
36.8.3. 호스트 시스템 이름 변경	396
36.8.4. ID 관리 클라이언트 재설치	396
36.8.5. 서비스 다시 추가, 인증서 재생, 호스트 그룹 다시 추가	396
36.9. 호스트 항목 비활성화 및 재활성화	397
36.9.1. 호스트 비활성화	397
36.9.2. 호스트 재활성화	398
37장. IDM 웹 UI에서 호스트 항목 추가	399
37.1. IDM의 호스트	399
37.2. 호스트 등록	400
37.2.1. 호스트 등록에 필요한 사용자 권한	400
IdM LDAP에서 수동으로 호스트 항목을 생성하는 데 필요한 사용자 권한	401
클라이언트를 IdM 도메인에 가입하는 사용자 권한	401
37.2.2. IdM 호스트 및 사용자의 등록 및 인증: 비교	401
37.2.3. IdM 호스트의 대체 인증 옵션	403
37.3. IDM LDAP의 호스트 항목	403
37.3.1. 호스트 항목 구성 속성	404
37.4. 웹 UI에서 호스트 항목 추가	405
38장. ANSIBLE 플레이북을 사용하여 호스트 관리	408
38.1. IDM의 호스트	408
38.2. 호스트 등록	409
38.2.1. 호스트 등록에 필요한 사용자 권한	410
IdM LDAP에서 수동으로 호스트 항목을 생성하는 데 필요한 사용자 권한	410
클라이언트를 IdM 도메인에 가입하는 사용자 권한	410
38.2.2. IdM 호스트 및 사용자의 등록 및 인증: 비교	411
38.2.3. IdM 호스트의 대체 인증 옵션	412
38.3. ANSIBLE PLAYBOOK을 사용하여 FQDN과 함께 IDM 호스트 항목이 있는지 확인	413
38.4. ANSIBLE PLAYBOOK을 사용하여 DNS 정보가 포함된 IDM 호스트 항목이 있는지 확인	415
38.5. ANSIBLE PLAYBOOK을 사용하여 임의의 암호와 함께 여러 IDM 호스트 항목이 있는지 확인	417
38.6. ANSIBLE PLAYBOOK을 사용하여 여러 IP 주소가 있는 IDM 호스트 항목이 있는지 확인	419
38.7. ANSIBLE 플레이북을 사용하여 IDM 호스트 항목이 없는지 확인	421
39장. IDM CLI를 사용하여 호스트 그룹 관리	424
39.1. IDM의 호스트 그룹	424
39.2. CLI를 사용하여 IDM 호스트 그룹 보기	425
39.3. CLI를 사용하여 IDM 호스트 그룹 생성	426
39.4. CLI를 사용하여 IDM 호스트 그룹 삭제	426
39.5. CLI를 사용하여 IDM 호스트 그룹 멤버 추가	427
39.6. CLI를 사용하여 IDM 호스트 그룹 멤버 제거	428
39.7. CLI를 사용하여 IDM 호스트 그룹 멤버 관리자 추가	430
39.8. CLI를 사용하여 IDM 호스트 그룹 멤버 관리자 제거	431
40장. IDM 웹 UI를 사용하여 호스트 그룹 관리	434
40.1. IDM의 호스트 그룹	434
40.2. IDM 웹 UI에서 호스트 그룹 보기	435
40.3. IDM 웹 UI에서 호스트 그룹 생성	437

40.4. IDM 웹 UI에서 호스트 그룹 삭제	437
40.5. IDM 웹 UI에서 호스트 그룹 구성원 추가	438
40.6. IDM 웹 UI에서 호스트 그룹 구성원 제거	439
40.7. 웹 UI를 사용하여 IDM 호스트 그룹 멤버 관리자 추가	440
40.8. 웹 UI를 사용하여 IDM 호스트 그룹 멤버 관리자 제거	443
41장. ANSIBLE 플레이북을 사용하여 호스트 그룹 관리	445
41.1. IDM의 호스트 그룹	445
41.2. ANSIBLE PLAYBOOK을 사용하여 IDM 호스트 그룹이 있는지 확인	446
41.3. ANSIBLE PLAYBOOK을 사용하여 IDM 호스트 그룹에 호스트가 있는지 확인	448
41.4. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹 중첩	450
41.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 멤버 관리자가 있는지 확인	451
41.6. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 호스트가 없는지 확인합니다.	453
41.7. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 중첩된 호스트 그룹이 없는지 확인합니다.	455
41.8. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹이 없는지 확인	457
41.9. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 멤버 관리자가 없는지 확인합니다.	459
42장. KERBEROS 티켓 정책 관리	461
42.1. IDM KDC의 역할	461
42.2. IDM KERBEROS 티켓 정책 유형	463
42.3. KERBEROS 인증 표시기	464
42.3.1. 인증 표시기 및 IdM 서비스	464
42.4. IDM 서비스에 대한 인증 표시기 시행	465
42.4.1. IdM 서비스 항목 및 Kerberos 키 탭 생성	465
42.4.2. IdM 서비스와 인증 표시기 연결	466
42.4.3. IdM 서비스의 Kerberos 서비스 티켓 검색	468
42.4.4. 추가 리소스	469
42.5. 글로벌 티켓 라이프사이클 정책 구성	470
42.6. 인증 표시기별로 글로벌 티켓 정책 구성	471
42.7. 사용자의 기본 티켓 정책 구성	472
42.8. 사용자에게 대한 개별 인증 표시기 티켓 정책 구성	473
42.9. KRBTPOLICY-MOD 명령의 인증 표시 옵션	474
43장. IDM에서 KDC 프로ksi 사용	476
43.1. KKDCP를 사용하도록 IDM 클라이언트 구성	476
43.2. IDM 서버에서 KKDCP가 활성화되었는지 확인	477
43.3. IDM 서버에서 KKDCP 비활성화	478
43.4. IDM 서버에서 KKDCP 재활성화	478
43.5. 추가 리소스	479
44장. IDM 암호 정책 정의	480
44.1. 암호 정책이란 무엇입니까	480
44.2. IDM의 암호 정책	480
44.3. ANSIBLE 플레이북을 사용하여 IDM에 암호 정책이 있는지 확인	482
44.4. IDM의 추가 암호 정책 옵션	484
44.5. IDM 그룹에 추가 암호 정책 옵션 적용	485
45장. 암호 알림 만료 관리	489
45.1. 암호 만료 알림 도구란 무엇입니까?	489
45.2. 만료 암호 알림 도구 설치	490
45.3. EPN 도구를 실행하여 암호가 만료되는 사용자에게 이메일 보내기	490
45.4. 암호가 만료되는 모든 사용자에게 IPA-EPN.TIMER를 활성화	493
45.5. 만료 암호 알림 이메일 템플릿 수정	493
46장. IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여	496

46.1. IDM 클라이언트에서 SUDO 액세스	496
46.2. CLI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여	496
46.3. IDM 웹 UI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여	499
46.4. IDM 클라이언트에서 명령을 서비스 계정으로 실행하는 CLI에서 SUDO 규칙 생성	503
46.5. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 IDM 웹UI에 SUDO 규칙 생성	506
46.6. IDM 클라이언트에서 SUDO에 대해 GSSAPI 인증 활성화	514
46.7. GSSAPI 인증 활성화 및 IDM 클라이언트에서 SUDO에 대한 KERBEROS 인증 표시기 적용	517
46.8. PAM 서비스에 대한 GSSAPI 인증을 제어하는 SSSD 옵션	520
46.9. SUDO에 대한 GSSAPI 인증 문제 해결	522
46.10. ANSIBLE 플레이북을 사용하여 IDM 클라이언트에서 IDM 사용자에게 SUDO 액세스 권한 확인	524
47장. ANSIBLE 플레이북을 사용하여 IDM에 호스트 기반 액세스 제어 규칙이 있는지 확인	528
47.1. IDM의 호스트 기반 액세스 제어 규칙	528
47.2. ANSIBLE 플레이북을 사용하여 IDM에 HBAC 규칙이 있는지 확인	528
48장. 복제 토폴로지 관리	531
48.1. 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트 설명	531
복제 계약	531
토폴로지 접미사	532
토폴로지 세그먼트	533
48.2. 토폴로지 그래프를 사용하여 복제 토폴로지 관리	535
토폴로지 그래프 액세스	535
토폴로지 그래프 해석	535
토폴로지 보기 사용자 정의	536
48.3. 웹 UI를 사용하여 두 서버 간 복제 설정	537
48.4. 웹 UI를 사용하여 두 서버 간 복제 중지	539
48.5. CLI를 사용하여 두 서버 간 복제 설정	540
48.6. CLI를 사용하여 두 서버 간 복제 중지	541
48.7. 웹 UI를 사용하여 토폴로지에서 서버 제거	543
48.8. CLI를 사용하여 토폴로지에서 서버 제거	544
48.9. 웹 UI를 사용하여 IDM 서버에서 서버 역할 보기	545
48.10. CLI를 사용하여 IDM 서버에서 서버 역할 보기	546
48.11. CA 갱신 서버 및 CRL 게시 서버로 복제본 승격	547
48.12. 숨겨진 복제본 데모 또는 승격	547
49장. IDENTITY MANAGEMENT의 공개 키 인증서	549
49.1. IDM의 인증 기관	549
49.2. 인증서와 KERBEROS 비교	550
49.3. 인증서를 사용하여 IDM의 사용자를 인증하는 장단점	551
50장. 통합 IDM CA를 사용한 사용자, 호스트 및 서비스에 대한 인증서 관리	553
50.1. IDM 웹 UI를 사용하여 사용자, 호스트 또는 서비스에 대한 새 인증서 요청	554
50.2. CERTUTIL을 사용하여 IDM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청	556
50.3. OPENSSL을 사용하여 IDM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청	557
50.4. 추가 리소스	559
51장. IDM에서 작동하도록 인증서 형식 변환	560
51.1. IDM의 인증서 형식 및 인코딩	560
시스템 설정	560
인증서 인코딩	560
사용자 인증	561
유용한 인증서 명령	562
51.2. IDM 사용자 계정으로 로드할 외부 인증서를 변환	562
51.2.1. Idm CLI에서 외부 인증서를 변환하여 Idm 사용자 계정으로 로드	562

51.2.2. IdM 사용자 계정으로 로드하기 위해 IdM 웹 UI에서 외부 인증서를 변환합니다.	564
51.3. 브라우저에 인증서 로드 준비	565
51.3.1. NSS 데이터베이스에서 PKCS #12 파일로 인증서 및 개인 키 내보내기	565
51.3.2. 인증서와 개인 키 PEM 파일을 PKCS #12 파일에 결합	566
51.4. IDM의 인증서 관련 명령 및 형식	566
52장. IDENTITY MANAGEMENT에서 인증서 프로필 생성 및 관리	568
52.1. 인증서 프로필이란 무엇입니까?	568
52.2. 인증서 프로필 생성	569
52.3. CA 액세스 제어 목록이란 무엇입니까?	571
52.4. 인증서 프로필에 대한 액세스를 제어하는 CA ACL 정의	572
52.5. 인증서 프로필 및 CA ACL을 사용하여 인증서 발행	574
52.6. 인증서 프로필 수정	576
52.7. 인증서 프로필 구성 매개변수	577
53장. IDM에서 인증서의 유효성 관리	581
IdM CA에서 발급한 기존 인증서의 유효성 관리	581
IdM CA에서 발급한 향후 인증서의 유효성 관리	582
53.1. 인증서 만료일 보기	582
53.1.1. IdM WebUI에서 인증서 만료일 보기	582
53.1.2. CLI에서 인증서 만료일 보기	583
53.2. IDM CA가 통합된 인증서 취소	583
53.2.1. 인증서 해지 이유	583
53.2.2. IdM WebUI를 사용하여 통합된 IdM CA로 인증서 해지	584
53.2.3. IdM CLI를 사용하여 통합된 IdM CA로 인증서 해지	585
53.3. 통합된 IDM CA를 사용하여 인증서 복원	586
53.3.1. IdM WebUI를 사용하여 통합된 IdM CA로 인증서 복원	586
53.3.2. IdM CLI를 사용하여 통합된 IdM CA로 인증서 복원	587
54장. 스마트 카드 인증을 위한 ID 관리 구성	588
54.1. 스마트 카드 인증을 위해 IDM 서버 구성	589
54.2. 스마트 카드 인증을 위해 IDM 클라이언트 구성	592
54.3. IDM 웹 UI의 사용자 항목에 인증서 추가	594
54.4. IDM CLI의 사용자 항목에 인증서 추가	596
54.5. 스마트 카드 관리 및 사용을 위한 도구 설치	597
54.6. 스마트 카드에 인증서 저장	598
54.7. 스마트 카드로 IDM에 로그인	601
54.8. 스마트 카드 인증을 사용하여 GDM 액세스 구성	603
54.9. 스마트 카드 인증을 사용하여 SU 액세스 구성	604
55장. IDM의 스마트 카드 인증을 위해 ADCS에서 발급한 인증서 구성	605
55.1. 스마트 카드 인증	606
55.2. 신뢰 구성 및 인증서 사용에 필요한 WINDOWS SERVER 설정	607
55.3. SFTP를 사용하여 ACTIVE DIRECTORY에서 인증서 복사	607
55.4. ADCS 인증서를 사용하여 스마트 카드 인증을 위해 IDM 서버 및 클라이언트 구성	608
55.5. PFX 파일 변환	611
55.6. 스마트 카드 관리 및 사용을 위한 도구 설치	612
55.7. 스마트 카드에 인증서 저장	612
55.8. SSSD.CONF에서 시간 제한 설정	615
55.9. 스마트 카드 인증을 위한 인증서 매핑 규칙 생성	616
56장. ID 관리에서 인증서 매핑 규칙 구성	618
56.1. 스마트 카드 인증을 구성하기 위한 인증서 매핑 규칙	618
56.1.1. Active Directory 도메인을 사용하여 신뢰에 대한 인증서 매핑 규칙	619

56.1.2. IdM에서 ID 매핑 규칙의 구성 요소	619
56.1.3. 일치하는 규칙에 사용하기 위해 인증서에서 발급자 가져오기	621
56.1.4. 추가 리소스	622
56.2. IDM에 저장된 사용자에게 대한 인증서 매핑 구성	622
56.2.1. IdM에서 인증서 매핑 규칙 추가	622
56.2.1.1. IdM 웹 UI에서 인증서 매핑 규칙 추가	623
56.2.1.2. IdM CLI에서 인증서 매핑 규칙 추가	624
56.2.2. IdM의 사용자 항목에 인증서 매핑 데이터 추가	625
56.2.2.1. IdM 웹 UI의 사용자 항목에 인증서 매핑 데이터 추가	625
56.2.2.2. IdM CLI의 사용자 항목에 인증서 매핑 데이터 추가	628
56.3. AD 사용자 항목이 전체 인증서가 포함된 사용자에게 대한 인증서 매핑 구성	629
56.3.1. AD 항목이 전체 인증서가 포함된 사용자를 위한 인증서 매핑 규칙 추가	629
56.3.1.1. IdM 웹 UI에서 인증서 매핑 규칙 추가	629
56.3.1.2. IdM CLI에서 인증서 매핑 규칙 추가	631
56.4. AD가 사용자 계정에 매핑하도록 구성된 경우 인증서 매핑 구성	631
56.4.1. 신뢰할 수 있는 AD 도메인이 사용자 인증서를 매핑하도록 구성된 경우 인증서 매핑 규칙 추가	632
56.4.1.1. IdM 웹 UI에서 인증서 매핑 규칙 추가	632
56.4.1.2. IdM CLI에서 인증서 매핑 규칙 추가	633
56.4.2. AD 측에서 인증서 매핑 데이터 확인	634
56.5. AD 사용자 항목에 인증서 또는 매핑 데이터가 포함되어 있지 않은 경우 인증서 매핑 구성	635
56.5.1. AD 사용자 항목에 인증서 또는 매핑 데이터가 없는 경우 인증서 매핑 규칙 추가	635
56.5.1.1. IdM 웹 UI에서 인증서 매핑 규칙 추가	636
56.5.1.2. IdM CLI에서 인증서 매핑 규칙 추가	637
56.5.2. AD의 사용자 항목에 인증서 또는 매핑 데이터가 포함되어 있지 않은 경우 AD 사용자 ID에 인증서를 추가합니다.	638
56.5.2.1. IdM 웹 UI에서 AD 사용자 ID 재정의에 인증서 추가	638
56.5.2.2. IdM CLI에서 AD 사용자 ID 재정의에 인증서 추가	640
56.6. 여러 ID 매핑 규칙을 하나로 결합	641
57장. IDM 클라이언트의 데스크탑에 저장된 인증서를 사용하여 인증 구성	643
57.1. 웹 UI에서 인증서 인증을 위한 ID 관리 서버 구성	643
57.2. 새 사용자 인증서 요청 및 클라이언트로 내보내기	644
57.3. 인증서와 사용자가 함께 연결되었는지 확인	646
57.4. 인증서 인증을 활성화하도록 브라우저 구성	647
57.5. ID 관리 사용자로 인증서를 사용하여 ID 관리 웹 UI에 인증	651
57.6. 인증서를 사용하여 CLI에 대한 인증을 활성화하도록 IDM 클라이언트 구성	652
58장. IDM CA 갱신 서버 사용	654
58.1. IDM CA 갱신 서버 설명	654
CA 갱신 서버의 역할	654
CA 복제본에서 certmonger 서비스의 역할	655
IdM CA 갱신 서버의 올바른 기능	655
58.2. IDM CA 갱신 서버 변경 및 재설정	656
58.3. IDM의 자체 서명된 CA로 외부에서 전환	658
58.4. 외부 서명 인증서를 사용하여 IDM CA 갱신 서버 갱신	659
59장. IDM이 오프라인 상태일 때 만료된 시스템 인증서 갱신	663
59.1. CA 갱신 서버에서 만료된 시스템 인증서 갱신	663
59.2. 갱신 후 IDM 도메인에서 다른 IDM 서버 확인	664
59.3. 웹 서버 및 LDAP 서버 인증서 교체	666
60장. IDM CA 서버에서 CRL 생성	669
60.1. IDM 서버에서 CRL 생성 중지	669
60.2. IDM 복제본 서버에서 CRL 생성 시작	670

61장. CA 갱신 서버 및 CRL 게시자 역할을 수행하는 서버 삭제	672
62장. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기	676
62.1. CERTMONGER 개요	676
certmonger 의 기능	676
인증서 certmonger 트랙 유형	676
certmonger 구성 요소	677
62.2. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기	678
62.3. 서비스 인증서를 요청하는 CERTMONGER의 통신 흐름	680
62.4. CERTMONGER에서 추적한 인증서 요청 세부 정보 보기	684
62.5. 인증서 추적 시작 및 중지	686
62.6. 인증서 수동 갱신	687
62.7. CERTMONGER에서 CA 복제본에서 IDM 인증서 추적을 다시 시작	688
63장. RHEL 시스템 역할을 사용하여 인증서 요청	690
63.1. 인증서 시스템 역할	690
63.2. 인증서 시스템 역할을 사용하여 자체 서명된 새 인증서 요청	691
63.3. 인증서 시스템 역할을 사용하여 IDM CA에서 새 인증서 요청	693
63.4. 인증서 시스템 역할을 사용하여 인증서 발급 전후에 실행할 명령 지정	695
64장. 인증서의 하위 집합만 신뢰하도록 애플리케이션 제한	698
64.1. 경량 하위 CA 관리	699
64.1.1. IdM 웹UI에서 하위 CA 생성	700
64.1.2. IdM 웹UI에서 하위 CA 삭제	701
64.1.3. IdM CLI에서 하위 CA 생성	703
64.1.4. IdM CLI에서 하위 CA 비활성화	705
64.1.5. IdM CLI에서 하위 CA 삭제	706
64.2. IDM WEBUI에서 하위 CA 인증서 다운로드	708
64.3. 웹 서버 및 클라이언트 인증을 위한 CA ACL 생성	708
64.3.1. IdM CLI에서 CA ACL 보기	708
64.3.2. webserver-ca에서 발급한 인증서를 사용하여 웹 클라이언트에 인증하기 위한 CA ACL 생성	709
64.3.3. webclient-ca에서 발급한 인증서를 사용하여 사용자 웹 브라우저에 대한 CA ACL 생성	711
64.4. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기	713
64.5. 서비스 인증서를 요청하는 CERTMONGER의 통신 흐름	715
64.6. 단일 인스턴스 APACHE HTTP 서버 설정	720
64.7. APACHE HTTP 서버에 TLS 암호화 추가	721
64.8. APACHE HTTP 서버에서 지원되는 TLS 프로토콜 버전 설정	724
64.9. APACHE HTTP 서버에서 지원되는 암호 설정	726
64.10. TLS 클라이언트 인증서 인증 구성	727
64.11. 새 사용자 인증서 요청 및 클라이언트로 내보내기	729
64.12. 인증서 인증을 활성화하도록 브라우저 구성	731
65장. 관련 인증서의 특정 그룹 무효화	734
65.1. IDM CLI에서 CA ACL 비활성화	734
65.2. IDM 하위 CA 비활성화	736
66장. IDM의 자격 증명 모음	737
66.1. 자격 증명 모음 및 이점	737
66.2. VAULT 소유자, 멤버 및 관리자	738
66.3. 표준, 대칭 및 비대칭 자격 증명 모음	740
66.4. 사용자, 서비스 및 공유 자격 증명 모음	740
66.5. VAULT 컨테이너	741
66.6. 기본 IDM 자격 증명 모음 명령	741
66.7. IDM에 키 복구 권한 설치	742

67장. IDM 사용자 자격 증명 모음 사용: 시크릿 저장 및 검색	744
67.1. 사용자 자격 증명 모음에 시크릿 저장	744
67.2. 사용자 자격 증명 모음에서 시크릿 검색	745
68장. ANSIBLE을 사용하여 IDM 사용자 자격 증명 모음 관리: 시크릿 저장 및 검색	747
68.1. ANSIBLE을 사용하여 IDM에 표준 사용자 자격 증명 모음이 있는지 확인	747
68.2. ANSIBLE을 사용하여 IDM의 표준 사용자 자격 증명 모음에 비밀 보관	749
68.3. ANSIBLE을 사용하여 IDM의 표준 사용자 자격 증명 모음에서 시크릿 검색	751
69장. IDM 서비스 시크릿 관리: 시크릿 저장 및 검색	755
69.1. 비대칭 자격 증명 모음에 IDM 서비스 시크릿 저장	756
69.2. IDM 서비스 인스턴스의 서비스 시크릿 검색	758
69.3. 손상된 경우 IDM 서비스 자격 증명 모음 시크릿 변경	758
70장. ANSIBLE을 사용하여 IDM 서비스 자격 증명 모음 관리: 시크릿 저장 및 검색	760
70.1. ANSIBLE을 사용하여 IDM에 비대칭 서비스 자격 증명 모음이 있는지 확인	761
70.2. ANSIBLE을 사용하여 멤버 서비스 추가	764
70.3. ANSIBLE을 사용하여 비대칭 자격 증명 모음에 IDM 서비스 시크릿 저장	766
70.4. ANSIBLE을 사용하여 IDM 서비스의 서비스 시크릿 검색	768
70.5. ANSIBLE을 사용하여 손상된 경우 IDM 서비스 자격 증명 모음 시크릿 변경	771
71장. ANSIBLE을 사용하여 IDM에 서비스가 있는지 여부 확인	777
71.1. ANSIBLE 플레이북을 사용하여 IDM에 HTTP 서비스가 있는지 확인	778
71.2. ANSIBLE 플레이북을 사용하여 비IDM 클라이언트에서 IDM에 HTTP 서비스가 있는지 확인	780
71.3. ANSIBLE PLAYBOOK을 사용하여 DNS 없이 IDM 클라이언트에 HTTP 서비스가 있는지 확인	782
71.4. ANSIBLE PLAYBOOK을 사용하여 IDM 서비스 항목에 외부 서명된 인증서가 있는지 확인	784
71.5. ANSIBLE 플레이북을 사용하여 IDM 사용자, 그룹, 호스트 또는 호스트 그룹을 사용하여 서비스의 키탭을 생성할 수 있습니다.	787
71.6. ANSIBLE 플레이북을 사용하여 IDM 사용자, 그룹, 호스트 또는 호스트 그룹을 통해 서비스의 키탭 검색 가능	790
71.7. ANSIBLE PLAYBOOK을 사용하여 서비스의 KERBEROS 기본 별칭이 있는지 확인	793
71.8. ANSIBLE 플레이북을 사용하여 IDM에 HTTP 서비스가 없는지 확인	795
72장. AD 사용자가 IDM을 관리하도록 활성화	798
72.1. AD 사용자의 ID 덮어쓰기	798
72.2. ID 덮어쓰기를 사용하여 AD 사용자가 IDM 관리 활성화	798
72.3. IDM CLI를 AD 사용자로 관리	799
73장. 짧은 AD 사용자 이름 확인 순서 구성	801
73.1. 도메인 확인 순서가 작동하는 방식	801
73.2. IDM 서버에서 글로벌 도메인 확인 순서 설정	802
73.3. IDM 서버에서 ID 보기에 대한 도메인 확인 순서 설정	803
73.4. IDM 클라이언트의 SSSD에서 도메인 확인 순서 설정	805
74장. IDM에서 AD 사용자 주체 이름을 사용한 인증 활성화	807
74.1. IDM에서 신뢰하는 ADFOREST의 사용자 주요 이름	807
74.2. AD UPN이 IDM에서 최신 상태인지 확인	808
74.3. AD UPN 인증 문제의 문제 해결 데이터 수집	809
75장. IDM에서 정식화된 DNS 호스트 이름 사용	811
75.1. 호스트 주체에 별칭 추가	811
75.2. 클라이언트에서 서비스 주체에서 호스트 이름의 정식화 활성화	811
75.3. DNS 호스트 이름 정식화가 활성화된 호스트 이름을 사용하는 옵션	812
76장. ANSIBLE 플레이북을 사용하여 IDM에서 글로벌 DNS 구성 관리	813
76.1. IDM을 통해 NETWORKMANAGER에서 /ETC/RESOLV.CONF의 글로벌 전달자를 제거하지 않는 방법	814

76.2. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인	815
76.3. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 없는지 확인	817
76.4. IDM의 DNS 전달 정책	819
76.5. ANSIBLE 플레이북을 사용하여 IDM DNS 글로벌 구성에 전달 첫 번째 정책이 설정되었는지 확인합니다.	820
76.6. ANSIBLE 플레이북을 사용하여 글로벌 전달자가 IDM DNS에서 비활성화되었는지 확인합니다.	822
76.7. ANSIBLE 플레이북을 사용하여 IDM DNS에서 정방향 및 역방향 조회 영역의 동기화가 비활성화되었는지 확인합니다.	824
77장. IDM에서 DNS 영역 관리	827
77.1. 지원되는 DNS 영역 유형	828
77.2. IDM 웹 UI에서 기본 DNS 영역 추가	829
77.3. IDM CLI에서 기본 DNS 영역 추가	830
77.4. IDM 웹 UI에서 기본 DNS 영역 제거	831
77.5. IDM CLI에서 기본 DNS 영역 제거	832
77.6. DNS 구성 우선순위	832
77.7. 기본 IDM DNS 영역의 구성 특성	833
77.8. IDM 웹 UI에서 기본 DNS 영역 구성 편집	835
77.9. IDM CLI에서 기본 DNS 영역 구성 편집	837
77.10. IDM의 영역 전송	838
77.11. IDM 웹 UI에서 영역 전송 활성화	839
77.12. IDM CLI에서 영역 전송 활성화	839
77.13. 추가 리소스	840
78장. ANSIBLE 플레이북을 사용하여 IDM DNS 영역 관리	841
78.1. 지원되는 DNS 영역 유형	841
78.2. 기본 IDM DNS 영역의 구성 특성	843
78.3. ANSIBLE을 사용하여 IDM DNS에서 기본 영역 생성	845
78.4. ANSIBLE 플레이북을 사용하여 여러 변수와 함께 IDM에 기본 DNS 영역이 있는지 확인합니다.	847
78.5. IP 주소가 제공될 때 역방향 DNS 조회에 영역이 있는지 확인하는 데 ANSIBLE 플레이북을 사용합니다.	850
79장. IDM에서 DNS 위치 관리	853
79.1. DNS 기반 서비스 검색	853
79.2. DNS 위치에 대한 배포 고려 사항	855
79.3. DNS TIME TO LIVE (TTL)	855
79.4. IDM 웹 UI를 사용하여 DNS 위치 생성	856
79.5. IDM CLI를 사용하여 DNS 위치 생성	857
79.6. IDM 웹 UI를 사용하여 DNS 위치에 IDM 서버 할당	857
79.7. IDM CLI를 사용하여 DNS 위치에 IDM 서버 할당	859
79.8. 동일한 위치에서 IDM 서버를 사용하도록 IDM 클라이언트 구성	861
79.9. 추가 리소스	862
80장. ANSIBLE을 사용하여 IDM에서 DNS 위치 관리	863
80.1. DNS 기반 서비스 검색	863
80.2. DNS 위치에 대한 배포 고려 사항	865
80.3. DNS TIME TO LIVE (TTL)	865
80.4. ANSIBLE을 사용하여 IDM 위치가 있는지 확인	865
80.5. ANSIBLE을 사용하여 IDM 위치가 없는지 확인	867
80.6. 추가 리소스	869
81장. IDM에서 DNS 전달 관리	870
81.1. IDM DNS 서버의 두 가지 역할	871
81.2. IDM의 DNS 전달 정책	871
81.3. IDM 웹 UI에서 글로벌 전달자 추가	872
81.4. CLI에서 글로벌 전달자 추가	875

81.5. IDM 웹 UI에서 DNS 전달 영역 추가	876
81.6. CLI에서 DNS 전달 영역 추가	880
81.7. ANSIBLE을 사용하여 IDM에 DNS GLOBAL FORWARDER 설정	881
81.8. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인	883
81.9. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 없는지 확인	885
81.10. ANSIBLE을 사용하여 DNS GLOBAL FORWARDERS가 IDM에서 비활성화되었는지 확인	887
81.11. ANSIBLE을 사용하여 IDM에 DNS 전달 영역이 있는지 확인	888
81.12. ANSIBLE을 사용하여 DNS FORWARD ZONE에 IDM에 여러 개의 전달자가 있는지 확인	891
81.13. ANSIBLE을 사용하여 IDM에서 DNS 전달 영역이 비활성화되어 있는지 확인	893
81.14. ANSIBLE을 사용하여 IDM에 DNS 전달 영역이 없는지 확인	896
82장. IDM에서 DNS 레코드 관리	899
82.1. IDM의 DNS 레코드	899
82.2. IDM 웹 UI에서 DNS 리소스 레코드 추가	901
82.3. IDM CLI에서 DNS 리소스 레코드 추가	902
82.4. 일반적인 IPA DNSRECORD-* 옵션	903
82.5. IDM 웹 UI에서 DNS 레코드 삭제	906
82.6. IDM 웹 UI에서 전체 DNS 레코드 삭제	908
82.7. IDM CLI에서 DNS 레코드 삭제	909
82.8. 추가 리소스	909
83장. ANSIBLE을 사용하여 IDM에서 DNS 레코드 관리	910
83.1. IDM의 DNS 레코드	910
83.2. 일반적인 IPA DNSRECORD-* 옵션	911
83.3. ANSIBLE을 사용하여 IDM에 A 및 AAAA DNS 레코드가 있는지 확인	914
83.4. ANSIBLE을 사용하여 IDM에 A 및 PTR DNS 레코드가 있는지 확인	916
83.5. ANSIBLE을 사용하여 IDM에 여러 DNS 레코드가 있는지 확인	919
83.6. ANSIBLE을 사용하여 IDM에 여러 CNAME 레코드가 있는지 확인	921
83.7. ANSIBLE을 사용하여 IDM에 SRV 레코드가 있는지 확인	924
84장. ANSIBLE을 사용하여 IDM 서버 관리	927
84.1. ANSIBLE을 사용하여 IDM 서버가 있는지 확인	927
84.2. ANSIBLE을 사용하여 IDM 토폴로지에 IDM 서버가 없는지 확인	929
84.3. 마지막 IDM 서버 역할을 호스팅해도 IDM 서버가 없는지 확인	932
84.4. IDM 서버가 없지만 다른 IDM 서버와 반드시 연결이 끊어지지 않도록 확인	934
84.5. ANSIBLE 플레이북을 사용하여 기존 IDM 서버가 숨겨진지 확인	937
84.6. ANSIBLE 플레이북을 사용하여 기존 IDM 서버를 볼 수 있는지 확인	939
84.7. 기존 IDM 서버에 IDM DNS 위치가 할당되었는지 확인	941
84.8. 기존 IDM 서버에 IDM DNS 위치가 할당되지 않았는지 확인	944
85장. IDM 상태 점검 정보 수집	947
85.1. IDM의 상태 점검	947
85.1.1. 모듈은 독립적입니다	947
85.1.2. 두 가지 출력 형식	948
85.1.3. 결과	948
85.2. 로그 회전	948
85.3. IDM 상태 점검을 사용하여 로그 회전 구성	949
86장. IDM 상태 점검을 사용하여 서비스 확인	951
86.1. 서비스 상태 점검 테스트	951
86.2. 상태 점검을 사용하는 폴링 서비스	952
87장. IDM 상태 점검을 사용하여 IDM 및 AD 신뢰 구성 확인	954
87.1. IDM 및 AD 신뢰 상태 점검 테스트	954
87.2. HEALTHCHECK 도구를 사용하여 신뢰를 건너뛰니다.	955

88장. IDM 상태 점검을 사용하여 인증서 확인	957
88.1. IDM 인증서 상태 점검 테스트	957
88.2. 상태 점검 툴을 사용한 암호 인증서	959
89장. IDM 상태 점검을 사용하여 시스템 인증서 확인	961
89.1. 시스템 인증서 상태 점검 테스트	961
89.2. HEALTHCHECK를 사용한 시스템 인증서	962
90장. IDM 상태 점검을 사용하여 디스크 공간 확인	964
90.1. 디스크 공간 상태 점검 테스트	964
90.2. 상태 점검 도구를 사용하여 디스크 공간	965
91장. HEALTHCHECK를 사용하여 IDM 구성 파일에 대한 권한 확인	967
91.1. 파일 권한 상태 점검 테스트	967
91.2. HEALTHCHECK를 사용하는 구성 파일	969
92장. 상태 점검을 사용하여 IDM 복제 확인	971
92.1. 복제 상태 점검 테스트	971
92.2. 상태 점검을 사용하여 복제 복제	972
93장. IDM 상태 점검을 사용하여 DNS 레코드 확인	974
93.1. DNS 레코드 상태 점검 테스트	974
93.2. 상태 점검 툴을 사용하여 DNS 레코드 표시	974
94장. 숨겨진 복제본 데모 또는 승격	976
95장. IDENTITY MANAGEMENT 보안 설정	977
95.1. IDENTITY MANAGEMENT가 기본 보안 설정을 적용하는 방법	977
95.2. IDENTITY MANAGEMENT에서 익명 LDAP 바인딩	977
96장. IDM 도메인 멤버에서 SAMBA 설정	978
96.1. 도메인 구성원에 SAMBA 설치를 위한 IDM 도메인 준비	978
96.2. GPO를 사용하여 ACTIVE DIRECTORY에서 AES 암호화 유형 활성화	980
96.3. IDM 클라이언트에 SAMBA 서버 설치 및 구성	982
96.4. IDM에서 새 도메인을 신뢰하는 경우 수동으로 ID 매핑 구성 추가	984
96.5. 추가 리소스	986
97장. 다른 RED HAT 제품과 IDM 통합	987
98장. NIS에서 IDENTITY MANAGEMENT로 마이그레이션	988
98.1. IDM에서 NIS 활성화	988
98.2. NIS에서 IDM으로 사용자 항목 마이그레이션	989
98.3. NIS에서 IDM으로 사용자 그룹 마이그레이션	990
98.4. NIS에서 IDM으로 호스트 항목 마이그레이션	992
98.5. NIS에서 IDM으로 넷그룹 항목 마이그레이션	993
98.6. NIS에서 IDM으로 자동 마운트 맵 마이그레이션	995
99장. IDM에서 자동 마운트 사용	997
99.1. IDM에서 AUTOFS 및 자동 마운트	998
99.2. NFS 서버에 대한 IDM 키맵 구성	999
99.3. IDM에서 NFS 공유 내보내기	1000
99.4. IDM CLI를 사용하여 IDM에서 자동 마운트 위치 및 맵 구성	1002
99.5. 자동 마운트 위치에 IDM 클라이언트 추가	1004
99.6. IDM 클라이언트에 자동 마운트 설정	1004
99.7. IDM 사용자가 IDM 클라이언트의 NFS 공유에 액세스할 수 있는지 확인	1005

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.

Identity Management에서 계획된 용어 교체는 다음과 같습니다.

- 차단 목록 대체 블랙리스트
- 목록 교체 허용 화이트리스트
- 2차 대체 슬레이브
- master 라는 단어는 컨텍스트에 따라 더 정확한 언어로 교체됩니다.
 - IdM 서버가 IdM 마스터 교체
 - CA 갱신 서버가 CA 갱신 마스터 교체
 - CRL 게시자 서버가 CRL 마스터 교체
 - 멀티 공급자 대체 멀티 마스터

RED HAT 문서에 관한 피드백 제공

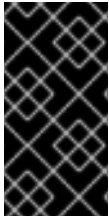
문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 이를 개선할 수 있는지 알려 주십시오. 이렇게 하려면 다음을 수행합니다.

- 특정 문구에 대한 간단한 주석은 다음과 같습니다.
 1. 문서가 *Multi-page HTML* 형식으로 표시되는지 확인합니다. 또한 문서 오른쪽 상단에 **Feedback** (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.
- 보다 상세하게 피드백을 제출하려면 다음과 같이 Bugzilla 티켓을 생성하십시오.
 1. [Bugzilla](#) 웹 사이트로 이동하십시오.
 2. Component로 **Documentation**을 선택하십시오.
 3. **Description** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
 4. **Submit Bug**를 클릭하십시오.

1장. 명령줄에서 IDENTITY MANAGEMENT에 로그인

IdM(Identity Management)은 Kerberos 프로토콜을 사용하여 SSO(Single Sign-On)를 지원합니다.

SSO(Single Sign-On)는 사용자가 올바른 사용자 이름과 암호를 한 번만 입력한 다음 시스템에서 자격 증명을 다시 입력하라는 메시지를 표시하지 않고 IdM 서비스에 액세스합니다.



중요

IdM에서 SSSD(System Security Services Daemon)는 사용자가 해당 Kerberos 주체 이름을 사용하여 IdM 클라이언트 시스템에서 데스크탑 환경에 성공적으로 로그인한 후 사용자의 티켓 제공 티켓(TGT)을 자동으로 가져옵니다. 즉, 로그인한 후에는 **kinit** 유틸리티를 사용하여 IdM 리소스에 액세스할 필요가 없습니다.

Kerberos 인증 정보 캐시를 지우거나 Kerberos TGT가 만료된 경우 Kerberos 티켓을 수동으로 요청하여 IdM 리소스에 액세스해야 합니다. 다음 섹션에서는 IdM에서 Kerberos를 사용할 때 기본 사용자 작업을 보여줍니다.

1.1. KINIT 를 사용하여 IDM에 수동으로 로그인합니다.

이 절차에서는 **kinit** 유틸리티를 사용하여 IdM(Identity Management) 환경에 수동으로 인증하는 방법을 설명합니다. **kinit** 유틸리티는 IdM 사용자를 대신하여 Kerberos 티켓 부여 티켓(TGT)을 가져와 캐시합니다.



참고

초기 Kerberos TGT를 삭제했거나 만료된 경우에만 이 절차를 사용하십시오. IdM 사용자로 로컬 시스템에 로그인할 때도 IdM에 자동으로 로그인됩니다. 즉, 로그인한 후에는 **kinit** 유틸리티를 사용하여 IdM 리소스에 액세스할 필요가 없습니다.

절차

1. IdM에 로그인하려면

- 로컬 시스템에 현재 로그인한 사용자의 사용자 이름 아래에 사용자 이름을 지정하지 않고 **kinit** 를 사용합니다. 예를 들어 로컬 시스템에서 **example_user** 로 로그인한 경우 다음을 수행합니다.

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

로컬 사용자의 사용자 이름이 IdM의 사용자 항목과 일치하지 않으면 인증 시도가 실패합니다.

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

- 로컬 사용자 이름에 해당하지 않는 Kerberos 주체를 사용하여 필수 사용자 이름을 **kinit** 유틸리티에 전달합니다. 예를 들어 **admin** 사용자로 로그인하려면 다음을 수행합니다.

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

- 선택적으로 로그인에 성공했는지 확인하려면 **klist** 유틸리티를 사용하여 캐시된 TGT를 표시합니다. 다음 예에서 캐시에는 **example_user** 주체의 티켓이 포함되어 있습니다. 즉, 이 특정 호스트에서는 현재 **example_user** 만 IdM 서비스에 액세스할 수 있습니다.

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

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

1.2. 사용자의 활성 KERBEROS 티켓 삭제

이 섹션에서는 사용자의 활성 Kerberos 티켓이 포함된 자격 증명 캐시를 지우는 방법을 설명합니다.

절차

- Kerberos 티켓을 삭제하려면 다음을 수행합니다.

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

- 선택적으로 Kerberos 티켓이 삭제되었는지 확인하려면 다음을 수행하십시오.

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

1.3. KERBEROS 인증을 위한 외부 시스템 구성

이 섹션에서는 IdM(Identity Management) 사용자가 Kerberos 자격 증명을 사용하여 외부 시스템에서 IdM에 로그인할 수 있도록 외부 시스템을 구성하는 방법에 대해 설명합니다.

외부 시스템에서 Kerberos 인증을 활성화하는 것은 인프라에 여러 영역 또는 중복되는 도메인이 포함된 경우 특히 유용합니다. 시스템이 **ipa-client-install** 을 통해 IdM 도메인에 등록되지 않은 경우에도 유용합니다.

IdM 도메인의 멤버가 아닌 시스템에서 IdM에 대한 Kerberos 인증을 활성화하려면 외부 시스템에 IdM별 Kerberos 구성 파일을 정의합니다.

사전 요구 사항

- krb5-workstation** 패키지는 외부 시스템에 설치됩니다.
패키지가 설치되어 있는지 여부를 확인하려면 다음 CLI 명령을 사용합니다.

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

절차

- IdM 서버에서 외부 시스템으로 **/etc/krb5.conf** 파일을 복사합니다. 예를 들면 다음과 같습니다.

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



주의

외부 시스템의 기존 **krb5.conf** 파일을 덮어쓰지 마십시오.

2. 외부 시스템에서 복사된 IdM Kerberos 구성 파일을 사용하도록 터미널 세션을 설정합니다.

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

KRB5_CONFIG 변수는 로그아웃할 때까지 일시적으로만 존재합니다. 이 손실을 방지하려면 다른 파일 이름으로 변수를 내보냅니다.

3. **/etc/krb5.conf.d/** 디렉토리에서 외부 시스템으로 Kerberos 구성 조각을 복사합니다.

이제 외부 시스템의 사용자는 **kinit** 유틸리티를 사용하여 IdM 서버에 대해 인증할 수 있습니다.

추가 리소스

- Kerberos에 대한 자세한 내용은 **krb5.conf(5)**, **kinit(1)**, **klist(1)** 및 **kdestroy(1)** 도움말 페이지를 참조하십시오.

2장. ID 관리 서비스 보기, 시작 및 중지

IdM(Identity Management) 서버는 DC(도메인 컨트롤러)로 작동하는 Red Hat Enterprise Linux 시스템입니다. IdM 서버에서 다양한 서비스가 실행되고 있습니다. 특히 Directory Server, CA(인증 기관), DNS 및 Kerberos가 있습니다.

2.1. IDM 서비스

2.1.1. IdM 서버에서 호스팅하는 서비스 목록

다음 서비스는 대부분 IdM 서버에 설치해야 하는 것은 아닙니다. 예를 들어 IdM 도메인 외부의 외부 서버에 CA(인증 기관) 또는 DNS 서버와 같은 서비스를 설치할 수 있습니다.

Kerberos

krb5kdc 및 kadmin 서비스

IdM은 **Kerberos** 프로토콜을 사용하여 SSO(Single-Sign-On)를 지원합니다. Kerberos를 사용하면 사용자는 올바른 사용자 이름과 암호를 한 번만 제공해야 하며 시스템에서 자격 증명을 다시 요구하지 않고 IdM 서비스에 액세스할 수 있습니다.

Kerberos는 다음 두 부분으로 나뉩니다.

- **krb5kdc** 서비스는 Kerberos 인증 서비스 및 KDC(키 배포 센터) 데몬입니다.
- **kadmin** 서비스는 Kerberos 데이터베이스 관리 프로그램입니다.

IdM에서 Kerberos를 사용하여 인증하는 방법에 대한 자세한 내용은 [명령줄에서 ID 관리에 로그인](#) 및 [웹 UI에서 IdM에 로그인을 참조하십시오. Kerberos 티켓 사용](#).

LDAP 디렉터리 서버

dirsrv 서비스

IdM **LDAP 디렉터리 서버** 인스턴스는 Kerberos, 사용자 계정, 호스트 항목, 서비스, 정책, DNS 등과 관련된 정보와 같은 모든 IdM 정보를 저장합니다. LDAP 디렉터리 서버 인스턴스는 [Red Hat Directory Server](#) 와 동일한 기술을 기반으로 합니다. 그러나 IdM 관련 작업에 맞게 조정됩니다.

인증 기관

pki-tomcatd 서비스

통합 CA(인증 기관)는 [Red Hat Certificate System](#) 과 동일한 기술을 기반으로 합니다. **pki** 는 인증서 시스템 서비스에 액세스하기 위한 명령줄 인터페이스입니다.

필요한 모든 인증서를 개별적으로 생성하고 제공하는 경우 통합 CA 없이 서버를 설치할 수도 있습니다.

자세한 내용은 [CA 서비스](#) 계획을 참조하십시오.

DNS(Domain Name System)

명명된 서비스

IdM은 동적 서비스 검색을 위해 **DNS** 를 사용합니다. IdM 클라이언트 설치 유틸리티는 DNS의 정보를 사용하여 클라이언트 시스템을 자동으로 구성할 수 있습니다. 클라이언트가 IdM 도메인에 등록되면 DNS를 사용하여 도메인 내에서 IdM 서버 및 서비스를 찾습니다. Red Hat Enterprise Linux의 DNS (Domain

Name System) 프로토콜의 **BIND** (Berkeley Internet Name Domain) 구현에는 이름 지정된 DNS 서버가 포함되어 있습니다. **named-pkcs11**은 PKCS#11 암호화 표준에 대한 기본 지원으로 구축된 BIND DNS 서버의 버전입니다.

자세한 내용은 [DNS 서비스 및 호스트 이름](#) 계획을 참조하십시오.

Apache HTTP Server

httpd 서비스

Apache HTTP 웹 서버는 IdM 웹 UI를 제공하며 인증 기관과 기타 IdM 서비스 간의 통신을 관리합니다.

Samba / Winbind

SMB 및 **winbind** 서비스

Samba는 Red Hat Enterprise Linux에서 CIFS(Common Internet File System) 프로토콜이라고도 하는 SMB(서버 메시지 블록) 프로토콜을 구현합니다. smb 서비스를 통해 SMB 프로토콜을 사용하면 파일 공유 및 공유 프린터와 같은 서버의 리소스에 액세스할 수 있습니다. AD(Active Directory) 환경을 사용하여 Trust를 구성한 경우 'Winbind' 서비스는 IdM 서버와 AD 서버 간의 통신을 관리합니다.

일회성 암호(OTP) 인증

ipa-otpd 서비스

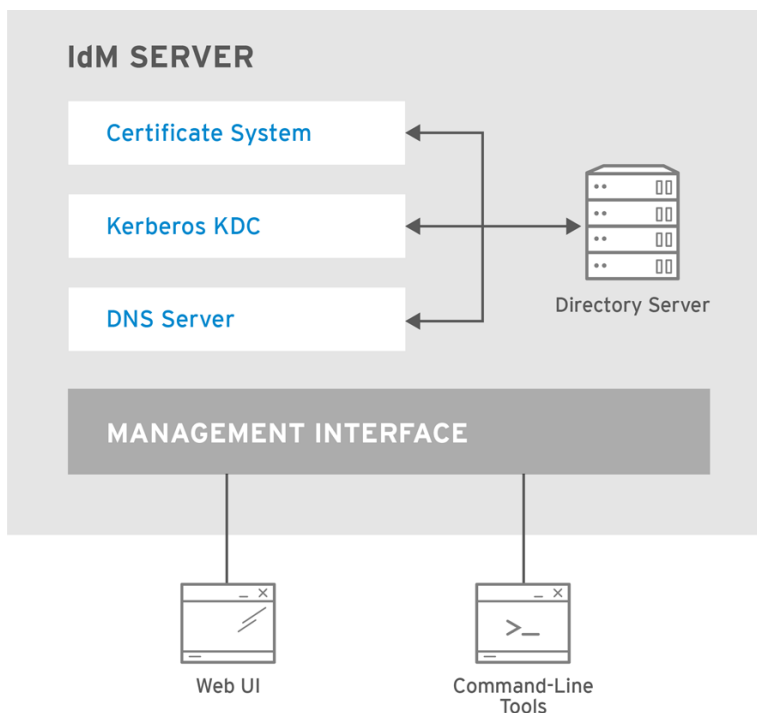
일회성 암호(OTP)는 이중 인증의 일부로 하나의 세션에서만 인증 토큰에 의해 생성되는 암호입니다. OTP 인증은 **ipa-otpd** 서비스를 통해 Red Hat Enterprise Linux에서 구현됩니다.

자세한 내용은 [일회성 암호를 사용하여 Identity Management Web UI](#) 로 로그인을 참조하십시오.

OpenDNSSEC

ipa-dnskeysyncd 서비스

OpenDNSSEC는 DNSSEC (DNS 보안 확장) 키 및 영역 서명을 유지하는 프로세스를 자동화하는 DNS 관리자입니다. **ipa-dnskeysyncd** 서비스는 IdM 디렉터리 서버와 OpenDNSSEC 간의 동기화를 관리합니다.



RHEL_404973_0516

2.1.2. IdM 클라이언트가 호스팅하는 서비스 목록

- 시스템 보안 서비스 데몬: **sssd** 서비스

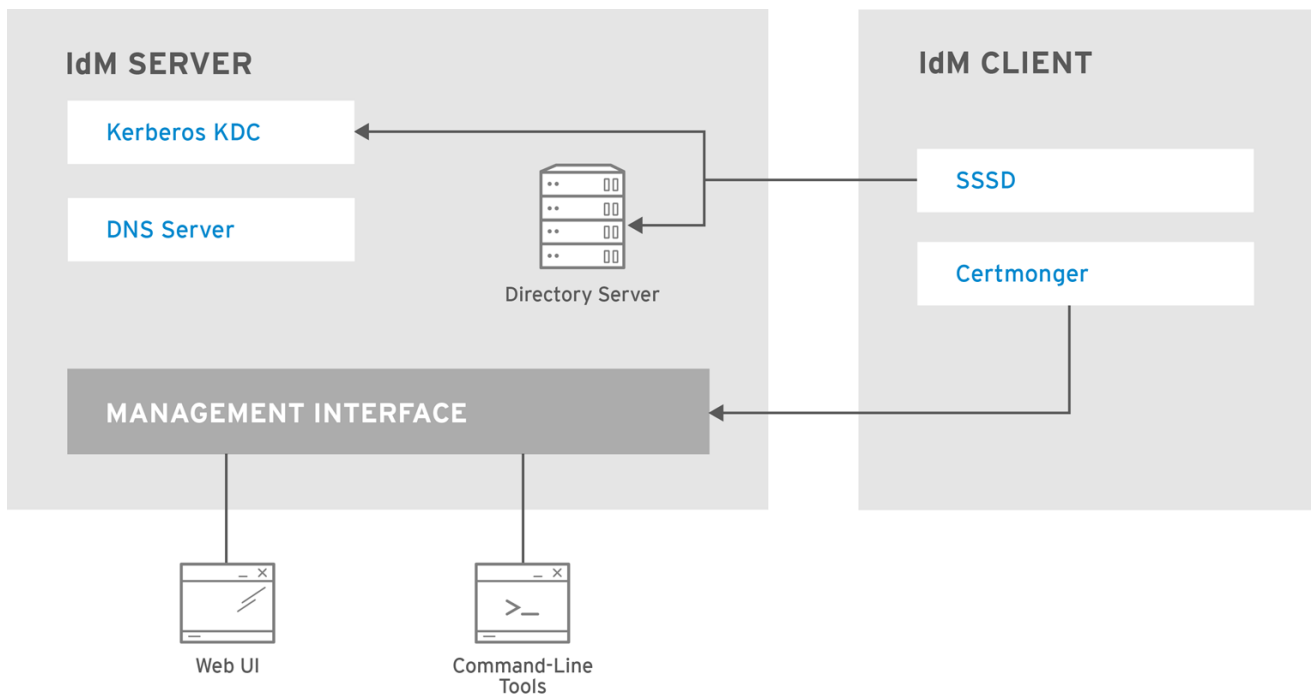
SSSD(시스템 보안 서비스 데몬)은 사용자 인증 및 캐싱 자격 증명을 관리하는 클라이언트 측 애플리케이션입니다. 캐싱을 사용하면 IdM 서버를 사용할 수 없거나 클라이언트가 오프라인 상태가 되면 로컬 시스템에서 일반 인증 작업을 계속할 수 있습니다.

자세한 내용은 [SSSD 및 이점 이해](#)를 참조하십시오.

- **certmonger**: **certmonger** 서비스

certmonger 서비스는 클라이언트의 인증서를 모니터링하고 갱신합니다. 시스템에서 서비스에 대한 새 인증서를 요청할 수 있습니다.

자세한 내용은 [certmonger를 사용하여 서비스의 IdM 인증서 가져오기](#)를 참조하십시오.



RHEL_404973_0516

2.2. IDM 서비스 상태 보기

IdM 서버에 구성된 IdM 서비스의 상태를 보려면 **ipactl status** 명령을 실행합니다.

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

서버의 **ipactl status** 명령 출력은 IdM 구성에 따라 다릅니다. 예를 들어 IdM 배포에 DNS 서버가 포함되지 않은 경우 **명명된** 서비스가 목록에 표시되지 않습니다.



참고

IdM 웹 UI를 사용하여 특정 IdM 서버에서 실행되는 모든 IdM 서비스의 상태를 볼 수 없습니다. 다른 서버에서 실행 중인 Kerberized 서비스는 IdM 웹 UI의 **ID → 서비스** 탭에서 볼 수 있습니다.

전체 서버 또는 개별 서비스만 시작하거나 중지할 수 있습니다.

전체 IdM 서버를 시작, 중지 또는 다시 시작하려면 다음을 참조하십시오.

- 2.3절. “전체 ID 관리 서버 시작 및 중지: **ipactl** 유틸리티”

개별 IdM 서비스를 시작, 중지 또는 다시 시작하려면 다음을 참조하십시오.

- 2.4절. “개별 ID 관리 서비스 시작 및 중지: **systemctl** 유틸리티”

IdM 소프트웨어 버전을 표시하려면 다음을 참조하십시오.

- 2.5절. “IdM 소프트웨어 버전을 표시하는 방법”

2.3. 전체 ID 관리 서버 시작 및 중지: **IPACTL** 유틸리티

설치된 모든 서비스와 함께 **ipactl** 유틸리티를 사용하여 전체 IdM 서버를 중지, 시작 또는 다시 시작합니다. **ipactl** 유틸리티를 사용하면 모든 서비스가 적절한 순서로 중지, 시작 또는 다시 시작됩니다. **ipactl** 명령을 실행하기 위해 유효한 Kerberos 티켓이 필요하지 않습니다.

ipactl 명령

전체 IdM 서버를 시작하려면 다음을 수행합니다.

```
# ipactl start
```

전체 IdM 서버를 중지하려면 다음을 수행합니다.

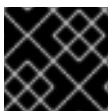
```
# ipactl stop
```

전체 IdM 서버를 다시 시작하려면 다음을 수행합니다.

```
# ipactl restart
```

IdM을 구성하는 모든 서비스의 상태를 표시하려면 다음을 수행합니다.

```
# ipactl status
```



중요

IdM 웹 UI를 사용하여 **ipactl** 명령을 수행할 수 없습니다.

2.4. 개별 ID 관리 서비스 시작 및 중지: **SYSTEMCTL** 유틸리티

일반적으로 IdM 구성 파일을 수동으로 변경하는 것은 권장되지 않습니다. 그러나 특정 상황에서는 관리자가 특정 서비스의 수동 구성을 수행해야 합니다. 이러한 상황에서는 **systemctl** 유틸리티를 사용하여 개별 IdM 서비스를 중지, 시작 또는 다시 시작합니다.

예를 들어 다른 IdM 서비스를 수정하지 않고 Directory Server 동작을 사용자 정의한 후 **systemctl** 을 사용합니다.

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

또한 Active Directory를 사용하여 IdM 트러스트를 처음 배포하는 경우 **/etc/sss/sss.conf** 파일을 수정하여 다음을 추가합니다.

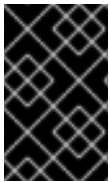
- 원격 서버에 대기 시간이 길어지는 환경에서 시간 제한 구성 옵션을 조정하는 특정 매개 변수
- Active Directory 사이트 유사성을 조정하기 위한 특정 매개 변수
- 글로벌 IdM 설정에서 제공하지 않는 특정 구성 옵션의 재정의

/etc/sss/sss.conf 파일에서 변경한 사항을 적용하려면 다음을 수행합니다.

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

SSSD(System Security Services Daemon)에서 구성을 자동으로 다시 읽거나 다시 적용하지 않으므로 **systemctl restart sssd.service** 를 실행해야 합니다.

IdM ID 범위에 영향을 미치는 변경 사항은 전체 서버 재부팅을 권장합니다.



중요

여러 IdM 도메인 서비스를 다시 시작하려면 항상 **ipactl** 을 사용하십시오. IdM 서버와 함께 설치된 서비스 간 종속성 때문에 시작하고 중지된 순서가 중요합니다. **ipactl** 유틸리티는 서비스가 적절한 순서로 시작되고 중지되었는지 확인합니다.

유용한 systemctl 명령

특정 IdM 서비스를 시작하려면 다음을 수행합니다.

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

특정 IdM 서비스를 중지하려면 다음을 수행합니다.

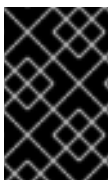
```
# systemctl stop name.service
```

특정 IdM 서비스를 다시 시작하려면 다음을 수행합니다.

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

특정 IdM 서비스의 상태를 보려면 다음을 수행합니다.

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



중요

IdM 웹 UI를 사용하여 IdM 서버에서 실행되는 개별 서비스를 시작하거나 중지할 수 없습니다. 웹 UI를 사용하여 **ID** → 서비스로 이동하고 서비스를 선택하여 Kerberized 서비스의 설정을 수정할 수 있습니다.

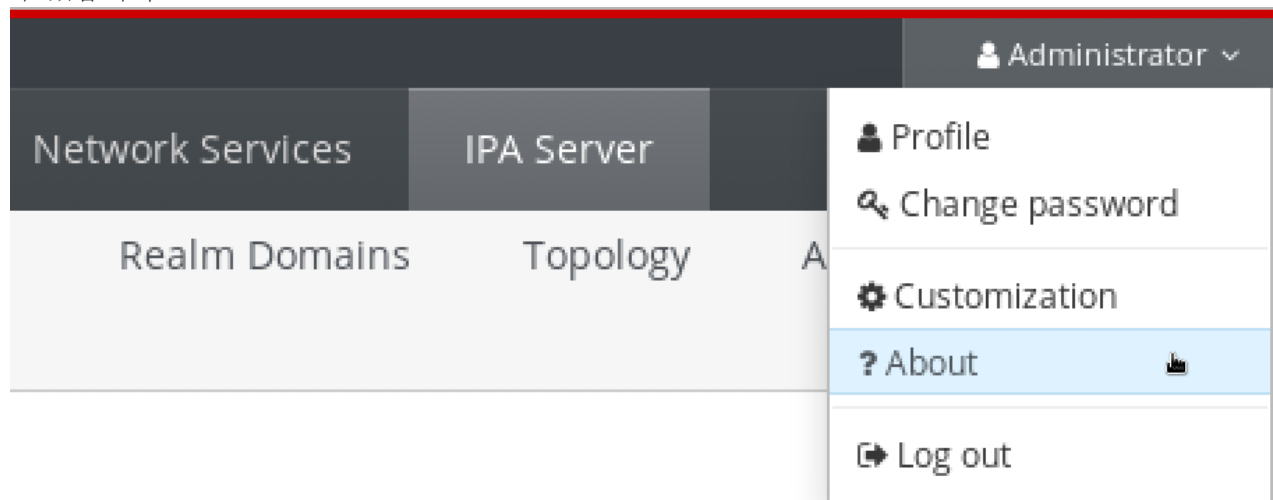
2.5. IDM 소프트웨어 버전을 표시하는 방법

다음을 사용하여 IdM 버전 번호를 표시할 수 있습니다.

- IdM WebUI
- IPA 명령
- rpm 명령

WebUI를 통해 버전 표시

IdM WebUI에서 오른쪽의 사용자 이름 메뉴에서 **About** (정보)을 선택하여 소프트웨어 버전을 표시할 수 있습니다.



ipa 명령으로 버전 표시

명령줄에서 **ipa --version** 명령을 사용합니다.

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

rpm 명령으로 버전 표시

IdM 서비스가 제대로 작동하지 않는 경우 **rpm** 유틸리티를 사용하여 현재 설치된 **ipa-server** 패키지의 버전 번호를 확인할 수 있습니다.

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

3장. IDM 명령줄 유틸리티 소개

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

사전 요구 사항

- 설치 및 액세스 가능한 IdM 서버.
자세한 내용은 [Identity Management 설치](#)를 참조하십시오.
- IPA 명령줄 인터페이스를 사용하려면 유효한 Kerberos 티켓을 사용하여 IdM을 인증합니다.
유효한 Kerberos 티켓을 가져오는 방법에 대한 자세한 내용은 [명령줄에서 Identity Management에 로그인](#)을 참조하십시오.

3.1. IPA 명령줄 인터페이스란 무엇입니까?

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

IdM을 관리하는 데 사용되는 많은 하위 명령(예: 새 사용자를 추가하는 **ipa user-add** 명령)을 지원합니다.

IPA CLI를 사용하면 다음을 수행할 수 있습니다.

- 네트워크의 사용자, 그룹, 호스트 및 기타 개체를 추가, 관리 또는 제거합니다.
- 인증서 관리.
- 검색 항목.
- 오브젝트 표시 및 나열.
- 액세스 권한 설정.
- 올바른 명령 구문에 대한 도움말을 가져옵니다.

3.2. IPA 도움말이란 무엇입니까?

IPA 도움말은 IdM 서버를 위한 기본 제공되는 설명서 시스템입니다.

IPA CLI(명령줄 인터페이스)는 로드된 IdM 플러그인 모듈에서 사용 가능한 도움말 주제를 생성합니다.
IPA 도움말을 성공적으로 실행하려면 다음을 수행해야 합니다.

- IdM 서버가 설치되어 실행 중이 되도록 합니다.
- 유효한 Kerberos 티켓으로 인증됩니다.

옵션 없이 **ipa help** 명령을 실행하면 기본 도움말 사용 및 가장 일반적인 명령 예제에 대한 정보가 표시됩니다.

옵션을 사용하여 도움말을 실행하면 다음과 같은 구문이 있습니다.

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

- `[] ^-dbbrackets`는 모든 매개변수가 선택 사항임을 의미하며 **ipa help** 만 쓸 수 있으며 명령이 실행됩니다.

- |cd-파이프 문자의 의미. 따라서 기본 **ipa help** 명령을 사용하여 IC 또는 COMMAND 또는 topic 또는 명령을 사용할 수 있습니다.
- 토픽 - **ipa help topic** 명령을 실행할 수 있으며 올바르게 실행됩니다. 명령은 IPA 도움말에서 다루는 항목(예: 사용자,인증서,서버 등)을 표시합니다.
- -- IC ICTI- 자릿 문자가 포함된 내용은 변수를 의미하므로 특정 주제(예: ipa도움말 사용자)를 사용할 수 있습니다.
- 명령에 ipa help 명령을 실행하면 ipa help 명령이 올바르게 실행됩니다. 명령은 IPA 도움말에서 다루는 명령 목록을 표시합니다(예: user-add,ca-enable,server-show 등).
- COMMAND TI- 자세한 문자가 있는 COMMAND는 변수를 의미하므로 특정 명령(예: ipa help user-add)을 사용할 수 있습니다.

3.3. IPA 도움말 항목 사용

다음 절차는 명령줄 인터페이스에서 IPA 도움말을 사용하여 이해하는 데 도움이 됩니다.

절차

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

```
$ ipa help topics
```

3. 주제 중 하나를 선택하고 다음 패턴에 따라 명령을 만듭니다. **ipa help [topic_name]** 은 **topic_name** 문자열 대신 이전 단계에서 나열된 항목 중 하나를 추가합니다.
예에서는 다음 주제를 사용합니다. **user**

```
$ ipa help user
```

4. IPA help 명령이 너무 길고 전체 텍스트를 볼 수 없는 경우 다음 구문을 사용하십시오.

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

아래로 스크롤하여 전체 도움말을 읽을 수 있습니다.

IPA CLI에는 사용자 주제에 대한 도움말 페이지가 표시됩니다. 개요를 읽은 후에는 topic 명령을 사용하여 작업하기 위한 패턴이 있는 여러 예제를 볼 수 있습니다.

3.4. IPA 도움말 명령 사용

다음 절차는 명령줄 인터페이스에서 IPA 도움말 명령 생성을 이해하는 데 도움이 됩니다.

절차

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

```
$ ipa help commands
```

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

\$ ipa help user-add

추가 리소스

- 자세한 내용은 **man ipa** 페이지를 참조하십시오.

3.5. IPA 명령 구조

IPA CLI는 다음 유형의 명령을 구분합니다.

- 기본 제공 명령 UTC-Built-in 명령은 모두 IdM 서버에서 사용할 수 있습니다.
- 플러그인 제공 명령

IPA 명령 구조를 사용하면 다양한 유형의 오브젝트를 관리할 수 있습니다. 예를 들면 다음과 같습니다.

- 사용자,
- 호스트,
- DNS 레코드,
- 인증서,

기타 여러 가지.

이러한 대부분의 오브젝트의 경우 IPA CLI에는 다음 명령이 포함됩니다.

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

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

기존 오브젝트를 변경하려면 오브젝트를 정의해야 하므로 명령에 오브젝트 **ipa user-mod USER_NAME [options]** 도 포함됩니다.

3.6. IPA 명령을 사용하여 IDM에 사용자 계정 추가

다음은 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
```

명령은 사용자 계정을 생성하는 데 필요한 기본 데이터를 추가할 수 있는 스크립트를 실행합니다.

3. Name(이름): 필드에 새 사용자의 이름을 입력하고 **Enter** 키를 누릅니다.
4. Last name:(이름:) 필드에 새 사용자의 성을 입력하고 **Enter** 키를 누릅니다.
5. User login [suggested user name]: 사용자 이름을 입력하거나 제안된 사용자 이름이 작동하는 경우 **Enter** 키를 누릅니다.
사용자 이름은 전체 IdM 데이터베이스에서 고유해야 합니다. 오류가 발생하면 사용자가 이미 존재하는 경우 **ipa user-add** 명령으로 처음부터 시작하고 다른 사용자 이름을 시도해야 합니다.

사용자 이름을 성공적으로 추가한 후에는 사용자 계정이 IdM 데이터베이스에 추가되고 다음 로그에 IPA CLI(명령줄 인터페이스)가 출력됩니다.

```
-----
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
```

위 화면에서 사용자 암호가 사용자 계정으로 설정되지 않았습니다. 암호도 추가하려면 다음 구문에서 **ipa user-add** 명령을 사용하십시오.


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

그러면 IPA CLI에서 사용자 이름 및 암호를 추가하거나 확인하도록 요청합니다.

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

추가 리소스

매개변수에 대한 자세한 내용을 보려면 명령줄에 다음 help 명령을 입력합니다.

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

3.7. IPA 명령을 사용하여 IDM에서 사용자 계정 수정

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

기본 명령 구문은 **user-add** 구문과 다릅니다. 변경 사항을 수행할 기존 사용자 계정을 정의해야 하기 때문입니다(예: 암호 추가).

사전 요구 사항

- IdM 서버에서 사용자 계정을 수정하려면 관리자 권한이 있어야 합니다.

절차

1. 터미널을 열고 IdM 서버에 연결합니다.
2. 암호를 추가하려면 명령을 입력합니다.

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

이 명령은 새 암호를 추가할 수 있는 스크립트를 실행합니다.

3. 새 암호를 입력하고 **Enter** 키를 누릅니다.

사용자 이름을 성공적으로 추가한 후에는 사용자 계정이 IdM 데이터베이스에 추가되고 다음 로그의 IPA CLI가 출력됩니다.

```
-----
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 사용자 암호가 계정에 대해 설정되었으며 사용자가 IdM에 로그인할 수 있습니다.

추가 리소스

매개변수에 대한 자세한 내용을 보려면 명령줄에 다음 `help` 명령을 입력합니다.

```
$ ipa help user-mod
```

3.8. IDM 유틸리티에 값 목록을 제공하는 방법

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

IdM은 다중 값 목록을 제공하는 다음 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 개체에 권한을 추가하는 **permissions-add** 명령을 보여줍니다. 예제에는 오브젝트가 언급되지 않습니다. 대신 ... 권한을 추가하려는 객체를 추가해야 합니다.

명령줄에서 이러한 다중 값 속성을 업데이트하면 IdM은 이전 값 목록을 새 목록으로 완전히 덮어씁니다. 따라서 다중 값 특성을 업데이트할 때 추가하려는 단일 값뿐만 아니라 전체 새 목록을 지정해야 합니다.

위의 명령에서 권한 목록에는 읽기, 쓰기, 삭제가 포함됩니다. **permission-mod** 명령으로 목록을 업데이트하기로 결정하면 모든 값을 추가해야 합니다. 그렇지 않으면 언급되지 않은 값이 삭제됩니다.

예 1: UTC - **ipa permission -mod** 명령은 이전에 추가한 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 포함되어 있지 않기 때문에 **--right=delete** 인수를 삭제합니다.

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

또는

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

3.9. IDM 유틸리티와 함께 특수 문자를 사용하는 방법

특수 문자를 **ipa** 명령에 포함하는 명령줄 인수를 전달할 때 백슬래시(\)를 사용하여 이 문자를 이스케이프합니다. 예를 들어 일반적인 특수 문자에는 대괄호(< 및 >), 앰퍼샌드(&), 별표(*) 또는 세로 막대(|)가 있습니다.

예를 들어 별표(*)를 이스케이프하려면 다음을 수행합니다.

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

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

4장. 명령줄에서 IDENTITY MANAGEMENT 항목 검색

다음 섹션에서는 개체를 찾거나 표시하는 데 도움이 되는 IPA 명령을 사용하는 방법에 대해 설명합니다.

4.1. IDM 항목 나열 개요

이 섹션에서는 특정 유형의 IdM 항목을 검색하는 데 도움이 되는 **ipa *-find** 명령에 대해 설명합니다.

find 명령을 모두 나열하려면 다음 **ipa help** 명령을 사용합니다.

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

특정 사용자가 IdM 데이터베이스에 포함되어 있는지 확인해야 할 수도 있습니다. 다음 명령을 사용하여 모든 사용자를 나열할 수 있습니다.

```
$ ipa user-find
```

지정된 속성에 키워드가 포함된 사용자 그룹을 나열하려면 다음을 수행합니다.

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

예를 들어 **ipa group-find admin** 명령은 이름 또는 설명에 문자열 **admin** 이 포함된 모든 그룹을 나열합니다.

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

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

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

사용자 그룹을 검색할 때 검색 결과를 특정 사용자가 포함된 그룹으로 제한할 수도 있습니다.

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

특정 사용자가 포함되지 않은 그룹을 검색하려면 다음을 수행합니다.

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

4.2. 특정 항목의 세부 정보 표시

ipa *-show 명령을 사용하여 특정 IdM 항목에 대한 세부 정보를 표시합니다.

절차

- `server.example.com`이라는 호스트에 대한 세부 정보를 표시하려면 다음을 수행합니다.

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

```
Host name: server.example.com
```

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

```
...
```

4.3. 검색 크기 및 시간 제한 조정

IdM 사용자 목록을 요청하는 등의 일부 쿼리는 매우 많은 항목을 반환할 수 있습니다. 이러한 검색 작업을 조정하면 `ipa` 사용자 검색과 같은 `ipa *-find` 명령을 실행하고 웹 UI에 해당 목록을 표시할 때 전체 서버 성능을 향상시킬 수 있습니다.

검색 크기 제한

클라이언트 CLI에서 서버로 전송하거나 IdM 웹 UI에 액세스하는 브라우저에서 전송된 요청에 대해 반환된 최대 항목 수를 정의합니다.

기본값: 100개 항목.

검색 시간 제한

서버에서 검색을 실행할 때까지 대기하는 최대 시간(초)을 정의합니다. 검색이 이 제한에 도달하면 서버에서 검색을 중지하고 해당 시간에서 검색된 항목을 반환합니다.

기본값: 2초.

값을 `-1`로 설정하면 IdM은 검색할 때 제한을 적용하지 않습니다.



중요

검색 크기 또는 시간 제한을 너무 높게 설정하면 서버 성능에 부정적인 영향을 줄 수 있습니다.

4.3.1. 명령줄에서 검색 크기 및 시간 제한 조정

다음 텍스트는 명령줄에서 검색 크기 및 시간 제한을 조정하는 방법을 설명합니다.

- 전 세계
- 특정 항목의 경우

절차

1. CLI에서 현재 검색 시간 및 크기 제한을 표시하려면 `ipa config-show` 명령을 사용합니다.

```
$ ipa config-show
```

```
Search time limit: 2
```

```
Search size limit: 100
```

2. 모든 쿼리에 대해 제한을 전역적으로 조정하려면 `ipa config-mod` 명령을 사용하고 `--searchrecordslimit` 및 `--searchtimelimit` 옵션을 추가합니다. 예를 들면 다음과 같습니다.

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

3. 특정 쿼리에 대해서만 제한을 조정하려면 명령에 **--sizelimit** 또는 **--timelimit** 옵션을 추가합니다.
예를 들면 다음과 같습니다.

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

4.3.2. 웹 UI에서 검색 크기 및 시간 제한 조정

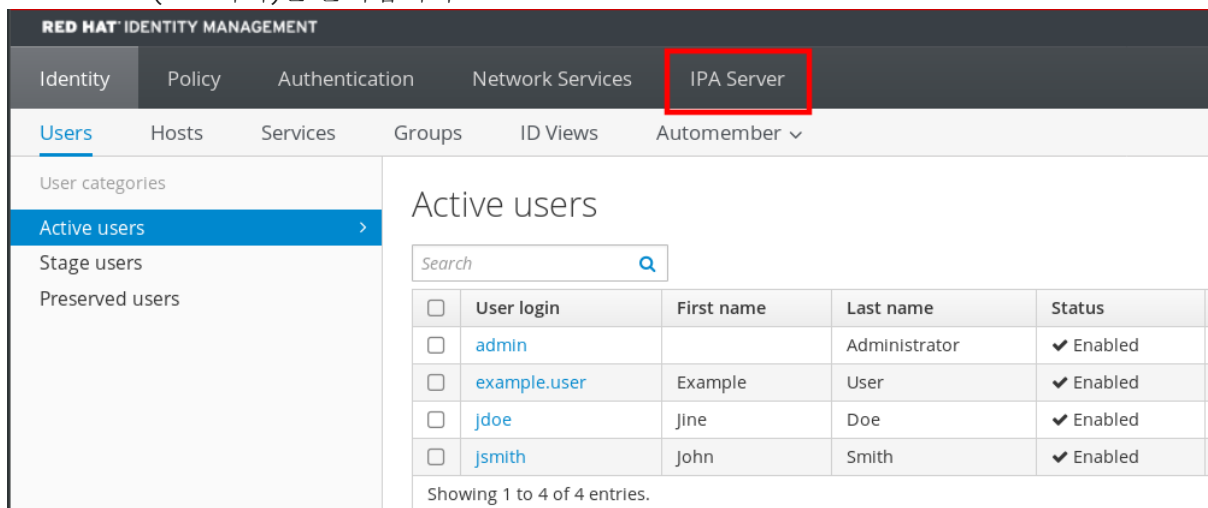
다음 텍스트는 IdM 웹 UI에서 검색 크기 및 시간 제한을 조정하는 방법을 설명합니다.

- 전 세계
- 특정 항목의 경우

절차

모든 쿼리에 대해 제한을 전역적으로 조정하려면 다음을 수행합니다.

1. IdM 웹 UI에 로그인합니다.
2. **IPA Server** (IPA 서버)를 클릭합니다.



3. **IPA Server**(IPA 서버) 탭에서 **Configuration**(구성)을 클릭합니다.
4. **Search Options** (검색 옵션) 영역에 필요한 값을 설정합니다.
기본값은 다음과 같습니다.
 - 검색 크기 제한: 100개 항목
 - 검색 시간 제한: 2초
5. 페이지 상단에 있는 **Save**(저장)를 클릭합니다.

The screenshot shows the Red Hat Identity Management web interface. The top navigation bar includes 'Identity', 'Policy', 'Authentication', 'Network Services', 'IPA Server', and 'API Browser'. The 'Configuration' tab is selected. Below the navigation bar, there are buttons for 'Refresh', 'Revert', and 'Save'. The 'Save' button is highlighted with a red box. The main content area is divided into two sections: 'Search Options' and 'User Options'. The 'Search Options' section has a 'Search size limit' of 50 and a 'Search time limit' of 4. The 'User Options' section has a 'User search fields' of 'uid,givenname,sn,telephonenumber,ou,title' and a 'Default e-mail domain' of 'ldm.example.com'.

RED HAT IDENTITY MANAGEMENT Administrator

Identity Policy Authentication Network Services IPA Server

Role-Based Access Control ID Ranges Realm Domains Topology API Browser Configuration

Configuration

Refresh Revert Save

Search Options

Search size limit * 50 Undo

Search time limit * 4 Undo

User Options

User search fields * uid,givenname,sn,telephonenumber,ou,title

Default e-mail domain ldm.example.com

값을 저장한 후 항목을 검색하고 결과를 확인합니다.

5장. 웹 브라우저에서 IDM 웹 UI에 액세스

다음 섹션에서는 IdM(ID 관리) 웹 UI에 대한 개요를 제공하고 액세스하는 방법을 설명합니다.

5.1. IDM 웹 UI란 무엇입니까

IdM(Identity Management) 웹 UI는 IdM 관리를 위한 웹 애플리케이션으로, IdM 명령줄 툴의 그래픽 대안입니다.

IdM 웹 UI에 다음과 같이 액세스할 수 있습니다.

- **IdM 사용자:** IdM 서버의 사용자에게 부여된 권한에 따라 제한된 작업 집합입니다. 기본적으로 활성 IdM 사용자는 IdM 서버에 로그인하여 자체 계정을 구성할 수 있습니다. 다른 사용자 또는 IdM 서버 설정의 설정을 변경할 수 없습니다.
- **관리자:** IdM 서버에 대한 전체 액세스 권한.
- **Active Directory 사용자:** 사용자에게 부여된 권한에 따라 작업 집합입니다. 이제 Active Directory 사용자가 Identity Management의 관리자가 될 수 있습니다. 자세한 내용은 [AD users to manage IdM](#)을 참조하십시오.

5.2. 웹 UI에 액세스하는 데 지원되는 웹 브라우저

IdM(Identity Management)은 웹 UI에 연결하기 위해 다음 브라우저를 지원합니다.

- Mozilla Firefox 38 이상
- Google Chrome 46 이상

5.3. 웹 UI에 액세스

다음 절차에서는 암호를 사용하여 IdM(Identity Management) 웹 UI에 처음 로그인하는 방법을 설명합니다.

처음 로그인한 후 인증할 IdM 서버를 구성할 수 있습니다.

- **Kerberos 티켓**
자세한 내용은 [6.1절. "Identity Management의 Kerberos 인증"](#)의 내용을 참조하십시오.
- **스마트 카드**
자세한 내용은 [스마트 카드 인증을 위한 IdM 서버 구성](#)을 참조하십시오.
- **한 번만 암호(OTP) 메시지-암호 및 Kerberos 인증과 결합될 수 있습니다.**
자세한 내용은 [7.2절. "Identity Management에서 한 번만 암호\(OTP\) 인증"](#)의 내용을 참조하십시오.

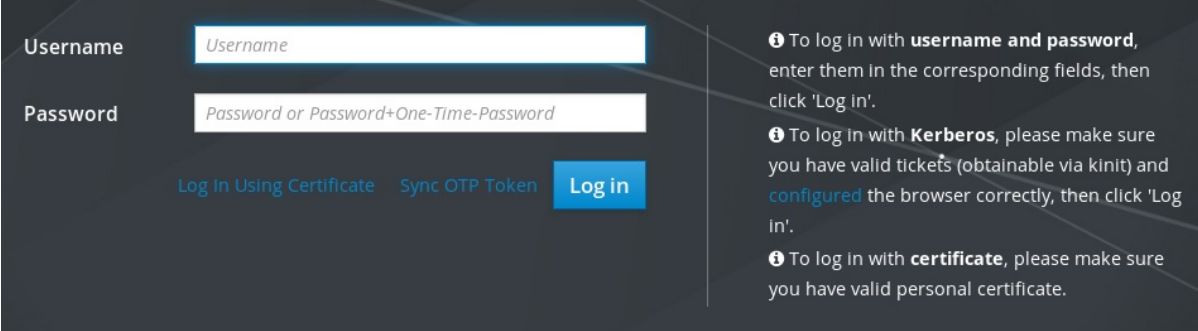
절차

1. IdM 서버 URL을 브라우저 주소 표시줄에 입력합니다. 이름은 다음 예와 유사하게 표시됩니다.

`https://server.example.com`

IdM 서버의 DNS 이름을 사용하여 **server.example.com** 을 변경하면 됩니다.

그러면 브라우저에서 IdM 웹 UI 로그인 화면이 열립니다.



IdM 웹 UI 로그인 화면. Username과 Password 필드가 있고, Log in Using Certificate, Sync OTP Token, Log in 버튼이 있습니다. 오른쪽에 로그인 방법 안내가 있습니다.

Username

Password

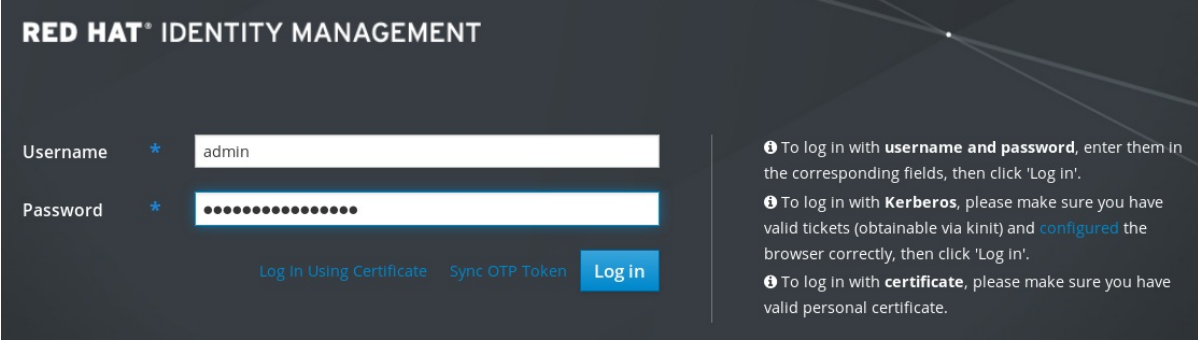
[Log in Using Certificate](#) [Sync OTP Token](#) [Log in](#)

❗ To log in with **username and password, enter them in the corresponding fields, then click 'Log in'.**

❗ To log in with **Kerberos, please make sure you have valid tickets (obtainable via kinit) and **configured** the browser correctly, then click 'Log in'.**

❗ To log in with **certificate, please make sure you have valid personal certificate.**

- 서버가 응답하지 않거나 로그인 화면이 열려 있지 않으면 연결 중인 IdM 서버의 DNS 설정을 확인합니다.
 - 자체 서명 인증서를 사용하는 경우 브라우저에서 경고를 발행합니다. 인증서를 확인하고 로그인을 진행하기 위해 보안 예외를 수락합니다.
보안 예외를 방지하려면 인증 기관에서 서명한 인증서를 설치합니다.
2. 웹 UI 로그인 화면에서 IdM 서버 설치 중에 추가한 관리자 계정 자격 증명을 입력합니다.
자세한 내용은 [ID 관리 서버 설치에서 참조하십시오](#). 통합 DNS를 사용하여 CA 를 통합했습니다.
- 개인 계정 자격 증명을 입력하거나 IdM 서버에 이미 입력한 경우에도 입력할 수 있습니다.



IdM 웹 UI 로그인 화면. Username과 Password 필드가 있고, Log in Using Certificate, Sync OTP Token, Log in 버튼이 있습니다. 오른쪽에 로그인 방법 안내가 있습니다.

Username *

Password *

[Log in Using Certificate](#) [Sync OTP Token](#) [Log in](#)

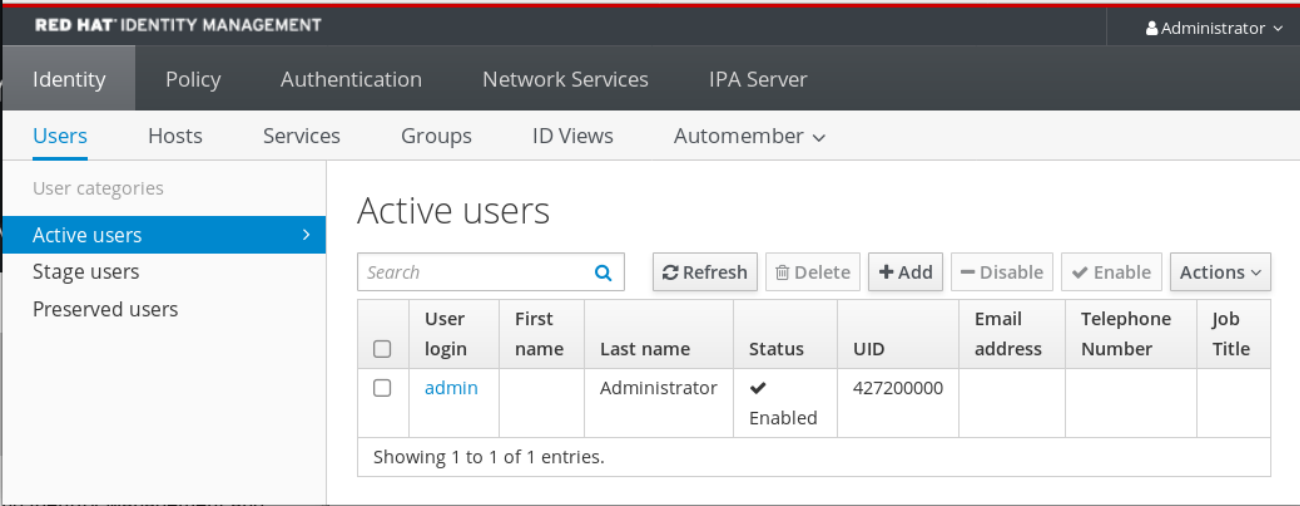
❗ To log in with **username and password, enter them in the corresponding fields, then click 'Log in'.**

❗ To log in with **Kerberos, please make sure you have valid tickets (obtainable via kinit) and **configured** the browser correctly, then click 'Log in'.**

❗ To log in with **certificate, please make sure you have valid personal certificate.**

3. **Log in** (로그인)을 클릭합니다.

로그인에 성공한 후 IdM 서버 구성을 시작할 수 있습니다.



IdM 웹 UI 관리자 화면. Identity, Policy, Authentication, Network Services, IPA Server 탭이 있습니다. Users 탭이 선택되어 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

Active users

Search [Refresh](#) [Delete](#) [+ Add](#) [- Disable](#) [✓ Enable](#) [Actions ▾](#)

☐	User login	First name	Last name	Status	UID	Email address	Telephone Number	Job Title
☐	admin		Administrator	✓ Enabled	427200000			

Showing 1 to 1 of 1 entries.

6장. 웹 UI에서 IDM에 로그인: KERBEROS 티켓 사용

다음 섹션에서는 IdM 웹 UI에 Kerberos 로그인을 활성화하고 Kerberos 인증을 사용하여 IdM에 액세스하는 데 사용하는 환경의 초기 구성에 대해 설명합니다.

사전 요구 사항

- 네트워크 환경에 IdM 서버가 설치되어 있습니다
자세한 내용은 [Red Hat Enterprise Linux 8에서 Identity Management 설치](#)를 참조하십시오.

6.1. IDENTITY MANAGEMENT의 KERBEROS 인증

IdM(Identity Management)은 Kerberos 프로토콜을 사용하여 SSO(Single Sign-On)를 지원합니다. SSO(Single Sign-On) 인증을 사용하면 올바른 사용자 이름과 암호를 한 번만 제공할 수 있으며 시스템에서 자격 증명을 다시 요구하지 않고 Identity Management 서비스에 액세스할 수 있습니다.

IdM 서버는 DNS 및 인증서 설정이 올바르게 구성된 경우 설치 직후 Kerberos 인증을 제공합니다. 자세한 내용은 [Identity Management 설치](#)를 참조하십시오.

호스트에서 Kerberos 인증을 사용하려면 다음을 설치합니다.

- IdM 클라이언트
자세한 내용은 [Identity Management 클라이언트 설치 준비](#)를 참조하십시오.
- krb5conf 패키지

6.2. KINIT 를 사용하여 IDM에 수동으로 로그인합니다.

이 절차에서는 **kinit** 유틸리티를 사용하여 IdM(Identity Management) 환경에 수동으로 인증하는 방법을 설명합니다. **kinit** 유틸리티는 IdM 사용자를 대신하여 Kerberos 티켓 부여 티켓(TGT)을 가져와 캐시합니다.



참고

초기 Kerberos TGT를 삭제했거나 만료된 경우에만 이 절차를 사용하십시오. IdM 사용자로 로컬 시스템에 로그인할 때도 IdM에 자동으로 로그인됩니다. 즉, 로그인한 후에는 **kinit** 유틸리티를 사용하여 IdM 리소스에 액세스할 필요가 없습니다.

절차

1. IdM에 로그인하려면

- 로컬 시스템에 현재 로그인한 사용자의 사용자 이름 아래에 사용자 이름을 지정하지 않고 **kinit** 를 사용합니다. 예를 들어 로컬 시스템에서 **example_user** 로 로그인한 경우 다음을 수행합니다.

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

로컬 사용자의 사용자 이름이 IdM의 사용자 항목과 일치하지 않으면 인증 시도가 실패합니다.

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

- 로컬 사용자 이름에 해당하지 않는 Kerberos 주체를 사용하여 필수 사용자 이름을 **kinit** 유틸리티에 전달합니다. 예를 들어 **admin** 사용자로 로그인하려면 다음을 수행합니다.

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

- 선택적으로 로그인에 성공했는지 확인하려면 **klist** 유틸리티를 사용하여 캐시된 TGT를 표시합니다. 다음 예에서 캐시에는 **example_user** 주체의 티켓이 포함되어 있습니다. 즉, 이 특정 호스트에서는 현재 **example_user** 만 IdM 서비스에 액세스할 수 있습니다.

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

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

6.3. KERBEROS 인증을 위한 브라우저 구성

Kerberos 티켓을 사용하여 인증을 활성화하려면 브라우저 구성이 필요할 수 있습니다.

다음 단계는 IdM 도메인에 액세스하기 위한 Kerberos 협상을 지원하는 데 도움이 됩니다.

각 브라우저는 Kerberos를 다른 방식으로 지원하므로 다른 설정이 필요합니다. IdM 웹 UI에는 다음 브라우저에 대한 지침이 포함되어 있습니다.

- Firefox
- Chrome

절차

- 웹 브라우저에서 IdM 웹 UI 로그인 대화 상자를 엽니다.
- Web UI 로그인 화면에서 브라우저 구성에 대한 링크를 클릭합니다.

RED HAT® IDENTITY MANAGEMENT

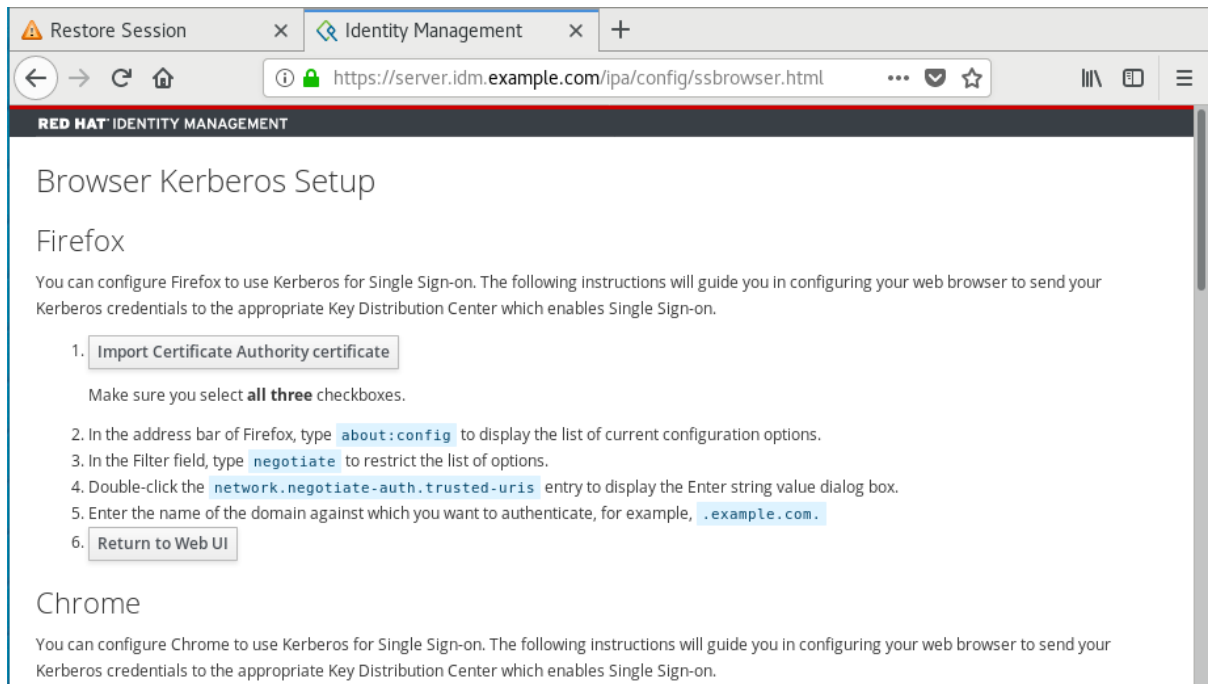
Username:

Password:

[Log in Using Certificate](#) [Sync OTP Token](#) [Log in](#)

ⓘ To log in with **username and password**, enter them in the corresponding fields, then click 'Log in'.
 ⓘ To log in with **Kerberos**, please make sure you have valid tickets (obtainable via kinit) and **configured** the browser correctly, then click 'Log in'.
 ⓘ To log in with **certificate**, please make sure you have valid personal certificate.

- 구성 페이지의 단계를 따릅니다.



설정 후 IdM 웹 UI로 돌아가서 **Log in** (로그인)을 클릭합니다.

6.4. KERBEROS 티켓을 사용하여 웹 UI에 로그인

이 절차에서는 Kerberos 티켓 부여 티켓(TGT)을 사용하여 IdM 웹 UI에 로그인하는 방법을 설명합니다.

TGT는 미리 정의된 시간에 만료됩니다. 기본 시간 간격은 24시간이며 IdM 웹 UI에서 변경할 수 있습니다.

시간 간격이 만료된 후 티켓을 갱신해야 합니다.

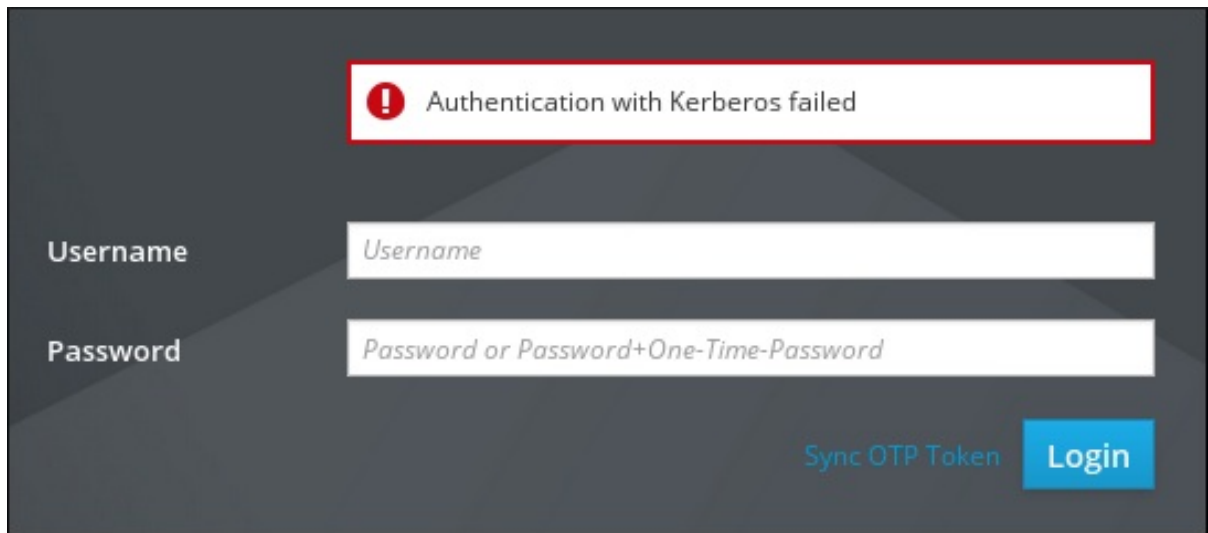
- kinit 명령 사용.
- 웹 UI 로그인 대화 상자에서 IdM 로그인 자격 증명 사용.

절차

- IdM 웹 UI를 엽니다.
Kerberos 인증이 올바르게 작동하고 유효한 티켓이 있으면 자동으로 인증되고 웹 UI가 열립니다.

티켓이 만료되면 먼저 자격 증명을 사용하여 인증해야 합니다. 그러나 다음에 로그인 대화 상자를 열지 않고 IdM 웹 UI가 자동으로 열립니다.

Authentication with Kerberos failed 라는 오류 메시지가 표시되면 브라우저가 Kerberos 인증에 맞게 구성되었는지 확인합니다. 6.3절. “Kerberos 인증을 위한 브라우저 구성”을 참조하십시오.



The screenshot shows a login form with a dark background. At the top, a red-bordered box contains a warning icon and the text "Authentication with Kerberos failed". Below this, there are two input fields: "Username" with a placeholder "Username" and "Password" with a placeholder "Password or Password+One-Time-Password". At the bottom right, there is a blue button labeled "Login" and a link labeled "Sync OTP Token".

6.5. KERBEROS 인증을 위한 외부 시스템 구성

이 섹션에서는 IdM(Identity Management) 사용자가 Kerberos 자격 증명을 사용하여 외부 시스템에서 IdM에 로그인할 수 있도록 외부 시스템을 구성하는 방법에 대해 설명합니다.

외부 시스템에서 Kerberos 인증을 활성화하는 것은 인프라에 여러 영역 또는 중복되는 도메인이 포함된 경우 특히 유용합니다. 시스템이 **ipa-client-install** 을 통해 IdM 도메인에 등록되지 않은 경우에도 유용합니다.

IdM 도메인의 멤버가 아닌 시스템에서 IdM에 대한 Kerberos 인증을 활성화하려면 외부 시스템에 IdM별 Kerberos 구성 파일을 정의합니다.

사전 요구 사항

- **krb5-workstation** 패키지는 외부 시스템에 설치됩니다.
패키지가 설치되어 있는지 여부를 확인하려면 다음 CLI 명령을 사용합니다.

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

절차

1. IdM 서버에서 외부 시스템으로 **/etc/krb5.conf** 파일을 복사합니다. 예를 들면 다음과 같습니다.

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



주의

외부 시스템의 기존 **krb5.conf** 파일을 덮어쓰지 마십시오.

2. 외부 시스템에서 복사된 IdM Kerberos 구성 파일을 사용하도록 터미널 세션을 설정합니다.

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

KRB5_CONFIG 변수는 로그아웃할 때까지 일시적으로만 존재합니다. 이 손실을 방지하려면 다른 파일 이름으로 변수를 내보냅니다.

3. **/etc/krb5.conf.d/** 디렉토리에서 외부 시스템으로 Kerberos 구성 조각을 복사합니다.
4. [6.3절. "Kerberos 인증을 위한 브라우저 구성"](#)에 설명된 대로 외부 시스템에서 브라우저를 구성합니다.

이제 외부 시스템의 사용자는 **kinit** 유틸리티를 사용하여 IdM 서버에 대해 인증할 수 있습니다.

6.6. ACTIVE DIRECTORY 사용자를 위한 웹 UI 로그인

Active Directory 사용자에게 대해 Web UI 로그인을 활성화하려면 기본 신뢰 보기에서 각 Active Directory 사용자에게 대한 ID 재정의의를 정의합니다. 예를 들면 다음과 같습니다.

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

7장. 일회성 암호를 사용하여 ID 관리 웹 UI에 로그인

IdM 웹 UI에 대한 액세스는 여러 가지 방법으로 보안을 할 수 있습니다. 기본 항목은 암호 인증입니다.

암호 인증의 보안을 강화하려면 두 번째 단계를 추가하고 자동으로 생성된 일회성 암호(OTP)가 필요합니다. 가장 일반적인 용도는 사용자 계정과 하드웨어 또는 소프트웨어 토큰으로 생성된 한 번만 암호를 결합하는 것입니다.

다음 섹션은 다음을 수행하는 데 도움이 됩니다.

- IdM에서 OTP 인증이 작동하는 방식을 이해합니다.
- IdM 서버에서 OTP 인증을 구성합니다.
- OTP 토큰을 생성하고 휴대폰의 FreeOTP 애플리케이션과 동기화합니다.
- 사용자 암호와 일회성 암호의 조합을 사용하여 IdM 웹 UI에 인증합니다.
- 웹 UI에서 토큰을 다시 동기화합니다.

7.1. 사전 요구 사항

- [웹 브라우저에서 IdM 웹 UI에 액세스](#)

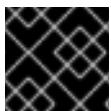
7.2. IDENTITY MANAGEMENT에서 한 번만 암호(OTP) 인증

일회성 암호는 인증 보안에 추가 단계를 수행합니다. 인증은 암호 + 자동으로 생성된 일회성 암호를 사용합니다.

한 번만 암호를 생성하려면 하드웨어 또는 소프트웨어 토큰을 사용할 수 있습니다. IdM은 소프트웨어 및 하드웨어 토큰을 모두 지원합니다.

ID 관리는 다음 두 가지 표준 OTP 메커니즘을 지원합니다.

- HMAC 기반 일회성 암호(HOTP) 알고리즘은 카운터를 기반으로 합니다. HMAC는 Hashed Message Authentication Code를 나타냅니다.
- TOTP(Time-Based One-Time Password) 알고리즘은 시간 기반 이동 요소를 지원하기 위한 HOTP의 확장입니다.



중요

IdM은 Active Directory 신뢰 사용자를 위한 OTP 로그인을 지원하지 않습니다.

7.3. 웹 UI에서 일회용 암호 활성화

The IdM Web UI allows you to configure hardware or software device to generate one-time passwords.

한 번 로그인 대화 상자의 전용 필드에 일반적인 암호를 입력한 직후에 암호가 입력됩니다.

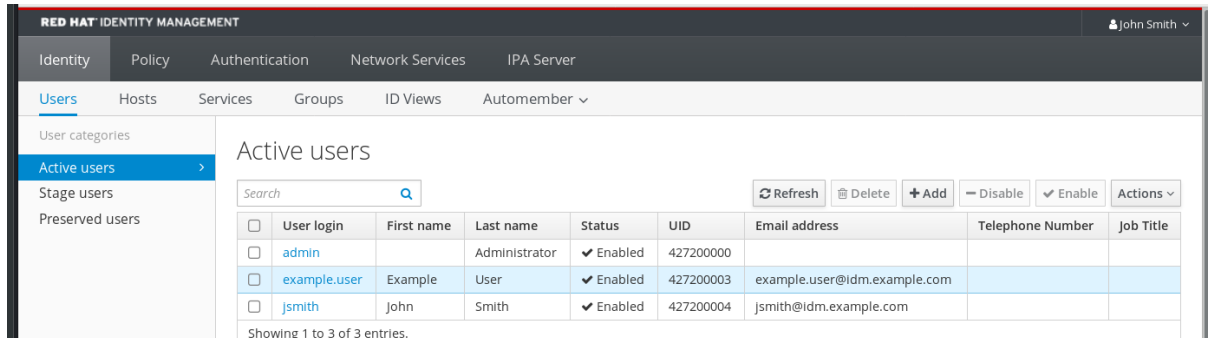
관리자만 사용자 설정에서 OTP 인증을 활성화할 수 있습니다.

사전 요구 사항

- 관리 권한

절차

1. 사용자 이름과 암호를 사용하여 IdM 웹 UI에 로그인합니다.
2. ID → 사용자 → 활성 사용자탭을 엽니다.



3. 사용자 이름을 클릭하여 사용자 설정을 엽니다.
4. 사용자 인증 유형에서 2단계 인증(암호 + OTP)을 선택합니다.
5. 저장을 클릭합니다.

이때 IdM 서버에서 OTP 인증이 활성화됩니다.

이제 사용자 또는 사용자 자체에서 새 토큰 ID를 사용자 계정에 할당해야 합니다.

7.4. 웹 UI에서 OTP 토큰 추가

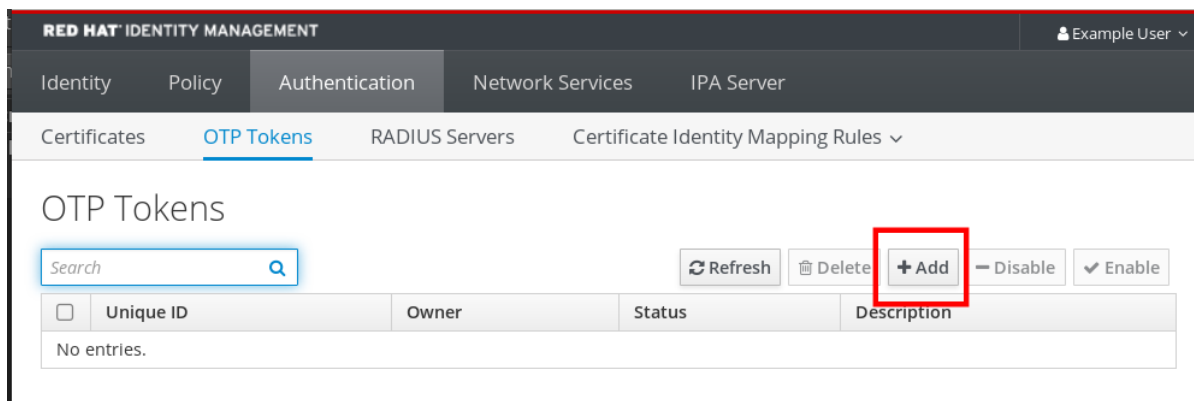
다음 섹션에서는 IdM 웹 UI 및 소프트웨어 토큰 생성기에 토큰을 추가하는 데 도움이 됩니다.

사전 요구 사항

- IdM 서버의 활성 사용자 계정.
- 관리자가 IdM 웹 UI의 특정 사용자 계정에 대해 OTP를 활성화했습니다.
- OTP 토큰을 생성하는 소프트웨어 장치입니다(예: FreeOTP).

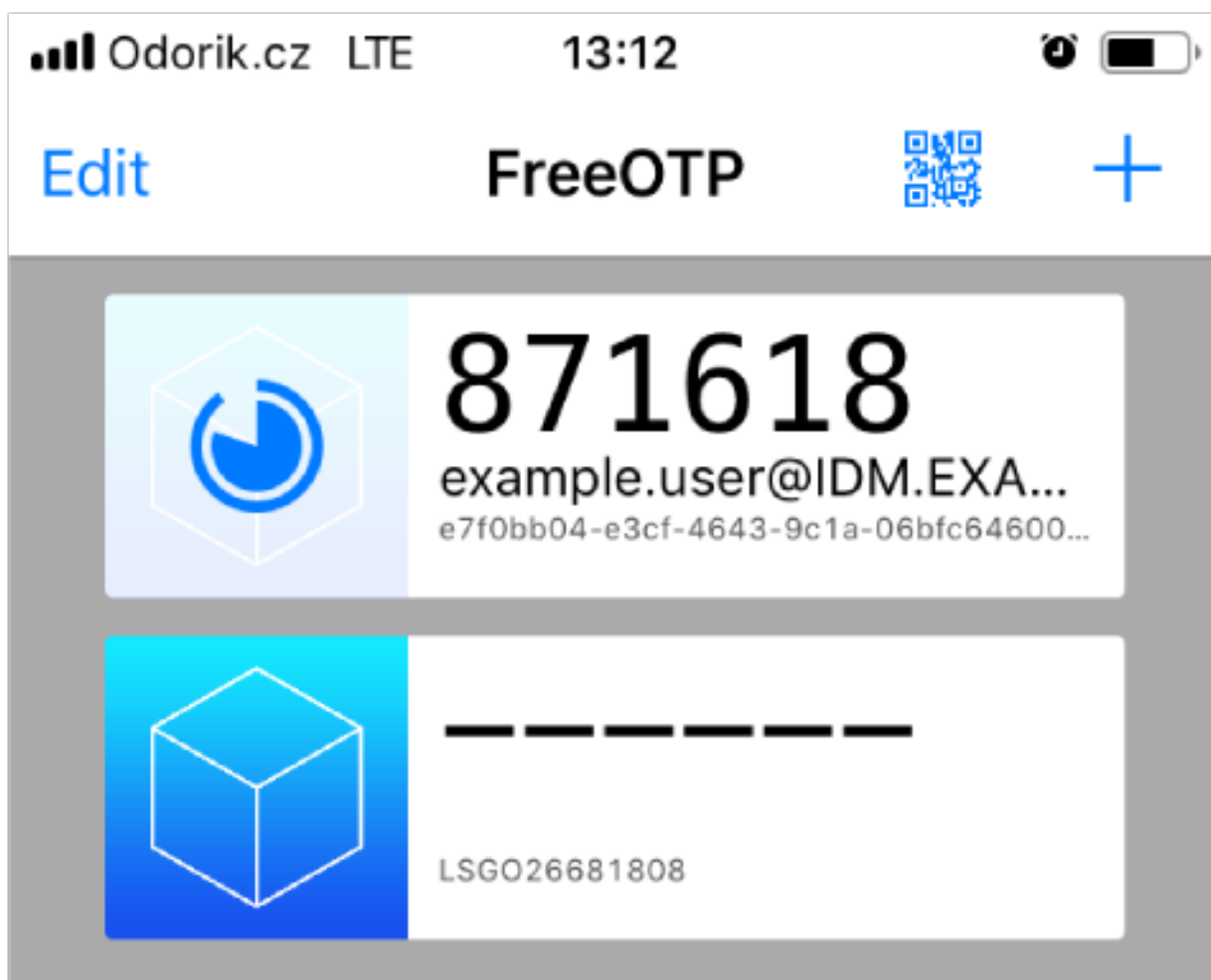
절차

1. 사용자 이름과 암호를 사용하여 IdM 웹 UI에 로그인합니다.
2. 휴대폰에서 토큰을 생성하려면 인증 → OTP 토큰 탭을 엽니다.
3. 추가를 클릭합니다.



4. **OTP 토큰 추가 대화 상자**에서 모두 채워지지 않은 상태로 두고 **Add(추가)**를 클릭합니다.
이 단계에서 IdM 서버는 서버에서 기본 매개 변수를 사용하여 토큰을 생성하고, 코드로 페이지를 엽니다.
5. 스마트폰 코드를 휴대폰에 복사합니다.
6. **OK(확인)**를 클릭하여 create code를 닫습니다.

이제 한 번 암호를 생성하여 IdM 웹 UI에 로그인할 수 있습니다.



7.5. 일회성 암호로 웹 UI에 로그인

이 절차에서는 일회성 암호(OTP)를 사용하여 IdM 웹 UI에 처음 로그인하는 방법을 설명합니다.

사전 요구 사항

- OTP 인증에 사용 중인 사용자 계정의 ID 관리 서버에서 OTP 구성이 활성화됩니다. 관리자뿐 아니라 사용자 자체도 OTP를 활성화할 수 있습니다.
OTP 구성을 활성화하려면 [7.3절. "웹 UI에서 일회용 암호 활성화"](#)를 참조하십시오.
- 구성된 OTP 토큰을 생성하는 하드웨어 또는 소프트웨어 장치입니다.

절차

1. Identity Management 로그인 화면에서 사용자 이름 또는 IdM 서버 관리자 계정의 사용자 이름을 입력합니다.
2. 위에 입력한 사용자 이름의 암호를 추가합니다.
3. 장치에서 한 번만 암호를 생성합니다.
4. 암호 바로 뒤에(공백 없음) 한 번 암호를 입력합니다.
5. **Log in** (로그인)을 클릭합니다.
인증에 실패하면 OTP 토큰을 동기화합니다.

CA에서 자체 서명된 인증서를 사용하는 경우 브라우저에서 경고를 발행합니다. 인증서를 확인하고 로그인을 진행하기 위해 보안 예외를 수락합니다.

IdM 웹 UI가 열려 있지 않으면 ID 관리 서버의 DNS 구성을 확인합니다.

로그인에 성공하면 IdM 웹 UI가 나타납니다.

The screenshot shows the Red Hat Identity Management web UI. The top navigation bar includes 'Identity', 'Policy', 'Authentication', 'Network Services', and 'IPA Server'. The 'Users' tab is selected in the left sidebar. The main content area is titled 'Active users' and contains a table with the following data:

	User login	First name	Last name	Status	UID	Email address	Telephone Number	Job Title
☐	admin		Administrator	✓ Enabled	427200000			

Below the table, it says 'Showing 1 to 1 of 1 entries.' Above the table, there are buttons for 'Search', 'Refresh', 'Delete', '+ Add', '- Disable', '✓ Enable', and 'Actions'.

7.6. 웹 UI를 사용하여 OTP 토큰 동기화

OTP(One Time Password)의 로그인에 실패하면 OTP 토큰이 올바르게 동기화되지 않습니다.

다음 텍스트는 토큰 재동기화에 대해 설명합니다.

사전 요구 사항

- 로그인 화면이 열립니다.
- OTP 토큰을 생성하는 장치입니다.

설치

1. IdM 웹 UI 로그인 화면에서 **Sync OTP Token**(OTP 토큰 동기화)을 클릭합니다.

RED HAT® IDENTITY MANAGEMENT

Username

Password

Log In Using Certificate: **Sync OTP Token** Log in

2. 로그인 화면에서 사용자 이름과 ID 관리 암호를 입력합니다.
3. 1회 암호를 생성하고 첫 번째 OTP 필드에 입력합니다.
4. 한 번 더 생성하고 Second OTP(두 번째 OTP) 필드에 입력합니다.
5. 필요한 경우 토큰 ID를 입력합니다.

RED HAT® IDENTITY MANAGEMENT

Username *

Password *

First OTP *

Second OTP *

Token ID

Cancel Sync OTP Token

6. **Sync OTP Token**(OTP 토큰 동기화)을 클릭합니다.

동기화가 완료되면 IdM 서버에 로그인할 수 있습니다.

7.7. 만료된 암호 변경

ID 관리 관리자는 다음 로그인 시 암호를 변경해야 할 수 있습니다. 즉, 암호를 변경할 때까지 IdM 웹 UI에 성공적으로 로그인할 수 없습니다.

웹 UI에 처음 로그인하는 동안 암호 만료가 발생할 수 있습니다.

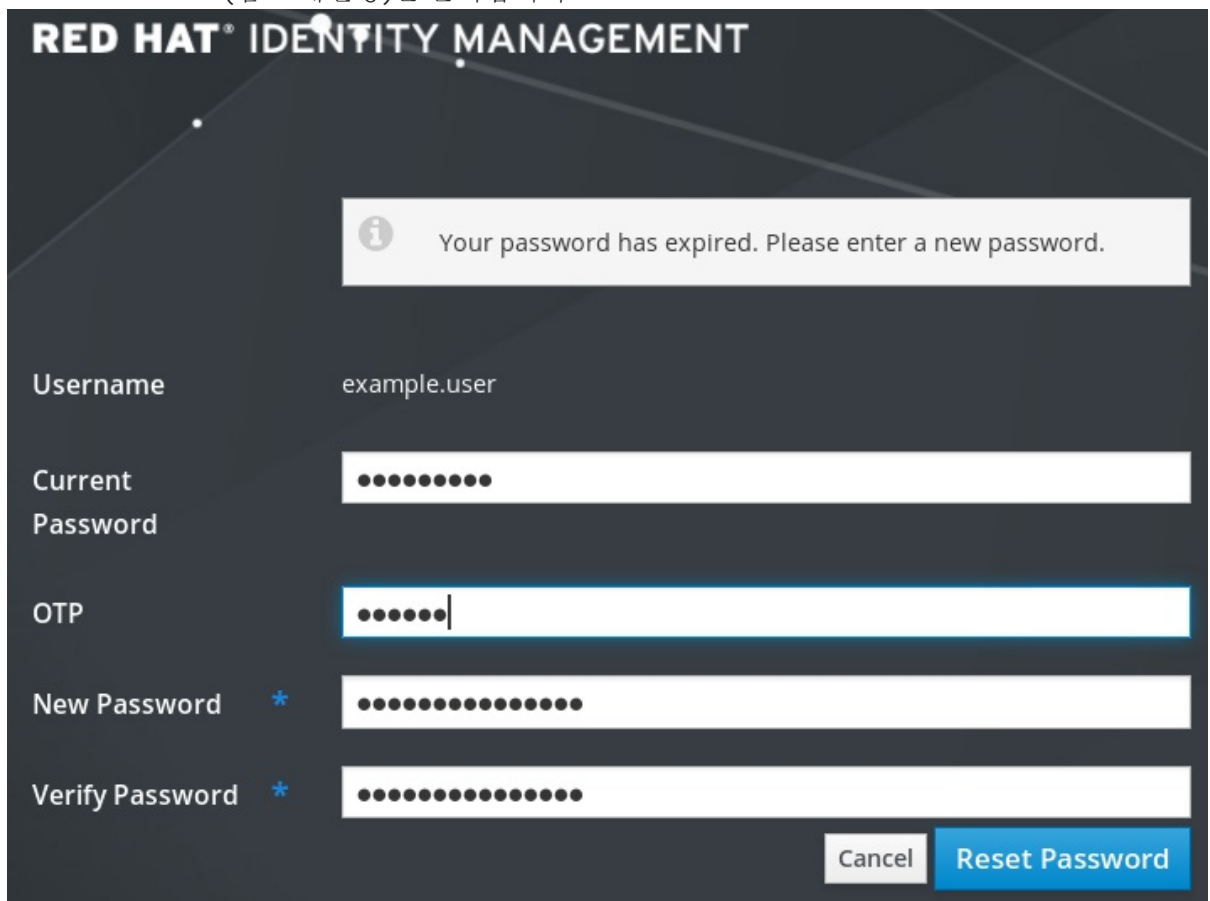
만료 암호 대화 상자가 나타나면 절차에 있는 지침을 따릅니다.

사전 요구 사항

- 로그인 화면이 열립니다.
- IdM 서버에 대한 활성 계정입니다.

절차

1. 암호 만료 로그인 화면에서 사용자 이름을 입력합니다.
2. 위에 입력한 사용자 이름의 암호를 추가합니다.
3. 일회성 암호 인증을 사용하는 경우 OTP 필드에서 일회성 암호를 생성합니다.
OTP 인증을 활성화하지 않은 경우 필드를 비워 둡니다.
4. 확인을 위해 새 암호를 두 번 입력합니다.
5. **Reset Password** (암호 재설정)를 클릭합니다.



The image shows a 'RED HAT IDENTITY MANAGEMENT' dialog box for password reset. At the top, a message states: 'Your password has expired. Please enter a new password.' Below this, there are five input fields: 'Username' (containing 'example.user'), 'Current Password' (masked with dots), 'OTP' (containing six dots and a cursor), 'New Password' (marked with a red asterisk and containing ten dots), and 'Verify Password' (marked with a red asterisk and containing ten dots). At the bottom right, there are two buttons: 'Cancel' and 'Reset Password'.

암호가 성공적으로 변경되면 일반적인 로그인 대화 상자가 표시됩니다. 새 암호로 로그인합니다.

8장. IDM의 SSSD를 사용한 인증 문제 해결

IdM(Identity Management) 환경의 인증에는 많은 구성 요소가 포함됩니다.

IdM 클라이언트에서 다음을 수행합니다.

- SSSD 서비스.
- NSS(Name Services Switch).
- PAM(Pluggable Authentication Module).

IdM 서버에서 다음을 수행합니다.

- SSSD 서비스.
- IdM 디렉터리 서버.
- IdM Kerberos KDC(키 배포 센터).

AD(Active Directory) 사용자로 인증하는 경우:

- AD 도메인 컨트롤러의 디렉터리 서버.
- AD 도메인 컨트롤러의 Kerberos 서버.

사용자를 인증하려면 SSSD 서비스를 사용하여 다음 기능을 수행할 수 있어야 합니다.

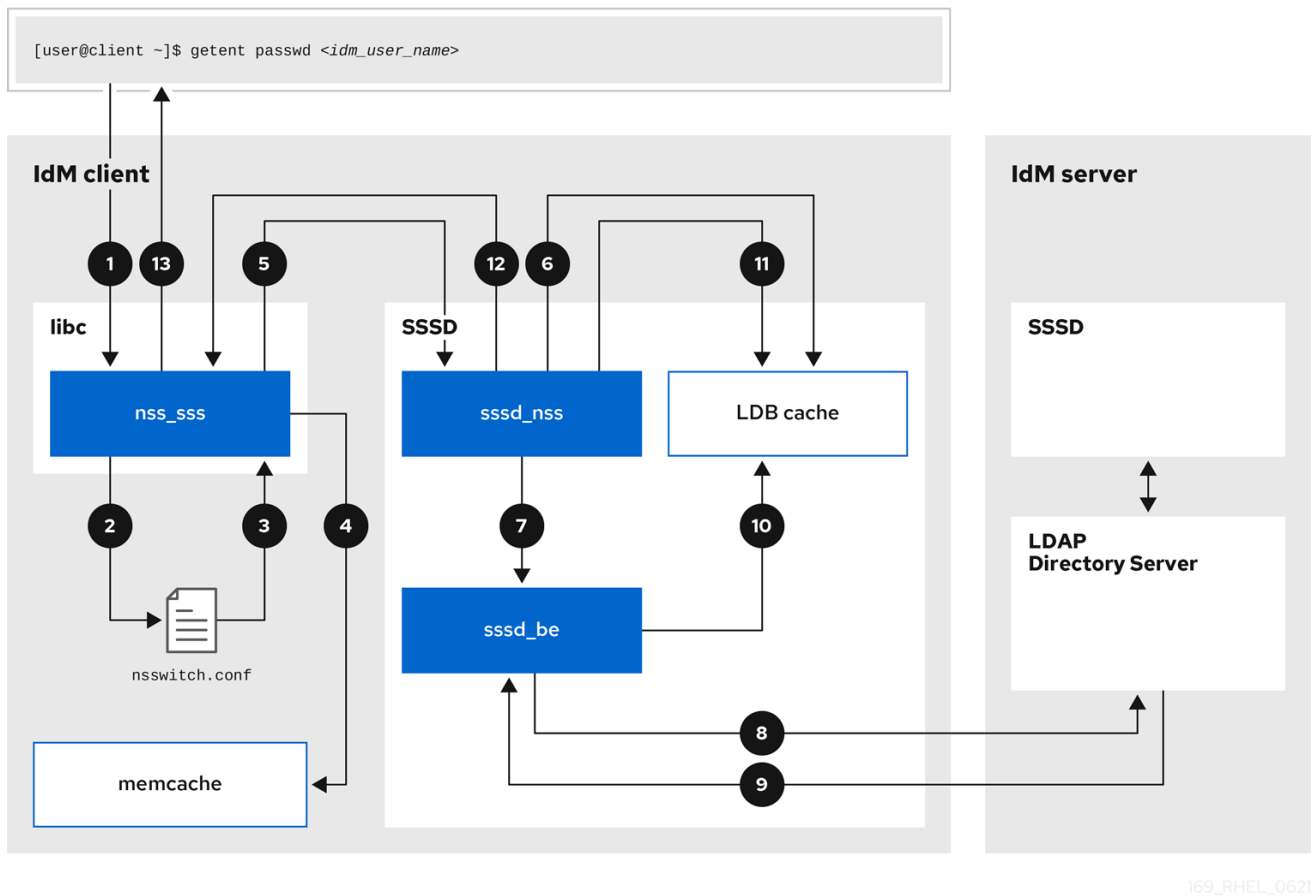
- 인증 서버에서 사용자 정보를 검색합니다.
- 사용자에게 자격 증명을 요청하고 해당 자격 증명을 인증 서버에 전달한 다음 결과를 처리합니다.

다음 섹션에서는 사용자 정보를 저장하는 SSSD 서비스와 서버 간에 정보가 이동하는 방법에 대해 설명하여 환경에서 인증 시도 실패 문제를 해결할 수 있습니다.

1. [SSSD를 사용하여 IdM 사용자 정보를 검색할 때 데이터 흐름](#)
2. [SSSD를 사용하여 AD 사용자 정보를 검색할 때 데이터 흐름](#)
3. [IdM에서 SSSD를 사용하여 사용자로 인증할 때 데이터 흐름](#)
4. [인증 문제 범위 줄이기](#)
5. [SSSD 로그 파일 및 로깅 수준](#)
6. [sssd.conf 파일에서 SSSD에 대한 세부 로깅 활성화](#)
7. [sssctl 명령을 사용하여 SSSD에 대한 세부 로깅 활성화](#)
8. [IdM 서버의 인증 문제 해결을 위해 SSSD 서비스에서 디버깅 로그 수집](#)
9. [IdM 클라이언트의 인증 문제를 해결하기 위해 SSSD 서비스에서 디버깅 로그 수집](#)
10. [SSSD 백엔드에서 클라이언트 요청 추적](#)

8.1. SSSD를 사용하여 IDM 사용자 정보를 검색할 때 데이터 흐름

다음 다이어그램은 **getent passwd <idm_user_name>** 명령을 사용하여 IdM 사용자 정보를 요청하는 동안 IdM 클라이언트와 IdM 서버 간의 정보 흐름을 간소화한 것입니다.



169_RHEL_0621

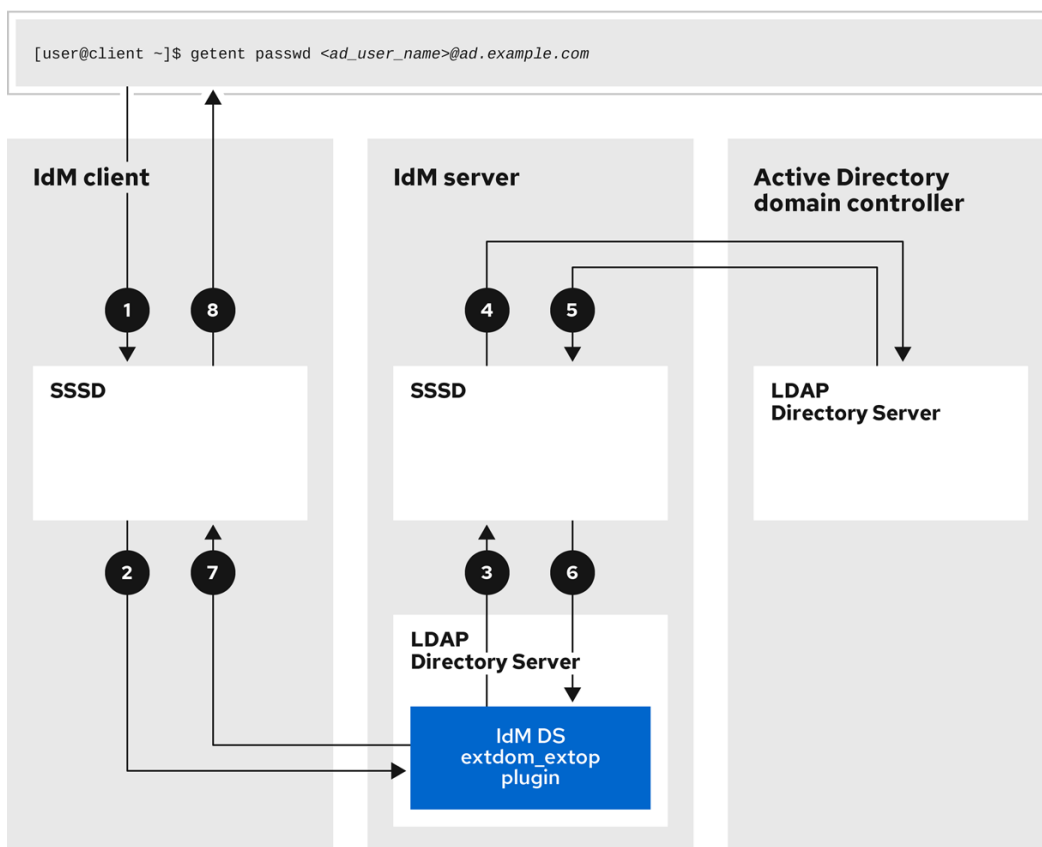
1. **getent** 명령은 **libc** 라이브러리의 **getpwnam** 호출을 트리거합니다.
2. **libc** 라이브러리는 **/etc/nsswitch.conf** 구성 파일을 참조하여 사용자 정보를 제공하는 서비스를 확인하고 SSSD 서비스에 대한 항목 **sss** 를 검색합니다.
3. **libc** 라이브러리는 **nss_sss** 모듈을 엽니다.
4. **nss_sss** 모듈은 메모리 매핑된 캐시에서 사용자 정보를 확인합니다. 데이터가 캐시에 있으면 **nss_sss** 모듈에서 해당 데이터가 반환됩니다.
5. 사용자 정보가 메모리 매핑 캐시에 없는 경우 요청이 SSSD **sssd_nss** 응답자 프로세스에 전달됩니다.
6. SSSD 서비스는 캐시를 확인합니다. 데이터가 캐시에 있고 유효한 경우 **sssd_nss** 응답자가 캐시에서 데이터를 읽고 애플리케이션에 반환합니다.
7. 데이터가 캐시에 없거나 만료된 경우 **sssd_nss** 응답자는 적절한 백엔드 프로세스를 쿼리하고 응답을 기다립니다. SSSD 서비스는 **sssd.conf** 구성 파일의 **id_provider=ipa** 설정을 통해 활성화되는 IdM 환경에서 IPA 백엔드를 사용합니다.
8. **sssd_be** 백엔드 프로세스는 IdM 서버에 연결하고 IdM LDAP 디렉터리 서버에서 정보를 요청합니다.
9. IdM 서버의 SSSD 백엔드는 IdM 클라이언트의 SSSD 백엔드 프로세스에 응답합니다.

10. 클라이언트의 SSSD 백엔드는 결과 데이터를 SSSD 캐시에 저장하고 응답자 프로세스에 캐시가 업데이트되었음을 경고합니다.
11. **sssd_nss** 프론트 엔드 응답자 프로세스는 SSSD 캐시에서 정보를 검색합니다.
12. **sssd_nss** 응답자는 사용자 정보를 **nss_sss** 응답자에게 전송하여 요청을 완료합니다.
13. **libc** 라이브러리는 사용자 정보를 요청한 애플리케이션에 반환합니다.

8.2. SSSD를 사용하여 AD 사용자 정보를 검색할 때 데이터 흐름

IdM 환경과 AD(Active Directory) 도메인 간에 교차 포드 신뢰를 구축한 경우 IdM 클라이언트에서 AD 사용자 정보를 검색할 때 IdM 사용자 정보를 검색할 때 정보 흐름과 매우 유사하며 AD 사용자 데이터베이스에 연결하는 추가 단계가 있습니다.

다음 다이어그램은 사용자가 **getent passwd <ad_user_name@ad.example.com>** 명령을 사용하여 AD 사용자에게 대한 정보를 요청할 때 정보 흐름을 간소화한 것입니다. 이 다이어그램에는 **SSSD 섹션을 사용하여 IdM 사용자 정보를 검색할 때 데이터 흐름에 설명된 내부 세부 정보가 포함되지 않습니다**. IdM 클라이언트에서 SSSD 서비스, IdM 서버의 SSSD 서비스 및 AD 도메인 컨트롤러의 LDAP 데이터베이스 간 통신에 중점을 둡니다.



169_RHEL_O621

1. IdM 클라이언트는 AD 사용자 정보를 위해 로컬 SSSD 캐시를 찾습니다.
2. IdM 클라이언트에 사용자 정보가 없거나 정보가 오래된 경우 클라이언트의 SSSD 서비스는 IdM 서버의 **extdom_extop** 플러그인에 연결하여 LDAP 확장 작업을 수행하고 정보를 요청합니다.
3. IdM 서버의 SSSD 서비스는 로컬 캐시에서 AD 사용자 정보를 찾습니다.

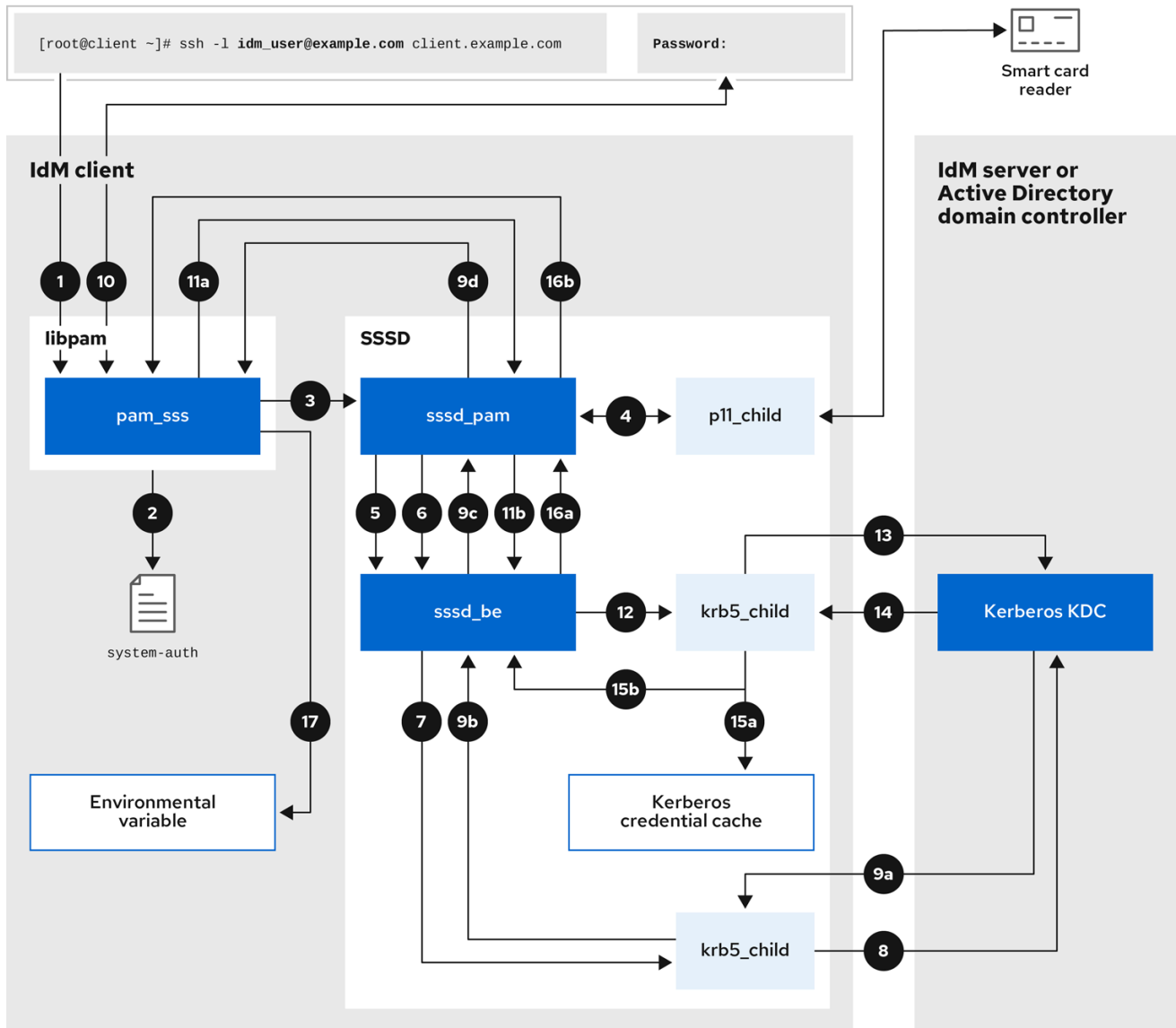
4. IdM 서버에 SSSD 캐시에 사용자 정보가 없거나 해당 정보가 오래된 경우 LDAP 검색을 수행하여 AD 도메인 컨트롤러에서 사용자 정보를 요청합니다.
5. IdM 서버의 SSSD 서비스는 AD 도메인 컨트롤러에서 AD 사용자 정보를 수신하여 해당 캐시에 저장합니다.
6. **extdom_extop** 플러그인은 IdM 서버의 SSSD 서비스에서 정보를 수신하여 LDAP 확장 작업을 완료합니다.
7. IdM 클라이언트의 SSSD 서비스는 LDAP 확장 작업에서 AD 사용자 정보를 수신합니다.
8. IdM 클라이언트는 AD 사용자 정보를 SSSD 캐시에 저장하고 정보를 요청한 애플리케이션에 반환합니다.

8.3. IDM에서 SSSD를 사용하여 사용자로 인증할 때 데이터 흐름

IdM 서버 또는 클라이언트에 사용자로 인증에는 다음 구성 요소가 포함됩니다.

- sshd 서비스와 같은 인증 요청을 시작하는 서비스입니다.
- PAM(Pluggable Authentication Module) 라이브러리 및 해당 모듈.
- SSSD 서비스, 해당 응답자 및 백엔드.
- 스마트 카드 인증이 구성된 경우 스마트 카드 리더.
- 인증 서버:
 - IdM 사용자는 IdM KDC(Kerberos Key Distribution Center)에 대해 인증됩니다.
 - AD(Active Directory) 사용자는 DC(AD 도메인 컨트롤러)에 대해 인증됩니다.

다음 다이어그램은 사용자가 명령줄의 SSH 서비스를 통해 호스트에 로컬로 로그인해야 하는 경우 정보 흐름을 단순화한 것입니다.



169_RHEL_0621

1. **ssh** 명령의 인증 시도는 **libpam** 라이브러리를 트리거합니다.
2. **libpam** 라이브러리는 인증 시도를 요청하는 서비스에 해당하는 `/etc/pam.d/` 디렉터리의 PAM 파일을 참조합니다. 로컬 호스트의 SSH 서비스를 통한 인증과 관련된 이 예제에서 **libpam** 라이브러리는 `/etc/pam.d/system-auth` 구성 파일을 확인하고 SSSD PAM에 대한 **pam_sss.so** 항목을 검색합니다.

```
auth sufficient pam_sss.so
```

3. 사용 가능한 인증 방법을 확인하기 위해 **libpam** 라이브러리는 **pam_sss** 모듈을 열고 SSSD 서비스의 **sssd_pam** PAM 응답자로 **SSS_PAM_PREAUTH** 요청을 보냅니다.
4. 스마트 카드 인증이 구성된 경우 SSSD 서비스는 임시 **p11_child** 프로세스를 생성하여 스마트 카드를 확인하고 인증서를 검색합니다.
5. 사용자에게 대해 스마트 카드 인증이 구성된 경우 **sssd_pam** 응답자가 스마트 카드의 인증서와 사용자와 일치하려고 시도합니다. **sssd_pam** 응답자는 그룹 멤버십이 액세스 제어에 영향을 줄 수 있으므로 사용자가 속한 그룹을 검색합니다.
6. **sssd_pam** 응답자는 **SSS_PAM_PREAUTH** 요청을 **sssd_be** 백엔드 응답자에게 전송하여 서버

가 지원하는 인증 방법(예: 암호 또는 2 단계 인증)을 확인합니다. SSSD 서비스에서 IPA 응답자를 사용하는 IdM 환경에서 기본 인증 방법은 Kerberos입니다. 이 예에서 사용자는 간단한 Kerberos 암호로 인증합니다.

7. **sssd_be** 응답자는 임시 **krb5_child** 프로세스를 생성합니다.
8. **krb5_child** 프로세스는 IdM 서버의 KDC에 연결하여 사용 가능한 인증 방법을 확인합니다.
9. KDC는 요청에 응답합니다.
 - a. **krb5_child** 프로세스는 응답을 평가하고 결과를 **sssd_be** 백엔드 프로세스로 다시 보냅니다.
 - b. **sssd_be** backend 프로세스는 결과를 수신합니다.
 - c. **sssd_pam** 응답자가 결과를 수신합니다.
 - d. **pam_sss** 모듈은 결과를 수신합니다.
10. 암호 인증이 사용자에게 대해 구성된 경우 **pam_sss** 모듈은 사용자에게 암호를 묻는 메시지를 표시합니다. 스마트 카드 인증이 구성된 경우 **pam_sss** 모듈은 사용자에게 스마트 카드 PIN을 요청합니다.
11. 이 모듈은 다음을 이동하는 사용자 이름과 암호를 사용하여 **SSS_PAM_AUTHENTICATE** 요청을 보냅니다.
 - a. **sssd_pam** 응답자.
 - b. **sssd_be** 백엔드 프로세스.
12. **sssd_be** 프로세스는 KDC에 연결할 임시 **krb5_child** 프로세스를 생성합니다.
13. **krb5_child** 프로세스는 KDC에서 사용자가 제공한 암호와 함께 Kerberos 티켓 부여 티켓(TGT)을 검색하려고 합니다.
14. **krb5_child** 프로세스는 인증 시도 결과를 수신합니다.
15. **krb5_child** 프로세스는 다음과 같습니다.
 - a. TGT를 자격 증명 캐시에 저장합니다.
 - b. 인증 결과를 **sssd_be** 백엔드 프로세스에 반환합니다.
16. 인증 결과는 **sssd_be** 프로세스에서 다음으로 이동합니다.
 - a. **sssd_pam** 응답자.
 - b. **pam_sss** 모듈.
17. **pam_sss** 모듈은 다른 애플리케이션에서 참조할 수 있도록 사용자 TGT 위치가 있는 환경 변수를 설정합니다.

8.4. 인증 문제 범위 줄이기

사용자를 인증하려면 사용자 정보를 저장하는 데이터베이스에서 SSSD 서비스를 사용하여 사용자 정보를 검색할 수 있어야 합니다. 다음 절차에서는 인증 프로세스의 다양한 구성 요소를 테스트하는 단계에 대해 설명하므로 사용자가 로그인할 수 없는 경우 인증 문제의 범위를 좁힐 수 있습니다.

절차

1. SSSD 서비스 및 해당 프로세스가 실행 중인지 확인합니다.

```
[root@client ~]# pstree -a | grep sssd
|-sssd -i --logger=files
|   |-sssd_be --domain implicit_files --uid 0 --gid 0 --logger=files
|   |-sssd_be --domain example.com --uid 0 --gid 0 --logger=files
|   |-sssd_ifp --uid 0 --gid 0 --logger=files
|   |-sssd_nss --uid 0 --gid 0 --logger=files
|   |-sssd_pac --uid 0 --gid 0 --logger=files
|   |-sssd_pam --uid 0 --gid 0 --logger=files
|   |-sssd_ssh --uid 0 --gid 0 --logger=files
|   |-sssd_sudo --uid 0 --gid 0 --logger=files
|-sssd_kcm --uid 0 --gid 0 --logger=files
```

2. 클라이언트가 IP 주소를 통해 사용자 데이터베이스 서버에 연결할 수 있는지 확인합니다.

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

이 단계가 실패하는 경우 네트워크와 방화벽 설정이 IdM 클라이언트와 서버 간의 직접 통신을 허용하는지 확인합니다. [firewalld 사용 및 구성](#)을 참조하십시오.

3. 클라이언트가 정규화된 호스트 이름을 통해 IdM LDAP 서버(IdM 사용자) 또는 AD 도메인 컨트롤러(AD 사용자의 경우)를 검색하고 연결할 수 있는지 확인합니다.

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

이 단계가 실패하면 **/etc/resolv.conf** 파일을 포함하여 DNS(동적 이름 서비스) 설정을 확인합니다. [DNS 서버 순서 구성](#)을 참조하십시오.

참고

기본적으로 SSSD 서비스는 SRV(DNS 서비스) 레코드를 통해 LDAP 서버 및 AD DC를 자동으로 검색하려고 합니다. 또는 **sssd.conf** 구성 파일에서 다음 옵션을 설정하여 특정 서버를 사용하도록 **SSSD** 서비스를 제한할 수 있습니다.

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

이러한 옵션을 사용하는 경우 해당 옵션에 나열된 서버에 연결할 수 있는지 확인합니다.

4. 클라이언트가 LDAP 서버에 인증하고 **ldapssearch** 명령을 사용하여 사용자 정보를 검색할 수 있는지 확인합니다.

- a. LDAP 서버가 **server.example.com** 과 같은 IdM 서버인 경우 호스트에 대한 Kerberos 티켓을 검색하고 호스트 Kerberos 주체를 사용하여 데이터베이스 검색 인증을 수행합니다.

```
[user@client ~]$ kinit -t 'host/client.example.com@EXAMPLE.COM'
[user@client ~]$ ldapsearch -LLL -Y GSSAPI -h server.example.com -b
"dc=example,dc=com" uid=<user_name>
```

- b. LDAP 서버가 **server.ad.example.com** 과 같은 DC(Active Directory) 도메인 컨트롤러인 경우 호스트에 대한 Kerberos 티켓을 검색하고 호스트 Kerberos 주체를 사용하여 데이터베이스 검색을 수행합니다.

```
[user@client ~]$ kinit -t 'CLIENT$@AD.EXAMPLE.COM'
[user@client ~]$ ldapsearch -LLL -Y GSSAPI -h server.ad.example.com -b
"dc=example,dc=com" sAMAccountname=<user_name>
```

- c. LDAP 서버가 일반 LDAP 서버이고 **sssd.conf** 파일에 **ldap_default_bind_dn** 및 **ldap_default_authtok** 옵션을 설정한 경우 동일한 **ldap_default_bind_dn** 계정으로 인증합니다.

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

이 단계가 실패하면 데이터베이스 설정이 호스트가 LDAP 서버를 검색할 수 있도록 허용하는지 확인합니다.

5. SSSD 서비스는 Kerberos 암호화를 사용하므로 로그인할 수 없는 사용자로 Kerberos 티켓을 가져올 수 있는지 확인합니다.

- a. LDAP 서버가 IdM 서버인 경우:

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

- b. LDAP 서버 데이터베이스가 AD 서버인 경우:

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

이 단계가 실패하면 Kerberos 서버가 제대로 작동하고 모든 서버가 시간 동기화되며 사용자 계정이 잠겨 있지 않은지 확인합니다.

6. 명령줄에서 사용자 정보를 검색할 수 있는지 확인합니다.

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

이 단계가 실패하면 클라이언트의 SSSD 서비스가 사용자 데이터베이스에서 정보를 수신할 수 있는지 확인합니다.

- a. **/var/log/messages** 로그 파일의 오류를 검토합니다.
- b. SSSD 서비스에서 자세한 로깅을 활성화하고 디버깅 로그를 수집한 다음 로그에서 문제 소스에 대한 표시를 확인합니다.
- c. (선택 사항) Red Hat 기술 지원 케이스를 열고 수집한 문제 해결 정보를 제공합니다.

7. **sssctl** 유틸리티를 사용하여 사용자가 로그인할 수 있는지 확인합니다.

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

이 단계가 실패하면 PAM 구성, IdM HBAC 규칙 및 IdM RBAC 규칙과 같은 인증 설정을 확인합니다.

- a. `/var/log/secure` 및 `/var/log/messages` 로그 파일에서 인증 오류를 검토합니다.
- b. SSSD 서비스에서 자세한 로깅을 활성화하고 디버깅 로그를 수집한 다음 로그에서 문제 소스에 대한 표시를 확인합니다.
- c. (선택 사항) Red Hat 기술 지원 케이스를 열고 수집한 문제 해결 정보를 제공합니다.

추가 리소스

- `sssd.conf` 파일에서 SSSD에 대한 세부 로깅 활성화
- `sssctl` 명령을 사용하여 SSSD에 대한 세부 로깅 활성화
- IdM 서버의 인증 문제 해결을 위해 SSSD 서비스에서 디버깅 로그 수집
- IdM 클라이언트의 인증 문제를 해결하기 위해 SSSD 서비스에서 디버깅 로그 수집

8.5. SSSD 로그 파일 및 로깅 수준

각 SSSD 서비스는 `/var/log/sss` 디렉토리에 있는 자체 로그 파일에 로그인합니다. **example.com** IdM 도메인에 있는 IdM 서버의 경우 로그 파일은 다음과 같을 수 있습니다.

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

8.5.1. SSSD 로그 파일 목적

krb5_child.log

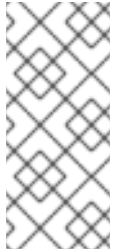
Kerberos 인증과 관련된 수명이 짧은 도우미 프로세스에 대한 로그 파일입니다.

ldap_child.log

LDAP 서버와의 통신을 위한 Kerberos 티켓을 가져오는 데 관련된 수명이 짧은 도우미 프로세스의 로그 파일입니다.

sssd_<example.com>.log

`sssd.conf` 파일의 각 도메인 섹션에 대해 SSSD 서비스는 LDAP 서버와의 통신에 대한 정보를 별도의 로그 파일에 기록합니다. 예를 들어, **example.com**이라는 IdM 도메인이 있는 환경에서 SSSD 서비스는 `sssd_example.com.log`라는 파일에 해당 정보를 기록합니다. 호스트가 **ad.example.com**이라는 AD 도메인과 직접 통합되는 경우 정보는 `sssd_ad.example.com.log`라는 파일에 기록됩니다.



참고

IdM 환경과 AD 도메인을 사용하는 교차 포트 신뢰가 있는 경우에도 AD 도메인에 대한 정보는 여전히 IdM 도메인의 로그 파일에 기록됩니다.

마찬가지로 호스트가 AD 도메인에 직접 통합되면 하위 도메인에 대한 정보는 기본 도메인의 로그 파일에 작성됩니다.

selinux_child.log

SELinux 정보를 검색하고 설정하는 수명이 짧은 도우미 프로세스에 대한 로그 파일입니다.

sssd.log

SSSD 모니터링 및 응답자 및 백엔드 프로세스와 통신하기 위한 로그 파일입니다.

sssd_ifp.log

시스템 버스를 통해 액세스할 수 있는 공용 D-Bus 인터페이스를 제공하는 InfoPipe 응답자의 로그 파일입니다.

sssd_nss.log

사용자 및 그룹 정보를 검색하는 NSS(Name Services Switch) 응답자의 로그 파일입니다.

sssd_pac.log

AD Kerberos 티켓에서 PAC를 수집하고 PAC에 대한 AD 사용자에게 대한 정보를 파생하여 AD에서 직접 요청하지 않도록 하는 Microsoft 권한 속성 인증서(PAC) 응답자의 로그 파일입니다.

sssd_pam.log

PAM(Pluggable Authentication Module) 응답자의 로그 파일입니다.

sssd_ssh.log

SSH 응답자 프로세스에 대한 로그 파일입니다.

8.5.2. SSSD 로깅 수준

디버그 수준을 설정하면 그 아래의 모든 디버그 수준도 활성화됩니다. 예를 들어 6에서 디버그 수준을 설정하면 디버그 수준 0~5도 활성화됩니다.

표 8.1. SSSD 로깅 수준

level	설명
0	치명적인 오류. SSSD 서비스가 시작되지 않거나 종료되는 오류입니다. RHEL 8.3 이전 버전의 기본 디버그 로그 수준입니다.
1	중요한 오류. SSSD 서비스를 종료하지 않지만 하나 이상의 주요 기능이 제대로 작동하지 않는 오류입니다.
2	심각한 오류. 특정 요청 또는 작업이 실패했음을 확인하는 동안 오류가 발생했습니다. RHEL 8.4 이상의 기본 디버그 로그 수준입니다.
3	사소한 오류. 레벨 2에서 작업 실패를 일으키는 오류.
4	구성 설정.

level	설명
5	기능 데이터.
6	작업 기능에 대한 메시지를 추적합니다.
7	내부 제어 기능에 대한 메시지를 추적합니다.
8	함수 내부 변수의 내용.
9	매우 낮은 수준의 추적 정보.

8.6. SSSD.CONF 파일에서 SSSD에 대한 세부 로깅 활성화

기본적으로 RHEL 8.4 이상의 SSSD 서비스는 심각한 오류(디버그 수준 2)만 로깅하지만 인증 문제 해결에 필요한 세부적인 수준에는 기록되지 않습니다.

SSSD 서비스 재시작 시 세부 로깅을 영구적으로 사용하려면 **/etc/sss/sss.conf** 구성 파일의 각 섹션에 **debug_level=<integer>** 옵션을 추가합니다. 여기서 **<integer>** 값은 0에서 9 사이의 숫자입니다. 최대 3개의 로그 실패를 디버그하고, 수준 8 이상에서는 많은 수의 세부 로그 메시지를 제공합니다. 수준 6은 인증 문제를 디버깅하기 위한 좋은 시작 지점입니다.

사전 요구 사항

- **sss.conf** 구성 파일을 편집하고 SSSD 서비스를 다시 시작하려면 루트 암호가 필요합니다.

절차

1. 텍스트 편집기에서 **/etc/sss/sss.conf** 파일을 엽니다.
2. **debug_level** 옵션을 파일의 모든 섹션에 추가하고 디버그 수준을 선택한 세부 정보 표시로 설정합니다.

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

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

[nss]
debug_level = 6

[pam]
debug_level = 6

[sudo]
debug_level = 6
```

```
[ssh]
debug_level = 6

[pac]
debug_level = 6

[ifp]
debug_level = 6
```

3. **sssd.conf** 파일을 저장하고 닫습니다.
4. SSSD 서비스를 다시 시작하여 새 구성 설정을 로드합니다.

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

추가 리소스

- [SSSD 로그 파일 및 로깅 수준](#)

8.7. SSSCTL 명령을 사용하여 SSSD에 대한 세부 로깅 활성화

기본적으로 RHEL 8.4 이상의 SSSD 서비스는 심각한 오류(디버그 수준 2)만 로깅하지만 인증 문제 해결에 필요한 세부적인 수준에는 기록되지 않습니다.

ssssctl debug-level <integer> 명령을 사용하여 명령줄에서 SSSD 서비스의 디버그 수준을 변경할 수 있습니다. 여기서 <integer> 값은 0에서 9 사이의 숫자입니다. 최대 3개의 로그 실패를 디버그하고, 수준 8 이상에서는 많은 수의 세부 로그 메시지를 제공합니다. 수준 6은 인증 문제를 디버깅하기 위한 좋은 시작 지점입니다.

사전 요구 사항

- **ssssctl** 명령을 실행하려면 루트 암호가 필요합니다.

절차

- **ssssctl debug-level** 명령을 사용하여 선택한 디버그 수준을 원하는 세부 정보 표시 수준으로 설정합니다.

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

추가 리소스

- [SSSD 로그 파일 및 로깅 수준](#)

8.8. IDM 서버의 인증 문제 해결을 위해 SSSD 서비스에서 디버깅 로그 수집

IdM 서버로 IdM 사용자로 인증하려고 할 때 문제가 발생하는 경우 서버의 SSSD 서비스에서 자세한 디버그 로깅을 활성화하고 사용자 정보 검색 시도의 로그를 수집합니다.

사전 요구 사항

- **ssssctl** 명령을 실행하고 SSSD 서비스를 다시 시작하려면 루트 암호가 필요합니다.

절차

1. IdM 서버에서 자세한 SSSD 디버그 로깅을 활성화합니다.

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

2. 인증 문제가 발생한 사용자에게 대해 SSSD 캐시의 오브젝트를 무효화하므로 LDAP 서버를 우회하지 않고 SSSD가 이미 캐시된 정보를 검색하지 않습니다.

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

3. 이전 SSSD 로그를 제거하여 데이터 세트 문제 해결을 최소화합니다.

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

4. 시도 전후에 타임스탬프를 수집하는 동시에 인증 문제가 발생한 사용자로 전환합니다. 이러한 타임스탬프는 데이터 세트의 범위를 더욱 좁힙니다.

```
[root@server sssd]# date; su idmuser; date
Mon Mar 29 15:33:48 EDT 2021
su: user idmuser does not exist
Mon Mar 29 15:33:49 EDT 2021
```

5. (선택 사항) 자세한 SSSD 로그를 계속 수집하지 않으려면 디버그 수준을 낮춥니다.

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

6. 실패한 요청에 대한 정보는 SSSD 로그를 검토하십시오. 예를 들어 `/var/log/sss/sss_example.com.log` 파일을 검토하면 SSSD 서비스에서 **cn=accounts,dc=example,dc=com** LDAP 하위 트리에서 사용자를 찾지 못했습니다. 이는 사용자가 없거나 다른 위치에 있음을 나타낼 수 있습니다.

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

7. 인증 문제의 원인을 확인할 수 없는 경우:

- a. 최근에 생성된 SSSD 로그를 수집합니다.

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

b. Red Hat 기술 지원 케이스를 열고 다음을 제공합니다.

i. SSSD 로그: **sssd-logs-29.tar**

ii. 로그에 해당하는 요청의 타임스탬프 및 사용자 이름을 포함한 콘솔 출력은 다음과 같습니다.

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

8.9. IDM 클라이언트의 인증 문제를 해결하기 위해 SSSD 서비스에서 디버깅 로그 수집

IdM 클라이언트에 IdM 사용자로 인증하려고 할 때 문제가 발생하는 경우 IdM 서버에 대한 사용자 정보를 검색할 수 있는지 확인합니다. IdM 서버에서 사용자 정보를 검색할 수 없는 경우 IdM 서버에서 정보를 검색하는 IdM 클라이언트에서 해당 정보를 검색할 수 없습니다.

인증 문제가 IdM 서버에서 발생하지 않았다는 것을 확인한 후 IdM 서버와 IdM 클라이언트에서 SSSD 디버깅 로그를 수집합니다.

사전 요구 사항

- 사용자는 IdM 서버가 아닌 IdM 클라이언트에만 인증 문제가 있습니다.
- **sssctl** 명령을 실행하고 SSSD 서비스를 다시 시작하려면 루트 암호가 필요합니다.

절차

1. 클라이언트에서 다음을 수행합니다. 텍스트 편집기에서 **/etc/sss/sss.conf** 파일을 엽니다.
2. 클라이언트에서 다음을 수행합니다. **ipa_server** 옵션을 파일의 **[domain]** 섹션에 추가하고 IdM 서버로 설정합니다. 이렇게 하면 IdM 클라이언트가 다른 IdM 서버를 자동 검색하지 않으므로 이 테스트를 하나의 클라이언트와 하나의 서버로만 제한합니다.

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

3. 클라이언트에서 다음을 수행합니다. **sss.conf** 파일을 저장하고 닫습니다.
4. 클라이언트에서 다음을 수행합니다. SSSD 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.

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

5. 서버 및 클라이언트에서 다음을 수행합니다. 자세한 SSSD 디버그 로깅을 활성화합니다.

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

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

6. 서버 및 클라이언트에서 다음을 수행합니다. 인증 문제가 발생한 사용자의 경우 SSSD 캐시에서 오브젝트를 무효화하므로 LDAP 데이터베이스를 무시하고 SSSD가 이미 캐시된 정보를 검색하지 않습니다.

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

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

7. 서버 및 클라이언트에서 다음을 수행합니다. 이전 SSSD 로그를 제거하여 데이터 세트 문제 해결을 최소화합니다.

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

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

8. 클라이언트에서 다음을 수행합니다. 시도 전후에 타임스탬프를 수집하는 동안 인증 문제가 발생하는 사용자로 전환합니다. 이러한 타임스탬프는 데이터 세트의 범위를 더욱 좁힙니다.

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

9. (선택 사항) 서버와 클라이언트에서 다음을 수행합니다. 자세한 SSSD 로그를 계속 수집하지 않으려면 디버그 수준을 낮춥니다.

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

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

10. 서버 및 클라이언트에서 다음을 수행합니다. 실패한 요청에 대한 정보는 SSSD 로그를 검토하십시오.

- 클라이언트 로그의 클라이언트에서 요청을 검토합니다.
- 서버 로그의 클라이언트에서 요청을 검토합니다.
- 서버 로그에서 요청 결과를 검토합니다.
- 서버에서 요청 결과를 수신하는 클라이언트의 결과를 검토합니다.

11. 인증 문제의 원인을 확인할 수 없는 경우:

- 최근 IdM 서버 및 IdM 클라이언트에서 생성된 SSSD 로그를 수집합니다. 호스트 이름 또는 역할에 따라 레이블을 지정합니다.

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

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

- Red Hat 기술 지원 케이스를 열고 다음을 제공합니다.

i. SSSD 디버그 로그:

A. 서버의 **sssd-logs-server-29.tar**B. 클라이언트에서 **sssd-logs-client-29.tar** 을 클라이언트에서 실행

ii. 로그에 해당하는 요청의 타임스탬프 및 사용자 이름을 포함한 콘솔 출력은 다음과 같습니다.

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

8.10. SSSD 백엔드에서 클라이언트 요청 추적

SSSD는 요청을 비동기적으로 처리하고 다른 요청의 메시지가 동일한 로그 파일에 추가되므로 고유한 요청 식별자와 클라이언트 ID를 사용하여 백엔드 로그에서 클라이언트 요청을 추적할 수 있습니다. 고유한 요청 식별자는 **RID#<integer>** 형식의 디버그 로그에 추가되고 **[CID #< integer>]** 형식의 클라이언트 ID가 추가됩니다. 이를 통해 개별 요청과 관련된 로그를 격리하고 요청을 여러 SSSD 구성 요소의 로그 파일에서 처음부터 끝까지 추적할 수 있습니다.

사전 요구 사항

- 디버그 로깅을 활성화했으며 IdM 클라이언트에서 요청이 제출되었습니다.
- SSSD 로그 파일의 내용을 표시하려면 루트 권한이 있어야 합니다.

절차

1. SSSD 로그 파일을 검토하려면 **less** 유틸리티를 사용하여 로그 파일을 엽니다. 예를 들어 **/var/log/sss/sssd_example.com.log**:

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

2. 클라이언트 요청에 대한 정보는 SSSD 로그를 검토합니다.

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

SSSD 로그 파일의 이 샘플 출력은 두 개의 다른 요청에 대한 고유 식별자 **RID#3** 및 **RID#4** 를 보여줍니다.

그러나 SSSD 클라이언트 인터페이스에 대한 단일 클라이언트 요청은 백엔드에서 여러 요청을 트리거하는 경우가 많으며 이로 인해 백엔드의 클라이언트 요청과 요청 간에 1-to-1의 상관 관계가 아닙니다. 백엔드의 여러 요청에는 RID 번호가 다르지만 각 초기 백엔드 요청에는 고유한 클라이언트 ID가 포함되어 있으므로 관리자는 여러 RID 번호를 단일 클라이언트 요청에 대해 추적할 수 있습니다.

다음 예제에서는 하나의 클라이언트 요청 [sssd.nss CID #1] 및 백엔드에서 생성된 여러 요청 [RID#5] 을 [RID#13]으로 보여줍니다.

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

9장. ANSIBLE 플레이북을 사용하여 글로벌 IDM 설정 구성

Ansible config 모듈을 사용하면 **IdM(Identity Management)**에 대한 글로벌 구성 매개 변수를 검색하고 설정할 수 있습니다.

이 장에는 다음 섹션이 포함되어 있습니다.

- [Ansible 플레이북을 사용하여 IdM 구성 검색](#)
- [Ansible 플레이북을 사용하여 IdM CA 갱신 서버 구성](#)
- [Ansible 플레이북을 사용하여 IdM 사용자에게 대한 기본 셸 구성](#)

9.1. ANSIBLE 플레이북을 사용하여 IDM 구성 검색

다음 절차에서는 **Ansible** 플레이북을 사용하여 현재 글로벌 **IdM** 구성에 대한 정보를 검색하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. **inventory** 파일(예: **inventory.file**)을 생성하고 **[ipaserver]** 섹션에서 **IdM** 구성을 검색하려는 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**에서 데이터를 검색하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

2. 편집을 위해 **/usr/share/doc/ansible-freeipa/playbooks/config/retmasterve-config.yml** **Ansible** 플레이북 파일을 엽니다.

```

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

  tasks:
    - name: Query IPA global configuration
      ipaconfig:
        ipaadmin_password: Secret123
        register: serverconfig

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

```

3.

다음을 변경하여 파일을 조정합니다.

- IdM 관리자의 암호입니다.
- 필요한 경우 기타 값입니다.

4.

파일을 저장합니다.

5.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/config/retrieve-config.yml
[...]
TASK [debug]
ok: [server.idm.example.com] => {
  "msg": {
    "ansible_facts": {
      "discovered_interpreter_
    },
    "changed": false,
    "config": {
      "ca_renewal_master_server": "server.idm.example.com",
      "configstring": [
        "AllowNThash",
        "KDC:Disable Last Success"
      ],
      "defaultgroup": "ipausers",
      "defaultshell": "/bin/bash",
      "emaildomain": "idm.example.com",

```

```

    "enable_migration": false,
    "groupsearch": [
        "cn",
        "description"
    ],
    "homedirectory": "/home",
    "maxhostname": "64",
    "maxusername": "64",
    "pac_type": [
        "MS-PAC",
        "nfs:NONE"
    ],
    "pwdexpnotify": "4",
    "searchrecordslimit": "100",
    "searchtimelimit": "2",
    "selinuxusermapdefault": "unconfined_u:s0-s0:c0.c1023",
    "selinuxusermaporder": [
        "guest_u:s0$guest_u:s0$user_"
    ],
    "usersearch": [
        "uid",
        "givenname",
        "sn",
        "telephonenumber",
        "ou",
        "title"
    ]
  },
  "failed": false
}
}

```

9.2. ANSIBLE 플레이북을 사용하여 IDM CA 갱신 서버 구성

임베디드 CA(인증 기관)를 사용하는 IdM(Identity Management) 배포에서 CA 갱신 서버는 IdM 시스템 인증서를 유지 관리하고 갱신합니다. 강력한 IdM 배포를 보장합니다.

IdM CA 갱신 서버의 역할에 대한 자세한 내용은 [IdM CA 갱신 서버](#) 사용을 참조하십시오.

다음 절차에서는 Ansible 플레이북을 사용하여 IdM CA 갱신 서버를 구성하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.

절차

1. 선택 사항: 현재 **IdM CA** 갱신 서버를 식별합니다.

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

2. 인벤토리 파일(예: **inventory.file**)을 생성하고 여기에 **ipaserver**를 정의합니다.

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

3. 편집을 위해 **/usr/share/doc/ansible-freeipa/playbooks/config/set-ca-renewal-master-server.yml** Ansible 플레이북 파일을 엽니다.

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

  tasks:
    - name: set ca_renewal_master_server
      ipaconfig:
        ipaadmin_password: SomeADMINpassword
        ca_renewal_master_server: carenewal.idm.example.com
```

4. 다음과 같이 변경하여 파일을 조정합니다.

- **ipaadmin_password** 변수로 설정한 **IdM** 관리자의 암호입니다.
- **ca_renewal_master_server** 변수에서 설정한 **CA** 갱신 서버의 이름입니다.

5. 파일을 저장합니다.

6.

Ansible 플레이북을 실행합니다. 플레이북 파일과 인벤토리 파일을 지정합니다.

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

검증 단계

CA 갱신 서버가 변경되었는지 확인할 수 있습니다.

1.

IdM 관리자로 **ipaserver** 에 로그인합니다.

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

2.

IdM CA 갱신 서버의 **ID**를 요청합니다.

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

출력에 **carenewal.idm.example.com** 서버가 새 **CA** 갱신 서버임을 보여줍니다.

9.3. ANSIBLE 플레이북을 사용하여 IDM 사용자에게 대한 기본 셸 구성

셸은 명령을 수락하고 해석하는 프로그램입니다. **bash, sh, ksh, zsh** 등의 **RHEL(Red Hat Enterprise Linux)**에서 몇 가지 셸을 사용할 수 있습니다. **Bash** 또는 **/bin/bash** 는 대부분의 **Linux** 시스템에서 널리 사용되는 셸이며, 일반적으로 **RHEL**의 사용자 계정의 기본 셸입니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 사용자의 기본 셸로 대체 셸인 **sh** 를 구성하는 방법을 설명합니다.

사전 요구 사항

•

IdM 관리자 암호를 알고 있습니다.

•

Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.

절차

1. 선택 사항: **retrieve-config.yml** Ansible 플레이북을 사용하여 **IdM** 사용자의 현재 셸을 식별합니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 구성](#) 검색을 참조하십시오.

2. 인벤토리 파일(예: **inventory.file**)을 생성하고 여기에 **ipaserver**를 정의합니다.

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

3. 편집을 위해 **/usr/share/doc/ansible-freeipa/playbooks/config/ensure-config-options-are-set.yml** Ansible 플레이북 파일을 엽니다.

```
---
- name: Playbook to ensure some config options are set
  hosts: ipaserver
  become: true

  tasks:
    # Set defaultlogin and maxusername
    - ipaconfig:
        ipaadmin_password: Secret123
        defaultshell: /bin/bash
        maxusername: 64
```

4. 다음을 변경하여 파일을 조정합니다.

- **ipaadmin_password** 변수로 설정한 **IdM** 관리자의 암호입니다.
- **default shell** 변수로 설정한 **IdM** 사용자의 기본 셸은 **/bin/sh**에 있습니다.

5. 파일을 저장합니다.

6. **Ansible** 플레이북을 실행합니다. 플레이북 파일과 인벤토리 파일을 지정합니다.

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

검증 단계

IdM에서 새 세션을 시작하여 기본 사용자 셸이 변경되었는지 확인할 수 있습니다.

1.

IdM 관리자로 **ipaserver**에 로그인합니다.

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

2.

현재 셸을 표시합니다.

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

로그인한 사용자는 **sh** 셸을 사용하고 있습니다.

추가 리소스

- **/usr/share/doc/ansible-freeipa/** 디렉터리에 있는 **README-config.md** Markdown 파일에서 글로벌 **IdM** 설정 및 가능한 변수 목록을 구성하는 샘플 **Ansible** 플레이북을 볼 수 있습니다.
- **/usr/share/doc/ansible-freeipa/playbooks/config** 디렉터리에서 다양한 **IdM** 구성 관련 작업에 대한 샘플 **Ansible** 플레이북을 볼 수 있습니다.

10장. 명령줄을 사용하여 사용자 계정 관리

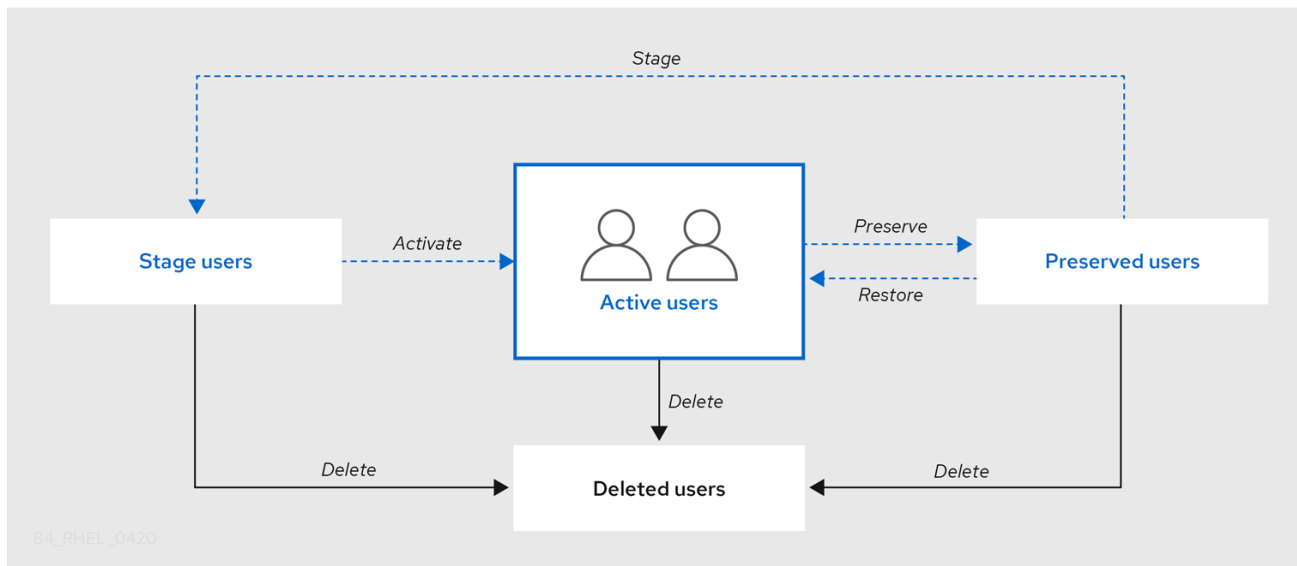
이 장에는 **IdM**의 사용자 라이프사이클에 대한 기본 설명(**ID 관리**)이 포함되어 있습니다. 다음 섹션에서는 다음을 수행하는 방법을 보여줍니다.

- 사용자 계정 만들기
- 단계 사용자 계정 활성화
- 사용자 계정 보존
- 활성화, 스테이징 또는 사용자 계정 삭제
- 보존된 사용자 계정 복원

10.1. 사용자 라이프 사이클

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

- 스테이징 사용자는 인증이 허용되지 않습니다. 초기 상태입니다. 활성화 사용자에게 필요한 일부 사용자 계정 속성은 예를 들어 그룹 멤버십을 설정할 수 없습니다.
- 활성화 사용자는 인증을 허용합니다. 필요한 모든 사용자 계정 속성은 이 상태에서 설정해야 합니다.
- 보존된 사용자는 비활성으로 간주되고 **IdM**에 인증할 수 없는 이전 활성화 사용자입니다. 보존된 사용자는 활성화 사용자로 보유한 계정 속성의 대부분을 유지하지만 사용자 그룹의 일부가 아닙니다.



IdM 데이터베이스에서 영구적으로 사용자 항목을 삭제할 수 있습니다.



중요

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

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



주의

admin 사용자를 삭제하지 마십시오. **admin**은 IdM에 필요한 사전 정의된 사용자 이므로 이 작업으로 인해 특정 명령에서 문제가 발생합니다. 대체 **admin** 사용자를 정의하고 사용하려는 경우 하나 이상의 다른 사용자에게 **admin** 권한을 부여한 후 **ipa** 사용자 비활성화 **admin**으로 사전 정의된 **admin** 사용자를 비활성화합니다.



주의

IdM에 로컬 사용자를 추가하지 마십시오. NSS(Name Service Switch)는 로컬 사용자 및 그룹을 확인하기 전에 항상 IdM 사용자 및 그룹을 확인합니다. 즉, IdM 그룹 멤버십은 로컬 사용자에게 작동하지 않습니다.

10.2. 명령줄을 사용하여 사용자 추가

다음과 같이 사용자를 추가할 수 있습니다.

- **활성 메시지** - 사용자가 적극적으로 사용할 수 있는 사용자 계정.
- **스테이징** - 암호사용자는 이러한 계정을 사용할 수 없습니다. 새 사용자 계정을 준비하려면 이 파일을 사용합니다. 사용자가 자신의 계정을 사용할 준비가 되면 활성화할 수 있습니다.

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

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



참고

IdM은 고유한 사용자 **ID(UID)**를 새 사용자 계정에 자동으로 할당합니다. 이 작업을 수동으로 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 없는 것을 방지하는 것이 좋습니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **Kerberos** 티켓을 획득했습니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.
2. 사용자 로그인, 사용자의 이름, 성을 추가하고 필요한 경우 이메일 주소를 추가할 수도 있습니다.

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

IdM에서는 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 추가하는 경우 **IdM**은 자동으로 이름을 저장할 때 소문자로 변환합니다. 따라서 **IdM**은 로그인할 때 항상 사용자 이름을 소문자로 입력해야 합니다. 또한 사용자 및 사용자 등 문자 캐싱에서만 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 명령에는 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 성공적으로 생성되었는지 확인할 수 있습니다.

\$ ipa user-find

이 명령은 세부 정보가 있는 모든 사용자 계정을 나열합니다.

10.3. 명령줄을 사용하여 사용자 활성화

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

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **Kerberos** 티켓을 획득했습니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

절차

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

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

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

\$ ipa user-find

이 명령은 세부 정보가 있는 모든 사용자 계정을 나열합니다.

10.4. 명령줄을 사용하여 사용자 보존

제거하려는 경우 사용자 계정을 보존할 수 있지만 옵션을 유지하여 나중에 복원할 수 있습니다. 사용자 계정을 보존하려면 **ipa user-del** 또는 **ipa stageuser-del** 명령에 **--preserve** 옵션을 사용합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **Kerberos** 티켓을 획득했습니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

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 불구하고 보존되었습니다.

10.5. 명령줄을 사용하여 사용자 삭제

IdM(ID 관리)을 사용하면 사용자를 영구적으로 삭제할 수 있습니다. 다음을 삭제할 수 있습니다.

- 다음 명령을 사용하는 활성 사용자: **ipa user-del**
- 다음 명령을 사용하여 사용자 준비: **ipa stageuser-del**

•

다음 명령을 사용하여 사용자 보존: **ipa user-del**

여러 사용자를 삭제하는 경우 **--continue** 옵션을 사용하여 오류에 관계없이 명령이 계속되도록 합니다. 명령이 완료되면 성공 및 실패한 작업의 요약이 **stdout** 표준 출력 스트림에 출력됩니다.

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

--continue 를 사용하지 않는 경우 명령은 오류가 발생할 때까지 사용자 삭제를 진행합니다. 이 경우 이 명령이 중지되고 종료됩니다.

사전 요구 사항

•

IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

•

Kerberos 티켓을 획득했습니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

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.

10.6. 명령줄을 사용하여 사용자 복원

보존된 사용자를 다음과 같이 복원할 수 있습니다.

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

사용자 계정을 복원해도 계정의 이전 속성이 모두 복원되지는 않습니다. 예를 들어 사용자의 암호가 복원되지 않으므로 다시 설정해야 합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **Kerberos** 티켓을 획득했습니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

절차

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

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

또는 사용자 계정을 스테이징된 상태로 복원할 수 있습니다.

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

검증 단계

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

\$ ipa user-find

이 명령은 세부 정보가 있는 모든 사용자 계정을 나열합니다.

11장. IDM 웹 UI를 사용하여 사용자 계정 관리

IdM(Identity Management)은 다양한 사용자 작업 환경을 관리하는 데 도움이 되는 여러 단계를 제공합니다.

사용자 계정 생성

직원이 회사에서 경력을 시작하기 전에 단계 사용자 계정을 만들고 직원이 사무실에 나타나면 계정을 활성화하려는 날을 미리 준비하십시오.

이 단계를 생략하고 활성 사용자 계정을 직접 만들 수 있습니다. 절차는 stage 사용자 계정 생성과 유사합니다.

사용자 계정 활성화

직원의 첫 근무일 계정 활성화.

사용자 계정 비활성화

사용자가 몇 개월 동안 부모의 휴가 로 이동하는 경우 계정을 일시적으로 비활성화해야 합니다.

사용자 계정 활성화

사용자가 반환되면 계정을 다시 활성화해야 합니다.

사용자 계정 보존

사용자가 퇴사하려는 경우 계정이 다시 복원될 가능성이 있는 경우 나중에 회사로 돌아갈 수 있기 때문에 계정을 삭제해야 합니다.

사용자 계정 복원

2년 후 사용자가 다시 시작되어 보존된 계정을 복원해야 합니다.

사용자 계정 삭제

직원이 해당 직원이 거부되면 백업 없이 계정을 삭제합니다.

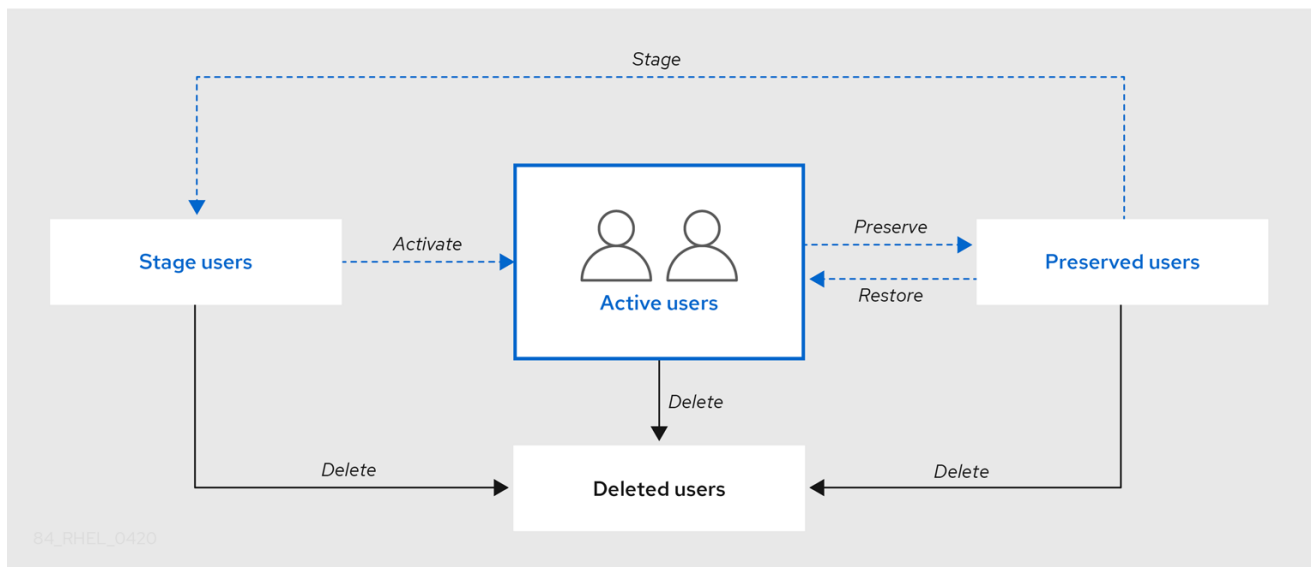
11.1. 사용자 라이프 사이클

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

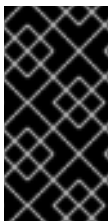
-

스테이징 사용자는 인증이 허용되지 않습니다. 초기 상태입니다. 활성 사용자에게 필요한 일부 사용자 계정 속성은 예를 들어 그룹 멤버십을 설정할 수 없습니다.

- 활성 사용자는 인증을 허용합니다. 필요한 모든 사용자 계정 속성은 이 상태에서 설정해야 합니다.
- 보존된 사용자는 비활성으로 간주되고 **IdM**에 인증할 수 없는 이전 활성 사용자입니다. 보존된 사용자는 활성 사용자로 보유한 계정 속성의 대부분을 유지하지만 사용자 그룹의 일부가 아닙니다.



IdM 데이터베이스에서 영구적으로 사용자 항목을 삭제할 수 있습니다.



중요

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

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



주의

admin 사용자를 삭제하지 마십시오. **admin**은 **IdM**에 필요한 사전 정의된 사용자이므로 이 작업으로 인해 특정 명령에서 문제가 발생합니다. 대체 **admin** 사용자를 정의하고 사용하려는 경우 하나 이상의 다른 사용자에게 **admin** 권한을 부여한 후 **ipa** 사용자 비활성화 **admin**으로 사전 정의된 **admin** 사용자를 비활성화합니다.



주의

IdM에 로컬 사용자를 추가하지 마십시오. **NSS(Name Service Switch)**는 로컬 사용자 및 그룹을 확인하기 전에 항상 **IdM** 사용자 및 그룹을 확인합니다. 즉, **IdM** 그룹 멤버십은 로컬 사용자에게 작동하지 않습니다.

11.2. 웹 UI에서 사용자 추가

일반적으로 새 직원이 작동하기 전에 새 사용자 계정을 만들어야 합니다. 이러한 단계 계정에 액세스할 수 없으며 나중에 활성화해야 합니다.



참고

또는 활성 사용자 계정을 직접 만들 수도 있습니다. 활성 사용자를 추가하려면 아래 절차를 따르고 **Active users** (활성 사용자) 탭에 사용자 계정을 추가합니다.

사전 요구 사항

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

절차

1. **IdM** 웹 UI에 로그인합니다.

자세한 내용은 웹 [브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.

2.
사용자 → 사용자 단계 탭으로 이동합니다.

또는 사용자 → 활성 사용자에게 사용자 계정을 추가할 수도 있지만 계정에 사용자 그룹을 추가할 수는 없습니다.
3.
+ **Add**(추가) 아이콘을 클릭합니다.
4.
Add stage user (단계 사용자 추가) 대화 상자에 새 사용자의 이름 및 이름을 입력합니다.
5.
[선택 사항] 사용자 로그인 필드에 로그인 이름을 추가합니다.

비워 두면 **IdM** 서버에서 다음 패턴으로 로그인 이름을 생성합니다. 이름 및 성의 첫 글자. 전체 로그인 이름은 최대 **32**자를 가질 수 있습니다.
6.
[선택 사항] **GID** 드롭다운 메뉴에서 사용자가 포함해야 하는 그룹을 선택합니다.
7.
[선택 사항] **Password and Verify password** (암호 및 암호 확인) 필드에 암호를 입력하고 확인한 후 둘 다 일치하는지 확인합니다.
8.
Add(추가) 단추를 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

이때 **Stage Users** (사용자 단계) 테이블에 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
☐	euser	Example	User	-1	euser@idm.example.com		

Showing 1 to 1 of 1 entries.



참고

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

11.3. IDM 웹 UI에서 단계 사용자 활성화

사용자가 **IdM**에 로그인하고 사용자를 **IdM** 그룹에 추가하려면 먼저 스테이징 사용자 계정을 활성화해야 합니다. 이 섹션에서는 단계 사용자 계정을 활성화하는 방법에 대해 설명합니다.

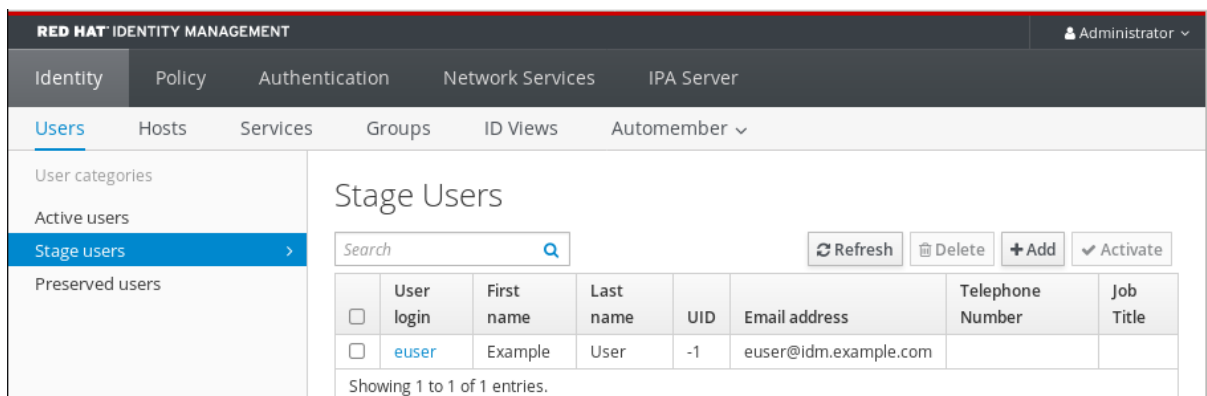
사전 요구 사항

- **IdM 웹 UI** 또는 사용자 관리자 역할을 관리하는 관리자 권한.
- **IdM**에서 하나 이상의 준비 사용자 계정.

절차

1. **IdM 웹 UI**에 로그인합니다.

자세한 내용은 웹 [브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.
2. 사용자 → 사용자 탭으로 이동합니다.
3. 활성화하려는 사용자 계정의 확인란을 클릭합니다.
4. **Activate(활성화)** 버튼을 클릭합니다.



5.

Confirmation(확인) 대화 상자에서 **OK (확인)** 버튼을 클릭합니다.

활성화에 성공하면 **IdM 웹 UI**에 사용자가 활성화되고 사용자 계정이 **Active users**로 이동한 녹색 확인이 표시됩니다. 계정이 활성 상태이며 사용자가 **IdM 도메인** 및 **IdM 웹 UI**에 인증할 수 있습니다. 처음으로 로그인할 때 암호를 변경하라는 메시지가 표시됩니다.

Active users

Search		Refresh		Delete		+ Add		- Disable		✓ Enable		Actions ▾	
☐	User login	First name	Last name	Status	UID	Email address	Telephone Number	Job Title					
☐	admin		Administrator	✓ Enabled	78000000								
☒	euser	Example	User	✓ Enabled	78000006	euser@idm.example.com							
☐	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.

11.4. 웹 UI에서 사용자 계정 비활성화

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

비활성화된 사용자 계정은 **IdM** 내에 계속 있으며 연결된 모든 정보는 변경되지 않습니다. 보존된 사용자 계정과 달리 비활성화된 사용자 계정이 활성 상태로 유지되며 사용자 그룹의 멤버가 될 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 관리하는 관리자 권한.**

절차

1.

IdM 웹 UI에 로그인합니다.

자세한 내용은 웹 [브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.

2.

사용자 → 활성 사용자 탭으로 이동합니다.

3.

비활성화하려는 사용자 계정의 확인란을 클릭합니다.

4.

Disable(비활성화) 버튼을 클릭합니다.

Active users

Search

Refresh Delete Add Disable Enable Actions ▾

☐	User login	First name	Last name	Status	UID	Email address	Telephone Number	Job Title
☐	admin		Administrator	✓ Enabled	78000000			
☒	euser	Example	User	✓ Enabled	78000006	euser@idm.example.com		
☐	preserved.user	Preserved	User	✓ Enabled	78000009	preserved.user@idm.example.com		

Showing 1 to 3 of 3 entries.

5.

Confirmation(확인) 대화 상자에서 **OK(확인)** 버튼을 클릭합니다.

비활성화 절차가 성공한 경우 **Active users** (활성 사용자) 테이블의 **Status(상태)** 열에서 확인할 수 있습니다.

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

11.5. 웹 UI에서 사용자 계정 활성화

IdM을 사용하면 비활성화된 활성 사용자 계정을 활성화할 수 있습니다. 사용자 계정을 활성화하면 비활성화된 계정을 활성화합니다.

사전 요구 사항

- **IdM 웹 UI** 또는 사용자 관리자 역할을 관리하는 관리자 권한.

절차

1. **IdM 웹 UI**에 로그인합니다.
2. 사용자 → 활성 사용자 탭으로 이동합니다.
3. 활성화하려는 사용자 계정의 확인란을 클릭합니다.
4. **Enable** 단추를 클릭합니다.

Active users

Search

Refresh Delete + Add **- Disable** **✓ Enable** Actions ▾

☐	User login	First name	Last name	Status	UID	Email address	Telephone Number	Job Title
☐	admin		Administrator	✓ Enabled	78000000			
☒	euser	Example	User	✓ Enabled	78000006	euser@idm.example.com		
☐	preserved.user	Preserved	User	✓ Enabled	78000009	preserved.user@idm.example.com		

Showing 1 to 3 of 3 entries.

5.

Confirmation(확인) 대화 상자에서 **OK (확인)** 버튼을 클릭합니다.

변경이 성공적으로 수행된 경우 **Active users** (활성 사용자) 테이블의 **Status(상태)** 열에서 확인할 수 있습니다.

11.6. IDM 웹 UI에서 활성 사용자 보존

사용자 계정을 보존하면 **Active users** 탭에서 계정을 제거할 수 있지만 이러한 계정을 **IdM**에 유지할 수 있습니다.

직원이 퇴사하는 경우 사용자 계정을 보존합니다. 몇 주 또는 몇 달 동안 사용자 계정을 비활성화하려면 (예:임시 휴가) 계정을 비활성화하십시오. 자세한 내용은 [11.4절. “웹 UI에서 사용자 계정 비활성화”](#)의 내용을 참조하십시오. 보존된 계정은 활성 상태가 아니므로 사용자가 내부 네트워크에 액세스하는 데 사용할 수 없지만 계정은 모든 데이터가 있는 데이터베이스에 남아 있습니다.

복원된 계정을 활성 모드로 다시 이동할 수 있습니다.



참고

보존 상태에 있는 사용자 목록은 과거 사용자 계정 기록을 제공할 수 있습니다.

사전 요구 사항

•

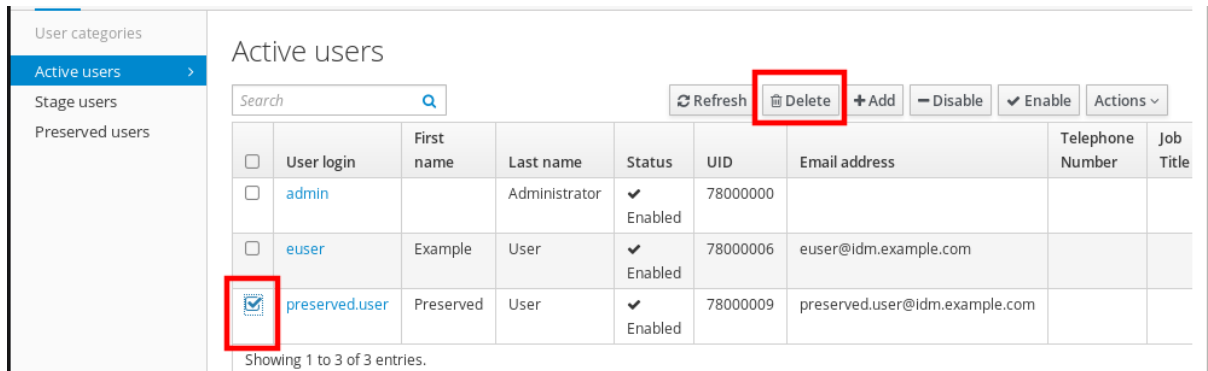
IdM(ID 관리) 웹 UI 또는 사용자 관리자 역할을 관리하는 관리자 권한.

절차

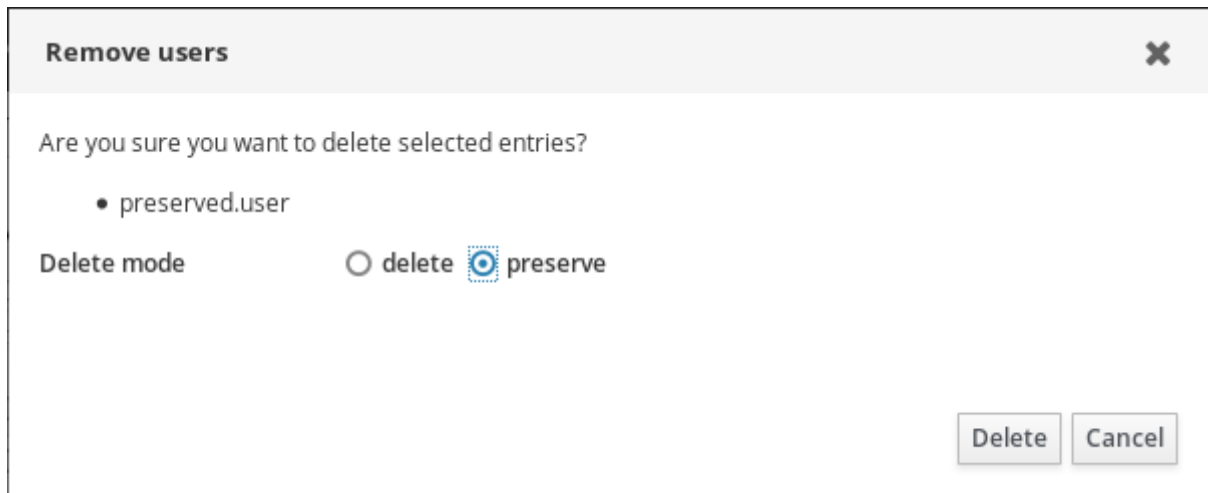
1. **IdM 웹 UI에 로그인 합니다.**

자세한 내용은 웹 브라우저에서 **IdM 웹 UI 액세스**를 참조하십시오.

2. 사용자 → 활성 사용자 탭으로 이동합니다.
3. 보존할 사용자 계정의 확인란을 클릭합니다.
4. **Delete(삭제)** 단추를 클릭합니다.



5. **Remove users(사용자 제거)** 대화 상자에서 **Delete mode(삭제 모드)** 라디오 버튼을 전환하여 보존합니다.
6. **Delete(삭제)** 단추를 클릭합니다.



결과적으로 사용자 계정이 보존된 사용자로 이동됩니다.

보존된 사용자를 복원해야 하는 경우 **IdM 웹 UI에서 사용자** 복원을 참조하십시오.

11.7. IDM 웹 UI에서 사용자 복원

IdM(ID 관리)을 사용하면 보존된 사용자 계정을 활성 상태로 다시 복원할 수 있습니다.

사전 요구 사항

- **IdM 웹 UI** 또는 사용자 관리자 역할을 관리하는 관리자 권한.

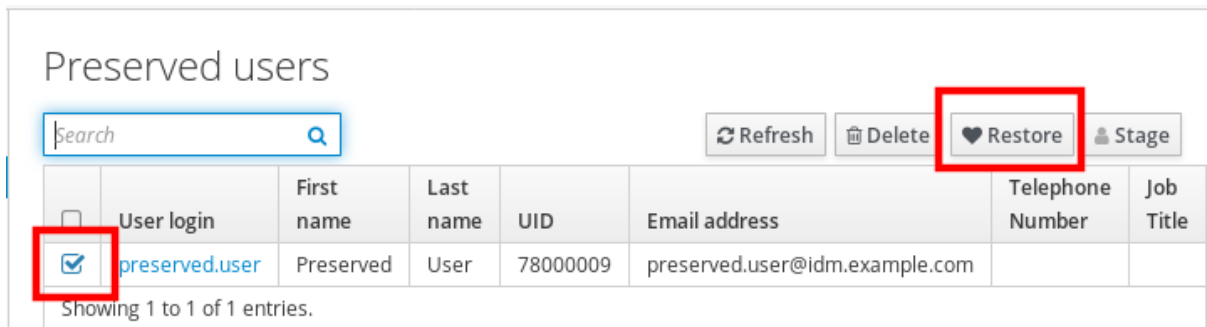
절차

1. **IdM 웹 UI**에 로그인합니다.

자세한 내용은 웹 [브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.
2. 사용자 → 보존된 사용자 탭으로 이동합니다.
3. 복원할 사용자 계정에서 체크박스를 클릭합니다.

4.

Restore (복구) 버튼을 클릭합니다.



5.

Confirmation(확인) 대화 상자에서 OK (확인) 버튼을 클릭합니다.

IdM 웹 UI는 녹색 확인을 표시하고 사용자 계정을 **Active users** (활성 사용자) 탭으로 이동합니다.

11.8. IDM 웹 UI에서 사용자 삭제

사용자를 삭제하는 것은 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 **IdM**을 통해 더 이상 액세스할 수 없습니다.

다음은 삭제할 수 있습니다.

- 활성 사용자 정보 - **IdM 웹 UI**는 다음과 같은 옵션을 제공합니다.
 - 일시적으로 사용자 보존

자세한 내용은 **IdM 웹 UI에서 활성 사용자 보존**을 참조하십시오.
 - 영구적으로 삭제
- 스테이징 사용자 - 단계 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 users(단계 사용자)** 탭
- 보존된 사용자 탭

사전 요구 사항

- **IdM 웹 UI** 또는 사용자 관리자 역할을 관리하는 관리자 권한.

절차

1. **IdM 웹 UI에 로그인합니다.**

자세한 내용은 웹 [브라우저에서 IdM 웹 UI 액세스](#)를 참조하십시오.
2. 사용자 → 활성 사용자 탭으로 이동합니다.

또는 사용자 → 사용자 단계 또는 사용자 → 보존된 사용자의 사용자 계정을 삭제할 수 있습니다.
3. **Delete(삭제)** 아이콘을 클릭합니다.
4. **Remove users(사용자 제거)** 대화 상자에서 **Delete mode(삭제 모드)** 라디오 버튼을 삭제 하도록 전환합니다.
5. **Delete(삭제)** 단추를 클릭합니다.

사용자 계정이 **IdM**에서 영구적으로 삭제되었습니다.

12장. ANSIBLE 플레이북을 사용하여 사용자 계정 관리

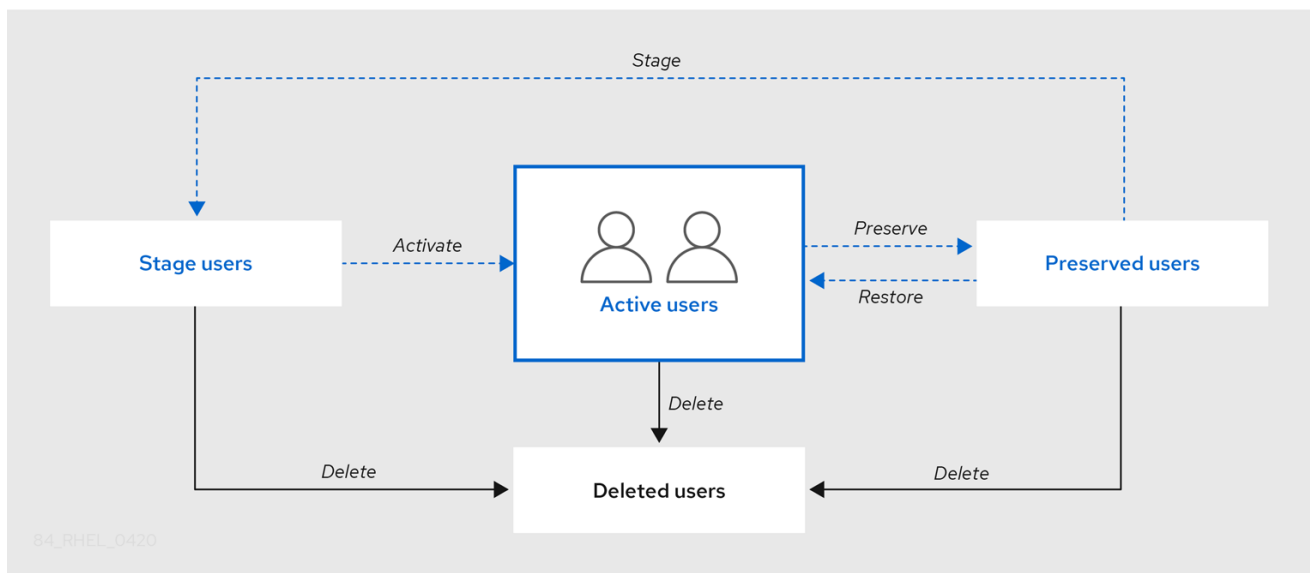
Ansible 플레이북을 사용하여 **IdM**에서 사용자를 관리할 수 있습니다. **사용자 라이프사이클** 을 제공한 후 이 장에서는 다음 작업에 **Ansible** 플레이북을 사용하는 방법을 설명합니다.

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

12.1. 사용자 라이프 사이클

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

- 스테이징 사용자는 인증이 허용되지 않습니다. 초기 상태입니다. 활성 사용자에게 필요한 일부 사용자 계정 속성은 예를 들어 그룹 멤버십을 설정할 수 없습니다.
- 활성 사용자는 인증을 허용합니다. 필요한 모든 사용자 계정 속성은 이 상태에서 설정해야 합니다.
- 보존된 사용자는 비활성으로 간주되고 **IdM**에 인증할 수 없는 이전 활성 사용자입니다. 보존된 사용자는 활성 사용자로 보유한 계정 속성의 대부분을 유지하지만 사용자 그룹의 일부가 아닙니다.



IdM 데이터베이스에서 영구적으로 사용자 항목을 삭제할 수 있습니다.



중요

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

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



주의

admin 사용자를 삭제하지 마십시오. **admin**은 IdM에 필요한 사전 정의된 사용자 이므로 이 작업으로 인해 특정 명령에서 문제가 발생합니다. 대체 **admin** 사용자를 정의하고 사용하려는 경우 하나 이상의 다른 사용자에게 **admin** 권한을 부여한 후 **ipa** 사용자 비활성화 **admin**으로 사전 정의된 **admin** 사용자를 비활성화합니다.



주의

IdM에 로컬 사용자를 추가하지 마십시오. NSS(Name Service Switch)는 로컬 사용자 및 그룹을 확인하기 전에 항상 IdM 사용자 및 그룹을 확인합니다. 즉, IdM 그룹 멤버십은 로컬 사용자에게 작동하지 않습니다.

12.2. 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 여기에 **ipaserver**를 정의합니다.

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

2. **IdM**에 있는 사용자의 데이터를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 간소화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/user/add-user.yml` 파일에서 예제를 복사하고 수정할 수 있습니다. 예를 들어 **idm_user**라는 사용자를 생성하고 **Password123**을 사용자 암호로 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

```

사용자를 추가하려면 다음 옵션을 사용해야 합니다.

- **name:** 로그인 이름
- **first:** 첫 번째 이름 문자열
- **last:** 성 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 이미 암호를 사용하여 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-ldM-user.yml

```

검증 단계

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

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

2. 관리자용 **Kerberos** 티켓을 요청합니다.

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

3. *idm_user*에 대한 정보를 요청합니다.

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

이름이 *idm_user*인 사용자는 **IdM**에 있습니다.

12.3. ANSIBLE PLAYBOOK을 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 여기에 **ipaserver**를 정의합니다.

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

2. **IdM**에서 확인할 사용자의 데이터로 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 간소화

하기 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 생성하고, `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 플레이북을 실행할 때마다 사용자 암호를 다시 설정합니다. 플레이북이 마지막으로 실행된 이후 사용자가 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 사용자 계정이 **IdM**에 있는지 확인할 수 있습니다.

1. 관리자로 **ipaserver**에 로그인합니다.

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

2. **idm_user_1**에 대한 정보 표시:

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

IdM에 **idm_user_1**이라는 사용자가 있습니다.

12.4. ANSIBLE 플레이북을 사용하여 JSON 파일에서 여러 **IDM** 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 여기에 **ipaserver**를 정의합니다.

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

- 2.

필요한 작업을 사용하여 **Ansible** 플레이북 파일을 생성합니다. 확인하고자 하는 사용자의 데이터로 **JSON** 파일을 참조합니다. 이 단계를 간소화하기 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 간소화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/user/users.json` 파일에서 예제를 복사하고 수정할 수 있습니다. 예를 들어 `idm_user_1`, `idm_user_2` 및 `idm_user_3` 사용자를 생성하고, `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 파일 및 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 IdM에 있는지 확인할 수 있습니다.

1. 관리자로 **ipaserver**에 로그인합니다.

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

2. **idm_user_1**에 대한 정보 표시:

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

IdM에 **idm_user_1**이라는 사용자가 있습니다.

12.5. ANSIBLE 플레이북을 사용하여 사용자가 없는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 특정 사용자가 IdM에 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 여기에 **ipaserver**를 정의합니다.

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

2.

IdM에 없는 사용자로 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 간소화하기 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 파일 및 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.

관리자로 **ipaserver** 에 로그인합니다.

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

2.

`idm_user_1` 에 대한 정보를 요청합니다.

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

이름이 `idm_user_1` 인 사용자는 IdM에 없습니다.

추가 리소스

- **/usr/share/doc/ansible-freeipa/** 디렉터리에 있는 **README-user.md Markdown** 파일에서 사용자 보존, 삭제, 비활성화, 잠금 해제 및 삭제와 같은 다른 **IdM** 사용자 관련 작업에 대한 샘플 **Ansible** 플레이북을 볼 수 있습니다. 파일에는 **ipausers** 변수 정의도 포함되어 있습니다.
- **/usr/share/doc/ansible-freeipa/playbooks/user** 디렉터리에 샘플 **Ansible** 플레이북을 볼 수도 있습니다.

13장. IDM CLI에서 사용자 그룹 관리

이 장에서는 **IdM CLI**를 사용한 사용자 그룹 관리 방법을 소개합니다.

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

IdM(Identity Management)의 사용자 그룹은 다음을 포함할 수 있습니다.

- **IdM 사용자**
- **기타 IdM 사용자 그룹**
- **외부 사용자 - IdM 외부에 존재하는 사용자**

13.1. IDM의 다양한 그룹 유형

IdM은 다음과 같은 유형의 그룹을 지원합니다.

POSIX 그룹(기본값)

POSIX 그룹은 구성원에 대해 **Linux POSIX** 특성을 지원합니다. **Active Directory**와 상호 작용하는 그룹은 **POSIX** 특성을 사용할 수 없습니다.

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

postIX 이외의 그룹

POST 이외의 그룹은 **POSIX** 특성을 지원하지 않습니다. 예를 들어, 이러한 그룹에는 **GID**가 정의되어 있지 않습니다.

이 유형의 그룹의 모든 멤버는 **IdM** 도메인에 속해야 합니다.

외부 그룹

외부 그룹을 사용하여 **IdM** 도메인 외부의 **ID** 저장소에 있는 그룹 구성원을 추가합니다(예:).

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

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

표 13.1. 기본적으로 생성된 사용자 그룹

그룹 이름	기본 그룹 멤버
ipausers	모든 IdM 사용자
admins	기본 admin 사용자를 포함하여 관리 권한이 있는 사용자
편집기	이는 더 이상 특수 권한이 없는 레거시 그룹입니다.
신뢰 관리자	Active Directory 신뢰 관리를 위한 권한이 있는 사용자

사용자를 사용자 그룹에 추가하면 사용자에게 그룹과 관련된 권한 및 정책이 부여됩니다. 예를 들어 사용자에게 관리 권한을 부여하려면 사용자를 **admins** 그룹에 추가합니다.



주의

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

또한 **IdM**에서 새 사용자가 생성될 때마다 **IdM**은 기본적으로 사용자 개인 그룹을 생성합니다. 개인 그룹에 대한 자세한 내용은 개인 그룹 없이 사용자 추가를 참조하십시오.

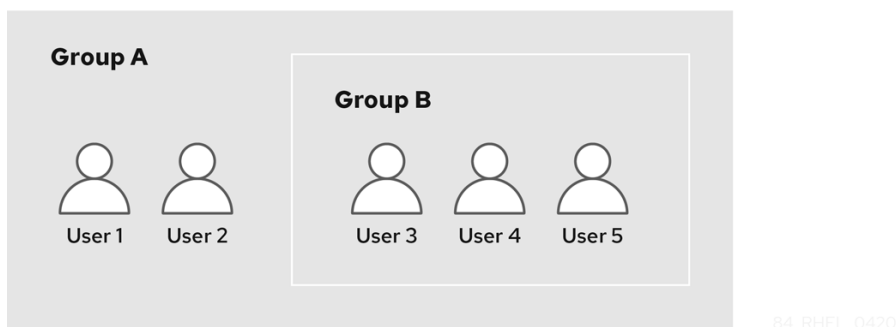
13.2. 직접 및 간접 그룹 구성원

IdM의 사용자 그룹 속성은 직접 및 간접 구성원 모두에 적용됩니다. B 그룹이 A 그룹의 구성원이면 B 그룹의 모든 사용자는 A 그룹의 간접 구성원으로 간주됩니다.

예를 들어 다음 다이어그램에서 다음을 수행합니다.

- User 1 및 User 2는 A 그룹의 직접 구성원입니다.
- User 3, User 4 및 User 5는 A 그룹의 간접 구성원입니다.

그림 13.1. 직접 및 간접 그룹 멤버십



사용자 그룹 A에 대한 암호 정책을 설정하면 이 정책은 사용자 그룹 B의 모든 사용자에게도 적용됩니다.

13.3. IDM CLI를 사용하여 사용자 그룹 추가

이 섹션에서는 IdM CLI를 사용하여 사용자 그룹을 추가하는 방법에 대해 설명합니다.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 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.

- **--POSIX** 이외의 그룹을 만드는 **nonposix**
- 외부 그룹을 만드는 **--external**

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

--gid= custom_GID 옵션을 사용하여 사용자 그룹을 추가할 때 사용자 지정 **GID**를 지정할 수 있습니다. 이 작업을 수행하는 경우 **ID** 충돌을 피하십시오. 사용자 지정 **GID**를 지정하지 않으면 **IdM**에서 사용 가능한 **ID** 범위에서 **GID**를 자동으로 할당합니다.



주의

IdM에 로컬 그룹을 추가하지 마십시오. **NSS(Name Service Switch)**는 로컬 사용자 및 그룹을 확인하기 전에 항상 **IdM** 사용자 및 그룹을 확인합니다. 즉, **IdM** 그룹 멤버십은 로컬 사용자에게 작동하지 않습니다.

13.4. 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 모든 비**POSIX** 그룹을 표시합니다.
 - **ipa group-find --external** 명령을 사용하여 모든 외부 그룹을 표시합니다.

다양한 그룹 유형에 대한 자세한 내용은 [IdM의 다양한 그룹 유형에서](#) 참조하십시오.

13.5. IDM CLI를 사용하여 사용자 그룹 삭제

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [Using kinit to log in IdM에서](#) 수동으로 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"
-----
```

13.6. IDM CLI를 사용하여 사용자 그룹에 멤버 추가

이 섹션에서는 **IdM CLI**를 사용하여 사용자 그룹에 멤버를 추가하는 방법을 설명합니다. 사용자 및 사용자 그룹을 사용자 그룹의 멤버로 추가할 수 있습니다. 자세한 내용은 [IdM 및 직접 및 간접 그룹 구성원의 다양한 그룹 유형을](#) 참조하십시오.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

절차

- `ipa group-add-member` 명령을 사용하여 사용자 그룹에 멤버를 추가합니다.

다음 옵션을 사용하여 멤버 유형을 지정합니다.
 - `--users` 는 IdM 사용자 추가
 - `--external` 은 IdM 도메인 외부에 있는 사용자 추가: `sudoAIN \user_name` 또는 `user_name@domain` 형식으로
 - `--groups` 는 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 구성원으로 추가할 때 재귀 그룹을 만들지 마십시오. 예를 들어 그룹 A가 그룹 B의 구성원인 경우 그룹 A의 구성원으로 그룹 B를 추가하지 마십시오. 반복적인 그룹은 예측할 수 없는 동작이 발생할 수 있습니다.



참고

사용자 그룹에 멤버를 추가하고 나면 업데이트에 Identity Management 환경의 모든 클라이언트에 분배하는 데 다소 시간이 걸릴 수 있습니다. 이는 지정된 호스트에서 사용자, 그룹 및 넷 그룹을 확인할 때 먼저 **SSSD(System Security Services Daemon)**에서 캐시를 살펴보고 서버 조회를 누락되거나 만료된 레코드에 대해서만 수행하기 때문입니다.

13.7. 사용자 개인 그룹없이 사용자 추가

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

- **UPG**의 이름은 새로 생성된 사용자와 동일합니다.
- 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.

13.7.1. 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.

다음 두 가지 방법으로 이 작업을 수행할 수 있습니다.

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

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

IdM에는 새 사용자에게 대한 **GID**가 필요하지만 기본 사용자 그룹 **ipausers**는 **POSTIX** 그룹이 아니므로 **GID**가 연결되어 있지 않기 때문입니다. 지정한 **GID**는 기존 그룹에 해당하지 않아도 됩니다.



참고

GID를 지정해도 새 그룹이 생성되지 않습니다. IdM에서 속성이 필요하므로 새 사용자의 **GID** 속성만 설정합니다.

13.7.2. 개인 그룹이 전역적으로 활성화된 경우 사용자 개인 그룹이 없는 사용자 추가

UPG(사용자 개인 그룹)를 시스템에서 활성화된 경우에도 사용자를 만들 수 있습니다. 이를 위해서는 새 사용자에게 대한 **GID**를 수동으로 설정해야 합니다. 이 기능이 필요한 이유에 대한 자세한 내용은 [13.7.1 절. “사용자 개인 그룹이 없는 사용자”](#)을 참조하십시오.

절차

- IdM이 **UPG**를 생성하지 못하도록 하려면 **ipa user-add** 명령에 **--noprivate** 옵션을 추가합니다.

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

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

13.7.3. 모든 사용자에게 대해 사용자 개인 그룹 전역 비활성화

사용자 개인 그룹(**UPG**)을 전역적으로 비활성화할 수 있습니다. 이로 인해 모든 새 사용자에게 대한 **UPG**가 생성되지 않습니다. 기존 사용자는 이 변경의 영향을 받지 않습니다.

절차

1. 관리자 권한을 얻습니다.

```
$ kinit admin
```

2. IdM은 **Directory Server Managed Entries** 플러그인을 사용하여 **UPG**를 관리합니다. 플러그인 인스턴스를 나열합니다.

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

3.

IdM이 UPG를 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 활성화하려면 `ipa-managed-entries -e "UPG Definition" enable` 명령을 사용합니다.

4.

Directory Server를 다시 시작하여 새 구성을 로드합니다.

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

UPG를 비활성화한 다음 사용자를 추가하려면 **GID**를 지정해야 합니다. 자세한 내용은 사용자 **개인 그룹이 전역적으로 비활성화된 경우 사용자 추가**를 참조하십시오.

검증 단계

•

UPG가 전역적으로 비활성화되어 있는지 확인하려면 **disable** 명령을 다시 사용합니다.

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

13.7.4. 사용자 개인 그룹이 전역적으로 비활성화된 경우 사용자 추가

사용자 개인 그룹(UPG)이 전역적으로 비활성화되면 IdM에서 새 사용자에게 **GID**를 자동으로 할당하지 않습니다. 사용자를 성공적으로 추가하려면 수동으로 또는 **automember** 규칙을 사용하여 **GID**를 할당해야 합니다. 이 기능이 필요한 이유에 대한 자세한 내용은 [13.7.1절. “사용자 개인 그룹이 없는 사용자”](#)를 참조하십시오.

전제 조건

•

모든 사용자에게 대해 UPG를 전역적으로 비활성화해야 합니다. 자세한 내용은 [모든 사용자에게 대해 사용자 개인 그룹 전역 비활성화](#)를 참조하십시오.

절차

- **UPG**를 생성할 때 새 사용자를 추가하는 것이 성공했는지 확인하려면 다음 중 하나를 선택하십시오.
 - 새 사용자를 추가할 때 사용자 지정 **GID**를 지정합니다. **GID**는 기존 사용자 그룹에 해당하지 않아도 됩니다.

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

13.8. IDM CLI를 사용하여 IDM 사용자 그룹에 멤버 관리자로 사용자 또는 그룹 추가

이 섹션에서는 **IdM CLI**를 사용하여 사용자 또는 그룹을 **IdM** 사용자 그룹에 구성원 관리자로 추가하는 방법에 대해 설명합니다. 멤버 관리자는 **IdM** 사용자 그룹에 사용자 또는 그룹을 추가할 수 있지만 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.
- 멤버 관리자로 추가하려는 사용자 또는 그룹의 이름과 관리하려는 그룹의 이름이 있어야 합니다.

절차

- **ipa group-add-member-manager** 명령을 사용하여 사용자를 **IdM** 사용자 그룹에 멤버 관리자로 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 이제 **group_a**의 멤버를 관리할 수 있습니다.

- **ipa group-add-member-manager** 명령을 사용하여 그룹을 **IdM** 사용자 그룹에 멤버 관리자로 추가합니다.

예를 들어 **group_admins** 그룹을 **group_a**의 멤버 관리자로 추가하려면 :

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

group_admins 그룹이 이제 **group_a**의 멤버를 관리할 수 있습니다.



참고

사용자 그룹에 멤버 관리자를 추가한 후 업데이트에 **Identity Management** 환경의 모든 클라이언트에 분배하는 데 다소 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.

13.9. IDM CLI를 사용하여 그룹 멤버 보기

이 섹션에서는 **IdM CLI**를 사용하여 그룹의 구성원을 보는 방법을 설명합니다. 직접 및 간접 그룹 구성

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

절차:

- 그룹의 구성원을 나열하려면 `ipa group-show group_name` 명령을 사용합니다. 예를 들면 다음과 같습니다.

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

참고

간접 구성원 목록에는 신뢰할 수 있는 **Active Directory** 도메인의 외부 사용자가 포함되지 않습니다. **Active Directory** 신뢰 사용자 오브젝트는 **Identity Management** 내에 **LDAP** 오브젝트로 존재하지 않기 때문에 ID 관리 인터페이스에 표시되지 않습니다.

13.10. 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 [Using kinit to log in IdM에서](#) 수동으로 참조하십시오.

절차

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

다음 옵션을 사용하여 제거할 멤버를 지정합니다.

- **--users** 는 IdM 사용자를 제거합니다
- **--external** 은 **pamAIn \user_name** 또는 **user_name@domain** 형식으로 IdM 도메인 외부에 있는 사용자를 제거합니다.
- **--groups** 는 IdM 사용자 그룹을 제거합니다

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
```

13.11. IDM CLI를 사용하여 IDM 사용자 그룹에서 사용자 또는 그룹을 구성원 관리자로 제거

이 섹션에서는 **IdM CLI**를 사용하여 **IdM** 사용자 그룹에서 사용자 또는 그룹을 멤버 관리자로 제거하는 방법을 설명합니다. 멤버 관리자는 **IdM** 사용자 그룹에서 사용자 또는 그룹을 제거할 수 있지만 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 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 **user 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 테스트는 더 이상 **group_a**의 구성원을 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
```

그룹 **group_admins**는 더 이상 **group_a**의 구성원을 관리할 수 없습니다.



참고

사용자 그룹에서 멤버 관리자를 제거한 후 업데이트에 **Identity Management** 환경의 모든 클라이언트에 분배하는 데 다소 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.

14장. IDM 웹 UI에서 사용자 그룹 관리

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

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

IdM(Identity Management)의 사용자 그룹은 다음을 포함할 수 있습니다.

- **IdM 사용자**
- **기타 IdM 사용자 그룹**
- **외부 사용자 - IdM 외부에 존재하는 사용자**

14.1. IDM의 다양한 그룹 유형

IdM은 다음과 같은 유형의 그룹을 지원합니다.

POSIX 그룹(기본값)

POSIX 그룹은 구성원에 대해 **Linux POSIX** 특성을 지원합니다. **Active Directory**와 상호 작용하는 그룹은 **POSIX** 특성을 사용할 수 없습니다.

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

postlX 이외의 그룹

POST 이외의 그룹은 **POSIX** 특성을 지원하지 않습니다. 예를 들어, 이러한 그룹에는 **GID**가 정의되어 있지 않습니다.

이 유형의 그룹의 모든 멤버는 **IdM** 도메인에 속해야 합니다.

외부 그룹

외부 그룹을 사용하여 **IdM** 도메인 외부의 **ID** 저장소에 있는 그룹 구성원을 추가합니다(예:).

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

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

표 14.1. 기본적으로 생성된 사용자 그룹

그룹 이름	기본 그룹 멤버
ipausers	모든 IdM 사용자
admins	기본 admin 사용자를 포함하여 관리 권한이 있는 사용자
편집기	이는 더 이상 특수 권한이 없는 레거시 그룹입니다.
신뢰 관리자	Active Directory 신뢰 관리를 위한 권한이 있는 사용자

사용자를 사용자 그룹에 추가하면 사용자에게 그룹과 관련된 권한 및 정책이 부여됩니다. 예를 들어 사용자에게 관리 권한을 부여하려면 사용자를 **admins** 그룹에 추가합니다.



주의

admins 그룹을 삭제하지 마십시오. **admins** 는 IdM에 필요한 사전 정의된 그룹이므로 이 작업으로 인해 특정 명령에 문제가 발생합니다.

또한 IdM에서 새 사용자가 생성될 때마다 IdM은 기본적으로 사용자 개인 그룹을 생성합니다. 개인 그룹에 대한 자세한 내용은 개인 그룹 없이 사용자 추가를 참조하십시오.

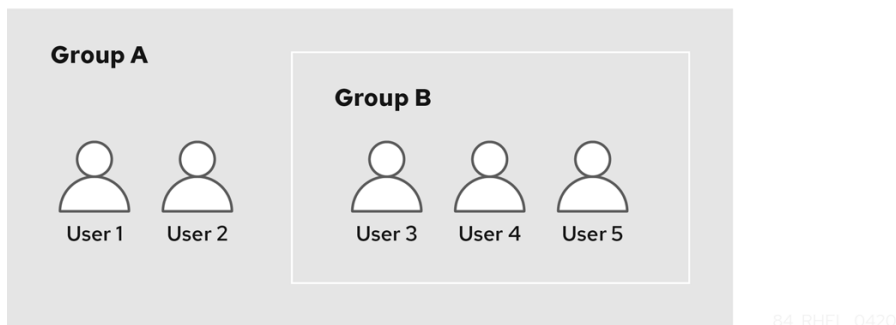
14.2. 직접 및 간접 그룹 구성원

IdM의 사용자 그룹 속성은 직접 및 간접 구성원 모두에 적용됩니다. B 그룹이 A 그룹의 구성원이면 B 그룹의 모든 사용자는 A 그룹의 간접 구성원으로 간주됩니다.

예를 들어 다음 다이어그램에서 다음을 수행합니다.

- User 1 및 User 2는 A 그룹의 직접 구성원입니다.
- User 3, User 4 및 User 5는 A 그룹의 간접 구성원입니다.

그림 14.1. 직접 및 간접 그룹 멤버십



사용자 그룹 A에 대한 암호 정책을 설정하면 이 정책은 사용자 그룹 B의 모든 사용자에게도 적용됩니다.

14.3. IDM 웹 UI를 사용하여 사용자 그룹 추가

이 섹션에서는 IdM 웹 UI를 사용하여 사용자 그룹을 추가하는 방법에 대해 설명합니다.

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.

절차

1.

ID → 그룹을 클릭하고 왼쪽 사이드바에서 사용자 그룹을 선택합니다.

2.

Add(추가) 를 클릭하여 그룹 추가를 시작합니다.

3.

그룹에 대한 정보를 입력합니다. 사용자 그룹 유형에 대한 자세한 내용은 [IdM의 다양한 그룹 유형에서](#) 참조하십시오.

그룹에 대한 사용자 지정 **GID**를 지정할 수 있습니다. 이 작업을 수행하는 경우 **ID** 충돌을 피하십시오. 사용자 지정 **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(추가) 를 클릭하여 확인합니다.

14.4. IDM 웹 UI를 사용하여 사용자 그룹 삭제

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

사전 요구 사항

- **IdM 웹 UI에 로그인되어 있습니다.**

절차

1. **ID → 그룹을 클릭하고 사용자 그룹을 선택합니다.**
2. **삭제할 그룹을 선택합니다.**
3. **삭제를 클릭합니다.**
4. **Delete(삭제)를 클릭하여 확인합니다.**

14.5. IDM 웹 UI를 사용하여 사용자 그룹에 멤버 추가

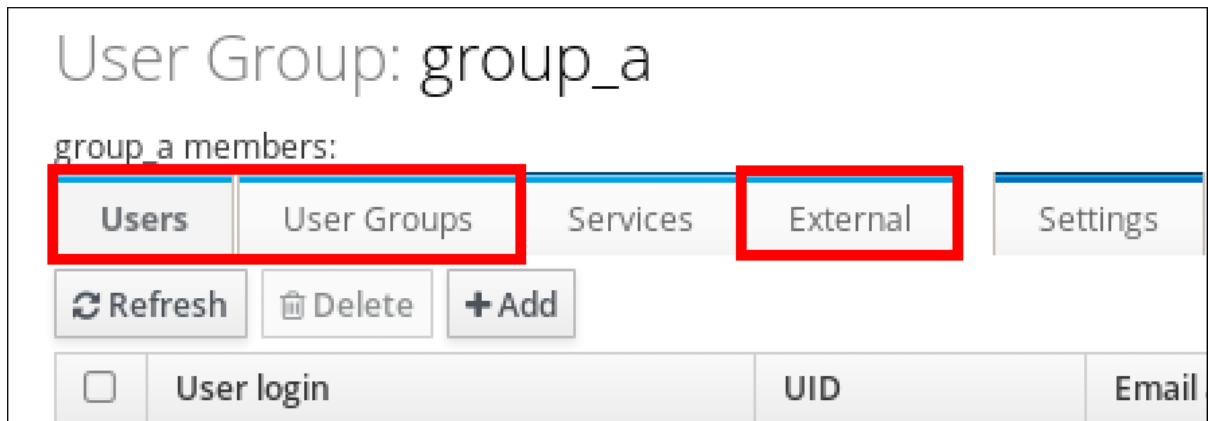
사용자 및 사용자 그룹을 사용자 그룹의 멤버로 추가할 수 있습니다. 자세한 내용은 [IdM 및 직접 및 간접 그룹 구성원의 다양한 그룹 유형을 참조하십시오.](#)

사전 요구 사항

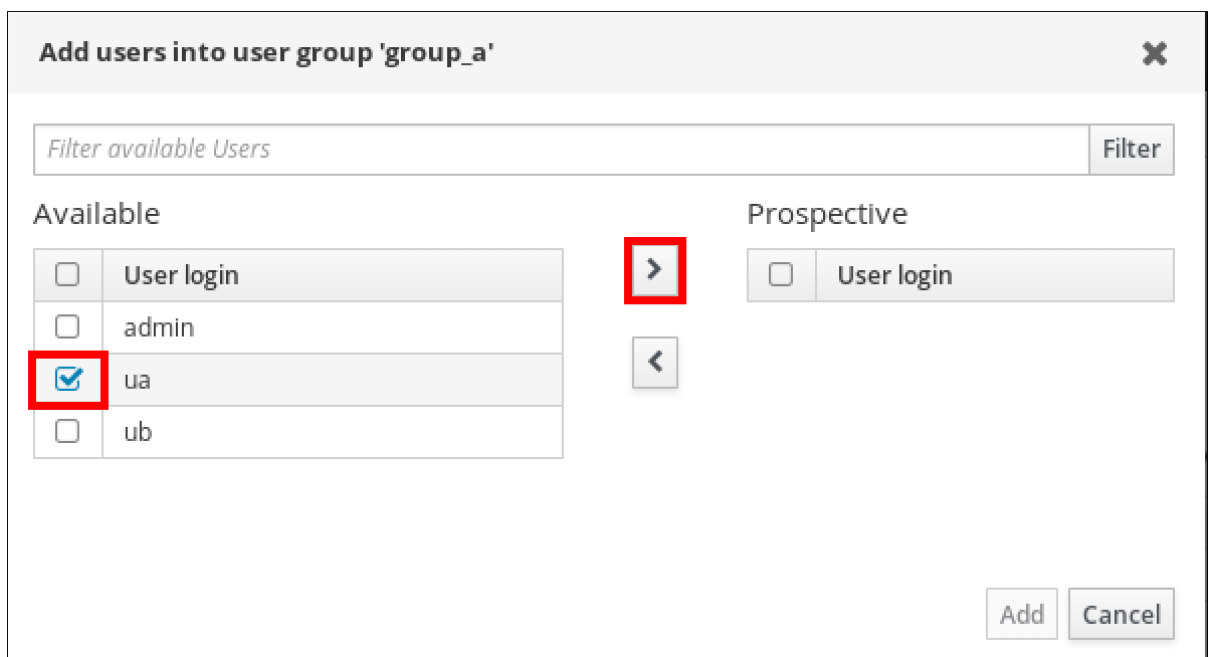
- **IdM 웹 UI에 로그인되어 있습니다.**

절차

1. **ID → 그룹을 클릭하고 왼쪽 사이드바 에서 사용자 그룹을 선택합니다.**
2. **그룹의 이름을 클릭합니다.**
3. **추가할 그룹 구성원 유형을 선택합니다. 사용자, 사용자 그룹 또는 외부.**



4. **추가를 클릭합니다.**
5. **추가할 구성원 하나 이상 옆에 있는 확인란을 선택합니다.**
6. **오른쪽 화살표를 클릭하여 선택한 구성원을 그룹으로 이동합니다.**



7. **Add(추가) 를 클릭하여 확인합니다.**

14.6. 웹 UI를 사용하여 IDM 사용자 그룹에 멤버 관리자로 사용자 또는 그룹 추가

이 섹션에서는 웹 UI를 사용하여 IDM 사용자 그룹에 구성원 관리자로 사용자 또는 그룹을 추가하는 방법에 대해 설명합니다. 멤버 관리자는 IDM 사용자 그룹에 사용자 또는 그룹을 추가할 수 있지만 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

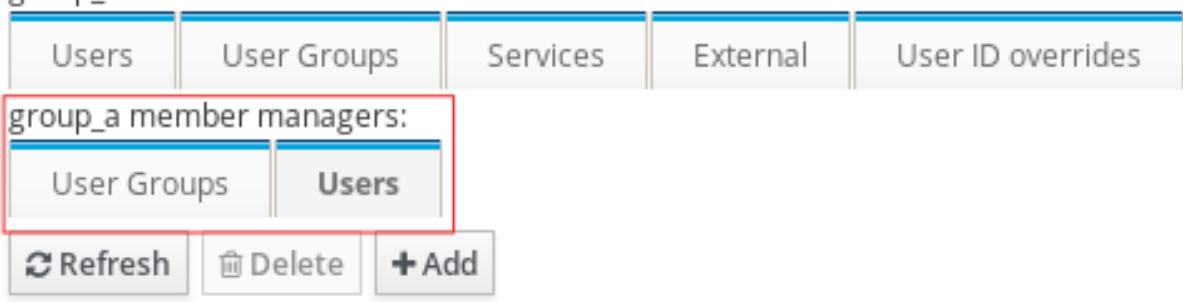
- **IdM 웹 UI에 로그인되어 있습니다.**
- **멤버 관리자로 추가하려는 사용자 또는 그룹의 이름과 관리하려는 그룹의 이름이 있어야 합니다.**

절차

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

User Group: group_a

group_a members:



4. **추가를 클릭합니다.**
5. **추가할 구성원 하나 이상 옆에 있는 확인란을 선택합니다.**
6. **오른쪽 화살표를 클릭하여 선택한 구성원을 그룹으로 이동합니다.**

Add users as member managers for user group 'group_a'
✕

Filter available Users
Filter

Available

☐	User login
☐	admin
☒	test1
☐	test2
☐	test_user
☐	test_user2
☐	tuser3

➤

➤

Prospective

☐	User login

Add

Cancel

7.

Add(추가) 를 클릭하여 확인합니다.



참고

사용자 그룹에 멤버 관리자를 추가한 후 업데이트에 **Identity Management** 환경의 모든 클라이언트에 분배하는 데 다소 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

☐	Group name
☐	project_admins

추가 리소스

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

14.7. IDM 웹 UI를 사용하여 그룹 구성원 보기

이 섹션에서는 **IdM 웹 UI**를 사용하여 그룹의 구성원을 보는 방법을 설명합니다. 직접 및 간접 그룹 구성원을 모두 볼 수 있습니다. 자세한 내용은 [직접 및 간접 그룹 구성원](#)을 참조하십시오.

사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.

절차

1. **ID** → 그룹을 선택합니다.
2. 왼쪽 사이드바에서 **User Groups** 를 선택합니다.
3. 보려는 그룹의 이름을 클릭합니다.
4. **Direct Membership** 과 **Indirect Membership** 간에 전환합니다.

Refresh	Delete	Add	Show Results ☒ Direct Membership ☐ Indirect Membership			
☐	User login	UID	Email address	Telephone Number	Job Title	

14.8. 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. 그룹의 이름을 클릭합니다.
3. 삭제할 그룹 구성원 유형을 선택합니다. 사용자, 사용자 그룹 또는 외부.

User Group: group_a

group_a members:

Users	User Groups	Services	External	Settings

Refresh	Delete	Add

☐	User login	UID	Email

4. 제거할 멤버 옆에 있는 확인란을 선택합니다.
5. 삭제를 클릭합니다.
6. **Delete(삭제)** 를 클릭하여 확인합니다.

14.9. WEB 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 특성을 변경할 수는 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 → 그룹**을 클릭하고 왼쪽 사이드바에서 사용자 그룹을 선택합니다.
2. 그룹의 이름을 클릭합니다.
3. 삭제할 멤버 관리자 유형을 선택합니다. 사용자 또는 사용자 그룹.

User Group: group_a

group_a members:



4. 제거할 멤버 관리자 옆의 확인란을 선택합니다.
5. 삭제를 클릭합니다.

6.

Delete(삭제) 를 클릭하여 확인합니다.

참고

사용자 그룹에서 멤버 관리자를 제거한 후 업데이트에 **Identity Management** 환경의 모든 클라이언트에 분배하는 데 다소 시간이 걸릴 수 있습니다.

검증 단계

- 사용자 또는 사용자 그룹의 멤버 관리자 목록에서 사용자 또는 사용자 그룹이 제거되었는지 확인합니다.

User Group: project

project members:

Users	User Groups	Services

project member managers:

User Groups	Users (1)

Refresh	Delete	Add

☐	Group name
No entries.	

추가 리소스

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

15장. 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 그룹은 다음을 포함할 수 있습니다.

- **IdM 사용자**
- **기타 IdM 사용자 그룹**
- **외부 사용자 - IdM 외부에 존재하는 사용자**

섹션에는 다음 항목이 포함되어 있습니다.

- **IdM의 다양한 그룹 유형**
- **직접 및 간접 그룹 구성원**
- **Ansible Playbook을 사용하여 IdM 그룹 및 그룹 구성원이 있는지 확인**
- **Ansible 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 있는지 확인**
- **Ansible 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 없는지 확인합니다.**

15.1. IDM의 다양한 그룹 유형

IdM은 다음과 같은 유형의 그룹을 지원합니다.

POSIX 그룹(기본값)

POSIX 그룹은 구성원에 대해 **Linux POSIX** 특성을 지원합니다. **Active Directory**와 상호 작용하는 그룹은 **POSIX** 특성을 사용할 수 없습니다.

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

postlX 이외의 그룹

POST 이외의 그룹은 **POSIX** 특성을 지원하지 않습니다. 예를 들어, 이러한 그룹에는 **GID**가 정의되어 있지 않습니다.

이 유형의 그룹의 모든 멤버는 **IdM** 도메인에 속해야 합니다.

외부 그룹

외부 그룹을 사용하여 **IdM** 도메인 외부의 **ID** 저장소에 있는 그룹 구성원을 추가합니다(예:).

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

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

표 15.1. 기본적으로 생성된 사용자 그룹

그룹 이름	기본 그룹 멤버
ipausers	모든 IdM 사용자
admins	기본 admin 사용자를 포함하여 관리 권한이 있는 사용자
편집기	이는 더 이상 특수 권한이 없는 레거시 그룹입니다.
신뢰 관리자	Active Directory 신뢰 관리를 위한 권한이 있는 사용자

사용자를 사용자 그룹에 추가하면 사용자에게 그룹과 관련된 권한 및 정책이 부여됩니다. 예를 들어 사용자에게 관리 권한을 부여하려면 사용자를 **admins** 그룹에 추가합니다.



주의

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

또한 **IdM**에서 새 사용자가 생성될 때마다 **IdM**은 기본적으로 사용자 개인 그룹을 생성합니다. 개인 그룹에 대한 자세한 내용은 개인 그룹 없이 사용자 추가를 참조하십시오.

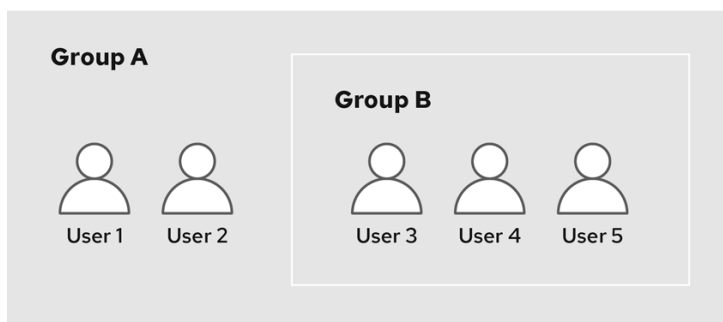
15.2. 직접 및 간접 그룹 구성원

IdM의 사용자 그룹 속성은 직접 및 간접 구성원 모두에 적용됩니다. **B** 그룹이 **A** 그룹의 구성원이면 **B** 그룹의 모든 사용자는 **A** 그룹의 간접 구성원으로 간주됩니다.

예를 들어 다음 다이어그램에서 다음을 수행합니다.

- **User 1** 및 **User 2**는 **A** 그룹의 직접 구성원입니다.
- **User 3**, **User 4** 및 **User 5**는 **A** 그룹의 간접 구성원입니다.

그림 15.1. 직접 및 간접 그룹 멤버십



84_RHEL_0420

사용자 그룹 **A**에 대한 암호 정책을 설정하면 이 정책은 사용자 그룹 **B**의 모든 사용자에게도 적용됩니다.

15.3. ANSIBLE PLAYBOOK을 사용하여 IDM 그룹 및 그룹 구성원이 있는지 확인

다음 절차에서는 사용자 및 사용자 그룹 모두 **Ansible** 플레이북을 사용하여 **IdM** 그룹 및 그룹 구성원이 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 여기에 **ipaserver**를 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:
        ipadmin_password: MySecret123
        name: ops
        gidnumber: 1234

    - name: Create group sysops
      ipagroup:
        ipadmin_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

ipa group-show 명령을 사용하여 **ops** 그룹에 **sysops** 및 **appops** 가 직접 멤버로 포함되고 **idm_user** 가 간접 구성원으로 포함되어 있는지 확인할 수 있습니다.

1.

관리자로 **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 - **idm_user** 사용자를 포함하는 후자는 **IdM**에 있습니다.

추가 리소스

- **Ansible**을 사용하여 사용자 그룹이 있는지 확인하는 방법에 대한 자세한 내용은 `/usr/share/doc/ansible-freeipa/README-group.md` Markdown 파일을 참조하십시오.

15.4. ANSIBLE PLAYBOOK을 사용하여 IDM 사용자 그룹에 멤버 관리자가 있는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 사용자 및 사용자 그룹 모두 **IdM** 멤버 관리자가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- 멤버 관리자로 추가하려는 사용자 또는 그룹의 이름과 관리하려는 그룹의 이름이 있어야 합니다.

절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 여기에 `ipaserver`를 정의합니다.

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

2. 필요한 사용자 및 그룹 구성원 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:
        ipadmin_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

ipa group-show 명령을 사용하여 **group_a** 그룹에 **test** 가 구성원 관리자로 포함되어 있고 **group_admins** 가 **group_a** 의 멤버 관리자인지 확인할 수 있습니다.

1.

관리자로 **ipaserver** 에 로그인합니다.

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

2.

managergroup1 에 대한 정보 표시 :

```
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** 도움말 페이지를 참조하십시오.

15.5. ANSIBLE PLAYBOOK을 사용하여 IDM 사용자 그룹에 멤버 관리자가 없는지 확인합니다.

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 여기에 **ipaserver**를 정의합니다.

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

2. 필요한 사용자 및 그룹 구성원 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:
        ipaadmin_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 구성원 관리자로 포함되어 있지 않은지, **group_admins**가 **group_a**의 멤버 관리자로 포함되어 있지 않은지 확인할 수 있습니다.

1. 관리자로 **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** 도움말 페이지를 참조하십시오.

16장. IDM CLI를 사용하여 그룹 멤버십 자동화

자동 그룹 멤버십을 사용하면 속성에 따라 자동으로 사용자와 호스트를 그룹에 할당할 수 있습니다. 예를 들면 다음을 수행할 수 있습니다.

- 직원 관리자, 위치 또는 기타 특성에 따라 직원의 사용자 항목을 그룹으로 나눕니다.
- 클래스, 위치 또는 기타 특성을 기반으로 호스트를 나눕니다.
- 모든 사용자 또는 모든 호스트를 하나의 글로벌 그룹에 추가합니다.

이 장에서는 다음 주제를 다룹니다.

- [자동 그룹 멤버십의 이점](#)
- [자동 구성원 규칙](#)
- [IdM CLI를 사용하여 자동 구성원 규칙 추가](#)
- [IdM CLI를 사용하여 자동 구성원 규칙에 조건 추가](#)
- [IdM CLI를 사용하여 기존 자동 구성원 규칙 보기](#)
- [IdM CLI를 사용하여 자동 구성원 규칙 삭제](#)
- [IdM CLI를 사용하여 automember 규칙에서 조건 제거](#)
- [IdM CLI를 사용하여 기존 항목에 자동 구성원 규칙 적용](#)

- **IdM CLI를 사용하여 기본 자동 구성원 그룹 구성**

16.1. 자동 그룹 멤버십의 이점

사용자에 대한 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 개선

사용자와 호스트는 엄격하게 정의되고 자동으로 평가된 기준에 따라 그룹에 할당됩니다.
- 그룹 기반 설정 관리 간소화

그룹에 대해 다양한 설정이 정의되고 개별 그룹 구성원에 적용됩니다(예: **sudo** 규칙, 자동 마운트 또는 액세스 제어). 사용자와 호스트를 그룹에 추가하면 이러한 설정을 자동으로 관리하기가 쉬워집니다.

16.2. 자동 구성원 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 후에는 관리자가 조건을 추가합니다. 대상 그룹에서 포함하거나 제외되는 사용자 또는 호스트를 지정합니다.

- 포함 조건

사용자 또는 호스트 항목이 포함 조건을 충족하면 대상 그룹에 포함됩니다.
- 배타적 조건

사용자 또는 호스트 항목이 독점 조건을 충족하면 대상 그룹에 포함되지 않습니다.

조건은 **Perl** 호환 정규 표현식(**PCRE**) 형식으로 정규 표현식으로 지정됩니다. **PCRE**에 대한 자세한 내용은 **pcresyntax(3)** 도움말 페이지를 참조하십시오.



참고

IdM은 포함 조건보다 먼저 배타적 조건을 평가합니다. 충돌이 발생하는 경우 독점 조건이 포함된 조건보다 우선합니다.

automember 규칙은 향후 생성된 모든 항목에 적용됩니다. 이러한 항목은 지정된 대상 그룹에 자동으로 추가됩니다. 항목이 여러 **automember** 규칙에 지정된 조건을 충족하면 모든 해당 그룹에 추가됩니다.

기존 항목은 새 규칙의 영향을 받지 않습니다. 기존 항목을 변경하려면 **IdM CLI**를 사용하여 기존 항목에 자동 구성원 규칙 적용을 참조하십시오.

16.3. IdM CLI를 사용하여 자동 구성원 규칙 추가

이 섹션에서는 **IdM CLI**를 사용하여 자동 구성원 규칙을 추가하는 방법에 대해 설명합니다. 자동 구성원 규칙에 대한 자세한 내용은 **Automember rules** 을 참조하십시오.

automember 규칙을 추가한 후에는 **automember** 규칙에 조건 추가에 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 구성원 규칙 적용을 참조하십시오.

사전 요구 사항



관리자로 로그인해야 합니다. 자세한 내용은 **Using kinit to log in IdM**에서 수동으로 참조하십시오.

- 새 규칙의 대상 그룹은 **IdM**에 있어야 합니다.

절차

1. **ipa automember-add** 명령을 입력하여 **automember** 규칙을 추가합니다.
2. 메시지가 표시되면 다음을 지정합니다.

- 자동 구성원 규칙. 대상 그룹 이름입니다.
- 그룹화 유형. 이는 규칙에서 사용자 그룹을 대상으로 하는지 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 자동 구성원 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 구성원 규칙 및 조건을 표시할 수 있습니다.

16.4. IDM CLI를 사용하여 자동 구성원 규칙에 조건 추가

이 섹션에서는 **IdM CLI**를 사용하여 **automember** 규칙에 조건을 추가하는 방법을 설명합니다. 자동 구성원 규칙에 대한 자세한 내용은 **Automember rules** 을 참조하십시오.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 **Using kinit to log in IdM**에서 수동으로 참조하

십시오.

- 대상 규칙은 **IdM**에 있어야 합니다. 자세한 내용은 **IdM CLI를 사용하여 자동 구성원 규칙 추가**를 참조하십시오.

절차

1. **ipa automember-add-condition** 명령을 사용하여 하나 이상의 포함 또는 독점적 조건을 정의합니다.
2. 메시지가 표시되면 다음을 지정합니다.
 - 자동 구성원 규칙. 대상 규칙 이름입니다. 자세한 내용은 **자동 구성원 규칙**을 참조하십시오.
 - 특성 키. 이는 필터를 적용할 항목 속성을 지정합니다. 예를 들어 사용자를 위한 **uid**.
 - 그룹화 유형. 이는 규칙에서 사용자 그룹을 대상으로 하는지 또는 호스트 그룹을 대상으로 하는지 여부를 지정합니다. 사용자 그룹을 대상으로 지정하려면 그룹을 입력합니다. 호스트 그룹을 대상으로 지정하려면 **hostgroup**을 입력합니다.
 - 포함 정규식 및 배타적 정규식. 이러한 조건은 하나 이상의 조건을 정규 표현식으로 지정합니다. 하나의 조건만 지정하려면 다른 조건을 입력하라는 메시지가 표시되면 **Enter** 키를 누릅니다.

예를 들어 다음 조건은 사용자 로그인 특성(**uid**)에서 임의의 값(.)을 가진 모든 사용자를 대상으로합니다.

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

또 다른 예로 **automembership** 규칙을 사용하여 **AD(Active Directory)**에서 동기화된 모든 **Windows** 사용자를 대상으로 할 수 있습니다. 이를 위해 모든 **AD** 사용자가 공유하는 **objects Class** 특성 에서 모든 사용자를 대상으로 하는 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 CLI를 사용하여 기존 automember 규칙 보기를 사용하여 IdM에 기존 자동 구성원 규칙 및 조건을 표시할 수 있습니다.**

16.5. IDM CLI를 사용하여 기존 자동 구성원 규칙 보기

이 섹션에서는 **IdM CLI**를 사용하여 기존 자동 구성원 규칙을 보는 방법에 대해 설명합니다.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 **Using kinit to log in IdM**에서 수동으로 참조하십시오.

절차

1. **ipa automember-find** 명령을 입력합니다.
2. 메시지가 표시되면 그룹 유형을 지정합니다.

- 사용자 그룹을 대상으로 지정하려면 그룹을 입력합니다.
- 호스트 그룹을 대상으로 지정하려면 **hostgroup** 을 입력합니다.

예를 들면 다음과 같습니다.

```
$ ipa automember-find
Grouping Type: group
-----
1 rules matched
-----
Automember Rule: user_group
Inclusive Regex: uid=.*
-----
Number of entries returned 1
-----
```

16.6. IDM CLI를 사용하여 자동 구성원 규칙 삭제

이 섹션에서는 **IdM CLI**를 사용하여 자동 구성원 규칙을 삭제하는 방법을 설명합니다.

자동 구성원 규칙을 삭제하면 규칙과 연결된 모든 조건도 삭제됩니다. 규칙에서 특정 조건만 제거하려면 **IdM CLI를 사용하여 automember 규칙에서 조건 제거**를 참조하십시오.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

절차

1. **ipa automember-del** 명령을 입력합니다.
2. 메시지가 표시되면 다음을 지정합니다.
 - 자동 구성원 규칙. 삭제할 규칙입니다.
 -

규칙 그룹화. 이는 삭제할 규칙이 사용자 그룹 또는 호스트 그룹에 대한지 여부를 지정합니다. 그룹 또는 호스트 그룹을 입력합니다.

16.7. IDM CLI를 사용하여 AUTOMEMBER 규칙에서 조건 제거

이 섹션에서는 **automember** 규칙에서 특정 조건을 제거하는 방법을 설명합니다.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

절차

1. **ipa automember-remove-condition** 명령을 입력합니다.
2. 메시지가 표시되면 다음을 지정합니다.
 - 자동 구성원 규칙. 조건을 제거할 규칙의 이름입니다.
 - 특성 키. 대상 항목 속성입니다. 예를 들어 사용자를 위한 **uid**.
 - 그룹화 유형. 이는 삭제할 조건이 사용자 그룹 또는 호스트 그룹에 대한지 여부를 지정합니다. 그룹 또는 호스트 그룹을 입력합니다.
 - 포함 정규식 및 배타적 정규식. 이러한 조건은 제거할 조건을 지정합니다. 하나의 조건만 지정하려면 다른 조건을 입력하라는 메시지가 표시되면 **Enter** 키를 누릅니다.

예를 들면 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
-----
```

16.8. IDM CLI를 사용하여 기존 항목에 자동 구성원 규칙 적용

automember 규칙은 규칙이 추가된 후 생성된 사용자 및 호스트 항목에 자동으로 적용됩니다. 규칙을 추가하기 전에 존재하는 항목에 소급으로 적용되지 않습니다.

이전에 추가한 항목에 자동 구성원 규칙을 적용하려면 자동 멤버십을 수동으로 다시 빌드해야 합니다. 자동 멤버십을 다시 빌드하면 기존의 모든 자동 구성원 규칙을 다시 평가하여 모든 사용자 또는 호스트 항목 또는 특정 항목에 적용합니다.



참고

자동 멤버십을 다시 빌드 해도 항목이 더 이상 그룹의 포함 조건과 일치하지 않는 경우에도 그룹에서 사용자 또는 호스트 항목이 제거되지 않습니다. 수동으로 제거하려면 **IdM CLI** 를 사용하여 사용자 그룹에서 멤버 제거 또는 **CLI**를 사용하여 **IdM** 호스트 그룹 구성원 제거를 참조하십시오.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 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 빌드하려면 **--users=target_user** 옵션을 사용합니다.

```
$ ipa automember-rebuild --users=target_user1 --users=target_user2
```

```
-----
Automember rebuild task finished. Processed (2) entries.
-----
```

- 지정된 호스트 또는 호스트에 대한 자동 멤버십을 다시 빌드하려면 **--hosts=client.idm.example.com** 옵션을 사용합니다.

16.9. IDM CLI를 사용하여 기본 자동 구성원 그룹 구성

기본 자동 구성원 그룹을 구성하면 **automember** 규칙과 일치하지 않는 새 사용자 또는 호스트 항목이 기본 그룹에 자동으로 추가됩니다.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 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 **default(fallback)** 그룹입니다.
 - 타겟이 사용자 그룹인지 또는 호스트 그룹인지 여부를 지정하는 유형 그룹화입니다. 사용자 그룹을 대상으로 지정하려면 그룹을 입력합니다. 호스트 그룹을 대상으로 지정하려면

hostgroup 을 입력합니다.

예를 들면 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 들면 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
```

17장. IDM 웹 UI를 사용하여 그룹 멤버십 자동화

자동 그룹 멤버십을 사용하면 속성에 따라 자동으로 사용자와 호스트를 그룹에 할당할 수 있습니다. 예를 들면 다음을 수행할 수 있습니다.

- 직원 관리자, 위치 또는 기타 특성에 따라 직원의 사용자 항목을 그룹으로 나눕니다.
- 클래스, 위치 또는 기타 특성을 기반으로 호스트를 나눕니다.
- 모든 사용자 또는 모든 호스트를 하나의 글로벌 그룹에 추가합니다.

이 장에서는 다음 주제를 다룹니다.

- [자동 그룹 멤버십의 이점](#)
- [자동 구성원 규칙](#)
- [IdM 웹 UI를 사용하여 자동 구성원 규칙 추가](#)
- [IdM 웹 UI를 사용하여 자동 구성원 규칙에 조건 추가](#)
- [IdM 웹 UI를 사용하여 기존 자동 구성원 규칙 및 조건 보기](#)
- [IdM 웹 UI를 사용하여 자동 구성원 규칙 삭제](#)
- [IdM 웹 UI를 사용하여 automember 규칙에서 조건 제거](#)
- [IdM 웹 UI를 사용하여 기존 항목에 자동 구성원 규칙 적용](#)

- [IdM 웹 UI를 사용하여 기본 사용자 그룹 구성](#)
- [IdM 웹 UI를 사용하여 기본 호스트 그룹 구성](#)

17.1. 자동 그룹 멤버십의 이점

사용자에 대한 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 개선

사용자와 호스트는 엄격하게 정의되고 자동으로 평가된 기준에 따라 그룹에 할당됩니다.
- 그룹 기반 설정 관리 간소화

그룹에 대해 다양한 설정이 정의되고 개별 그룹 구성원에 적용됩니다(예: **sudo** 규칙, 자동 마운트 또는 액세스 제어). 사용자와 호스트를 그룹에 추가하면 이러한 설정을 자동으로 관리하기가 쉬워집니다.

17.2. 자동 구성원 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 후에는 관리자가 조건을 추가합니다. 대상 그룹에서 포함하거나 제외되는 사용자 또는 호스트를 지정합니다.

- 포함 조건

사용자 또는 호스트 항목이 포함 조건을 충족하면 대상 그룹에 포함됩니다.

•

배타적 조건

사용자 또는 호스트 항목이 독점 조건을 충족하면 대상 그룹에 포함되지 않습니다.

조건은 **Perl** 호환 정규 표현식(**PCRE**) 형식으로 정규 표현식으로 지정됩니다. **PCRE**에 대한 자세한 내용은 **pcresyntax(3)** 도움말 페이지를 참조하십시오.



참고

IdM은 포함 조건보다 먼저 배타적 조건을 평가합니다. 충돌이 발생하는 경우 독점 조건이 포함된 조건보다 우선합니다.

automember 규칙은 향후 생성된 모든 항목에 적용됩니다. 이러한 항목은 지정된 대상 그룹에 자동으로 추가됩니다. 항목이 여러 **automember** 규칙에 지정된 조건을 충족하면 모든 해당 그룹에 추가됩니다.

기존 항목은 새 규칙의 영향을 받지 않습니다. 기존 항목을 변경하려면 **IdM 웹 UI를 사용하여 기존 항목에 자동 구성원 규칙 적용**을 참조하십시오.

17.3. IDM 웹 UI를 사용하여 자동 구성원 규칙 추가

이 섹션에서는 **IdM 웹 UI**를 사용하여 자동 구성원 규칙을 추가하는 방법에 대해 설명합니다. 자동 구성원 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 구성원 규칙 적용**을 참조하십시오.

사전 요구 사항

•

IdM 웹 UI에 로그인되어 있습니다.

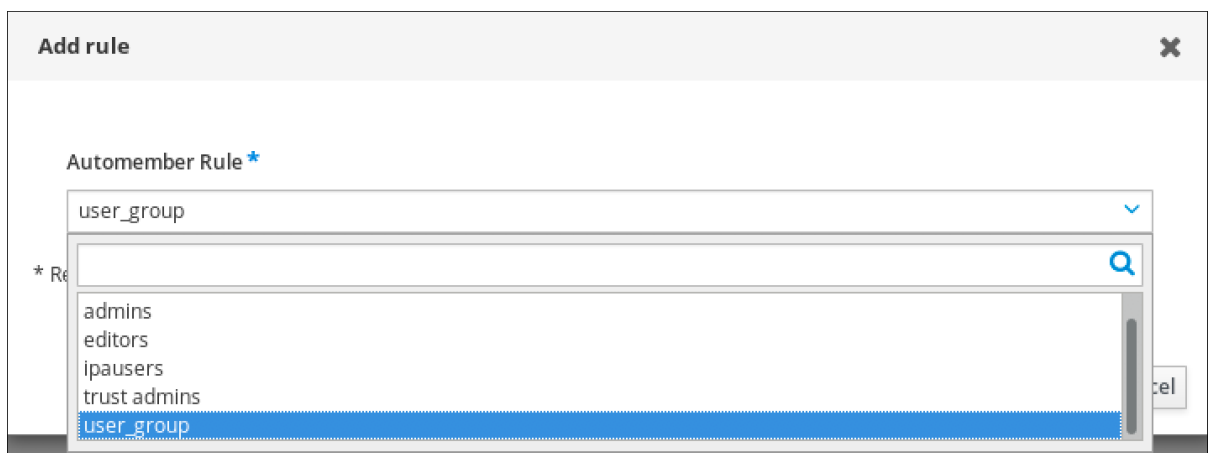
•

admins 그룹의 멤버여야 합니다.

- 새 규칙의 대상 그룹은 **IdM**에 있습니다.

절차

1. **ID** → 자동 구성원을 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택합니다.
2. 추가를 클릭합니다.
3. **Automember** 규칙 필드에서 규칙이 적용할 그룹을 선택합니다. 대상 그룹 이름입니다.



4. **Add(추가)** 를 클릭하여 확인합니다.
5. 선택 사항: **IdM 웹 UI**를 사용하여 **automember** 규칙에 조건 추가에 설명된 절차를 사용하여 새 규칙에 조건을 추가할 수 있습니다.

17.4. IdM 웹 UI를 사용하여 자동 구성원 규칙에 조건 추가

이 섹션에서는 **IdM 웹 UI**를 사용하여 **automember** 규칙에 조건을 추가하는 방법을 설명합니다. 자동 구성원 규칙에 대한 자세한 내용은 [Automember rules](#) 을 참조하십시오.

사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.

- 대상 규칙은 **IdM**에 있습니다.

절차

1. **ID** → 자동 구성원을 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택합니다.
2. 조건을 추가할 규칙을 클릭합니다.
3. **Inclusive** (포함) 또는 **Exclusive** (독점) 섹션에서 **Add**(추가)를 클릭합니다.

User group rule: user_group

General

Automember Rule

user_group

Description

Inclusive

☐	Attribute	Expression	Delete	+Add
☐	uid	.*		

Exclusive

☐	Attribute	Expression	Delete	+Add

4. **Attribute** (특성) 필드에서 **required** 속성을 선택합니다(예: *uid*).
5. **Expression**(표현식) 필드에 정규 표현식을 정의합니다.
6. 추가를 클릭합니다.

예를 들어 다음 조건은 사용자 **ID(uid)** 속성에서 임의의 값(.*을 가진 모든 사용자를 대상으로 합니다.

Add Condition into automember [X]

Attribute: uid

Expression *: .*

* Required field

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

17.5. IDM 웹 UI를 사용하여 기존 자동 구성원 규칙 및 조건 보기

이 섹션에서는 **IdM 웹 UI**를 사용하여 기존의 자동 구성원 규칙 및 조건을 보는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.

절차

1. **ID** → 자동 구성원을 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택하여 해당 자동 구성원 규칙을 확인합니다.
2. 선택 사항: 규칙을 클릭하여 **Inclusive** 또는 **Exclusive** 섹션에서 해당 규칙의 조건을 확인합니다.

User group rule: user_group

Refresh

Revert

Save

General

Automember Rule

user_group

Description

Inclusive

☐	Attribute	Expression	Delete	Add
☐	uid	,*		

Exclusive

☐	Attribute	Expression	Delete	Add

17.6. IDM 웹 UI를 사용하여 자동 구성원 규칙 삭제

이 섹션에서는 **IdM 웹 UI**를 사용하여 자동 구성원 규칙을 삭제하는 방법을 설명합니다.

자동 구성원 규칙을 삭제하면 규칙과 연결된 모든 조건도 삭제됩니다. 규칙에서 특정 조건만 제거하려면 **IdM 웹 UI를 사용하여 automember 규칙에서 조건** 제거를 참조하십시오.

사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.

절차

1. **ID** → 자동 구성원을 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택하여 해당 자동 구성원 규칙을 확인합니다.

2. 제거할 규칙 옆에 있는 확인란을 선택합니다.
3. 삭제를 클릭합니다.

User group rules

Search

Default user group

☐	Automember Rule	Description
☒	user_group	

1 rules matched

4. **Delete(삭제)** 를 클릭하여 확인합니다.

17.7. IDM 웹 UI를 사용하여 AUTOMEMBER 규칙에서 조건 제거

이 섹션에서는 **IdM 웹 UI**를 사용하여 **automember** 규칙에서 특정 조건을 제거하는 방법을 설명합니다.

사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.

절차

1. **ID** → 자동 구성원을 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택하여 해당 자동 구성원 규칙을 확인합니다.
2. 규칙을 클릭하여 **Inclusive** 또는 **Exclusive** 섹션에서 해당 규칙의 조건을 확인합니다.
3. 제거할 조건 옆에 있는 확인란을 선택합니다.

4. 삭제 버튼을 클릭합니다.

User group rule: user_group

Refresh Revert Save

General

Automember Rule

user_group

Description

Inclusive

☐	Attribute	Expression	Delete + Add
☒	uid	, *	Delete

Exclusive

☐	Attribute	Expression	Delete + Add

5. Delete(삭제) 버튼을 클릭하여 확인합니다.

17.8. IDM 웹 UI를 사용하여 기존 항목에 자동 구성원 규칙 적용

automember 규칙은 규칙이 추가된 후 생성된 사용자 및 호스트 항목에 자동으로 적용됩니다. 규칙을 추가하기 전에 존재하는 항목에 소급으로 적용되지 않습니다.

이전에 추가한 항목에 자동 구성원 규칙을 적용하려면 자동 멤버십을 수동으로 다시 빌드해야 합니다. 자동 멤버십을 다시 빌드하면 기존의 모든 자동 구성원 규칙을 다시 평가하여 모든 사용자 또는 호스트 항목 또는 특정 항목에 적용합니다.



참고

자동 멤버십을 다시 빌드 해도 항목이 더 이상 그룹의 포함 조건과 일치하지 않는 경우에도 그룹에서 사용자 또는 호스트 항목이 제거되지 않습니다. 수동으로 제거하려면 **IdM 웹 UI**를 사용하여 사용자 그룹에서 멤버 제거 또는 **IdM 웹 UI**에서 호스트 그룹 구성원 제거를 참조하십시오.

17.8.1. 모든 사용자 또는 호스트에 대한 자동 멤버십 다시 빌드

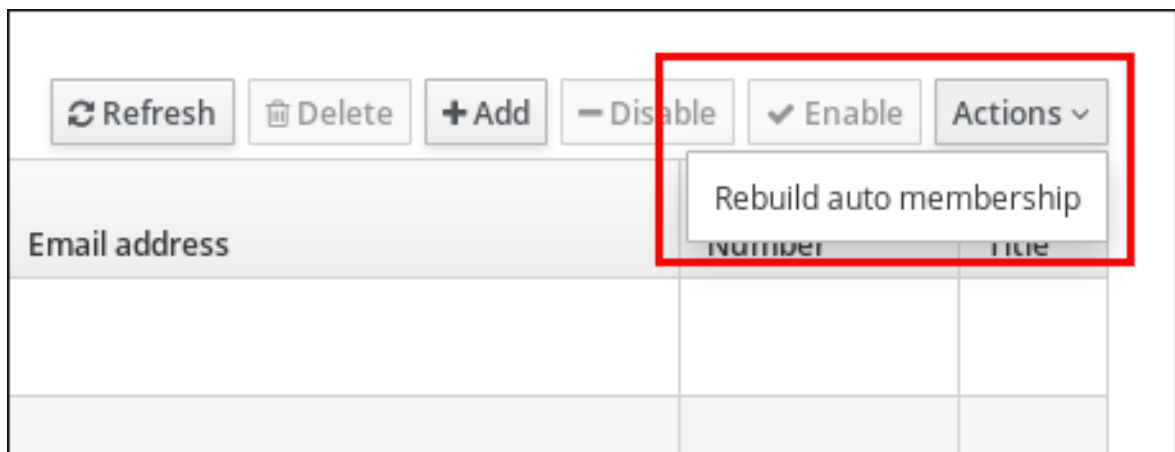
이 섹션에서는 모든 사용자 또는 호스트 항목에 대한 자동 멤버십을 다시 빌드하는 방법을 설명합니다.

사전 요구 사항

- **IdM 웹 UI에 로그인되어 있습니다.**
- **admins 그룹의 멤버여야 합니다.**

절차

1. **ID → 사용자 또는 호스트 를 선택합니다.**
2. **작업 → 자동 멤버십 다시 빌드를 클릭합니다.**



17.8.2. 단일 사용자 또는 호스트에 대한 자동 멤버십 다시 빌드

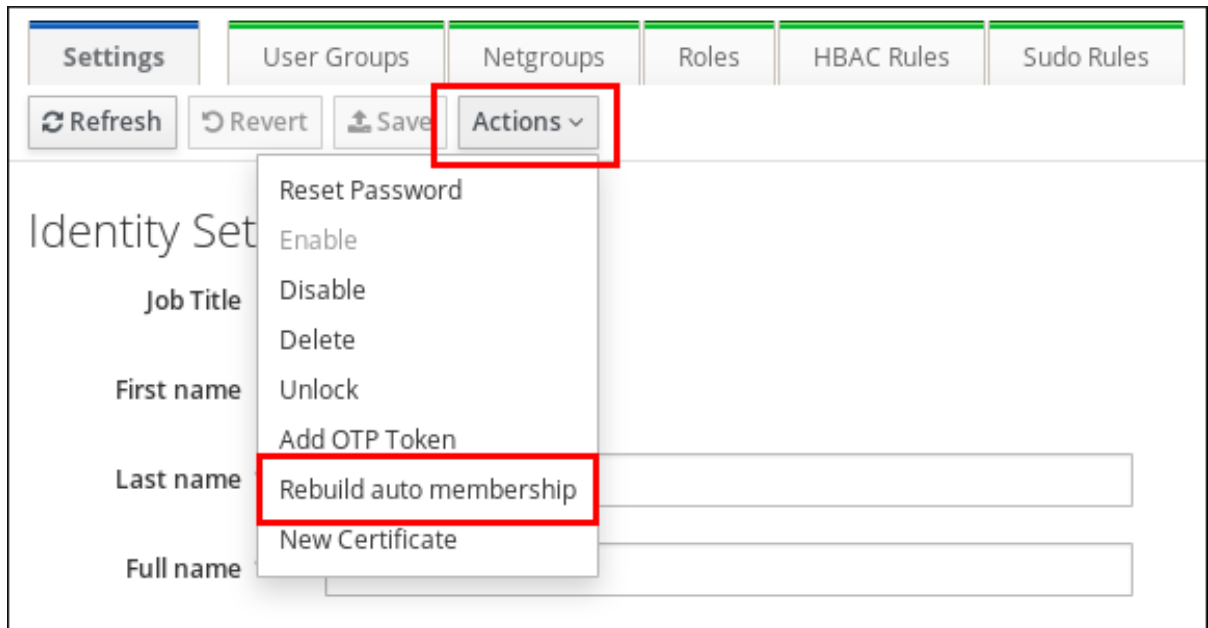
이 섹션에서는 특정 사용자 또는 호스트 항목에 대한 자동 멤버십을 다시 빌드하는 방법을 설명합니다.

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 멤버십 다시 빌드를 클릭합니다.



17.9. IDM 웹 UI를 사용하여 기본 사용자 그룹 구성

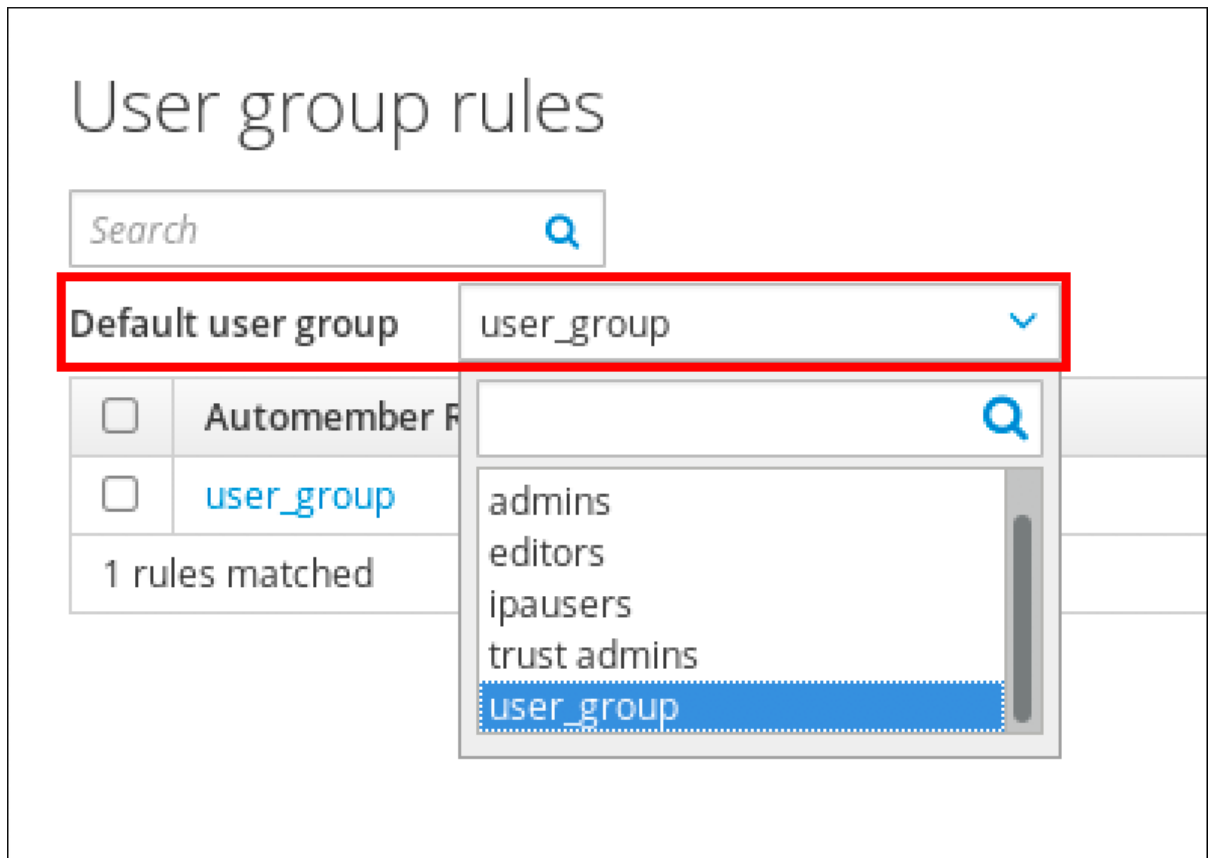
기본 사용자 그룹을 구성하면 **automember** 규칙과 일치하지 않는 새 사용자 항목이 이 기본 그룹에 자동으로 추가됩니다.

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.



17.10. IDM 웹 UI를 사용하여 기본 호스트 그룹 구성

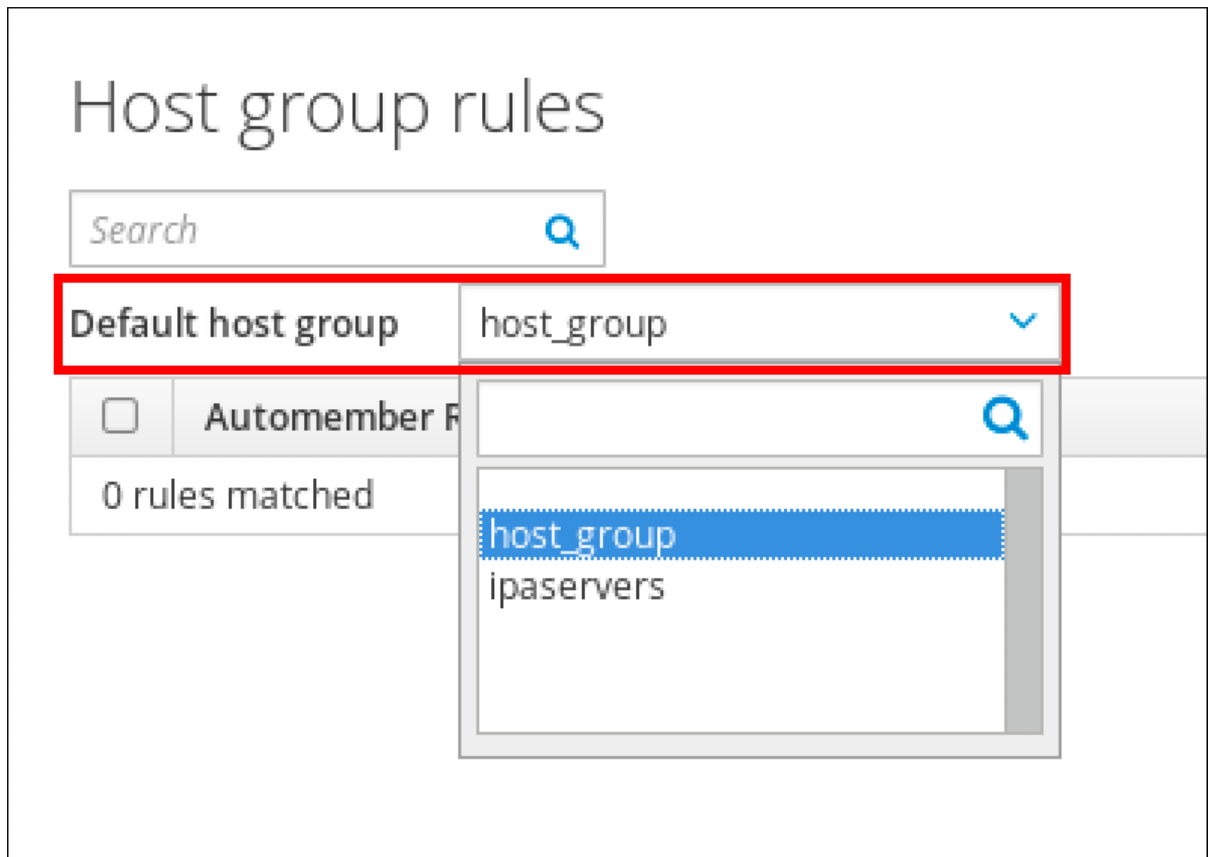
기본 호스트 그룹을 구성하면 **automember** 규칙과 일치하지 않는 새 호스트 항목이 이 기본 그룹에 자동으로 추가됩니다.

사전 요구 사항

- IDM 웹 UI에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.
- IDM에 기본값으로 설정할 대상 호스트 그룹이 있습니다.

절차

1. ID → 자동 구성원을 클릭하고 호스트 그룹 규칙을 선택합니다.
2. **Default** 호스트 그룹 필드에서 기본 호스트 그룹으로 설정할 그룹을 선택합니다.



18장. ANSIBLE을 사용하여 IDM의 그룹 멤버십 자동화

자동 그룹 멤버십을 사용하면 사용자 및 호스트 사용자 그룹과 호스트 그룹을 속성에 따라 자동으로 할당할 수 있습니다. 예를 들면 다음을 수행할 수 있습니다.

- 직원의 사용자 항목을 직원 관리자, 위치, 위치 또는 기타 특성에 따라 그룹으로 나눕니다. 명령행에 **ipa user-add --help** 를 입력하여 모든 속성을 나열할 수 있습니다.
- 호스트를 클래스, 위치 또는 기타 특성에 따라 그룹으로 나눕니다. 명령행에 **ipa host-add --help** 를 입력하여 모든 특성을 나열할 수 있습니다.
- 모든 사용자 또는 모든 호스트를 하나의 글로벌 그룹에 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 자동 관리 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 자동 관리 규칙이 없는지 확인합니다.**
- **Ansible을 사용하여 IdM 호스트 그룹 automember 규칙에 조건이 있는지 확인합니다.**

추가 리소스

- IdM의 사용자에게 대한 자세한 내용은 [Ansible 플레이북을 사용하여 사용자 계정](#) 관리를 참조하십시오.
- IdM의 호스트에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 호스트](#) 관리를 참조하십시오.
- IdM의 사용자 그룹에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 사용자 그룹](#) 관리를 참조하십시오.
- IdM의 호스트 그룹에 대한 자세한 내용은 [IdM CLI를 사용하여 호스트 그룹](#) 관리를 참조하십시오.

18.1. IDM 관리를 위한 ANSIBLE 제어 노드 준비

IdM(Identity Management)을 관리하는 시스템 관리자로서 Red Hat Ansible Engine을 사용하여 작업할 때 다음을 수행하는 것이 좋습니다.

- 홈 디렉터리에서 **Ansible** 플레이북 전용 하위 디렉터리(예: `~/MyPlaybooks`)를 생성합니다.
- `/usr/share/doc/ansible-freeipa/*` 및 `/usr/share/doc/rhel-system-roles/*` 디렉터리 및 하위 디렉터리에서 `~/MyPlaybooks` 디렉터리에 샘플 **Ansible** 플레이북을 복사 및 조정합니다.
- 인벤토리 파일을 `~/MyPlaybooks` 디렉터리에 포함합니다.

이 연습을 수행하면 모든 플레이북을 한 곳에서 찾을 수 있으며 **root** 권한을 호출하지 않고 플레이북을 실행할 수 있습니다.



참고

관리 노드에서 **root** 권한만 있으면 `ipaserver`, `ipareplica`, `ipa client` 및 `ipa backup` `ansible-freeipa` 역할을 실행할 수 있습니다. 이러한 역할을 수행하려면 디렉터리 및 `dnf` 소프트웨어 패키지 관리자에 대한 권한 있는 액세스 권한이 필요합니다.

이 섹션에서는 `~/MyPlaybooks` 디렉터리를 만들고 **Ansible** 플레이북을 저장하고 실행하는 데 사용할

수 있도록 구성하는 방법을 설명합니다.

사전 요구 사항

- 관리형 노드인 **server.idm.example.com** 및 **replica.idm.example.com**에 **IdM** 서버를 설치했습니다.
- 제어 노드에서 직접 관리형 노드인 **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 디렉터리에서 **Ansible** 구성 및 플레이북의 디렉터리를 생성합니다.

```
$ mkdir ~/MyPlaybooks/
```

2. **~/MyPlaybooks/** 디렉터리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

3. 다음 콘텐츠를 사용하여 **~/Myplaybooks/ansible.cfg** 파일을 만듭니다.

```
[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 이러한 위치에 있는 호스트에 대한 두 개의 호스트 그룹인 **eu** 와 **us** 를 정의합니다. 또한 이 구성은 **eu** 및 **us** 그룹의 모든 호스트를 포함하는 **ipaserver** 호스트 그룹을 정의합니다.

5.

[선택 사항] **SSH** 공개 및 개인 키를 생성합니다. 테스트 환경에서 액세스를 간소화하려면 개인 키에 암호를 설정하지 마십시오.

```
$ ssh-keygen
```

6.

SSH 공개 키를 각 관리 노드의 **IdM admin** 계정에 복사합니다.

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

이러한 명령을 입력할 때 **IdM** 관리자 암호를 입력해야 합니다.

추가 리소스

- [Ansible 플레이북을 사용하여 ID 관리 서버 설치.](#)
- [인벤토리 구축 방법.](#)

18.2. ANSIBLE을 사용하여 IDM 사용자 그룹에 대한 자동 관리 규칙이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 그룹에 대한 자동 관리 규칙이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 **testing_group** 사용자 그룹에 대해 **automember** 규칙이 있는지 확인합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **testing_group** 사용자 그룹이 **IdM**에 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)으로 **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. 편집을 위해 `automember-group-present-copy.yml` 파일을 엽니다.
4. `ipaautomember` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 **IdM** 관리자의 암호로 설정합니다.
- `name` 변수를 `testing_group` 으로 설정합니다.
- `automember_type` 변수를 `group` 으로 설정합니다.
- `state` 변수가 `present`로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

추가 리소스

- 자동 관리 규칙의 개념에 대한 자세한 내용은 자동 [그룹 멤버십](#) 및 [자동 관리 규칙의 이점](#)을 참조하십시오.
- **Ansible**을 사용하여 **automember** 규칙에 조건을 추가하는 방법에 대한 자세한 내용은 [Ansible 사용을 사용하여 IdM 사용자 그룹 automember 규칙에 조건이 있는지 확인합니다.](#)
- **ipaautomember** 모듈을 사용하는 추가 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README- automember.md** 파일. 이 파일에는 **ipaautomember** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/automember` 디렉토리.

18.3. ANSIBLE을 사용하여 지정된 조건이 **IDM** 사용자 그룹 **AUTOMEMBER** 규칙에 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 그룹의 **automember** 규칙

에 지정된 조건이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 **testing_group** 그룹에 대해 **automember** 규칙에 **UID** 관련 조건이 있어야 합니다. **.*** 조건을 지정하면 향후 모든 **IdM** 사용자가 자동으로 **testing_group**의 구성원이 되도록 합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **testing_group** 사용자 그룹과 **automember** 사용자 그룹 규칙이 **IdM**에 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)으로 **Ansible** 인벤토리 파일을 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 복사하고 이름을 로 지정합니다(예: **automember-usergroup-rule-present.yml**):


```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-present.yml automember-usergroup-rule-present.yml
```
3. 편집을 위해 **automember-usergroup-rule-present.yml** 파일을 엽니다.

4.

다음 매개변수를 수정하여 파일을 조정합니다.

- 예를 들어, 사용 사례에 해당하는 플레이북 이름을 로 변경합니다. 사용자 그룹 규칙 멤버 **present** 자동 메모리.
- 사용 사례에 해당하는 작업의 이름을 변경합니다. 예를 들면 다음과 같습니다. 사용자 그룹의 **automember** 조건이 있는지 확인합니다.
- **ipaautomember** 작업 섹션에서 다음 변수를 설정합니다.
 - **ipadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
 - **name** 변수를 **testing_group** 으로 설정합니다.
 - **automember_type** 변수를 **group** 으로 설정합니다.
 - **state** 변수가 **present**로 설정되어 있는지 확인합니다.
 - **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
 - 포함 키 변수를 **UID** 로 설정합니다.
 - 포함 표현식 변수를 로 설정합니다.*

이는 현재 예제에 대한 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 파일 및 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.

사용자를 추가합니다. 예를 들면 다음과 같습니다.

```

$ 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

- 기존 사용자 항목에 업데이트된 자동 공유 규칙을 [적용하려면 IdM CLI를 사용하여 기존 항목에 자동 메모리 규칙](#) 적용을 참조하십시오.
- 자동 관리 규칙의 개념에 대한 자세한 내용은 자동 [그룹 멤버십](#) 및 [자동 관리 규칙의 이점](#)을 참조하십시오.

- **ipaautomember** 모듈을 사용하는 추가 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-automember.md** 파일. 이 파일에는 **ipaautomember** 변수 정의도 포함되어 있습니다.
- `/usr/share/doc/ansible-freeipa/playbooks/automember` 디렉토리.

18.4. ANSIBLE을 사용하여 IDM 사용자 그룹 AUTOMEMBER 규칙에서 조건이 없는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 그룹의 자동 관리 규칙에서 상태가 없는지 확인하는 방법을 설명합니다. 이 예제에서는 **automember** 규칙에 조건이 없으면 초기화가 **dp** 인 사용자를 포함하도록 지정합니다. **automember** 규칙은 **testing_group** 그룹에 적용됩니다. 조건을 적용하면 초기 사용자의 **dp**가 **testing_group**의 구성원이 될 수 없게 됩니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **testing_group** 사용자 그룹과 **automember** 사용자 그룹 규칙이 **IdM**에 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)으로 **Ansible** 인벤토리 파일을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2.

`/usr/share/doc/ansible-freeipa/playbooks/automember/` 디렉터리에 있는 **automember-hostgroup-rule-absent.yml** Ansible 플레이북 파일을 복사하고 이름을 로 지정합니다(예: **automember-usergroup-rule-absent.yml**):

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-absent.yml automember-usergroup-rule-absent.yml
```

3.

편집을 위해 **automember-usergroup-rule-absent.yml** 파일을 엽니다.

4.

다음 매개변수를 수정하여 파일을 조정합니다.

- 예를 들어, 사용 사례에 해당하는 플레이북 이름을 로 변경합니다. 사용자 그룹 규칙 멤버가 없습니다.
- 사용 사례에 해당하는 작업의 이름을 변경합니다. 예를 들면 다음과 같습니다. 사용자 그룹의 **automember** 조건이 있는지 확인합니다.
- **ipaautomember** 작업 섹션에서 다음 변수를 설정합니다.
 - **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.
 - **name** 변수를 **testing_group** 으로 설정합니다.
 - **automember_type** 변수를 **group** 으로 설정합니다.
 - **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.
 - **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
 - 포함 키 변수를 **initials** 로 설정합니다.

○

포함 표현식 변수를 **dp** 로 설정합니다.

이는 현재 예제에 대한 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 파일 및 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 **Inclusive Regex: initials=dp** 항목이 없으면 **testing_group automember** 규칙에 지정된 조건이 포함되어 있지 않음을 확인합니다.

추가 리소스

- 기존 사용자 항목에 업데이트된 자동 공유 규칙을 적용하려면 **IdM CLI**를 사용하여 기존 항목에 자동 메모리 규칙 적용을 참조하십시오.
- 자동 관리 규칙의 개념에 대한 자세한 내용은 자동 그룹 멤버십 및 자동 관리 규칙의 이점을 참조하십시오.
- **ipaautomember** 모듈을 사용하는 추가 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-automember.md** 파일. 이 파일에는 **ipaautomember** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/automember` 디렉토리.

18.5. ANSIBLE을 사용하여 IDM 사용자 그룹에 대한 자동 관리 규칙이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 그룹에 대해 **automember** 규칙이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 **testing_group** 그룹에 대해 **automember** 규칙이 없습니다.



참고

자동 구성원 규칙을 삭제하면 규칙과 연결된 모든 조건도 삭제됩니다. 규칙에서 특정 조건만 제거하려면 **Ansible** 사용을 통해 **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)으로 **Ansible 인벤토리 파일**을 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. 편집을 위해 **automember-group-absent-copy.yml** 파일을 엽니다.
4. **ipaautomember** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 **testing_group** 으로 설정합니다.
- **automember_type** 변수를 **group** 으로 설정합니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory automember-group-absent.yml
```

추가 리소스

- 자동 관리 규칙의 개념에 대한 자세한 내용은 자동 [그룹 멤버십](#) 및 [자동 관리 규칙의 이점](#)을 참조하십시오.
- **ipaautomember** 모듈을 사용하는 추가 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-automember.md** 파일. 이 파일에는 **ipaautomember** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/automember` 디렉토리.

18.6. ANSIBLE을 사용하여 IDM 호스트 그룹 AUTOMEMBER 규칙에 조건이 있는지 확인합니다.

이 섹션에서는 **Ansible**을 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 구성원인지 확인하는 방법을 설명합니다. `*.example.org`는 **primary_dns_domain_hosts** 호스트 그룹의 멤버가 아닙니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
-

IdM에는 `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)으로 **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. 편집을 위해 `automember-hostgroup-rule-present-copy.yml` 파일을 엽니다.

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

- `ipadmin_password` 변수를 IdM 관리자의 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.
- 포함 키 변수가 **fqdn;** 으로 설정되어 있는지 확인합니다.
- 해당 포함 표현식 변수를 ***.idm.example.com** 으로 설정합니다.
- 독점 키 변수를 **fqdn;** 으로 설정합니다.
- 해당 배타적 표현식 변수를 ***.example.org** 로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory automember-hostgroup-rule-present-copy.yml
```

추가 리소스

- 기존 호스트 항목에 업데이트된 자동 공유 규칙을 적용하려면 **IdM CLI**를 사용하여 기존 항목에 자동 메모리 규칙 적용을 참조하십시오.
- 자동 관리 규칙의 개념에 대한 자세한 내용은 자동 그룹 멤버십 및 자동 관리 규칙의 이점을 참조하십시오.
- **ipaautomember** 모듈을 사용하는 추가 샘플 Ansible 플레이북은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README- automember.md** 파일. 이 파일에는 **ipaautomember** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/automember` 디렉토리.

19장. CLI를 사용하여 IDM에서 셀프 서비스 규칙 관리

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

19.1. IDM의 셀프 서비스 액세스 제어

셀프 서비스 액세스 제어 규칙은 **IdM(Identity Management)** 엔티티가 **IdM Directory Server** 항목에서 수행할 수 있는 작업을 정의합니다. 예를 들어 **IdM** 사용자는 자신의 암호를 업데이트할 수 있습니다.

이 제어 방법을 사용하면 인증된 **IdM** 엔티티에서 **LDAP** 항목 내에서 특정 속성을 편집할 수 있지만 전체 항목에서 작업을 추가하거나 삭제할 수는 없습니다.



주의

셀프 서비스 액세스 제어 규칙으로 작업할 때 주의하십시오. 액세스 제어 규칙을 부적절하게 구성하면 엔티티의 권한을 의도치 않게 높일 수 있습니다.

19.2. CLI를 사용하여 셀프 서비스 규칙 생성

이 절차에서는 **CLI(명령줄 인터페이스)**를 사용하여 **IdM**에서 셀프 서비스 액세스 규칙을 생성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

절차

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

--permissions

읽기 및 쓰기 권한을 설정합니다. **ACI**(액세스 제어 명령)가 부여됩니다.

--attrs

이 **ACI**에서 권한을 부여하는 전체 속성 목록을 설정합니다.

예를 들어 사용자가 자신의 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
```

19.3. CLI를 사용하여 셀프 서비스 규칙 편집

이 절차에서는 명령줄 인터페이스(CLI)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 [Using kinit to log in IdM에서](#) 수동으로 참조하십시오.

절차

1. **선택 사항:** **ipa selfservice-find** 명령을 사용하여 기존 셀프 서비스 규칙을 표시합니다.
2. **선택 사항:** **ipa selfservice-show** 명령을 사용하여 수정할 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 항상 기존 권한 및 속성의 전체 목록과 정의하고자 하는 새 권한 목록을 포함합니다.

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
```

19.4. CLI를 사용하여 셀프 서비스 규칙 삭제

이 절차에서는 명령줄 인터페이스(CLI)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 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

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.

20장. IDM 웹 UI를 사용하여 셀프 서비스 규칙 관리

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

20.1. IDM의 셀프 서비스 액세스 제어

셀프 서비스 액세스 제어 규칙은 **IdM(Identity Management)** 엔티티가 **IdM Directory Server** 항목에서 수행할 수 있는 작업을 정의합니다. 예를 들어 **IdM** 사용자는 자신의 암호를 업데이트할 수 있습니다.

이 제어 방법을 사용하면 인증된 **IdM** 엔티티에서 **LDAP** 항목 내에서 특정 속성을 편집할 수 있지만 전체 항목에서 작업을 추가하거나 삭제할 수는 없습니다.



주의

셀프 서비스 액세스 제어 규칙으로 작업할 때 주의하십시오. 액세스 제어 규칙을 부적절하게 구성하면 엔티티의 권한을 의도치 않게 높일 수 있습니다.

20.2. IDM 웹 UI를 사용하여 셀프 서비스 규칙 생성

다음 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.

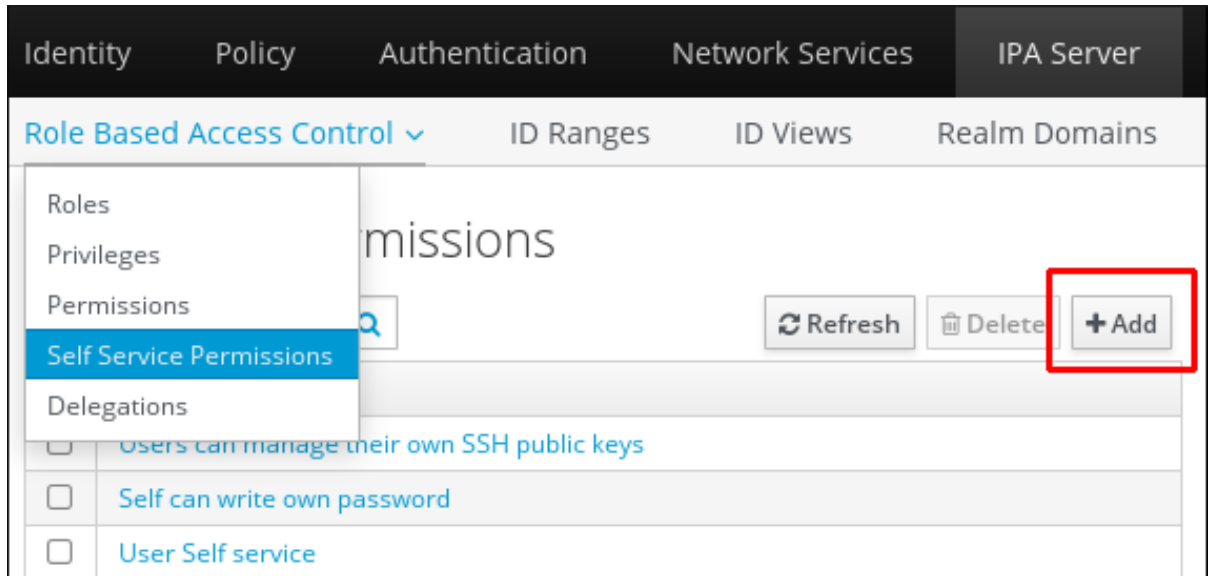
절차

1. **IPA Server** 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** (추가)를 클릭합니다.



3.

Add self Service Permission(셀프 서비스 권한 추가) 창이 열립니다. 셀프 서비스 이름 필드에 새 셀프 서비스 규칙의 이름을 입력합니다. 공백을 사용할 수 있습니다.

Add Self Service Permission

Self-service *
name Adding Personal Info

Attributes *
Filter Add

☐ audio	☐ businesscategory
☐ carlicense	☐ cn
☐ departmentnumber	☐ description
☐ homedirectory	☐ homephone
☐ homepostaladdress	☐ inetuserhttpurl
☐ inetuserstatus	☒ initials
☐ internationalisdnumber	☐ ipasshpubkey
☐ ipatokenradiusconfiglink	☐ ipatokenradiususername
☐ ipauniqueid	☐ ipauserauthtype
☒ jpegphoto	☐ krbcanonicalname

* Required field

Add Add and Add Another Add and Edit Cancel

4. 사용자가 편집할 속성 옆에 있는 확인란을 선택합니다.
5. 선택 사항: 액세스를 제공할 속성이 나열되지 않은 경우 목록을 추가할 수 있습니다.
 - a. **Add(추가)** 버튼을 클릭합니다.
 - b. 다음 **Add Custom Attribute**(사용자 지정 속성 추가) 창의 **Attribute** (특성) 텍스트 필드에 특성 이름을 입력합니다.
 - c. **OK(확인)** 버튼을 클릭하여 속성을 추가합니다.
 - d. 새 속성이 선택되었는지 확인합니다.
6. 양식 하단에서 **Add(추가)** 버튼을 클릭하여 새 셀프 서비스 규칙을 저장합니다. 또는 **Add and Edit** (추가 및 편집) 버튼을 클릭하여 셀프 서비스 규칙을 저장하고 계속 편집하기

나 추가 및 추가 버튼을 클릭하여 추가 규칙을 저장하고 추가할 수 있습니다.

20.3. IDM 웹 UI를 사용하여 셀프 서비스 규칙 편집

다음 절차에서는 웹 인터페이스(**IdM 웹 UI**)를 사용하여 **IdM**에서 셀프 서비스 액세스 규칙을 편집하는 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 Server** 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 *

Filter Add

☐ audio	☒ businesscategory
☒ carlicense	☒ cn
☐ departmentnumber	☒ description
☐ destinationindicator	☒ displayname
☐ employeenumber	☒ employeetype
☒ facsimiletelephonenumber	☒ gecos
☐ gidnumber	☒ givenname
☐ homedirectory	☒ homephone
☐ homepostaladdress	☒ inetuserhttpurl
☐ inetuserstatus	☒ initials

3.

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

4.

Save(저장) 버튼을 클릭하여 셀프 서비스 규칙에 변경 사항을 저장합니다.

20.4. IDM 웹 UI를 사용하여 셀프 서비스 규칙 삭제

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

사전 요구 사항

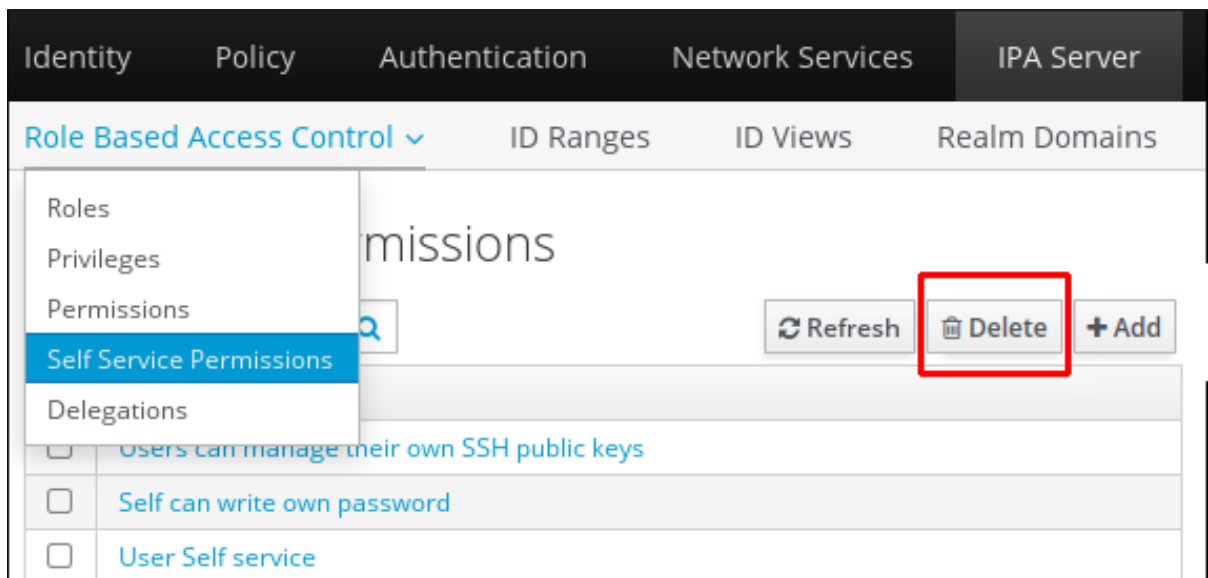
•

IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

절차

1. IPA Server 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 Delete (삭제) 버튼을 클릭합니다.



3. 대화 상자가 열리고 Delete(삭제)를 클릭하여 확인합니다.

21장. ANSIBLE 플레이북을 사용하여 IDM에서 셀프 서비스 규칙 관리

이 섹션에서는 **IdM(Identity Management)**의 셀프 서비스 규칙을 소개하고 **Ansible** 플레이북을 사용하여 셀프 서비스 액세스 규칙을 생성하고 편집하는 방법을 설명합니다. 셀프 서비스 액세스 제어 규칙을 사용하면 **IdM** 엔터티에서 **IdM** 디렉터리 서버 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 확인합니다.](#)
- [Ansible을 사용하여 셀프 서비스 규칙에 특정 특성이 있는지 확인합니다.](#)
- [Ansible을 사용하여 셀프 서비스 규칙에 특정 속성이 없는지 확인합니다.](#)

21.1. IDM의 셀프 서비스 액세스 제어

셀프 서비스 액세스 제어 규칙은 **IdM(Identity Management)** 엔터티가 **IdM Directory Server** 항목에서 수행할 수 있는 작업을 정의합니다. 예를 들어 **IdM** 사용자는 자신의 암호를 업데이트할 수 있습니다.

이 제어 방법을 사용하면 인증된 **IdM** 엔터티에서 **LDAP** 항목 내에서 특정 속성을 편집할 수 있지만 전체 항목에서 작업을 추가하거나 삭제할 수는 없습니다.



주의

셀프 서비스 액세스 제어 규칙으로 작업할 때 주의하십시오. 액세스 제어 규칙을 부적절하게 구성하면 엔터티의 권한을 의도치 않게 높일 수 있습니다.

21.2. 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 자신의 이름 세부 정보 규칙을 관리할 수 있으며 사용자에게 자신의 지정된 이름,디스플레이 이름, 제목 및 초기 속성을 변경할 수 있습니다. 이를 통해 원하는 경우 표시 이름 또는 초기값을 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/` 디렉토리에 이러한 옵션을 구성하는 **IdM** 서버의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉토리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/selfservice/` 디렉토리에 있는 **selfservice-present.yml** 파일의 복사본을 만듭니다.

```
$ 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 목록(읽기 및 쓰기)으로 설정합니다.
- **attribute** 변수를 **givenname, displayname, title, initials** 등 사용자가 자체적으로 관리할 수 있는 속성 목록으로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

추가 리소스

- 셀프 서비스 규칙 개념에 대한 자세한 내용은 **IdM의 셀프 서비스 액세스 제어**를 참조하십시오.
- **ipaselfservice** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- selfservice.md** 파일. 이 파일에는 **ipaselfservice** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/selfservice` 디렉토리.

21.3. ANSIBLE을 사용하여 셀프 서비스 규칙이 없는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 구성에 지정된 셀프 서비스 규칙이 없는지 확인하는 방법을 설명합니다. 아래 예제에서는 **IdM**에 사용자가 자체 이름 세부 정보 셀프 서비스 규칙이 없는지 확인하는 방법을 설명합니다. 이렇게 하면 사용자가 예를 들어 자체 표시 이름 또는 초기값을 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/` 디렉터리에 이러한 옵션을 구성하는 **IdM** 서버의 정규화된 도메인 이름

(FQDN)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 복사본을 만듭니다.

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

3. `selfservice-absent-copy.yml` Ansible 플레이북 파일을 편집하여 편집합니다.

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.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

추가 리소스

- 셀프 서비스 규칙 개념에 대한 자세한 내용은 [IdM의 셀프 서비스 액세스 제어](#)를 참조하십시오.
- **ipaselfservice** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- selfservice.md** 파일. 이 파일에는 **ipaselfservice** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/selfservice` 디렉토리.

21.4. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 특성이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 기존 셀프 서비스 규칙에 특정 설정이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 사용자가 자신의 이름 세부 정보 셀프 서비스 규칙에 성 멤버 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/` 디렉터리에 이러한 옵션을 구성하는 **IdM** 서버의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- 사용자는 **IdM**에 있는 고유한 이름 세부 정보 셀프 서비스 규칙을 관리할 수 있습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/selfservice/` 디렉터리에 있는 `selfservice-member-present.yml` 파일의 복사본을 만듭니다.

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

3. `selfservice-member-present-copy.yml` **Ansible** 플레이북 파일을 편집하여 편집합니다.
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 성으로 설정합니다.
- `action` 변수를 `member` 로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

추가 리소스

- 셀프 서비스 규칙 개념에 대한 자세한 내용은 [IdM의 셀프 서비스 액세스 제어를 참조하십시오](#).
- **ipaselfservice** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- selfservice.md** 파일. 이 파일에는 **ipaselfservice** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/selfservice` 디렉토리.

21.5. 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 예에서 사용자는 이름 세부 정보 셀프 서비스 규칙에 지정된 이름과 성 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.
 - `~/MyPlaybooks/` 디렉터리에 이러한 옵션을 구성하는 IdM 서버의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- 사용자는 IdM에 있는 고유한 이름 세부 정보 셀프 서비스 규칙을 관리할 수 있습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

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

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

3. **selfservice-member-absent-copy.yml** Ansible 플레이북 파일을 편집하여 편집합니다.
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 주어진 이름 및 성으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.
- **state** 변수를 **absent** 로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

추가 리소스

- 셀프 서비스 규칙 개념에 대한 자세한 내용은 [IdM의 셀프 서비스 액세스 제어](#)를 참조하십시오.
- **ipaselfservice** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 -

/usr/share/doc/ansible-freeipa/ 디렉터리에 있는 **README-selfservice.md** 파일. 이 파일에는 **ipaselfservice** 변수 정의도 포함되어 있습니다.

-

/usr/share/doc/ansible-freeipa/playbooks/selfservice 디렉토리.

22장. IDM CLI를 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한을 위임

위임은 **IdM**의 액세스 제어 방법 중 하나로 셀프 서비스 규칙 및 역할 기반 액세스 제어(**RBAC**)와 함께 사용됩니다. 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](#)

22.1. 위임 규칙

위임 규칙을 만들어 사용자 그룹에 권한을 위임하여 사용자를 관리할 수 있습니다.

위임 규칙을 사용하면 특정 사용자 그룹이 다른 사용자 그룹의 특정 속성에 대한 쓰기(편집) 작업을 수행할 수 있습니다. 이 형태의 액세스 제어 규칙은 위임 규칙에서 지정한 속성의 값 편집으로 제한됩니다. 전체 항목을 추가 또는 제거하는 기능을 부여하지 않습니다. 지정되지 않은 속성에 대해 전체 항목을 추가하거나 제어하는 기능을 부여하지 않습니다.

위임 규칙은 **IdM**의 기존 사용자 그룹에 권한을 부여합니다. 예를 들어 위임을 사용하여 **managers** 사용자 그룹이 **employees** 사용자 그룹에서 선택한 사용자 속성을 관리할 수 있습니다.

22.2. IDM CLI를 사용하여 위임 규칙 생성

이 섹션에서는 **IdM CLI**를 사용하여 위임 규칙을 생성하는 방법에 대해 설명합니다.

사전 요구 사항

- **admins** 그룹의 멤버로 로그인했습니다.

절차

- **ipa delegation-add** 명령을 입력합니다. 다음 옵션을 지정합니다.
 - **--group**: 사용자 그룹의 사용자 항목에 권한을 부여하는 그룹입니다.
 - **--memberof**: 위임 그룹의 멤버가 항목을 편집할 수 있는 그룹입니다.
 - **--permissions**: 사용자가 지정된 속성을 보고(읽기) 지정된 속성을 보고 주어진 속성을 추가하거나 변경할 수 있는지(쓰기). 권한을 지정하지 않으면 쓰기 권한만 추가됩니다.
 - **--attrs**: 멤버 그룹의 사용자가 보거나 편집할 수 있는 속성입니다.

예를 들면 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
```

22.3. IDM CLI를 사용하여 기존 위임 규칙 보기

이 섹션에서는 **IdM CLI**를 사용하여 기존 위임 규칙을 보는 방법에 대해 설명합니다.

사전 요구 사항

- **admins** 그룹의 멤버로 로그인했습니다.

절차

- **ipa delegation-find** 명령을 입력합니다.

```
$ ipa delegation-find
```

```
-----
```

```
1 delegation matched
```

```
-----
```

```
Delegation name: basic manager attributes
```

```
Permissions: read, write
```

```
Attributes: businesscategory, departmentnumber, employeeenumber, employeetype
```

```
Member user group: employees
```

```
User group: managers
```

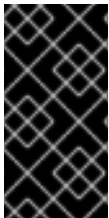
```
-----
```

```
Number of entries returned 1
```

```
-----
```

22.4. 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 로그인했습니다.

절차

- 원하는 변경 사항을 사용하여 **ipa delegation-mod** 명령을 입력합니다. 예를 들어 기본 관리자 속성 예제 규칙에 **displayname** 속성을 추가하려면 다음을 수행합니다.

```
$ ipa delegation-mod "basic manager attributes" --attrs=businesscategory --
attrs=departmentnumber --attrs=employeetype --attrs=employeeenumber --
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

```

22.5. IDM CLI를 사용하여 위임 규칙 삭제

이 섹션에서는 **IdM CLI**를 사용하여 기존 위임 규칙을 삭제하는 방법을 설명합니다.

사전 요구 사항

- **admins** 그룹의 멤버로 로그인했습니다.

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"
-----

```

23장. IDM WEBUI를 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한을 위임

위임은 **IdM**의 액세스 제어 방법 중 하나로 셀프 서비스 규칙 및 역할 기반 액세스 제어(**RBAC**)와 함께 사용됩니다. 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](#)

23.1. 위임 규칙

위임 규칙을 만들어 사용자 그룹에 권한을 위임하여 사용자를 관리할 수 있습니다.

위임 규칙을 사용하면 특정 사용자 그룹이 다른 사용자 그룹의 특정 속성에 대한 쓰기(편집) 작업을 수행할 수 있습니다. 이 형태의 액세스 제어 규칙은 위임 규칙에서 지정한 속성의 값 편집으로 제한됩니다. 전체 항목을 추가 또는 제거하는 기능을 부여하지 않습니다. 지정되지 않은 속성에 대해 전체 항목을 추가하거나 제어하는 기능을 부여하지 않습니다.

위임 규칙은 **IdM**의 기존 사용자 그룹에 권한을 부여합니다. 예를 들어 위임을 사용하여 **managers** 사용자 그룹이 **employees** 사용자 그룹에서 선택한 사용자 속성을 관리할 수 있습니다.

23.2. IDM WEBUI를 사용하여 위임 규칙 생성

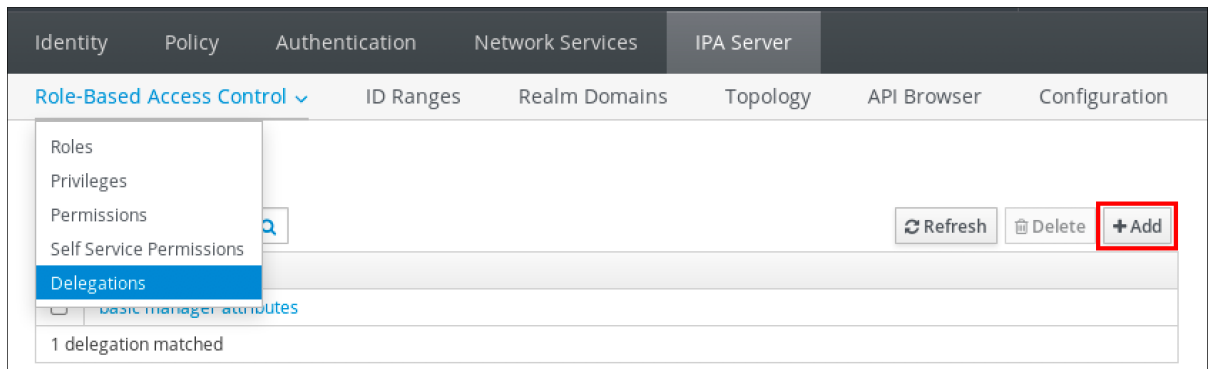
이 섹션에서는 **IdM WebUI**를 사용하여 위임 규칙을 생성하는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM 웹 UI에 admins 그룹의 멤버로 로그인했습니다.**

절차

1. **IPA 서버 메뉴에서 역할 기반 액세스 제어 → 위임을 클릭합니다.**
2. **추가를 클릭합니다.**



3. **Add delegation(디렉토리 추가) 창에서 다음을 수행합니다.**
 - a. **새 위임 규칙의 이름을 지정합니다.**
 - b. **사용자가 지정된 속성(위키)을 보고 지정된 속성을 추가하거나 변경할 수 있는지 여부를 나타내는 확인란을 선택하여 권한을 설정합니다.**
 - c. **User group(사용자 그룹) 드롭다운 메뉴에서 권한을 부여 받고 있는 그룹을 선택하여 멤버 그룹에서 사용자 항목을 보거나 편집할 수 있습니다.**
 - d. **Member 사용자 그룹 드롭다운 메뉴에서 위임 그룹의 구성원이 편집할 수 있는 그룹을 선택합니다.**
 - e. **attributes(속성) 상자에서 권한을 부여할 속성별로 확인란을 선택합니다.**

Add delegation

Delegation name *

Permissions
☒ read
☒ write

User group *

Member user group *

Attributes *

Filter

☐ audio	☒ businesscategory
☐ carlicense	☐ cn
☒ departmentnumber	☐ description
☐ destinationindicator	☐ displayname
☒ employeeenumber	☒ employeetype
☐ facsimiletelephonenumber	☐ gecos
☐ gidnumber	☐ givenname
☐ homedirectory	☐ homephone
☐ homepostaladdress	☐ inetuserhttpurl
☐ inetuserstatus	☐ initials
☐ internationalisdnumber	☐ ipacertmapdata
☐ ipakrbauthzdata	☐ ipanthash
☐ ipanthomedirectory	☐ ipanthomedirectorydrive
☐ ipantlogonscript	☐ ipantprofilepath
☐ ipantsecurityidentifier	☐ ipasshpubkey
☐ ipatokenradiusconfiglink	☐ ipatokenradiususername
☐ ipauniqueid	☐ ipauserauthtype
☐ jpegphoto	☐ krballowedtodelegateto
☐ krbcanonicalname	☐ krbextradata

* Required field

f.

Add(추가) 버튼을 클릭하여 새 위임 규칙을 저장합니다.

23.3. IDM WEBUI를 사용하여 기존 위임 규칙 보기

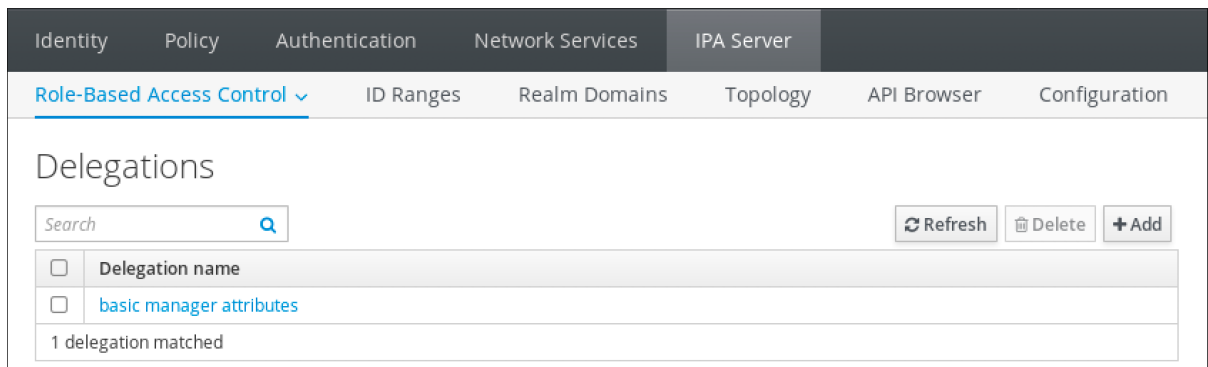
이 섹션에서는 **IdM WebUI**를 사용하여 기존 위임 규칙을 보는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM 웹 UI에 admins 그룹의 멤버로 로그인했습니다.**

절차

- **IPA 서버 메뉴에서 역할 기반 액세스 제어 → 위임을 클릭합니다.**



23.4. IDM WEBUI를 사용하여 위임 규칙 수정

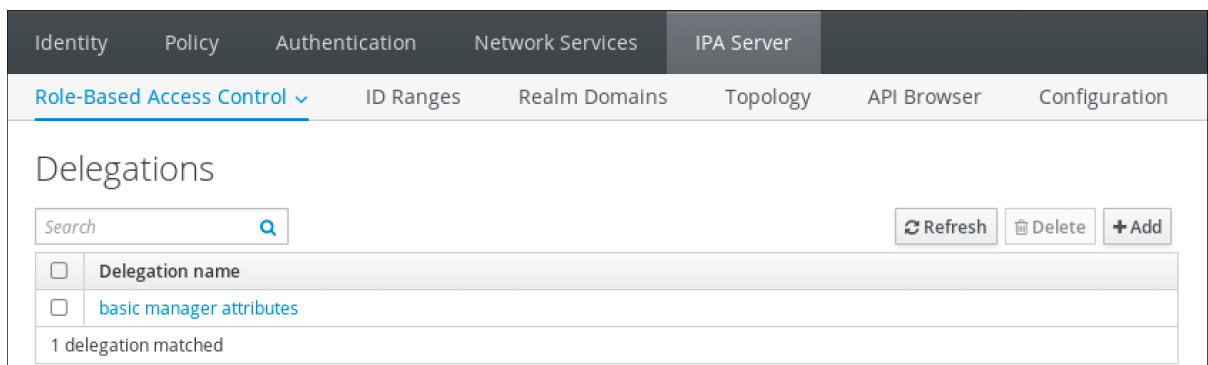
이 섹션에서는 **IdM WebUI**를 사용하여 기존 위임 규칙을 수정하는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM 웹 UI에 admins 그룹의 멤버로 로그인했습니다.**

절차

1. **IPA 서버 메뉴에서 역할 기반 액세스 제어 → 위임을 클릭합니다.**



2.

수정할 규칙을 클릭합니다.

3.

원하는 대로 변경합니다.

•

규칙의 이름을 변경합니다.

•

사용자가 지정된 속성(읽기)을 보고 지정된 속성을 추가하거나 변경할 수 있는지 여부를 나타내는 확인란을 선택하여 부여된 권한을 변경합니다.

•

User group(사용자 그룹) 드롭다운 메뉴에서 권한을 부여 받고 있는 그룹을 선택하여 멤버 그룹에서 사용자 항목을 보거나 편집할 수 있습니다.

•

Member 사용자 그룹 드롭다운 메뉴에서 위임 그룹의 구성원이 편집할 수 있는 그룹을 선택합니다.

•

attributes(속성) 상자에서 권한을 부여할 속성별로 확인란을 선택합니다. 속성에 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 *

☐ audio	☒ businesscategory	☐ carlicense
☐ cn	☒ departmentnumber	☐ description
☐ destinationindicator	☒ displayname	☒ employeenumber
☒ employeetype	☐ facsimiletelephonenumber	☐ gecoss
☐ gidnumber	☐ givenname	☒ homedirectory
☐ homephone	☐ homepostaladdress	☐ inetuserhttpurl
☐ inetuserstatus	☐ initials	☐ internationalisdnnumber
☐ ipacertmapdata	☐ ipakrbauthzdata	☐ ipanhash
☐ ipanthomedirectory	☐ ipanthomedirectorydrive	☐ ipantlogonscript
☐ ipantprofilepath	☐ ipantsecurityidentifier	☐ ipasshpubkey

- **Save(저장)** 버튼을 클릭하여 변경 사항을 저장합니다.

23.5. IDM WEBUI를 사용하여 위임 규칙 삭제

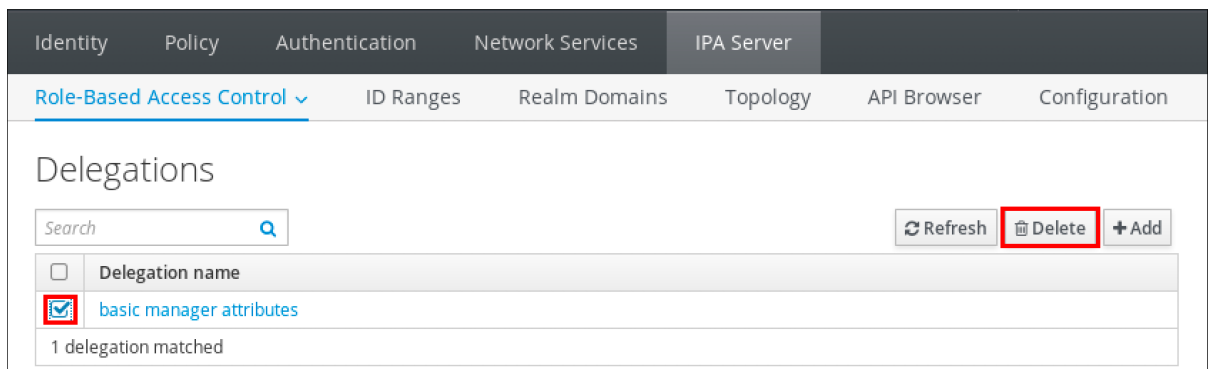
이 섹션에서는 **IdM WebUI**를 사용하여 기존 위임 규칙을 삭제하는 방법을 설명합니다.

사전 요구 사항

- **IdM 웹 UI에 admins 그룹의 멤버로 로그인했습니다.**

절차

1. **IPA 서버 메뉴에서 역할 기반 액세스 제어 → 위임을 클릭합니다.**
2. **제거할 규칙 옆에 있는 확인란을 선택합니다.**
3. **삭제를 클릭합니다.**



4. **Delete(삭제) 를 클릭하여 확인합니다.**

24장. ANSIBLE 플레이북을 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한을 위임

위임은 IdM의 액세스 제어 방법 중 하나로 셀프 서비스 규칙 및 역할 기반 액세스 제어(RBAC)와 함께 사용됩니다. 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.](#)
- [위임 규칙에 특정 속성이 있는지 확인하려면 Ansible을 사용합니다.](#)
- [위임 규칙에 특정 속성이 없는지 확인하려면 Ansible을 사용합니다.](#)

24.1. 위임 규칙

위임 규칙을 만들어 사용자 그룹에 권한을 위임하여 사용자를 관리할 수 있습니다.

위임 규칙을 사용하면 특정 사용자 그룹이 다른 사용자 그룹의 특정 속성에 대한 쓰기(편집) 작업을 수행할 수 있습니다. 이 형태의 액세스 제어 규칙은 위임 규칙에서 지정한 속성의 값 편집으로 제한됩니다. 전체 항목을 추가 또는 제거하는 기능을 부여하지 않습니다. 지정되지 않은 속성에 대해 전체 항목을 추가하거나 제어하는 기능을 부여하지 않습니다.

위임 규칙은 IdM의 기존 사용자 그룹에 권한을 부여합니다. 예를 들어 위임을 사용하여 **managers** 사용자 그룹이 **employees** 사용자 그룹에서 선택한 사용자 속성을 관리할 수 있습니다.

24.2. IDm용 ANSIBLE 인벤토리 파일 생성

Ansible을 사용하는 경우 홈 디렉터리에서 `/usr/share/doc/ansible-freeipa/*` 및 `/usr/share/doc-roles/*` 하위 디렉터리를 복사하고 적용하는 **Ansible** 플레이북 전용 하위 디렉터리를 생성하는 것이 좋습니다. 이 연습에는 다음과 같은 이점이 있습니다.

- 모든 플레이북을 한 곳에서 찾을 수 있습니다.
- 루트 권한을 호출하지 않고 플레이북을 실행할 수 있습니다.

절차

1. 홈 디렉터리에서 **Ansible** 구성 및 플레이북의 디렉터리를 생성합니다.

```
$ mkdir ~/MyPlaybooks/
```

2. `~/MyPlaybooks/` 디렉터리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

3. 다음 콘텐츠를 사용하여 `~/Myplaybooks/ansible.cfg` 파일을 만듭니다.

```
[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 이러한 위치에 있는 호스트에 대한 두 개의 호스트 그룹인 **eu** 와 **us** 를 정의합니다.

또한 이 구성은 **eu** 및 **us** 그룹의 모든 호스트를 포함하는 **ipaserver** 호스트 그룹을 정의합니다.

24.3. ANSIBLE을 사용하여 위임 규칙이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 새 **IdM** 위임 규칙에 대한 권한을 정의하고 해당 규칙이 있는지 확인하는 방법을 설명합니다. 이 예제에서 새 기본 관리자 속성 위임 규칙은 **managers** 그룹에 **employees** 그룹 구성원에 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 정규화된 도메인 이름(FQDN)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/` 디렉토리에 있는 `delegation-present.yml` 파일의 복사본을 만듭니다.

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

3. 편집 할 `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 목록(읽기 및 쓰기)으로 설정합니다.
- 속성 변수를 위임된 사용자 그룹이 관리할 수 있는 속성 목록(`business category, departmentnumber, employeenumber, employeetype`) 으로 설정합니다.
- 그룹 변수를 특정 보기 또는 수정에 대한 액세스 권한이 부여되는 그룹의 이름으로 설정합니다.
- `membergroup` 변수를 속성을 보거나 수정할 수 있는 그룹의 이름으로 설정합니다.

이는 현재 예제에 대한 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:
    ipadmin_password: Secret123
    name: "basic manager attributes"
    permission: read, write
    attribute:
      - businesscategory
      - departmentnumber
      - employeenumber
      - employeeetype
    group: managers
    membergroup: employees

```

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

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-present-copy.yml
```

추가 리소스

- 위임 규칙 개념에 대한 자세한 내용은 [위임 규칙](#)을 참조하십시오.
- **ipadelegation** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- delegation.md** 파일. 이 파일에는 **ipadelegation** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉토리.

24.4. ANSIBLE을 사용하여 위임 규칙이 없는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 구성에 지정된 위임 규칙이 없는지 확인하는 방법을 설명합니다. 아래 예제에서는 사용자 지정 기본 관리자 속성 위임 규칙이 **IdM**에 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 정규화된 도메인 이름(FQDN)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-absent.yml** 파일의 복사본을 만듭니다.

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

3. 편집 할 **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.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Delegation absent
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure delegation "basic manager attributes" is absent
      ipadelegation:
        ipadmin_password: Secret123
        name: "basic manager attributes"
        state: absent
```

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

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-absent-copy.yml
```

추가 리소스

- 위임 규칙 개념에 대한 자세한 내용은 [위임 규칙](#)을 참조하십시오.
- **ipadelegation** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- delegation.md** 파일. 이 파일에는 **ipadelegation** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉토리.

24.5. 위임 규칙에 특정 속성이 있는지 확인하려면 **ANSIBLE**을 사용합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 위임 규칙에 특정 설정이 있는지 확인하는 방법을 설명합니다. 이 플레이북을 사용하여 이전에 만든 위임 역할을 수정할 수 있습니다. 이 예제에서는 기본 관리

자 속성 위임 규칙에 **departmentnumber** 멤버 속성만 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.
 - 이러한 옵션을 구성하는 **IdM** 서버의 정규화된 도메인 이름(**FQDN**)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
 - **Ansible** 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.
- 기본 관리자 속성 위임 규칙은 **IdM**에 있습니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/delegation/** 디렉터리에 있는 **delegation-member-present.yml** 파일의 복사본을 만듭니다.

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

3. 편집 할 **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 위임 규칙의 이름으로 설정하여 수정합니다.
- 특성 변수를 **departmentnumber** 로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-member-present-copy.yml
```

추가 리소스

- 위임 규칙 개념에 대한 자세한 내용은 [위임 규칙](#)을 참조하십시오.

- **ipadelegation** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- delegation.md** 파일. 이 파일에는 **ipadelegation** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉토리.

24.6. 위임 규칙에 특정 속성이 없는지 확인하려면 **ANSIBLE**을 사용합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 위임 규칙에 특정 설정이 없는지 확인하는 방법을 설명합니다. 이 플레이북을 사용하여 위임 역할이 바람직하지 않은 액세스 권한을 부여하지 않도록 할 수 있습니다. 이 예제에서는 기본 관리자 속성 위임 규칙에 **employeenumber** 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 정규화된 도메인 이름(FQDN)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
 - **Ansible** 인벤토리 파일은 `~/MyPlaybooks/` 디렉터리에 있습니다.
- 기본 관리자 속성 위임 규칙은 **IdM**에 있습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/delegation/` 디렉토리에 있는 `delegation-member-absent.yml` 파일의 복사본을 만듭니다.

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

3. 편집 할 `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 위임 규칙의 이름으로 설정하여 수정합니다.
- 특성 변수를 `employeenumber` 및 `employee type` 으로 설정합니다.
- `action` 변수를 `member` 로 설정합니다.
- `state` 변수를 `absent` 로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Delegation member absent
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure delegation "basic manager attributes" member attributes
      employeenumber and employeetype are absent
      ipadelegation:
        ipaadmin_password: Secret123
```

```

name: "basic manager attributes"
attribute:
- employeenumber
- employeetype
action: member
state: absent

```

5.

파일을 저장합니다.

6.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-member-absent-copy.yml
```

추가 리소스

- 위임 규칙 개념에 대한 자세한 내용은 [위임 규칙](#)을 참조하십시오.
- **ipadelegation** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- delegation.md** 파일. 이 파일에는 **ipadelegation** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉토리.

25장. CLI를 사용하여 IDM에서 역할 기반 액세스 제어 관리

이 장에서는 **IdM(Identity Management)**의 역할 기반 액세스 제어를 소개하고 명령줄 인터페이스(**CLI**)에서 다음 작업을 설명합니다.

- 권한 관리
- 권한 관리
- 역할 관리

25.1. IDM의 역할 기반 액세스 제어

IdM의 역할 기반 액세스 제어(**RBAC**)는 셸프 서비스 및 위임 액세스 제어와 비교하여 사용자에게 매우 다른 종류의 권한을 부여합니다.

역할 기반 액세스 제어는 다음 세 부분으로 구성됩니다.

- 권한은 사용자 추가 또는 삭제, 그룹 수정, 읽기 액세스 활성화 등의 특정 작업을 수행할 수 있는 권한을 부여합니다.
- 권한은 권한을 결합합니다(예: 새 사용자를 추가하는 데 필요한 모든 권한).
- 역할은 사용자, 사용자 그룹, 호스트 또는 호스트 그룹에 일련의 권한을 부여합니다.

25.1.1. IdM의 권한

권한은 역할 기반 액세스 제어의 최하위 단위이며, 해당 작업이 적용되는 **LDAP** 항목과 함께 작업을 정의합니다. 구성 요소와 비교했을 때 필요한 수의 권한에 권한을 할당할 수 있습니다. 하나 이상의 권한은 다음을 허용하는 작업을 정의합니다.

- write

- **read**
- **search**
- **비교**
- **add**
- **delete**
- **all**

이러한 작업은 세 개의 기본 대상에 적용됩니다 :

- **subtree:** 도메인 이름(DN), 이 DN 아래의 하위 트리
- **대상 필터: LDAP 필터**
- **대상:** 항목을 지정하는 데 사용할 수 있는 와일드카드가 있는 DN

또한 다음과 같은 편의성 옵션은 해당 특성을 설정합니다.

- **type:** 유형의 개체 (사용자, 그룹 등)는 **subtree** 및 **target filter**를 설정합니다.
- **memberOf:** 그룹의 구성원; 대상 필터를 설정합니다.
- **targetGroup:** 특정 그룹을 수정할 액세스 권한을 부여합니다(예: 그룹 멤버십을 관리할 수 있는 권한 부여). 대상을 설정합니다.

IdM 권한을 사용하면 어떤 사용자가 어떤 오브젝트에 대한 액세스 권한이 있는지 그리고 이러한 오브젝트의 속성까지 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, 모든 인증된 사용자 또는 특정 권한 있는 사용자 그룹에 변경할 수 있습니다.

예를 들어 이 접근 방식의 권한의 유연성은 사용자 또는 그룹에 대한 액세스만 제한하려는 관리자에게 유용합니다. 사용자 또는 그룹이 액세스해야 하는 특정 섹션으로만 액세스하고 다른 섹션에 완전히 숨길 수 있도록 합니다.



참고

권한에는 다른 권한이 포함될 수 없습니다.

25.1.2. 기본 관리 권한

관리 권한은 기본적으로 **IdM**과 함께 제공되는 권한입니다. 다음과 같은 차이점을 사용하여 사용자가 생성한 다른 권한처럼 작동합니다.

- 해당 파일을 삭제하거나 이름, 위치, 대상 특성을 수정할 수 없습니다.
- 여기에는 세 가지 속성 세트가 있습니다.
 - 기본 속성은 **IdM**에서 관리하므로 사용자가 수정할 수 없습니다.
 - 사용자가 추가한 속성인 포함된 속성
 - 사용자가 제거하는 속성인 제외된 속성

관리되는 권한은 기본 및 포함된 특성 세트에 표시되지만 제외된 속성 세트에는 표시되지 않는 모든 속성에 적용됩니다.



참고

관리 권한을 삭제할 수는 없지만 바인딩 유형을 권한으로 설정하고 모든 권한에서 관리 권한을 제거하면 효과적으로 비활성화됩니다.

관리되는 모든 권한의 이름은 시스템:(예: 시스템): **Sudo rule** 또는 시스템을 추가합니다: **Services** 를 수정합니다. 이전 버전의 **IdM**에서는 기본 권한에 다른 체계를 사용했습니다. 예를 들어 사용자는 삭제할 수 없으며 권한에만 할당할 수 있었습니다. 이러한 기본 권한은 대부분 관리 권한으로 설정되었지만 다음 권한은 여전히 이전 체계를 사용합니다.

- **Automember Rebuild Membership** 작업 추가
- 구성 하위 항목 추가
- 복제 계약 추가
- 인증서 제거 보류
- **CA**에서 인증서 상태 가져오기
- **DNA** 범위 읽기
- **DNA** 범위 수정
- **PassSync** 관리자 구성 읽기
- **PassSync** 관리자 구성 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 특성을 변경할 수 없으므로 명령이 실패합니다. 웹 UI에서 관리 권한을 수정하려고 하면 수정할 수 없는 속성이 비활성화됩니다.

25.1.3. IdM의 권한

권한은 역할에 적용할 수 있는 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 사용자의 계정을 설정하려면 다음과 같은 권한이 필요합니다.

- 새 사용자 항목 생성
- 사용자 암호 재설정
- 새 사용자를 기본 IPA 사용자 그룹에 추가

이러한 세 가지 하위 수준 작업을 이름이 지정된 사용자 지정 권한의 형태로 더 높은 수준의 작업(예: 사용자 추가)에 결합하면 시스템 관리자가 역할을 쉽게 관리할 수 있습니다. IdM에는 이미 몇 가지 기본 권한이 포함되어 있습니다. 사용자 및 사용자 그룹 외에도 호스트 및 호스트 그룹은 네트워크 서비스에도 권한이 할당됩니다. 이 방법을 사용하면 특정 네트워크 서비스를 사용하는 호스트 집합의 사용자 집합에 의한 작업을 세부적으로 제어할 수 있습니다.



참고

권한에는 다른 권한이 없을 수 있습니다.

25.1.4. IdM의 역할

역할은 역할에 지정된 사용자가 보유하는 권한 목록입니다. 실제로 권한은 지정된 하위 수준 작업을 수행할 수 있는 기능을 부여합니다(예: 사용자 항목 생성, 그룹에 항목 추가). 권한은 더 높은 수준의 작업(예: 지정된 그룹에 새 사용자 만들기)에 필요한 하나 이상의 권한을 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 톨로 사용되지 않습니다.



참고

역할에 다른 역할은 포함할 수 없습니다.

25.1.5. ID 관리에 사전 정의된 역할

Red Hat Identity Management는 다음과 같은 일련의 사전 정의 역할을 제공합니다.

표 25.1. ID 관리에서 사전 정의된 역할

Role	권한	설명
헬프데스크	사용자 및 암호 재설정, 그룹 멤버십 수정	간단한 사용자 관리 작업 수행 담당
IT 보안 전문가	Netgroups 관리자, HBAC 관리자, Sudo 관리자	호스트 기반 액세스 제어와 같은 보안 정책 관리, sudo 규칙

Role	권한	설명
IT 전문가	호스트 관리자, 호스트 그룹 관리자, 서비스 관리자, 자동 마운트 관리자	호스트 관리 담당
보안 설계자	위임 관리자, 복제 관리자, 쓰기 IPA 구성, 암호 정책 관리자	Identity Management 환경 관리, 신뢰 생성, 복제 계약 생성
사용자 관리자	사용자 관리자, 그룹 관리자, 단계 사용자 관리자	사용자 및 그룹 생성 담당

25.2. 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 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

절차

1. **ipa permission-add** 명령을 사용하여 새 권한 항목을 생성합니다.
예를 들어 **dns admin** 이라는 권한을 추가하려면 :

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

2. 다음 옵션을 사용하여 권한의 속성을 지정합니다.

- **--bindtype** 은 바인드 규칙 유형을 지정합니다. 이 옵션은 모든, 익명 및 권한 인수를 허용합니다. 권한 바인드 유형은 역할을 통해 이 권한을 부여 받은 사용자만 수행할 수 있음을 나타냅니다.
예를 들면 다음과 같습니다.

```
$ ipa permission-add "dns admin" --bindtype=all
```

--bindtype 을 지정하지 않으면 **permission** 이 기본값입니다.



참고

기본이 아닌 바인드 규칙 유형의 권한을 권한에 추가할 수 없습니다. 또한 권한이 이미 권한에 있는 권한을 기본이 아닌 바인드 규칙 유형으로 설정할 수 없습니다.

•

--right 는 권한에 의해 부여된 권한을 나열하고 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 특성을 설정할 수 있습니다. 예를 들면 다음과 같습니다.

```
$ ipa permission-add "dns admin" --right=read --right=write
```

```
$ ipa permission-add "dns admin" --right={read,write}
```



참고

추가 및 삭제 는 읽기,검색,비교, 쓰기 등의 항목 수준의 작업(예: 사용자 삭제, 그룹 추가 등)입니다. **userCertificate**에 쓸 수는 있지만 **user Password** 는 읽을 수 없습니다.

•

--attrs 는 권한이 부여되는 속성 목록을 제공합니다.

여러 **--attrs** 옵션을 사용하거나 중괄호 안의 쉼표로 구분된 목록에 옵션을 나열하여 여러 특성을 설정할 수 있습니다. 예를 들면 다음과 같습니다.

```
$ ipa permission-add "dns admin" --attrs=description --attrs=automountKey
```

```
$ ipa permission-add "dns admin" --attrs={description,automountKey}
```

--attrs 로 제공되는 속성은 존재해야 하며 지정된 오브젝트 유형에 대해 허용된 속성이 있어야 합니다. 그렇지 않으면 스키마 구문 오류로 인해 명령이 실패합니다.

•

--type 은 권한이 적용되는 항목 오브젝트 유형(예: 사용자, 호스트 또는 서비스)을 정의합니다. 각 유형에는 허용된 속성 집합이 있습니다.

예를 들면 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 간소화된 플랫 디렉터리 트리 구조를 사용하기 때문에 **--subtree** 를 사용하여 다른 구성의 컨테이너 또는 상위 항목인 자동 마운트 위치와 같은 일부 항목을 대상으로 지정할 수 있습니다. 예를 들면 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 표시되어 관리자가 쉽게 사용할 수 있도록 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 대상 필터를 지정된 그룹의 멤버로 설정합니다. 예를 들어 이 권한이 있는 사용자가 **engineer** 그룹의 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 그룹으로 **target**을 설정합니다. 예를 들어 권한이 있는 사용자가 **engineers** 그룹에 **member** 속성을 쓰도록 하려면 (구성원 추가 또는 제거 가능).

```
$ ipa permission-add ManageMembers --right="write" --
subtree=cn=groups,cn=accounts,dc=example,dc=test --attr=member --
targetgroup=engineers
```

선택적으로 **DN**(대상 도메인 이름)을 지정할 수 있습니다.

- **--target** 은 권한을 적용할 **DN**을 지정합니다. 와일드카드가 허용됩니다.
- **--targetto** 는 항목을 이동할 수 있는 **DN** 하위 트리를 지정합니다.
- **--targetfrom** 은 항목을 이동할 수 있는 **DN** 하위 트리를 지정합니다.

25.3. 기존 권한의 명령 옵션

필요에 따라 다음 변형을 사용하여 기존 권한을 수정합니다.

- 기존 권한을 편집하려면 **ipa permission-mod** 명령을 사용합니다. 권한을 추가할 때 와 동일한 명령 옵션을 사용할 수 있습니다.
- 기존 권한을 찾으려면 **ipa permission-find** 명령을 사용합니다. 권한을 추가할 때 와 동일한 명령 옵션을 사용할 수 있습니다.
- 특정 권한을 보려면 **ipa permission-show** 명령을 사용합니다.
--raw 인수는 생성된 원시 **389-ds ACI**를 보여줍니다. 예를 들면 다음과 같습니다.

```
$ ipa permission-show <permission> --raw
```

- **ipa permission-del** 명령은 권한을 완전히 삭제합니다.

추가 리소스

ipa 권한 명령에 대한 자세한 내용은 **ipa** 도움말 페이지 및 **ipa help** 명령을 참조하십시오.

25.4. 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 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.
- **기존 권한.** 권한에 대한 자세한 내용은 [CLI에서 IdM 권한 관리](#)를 참조하십시오.

절차

1.

ipa privilege-add 명령

예(를) 사용하여 권한 항목을 추가합니다. 예: 설명으로 *파일 시스템 관리* 라는 권한을 추가하려면 다음을 실행합니다.

```
$ ipa privilege-add "managing filesystems" --desc="for filesystems"
```

2.

예를 들어, **privilege-add-permission** 명령을 사용하여 권한 그룹에 필요한 권한을 할당합니다. 예를 들어 *자동 마운트 관리* 및 *ftp 서비스 관리* 라는 권한을 *파일 시스템 권한 관리* 에 추가하려면 다음을 수행합니다.

```
$ ipa privilege-add-permission "managing filesystems" --permissions="managing automount" --permissions="managing ftp services"
```

25.5. 기존 권한에 대한 명령 옵션

필요에 따라 기존 권한을 수정하려면 다음 변형을 사용합니다.

- 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 권한 명령에 대한 자세한 내용은 **ipa** 도움말 페이지 및 **ipa help** 명령을 참조하십시오.

25.6. 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 [Using kinit to log in IdM에서 수동으로](#) 참조하십시오.
- 기존 권한. 권한에 대한 자세한 내용은 [CLI에서 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 필요한 권한을 역할에 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 생성한 **useradmin** 역할에 추가하려면 다음을 수행합니다.

```
$ ipa role-add-member --groups=useradmins useradmin
Role name: useradmin
Description: User Administrator
Member groups: useradmins
Privileges: user administrators
-----
Number of members added 1
-----
```

25.7. 기존 역할에 대한 명령 옵션

필요에 따라 다음 변형을 사용하여 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 역할 명령에 대한 자세한 내용은 **ipa** 도움말 페이지 및 **ipa help** 명령을 참조하십시오.

26장. IDM 웹 UI를 사용하여 역할 기반 액세스 제어 관리

이 장에서는 **IdM(Identity Management)**의 역할 기반 액세스 제어를 소개하고 웹 인터페이스(웹 UI)에서 다음 작업을 설명합니다.

- 권한 관리
- 권한 관리
- 역할 관리

26.1. IDM의 역할 기반 액세스 제어

IdM의 역할 기반 액세스 제어(**RBAC**)는 셸프 서비스 및 위임 액세스 제어와 비교하여 사용자에게 매우 다른 종류의 권한을 부여합니다.

역할 기반 액세스 제어는 다음 세 부분으로 구성됩니다.

- 권한은 사용자 추가 또는 삭제, 그룹 수정, 읽기 액세스 활성화 등의 특정 작업을 수행할 수 있는 권한을 부여합니다.
- 권한은 권한을 결합합니다(예: 새 사용자를 추가하는 데 필요한 모든 권한).
- 역할은 사용자, 사용자 그룹, 호스트 또는 호스트 그룹에 일련의 권한을 부여합니다.

26.1.1. IdM의 권한

권한은 역할 기반 액세스 제어의 최하위 단위이며, 해당 작업이 적용되는 **LDAP** 항목과 함께 작업을 정의합니다. 구성 요소와 비교했을 때 필요한 수의 권한에 권한을 할당할 수 있습니다. 하나 이상의 권한은 다음을 허용하는 작업을 정의합니다.

- write

- **read**
- **search**
- **비교**
- **add**
- **delete**
- **all**

이러한 작업은 세 개의 기본 대상에 적용됩니다 :

- **subtree:** 도메인 이름(DN), 이 DN 아래의 하위 트리
- **대상 필터:** LDAP 필터
- **대상:** 항목을 지정하는 데 사용할 수 있는 와일드카드가 있는 DN

또한 다음과 같은 편의성 옵션은 해당 특성을 설정합니다.

- **type:** 유형의 개체 (사용자, 그룹 등)는 **subtree** 및 **target filter**를 설정합니다.
- **memberOf:** 그룹의 구성원; 대상 필터를 설정합니다.
- **targetGroup:** 특정 그룹을 수정할 액세스 권한을 부여합니다(예: 그룹 멤버십을 관리할 수 있는 권한 부여). 대상을 설정합니다.

IdM 권한을 사용하면 어떤 사용자가 어떤 오브젝트에 대한 액세스 권한이 있는지 그리고 이러한 오브젝트의 속성까지 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, 모든 인증된 사용자 또는 특정 권한 있는 사용자 그룹에 변경할 수 있습니다.

예를 들어 이 접근 방식의 권한의 유연성은 사용자 또는 그룹에 대한 액세스만 제한하려는 관리자에게 유용합니다. 사용자 또는 그룹이 액세스해야 하는 특정 섹션으로만 액세스하고 다른 섹션에 완전히 숨길 수 있도록 합니다.



참고

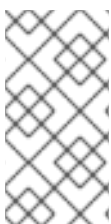
권한에는 다른 권한이 포함될 수 없습니다.

26.1.2. 기본 관리 권한

관리 권한은 기본적으로 **IdM**과 함께 제공되는 권한입니다. 다음과 같은 차이점을 사용하여 사용자가 생성한 다른 권한처럼 작동합니다.

- 해당 파일을 삭제하거나 이름, 위치, 대상 특성을 수정할 수 없습니다.
- 여기에는 세 가지 속성 세트가 있습니다.
 - 기본 속성은 **IdM**에서 관리하므로 사용자가 수정할 수 없습니다.
 - 사용자가 추가한 속성인 포함된 속성
 - 사용자가 제거하는 속성인 제외된 속성

관리되는 권한은 기본 및 포함된 특성 세트에 표시되지만 제외된 속성 세트에는 표시되지 않는 모든 속성에 적용됩니다.



참고

관리 권한을 삭제할 수는 없지만 바인딩 유형을 권한으로 설정하고 모든 권한에서 관리 권한을 제거하면 효과적으로 비활성화됩니다.

관리되는 모든 권한의 이름은 시스템:(예: 시스템): **Sudo rule** 또는 시스템을 추가합니다: **Services** 를 수정합니다. 이전 버전의 **IdM**에서는 기본 권한에 다른 체계를 사용했습니다. 예를 들어 사용자는 삭제할 수 없으며 권한에만 할당할 수 있었습니다. 이러한 기본 권한은 대부분 관리 권한으로 설정되었지만 다음 권한은 여전히 이전 체계를 사용합니다.

- **Automember Rebuild Membership** 작업 추가
- 구성 하위 항목 추가
- 복제 계약 추가
- 인증서 제거 보류
- **CA**에서 인증서 상태 가져오기
- **DNA** 범위 읽기
- **DNA** 범위 수정
- **PassSync** 관리자 구성 읽기
- **PassSync** 관리자 구성 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 특성을 변경할 수 없으므로 명령이 실패합니다. 웹 UI에서 관리 권한을 수정하려고 하면 수정할 수 없는 속성이 비활성화됩니다.

26.1.3. IdM의 권한

권한은 역할에 적용할 수 있는 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** 사용자의 계정을 설정하려면 다음과 같은 권한이 필요합니다.

- 새 사용자 항목 생성
- 사용자 암호 재설정
- 새 사용자를 기본 **IPA** 사용자 그룹에 추가

이러한 세 가지 하위 수준 작업을 이름이 지정된 사용자 지정 권한의 형태로 더 높은 수준의 작업(예: 사용자 추가)에 결합하면 시스템 관리자가 역할을 쉽게 관리할 수 있습니다. **IdM**에는 이미 몇 가지 기본 권한이 포함되어 있습니다. 사용자 및 사용자 그룹 외에도 호스트 및 호스트 그룹은 네트워크 서비스에도 권한이 할당됩니다. 이 방법을 사용하면 특정 네트워크 서비스를 사용하는 호스트 집합의 사용자 집합에 의한 작업을 세부적으로 제어할 수 있습니다.



참고

권한에는 다른 권한이 없을 수 있습니다.

26.1.4. IdM의 역할

역할은 역할에 지정된 사용자가 보유하는 권한 목록입니다. 실제로 권한은 지정된 하위 수준 작업을 수행할 수 있는 기능을 부여합니다(예: 사용자 항목 생성, 그룹에 항목 추가). 권한은 더 높은 수준의 작업(예: 지정된 그룹에 새 사용자 만들기)에 필요한 하나 이상의 권한을 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 톨로 사용되지 않습니다.



참고

역할에 다른 역할은 포함할 수 없습니다.

26.1.5. ID 관리에 사전 정의된 역할

Red Hat Identity Management는 다음과 같은 일련의 사전 정의 역할을 제공합니다.

표 26.1. ID 관리에서 사전 정의된 역할

Role	권한	설명
헬프데스크	사용자 및 암호 재설정, 그룹 멤버십 수정	간단한 사용자 관리 작업 수행 담당
IT 보안 전문가	Netgroups 관리자, HBAC 관리자, Sudo 관리자	호스트 기반 액세스 제어와 같은 보안 정책 관리, sudo 규칙

Role	권한	설명
IT 전문가	호스트 관리자, 호스트 그룹 관리자, 서비스 관리자, 자동 마운트 관리자	호스트 관리 담당
보안 설계자	위임 관리자, 복제 관리자, 쓰기 IPA 구성, 암호 정책 관리자	Identity Management 환경 관리, 신뢰 생성, 복제 계약 생성
사용자 관리자	사용자 관리자, 그룹 관리자, 단계 사용자 관리자	사용자 및 그룹 생성 담당

26.2. IDM 웹 UI에서 권한 관리

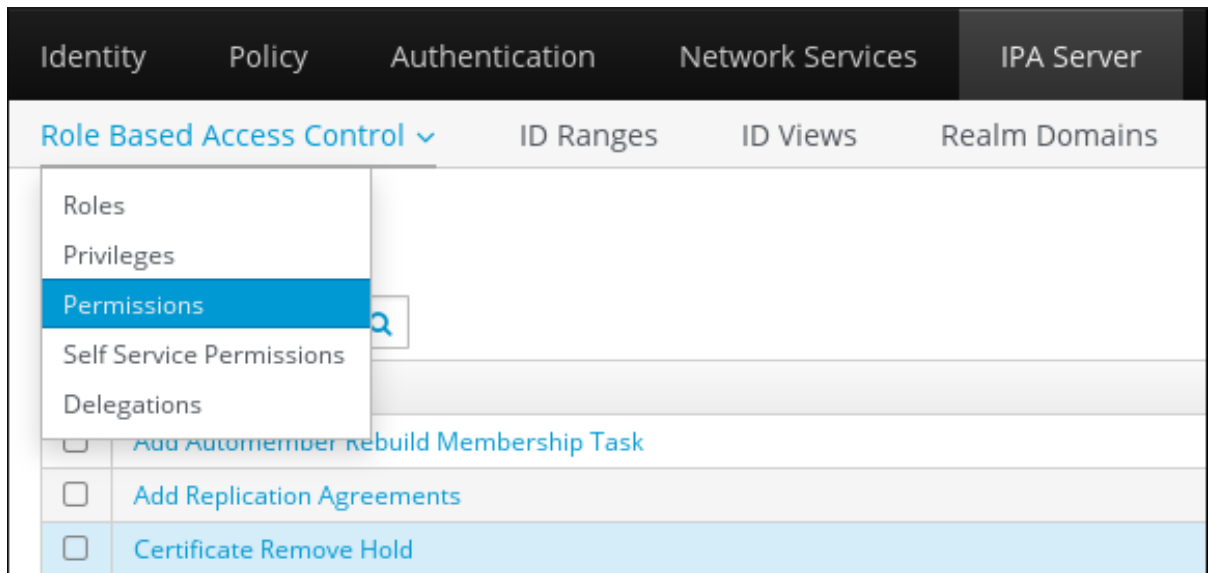
이 섹션에서는 웹 인터페이스(IdM 웹 UI)를 사용하여 IdM(Identity Management)에서 권한을 관리하는 방법을 설명합니다.

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.

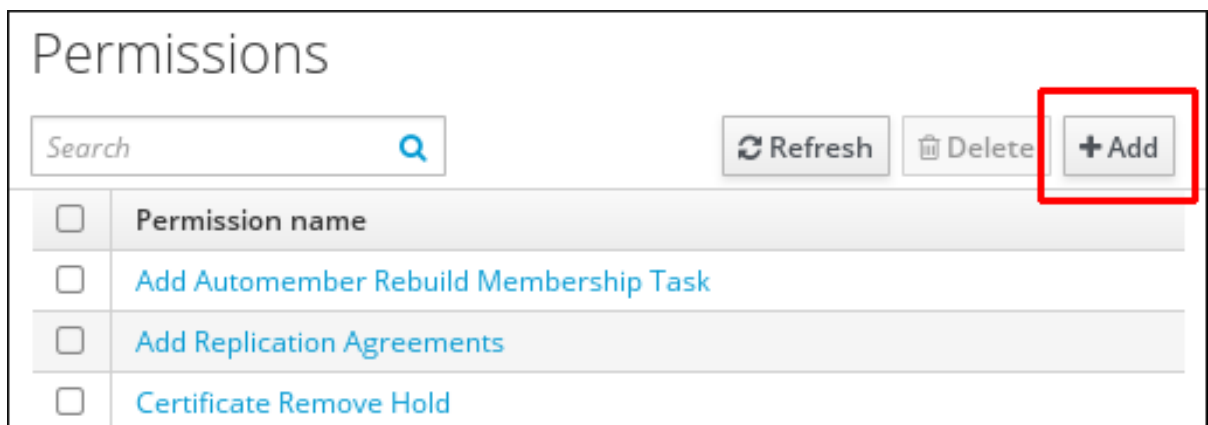
절차

1. 새 권한을 추가하려면 IPA Server 탭에서 **Role-Based Access Control** (역할 기반 액세스 제어) 하위 메뉴를 열고 **Permissions:**을 선택합니다.



2.

권한 목록이 열립니다. 권한 목록 상단에 있는 **Add** 단추를 클릭합니다.



3.

Add Permission(권한 추가) 양식이 열립니다. 새 권한의 이름을 지정하고 그에 따라 속성을 정의합니다.

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

4.

적절한 **Bind** 규칙 유형을 선택합니다.

•

권한은 기본 권한 유형으로, 권한 및 역할을 통해 액세스 권한을 부여합니다.

- **all** 은 권한이 인증된 모든 사용자에게 적용되도록 지정합니다.
- **anonymous** 는 인증되지 않은 사용자를 포함하여 모든 사용자에게 권한이 적용되도록 지정합니다.



참고

기본이 아닌 바인드 규칙 유형의 권한을 권한에 추가할 수 없습니다. 또한 권한이 이미 권한에 있는 권한을 기본이 아닌 바인드 규칙 유형으로 설정할 수 없습니다.

5. 부여된 권리에서 이 권한에 대해 부여할 권한을 선택하십시오 .

6. 권한에 대한 대상 항목을 식별하는 메서드를 정의합니다.

- **type**은 사용자, 호스트 또는 서비스와 같은 항목 유형을 지정합니다. **Type** (유형) 설정에 값을 선택하면 해당 항목 유형에 대해 이 **ACI**를 통해 액세스할 수 있는 모든 가능한 속성 목록이 유효 속성 아래에 표시됩니다. **Type**(유형)을 정의하면 사전 정의된 값 중 하나로 **Subtree** 및 **Target DN** 이 설정됩니다.
- **subtree** (필수)는 **subtree** 항목을 지정합니다. 그러면 이 **subtree** 항목 아래의 모든 항목을 대상으로 지정합니다. **Subtree** 는 와일드카드 또는 존재하지 않는 도메인 이름(**DN**)을 허용하지 않으므로 기존 하위 트리 항목을 제공합니다. 예:
cn=automount,dc=example,dc=com
- 추가 대상 필터는 **LDAP** 필터를 사용하여 권한이 적용되는 항목을 식별합니다. 필터는 유효한 **LDAP** 필터일 수 있습니다(예: **!(objectclass=posixgroup)**)
IdM은 지정된 필터의 유효성을 자동으로 확인합니다. 잘못된 필터를 입력하면 **IdM**에서 권한을 저장하려고 할 때 이 문제에 대해 경고합니다.
- 대상 **DN** 은 **DN**(도메인 이름)을 지정하고 와일드카드를 허용합니다. 예:
uid=*,cn=users,cn=accounts,dc=com
- 그룹의 멤버 는 대상 필터를 지정된 그룹의 멤버로 설정합니다. 필터 설정을 지정하고 **Add** 를 클릭하면 **IdM**에서 필터를 검증합니다. 모든 권한 설정이 올바르면 **IdM**에서 검색을

수행합니다. 일부 권한 설정이 올바르지 않으면 **IdM**은 어떤 설정이 잘못 설정되었는지 알려주는 메시지를 표시합니다.

7.

권한에 속성을 추가합니다.

- 유형을 설정하는 경우 사용 가능한 **ACI** 속성 목록에서 유효한 속성을 선택합니다.
- **Type** (유형)을 사용하지 않은 경우 **Effective attributes**(유효한 속성) 필드에 수동으로 속성을 추가하여 속성을 추가합니다. 한 번에 단일 속성을 추가합니다. 여러 특성을 추가하려면 **Add**(추가)를 클릭하여 다른 입력 필드를 추가합니다.



중요

권한에 대한 속성을 설정하지 않으면 권한에는 기본적으로 모든 속성이 포함됩니다.

8.

양식 하단에 **Add**(추가) 버튼을 사용하여 권한 추가를 완료합니다.

- **Add**(추가) 버튼을 클릭하여 권한을 저장하고 권한 목록으로 돌아갑니다.
- 또는 권한을 저장하고 **Add**(추가) 버튼을 클릭하여 동일한 양식에 추가 권한을 추가할 수 있습니다.
- **Add and Edit**(추가 및 편집) 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 목록의 이름 옆에 있는 확인란을 선택한 후 삭제 버튼을 클릭하여 **The Remove permissions** (권한 제거) 대화 상자를 표시합니다.



참고

기본 관리 권한에 대한 작업은 제한됩니다. 수정할 수 없는 속성은 **IdM 웹 UI**에서 비활성화되며 관리 권한을 완전히 삭제할 수 없습니다. 그러나 모든 권한에서 관리되는 권한을 제거하여 권한으로 바인드 유형이 설정된 관리 권한을 효과적으로 비활성화할 수 있습니다.

예를 들어 권한이 있는 사용자가 **engineers** 그룹에 **member** 속성을 작성하도록 하려면 (이므로) 멤버를 추가하거나 제거할 수 있습니다.

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

26.3. IDM WEBUI에서 권한 관리

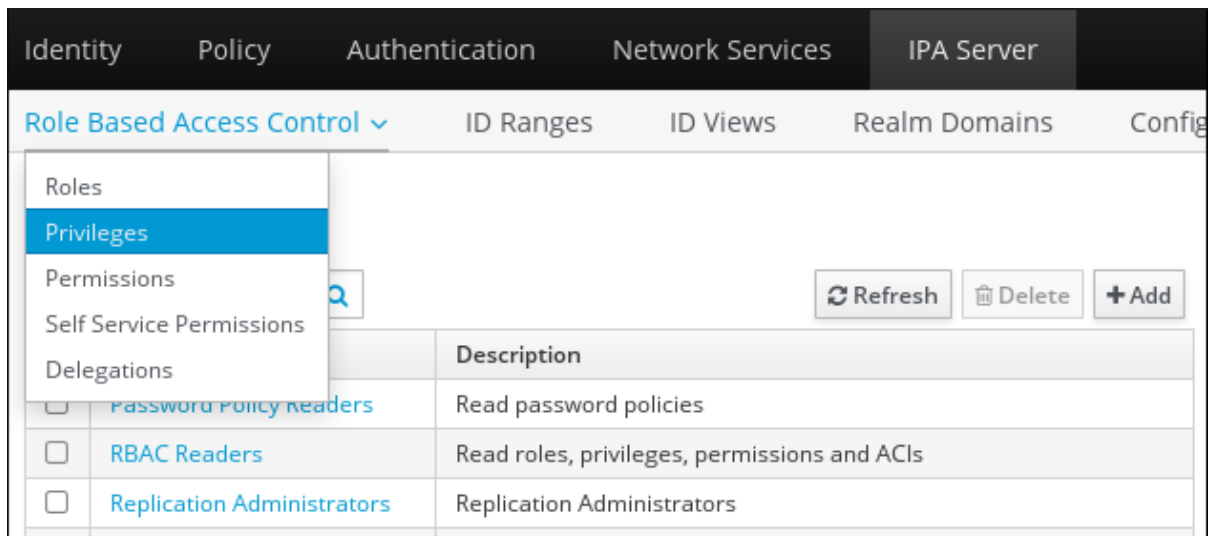
이 섹션에서는 웹 인터페이스(IdM 웹 UI)를 사용하여 IdM에서 권한을 관리하는 방법을 설명합니다.

사전 요구 사항

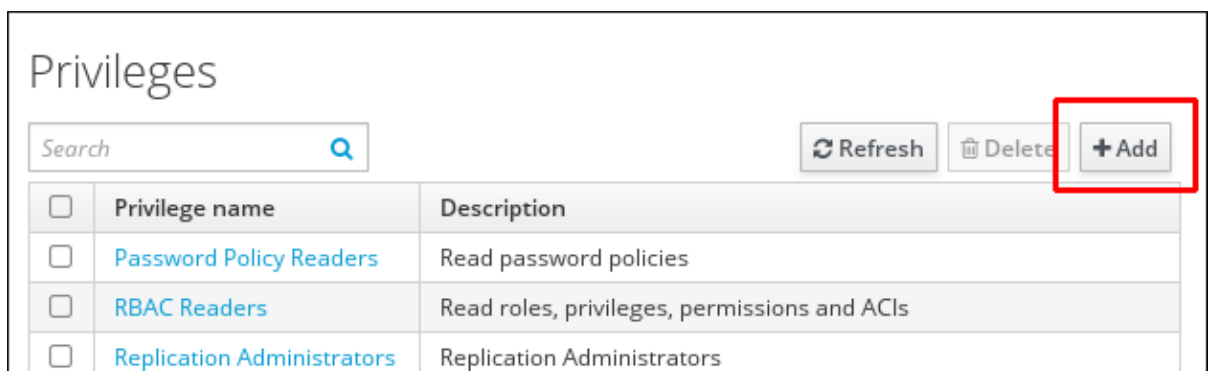
- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**
- **IdM 웹 UI에 로그인되어 있습니다.** 자세한 내용은 웹 브라우저에서 **IdM 웹 UI 액세스**를 참조하십시오.
- **기존 권한.** 권한에 대한 자세한 내용은 **IdM 웹 UI에서 권한** 관리를 참조하십시오.

절차

1. 새 권한을 추가하려면 **IPA Server** 탭에서 **Role-Based Access Control** (역할 기반 액세스 제어) 하위 메뉴를 열고 **Privileges**를 선택합니다.



2. 권한 목록이 열립니다. 권한 목록 상단에 있는 **Add** 단추를 클릭합니다.



3.

Add Privilege(권한 추가) 양식이 열립니다. 권한의 이름과 설명을 입력합니다.

Add Privilege [X]

Privilege name *

Description

* Required field

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

4.

Add and Edit(추가 및 편집) 버튼을 클릭하여 새 권한을 저장하고 권한 구성 페이지로 계속하여 권한을 추가합니다.

5.

permissions 목록에서 권한 이름을 클릭하여 권한 속성을 편집합니다. 권한 구성 페이지가 열립니다.

6.

Permissions(권한) 탭에는 선택한 권한에 포함된 권한 목록이 표시됩니다. 목록 상단에 있는 **Add**(추가) 버튼을 클릭하여 권한을 권한에 추가합니다.

Privilege: New Privilege

Permissions (5)

Settings

Roles

Refresh

Delete

+ Add

☐	Permission name
☐	System: Add Groups
☐	System: Add HBAC Rule
☐	System: Add Hosts
☐	System: Add Privileges
☐	System: Add Roles

7.

추가할 각 권한의 이름 옆에 있는 확인란을 선택하고 > 버튼을 사용하여 권한을 **Prospective** 열로 이동합니다.

Add Privilege New Privilege into Permissions

Filter available Permissions

Filter

Available

	Policy costemplate
☐	System: Add HBAC Service Groups
☐	System: Add HBAC Services
☐	System: Add Hostgroups
☐	System: Add krbPrincipalName to a Host
☐	System: Add Netgroups
☐	System: Add SELinux User Maps
☐	System: Add Sudo Command Group
☐	System: Delete Group Password Policy

Prospective

☐	Permission name
☐	System: Add Services
☐	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. 권한 에서 권한 제거 대화 상자가 열립니다.

10.

선택 사항: 기존 권한을 삭제해야 하는 경우 목록의 이름 옆에 있는 확인란을 선택한 후 삭제 버튼을 클릭합니다. 권한 제거 대화 상자가 열립니다.

26.4. IDM 웹 UI에서 역할 관리

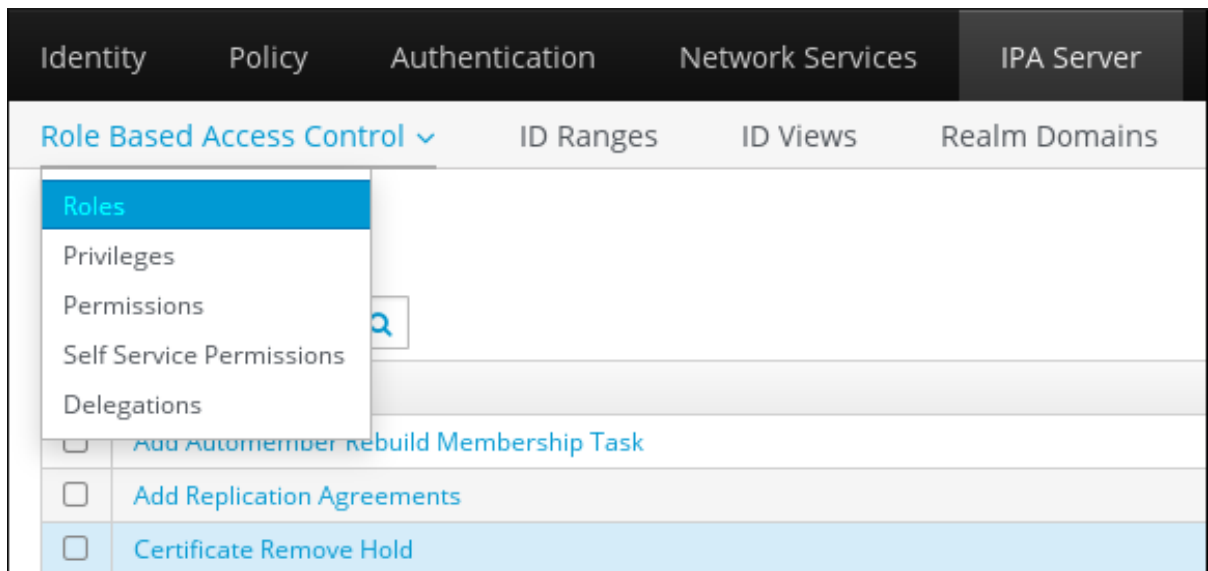
이 섹션에서는 웹 인터페이스(**IdM 웹 UI**)를 사용하여 **IdM(Identity Management)**에서 역할을 관리하는 방법을 설명합니다.

사전 요구 사항

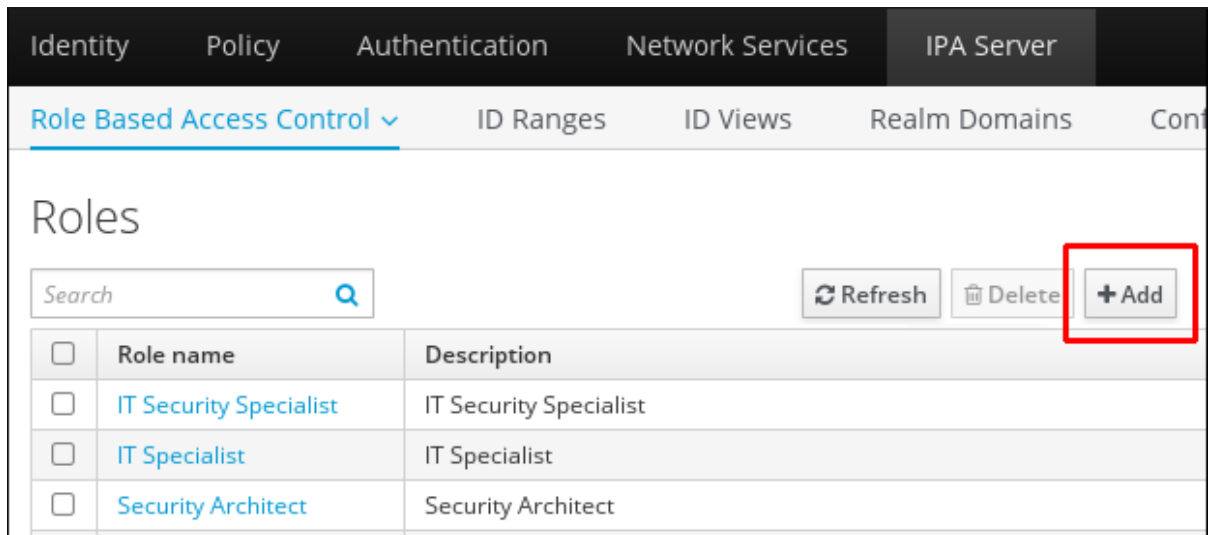
- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **IdM 웹 UI**에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 **IdM 웹 UI 액세스**를 참조하십시오.
- 기존 권한. 권한에 대한 자세한 내용은 **IdM 웹 UI**에서 **권한** 관리를 참조하십시오.

절차

1. 새 역할을 추가하려면 **IPA Server** 탭에서 **Role-Based Access Control** (역할 기반 액세스 제어) 하위 메뉴를 열고 **Roles** (역할)를 선택합니다.



2. 역할 목록이 열립니다. 역할 기반 액세스 제어 지침 목록 상단에 있는 **Add** (추가) 버튼을 클릭합니다.



3.

Add Role(역할 추가) 형식이 열립니다. 역할 이름과 설명을 입력합니다.

The screenshot shows the 'Add Role' dialog box. It has a title bar with 'Add Role' and a close button. The main area contains two input fields: 'Role name' (with a red asterisk indicating it is required) and 'Description'. The 'Role name' field contains the text 'Example Role' and the 'Description' field contains the text 'For engineers'. Below the input fields, there is a note '* Required field'. At the bottom of the dialog, there are four buttons: 'Add', 'Add and Add Another', 'Add and Edit', and 'Cancel'.

4.

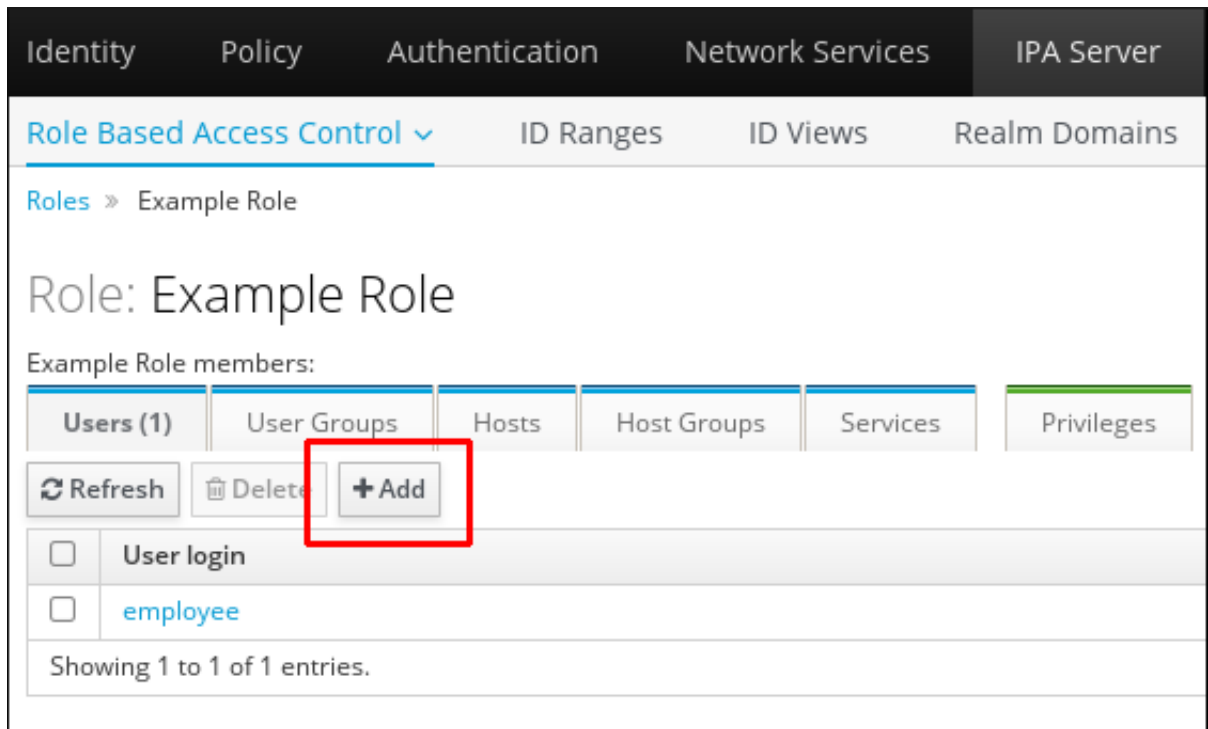
Add and Edit(추가 및 편집) 버튼을 클릭하여 새 역할을 저장하고 역할 구성 페이지로 이동하여 권한과 사용자를 추가합니다.

5.

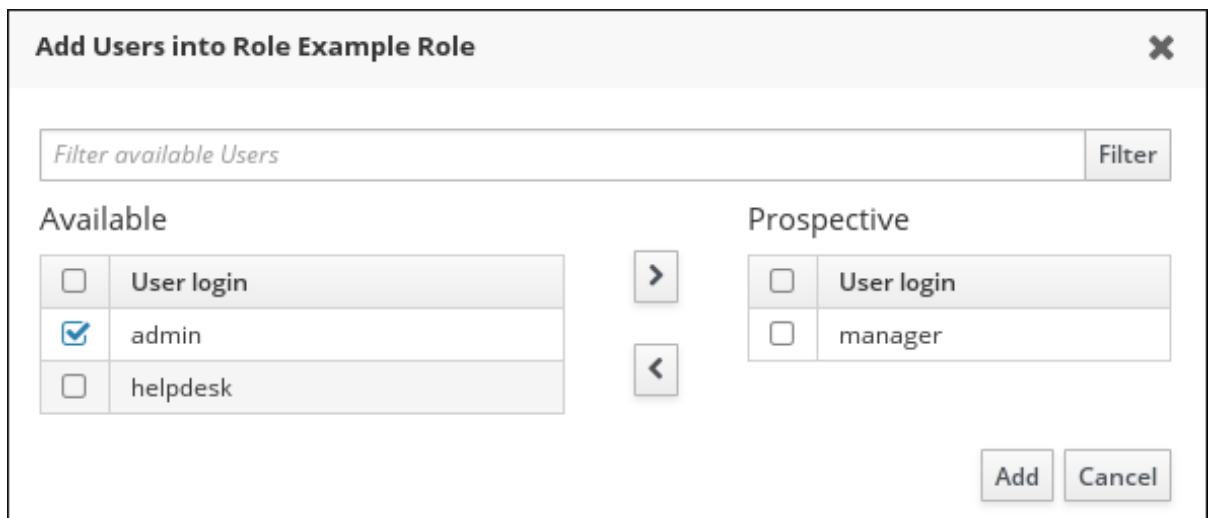
역할 목록에서 역할 이름을 클릭하여 역할 속성을 편집합니다. 역할 구성 페이지가 열립니다.

6.

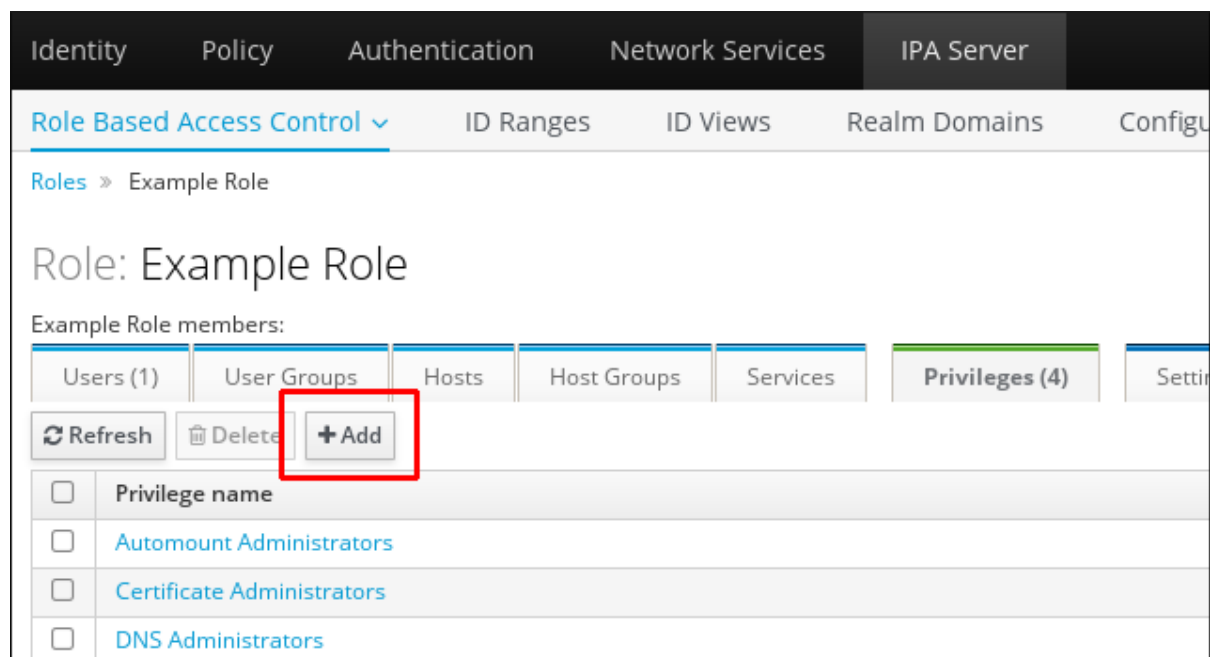
관련 목록 상단에 있는 **Add(추가)** 버튼을 클릭하여 **Users**, **Users Groups** (사용자, 사용자 그룹), **Hosts** (호스트), **Host Groups** (호스트 그룹) 또는 **Services** (서비스) 탭을 사용하여 멤버를 추가합니다.



7. 열리는 창에서 왼쪽의 구성원을 선택하고 > 버튼을 사용하여 **Prospective** 열로 이동합니다.



8. **Privileges** (권한) 탭의 상단에서 **Add**(추가)를 클릭합니다.



9. 왼쪽에서 권한을 선택하고 > 버튼을 사용하여 **Prospective** 열로 이동합니다.

Add Role Example Role into Privileges

Filter available Privileges

Filter

Available

☐	Privilege name
☐	ADTrust Agents
☐	Automember Readers
☐	Automember Task Administrator
☐	DNS Servers
☐	HBAC Administrator
☐	Host Enrollment
☐	IPA Masters Readers
☐	Kerberos Ticket Policy Readers
☐	Modify Group membership
☐	Modify Users and Reset passwords
☐	Netgroups Administrators
☐	New Privilege
☐	Password Policy Administrator
☐	Password Policy Readers

>

<

Prospective

☐	Privilege name
☐	Group Administrators
☐	Host Administrators
☐	Host Group Administrators

Add

Cancel

10. 저장하려면 **Add**(추가) 단추를 클릭합니다.

11. **선택 사항:** 역할에서 권한 또는 멤버를 제거해야 하는 경우 제거할 엔터티 이름 옆에 있는 확인란을 선택한 후 **Delete** (삭제) 단추를 클릭합니다. 대화 상자가 열립니다.

12. **선택 사항:** 기존 역할을 제거해야 하는 경우 목록의 이름 옆에 있는 확인란을 선택한 후 **Delete** (삭제) 버튼을 클릭하여 **Remove roles** (역할 제거) 대화 상자를 표시합니다.

27장. ANSIBLE 플레이북을 사용하여 IDM을 관리하기 위한 환경 준비

IdM(Identity Management)을 관리하는 시스템 관리자로서 Red Hat Ansible Engine을 사용하여 작업할 때 다음을 수행하는 것이 좋습니다.

- 홈 디렉터리에서 **Ansible** 플레이북 전용 하위 디렉터리(예: `~/MyPlaybooks`)를 생성합니다.
- `/usr/share/doc/ansible-freeipa/*` 및 `/usr/share/doc/rhel-system-roles/*` 디렉터리 및 하위 디렉터리에서 `~/MyPlaybooks` 디렉터리에 샘플 **Ansible** 플레이북을 복사 및 조정합니다.
- 인벤토리 파일을 `~/MyPlaybooks` 디렉터리에 포함합니다.

이 연습을 사용하면 한 곳에서 모든 플레이북을 찾을 수 있으며 루트 권한을 호출하지 않고도 플레이북을 실행할 수 있습니다.



참고

관리 노드에서 **root** 권한만 있으면 **ipaserver**, **ipareplica**, **ipa client** 및 **ipa backup ansible-freeipa** 역할을 실행할 수 있습니다. 이러한 역할을 수행하려면 디렉토리 및 **dnf** 소프트웨어 패키지 관리자 권한에 대한 액세스 권한이 필요합니다.

이 섹션에서는 `~/MyPlaybooks` 디렉터리를 만들고 **Ansible** 플레이북을 저장하고 실행하는 데 사용할 수 있도록 구성하는 방법을 설명합니다.

사전 요구 사항

- 관리형 노드인 **server.idm.example.com** 및 **replica.idm.example.com**에 **IdM** 서버를 설치했습니다.
- 제어 노드에서 직접 관리형 노드인 **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 디렉터리에서 **Ansible** 구성 및 플레이북의 디렉터리를 생성합니다.

```
$ mkdir ~/MyPlaybooks/
```

2. ~/MyPlaybooks/ 디렉터리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

3. 다음 콘텐츠를 사용하여 ~/Myplaybooks/ansible.cfg 파일을 만듭니다.

```
[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 이러한 위치에 있는 호스트에 대한 두 개의 호스트 그룹인 **eu** 와 **us** 를 정의합니다. 또한 이 구성은 **eu** 및 **us** 그룹의 모든 호스트를 포함하는 **ipaserver** 호스트 그룹을 정의합니다.

5. [선택 사항] SSH 공개 및 개인 키를 생성합니다. 테스트 환경에서 액세스를 간소화하려면 개인 키에 암호를 설정하지 마십시오.

```
$ ssh-keygen
```

6. SSH 공개 키를 각 관리 노드의 **IdM admin** 계정에 복사합니다.

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

이러한 명령을 사용하려면 **IdM** 관리자 암호를 입력해야 합니다.

추가 리소스

- **Ansible** 플레이북을 사용하여 **IdM** 서버 설치에 대한 자세한 내용은 **Ansible** 플레이북을 [사용하여 ID 관리 서버 설치를 참조하십시오](#).
- 예를 포함하여 **Ansible** 인벤토리 파일에 사용 가능한 형식 개요는 [How to build your inventory](#)([인벤토리 빌드 방법](#))를 참조하십시오.

28장. 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 활성화 등의 특정 작업을 수행할 수 있는 권한을 부여합니다.
- 권한은 권한을 결합합니다(예: 새 사용자를 추가하는 데 필요한 모든 권한).
- 역할은 사용자, 사용자 그룹, 호스트 또는 호스트 그룹에 일련의 권한을 부여합니다.

특히 대기업에서는 **RBAC**를 사용하면 개별 책임을 수행하는 관리자의 계층적 시스템을 만들 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 확인합니다.**
- **Ansible을 사용하여 IdM RBAC 역할이 없는지 확인합니다.**

- **Ansible**을 사용하여 사용자 그룹이 **IdM RBAC** 역할에 할당되었는지 확인합니다.
- **Ansible**을 사용하여 특정 사용자가 **IdM RBAC** 역할에 할당되지 않도록 합니다.
- 서비스가 **IdM RBAC** 역할의 멤버인지 확인하려면 **Ansible**을 사용합니다.
- 호스트가 **IdM RBAC** 역할의 멤버인지 확인하려면 **Ansible**을 사용합니다.
- **Ansible**을 사용하여 호스트 그룹이 **IdM RBAC** 역할의 멤버인지 확인합니다.

28.1. IDM의 권한

권한은 역할 기반 액세스 제어의 최하위 단위이며, 해당 작업이 적용되는 **LDAP** 항목과 함께 작업을 정의합니다. 구성 요소와 비교했을 때 필요한 수의 권한에 권한을 할당할 수 있습니다. 하나 이상의 권한은 다음을 허용하는 작업을 정의합니다.

- **write**
- **read**
- **search**
- **비교**
- **add**
- **delete**
- **all**

이러한 작업은 세 개의 기본 대상에 적용됩니다 :

- **subtree:** 도메인 이름(DN), 이 DN 아래의 하위 트리
- 대상 필터: **LDAP** 필터
- 대상: 항목을 지정하는 데 사용할 수 있는 와일드카드가 있는 **DN**

또한 다음과 같은 편의성 옵션은 해당 특성을 설정합니다.

- **type:** 유형의 개체 (사용자, 그룹 등)는 **subtree** 및 **target filter**를 설정합니다.
- **memberOf:** 그룹의 구성원; 대상 필터를 설정합니다.
- **targetGroup:** 특정 그룹을 수정할 액세스 권한을 부여합니다(예: 그룹 멤버십을 관리할 수 있는 권한 부여). 대상을 설정합니다.

IdM 권한을 사용하면 어떤 사용자가 어떤 오브젝트에 대한 액세스 권한이 있는지 그리고 이러한 오브젝트의 속성까지 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, 모든 인증된 사용자 또는 특정 권한 있는 사용자 그룹에 변경할 수 있습니다.

예를 들어 이 접근 방식의 권한의 유연성은 사용자 또는 그룹에 대한 액세스만 제한하려는 관리자에게 유용합니다. 사용자 또는 그룹이 액세스해야 하는 특정 섹션으로만 액세스하고 다른 섹션에 완전히 숨길 수 있도록 합니다.



참고

권한에는 다른 권한이 포함될 수 없습니다.

28.2. 기본 관리 권한

관리 권한은 기본적으로 **IdM**과 함께 제공되는 권한입니다. 다음과 같은 차이점을 사용하여 사용자가 생성한 다른 권한처럼 작동합니다.

- 해당 파일을 삭제하거나 이름, 위치, 대상 특성을 수정할 수 없습니다.
- 여기에는 세 가지 속성 세트가 있습니다.
 - 기본 속성은 **IdM**에서 관리하므로 사용자가 수정할 수 없습니다.
 - 사용자가 추가한 속성인 포함된 속성
 - 사용자가 제거하는 속성인 제외된 속성

관리되는 권한은 기본 및 포함된 특성 세트에 표시되지만 제외된 속성 세트에는 표시되지 않는 모든 속성에 적용됩니다.



참고

관리 권한을 삭제할 수는 없지만 바인딩 유형을 권한으로 설정하고 모든 권한에서 관리 권한을 제거하면 효과적으로 비활성화됩니다.

관리되는 모든 권한의 이름은 시스템:(예: 시스템): **Sudo rule** 또는 시스템을 추가합니다: **Services** 를 수정합니다. 이전 버전의 **IdM**에서는 기본 권한에 다른 체계를 사용했습니다. 예를 들어 사용자는 삭제할 수 없으며 권한에만 할당할 수 있었습니다. 이러한 기본 권한은 대부분 관리 권한으로 설정되었지만 다음 권한은 여전히 이전 체계를 사용합니다.

- **Automember Rebuild Membership** 작업 추가
- 구성 하위 항목 추가
- 복제 계약 추가
- 인증서 제거 보류

- **CA에서 인증서 상태 가져오기**
- **DNA 범위 읽기**
- **DNA 범위 수정**
- **PassSync 관리자 구성 읽기**
- **PassSync 관리자 구성 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 특성을 변경할 수 없으므로 명령이 실패합니다. 웹 UI에서 관리 권한을 수정하려고 하면 수정할 수 없는 속성이 비활성화됩니다.

28.3. IDM의 권한

권한은 역할에 적용할 수 있는 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** 사용자의 계정을 설정하려면 다음과 같은 권한이 필요합니다.

•

새 사용자 항목 생성

•

사용자 암호 재설정

•

새 사용자를 기본 **IPA** 사용자 그룹에 추가

이러한 세 가지 하위 수준 작업을 이름이 지정된 사용자 지정 권한의 형태로 더 높은 수준의 작업(예: 사용자 추가)에 결합하면 시스템 관리자가 역할을 쉽게 관리할 수 있습니다. **IdM**에는 이미 몇 가지 기본 권한이 포함되어 있습니다. 사용자 및 사용자 그룹 외에도 호스트 및 호스트 그룹은 네트워크 서비스에도 권한이 할당됩니다. 이 방법을 사용하면 특정 네트워크 서비스를 사용하는 호스트 집합의 사용자 집합에 의한 작업을 세부적으로 제어할 수 있습니다.



참고

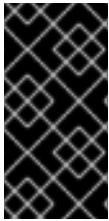
권한에는 다른 권한이 없을 수 있습니다.

28.4. IDM의 역할

역할은 역할에 지정된 사용자가 보유하는 권한 목록입니다.

실제로 권한은 지정된 하위 수준 작업을 수행할 수 있는 기능을 부여합니다(예: 사용자 항목 생성, 그룹에 항목 추가) 권한은 더 높은 수준의 작업(예: 지정된 그룹에 새 사용자 만들기)에 필요한 하나 이상의 권한

을 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 톨로 사용되지 않습니다.



참고

역할에 다른 역할은 포함할 수 없습니다.

28.5. ID 관리에 사전 정의된 역할

Red Hat Identity Management는 다음과 같은 일련의 사전 정의 역할을 제공합니다.

표 28.1. ID 관리에서 사전 정의된 역할

Role	권한	설명
헬프데스크	사용자 및 암호 재설정, 그룹 멤버십 수정	간단한 사용자 관리 작업 수행 담당
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 담당
보안 설계자	위임 관리자, 복제 관리자, 쓰기 IPA 구성, 암호 정책 관리자	Identity Management 환경 관리, 신뢰 생성, 복제 계약 생성
사용자 관리자	사용자 관리자, 그룹 관리자, 단계 사용자 관리자	사용자 및 그룹 생성 담당

28.6. ANSIBLE을 사용하여 권한이 있는 IDM RBAC 역할이 있는지 확인합니다.

기본 역할이 제공하는 기본 역할보다 IdM(Identity Management)의 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 정규화된 도메인 이름(FQDN)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 복사본을 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.
- 권한 목록을 새 역할에 포함하려는 **IdM** 권한의 이름으로 설정합니다.
- 또는 **user** 변수를 새 역할을 부여하려는 사용자 이름으로 설정합니다.
- 필요한 경우 **group** 변수를 새 역할을 부여할 그룹 이름으로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-user-present-copy.yml
```

추가 리소스

- **Ansible Vault**를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 [Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오](#).
- **IdM**의 역할 개념에 대한 자세한 내용은 **IdM**의 [역할을 참조하십시오](#).
- **iparole** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용할 수 있는 **README- role** 파일. 이 파일에는 **iparole** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉토리.

28.7. ANSIBLE을 사용하여 IDM RBAC 역할이 없는지 확인합니다.

IdM(Identity Management)에서 역할 기반 액세스 제어(**RBAC**)를 관리하는 시스템 관리자는 실수로 사용자에게 할당하지 않도록 사용되지 않는 역할이 없는지 확인해야 합니다.

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 정규화된 도메인 이름(**FQDN**)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-is-absent.yml` 파일의 복사본을 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.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-is-absent-copy.yml
```

추가 리소스

- **Ansible Vault**를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 [Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오](#).
- **IdM**의 역할 개념에 대한 자세한 내용은 **IdM의 역할을 참조하십시오**.
- **iparole** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- role** 마크다운 파일. 이 파일에는 **iparole** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉토리.

28.8. ANSIBLE을 사용하여 사용자 그룹이 **IDM RBAC** 역할에 할당되었는지 확인합니다.

IdM(Identity Management)에서 역할 기반 액세스 제어(**RBAC**)를 관리하는 시스템 관리자는 특정 사용자 그룹(예: 신참 관리자)에 역할을 할당할 수 있습니다.

다음 예제에서는 **Ansible** 플레이북을 사용하여 기본 제공 **IdM RBAC** 헬프데스크 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 정규화된 도메인 이름(FQDN)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 복사본을 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.
- **name** 변수를 할당할 역할의 이름으로 설정합니다.
- **그룹** 변수를 그룹 이름으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-group-present-copy.yml
```

추가 리소스

- **Ansible Vault**를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 [Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오](#).
- **IdM**의 역할 개념에 대한 자세한 내용은 **IdM**의 [역할을 참조하십시오](#).
- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북에 대해서는 `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- role** Markdown 파일을 참조하십시오. 파일에는 **iparole** 변수의 정의도 포함되어 있습니다.
- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북은 `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉터리를 참조하십시오.

28.9. ANSIBLE을 사용하여 특정 사용자가 IDM RBAC 역할에 할당되지 않도록 합니다.

IdM(Identity Management)에서 역할 기반 액세스 제어(**RBAC**)를 관리하는 시스템 관리자는 특정 사용자에게 **RBAC** 역할이 할당되지 않도록 할 수 있습니다(예: 회사 내 다른 위치로 이동).

다음 절차에서는 **Ansible** 플레이북을 사용하여 **user_01** 및 **user_02** 라는 사용자가 **helpdesk** 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 정규화된 도메인 이름(FQDN)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 복사본을 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.
- **name** 변수를 할당할 역할의 이름으로 설정합니다.

- 사용자 목록을 사용자 이름으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.
- **state** 변수를 **absent** 로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-user-absent-copy.yml
```

추가 리소스

- **Ansible Vault**를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 [Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오](#).
- **IdM**의 역할 개념에 대한 자세한 내용은 **IdM**의 [역할을 참조하십시오](#).
-

iparole 모듈을 사용하는 샘플 **Ansible** 플레이북에 대해서는 `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- role Markdown** 파일을 참조하십시오. 파일에는 **iparole** 변수의 정의도 포함되어 있습니다.

- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북은 `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉터리를 참조하십시오.

28.10. 서비스가 IDM RBAC 역할의 멤버인지 확인하려면 ANSIBLE을 사용합니다.

IdM(Identity Management)에서 역할 기반 액세스 제어(**RBAC**)를 관리하는 시스템 관리자는 **IdM**에 등록된 특정 서비스가 특정 역할의 구성원인지 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 정규화된 도메인 이름(**FQDN**)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
- **Ansible** 인벤토리 파일은 `~/<MyPlaybooks>/` 디렉터리에 있습니다.
- **web_administrator** 역할은 **IdM**에 있습니다.
- **HTTP/client01.idm.example.com@IDM.EXAMPLE.COM** 서비스는 **IdM**에 있습니다.

절차

1. `~/<MyPlaybooks>/` 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

- 2.

`/usr/share/doc/ansible-freeipa/playbooks/role/` 디렉터리에 있는 `role-member-service-present.yml` 파일의 복사본을 만듭니다.

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

3. `role-member-service-present-copy.yml` Ansible 플레이북 파일을 편집하여 편집합니다.

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

- `ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.
- `name` 변수를 할당할 역할의 이름으로 설정합니다.
- 서비스 목록을 서비스 이름으로 설정합니다.
- `action` 변수를 `member` 로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 Ansible 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-service-present-copy.yml
```

추가 리소스

- **Ansible Vault**를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 [Ansible Vault를 사용하여 콘텐츠 암호화](#)를 참조하십시오.
- **IdM**의 역할 개념에 대한 자세한 내용은 **IdM의 역할**을 참조하십시오.
- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북에 대해서는 `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- role Markdown** 파일을 참조하십시오. 파일에는 **iparole** 변수의 정의도 포함되어 있습니다.
- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북은 `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉터리를 참조하십시오.

28.11. 호스트가 IDM RBAC 역할의 멤버인지 확인하려면 ANSIBLE을 사용합니다.

IdM(Identity Management)에서 역할 기반 액세스 제어를 관리하는 시스템 관리자는 특정 호스트 또는 호스트 그룹이 특정 역할과 연결되어 있는지 확인할 수 있습니다. 다음 예제에서는 사용자 지정 **web_administrator** 역할이 **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 정규화된 도메인 이름(**FQDN**)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- **Ansible** 인벤토리 파일은 `~/<MyPlaybooks>/` 디렉터리에 있습니다.

- **web_administrator** 역할은 **IdM**에 있습니다.
- **client01.idm.example.com** 호스트는 **IdM**에 있습니다.

절차

1. **~/<MyPlaybooks>/** 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

2. **/usr/share/doc/ansible-freeipa/playbooks/role/** 디렉터리에 있는 **role-member-host-present.yml** 파일의 복사본을 만듭니다.

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

3. **role-member-host-present-copy.yml** Ansible 플레이북 파일을 편집하여 편집합니다.
4. **iparole** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 할당할 역할의 이름으로 설정합니다.
- 호스트 목록을 호스트 이름으로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-host-present-copy.yml
```

추가 리소스

- **Ansible Vault**를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 [Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오](#).
- **IdM**의 역할 개념에 대한 자세한 내용은 **IdM의 역할을 참조하십시오**.
- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북에 대해서는 **/usr/share/doc/ansible-freeipa/** 디렉터리에 있는 **README- role Markdown** 파일을 참조하십시오. 파일에는 **iparole** 변수의 정의도 포함되어 있습니다.
- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북은 **/usr/share/doc/ansible-freeipa/playbooks/iparole** 디렉터리를 참조하십시오.

28.12. ANSIBLE을 사용하여 호스트 그룹이 IDM RBAC 역할의 멤버인지 확인합니다.

IdM(Identity Management)에서 역할 기반 액세스 제어를 관리하는 시스템 관리자는 특정 호스트 또는 호스트 그룹이 특정 역할과 연결되어 있는지 확인할 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 정규화된 도메인 이름(FQDN)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 **~/<MyPlaybooks>/** 디렉터리에 있습니다.
- **web_administrator** 역할은 IdM에 있습니다.
- **web_servers** 호스트 그룹은 IdM에 있습니다.

절차

1. **~/<MyPlaybooks>/** 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

2. **/usr/share/doc/ansible-freeipa/playbooks/role/** 디렉터리에 있는 **role-member-hostgroup-present.yml** 파일의 복사본을 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.
- **name** 변수를 할당할 역할의 이름으로 설정합니다.
- **hostgroup** 목록을 호스트 그룹의 이름으로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-hostgroup-present-copy.yml
```

추가 리소스

- **Ansible Vault**를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 [Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오](#).
- **IdM**의 역할 개념에 대한 자세한 내용은 **IdM**의 [역할을 참조하십시오](#).
- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북에 대해서는 `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- role Markdown** 파일을 참조하십시오. 파일에는 **iparole** 변수의 정의도 포함되어 있습니다.
- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북은 `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉터리를 참조하십시오.

29장. ANSIBLE 플레이북을 사용하여 RBAC 권한 관리

RBAC(역할 기반 액세스 제어)는 역할, 권한 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. 특히 대기업에서는 **RBAC**를 사용하면 개별 책임을 수행하는 관리자의 계층적 시스템을 만들 수 있습니다.

이 장에서는 **IdM(Identity Management)**에서 **RBAC** 권한을 관리하기 위해 **Ansible** 플레이북을 사용하는 작업에 대해 설명합니다.

- **Ansible**을 사용하여 사용자 지정 **RBAC** 권한이 있는지 확인합니다.
- **Ansible**을 사용하여 멤버 권한이 사용자 지정 **IdM RBAC** 권한에 있는지 확인
- **Ansible**을 사용하여 **IdM RBAC** 권한에 권한이 포함되지 않도록 합니다.
- **Ansible**을 사용하여 사용자 지정 **IdM RBAC** 권한 이름 변경
- **Ansible**을 사용하여 **IdM RBAC** 권한이 없는지 확인합니다.

사전 요구 사항

- **RBAC**의 개념과 원칙을 이해합니다.

29.1. ANSIBLE을 사용하여 사용자 지정 ID RBAC 권한이 있는지 확인

IdM(Identity Management) 역할 기반 액세스 제어(**RBAC**)에서 사용자 지정 권한을 완전히 작동하려면 단계를 진행해야 합니다.

1. 권한이 연결되지 않고 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 결합하기 위한 **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 정규화된 도메인 이름(FQDN)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 복사본을 만듭니다.

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

3. 편집을 위해 **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 권한인 **full_host_administration** 의 이름으로 설정합니다.

●

선택적으로 **description** 변수를 사용하여 권한을 설명합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

29.2. ANSIBLE를 사용하여 멤버 권한이 사용자 지정 IDM RBAC 권한에 있는지 확인

IdM(Identity Management) 역할 기반 액세스 제어(**RBAC**)에서 사용자 지정 권한을 완전히 작동하려면 단계를 진행해야 합니다.

1.

권한이 연결되지 않고 권한을 생성합니다.

2.

선택한 권한을 권한에 추가합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 이전 단계에서 만든 권한에 권한을 추가하는 방법을 설명합니다. 이 예제에서는 호스트 관리와 관련된 모든 **IdM** 권한을 **full_host_administration** 이라는 권한에 추가하는 방법을 설명합니다. 기본적으로 권한은 **Host Enrollment**(호스트 등록), **Host Administrators**(호스트 관리자) 및 **Host Group Administrator** (호스트 그룹 관리자) 권한 사이에 배포됩니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 정규화된 도메인 이름(FQDN)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 **Ansible을 사용하여 사용자 지정 IdM RBAC 권한이 있는지를 참조하십시오.**

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/privilege/** 디렉터리에 있는 **privilege-member-present.yml** 파일의 복사본을 만듭니다.

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

3. 편집을 위해 **privilege-member-present-copy.yml** Ansible 플레이북 파일을 엽니다.

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

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.

- **name** 변수를 권한 이름으로 설정합니다.
- 권한 목록을 권한에 포함할 권한의 이름으로 설정합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 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:
        ipadmin_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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

29.3. ANSIBLE을 사용하여 **IDM RBAC** 권한에 권한이 포함되지 않도록 합니다.

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 관리자가 보안 위험이라고 간주하기 때문에 기본 **Certificate Administrators** 권한에서 **CA ACL** 권한을 무시하는 요청 인증서를 제거하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 정규화된 도메인 이름(FQDN)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-member-present.yml** 파일의 복사본을 만듭니다.

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

3. 편집을 위해 **privilege-member-absent-copy.yml** Ansible 플레이북 파일을 엽니다.

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

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.
- 권한 목록을 권한에서 제거할 권한의 이름으로 설정합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

29.4. ANSIBLE을 사용하여 사용자 지정 IDM RBAC 권한 이름 변경

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어를 사용자 지정할 수 있습니다.

다음 절차에서는 권한 이름을 변경하는 방법을 설명합니다. 예를 들어 몇 가지 권한을 제거했기 때문입니다. 결과적으로 권한 이름은 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 정규화된 도메인 이름(FQDN)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.
- **full_host_administration** 권한이 있습니다. 권한을 추가하는 방법에 대한 자세한 내용은 **Ansible을 사용하여 사용자 지정 IdM RBAC 권한이 있는지를 참조하십시오.**

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2.

`/usr/share/doc/ansible-freeipa/playbooks/privilege/` 디렉터리에 있는 `privilege-present.yml` 파일의 복사본을 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 권한의 현재 이름으로 설정합니다.
- `rename` 변수를 추가하고 권한의 새 이름으로 설정합니다.
- `state` 변수를 추가하고 이름을 바꿉니다.

5.

플레이북 자체의 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
---
- name: Rename a privilege
  hosts: ipaserver
  become: true
```

6.

플레이북의 작업의 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
[...]
tasks:
- name: Ensure the full_host_administration privilege is renamed to
  limited_host_administration
  ipaprivilege:
  [...]
```

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory rename-privilege.yml
```

29.5. ANSIBLE을 사용하여 IDM RBAC 권한이 없는지 확인합니다.

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 절차의 결과로 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 정규화된 도메인 이름(FQDN)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 복사본을 만듭니다.

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

3. 편집을 위해 `privilege-absent-copy.yml` Ansible 플레이북 파일을 엽니다.

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

- `ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.
- `name` 변수를 제거할 권한의 이름으로 설정합니다.
- `state` 변수가 `absent` 로 설정되어 있는지 확인합니다.

5. 플레이북의 작업의 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
[...]
tasks:
- name: Ensure privilege "CA administrator" is absent
  ipaprivilege:
  [...]
```

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

추가 리소스

- **IdM RBAC**의 권한 개념에 대한 자세한 내용은 [IdM](#)의 권한을 참조하십시오.
- **IdM RBAC**의 권한 개념에 대한 자세한 내용은 [Permissions in IdM](#)을 참조하십시오.
- **ipaprivilege** 모듈을 사용하는 샘플 **Ansible** 플레이북은 `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README-privilege** 파일을 참조하십시오. 파일에는 **ipaprivilege** 변수 정의도 포함되어 있습니다.
- **ipaprivilege** 모듈을 사용하는 샘플 **Ansible** 플레이북은 `/usr/share/doc/ansible-freeipa/playbooks/ipaprivilege` 디렉터리를 참조하십시오.

30장. 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 대기업에서는 **RBAC**를 사용하면 개별 책임을 수행하는 관리자의 계층적 시스템을 만들 수 있습니다.

이 장에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에서 **RBAC** 권한을 관리할 때 수행되는 다음 작업을 설명합니다.

- **Ansible**을 사용하여 **RBAC** 권한이 있는지 확인합니다.
- **Ansible**을 사용하여 속성이 있는 **RBAC** 권한이 있는지 확인합니다.
- **Ansible**을 사용하여 **RBAC** 권한이 없는지 확인합니다.
- **Ansible**을 사용하여 속성이 **IdM RBAC** 권한의 멤버인지 확인합니다.
- **Ansible**을 사용하여 속성이 **IdM RBAC** 권한의 멤버가 아닌지 확인합니다.
- **Ansible**을 사용하여 **IdM RBAC** 권한 이름 변경

사전 요구 사항

- **RBAC**의 개념과 원칙을 이해합니다.

30.1. ANSIBLE을 사용하여 RBAC 권한이 있는지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 권한에 권한을 추가할 수 있도록 **IdM**에 권한을 제공하는 방법을 설명합니다. 예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한은 호스트에만 적용할 수 있습니다.
- 권한이 포함된 권한이 부여된 사용자는 항목에서 다음과 같은 가능한 작업을 모두 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 **~/MyPlaybooks/** 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 **만들고 구성** 했다고 가정합니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2.

`/usr/share/doc/ansible-freeipa/playbooks/permission/` 디렉터리에 있는 `permission-present.yml` 파일의 복사본을 만듭니다.

```
$ 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 대응하도록 작업 이름을 조정합니다.
- `ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.
- `name` 변수를 권한의 이름으로 설정합니다.
- `object_type` 변수를 `host` 로 설정합니다.
- 올바른 변수를 `all` 로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

30.2. ANSIBLE을 사용하여 속성이 있는 RBAC 권한이 있는지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 권한에 권한을 추가할 수 있도록 **IdM**에 권한을 제공하는 방법을 설명합니다. 예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한은 호스트를 추가하는 데만 사용할 수 있습니다.
- 권한이 포함된 권한이 부여된 사용자는 호스트 항목에서 다음과 같은 모든 작업을 수행할 수 있습니다.
 - 쓰기
 - 읽기
 - 검색
 - 비교
 - add

○

delete

●

MyPermission 권한이 포함된 권한이 부여된 사용자가 생성한 호스트 항목은 **description** 값을 가질 수 있습니다.



참고

권한을 만들거나 수정할 때 지정할 수 있는 속성 유형은 **IdM LDAP** 스키마에 의해 제한되지 않습니다. 그러나 **object_type**이 호스트 일 경우 **attrs:car_licence** 를 지정하면 나중에 **ipa**가 됩니다. **ERROR: 속성 "car-license" not allowed error message when you try to a specificcar licence value to a host.**

사전 요구 사항

●

IdM 관리자 암호를 알고 있습니다.

●

Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.

●

이 예제에서는 **~/MyPlaybooks/** 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 **만들고 구성** 했다고 가정합니다.

절차

1.

~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2.

/usr/share/doc/ansible-freeipa/playbooks/permission/ 디렉터리에 있는 **permission-present.yml** 파일의 복사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-present.yml  
permission-present-with-attribute.yml
```

3.

편집을 위해 **permission-present-with-attribute.yml** **Ansible** 플레이북 파일을 엽니다.

4.

ipapermission 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한의 이름으로 설정합니다.
- **object_type** 변수를 **host** 로 설정합니다.
- 올바른 변수를 **all** 로 설정합니다.
- **attrs** 변수를 **description** 으로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory permission-present-with-attribute.yml
```

추가 리소스

- IdM 스키마에 대한 자세한 내용은 RHEL 7의 *Linux Domain Identity, Authentication and Policy Guide*의 [사용자 및 그룹 스키마](#)를 참조하십시오.

30.3. ANSIBLE을 사용하여 RBAC 권한이 없는지 확인합니다.

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 IdM에 권한이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 [ansible-freeipa](#) 패키지를 설치했습니다.
- 이 예제에서는 `~/MyPlaybooks/` 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 [만들고 구성](#) 했다고 가정합니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

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

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

3. 편집을 위해 `permission-absent-copy.yml` Ansible 플레이북 파일을 엽니다.

4.

ipapermission 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한의 이름으로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

30.4. ANSIBLE을 사용하여 속성이 **IDM RBAC** 권한의 멤버인지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 속성이 **IdM**에서 **RBAC** 권한의 멤버인지 확인하는 방법을 설명합니다. 결과적으로 권한이 있는 사용자는 특성이 있는 항목을 생성할 수 있습니다.

이 예제에서는 **MyPermission** 권한을 포함하는 권한이 있는 사용자가 생성한 호스트 항목에 **gecos** 및 **description** 값을 가질 수 있도록 하는 방법을 설명합니다.



참고

권한을 만들거나 수정할 때 지정할 수 있는 속성 유형은 **IdM LDAP** 스키마에 의해 제한되지 않습니다. 그러나 **object_type**이 호스트 일 경우 **attrs:car_licence** 를 지정하면 나중에 **ipa**가 됩니다. **ERROR: 속성 "car-license" not allowed error message when you try to a specificcar licence value to a host.**

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 **~/MyPlaybooks/** 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 **만들고 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** 파일의 복사본을 만듭니다.

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

3. 편집을 위해 **permission-member-present-copy.yml** **Ansible** 플레이북 파일을 엽니다.
4. **ipapermission** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- `ipaadmin_password` 변수를 `IdM` 관리자의 암호로 설정합니다.
- `name` 변수를 권한의 이름으로 설정합니다.
- `attrs` 목록을 `description` 및 `gecos` 변수로 설정합니다.
- 작업 변수가 `member` 로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

30.5. ANSIBLE을 사용하여 속성이 **IDM RBAC** 권한의 멤버가 아닌지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 속성이 **IdM**에서 **RBAC** 권한의 멤버가 아닌지 확인하는 방법을 설명합니다. 결과적으로 권한이 있는 사용자가 **IdM LDAP**에 항목을 생성하면 해당 항목에 특성과 연결된 값이 있을 수 없습니다.

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 **~/MyPlaybooks/** 디렉토리를 샘플 플레이북 복사본을 저장할 중앙 위치로 **만들고 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** 파일의 복사본을 만듭니다.

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

3. 편집을 위해 **permission-member-absent-copy.yml** Ansible 플레이북 파일을 엽니다.

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

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한의 이름으로 설정합니다.
- **attrs** 변수를 **description** 으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.
- **state** 변수가 **absent**로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

30.6. 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 변경하는 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 **~/MyPlaybooks/** 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 **만들고 구성** 했다고 가정합니다.
- **MyPermission** 은 **IdM**에 있습니다.
- **MyNewPermission** 은 **IdM**에 존재하지 않습니다.

절차

1.

~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2.

/usr/share/doc/ansible-freeipa/playbooks/permission/ 디렉터리에 있는 **permission-renamed.yml** 파일의 복사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-renamed.yml
permission-renamed-copy.yml
```

3. 편집을 위해 **permission-renamed-copy.yml** Ansible 플레이북 파일을 엽니다.

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

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한의 이름으로 설정합니다.

이는 현재 예제에 대한 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 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory permission-renamed-copy.yml
```

30.7. 추가 리소스

- IdM RBAC의 권한 개념에 대한 자세한 내용은 [Permissions in IdM](#)을 참조하십시오.
- IdM RBAC의 권한 개념에 대한 자세한 내용은 [IdM의 권한](#)을 참조하십시오.
- `ipapermission` 모듈을 사용하는 샘플 Ansible 플레이북에 대해서는 `/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용할 수 있는 `README-permission` 파일을 참조하십시오. 파일에는 `ipapermission` 변수 정의도 포함되어 있습니다.
- `ipapermission` 모듈을 사용하는 샘플 Ansible 플레이북은 `/usr/share/doc/ansible-freeipa/playbooks/ipapermission` 디렉터리를 참조하십시오.

31장. ID 보기를 사용하여 IDM 클라이언트에서 사용자 속성 값을 재정의

IdM(Identity Management) 사용자가 **IdM** 서버에 저장된 일부 사용자 또는 그룹 속성(예: 로그인 이름, 홈 디렉터리, 인증 또는 **SSH** 키)을 재정의하려는 경우 **IdM** 관리자가 **IdM ID** 보기를 사용하여 특정 **IdM** 클라이언트에 대해 이러한 값을 재정의할 수 있습니다. 예를 들어 사용자가 **IdM**에 로그인하는 데 가장 일반적으로 사용하는 **IdM** 클라이언트에서 사용자에게 대해 다른 홈 디렉터리를 지정할 수 있습니다.

이 장에서는 **IdM**에 클라이언트로 등록된 호스트에서 **IdM** 사용자와 관련된 **POSIX** 특성 값을 재정의하는 방법을 설명합니다. 특히 장에서는 사용자 로그인 이름과 홈 디렉토리를 재정의하는 방법을 설명합니다.

이 장에는 다음 섹션이 포함되어 있습니다.

- [ID 보기](#)
- [SSSD 성능에 대한 ID 뷰의 잠재적 영향](#)
- [ID 보기가 재정의할 수 있는 속성](#)
- [ID 보기 명령에 대한 도움말 가져오기](#)
- [ID 보기를 사용하여 특정 호스트의 IdM 사용자 로그인 이름 덮어쓰기](#)
- [IdM ID 보기 수정](#)
- [IdM 클라이언트에서 IdM 사용자 홈 디렉터리 재정의에 대해 ID 보기 추가](#)
- [IdM 호스트 그룹에 ID 보기 적용](#)

31.1. ID 보기

IdM(Identity Management)의 ID 보기는 다음 정보를 지정하는 **IdM** 클라이언트 측 보기입니다.

- 중앙에서 정의된 **POSIX** 사용자 또는 그룹 속성의 새 값

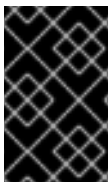
- 새 값이 적용되는 클라이언트 호스트 또는 호스트입니다.

ID 보기에는 하나 이상의 재정의가 포함되어 있습니다. 재정의는 중앙에서 정의된 **POSIX** 특성 값을 특정 대체합니다.

IdM 서버에서 중앙에서 **IdM** 클라이언트에 대한 **ID** 보기만 정의할 수 있습니다. **IdM** 클라이언트의 클라이언트 측 덮어쓰기를 로컬로 구성할 수 없습니다.

예를 들어 **ID** 뷰를 사용하여 다음 목표를 달성할 수 있습니다.

- 다양한 환경에 대한 다른 특성 값을 정의합니다. 예를 들어 **IdM** 관리자 또는 다른 **IdM** 사용자가 다른 **IdM** 클라이언트에 다른 홈 디렉토리를 가질 수 있습니다. **/home/encrypted/username** 을 한 **IdM** 클라이언트 및 **/dropbox/username** 에서 이 사용자의 홈 디렉토리로 구성할 수 있습니다. 이 상황에서 **ID** 보기를 사용하는 것은 대체 방법으로 사용할 수 있습니다. 예를 들어 **fallback_homedir,override_homedir** 또는 클라이언트의 **/etc/sss/sss.conf** 파일의 기타 홈 디렉터리 변수는 모든 사용자에게 영향을 미칩니다. 예제 절차는 **IdM 클라이언트에서 IdM 사용자 홈 디렉토리를 덮어쓰려면 ID 보기 추가**를 참조하십시오.
- 이전에 생성한 특성 값을 사용자의 **UID** 재정의와 같은 다른 값으로 바꿉니다. 이 기능은 **LDAP** 측에서 수행하기 어려운 시스템 전체 변경을 수행하려는 경우 유용할 수 있습니다(예: **IdM** 사용자의 **UID 1009**). **IdM** 사용자 **UID**를 생성하는 데 사용되는 **IdM ID** 범위는 **1000** 또는 **10000**으로 시작하지 않습니다. **IdM** 사용자가 모든 **IdM** 클라이언트에서 **UID 1009**를 사용하여 로컬 사용자를 가장하는 이유가 있는 경우 **ID** 보기를 사용하여 **IdM**에서 사용자를 생성할 때 생성된 이 **IdM** 사용자의 **UID**를 재정의할 수 있습니다.



중요

IdM 서버가 아닌 **IdM** 클라이언트에만 **ID** 뷰를 적용할 수 있습니다.

추가 리소스

- **AD(Active Directory)**와 관련된 환경에서 **ID** 보기를 사용할 수도 있습니다. 자세한 내용은 **Windows** 통합 가이드의 **ID 보기 및 기존 환경 마이그레이션 장**을 참조하십시오.
- 중앙 집중식 **ID** 관리 도메인에 속하지 않는 호스트에 대한 **ID** 뷰를 구성할 수도 있습니다. 자

세한 내용은 시스템 수준 인증 가이드의 [SSSD 클라이언트 측 보기 장](#)을 참조하십시오.

31.2. SSSD 성능에 대한 ID 뷰의 잠재적 영향

ID 뷰를 정의할 때 IdM은 IdM 서버의 SSSD(System Security Services Daemon) 캐시에 원하는 덮어쓰기 값을 배치합니다. 그러면 IdM 클라이언트에서 실행 중인 SSSD가 서버 캐시에서 덮어쓰기 값을 검색합니다.

특정 최적화 및 ID 보기를 동시에 실행할 수 없기 때문에 ID 보기를 적용하면 SSSD(System Security Services Daemon) 성능에 부정적인 영향을 미칠 수 있습니다. 예를 들어 ID 뷰는 SSSD가 서버에서 그룹을 조회하는 프로세스를 최적화하지 못하도록 합니다.

- ID 보기를 사용하면 그룹 이름이 재정의된 경우 SSSD에서 그룹 구성원 이름에서 반환된 모든 멤버를 확인해야 합니다.
- ID 보기가 없으면 SSSD는 그룹 오브젝트의 **member** 속성에서만 사용자 이름만 수집할 수 있습니다.

이러한 부정적인 영향은 SSSD 캐시가 비어 있거나 캐시를 지운 후 모든 항목을 잘못 만드는 경우 가장 드러납니다.

31.3. ID 보기가 재정의할 수 있는 속성

ID 보기는 사용자 및 그룹 ID 재정의로 구성됩니다. 재정의는 새로운 POSIX 특성 값을 정의합니다.

사용자 및 그룹 ID 재정의는 다음 POSIX 속성에 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(userCertificate)**

그룹 속성

- **그룹 이름(cn)**
- **그룹 GID 번호(gidNumber)**

31.4. ID 보기 명령에 대한 도움말 가져오기

IdM CLI(명령줄 인터페이스)에서 **IdM(Identity Management) ID** 뷰와 관련된 명령에 대한 도움말을 얻을 수 있습니다.

사전 요구 사항

- **IdM 사용자의 Kerberos 티켓을 획득했습니다.**

절차

- **ID 보기 및 재정의**를 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
[...]

```

31.5. ID 보기를 사용하여 특정 호스트의 IDM 사용자 로그인 이름 덮어쓰기

이 섹션에서는 **IdM(Identity Management)** 시스템 관리자로 특정 **IdM** 사용자와 관련된 **POSIX** 속성 값을 재정의하는 특정 **IdM** 클라이언트에 대한 **ID** 보기를 생성할 수 있는 방법을 설명합니다. 이 절차에서는 **idm_user**라는 **IdM** 사용자가 **user_1234** 로그인 이름을 사용하여 **host1**이라는 **IdM** 클라이언트에 로그인할 수 있는 **ID** 보기의 예를 사용합니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.

절차

1. 새 **ID** 보기 생성. 예를 들어 **example_for_host1**이라는 **ID** 보기를 생성하려면 다음을 실행합니다.

```
$ 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 재정의하려면 다음을 수행합니다.

- **ipa idoverrideuser-add** 명령을 이렇게 합니다.

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 보기에 있는 계정의 기존 인증서를 모두 대체합니다. 추가 인증서를 추가하려면 대신 **ipa idoverrideuser-add-cert** 명령을 사용하십시오.

```
$ ipa idoverrideuser-add-cert example_for_host1 user --
certificate="MIIETCC..."
```

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 보기를 적용하지만 ID 보기를 호스트 그룹 자체와 연결하지 않습니다. 대신, **--hostgroups** 옵션은 지정된 호스트 그룹의 멤버를 확장하고 **--hosts** 옵션을 각 호스트 그룹에 개별적으로 적용합니다.

즉, 나중에 호스트가 호스트 그룹에 추가되면 ID 보기가 새 호스트에 적용되지 않습니다.

5. 새 구성을 **host1.idm.example.com** 시스템에 즉시 적용하려면 다음을 수행합니다.

- a. 시스템에 **root**로 **SSH** 연결을 수행합니다.

```
$ 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 자격 증명이 있는 경우 이를 사용하여 **host1**의 **IdM**에 로그인할 수 있습니다.

1. **user_1234**를 로그인 이름으로 사용하여 **host1**에 **SSH**로 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** 자격 증명이 있는 경우 이를 사용하여 **id** 명령의 출력을 **idm_user** 및 **user_1234** 에 대해 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)
```

31.6. 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 **ID** 보기를 수정하여 **idm_user** 라는 사용자가 **host1.idm.example.com** **IdM** 클라이언트에서 **/home/user_1234/** 디렉터리를 사용자 홈 디렉터리로 사용하도록 설정하는 방법을 설명합니다.

사전 요구 사항

•

host1.idm.example.com 에 대한 루트 액세스 권한이 있습니다.

•

필요한 권한(예: **admin**)이 있는 사용자로 로그인했습니다.

•

host1 **IdM** 클라이언트에 적용되는 **idm_user** 에 대해 **ID** 보기가 구성되어 있습니다.

절차

1.

root로 **idm_user** 가 **host1.idm.example.com** 에서 사용자 홈 디렉터리로 사용할 디렉터리를 생성합니다.

```
[root@host1 /]# mkdir /home/user_1234/
```

2.

디렉터리의 소유권을 변경합니다.

```
[root@host1 /]# chown idm_user:idm_user /home/user_1234/
```

3.

ID 보기가 현재 적용되는 호스트를 포함하여 ID 보기를 표시합니다. **example_for_host1** 이라는 ID 보기를 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
```

출력에 ID 보기가 현재 **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
- **home dir** 옵션을 추가합니다.

```
$ ipa idoverrideuser-mod example_for_host1 idm_user --
homedir=/home/user_1234
-----
Modified an 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 수행합니다.

a.

시스템에 **root**로 **SSH** 연결을 수행합니다.

```
$ 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** 를 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/
```

31.7. IDM 클라이언트에서 IDM 사용자 홈 디렉터리 재정의의 위해 ID 보기 추가

IdM(Identity Management)의 **ID** 보기는 특정 **IdM** 사용자와 관련된 **POSIX** 특성 값을 재정의합니다. 이 섹션에서는 사용자가 **/home/ user_1234/** 디렉터리를 **/home/ idm_user/** 대신 사용자 홈 디렉터리로 사용할 수 있도록 **host1** 이라는 **IdM** 클라이언트에서 **idm_user** 에 적용되는 **ID** 보기를 생성하는 방법을 설명합니다.

사전 요구 사항

-

host1.idm.example.com 에 대한 루트 액세스 권한이 있습니다.

- 필요한 권한(예: **admin**)이 있는 사용자로 로그인했습니다.

절차

1. **root**로 **idm_user**가 **host1.idm.example.com**에서 사용자 홈 디렉터리로 사용할 디렉터리를 생성합니다.

```
[root@host1 /]# mkdir /home/user_1234/
```

2. 디렉터리의 소유권을 변경합니다.

```
[root@host1 /]# chown idm_user:idm_user /home/user_1234/
```

3. ID 뷰를 생성합니다. 예를 들어 **example_for_host1**이라는 ID 보기를 생성하려면 다음을 실행합니다.

```
$ 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 디렉터리를 재정의하려면 다음을 수행합니다.

- **ipa idoverrideuser-add** 명령을 입력합니다.

- ID 보기의 이름 추가

- 앵커라고도 하는 사용자 이름 추가

- **home dir** 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 보기를 적용하지만 ID 보기를 호스트 그룹 자체와 연결하지 않습니다. 대신, `--hostgroups` 옵션은 지정된 호스트 그룹의 멤버를 확장하고 `--hosts` 옵션을 각 호스트 그룹에 개별적으로 적용합니다.

즉, 나중에 호스트가 호스트 그룹에 추가되면 ID 보기가 새 호스트에 적용되지 않습니다.

6.

새 구성을 `host1.idm.example.com` 시스템에 즉시 적용하려면 다음을 수행합니다.

a.

시스템에 **root**로 **SSH** 연결을 수행합니다.

```
$ 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** 를 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 디렉터리를 출력합니다.

```
[idm_user@host1 ~]$ pwd
/home/user_1234/
```

31.8. IDM 호스트 그룹에 ID 보기 적용

ipa idview-apply 명령은 **--hostgroups** 옵션을 허용합니다. 그러나 옵션은 현재 지정된 호스트 그룹에 속해 있지만 ID 보기를 호스트 그룹 자체와 동적으로 연결하지 않는 호스트에 ID 뷰를 적용하는 일회성 작업 역할을 합니다. **hostgroups** 옵션은 지정된 호스트 그룹의 멤버를 확장하고 **--hosts** 옵션을 모든 호스트에 개별적으로 적용합니다.

나중에 호스트 그룹에 새 호스트를 추가하는 경우 **--hosts** 옵션과 함께 **ipa idview-apply** 명령을 사용하여 새 호스트에 ID 보기를 수동으로 적용해야 합니다.

마찬가지로 호스트 그룹에서 호스트를 제거해도 제거 후에도 ID 뷰가 여전히 호스트에 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 호스트를 추가하는 방법.
2. 호스트 그룹에 ID 보기를 적용하는 방법.
3. 새 호스트를 호스트 그룹에 추가하고 ID 보기를 새 호스트에 적용하는 방법.

사전 요구 사항

- 호스트 그룹에 적용할 ID 보기가 IdM에 있는지 확인합니다. 예를 들어 특정 IdM 클라이언트에서 IdM 사용자 로그인 이름을 재정의하기 위해 ID 보기를 생성하려면 특정 호스트의 IdM 사용자 로그인 이름을 재정의하는 ID 보기를 참조하십시오.

절차

1.

호스트 그룹을 생성하고 호스트를 추가합니다.

a.

호스트 그룹을 만듭니다. 예를 들어 호스트 그룹을 생성하려면 다음을 실행합니다.

```
[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 host 102 및 host 103 을 hub timore 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 hub timore 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 새 호스트에 ID 보기를 적용합니다.

a.

새 호스트를 호스트 그룹에 추가합니다. 예를 들어 **somehost.idm.example.com** 호스트를 **parent timore** 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 **catalog timore** 호스트 그룹의 새로 추가된 호스트인 **somehost.idm.example.com**에 적용되지 않았음을 보여줍니다.

c.

ID 보기를 새 호스트에 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
```

출력에서 ID 보기가 이제 **satellite timore** 호스트 그룹의 새로 추가된 호스트인 **somehost.idm.example.com** 에 적용됨을 보여줍니다.

32장. 수동으로 ID 범위 조정

IdM 서버는 고유한 사용자 ID(UID) 및 그룹 ID(GID) 번호를 생성합니다. 복제본에 다양한 ID 범위를 만들고 할당하면 동일한 ID 번호를 생성하지 않도록 합니다. 기본적으로 이 프로세스는 자동입니다. 그러나 IdM 서버를 설치하는 동안 IdM ID 범위를 수동으로 조정하거나 복제본의 DNA ID 범위를 수동으로 정의할 수 있습니다.

32.1. ID 범위

ID 번호는 *ID 범위*로 구분됩니다. 개별 서버 및 복제본에 대해 별도의 숫자 범위를 유지하면 항목에 대해 발행된 ID 번호를 다른 서버 또는 복제본의 다른 항목에서 이미 사용할 가능성이 제거됩니다.

두 가지 유형의 ID 범위가 있습니다.

- 첫 번째 서버를 설치하는 동안 할당되는 **IdM ID 범위**. 이 범위는 생성된 후에는 수정할 수 없습니다. 그러나 필요한 경우 원래 ID 외에도 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. 이러한 값은 기존 IdM ID 범위에 적합해야 합니다. 자세한 내용은 [수동으로 DNA ID 범위 조정](#)을 참조하십시오.

복제본에 다음 **DNA ID 범위**가 할당될 수도 있습니다. 복제본은 현재 범위의 ID가 부족할 때 다음 범위를 사용합니다. 다음 범위는 복제본이 삭제되거나 [수동으로 설정](#)할 수 있는 경우 [자동으로 할당](#)됩니다.

도메인의 백엔드 **389 Directory Server** 인스턴스의 일부로 **DNA** 플러그인에 의해 서버와 복제본 간에 범위가 업데이트되고 공유됩니다.

DNA 범위 정의는 두 가지 특성으로 설정되며, 서버의 다음에 사용 가능한 숫자(**DNA 범위의 낮은 끝**) 및 최대 값(**DNA 범위의 상단 끝**)으로 설정됩니다. 초기 하단 범위는 플러그인 인스턴스 구성 중에 설정됩니다. 그런 다음 플러그인은 하단 값을 업데이트합니다. 사용 가능한 숫자를 범위로 분할하면 서버는 서로 겹치지 않고 번호를 계속 할당할 수 있습니다.

32.2. 자동 ID 범위 할당

기본적으로 IdM ID 범위는 IdM 서버 설치 중에 자동으로 할당됩니다. **ipa-server-install** 명령은 총 10,000개 범위의 ID 범위를 임의로 선택하고 할당합니다. 이와 같이 임의의 범위를 선택하면 향후 두 개의

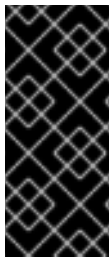
별도의 IdM 도메인을 병합하기로 결정한 경우 ID 충돌 가능성이 크게 줄어듭니다.



참고

IdM ID 범위를 생성한 후에는 수정할 수 없습니다. **DNA ID 범위 조정에 설명된 명령을 사용하여 수동으로 DCA(Distributed Numeric Assignment) ID 범위를 수동으로 조정할 수 있습니다. IdM ID 범위와 일치하는 DNA 범위가 설치 중에 자동으로 생성됩니다.**

IdM 서버가 1개 설치되어 있으면 전체 **DNA ID** 범위를 제어합니다. 새 복제본을 설치하고 복제본이 고유한 **DNA ID** 범위를 요청하면 서버 분할의 초기 ID 범위가 분리되고 서버와 복제본 간에 배포됩니다. 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 각 부분을 사용합니다. 또한 복제본이 할당된 ID 범위를 제거하고 100개 미만의 ID를 유지하는 경우에도 복제본은 사용 가능한 다른 서버에 연결하여 새 **DNA ID** 범위를 요청합니다.



중요

복제본을 설치하면 ID 범위가 즉시 수신 되지 않습니다. 복제본은 **DNA** 플러그인이 처음 사용되는 경우(예: 사용자를 처음 추가할 때) ID 범위를 수신합니다. 그런 다음 복제본의 ID 범위가 정의되어 있지 않습니다.

복제본에서 **DNA ID** 범위를 요청하기 전에 초기 서버가 작동을 중지하면 복제본이 서버에 연결하여 ID 범위를 요청할 수 없습니다. 복제본에 새 사용자를 추가하려고 하면 실패합니다. 이러한 **상황에서는 비활성화된 서버에 할당된 ID 범위를 확인하고 복제본에 ID 범위를 수동으로 할당할 수 있습니다.**

32.3. 서버 설치 중 IdM ID 범위 수동 할당

임의로 할당하는 대신 기본 동작을 재정의하고 IdM ID 범위를 수동으로 설정할 수 있습니다.



중요

UID 값이 1000 이하인 ID 범위를 설정하지 마십시오. 이러한 값은 시스템용으로 예약되어 있습니다. 또한 0 값을 포함할 ID 범위를 설정하지 마십시오. **SSSD** 서비스에서 0 ID 값을 처리하지 않습니다.

절차



ipa-server-install 과 함께 다음 두 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
-----
```

32.4. 새 ID ID 범위 추가

경우에 따라 원래 ID 외에도 새 **IdM** ID 범위를 생성할 수 있습니다(예: 복제본의 ID가 부족하고 원래 **IdM** ID 범위가 고갈된 경우).

중요

새 **IdM** ID 범위를 추가하면 새 **DNA ID** 범위가 자동으로 생성되지 않습니다. 필요에 따라 새 **DNA ID** 범위를 수동으로 할당해야 합니다. 이 작업을 수행하는 방법에 대한 자세한 내용은 [DNA ID 범위 조정을 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 지우고 서비스를 다시 시작하지 않으면 SSSD에서 새 ID 범위를 확인하는 데 다소 시간이 걸립니다. 보다 구체적으로, IdM 서버에 저장된 도메인 목록 및 기타 구성 데이터를 업데이트할 때 범위를 확인합니다.

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
-----
```

32.5. 신뢰할 수 있는 AD를 제거한 후 ID 범위 제거

IdM 및 AD(Active Directory) 환경 간에 신뢰가 제거된 경우 연결된 ID 범위를 제거할 수 있습니다.



주의

신뢰할 수 있는 도메인과 연결된 ID 범위에 할당된 ID는 여전히 IdM에 등록된 시스템의 파일 및 디렉터리 소유권에 사용될 수 있습니다.

제거한 AD 트러스트에 해당하는 ID 범위를 제거하면 AD 사용자가 소유한 파일과 디렉터리의 소유권을 확인할 수 없습니다.

사전 요구 사항

- AD 환경에 대한 신뢰성을 제거했습니다.

절차

1. 현재 사용 중인 모든 ID 범위를 표시합니다.

```
[root@server ~]# ipa idrange-find
```

2. 제거한 신뢰와 연결된 ID 범위의 이름을 식별합니다. ID 범위 이름의 첫 번째 부분은 신뢰의 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

- 명령줄에서 AD에 대한 신뢰성을 제거하는 단계는 [명령줄을 사용하여 신뢰 제거를 참조하십시오.](#)
- IdM 웹 UI에서 AD에 대한 신뢰성을 제거하는 단계는 [IdM 웹 UI를 사용하여 신뢰 제거를 참조하십시오.](#)

32.6. 현재 할당된 DNA ID 범위 표시

현재 활성화된 DNA(Distributed Numeric Assignment) ID 범위와 서버에 모두 DNA(Distributed Numeric Assignment) ID 범위를 할당한 경우 해당 다음 DNA 범위를 표시할 수 있습니다.

절차

- 토폴로지의 서버에 구성된 DNA ID 범위를 표시하려면 다음 명령을 사용합니다.
 - `ipa-replica-manage dnrangle-show` 는 모든 서버에 설정된 현재 DNA ID 범위를 표시하거나 지정된 서버에만 서버를 지정하는 경우 다음을 수행합니다.

```
# ipa-replica-manage dnrangle-show
serverA.example.com: 1001-1500
serverB.example.com: 1501-2000
serverC.example.com: No range set

# ipa-replica-manage dnrangle-show serverA.example.com
serverA.example.com: 1001-1500
```

- `ipa-replica-manage dnanextrange-show` 는 현재 모든 서버에 설정된 다음 DNA ID 범위를 표시하거나 지정된 서버에만 서버를 지정하는 경우 다음을 표시합니다.

```
# ipa-replica-manage dnanextrange-show
serverA.example.com: 2001-2500
serverB.example.com: No on-deck range set
serverC.example.com: No on-deck range set

# ipa-replica-manage dnanextrange-show serverA.example.com
serverA.example.com: 2001-2500
```

32.7. 자동 DNA ID 범위 확장

RHEL 8 IdM(Identity Management) 복제본을 삭제하면 기본적으로 복제에 할당된 **ID 범위**가 토폴로지의 나머지 다른 복제본으로 전송되지 않습니다. **ID 범위**를 사용하려면 수동으로 다른 복제본으로 전송해야 합니다.

절차

이 예제에서는 **serverB.example.com**을 삭제하고 **server C.example.com**에 전송하여 **ID 범위**를 보존하려고 합니다.

1. **serverB.example.com**에 현재 할당된 범위를 확인하려면 **ipa-replica-manage dnrange-show** 명령을 사용합니다.

```
# ipa-replica-manage dnrange-show
serverA.example.com: 1001-1500
serverB.example.com: 1501-2000
serverC.example.com: No range set
```

출력에는 **serverB.example.com**에 **1501-2000 ID 범위**가 할당되었음을 보여줍니다.



참고

서버에 설정된 다음 **ID 범위**를 확인하려면 대신 **ipa-replica-manage dnanextrange-show**를 사용합니다.

2. **ipa-replica-manage dnrange-set** 명령을 사용하여 **ID 범위**를 **serverC.example.com**에 할당합니다.

```
# ipa-replica-manage dnrange-set serverC.example.com 1501-2000
```

검증 단계

1. **serverB.example.com** 삭제 :

```
# ipa-server-del serverB.example.com
```

2. **serverC.example.com**에 설정된 **DNA ID 범위**를 확인합니다.

```
# ipa-replica-manage dnrange-show serverC.example.com
serverC.example.com: 1501-2000
```

32.8. 수동 DNA ID 범위 조정

특정 상황에서는 다음과 같은 경우 **DNA(Distributed Numeric Assignment) ID** 범위를 수동으로 조정해야 합니다.

- 복제본에 **ID**가 부족하고 **IdM ID** 범위가 제거되었습니다.

복제본은 할당된 **DNA ID** 범위를 고갈시키고 **IdM** 범위에서 사용 가능한 **ID**를 더 이상 사용할 수 없기 때문에 추가 **ID**를 요청하지 못했습니다.

이러한 상황을 해결하려면 복제본에 할당된 **DNA ID** 범위를 확장합니다. 다음 두 가지 방법으로 이 작업을 수행할 수 있습니다.

- 다른 복제본에 할당된 **DNA ID** 범위를 단축한 다음 새로 사용 가능한 값을 고갈된 복제본에 할당합니다.
- 새 **IdM ID** 범위를 생성한 다음 이 **IdM** 범위 내에서 복제본의 새 **DNA ID** 범위를 설정합니다.

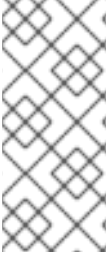
새 **IdM ID** 범위를 생성하는 방법에 대한 자세한 내용은 [새 IdM ID 범위 추가](#)를 참조하십시오.

- 복제본의 작동이 중지되었습니다

복제본의 **DNA ID** 범위는 종료되고 삭제해야 하는 경우 자동으로 검색되지 않습니다. 즉, 이전에 복제본에 할당된 **DNA ID** 범위를 사용할 수 없게 됩니다. **DNA ID** 범위를 복구하고 다른 복제본에서 사용할 수 있도록 합니다.

작동을 중지하고 다른 서버에 할당하는 복제본에 속하는 **DNA ID** 범위를 복구하려면 먼저 해당 범위를 다른 서버에 수동으로 할당하기 전에 **ID 범위 값이 무엇인지 확인해야** 합니다. 또한 중복된 **UID** 또는 **GID**를 방지하려면 복구된 범위의 **ID** 값이 사용자 또는 그룹에 할당되지 않았는지 확인합니다. 기존 사용자와 그룹의 **UID** 및 **GID**를 검사하여 이 작업을 수행할 수 있습니다.

수동으로 **DNA ID** 범위 조정 명령을 사용하여 복제본의 **DNA ID** 범위를 수동으로 조정할 수 있습니다.



참고

새 **DNA ID** 범위를 할당하면 서버 또는 복제본에 있는 기존 항목의 **UID**가 동일하게 유지됩니다. 현재 **DNA ID** 범위를 변경해도 **IdM**은 과거에 할당된 범위에 대한 레코드를 유지하므로 문제가 발생하지 않습니다.

32.9. 수동으로 DNA ID 범위 조정

경우에 따라 기능이 아닌 복제본에 할당된 **DNA ID** 범위를 다시 할당하기 위해 기존 복제본의 **DNA(Distributed Numeric Assignment) ID** 범위를 수동으로 조정해야 할 수 있습니다. 자세한 내용은 [Manual DNA ID 범위 조정](#)을 참조하십시오.

DNA ID 범위를 수동으로 조정할 때 새로 조정된 범위가 **IdM ID** 범위에 포함되어 있는지 확인합니다. `ipa idrange-find` 명령을 사용하여 확인할 수 있습니다. 그렇지 않으면 명령이 실패합니다.



중요

중복되는 **ID** 범위를 만들지 않도록 주의하십시오. 서버 또는 복제본에 할당한 **ID** 범위가 겹치는 경우 서로 다른 두 개의 서버에서 동일한 **ID** 값을 다른 항목에 할당할 수 있습니다.

사전 요구 사항

- 선택 사항:** 작동하지 않는 복제본에서 **DNA ID** 범위를 복구하는 경우 먼저 [현재 할당된 DNA ID 범위 표시에 설명된 명령을 사용하여 ID 범위를 찾습니다](#).

절차

- 지정된 서버에 대한 현재 **DNA ID** 범위를 정의하려면 `ipa-replica-manage dnarange-set` 를 사용합니다.

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

- 지정된 서버에 대한 다음 **DNA ID** 범위를 정의하려면 `ipa-replica-manage dnanextrange-set` 을 사용합니다.

```
# ipa-replica-manage dnanextrange-set serverB.example.com 1500-5000
```

검증 단계

-

현재 할당된 **DNA ID** 범위 표시에 설명된 명령을 사용하여 새로운 **DNA** 범위가 올바르게 설정되었는지 확인할 수 있습니다.

33장. ANSIBLE을 사용하여 IDM의 복제 토폴로지 관리

여러 IdM(Identity Management) 서버를 유지 관리하고 중복을 위해 서로 복제하여 서버 손실을 완화하거나 방지할 수 있습니다. 예를 들어, 한 서버가 실패하면 다른 서버는 계속 도메인에 서비스를 제공합니다. 나머지 서버 중 하나를 기반으로 새 복제본을 만들어 손실된 서버를 복구할 수도 있습니다.

IdM 서버에 저장된 데이터는 복제 계약에 따라 복제됩니다. 두 서버에 복제 계약이 구성된 경우 해당 데이터를 공유합니다. 복제된 데이터는 토폴로지 접미사에 저장됩니다. 두 복제본에 접미사 간 복제 계약이 있는 경우 접미사는 토폴로지 세그먼트를 형성합니다.

이 장에서는 Red Hat Ansible Engine 을 사용하여 IdM 복제 계약, 토폴로지 세그먼트 및 토폴로지 접미사를 관리하는 방법을 설명합니다. 장에는 다음 섹션이 포함되어 있습니다.

- [Ansible을 사용하여 IdM에 복제 계약이 있는지 확인](#)
- [Ansible을 사용하여 여러 IdM 복제본 간에 복제 계약이 있는지 확인](#)
- [Ansible을 사용하여 두 복제본 간에 복제 계약이 있는지 확인](#)
- [Ansible을 사용하여 토폴로지 접미사가 IdM에 있는지 확인합니다.](#)
- [Ansible을 사용하여 IdM 복제본 다시 초기화](#)
- [Ansible을 사용하여 IdM에 복제 계약이 없음](#)

33.1. ANSIBLE을 사용하여 IDM에 복제 계약이 있는지 확인

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약에 따라 복제됩니다. 두 서버에 복제 계약이 구성된 경우 데이터를 공유합니다. 복제 계약은 항상 양방향입니다. 데이터는 첫 번째 복제본에서 다른 복제본으로는 물론 다른 복제본으로 복제됩니다.

이 섹션에서는 Ansible 플레이북을 사용하여 `server.idm.example.com`과 `replica.idm.example.com` 간에 도메인 유형의 복제 계약이 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- 토폴로지 에서 복제본 연결에 나열된 **IdM** 토폴로지 설계 권장 사항을 이해해야 합니다.
- **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/` 디렉터리에 이러한 옵션을 구성하는 **IdM** 서버의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/topology/` 디렉터리에 있는 **add-topology segment.yml** Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegment.yml  
add-topologysegment-copy.yml
```

3. **add-topologysegment-copy.yml** 파일을 열어 편집합니다.
4. **ipatopologysegment** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
 - **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.

- 추가하려는 세그먼트 유형에 따라 접미사 변수를 **domain** 또는 **ca** 로 설정합니다.
- 복제 계약의 왼쪽 노드가 되고자 하는 **IdM** 서버의 이름으로 왼쪽 변수를 설정합니다.
- 올바른 변수를 복제 계약의 올바른 노드로 설정하고자 하는 **IdM** 서버의 이름으로 설정합니다.
- **state** 변수가 **present**로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - name: Add topology segment
    ipatopologysegment:
      ipadmin_password: Secret123
      suffix: domain
      left: server.idm.example.com
      right: replica.idm.example.com
      state: present
```

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

```
$ ansible-playbook -v -i inventory add-topologysegment-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 [설명 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트](#) 를 참조하십시오.
- **ipatopologysegment** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.

- `/usr/share/doc/ansible-freeipa/` 디렉터리의 **README- topology.md** 파일. 이 파일에는 `ipatopologysegment` 변수 정의도 포함되어 있습니다.
- `/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉토리.

33.2. ANSIBLE을 사용하여 여러 IDM 복제본 간에 복제 계약이 있는지 확인

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약에 따라 복제됩니다. 두 서버에 복제 계약이 구성된 경우 데이터를 공유합니다. 복제 계약은 항상 양방향입니다. 데이터는 첫 번째 복제본에서 다른 복제본으로는 물론 다른 복제본으로 복제됩니다.

이 섹션에서는 **IdM**의 여러 복제본 쌍 간에 복제 계약이 있는지 확인하는 방법에 대해 설명합니다.

사전 요구 사항

- 토폴로지에서 복제본 연결에 나열된 **IdM** 토폴로지 설계 권장 사항을 이해했는지 확인하십시오. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/single/planning_identity_management/index#connecting-the-replicas-in-a-topology_planning-the-replica-topology
- **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/` 디렉토리에 이러한 옵션을 구성하는 **IdM** 서버의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉토리로 이동합니다.

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

2.

`/usr/share/doc/ansible-freeipa/playbooks/topology/` 디렉터리에 있는 `add-topologysegments.yml` Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegments.yml
add-topologysegments-copy.yml
```

3.

`add-topologysegments-copy.yml` 파일을 열어 편집합니다.

4.

`vars` 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 `IdM` 관리자의 암호로 설정합니다.
- 모든 토폴로지 세그먼트에 대해 `ipatopology_segments` 섹션에 행을 추가하고 다음 변수를 설정합니다.
 - 추가하려는 세그먼트 유형에 따라 접미사 변수를 `domain` 또는 `ca` 로 설정합니다.
 - 복제 계약의 왼쪽 노드가 되고자 하는 `IdM` 서버의 이름으로 왼쪽 변수를 설정합니다.
 - 올바른 변수를 복제 계약의 올바른 노드로 설정하고자 하는 `IdM` 서버의 이름으로 설정합니다.

5.

`add-topologysegments-copy.yml` 파일의 `tasks` 섹션에서 `state` 변수가 `present`로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Add topology segments
  hosts: ipaserver
  become: true
  gather_facts: false

  vars:
```

```

ipaadmin_password: Secret123
ipatopology_segments:
- {suffix: domain, left: replica1.idm.example.com , right: replica2.idm.example.com }
- {suffix: domain, left: replica2.idm.example.com , right: replica3.idm.example.com }
- {suffix: domain, left: replica3.idm.example.com , right: replica4.idm.example.com }
- {suffix: domain+ca, left: replica4.idm.example.com , right:
replica1.idm.example.com }

tasks:
- name: Add topology segment
  ipatopologysegment:
    ipaadmin_password: "{{ ipaadmin_password }}"
    suffix: "{{ item.suffix }}"
    name: "{{ item.name | default(omit) }}"
    left: "{{ item.left }}"
    right: "{{ item.right }}"
    state: present
    #state: absent
    #state: checked
    #state: reinitialized
    loop: "{{ ipatopology_segments | default([]) }}"

```

6.

파일을 저장합니다.

7.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory add-topologysegments-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 [설명 복제 계약](#), [토폴로지 접미사 및 토폴로지 세그먼트](#)를 참조하십시오.
- **ipatopologysegment** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리의 **README- topology.md** 파일. 이 파일에는 **ipatopologysegment** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉토리.

33.3. ANSIBLE을 사용하여 두 복제본 간에 복제 계약이 있는지 확인

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약에 따라 복제됩니다. 두 서버에 복제 계약이 구성된 경우 데이터를 공유합니다. 복제 계약은 항상 양방향입니다. 데이터는 첫 번째 복제본에서 다른 복제본으로는 물론 다른 복제본으로 복제됩니다.

이 섹션에서는 **IdM**의 여러 복제본 쌍 간에 복제 계약이 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- 토폴로지 에서 복제본 연결에 나열된 **IdM(Identity Management)** 토폴로지 설계 권장 사항을 이해해야 합니다.
- **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/` 디렉터리에 이러한 옵션을 구성하는 **IdM** 서버의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/topology/` 디렉터리에 있는 **check-topologysegments.yml** **Ansible** 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/check-topologysegments.yml
check-topologysegments-copy.yml
```

3.

check-topologysegments-copy.yml 파일을 열어 편집합니다.

4.

vars 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

•

ipaadmin_password 변수를 **IdM** 관리자의 암호로 설정합니다.

•

모든 토폴로지 세그먼트에 대해 **ipatopology_segments** 섹션에 행을 추가하고 다음 변수를 설정합니다.

○

추가하려는 세그먼트 유형에 따라 접미사 변수를 **domain** 또는 **ca** 로 설정합니다.

○

복제 계약의 왼쪽 노드가 되고자 하는 **IdM** 서버의 이름으로 왼쪽 변수를 설정합니다.

○

올바른 변수를 복제 계약의 올바른 노드로 설정하고자 하는 **IdM** 서버의 이름으로 설정합니다.

5.

check-topologysegments-copy.yml 파일의 **tasks** 섹션에서 **state** 변수가 **present**로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Add topology segments
  hosts: ipaserver
  become: true
  gather_facts: false

  vars:
    ipaadmin_password: Secret123
    ipatopology_segments:
      - {suffix: domain, left: replica1.idm.example.com, right: replica2.idm.example.com }
      - {suffix: domain, left: replica2.idm.example.com , right: replica3.idm.example.com }
      - {suffix: domain, left: replica3.idm.example.com , right: replica4.idm.example.com }
      - {suffix: domain+ca, left: replica4.idm.example.com , right:
        replica1.idm.example.com }

  tasks:
    - name: Check topology segment
      ipatopologysegment:
```



```

ipaadmin_password: "{{ ipaadmin_password }}"
suffix: "{{ item.suffix }}"
name: "{{ item.name | default(omit) }}"
left: "{{ item.left }}"
right: "{{ item.right }}"
state: checked
loop: "{{ ipatopology_segments | default([]) }}"

```

6.

파일을 저장합니다.

7.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory check-topologysegments-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 [설명 복제 계약](#), [토폴로지 접미사](#) 및 [토폴로지 세그먼트](#) 를 참조하십시오.
- **ipatopologysegment** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리의 **README- topology.md** 파일. 이 파일에는 **ipatopologysegment** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉토리.

33.4. ANSIBLE을 사용하여 토폴로지 접미사가 IDM에 있는지 확인합니다.

IdM(Identity Management)의 복제 계약 컨텍스트에서 토폴로지 접미사는 복제되는 데이터를 저장합니다. **IdM**은 두 가지 유형의 토폴로지 접미사인 **domain** 과 **ca** 를 지원합니다. 각 접미사는 별도의 백엔드인 별도의 복제 토폴로지를 나타냅니다. 복제 계약이 구성되면 서로 다른 두 서버에서 동일한 유형의 두 토폴로지 접미사를 결합합니다.

도메인 접미사에는 사용자, 그룹 및 정책과 같은 모든 도메인 관련 데이터가 포함됩니다. **ca** 접미사에는 **Certificate System** 구성 요소에 대한 데이터가 포함되어 있습니다. **CA**(인증 기관)가 설치된 서버에만 존재합니다.

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM**에 토폴로지 접미사가 있는지 확인하는 방법을 설명

합니다. 이 예제에서는 도메인 접미사가 **IdM**에 있는지 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.
 - `~/MyPlaybooks/` 디렉터리에 이러한 옵션을 구성하는 **IdM** 서버의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/topology/` 디렉터리에 있는 `verify-topology suffix.yml` **Ansible** 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/ verify-topologysuffix.yml
verify-topologysuffix-copy.yml
```

3. 편집을 위해 `verify-topologysuffix-copy.yml` **Ansible** 플레이북 파일을 엽니다.
4. `ipatologysuffix` 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 **IdM** 관리자의 암호로 설정합니다.
- 접미사 변수를 `domain` 로 설정합니다. `ca` 접미사가 있는지 확인하는 경우 변수를 `ca` 로

설정합니다.

- **state** 변수가 **verify** 로 설정되어 있는지 확인합니다 . 다른 옵션은 없습니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
    - name: Verify topology suffix
      ipatopologysuffix:
        ipaadmin_password: Secret123
        suffix: domain
        state: verified
```

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

```
$ ansible-playbook -v -i inventory verify-topologysuffix-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 [설명 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트](#) 를 참조하십시오.
- **ipatopologysegment** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리의 **README- topology.md** 파일. 이 파일에는 **ipatopologysuffix** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉토리.

33.5. ANSIBLE을 사용하여 IDM 복제본 다시 초기화

복제본이 장기간 오프라인 상태이거나 데이터베이스가 손상된 경우 다시 초기화할 수 있습니다. 다시 초기화하면 업데이트된 데이터 세트로 복제본을 새로 고칩니다. 예를 들어 백업에서 권한 있는 복원이 필요한 경우 다시 초기화할 수 있습니다.



참고

복제 업데이트와 대조적으로 복제본이 변경된 항목만 상호 전송하는 반면 다시 초기화하면 전체 데이터베이스가 새로 고칩니다.

명령을 실행하는 로컬 호스트는 다시 초기화된 복제본입니다. 데이터를 가져오는 복제본을 지정하려면 **direction** 옵션을 사용합니다.

이 섹션에서는 **Ansible** 플레이북을 사용하여 **server.idm.example.com**에서 **replica.idm.example.com**의 도메인 데이터를 다시 초기화하는 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.
 - **~/MyPlaybooks/** 디렉터리에 이러한 옵션을 구성하는 **IdM** 서버의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

- 2.

`/usr/share/doc/ansible-freeipa/playbooks/topology/` 디렉터리에 있는 `reinitialize-topology segment.yml` Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/reinitialize-topologysegment.yml reinitialize-topologysegment-copy.yml
```

3.

편집을 위해 `reinitialize-topologysegment-copy.yml` 파일을 엽니다.

4.

`ipatopologysegment` 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 **IdM** 관리자의 암호로 설정합니다.
- 접미사 변수를 **domain** 로 설정합니다. **ca** 데이터를 다시 초기화하는 경우 변수를 **ca** 로 설정합니다.
- 왼쪽 변수를 복제 계약의 왼쪽 노드로 설정합니다.
- 올바른 변수를 복제 계약의 올바른 노드로 설정합니다.
- 방향 변수를 다시 초기화 데이터 방향으로 설정합니다. 왼쪽에서 오른쪽 방향으로 데이터를 왼쪽 노드에서 오른쪽 노드로 이동합니다.
- `state` 변수가 다시 초기화 되도록 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - name: Reinitialize topology segment
    ipatopologysegment:
      ipaadmin_password: Secret123
      suffix: domain
      left: server.idm.example.com
```

```
right: replica.idm.example.com
direction: left-to-right
state: reinitialized
```

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

```
$ ansible-playbook -v -i inventory reinitialize-topologysegment-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 [설명 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트](#)를 참조하십시오.
- **ipatopologysegment** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리의 **README- topology.md** 파일. 이 파일에는 **ipatopologysegment** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉토리.

33.6. ANSIBLE을 사용하여 IDM에 복제 계약이 없음

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약에 따라 복제됩니다. 두 서버에 복제 계약이 구성된 경우 데이터를 공유합니다. 복제 계약은 항상 양방향입니다. 데이터는 첫 번째 복제본에서 다른 복제본으로는 물론 다른 복제본으로 복제됩니다.

이 섹션에서는 두 복제본 간의 복제 연결이 **IdM**에 없는지 확인하는 방법에 대해 설명합니다. 이 예제에서는 **replica01.idm.example.com** 및 **replica02.idm.example.com** **IdM** 서버 사이에 도메인 유형의 복제 계약이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- 토폴로지에서 복제본 연결에 나열된 **IdM** 토폴로지 설계 권장 사항을 이해했는지 확인하십시오. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-

single/planning_identity_management/index#connecting-the-replicas-in-a-topology_planning-the-replica-topology

- 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/` 디렉터리에 이러한 옵션을 구성하는 IdM 서버의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/topology/` 디렉터리에 있는 **delete-topologysegment.yml** Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/delete-topologysegment.yml
delete-topologysegment-copy.yml
```

3. 편집을 위해 **delete-topologysegment-copy.yml** 파일을 엽니다.

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

- **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.
- 접미사 변수를 **domain** 로 설정합니다. 또는 왼쪽 및 오른쪽 노드 간에 **ca** 데이터가 복제되지 않도록 하는 경우 변수를 **ca** 로 설정합니다.

- 왼쪽 변수를 복제 계약의 왼쪽 노드인 **IdM** 서버의 이름으로 설정합니다.
- 올바른 변수를 복제 계약의 올바른 노드인 **IdM** 서버의 이름으로 설정합니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - name: Delete topology segment
    ipatopologysegment:
      ipaadmin_password: Secret123
      suffix: domain
      left: replica01.idm.example.com
      right: replica02.idm.example.com:
      state: absent
```

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

```
$ ansible-playbook -v -i inventory delete-topologysegment-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 [설명 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트](#) 를 참조하십시오.
- **ipatopologysegment** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉터리의 **README- topology.md** 파일. 이 파일에

는 `ipatopologysegment` 변수 정의도 포함되어 있습니다.

- `/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉토리.

33.7. 추가 리소스

- **IdM** 배포의 토폴로지를 계획하는 방법에 대한 자세한 내용은 [복제 토폴로지](#) 계획을 참조하십시오.
- **IdM** 복제본을 설치하는 방법에 대한 자세한 내용은 **IdM** 복제본 설치를 참조하십시오 .

34장. 사용자의 외부 프로비저닝을 위한 IDM 구성

시스템 관리자는 ID 관리를 위해 외부 솔루션에서 사용자의 프로비저닝을 지원하도록 **IdM(Identity Management)**을 구성할 수 있습니다.

ipa 유틸리티를 사용하는 대신 외부 프로비저닝 시스템 관리자는 **ldapmodify** 유틸리티를 사용하여 **IdM LDAP**에 액세스할 수 있습니다. 관리자는 **ldapmodify** 또는 **LDIF** 파일을 사용하여 **CLI**에서 개별 단계 사용자를 추가할 수 있습니다.

IdM 관리자는 검증된 사용자만 추가하도록 외부 프로비저닝 시스템을 완전히 신뢰한다고 가정합니다. 그러나 동시에 외부 프로비저닝 시스템의 관리자에게 사용자 관리자의 **IdM** 역할을 할당하여 새 활성 사용자를 직접 추가할 수 없습니다.

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

이 장에는 다음 섹션이 포함되어 있습니다.

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

추가 자료

더 높은 권한이 필요한 사용자 및 그룹 관리 작업을 수행하기 위해 **ldapmodify** 를 전체 **IdM** 관리자로 사용하기 위한 예제 및 템플릿은 **IdM 사용자를 외부적으로 관리하는 데 ldapmodify 사용**을 참조하십시오

오.

34.1. 단계 사용자 계정의 자동 활성화를 위한 IDM 계정 준비

다음 절차에서는 외부 프로비저닝 시스템에서 사용할 두 개의 **IdM** 사용자 계정을 구성하는 방법을 설명합니다. 적절한 암호 정책이 있는 그룹에 계정을 추가하면 외부 프로비저닝 시스템에서 **IdM**의 사용자 프로비저닝을 관리할 수 있습니다. 다음에서는 외부 시스템에서 단계 사용자를 추가하는 데 사용하는 사용자 계정은 **provisionator** 라고 합니다. 스테이징 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 만듭니다.

- a. **provisionator** 사용자 계정을 추가합니다.

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

- a. 프로비저너 사용자에게 필요한 권한을 부여합니다.

- i. 사용자 지정 역할인 **System Provisioning** 을 생성하여 스테이징 사용자 추가를 관리합니다.

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

- ii. 역할에 **Stage User Provisioning**(사용자 프로비저닝 단계) 권한을 추가합니다. 이 권한은 스테이징 사용자를 추가할 수 있는 기능을 제공합니다.

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

iii.

provisionator 사용자를 역할에 추가합니다.

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

iv.

프로비저너가 **IdM**에 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 활성화 기를 만듭니다.

a.

활성화기 사용자 계정을 추가합니다.

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

b.

사용자를 기본 **User Administrator** 역할에 추가하여 필요한 권한을 활성화 사용자에게 부여합니다.

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

4.

애플리케이션 계정의 사용자 그룹을 생성합니다.

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

5.

그룹의 암호 정책을 업데이트합니다. 다음 정책은 계정의 암호 만료 및 잠금을 방지하지만 복잡한 암호를 요구하여 잠재적인 위험을 보상합니다.

```
$ 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) 암호 정책이 **IdM**에 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
```

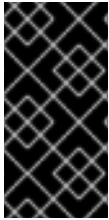
새 **IdM** 사용자 암호가 즉시 만료되기 때문에 암호를 변경해야 합니다.

추가 리소스:

- 새 사용자 추가에 대한 자세한 내용은 [명령줄을 사용하여 사용자 계정](#) 관리를 참조하십시오.
- 사용자에게 다른 사용자 계정을 관리하는 데 필요한 권한을 부여하는 방법에 대한 자세한 내용은 사용자 [위임](#)을 참조하십시오.
- **IdM** 암호 정책 관리에 대한 자세한 내용은 [IdM 암호 정책 정의](#)를 참조하십시오.

34.2. IDM 스테이징 사용자 계정의 자동 활성화 구성

다음 절차에서는 스테이징 사용자를 활성화하는 스크립트를 생성하는 방법을 설명합니다. 시스템은 지정된 시간 간격에 따라 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 스크립트에서 **IdM**에 추가 검증이 필요하지 않다고 가정합니다.

IdM 서버 중 하나에서만 활성화 프로세스를 활성화할 수 있습니다.

사전 요구 사항

- 프로비저너 및 활성 기 계정은 **IdM**에 있습니다. 자세한 내용은 [단계 사용자 계정 자동 활성화를 위한 IdM 계정 준비](#)를 참조하십시오.
- 프로시저를 실행 중인 **IdM** 서버에 대한 **root** 권한이 있습니다.
- **IdM** 관리자로 로그인했습니다.
- 외부 프로비저닝 시스템을 신뢰합니다.

절차

1. 활성화 계정에 대한 **keytab** 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 **keytab** 파일을 생성합니다. 그런 다음 키탭 파일을 다른 서버에 복사합니다.

2. 다음 내용으로 **/usr/local/sbin/ipa-activate-all** 스크립트를 생성하여 모든 사용자를 활성화합니다.

```
#!/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 내용을 사용하여 **systemd** 타이머 **/etc/systemd/system/ipa-activate-all.timer** 을 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:

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

8.

ipa-activate-all.timer 시작 :

```
# 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.
```

34.3. LDIF 파일에 정의된 IDM 단계 사용자 추가

이 섹션에서는 외부 프로비저닝 시스템의 관리자가 **IdM LDAP**에 액세스하고 **LDIF** 파일을 사용하여 스테이징 사용자를 추가하는 방법에 대해 설명합니다. 아래 예제에서는 단일 사용자를 추가하는 방법을 보여 주지만, 일괄 모드로 하나의 파일에 여러 사용자를 추가할 수 있습니다.

사전 요구 사항

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

○

uid**절차**

1.

외부 서버에서 새 사용자에 대한 정보가 포함된 **LDIF** 파일을 생성합니다.

```
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 프로비저너로 연결합니다.

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

4.

IdM 서버에서 프로비저너 계정의 **Kerberos** 티켓을 받습니다.

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

5.

f 옵션과 **LDIF** 파일의 이름을 사용하여 **ldapadd** 명령을 입력합니다. **IdM** 서버의 이름과 포트 번호를 지정합니다.

```
~]$ ldapadd -h server.idm.example.com -p 389 -f add-stageidmuser.ldif
SASL/GSSAPI authentication started
SASL username: provisionator@IDM.EXAMPLE.COM
SASL SSF: 256
```

SASL data security layer installed.
adding the entry "uid=stageidmuser,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com"

34.4. LDAPMODIFY를 사용하여 CLI에서 직접 IDM 단계 사용자 추가

이 섹션에서는 외부 프로비저닝 시스템의 관리자가 **IdM(Identity Management) LDAP**에 액세스하고 **ldapmodify** 유틸리티를 사용하여 **stage** 사용자를 추가하는 방법에 대해 설명합니다.

사전 요구 사항

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

절차

1. **IdM ID** 및 자격 증명을 사용하여 **SSH** 프로토콜을 사용하여 **IdM** 서버에 연결합니다.

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

2. 새 단계 사용자를 추가하는 역할의 **IdM** 사용자인 프로비저너 계정의 **TGT**를 가져옵니다.

```
$ kinit provisionator
```

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

```
# ldapmodify -h server.idm.example.com -p 389 -Y GSSAPI
SASL/GSSAPI authentication started
SASL username: provisionator@IDM.EXAMPLE.COM
SASL SSF: 56
SASL data security layer installed.
```

4. 추가하려는 사용자의 **dn** 을 입력합니다.

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

5. 수행 중인 변경 유형으로 **add** 를 입력합니다.

```
changetype: add
```

6. 사용자 라이프 사이클의 올바른 처리를 허용하는 데 필요한 **LDAP** 오브젝트 클래스 범주를 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. 사용자의 성을 입력합니다.

```
sn: Jensen
```

10. **Enter** 를 다시 눌러 이것이 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

stage 항목의 콘텐츠를 확인하여 프로비저닝 시스템에서 필요한 모든 **POSIX** 특성을 추가하고 **stage** 항목을 활성화할 준비가 되었는지 확인합니다.

- 새 **stage** 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** 특성에 의해 명시적으로 비활성화됩니다.

35장. LDAPMODIFY를 사용하여 IDM 사용자 외부 관리

ldapmodify 및 **ldap delete** 유틸리티를 사용하여 명령줄 인터페이스(CLI)에서 직접 **IdM(Identity Management) LDAP**를 수정할 수 있습니다. 유틸리티는 디렉터리 내용을 추가, 편집 및 삭제하는 완전한 기능을 제공합니다. 이러한 유틸리티를 사용하여 서버의 구성 항목과 사용자 항목의 데이터를 모두 관리할 수 있습니다. 유틸리티를 사용하여 하나 이상의 디렉터를 대량으로 관리하는 스크립트를 작성할 수도 있습니다.

35.1. 외부에서 IDM 사용자 계정을 관리하는 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 단계 사용자 추가
- 사용자 속성 수정
- 사용자 활성화
- 사용자 비활성화
- 사용자 보존

템플릿은 **LDAP(LDIF)**로 포맷됩니다. **LDIF**는 **LDAP** 디렉터리 콘텐츠를 표시하고 요청을 업데이트하기 위한 표준 일반 텍스트 데이터 교환 형식입니다.

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

자세한 예제 절차는 다음 섹션을 참조하십시오.

- **LDIF** 파일에 정의된 **IdM** 단계 사용자 추가

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

새 스테이징 사용자를 추가하기 위한 템플릿

- **UID 및 GID가 자동으로 할당된 사용자를 추가하는 템플릿. 생성된 항목의 고유 이름(DN)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
```

- **정적으로 UID 및 GID가 할당된 사용자를 추가하는 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 사용자 수정을 위한 템플릿

- **사용자의 속성 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
```

nsAccountLock 특성을 업데이트해도 단계 및 보존된 사용자에게는 영향을 미치지 않습니다. 업데이트 작업이 성공적으로 완료되었지만 속성 값은 **nsAccountLock**으로 유지됩니다. **TRUE**.

- 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 사용자의 로그인을 통해 검색하여 사용자의 고유 이름(DN)을 획득합니다. 다음 예에서 ***user_allowed_to_modify_user_entries*** 사용자는 활성화 기 또는 **IdM** 관리자와 같이 사용자 및 그룹 정보를 수정할 수 있는 사용자입니다. 예제의 암호는 이 사용자의 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
```

35.2. 외부에서 IDМ 그룹 계정을 관리하는 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 생성
- 기존 그룹 삭제
- 그룹에 구성원 추가
- 그룹에서 멤버 제거

템플릿은 **LDAP(LDIF)**로 포맷됩니다. **LDIF**는 **LDAP** 디렉터리 콘텐츠를 표시하고 요청을 업데이트하기 위한 표준 일반 텍스트 데이터 교환 형식입니다.

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

새 그룹 생성


```
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 멤버 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 구성원으로 업데이트되지 않습니다. 활성 사용자만 그룹에 속할 수 있습니다.

- 그룹에서 멤버 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 그룹 이름을 사용하여 검색하여 그룹의 고유 이름(DN)을 가져옵니다.

```
# ldapsearch -YGSSAPI -H ldap://server.idm.example.com -b
"cn=groups,cn=accounts,dc=idm,dc=example,dc=com" "cn=group_name"
dn: cn=group_name,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
ipaNTSecurityIdentifier: S-1-5-21-1650388524-2605035987-2578146103-11017
cn: testgroup
objectClass: top
objectClass: groupofnames
objectClass: nestedgroup
objectClass: ipausergroup
objectClass: ipaobject
objectClass: posixgroup
objectClass: ipantgroupattrs
ipaUniqueID: 569bf864-9d45-11ea-bea3-525400f6f085
gidNumber: 1997010017
```

35.3. LDAPMODIFY를 사용하여 IDM 사용자 보존

이 섹션에서는 **ldapmodify** 를 사용하여 **IdM** 사용자를 보존하는 방법에 대해 설명합니다. 즉, 직원이 퇴사한 후 사용자 계정을 비활성화하는 방법을 설명합니다.

사전 요구 사항

- 사용자를 보존하기 위해 역할로 **IdM** 사용자로 인증할 수 있습니다.

절차

1. 사용자를 보존하려면 역할을 가진 **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. 보존할 사용자의 **dn** 을 입력합니다.

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

4. 수행할 변경 유형으로 **modrdn** 을 입력합니다.

```
changetype: modrdn
```

5. 사용자에게 대한 **newrdn** 을 지정합니다.

```
newrdn: uid=user1
```

6. 사용자를 보존할 것을 나타냅니다.

```
deleteoldrdn: 0
```

7. 새로운 우수 **DN** 을 지정합니다.

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

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

8. **Enter** 를 다시 눌러 이것이 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
-----
```

36장. IDM CLI에서 호스트 관리

이 장에서는 **IdM(Identity Management)**의 [호스트](#) 및 [호스트 항목](#) 과 **IdM CLI**에서 호스트 및 호스트 항목을 관리할 때 수행되는 다음 작업을 소개합니다.

- [호스트 등록](#)
- [IdM 호스트 항목 추가](#)
- [IdM 호스트 항목 삭제](#)
- [호스트 재등록](#)
- [호스트 이름 변경](#)
- [호스트 비활성화](#)
- [호스트 재활성화](#)

또한 장에는 사전 요구 사항, 컨텍스트 및 이러한 작업 결과에 대한 [개요 테이블](#)이 포함되어 있습니다.

36.1. IDM의 호스트

IdM(Identity Management)은 이러한 ID를 관리합니다.

- 사용자
- 서비스
- 호스트

호스트는 시스템을 나타냅니다. **IdM ID**로 호스트에 **IdM LDAP**에 항목이 있으며, 이는 **IdM** 서버의 디렉터리 서버 인스턴스 **389**입니다.

IdM LDAP의 **host** 항목은 다른 호스트와 도메인 내의 서비스 간 관계를 설정하는 데 사용됩니다. 이러한 관계는 도메인 내의 호스트에 권한 부여 및 제어 *위임*의 일부입니다. 호스트 기반 액세스 제어(**HBAC**) 규칙에서 모든 호스트를 사용할 수 있습니다.

IdM 도메인은 공통 **ID** 정보, 공통 정책 및 공유 서비스가 포함된 시스템 간 공통성을 설정합니다. 도메인 기능에 속한 시스템은 도메인의 클라이언트에 속하므로, 도메인에서 제공하는 서비스를 사용합니다. **IdM** 도메인은 시스템 전용 세 가지 기본 서비스를 제공합니다.

- **DNS**
- **Kerberos**
- 인증서 관리

IdM의 호스트는 실행 중인 서비스와 긴밀하게 연결됩니다.

- 서비스 항목은 호스트와 연결됩니다.
- 호스트는 호스트와 서비스 **Kerberos** 사용자를 모두 저장합니다.

36.2. 호스트 등록

이 섹션에서는 **IdM** 클라이언트로 호스트 등록 및 등록 중 및 등록 후에 발생하는 사항에 대해 설명합니다. 이 섹션에서는 **IdM** 호스트 및 **IdM** 사용자의 등록을 비교합니다. 또한 이 섹션에서는 호스트에서 사용할 수 있는 대체 유형의 인증에 대해 간단히 설명합니다.

호스트 등록은 다음으로 구성됩니다.

- **IdM LDAP**에서 호스트 항목 생성: **IdM CLI**에서 **ipa host-add** 명령을 사용하거나 이에 상응하는 **IdM 웹 UI** 작업을 사용할 수 있습니다.

- 호스트에서 **IdM** 서비스 구성(예: **SSSD**(시스템 보안 서비스 데몬), **Kerberos**, **certmonger**) 호스트를 **IdM** 도메인에 연결합니다.

두 작업은 별도로 또는 함께 수행할 수 있습니다.

개별적으로 수행되는 경우 두 가지 작업을 서로 다른 수준의 권한을 가진 두 사용자 간에 나눌 수 있습니다. 이는 대량 배포에 유용합니다.

ipa-client-install 명령은 두 작업을 함께 수행할 수 있습니다. 해당 항목이 아직 없는 경우 명령은 **IdM LDAP**에 호스트 항목을 생성하고 호스트에 대한 **Kerberos** 및 **SSSD** 서비스를 둘 다 구성합니다. 이 명령을 사용하면 **IdM** 도메인 내에 호스트를 가져와 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.

36.2.1. 호스트 등록에 필요한 사용자 권한

호스트 등록 작업을 수행하려면 권한이 없는 사용자가 원하지 않는 시스템을 **IdM** 도메인에 추가하지 못하도록 인증해야 합니다. 필요한 권한은 다음과 같은 여러 요인에 따라 다릅니다.

- **ipa-client-install** 실행과 별도로 호스트 항목을 생성하는 경우
- 등록에 **OTP**(일회용 암호)를 사용하는 경우

IdM LDAP에서 수동으로 호스트 항목을 생성하는 데 필요한 사용자 권한

ipa host-add CLI 명령 또는 **IdM 웹 UI**를 사용하여 **IdM LDAP**에서 호스트 항목을 생성하는 데 필요한 사용자 권한은 호스트 관리자입니다. 호스트 관리자 권한은 **IT** 전문가 역할을 통해 얻을 수 있습니다.

클라이언트를 **IdM** 도메인에 가입하는 사용자 권한

호스트는 **ipa-client-install** 명령을 실행하는 동안 **IdM** 클라이언트로 구성됩니다. **ipa-client-install** 명령을 실행하는 데 필요한 인증 정보의 수준은 다음 중 본인이 찾은 시나리오에 따라 다릅니다.

- **IdM LDAP**의 호스트 항목이 존재하지 않습니다. 이 시나리오의 경우 전체 관리자 자격 증명 또는 **Host Administrators** 역할이 필요합니다. 전체 관리자는 **admins** 그룹의 구성원입니다. **Host Administrators** 역할은 호스트를 추가하고 호스트를 등록할 수 있는 권한을 제공합니다. 이

시나리오에 대한 자세한 내용은 [사용자 자격 증명: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오](#).

- **IdM LDAP의 host 항목이 있습니다.** 이 시나리오의 경우 **ipa-client-install** 을 실행하려면 제한된 관리자의 인증 정보가 필요합니다. 이 경우 제한된 관리자에게는 호스트 등록 권한을 제공하는 **Enrollment Administrator** 역할이 있습니다. 자세한 내용은 [사용자 credentials: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오](#).
- **IdM LDAP의 호스트 항목이 존재하며 전체 또는 제한된 관리자가 호스트에 대한 OTP가 생성되었습니다.** 이 시나리오에서는 **--password** 옵션을 사용하여 **ipa-client-install** 명령을 실행하여 올바른 OTP를 제공하는 경우 **IdM** 클라이언트를 일반 사용자로 설치할 수 있습니다. 자세한 내용은 [일회성 암호를 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치](#).

등록 후 **IdM** 호스트는 모든 새 세션을 인증하여 **IdM** 리소스에 액세스할 수 있도록 합니다. **IdM** 서버가 시스템을 신뢰하고 해당 시스템에 설치된 클라이언트 소프트웨어에서 **IdM** 연결을 수락하려면 시스템 인증이 필요합니다. 클라이언트를 인증한 후 **IdM** 서버는 요청에 응답할 수 있습니다.

36.2.2. IdM 호스트 및 사용자의 등록 및 인증: 비교

IdM의 사용자와 호스트 사이에는 많은 유사점이 있습니다. 이 섹션에서는 등록 단계에서 확인할 수 있는 몇 가지 유사점과 배포 단계에서 인증에 관한 몇 가지 유사 사항에 대해 설명합니다.

- 등록 단계([표 36.1. “사용자 및 호스트 등록”](#)):
 - 관리자는 사용자 또는 호스트 모두에 대해 **IdM**에 실제로 참여하기 전에 사용자 및 호스트에 대해 **LDAP** 항목을 생성할 수 있습니다. **stage** 사용자의 경우 명령은 **ipa stageuser-add** 입니다. 호스트에 대해 명령은 **ipa host-add** 입니다.
 - 키 테이블 또는 일부 익스텐트와 유사한 대칭 키인 **keytab**을 포함하는 파일은 호스트에서 **ipa-client-install** 명령 실행 중에 생성됩니다. 그러면 호스트가 **IdM** 영역에 결합됩니다. 비동기적으로 사용자는 계정을 활성화할 때 암호를 생성하여 **IdM** 영역에 가입해야 합니다.
 - 사용자 **password**는 사용자의 기본 인증 방법이지만 **keytab**은 호스트의 기본 인증 방법입니다. 키탭은 호스트의 파일에 저장됩니다.

표 36.1. 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] (호스트 자체에서 실행해야 합니다)

●

배포 단계(표 36.2. “사용자 및 호스트 세션 인증”):

○

사용자가 새 세션을 시작하면 사용자는 암호를 사용하여 인증합니다. 마찬가지로, 콘솔이 켜질 때마다 호스트는 해당 키탭 파일을 표시하여 인증합니다. **SSSD(System Security Services Daemon)**는 이 프로세스를 백그라운드에서 관리합니다.

○

인증에 성공하면 사용자 또는 호스트에서 티켓(TGT)을 부여하는 **Kerberos** 티켓을 가져옵니다.

○

그런 다음 TGT를 사용하여 특정 서비스의 특정 티켓을 받습니다.

표 36.2. 사용자 및 호스트 세션 인증

	사용자	호스트
기본 인증 수단	암호	키탭
세션 시작 (일반 사용자)	\$ kinit <i>user_name</i>	[호스트의 전환]
인증에 성공한 결과	특정 서비스에 대한 액세스를 얻는데 사용할 TGT	특정 서비스에 대한 액세스를 얻는데 사용할 TGT

TGT 및 기타 **Kerberos** 티켓은 서버에서 정의한 **Kerberos** 서비스 및 정책의 일부로 생성됩니다. **Kerberos** 티켓을 처음 부여하고, **Kerberos** 자격 증명을 갱신하며, **Kerberos** 세션 삭제도 **IdM** 서비스에 의해 자동으로 처리됩니다.

36.2.3. IdM 호스트의 대체 인증 옵션

keytab 외에도 **IdM**은 두 가지 유형의 시스템 인증을 지원합니다.

•

SSH 키. 호스트의 SSH 공개 키가 생성되어 호스트 항목에 업로드됩니다. 여기에서 SSSD(시스템 보안 서비스 데몬)는 IdM을 ID 프로바이더로 사용하며 OpenSSH 및 기타 서비스와 함께 작업하여 IdM에 있는 공용 키를 참조할 수 있습니다.

•

시스템 인증서. 이 경우 시스템은 IdM 서버의 인증 기관에서 발급한 SSL 인증서를 사용한 다음 IdM의 디렉터리 서버에 저장됩니다. 그러면 서버에 인증할 때 인증서가 존재하도록 시스템으로 전송됩니다. 클라이언트의 인증서는 [certmonger](#) 라는 서비스에서 관리합니다.

36.3. 호스트 작업

이 섹션에는 호스트 등록 및 활성화와 관련된 가장 일반적인 작업과 전제 조건, 컨텍스트 및 실행 결과에 대해 설명합니다.

표 36.3. 호스트 운영 파트 1

동작	작업의 전제 조건은 무엇입니까?	명령을 실행하는 것은 언제 적합합니까?	시스템 관리자가 작업을 어떻게 수행합니까? 실행 중인 명령은 무엇입니까?
고객 등록	<i>Installing Identity Management</i> 에서 Identity Management 클라이언트 설치 준비 를 참조하십시오.	호스트가 IdM 영역에 가입하려는 경우.	IdM 도메인에서 클라이언트로 시스템을 등록하는 작업은 2부로 구성된 프로세스입니다. ipa host-add 명령이 실행되는 경우 클라이언트에 대한 호스트 항목이 생성되고(389 Directory Server 인스턴스에 저장되고) 클라이언트를 프로비저닝하기 위해 keytab이 생성됩니다. 두 부분 모두 ipa-client-install 명령으로 자동으로 수행됩니다. 이러한 단계를 별도로 수행할 수도 있습니다. 이를 통해 관리자는 실제로 클라이언트를 구성하기 전에 시스템과 IdM을 준비할 수 있습니다. 따라서 대규모 배포를 비롯한 보다 유연한 설정 시나리오가 가능합니다.
클라이언트 비활성화	호스트에 IdM에 항목이 있어야 합니다. 호스트에 활성 키탭이 있어야 합니다.	유지 관리를 위해 IdM 영역에서 임시로 호스트를 제거하려는 경우.	ipa host-disable host_name
클라이언트 활성화	호스트에 IdM에 항목이 있어야 합니다.	일시적으로 비활성화된 호스트를 다시 활성화하려면 다음을 수행합니다.	ipa-getkeytab

동작	작업의 전체 조건은 무엇입니까?	명령을 실행하는 것은 언제 적합합니까?	시스템 관리자가 작업을 어떻게 수행합니까? 실행 중인 명령은 무엇입니까?
클라이언트 재등록	호스트에는 IdM에 en 항목이 있어야 합니다.	원래 호스트가 손실되었지만 동일한 호스트 이름을 가진 호스트를 설치한 경우.	ipa-client-install --keytab 또는 ipa-client-install --force-join
클라이언트 등록 해제	호스트에 IdM에 항목이 있어야 합니다.	IdM 영역에서 호스트를 영구적으로 제거하려는 경우.	ipa-client-install --uninstall

표 36.4. 호스트 작업 2부

동작	관리자가 어떤 시스템에서 명령을 실행할 수 있습니까?	작업이 수행되면 어떻게 됩니까? IdM에서 호스트의 기능에 대한 결과는 무엇입니까? 도입/제거된 제한 사항은 무엇입니까?
고객 등록	2단계 등록의 경우: ipa host-add 는 IdM 클라이언트에서 실행될 수 있습니다. ipa-client-install 의 두 번째 단계는 클라이언트 자체에서 실행해야 합니다.	기본적으로 인증 및 권한 부여를 위해 IdM 서버에 연결하도록 SSSD를 구성합니다. 선택적으로 Kerberos 및 LDAP를 통해 IdM 서버에서 작동하도록 PAM(Pluggable Authentication Module) 및 NSS(Name Switching Service)를 구성할 수 있습니다.
클라이언트 비활성화	IdM의 모든 시스템, 호스트 자체	호스트의 Kerberos 키와 SSL 인증서가 무효화되고 호스트에서 실행 중인 모든 서비스가 비활성화됩니다.
클라이언트 활성화	IdM의 모든 시스템. 비활성화 호스트에서 실행되는 경우 LDAP 자격 증명을 제공해야 합니다.	호스트의 Kerberos 키와 SSL 인증서가 다시 유효해지며 호스트에서 실행되는 모든 IdM 서비스가 다시 활성화됩니다.
클라이언트 재등록	다시 등록할 호스트입니다. LDAP 자격 증명을 제공해야 합니다.	호스트에 대해 새 Kerberos 키가 생성되어 이전 키가 교체됩니다.

동작	관리자가 어떤 시스템에서 명령을 실행할 수 있습니까?	작업이 수행되면 어떻게 됩니까? IdM에서 호스트의 기능에 대한 결과는 무엇입니까? 도입/제거된 제한 사항은 무엇입니까?
클라이언트 등록 해제	등록 취소할 호스트입니다.	명령은 IdM을 구성 해제하고 시스템을 이전 상태로 되돌리려고 시도합니다. 이 프로세스의 일부는 IdM 서버에서 호스트 등록을 취소하는 것입니다. Unenrollment은 IdM 서버에서 키 키를 비활성화하도록 구성됩니다. <code>/etc/krb5.keytab</code> 의 시스템 주체 (<code>host/<fqdn>@REALM</code>)는 IdM 서버에 인증하여 등록 취소합니다. 이 주체가 없으면 등록 취소가 실패하고 관리자가 호스트 주체 (<code>ipahost-disable <fqdn></code>)를 비활성화해야 합니다.

36.4. IDM LDAP의 호스트 항목

이 섹션에서는 IdM(Identity Management)의 호스트 항목과 어떤 속성을 포함할 수 있는지에 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 시스템 정보



참고

IdM 웹 UI ID → 호스트 탭에는 IdM LDAP에 저장된 특정 호스트에 대한 모든 정보가 표시되지 않습니다.

36.4.1. 호스트 항목 구성 속성

호스트 항목은 실제 위치, MAC 주소, 키 및 인증서와 같이 시스템 구성 외부에 있는 호스트에 대한 정보를 포함할 수 있습니다.

이 정보는 수동으로 만드는 경우 호스트 항목을 생성할 때 설정할 수 있습니다. 또는 호스트가 도메인에 등록된 후 대부분의 이 정보를 **host** 항목에 추가할 수 있습니다.

표 36.5. 호스트 구성 속성

UI 필드	명령줄 옵션	설명
설명	--desc = <i>description</i>	호스트에 대한 설명입니다.
지역	--locality = <i>locality</i>	호스트의 지리적 위치.
위치	--location = <i>location</i>	호스트의 실제 위치(예: 데이터 센터 랙).
플랫폼	--platform = <i>string</i>	호스트 하드웨어 또는 아키텍처.
운영 체제	--os = <i>string</i>	호스트의 운영 체제 및 버전.
MAC 주소	--macaddress = <i>address</i>	호스트의 MAC 주소입니다. 이것은 다중 값 속성입니다. MAC 주소는 NIS 플러그인에서 호스트에 대한 NIS ethers 맵을 만드는 데 사용됩니다.
SSH 공개 키	--sshpkey = <i>string</i>	호스트에 대한 전체 SSH 공개 키. 이것은 다중 값 속성이므로 여러 키를 설정할 수 있습니다.
주체 이름 (편집할 수 없음)	--principalname = <i>principal</i>	호스트의 Kerberos 주체 이름입니다. 다른 주체가 -p 에 명시적으로 설정되지 않는 한 클라이언트 설치 중에 호스트 이름이 기본값입니다. 이 설정은 명령줄 도구를 사용하여 변경할 수 있지만 UI에서는 변경할 수 없습니다.
일회성 암호 설정	--password = <i>string</i>	이 옵션은 대규모 등록에 사용할 수 있는 호스트의 암호를 설정합니다.
-	--random	이 옵션은 대량 등록에 사용할 임의 암호를 생성합니다.
-	--certificate = <i>string</i>	호스트에 대한 인증서 Blob.
-	--updatedns	이렇게 하면 IP 주소가 변경되면 호스트가 DNS 항목을 동적으로 업데이트할 수 있는지 여부를 설정합니다.

36.5. 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 생성합니다. 이 명령은 IdM 디렉터리 서버에 호스트 항목을 추가합니다. CLI에 **ipa help host** 를 입력하여 **host-add**에 사용할 수 있는 전체 옵션 목록을 가져와 **ipa host manpage**를 참조합니다.

IdM에 호스트를 추가할 때 몇 가지 시나리오가 있습니다.

- 가장 기본적인 경우 클라이언트 호스트 이름만 지정하여 **Kerberos** 영역에 클라이언트를 추가하고 **IdM LDAP** 서버에서 항목을 생성합니다.

```
$ ipa host-add client1.example.com
```

- IdM 서버가 **DNS**를 관리하도록 구성된 경우 **--ip-address** 옵션을 사용하여 호스트를 **DNS** 리소스 레코드에 추가합니다.

예 36.1. 정적 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 구성될 때 **IP** 주소를 알 수 없는 경우 **ipa host-add** 명령과 함께 **--force** 옵션을 사용합니다.

예 36.2. DHCP를 사용하여 호스트 항목 생성

```
$ ipa host-add --force client1.example.com
```

예를 들어 랩탑은 IdM 클라이언트로 사전 구성될 수 있지만 구성 시 **IP** 주소는 없습니다. **--force** 를 사용하면 기본적으로 IdM **DNS** 서비스에 자리 표시자 항목이 생성됩니다. **DNS** 서비스에서 레코드를 동적으로 업데이트하면 호스트의 현재 **IP** 주소가 탐지되고 해당 **DNS** 레코드가 업데이트됩니다.

36.6. IDM CLI에서 호스트 항목 삭제

- host-del** 명령을 사용하여 호스트 레코드를 삭제합니다. IdM 도메인에 통합된 **DNS**가 있는 경

우 **--updatedns** 옵션을 사용하여 DNS에서 호스트에 대한 모든 종류의 관련 레코드를 제거합니다.

```
$ ipa host-del --updatedns client1.example.com
```

36.7. ID 관리 클라이언트 재등록

36.7.1. IdM의 클라이언트 등록

이 섹션에서는 **IdM(Identity Management)** 클라이언트를 다시 등록하는 방법을 설명합니다.

클라이언트 시스템이 삭제되고 **IdM** 서버와 연결이 끊어진 경우(예: 클라이언트의 하드웨어 실패로 인해) 키맵이 있는 경우 클라이언트를 다시 등록할 수 있습니다. 이 시나리오에서는 동일한 호스트 이름이 있는 **IdM** 환경에서 클라이언트를 다시 가져오려고 합니다.

다시 등록하는 동안 클라이언트는 새 **Kerberos** 키와 **SSH** 키를 생성하지만 **LDAP** 데이터베이스의 클라이언트 **ID**는 변경되지 않습니다. 다시 등록한 후 호스트에 **IdM** 서버와의 연결이 손실되기 전에 이전과 동일한 **FQDN** 을 사용하는 동일한 **LDAP** 오브젝트에 키 및 기타 정보가 있습니다.

중요

도메인 항목이 아직 활성 상태인 클라이언트만 다시 등록할 수 있습니다. 클라이언트를 설치 제거하거나(**ipa -client-install --uninstall**사용) 호스트 항목(**ipa host-disable**사용)을 비활성화한 경우 다시 등록할 수 없습니다.

이름을 변경한 후에는 클라이언트를 다시 등록할 수 없습니다. 이는 **ID** 관리에서 **LDAP**에서 클라이언트 항목의 키 특성이 클라이언트의 호스트 이름인 해당 **FQDN** 이기 때문입니다. 클라이언트의 **LDAP** 오브젝트가 변경되지 않은 상태로 남아 있는 클라이언트를 다시 등록하는 것과 달리, 클라이언트의 이름을 변경한 결과는 새로운 **FQDN** 을 사용하여 다른 **LDAP** 오브젝트에 키와 기타 정보가 있다는 것입니다. 따라서 클라이언트 이름을 변경할 수 있는 유일한 방법은 **IdM**에서 호스트를 제거하고 호스트의 호스트 이름을 변경한 후 새 이름으로 **IdM** 클라이언트로 설치하는 것입니다. 클라이언트 이름 변경 방법에 대한 자세한 내용은 [36.8절. “Identity Management 클라이언트 시스템 이름 변경”](#) 을 참조하십시오.

36.7.1.1. 클라이언트를 다시 등록하는 동안 발생하는 사항

다시 등록하는 동안 **ID** 관리:

- 원래 호스트 인증서 취소

- 새 SSH 키 생성
- 새 키맵 생성

36.7.2. 사용자 인증 정보를 사용하여 클라이언트 재등록: 대화형 등록

이 절차에서는 권한 있는 사용자의 자격 증명을 사용하여 대화식으로 ID 관리 클라이언트를 다시 등록하는 방법을 설명합니다.

1. 동일한 호스트 이름으로 클라이언트 시스템을 다시 생성합니다.
2. 클라이언트 시스템에서 **ipa-client-install --force-join** 명령을 실행합니다.

```
# ipa-client-install --force-join
```

3. 이 스크립트는 클라이언트를 다시 등록하는 데 ID를 사용하는 사용자를 묻습니다. 예를 들어 Enrollment Administrator 역할이 있는 **hostadmin** 사용자는 다음과 같습니다.

```
User authorized to enroll computers: hostadmin
Password for hostadmin@EXAMPLE.COM:
```

추가 리소스

- 인증된 사용자의 인증 정보를 사용하여 클라이언트 등록에 대한 자세한 절차는 [사용자 인증 정보를 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치 ID 관리 설치 중.](#)

36.7.3. 클라이언트 키맵을 사용하여 클라이언트를 다시 등록합니다. 비대화형 재등록

사전 요구 사항

- 원래 클라이언트 키맵 파일을 백업합니다(예: **/tmp** 또는 **/root** 디렉토리).

절차

이 절차에서는 클라이언트 시스템의 키맵을 사용하여 IdM(Identity Management) 클라이언트를 비대화식으로 등록하는 방법을 설명합니다. 예를 들어 클라이언트 **keytab**을 사용하여 다시 등록하는 것은 자

동화된 설치에 적합합니다.

1. 동일한 호스트 이름으로 클라이언트 시스템을 다시 생성합니다.
2. 백업 위치의 **keytab** 파일을 다시 생성된 클라이언트 시스템의 **/etc/** 디렉토리로 복사합니다.
3. **ipa-client-install** 유틸리티를 사용하여 클라이언트를 다시 등록하고 **--keytab** 옵션으로 **keytab** 위치를 지정합니다.

```
# ipa-client-install --keytab /etc/krb5.keytab
```



참고

keytab 옵션에 지정된 **key tab** 은 인증을 시작하여 등록을 시작할 때만 사용됩니다. 다시 등록하는 동안 **IdM**은 클라이언트에 대한 새 키탭을 생성합니다.

36.7.4. 설치 후 Identity Management 클라이언트 테스트

명령줄 인터페이스는 **ipa-client-install** 이 성공적으로 수행되었지만 자체 테스트를 수행할 수도 있음을 알려줍니다.

ID 관리 클라이언트가 서버에 정의된 사용자에게 대한 정보를 가져올 수 있는지 테스트하려면 서버에 정의된 사용자를 확인할 수 있는지 확인합니다. 예를 들어 기본 **admin** 사용자를 확인하려면 다음을 수행합니다.

```
[user@client1 ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

인증이 올바르게 작동하는지 테스트하려면 **su -** 를 다른 **IdM** 사용자로 실행하십시오.

```
[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 ~]$
```

36.8. IDENTITY MANAGEMENT 클라이언트 시스템 이름 변경

다음 섹션에서는 ID 관리 클라이언트 시스템의 호스트 이름을 변경하는 방법을 설명합니다.



주의

클라이언트 이름 바꾸기는 수동 절차입니다. 호스트 이름을 변경할 필요가 없는 한 수행하지 마십시오.

Identity Management 클라이언트의 이름 변경에는 다음이 포함됩니다.

1. 호스트 준비. 자세한 내용은 변경에 **필요한 IdM 클라이언트 준비**를 참조하십시오.
2. 호스트에서 **IdM 클라이언트 설치 제거**. 자세한 내용은 [36.8.2절. “ID 관리 클라이언트 설치 제거”](#)의 내용을 참조하십시오.
3. 호스트 이름 바꾸기. 자세한 내용은 [36.8.3절. “호스트 시스템 이름 변경”](#)의 내용을 참조하십시오.
4. 호스트에 새 이름으로 **IdM 클라이언트 설치**. 자세한 내용은 [36.8.4절. “ID 관리 클라이언트 재 설치”](#)의 내용을 참조하십시오.
5. **IdM 클라이언트 설치 후 호스트 구성**. 자세한 내용은 [36.8.5절. “서비스 다시 추가, 인증서 재생, 호스트 그룹 다시 추가”](#)의 내용을 참조하십시오.

36.8.1. 변경 작업을 위해 IdM 클라이언트 준비

현재 클라이언트를 제거하기 전에 클라이언트의 특정 설정을 기록해 두십시오. 새 호스트 이름으로 시스템을 다시 등록한 후 이 구성을 적용합니다.

- 시스템에서 실행 중인 서비스를 확인합니다.

○

ipa service-find 명령을 사용하고 출력에서 인증서가 있는 서비스를 식별합니다.

```
$ ipa service-find old-client-name.example.com
```

○

또한 각 호스트에는 **ipa service-find** 출력에 표시되지 않는 기본 호스트 서비스가 있습니다. **host** 서비스의 서비스 주체(host principal라고도 함)는 **host/old-client-name.example.com** 입니다.

●

ipa service-find old-client-name.example.com 에서 표시되는 모든 서비스 주체에 대해 **old-client-name.example.com** 시스템에서 해당 키 탭의 위치를 확인합니다.

```
# find / -name "*.keytab"
```

클라이언트 시스템의 각 서비스에는 **ldap/old-client-name.example.com@EXAMPLE.COM** 과 같은 **service_name/host_name@REALM** 형식의 Kerberos 주체가 있습니다.

●

시스템이 속한 모든 호스트 그룹을 식별합니다.

```
# ipa hostgroup-find old-client-name.example.com
```

36.8.2. ID 관리 클라이언트 설치 제거

클라이언트를 설치 제거하면 ID 관리 도메인에서 클라이언트가 제거되며 **SSSD(System Security Services Daemon)**와 같은 시스템 서비스의 모든 특정 ID 관리 구성도 제거됩니다. 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 식별된 각 키탭에 대해 이전 주체를 제거하십시오.

```
[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
```

36.8.3. 호스트 시스템 이름 변경

필요에 따라 시스템의 이름을 변경합니다. 예를 들면 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.

36.8.4. ID 관리 클라이언트 재설치

ID 관리 클라이언트 설치에 설명된 절차에 따라 이름이 변경된 호스트에 클라이언트를 설치합니다. [기본 시나리오 ID 관리 설치](#) 시.

36.8.5. 서비스 다시 추가, 인증서 재생, 호스트 그룹 다시 추가

1.

IdM(Identity Management) 서버에서 **IdM 클라이언트의 이름 변경**을 위해 식별된 모든 서비스에 대해 새 키 탭을 추가합니다.

```
[root@server ~]# ipa service-add service_name/new-client-name
```

2.

교체를 위해 **IdM 클라이언트 준비**에 할당된 인증서가 할당된 서비스에 대한 인증서를 생성합니다. 다음을 수행할 수 있습니다.

- **IdM 관리 도구 사용**
- **certmonger 유틸리티 사용**

3.

교체를 위해 **IdM 클라이언트 준비에서 식별된 호스트 그룹에 클라이언트를 다시 추가합니다.**

36.9. 호스트 항목 비활성화 및 재활성화

이 섹션에서는 **IdM(Identity Management)**에서 호스트를 비활성화하고 다시 활성화하는 방법에 대해 설명합니다.

36.9.1. 호스트 비활성화

IdM에서 호스트 항목을 비활성화하려면 다음 절차를 완료합니다.

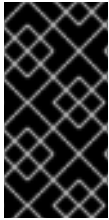
도메인 서비스, 호스트 및 사용자가 활성 호스트에 액세스할 수 있습니다. 유지 관리상의 이유로 활성 호스트를 일시적으로 제거해야 하는 상황이 있을 수 있습니다. 예를 들면 다음과 같습니다. 이러한 상황에서 호스트를 삭제하는 것은 호스트 항목 및 연결된 모든 구성을 영구적으로 제거하므로 바람직하지 않습니다. 대신 호스트를 비활성화하는 옵션을 선택합니다.

호스트를 비활성화하면 도메인 사용자가 도메인에서 영구적으로 제거하지 않고 액세스할 수 없습니다. 이 작업은 **host-disable** 명령을 사용하여 수행할 수 있습니다. 호스트를 비활성화하면 현재 활성 키가 종료됩니다.

예를 들면 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 호스트가 비활성화될 뿐만 아니라, 해당 호스트에서 구성된 모든 서비스도 비활성화합니다.

36.9.2. 호스트 재활성화

이 섹션에서는 비활성화된 **IdM** 호스트를 다시 활성화하는 방법에 대해 설명합니다.

호스트를 비활성화하면 활성 키맵이 종료되어 구성 항목을 사용하지 않고 **IdM** 도메인에서 호스트가 제거되었습니다.

호스트를 다시 활성화하려면 **ipa-getkeytab** 명령을 사용하여 다음을 추가합니다.

- 키맵을 요청할 **IdM** 서버를 지정하는 **-s** 옵션
- 주체 이름을 지정하는 **-p** 옵션
- **keytab**을 저장할 파일을 지정하는 **-k** 옵션.

예를 들어 **client.example.com**에 대해 **server.example.com** 에서 새 호스트 키맵을 요청하고 **/etc/krb5.keytab** 파일에 키맵을 저장하려면 다음을 수행합니다.

```
$ ipa-getkeytab -s server.example.com -p host/client.example.com -k /etc/krb5.keytab -D
"cn=directory manager" -w password
```



참고

관리자 자격 증명을 사용하여 **-D**
"uid=admin,cn=users,cn=accounts,dc=example,dc=com" 을 지정할 수도 있습니다.
 자격 증명은 호스트에 대한 키맵을 만들 수 있는 사용자에게 대응하는 것이 중요합니다.

ipa-getkeytab 명령이 활성 **IdM** 클라이언트 또는 서버에서 실행되면 사용자에게 **TGT**를 사용하여 얻은 경우 **LDAP** 자격 증명(예: **kinit admin**) 없이 실행할 수 있습니다. 비활성화된 호스트에서 직접 명령을 실행하려면 **IdM** 서버에 인증할 **LDAP** 자격 증명을 제공합니다.

37장. IDM 웹 UI에서 호스트 항목 추가

이 장에서는 **IdM(Identity Management)**의 호스트와 **IdM 웹 UI**에 호스트 항목을 추가하는 작업을 소개합니다.

37.1. IDM의 호스트

IdM(Identity Management)은 이러한 **ID**를 관리합니다.

- 사용자
- 서비스
- 호스트

호스트는 시스템을 나타냅니다. **IdM ID**로 호스트에 **IdM LDAP**에 항목이 있으며, 이는 **IdM** 서버의 디렉터리 서버 인스턴스 **389**입니다.

IdM LDAP의 **host** 항목은 다른 호스트와 도메인 내의 서비스 간 관계를 설정하는 데 사용됩니다. 이러한 관계는 도메인 내의 호스트에 권한 부여 및 제어 *위임*의 일부입니다. 호스트 기반 액세스 제어(**HBAC**) 규칙에서 모든 호스트를 사용할 수 있습니다.

IdM 도메인은 공통 **ID** 정보, 공통 정책 및 공유 서비스가 포함된 시스템 간 공통성을 설정합니다. 도메인 기능에 속한 시스템은 도메인의 클라이언트에 속하므로, 도메인에서 제공하는 서비스를 사용합니다. **IdM** 도메인은 시스템 전용 세 가지 기본 서비스를 제공합니다.

- **DNS**
- **Kerberos**
- 인증서 관리

IdM의 호스트는 실행 중인 서비스와 긴밀하게 연결됩니다.

- 서비스 항목은 호스트와 연결됩니다.
- 호스트는 호스트와 서비스 **Kerberos** 사용자를 모두 저장합니다.

37.2. 호스트 등록

이 섹션에서는 IdM 클라이언트로 호스트 등록 및 등록 중 및 등록 후에 발생하는 사항에 대해 설명합니다. 이 섹션에서는 IdM 호스트 및 IdM 사용자의 등록을 비교합니다. 또한 이 섹션에서는 호스트에서 사용할 수 있는 대체 유형의 인증에 대해 간단히 설명합니다.

호스트 등록은 다음으로 구성됩니다.

- IdM LDAP에서 호스트 항목 생성: IdM CLI에서 **ipa host-add** 명령을 사용하거나 이에 상응하는 IdM 웹 UI 작업을 사용할 수 있습니다.
- 호스트에서 IdM 서비스 구성(예: SSSD(시스템 보안 서비스 데몬), Kerberos, certmonger) 호스트를 IdM 도메인에 연결합니다.

두 작업은 별도로 또는 함께 수행할 수 있습니다.

개별적으로 수행되는 경우 두 가지 작업을 서로 다른 수준의 권한을 가진 두 사용자 간에 나눌 수 있습니다. 이는 대량 배포에 유용합니다.

ipa-client-install 명령은 두 작업을 함께 수행할 수 있습니다. 해당 항목이 아직 없는 경우 명령은 IdM LDAP에 호스트 항목을 생성하고 호스트에 대한 Kerberos 및 SSSD 서비스를 둘 다 구성합니다. 이 명령을 사용하면 IdM 도메인 내에 호스트를 가져와 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.

37.2.1. 호스트 등록에 필요한 사용자 권한

호스트 등록 작업을 수행하려면 권한이 없는 사용자가 원하지 않는 시스템을 IdM 도메인에 추가하지 못하도록 인증해야 합니다. 필요한 권한은 다음과 같은 여러 요인에 따라 다릅니다.

- **ipa-client-install** 실행과 별도로 호스트 항목을 생성하는 경우
- 등록에 **OTP**(일회용 암호)를 사용하는 경우

IdM LDAP에서 수동으로 호스트 항목을 생성하는 데 필요한 사용자 권한

ipa host-add CLI 명령 또는 **IdM 웹 UI**를 사용하여 **IdM LDAP**에서 호스트 항목을 생성하는 데 필요한 사용자 권한은 호스트 관리자입니다. 호스트 관리자 권한은 IT 전문가 역할을 통해 얻을 수 있습니다.

클라이언트를 **IdM** 도메인에 가입하는 사용자 권한

호스트는 **ipa-client-install** 명령을 실행하는 동안 **IdM** 클라이언트로 구성됩니다. **ipa-client-install** 명령을 실행하는 데 필요한 인증 정보의 수준은 다음 중 본인이 찾은 시나리오에 따라 다릅니다.

- **IdM LDAP**의 호스트 항목이 존재하지 않습니다. 이 시나리오의 경우 전체 관리자 자격 증명 또는 **Host Administrators** 역할이 필요합니다. 전체 관리자는 **admins** 그룹의 구성원입니다. **Host Administrators** 역할은 호스트를 추가하고 호스트를 등록할 수 있는 권한을 제공합니다. 이 시나리오에 대한 자세한 내용은 [사용자 자격 증명: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오](#).
- **IdM LDAP**의 **host** 항목이 있습니다. 이 시나리오의 경우 **ipa-client-install** 을 실행하려면 제한된 관리자의 인증 정보가 필요합니다. 이 경우 제한된 관리자에게는 호스트 등록 권한을 제공하는 **Enrollment Administrator** 역할이 있습니다. 자세한 내용은 [사용자 credentials: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오](#).
- **IdM LDAP**의 호스트 항목이 존재하며 전체 또는 제한된 관리자가 호스트에 대한 **OTP**가 생성되었습니다. 이 시나리오에서는 **--password** 옵션을 사용하여 **ipa-client-install** 명령을 실행하여 올바른 **OTP**를 제공하는 경우 **IdM** 클라이언트를 일반 사용자로 설치할 수 있습니다. 자세한 내용은 [일회성 암호를 사용하여 클라이언트 설치를 참조하십시오](#). [대화식 설치](#).

등록 후 **IdM** 호스트는 모든 새 세션을 인증하여 **IdM** 리소스에 액세스할 수 있도록 합니다. **IdM** 서버가 시스템을 신뢰하고 해당 시스템에 설치된 클라이언트 소프트웨어에서 **IdM** 연결을 수락하려면 시스템 인증이 필요합니다. 클라이언트를 인증한 후 **IdM** 서버는 요청에 응답할 수 있습니다.

37.2.2. IdM 호스트 및 사용자의 등록 및 인증: 비교

IdM의 사용자와 호스트 사이에는 많은 유사점이 있습니다. 이 섹션에서는 등록 단계에서 확인할 수 있는 몇 가지 유사점과 배포 단계에서 인증에 관한 몇 가지 유사 사항에 대해 설명합니다.

-

등록 단계(표 37.1. “사용자 및 호스트 등록”):

-

관리자는 사용자 또는 호스트 모두에 대해 IdM에 실제로 참여하기 전에 사용자 및 호스트에 대해 LDAP 항목을 생성할 수 있습니다. **stage** 사용자의 경우 명령은 **ipa stageuser-add** 입니다. 호스트에 대해 명령은 **ipa host-add** 입니다.

-

키 테이블 또는 일부 익스텐트와 유사한 대칭 키인 **keytab**을 포함하는 파일은 호스트에서 **ipa-client-install** 명령 실행 중에 생성됩니다. 그러면 호스트가 IdM 영역에 결합됩니다. 비동기적으로 사용자는 계정을 활성화할 때 암호를 생성하여 IdM 영역에 가입해야 합니다.

-

사용자 **password**는 사용자의 기본 인증 방법이지만 **keytab**은 호스트의 기본 인증 방법입니다. 키탭은 호스트의 파일에 저장됩니다.

표 37.1. 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] (호스트 자체에서 실행해야 합니다)

-

배포 단계(표 37.2. “사용자 및 호스트 세션 인증”):

-

사용자가 새 세션을 시작하면 사용자는 암호를 사용하여 인증합니다. 마찬가지로, 콘솔이 켜질 때마다 호스트는 해당 키탭 파일을 표시하여 인증합니다. **SSSD(System Security Services Daemon)**는 이 프로세스를 백그라운드에서 관리합니다.

-

인증에 성공하면 사용자 또는 호스트에서 티켓(TGT)을 부여하는 **Kerberos** 티켓을 가져옵니다.

-

그런 다음 **TGT**를 사용하여 특정 서비스의 특정 티켓을 받습니다.

표 37.2. 사용자 및 호스트 세션 인증

사용자		호스트
기본 인증 수단	암호	키탭
세션 시작 (일반 사용자)	\$ kinit <i>user_name</i>	[호스트의 전환]
인증에 성공한 결과	특정 서비스에 대한 액세스를 얻는데 사용할 TGT	특정 서비스에 대한 액세스를 얻는데 사용할 TGT

TGT 및 기타 Kerberos 티켓은 서버에서 정의한 Kerberos 서비스 및 정책의 일부로 생성됩니다. Kerberos 티켓을 처음 부여하고, Kerberos 자격 증명을 갱신하며, Kerberos 세션 삭제도 IdM 서비스에 의해 자동으로 처리됩니다.

37.2.3. IdM 호스트의 대체 인증 옵션

keytab 외에도 IdM은 두 가지 유형의 시스템 인증을 지원합니다.

- SSH 키. 호스트의 SSH 공개 키가 생성되어 호스트 항목에 업로드됩니다. 여기에서 SSSD(시스템 보안 서비스 데몬)는 IdM을 ID 프로바이더로 사용하며 OpenSSH 및 기타 서비스와 함께 작업하여 IdM에 있는 공용 키를 참조할 수 있습니다.
- 시스템 인증서. 이 경우 시스템은 IdM 서버의 인증 기관에서 발급한 SSL 인증서를 사용한 다음 IdM의 디렉터리 서버에 저장됩니다. 그러면 서버에 인증할 때 인증서가 존재하도록 시스템으로 전송됩니다. 클라이언트의 인증서는 [certmonger](#) 라는 서비스에서 관리합니다.

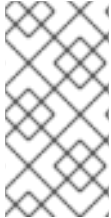
37.3. IDM LDAP의 호스트 항목

이 섹션에서는 IdM(Identity Management)의 호스트 항목과 어떤 속성을 포함할 수 있는지에 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 시스템 정보



참고

IdM 웹 UI ID → 호스트 탭에는 **IdM LDAP**에 저장된 특정 호스트에 대한 모든 정보가 표시되지 않습니다.

37.3.1. 호스트 항목 구성 속성

호스트 항목은 실제 위치, **MAC** 주소, 키 및 인증서와 같이 시스템 구성 외부에 있는 호스트에 대한 정보를 포함할 수 있습니다.

이 정보는 수동으로 만드는 경우 호스트 항목을 생성할 때 설정할 수 있습니다. 또는 호스트가 도메인에 등록된 후 대부분의 이 정보를 **host** 항목에 추가할 수 있습니다.

표 37.3. 호스트 구성 속성

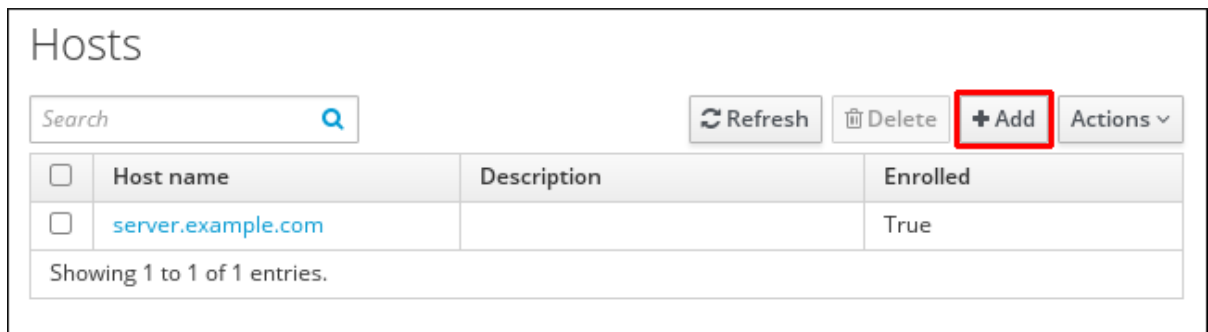
UI 필드	명령줄 옵션	설명
설명	--desc = <i>description</i>	호스트에 대한 설명입니다.
지역	--locality = <i>locality</i>	호스트의 지리적 위치.
위치	--location = <i>location</i>	호스트의 실제 위치(예: 데이터 센터 랙).
플랫폼	--platform = <i>string</i>	호스트 하드웨어 또는 아키텍처.
운영 체제	--os = <i>string</i>	호스트의 운영 체제 및 버전.
MAC 주소	--macaddress = <i>address</i>	호스트의 MAC 주소입니다. 이것은 다중 값 속성입니다. MAC 주소는 NIS 플러그인에서 호스트에 대한 NISethers 맵을 만드는 데 사용됩니다.
SSH 공개 키	--sshpubkey = <i>string</i>	호스트에 대한 전체 SSH 공개 키. 이것은 다중 값 속성이므로 여러 키를 설정할 수 있습니다.

UI 필드	명령줄 옵션	설명
주체 이름 (편집할 수 없음)	--principalname=principal	호스트의 Kerberos 주체 이름입니다. 다른 주체가 -p: 에 명시적으로 설정되지 않는 한 클라이언트 설치 중에 호스트 이름이 기본값입니다. 이 설정은 명령줄 도구를 사용하여 변경할 수 있지만 UI에서는 변경할 수 없습니다.
일회성 암호 설정	--password=string	이 옵션은 대규모 등록에 사용할 수 있는 호스트의 암호를 설정합니다.
-	--random	이 옵션은 대량 등록에 사용할 임의 암호를 생성합니다.
-	--certificate=string	호스트에 대한 인증서 Blob.
-	--updatedns	이렇게 하면 IP 주소가 변경되면 호스트가 DNS 항목을 동적으로 업데이트할 수 있는지 여부를 설정합니다.

37.4. 웹 UI에서 호스트 항목 추가

1. **Identity(ID)** 탭을 열고 **Hosts(호스트)** 하위 탭을 선택합니다.
2. **hosts** 목록 상단에 있는 **Add(추가)**를 클릭합니다.

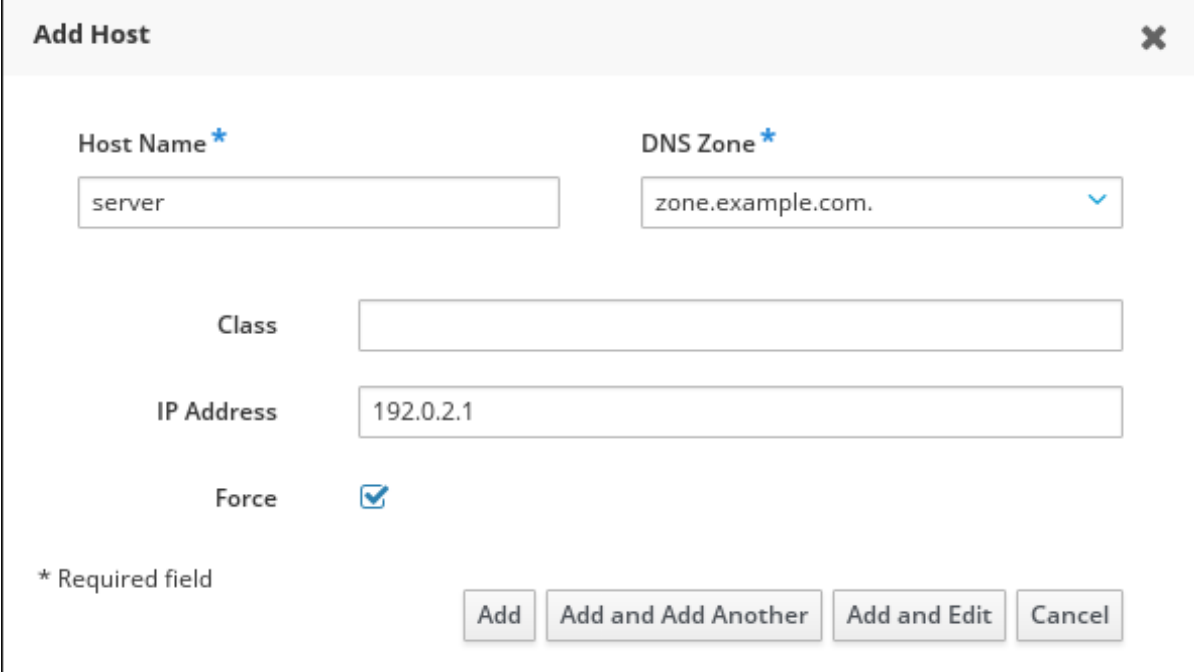
그림 37.1. 호스트 항목 추가



3. 시스템 이름을 입력하고 드롭다운 목록에서 구성된 영역에서 도메인을 선택합니다. 호스트에 이미 정적 IP 주소가 할당된 경우 **DNS** 항목이 완전히 생성되도록 **host** 항목으로 해당 주소를 포함합니다.

Class 필드는 현재 특별한 용도가 없습니다.

그림 37.2. 호스트 마법사 추가



The image shows a 'Add Host' dialog box with a close button (X) in the top right corner. It contains the following fields and controls:

- Host Name ***: A text input field containing 'server'.
- DNS Zone ***: A dropdown menu showing 'zone.example.com.' with a blue checkmark icon on the right.
- Class**: An empty text input field.
- IP Address**: A text input field containing '192.0.2.1'.
- Force**: A checkbox that is checked, with a blue checkmark icon to its right.
- * Required field**: A label indicating that fields with an asterisk are required.
- Buttons**: Four buttons at the bottom right: 'Add', 'Add and Add Another', 'Add and Edit', and 'Cancel'.

DNS 영역은 **IdM**에서 생성할 수 있습니다. **IdM** 서버가 **DNS** 서버를 관리하지 않으면 일반 텍스트 필드와 같이 메뉴 영역에 수동으로 영역을 입력할 수 있습니다.



참고

DNS를 통해 호스트를 확인할 수 있는지 여부를 건너뛰려면 **Force** (강제) 확인란을 선택합니다.

4.

Add and Edit(추가 및 편집) 버튼을 클릭하여 확장된 항목 페이지로 바로 이동하고 추가 속성 정보를 입력합니다. 호스트 하드웨어 및 물리적 위치에 대한 정보는 **host** 항목에 포함할 수 있습니다.

그림 37.3. 항목 페이지 확장

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

38장. ANSIBLE 플레이북을 사용하여 호스트 관리

Ansible은 시스템을 구성하고, 소프트웨어를 배포하고, 롤링 업데이트를 수행하는 데 사용되는 자동화 도구입니다. **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 작업에 대해 설명합니다.

- [IdM의 호스트](#)
- [호스트 등록](#)
- [FQDN에서만 정의된 IdM 호스트 항목이 있는지 확인](#)
- [IP 주소가 있는 IdM 호스트 항목이 있는지 확인](#)
- [임의의 암호를 사용하여 여러 IdM 호스트 항목이 있는지 확인](#)
- [여러 IP 주소가 있는 IdM 호스트 항목이 있는지 확인](#)
- [IdM 호스트 항목이 없는지 확인](#)

38.1. ID의 호스트

IdM(Identity Management)은 이러한 ID를 관리합니다.

- 사용자
- 서비스

-

호스트

호스트는 시스템을 나타냅니다. **IdM ID**로 호스트에 **IdM LDAP**에 항목이 있으며, 이는 **IdM** 서버의 디렉터리 서버 인스턴스 **389**입니다.

IdM LDAP의 **host** 항목은 다른 호스트와 도메인 내의 서비스 간 관계를 설정하는 데 사용됩니다. 이러한 관계는 도메인 내의 호스트에 권한 부여 및 제어 *위임*의 일부입니다. 호스트 기반 액세스 제어(**HBAC**) 규칙에서 모든 호스트를 사용할 수 있습니다.

IdM 도메인은 공통 **ID** 정보, 공통 정책 및 공유 서비스가 포함된 시스템 간 공통성을 설정합니다. 도메인 기능에 속한 시스템은 도메인의 클라이언트에 속하므로, 도메인에서 제공하는 서비스를 사용합니다. **IdM** 도메인은 시스템 전용 세 가지 기본 서비스를 제공합니다.

-

DNS

-

Kerberos

-

인증서 관리

IdM의 호스트는 실행 중인 서비스와 긴밀하게 연결됩니다.

-

서비스 항목은 호스트와 연결됩니다.

-

호스트는 호스트와 서비스 **Kerberos** 사용자를 모두 저장합니다.

38.2. 호스트 등록

이 섹션에서는 **IdM** 클라이언트로 호스트 등록 및 등록 중 및 등록 후에 발생하는 사항에 대해 설명합니다. 이 섹션에서는 **IdM** 호스트 및 **IdM** 사용자의 등록을 비교합니다. 또한 이 섹션에서는 호스트에서 사용할 수 있는 대체 유형의 인증에 대해 간단히 설명합니다.

호스트 등록은 다음으로 구성됩니다.

- **IdM LDAP에서 호스트 항목 생성:** IdM CLI에서 **ipa host-add** 명령을 사용하거나 이에 상응하는 **IdM 웹 UI** 작업을 사용할 수 있습니다.
- 호스트에서 IdM 서비스 구성(예: **SSSD**(시스템 보안 서비스 데몬), **Kerberos**, **certmonger**) 호스트를 IdM 도메인에 연결합니다.

두 작업은 별도로 또는 함께 수행할 수 있습니다.

개별적으로 수행되는 경우 두 가지 작업을 서로 다른 수준의 권한을 가진 두 사용자 간에 나눌 수 있습니다. 이는 대량 배포에 유용합니다.

ipa-client-install 명령은 두 작업을 함께 수행할 수 있습니다. 해당 항목이 아직 없는 경우 명령은 IdM LDAP에 호스트 항목을 생성하고 호스트에 대한 **Kerberos** 및 **SSSD** 서비스를 둘 다 구성합니다. 이 명령을 사용하면 IdM 도메인 내에 호스트를 가져와 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.

38.2.1. 호스트 등록에 필요한 사용자 권한

호스트 등록 작업을 수행하려면 권한이 없는 사용자가 원하지 않는 시스템을 IdM 도메인에 추가하지 못하도록 인증해야 합니다. 필요한 권한은 다음과 같은 여러 요인에 따라 다릅니다.

- **ipa-client-install** 실행과 별도로 호스트 항목을 생성하는 경우
- 등록에 **OTP**(일회용 암호)를 사용하는 경우

IdM LDAP에서 수동으로 호스트 항목을 생성하는 데 필요한 사용자 권한

ipa host-add CLI 명령 또는 IdM 웹 UI를 사용하여 IdM LDAP에서 호스트 항목을 생성하는 데 필요한 사용자 권한은 호스트 관리자입니다. 호스트 관리자 권한은 IT 전문가 역할을 통해 얻을 수 있습니다.

클라이언트를 IdM 도메인에 가입하는 사용자 권한

호스트는 **ipa-client-install** 명령을 실행하는 동안 IdM 클라이언트로 구성됩니다. **ipa-client-install** 명령을 실행하는 데 필요한 인증 정보의 수준은 다음 중 본인이 찾은 시나리오에 따라 다릅니다.

- IdM LDAP의 호스트 항목이 존재하지 않습니다. 이 시나리오의 경우 전체 관리자 자격 증명

또는 **Host Administrators** 역할이 필요합니다. 전체 관리자는 **admins** 그룹의 구성원입니다. **Host Administrators** 역할은 호스트를 추가하고 호스트를 등록할 수 있는 권한을 제공합니다. 이 시나리오에 대한 자세한 내용은 [사용자 자격 증명: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오.](#)

- **IdM LDAP의 host 항목이 있습니다.** 이 시나리오의 경우 **ipa-client-install** 을 실행하려면 제한된 관리자의 인증 정보가 필요합니다. 이 경우 제한된 관리자에게는 호스트 등록 권한을 제공하는 **Enrollment Administrator** 역할이 있습니다. 자세한 내용은 [사용자 **credentials**: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오.](#)
- **IdM LDAP의 호스트 항목이 존재하며 전체 또는 제한된 관리자가 호스트에 대한 OTP가 생성되었습니다.** 이 시나리오에서는 **--password** 옵션을 사용하여 **ipa-client-install** 명령을 실행하여 올바른 OTP를 제공하는 경우 **IdM** 클라이언트를 일반 사용자로 설치할 수 있습니다. 자세한 내용은 [일회성 암호를 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치.](#)

등록 후 **IdM** 호스트는 모든 새 세션을 인증하여 **IdM** 리소스에 액세스할 수 있도록 합니다. **IdM** 서버가 시스템을 신뢰하고 해당 시스템에 설치된 클라이언트 소프트웨어에서 **IdM** 연결을 수락하려면 시스템 인증이 필요합니다. 클라이언트를 인증한 후 **IdM** 서버는 요청에 응답할 수 있습니다.

38.2.2. IdM 호스트 및 사용자의 등록 및 인증: 비교

IdM의 사용자와 호스트 사이에는 많은 유사점이 있습니다. 이 섹션에서는 등록 단계에서 확인할 수 있는 몇 가지 유사점과 배포 단계에서 인증에 관한 몇 가지 유사 사항에 대해 설명합니다.

- 등록 단계([표 38.1. “사용자 및 호스트 등록”](#)):
 - 관리자는 사용자 또는 호스트 모두에 대해 **IdM**에 실제로 참여하기 전에 사용자 및 호스트에 대해 **LDAP** 항목을 생성할 수 있습니다. **stage** 사용자의 경우 명령은 **ipa stageuser-add** 입니다. 호스트에 대해 명령은 **ipa host-add** 입니다.
 - 키 테이블 또는 일부 익스텐트와 유사한 대칭 키인 **keytab**을 포함하는 파일은 호스트에서 **ipa-client-install** 명령 실행 중에 생성됩니다. 그러면 호스트가 **IdM** 영역에 결합됩니다. 비동기적으로 사용자는 계정을 활성화할 때 암호를 생성하여 **IdM** 영역에 가입해야 합니다.
 - 사용자 **password**는 사용자의 기본 인증 방법이지만 **keytab**은 호스트의 기본 인증 방법입니다. 키탭은 호스트의 파일에 저장됩니다.

표 38.1. 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] (호스트 자체에서 실행해야 합니다)

- 배포 단계(표 38.2. “사용자 및 호스트 세션 인증”):
 - 사용자가 새 세션을 시작하면 사용자는 암호를 사용하여 인증합니다. 마찬가지로, 콘솔이 켜질 때마다 호스트는 해당 키탭 파일을 표시하여 인증합니다. **SSSD(System Security Services Daemon)**는 이 프로세스를 백그라운드에서 관리합니다.
 - 인증에 성공하면 사용자 또는 호스트에서 티켓(TGT)을 부여하는 **Kerberos** 티켓을 가져옵니다.
 - 그런 다음 **TGT**를 사용하여 특정 서비스의 특정 티켓을 받습니다.

표 38.2. 사용자 및 호스트 세션 인증

	사용자	호스트
기본 인증 수단	암호	키탭
세션 시작 (일반 사용자)	\$ kinit <i>user_name</i>	[호스트의 전환]
인증에 성공한 결과	특정 서비스에 대한 액세스를 얻는데 사용할 TGT	특정 서비스에 대한 액세스를 얻는데 사용할 TGT

TGT 및 기타 **Kerberos** 티켓은 서버에서 정의한 **Kerberos** 서비스 및 정책의 일부로 생성됩니다. **Kerberos** 티켓을 처음 부여하고, **Kerberos** 자격 증명을 갱신하며, **Kerberos** 세션 삭제도 **IdM** 서비스에 의해 자동으로 처리됩니다.

38.2.3. IdM 호스트의 대체 인증 옵션

keytab 외에도 **IdM**은 두 가지 유형의 시스템 인증을 지원합니다.

-

SSH 키. 호스트의 SSH 공개 키가 생성되어 호스트 항목에 업로드됩니다. 여기에서 SSSD(시스템 보안 서비스 데몬)는 IdM을 ID 프로바이더로 사용하며 OpenSSH 및 기타 서비스와 함께 작업하여 IdM에 있는 공용 키를 참조할 수 있습니다.

-

시스템 인증서. 이 경우 시스템은 IdM 서버의 인증 기관에서 발급한 SSL 인증서를 사용한 다음 IdM의 디렉터리 서버에 저장됩니다. 그러면 서버에 인증할 때 인증서가 존재하도록 시스템으로 전송됩니다. 클라이언트의 인증서는 [certmonger](#) 라는 서비스에서 관리합니다.

38.3. ANSIBLE PLAYBOOK을 사용하여 FQDN과 함께 IDM 호스트 항목이 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에 호스트 항목이 있는지에 대해 설명합니다. 호스트 항목은 FQDN(정규화 된 도메인 이름)으로만 정의합니다.

다음 조건 중 하나가 적용되는 경우 호스트의 FQDN 이름을 지정하는 것으로 충분합니다.

-

IdM 서버는 DNS를 관리하도록 구성되어 있지 않습니다.

-

호스트에 고정 IP 주소가 없거나 호스트가 구성된 시점에는 IP 주소를 알 수 없습니다. FQDN에서만 정의한 호스트를 추가하면 기본적으로 IdM DNS 서비스에 자리 표시자 항목이 생성됩니다. 예를 들어 랩탑은 IdM 클라이언트로 사전 구성될 수 있지만 구성 시 IP 주소는 없습니다. DNS 서비스에서 레코드를 동적으로 업데이트하면 호스트의 현재 IP 주소가 탐지되고 해당 DNS 레코드가 업데이트됩니다.



참고

Ansible이 없으면 `ipa host-add` 명령을 사용하여 호스트 항목이 IdM에 생성됩니다. IdM에 호스트를 추가한 결과는 IdM에 있는 호스트의 상태입니다. Ansible은 먹등에 의존하므로 Ansible을 사용하여 IdM에 호스트를 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 여기에 **ipaserver**를 정의합니다.

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

2.

확인할 IdM에 있는 호스트의 FQDN으로 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 간소화하기 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:
      ipaadmin_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 절차에서는 IdM LDAP 서버에 호스트 항목이 생성되지만 호스트를 IdM Kerberos 영역에 등록하지 않습니다. 따라서 호스트를 IdM 클라이언트로 배포해야 합니다. 자세한 내용은 [Ansible 플레이북을 사용하여 Identity Management 클라이언트 설치를 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 `host01.idm.example.com` 이 `IdM`에 있는지 확인합니다.

38.4. ANSIBLE PLAYBOOK을 사용하여 DNS 정보가 포함된 IDM 호스트 항목이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 호스트 항목이 있는지에 대해 설명합니다. 호스트 항목은 **FQDN**(정규화 된 도메인 이름) 및 해당 **IP** 주소로 정의합니다.



참고

Ansible이 없으면 `ipa host-add` 명령을 사용하여 호스트 항목이 `IdM`에 생성됩니다. `IdM`에 호스트를 추가한 결과는 `IdM`에 있는 호스트의 상태입니다. **Ansible**은 먹등에 의존하므로 **Ansible**을 사용하여 `IdM`에 호스트를 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 여기에 `ipaserver`를 정의합니다.

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

2. 확인하고자 하는 `IdM`에 있는 호스트의 정규화된 도메인 이름 (**FQDN**)을 사용하여 **Ansible** 플레이북 파일을 생성합니다. 또한 `IdM` 서버가 **DNS**를 관리하고 호스트의 **IP** 주소를 알고 있는 경우 `ip_address` 매개 변수의 값을 지정합니다. 호스트가 **DNS** 리소스 레코드에 있으려면 **IP** 주소가 필요합니다. 이 단계를 간소화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/host/host-`

present.yml 파일에서 예제를 복사하고 수정할 수 있습니다. 다음과 같은 기타 추가 정보를 포함할 수도 있습니다.

```
---
- name: Host present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure host01.idm.example.com is present
    ipahost:
      ipadmin_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 절차에서는 **IdM LDAP** 서버에 호스트 항목이 생성되지만 호스트를 **IdM Kerberos** 영역에 등록하지 않습니다. 따라서 호스트를 **IdM** 클라이언트로 배포해야 합니다. 자세한 내용은 [Ansible 플레이북을 사용하여 Identity Management 클라이언트 설치를 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 IdM에 host01.idm.example.com 이 있는지 확인합니다.

38.5. ANSIBLE PLAYBOOK을 사용하여 임의의 암호와 함께 여러 IDM 호스트 항목이 있는지 확인

ipahost 모듈을 사용하면 시스템 관리자가 하나의 **Ansible** 작업만 사용하여 IdM에 여러 호스트 항목이 있는지 확인할 수 있습니다. 이 섹션에서는 **FQDN**(정규화 된 도메인 이름)으로만 정의된 여러 호스트 항목이 있는지 확인하는 방법에 대해 설명합니다. **Ansible** 플레이북을 실행하면 호스트에 대한 임의 암호가 생성됩니다.



참고

Ansible이 없으면 **ipa host-add** 명령을 사용하여 호스트 항목이 IdM에 생성됩니다. IdM에 호스트를 추가한 결과는 IdM에 있는 호스트의 상태입니다. **Ansible**은 먹등에 의존하므로 **Ansible**을 사용하여 IdM에 호스트를 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 여기에 **ipaserver** 를 정의합니다.

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

2.

확인하려는 IdM에 있는 호스트의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible** 플레이북 파일을 생성합니다. 호스트가 이미 IdM에 있고 `update_password`가 `on_create`로 제한된 경우에도 **Ansible** 플레이북에서 각 호스트의 임의 암호를 생성하도록 하려면 `random: yes` 및 `force: yes` 옵션을 추가합니다. 이 단계를 간소화하기 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

임의의 OTP(일회 암호)를 사용하여 호스트를 IdM 클라이언트로 배포하려면 **Ansible** 플레이북을 사용하여 IdM 클라이언트 등록에 대한 권한 부여 옵션을 참조하거나 일회성 암호를 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치.

검증 단계

1.

IdM 서버에 **admin**으로 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
```

출력은 임의의 암호가 있는 IdM에 `host01.idm.example.com` 이 있는지 확인합니다.

38.6. ANSIBLE PLAYBOOK을 사용하여 여러 IP 주소가 있는 IDM 호스트 항목이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 호스트 항목이 있는지 확인하는 방법을 설명합니다. 호스트 항목은 **FQDN**(정규화 된 도메인 이름) 및 여러 **IP** 주소로 정의합니다.



참고

ipa 호스트 유틸리티와 달리 **Ansible ipahost** 모듈은 호스트에 대해 여러 개의 **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 여기에 `ipaserver` 를 정의합니다.

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

2.

Ansible 플레이북 파일을 생성합니다. 확인하려는 **IdM** 에 있는 호스트의 정규화된 도메인 이

름 (FQDN)을 `ipahost` 변수의 이름으로 지정합니다. `ip_address` 구문을 사용하여 별도의 행에 여러 IPv4 및 IPv6 `ip_address` 값을 지정합니다. 이 단계를 간소화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/host/host-member-ipaddresses-present.yml` 과 일에서 예제를 복사하고 수정할 수 있습니다. 다음과 같은 추가 정보를 포함할 수도 있습니다.

```
---
- name: Host member IP addresses present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure host01.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 호스트를 **IdM Kerberos** 영역에 등록하지 않습니다. 따라서 호스트를 **IdM** 클라이언트로 배포해야 합니다. 자세한 내용은 [Ansible 플레이북을 사용하여 Identity Management 클라이언트 설치를 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 `host01.idm.example.com` 이 IdM에 있는지 확인합니다.

3.

호스트의 여러 IP 주소가 IdM DNS 레코드에 있는지 확인하려면 `ipa dnsrecord-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
```

출력은 플레이북에 지정된 모든 IPv4 및 IPv6 주소가 `host 01.idm.example.com` 호스트 항목과 올바르게 연결되어 있는지 확인합니다.

38.7. ANSIBLE 플레이북을 사용하여 IDM 호스트 항목이 없는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에 호스트 항목이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 인증 정보

절차

1.

인벤토리 파일(예: `inventory.file`)을 생성하고 여기에 `ipaserver` 를 정의합니다.

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

2.

확인하려는 IdM에 없는 호스트의 정규화된 도메인 이름 (FQDN)을 사용하여 **Ansible** 플레이북 파일을 생성합니다. IdM 도메인에 통합된 DNS가 있는 경우 **updatedns: yes** 옵션을 사용하여 DNS에서 호스트의 연결된 레코드를 제거합니다.

이 단계를 간소화하기 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:
      ipaadmin_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

절차 결과는 다음과 같습니다.

- IdM Kerberos 영역에 존재하지 않는 호스트.
- 호스트 항목이 IdM LDAP 서버에 존재하지 않습니다.

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.

admin으로 **ipaserver**에 로그인합니다.

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

2.

host01.idm.example.com 에 대한 정보를 표시 :

```
$ ipa host-show host01.idm.example.com
ipa: ERROR: host01.idm.example.com: host not found
```

출력은 호스트가 IdM에 없는지 확인합니다.

추가 리소스

- **ipahost** 변수의 정의와 샘플 **Ansible** 플레이북을 통해 **/usr/share/doc/ansible-freeipa/README-host.md** Markdown 파일에서 호스트의 존재, 누락 및 비활성화를 확인할 수 있습니다.
- 추가 플레이북은 **/usr/share/doc/ansible-freeipa/playbooks/host** 디렉터리에 있습니다.

39장. IDM CLI를 사용하여 호스트 그룹 관리

이 장에서는 **IdM(Identity Management)**의 호스트 그룹을 소개하고 **CLI(명령줄 인터페이스)**에서 호스트 그룹과 멤버를 관리하기 위한 다음 작업을 설명합니다.

- 호스트 그룹 및 해당 구성원 보기
- 호스트 그룹 생성
- 호스트 그룹 삭제
- 호스트 그룹 구성원 추가
- 호스트 그룹 구성원 제거
- 호스트 그룹 멤버 관리자 추가
- 호스트 그룹 구성원 관리자 제거

39.1. 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 특성이 있는 **IdM** 호스트 집합이 포함된 엔티티입니다. 예를 들어 회사 부서, 물리적 위치 또는 액세스 제어 요구 사항을 기반으로 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 구성원

IdM의 그룹 속성은 직접 및 간접 구성원 모두에 적용됩니다. 호스트 그룹 **B**가 호스트 그룹 **A**의 구성원 이면 호스트 그룹 **B**의 모든 구성원이 호스트 그룹 **A**의 간접 구성원으로 간주됩니다.

39.2. CLI를 사용하여 IDM 호스트 그룹 보기

이 섹션에서는 **CLI**(명령줄 인터페이스)를 사용하여 **IdM** 호스트 그룹을 보는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

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 특성을 표시하려면 **--all** 옵션을 추가합니다. 예를 들면 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
-----
```

39.3. CLI를 사용하여 IDM 호스트 그룹 생성

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 **IdM** 호스트 그룹을 생성하는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

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
-----
```

39.4. 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 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.

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 **IdM**에서 그룹 구성원이 삭제되지 않습니다.

39.5. CLI를 사용하여 IDM 호스트 그룹 멤버 추가

단일 명령을 사용하여 호스트 및 호스트 그룹을 **IdM** 호스트 그룹에 구성원을 추가할 수 있습니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.
- 선택사항입니다. **ipa hostgroup-find** 명령을 사용하여 호스트 및 호스트 그룹을 찾습니다.

절차

1. 멤버를 호스트 그룹에 추가하려면 **ipa hostgroup-add-member** 를 사용하고 관련 정보를 제공합니다. 다음 옵션을 사용하여 추가할 멤버 유형을 지정할 수 있습니다.

•

hosts 옵션을 사용하여 IdM 호스트 그룹에 하나 이상의 호스트를 추가합니다. 예를 들어 **example_member**라는 호스트를 **group_name** 그룹에 추가하려면 다음을 실행합니다.

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

•

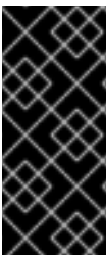
host groups 옵션을 사용하여 IdM 호스트 그룹에 하나 이상의 호스트 그룹을 추가합니다. 예를 들어 **nested_group**이라는 호스트 그룹을 **group_name**이라는 그룹에 추가하려면 :

```
$ 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 명령으로 여러 호스트와 여러 호스트 그룹을 IdM 호스트 그룹에 추가할 수 있습니다.

```
$ ipa hostgroup-add-member group_name --hosts={host1,host2} --hostgroups={group1,group2}
```



중요

호스트 그룹을 다른 호스트 그룹의 구성원으로 추가할 때 재귀 그룹을 생성하지 마십시오. 예를 들어 그룹 A가 그룹 B의 구성원인 경우 그룹 A의 구성원으로 그룹 B를 추가하지 마십시오. 반복적인 그룹은 예측할 수 없는 동작이 발생할 수 있습니다.

39.6. 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 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.
- **선택사항입니다.** `ipa hostgroup-find` 명령을 사용하여 그룹에 제거할 멤버가 포함되어 있는지 확인합니다.

절차

1.

호스트 그룹 구성원을 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 이름이 `group_name` 인 그룹에서 `example_member` 라는 호스트를 제거하려면 다음을 수행하십시오.

```
$ ipa hostgroup-remove-member group_name --hosts example_member
Host-group: group_name
Description: My host group
-----
Number of members removed 1
-----
```

- **hostnames 옵션을 사용하여 IdM 호스트 그룹에서 하나 이상의 호스트 그룹을 제거합니다.**
예를 들어, `nested_group` 이라는 호스트 그룹을 `group_name`:

```
$ 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 IdM에서 그룹 구성원이 삭제되지 않습니다.

- 다음 구문을 사용하여 하나의 단일 명령으로 IdM 호스트 그룹에서 여러 호스트와 여러 호스트

그룹을 제거할 수 있습니다.

```
$ ipa hostgroup-remove-member group_name --hosts={host1,host2} --hostgroups={group1,group2}
```

39.7. CLI를 사용하여 IDM 호스트 그룹 멤버 관리자 추가

단일 명령을 사용하여 호스트 및 호스트 그룹과 구성원 관리자를 **IdM** 호스트 그룹에 추가할 수 있습니다. 멤버 관리자는 **IdM** 호스트 그룹에 호스트 또는 호스트 그룹을 추가할 수 있지만 호스트 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.
- 멤버 관리자로 추가하려는 호스트 또는 호스트 그룹의 이름과 관리할 호스트 그룹의 이름이 있어야 합니다.

절차

1. **선택사항**입니다. **ipa hostgroup-find** 명령을 사용하여 호스트 및 호스트 그룹을 찾습니다.
2. 멤버 관리자를 호스트 그룹에 추가하려면 **ipa hostgroup-add-member-manager** 를 사용합니다.

예를 들어, **example_member** 라는 사용자를 멤버 관리자로 **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.

하나 이상의 호스트 그룹을 **IdM** 호스트 그룹에 멤버 관리자로 추가하려면 **--groups** 옵션을 사용합니다.

예를 들어 **admin_group**이라는 호스트 그룹을 멤버 관리자로 **group_name**이라는 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 업데이트에 **Identity Management** 환경의 모든 클라이언트에 분배하는 데 다소 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.

39.8. CLI를 사용하여 IDM 호스트 그룹 멤버 관리자 제거

단일 명령을 사용하여 **IdM** 호스트 그룹에서 멤버 관리자뿐만 아니라 호스트 그룹을 제거할 수 있습니다. 멤버 관리자는 **IdM** 호스트 그룹에서 호스트 그룹 구성원 관리자를 제거할 수 있지만 호스트 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 [Using kinit to log in IdM](#)에서 수동으로 참조하십시오.
- 제거 중인 기존 멤버 관리자 호스트 그룹의 이름과 관리 중인 호스트 그룹의 이름이 있어야 합니다.

절차

1. **선택사항**입니다. **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** 이라는 그룹에서 멤버 관리자로 **example_member** 라는 사용자를 제거하려면 다음을 수행하십시오.

```
$ 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 **group_name** 이라는 그룹에서 멤버 관리자로 **nested_group** 이라는 호스트 그룹을 제거하려면 다음을 수행하십시오.


```
$ 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 업데이트에 Identity Management 환경의 모든 클라이언트에 분배하는 데 다소 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.

40장. IDM 웹 UI를 사용하여 호스트 그룹 관리

이 장에서는 **IdM(Identity Management)**의 호스트 그룹을 소개하고 웹 인터페이스(**웹 UI**)에서 호스트 그룹과 멤버를 관리하기 위한 다음 작업을 설명합니다.

- 호스트 그룹 및 해당 구성원 보기
- 호스트 그룹 생성
- 호스트 그룹 삭제
- 호스트 그룹 구성원 추가
- 호스트 그룹 구성원 제거
- 호스트 그룹 멤버 관리자 추가
- 호스트 그룹 구성원 관리자 제거

40.1. 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 특성이 있는 **IdM** 호스트 집합이 포함된 엔티티입니다. 예를 들어 회사 부서, 물리적 위치 또는 액세스 제어 요구 사항을 기반으로 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 구성원

IdM의 그룹 속성은 직접 및 간접 구성원 모두에 적용됩니다. 호스트 그룹 **B**가 호스트 그룹 **A**의 구성원 이면 호스트 그룹 **B**의 모든 구성원이 호스트 그룹 **A**의 간접 구성원으로 간주됩니다.

40.2. IDM 웹 UI에서 호스트 그룹 보기

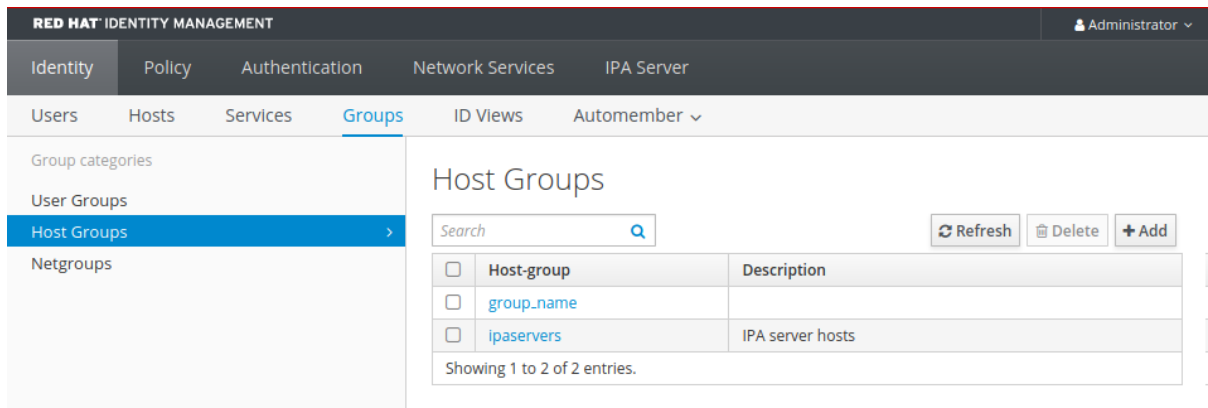
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 **IdM** 호스트 그룹을 보는 방법에 대해 설명합니다.

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 참조 하십시오.

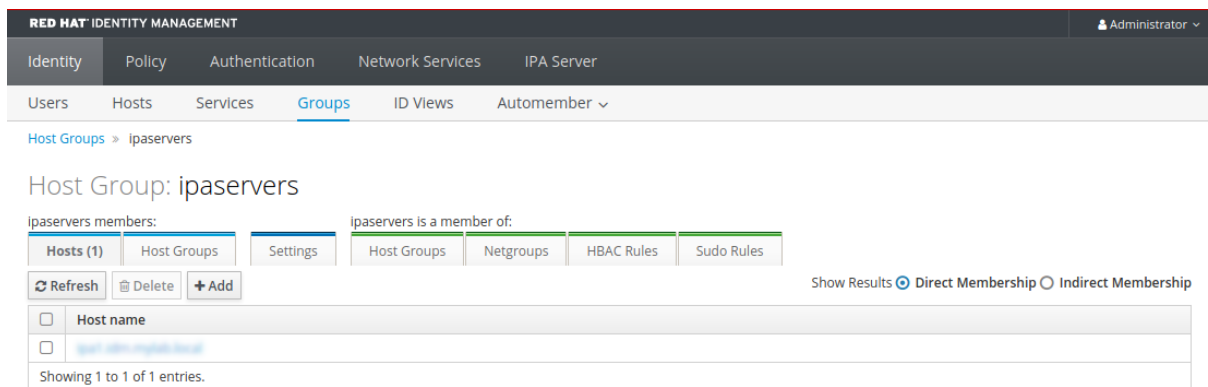
절차

1. **Identity → Groups(ID → 그룹)**를 클릭하고 **Host Groups(호스트 그룹)** 탭을 선택합니다.
 - 페이지에는 기존 호스트 그룹과 해당 설명이 나열됩니다.
 - 특정 호스트 그룹을 검색할 수 있습니다.



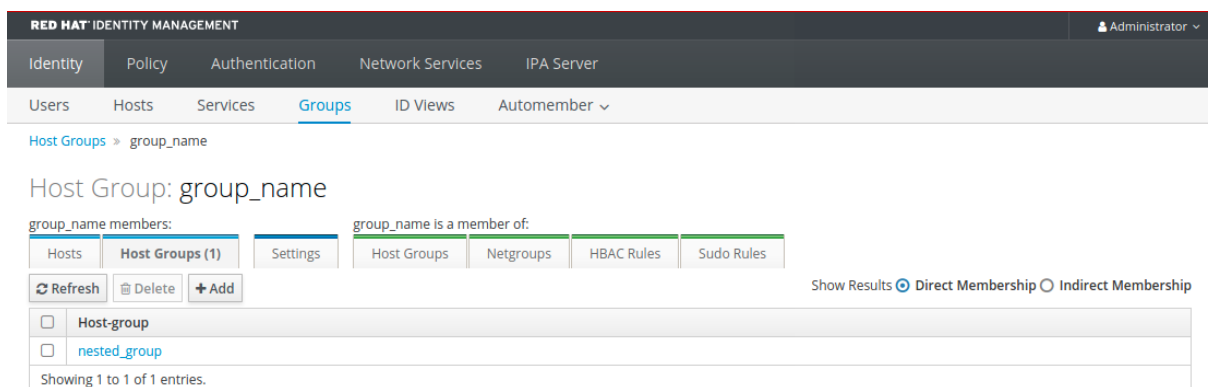
2.

목록에서 그룹을 클릭하여 이 그룹에 속하는 호스트를 표시합니다. 직접 또는 간접 구성원으로 결과를 제한할 수 있습니다.



3.

Host Groups(호스트 그룹) 탭을 선택하여 이 그룹(**nested 호스트 그룹**)에 속하는 호스트 그룹을 표시합니다. 직접 또는 간접 구성원으로 결과를 제한할 수 있습니다.



40.3. IDM 웹 UI에서 호스트 그룹 생성

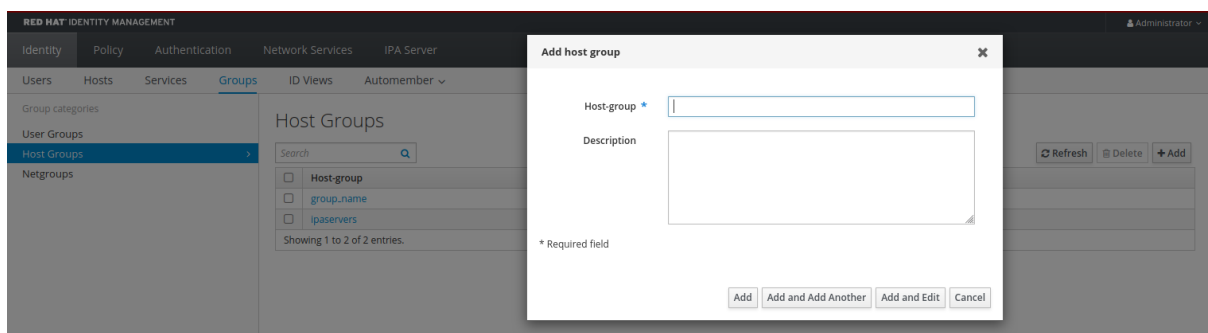
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 **IdM** 호스트 그룹을 생성하는 방법에 대해 설명합니다.

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. **Identity** → **Groups(ID → 그룹)**를 클릭하고 **Host Groups(호스트 그룹)** 탭을 선택합니다.
2. **추가**를 클릭합니다. **Add host group (호스트 그룹 추가)** 대화 상자가 표시됩니다.
3. 그룹에 대한 정보 제공: **name(필수)** 및 **설명(선택 사항)**.
4. **Add(추가)**를 클릭하여 확인합니다.



40.4. IDM 웹 UI에서 호스트 그룹 삭제

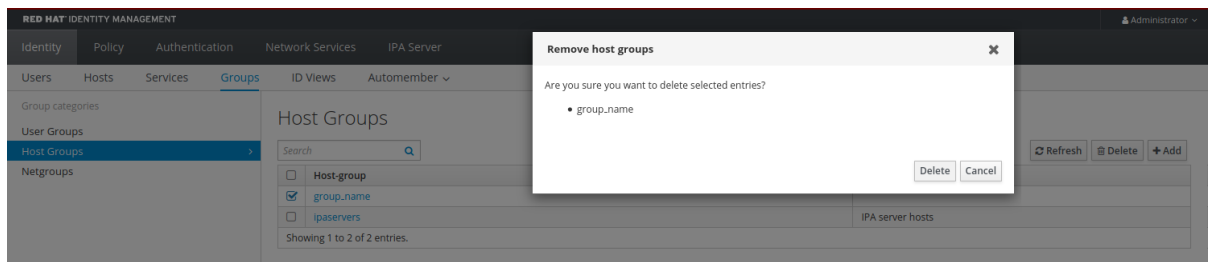
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 **IdM** 호스트 그룹을 삭제하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. **ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.**
2. **제거할 IdM 호스트 그룹을 선택하고 Delete(삭제)를 클릭합니다.** 확인 대화 상자가 나타납니다.
3. **삭제를 클릭하여 확인합니다.**



참고

호스트 그룹을 제거해도 IdM의 그룹 구성원이 삭제되지 않습니다.

40.5. IDM 웹 UI에서 호스트 그룹 구성원 추가

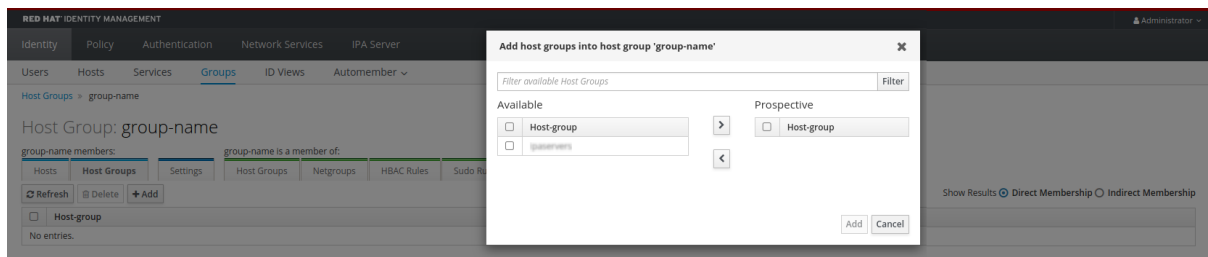
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM에 호스트 그룹 구성원을 추가하는 방법을 설명합니다.

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 → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.**
2. **멤버를 추가할 그룹의 이름을 클릭합니다.**
3. **추가할 구성원 유형에 따라 **Hosts**(호스트) 또는 **Host groups** (호스트 그룹) 탭을 클릭합니다. 해당 대화 상자가 나타납니다.**
4. **추가할 호스트 또는 호스트 그룹을 선택하고 > 화살표 버튼을 클릭하여 **Prospective** 열로 이동합니다.**
5. **Add(추가) 를 클릭하여 확인합니다.**



40.6. IDM 웹 UI에서 호스트 그룹 구성원 제거

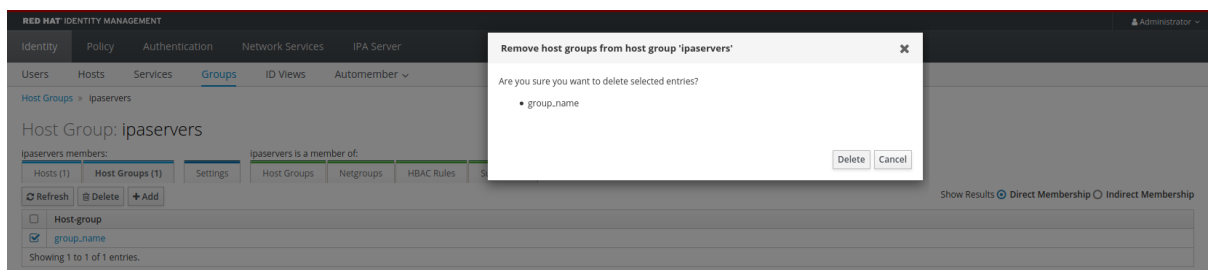
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM에서 호스트 그룹 구성원을 제거하는 방법을 설명합니다.

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 참조하십시오.**

절차

1. **ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.**
2. **멤버를 제거할 그룹의 이름을 클릭합니다.**
3. **제거할 구성원 유형에 따라 **Hosts**(호스트) 또는 **Host groups** (호스트 그룹) 탭을 클릭합니다.**
4. **제거할 멤버 옆에 있는 확인란을 선택합니다.**
5. **Delete(삭제)를 클릭합니다. 확인 대화 상자가 나타납니다.**



6. **Delete(삭제)를 클릭하여 확인합니다. 선택한 멤버가 삭제됩니다.**

40.7. 웹 UI를 사용하여 IDM 호스트 그룹 멤버 관리자 추가

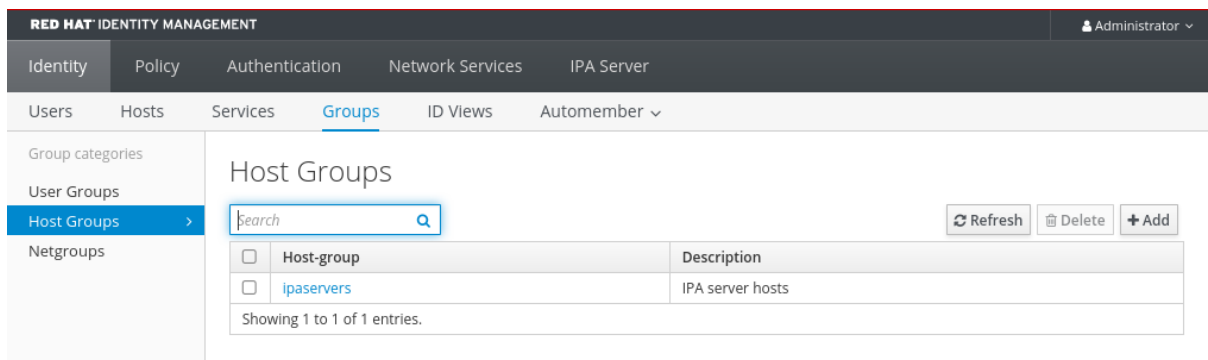
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM에서 사용자 또는 사용자 그룹을 호스트 그룹 멤버 관리자로 추가하는 방법에 대해 설명합니다. 멤버 관리자는 IdM 호스트 그룹에 호스트 그룹 구성원 관리자를 추가할 수 있지만 호스트 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**
- **IdM 웹 UI에 로그인되어 있습니다.** 자세한 내용은 웹 브라우저에서 **IdM 웹 UI 액세스**를 참조하십시오.
- **멤버 관리자로 추가하려는 호스트 그룹 이름과 관리할 호스트 그룹의 이름이 있어야 합니다.**

절차

1. **ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.**



2. **멤버 관리자를 추가할 그룹의 이름을 클릭합니다.**
3. **추가하려는 멤버 관리자 유형에 따라 멤버 **managerss**(사용자 그룹) 또는 **Users** (사용자) 탭을 클릭합니다.** 해당 대화 상자가 나타납니다.
4. **추가를 클릭합니다.**

Add groups as member managers for host group 'ipaservers'

Filter available User Groups
Filter

Available

☐	Group name
☐	editors
☐	ipasers
☐	trust admins

>
<

Prospective

☐	Group name

Add
Cancel

5.

추가할 사용자 또는 사용자 그룹을 선택하고 > 화살표 버튼을 클릭하여 **Prospective** 열로 이동합니다.

6.

Add(추가) 를 클릭하여 확인합니다.



참고

호스트 그룹에 멤버 관리자를 추가한 후 업데이트에 **Identity Management** 환경의 모든 클라이언트에 분배하는 데 다소 시간이 걸릴 수 있습니다.

검증 단계

•

Host Group(호스트 그룹) 대화 상자에서 사용자 그룹 또는 사용자가 그룹 또는 사용자의 멤버 관리자 목록에 추가되었는지 확인합니다.

RED HAT IDENTITY MANAGEMENT
Administrator

Identity Policy Authentication Network Services IPA Server

Users Hosts Services Groups ID Views Automember

Host Groups » ipaservers

Host Group: ipaservers

ipaservers members:

Hosts (1) Host Groups Settings

Refresh Delete Add

ipaservers is a member of:

Host Groups Netgroups HBAC Rules Sudo Rules

ipaservers member managers:

User Groups (1) Users

☐ Group name

☐ admins

Showing 1 to 1 of 1 entries.

40.8. 웹 UI를 사용하여 IDM 호스트 그룹 멤버 관리자 제거

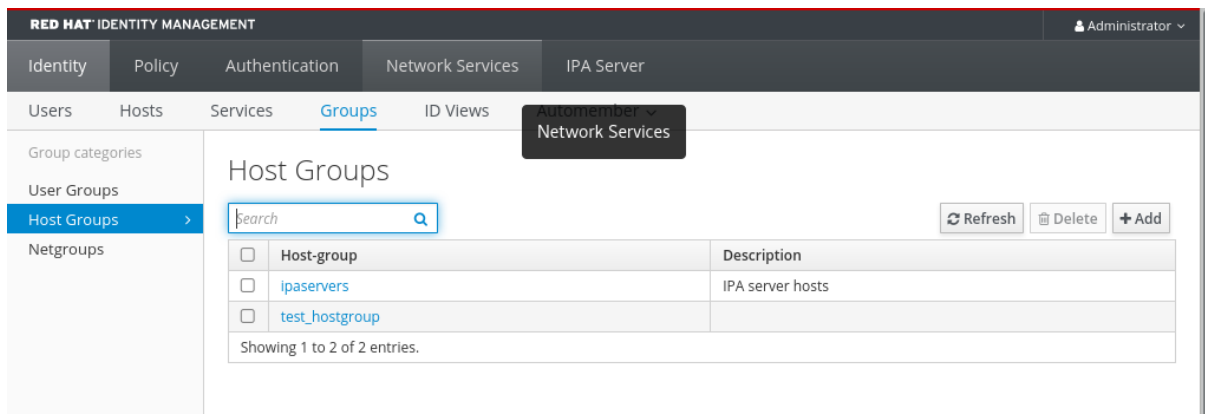
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 **IdM**에서 사용자 또는 사용자 그룹을 호스트 그룹 멤버 관리자로 제거하는 방법을 설명합니다. 멤버 관리자는 **IdM** 호스트 그룹에서 호스트 그룹 구성원 관리자를 제거할 수 있지만 호스트 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **IdM** 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 **IdM 웹 UI 액세스**를 참조하십시오.
- 제거 중인 기존 멤버 관리자 호스트 그룹의 이름과 관리 중인 호스트 그룹의 이름이 있어야 합니다.

절차

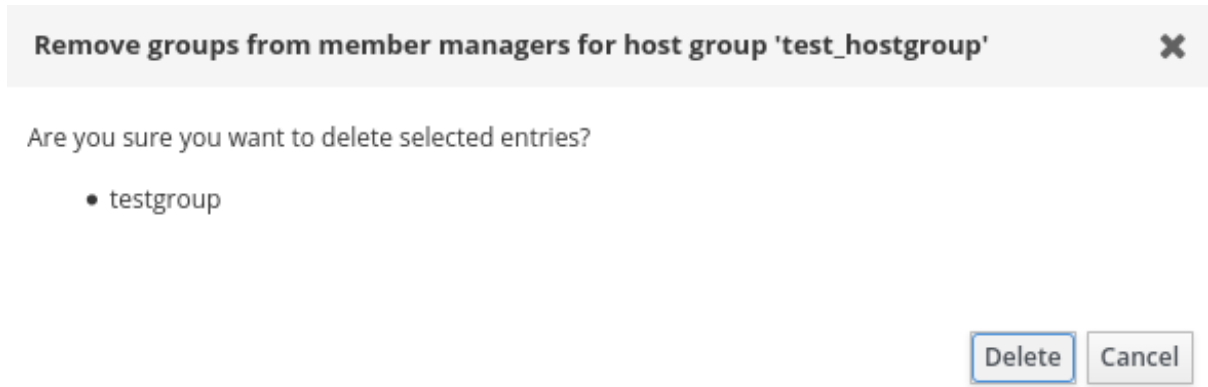
1. **ID** → **그룹**을 클릭하고 **호스트 그룹** 탭을 선택합니다.



2. 멤버 관리자를 제거할 그룹의 이름을 클릭합니다.
3. 제거할 멤버 관리자의 유형에 따라 멤버 **managerss**(사용자 그룹) 또는 **Users** (사용자) 탭을 클릭합니다. 해당 대화 상자가 나타납니다.
4. 제거할 사용자 또는 사용자 그룹을 선택하고 **Delete**(삭제)를 클릭합니다.

5.

Delete(삭제) 를 클릭하여 확인합니다.



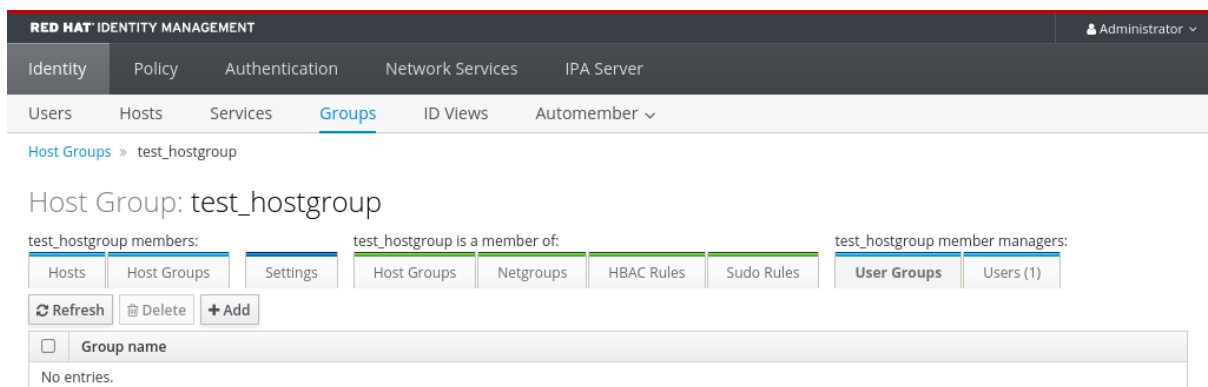
참고

호스트 그룹에서 멤버 관리자를 제거한 후 업데이트에 Identity Management 환경의 모든 클라이언트에 분배하는 데 다소 시간이 걸릴 수 있습니다.

검증 단계

•

Host Group(호스트 그룹) 대화 상자에서 사용자 그룹 또는 사용자가 그룹 또는 사용자의 멤버 관리자 목록에서 제거되었는지 확인합니다.



41장. ANSIBLE 플레이북을 사용하여 호스트 그룹 관리

이 장에서는 **IdM(Identity Management)**의 **호스트 그룹**을 소개하고 **Ansible**을 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 중첩된 호스트 그룹이 없는지 확인합니다.**
- **IdM 호스트 그룹에서 멤버 관리자가 없는지 확인**

41.1. 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 특성이 있는 **IdM 호스트 집합**이 포함된 엔티티입니다. 예를 들어 회사 부서, 물리적 위치 또는 액세스 제어 요구 사항을 기반으로 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 구성원

IdM의 그룹 속성은 직접 및 간접 구성원 모두에 적용됩니다. 호스트 그룹 **B**가 호스트 그룹 **A**의 구성원이면 호스트 그룹 **B**의 모든 구성원이 호스트 그룹 **A**의 간접 구성원으로 간주됩니다.

41.2. ANSIBLE PLAYBOOK을 사용하여 IDENTITY MANAGEMENT 호스트 그룹이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 IdM(Identity Management)에 호스트 그룹이 있는지 확인하는 방법을 설명합니다.



참고

Ansible이 없으면 `ipa hostgroup-add` 명령을 사용하여 호스트 그룹 항목이 IdM에 생성됩니다. IdM에 호스트 그룹을 추가한 결과는 IdM에 있는 호스트 그룹의 상태입니다. **Ansible**은 멍등에 의존하므로 **Ansible**을 사용하여 호스트 그룹을 IdM에 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. **inventory** .file과 같은 인벤토리 파일을 생성하고 타겟 IdM 서버 목록으로 **ipaserver** 를 정의합니다.

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

2.

필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 예를 들어 **database**라는 호스트 그룹이 있는지 확인하려면 - **ipahostgroup** 작업에 **name: database**를 지정합니다. 이 단계를 간소화하기 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 **IdM**에 호스트 그룹을 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.

admin으로 **ipaserver** 에 로그인합니다.

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

2.

관리자용 **Kerberos** 티켓을 요청합니다.

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

3.

확인하고자 하는 IdM에 있는 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
Host-group: databases
```

데이터베이스 호스트 그룹은 IdM에 있습니다.

41.3. ANSIBLE PLAYBOOK을 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 플레이북에서 참조할 호스트는 IdM에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 항목이 있는지](#) 확인하십시오.
- Ansible 플레이북 파일에서 참조하는 호스트 그룹이 IdM에 추가되었습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지](#) 확인합니다.

절차

1.

inventory .file과 같은 인벤토리 파일을 생성하고 타겟 IdM 서버 목록으로 **ipaserver** 를 정의합니다.

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

2.

필요한 호스트 정보를 사용하여 **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 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-present-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 실행될 때 데이터베이스 그룹 자체를 추가하기 위한 시도가 수행되지 않았음을 나타냅니다. 대신 **db.idm.example.com** 을 데이터베이스에 추가하려는 시도만 수행합니다.

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. 관리자용 **Kerberos** 티켓을 요청합니다.

```

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

```

3. 호스트 그룹에 대한 정보를 표시하여 어떤 호스트가 있는지 확인합니다.

```

$ ipa hostgroup-show databases
Host-group: databases
Member hosts: db.idm.example.com

```

db.idm.example.com 호스트는 데이터베이스 호스트 그룹의 멤버로 있습니다.

41.4. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹 중첩

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 호스트 그룹에 중첩된 호스트 그룹이 있는지에 대해 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 확인](#)합니다.

절차

1. **inventory .file**과 같은 인벤토리 파일을 생성하고 타겟 **IdM** 서버 목록으로 **ipaserver**를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 중첩된 호스트 그룹 **A**가 **Ansible** 플레이북의 호스트 그룹 **B**에 있는지 확인하려면 - **ipahostgroup** 변수 중 **name** 변수를 사용하여 호스트 그룹 **B**의 이름을 지정합니다. **host group** 변수를 사용하여 중첩 호스트 그룹의 이름을 지정합니다. 이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-present-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:
        ipadmin_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 실행될 때 데이터베이스 그룹 자체를 IdM에 추가하기 위한 시도가 수행되지 않았음을 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. 관리자용 Kerberos 티켓을 요청합니다.

```

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

```

3. 중첩 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.

41.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 멤버 관리자가 있는지 확인

다음 절차에서는 Ansible 플레이북을 사용하여 IdM 호스트 및 호스트 그룹에 구성원 관리자가 있는지

확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- 멤버 관리자로 추가하려는 호스트 또는 호스트 그룹의 이름과 관리할 호스트 그룹의 이름이 있어야 합니다.

절차

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

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

2. 필요한 호스트 및 호스트 그룹 멤버 관리 정보를 사용하여 **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:
        ipadmin_password: MySecret123
        name: group_name
        membermanager_user: example_member

    - name: Ensure member manager group project_admins is present for group_name
      ipahostgroup:
        ipadmin_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.

관리자로 **ipaserver** 에 로그인합니다.

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

2.

testhostgroup 에 대한 정보 표시 :

```
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** 도움말 페이지를 참조하십시오.

41.6. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 호스트가 없는지 확인합니다.

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**의 호스트 그룹에서 호스트가 없는지 확인하는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.

- Ansible 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.
- Ansible 플레이북에서 참조할 호스트는 IdM에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 항목이 있는지](#) 확인하십시오.
- Ansible 플레이북 파일에서 참조하는 호스트 그룹은 IdM에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지](#) 확인합니다.

절차

1.

inventory .file과 같은 인벤토리 파일을 생성하고 타겟 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 **host** 매개 변수를 사용하여 확인할 호스트 그룹에 없는 호스트 이름을 지정합니다. 이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-absent-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 실행될 때 데이터베이스 그룹 자체를 제거하기 위한 시도가 수행되지 않았음을 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.

관리자용 Kerberos 티켓을 요청합니다.

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

3.

호스트 그룹 및 포함된 호스트에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
Host-group: databases
Member host-groups: mysql-server, oracle-server
```

db.idm.example.com 호스트가 databases 호스트 그룹에 존재하지 않습니다.

41.7. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 중첩된 호스트 그룹이 있는지 확인합니다.

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)의 외부 호스트 그룹에서 중첩된 호스트 그룹이 있는지 확인하는 방법에 대해 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 확인](#)합니다.

절차

1.

inventory .file과 같은 인벤토리 파일을 생성하고 타겟 **IdM** 서버 목록으로 **ipaserver** 를 정의합니다.

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

2.

필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. **ipahostgroup** 변수 중에 **name** 변수를 사용하여 외부 호스트 그룹의 이름을 지정합니다. 호스트 그룹 변수를 사용하여 중첩 호스트 그룹의 이름을 지정합니다. 이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-absent-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 데이터베이스 호스트 그룹에 **mysql-server** 및 **oracle-server** 호스트 그룹이 없는지 확인합니다. **action: member** 행은 플레이북이 실행될 때 데이터베이스 그룹 자체가 **IdM**에서 삭제되었는지 확인하기 위한 시도가 수행되지 않았음을 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. 관리자용 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 없는지 확인합니다.

41.8. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹이 없는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 호스트 그룹이 없는지 확인하는 방법을 설명합니다.

참고

Ansible이 없으면 **ipa hostgroup-del** 명령을 사용하여 호스트 그룹 항목이 **IdM**에서 제거됩니다. **IdM**에서 호스트 그룹을 제거한 결과는 **IdM**에 없는 호스트 그룹의 상태입니다. **Ansible**은 멍등에 의존하므로 **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. **inventory .file**과 같은 인벤토리 파일을 생성하고 타겟 **IdM** 서버 목록으로 **ipaserver** 를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 간소화하기 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 **IdM**에서 데이터베이스 호스트 그룹이 없는지 확인합니다. **state: absent** 는 이미 삭제되지 않는 한 **IdM**에서 호스트 그룹을 삭제하도록 요청합니다.

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. 관리자용 **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
```

데이터베이스 호스트 그룹은 **IdM**에 존재하지 않습니다.

41.9. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 멤버 관리자가 없는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 호스트 및 호스트 그룹에 구성원 관리자가 없는지 확인합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- 멤버 관리자로 제거하려는 사용자 또는 사용자 그룹의 이름과 관리 중인 호스트 그룹의 이름이 있어야 합니다.

절차

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

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

2. 필요한 호스트 및 호스트 그룹 멤버 관리 정보를 사용하여 **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.

관리자로 **ipaserver** 에 로그인합니다.

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

2.

testhostgroup 에 대한 정보 표시 :

```
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** 도움말 페이지를 참조하십시오.

42장. KERBEROS 티켓 정책 관리

IdM(Identity Management)의 **Kerberos** 티켓 정책은 **Kerberos** 티켓 액세스, 기간 및 갱신에 대한 제한을 설정합니다. **IdM** 서버에서 실행되는 **KDC(키 배포 센터)**에 대한 **Kerberos** 티켓 정책을 구성할 수 있습니다.

이 장에서는 다음과 같은 **Kerberos** 티켓 관리 주제와 작업을 설명합니다.

- [IdM KDC의 역할](#)
- [IdM Kerberos 티켓 정책 유형](#)
- [Kerberos 인증 표시기](#)
- [IdM 서비스에 대한 인증 표시기 시행](#)
- [글로벌 티켓 라이프사이클 정책 구성](#)
- [인증 표시기별로 글로벌 티켓 정책 구성](#)
- [사용자의 기본 티켓 정책 구성](#)
- [사용자에 대한 개별 인증 표시기 티켓 정책 구성](#)
- [krbtpolicy-mod 명령의 인증 표시 옵션](#)

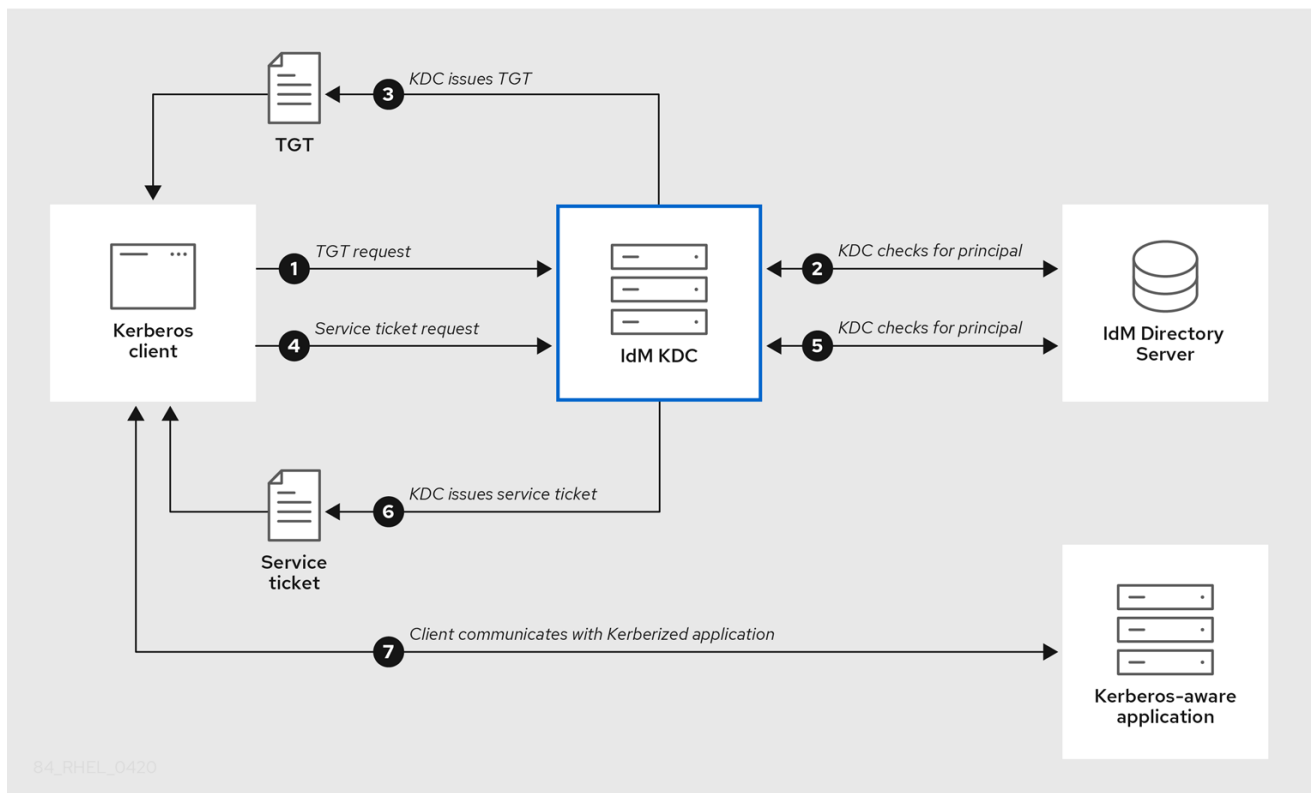
42.1. IDM KDC의 역할

ID 관리의 인증 메커니즘은 **KDC(키 배포 센터)**에서 설정한 **Kerberos** 인프라를 사용합니다. **KDC**는 자격 증명 정보를 저장하고 **IdM** 네트워크 내의 엔티티에서 시작된 데이터의 신뢰성을 보장하는 신뢰할 수 있는 권한입니다.

각 **IdM** 사용자, 서비스 및 호스트는 **Kerberos** 클라이언트 역할을 하며 고유한 **Kerberos 주체**로 식별됩니다.

- **users: identifier@REALM** (예: **admin@EXAMPLE.COM**)
- **서비스: service/fully-qualified-hostname@REALM** (예: **http/server.example.com@EXAMPLE.COM**)
- **호스트의 경우: host/fully-qualified-hostname@REALM** (예: **host/client.example.com@EXAMPLE.COM**)

다음 이미지는 **Kerberos** 클라이언트, **KDC** 및 클라이언트가 통신하려는 **Kerberized** 애플리케이션 간의 통신을 간소화한 것입니다.



1. **Kerberos** 클라이언트는 **Kerberos** 주체로 인증하여 **KDC**를 식별합니다. 예를 들어 **IdM** 사용자는 **kinit 사용자 이름**을 수행하고 암호를 제공합니다.
2. **KDC**는 데이터베이스의 주체를 확인하고, 클라이언트를 인증하고, **Kerberos 티켓 정책**을 평가하여 요청을 부여할지 여부를 결정합니다.

3. **KDC**는 고객에게 해당 티켓 정책에 따라 라이프사이클 및 **인증 표시기**가 포함된 티켓(**TGT**)을 발행합니다.
4. **TGT**를 사용하면 클라이언트는 **KDC**의 **서비스 티켓**을 요청하여 대상 호스트에서 **Kerberized** 서비스와 통신합니다.
5. **KDC**는 고객의 **TGT**가 여전히 유효한지 확인하고, 티켓 정책에 대해 서비스 티켓 요청을 평가합니다.
6. **KDC**는 고객에게 **서비스 티켓**을 발행합니다.
7. 서비스 티켓을 사용하여 클라이언트는 대상 호스트에서 서비스와 암호화된 통신을 시작할 수 있습니다.

42.2. IDM KERBEROS 티켓 정책 유형

IdM Kerberos 티켓 정책은 다음과 같은 티켓 정책 유형을 구현합니다.

연결 정책

서로 다른 보안 수준으로 **Kerberized** 서비스를 보호하기 위해 연결 정책을 정의하여 **TGT**(티켓 제공 티켓)를 검색하는 데 사용하는 사전 인증 메커니즘을 기반으로 규칙을 적용할 수 있습니다.

예를 들어 **client1.example.com**에 연결하는 데 스마트 카드 인증이 필요하며, **client2.example.com**의 테스트 서비스 애플리케이션에 액세스하려면 이중 인증이 필요합니다.

연결 정책을 적용하려면 **인증 표시기**를 서비스와 연결합니다. 서비스 티켓 요청에 필요한 인증 표시기가 있는 클라이언트만 해당 서비스에 액세스할 수 있습니다. 자세한 내용은 **Kerberos 인증 표시기**를 참조하십시오.

티켓 라이프 사이클 정책

각 **Kerberos** 티켓은 수명 과 잠재적인 갱신 기간이 있습니다. 티켓을 최대 수명에 도달하기 전에 갱신할 수 있지만 최대 갱신 기간을 초과하면 안 됩니다.

기본 글로벌 티켓 수명은 1일(86400초)이며 기본 글로벌 최대 갱신 기간은 1주(604800초)입니다. 이러한 글로벌 값을 조정하려면 **글로벌 티켓 라이프사이클 정책 구성**을 참조하십시오.

또한 자체 티켓 라이프사이클 정책을 정의할 수도 있습니다.

- 각 인증 지표에 대해 서로 다른 글로벌 티켓 라이프사이클 값을 구성하려면 [인증당 글로벌 티켓 정책 구성을 참조하십시오](#).
- 사용된 인증 방법에 관계없이 적용되는 단일 사용자의 티켓 라이프사이클 값을 정의하려면 [사용자에 대한 기본 티켓 정책 구성을 참조하십시오](#).
- 단일 사용자에게만 적용되는 각 인증 지표에 대한 개별 티켓 라이프사이클 값을 정의하려면 사용자에 [대한 개별 인증 표시기 티켓 정책 구성을 참조하십시오](#).

42.3. KERBEROS 인증 표시기

KDC(Kerberos Key Distribution Center)는 클라이언트가 ID를 증명하는 데 사용한 사전 인증 메커니즘을 기반으로 인증 표시기를 TGT(티켓 제공 티켓)에 연결합니다.

otp

이중 인증 (암호 + 일회 암호)

마케도니아

RAID 인증(일반적으로 802.1x 인증의 경우)

pkinit

PKINIT, 스마트 카드 또는 인증서 인증

강화

강화된 암호 (SPAKE 또는 FAST)^[1]

그런 다음 KDC는 TGT의 인증 표시기를 통해 발생하는 모든 서비스 티켓 요청에 연결합니다. KDC는 인증 지표에 따라 서비스 액세스 제어, 최대 티켓 수명, 최대 재생성 기간 등의 정책을 시행합니다.

42.3.1. 인증 표시기 및 IdM 서비스

서비스 또는 호스트를 인증 표시기와 연결할 경우 해당 인증 메커니즘을 사용하여 TGT를 획득한 클라

이엔트만 액세스할 수 있습니다. 애플리케이션 또는 서비스가 아닌 **KDC**는 서비스 티켓 요청의 인증 표시기를 확인하고 **Kerberos** 연결 정책에 따라 요청을 허용하거나 거부합니다.

예를 들어 호스트 **secure.example.com**에 연결하기 위해 이중 인증을 요구하려면 **otp** 인증 표시기를 **host/secure.example.com@EXAMPLE.COM Kerberos** 주체와 연결합니다. **KDC**에서 초기 **TGT**를 얻기 위해 일회성 암호를 사용한 사용자만 로그인할 수 있습니다.

서비스 또는 호스트에 인증 표시기가 할당되지 않은 경우 메커니즘에서 인증된 티켓을 수락합니다.

추가 리소스

- [IdM 서비스와 인증 표시기를 연결하려면 IdM 서비스에 대한 인증 표시기 사용을 참조하십시오.](#)

42.4. IDm 서비스에 대한 인증 표시기 시행

이 절차에서는 **IdM** 서비스 생성 및 수신 서비스 티켓 요청의 특정 **Kerberos** 인증 표시기가 필요하도록 구성하는 방법을 설명합니다.

인증 표시기를 **IdM** 서비스와 연결하면 특정 사전 인증 메커니즘을 사용한 클라이언트만 초기 티켓 제공 티켓(**TGT**)을 받을 수 있습니다.

42.4.1. IdM 서비스 항목 및 Kerberos 키 탭 생성

IdM 호스트에서 실행되는 서비스에 대해 **IdM** 서비스 항목을 **IdM**에 추가하면 해당 **Kerberos** 사용자가 생성되고 서비스에서 **SSL** 인증서, **Kerberos** 키탭 또는 둘 다를 요청할 수 있습니다.

다음 절차에서는 **IdM** 서비스 항목을 생성하고 관련 **Kerberos keytab**을 생성하여 해당 서비스와의 통신을 암호화하는 방법을 설명합니다.

사전 요구 사항

- 서비스는 **Kerberos** 주체, **SSL** 인증서 또는 둘 다를 저장할 수 있습니다.

절차

1.

ipa service-add 명령으로 **IdM** 서비스를 추가하여 연결된 **Kerberos** 주체를 생성합니다. 예

를 들어 호스트 **client.example.com**에서 실행되는 테스트 서비스 애플리케이션에 대한 **IdM** 서비스 항목을 생성하려면 다음을 수행합니다.

```
[root@client ~]# ipa service-add testservice/client.example.com
-----
Modified service "testservice/client.example.com@EXAMPLE.COM"
-----
Principal name: testservice/client.example.com@EXAMPLE.COM
Principal alias: testservice/client.example.com@EXAMPLE.COM
Managed by: client.example.com
```

2.

클라이언트에서 서비스에 대한 **Kerberos** 키탭을 생성하고 저장합니다.

```
[root@client ~]# ipa-getkeytab -k /etc/testservice.keytab -p
testservice/client.example.com
Keytab successfully retrieved and stored in: /etc/testservice.keytab
```

검증 단계

1.

ipa service-show 명령을 사용하여 **IdM** 서비스에 대한 정보를 표시합니다.

```
[root@server ~]# ipa service-show testservice/client.example.com
Principal name: testservice/client.example.com@EXAMPLE.COM
Principal alias: testservice/client.example.com@EXAMPLE.COM
Keytab: True
Managed by: client.example.com
```

2.

klist 명령을 사용하여 서비스의 **Kerberos keytab** 내용을 표시합니다.

```
[root@server etc]# klist -ekt /etc/testservice.keytab
Keytab name: FILE:/etc/testservice.keytab
KVNO Timestamp          Principal
-----
2 04/01/2020 17:52:55 testservice/client.example.com@EXAMPLE.COM (aes256-cts-
hmac-sha1-96)
2 04/01/2020 17:52:55 testservice/client.example.com@EXAMPLE.COM (aes128-cts-
hmac-sha1-96)
2 04/01/2020 17:52:55 testservice/client.example.com@EXAMPLE.COM (camellia128-
cts-cmac)
2 04/01/2020 17:52:55 testservice/client.example.com@EXAMPLE.COM (camellia256-
cts-cmac)
```

42.4.2. IdM 서비스와 인증 표시기 연결

다음 절차에서는 들어오는 서비스 티켓 요청의 특정 **Kerberos** 인증 지표가 필요하도록 서비스 구성을 설명합니다.

사전 요구 사항

- **IdM** 호스트에서 실행되는 서비스에 대한 **IdM** 서비스 항목을 생성했습니다. **IdM 서비스 항목** 및 **Kerberos keytab** 생성을 참조하십시오.



주의

내부 **IdM** 서비스에 인증 표시기를 할당 하지 마십시오. 다음 **IdM** 서비스는 **PKINIT** 및 다단계 인증 방법에 필요한 대화형 인증 단계를 수행할 수 없습니다.

```
host/server.example.com@EXAMPLE.COM
HTTP/server.example.com@EXAMPLE.COM
ldap/server.example.com@EXAMPLE.COM
DNS/server.example.com@EXAMPLE.COM
cifs/server.example.com@EXAMPLE.COM
```

절차

- **ipa service-mod** 명령을 사용하여 **--auth-ind** 인수로 식별된 서비스에 대해 하나 이상의 필수 인증 표시기를 지정합니다.

인증 방법	--auth-ind 값
이중 인증	otp
RADIUS 인증	마케도니아
PKINIT, 스마트 카드 또는 인증서 인증	pkinit
강화된 암호 (SPAKE 또는 FAST)	강화

예를 들어 호스트 **client.example.com**에서 테스트 서비스 주체 에 대한 서비스 티켓을 검색하기 위해 사용자가 스마트 카드 또는 **OTP** 인증으로 인증되도록 하려면 다음을 수행합니다.

```
[root@server ~]# ipa service-mod testservice/client.example.com@EXAMPLE.COM --auth-ind otp --auth-ind pkinit
```

```
-----
Modified service "testservice/client.example.com@EXAMPLE.COM"
```

```
-----
Principal name: testservice/client.example.com@EXAMPLE.COM
Principal alias: testservice/client.example.com@EXAMPLE.COM
Authentication Indicators: otp, pkinit
Managed by: client.example.com
```

참고

서비스에서 모든 인증 표시기를 제거하려면 빈 표시기 목록을 제공합니다.

```
[root@server ~]# ipa service-mod
testservice/client.example.com@EXAMPLE.COM --auth-ind "
-----
Modified service "testservice/client.example.com@EXAMPLE.COM"
-----
Principal name: testservice/client.example.com@EXAMPLE.COM
Principal alias: testservice/client.example.com@EXAMPLE.COM
Managed by: client.example.com
```

검증 단계

- **ipa service-show** 명령을 사용하여 필요한 인증 표시기를 포함하여 **IdM** 서비스에 대한 정보를 표시합니다.

```
[root@server ~]# ipa service-show testservice/client.example.com
Principal name: testservice/client.example.com@EXAMPLE.COM
Principal alias: testservice/client.example.com@EXAMPLE.COM
Authentication Indicators: otp, pkinit
Keytab: True
Managed by: client.example.com
```

추가 리소스

- **IdM** 서비스에 대한 서비스 티켓 요청을 테스트하려면 **IdM** 서비스에 대한 **Kerberos** 서비스 티켓 검색을 참조하십시오.

42.4.3. IdM 서비스의 Kerberos 서비스 티켓 검색

다음 절차에서는 **IdM** 서비스에 대한 **Kerberos** 서비스 티켓을 검색하는 방법을 설명합니다. 이 절차를 사용하여 **Kerberos** 티켓 정책을 테스트할 수 있습니다.

사전 요구 사항

- 작업 중인 서비스가 내부 **IdM** 서비스가 아닌 경우 해당 **IdM** 서비스 항목이 생성되었습니다.

IdM 서비스 항목 및 Kerberos keytab 생성을 참조하십시오.

- **Kerberos** 티켓 부여 티켓(TGT)이 있습니다.

절차

- **kvno** 명령을 **-S** 옵션과 함께 사용하여 서비스 티켓을 검색하고 **IdM** 서비스의 이름과 이를 관리하는 호스트의 정규화된 도메인 이름을 지정합니다.

```
[root@server ~]# kvno -S testservice client.example.com
testservice/client.example.com@EXAMPLE.COM: kvno = 1
```

참고

IdM 서비스에 액세스해야 하며 현재 티켓 제공 티켓(TGT)에 연결된 필수 인증 표시기가 없는 경우 현재 **Kerberos** 자격 증명 캐시를 **kdestroy** 명령으로 지우고 새로운 TGT를 검색합니다.

```
[root@server ~]# kdestroy
```

예를 들어, 암호를 사용하여 인증하여 처음에 TGT를 검색하고 해당 인증서와 연결된 **pkinit** 인증 표시기가 있는 **IdM** 서비스에 액세스해야 하는 경우 현재 자격 증명 캐시를 삭제하고 스마트 카드로 다시 인증해야 합니다. **Kerberos 인증 표시기** 를 참조하십시오.

검증 단계

- **klist** 명령을 사용하여 서비스 티켓이 기본 **Kerberos** 자격 증명 캐시에 있는지 확인합니다.

```
[root@server etc]# klist_
Ticket cache: KCM:1000
Default principal: admin@EXAMPLE.COM

Valid starting    Expires          Service principal
04/01/2020 12:52:42 04/02/2020 12:52:39 krbtgt/EXAMPLE.COM@EXAMPLE.COM
04/01/2020 12:54:07 04/02/2020 12:52:39
testservice/client.example.com@EXAMPLE.COM
```

42.4.4. 추가 리소스

- **Kerberos** 인증 지표에 대한 자세한 내용은 **42.3절. “Kerberos 인증 표시기”**의 내용을 참조하십시오.

42.5. 글로벌 티켓 라이프사이클 정책 구성

글로벌 티켓 정책은 모든 서비스 티켓과 사용자별 티켓 정책이 정의되어 있지 않은 사용자에게 적용됩니다.

다음 절차에서는 **ipa krbtpolicy-mod** 명령을 사용하여 글로벌 **Kerberos** 티켓 정책에 대한 최대 티켓 수명 및 최대 티켓 갱신 기간을 조정하는 방법을 설명합니다.

ipa krbtpolicy-mod 명령을 사용하는 동안 다음 인수 중 하나 이상을 지정합니다.

- **--초 단위 최대 티켓 수명의 maxlife**
- **--maxrenew: 최대 재생 수명 (초)**

절차

1. 글로벌 티켓 정책을 수정하려면 다음을 수행합니다.

```
[root@server ~]# ipa krbtpolicy-mod --maxlife=$((8*60*60)) --maxrenew=$((24*60*60))
Max life: 28800
Max renew: 86400
```

이 예에서 최대 수명은 8시간($8 * 60분 * 60초$)으로 설정되고 최대 갱신 기간은 1일($24 * 60분 * 60초$)으로 설정됩니다.

2. 선택 사항: 글로벌 **Kerberos** 티켓 정책을 기본 설치 값으로 재설정하려면 다음을 수행합니다.

```
[root@server ~]# ipa krbtpolicy-reset
Max life: 86400
Max renew: 604800
```

검증 단계

- 글로벌 티켓 정책을 표시합니다.

■

```
[root@server ~]# ipa krbtpolicy-show
Max life: 28800
Max renew: 86640
```

추가 리소스

- 단일 사용자의 기본 티켓 정책을 조정하려면 사용자의 [기본 티켓 정책 구성을 참조하십시오](#).
- 단일 사용자의 각 인증 지표에 대한 개별 티켓 정책을 구성하려면 사용자의 [개별 인증 표시기 티켓 정책 구성을 참조하십시오](#).

42.6. 인증 표시기별로 글로벌 티켓 정책 구성

이 절차에서는 각 인증 지표에 대한 글로벌 최대 티켓 수명 및 최대 재생 기간을 조정하는 방법에 대해 설명합니다. 이러한 설정은 사용자당 티켓 정책이 정의되지 않은 사용자에게 적용됩니다.

ipa krbtpolicy-mod 명령을 사용하여 [연결된 인증](#) 표시자에 따라 Kerberos 티켓의 최대 수명 또는 최대 재생 기간을 지정합니다.

절차

- 예를 들어, 글로벌 2단계 티켓 수명과 갱신 기간 값을 1주로 설정하고 글로벌 스마트 카드 티켓 수명과 갱신 기간 값을 2주로 설정하려면 다음을 수행합니다.

```
[root@server ~]# ipa krbtpolicy-mod --otp-maxlife=604800 --otp-maxrenew=604800 --pkinit-maxlife=172800 --pkinit-maxrenew=172800
```

검증 단계

- 글로벌 티켓 정책을 표시합니다.

```
[root@server ~]# ipa krbtpolicy-show
Max life: 86400
OTP max life: 604800
PKINIT max life: 172800
Max renew: 604800
OTP max renew: 604800
PKINIT max renew: 172800
```

OTP 및 PKINIT 값은 글로벌 기본 최대 수명 및 최대 갱신 값과 다릅니다.

추가 리소스

- **ipa krbtpolicy-mod** 명령의 인증 표시기 옵션 목록은 **krbtpolicy-mod** 명령 [의 인증 표시기 옵션을 참조하십시오](#).
- 단일 사용자의 기본 티켓 정책을 조정하려면 사용자의 [기본 티켓 정책 구성을 참조하십시오](#).
- 단일 사용자의 각 인증 지표에 대한 개별 티켓 정책을 구성하려면 사용자의 [개별 인증 표시기 티켓 정책 구성을 참조하십시오](#).

42.7. 사용자의 기본 티켓 정책 구성

단일 사용자에게만 적용되는 고유한 **Kerberos** 티켓 정책을 정의할 수 있습니다. 이러한 사용자별 설정은 모든 인증 지표에 대한 글로벌 티켓 정책을 재정의합니다.

ipa krbtpolicy-mod *username* 명령을 사용하고 다음 인수 중 하나 이상을 지정합니다.

- **--초 단위 최대 티켓 수명의 maxlife**
- **--maxrenew: 최대 재생 수명 (초)**

절차

1. 예를 들어 **IdM** 관리자의 최대 티켓 수명 기간을 **2일**, 최대 갱신 기간을 **2주**로 설정하려면 다음을 수행합니다.

```
[root@server ~]# ipa krbtpolicy-mod admin --maxlife= 172800 --maxrenew= 1209600
Max life: 172800
Max renew: 1209600
```

2. 선택 사항: 사용자의 티켓 정책을 재설정하려면 다음을 수행합니다.

```
[root@server ~]# ipa krbtpolicy-reset admin
```

검증 단계

- 사용자에게 적용되는 유효 **Kerberos** 티켓 정책을 표시합니다.

```
[root@server ~]# ipa krbtpolicy-show admin
Max life: 172800
Max renew: 1209600
```

추가 리소스

- 모든 사용자의 글로벌 티켓 정책을 조정하려면 [글로벌 티켓 라이프사이클 정책 구성](#)을 참조하십시오.
- 인증 지표별로 서로 다른 기본 티켓 정책을 구성하려면 [인증당 글로벌 티켓 정책 구성](#)을 참조하십시오.

42.8. 사용자에게 대한 개별 인증 표시기 티켓 정책 구성

관리자는 인증 표시기별로 다른 사용자에게 대해 **Kerberos** 티켓 정책을 정의할 수 있습니다. 예를 들어 **IdM** 관리자가 **OTP** 인증을 통해 얻은 경우 **2일** 동안 티켓을 갱신할 수 있도록 정책을 구성하고 스마트 카드 인증을 통해 **1주일**에 걸쳐 티켓을 갱신할 수 있습니다.

이러한 인증별 표시기 설정은 *사용자의* 기본 티켓 정책, 글로벌 기본 티켓 정책 및 모든 글로벌 인증 표시 티켓 정책을 재정의합니다.

ipa krbtpolicy-mod *username* 명령을 사용하여 연결된 [인증 지표](#)에 따라 사용자 **Kerberos** 티켓의 최대 수명 및 최대 재생 기간 값을 설정합니다.

절차

1. 예를 들어 **IdM** 관리자가 일회성 암호 인증을 받은 경우 **Kerberos** 티켓을 이틀 동안 갱신하도록 허용하려면 **--otp-maxrenew** 옵션을 설정합니다.

```
[root@server ~]# ipa krbtpolicy-mod admin --otp-maxrenew=$((2*24*60*60))
OTP max renew: 172800
```

2. 선택 사항: 사용자의 티켓 정책을 재설정하려면 다음을 수행합니다.

```
[root@server ~]# ipa krbtpolicy-reset username
```

검증 단계

- 사용자에게 적용되는 유효 **Kerberos** 티켓 정책을 표시합니다.

```
[root@server ~]# ipa krbtpolicy-show admin
Max life: 28800
Max renew: 86640
```

추가 리소스

- **ipa krbtpolicy-mod** 명령의 인증 표시기 옵션 목록은 **krbtpolicy-mod** 명령 [의 인증 표시기 옵션을 참조하십시오](#).
- 단일 사용자의 기본 티켓 정책을 조정하려면 사용자의 [기본 티켓 정책 구성을 참조하십시오](#).
- 모든 사용자의 글로벌 티켓 정책을 조정하려면 [글로벌 티켓 라이프사이클 정책 구성을 참조하십시오](#).
- 인증 지표별로 다양한 글로벌 티켓 정책을 구성하려면 [인증당 글로벌 티켓 정책 구성을 참조하십시오](#).

42.9. KRBTPOLICY-MOD 명령의 인증 표시 옵션

다음 인수를 사용하여 인증 표시기 값을 지정합니다.

표 42.1. **krbtpolicy-mod** 명령의 인증 표시 옵션

인증 표시기	최대 수명에 대한 인수	최대 갱신 기간에 대한 인수
otp	--otp-maxlife	--otp-maxrenew
마케도니아	--radius-maxlife	--radius-maxrenew
pkinit	--pkinit-maxlife	--pkinit-maxrenew
강화	--hardened-maxlife	--hardened-maxrenew

[1]

강화된 암호는 단일 파티 공개 키-키 인증 키 교환(SPAKE) 사전 인증 및/또는 FAST(Secure 터널링)

를 통한 유연한 인증을 사용하여 무차별 암호 사전 공격으로부터 보호됩니다.

43장. IDM에서 KDC 프록시 사용

일부 관리자는 배포에 기본 **Kerberos** 포트에 액세스할 수 없도록 선택할 수 있습니다. 사용자, 호스트 및 서비스에서 **Kerberos** 자격 증명을 얻을 수 있도록 허용하기 위해 **HTTPS** 서비스를 **HTTPS** 포트 **443**을 통해 **Kerberos**와 통신하는 프록시로 사용할 수 있습니다.

IdM(Identity Management)에서 **KKDCP(Kerberos 키 배포 센터 프록시)**는 이 기능을 제공합니다.

IdM 서버에서 **KKDCP**는 기본적으로 활성화되어 있으며 <https://server.idm.example.com/KdcProxy>에서 사용할 수 있습니다. **IdM** 클라이언트에서 **KKDCP**에 액세스하도록 **Kerberos** 구성을 변경해야 합니다.

이 장에는 다음 섹션이 포함되어 있습니다.

- [KKDCP를 사용하도록 IdM 클라이언트 구성](#)
- [IdM 서버에서 KKDCP가 활성화되었는지 확인](#)
- [IdM 서버에서 KKDCP 비활성화](#)
- [IdM 서버에서 KKDCP 재활성화](#)

43.1. KKDCP를 사용하도록 IDM 클라이언트 구성

IdM(Identity Management) 시스템 관리자는 **IdM** 서버에서 **Kerberos KKDCP(키 배포 센터 프록시)**를 사용하도록 **IdM** 클라이언트를 구성할 수 있습니다. 이는 기본 **Kerberos** 포트에 **IdM** 서버에서 액세스할 수 없으며 **HTTPS** 포트 **443**이 **Kerberos** 서비스에 액세스하는 유일한 방법인 경우 유용합니다.

사전 요구 사항

- **IdM** 클라이언트에 대한 루트 액세스 권한이 있습니다.

절차

1. 편집을 위해 `/etc/krb5.conf` 파일을 엽니다.
2. `[realms]` 섹션에서 `kdc,admin_server, k passwd_server` 옵션에 대한 **KKDCP**의 URL을 입력합니다.

```
[realms]
EXAMPLE.COM = {
    kdc = https://kdc.example.com/KdcProxy
    admin_server = https://kdc.example.com/KdcProxy
    kpasswd_server = https://kdc.example.com/KdcProxy
    default_domain = example.com
}
```

중복성을 위해 `kdc,admin_server, k passwd_server` 매개 변수를 여러 번 추가하여 다른 **KKDCP** 서버를 나타낼 수 있습니다.

3. **sssd** 서비스를 다시 시작하여 변경 사항을 적용합니다.

```
~]# systemctl restart sssd
```

43.2. IDM 서버에서 **KKDCP**가 활성화되었는지 확인

IdM(Identity Management) 서버에서는 특성 및 값 쌍 `ipaConfigString=kdcProxyEnabled` 가 디렉터리에 존재하는 경우 **Apache** 웹 서버가 시작될 때마다 **KDCP**(Kerberos Key Distribution Center Proxy)가 자동으로 활성화됩니다. 이 경우에는 심볼릭 링크 `/etc/httpd/conf.d/ipa-kdc-proxy.conf` 가 생성됩니다.

권한이 없는 사용자로도 **IdM** 서버에서 **KKDCP**가 활성화되어 있는지 확인할 수 있습니다.

절차

- 심볼릭 링크가 있는지 확인합니다.

```
$ ls -l /etc/httpd/conf.d/ipa-kdc-proxy.conf
lrwxrwxrwx. 1 root root 36 Jun 21 2020 /etc/httpd/conf.d/ipa-kdc-proxy.conf ->
/etc/ipa/kdcproxy/ipa-kdc-proxy.conf
```

출력은 **KKDCP**가 활성화되었는지 확인합니다.

43.3. IDM 서버에서 KKDCP 비활성화

IdM(Identity Management) 시스템 관리자는 **IdM** 서버에서 **Kerberos KKDCP**(키 배포 센터 프록시)를 비활성화할 수 있습니다.

사전 요구 사항

- **IdM** 서버에 대한 루트 액세스 권한이 있습니다.

절차

1. 디렉터리에서 **ipaConfigString=kdcProxyEnabled** 속성 및 값을 제거합니다.

```
# ipa-ldap-updater /usr/share/ipa/kdcproxy-disable.uldif
Update complete
The ipa-lldap-updater command was successful
```

2. **httpd** 서비스를 다시 시작하십시오.

```
# systemctl restart httpd.service
```

KKDCP는 현재 **IdM** 서버에서 비활성화되어 있습니다.

검증 단계

- 심볼릭 링크가 없는지 확인합니다.

```
$ ls -l /etc/httpd/conf.d/ipa-kdc-proxy.conf
ls: cannot access '/etc/httpd/conf.d/ipa-kdc-proxy.conf': No such file or directory
```

43.4. IDM 서버에서 KKDCP 재활성화

IdM 서버에서 **KKDCP**(Kerberos 키 배포 센터 프록시)는 기본적으로 활성화되어 있으며 **https://server.idm.example.com/KdcProxy** 에서 사용할 수 있습니다.

KKDCP가 서버에서 비활성화된 경우 다시 활성화할 수 있습니다.

사전 요구 사항

- **IdM 서버에 대한 루트 액세스 권한이 있습니다.**

절차

1. **ipaConfigString=kdcProxyEnabled** 속성 및 값 쌍을 디렉터리에 추가합니다.

```
# ipa-lldap-updater /usr/share/ipa/kdcproxy-enable.uldif
Update complete
The ipa-lldap-updater command was successful
```

2. **httpd** 서비스를 다시 시작하십시오.

```
# systemctl restart httpd.service
```

KKDCP는 현재 **IdM** 서버에서 활성화되었습니다.

검증 단계

- 심볼릭 링크가 있는지 확인합니다.

```
$ ls -l /etc/httpd/conf.d/ipa-kdc-proxy.conf
lrwxrwxrwx. 1 root root 36 Jun 21 2020 /etc/httpd/conf.d/ipa-kdc-proxy.conf ->
/etc/ipa/kdcproxy/ipa-kdc-proxy.conf
```

43.5. 추가 리소스

- **Active Directory** 영역에 대한 **KKDCP** 구성에 대한 자세한 내용은 **Red Hat Knowledgebase**의 [AD Kerberos 통신을 위한 KDC 프록시로 IPA 서버 구성](#) 을 참조하십시오.

44장. IDM 암호 정책 정의

이 장에서는 **IdM(Identity Management)** 암호 정책과 **Ansible** 플레이북을 사용하여 **IdM**에 새 암호 정책을 추가하는 방법을 설명합니다.

44.1. 암호 정책이란 무엇입니까

암호 정책은 암호가 충족해야 하는 규칙 집합입니다. 예를 들어 암호 정책은 최소 암호 길이와 최대 암호 기간을 정의할 수 있습니다. 이 정책의 영향을 받는 모든 사용자는 충분히 긴 암호를 설정하고 지정된 조건을 충족하기 위해 암호를 자주 변경해야 합니다. 이러한 방식으로 암호 정책은 사용자의 암호를 검색하고 오용할 위험을 줄이는 데 도움이 됩니다.

44.2. IDM의 암호 정책

IdM(Identity Management) 사용자가 **IdM(Identity Management)** 도메인에 인증하는 가장 일반적인 방법입니다. 암호 정책은 이러한 **IdM** 사용자 암호를 충족해야 하는 요구 사항을 정의합니다.



참고

IdM 암호 정책은 기본 **LDAP** 디렉터리에 설정되지만 **Kerberos KDC**(키 배포 센터)는 암호 정책을 적용합니다.

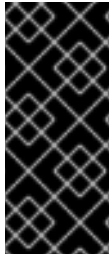
암호 정책 속성은 **IdM**에서 암호 정책을 정의하는 데 사용할 수 있는 속성을 나열합니다.

표 44.1. 암호 정책 속성

속성	설명	예제
최대 수명	암호가 유효한 최대 시간(사용자가 재설정해야 함).	최대 수명 = 90 사용자 암호는 90일 동안만 유효합니다. 그런 다음 IdM은 사용자에게 변경하라는 메시지를 표시합니다.
최소 수명 주기	두 개의 암호 변경 작업 간에 경과해야 하는 최소 시간(시간)입니다.	최소 수명 주기 = 1 사용자가 암호를 변경한 후 1시간 이상 기다린 후 다시 변경해야 합니다.

속성	설명	예제
기록 크기	저장된 이전 암호의 수입입니다. 사용자는 암호 기록에서 암호를 재사용할 수 없지만 저장되지 않은 이전 암호를 재사용할 수 있습니다.	기록 크기 = 0 이 경우 암호 기록이 비어 있으며 사용자는 이전 암호를 재사용할 수 있습니다.
문자 클래스	사용자가 암호에서 사용해야 하는 다른 문자 클래스의 수입입니다. 문자 클래스는 다음과 같습니다. * 대문자 * 소문자 * 숫자 * 쉼표 (,), 마침표 (.), 별표 (*)와 같은 특수 문자 * 기타 UTF-8 문자 행에서 세 번 이상 문자를 사용하면 문자 클래스가 1씩 줄어듭니다. 예를 들면 다음과 같습니다. * Secret1에는 대문자, 소문자, 숫자 등 3개 문자 클래스가 있습니다. * Secret111에는 두 개의 문자 클래스가 있습니다.: 대문자, 소문자, 숫자 및 1을 반복적으로 사용할 경우 -1 위약금입니다.	문자 클래스 = 0 필요한 기본 클래스 수는 0입니다. 번호를 구성하려면 ipa pwpolicy-mod 명령을 -minclasses 옵션과 함께 실행합니다. 이 표 아래에 있는 중요 참고 사항을 참조하십시오.
최소 길이	암호의 최소 문자 수입입니다. 추가 암호 정책 옵션이 설정된 경우 최소 길이 옵션이 설정된 값과 관계없이 최소 암호 길이는 6입니다.	최소 길이 = 8 사용자는 8자보다 짧은 암호를 사용할 수 없습니다.
최대 실패	IdM이 사용자 계정을 잠그기 전에 실패한 최대 로그인 횟수입니다.	최대 실패 = 6 IdM은 사용자가 잘못된 암호를 7번 입력하면 사용자 계정을 잠급니다.
실패 재설정 간격	IdM이 실패한 로그인 시도의 현재 수를 재설정 한 후 시간(초)입니다.	실패 재설정 간격 = 60 사용자가 Max 실패에 정의된 실패한 로그인 시도 횟수가 1분 이상 기다린 후 사용자는 사용자 계정 잠금의 위험 없이 다시 로그인을 시도할 수 있습니다.

속성	설명	예제
잠금 기간	최대 실패 에 정의된 실패한 로그인 횟수가 지난 후 사용자 계정이 잠긴 시간(초)입니다.	잠금 기간 = 600 잠긴 계정이 있는 사용자는 10분 동안 로그인할 수 없습니다.



중요

국제 문자와 기호에 액세스할 수 없는 다양한 하드웨어 세트가 있는 경우 문자 클래스 요구 사항에 영어 알파벳 및 공통 기호를 사용하십시오. 암호의 문자 클래스 정책에 대한 자세한 내용은 [Red Hat Knowledgebase의 암호에 유효한 문자](#)를 참조하십시오.

44.3. ANSIBLE 플레이북을 사용하여 IDM에 암호 정책이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 암호 정책이 있는지 확인하는 방법을 설명합니다.

IdM의 기본 **global_policy** 암호 정책에서 암호의 다양한 문자 클래스 수가 **0**으로 설정됩니다. 기록 크기도 **0**으로 설정되어 있습니다.

Ansible 플레이북을 사용하여 **IdM** 그룹에 대해 더 강력한 암호 정책을 적용하려면 이 절차를 완료합니다.



참고

IdM 그룹에 대한 암호 정책만 정의할 수 있습니다. 개별 사용자에게 대한 암호 정책을 정의할 수 없습니다.

사전 요구 사항

- **Ansible** 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.
- **IdM** 관리자 암호를 알고 있습니다.
- 암호 정책이 **IdM**에 있는지 확인하는 그룹입니다.

절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 `[ipaserver]` 섹션에 IdM 서버의 FQDN 을 정의합니다.

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

2. 확인할 암호 정책을 정의하는 **Ansible** 플레이북 파일을 만듭니다. 이 단계를 간소화하려면 `/usr/share/doc/ansible-freeipa/playbooks/pwpolicy/pwpolicy_present.yml` 파일에서 예제를 복사하고 수정합니다.

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

  tasks:
  - name: Ensure presence of pwpolicy for group ops
    ipapwpolicy:
      ipaadmin_password: MySecret123
      name: ops
      minlife: 7
      maxlife: 49
      history: 5
      priority: 1
      lockouttime: 300
      minlength: 8
      minclasses: 4
      maxfail: 3
      failinterval: 5
```

개별 변수의 의미에 대한 자세한 내용은 [암호 정책 속성](#)을 참조하십시오.

3. 플레이북을 실행합니다.

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

Ansible 플레이북을 사용하여 IdM에 **ops** 그룹의 암호 정책이 있는지 확인합니다.

중요

ops 암호 정책의 우선 순위는 **1**로 설정되어 있지만 **global_policy** 암호 정책에는 우선 순위가 설정되어 있지 않습니다. 이러한 이유로 **ops** 정책은 **ops** 그룹에 대해 **global_policy**를 자동으로 대체하며 즉시 적용됩니다.

global_policy는 사용자에 대한 그룹 정책이 설정되지 않은 경우 대체 정책으로 작동하며 그룹 정책보다 우선할 수 없습니다.

추가 리소스

- IdM 및 플레이북 변수에 대한 암호 정책을 정의하는 데 **Ansible**을 사용하는 방법에 대한 자세한 내용은 `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- pwpolicy.md** Markdown 파일을 참조하십시오.
- 암호 정책 우선 순위가 IdM에서 작동하는 방법에 대한 자세한 내용은 **RHEL 7** 설명서의 [암호 정책 우선 순위](#)를 참조하십시오.

44.4. IDM의 추가 암호 정책 옵션

IdM(Identity Management) 관리자는 **libpwquality** 기능 세트를 기반으로 추가 암호 정책 옵션을 활성화하여 기본 암호 요구 사항을 강화할 수 있습니다. 추가 암호 정책 옵션에는 다음이 포함됩니다.

--maxrepeat

새 암호에서 동일한 연속 문자의 최대 허용 가능한 수를 지정합니다.

--maxsequence

새 암호의 최대 단일 문자 시퀀스 길이를 지정합니다. 이러한 시퀀스의 예는 **12345** 또는 **fedcb**입니다. 이러한 암호는 대부분 단순성 검사를 통과하지 않습니다. 유일한 예외는 시퀀스가 암호의 일부일 때입니다.

--dictcheck

0이 아닌 경우 가능한 수정 가능한 암호를 사용하여 사전의 단어와 일치하는지 확인합니다. 현재 **libpwquality**는 **the cracklib** 라이브러리를 사용하여 사전 검사를 수행합니다.

--usercheck

0이 아니면 암호를 수정할 수 있는지 여부를 확인하고 일부 형식으로 사용자 이름을 포함합니다. **3**자 미만 사용자 이름에 대해서는 수행되지 않습니다.

추가 암호 정책 옵션은 기존 암호에 적용할 수 없습니다. 추가 옵션을 적용하는 경우 **IdM**은 암호의 최소 문자 수인 **--minlength** 옵션을 6자로 자동 설정합니다.



참고

RHEL 7 및 **RHEL 8** 서버와 혼합된 환경에서는 **RHEL 8.4** 이상에서 실행되는 서버에만 추가 암호 정책 설정을 적용할 수 있습니다. 사용자가 **IdM** 클라이언트에 로그인하고 **IdM** 클라이언트가 **RHEL 8.3** 이하에서 실행되는 **IdM** 서버와 통신하는 경우 시스템 관리자가 설정한 새 암호 정책 요구 사항이 적용되지 않습니다. 일관된 동작을 보장하려면 모든 서버를 **RHEL 8.4** 이상으로 업그레이드하거나 업데이트합니다.

추가 리소스:

- [IdM 그룹에 추가 암호 정책 적용](#)
- [pwquality\(3\) 도움말 페이지](#)

44.5. IDM 그룹에 추가 암호 정책 옵션 적용

이 섹션에서는 **IdM(Identity Management)**에 추가 암호 정책 옵션을 적용하는 방법을 설명합니다. 이 예제에서는 새 암호에 사용자의 각 사용자 이름이 포함되지 않고 암호에 연속해서 두 개 이상의 동일한 문자가 포함되어 있는지 확인하여 **managers** 그룹에 대한 암호 정책을 강화하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.
- **managers** 그룹은 **IdM**에 있습니다.
- **managers** 암호 정책은 **IdM**에 있습니다.

절차

1. **managers** 그룹의 사용자가 제안한 모든 새 암호에 사용자 이름 검사를 적용합니다.

```
$ ipa pwpolicy-mod --usercheck=True managers
```



참고

암호 정책의 이름을 지정하지 않으면 기본 **global_policy**가 수정됩니다.

2.

managers 암호 정책에서 동일한 연속 문자의 최대 수를 **2**로 설정합니다.

```
$ ipa pwpolicy-mod --maxrepeat=2 managers
```

2개 이상의 동일한 연속 문자가 포함된 경우 이제 암호가 허용되지 않습니다. 예를 들어 **eR873mUi111YJQ** 조합은 연속적으로 3개의 **1s**가 포함되어 있기 때문에 허용되지 않습니다.

검증

1.

test_user라는 테스트 사용자를 추가합니다.

```
$ ipa user-add test_user
First name: test
Last name: user
-----
Added user "test_user"
-----
```

2.

test 사용자를 **managers** 그룹에 추가합니다.

a.

IdM 웹 UI에서 **Identity** → **Groups(ID 그룹)** **User Groups (사용자 그룹)**를 클릭합니다.

b.

managers를 클릭합니다.

c.

추가를 클릭합니다.

d.

Add users into user group 'managers' 페이지에서 **test_user**를 확인합니다.

e.

> 화살표를 클릭하여 사용자를 **Prospective** 열로 이동합니다.

- f.
 - 추가를 클릭합니다.
3.
 - 테스트 사용자의 암호를 재설정합니다.
 - a.
 - Identity(ID) Users(사용자)** 로 이동합니다.
 - b.
 - test_user** 를 클릭합니다.
 - c.
 - Actions(작업)** 메뉴에서 **Reset Password(암호 재설정)**를 클릭합니다.
 - d.
 - 사용자의 임시 암호를 입력합니다.
4.
 - 명령줄에서 **test_user** 에 대한 **Kerberos** 티켓 부여 티켓(TGT)을 가져옵니다.

\$ kinit test_user

- a.
 - 임시 암호를 입력합니다.
- b.
 - 시스템에서 암호를 변경해야 함을 알려줍니다. 사용자 이름이 **test_user** 인 암호를 입력합니다.

Password expired. You must change it now.
Enter new password:
Enter it again:



참고

Kerberos에는 세부적인 오류 암호 정책 보고가 없으며 특정 경우에는 암호가 거부된 명확한 이유를 제공하지 않습니다.

- c.
 - 시스템은 입력한 암호가 거부되었음을 알려줍니다. 연속해서 **3개** 이상의 동일한 문자를 포함하는 암호를 입력합니다.

```

Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

```

```

Enter new password:
Enter it again:

```

d.

시스템은 입력한 암호가 거부되었음을 알려줍니다. **managers** 암호 정책의 기준을 충족하는 암호를 입력합니다.

```

Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

```

```

Enter new password:
Enter it again:

```

5.

가져온 TGT 보기:

```

$ klist
Ticket cache: KCM:0:33945
Default principal: test_user@IDM.EXAMPLE.COM

Valid starting    Expires          Service principal
07/07/2021 12:44:44 07/08/2021 12:44:44
krbtgt@IDM.EXAMPLE.COM@IDM.EXAMPLE.COM

```

managers 암호 정책이 **managers** 그룹의 사용자에게 대해 올바르게 작동합니다.

추가 리소스

•

[IdM의 추가 암호 정책](#)

45장. 암호 알림 만료 관리

ipa-client-epn 패키지에서 제공하는 **EPN(암호 만료 알림)** 도구를 사용하여 암호가 구성된 시간 내에 만료되는 **IdM(Identity Management)** 사용자 목록을 구축할 수 있습니다. **EPN** 도구를 설치, 구성 및 사용하려면 관련 섹션을 참조하십시오.

- [암호 만료 알림 도구란 무엇입니까?](#)
- [만료 암호 알림 도구 설치](#)
- [EPN 도구를 실행하여 암호가 만료되는 사용자에게 이메일 보내기](#)
- [암호가 만료되는 모든 사용자에게 `ipa-epn.timer`를 활성화](#)
- [만료 암호 알림 이메일 템플릿 수정](#)

45.1. 암호 만료 알림 도구란 무엇입니까?

EPN(Expiring Password Notification) 툴은 암호가 구성된 시간 내에 만료되는 **IdM(Identity Management)** 사용자 목록을 빌드하는 데 사용할 수 있는 독립 실행형 도구입니다.

IdM 관리자는 **EPN**을 사용하여 다음을 수행할 수 있습니다.

- 영향을 받는 사용자 목록을 **JSON** 형식으로 표시합니다. 이 형식은 시험 실행 모드로 실행될 때 생성됩니다.
- 지정된 날짜 또는 날짜 범위에 대해 전송할 이메일 수를 계산합니다.
- 암호 만료 이메일 알림 전송.
- **EPN** 도구를 매일 실행하고 암호가 정의된 향후 날짜 범위 내에 만료되는 사용자에게 이메일을 보내도록 **ipa-epn.timer** 을 구성합니다.

- 사용자에게 전송하도록 이메일 알림을 사용자 지정합니다.



참고

사용자 계정을 비활성화하면 암호가 만료될 경우 이메일 알림이 전송되지 않습니다.

45.2. 만료 암호 알림 도구 설치

다음 절차에서는 **EPN**(수명 암호 알림) 도구를 설치하는 방법을 설명합니다.

사전 요구 사항

- **IdM(Identity Management)** 복제본에 **EPN** 도구를 설치하거나 스마트 호스트로 구성된 로컬 **Postfix SMTP** 서버를 사용하여 **IdM** 클라이언트에 설치합니다.

절차

- **EPN** 도구를 설치합니다.

```
# yum install ipa-client-epn
```

45.3. EPN 도구를 실행하여 암호가 만료되는 사용자에게 이메일 보내기

다음 절차에서는 암호가 만료되는 사용자에게 이메일을 보내는 방법을 설명합니다.



참고

EPN 도구는 상태 비저장 도구입니다. **EPN** 도구가 지정된 날짜에 암호가 만료되는 사용자를 이메일을 보내지 못하면 **EPN** 틀에서 해당 사용자 목록을 저장하지 않습니다.

사전 요구 사항

- **ipa-client-epn** 패키지가 설치되어 있습니다. [암호 만료 알림 도구](#) 설치를 참조하십시오.
- 필요한 경우 **ipa-epn** 이메일 템플릿을 사용자 지정합니다. [만료 암호 알림 이메일 템플릿](#) 수정을 참조하십시오.

절차

1. **epn.conf** 구성 파일을 업데이트하여 **EPN** 도구에 대한 옵션을 설정하여 향후 암호 만료를 알립니다.

```
# vi /etc/ipa/epn.conf
```

2. 필요에 따라 **notify_ttls** 를 업데이트합니다. 기본값은 암호가 **28, 14, 7, 3** 및 **1**일 후에 만료되는 사용자에게 알리는 것입니다.

```
notify_ttls = 28, 14, 7, 3, 1
```

3. **SMTP** 서버 및 포트를 구성합니다.

```
smtp_server = localhost
smtp_port = 25
```

4. 이메일 만료 통지가 전송되는 이메일 주소를 지정합니다. 실패한 이메일은 이 주소로 반환됩니다.

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

5. **/etc/ipa/epn.conf** 파일을 저장합니다.

6. 시험 실행 모드에서 **EPN** 도구를 실행하여 **--dry-run** 옵션 없이 도구를 실행하는 경우 암호 만료 이메일 알림이 전송되는 사용자 목록을 생성합니다.

```
ipa-epn --dry-run
[
  {
    "uid": "user5",
    "cn": "user 5",
    "krbpasswordexpiration": "2020-04-17 15:51:53",
    "mail": "['user5@ipa.test']"
  }
]
[
  {
    "uid": "user6",
    "cn": "user 6",
```

```
"krbpasswordexpiration": "2020-12-17 15:51:53",
"mail": "['user5@ipa.test']"
}
]
The IPA-EPN command was successful
```



참고

반환된 사용자 목록이 매우 크고 **--dry-run** 옵션 없이 툴을 실행하면 이메일 서버에서 문제가 발생할 수 있습니다.

7.

시험 실행 모드에서 **EPN** 도구를 실행할 때 반환된 모든 사용자 목록에 만료 이메일을 보내려면 **--dry-run** 옵션을 사용하지 않고 **EPN** 도구를 실행합니다.

```
ipa-epn
[
{
"uid": "user5",
"cn": "user 5",
"krbpasswordexpiration": "2020-10-01 15:51:53",
"mail": "['user5@ipa.test']"
}
]
[
{
"uid": "user6",
"cn": "user 6",
"krbpasswordexpiration": "2020-12-17 15:51:53",
"mail": "['user5@ipa.test']"
}
]
The IPA-EPN command was successful
```

8.

모든 모니터링 시스템에 **EPN**을 추가하고 **--from-nbdays** 및 **--to-nbdays** 옵션으로 호출하여 특정 시간 프레임 내에 만료할 사용자 암호 수를 결정할 수 있습니다.

```
# ipa-epn --from-nbdays 8 --to-nbdays 12
```



참고

--from-nbdays 및 **--to-nbdays** 옵션을 사용하여 **EPN** 툴을 호출하면 시험 실행 모드에서 자동으로 실행됩니다.

검증 단계

- **EPN** 도구를 실행하고 이메일 알림이 전송되었는지 확인합니다.

추가 리소스

- **ipa-epn** 도움말 페이지를 참조하십시오.
- **epn.conf** 도움말 페이지를 참조하십시오.

45.4. 암호가 만료되는 모든 사용자에게 **IPA-EPN.TIMER**를 활성화

이 절차에서는 **ipa-epn.timer** 을 사용하여 암호가 만료되는 사용자에게 이메일을 보내는 방법을 설명합니다. **ipa-epn.timer** 는 **epn.conf** 파일을 구문 분석하고 해당 파일에 구성된 정의된 향후 날짜 범위 내에 암호가 만료되는 사용자에게 이메일을 전송합니다.

사전 요구 사항

- **ipa-client-epn** 패키지가 설치되어 있습니다. [만료되는 암호 알림 도구설치](#)를 참조하십시오.
- 필요한 경우 **ipa-epn** 이메일 템플릿을 사용자 지정합니다. [만료 암호 알림 이메일 템플릿 수정](#)을 참조하십시오.

절차

- **ipa-epn.timer** 를 시작합니다 :

```
systemctl start ipa-epn.timer
```

타이머를 시작하면 기본적으로 **EPN** 톨은 매일 오전 1시에 실행됩니다.

추가 리소스

- **ipa-epn** 도움말 페이지를 참조하십시오.

45.5. 만료 암호 알림 이메일 템플릿 수정

이 절차에서는 **EPN(Expiring Password Notification)** 이메일 메시지 템플릿을 사용자 지정하는 방법을 설명합니다.

사전 요구 사항

- **ipa-client-epn** 패키지가 설치되어 있습니다.

절차

1. **EPN** 메시지 템플릿을 엽니다.

```
# vi /etc/ipa/epn/expire_msg.template
```

2. 필요에 따라 템플릿 텍스트를 업데이트합니다.

```
Hi {{ fullname }},

Your password will expire on {{ expiration }}.

Please change it as soon as possible.
```

템플릿에서 다음 변수를 사용할 수 있습니다.

- 사용자 ID: **uid**
- 전체 이름: **fullname**
- 이름: **first**
- 성: **성**
- 암호 만료 날짜: **만료**

3.

메시지 템플릿 파일을 저장합니다.

검증 단계

-

EPN 도구를 실행하고 이메일 알림에 업데이트된 텍스트가 포함되어 있는지 확인합니다.

추가 리소스

-

ipa-epn 도움말 페이지를 참조하십시오.

46장. IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

46.1. IDM 클라이언트에서 SUDO 액세스

시스템 관리자는 루트가 아닌 사용자가 일반적으로 **root** 사용자에게 예약된 관리 명령을 실행할 수 있도록 **sudo** 액세스 권한을 부여할 수 있습니다. 따라서 사용자가 일반적으로 **root** 사용자로 예약된 관리 명령을 수행해야 하는 경우 **sudo** 를 사용하여 해당 명령 앞에 추가합니다. 암호를 입력한 후에는 루트 사용자인 것처럼 명령이 실행됩니다. 데이터베이스 서비스 계정과 같은 다른 사용자 또는 그룹으로 **sudo** 명령을 실행하려면 **sudo** 규칙에 대해 **RunAs** 별칭을 구성할 수 있습니다.

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

- 로컬로 **/etc/sudoers** 파일에서
- **IdM**에서 중앙 집중식으로

이 섹션에서는 **CLI(명령줄 인터페이스)** 및 **IdM 웹 UI**를 사용하여 **IdM** 클라이언트에 대한 중앙 **sudo** 규칙을 생성하는 방법을 설명합니다.

RHEL 8.4 이상에서는 **UNIX** 기반 운영 체제가 **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_gss.so** **PAM(Pluggable Authentication Module)**을 사용하여 **SSSD** 서비스를 통해 **GSSAPI** 인증을 호출할 수 있으므로 사용자는 유효한 **Kerberos** 티켓을 사용하여 **sudo** 명령을 인증할 수 있습니다.

추가 리소스

- **RHEL** 호스트에서 로컬 **sudo** 규칙 생성에 대한 자세한 내용은 [sudo 액세스 관리](#)를 참조하십시오.

46.2. CLI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

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

예를 들어 **idm_user** 계정에 **idm_sbin/reboot** 명령을 실행할 수 있는 권한을 부여하려면 **idm_user_reboot sudo** 규칙을 만들려면 다음 절차를 완료합니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.
- **IdM**에서 **idm_user**에 대한 사용자 계정을 생성하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. **CLI**를 사용하여 새 **IdM** 사용자를 추가하는 방법에 대한 자세한 내용은 [명령줄을 사용하여 사용자](#) 추가를 참조하십시오.
- **idm client** 호스트에 로컬 **idm_user** 계정이 없습니다. **idm_user** 사용자는 로컬 **/etc/passwd** 파일에 나열되지 않습니다.

절차

1. **IdM** 관리자로 **Kerberos** 티켓을 검색합니다.

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

2. **/usr/sbin/reboot** 명령을 **sudo** 명령의 **IdM** 데이터베이스에 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 idm client 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:
```

46.3. IDM 웹 UI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

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

idm_user_reboot sudo 규칙을 생성하여 **idm_user** 계정에 **idm client** 시스템에서 **/usr/sbin/reboot** 명령을 실행할 수 있는 권한을 부여하려면 이 절차를 완료합니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- IdM에서 **idm_user** 에 대한 사용자 계정을 생성하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. 명령줄 인터페이스를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄 을 사용하여 사용자 추가를 참조하십시오.
- **idm client** 호스트에 로컬 **idm_user** 계정이 없습니다. **idm_user** 사용자는 로컬 **/etc/passwd** 파일에 나열되지 않습니다.

절차

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

그림 46.1. 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. **추가**를 클릭합니다.
2. 새 **sudo** 명령 항목을 사용하여 **idm_user**가 **idm client** 시스템을 재부팅할 수 있도록 **sudo** 규칙을 생성합니다.
 - a. 정책 → **Sudo** → **Sudo** 규칙으로 이동합니다.
 - b. 오른쪽 상단 모서리에서 **Add**(추가)를 클릭하여 **Add sudo rule(sudo 규칙 추가)** 대화 상자를 엽니다.

- c. **sudo** 규칙의 이름을 **idm_user_reboot** 로 입력합니다.
- d. **Add and Edit**(추가 및 편집)를 클릭합니다.
- e. 사용자를 지정합니다.
 - i. **who**(사용자) 섹션에서 **Specified Users and Groups**(지정된 사용자 및 그룹) 라디오 단추를 선택합니다.
 - ii. **User** 카테고리에서 규칙이 하위 섹션에 적용되는 경우 **Add**(추가)를 클릭하여 **Add users into sudo rule "idm_user_reboot"** 대화 상자를 엽니다.
 - iii. **Add users into sudo rule "idm_user_reboot"** 대화 상자의 **Available** 열에서 **idm_user** 확인란을 선택하고 **Prospective** 열로 이동합니다.
 - iv. 추가를 클릭합니다.
- f. 호스트를 지정합니다.
 - i. **Access this host** (이 호스트에 액세스) 섹션에서 **Specified Hosts and Groups**(지정된 호스트 및 그룹) 라디오 단추를 선택합니다.
 - ii. 호스트 범주에서 이 규칙이 하위 섹션에 적용되는 경우 **Add**(추가)를 클릭하여 **Add hosts into sudo rule "idm_user_reboot"** 대화 상자를 엽니다.
 - iii. **Add hosts to sudo rule "idm_user_reboot"** 대화 상자의 **Available** 열에서 **idmclient.idm.example.com** 확인란을 선택하고 **Prospective** 열로 이동합니다.
 - iv. 추가를 클릭합니다.
- g. 명령을 지정합니다.

- i. **Command** 카테고리에서 규칙이 **Run Commands(명령 실행)** 섹션의 하위 섹션에 적용되는 경우 **Specified Commands and Groups(지정된 명령)** 및 **Groups (그룹)** 라디오 버튼을 선택합니다.
- ii. **Sudo Allow Commands (Sudo Allow Commands)** 하위 섹션에서 **Add allow sudo** 명령을 **sudo rule "idm_user_reboot"** 대화 상자에 엽니다.
- iii. **Add allow sudo** 명령을 **sudo** 규칙 "idm_user_reboot" 대화 상자의 **Available** 열에서 **/usr/sbin/reboot** 확인란을 선택하고 **Prospective** 열로 이동합니다.
- iv. **Add(추가)**를 클릭하여 **idm_sudo_reboot** 페이지로 돌아갑니다.

IdM sudo 규칙 추가

+ image::IdM-sudo-rule-WebUI.png[추가된 **sudo** 규칙 개요의 스크린샷입니다. 규칙이 적용되는 사용자 테이블이 있는 "Who" 섹션이 있습니다. 규칙이 적용되는 호스트 테이블이 있는 "이 호스트 액세스" 섹션이 있습니다. 규칙과 관련된 명령 테이블이 있는 "실행 명령" 섹션이 있습니다.]

- a. 왼쪽 상단 모서리에서 **Save(저장)**를 클릭합니다.

새 규칙은 기본적으로 활성화되어 있습니다.



참고

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

검증 단계

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 구성되면 시스템이 재부팅됩니다.

46.4. IDM 클라이언트에서 명령을 서비스 계정으로 실행하는 CLI에서 SUDO 규칙 생성

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

이 예제를 사용하면 **idm_user** 계정이 **idmclient** 호스트에서 타사 서비스 계정으로 **/opt/third-party-app/bin/report** 명령을 실행할 수 있도록 **run_third-party-app_report** 라는 명령줄에서 **sudo** 규칙을 만듭니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- IdM에서 **idm_user**에 대한 사용자 계정을 생성하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. CLI를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 [명령줄을 사용하여 사용자](#) 추가를 참조하십시오.
- **idm client** 호스트에 로컬 **idm_user** 계정이 없습니다. **idm_user** 사용자는 로컬 **/etc/passwd** 파일에 나열되지 않습니다.
- **idmclient** 호스트에 **third-party-app**이라는 사용자 지정 애플리케이션이 설치되어 있어야 합니다.
- **third-party-app** 애플리케이션의 **report** 명령은 **/opt/third-party-app/bin/report** 디렉터리에 설치됩니다.
- 타사-애플리케이션 애플리케이션에 대한 명령을 실행하기 위한 **third partyapp**이라는 로컬 서비스 계정을 생성했습니다.

절차

1. IdM 관리자로 Kerberos 티켓을 검색합니다.

■

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

2.

`/opt/third-party-app/bin/report` 명령을 `sudo` 명령의 IdM 데이터베이스에 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.

`run_third-party-app_report` 라는 `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.

`users =<user>` 옵션을 사용하여 `sudorule-add-runasuser` 명령에 `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 같이 IdM 외부에 지정된 사용자(또는 `--groups=*`)는 IdM 외부에 있을 수 있습니다. 그룹 이름에 대해 % 접두사를 추가하지 마십시오.

5.

`/opt/third-party-app/bin/report` 명령을 `idm_user_reboot` 규칙에 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.

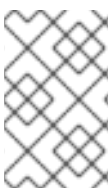
IdM `idmclient` 호스트에 `run_third-party-app_report` 규칙을 적용합니다.

```
[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.

`third partyapp` 서비스 계정으로 `report` 명령을 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.
```

46.5. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 IDM 웹UI에 SUDO 규칙 생성

IdM에서는 **RunAs** 별칭을 사용하여 **sudo** 규칙을 구성하여 **sudo** 명령을 다른 사용자 또는 그룹으로 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 `idm_user` 계정이 `idmclient` 호스트에서 타사 서비스 계정으로 `/opt/third-party-app/bin/report` 명령을 실행할 수 있도록 `run_third-party-app_report` 라는 IdM WebUI에 **sudo** 규칙을 생성합니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- IdM에서 `idm_user`에 대한 사용자 계정을 생성하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. CLI를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 [명령줄을 사용하여 사용자 추가](#)를 참조하십시오.

- **idm client** 호스트에 로컬 **idm_user** 계정이 없습니다. **idm_user** 사용자는 로컬 **/etc/passwd** 파일에 나열되지 않습니다.
- **idmclient** 호스트에 **third-party-app**이라는 사용자 지정 애플리케이션이 설치되어 있어야 합니다.
- **third-party-app** 애플리케이션의 **report** 명령은 **/opt/third-party-app/bin/report** 디렉터리에 설치됩니다.
- 타사-애플리케이션 애플리케이션에 대한 명령을 실행하기 위한 **third partyapp**이라는 로컬 서비스 계정을 생성했습니다.

절차

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

Add sudo command [X]

Sudo Command *

Description

* Required field

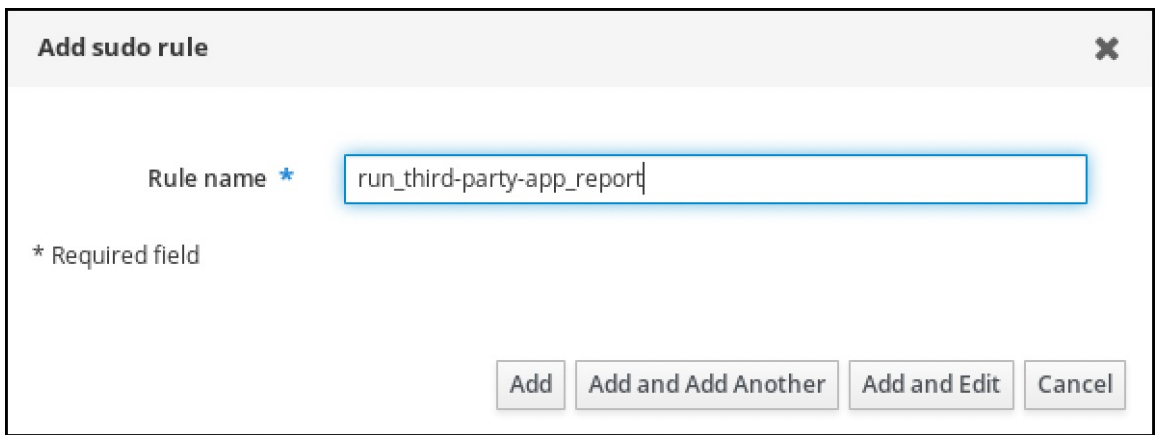
- d.
 추가를 클릭합니다.

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

- a.
 정책 → **Sudo** → **Sudo** 규칙으로 이동합니다.

- b.
 오른쪽 상단 모서리에서 **Add**(추가)를 클릭하여 **Add sudo rule(sudo 규칙 추가)** 대화 상자를 엽니다.

- c.
 sudo 규칙 이름 **run_third-party-app_report** 를 입력합니다.



- d.
 Add and Edit(추가 및 편집)를 클릭합니다.

- e.
 사용자를 지정합니다.

- i.
 who(사용자) 섹션에서 **Specified Users and Groups**(지정된 사용자 및 그룹) 라디오 단추를 선택합니다.

- ii.
 User(사용자) 카테고리에서 규칙이 하위 섹션에 적용되는 경우 **Add**(추가)를 클릭하여 **sudo** 규칙 "**run_third-party-app_report**" 대화 상자를 엽니다.

- iii.
 Available (사용 가능) 열의 **Add users into sudo** 규칙 "**run_third-party-app_report**" 대화 상자에서 **idm_user** 확인란을 선택한 다음 **Prospective** 열로 이동합니다.

Add users into sudo rule 'run_third-party-app_report'

Filter available Users Filter

Available

☐	Users
☐	admin

Prospective

☐	Users
☒	idmuser

External

Add Cancel

iv.

추가를 클릭합니다.

f.

호스트를 지정합니다.

i.

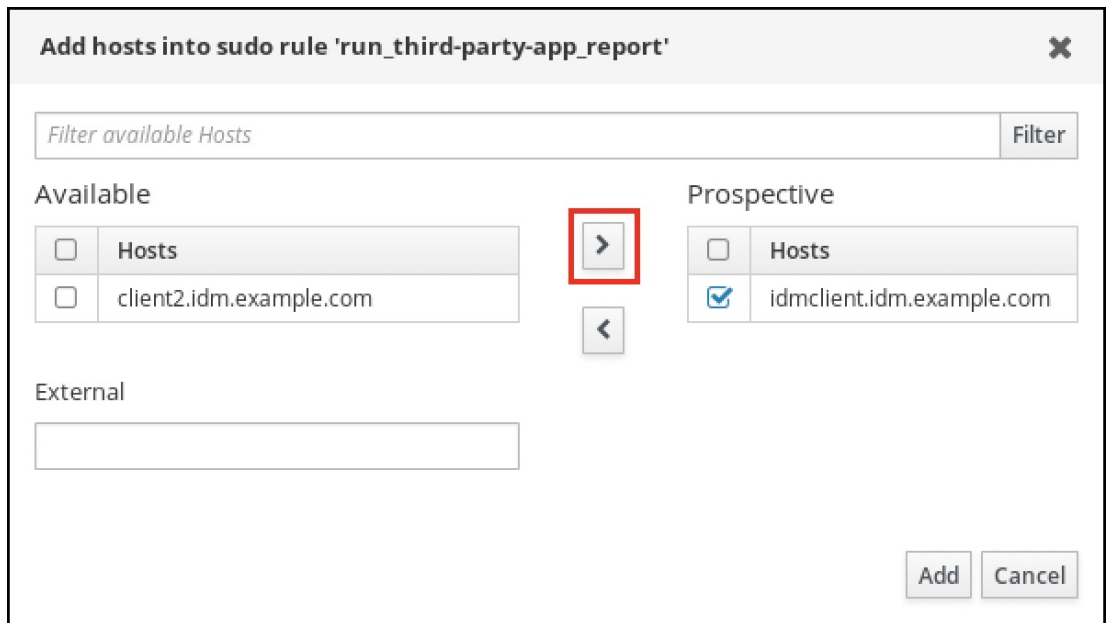
Access this host (이 호스트에 액세스) 섹션에서 **Specified Hosts and Groups**(지정된 호스트 및 그룹) 라디오 단추를 선택합니다.

ii.

이 규칙이 하위 섹션에 적용되는 호스트 범주에서 **Add**(추가)를 클릭하여 **sudo** 규칙 "run_third-party-app_report" 대화 상자를 엽니다.

iii.

Available (사용 가능) 열의 **Add hosts rule "run_third-party-app_report"** 대화 상자에서 **idmclient.idm.example.com** 확인란을 선택하여 **Prospective** 열로 이동합니다.



iv.

추가를 클릭합니다.

g.

명령을 지정합니다.

i.

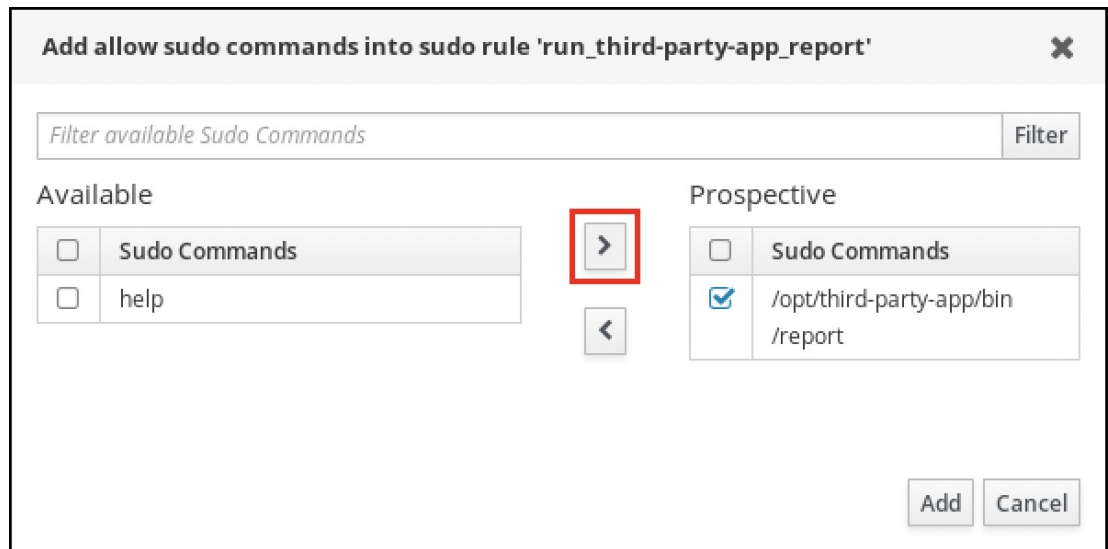
Command 카테고리에서 규칙이 **Run Commands**(명령 실행) 섹션의 하위 섹션에 적용되는 경우 **Specified Commands and Groups**(지정된 명령) 및 **Groups** (그룹) 라디오 버튼을 선택합니다.

ii.

Sudo Allow Commands 하위 섹션에서 **Add allow sudo** 명령을 **sudo** 규칙 "run_third-party-app_report" 대화 상자에 엽니다.

iii.

Add allow sudo 명령을 **sudo** 규칙 "run_third-party-app_report" 대화 상자에서 **Available** 열의 **/opt/third-party-app/bin/report** 확인란을 선택하여 **Prospective** 열로 이동합니다.



iv.

Add(추가)를 클릭하여 **run_third-party-app_report** 페이지로 돌아갑니다.

h.

RunAs 사용자를 지정합니다.

i.

As whom 섹션에서 **Specified Users and Groups**(지정된 사용자 및 그룹) 라디오 단추를 선택합니다.

ii.

RunAs Users(As 사용자 실행) 하위 섹션에서 **Add As**(추가)를 클릭하여 **AddAs users into sudo** 규칙 "**run_third-party-app_report**" 대화 상자를 엽니다.

iii.

RunAs 사용자를 **sudo** 규칙 "**run_third-party-app_report**" 대화 상자에서 **External** 상자에 **third partyapp** 서비스 계정을 입력하고 **Prospective** 열로 이동합니다.

Add RunAs users into sudo rule 'run_third-party-app_report'

Filter available Users Filter

Available

☐	Users
☐	admin
☐	employee
☐	helpdesk
☐	manager

Prospective

☐	Users

External

thirdpartyapp

Add Cancel

iv.

Add(추가)를 클릭하여 **run_third-party-app_report** 페이지로 돌아갑니다.

i.

왼쪽 상단 모서리에서 **Save(저장)**를 클릭합니다.

새 규칙은 기본적으로 활성화되어 있습니다.

그림 46.2. sudo 규칙의 세부 정보

Who

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

☐ Users	External	Delete + Add
☐ idm_user		

☐ User Groups Delete [+ Add](#)

Access this host

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

☐ Hosts	External	Delete + Add
☐ idmclient.idm.example.com		

☐ Host Groups Delete [+ Add](#)

Run Commands

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

Allow

☐ Sudo Allow Commands	Delete + Add
☐ /opt/third-party-app/bin/report	

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

Deny

☐ Sudo Deny Commands	Delete + Add
☐ Sudo Deny Command Groups	Delete + Add

As Whom

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

☐ RunAs Users	External	Delete + Add
☐ thirdpartyapp	True	

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

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

☐ RunAs Groups	External	Delete + Add



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.

third partyapp 서비스 계정으로 **report** 명령을 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.
```

46.6. IDM 클라이언트에서 SUDO에 대해 GSSAPI 인증 활성화

다음 절차에서는 **sudo** 및 **pam_sss_gss.so** PAM 모듈을 통해 **IdM** 클라이언트에서 **GSSAPI**(Generic Security Service Application Program Interface) 인증 활성화를 설명합니다. 이 구성을 사용하면 **IdM** 사용자가 **Kerberos** 티켓을 사용하여 **sudo** 명령에 인증할 수 있습니다.

사전 요구 사항

- **IdM** 호스트에 적용되는 **IdM** 사용자에게 **sudo** 규칙을 생성했습니다. 이 예제에서는 **idm_user** 계정에 **idm_sbin/reboot** 명령을 실행할 수 있는 권한을 부여하는 **idm_user_reboot sudo** 규칙을 생성했습니다.
- **idmclient** 호스트는 **RHEL 8.4** 이상을 실행하고 있습니다.

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

절차

1. **/etc/sss/sss.conf** 구성 파일을 엽니다.
2. 다음 항목을 **[domain/<domain_name>]** 섹션에 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** 파일에 있는 **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 닫습니다.
8. **/etc/pam.d/sudo-i** PAM 구성 파일을 엽니다.
9. 다음 항목을 **/etc/pam.d/sudo-i** 파일에 있는 **auth** 섹션의 첫 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.

티켓이 idm_user 계정으로 티켓을 부여했는지 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 삭제하고 올바른 자격 증명을 요청하십시오.

```
[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 UI**를 사용하여 **IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한 부여**
- **CLI**를 사용하여 **IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한 부여.**
- **pam_sss_gss (8)** 도움말 페이지
- **sssd.conf (5)** 도움말 페이지

46.7. GSSAPI 인증 활성화 및 IDM 클라이언트에서 SUDO에 대한 KERBEROS 인증 표시기 적용

다음 절차에서는 **sudo** 및 **pam_sss_gss.so** PAM 모듈을 통해 **IdM 클라이언트에서 GSSAPI(Generic Security Service Application Program Interface)** 인증 활성화를 설명합니다. 또한 스마트 카드로 로그인한 사용자만 **Kerberos** 티켓을 사용하여 해당 명령에 인증됩니다.



참고

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

사전 요구 사항

- **IdM 호스트에 적용되는 IdM 사용자에게 대한 sudo 규칙을 생성했습니다.** 이 예제에서는 **idm_user** 계정에 **idm_sbin/reboot** 명령을 실행할 수 있는 권한을 부여하는 **idm_user_reboot sudo** 규칙을 생성했습니다.
- **idmclient 호스트에 대해 스마트 카드 인증을 구성했습니다.**
- **idmclient 호스트는 RHEL 8.4 이상을 실행하고 있습니다.**
- **/etc/pam.d / 디렉토리에서 /etc/sss/sss.conf 파일과 PAM 파일을 수정하려면 root 권한이 필요합니다.**

절차

1. `/etc/sss/sss.conf` 구성 파일을 엽니다.
2. 다음 항목을 `[domain/<domain_name>]` 섹션에 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` 파일에 있는 `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 닫습니다.
8. `/etc/pam.d/sudo-i` PAM 구성 파일을 엽니다.
9. 다음 항목을 `/etc/pam.d/sudo-i` 파일에 있는 `auth` 섹션의 첫 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 티켓이 제공된지 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

- [IdM 용어 목록의 GSSAPI 항목](#)
- [스마트 카드 인증을 위한 ID 관리 구성](#)
- [Kerberos 인증 표시기](#)
- [IdM 웹 UI를 사용하여 IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한 부여](#)
- [CLI를 사용하여 IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한 부여.](#)
- [pam_sss_gss \(8\) 도움말 페이지](#)
- [sssd.conf \(5\) 도움말 페이지](#)

46.8. PAM 서비스에 대한 GSSAPI 인증을 제어하는 SSSD 옵션

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

`pam_gssapi_services`

SSSD를 사용한 GSSAPI 인증은 기본적으로 비활성화되어 있습니다. 이 옵션을 사용하여 `pam_ss_gss.so` PAM 모듈을 사용하여 GSSAPI 인증을 시도할 수 있는 PAM 서비스의 쉼표로 구분된 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 서비스에 대한 PAM 액세스 권한을 부여하는 데 필요한 Kerberos 인증 표시기를 나열합니다. 쌍은 `<PAM_service> :_<required_authentication_indicator>_` 형식이어야 합니다.

유효한 인증 표시기는 다음과 같습니다.

- 이중 인증을 위한 **OTP**
- **RADIUS** 인증을 위한 준비
- **PKINIT**, 스마트 카드 또는 인증서 인증을 위한 **Pk init**
- 강화된 암호를 위한 강화

pam_gssapi_check_upn

이 옵션은 활성화되어 있으며 기본적으로 **true** 로 설정됩니다. 이 옵션을 활성화하면 **SSSD** 서비스에 사용자 이름이 **Kerberos** 자격 증명과 일치해야 합니다. **false**인 경우 **pam_sss_gss.so** PAM 모듈은 필수 서비스 티켓을 가져올 수 있는 모든 사용자를 인증합니다.

예

다음 옵션을 사용하면 **sudo** 및 **sudo -i** 서비스에 **Kerberos** 인증을 사용하려면 **sudo** 사용자가 일회성 암호로 인증해야 하며 사용자 이름은 **Kerberos** 주체와 일치해야 합니다. 이러한 설정은 **[pam]** 섹션에 있으므로 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]** 섹션에 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** 명령에는 인증서 또는 스마트 카드 인증 인증기가 필요합니다.

- **sudo -i** 명령에 대해 일회성 암호 인증 프로그램이 필요합니다.
- 일치하는 사용자 이름과 **Kerberos** 주체 적용.

ad.example.com 도메인의 경우

- **sudo** 서비스에 대해서만 **GSSAPI** 인증을 활성화합니다.
- 일치하는 사용자 이름과 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 표시기](#)

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

절차

- 다음 오류가 표시되면 **Kerberos** 서비스에서 호스트 이름을 기반으로 서비스 티켓에 대한 올바른 영역을 확인할 수 없습니다.

Server not found in Kerberos database

이 경우 `/etc/krb5.conf` **Kerberos** 구성 파일의 `[domain_realm]` 섹션에 직접 호스트 이름을 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** 인증 정보가 없습니다.

No Kerberos credentials available

이 경우 `kinit` 유틸리티를 사용하여 **Kerberos** 인증 정보를 검색하거나 **SSSD**로 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 중 하나가 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**로 인증했는지 또는 `/etc/sss/sssd.conf` 파일에서 `pam_gssapi_check_upn` 옵션을 비활성화하는 것이 좋습니다.

```
[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.
- PAM 파일의 모든 **pam_sss_gss.so** 항목(예: **/etc/pam.d/ sudo** 및 **/etc/pam.d/sudo -i**)의 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 예제에서 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
```

46.10. ANSIBLE 플레이북을 사용하여 IDM 클라이언트에서 IDM 사용자에게 SUDO 액세스 권한 확인

IdM(Identity Management)에서는 특정 명령에 대한 **sudo** 액세스 권한이 특정 IdM 호스트의 IdM 사용

자 계정에 부여되도록 할 수 있습니다.

`idm_user_reboot` 라는 `sudo` 규칙이 있는지 확인하려면 다음 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.
- IdM에 `idm_user` 의 사용자 계정이 있는지 확인하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. 명령줄 인터페이스를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄 을 사용하여 사용자 추가를 참조하십시오.
- `idm client` 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 여기에 `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 간소화하기 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-on-idmclient-is-present.yml** Ansible 플레이북을 만듭니다. **sudo** 규칙을 사용하면 **idm_user**가 **idmclient** 시스템을 재부팅할 수 있습니다. 이 단계를 간소화하기 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_user가 **sudo**를 사용하여 **idmclient**를 재부팅할 수 있는지 확인하여 **IdM** 서버에서 **IdM** 서버가 작동하는지 확인하는 **sudo** 규칙이 **idmclient**에서 작동하는지 테스트합니다. 서버에서 변경한 사항이 클

라이언트에 적용되는 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 가 올바르게 구성되면 시스템이 재부팅됩니다.

추가 자료

- 플레이북 변수에 대한 설명을 포함하는 **Ansible** 플레이북을 사용하여 **sudo** 명령, 명령 그룹 및 규칙을 적용하는 방법에 대한 자세한 내용은 **README-sudocmd.md**, **README-sudocmdgroup.md** 및 **/usr/share/doc/ansible-freeipa/** 디렉터리에 있는 **README-sudorule.md** Markdown 파일을 참조하십시오.

47장. ANSIBLE 플레이북을 사용하여 IDM에 호스트 기반 액세스 제어 규칙이 있는지 확인

이 장에서는 **IdM(Identity Management)** 호스트 기반 액세스 정책과 **Ansible** 을 사용하여 정의하는 방법에 대해 설명합니다.

Ansible은 시스템을 구성하고, 소프트웨어를 배포하고, 롤링 업데이트를 수행하는 데 사용되는 자동화 도구입니다. **IdM(Identity Management)** 지원이 포함되어 있습니다.

47.1. IDM의 호스트 기반 액세스 제어 규칙

호스트 기반 액세스 제어(**HBAC**) 규칙은 서비스 그룹의 서비스 또는 서비스를 사용하여 어떤 호스트나 호스트 그룹에 액세스할 수 있는 사용자 또는 사용자 그룹을 정의합니다. 시스템 관리자는 **HBAC** 규칙을 사용하여 다음 목표를 달성할 수 있습니다.

- 도메인의 지정된 시스템에 대한 액세스를 특정 사용자 그룹의 멤버로 제한합니다.
- 특정 서비스만 도메인의 시스템에 액세스하도록 허용합니다.

기본적으로 **IdM**은 **allow_all** 이라는 기본 **HBAC** 규칙으로 구성되며, 이는 전체 **IdM** 도메인의 모든 관련 서비스를 통해 모든 사용자에게 대한 모든 호스트에 대한 범용 액세스를 의미합니다.

기본 **allow_all** 규칙을 자체 **HBAC** 규칙 세트로 교체하여 다른 호스트에 대한 액세스를 미세 조정할 수 있습니다. 중앙 집중적이고 단순화된 액세스 제어 관리를 위해 개별 사용자, 호스트 또는 서비스 대신 사용자 그룹, 호스트 그룹 또는 서비스 그룹에 **HBAC** 규칙을 적용할 수 있습니다.

47.2. ANSIBLE 플레이북을 사용하여 IDM에 HBAC 규칙이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 **HBAC**(호스트 기반 액세스 제어) 규칙이 있는지 확인하는 방법에 대해 설명합니다.

사전 요구 사항

- **ansible-freeipa** 패키지는 **Ansible** 컨트롤러에 설치됩니다.

- **IdM 관리자 암호를 알고 있습니다.**
- **HBAC 규칙에 사용할 사용자 및 사용자 그룹이 IdM에 있습니다.** 자세한 내용은 [Ansible 플레이북을 사용하여 사용자 계정 관리](#) 및 [Ansible 플레이북을 사용하여 IdM 그룹 및 그룹 구성원이 있는지 확인](#) 하십시오.
- **HBAC 규칙을 적용하려는 호스트 및 호스트 그룹은 IdM에 있습니다.** 자세한 내용은 [Ansible 플레이북을 사용하여 호스트 관리](#) 및 [Ansible 플레이북을 사용하여 호스트 관리를 참조](#) 하십시오.

절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 여기에 `ipaserver` 를 정의합니다.

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

2. 확인할 **HBAC** 정책을 정의하는 **Ansible** 플레이북 파일을 만듭니다. 이 단계를 간소화하기 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. **IdM 웹 UI에 관리자로 로그인합니다.**
2. **정책 → 호스트 기반 액세스 제어 → HBAC 테스트로 이동합니다.**
3. **who(사용자) 탭에서 idm_user를 선택합니다.**
4. **Accessing(액세스) 탭에서 client.idm.example.com 을 선택합니다.**
5. **Via service 탭에서 sshd 를 선택합니다.**
6. **Rules(규칙) 탭에서 로그인을 선택합니다.**
7. **Run test(테스트 실행) 탭에서 Run test(테스트 실행) 단추를 클릭합니다. ACCESS GRANTED가 표시되면 HBAC 규칙이 성공적으로 구현됩니다.**

추가 리소스

- **Ansible을 사용하여 HBAC 서비스, 서비스 그룹 및 규칙을 구성하는 방법에 대한 자세한 내용은 README-hbacsvc.md, README-hbacsvgroup.md, README-hbacrule.md Markdown 파일을 참조하십시오. 이러한 파일은 /usr/share/doc/ansible-freeipa 디렉토리에서 사용할 수 있습니다. /usr/share/doc/ansible-freeipa/playbooks 디렉토리의 관련 하위 디렉토리에서 사용할 수 있는 플레이북도 참조하십시오.**

48장. 복제 토폴로지 관리

이 장에서는 **IdM(Identity Management)** 도메인의 서버 간 복제를 관리하는 방법을 설명합니다.

48.1. 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트 설명

이 섹션에서는 다음 개념을 설명합니다.

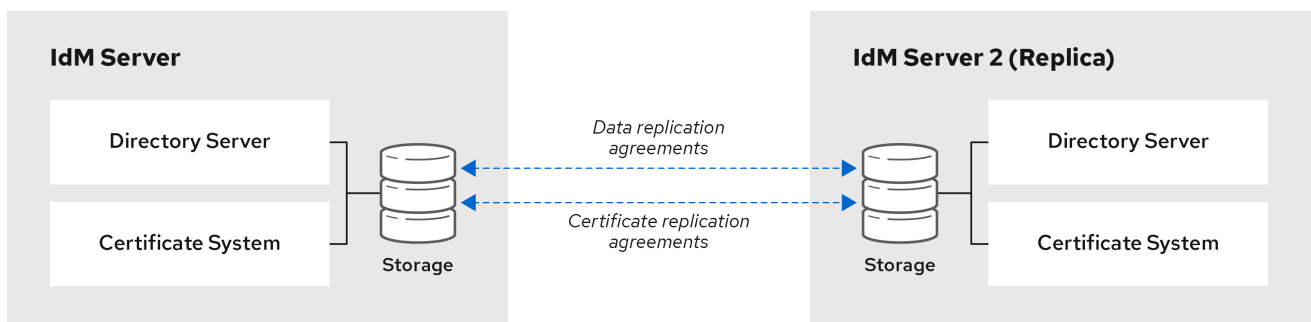
- 복제 계약
- 토폴로지 접미사
- 토폴로지 세그먼트

복제 계약

관리자가 기존 서버를 기반으로 복제본을 만들면 **IdM(Identity Management)**에서 초기 서버와 복제본 간에 **복제 계약**을 생성합니다. 복제 계약을 통해 두 서버 간에 데이터와 구성이 지속적으로 복제됩니다.

IdM은 여러 읽기/쓰기 복제본 복제를 사용합니다. 이 구성에서는 복제 계약에 가입된 모든 복제본은 업데이트를 수신하고 제공하므로 공급업체와 소비자 간주됩니다. 복제 계약은 항상 양방향입니다.

그림 48.1. 서버 및 복제본 계약



64_RHEL_0120

IdM은 다음 두 가지 유형의 복제 계약을 사용합니다.

도메인 복제 계약

이러한 계약은 **ID** 정보를 복제합니다.

인증서 복제 계약

이러한 계약은 인증서 정보를 복제합니다.

두 복제 채널은 모두 독립적입니다. 두 서버에는 둘 사이의 복제 계약 유형이 하나 또는 둘 다 있을 수 있습니다. 예를 들어 서버 **A**와 서버 **B**에 도메인 복제 계약만 구성된 경우 인증서 정보가 아닌 **ID** 정보만 복제됩니다.

토폴로지 접미사

*토폴로지 접미사*는 복제된 데이터를 저장합니다. **IdM**은 두 가지 유형의 토폴로지 접미사인 **domain**과 **ca**를 지원합니다. 각 접미사는 별도의 복제 토폴로지인 별도의 서버인 을 나타냅니다.

복제 계약이 구성되면 서로 다른 두 서버에서 동일한 유형의 두 토폴로지 접미사를 결합합니다.

도메인 접미사: **dc=example,dc=com**

도메인 접미사에는 모든 도메인 관련 데이터가 포함됩니다.

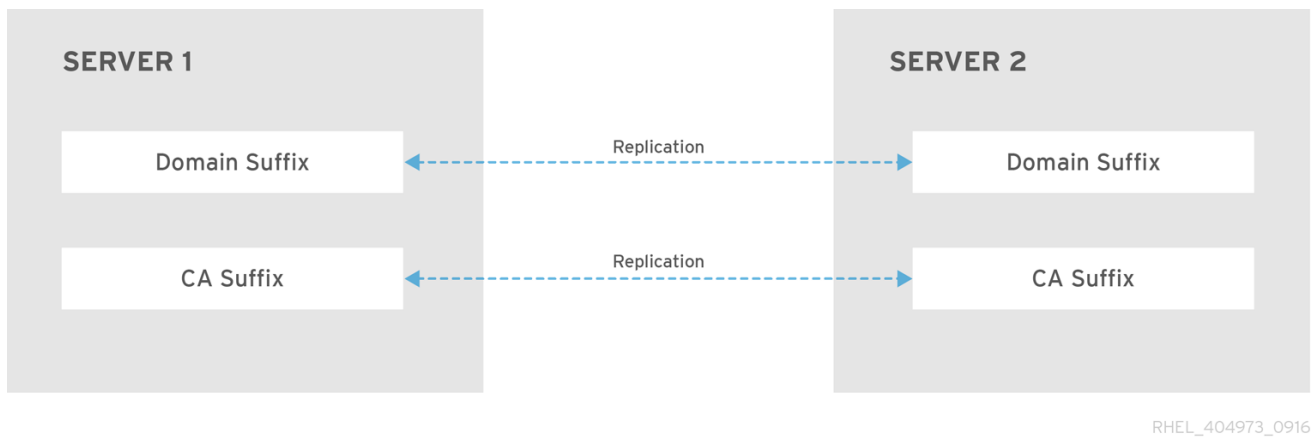
두 복제본에 도메인 접미사 간 복제 계약이 있는 경우 사용자, 그룹 및 정책과 같은 디렉터리 데이터를 공유합니다.

ca 접미사: **o=ipaca**

ca 접미사에는 **Certificate System** 구성 요소에 대한 데이터가 포함되어 있습니다. **CA**(인증 기관)가 설치된 서버에만 존재합니다.

두 복제본에 **ca** 접미사 간 복제 계약이 있는 경우 인증서 데이터를 공유합니다.

그림 48.2. 토폴로지 접미사



초기 토폴로지 복제 계약은 새 복제본을 설치할 때 **ipa-replica-install** 스크립트에 의해 두 서버 간에 설정됩니다.

예 48.1. 토폴로지 접미사 보기

ipa topologysuffix-find 명령은 토폴로지 접미사 목록을 표시합니다.

```
$ ipa topologysuffix-find
-----
2 topology suffixes matched
-----
Suffix name: ca
Managed LDAP suffix DN: o=ipaca

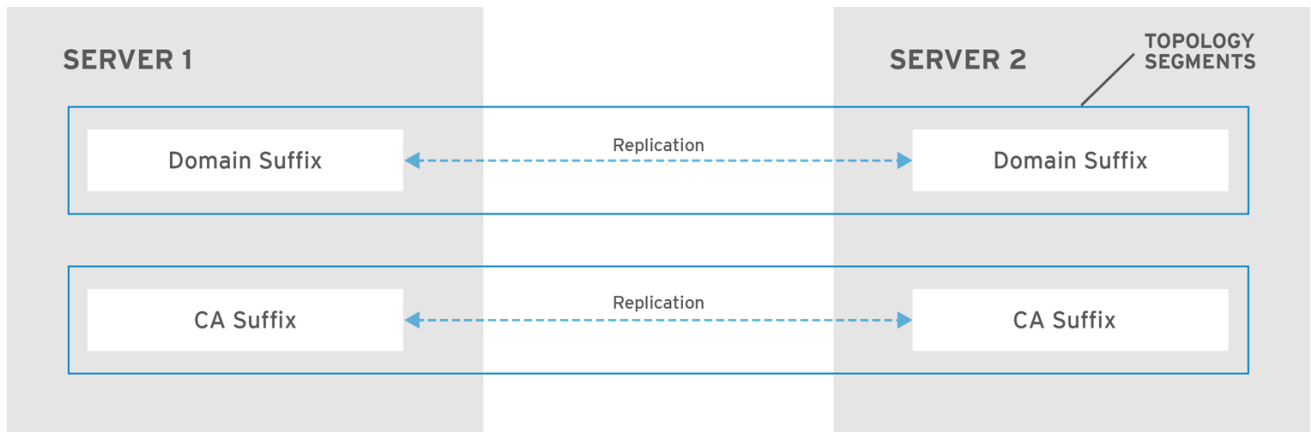
Suffix name: domain
Managed LDAP suffix DN: dc=example,dc=com
-----
Number of entries returned 2
-----
```

토폴로지 세그먼트

두 복제본에 접미사 간 복제 계약이 있는 경우 접미사는 **토폴로지 세그먼트**를 형성합니다. 각 토폴로지 세그먼트는 **왼쪽 노드**와 **오른쪽 노드**로 구성됩니다. 노드는 복제 계약에 가입된 서버를 나타냅니다.

IdM의 토폴로지 세그먼트는 항상 양방향입니다. 각 세그먼트는 서버 **A**에서 서버 **B**로, 서버 **B**에서 서버 **A**로의 두 가지 복제 계약을 나타냅니다. 따라서 데이터는 두 방향으로 복제됩니다.

그림 48.3. 토폴로지 세그먼트



RHEL_404973_0916

예 48.2. 토폴로지 세그먼트 보기

ipa topologysegment-find 명령은 도메인 또는 **CA** 접미사에 대해 구성된 현재 토폴로지 세그먼트를 표시합니다. 예를 들어 **domain** 접미사는 다음과 같습니다.

```
$ ipa topologysegment-find
Suffix name: domain
-----
1 segment matched
-----
Segment name: server1.example.com-to-server2.example.com
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
-----
Number of entries returned 1
-----
```

이 예제에서는 도메인 관련 데이터는 **server1.example.com** 및 **server 2.example.com** 두 서버 간에만 복제됩니다.

특정 세그먼트에 대한 세부 정보만 표시하려면 **ipa topologysegment-show** 명령을 사용합니다.

```
$ ipa topologysegment-show
Suffix name: domain
Segment name: server1.example.com-to-server2.example.com
Segment name: server1.example.com-to-server2.example.com
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
```

48.2. 토폴로지 그래프를 사용하여 복제 토폴로지 관리

웹 UI의 토폴로지 그래프는 도메인의 서버 간 관계를 보여줍니다. 웹 UI를 사용하여 토폴로지의 표현을 조작하고 변환할 수 있습니다.

토폴로지 그래프 액세스

토폴로지 그래프에 액세스하려면 다음을 수행합니다.

1. **IPA Server Topology Topology Graph(IPA 서버토폴로지 토폴로지 그래프)** 를 선택합니다.
2. 그래프에 즉시 반영되지 않은 토폴로지를 변경한 경우 새로 고침 을 클릭합니다.

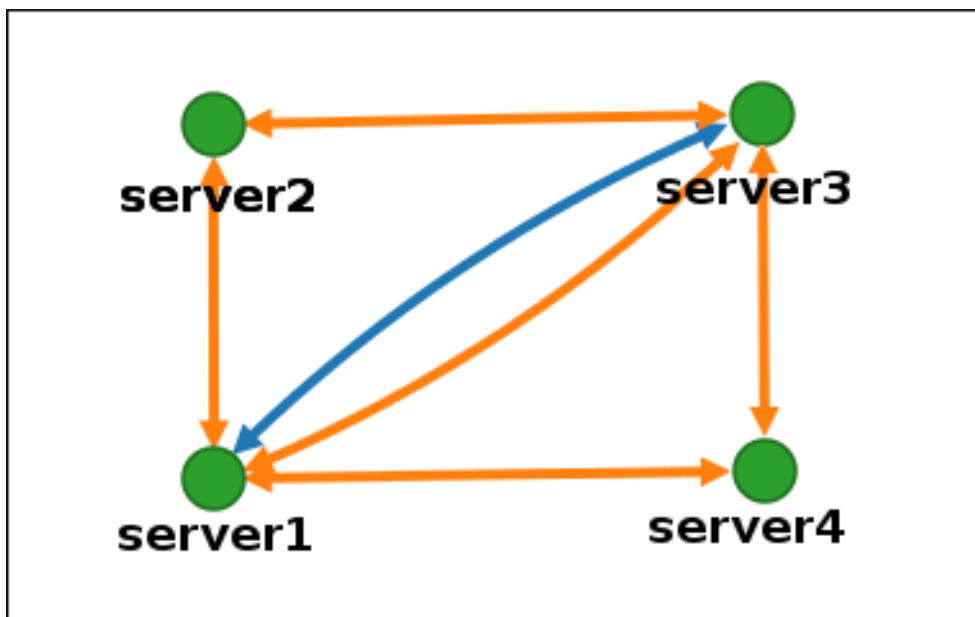
토폴로지 그래프 해석

도메인 복제 계약에 연결된 서버는 주황색 화살표를 통해 연결됩니다. CA 복제 계약에 가입된 서버는 파란색 화살표를 통해 연결됩니다.

토폴로지 그래프 예: 권장 토폴로지

[fig.mng-top-rec](#) 는 네 대의 서버에 권장되는 토폴로지 중 하나를 보여줍니다. 각 서버는 다른 두 대 이상의 서버에 연결되어 있고, 둘 이상의 서버가 CA 마스터입니다.

그림 48.4. 권장 토폴로지 예



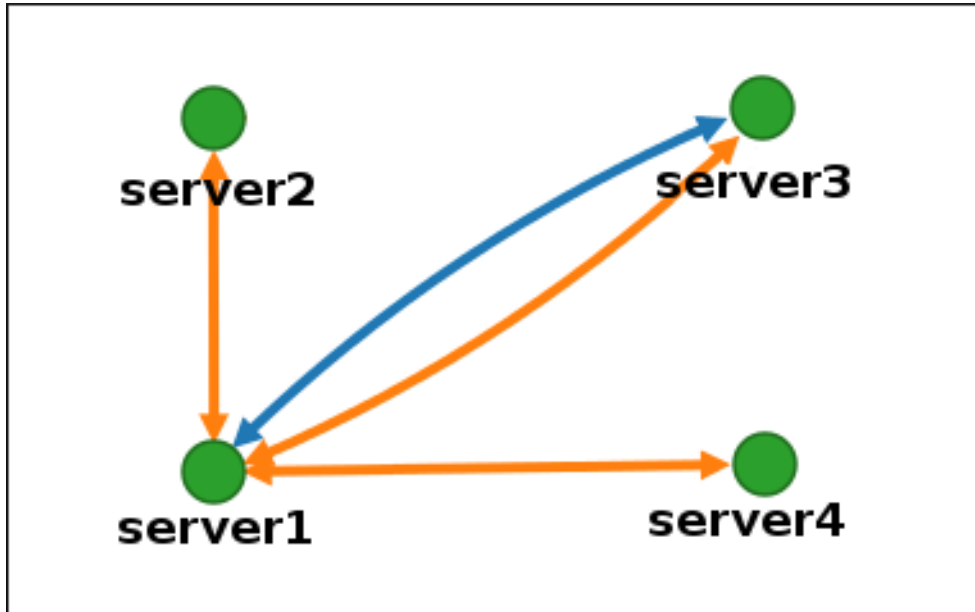
토폴로지 그래프 예: 권장되는 토폴로지

[fig.mng-top-single](#) 에서 **server1** 은 단일 오류 지점입니다. 다른 모든 서버는 이 서버와 복제 계약

을 맺지만 다른 서버와는 호환되지 않습니다. 따라서 **server1** 이 실패하면 다른 모든 서버가 격리됩니다.

다음과 같은 토폴로지를 생성하지 마십시오.

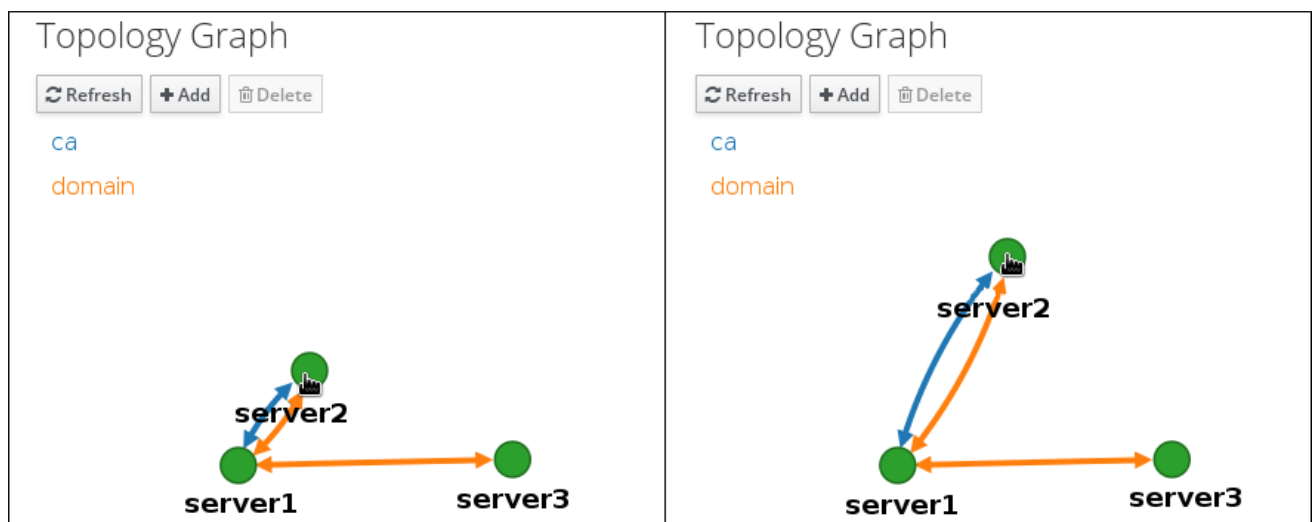
그림 48.5. 권장되지 않는 토폴로지 예: 단일 오류 지점



토폴로지 보기 사용자 정의

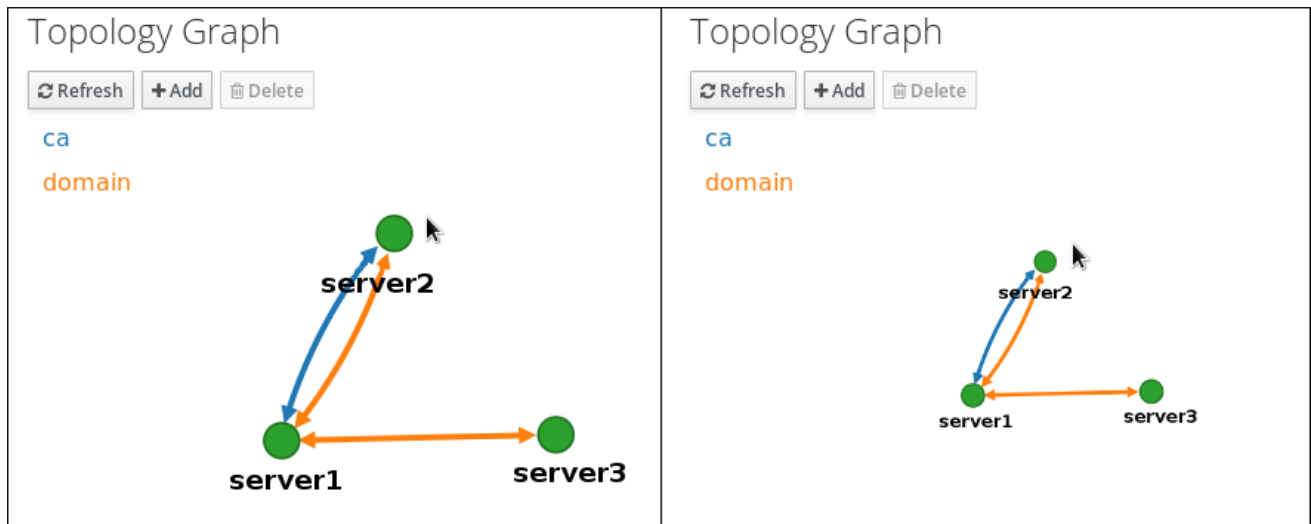
마우스를 드래그하여 개별 토폴로지 노드를 이동할 수 있습니다.

그림 48.6. 토폴로지 그래프 노드 이동



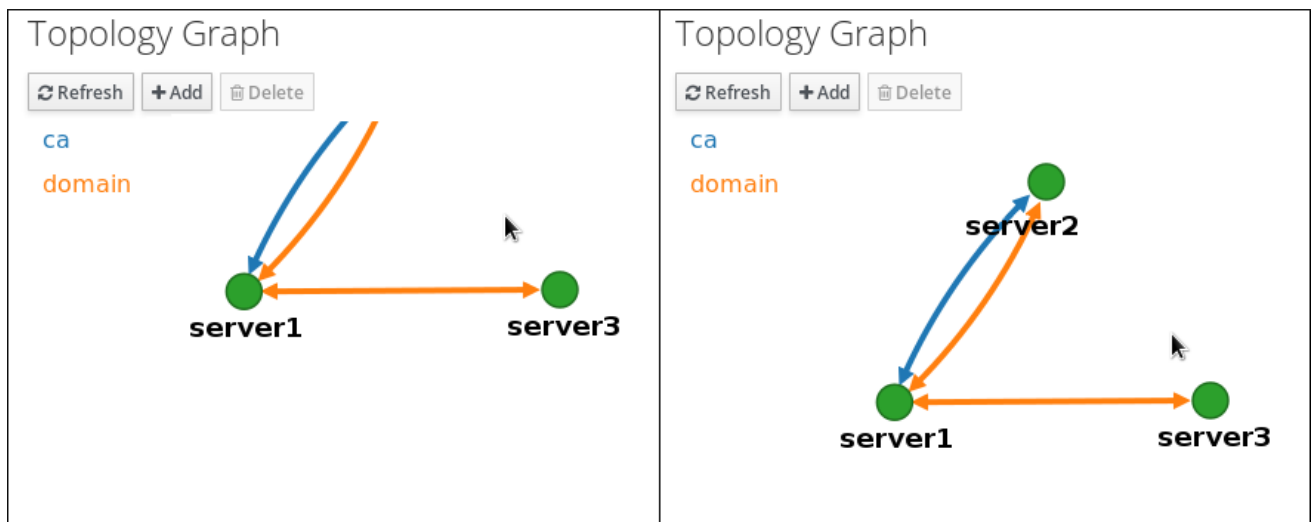
마우스바퀴를 사용하여 토폴로지 그래프를 확대하고 축소할 수 있습니다.

그림 48.7. 토폴로지 그래프 확대



왼쪽 마우스 버튼을 눌러 토폴로지 그래프의 팔레트를 이동할 수 있습니다.

그림 48.8. 토폴로지 그래프 팔레트 이동



48.3. 웹 UI를 사용하여 두 서버 간 복제 설정

IdM(Identity Management)의 웹 인터페이스를 사용하여 두 개의 서버를 선택하고 그 사이에 새 복제 계약을 만들 수 있습니다.

사전 요구 사항

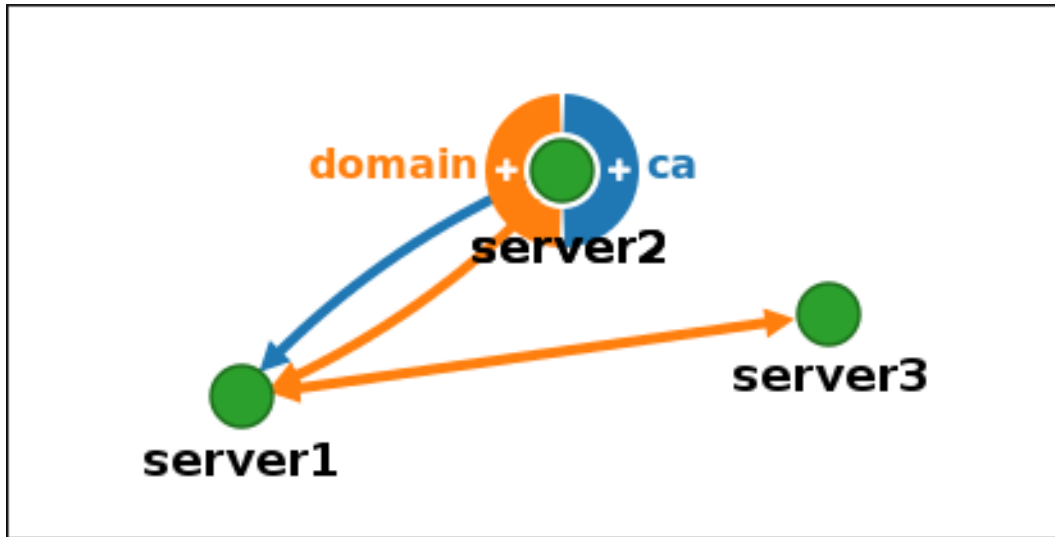
- **IdM 관리자 자격 증명**이 있습니다.

절차

1.

토폴로지 그래프에서 마우스를 서버 노드 중 하나 위로 가리킵니다.

그림 48.9. 도메인 또는 CA 옵션



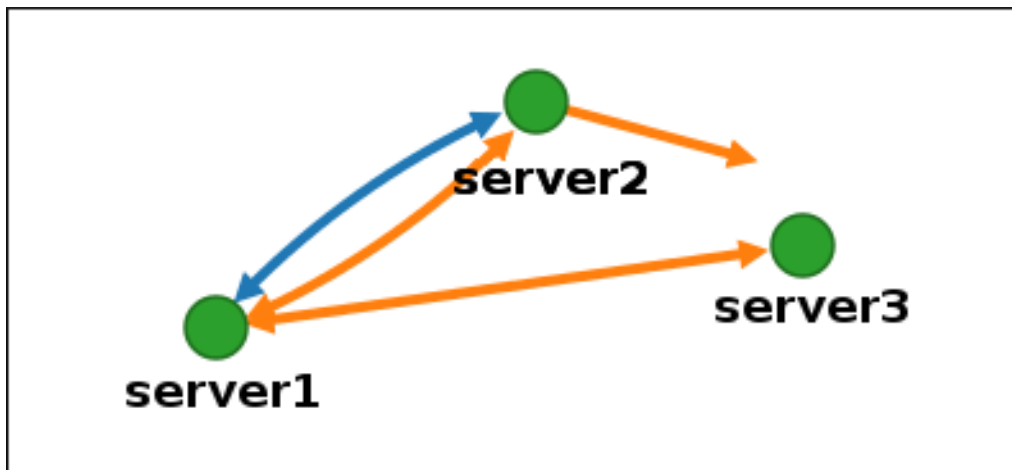
2.

생성할 토폴로지 세그먼트 유형에 따라 동그라미의 도메인 또는 **ca** 부분을 클릭합니다.

3.

새 복제 계약을 나타내는 새 화살표가 마우스 포인터 아래에 표시됩니다. 마우스를 다른 서버 노드로 이동하고 클릭합니다.

그림 48.10. 새 세그먼트 생성

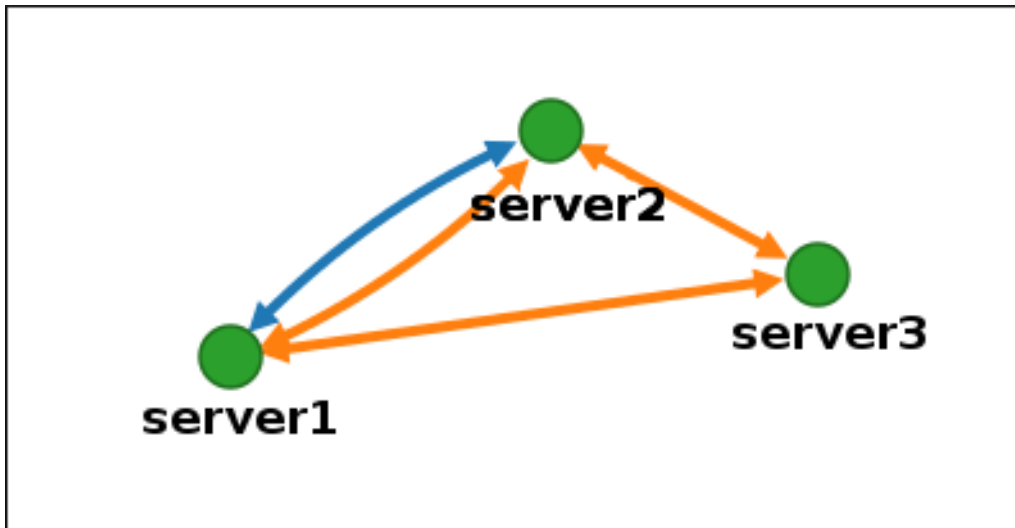


4.

Add topology segment (토폴로지 세그먼트 추가) 창에서 **Add(추가)**를 클릭하여 새 세그먼트의 속성을 확인합니다.

두 서버 간의 새 토폴로지 세그먼트는 복제 계약에 참여합니다. 토폴로지 그래프에 업데이트된 복제 토폴로지가 표시됩니다.

그림 48.11. 새 세그먼트 생성



48.4. 웹 UI를 사용하여 두 서버 간 복제 중지

IdM(Identity Management)의 웹 인터페이스를 사용하여 서버에서 복제 연결을 제거할 수 있습니다.

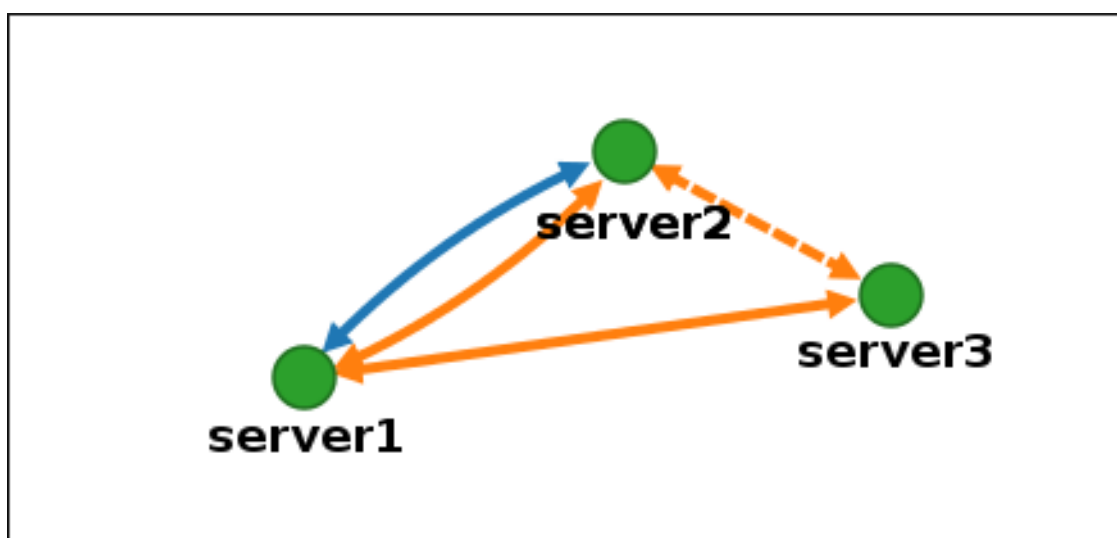
사전 요구 사항

- IdM 관리자 자격 증명이 있습니다.

절차

1. 제거할 복제 계약을 나타내는 화살표를 클릭합니다. 화살표를 강조 표시합니다.

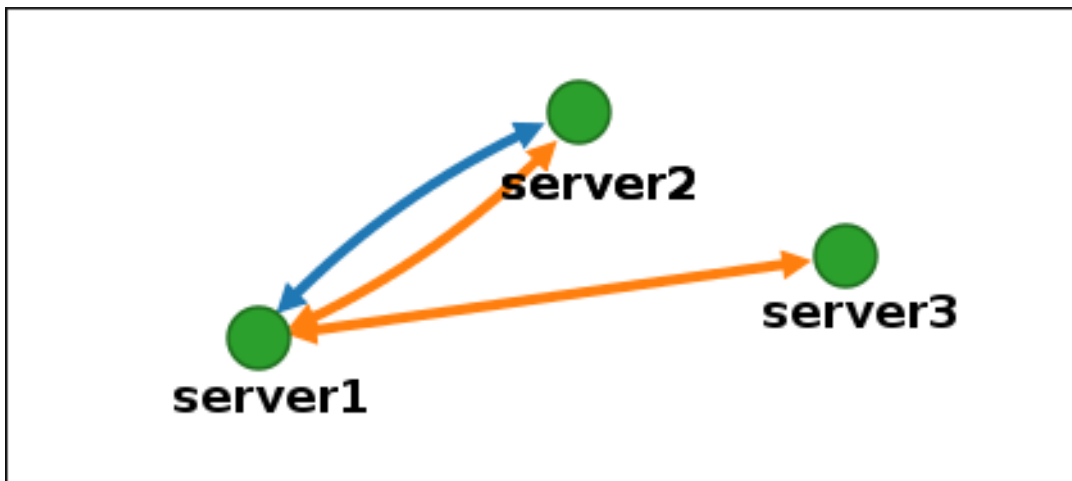
그림 48.12. 토폴로지 세그먼트가 강조 표시됨



2. 삭제를 클릭합니다.
3. **Confirmation (확인)** 창에서 **OK(확인)**를 클릭합니다.

IdM은 두 서버 간의 토폴로지 세그먼트를 제거하여 복제 계약을 삭제합니다. 토폴로지 그래프에 업데이트된 복제 토폴로지가 표시됩니다.

그림 48.13. 토폴로지 세그먼트 삭제



48.5. CLI를 사용하여 두 서버 간 복제 설정

ipa topologysegment-add 명령을 사용하여 두 서버 간 복제 계약을 구성할 수 있습니다.

사전 요구 사항

- **IdM** 관리자 자격 증명이 있습니다.

절차

1. **ipa topologysegment-add** 명령을 사용하여 두 서버의 토폴로지 세그먼트를 생성합니다. 메시지가 표시되면 다음을 제공합니다.
 - 필수 토폴로지 접미사: **domain** 또는 **ca**
 - 두 서버를 나타내는 왼쪽 노드와 오른쪽 노드

- 필요한 경우 세그먼트의 사용자 정의 이름

예를 들면 다음과 같습니다.

```
$ ipa topologysegment-add
Suffix name: domain
Left node: server1.example.com
Right node: server2.example.com
Segment name [server1.example.com-to-server2.example.com]: new_segment
-----
Added segment "new_segment"
-----
Segment name: new_segment
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
```

새 세그먼트를 추가하면 복제 계약의 서버에 참여합니다.

2. **선택 사항:** `ipa topologysegment-show` 명령을 사용하여 새 세그먼트가 구성되었는지 확인합니다.

```
$ ipa topologysegment-show
Suffix name: domain
Segment name: new_segment
Segment name: new_segment
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
```

48.6. CLI를 사용하여 두 서버 간 복제 중지

`ipa topology_segment-del` 명령을 사용하여 명령줄에서 복제 계약을 종료할 수 있습니다.

사전 요구 사항

- IdM 관리자 자격 증명이 있습니다.

절차

1. 복제를 중지하려면 서버 간에 해당 복제 세그먼트를 삭제해야 합니다. 이렇게 하려면 세그먼트 이름을 알아야 합니다.

이름을 모르는 경우 **ipa topologysegment-find** 명령을 사용하여 모든 세그먼트를 표시하고 출력에서 필요한 세그먼트를 찾습니다. 메시지가 표시되면 필수 토폴로지 접미사인 **domain** 또는 **ca** 를 제공합니다. 예를 들면 다음과 같습니다.

```
$ ipa topologysegment-find
Suffix name: domain
-----
8 segments matched
-----
Segment name: new_segment
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
...
-----
Number of entries returned 8
-----
```

2.

ipa topologysegment-del 명령을 사용하여 두 서버에 가입하는 토폴로지 세그먼트를 제거합니다.

```
$ ipa topologysegment-del
Suffix name: domain
Segment name: new_segment
-----
Deleted segment "new_segment"
-----
```

세그먼트를 삭제하면 복제 계약이 제거됩니다.

3.

선택 사항: **ipa topologysegment-find** 명령을 사용하여 세그먼트가 더 이상 나열되지 않는지 확인합니다.

```
$ ipa topologysegment-find
Suffix name: domain
-----
7 segments matched
-----
Segment name: server2.example.com-to-server3.example.com
Left node: server2.example.com
Right node: server3.example.com
Connectivity: both
...
-----
```

Number of entries returned 7

48.7. 웹 UI를 사용하여 토폴로지에서 서버 제거

IdM(Identity Management) 웹 인터페이스를 사용하여 토폴로지에서 서버를 제거할 수 있습니다.

사전 요구 사항

- **IdM** 관리자 자격 증명이 있습니다.
- 제거하려는 서버가 토폴로지의 나머지 부분에 다른 서버를 연결하는 유일한 서버가 아닙니다. 이로 인해 다른 서버가 허용되지 않습니다.
- 삭제하려는 서버는 마지막 **CA** 또는 **DNS** 서버가 아닙니다.



주의

서버를 제거하는 것은 되돌릴 수 없는 작업입니다. 서버를 제거하면 토폴로지로는 다시 도입하는 유일한 방법은 시스템에 새 복제본을 설치하는 것입니다.

절차

시스템에서 서버 구성 요소를 설치 제거하지 않고 토폴로지에서 서버를 제거하려면 다음을 수행합니다.

1. **IPA Server** → **Topology** → **IPA Servers** 를 선택합니다.
2. 삭제할 서버 이름을 클릭합니다.

그림 48.14. 서버 선택

IPA Servers				
Search			Refresh	
☐	Server name	Min domain level	Max domain level	Managed suffixes
☐	server1.example.com	0	1	domain, ca
☐	server2.example.com	0	1	domain
☐	server3.example.com	0	1	domain, ca
Showing 1 to 3 of 3 entries.				

3.

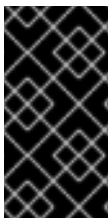
Delete Server(서버 삭제)를 클릭합니다.

48.8. CLI를 사용하여 토폴로지에서 서버 제거

명령줄 인터페이스를 사용하여 토폴로지에서 서버를 제거할 수 있습니다.

사전 요구 사항

- IdM 관리자 자격 증명이 있습니다.
- 제거하려는 서버가 토폴로지의 나머지 부분에 다른 서버를 연결하는 유일한 서버가 아닙니다. 이로 인해 다른 서버가 격리되어 허용되지 않습니다.
- 삭제하려는 서버는 마지막 CA 또는 DNS 서버가 아닙니다.



중요

서버를 제거하는 것은 되돌릴 수 없는 작업입니다. 서버를 제거하면 토폴로지로는 다시 도입하는 유일한 방법은 시스템에 새 복제본을 설치하는 것입니다.

절차

server1.example.com을 제거하려면 다음을 수행합니다.

1.

다른 서버에서 **ipa server-del** 명령을 실행하여 **server 1.example.com** 을 제거합니다. 명령

은 서버를 가리키는 모든 토폴로지 세그먼트를 제거합니다.

```
[user@server2 ~]$ ipa server-del
Server name: server1.example.com
Removing server1.example.com from replication topology, please wait...

Deleted IPA server "server1.example.com"
```

2.

선택 사항: `server1.example.com` 에서 `ipa server-install --uninstall` 명령을 실행하여 시스템에서 서버 구성 요소를 제거합니다.

```
[root@server1 ~]# ipa server-install --uninstall
```

48.9. 웹 UI를 사용하여 IDM 서버에서 서버 역할 보기

IdM 서버에 설치된 서비스를 기반으로 다양한 **서버 역할**을 수행할 수 있습니다. 예를 들면 다음과 같습니다.

- **CA 서버**
- **DNS 서버**
- **KRA(핵심 복구 기관) 서버.**

지원되는 서버 역할의 전체 목록은 **IPA 서버 토폴로지 서버** → **역할**을 참조하십시오.



참고

- 역할 상태가 없을 경우 토폴로지에서 역할을 수행하고 있는 서버가 없음을 나타냅니다.
- 역할 상태가 활성화된 경우 토폴로지에서 하나 이상의 서버가 역할을 수행하고 있음을 나타냅니다.

그림 48.15. 웹 UI의 서버 역할

Server Roles	
Refresh	
Role name	Role status
AD trust agent	absent
AD trust controller	absent
CA server	enabled

48.10. CLI를 사용하여 IDM 서버에서 서버 역할 보기

IdM 서버에 설치된 서비스를 기반으로 다양한 서버 역할을 수행할 수 있습니다. 예를 들면 다음과 같습니다.

- CA 서버
- DNS 서버
- KRA(핵심 복구 기관) 서버.

다음 명령을 사용하여 토폴로지에서 역할을 수행하는 서버를 볼 수 있습니다.

- `ipa config-show` 명령은 모든 CA 서버 및 현재 CA 갱신 서버를 표시합니다.

```
$ ipa config-show
```

```
...
```

```
IPA masters: server1.example.com, server2.example.com, server3.example.com
```

```
IPA CA servers: server1.example.com, server2.example.com
```

```
IPA CA renewal master: server1.example.com
```

-

`ipa server-show` 명령은 특정 서버에서 활성화된 역할 목록을 표시합니다. 예를 들어 `server.example.com`에서 활성화된 역할 목록은 다음과 같습니다.

```
$ ipa server-show
```

```
Server name: server.example.com
```

```
...
```

Enabled server roles: CA server, DNS server, KRA server

- `ipa server-find --servrole` 은 특정 서버 역할이 활성화된 모든 서버를 검색합니다. 예를 들어 모든 CA 서버를 검색하려면 다음을 수행합니다.

```
$ ipa server-find --servrole "CA server"
```

```
-----  
2 IPA servers matched  
-----
```

```
Server name: server1.example.com
```

```
...
```

```
Server name: server2.example.com
```

```
...
```

```
-----  
Number of entries returned 2  
-----
```

48.11. CA 갱신 서버 및 CRL 게시 서버로 복제본 승격

IdM 배포에서 포함된 CA(인증 기관)를 사용하는 경우 IdM CA 서버 중 하나가 CA 하위 시스템 인증서 갱신을 관리하는 서버인 CA 갱신 서버 역할을 합니다. IdM CA 서버 중 하나는 인증서 해지 목록을 생성하는 서버인 IdM CRL 게시자 서버 역할을 합니다. 기본적으로 CA 갱신 서버 및 CRL 게시자 서버 역할은 시스템 관리자가 `ipa-server-install` 또는 `ipa-ca-install` 명령을 사용하여 CA 역할을 설치한 첫 번째 서버에 설치됩니다.

사전 요구 사항

- IdM 관리자 자격 증명이 있습니다.

절차

- 현재 CA 갱신 서버를 변경합니다.
- CRL을 생성하도록 복제본을 구성합니다.

48.12. 숨겨진 복제본 데모 또는 승격

복제본을 설치한 후에는 복제본이 숨겨져 있는지 또는 표시되는지 여부를 구성할 수 있습니다.

숨겨진 복제본에 대한 자세한 내용은 [숨겨진 복제본 모드](#)를 참조하십시오.

복제본이 **CA** 갱신 서버인 경우 이 복제본을 숨기기 전에 서비스를 다른 복제본으로 이동합니다. 자세한 내용은 [IdM CA 갱신 서버 변경 및 재설정](#)을 참조하십시오.



참고

RHEL 8.1에서 기술 프리뷰로 도입된 숨겨진 복제본 기능은 **RHEL 8.2**부터 완전히 지원됩니다.

절차

- 복제본을 숨기려면 다음을 입력합니다.

```
# ipa server-state replica.idm.example.com --state=hidden
```

또는 다음 명령을 사용하여 복제본을 볼 수 있습니다.

```
# ipa server-state replica.idm.example.com --state=enabled
```

49장. IDENTITY MANAGEMENT의 공개 키 인증서

이 장에서는 **IdM(Identity Management)**에서 사용자, 호스트 및 서비스를 인증하는 데 사용되는 **X.509** 공개 키 인증서를 설명합니다. **X.509** 인증서는 인증 외에도 디지털 서명 및 암호화를 통해 개인 정보 보호, 무결성 및 비응답을 제공할 수 있습니다.

인증서에는 다음 정보가 포함됩니다.

- 인증서가 인증하는 주체입니다.
- 발급자, 즉 인증서에 서명한 **CA**입니다.
- 인증서의 유효성의 시작 및 종료일입니다.
- 인증서의 유효한 용도입니다.
- 주체의 공개 키입니다.

공개 키로 암호화된 메시지는 해당 개인 키로만 해독할 수 있습니다. 인증서와 포함된 공개 키는 공개적으로 사용할 수 있지만 사용자, 호스트 또는 서비스는 개인 키 비밀을 유지해야 합니다.

49.1. IDM의 인증 기관

인증 기관은 신뢰 계층에서 작동합니다. 내부 **CA(인증 기관)**가 있는 **IdM** 환경에서 모든 **IdM** 호스트, 사용자 및 서비스는 **CA**에서 서명한 인증서를 신뢰합니다. 이 루트 **CA** 이외에 **IdM**은 루트 **CA**에서 인증서에 차례로 서명할 수 있는 기능을 부여한 하위 **CA**를 지원합니다. 일반적으로 이러한 하위 **CA**가 서명할 수 있는 인증서는 특정 종류의 인증서(예: **VPN** 인증서)입니다. 마지막으로 **IdM**은 외부 **CA** 사용을 지원합니다. [아래](#) 표는 **IdM**에서 개별 유형의 **CA**를 사용하는 세부 사항을 보여줍니다.

표 49.1. IdM에서 통합 및 외부 CA 사용 비교

CA 이름	설명	사용	유용한 링크
ipa CA	Dogtag 업스트림 프로젝트 기반의 통합 CA	통합 CA는 사용자, 호스트 및 서비스에 대한 인증서를 생성, 취소 및 발급할 수 있습니다.	ipa CA를 사용하여 새 사용자 인증서를 요청하고 클라이언트로 내보내기
IdM 하위 CA	ipa CA에 종속되는 통합 CA	IdM 하위 CA는 ipa CA에서 인증서 서명 기능을 부여한 CA입니다. 이러한 인증서는 VPN 인증서와 같은 특정 종류입니다.	인증서의 하위 집합만 신뢰하도록 애플리케이션 제한
외부 CA	외부 CA는 통합된 IdM CA 또는 하위 CA가 아닌 다른 CA입니다.	IdM 도구를 사용하여 이러한 CA에서 발급한 인증서를 사용자, 서비스 또는 호스트에 추가하고 제거합니다.	RHEL 7 설명서에서 외부 CA에서 발급한 인증서 관리

인증서 관점에서 자체 서명 IdM CA가 서명하고 외부적으로 서명하는 것은 차이가 없습니다.

CA의 역할에는 다음과 같은 목적이 포함됩니다.

- 디지털 인증서를 발급합니다.
- 인증서에 서명하면 인증서에 있는 라는 주체가 공개 키를 소유함을 인증합니다. 제목은 사용자, 호스트 또는 서비스일 수 있습니다.
- CRL(Certificate Revocation Lists) 및 OCSP(Online Certificate Status Protocol)를 통해 인증서를 취소하고 해지할 수 있습니다.

추가 리소스

- IdM 서버의 지원되는 CA 구성에 대한 자세한 내용은 [CA 서비스](#) 계획을 참조하십시오.

49.2. 인증서와 KERBEROS 비교

인증서는 Kerberos 티켓에 의해 수행되는 와 유사한 기능을 수행합니다. Kerberos는 비보안 네트워크

를 통해 통신하는 노드가 안전한 방식으로 서로의 ID를 증명할 수 있는 티켓을 기반으로 작동하는 컴퓨터 네트워크 인증 프로토콜입니다. 다음 표는 Kerberos 및 X.509 인증서를 비교한 결과를 보여줍니다.

표 49.2. 인증서와 Kerberos 비교

특성	Kerberos	X.509
인증	있음	있음
개인 정보 보호	선택 사항	있음
무결성	선택 사항	있음
관련된 암호화 유형	대칭	비대칭
기본 유효 기간	짧은(1일)	긴(2년)

기본적으로 Identity Management의 Kerberos는 통신 당사자의 ID만 허용합니다.

49.3. 인증서를 사용하여 IDM의 사용자를 인증하는 장단점

IdM의 사용자를 인증하는 인증서를 사용하면 다음과 같은 사항이 포함됩니다.

- 스마트 카드에서 개인 키를 보호하는 PIN은 일반적으로 일반 암호보다 덜 복잡하고 기억하기 쉽습니다.
- 장치에 따라 스마트 카드에 저장된 개인 키를 내보낼 수 없습니다. 이는 추가적인 보안 기능을 제공합니다.
- 스마트 카드는 로그아웃을 자동으로 만들 수 있습니다. reader에서 스마트 카드를 제거할 때 사용자를 로그아웃하도록 IdM을 구성할 수 있습니다.
- 개인 키를 포착하려면 스마트 카드로 실제 물리적으로 액세스해야 하므로 해커 공격으로부터 스마트 카드를 안전하게 보호할 수 있습니다.
-

스마트 카드 인증은 이중 인증의 예입니다. 이 인증에는 있는 내용(카드)과 PIN(PIN)이 모두 필요합니다.

- 스마트 카드는 이메일 암호화와 같은 다른 목적에 사용할 수 있는 키를 제공하기 때문에 암호보다 더 유연합니다.
- IdM 클라이언트인 공유 시스템에서 스마트 카드를 사용하면 일반적으로 시스템 관리자에게 추가 구성 문제가 발생하지 않습니다. 실제로 스마트 카드 인증은 공유 시스템에 이상적인 선택입니다.

IdM에서 사용자를 인증하기 위해 인증서를 사용하는 단점은 다음과 같은 사항이 포함됩니다.

- 사용자는 스마트 카드나 인증서를 가져오는 것을 잊어버리거나 잊고 효과적으로 잠길 수 있습니다.
- PIN을 여러 번 잘못 입력하면 카드가 잠길 수 있습니다.
- 일반적으로 일종의 보안 책임자 또는 승인자의 요청과 권한 사이의 중간 단계가 있습니다. IdM에서 보안 책임자 또는 관리자는 **ipa cert-request** 명령을 실행해야 합니다.
- 스마트 카드와 리더는 벤더와 드라이버에 고유한 경향이 있습니다. 많은 독자가 다른 카드에 사용할 수 있지만 특정 벤더의 스마트 카드는 다른 벤더의 리더나 설계되지 않은 리더 유형에서 작동하지 않을 수 있습니다.
- 인증서와 스마트 카드는 관리자를 위한 고급 학습 곡선을 갖추고 있습니다.

50장. 통합 IDM CA를 사용한 사용자, 호스트 및 서비스에 대한 인증서 관리

이 장에서는 통합 CA, ipa CA 및 하위 CA를 사용하여 IdM(Identity Management)에서 인증서를 관리하는 방법을 설명합니다.

이 장에는 다음 섹션이 포함되어 있습니다.

- [IdM 웹 UI를 사용하여 사용자, 호스트 또는 서비스에 새 인증서 요청.](#)
- [IdM CLI를 사용하여 IdM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청:](#)
 - [certutil을 사용하여 IdM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청](#)
 - [certutil 유틸리티를 사용하여 IdM CA에서 새 사용자 인증서를 요청하고 IdM 클라이언트로 내보내는 특정 예는 새 사용자 인증서 요청 및 클라이언트로 내보내는 것을 참조하십시오.](#)
 - [openssl을 사용하여 IdM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청](#)

[certmonger](#) 유틸리티를 사용하여 IdM CA에서 서비스에 대한 새 인증서를 요청할 수도 있습니다. 자세한 내용은 [certmonger를 사용하여 IdM CA에서 서비스에 대한 새 인증서 요청](#)을 참조하십시오.

사전 요구 사항

- [IdM 배포에 통합된 CA가 포함되어 있습니다.](#)
 - [IdM에서 CA 서비스 계획을 수립하는 방법에 대한 자세한 내용은 CA 서비스 계획을 참조하십시오.](#)
 - [root CA로 통합된 DNS 및 통합 CA를 사용하여 IdM 서버를 설치하는 방법에 대한 자세한 내용은 IdM 서버 설치를 참조하십시오. 통합 DNS, 루트 CA로 통합 CA 포함](#)
 - [root CA로 통합된 DNS 및 외부 CA를 사용하여 IdM 서버를 설치하는 방법에 대한 자세한](#)

한 내용은 [IdM 서버 설치를 참조하십시오](#). 통합 [DNS](#), 외부 [CA](#)가 루트 [CA](#)로 포함

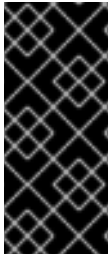
- 통합 [DNS](#) 없이 IdM 서버를 설치하고 루트 [CA](#)로 통합 [CA](#)를 설치하는 방법에 대한 자세한 내용은 [IdM 서버 설치를 참조하십시오](#). 통합 [DNS](#) 없이, 루트 [CA](#)로 통합 [CA](#).
- [선택 사항] IdM 배포에서는 인증서를 사용한 인증 사용자를 지원합니다.
 - IdM 클라이언트 파일 시스템에 저장된 인증서를 사용하여 사용자 인증을 지원하도록 IdM 배포를 구성하는 방법에 대한 자세한 내용은 IdM 클라이언트 [데스크탑에 저장된 인증서를 사용하여 인증 구성을 참조하십시오](#).
 - IdM 클라이언트에 스마트 카드가 저장된 인증서를 사용하여 사용자 인증을 지원하도록 IdM 배포를 구성하는 방법에 대한 자세한 내용은 [스마트 카드 인증을 위한 ID 관리](#) 구성을 참조하십시오.
 - Active Directory 인증서 시스템에서 발급한 스마트 카드로 사용자 인증을 지원하도록 IdM 배포를 구성하는 방법에 대한 자세한 내용은 [IdM의 스마트 카드 인증을 위해 ADCS에서 발급한 인증서](#) 구성을 참조하십시오.

50.1. IDM 웹 UI를 사용하여 사용자, 호스트 또는 서비스에 대한 새 인증서 요청

이 섹션에서는 IdM(Identity Management) 웹 UI를 사용하여 통합된 IdM(인증 기관) CA 또는 하위 CA에서 IdM 엔터티에 대한 새 인증서를 요청하는 방법에 대해 설명합니다.

IdM 엔터티에는 다음이 포함됩니다.

- 사용자
- 호스트
- 서비스



중요

일반적으로 서비스는 개인 키가 저장된 전용 서비스 노드에서 실행됩니다. 서비스의 개인 키를 IdM 서버에 복사하는 것은 안전하지 않은 것으로 간주됩니다. 따라서 서비스에 대한 인증서를 요청하는 경우 서비스 노드에 **CSR**(인증서 서명 요청)을 생성합니다.

사전 요구 사항

- IdM 배포에 통합된 CA가 포함되어 있습니다.
- IdM 관리자로 IdM 웹 UI에 로그인되어 있습니다.

절차

1. Identity(ID) 탭에서 **Users** (사용자), **Hosts**(호스트) 또는 **Services** (서비스) 하위 탭을 선택합니다.
2. 사용자, 호스트 또는 서비스의 이름을 클릭하여 구성 페이지를 엽니다.

그림 50.1. 호스트 목록

Hosts			
Search		Refresh	Delete
+ Add		Actions v	
☐	Host name	Description	Enrolled
☐	server.example.com		True
Showing 1 to 1 of 1 entries.			

3. **Actions**(작업) **New Certificate**(새 인증서) 를 클릭합니다.
4. 선택 사항: 발급 CA 및 프로필 ID를 선택합니다.
5. 화면에서 **certutil CLI**(명령줄) 유틸리티 사용 지침을 따르십시오.

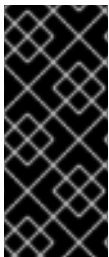
6.

Issue(문제)를 클릭합니다.

50.2. CERTUTIL을 사용하여 IDM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청

certutil 유틸리티를 사용하여 표준 **IdM** 상황에서 **IdM(Identity Management)** 사용자, 호스트 또는 서비스에 대한 인증서를 요청할 수 있습니다. 호스트 또는 서비스 **Kerberos** 별칭이 인증서를 사용할 수 있도록 하려면 **openssl** 유틸리티를 사용하여 인증서를 대신 요청합니다.

이 섹션에서는 **certutil** 을 사용하여 **IdM** 사용자, 호스트 또는 서비스에 대한 인증서를 요청하는 방법에 대해 설명합니다.



중요

일반적으로 서비스는 개인 키가 저장된 전용 서비스 노드에서 실행됩니다. 서비스의 개인 키를 **IdM** 서버에 복사하는 것은 안전하지 않은 것으로 간주됩니다. 따라서 서비스에 대한 인증서를 요청하는 경우 서비스 노드에 **CSR(인증서 서명 요청)**을 생성합니다.

사전 요구 사항

- **IdM** 배포에 통합된 **CA**가 포함되어 있습니다.
- **IdM** 관리자로 **IdM CLI(명령줄 인터페이스)**에 로그인되어 있습니다.

절차

1. 인증서 데이터베이스의 임시 디렉토리를 생성합니다.

```
# mkdir ~/certdb/
```

2. 다음과 같은 새 임시 인증서 데이터베이스를 생성합니다.

```
# certutil -N -d ~/certdb/
```

3. **CSR**을 생성하고 출력을 파일로 리디렉션합니다. 예를 들어 **4096비트** 인증서에 대한 **CSR**을 생성하고 제목을 **CN=server.example.com,O=EXAMPLE.COM**로 설정하려면 다음을 수행합니다.

```
# certutil -R -d ~/certdb/ -a -g 4096 -s "CN=server.example.com,O=EXAMPLE.COM" -8
server.example.com > certificate_request.csr
```

4.

IdM 서버에서 실행 중인 CA에 인증서 요청 파일을 제출합니다. 새로 발급된 인증서와 연결할 Kerberos 주체를 지정합니다.

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

IdM의 ipa cert-request 명령은 다음 기본값을 사용합니다.

•

calPAserviceCert 인증서 프로필

사용자 지정 프로필을 선택하려면 **--profile-id** 옵션을 사용합니다.

•

통합 IdM 루트 CA, ipa

하위 CA를 선택하려면 **--ca** 옵션을 사용합니다.

추가 리소스

•

ipa cert-request 명령에 대한 자세한 내용은 ipa cert-request --help 명령의 출력을 참조하십시오.

•

사용자 정의 인증서 프로필 생성에 대한 자세한 내용은 [Identity Management에서 인증서 프로필 생성 및 관리](#)를 참조하십시오.

50.3. OPENSSSL을 사용하여 IDM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청

호스트 또는 서비스의 Kerberos 별칭이 인증서를 사용할 수 있도록 하려면 openssl 유틸리티를 사용하여 IdM(Identity Management) 호스트 또는 서비스에 대한 인증서를 요청할 수 있습니다. 표준 상황에서는 대신 certutil 유틸리티를 사용하여 새 인증서를 요청하는 것이 좋습니다.

이 섹션에서는 openssl 을 사용하여 IdM 호스트 또는 ipa 에서 IdM 인증 기관인 서비스를 요청하는 방법을 설명합니다.



중요

일반적으로 서비스는 개인 키가 저장된 전용 서비스 노드에서 실행됩니다. 서비스의 개인 키를 IdM 서버에 복사하는 것은 안전하지 않은 것으로 간주됩니다. 따라서 서비스에 대한 인증서를 요청하는 경우 서비스 노드에 **CSR**(인증서 서명 요청)을 생성합니다.

사전 요구 사항

- IdM 배포에 통합된 CA가 포함되어 있습니다.
- IdM 관리자로 IdM CLI(명령줄 인터페이스)에 로그인되어 있습니다.

절차

1. Kerberos 주체 **test/server.example.com**에 대해 하나 이상의 별칭을 만듭니다. 예를 들어 **test1/server.example.com** 및 **test2/server.example.com**입니다.
2. CSR에서 **dnsName(server.example.com)** 및 **otherName(test2/server.example.com)**에 대한 **subjectAltName**을 추가합니다. 이렇게 하려면 **UPN otherName** 및 **subjectAltName**을 지정하는 다음 행을 포함하도록 **openssl.conf** 파일을 구성합니다.

```
otherName=1.3.6.1.4.1.311.20.2.3;UTF8:test2/server.example.com@EXAMPLE.COM
DNS.1 = server.example.com
```

3. **openssl** 을 사용하여 인증서 요청을 생성합니다.

```
openssl req -new -newkey rsa:2048 -keyout test2service.key -sha256 -nodes -out
certificate_request.csr -config openssl.conf
```

4. IdM 서버에서 실행 중인 CA에 인증서 요청 파일을 제출합니다. 새로 발급된 인증서와 연결할 Kerberos 주체를 지정합니다.

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

IdM의 **ipa cert-request** 명령은 다음 기본값을 사용합니다.

- **calPAserviceCert** 인증서 프로파일

사용자 지정 프로필을 선택하려면 **--profile-id** 옵션을 사용합니다.

- 통합 IdM 루트 CA, ipa

하위 CA를 선택하려면 **--ca** 옵션을 사용합니다.

추가 리소스

- **ipa cert-request** 명령에 대한 자세한 내용은 **ipa cert-request --help** 명령의 출력을 참조하십시오.
- 사용자 정의 인증서 프로필 생성에 대한 자세한 내용은 [Identity Management에서 인증서 프로필 생성 및 관리](#)를 참조하십시오.

50.4. 추가 리소스

- IdM CA를 사용하여 인증서를 취소하는 방법에 대한 자세한 내용은 [통합 IdM CA가 있는 인증서 공개](#)를 참조하십시오.
- IdM CA를 사용하여 인증서를 복원하는 방법에 대한 자세한 내용은 [통합 IdM CA가 있는 인증서 복원](#)을 참조하십시오.
- IdM 하위 CA에서 발급한 인증서만 신뢰하도록 애플리케이션을 제한하는 방법에 대한 자세한 내용은 [애플리케이션 제한에서 인증서의 하위 집합만 신뢰하도록 애플리케이션 제한](#)을 참조하십시오.

51장. IDM에서 작동하도록 인증서 형식 변환

이 사용자 사례에서는 **IdM** 시스템 관리자로서 특정 **IdM** 명령으로 올바른 형식의 인증서를 사용하고 있는지 확인하는 방법을 설명합니다. 예를 들어 다음과 같은 상황에서 유용합니다.

- 외부 인증서를 사용자 프로필에 로드하고 있습니다. 자세한 내용은 [51.2절. “IdM 사용자 계정으로 로드할 외부 인증서를 변환”](#)의 내용을 참조하십시오.
- 사용자가 외부 인증 기관에서 발급한 인증서가 있는 스마트 카드를 사용하여 **IdM**을 인증할 수 있도록 스마트 카드 인증을 위해 **IdM** 서버를 구성하거나 스마트 카드 인증을 위해 **IdM** 클라이언트를 구성할 때 외부 **CA** 인증서를 사용하고 있습니다.
- **NSS** 데이터베이스에서 인증서와 개인 키를 모두 포함하는 **pkcs #12** 형식으로 인증서를 내보내고 있습니다. 자세한 내용은 [51.3.1절. “NSS 데이터베이스에서 PKCS #12 파일로 인증서 및 개인 키 내보내기”](#)의 내용을 참조하십시오.

51.1. IDM의 인증서 형식 및 인코딩

사용자가 제공하는 인증서를 인증서 또는 사용자의 **IdM** 프로필에 저장된 인증서 데이터와 비교하여 **IdM**의 스마트 카드 인증을 포함한 인증서 인증이 진행됩니다.

시스템 설정

IdM 프로필에 저장된 것은 해당 개인 키가 아닌 인증서에만 저장됩니다. 인증하는 동안 사용자는 해당 개인 키를 소유하고 있음을 표시해야 합니다. 사용자는 인증서와 개인 키를 모두 포함하는 **PKCS #12** 파일을 제공하거나 인증서가 포함된 두 개의 파일을 제공하고 개인 키가 포함된 다른 파일을 제공하여 이러한 작업을 수행합니다.

따라서 사용자 프로필에 인증서를 로드하는 것과 같은 프로세스는 개인 키가 포함되지 않은 인증서 파일만 허용합니다.

마찬가지로 시스템 관리자가 외부 **CA** 인증서를 제공하는 경우 공용 데이터(개인 키가 없는 인증서)만 제공합니다. **IdM** 서버 또는 스마트 카드 인증을 위해 **IdM** 클라이언트를 구성하는 **ipa-adviser** 유틸리티에서는 입력 파일에 외부 **CA**의 인증서가 포함되지 않고 개인 키가 아닌 외부 **CA**의 인증서가 포함되어야 합니다.

인증서 인코딩

일반적인 인증서 인코딩에는 두 가지가 있습니다. 개인 정보 보호 강화 전자 메일(**PEM**) 및 고유 인코딩 규칙(**DER**). **base64** 형식은 **PEM** 형식과 거의 동일하지만 **-----BEGIN CERTIFICATE-----/-----END**

CERTIFICATE----- 헤더와 바닥글을 포함하지 않습니다.

DER를 사용하여 인코딩된 인증서는 바이너리 **X509** 디지털 인증서 파일입니다. 이전 파일로는 인증서를 사람이 읽을 수 없습니다. **DER** 파일은 **.der** 파일 확장자를 사용하는 경우도 있지만 **.crt** 및 **.cer** 파일 확장자가 있는 파일에도 **DER** 인증서가 포함된 경우가 있습니다. 키가 포함된 **DER** 파일의 이름은 **.key**입니다.

PEM Base64를 사용하여 인코딩된 인증서는 사람이 읽을 수 있는 파일입니다. 파일에는 **"----- BEGIN ..."** 행이 앞에 있는 **ASCII(Base64)**의 마운트된 데이터가 포함되어 있습니다. **PEM** 파일에서 **.pem** 파일 확장자를 사용하는 경우도 있지만 **.crt** 및 **.cer** 파일 확장자가 있는 파일에도 **PEM** 인증서가 포함된 경우가 있습니다. 키가 포함된 **PEM** 파일의 이름은 **.key**로 지정할 수 있습니다.

다른 **ipa** 명령에는 수락하는 인증서 유형에 대한 제한 사항이 다릅니다. 예를 들어 **ipa user-add-cert** 명령은 **base64** 형식으로 인코딩된 인증서만 수락하지만 **ipa-server-certinstall**은 **PEM**, **DER**, **PKCS #7**, **PKCS #8** 및 **PKCS #12** 인증서를 허용합니다.

표 51.1. 인증서 인코딩

인코딩 형식	사람이 읽을 수 있음	일반적인 파일 이름 확장	인코딩 형식을 허용하는 샘플 IdM 명령
PEM/base64	있음	.pem, .crt, .cer	ipa user-add-cert, ipa-server-certinstall, ...
DER	없음	.der, .crt, .cer	ipa-server-certinstall, ...

51.4절. “IdM의 인증서 관련 명령 및 형식” 인증서가 수락하는 인증서 형식을 사용하여 추가 **ipa** 명령을 나열합니다.

사용자 인증

웹 UI를 사용하여 IdM에 액세스하는 경우 사용자는 브라우저의 데이터베이스에 둘 다 저장되어 인증서에 해당하는 개인 키가 포함되어 있음을 증명합니다.

CLI를 사용하여 IdM에 액세스하는 경우 다음 방법 중 하나를 통해 인증서에 해당하는 개인 키가 포함되어 있음을 사용자가 증명합니다.

-

사용자는 **kinit -X** 명령의 **X509_user_identity** 매개변수 값으로 인증서와 키가 모두 포함된 스마트 카드 모듈의 경로를 추가합니다.

```
$ kinit -X X509_user_identity='PKCS11:opensc-pkcs11.so' idm_user
```

•

사용자는 `kinit -X` 명령의 `X509_user_identity` 매개변수 값으로 두 개의 파일을 추가하고, 인증서와 다른 하나는 개인 키를 포함합니다.

```
$ kinit -X X509_user_identity='FILE:/path/to/cert.pem,/path/to/cert.key' idm_user
```

유용한 인증서 명령

주체 및 발급자와 같은 인증서 데이터를 보려면 다음을 수행합니다.

```
$ openssl x509 -noout -text -in ca.pem
```

두 인증서가 다른 행을 비교하려면 다음을 수행하십시오.

```
$ diff cert1.crt cert2.crt
```

두 인증서가 두 열에 표시된 출력과 다른 행을 비교하려면 다음을 수행하십시오.

```
$ diff cert1.crt cert2.crt -y
```

51.2. IDM 사용자 계정으로 로드할 외부 인증서를 변환

이 섹션에서는 외부 인증서가 올바르게 인코딩되고 사용자 항목에 추가하기 전에 포맷되도록 하는 방법을 설명합니다.

사전 요구 사항

•

인증서가 **Active Directory** 인증 기관에서 발급되었으며 **PEM** 인코딩을 사용하는 경우 **PEM** 파일이 **UNIX** 형식으로 변환되었는지 확인합니다. 파일을 변환하려면 **eponymous** 패키지에서 제공하는 **dos2unix** 유틸리티를 사용합니다.

51.2.1. IdM CLI에서 외부 인증서를 변환하여 IdM 사용자 계정으로 로드

IdM CLI 는 첫 번째 및 마지막 행(-----BEGIN CERTIFICATE----- 및 -----END CERTIFICATE-----)이 제거된 **PEM** 인증서만 허용합니다.

절차

1.

인증서를 **PEM** 형식으로 변환합니다.

•

인증서가 **DER** 형식인 경우:

```
$ openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

•

파일 이름이 **.pfx** 및 **p12** 인 **PKCS #12** 형식에 있고 인증서, 개인 키 및 기타 데이터가 포함된 경우 **openssl pkcs12** 유틸리티를 사용하여 인증서를 추출합니다. 메시지가 표시되면 파일에 저장된 개인 키를 보호하는 암호를 입력합니다.

```
$ openssl pkcs12 -in cert_and_key.p12 -clcerts -nokeys -out cert.pem
Enter Import Password:
```

2.

관리자의 자격 증명을 가져옵니다.

```
$ kinit admin
```

3.

다음 방법 중 하나로 **IdM CLI** 를 사용하여 사용자 계정에 인증서를 추가합니다.

•

ipa user-add-cert 명령에 문자열을 추가하기 전에 **sed** 유틸리티를 사용하여 **PEM** 파일의 첫 번째 및 마지막 행(**-----BEGIN CERTIFICATE-----** 및 **-----END CERTIFICATE-----**)을 제거합니다.

```
$ ipa user-add-cert some_user --certificate="$(sed -e '/BEGIN
CERTIFICATE/d;/END CERTIFICATE/d' cert.pem)"
```

•

첫 번째 및 마지막 행(**-----BEGIN CERTIFICATE-----** 및 **-----END CERTIFICATE-----**) 없이 인증서 파일의 내용을 **ipa user-add-cert** 명령에 복사하여 붙여넣습니다.

```
$ ipa user-add-cert some_user --
certificate=MIIDlzCCAn+gAwIBAgIBATANBgkqhki...
```



참고

첫 번째 및 마지막 행을 먼저 제거하지 않고 인증서가 **ipa user-add-cert** 명령에 입력으로 포함된 **PEM** 파일을 전달할 수 없습니다(----- BEGIN CERTIFICATE----- 및 -----END CERTIFICATE-----).

```
$ ipa user-add-cert some_user --cert=some_user_cert.pem
```

이 명령을 실행하면 "ipa: 오류: Base64 디코딩에 실패했습니다: 잘못된 패딩" 오류 메시지.

4.

선택적으로 시스템에서 인증서를 수락했는지 확인하려면 다음을 수행하십시오.

```
[idm_user@r8server]$ ipa user-show some_user
```

51.2.2. IdM 사용자 계정으로 로드하기 위해 IdM 웹 UI에서 외부 인증서를 변환합니다.

절차

1.

CLI 를 사용하여 인증서를 **PEM** 형식으로 변환합니다.

•

인증서가 **DER** 형식인 경우:

```
$ openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

•

파일 이름이 **.pfx** 및 **p12** 인 **PKCS #12** 형식에 있고 인증서, 개인 키 및 기타 데이터가 포함된 경우 **openssl pkcs12** 유틸리티를 사용하여 인증서를 추출합니다. 메시지가 표시되면 파일에 저장된 개인 키를 보호하는 암호를 입력합니다.

```
$ openssl pkcs12 -in cert_and_key.p12 -clcerts -nokeys -out cert.pem
Enter Import Password:
```

2.

편집기에서 인증서를 열고 내용을 복사합니다. "----- BEGIN CERTIFICATE-----" 및 "-----END CERTIFICATE-----" 머리글과 바닥글 행을 포함할 수 있지만, **PEM** 및 **base64** 형식이 모두 IdM 웹 UI에서 허용되므로 필요하지는 않습니다.

3.

IdM 웹 UI에서 보안 책임자로 로그인합니다.

4. **ID → 사용자 → some_user** 로 이동합니다.
5. 인증서 옆에 있는 **Add(추가)** 를 클릭합니다.
6. 인증서의 **PEM** 형식 콘텐츠를 열리는 창에 붙여넣습니다.
7. 추가를 클릭합니다.

시스템에서 인증서를 수락한 경우 사용자 프로필의 인증서 중 나열된 것을 확인할 수 있습니다.

51.3. 브라우저에 인증서 로드 준비

사용자 인증서를 브라우저로 가져오기 전에 인증서 및 해당 개인 키가 **PKCS #12** 형식으로 되어 있는지 확인합니다. 추가 준비 작업이 필요한 두 가지 일반적인 상황이 있습니다.

- 인증서는 **NSS** 데이터베이스에 있습니다. 이 상황에서 진행되는 방법에 대한 자세한 내용은 **51.3.1절. “NSS 데이터베이스에서 PKCS #12 파일로 인증서 및 개인 키 내보내기”**의 내용을 참조하십시오.
- 인증서와 개인 키는 두 개의 개별 **PEM** 파일에 있습니다. 이 상황에서 진행되는 방법에 대한 자세한 내용은 **51.3.2절. “인증서와 개인 키 PEM 파일을 PKCS #12 파일에 결합”**의 내용을 참조하십시오.

나중에 **PEM** 형식의 **CA** 인증서와 **PKCS #12** 형식의 사용자 인증서를 브라우저로 가져오려면 브라우저 구성의 절차에 따라 **인증서 인증을 활성화**하고 **ID 관리 사용자**로 인증서를 사용하여 **ID 관리 웹 UI**에 인증합니다.

51.3.1. NSS 데이터베이스에서 PKCS #12 파일로 인증서 및 개인 키 내보내기

절차

1. **pk12util** 명령을 사용하여 **NSS** 데이터베이스에서 **PKCS12** 형식으로 인증서를 내보냅니다. 예를 들어 **~/certdb** 디렉터리에 저장된 **NSS** 데이터베이스에서 **some_user** 가 있는 인증서를 **~/some_user.p12** 파일로 내보내려면 다음을 수행합니다.

```
$ pk12util -d ~/certdb -o ~/some_user.p12 -n some_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

2.

.p12 파일에 적절한 권한을 설정합니다.

```
# chmod 600 ~/some_user.p12
```

PKCS #12 파일에도 개인 키가 포함되어 있으므로 다른 사용자가 파일을 사용하지 못하도록 보호해야 합니다. 그렇지 않으면 사용자를 가장할 수 있습니다.

51.3.2. 인증서와 개인 키 PEM 파일을 PKCS #12 파일에 결합

이 섹션에서는 인증서와 별도의 **PEM** 파일에 저장된 해당 키를 **PKCS #12** 파일로 결합하는 방법에 대해 설명합니다.

절차

•

certfile.cer에 저장된 인증서와 **certfile.key**에 저장된 키를 인증서와 키를 모두 포함하는 **certfile.p12** 파일에 결합하려면 다음을 수행합니다.

```
$ openssl pkcs12 -export -in certfile.cer -inkey certfile.key -out certfile.p12
```

51.4. IDM의 인증서 관련 명령 및 형식

테이블 **IdM 인증서 명령 및 형식**은 허용 가능한 형식으로 **IdM**에 인증서 관련 명령을 표시합니다.

표 51.2. IdM 인증서 명령 및 형식

명령	허용 가능한 형식	참고
ipa user-add-cert some_user --certificate	base64 PEM 인증서	
ipa-server-certinstall	PEM 및 DER 인증서, PKCS#7 인증서 체인, PKCS#8 및 원시 개인 키; PKCS#12 인증서 및 개인 키	
ipa-cacert-manage install	DER; PEM; PKCS#7	

명령	허용 가능한 형식	참고
ipa-cacert-manage reboot --external-cert-file	PEM 및 DER 인증서, PKCS#7 인증서 체인	
ipa-ca-install --external-cert-file	PEM 및 DER 인증서, PKCS#7 인증서 체인	
ipa cert-show <cert serial> --certificate-out /path/to/file.pem	해당 없음	인증서가 <cert_serial> 일련 번호가 있는 PEM 인코딩 file.pem 파일을 생성합니다.
ipa cert-show <cert serial> --certificate-out /path/to/file.pem	해당 없음	인증서가 <cert_serial> 일련 번호가 있는 PEM 인코딩 file.pem 파일을 생성합니다. chain 옵션을 사용하는 경우 PEM 파일에는 인증서 체인을 포함한 인증서가 포함되어 있습니다.
ipa cert-request --certificate-out=FILE /path/to/req.csr	해당 없음	새 인증서를 사용하여 PEM 형식으로 req.csr 파일을 만듭니다.
ipa cert-request --certificate-out=FILE /path/to/req.csr	해당 없음	새 인증서를 사용하여 PEM 형식으로 req.csr 파일을 만듭니다. chain 옵션을 사용하는 경우 PEM 파일에는 인증서 체인을 포함한 인증서가 포함되어 있습니다.

52장. IDENTITY MANAGEMENT에서 인증서 프로파일 생성 및 관리

인증서 프로파일은 인증서에 서명하여 **CSR**(인증서 서명 요청)이 허용되는지 여부를 판단할 때 **CA**(인증 기관)에서 사용하며, 이 경우 인증서에 어떤 기능 및 확장 기능이 있는지 확인합니다. 인증서 프로파일은 특정 유형의 인증서 발행과 연결됩니다. 인증서 프로파일과 **CA ACL**(액세스 제어 목록)을 결합하면 사용자 정의 인증서 프로파일에 대한 액세스를 정의하고 제어할 수 있습니다.

인증서 프로파일을 생성하는 방법을 설명하는 절차에서는 **S/MIME** 인증서를 예제로 사용합니다. 일부 이메일 프로그램은 **S/MIME**(Secure Multipurpose Internet Mail Extension) 프로토콜을 사용하여 디지털 서명 및 암호화된 이메일을 지원합니다. **S/MIME**를 사용하여 이메일 메시지에 서명 또는 암호화하려면 **S/MIME** 인증서가 있도록 메시지 발신자가 필요합니다.

- [인증서 프로파일이란 무엇입니까?](#)
- [인증서 프로파일 생성](#)
- [CA 액세스 제어 목록이란 무엇입니까?](#)
- [인증서 프로파일에 대한 액세스를 제어하는 CA ACL 정의](#)
- [인증서 프로파일 및 CA ACL을 사용하여 인증서 발행](#)
- [인증서 프로파일 수정](#)
- [인증서 프로파일 구성 매개변수](#)

52.1. 인증서 프로파일이란 무엇입니까?

인증서 프로파일을 사용하여 다음과 같은 인증서 발급에 대한 제약 조건 및 인증서 내용을 확인할 수 있습니다.

- 인증서 서명 요청을 설정하는 데 사용할 서명 알고리즘입니다.

- 인증서의 기본 유효 기간입니다.
- 해지 이유는 인증서를 취소하는 데 사용할 수 있습니다.
- 주체의 일반 이름이 **subject** 대체 이름 필드에 복사되는 경우.
- 인증서에 있어야 하는 기능 및 확장 기능.

단일 인증서 프로파일은 특정 유형의 인증서 발행과 연결됩니다. **IdM**에서 사용자, 서비스 및 호스트에 대해 다양한 인증서 프로파일을 정의할 수 있습니다. **IdM**에는 기본적으로 다음 인증서 프로파일이 포함되어 있습니다.

- **calPAserviceCert**
- **IECUserRoles**
- **KDCs_PKINIT_Certs** (내부에서 사용)

또한 사용자 지정 프로파일을 생성하고 가져올 수 있으므로 특정 용도로 인증서를 발행할 수 있습니다. 예를 들어 특정 프로파일 사용을 하나의 사용자 또는 하나의 그룹으로만 제한하여 다른 사용자 및 그룹이 해당 프로파일을 사용하지 못하도록 하여 인증을 위한 인증서를 발행할 수 있습니다. 사용자 정의 인증서 프로파일을 생성하려면 **ipa certprofile** 명령을 사용합니다.

추가 리소스

- **ipa certprofile** 명령에 대한 자세한 내용을 보려면 **ipa help certprofile** 명령을 실행합니다.

52.2. 인증서 프로파일 생성

다음 절차에서는 **S/MIME** 인증서를 요청하는 프로파일 구성 파일을 생성하여 명령줄을 통해 인증서 프로파일을 생성하는 방법을 설명합니다.

절차

1.

기존 기본 프로필을 복사하여 사용자 정의 프로필을 생성합니다.

```
$ ipa certprofile-show --out smime.cfg calPAserviceCert
-----
Profile configuration stored in file 'smime.cfg'
-----
Profile ID: calPAserviceCert
Profile description: Standard profile for network services
Store issued certificates: TRUE
```

2.

텍스트 편집기에서 새로 생성된 프로필 구성 파일을 엽니다.

```
$ vi smime.cfg
```

3.

Profile ID 를 프로필 사용을 반영하는 이름으로 변경합니다(예: **smime**).



참고

새로 생성된 프로필을 가져오는 경우 **profileid** 필드가 있으면 명령줄에 지정된 ID와 일치해야 합니다.

4.

Extended Key Usage(확장 키 사용) 구성을 업데이트합니다. 기본 확장 키 사용 확장 구성은 TLS 서버 및 클라이언트 인증을 위한 것입니다. 예를 들어 **S/MIME**의 경우 이메일 보호를 위해 확장 키 사용량을 구성해야 합니다.

```
policyset.serverCertSet.7.default.params.exKeyUsageOIDs=1.3.6.1.5.5.7.3.4
```

5.

새 프로필을 가져옵니다.

```
$ ipa certprofile-import smime --file smime.cfg \
--desc "S/MIME certificates" --store TRUE
-----
Imported profile "smime"
-----
Profile ID: smime
Profile description: S/MIME certificates
Store issued certificates: TRUE
```

검증 단계

- 새 인증서 프로파일을 가져왔는지 확인합니다.

```
$ ipa certprofile-find
```

```
-----  
4 profiles matched  
-----
```

```
Profile ID: calPAserviceCert
```

```
Profile description: Standard profile for network services
```

```
Store issued certificates: TRUE
```

```
Profile ID: IECUserRoles
```

```
Profile description: User profile that includes IECUserRoles extension from request
```

```
Store issued certificates: TRUE
```

```
Profile ID: KDCs_PKINIT_Certs
```

```
Profile description: Profile for PKINIT support by KDCs
```

```
Store issued certificates: TRUE
```

```
Profile ID: smime
```

```
Profile description: S/MIME certificates
```

```
Store issued certificates: TRUE
```

```
-----  
Number of entries returned 4  
-----
```

추가 리소스

- **certprofile** 플러그인에 대한 자세한 내용을 보려면 **ipa help certprofile** 명령을 실행합니다.
- 확장 키 사용 확장에 대한 자세한 내용은 [RFC 5280, 4.2.1.12](#) 절을 참조하십시오.

52.3. CA 액세스 제어 목록이란 무엇입니까?

CA ACL(인증 기관 액세스 제어 목록) 규칙은 어떤 주체에 인증서를 발급하는 데 사용할 수 있는 프로파일을 정의합니다. **CA ACL**을 사용하여 이 작업을 수행할 수 있습니다. 예를 들면 다음과 같습니다.

- 특정 프로파일이 있는 인증서를 발행할 수 있는 사용자, 호스트 또는 서비스 결정
- 인증서를 발행할 수 있는 **IdM** 인증 기관 또는 하위 **CA** 확인

예를 들어 **CA ACL**을 사용하면 런던 사무소에 있는 사무실에서 근무하는 직원을 위해 설계된 프로파일

사용을 런던 사무소 관련 **IdM** 사용자 그룹의 멤버인 사용자로만 제한할 수 있습니다.

CA ACL 규칙 관리를 위한 **ipa caacl** 유틸리티를 사용하면 권한이 있는 사용자는 지정된 **CA ACL**을 추가, 표시, 수정 또는 삭제할 수 있습니다.

추가 리소스

- **ipa caacl** 명령에 대한 자세한 내용을 보려면 **ipa help caacl** 명령을 실행합니다.

52.4. 인증서 프로필에 대한 액세스를 제어하는 **CA ACL** 정의

다음 절차에서는 **caacl** 유틸리티를 사용하여 사용자 정의 인증서 프로필에 대한 그룹 액세스를 허용하는 **ACL(CA Access Control List)** 규칙을 정의하는 방법을 설명합니다. 이 경우 절차에서는 **smime** 인증서 프로필에 대해 해당 그룹의 사용자가 액세스할 수 있도록 **S/MIME** 사용자 그룹 및 **CA ACL**을 생성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자의 자격 증명을 획득했는지 확인합니다.

절차

1. 인증서 프로필 사용자에게 새 그룹을 생성합니다.

```
$ ipa group-add smime_users_group
-----
Added group "smime users group"
-----
Group name: smime_users_group
GID: 75400001
```

2. **smime_user_group** 그룹에 추가할 새 사용자를 생성합니다.

```
$ ipa user-add smime_user
First name: smime
Last name: user
-----
Added user "smime_user"
-----
User login: smime_user
First name: smime
```

```

Last name: user
Full name: smime user
Display name: smime user
Initials: TU
Home directory: /home/smime_user
GECOS: smime user
Login shell: /bin/sh
Principal name: smime_user@IDM.EXAMPLE.COM
Principal alias: smime_user@IDM.EXAMPLE.COM
Email address: smime_user@idm.example.com
UID: 1505000004
GID: 1505000004
Password: False
Member of groups: ipausers
Kerberos keys available: False

```

3.

smime_user 를 **smime_users_group** 그룹에 추가합니다.

```

$ ipa group-add-member smime_users_group --users=smime_user
Group name: smime_users_group
GID: 1505000003
Member users: smime_user
-----
Number of members added 1
-----

```

4.

그룹의 사용자가 인증서 프로필에 액세스할 수 있도록 **CA ACL**을 생성합니다.

```

$ ipa caacl-add smime_acl
-----
Added CA ACL "smime_acl"
-----
ACL name: smime_acl
Enabled: TRUE

```

5.

CA ACL에 사용자 그룹을 추가합니다.

```

$ ipa caacl-add-user smime_acl --group smime_users_group
ACL name: smime_acl
Enabled: TRUE
User Groups: smime_users_group
-----
Number of members added 1
-----

```

6.

CA ACL에 인증서 프로필을 추가합니다.

```
$ ipa caacl-add-profile smime_acl --certprofile smime
ACL name: smime_acl
Enabled: TRUE
Profiles: smime
User Groups: smime_users_group
-----
Number of members added 1
-----
```

검증 단계

- 생성한 **CA ACL**의 세부 정보를 확인합니다.

```
$ ipa caacl-show smime_acl
ACL name: smime_acl
Enabled: TRUE
Profiles: smime
User Groups: smime_users_group
...
```

추가 리소스

- **ipa** 도움말 페이지를 참조하십시오.
- **ipa caacl** 명령에 대한 자세한 내용은 **ipa help caacl** 명령을 참조하십시오.

52.5. 인증서 프로파일 및 **CA ACL**을 사용하여 인증서 발행

CA ACL(인증 기관 액세스 제어 목록)에서 허용한 경우 인증서 프로ファイルを 사용하여 인증서를 요청할 수 있습니다. 다음 절차에서는 **CA ACL**을 통해 액세스 권한이 부여된 사용자 정의 인증서 프로 파일을 사용하여 사용자의 **S/MIME** 인증서를 요청하는 방법을 설명합니다.

사전 요구 사항

- 인증서 프로 파일이 생성되었습니다.
- 사용자가 필요한 인증서 프로 파일을 사용하여 인증서를 요청할 수 있는 **CA ACL**이 생성되었습니다.

참고

사용자가 **cert-request** 명령을 수행하는지 **CA ACL** 검사를 바이패스할 수 있습니다.

- 는 **admin** 사용자입니다.
- **CA ACL** 권한을 무시하는 요청 인증서가 있습니까.

절차

1.

사용자에 대한 인증서 요청을 생성합니다. 예를 들어 **OpenSSL**을 사용합니다.

```
$ openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout private.key -out
cert.csr -subj '/CN=smime_user'
```

2.

IdM CA에서 사용자에게 대한 새 인증서를 요청합니다.

```
$ ipa cert-request cert.csr --principal=smime_user --profile-id=smime
```

선택적으로 명령에 **--ca sub-CA_name** 옵션을 전달하여 루트 **CA** 대신 하위 **CA**에서 인증서를 요청합니다.

검증 단계

•

새로 발급한 인증서가 사용자에게 할당되었는지 확인합니다.

```
$ ipa user-show user
User login: user
...
Certificate: MIICfzCCAWcCAQA...
...
```

추가 리소스

•

ipa(a) 도움말 말 페이지를 참조하십시오.

•

ipa user-show 명령에 대한 자세한 내용은 **ipa help user-show** 명령을 참조하십시오.

- ***ipa cert-request* 명령에 대한 자세한 내용은 *ipa help cert-request* 명령을 참조하십시오.**
- ***openssl(1ssl)* 도움말 페이지를 참조하십시오.**

52.6. 인증서 프로필 수정

다음 절차에서는 ***ipa certprofile-mod*** 명령을 사용하여 명령줄을 통해 인증서 프로필을 직접 수정하는 방법을 설명합니다.

절차

1.

수정할 인증서 프로필의 인증서 프로필 ID를 확인합니다. 현재 IdM에 저장된 모든 인증서 프로필을 표시하려면 다음을 수행합니다.

```
# ipa certprofile-find
```

```
-----
4 profiles matched
-----
Profile ID: calPAserviceCert
Profile description: Standard profile for network services
Store issued certificates: TRUE

Profile ID: IECUserRoles
...

Profile ID: smime
Profile description: S/MIME certificates
Store issued certificates: TRUE
-----
Number of entries returned
-----
```

2.

인증서 프로필 설명을 수정합니다. 예를 들어 기존 프로필을 사용하여 **S/MIME** 인증서에 대한 사용자 정의 인증서 프로필을 생성한 경우 새 사용량에 따라 설명을 변경합니다.

```
# ipa certprofile-mod smime --desc "New certificate profile description"
-----
Modified Certificate Profile "smime"
-----
Profile ID: smime
Profile description: New certificate profile description
Store issued certificates: TRUE
```


3.

텍스트 편집기에서 고객 인증서 프로파일 파일을 열고 요구 사항에 맞게 를 수정합니다.

```
# vi smime.cfg
```

인증서 프로파일 구성 파일에서 구성할 수 있는 옵션에 대한 자세한 내용은 [인증서 프로파일 구성 매개 변수](#)를 참조하십시오.

4.

기존 인증서 프로파일 구성 파일을 업데이트합니다.

```
# ipa certprofile-mod _profile_ID_ --file=smime.cfg
```

검증 단계

•

인증서 프로파일 이 업데이트되었는지 확인합니다.

```
$ ipa certprofile-show smime
Profile ID: smime
Profile description: New certificate profile description
Store issued certificates: TRUE
```

추가 리소스

•

[ipa\(a\)](#) 도움말 페이지를 참조하십시오.

•

[ipa certprofile-mod](#) 명령에 대한 자세한 내용은 [ipa help certprofile-mod](#) 명령을 참조하십시오.

52.7. 인증서 프로파일 구성 매개변수

인증서 프로파일 구성 매개 변수는 CA 프로파일 디렉터리의 `profile_name.cfg` 파일에 `/var/lib/pki/pki-tomcat/ca/profiles/ca`에 저장됩니다. 프로파일의 모든 매개 변수(기본값, 입력, 출력 및 제약 조건)는 단일 정책 세트 내에 구성됩니다. 인증서 프로파일 에 설정된 정책에는 이름 `policyset.policyName.policyNumber`가 있습니다. 예를 들어 정책의 경우 `serverCertSet`를 설정합니다.

```
policyset.list=serverCertSet
policyset.serverCertSet.list=1,2,3,4,5,6,7,8
policyset.serverCertSet.1.constraint.class_id=subjectNameConstraintImpl
policyset.serverCertSet.1.constraint.name=Subject Name Constraint
policyset.serverCertSet.1.constraint.params.pattern=CN=[^,]+,.
```

```

policyset.serverCertSet.1.constraint.params.accept=true
policyset.serverCertSet.1.default.class_id=subjectNameDefaultImpl
policyset.serverCertSet.1.default.name=Subject Name Default
policyset.serverCertSet.1.default.params.name=CN=$request.req_subject_name.cn$, OU=pki-
ipa, O=IPA
policyset.serverCertSet.2.constraint.class_id=validityConstraintImpl
policyset.serverCertSet.2.constraint.name=Validity Constraint
policyset.serverCertSet.2.constraint.params.range=740
policyset.serverCertSet.2.constraint.params.notBeforeCheck=false
policyset.serverCertSet.2.constraint.params.notAfterCheck=false
policyset.serverCertSet.2.default.class_id=validityDefaultImpl
policyset.serverCertSet.2.default.name=Validity Default
policyset.serverCertSet.2.default.params.range=731
policyset.serverCertSet.2.default.params.startTime=0

```

각 정책 세트에는 평가해야 하는 순서대로 정책 ID 번호별로 인증서 프로필에 대해 구성된 정책 목록이 포함되어 있습니다. 서버는 수신하는 각 요청에 대해 설정된 각 정책을 평가합니다. 단일 인증서 요청이 수신되면 하나의 세트가 평가되고 프로필의 다른 모든 세트는 무시됩니다. 이중 키 쌍이 발행되면 첫 번째 정책 집합이 첫 번째 인증서 요청에 대해 평가되고, 두 번째 키 쌍은 두 번째 인증서 요청에 대해 평가됩니다. 이중 키 쌍을 발행할 때 단일 인증서 또는 두 개 이상의 집합을 발행할 때 두 개 이상의 정책 세트가 필요하지 않습니다.

표 52.1. 인증서 프로필 구성 파일 매개변수

매개변수	설명
Desc	최종 엔티티 페이지에 표시되는 인증서 프로필에 대한 무료 텍스트 설명입니다. 예를 들어, desc=이 인증서 프로필은 에이전트 인증을 사용하여 서버 인증서를 등록하기 위한 것입니다.
활성화	엔드 엔티티 페이지를 통해 액세스할 수 있도록 프로필을 활성화합니다. 예를 들면 enable=true 입니다.
auth.instance_id	인증서 요청을 인증하는 데 사용할 인증 관리자 플러그인을 설정합니다. 자동 등록을 위해 인증에 성공하면 CA에서 즉시 인증서를 발행합니다. 인증에 실패하거나 인증 플러그인이 지정되지 않은 경우 에이전트에서 수동으로 승인하도록 요청이 대기 중입니다. 예를 들면 auth.instance_id=AgentCertAuth 입니다.

매개변수	설명
authz.acl	권한 부여 제한 조건을 지정합니다. 이는 그룹 ACL(평가 제어 목록)을 설정하는 데 주로 사용됩니다. 예를 들어 caCMCUserCert 매개변수를 사용하려면 CMC 요청의 서명자가 Certificate Manager Agents 그룹에 속해야 합니다. authz.acl=group="인증 관리자 에이전트 디렉터리 기반 사용자 인증서 갱신에서 이 옵션은 원래 요청자 및 현재 인증된 사용자가 동일한지 확인하는 데 사용됩니다. 권한을 평가하려면 엔터티에서 인증(바인드 또는 기본적으로 시스템에 로그인)해야 합니다.
name	인증서 프로파일의 이름입니다. 예를 들면 name=Agent-Authenticated 서버 인증서 등록입니다. 이 이름은 최종 사용자 등록 또는 갱신 페이지에 표시됩니다.
input.list	인증서 프로파일에 허용된 입력을 이름으로 나열합니다. 예를 들면 input.list=i1,i2입니다.
input.input_id.class_id	입력 ID(input.list에 나열된 입력 이름)를 통해 입력의 java 클래스 이름을 나타냅니다. 예를 들면 input.i1.class_id=certReqInputImpl입니다.
output.list	이름별로 인증서 프로파일에 사용할 수 있는 출력 형식을 나열합니다. 예를 들면 output.list=o1입니다.
output.output_id.class_id	output.list에서 이름이 지정된 출력 형식의 java 클래스 이름을 지정합니다. 예를 들면 output.o1.class_id=certOutputImpl입니다.
policyset.list	구성된 인증서 프로파일 규칙을 나열합니다. 이중 인증서의 경우 하나의 규칙 집합이 서명 키에 적용되고 다른 하나는 암호화 키에 적용됩니다. 단일 인증서는 하나의 인증서 프로파일 규칙 세트만 사용합니다. 예를 들면 policyset.list=serverCertSet입니다.
policyset.policyset_id.list	평가해야 하는 정책 ID 번호에 따라 인증서 프로파일에 대해 구성된 정책 내의 정책을 나열합니다. 예를 들면 policyset.serverCertSet.list=1,2,3,4,5,6,7,8입니다.
policyset.policyset_id.policy_number.constraint.class_id	프로파일 규칙에 구성된 기본값에 설정된 제약 조건 플러그인의 java 클래스 이름을 나타냅니다. 예를 들면 policyset.serverCertSet.1.constraint.class_id=subjectNameConstraintImpl입니다.

매개변수	설명
policyset.policyset_id.policy_number.constraint.name	제한 조건의 사용자 정의 이름을 제공합니다. 예를 들면 policyset.serverCertSet.1.constraint.name=Subject Name Constraint입니다.
policyset.policyset_id.policy_number.constraint.params.attribute	계약 조건에 허용된 속성의 값을 지정합니다. 가능한 속성은 제한 조건 유형에 따라 다릅니다. 예를 들어 policyset.serverCertSet.1.constraint.params.pattern=CN=.*입니다.
policyset.policyset_id.policy_number.default.class_id	프로필 규칙에 기본 세트의 java 클래스 이름을 제공합니다. For example, policyset.serverCertSet.1.default.class_id=userSubjectNameDefaultImpl
policyset.policyset_id.policy_number.default.name	은 사용자 정의 이름을 제공합니다. 예를 들어 policyset.serverCertSet.1.default.name=Subject Name Default
policyset.policyset_id.policy_number.default.params.attribute	기본값에 허용된 속성의 값을 지정합니다. 가능한 속성은 기본값 유형에 따라 다릅니다. 예를 들어 policyset.serverCertSet.1.default.params.name=CN=(Name)\$request.requestor_name\$입니다.

53장. IDM에서 인증서의 유효성 관리

IdM(Identity Management)에서는 향후 발급하려는 기존 인증서와 인증서의 유효성을 관리할 수 있지만 메서드는 다릅니다.

IdM CA에서 발급한 기존 인증서의 유효성 관리

IdM에서는 인증서 만료일을 확인하는 다음 방법을 사용할 수 있습니다.

- [IdM WebUI에서 만료 일자 보기](#)
- [CLI에서 만료 날짜 보기.](#)

다음과 같은 방법으로 **IdM CA**에서 발급한 기존 인증서의 유효성을 관리할 수 있습니다.

- 원래 **CSR**(인증서 서명 요청) 또는 개인 키에서 생성된 새 **CSR**를 사용하여 새 인증서를 요청하여 인증서를 갱신합니다. 다음 유틸리티를 사용하여 새 인증서를 요청할 수 있습니다.

certmonger

certmonger 를 사용하여 서비스 인증서를 요청할 수 있습니다. 인증서가 만료되기 전에 **certmonger** 는 인증서를 자동으로 갱신하므로 서비스 인증서의 유효성을 계속 유지할 수 있습니다. 자세한 내용은 [certmonger를 사용하여 서비스의 IdM 인증서 가져오기를](#) 참조하십시오.

certutil

certutil 을 사용하여 사용자, 호스트 및 서비스 인증서를 갱신할 수 있습니다. 사용자 인증서를 요청하는 방법에 대한 자세한 내용은 [새 사용자 인증서 요청 및 클라이언트로 내보내기](#)를 참조하십시오.

OpenSSL

openssl 을 사용하여 사용자, 호스트 및 서비스 인증서를 갱신할 수 있습니다.

- 인증서를 취소합니다. 자세한 내용은 다음을 참조하십시오.
 - [IdM WebUI를 사용하여 IdM CA가 통합된 인증서를 취소합니다.](#)

- **IdM CLI를 사용하여 IdM CA가 통합된 인증서를 취소합니다.**
- 인증서를 일시적으로 폐기한 경우 복원합니다. 자세한 내용은 다음을 참조하십시오.
 - **IdM WebUI를 사용하여 통합 IdM CA로 인증서 복원**
 - **IdM CLI를 사용하여 통합된 IdM CA로 인증서 복원.**

IdM CA에서 발급한 향후 인증서의 유효성 관리

IdM CA에서 발급한 향후 인증서의 유효성을 관리하려면 인증서 프로필을 수정, 가져오기 또는 생성합니다. 자세한 내용은 [Identity Management에서 인증서 프로필 생성 및 관리](#)를 참조하십시오.

53.1. 인증서 만료일 보기

53.1.1. IdM WebUI에서 인증서 만료일 보기

IdM WebUI를 사용하여 IdM CA에서 발급한 모든 인증서의 만료 날짜를 볼 수 있습니다.

사전 요구 사항

- 관리자의 자격 증명을 획득했는지 확인합니다.

절차

1. 인증 메뉴에서 인증서 > 인증서를 클릭합니다.
2. 인증서의 일련 번호를 클릭하여 인증서 정보 페이지를 엽니다.

그림 53.1. 인증서 목록

Certificates		
Subject ▼ Search Refresh + Issue		
☐	Serial Number	Subject
☐	1	CN=Certificate Authority,O=EXAMPLE.COM
☐	2	CN=OCSP Subsystem,O=EXAMPLE.COM
☐	3	CN=server.example.com,O=EXAMPLE.COM
☐	4	CN=CA Subsystem,O=EXAMPLE.COM

3.

인증서 정보 페이지에서 **Expires On** 정보를 찾습니다.

53.1.2. CLI에서 인증서 만료일 보기

CLI(명령줄 인터페이스)를 사용하여 인증서 만료일을 볼 수 있습니다.

절차

- **openssl** 유틸리티를 사용하여 사람이 읽을 수 있는 형식으로 파일을 엽니다.

```
$ openssl x509 -noout -text -in ca.pem
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: O = IDM.EXAMPLE.COM, CN = Certificate Authority
    Validity
      Not Before: Oct 30 19:39:14 2017 GMT
      Not After : Oct 30 19:39:14 2037 GMT
```

53.2. IDM CA가 통합된 인증서 취소

53.2.1. 인증서 해지 이유

취소된 인증서가 유효하지 않으며 인증에 사용할 수 없습니다. 이유 6을 제외하고 모든 취소는 영구적입니다. 자격증 보유.

기본 취소 이유는 0: **unspecified** 입니다.

표 53.1. 해지 사유

ID	이유	설명
0	지정되지 않음	
1	Key Compromized	인증서가 발급된 키는 더 이상 신뢰할 수 없습니다. 가능한 원인: 누락된 토큰, 부적절하게 액세스된 파일.
2	CA가 압축됨	인증서가 발급된 CA는 더 이상 신뢰할 수 없습니다.
3	제휴 변경됨	가능한 원인 : * 한 명이 퇴사했거나 다른 부서로 이전했습니다. * 호스트 또는 서비스가 폐기되고 있습니다.
4	대체됨	최신 인증서가 현재 인증서를 교체했습니다.
5	작업 중단	호스트 또는 서비스가 해제되고 있습니다.
6	인증서 보유	인증서가 일시적으로 폐기됩니다. 나중에 인증서를 복원할 수 있습니다.
8	CRL에서 제거	인증서는 CRL(인증서 취소 목록)에 포함되지 않습니다.
9	권한 보유	사용자, 호스트 또는 서비스는 더 이상 인증서를 사용할 수 없습니다.
10	servera(속성 기관) Compromise	AA 인증서는 더 이상 신뢰할 수 없습니다.

53.2.2. IdM WebUI를 사용하여 통합된 IdM CA로 인증서 해지

인증서의 개인 키를 분실한 경우 오용을 방지하기 위해 인증서를 취소해야 합니다. IdM WebUI를 사용하여 IdM CA에서 발급한 인증서를 취소하려면 이 절차를 완료합니다.

절차

1. **인증 > 인증서 > 인증서를 클릭합니다.**
2. **인증서의 일련 번호를 클릭하여 인증서 정보 페이지를 엽니다.**

그림 53.2. 인증서 목록

Certificates		
Subject ▼ Search  Refresh  + Issue		
☐	Serial Number	Subject
☐	1	CN=Certificate Authority,O=EXAMPLE.COM
☐	2	CN=OCSP Subsystem,O=EXAMPLE.COM
☐	3	CN=server.example.com,O=EXAMPLE.COM
☐	4	CN=CA Subsystem,O=EXAMPLE.COM

3.

인증서 정보 페이지에서 **Actions(작업) revoke Certificate(인증서 공개)** 를 클릭합니다.

4.

취소 이유를 선택하고 개정을 클릭합니다. 자세한 내용은 [53.2.1절. “인증서 해지 이유”](#) 을 참조하십시오.

53.2.3. IdM CLI를 사용하여 통합된 IdM CA로 인증서 해지

인증서의 개인 키를 분실한 경우 오용을 방지하기 위해 인증서를 취소해야 합니다. IdM CLI를 사용하여 IdM CA에서 발급한 인증서를 취소하려면 이 절차를 완료합니다.

절차

- **ipa cert-revoke** 명령을 사용하고 다음을 지정합니다.
 - 인증서 일련 번호
 - 취소 사유의 ID 번호입니다. 자세한 내용은 [53.2.1절. “인증서 해지 이유”](#) 을 참조하십시오.

예를 들어 이유 1로 인해 일련 번호가 1032 인 인증서를 취소하려면 다음을 수행합니다. 키 압축을 입력합니다.

```
$ ipa cert-revoke 1032 --revocation-reason=1
```

새 인증서를 요청하는 방법에 대한 자세한 내용은 다음 설명서를 참조하십시오.

- 새 사용자 인증서 요청 및 클라이언트로 내보내기
- **certmonger**를 사용하여 서비스에 대한 **IdM** 인증서 가져오기.

53.3. 통합된 IDM CA를 사용하여 인증서 복원

이유 6으로 인해 인증서를 해지한 경우: 인증서 보유, 인증서의 개인 키가 손상되지 않은 경우 다시 복원할 수 있습니다. 인증서를 복원하려면 다음 절차 중 하나를 사용하십시오.

- **IdM WebUI**를 사용하여 통합 **IdM CA**로 인증서를 복원합니다.
- **IdM CLI**를 사용하여 통합된 **IdM CA**로 인증서를 복원합니다.

53.3.1. IdM WebUI를 사용하여 통합된 IdM CA로 인증서 복원

IdM WebUI를 사용하여 **Reason 6**로 인해 취소된 **IdM** 인증서를 복원하려면 이 절차를 완료합니다. 자격증 보유.

절차

1. 인증 메뉴에서 인증서 > 인증서를 클릭합니다.
2. 인증서의 일련 번호를 클릭하여 인증서 정보 페이지를 엽니다.

그림 53.3. 인증서 목록

Certificates		
Subject ▼		Search 🔍
		Refresh ↻ + Issue
☐	Serial Number	Subject
☐	1	CN=Certificate Authority,O=EXAMPLE.COM
☐	2	CN=OCSP Subsystem,O=EXAMPLE.COM
☐	3	CN=server.example.com,O=EXAMPLE.COM
☐	4	CN=CA Subsystem O=EXAMPLE.COM

3.

인증서 정보 페이지에서 **Actions(작업) Restore Certificate(인증서 복원)** 를 클릭합니다.

53.3.2. IdM CLI를 사용하여 통합된 IdM CA로 인증서 복원

IdM CLI를 사용하여 Reason 6로 인해 취소된 IdM 인증서를 복원하려면 이 절차를 완료합니다. 자격증 보유.

절차

- **ipa cert-remove-hold 명령을 사용하고 인증서 일련 번호를 지정합니다. 예를 들면 다음과 같습니다.**

```
$ ipa cert-remove-hold 1032
```

54장. 스마트 카드 인증을 위한 ID 관리 구성

스마트 카드를 기반으로 하는 인증은 암호에 대한 대안입니다. 개인 키와 인증서의 형태로 스마트 카드에 사용자 자격 증명을 저장할 수 있으며 특수 소프트웨어 및 하드웨어를 사용하여 액세스할 수 있습니다. 스마트 카드를 리더 또는 **USB** 포트에 저장하고 암호를 제공하는 대신 스마트 카드의 **PIN** 코드를 제공합니다.

IdM(Identity Management)은 다음을 사용하여 스마트 카드 인증을 지원합니다.

- **IdM 인증 기관에서 발급한 사용자 인증서**
- **외부 인증 기관에서 발급한 사용자 인증서**

이 사용자 사례에서는 두 유형의 인증서 모두에 대해 **IdM**에 스마트 카드 인증을 설정하는 방법을 보여줍니다. 사용자 사례에서 **the smartcard_ca.pem CA** 인증서는 신뢰할 수 있는 외부 인증 기관의 인증서를 포함하는 파일입니다.

사용자 사례에는 다음 모듈이 포함되어 있습니다.

- **스마트 카드 인증을 위해 IdM 서버 구성**
- **스마트 카드 인증을 위해 IdM 클라이언트 구성**
- **IdM 웹 UI의 사용자 항목에 인증서 추가**
- **IdM CLI의 사용자 항목에 인증서 추가**
- **스마트 카드 관리 및 사용을 위한 도구 설치**
- **스마트 카드에 인증서 저장**

- 스마트 카드로 **IdM**에 로그인
- 스마트 카드 인증을 사용하여 **GDM** 액세스 구성
- 스마트 카드 인증을 사용하여 **su** 액세스 구성

54.1. 스마트 카드 인증을 위해 **IDM** 서버 구성

LDAP 고유 이름(**DN**)이 **CN=Certificate Authority,DC=EXAMPLE,DC=ORG**인 **EXAMPLE,DC=ORG** 도메인의 인증 기관에서 발급한 사용자에게 대해 스마트 카드 인증을 활성화하려면 **IdM** 서버를 구성하는 스크립트를 사용하여 실행할 수 있도록 권한의 인증서를 가져와야 합니다. 예를 들어, 기관에서 인증서를 발급한 웹 페이지에서 인증서를 다운로드할 수 있습니다. 자세한 내용은 [인증서 인증을 활성화하기 위해 브라우저 구성의 1단계 - 4a](#)를 참조하십시오.

IdM 인증 기관에서 인증서를 발급한 **IdM** 사용자에게 대해 스마트 카드 인증을 사용하려면 **IdM CA**가 실행 중인 **IdM** 서버의 **/etc/ipa/ca.crt** 파일에서 **CA** 인증서를 가져옵니다.

이 섹션에서는 스마트 카드 인증을 위해 **IdM** 서버를 구성하는 방법에 대해 설명합니다. 먼저, **PEM** 형식의 **CA** 인증서가 있는 파일을 가져온 다음 기본 제공 **ipa-adviser** 스크립트를 실행합니다. 마지막으로 시스템 구성을 다시 로드합니다.

사전 요구 사항

- **IdM** 서버에 대한 루트 액세스 권한이 있습니다.
- 루트 **CA** 인증서와 하위 **CA** 인증서가 있습니다.

절차

1. 설정을 수행할 디렉터리를 생성합니다.

```
[root@server]# mkdir ~/SmartCard/
```

2. 디렉터리로 이동합니다.

```
[root@server]# cd ~/SmartCard/
```

3.

PEM 형식의 파일에 저장된 관련 **CA** 인증서를 가져옵니다. **CA** 인증서가 **DER**와 같은 다른 형식의 파일에 저장된 경우 이를 **PEM** 형식으로 변환합니다. **IdM** 인증 기관 인증서는 **/etc/ipa/ca.crt** 파일에 있습니다.

DER 파일을 **PEM** 파일로 변환합니다.

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

4.

편의를 위해 구성을 수행하려는 디렉터리에 인증서를 복사합니다.

```
[root@server SmartCard]# cp /etc/ipa/ca.crt ~/SmartCard/
[root@server SmartCard]# cp /tmp/smartcard_ca.pem ~/SmartCard/
```

5.

선택적으로 외부 인증 기관의 인증서를 사용하는 경우 **openssl x509** 유틸리티를 사용하여 **PEM** 형식으로 파일의 내용을 확인하여 발급자 및 주체 값이 올바른지 확인합니다.

```
[root@server SmartCard]# openssl x509 -noout -text -in smartcard_ca.pem | more
```

6.

관리자 권한을 사용하여 기본 제공 **ipa-adviser** 유틸리티로 구성 스크립트를 생성합니다.

```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# sudo ipa-adviser config-server-for-smart-card-auth > config-server-for-smart-card-auth.sh
```

config-server-for-smart-card-auth.sh 스크립트는 다음 작업을 수행합니다.

- **IdM Apache HTTP** 서버를 구성합니다.
- **KDC**(키 배포 센터)에서 **Kerberos**(**PKINIT**)의 초기 인증에 대한 공개 키 암호화를 활성화합니다.
- 스마트 카드 권한 부여 요청을 수락하도록 **IdM** 웹 **UI**를 구성합니다.

7.

스크립트를 실행하여 루트 **CA** 및 하위 **CA** 인증서를 포함하는 **PEM** 파일을 인수로 추가합니

다.

```
[root@server SmartCard]# chmod +x config-server-for-smart-card-auth.sh
[root@server SmartCard]# ./config-server-for-smart-card-auth.sh smartcard_ca.pem
ca.crt
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```



참고

루트 **CA** 인증서를 하위 **CA** 인증서 전에 인수로 추가하고 **CA** 또는 하위 **CA** 인증서가 만료되지 않았는지 확인합니다.

8.

선택적으로 사용자 인증서를 발행한 인증 기관에서 **OCSP(Online Certificate Status Protocol)** 응답자를 제공하지 않는 경우 **IdM 웹 UI**에 대한 인증을 **OCSP** 검사를 비활성화해야 할 수도 있습니다.

a.

`/etc/httpd/conf.d/ssl.conf` 파일에서 **SSL_OCSP_Enable** 매개변수를 **off**로 설정합니다.

SSL_OCSP_Enable off

b.

변경 사항을 즉시 적용하려면 **Apache** 데몬(**httpd**)을 다시 시작하십시오.

```
[root@server SmartCard]# sudo systemctl restart httpd
```



주의

IdM CA에서 발급한 사용자 인증서만 사용하는 경우 **OCSP** 검사를 비활성화하지 마십시오. **OCSP** 응답자는 **IdM**의 일부입니다.

OCSP 검사를 계속 활성화하는 방법에 대한 지침이 있지만 사용자 인증서가 **OCSP** 서비스 요청을 수신 대기하는 **CA**에 대한 정보가 포함되지 않은 경우 **IdM** 서버가 사용자 인증서를 거부하

지 않는 경우 [Apache mod_ssl 구성 옵션의 SSLOCSPDefaultResponder](#) 지시문을 참조하십시오.

이제 서버는 스마트 카드 인증을 위해 구성됩니다.



참고

전체 토폴로지에서 스마트 카드 인증을 사용하려면 각 **IdM** 서버에서 절차를 실행합니다.

54.2. 스마트 카드 인증을 위해 IDM 클라이언트 구성

이 섹션에서는 스마트 카드 인증을 위해 **IdM** 클라이언트를 구성하는 방법에 대해 설명합니다. 인증을 위해 스마트 카드를 사용하는 동안 연결하려는 각 **IdM** 시스템, 클라이언트 또는 서버에서 절차를 실행해야 합니다. 예를 들어 호스트 **A**에서 호스트 **B**로의 **ssh** 연결을 활성화하려면 스크립트를 호스트 **B**에서 실행해야 합니다.

관리자는 을 사용하여 스마트 카드 인증을 활성화하려면 다음 절차를 실행합니다.

- **ssh** 프로토콜

자세한 내용은 [스마트 카드 인증을 사용하여 SSH 액세스 구성](#)을 참조하십시오.

- 콘솔 로그인

- **GDM**(Gnome Display Manager)

- **su** 명령

이 절차는 **IdM** 웹 UI를 인증하는 데 필요하지 않습니다. **IdM** 웹 UI에 인증에는 두 개의 호스트가 있으며, 둘 다 **IdM** 클라이언트여야 합니다.

- 브라우저가 실행 중인 시스템 - **IdM** 도메인 외부일 수 있음

- **httpd** 가 실행 중인 **IdM** 서버

다음 절차에서는 **IdM** 서버가 아닌 **IdM** 클라이언트에서 스마트 카드 인증을 구성한다고 가정합니다. 따라서 구성 스크립트를 생성하는 **IdM** 서버와 스크립트를 실행할 **IdM** 클라이언트라는 두 대의 컴퓨터가 필요합니다.

사전 요구 사항

- 스마트 카드 인증을 위해 **IdM** 서버 구성에 설명된 대로 **IdM** 서버가 스마트 카드 인증용으로 구성되어 있습니다.
- **IdM** 서버와 **IdM** 클라이언트에 루트 액세스 권한이 있습니다.
- 루트 **CA** 인증서 및 하위 **CA** 인증서에 액세스할 수 있습니다.
- 원격 사용자가 성공적으로 로그인할 수 있도록 **mkhomedir** 옵션을 사용하여 **IdM** 클라이언트를 설치했습니다. 홈 디렉토리를 만들지 않으면 기본 로그인 위치는 **root**입니다.

절차

1. **IdM** 서버에서 관리자 권한을 사용하여 **ipa-adviser** 로 구성 스크립트를 생성합니다.

```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# ipa-adviser config-client-for-smart-card-auth > config-client-for-smart-card-auth.sh
```

config-client-for-smart-card-auth.sh 스크립트는 다음 작업을 수행합니다.

- 스마트 카드 데몬을 구성합니다.
- 시스템 전체 신뢰 저장소를 설정합니다.
- 데스크탑에 스마트 카드 로그인을 허용하도록 **SSSD(System Security Services Daemon)**를 구성합니다.

2.

IdM 서버에서 스크립트를 IdM 클라이언트 시스템에서 선택한 디렉터리에 복사합니다.

```
[root@server SmartCard]# scp config-client-for-smart-card-auth.sh
root@client.idm.example.com:/root/SmartCard/
Password:
config-client-for-smart-card-auth.sh    100% 2419    3.5MB/s  00:00
```

3.

IdM 서버에서 편의를 위해 이전 단계에서 사용한 것과 같이 IdM 클라이언트 시스템의 동일한 디렉터리에 사용하기 위해 PEM 형식의 CA 인증서 파일을 복사합니다.

```
[root@server SmartCard]# scp {smartcard_ca.pem,ca.crt}
root@client.idm.example.com:/root/SmartCard/
Password:
smartcard_ca.pem          100% 1237    9.6KB/s  00:00
ca.crt                   100% 2514   19.6KB/s  00:00
```

4.

클라이언트 시스템에서 스크립트를 실행하여 CA 인증서를 인수로 포함하는 PEM 파일을 추가합니다.

```
[root@client SmartCard]# kinit admin
[root@client SmartCard]# chmod +x config-client-for-smart-card-auth.sh
[root@client SmartCard]# ./config-client-for-smart-card-auth.sh smartcard_ca.pem
ca.crt
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```



참고

루트 CA 인증서를 하위 CA 인증서 전에 인수로 추가하고 CA 또는 하위 CA 인증서가 만료되지 않았는지 확인합니다.

이제 클라이언트는 스마트 카드 인증을 위해 구성됩니다.

54.3. IDM 웹 UI의 사용자 항목에 인증서 추가

이 절차에서는 IdM 웹 UI의 사용자 항목에 외부 인증서를 추가하는 방법을 설명합니다.

전체 인증서를 업로드하는 대신, 인증서 매핑 데이터를 **IdM**의 사용자 항목에 업로드할 수도 있습니다. 전체 인증서 또는 인증서 매핑 데이터를 포함하는 사용자 항목을 해당 인증서 매핑 규칙과 함께 사용하여 시스템 관리자를 위한 스마트 카드 인증 구성을 용이하게 할 수 있습니다. 자세한 내용은 [스마트 카드의 인증 구성을 위한 인증서 매핑 규칙을 참조하십시오.](#)



참고

IdM 인증 기관에서 사용자의 인증서를 발급한 경우 인증서는 이미 사용자 항목에 저장되어 있으며 이 섹션을 건너뛸 수 있습니다.

사전 요구 사항

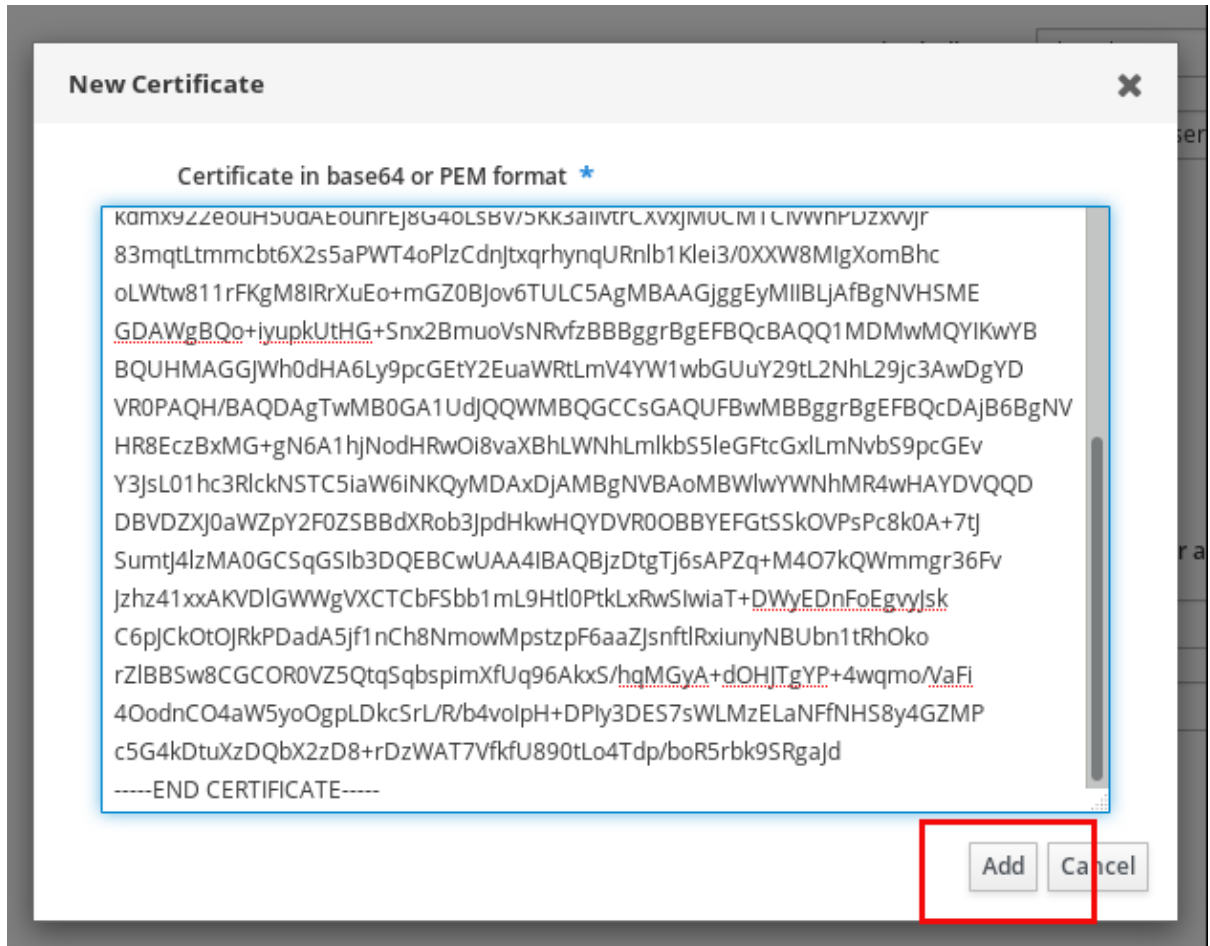
- 사용자 항목에 추가할 인증서가 있습니다.

절차

1. 다른 사용자에게 인증서를 추가하려면 **IdM** 웹 UI에 관리자로 로그인합니다. 자체 프로필에 인증서를 추가하는 데 관리자의 자격 증명이 필요하지 않습니다.
2. 사용자 → 활성 사용자 → **sc_user** 로 이동합니다.
3. **Certificate** (인증서) 옵션을 찾아 **Add**(추가)를 클릭합니다.
4. 명령줄 인터페이스에서 **cat** 유틸리티 또는 텍스트 편집기를 사용하여 **PEM** 형식으로 인증서를 표시합니다.


```
[user@client SmartCard]$ cat testuser.crt
```
5. CLI에서 인증서를 복사하여 웹 UI에서 연 창에 붙여넣습니다.
6. 추가를 클릭합니다.

그림 54.1. IdM 웹 UI에서 새 인증서 추가



이제 **sc_user** 항목에 외부 인증서가 포함됩니다.

54.4. IDM CLI의 사용자 항목에 인증서 추가

이 절차에서는 **IdM CLI**의 사용자 항목에 외부 인증서를 추가하는 방법을 설명합니다.

전체 인증서를 업로드하는 대신, 인증서 매핑 데이터를 **IdM**의 사용자 항목에 업로드할 수도 있습니다. 전체 인증서 또는 인증서 매핑 데이터를 포함하는 사용자 항목을 해당 인증서 매핑 규칙과 함께 사용하여 시스템 관리자를 위한 스마트 카드 인증 구성을 용이하게 할 수 있습니다. 자세한 내용은 [스마트 카드의 인증 구성을 위한 인증서 매핑 규칙](#)을 참조하십시오.



참고

IdM 인증 기관에서 사용자의 인증서를 발급한 경우 인증서는 이미 사용자 항목에 저장되어 있으며 이 섹션을 건너뛸 수 있습니다.

사전 요구 사항

- 사용자 항목에 추가할 인증서가 있습니다.

절차

1. 다른 사용자에게 인증서를 추가하려면 **IdM CLI**에 관리자로 로그인합니다.

```
[user@client SmartCard]$ kinit admin
```

자체 프로필에 인증서를 추가하는 데 관리자의 인증 정보가 필요하지 않습니다.

```
[user@client SmartCard]$ kinit sc_user
```

2. **ipa user-add-cert** 명령에서 필요한 형식인 헤더와 바닥글이 제거되고 한 줄로 연결된 인증서가 포함된 환경 변수를 생성합니다.

```
[user@client SmartCard]$ export CERT=`openssl x509 -outform der -in testuser.crt | base64 -w0 -`
```

testuser.crt 파일의 인증서는 **PEM** 형식이어야 합니다.

3. **ipa user-add-cert** 명령을 사용하여 **sc_user**의 프로필에 인증서를 추가합니다.

```
[user@client SmartCard]$ ipa user-add-cert sc_user --certificate=$CERT
```

이제 **sc_user** 항목에 외부 인증서가 포함됩니다.

54.5. 스마트 카드 관리 및 사용을 위한 도구 설치

스마트 카드를 구성하려면 인증서를 생성하고 스마트 카드에 저장할 수 있는 도구가 필요합니다.

다음은 수행해야 합니다.

- 인증서를 관리하는 데 도움이 되는 **gnutls-utils** 패키지를 설치합니다.

- 스마트 카드로 작업할 라이브러리 및 유틸리티 집합을 제공하는 **opensc** 패키지를 설치합니다.
- 스마트 카드 리더와 통신하는 **pcscd** 서비스를 시작합니다.

절차

1. **opensc** 및 **gnutls-utils** 패키지를 설치합니다.

```
# dnf -y install opensc gnutls-utils
```

2. **pcscd** 서비스를 시작합니다.

```
# systemctl start pcscd
```

pcscd 서비스가 실행 중인지 확인합니다.

54.6. 스마트 카드에 인증서 저장

이 섹션에서는 다음을 구성하는 데 도움이 되는 **pkcs15-init** 도구를 사용한 스마트 카드 구성에 대해 설명합니다.

- 스마트 카드 삭제
- 새로운 **PIN** 및 선택적 **PIN** 잠금 해제 키 (**PUK**) 설정
- 스마트 카드에서 새 슬롯 생성
- 인증서, 개인 키 및 공개 키 저장
- 스마트 카드 설정을 잠그십시오 (일부 스마트 카드에는 이러한 유형의 최종이 필요합니다)

사전 요구 사항

- **pkcs15-init** 도구를 포함하는 **opensc** 패키지가 설치됩니다.
- 자세한 내용은 [스마트 카드 관리 및 사용을 위한 툴](#) 설치를 참조하십시오.
- 카드가 리더에 삽입되고 컴퓨터에 연결됩니다.
 - 스마트 카드에 저장할 개인 키, 공개 키 및 인증서가 있습니다. 이 절차에서 **testuser.key**, **testuserpublic.key** 및 **testuser.crt** 는 개인 키, 공개 키 및 인증서에 사용되는 이름입니다.
 - 현재 스마트 카드 사용자 **PIN** 및 보안 책임자 **PIN(SO-PIN)**

절차

1. 스마트 카드를 지우고 **PIN**으로 인증하십시오:

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

카드가 지워졌습니다.

2. 스마트 카드를 초기화하고 사용자 **PIN** 및 **PUK**, 보안 책임자 **PIN** 및 **PUK**를 설정합니다.

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
  --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

pkcs15-init 툴은 스마트 카드에 새 슬롯을 생성합니다.

3. 슬롯의 레이블 및 인증 **ID**를 설정합니다.

```
$ pkcs15-init --store-pin --label testuser \
  --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

레이블은 사람이 읽을 수 있는 값(이 경우 **testuser**)으로 설정됩니다. **auth-id** 는 16진수 2개 값이어야 합니다(이 경우 **01** 로 설정됨).

4.

스마트 카드의 새 슬롯에 개인 키 저장 및 레이블을 지정합니다.

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \
  --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



참고

id에 대해 지정한 값은 개인 키와 인증서를 저장할 때 동일해야 합니다. **id**에 대한 값을 지정하지 않으면 틀로 더 복잡한 값을 계산하므로 고유한 값을 쉽게 정의할 수 있습니다.

5.

스마트 카드의 새 슬롯에 인증서를 저장하고 레이블을 지정합니다.

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \
  --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6.

(선택 사항) 스마트 카드의 새 슬롯에 공개 키 저장 및 레이블을 지정합니다.

```
$ pkcs15-init --store-public-key testuserpublic.key
  --label testuserpublic_key --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



참고

공개 키가 개인 키 및/또는 인증서에 해당하는 경우 해당 개인 키 및/또는 인증서와 동일한 **ID**를 지정해야 합니다.

7.

(선택 사항) 일부 스마트 카드는 설정을 잠금하여 카드를 종료해야 합니다.

\$ pkcs15-init -F

이 단계에서는 새로 생성된 슬롯에 인증서, 개인 키 및 공개 키가 포함되어 있습니다. 사용자 **PIN** 및 **PUK**, 보안 책임자 **PIN** 및 **PUK**도 생성했습니다.

54.7. 스마트 카드로 IDM에 로그인

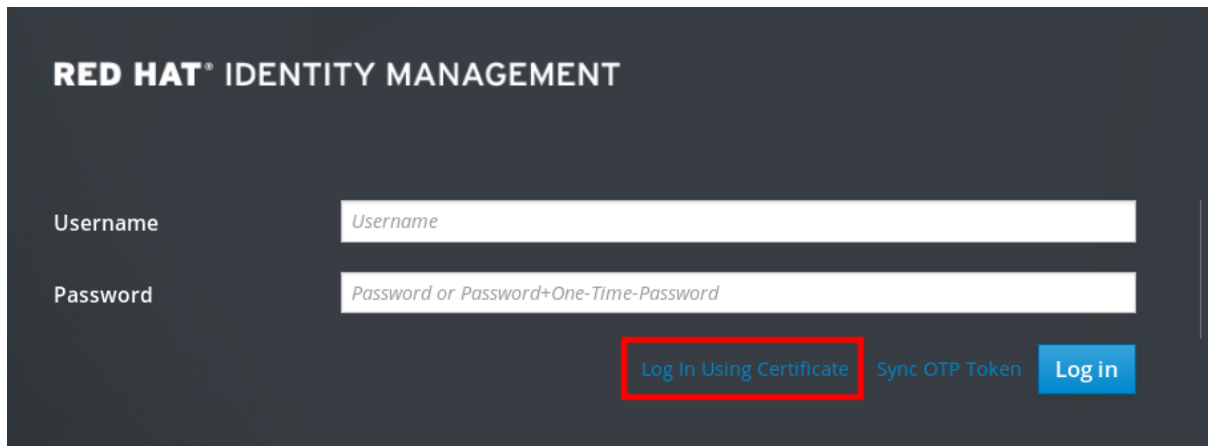
이 섹션에서는 **IdM** 웹 **UI**에 로그인하기 위해 스마트 카드 사용에 대한 정보를 제공합니다.

사전 요구 사항

- 웹 브라우저는 스마트 카드 인증을 사용하도록 구성됩니다.
- **IdM** 서버는 스마트 카드 인증을 위해 구성되었습니다.
- 스마트 카드에 설치된 인증서는 **IdM** 서버로 알려져 있습니다.
- 스마트 카드 잠금을 해제하려면 **PIN**이 필요합니다.
- 스마트 카드가 리더에 연결되었습니다.

절차

1. 브라우저에서 **IdM** 웹 **UI**를 엽니다.
2. 인증서를 사용하여 로그인을 클릭합니다.



RED HAT® IDENTITY MANAGEMENT

Username

Password

Log In Using Certificate Sync OTP Token Log in

3.

Password Required(암호 필요) 대화 상자가 열리면 **PIN**을 추가하여 스마트 카드 잠금을 해제하고 **OK** (확인) 버튼을 클릭합니다.

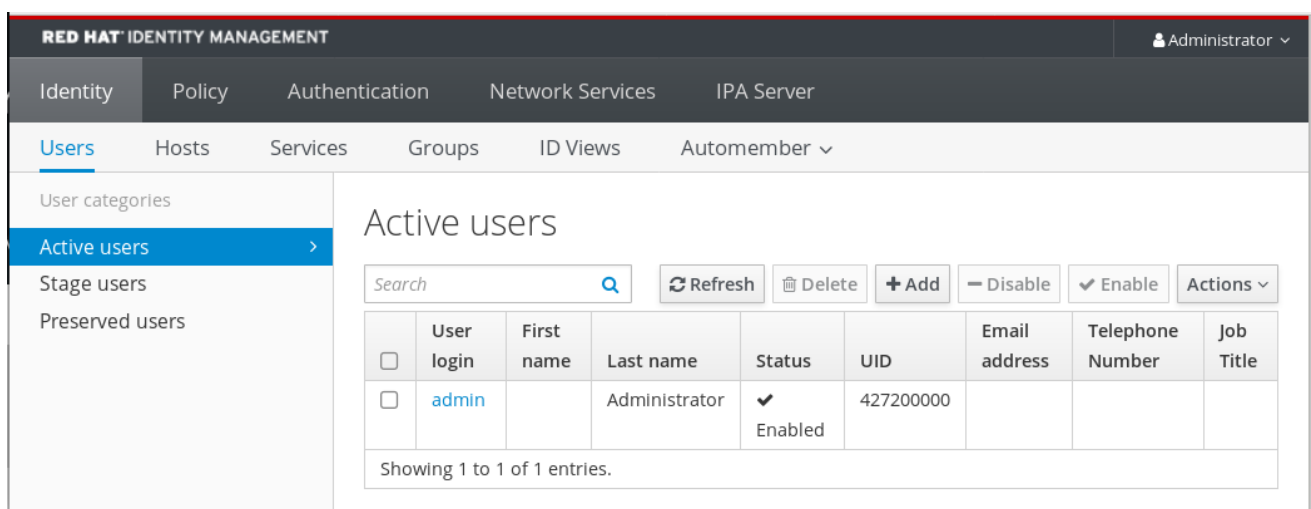
사용자 식별 요청 대화 상자가 열립니다.

스마트 카드에 인증서가 두 개 이상 포함된 경우 아래의 드롭다운 목록에서 인증에 사용할 인증서를 선택하십시오. 식별로 표시할 인증서를 선택하십시오.

4.

OK(확인) 버튼을 클릭합니다.

이제 **IdM** 웹 **UI**에 성공적으로 로그인되었습니다.



RED HAT® IDENTITY MANAGEMENT Administrator

Identity Policy Authentication Network Services IPA Server

Users Hosts Services Groups ID Views Automember

User categories

Active users

Stage users

Preserved users

Active users

Search Refresh Delete Add Disable Enable Actions

	User login	First name	Last name	Status	UID	Email address	Telephone Number	Job Title
☐	admin		Administrator	✓ Enabled	427200000			

Showing 1 to 1 of 1 entries.

54.8. 스마트 카드 인증을 사용하여 GDM 액세스 구성

GDM(Gnome Desktop Manager)에는 인증이 필요합니다. 그러나 암호를 사용할 수도 있지만 인증에 스마트 카드를 사용할 수도 있습니다.

이 섹션에서는 **GDM**에 액세스하기 위한 스마트 카드 인증에 대해 설명합니다.

스마트 카드 인증 사용의 이점은 사용자 계정이 **Identity Management** 도메인의 일부인 경우, **TGT**(티켓 부여 티켓)도 얻을 수 있다는 것입니다.

사전 요구 사항

- 스마트 카드에는 인증서와 개인 키가 포함되어 있습니다.
- 사용자 계정은 **IdM** 도메인의 멤버입니다.
- 스마트 카드의 인증서는 다음을 통해 사용자 항목에 매핑됩니다.
 - 특정 사용자 항목에 인증서 할당. 자세한 내용은 [IdM 웹 UI의 사용자 항목에 인증서 추가 또는 IdM CLI의 사용자 항목에 인증서 추가](#)를 참조하십시오.
 - 계정에 적용되는 인증서 매핑 데이터입니다. 자세한 내용은 [스마트 카드의 인증 구성을 위한 인증서 매핑 규칙](#)을 참조하십시오.

절차

1. 리더에 스마트 카드를 삽입합니다.
2. 스마트 카드 **PIN**을 입력합니다.
3. **Sign In** (로그인)을 클릭합니다.

RHEL 시스템에 성공적으로 로그인했으며 IdM 서버에서 TGT를 제공합니다.

검증 단계

- 터미널 창에서 **klist** 를 입력하고 결과를 확인합니다.

```
$ klist
Ticket cache: KEYRING:persistent:1358900015:krb_cache_TObtNMd
Default principal: example.user@REDHAT.COM

Valid starting    Expires          Service principal
04/20/2020 13:58:24 04/20/2020 23:58:24  krbtgt/EXAMPLE.COM@EXAMPLE.COM
renew until 04/27/2020 08:58:15
```

54.9. 스마트 카드 인증을 사용하여 SU 액세스 구성

다른 사용자로 변경하려면 인증이 필요합니다. 암호 또는 인증서를 사용할 수 있습니다. 이 섹션에서는 **su** 명령으로 스마트 카드 사용에 대해 설명합니다. **su** 명령을 입력한 후 스마트 카드 **PIN**을 묻는 메시지가 표시됩니다.

사전 요구 사항

- 스마트 카드에는 인증서와 개인 키가 포함되어 있습니다.
- 카드가 리더에 삽입되고 컴퓨터에 연결됩니다.

절차

- 터미널 창에서 **su** 명령을 사용하여 다른 사용자로 변경합니다.

```
$ su - example.user
PIN for smart_card
```

구성이 성공적으로 수행되면 스마트 카드 **PIN**을 입력하라는 메시지가 표시됩니다.

55장. IDM의 스마트 카드 인증을 위해 ADCS에서 발급한 인증서 구성

이 시나리오에서는 다음 상황을 설명합니다.

- 배포는 **IdM(Identity Management)**과 **AD(Active Directory)** 간의 교차 포리스트 신뢰성을 기반으로 합니다.
- 계정이 **AD**에 저장된 사용자에게 대해 스마트 카드 인증을 허용하려고 합니다.
- 인증서는 **ADCS(Active Directory Certificate Services)**에 생성 및 저장됩니다.

구성은 다음 단계에서 수행됩니다.

- **Active Directory**에서 **IdM** 서버 및 클라이언트로 **CA** 및 사용자 인증서 복사
- **ADCS** 인증서를 사용하여 스마트 카드 인증을 위해 **IdM** 서버 및 클라이언트 구성
- 인증서 및 개인 키를 스마트 카드로 저장할 수 있도록 **PKCS#12** 파일을 변환
- **sssd.conf** 파일에서 시간 제한 구성
- 스마트 카드 인증을 위한 인증서 매핑 규칙 생성

사전 요구 사항

- **IdM(Identity Management)** 및 **AD(Active Directory)** 신뢰가 설치되어 있습니다.
자세한 내용은 **IdM**과 **AD** 간의 신뢰 설치를 참조하십시오.
- **ADCS(Active Directory Certificate Services)**가 설치되어 사용자를 위한 인증서가 생성됩니다.

55.1. 스마트 카드 인증

스마트 카드는 카드에 저장된 인증서를 사용하여 개인 인증을 제공할 수 있는 물리적 장치입니다. 개인 인증은 사용자 암호와 동일한 방식으로 스마트 카드를 사용할 수 있음을 의미합니다.

개인 키와 인증서의 형태로 스마트 카드에 사용자 자격 증명을 저장할 수 있으며 특수 소프트웨어 및 하드웨어를 사용하여 액세스 할 수 있습니다. 스마트 카드를 리더나 **USB** 소켓에 배치하고 암호를 제공하는 대신 스마트 카드의 **PIN** 코드를 제공합니다.

특정 **IdM** 클라이언트에서 스마트 카드 인증이 작동하는 방법을 구성할 수 있습니다.

- 사용자는 사용자 이름 및 암호로 또는 스마트 카드로 인증할 수 있습니다.
- 사용자가 스마트 카드로 인증할 수 있으며, 비밀번호는 허용되지 않습니다.
- 사용자는 제거 시 기능 잠금으로 로그아웃을 위해 스마트 카드를 사용할 수 있으며 암호는 허용되지 않습니다.

IdM(Identity Management)은 다음을 사용하여 스마트 카드 인증을 지원합니다.

- **IdM** 인증 기관에서 발급한 사용자 인증서. 자세한 내용은 [스마트 카드 인증을 위한 ID 관리](#) 구성을 참조하십시오.
- **ADCS** 인증 기관에서 발급한 사용자 인증서입니다. 자세한 내용은 [IdM에서 스마트 카드 인증을 위해 ADCS에서 발급한 인증서](#) 구성을 참조하십시오.
- **RHEL** 시스템에서 생성된 로컬 인증 기관에서 발급한 사용자 인증서. 자세한 내용은 [로컬 인증서 구성 및 스마트 카드로 가져오기](#)를 참조하십시오.
- 외부 인증 기관에서 발급한 사용자 인증서.



참고

스마트 카드 인증 사용을 시작하려면 하드웨어 요구 사항을 참조하십시오. [RHEL8의 스마트 카드 지원](#).

55.2. 신뢰 구성 및 인증서 사용에 필요한 WINDOWS SERVER 설정

이 섹션에는 **Windows Server**에서 구성해야 하는 사항이 요약되어 있습니다.

- **ADCS(Active Directory Certificate Services)**가 설치되어 있습니다.
- 인증 기관이 생성됨
- [선택 사항] 인증 기관 웹 등록을 사용하는 경우 **IIS(Internet Information Services)**를 구성해야 합니다.

인증서를 내보냅니다.

- 키에는 **2048 비트** 이상이 있어야 합니다
- 개인 키 포함
- 다음 형식의 인증서가 필요합니다. 개인 정보 교환기 - **PKCS #12(.PFX)**
 - 인증서 개인 정보 보호 활성화

55.3. SFTP를 사용하여 ACTIVE DIRECTORY에서 인증서 복사

스마트 카드 인증 기능을 사용하려면 다음 인증서 파일을 복사해야 합니다.

- **CER** 형식의 루트 **CA** 인증서: **IdM** 서버의 **adcs-winserver-ca.cer**입니다.

•

PFX 형식의 개인 키가 있는 사용자 인증서: IdM 클라이언트의 `aduser1.pfx`.



참고

이 절차에서는 **SSH** 액세스가 허용되어야 합니다. **SSH**를 사용할 수 없는 경우 사용자는 **AD** 서버에서 **IdM** 서버 및 클라이언트에 파일을 복사해야 합니다.

절차

1.

IdM 서버에서 연결하고 `adcs-winservice-ca.cer` 루트 인증서를 IdM 서버에 복사합니다.

```
root@idmservice ~]# sftp Administrator@winservice.ad.example.com
Administrator@winservice.ad.example.com's password:
Connected to Administrator@winservice.ad.example.com.
sftp> cd <Path to certificates>
sftp> ls
adcs-winservice-ca.cer  aduser1.pfx
sftp>
sftp> get adcs-winservice-ca.cer
Fetching <Path to certificates>/adcs-winservice-ca.cer to adcs-winservice-ca.cer
<Path to certificates>/adcs-winservice-ca.cer      100% 1254  15KB/s 00:00
sftp quit
```

2.

IdM 클라이언트에서 연결하고 `aduser1.pfx` 사용자 인증서를 클라이언트에 복사합니다.

```
[root@client1 ~]# sftp Administrator@winservice.ad.example.com
Administrator@winservice.ad.example.com's password:
Connected to Administrator@winservice.ad.example.com.
sftp> cd /<Path to certificates>
sftp> get aduser1.pfx
Fetching <Path to certificates>/aduser1.pfx to aduser1.pfx
<Path to certificates>/aduser1.pfx      100% 1254  15KB/s 00:00
sftp quit
```

이제 **CA** 인증서가 **IdM** 서버에 저장되고 사용자 인증서가 클라이언트 시스템에 저장됩니다.

55.4. ADCS 인증서를 사용하여 스마트 카드 인증을 위해 IdM 서버 및 클라이언트 구성

IdM 환경에서 스마트 카드 인증을 사용할 수 있도록 IdM(ID 관리) 서버 및 클라이언트를 구성해야 합니다. IdM에는 필요한 모든 변경을 수행하는 `ipa-adviser` 스크립트가 포함되어 있습니다.

- 필요한 패키지 설치
- **IdM 서버 및 클라이언트 구성**
- **CA 인증서를 예상 위치에 복사**

IdM 서버에서 ipa-adviser 를 실행할 수 있습니다.

이 절차에서는 다음을 설명합니다.

- **IdM 서버에서 다음을 수행합니다. 스마트 카드 인증을 위해 IdM 서버를 구성하기 위해 ipa-adviser 스크립트 준비.**
- **IdM 서버에서 다음을 수행합니다. 스마트 카드 인증을 위해 IdM 클라이언트를 구성하기 위해 ipa-adviser 스크립트를 준비합니다.**
- **IdM 서버에서 다음을 수행합니다. AD 인증서를 사용하여 IdM 서버에 ipa-adviser 서버 스크립트를 적용합니다.**
- **클라이언트 스크립트를 IdM 클라이언트 시스템으로 이동.**
- **IdM 클라이언트에서 다음을 수행합니다. AD 인증서를 사용하여 IdM 클라이언트에 ipa-adviser 클라이언트 스크립트를 적용합니다.**

사전 요구 사항

- 인증서가 **IdM 서버**에 복사되었습니다.
- **Kerberos** 티켓을 받습니다.
- 관리 권한이 있는 사용자로 로그인합니다.

절차

1.

IdM 서버에서 ipa-advise 스크립트를 사용하여 클라이언트를 구성합니다.

```
[root@idmserver ~]# ipa-advise config-client-for-smart-card-auth > sc_client.sh
```

2.

IdM 서버에서 ipa-advise 스크립트를 사용하여 서버를 구성합니다.

```
[root@idmserver ~]# ipa-advise config-server-for-smart-card-auth > sc_server.sh
```

3.

IdM 서버에서 스크립트를 실행합니다.

```
[root@idmserver ~]# sh -x sc_server.sh adcs-winservice-ca.cer
```

•

IdM Apache HTTP 서버를 구성합니다.

•

KDC(키 배포 센터)에서 Kerberos(PKINIT)의 초기 인증에 대한 공개 키 암호화를 활성화합니다.

•

스마트 카드 권한 부여 요청을 수락하도록 IdM 웹 UI를 구성합니다.

4.

sc_client.sh 스크립트를 클라이언트 시스템에 복사합니다.

```
[root@idmserver ~]# scp sc_client.sh root@client1.idm.example.com:/root
```

Password:

```
sc_client.sh          100% 2857  1.6MB/s  00:00
```

5.

Windows 인증서를 클라이언트 시스템에 복사합니다.

```
[root@idmserver ~]# scp adcs-winservice-ca.cer root@client1.idm.example.com:/root
```

Password:

```
adcs-winservice-ca.cer      100% 1254  952.0KB/s  00:00
```

6.

클라이언트 시스템에서 클라이언트 스크립트를 실행합니다.

```
[root@idmclient1 ~]# sh -x sc_client.sh adcs-winservice-ca.cer
```

CA 인증서는 **IdM** 서버 및 클라이언트 시스템에서 올바른 형식으로 설치되고 다음 단계는 사용자 인증서를 스마트 카드 자체에 복사하는 것입니다.

55.5. PFX 파일 변환

PFX (PKCS#12) 파일을 스마트 카드에 저장하기 전에 다음을 수행해야 합니다.

- 파일을 **PEM** 형식으로 변환합니다.
- 개인 키와 인증서를 두 개의 다른 파일로 추출합니다

사전 요구 사항

- **PFX** 파일은 **IdM** 클라이언트 시스템에 복사됩니다.

절차

1. **IdM** 클라이언트에서 **PEM** 형식으로 변경합니다.

```
[root@idmclient1 ~]# openssl pkcs12 -in aduser1.pfx -out aduser1_cert_only.pem -
clcerts -nodes
Enter Import Password:
```

2. 키를 별도의 파일에 추출합니다.

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -nocerts -out adduser1.pem >
aduser1.key
```

3. 공용 인증서를 별도의 파일에 추출합니다.

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -clcerts -nokeys -out
aduser1_cert_only.pem > aduser1.crt
```

이제 **aduser1.key** 및 **aduser1.crt** 를 스마트 카드로 저장할 수 있습니다.

55.6. 스마트 카드 관리 및 사용을 위한 도구 설치

스마트 카드를 구성하려면 인증서를 생성하고 스마트 카드에 저장할 수 있는 도구가 필요합니다.

다음은 수행해야 합니다.

- 인증서를 관리하는 데 도움이 되는 **gnutls-utils** 패키지를 설치합니다.
- 스마트 카드로 작업할 라이브러리 및 유틸리티 집합을 제공하는 **opensc** 패키지를 설치합니다.
- 스마트 카드 리더와 통신하는 **pcscd** 서비스를 시작합니다.

절차

1. **opensc** 및 **gnutls-utils** 패키지를 설치합니다.

```
# dnf -y install opensc gnutls-utils
```

2. **pcscd** 서비스를 시작합니다.

```
# systemctl start pcscd
```

pcscd 서비스가 실행 중인지 확인합니다.

55.7. 스마트 카드에 인증서 저장

이 섹션에서는 다음을 구성하는 데 도움이 되는 **pkcs15-init** 도구를 사용한 스마트 카드 구성에 대해 설명합니다.

- 스마트 카드 삭제

- 새로운 **PIN** 및 선택적 **PIN** 잠금 해제 키 (**PUK**) 설정
- 스마트 카드에서 새 슬롯 생성
- 인증서, 개인 키 및 공개 키 저장
- 스마트 카드 설정을 잠그십시오 (일부 스마트 카드에는 이러한 유형의 최종이 필요합니다)

사전 요구 사항

- **pkcs15-init** 도구를 포함하는 **opensc** 패키지가 설치됩니다.
- 자세한 내용은 [스마트 카드 관리 및 사용을 위한 툴](#) 설치를 참조하십시오.
- 카드가 리더에 삽입되고 컴퓨터에 연결됩니다.
- 스마트 카드에 저장할 개인 키, 공개 키 및 인증서가 있습니다. 이 절차에서 **testuser.key**, **testuserpublic.key** 및 **testuser.crt** 는 개인 키, 공개 키 및 인증서에 사용되는 이름입니다.
- 현재 스마트 카드 사용자 **PIN** 및 보안 책임자 **PIN(SO-PIN)**

절차

1. 스마트 카드를 지우고 **PIN**으로 인증하십시오:

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

카드가 지워졌습니다.

2.

스마트 카드를 초기화하고 사용자 **PIN** 및 **PUK**, 보안 책임자 **PIN** 및 **PUK**를 설정합니다.

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
  --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

pkcs15-init 톨은 스마트 카드에 새 슬롯을 생성합니다.

3.

슬롯의 레이블 및 인증 **ID**를 설정합니다.

```
$ pkcs15-init --store-pin --label testuser \
  --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

레이블은 사람이 읽을 수 있는 값(이 경우 **testuser**)으로 설정됩니다. **auth-id**는 16진수 2개 값이어야 합니다(이 경우 **01**로 설정됨).

4.

스마트 카드의 새 슬롯에 개인 키 저장 및 레이블을 지정합니다.

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \
  --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



참고

id에 대해 지정한 값은 개인 키와 인증서를 저장할 때 동일해야 합니다. **id**에 대한 값을 지정하지 않으면 톨로 더 복잡한 값을 계산하므로 고유한 값을 쉽게 정의할 수 있습니다.

5.

스마트 카드의 새 슬롯에 인증서를 저장하고 레이블을 지정합니다.

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \
  --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6.

(선택 사항) 스마트 카드의 새 슬롯에 공개 키 저장 및 레이블을 지정합니다.

```
$ pkcs15-init --store-public-key testuserpublic.key
--label testuserpublic_key --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



참고

공개 키가 개인 키 및/또는 인증서에 해당하는 경우 해당 개인 키 및/또는 인증서와 동일한 ID를 지정해야 합니다.

7.

(선택 사항) 일부 스마트 카드는 설정을 잠금하여 카드를 종료해야 합니다.

```
$ pkcs15-init -F
```

이 단계에서는 새로 생성된 슬롯에 인증서, 개인 키 및 공개 키가 포함되어 있습니다. 사용자 PIN 및 PUK, 보안 책임자 PIN 및 PUK도 생성했습니다.

55.8. SSSD.CONF에서 시간 제한 설정

스마트 카드 인증서를 사용한 인증은 SSSD에서 사용하는 기본 시간 초과보다 오래 걸릴 수 있습니다. 시간 초과 만료는 다음과 같이 발생할 수 있습니다.

- 느린 리더
- 전달은 물리적 장치를 가상 환경으로 구성합니다.
- 스마트 카드에 너무 많은 인증서가 저장되어 있습니다
- OCSP를 사용하여 인증서를 확인하는 경우 OCSP (Online Certificate Status Protocol) 응답기에서 느린 응답

이 경우 `sssd.conf` 파일에서 다음 시간 제한을 초과할 수 있습니다(예: 60초).

- `p11_child_timeout`

- **`krb5_auth_timeout`**

사전 요구 사항

- **`root`로 로그인해야 합니다.**

절차

1. **`sssd.conf` 파일을 엽니다.**

```
[root@idmclient1 ~]# vim /etc/sss/sss.conf
```

2. **`p11_child_timeout` 값을 변경합니다.**

```
[pam]
p11_child_timeout = 60
```

3. **`krb5_auth_timeout` 값을 변경합니다.**

```
[domain/IDM.EXAMPLE.COM]
krb5_auth_timeout = 60
```

4. **설정을 저장합니다.**

이제 시간 초과로 인증에 실패하기 전에 스마트 카드와의 상호 작용을 1분(60초) 동안 실행할 수 있습니다.

55.9. 스마트 카드 인증을 위한 인증서 매핑 규칙 생성

AD(Active Directory) 및 **IdM(Identity Management)**에 계정이 있는 사용자에게 대해 하나의 인증서를 사용하려는 경우 **IdM** 서버에서 인증서 매핑 규칙을 생성할 수 있습니다.

이러한 규칙을 만든 후 사용자는 두 도메인에서 스마트 카드로 인증할 수 있습니다.

인증서 매핑 규칙에 대한 자세한 내용은 [스마트 카드의 인증 구성을 위한 인증서 매핑 규칙을 참조하십시오](#)

시오.

56장. ID 관리에서 인증서 매핑 규칙 구성

56.1. 스마트 카드 인증을 구성하기 위한 인증서 매핑 규칙

인증서 매핑 규칙은 사용자가 **IdM(Identity Management)** 관리자가 특정 사용자 인증서에 액세스할 수 없는 시나리오에서 인증서를 사용하여 인증할 수 있는 편리한 방법입니다. 이러한 액세스 부족은 일반적으로 외부 인증 기관에서 인증서를 발급했기 때문에 발생합니다. 특별한 사용 사례는 **IdM** 도메인이 신뢰 관계에 있는 **AD(인증 기관)**에서 발급한 인증서로 표시됩니다.

또한 인증서 매핑 규칙은 **IdM** 환경이 스마트 카드를 사용하는 많은 사용자와 함께 사용하는 경우에도 편리합니다. 이 경우 전체 인증서를 추가하는 것이 복잡할 수 있습니다. 주체와 발급자는 대부분의 시나리오에서 예측 가능하므로 전체 인증서보다 미리 추가하기가 더 쉽습니다. 시스템 관리자는 인증서 매핑 규칙을 생성하고 인증서를 특정 사용자에게 발급하기 전에도 사용자 항목에 인증서 매핑 데이터를 추가할 수 있습니다. 인증서가 발급되면 전체 인증서가 사용자 항목에 아직 업로드되지 않았더라도 사용자는 인증서를 사용하여 로그인할 수 있습니다.

또한 인증서를 정기적으로 갱신해야 하므로 인증서 매핑 규칙은 관리 오버헤드를 줄입니다. 사용자의 인증서가 갱신되면 관리자가 사용자 항목을 업데이트할 필요가 없습니다. 예를 들어 매핑이 주체 및 발급자 값을 기반으로 하고 새 인증서에 이전 인증서와 동일한 제목과 발급자가 있는 경우 매핑이 계속 적용됩니다. 반대로 전체 인증서가 사용된 경우 관리자는 새 인증서를 사용자 항목에 업로드하여 이전 인증서를 교체해야 합니다.

인증서 매핑을 설정하려면 다음을 수행합니다.

1. 관리자는 인증서 매핑 데이터(일반적으로 발급자 및 주체) 또는 전체 인증서를 사용자 계정으로 로드해야 합니다.
2. 관리자가 사용자가 **IdM**에 성공적으로 로그인할 수 있도록 인증서 매핑 규칙을 생성해야 합니다.
 - a. 해당 계정에 인증서 매핑 데이터 항목이 포함되어 있습니다.
 - b. 인증서 매핑 데이터 항목이 인증서의 정보와 일치하는 경우

매핑 규칙을 구성하는 개별 구성 요소에 대한 자세한 내용은 **IdM에서 ID 매핑 규칙의 구성 요소**를 참조하십시오. 일치하는 규칙에 사용하기 위해 인증서에서 발급자 가져오기를 참조하십시오.

그런 다음 최종 사용자가 인증서를 표시하여 **파일 시스템** 또는 **스마트 카드**에 저장하면 인증이 성공적으로 수행됩니다.

56.1.1. Active Directory 도메인을 사용하여 신뢰에 대한 인증서 매핑 규칙

이 섹션에서는 **IdM** 배포가 **AD(Active Directory)** 도메인과 신뢰 관계에 있는 경우 가능한 다양한 인증서 매핑 사용 사례에 대해 간단히 설명합니다.

인증서 매핑 규칙은 신뢰할 수 있는 **AD** 인증서 시스템이 발급한 스마트 카드 인증서를 보유한 사용자에게 **IdM** 리소스에 액세스할 수 있는 편리한 방법입니다. **AD** 구성에 따라 다음 시나리오가 가능합니다.

- **AD**에서 인증서를 발급하지만 사용자 및 인증서가 **IdM**에 저장된 경우 매핑 및 인증 요청의 전체 처리가 **IdM** 측에서 수행됩니다. 이 시나리오 구성에 대한 자세한 내용은 **IdM에 저장된 사용자에 대한 인증서 매핑** 구성을 참조하십시오.
- 사용자가 **AD**에 저장된 경우 인증 요청 처리가 **AD**에서 수행됩니다. 다음과 같은 세 가지 하위 사례가 있습니다.
 - **AD** 사용자 항목에는 전체 인증서가 포함되어 있습니다. 이 시나리오에서 **IdM**을 구성하는 방법에 대한 자세한 내용은 **AD 사용자 항목이 전체 인증서가 포함된 사용자에게 대한 인증서 매핑** 구성을 참조하십시오.
 - 사용자 인증서를 사용자 계정에 매핑하도록 **AD**가 구성됩니다. 이 경우 **AD** 사용자 항목에 전체 인증서가 포함되지 않지만 **altSecurityIdentities** 라는 속성을 포함합니다. 이 시나리오에서 **IdM**을 구성하는 방법에 대한 자세한 내용은 **AD가 사용자 계정에 매핑되도록 구성된 경우 인증서 매핑** 구성을 참조하십시오.
 - **AD** 사용자 항목에 전체 인증서와 매핑 데이터가 포함되지 않습니다. 이 경우 유일한 해결책은 **ipa idoverrideuser-add** 명령을 사용하여 **IdM**의 **AD** 사용자 ID 재정의에 전체 인증서를 추가하는 것입니다. 자세한 내용은 **AD 사용자 항목에 인증서 또는 매핑 데이터가 포함되어 있지 않은 경우 인증서 매핑** 구성을 참조하십시오.

56.1.2. IdM에서 ID 매핑 규칙의 구성 요소

이 섹션에서는 **IdM**의 ID 매핑 규칙 구성 요소와 이를 구성하는 방법에 대해 설명합니다. 각 구성 요소에는 재정의할 수 있는 기본값이 있습니다. 웹 UI 또는 **CLI**에서 구성 요소를 정의할 수 있습니다. **CLI**에서 ID 매핑 규칙은 **ipa certmaprule-add** 명령을 사용하여 생성됩니다.

매핑 규칙

매핑 규칙 구성 요소는 인증서를 하나 이상의 사용자 계정과 연결(또는 맵)합니다. 규칙은 인증서를 원하는 사용자 계정과 연결하는 **LDAP** 검색 필터를 정의합니다.

다른 **CA**(인증 기관)에서 발급한 인증서는 다른 속성을 가질 수 있으며 다른 도메인에서 사용할 수 있습니다. 따라서 **IdM**은 무조건 매핑 규칙을 적용하지 않고 적절한 인증서에만 적용합니다. 일치하는 규칙을 사용하여 적절한 인증서를 정의합니다.

매핑 규칙 옵션을 비워 두면 **userCertificate** 특성에서 인증서가 **DER** 인코딩 바이너리 파일로 검색됩니다.

map rule 옵션을 사용하여 **CLI**에서 매핑 규칙을 정의합니다.

일치 규칙

일치하는 규칙 구성 요소는 매핑 규칙을 적용할 인증서를 선택합니다. 기본 일치 규칙은 **digitalSignature** 키 사용 및 **clientAuth** 확장 키 사용과 인증서와 일치합니다.

match rule 옵션을 사용하여 **CLI**에 일치하는 규칙을 정의합니다.

도메인 목록

domain 목록은 **IdM**에서 **ID** 매핑 규칙을 처리할 때 사용자를 검색하려는 **ID** 도메인을 지정합니다. 옵션을 지정하지 않으면 **IdM** 클라이언트가 속한 로컬 도메인의 사용자만 검색합니다.

domain 옵션을 사용하여 **CLI**에서 도메인을 정의합니다.

우선 순위

여러 규칙을 인증서에 적용할 수 있는 경우 우선 순위가 가장 높은 규칙이 우선합니다. 다른 모든 규칙은 무시됩니다.

- 숫자 값이 낮을수록 **ID** 매핑 규칙의 우선 순위가 높습니다. 예를 들어 우선 순위 1이 있는 규칙은 우선 순위가 2인 규칙보다 우선 순위가 높습니다.
- 규칙의 우선 순위 값이 정의되지 않은 경우 우선 순위가 가장 낮습니다.

우선순위 옵션을 사용하여 CLI에서 매핑 규칙 우선 순위를 정의합니다.

인증서 매핑 규칙 예

CLI를 사용하여 **EXAMPLE.ORG** 조직의 스마트 카드 CA 에서 발급한 인증서에 대한 인증을 허용하는 **simple_rule** 이라는 인증서 매핑 규칙을 정의하기 위해 해당 인증서의 주체 가 IdM의 사용자 계정의 **certmapdata** 항목과 일치하는 경우 다음을 정의합니다.

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=Smart Card
CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})'
```

56.1.3. 일치하는 규칙에 사용하기 위해 인증서에서 발급자 가져오기

다음 절차에서는 인증서 매핑 규칙의 일치 규칙에 복사하여 붙여넣을 수 있도록 인증서에서 발급자 정보를 가져오는 방법을 설명합니다. 일치하는 규칙에 필요한 발급자 형식을 가져오려면 **openssl x509** 유틸리티를 사용합니다.

사전 요구 사항

- 사용자 인증서가 .pem 또는. crt 형식으로 되어 있습니다

절차

1. 인증서에서 사용자 정보를 가져옵니다. 다음을 사용하여 **openssl x509** 인증서 표시 및 서명 유틸리티를 사용합니다.
 - 인코딩된 버전의 요청을 방지하는 **no out** 옵션
 - 발급자 이름을 출력하는 **-issuer** 옵션
 - 인증서를 읽을 입력 파일 이름을 지정하는 **-in** 옵션
 - 가장 구체적인 상대 고유 이름(RDN)이 있는 출력을 먼저 표시하기 위한 **RFC2253** 값이 있는 **-nameopt** 옵션

입력 파일에 ID 관리 인증서가 포함된 경우 명령의 출력에는 조직 정보를 사용하여 발급

자가 정의됨을 표시합니다.

```
# openssl x509 -noout -issuer -in idm_user.crt -nameopt RFC2253
issuer=CN=Certificate Authority,O=REALM.EXAMPLE.COM
```

입력 파일에 **Active Directory** 인증서가 포함된 경우 명령의 출력에 도메인 구성 요소 정보를 사용하여 발급자가 정의됨을 표시합니다.

```
# openssl x509 -noout -issuer -in ad_user.crt -nameopt RFC2253
issuer=CN=AD-WIN2012R2-CA,DC=AD,DC=EXAMPLE,DC=COM
```

2.

선택적으로 일치하는 규칙을 기반으로 CLI에서 새 매핑 규칙을 생성하려면 인증서 발급자가 **ad.example.com** 도메인의 추출된 **AD-WIN2012R2-CA** 여야 하며 인증서의 제목은 **IdM**의 사용자 계정의 **certmapdata** 항목과 일치해야 합니다.

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=AD-WIN2012R2-CA,DC=AD,DC=EXAMPLE,DC=COM' --maprule '(ipacertmapdata=X509:<l>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})'
```

56.1.4. 추가 리소스

- **sss-certmap(5)** 도움말 페이지를 참조하십시오.

56.2. IDM에 저장된 사용자에게 대한 인증서 매핑 구성

이 사용자 사례에서는 인증서 인증을 구성하는 사용자가 **IdM**에 저장된 경우 시스템 관리자가 **IdM**에서 인증서 매핑을 활성화하기 위해 수행해야 하는 단계에 대해 설명합니다.

사전 요구 사항

- 사용자 계정은 **IdM**에 있습니다.
- 관리자에게는 사용자 항목에 추가할 전체 인증서 또는 인증서 매핑 데이터가 있습니다.

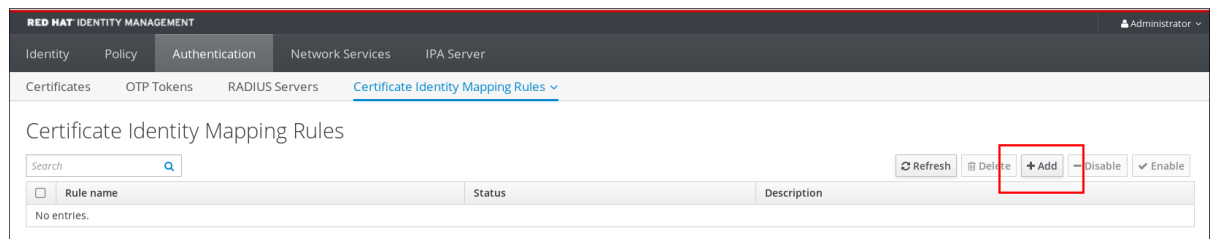
56.2.1. IdM에서 인증서 매핑 규칙 추가

이 섹션에서는 매핑 규칙에 지정된 조건과 일치하는 인증서가 있는 **IdM** 사용자가 인증서 매핑 데이터 항목을 인증할 수 있도록 인증서 매핑 규칙을 설정하는 방법에 대해 설명합니다.

56.2.1.1. IdM 웹 UI에서 인증서 매핑 규칙 추가

1. **IdM 웹 UI에 관리자로 로그인합니다.**
2. **인증 → 인증서 ID 매핑 규칙 → 인증서 ID 매핑 규칙으로 이동합니다.**
3. **추가를 클릭합니다.**

그림 56.1. IdM 웹 UI에서 새 인증서 매핑 규칙 추가



4. **규칙 이름을 입력합니다.**
5. **매핑 규칙을 입력합니다. 예를 들어 IdM이 제공된 인증서의 발급자 및 주체 항목을 검색하고, 제공된 인증서의 이러한 두 항목에 있는 정보에 대해 인증 여부를 결정한 경우 다음을 수행합니다.**

(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})

6. **일치하는 규칙을 입력합니다. 예를 들어 EXAMPLE.ORG 조직의 스마트 카드 CA 에서 발급한 인증서만 IdM에 대한 인증을 허용하려면 다음을 수행합니다.**

<ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG

그림 56.2. IdM 웹 UI에서 인증서 매핑 규칙 세부 정보 입력

7. 대화 상자의 맨 아래에서 **Add** (추가)를 클릭하여 규칙을 추가하고 상자를 닫습니다.

8. **SSSD**(시스템 보안 서비스 데몬)는 정기적으로 인증서 매핑 규칙을 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

이제 스마트 카드 인증서에서 찾은 스마트 카드 인증서와 IdM 사용자 항목의 인증서 매핑 데이터와 매핑 규칙에 지정된 데이터 유형을 비교하는 인증서 매핑 규칙 세트가 있습니다. 일치 항목을 찾으면 일치하는 사용자를 인증합니다.

56.2.1.2. IdM CLI에서 인증서 매핑 규칙 추가

1. 관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2. 매핑 규칙과 매핑 규칙이 기반으로 하는 일치 규칙을 입력합니다. 예를 들어 제공된 인증서에서 발급자 및 주체 항목을 검색하고 **EXAMPLE.ORG** 조직의 스마트 카드 **CA**에서 발급한 인증서만 인식하여 제공된 인증서의 이 두 항목에 있는 정보를 인증 또는 사용하지 않도록 결정하려면 다음을 수행합니다.


```
# ipa certmaprule-add rule_name --matchrule '<ISSUER>CN=Smart Card
CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})'

-----
Added Certificate Identity Mapping Rule "rule_name"
-----

Rule name: rule_name
Mapping rule: (ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})
Matching rule: <ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG
Enabled: TRUE
```

3.

SSSD(시스템 보안 서비스 데몬)는 정기적으로 인증서 매핑 규칙을 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

이제 스마트 카드 인증서에서 찾은 스마트 카드 인증서와 **IdM** 사용자 항목의 인증서 매핑 데이터와 매핑 규칙에 지정된 데이터 유형을 비교하는 인증서 매핑 규칙 세트가 있습니다. 일치 항목을 찾으면 일치하는 사용자를 인증합니다.

56.2.2. IdM의 사용자 항목에 인증서 매핑 데이터 추가

이 섹션에서는 모두 인증서 매핑 데이터 항목에 지정된 값이 포함되어 있는 한 사용자가 여러 인증서를 사용하여 인증할 수 있도록 인증서 매핑 데이터를 **IdM** 사용자 항목에 입력하는 방법을 설명합니다.

56.2.2.1. IdM 웹 UI의 사용자 항목에 인증서 매핑 데이터 추가

1.

IdM 웹 UI에 관리자로 로그인합니다.

2.

사용자 → 활성 사용자 → **idm_user** 로 이동합니다.

3.

인증서 매핑 데이터 옵션을 찾은 후 **Add(추가)**를 클릭합니다.

4.

idm_user 인증서가 있는 경우:

a.

명령줄 인터페이스에서 **cat** 유틸리티 또는 텍스트 편집기를 사용하여 인증서를 표시합니다.

```
[root@server ~]# cat idm_user_certificate.pem
-----BEGIN CERTIFICATE-----
MIIFFTCCA/2gAwIBAgIBejANBgkqhkiG9w0BAQsFADA6MRgwFgYDVQQKDA9JRE
0u
RVhBTVMR5SDT00xHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0
x
ODA5MDIxODE1MzlaFw0yMDA5MDIxODE1MzlaMCwxGDAWBgNVBAoMD0IETS5F
WEFN
[...output truncated...]
```

b.

인증서를 복사합니다.

c.

IdM 웹 UI에서 인증서 옆에 있는 **Add(추가)**를 클릭하고 인증서를 열리는 창에 붙여넣습니다.

그림 56.3. 사용자의 인증서 매핑 데이터 추가: **certificate**

User: demouser
demouser is a member of:

Settings | User Groups | Netgroups | Roles | HBAC Rules | Sudo Rules

Refresh Revert Save Actions

Identity Settings

Job Title

First name * Demo

Last name * User

Full name * Demo User

Display name Demo User

Initials DU

GECOS Demo User

Class

Account Settings

User login demouser

Password *****

Password expiration 2016-07-14 10:14:41Z

UID 373000005

GID 373000005

Principal alias demouser@IDM.EXAMPLE.COM Delete

Add

Kerberos principal expiration YYYY-MM-DD hh : mn UTC

Login shell /bin/sh

Home directory /home/demouser

SSH public keys Add

Certificates Add

또는 **idm_user** 인증서가 없지만 발급자 및 인증서 주체를 알고 있는 경우 발급자 및 제목의 라디오 버튼을 선택하고 두 개의 상자에 값을 입력하십시오.

그림 56.4. 사용자의 인증서 매핑 데이터 추가: 발급자 및 제목

The screenshot shows a dialog box titled "Add Certificate Mapping Data". It has two radio buttons: "Certificate mapping data" (unselected) and "Issuer and subject" (selected). The "Issuer and subject" section is highlighted with a red box. It contains two text input fields: "Issuer" with the value "O=EXAMPLE.ORG,CN=Smart Card CA" and "Subject" with the value "CN=test,O=EXAMPLE.ORG". At the bottom right of the dialog are "Add" and "Cancel" buttons. The background window shows a table with columns "GID" and "Certificate mapping", and a row with the value "1997000009".

5.

추가를 클릭합니다.

6.

선택적으로 .pem 형식의 전체 인증서에 액세스할 수 있는 경우 사용자 및 인증서가 연결되어 있는지 확인합니다.

a.

sss_cache 유틸리티를 사용하여 **SSSD** 캐시의 **idm_user** 레코드를 무효화하고 **idm_user** 정보를 강제로 다시 로드합니다.

```
# sss_cache -u idm_user
```

b.

IdM 사용자의 인증서가 포함된 파일 이름으로 **ipa certmap-match** 명령을 실행합니다.

```
# ipa certmap-match idm_user_cert.pem
```

```
-----
1 user matched
-----
```

```
Domain: IDM.EXAMPLE.COM
```

```
User logins: idm_user
```

```
-----
Number of entries returned 1
-----
```

출력에서 인증서 매핑 데이터가 **idm_user**에 추가되고 해당 매핑 규칙이 있음을 확인합니다. 즉, 정의된 인증서 매핑 데이터와 일치하는 인증서를 사용하여 **idm_user**로 인증할 수 있습니다.

56.2.2.2. IdM CLI의 사용자 항목에 인증서 매핑 데이터 추가

1.

관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2.

`idm_user`의 인증서가 필요할 경우 `ipa user-add-cert` 명령을 사용하여 사용자 계정에 인증서를 추가합니다.

```
# CERT=`cat idm_user_cert.pem | tail -n +2 | head -n -1 | tr -d '\r\n\'`
# ipa user-add-certmapdata idm_user --certificate $CERT
```

또는 `idm_user` 인증서가 없지만 발급자 및 `idm_user` 인증서의 주체를 알고 있는 경우:

```
# ipa user-add-certmapdata idm_user --subject "O=EXAMPLE.ORG,CN=test" --issuer
"CN=Smart Card CA,O=EXAMPLE.ORG"
```

```
-----
Added certificate mappings to user "idm_user"
-----
```

```
User login: idm_user
Certificate mapping data: X509:<I>O=EXAMPLE.ORG,CN=Smart Card
CA<S>CN=test,O=EXAMPLE.ORG
```

3.

선택적으로 `.pem` 형식의 전체 인증서에 액세스할 수 있는 경우 사용자 및 인증서가 연결되어 있는지 확인합니다.

a.

`sss_cache` 유틸리티를 사용하여 `SSSD` 캐시의 `idm_user` 레코드를 무효화하고 `idm_user` 정보를 강제로 다시 로드합니다.

```
# sss_cache -u idm_user
```

b.

IdM 사용자의 인증서가 포함된 파일 이름으로 `ipa certmap-match` 명령을 실행합니다.

```
# ipa certmap-match idm_user_cert.pem
```

```
-----
1 user matched
-----
```

```
Domain: IDM.EXAMPLE.COM
```

User logins: idm_user

Number of entries returned 1

출력에서 인증서 매핑 데이터가 **idm_user**에 추가되고 해당 매핑 규칙이 있음을 확인합니다. 즉, 정의된 인증서 매핑 데이터와 일치하는 인증서를 사용하여 **idm_user**로 인증할 수 있습니다.

56.3. AD 사용자 항목이 전체 인증서가 포함된 사용자에게 대한 인증서 매핑 구성

이 사용자 사례에서는 **IdM** 배포를 **AD(Active Directory)**와 신뢰하는 경우 **IdM**에서 인증서 매핑을 활성화하는 데 필요한 단계를 설명하고, 사용자는 **AD**에 저장되고 **AD**의 사용자 항목에 전체 인증서가 포함되어 있습니다.

사전 요구 사항

- 사용자에게 **IdM**에 계정이 없습니다.
- 사용자는 인증서를 포함하는 **AD**에 계정이 있습니다.
- **IdM** 관리자는 **IdM** 인증서 매핑 규칙을 기반으로 할 수 있는 데이터에 액세스할 수 있습니다.

56.3.1. AD 항목이 전체 인증서가 포함된 사용자를 위한 인증서 매핑 규칙 추가

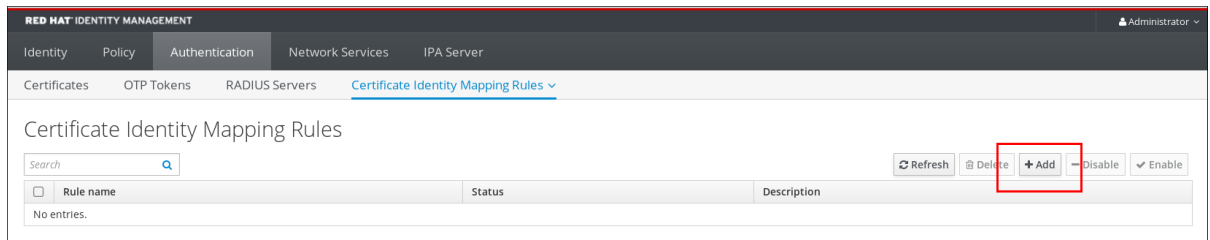
56.3.1.1. IdM 웹 UI에서 인증서 매핑 규칙 추가

1. **IdM 웹 UI**에 관리자로 로그인합니다.
2. 인증 → 인증서 ID 매핑 규칙 → 인증서 ID 매핑 규칙으로 이동합니다.

3.

추가를 클릭합니다.

그림 56.5. IdM 웹 UI에서 새 인증서 매핑 규칙 추가



4.

규칙 이름을 입력합니다.

5.

매핑 규칙을 입력합니다. 인증을 위해 IdM에 제공되는 전체 인증서를 AD에서 사용할 수 있는 인증서와 비교하려면 다음을 수행합니다.

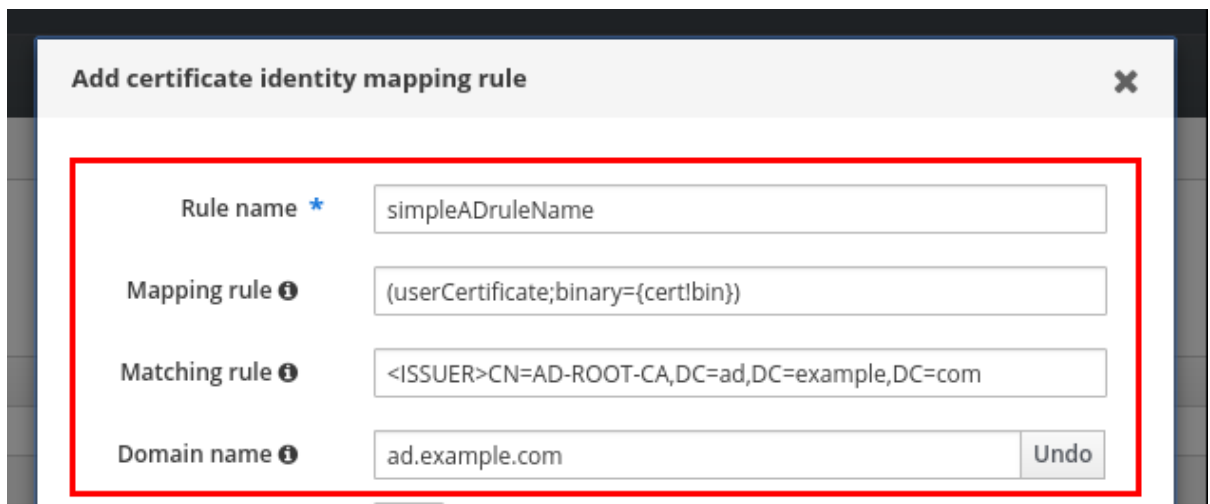
```
(userCertificate;binary={cert!bin})
```

6.

일치하는 규칙을 입력합니다. 예를 들어 **AD.EXAMPLE.COM** 도메인의 **AD-ROOT-CA** 에서 발급한 인증서만 인증하려면 다음을 수행하십시오.

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

그림 56.6. 인증서가 AD에 저장된 사용자의 인증서 매핑 규칙



7.

추가를 클릭합니다.

8.

SSSD(시스템 보안 서비스 데몬)는 정기적으로 인증서 매핑 규칙을 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **CLI**에서 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

56.3.1.2. IdM CLI에서 인증서 매핑 규칙 추가

1.

관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2.

매핑 규칙과 매핑 규칙이 기반으로 하는 일치 규칙을 입력합니다. 인증에 대해 제공되는 전체 인증서를 **AD.EXAMPLE.COM** 도메인의 **AD-ROOT-CA** 가 인증할 수 있도록 허용하는 경우:

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --domain ad.example.com
```

```
-----
Added Certificate Identity Mapping Rule "simpleADrule"
-----
```

```
Rule name: simpleADrule
Mapping rule: (userCertificate;binary={cert!bin})
Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE
```

3.

SSSD(시스템 보안 서비스 데몬)는 정기적으로 인증서 매핑 규칙을 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

56.4. AD가 사용자 계정에 매핑하도록 구성된 경우 인증서 매핑 구성

이 사용자 사례에서는 **IdM** 배포를 **AD(Active Directory)**와 신뢰하는 경우 **IdM**에서 인증서 매핑을 활성화하는 데 필요한 단계를 설명하고, 사용자는 **AD**에 저장되고 **AD**의 사용자 항목에 인증서 매핑 데이터가 포함되어 있습니다.

사전 요구 사항

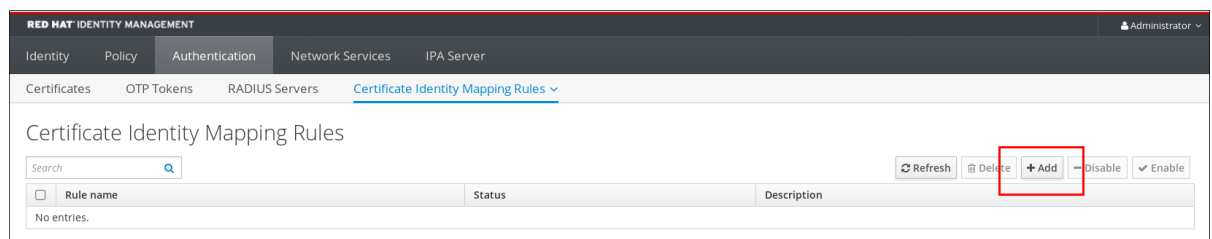
- 사용자에게 **IdM**에 계정이 없습니다.
- 사용자는 **IdM certmapdata** 속성과 동등한 **altSecurityIdentities** 특성을 포함하는 **AD**에 계정이 있습니다.
- **IdM** 관리자는 **IdM** 인증서 매핑 규칙을 기반으로 할 수 있는 데이터에 액세스할 수 있습니다.

56.4.1. 신뢰할 수 있는 **AD** 도메인이 사용자 인증서를 매핑하도록 구성된 경우 인증서 매핑 규칙 추가

56.4.1.1. **IdM** 웹 UI에서 인증서 매핑 규칙 추가

1. **IdM** 웹 UI에 관리자로 로그인합니다.
2. 인증 → 인증서 ID 매핑 규칙 → 인증서 ID 매핑 규칙으로 이동합니다.
3. 추가를 클릭합니다.

그림 56.7. **IdM** 웹 UI에서 새 인증서 매핑 규칙 추가



4. 규칙 이름을 입력합니다.
5. 매핑 규칙을 입력합니다. 예를 들어 **AD DC**가 제공된 인증서의 발급자 및 주체 항목을 검색하고, 제공된 인증서의 이러한 두 항목에 있는 정보에 대해 인증 여부를 결정한 경우 다음을 수행합니다.


```
(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})
```

6.

일치하는 규칙을 입력합니다. 예를 들어 **AD.EXAMPLE.COM** 도메인의 **AD-ROOT-CA** 에서 발급한 인증서만 **IdM**에 인증하도록 허용하려면 다음을 수행합니다.

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

7.

도메인을 입력합니다.

```
ad.example.com
```

그림 56.8. AD가 매핑에 대해 구성된 경우 인증서 매핑 규칙

8.

추가를 클릭합니다.

9.

SSSD(시스템 보안 서비스 데몬)는 정기적으로 인증서 매핑 규칙을 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **CLI**에서 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

56.4.1.2. IdM CLI에서 인증서 매핑 규칙 추가

1.

관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2.

매핑 규칙과 매핑 규칙이 기반으로 하는 일치 규칙을 입력합니다. 예를 들어 제공된 인증서에서 발급자 및 주체 항목을 **AD**로 검색하고 **AD.EXAMPLE.COM** 도메인의 **AD-ROOT-CA**에서 발급한 인증서만 허용하려면 다음을 수행합니다.

```
# ipa certmaprule-add ad_configured_for_mapping_rule --matchrule
'<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule
'(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})' --
domain=ad.example.com

-----
Added Certificate Identity Mapping Rule "ad_configured_for_mapping_rule"
-----

Rule name: ad_configured_for_mapping_rule
Mapping rule: (altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>
{subject_dn!ad_x500})
Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE
```

3.

SSSD(시스템 보안 서비스 데몬)는 정기적으로 인증서 매핑 규칙을 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

56.4.2. AD 측에서 인증서 매핑 데이터 확인

altSecurityIdentities 속성은 **IdM**의 **certmapdata** 사용자 속성에 해당하는 **AD(Active Directory)**입니다. 사용자 인증서를 사용자 계정에 매핑하도록 신뢰할 수 있는 **AD** 도메인이 구성된 경우, **IdM** 시스템 관리자는 **AD**의 사용자 항목에 **altSecurityIdentities** 특성이 올바르게 설정되었는지 확인해야 합니다.

AD에 **AD**에 저장된 사용자에게 대한 올바른 정보가 포함되어 있는지 확인하려면 **ldapsearch** 명령을 사용합니다.

•

예를 들어 아래 명령을 입력하여 다음 조건이 적용되는 **adserver.ad.example.com** 서버를 확인합니다.

○

altSecurityIdentities 속성은 **ad_user**의 사용자 항목에 설정되어 있습니다.

○

matchrule은 다음 조건이 적용되는 것으로 지정합니다.

■

ad_user가 AD 인증에 사용하는 인증서는 **ad.example.com** 도메인의 **AD-ROOT-CA**에 의해 발급되었습니다.

■

제목은 **<S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user**입니다.

```
$ ldapsearch -o ldif-wrap=no -LLL -h adserver.ad.example.com \
-p 389 -D cn=Administrator,cn=users,dc=ad,dc=example,dc=com \
-W -b cn=users,dc=ad,dc=example,dc=com "(cn=ad_user)" \
altSecurityIdentities
Enter LDAP Password:
dn: CN=ad_user,CN=Users,DC=ad,DC=example,DC=com
altSecurityIdentities: X509:<l>DC=com,DC=example,DC=ad,CN=AD-ROOT-
CA<S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user
```

56.5. AD 사용자 항목에 인증서 또는 매핑 데이터가 포함되어 있지 않은 경우 인증서 매핑 구성

이 사용자 사례에서는 **IdM** 배포를 **AD(Active Directory)**와 신뢰하는 경우 **IdM**에서 인증서 매핑을 활성화하는 데 필요한 단계를 설명하고, 사용자는 **AD**에 저장되고 **AD**의 사용자 항목에 전체 인증서나 인증서 매핑 데이터가 포함되지 않습니다.

사전 요구 사항

●

사용자에게 **IdM**에 계정이 없습니다.

●

사용자는 **AD**에 전체 인증서와 **altSecurityIdentities** 속성이 포함되지 않으며 **IdM** **certmapdata** 속성과 동등한 **AD**가 포함되지 않습니다.

●

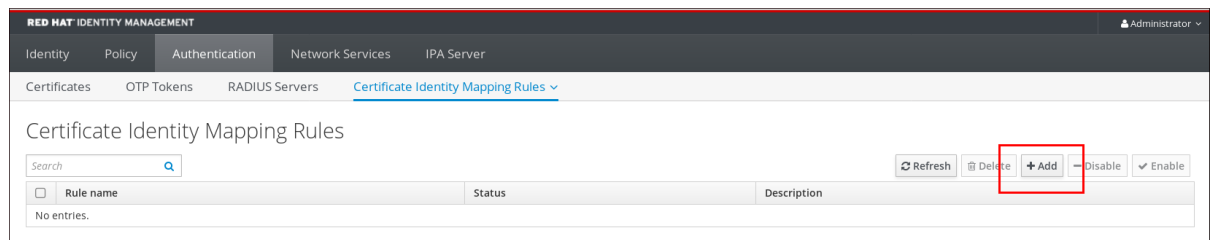
IdM 관리자에게는 **IdM**에서 **AD** 사용자의 사용자 ID 재정의에 추가할 전체 **AD** 사용자 인증서가 있습니다.

56.5.1. AD 사용자 항목에 인증서 또는 매핑 데이터가 없는 경우 인증서 매핑 규칙 추가

56.5.1.1. IdM 웹 UI에서 인증서 매핑 규칙 추가

1. **IdM 웹 UI에 관리자로 로그인합니다.**
2. **인증 → 인증서 ID 매핑 규칙 → 인증서 ID 매핑 규칙으로 이동합니다.**
3. **추가를 클릭합니다.**

그림 56.9. IdM 웹 UI에서 새 인증서 매핑 규칙 추가



4. **규칙 이름을 입력합니다.**
5. **매핑 규칙을 입력합니다. IdM에 있는 AD 사용자 항목의 사용자 ID 재정의 항목에 저장된 인증서와 비교하여 인증을 위해 IdM에 제공되는 전체 인증서를 보유하려면 다음을 수행합니다.**

(userCertificate;binary={cert!bin})

6. **일치하는 규칙을 입력합니다. 예를 들어 AD.EXAMPLE.COM 도메인의 AD-ROOT-CA 에서 발급한 인증서만 인증하려면 다음을 수행하십시오.**

<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com

7. **도메인 이름을 입력합니다. 예를 들어 ad.example.com 도메인에서 사용자를 검색하려면 다음을 수행합니다.**

그림 56.10. 인증서 또는 매핑 데이터가 AD에 저장되지 않은 사용자의 인증서 매핑 규칙

8.

추가를 클릭합니다.

9.

SSSD(시스템 보안 서비스 데몬)는 정기적으로 인증서 매핑 규칙을 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **CLI**에서 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

56.5.1.2. IdM CLI에서 인증서 매핑 규칙 추가

1.

관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2.

매핑 규칙과 매핑 규칙이 기반으로 하는 일치 규칙을 입력합니다. **IdM**의 **AD** 사용자 항목의 사용자 ID 덮어쓰기 항목에 저장된 인증서와 비교하여 인증을 위해 제공되는 전체 인증서를 보유하려면 **AD.EXAMPLE.COM** 도메인의 **AD-ROOT-CA** 에서 발급한 인증서만 인증할 수 있습니다.

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --domain ad.example.com
```

```
-----
Added Certificate Identity Mapping Rule "simpleADrule"
-----
```

```
Rule name: simpleADrule
Mapping rule: (userCertificate;binary={cert!bin})
```

Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE

3.

SSSD(시스템 보안 서비스 데몬)는 정기적으로 인증서 매핑 규칙을 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

56.5.2. AD의 사용자 항목에 인증서 또는 매핑 데이터가 포함되어 있지 않은 경우 AD 사용자 ID에 인증서를 추가합니다.

56.5.2.1. IdM 웹 UI에서 AD 사용자 ID 재정의에 인증서 추가

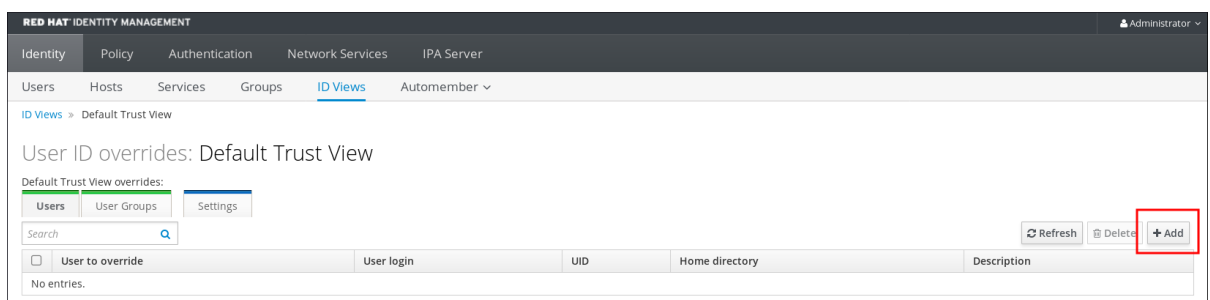
1.

ID → ID 보기 → 기본 신뢰 보기로 이동합니다.

2.

추가를 클릭합니다.

그림 56.11. IdM 웹 UI에서 새 사용자 ID 덮어쓰기 추가



3.

User to override (재정의할 사용자) 필드에 ad_user@ad.example.com 을 입력합니다.

4.

ad_user 인증서를 복사하여 Certificate(인증서) 필드에 붙여넣습니다.

그림 56.12. AD 사용자에게 대한 사용자 ID 덮어쓰기 구성

Add User ID override

User to override *

User login

GECOS

UID

GID

Certificate

5.

추가를 클릭합니다.

6.

선택적으로 사용자 및 인증서가 연결되어 있는지 확인합니다.

a.

sss_cache 유틸리티를 사용하여 **SSSD** 캐시의 **ad_user@ad.example.com** 레코드를 무효화하고 **ad_user@ad.example.com** 정보를 강제로 다시 로드합니다.

```
# sss_cache -u ad_user@ad.example.com
```

b.

AD 사용자의 인증서가 포함된 파일 이름으로 **ipa certmap-match** 명령을 실행합니다.

```
# ipa certmap-match ad_user_cert.pem
```

```
-----
```

```
1 user matched
```

```
-----
```

```
Domain: AD.EXAMPLE.COM
```

```
User logins: ad_user@ad.example.com
```

```
-----
```

```
Number of entries returned 1
```

```
-----
```

출력은 인증서 매핑 데이터가 `ad_user@ad.example.com` 에 추가되었고 **AD 사용자 항목에 인증서 또는 매핑 데이터가 없는 경우 인증서 매핑 규칙 추가에 정의된 해당 매핑 규칙**이 있는지 확인합니다. 즉, 정의된 인증서 매핑 데이터와 일치하는 인증서를 사용하여 `ad_user@ad.example.com` 로 인증할 수 있습니다.

56.5.2.2. IdM CLI에서 AD 사용자 ID 재정의에 인증서 추가

1. 관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2. `ipa idoverrideuser-add-cert` 명령을 사용하여 `ad_user@ad.example.com` 의 인증서를 사용자 계정에 추가합니다.

```
# CERT=`cat ad_user_cert.pem | tail -n +2 | head -n -1 | tr -d '\r\n'`
# ipa idoverrideuser-add-cert ad_user@ad.example.com --certificate $CERT
```

3. 선택적으로 사용자 및 인증서가 연결되어 있는지 확인합니다.

- a. `sss_cache` 유틸리티를 사용하여 `SSSD` 캐시의 `ad_user@ad.example.com` 레코드를 무효화하고 `ad_user@ad.example.com` 정보를 강제로 다시 로드합니다.

```
# sss_cache -u ad_user@ad.example.com
```

- b. **AD 사용자의 인증서가 포함된 파일 이름으로 `ipa certmap-match` 명령을 실행합니다.**

```
# ipa certmap-match ad_user_cert.pem
-----
1 user matched
-----
Domain: AD.EXAMPLE.COM
User logins: ad_user@ad.example.com
-----
Number of entries returned 1
-----
```

출력은 인증서 매핑 데이터가 `ad_user@ad.example.com` 에 추가되었고 **AD 사용자 항목에 인증서 또는 매핑 데이터가 없는 경우 인증서 매핑 규칙 추가에 정의된 해당 매핑 규칙**

이 있는지 확인합니다. 즉, 정의된 인증서 매핑 데이터와 일치하는 인증서를 사용하여 `ad_user@ad.example.com` 로 인증할 수 있습니다.

56.6. 여러 ID 매핑 규칙을 하나로 결합

여러 ID 매핑 규칙을 하나의 결합 규칙으로 결합하려면 `|` (또는) 문자를 사용하여 개별 매핑 규칙 앞에 두고 `()` 대괄호를 사용하여 구분합니다. 예를 들면 다음과 같습니다.

인증서 매핑 필터 예 1

```
$ ipa certmaprule-add ad_cert_for_ipa_and_ad_users \ --maprule='(|(ipacertmapdata=X509:<l>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})(altSecurityIdentities=X509:<l>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}))' \ --matchrule='<ISSUER>CN=AD-ROOT-
CA,DC=ad,DC=example,DC=com' \ --domain=ad.example.com
```

위 예제에서 `--maprule` 옵션의 필터 정의에는 다음 기준이 포함됩니다.

- `ipacertmapdata=X509:<l>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}` 는 IdM의 인증서 추가에 설명된 대로 스마트 카드 인증서에서 IdM 사용자 계정의 `ipacertmapdata` 속성 값에 주체와 발급자를 연결하는 필터입니다. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_and_managing_identity_management/configuring_and_managing-idm#proc-add-maprule_conf-certmap-for-users-in-idm
- `altSecurityIdentities=X509:<l>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}` 은 스마트 카드 인증서에서 AD 사용자 계정의 `altSecurityIdentities` 속성 값에 주체와 발급자를 연결하는 필터입니다. 신뢰할 수 있는 AD 도메인이 사용자 인증서를 매핑하도록 구성된 경우 인증서 매핑 규칙 추가에 설명된 대로
- `domain=ad.example.com` 옵션을 추가하면 지정된 인증서에 매핑된 사용자가 로컬 `idm.example.com` 도메인뿐만 아니라 `ad.example.com` 도메인에서도 검색됩니다.

`map rule` 옵션의 필터 정의는 논리 연산자 `|` (또는)를 허용하므로 여러 기준을 지정할 수 있습니다. 이

경우 규칙은 기준 중 하나 이상을 충족하는 모든 사용자 계정을 매핑합니다.

인증서 매핑 필터 예 2

```
$ ipa certmaprule-add ipa_cert_for_ad_users \
--maprule='((userCertificate;binary={cert!bin})(ipacertmapdata=X509:<I>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})(altSecurityIdentities=X509:<I>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}))' \
--matchrule='<ISSUER>CN=Certificate Authority,O=REALM.EXAMPLE.COM' \
--domain=idm.example.com --domain=ad.example.com
```

위 예제에서 `--maprule` 옵션의 필터 정의에는 다음 기준이 포함됩니다.

- `userCertificate;binary={cert!bin}` 은 전체 인증서를 포함하는 사용자 항목을 반환하는 필터입니다. AD 사용자의 경우 이 유형의 필터를 만드는 것은 **AD 사용자 항목에 인증서 또는 매핑 데이터가 포함되어 있지 않은 경우 인증서 매핑 규칙 추가**에 자세히 설명되어 있습니다.
- `ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}` 은 IdM에 인증서 매핑 규칙 추가에 설명된 대로 스마트 카드 인증서에서 스마트 카드 인증서의 값과 발급자를 IdM 사용자 계정의 `ipacertmapdata` 속성 값으로 연결하는 필터입니다.
- `altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}` 은 스마트 카드 인증서에서 AD 사용자 계정의 `altSecurityIdentities` 속성 값에 주체와 발급자를 연결하는 필터입니다. 신뢰할 수 있는 AD 도메인이 사용자 인증서를 매핑하도록 구성된 경우 인증서 매핑 규칙 추가에 설명된 대로.

`map rule` 옵션의 필터 정의는 논리 연산자 `|` (또는)를 허용하므로 여러 기준을 지정할 수 있습니다. 이 경우 규칙은 기준 중 하나 이상을 충족하는 모든 사용자 계정을 매핑합니다.

57장. IDM 클라이언트의 데스크탑에 저장된 인증서를 사용하여 인증 구성

IdM(Identity Management)을 구성하여 **IdM** 시스템 관리자는 사용자가 **CA(인증 기관)**가 사용자에게 발급한 인증서를 사용하여 **IdM** 웹 **UI** 및 **CLI(명령줄 인터페이스)**에 대한 인증을 활성화할 수 있습니다.

웹 브라우저는 **IdM** 도메인에 속하지 않는 시스템에서 실행할 수 있습니다.

이 사용자 사례에서는 **IdM** 클라이언트의 데스크탑에 저장된 인증서를 사용하여 **ID** 관리 웹 **UI** 및 **CLI**에 효율적으로 로그인을 구성하고 테스트하는 방법에 대한 지침을 제공합니다. 이 사용자 사례를 따르는 경우

- 인증서를 사용하여 인증하려는 사용자에게 이미 인증서가 있는 경우 **57.2절. “새 사용자 인증서 요청 및 클라이언트로 내보내기”** 을(를) 건너뛸 수 있습니다.
- **IdM CA**에서 사용자 인증서를 발급한 경우 **57.3절. “인증서와 사용자가 함께 연결되었는지 확인”** 을(를) 건너뛸 수 있습니다.



참고

ID 관리 사용자만 인증서를 사용하여 웹 **UI**에 로그인할 수 있습니다. **Active Directory** 사용자는 사용자 이름 및 암호를 사용하여 로그인할 수 있습니다.

57.1. 웹 UI에서 인증서 인증을 위한 ID 관리 서버 구성

IdM(Identity Management) 관리자는 사용자가 인증서를 사용하여 **IdM** 환경에 인증할 수 있습니다.

절차

Identity Management 관리자로서 다음을 수행합니다.

1. **Identity Management** 서버에서 관리자 권한을 획득하고 셸 스크립트를 생성하여 서버를 구성합니다.

a.

ipa-adviser config-server-for-smart-card-auth 명령을 실행하고 출력을 파일에 저장합니다(예: **server_certificate_script.sh**).

```
# kinit admin
# ipa-adviser config-server-for-smart-card-auth > server_certificate_script.sh
```

b.

chmod 유틸리티를 사용하여 파일에 실행 권한을 추가합니다.

```
# chmod +x server_certificate_script.sh
```

2.

Identity Management 도메인의 모든 서버에서 **server_certificate_script.sh** 스크립트를 실행합니다.

a.

IdM 인증 기관 인증서의 경로인 **/etc/ipa/ca.crt**를 **IdM CA**가 인증서 인증을 활성화하려는 사용자의 인증서를 발급한 유일한 인증 기관인 경우 입력으로 입력됩니다.

```
# ./server_certificate_script.sh /etc/ipa/ca.crt
```

b.

서로 다른 외부 **CA**가 인증서 인증을 활성화하려는 사용자의 인증서에 서명한 경우 관련 **CA** 인증서를 입력으로 만드는 경로를 사용하여 다음을 수행합니다.

```
# ./server_certificate_script.sh /tmp/ca1.pem /tmp/ca2.pem
```

참고

전체 토폴로지에서 사용자의 인증서 인증을 활성화하려는 경우 나중에 시스템에 추가하는 각 새 복제본에서 스크립트를 실행해야 합니다.

57.2. 새 사용자 인증서 요청 및 클라이언트로 내보내기

IdM(Identity Management) 관리자는 **IdM** 환경의 사용자에게 대한 인증서를 생성하여 사용자의 인증서 인증을 활성화하려는 **IdM** 클라이언트에 내보낼 수 있습니다.

참고

인증서를 사용하여 인증하려는 사용자에게 인증서가 이미 있는 경우 이 섹션을 건너뛸 수 있습니다.

절차

1.

선택적으로 `~/certdb/` 와 같은 새 디렉터리를 생성하고 임시 인증서 데이터베이스로 만듭니다. 메시지가 표시되면 **NSS** 인증서 **DB** 암호를 생성하여 후속 단계에서 생성할 인증서의 키를 암호화합니다.

```
# mkdir ~/certdb/
# certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.
```

```
Enter new password:
Re-enter password:
```

2.

CSR(인증서 서명 요청)을 만들고 출력을 파일로 리디렉션합니다. 예를 들어 **IDM.EXAMPLE.COM** 영역에서 **idm_user** 사용자에게 대해 **certificate_request.csr** 이라는 이름으로 **CSR**을 생성하려면 찾기 쉽도록 인증서 개인 키의 닉네임을 **idm_user** 로 설정하고 제목을 **CN=idm_user,O=IDM.EXAMPLE.COM**:

```
# certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s
"CN=idm_user,O=IDM.EXAMPLE.COM" > certificate_request.csr
```

3.

메시지가 표시되면 **certutil** 을 사용하여 임시 데이터베이스를 만들 때 입력한 것과 동일한 암호를 입력합니다. 그런 다음 중지하도록 지시할 때까지 **rundlonly**를 계속 입력합니다.

Enter Password or Pin for "NSS Certificate DB":

A random seed must be generated that will be used in the creation of your key. One of the easiest ways to create a random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter is full. DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!

Continue typing until the progress meter is full:

4.

인증서 요청 파일을 서버에 제출합니다. 새로 발급된 인증서와 연결할 **Kerberos** 사용자, 인증서를 저장할 출력 파일 및 선택적으로 인증서 프로필을 지정합니다. 예를 들어, 사용자 역할 확장이 추가된 프로필, **idm_user@IDM.EXAMPLE.COM** 주체의 인증서를 가져오고 **~/idm_user.pem** 파일에 저장하려면 다음을 수행합니다.

```
# ipa cert-request certificate_request.csr --principal=idm_user@IDM.EXAMPLE.COM --
profile-id=IECUserRoles --certificate-out=~/idm_user.pem
```

5.

NSS 데이터베이스에 인증서를 추가합니다. **n** 옵션을 사용하여 인증서가 **NSS 데이터베이스**의 개인 키와 일치하도록 이전에 **CSR**을 만들 때 사용한 것과 동일한 앵커를 설정합니다. **t** 옵션은 신뢰 수준을 설정합니다. 자세한 내용은 **certutil(1)** 도움말 페이지를 참조하십시오. **i** 옵션은 입력 인증서 파일을 지정합니다. 예를 들어 **NSS 데이터베이스**에 **~/certdb/** 데이터베이스의 **~/idm_user.pem** 파일에 저장된 **idm_user** 닉네임 인증서를 추가하려면 다음을 수행합니다.

```
# certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

6.

NSS 데이터베이스의 키가 쿼리로 표시되지 않는지 확인합니다. 예를 들어 **~/certdb/** 데이터베이스에 저장된 인증서가 분리되지 않았는지 확인하려면 다음을 수행합니다.

```
# certutil -K -d ~/certdb/
< 0> rsa      5ad14d41463b87a095b1896cf0068ccc467df395  NSS Certificate
DB:idm_user
```

7.

pk12util 명령을 사용하여 **NSS 데이터베이스**에서 **PKCS12** 형식으로 인증서를 내보냅니다. 예를 들어 **/root/certdb** **NSS 데이터베이스**에서 **idm_user** 닉네임이 있는 인증서를 **~/idm_user.p12** 파일로 내보내려면 다음을 수행합니다.

```
# pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

8.

idm_user에 대한 인증서 인증을 활성화할 호스트에 인증서를 전송합니다.

```
# scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

9.

인증서가 전송된 호스트에서 보안상의 이유로 **.pkcs12** 파일이 **'other'** 그룹에 액세스할 수 없는 디렉토리를 만듭니다.

```
# chmod o-rwx /home/idm_user/
```

10.

보안상의 이유로 임시 **NSS 데이터베이스**와 **.pkcs12** 파일을 서버에서 제거합니다.

```
# rm ~/certdb/
# rm ~/idm_user.p12
```

57.3. 인증서와 사용자가 함께 연결되었는지 확인



참고

사용자 인증서를 **IdM CA**에서 발급한 경우 이 섹션을 건너뛸 수 있습니다.

인증서 인증이 작동하려면 인증서가 **IdM(Identity Management)**에 인증하는 데 사용할 사용자에게 연결되어 있는지 확인해야 합니다.

- **ID** 관리 환경에 속하지 않는 인증 기관에서 인증서를 제공하는 경우 사용자 계정과 인증서 연결에 설명된 절차에 따라 **사용자와 인증서를 연결합니다**.
- **ID** 관리 **CA**에서 인증서를 제공하는 경우 인증서는 이미 사용자 항목에 자동으로 추가되며 사용자 계정에 인증서를 연결할 필요가 없습니다. **IdM**에서 새 인증서를 생성하는 방법에 대한 자세한 내용은 **57.2절. “새 사용자 인증서 요청 및 클라이언트로 내보내기”**를 참조하십시오.

57.4. 인증서 인증을 활성화하도록 브라우저 구성

WebUI를 사용하여 **IdM(Identity Management)**에 로그인할 때 인증서로 인증하려면 사용자 및 관련 **CA(인증 기관)** 인증서를 **Mozilla Firefox** 또는 **Google Chrome** 브라우저로 가져와야 합니다. 브라우저가 실행 중인 호스트 자체는 **IdM** 도메인의 일부일 필요는 없습니다.

IdM은 웹 **UI**에 연결하기 위해 다음 브라우저를 지원합니다.

- **Mozilla Firefox 38 이상**
- **Google Chrome 46 이상**

다음 절차에서는 **Mozilla Firefox 57.0.1** 브라우저를 구성하는 방법을 보여줍니다.

사전 요구 사항

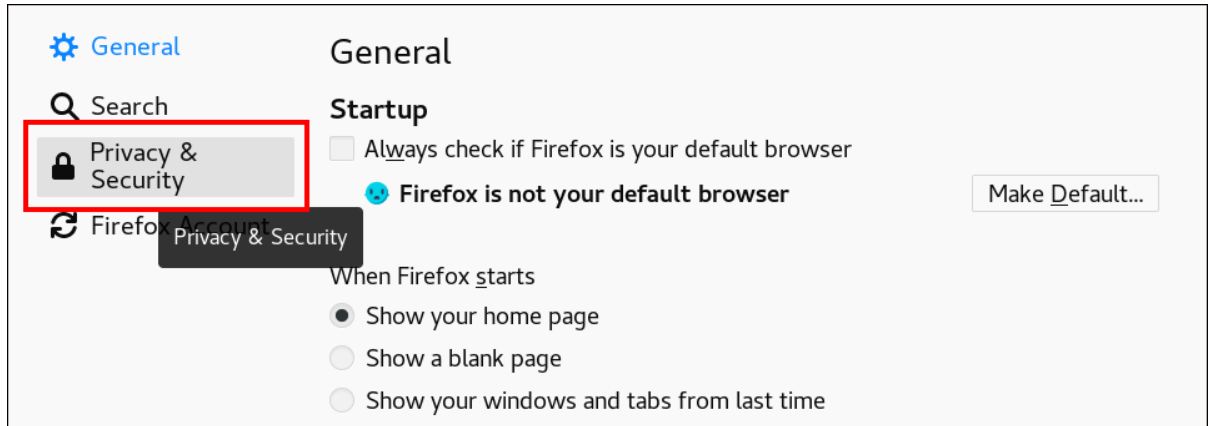
- **PKCS#12** 형식으로 브라우저에서 가져올 **사용자 인증서**가 있습니다.

절차

1.

Firefox를 열고 환경 설정 → 개인 정보 보호 및 보안 으로 이동합니다.

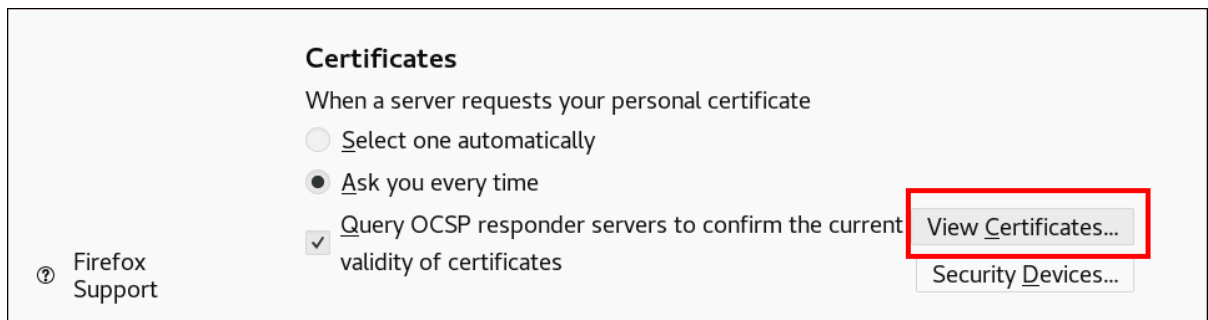
그림 57.1. 환경 설정의 개인 정보 보호 및 보안 섹션



2.

View Certificates (인증서 보기)를 클릭합니다.

그림 57.2. 개인 정보 보호 및 보안에서 인증서보기



3.

Certificates(인증서) 탭에서 **Import**(가져오기)를 클릭합니다. **PKCS12** 형식으로 사용자 인증서를 찾아 연 다음 **OK**(확인) 및 **OK** (확인)를 클릭합니다.

4.

ID 관리 인증 기관이 **Firefox**에서 신뢰할 수 있는 권한으로 인식되는지 확인합니다.

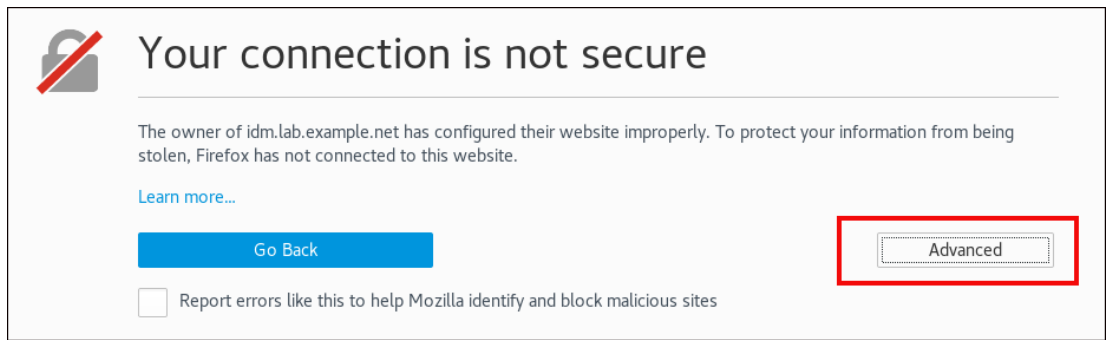
a.

IdM CA 인증서를 로컬에 저장합니다.

•

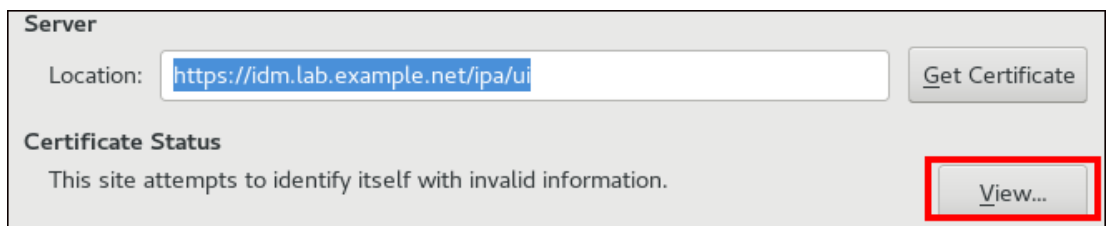
Firefox 주소 표시줄에 **IdM** 서버 이름을 작성하여 **IdM** 웹 UI로 이동합니다. 비보안 연결 경고 페이지에서 **Advanced**(고급)를 클릭합니다.

그림 57.3. 안전하지 않은 연결



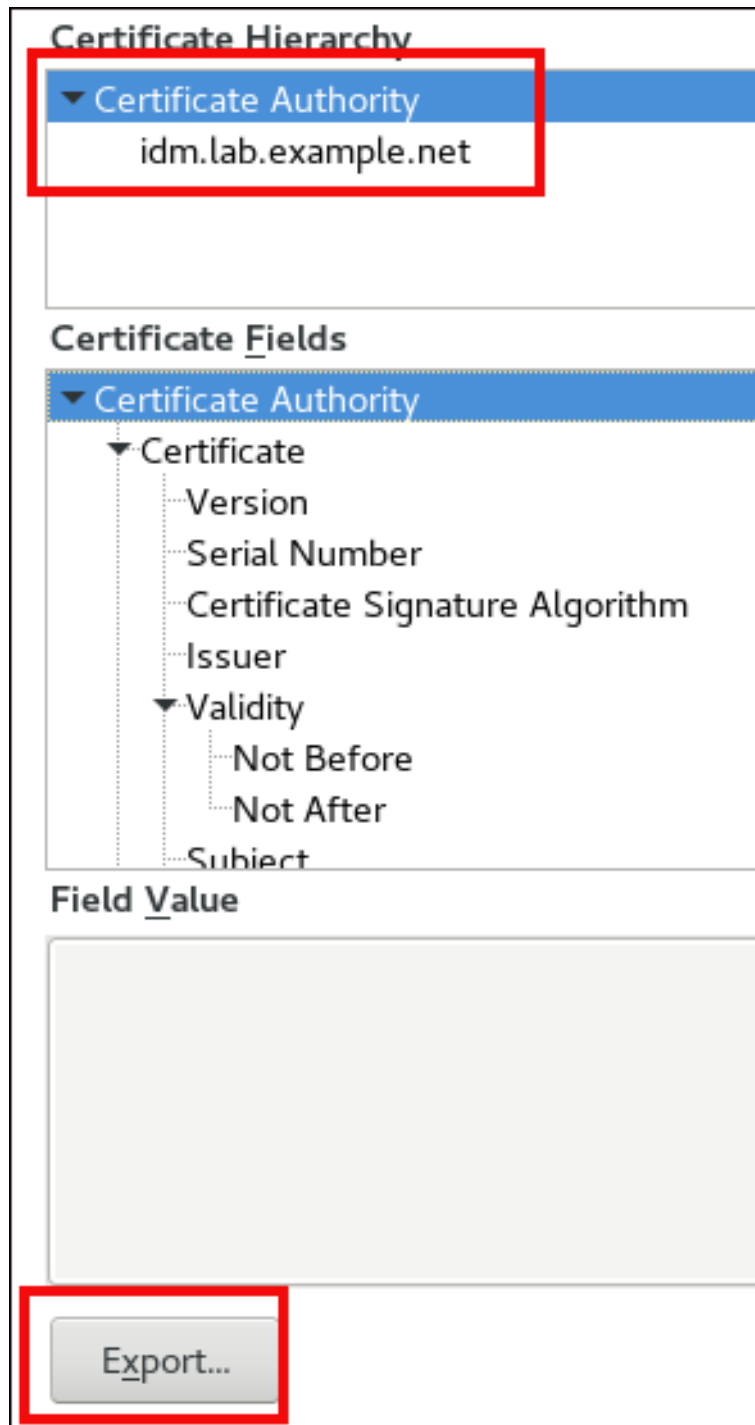
- 예외 추가. **View(보기)**를 클릭합니다.

그림 57.4. 인증서 세부 정보 보기



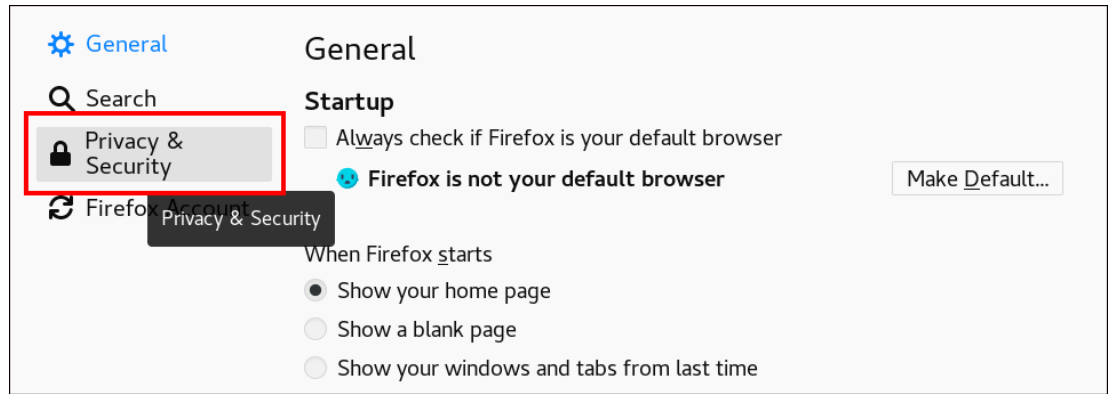
- Details(세부 정보)** 탭에서 인증 기관 필드를 강조 표시합니다.

그림 57.5. CA 인증서 내보내기



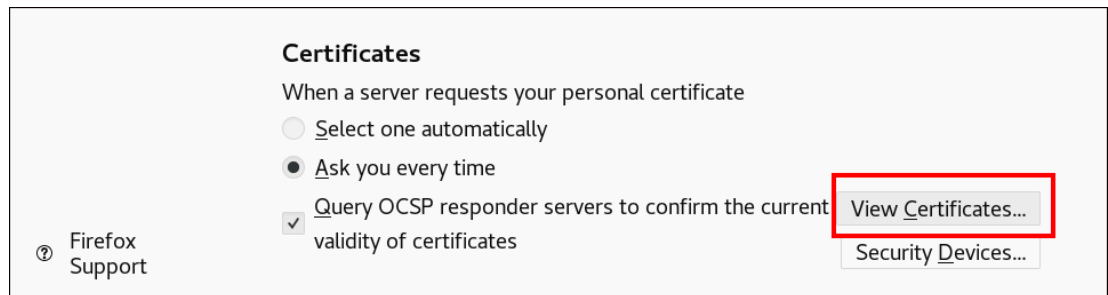
- **Export(내보내기)**를 클릭합니다. CA 인증서를(예: **CertificateAuthority.crt** 파일로 저장)한 다음 **Close(닫기)** 및 **Cancel(취소)**을 클릭합니다.
- b. 신뢰할 수 있는 인증 기관 인증서로 **Firefox**로 **IdM CA** 인증서를 가져옵니다.
- **Firefox**를 열고 **Preferences(기본 설정)**로 이동한 다음 **Privacy & Security(개인 정보 보호 및 보안)**를 클릭합니다.

그림 57.6. 환경 설정의 개인 정보 보호 및 보안 섹션



• **View Certificates (인증서 보기)**를 클릭합니다.

그림 57.7. 개인 정보 보호 및 보안에서 인증서보기



• **Authorities(권한)** 탭에서 **Import(가져오기)**를 클릭합니다.
CertificateAuthority.crt 파일에서 이전 단계에서 저장한 CA 인증서를 찾아 엽니다. 인증서를 신뢰하여 웹사이트를 식별한 다음 **OK(확인)** 및 **OK (확인)**를 클릭합니다.

5.

Identity Management User(ID 관리 사용자)로 인증서를 사용하여 **Identity Management Web UI**에 계속 인증합니다.

57.5. ID 관리 사용자로 인증서를 사용하여 ID 관리 웹 UI에 인증

이 절차에서는 ID 관리 클라이언트의 데스크탑에 저장된 인증서를 사용하여 IdM(Identity Management) 웹 UI에 대한 사용자 인증에 대해 설명합니다.

절차

1.

브라우저에서 **Identity Management(ID 관리) 웹 UI**(예 : <https://server.idm.example.com/ipa/ui>)로 이동합니다.

2.

Login(로그인) using Certificate(인증서 사용)를 클릭합니다.

. ID 관리 웹 UI에서 인증서를 사용하여 로그인

3.

사용자의 인증서가 이미 선택되어 있어야 합니다. 이 결정의 선택을 취소하고 **OK(확인)**를 클릭합니다.

이제 인증서에 해당하는 사용자로 인증되었습니다.

추가 리소스

•

스마트 카드에 저장된 인증서를 사용하여 IdM 웹 UI에 인증하는 방법에 대한 자세한 내용은 [스마트 카드 인증을 위한 ID 관리](#) 구성을 참조하십시오.

57.6. 인증서를 사용하여 CLI에 대한 인증을 활성화하도록 IDM 클라이언트 구성

IdM 클라이언트의 CLI(명령줄 인터페이스)에서 IdM 사용자에게 대한 인증서 인증을 수행하려면 IdM 사용자의 인증서와 개인 키를 IdM 클라이언트로 가져옵니다. 사용자 인증서 생성 및 전송에 대한 자세한 내용은 [57.2절. “새 사용자 인증서 요청 및 클라이언트로 내보내기”](#)을 참조하십시오.

절차

•

IdM 클라이언트에 로그인하고 사용자의 인증서와 개인 키가 준비된 .p12 파일을 준비합니다. TGT(권한)를 부여하는 Kerberos 티켓을 가져와 캐시하려면 `X509_username:/path/to/file.p12` 속성과 함께 `-X` 옵션을 사용하여 사용자 주체와 함께 `kinit` 명령을 실행하여 사용자의 X509 ID 정보를 찾을 위치를 지정합니다. 예를 들어 `~/idm_user.p12` 파일에 저장된 사용자의 ID 정보를 사용하여 `idm_user`의 TGT를 가져오려면 다음을 수행합니다.

```
$ kinit -X X509_idm_user='PKCS12:~/idm_user.p12' idm_user
```



참고

명령은 **.pem** 파일 형식도 지원합니다. **kinit -X X509_username='FILE:/path/to/cert.pem,/path/to/key' user_principal**

58장. IDM CA 갱신 서버 사용

58.1. IDM CA 갱신 서버 설명

임베디드 CA(인증 기관)를 사용하는 **IdM(Identity Management)** 배포에서 **CA** 갱신 서버는 **IdM** 시스템 인증서를 유지 관리하고 갱신합니다. 강력한 **IdM** 배포를 보장합니다.

IdM 시스템 인증서는 다음과 같습니다.

- **IdM CA 인증서**
- **OCSP 서명 인증서**
- **IdM CA 하위 시스템 인증서**
- **IdM CA 감사 서명 인증서**
- **IdM 갱신 에이전트 (RA) 인증서**
- **KRA 전송 및 스토리지 인증서**

시스템 인증서의 특징화는 해당 키를 모든 **CA** 복제본에서 공유한다는 것입니다. 반대로 **IdM** 서비스 인증서(예: **LDAP**, **HTTP** 및 **PKINIT** 인증서)에는 다양한 **IdM CA** 서버의 키 쌍과 주체 이름이 다릅니다.

IdM 토폴로지에서 기본적으로 첫 번째 **IdM CA** 서버는 **CA** 갱신 서버입니다.



참고

업스트림 설명서에서 **IdM CA**는 **Dogtag** 라고 합니다.

CA 갱신 서버의 역할

IdM CA, IdM CA 하위 시스템 및 IdM RA 인증서는 **IdM** 배포에 중요합니다. 각 인증서는 **/etc/pki/pki-tomcat/** 디렉터리 및 **LDAP** 데이터베이스 항목의 **NSS** 데이터베이스에 저장됩니다. **LDAP**에 저장된 인증서는 **NSS** 데이터베이스에 저장된 인증서와 일치해야 합니다. 일치하지 않는 경우 **IdM** 프레임워크와 **IdM CA**와 **IdM CA**와 **IdM CA** 간에 인증 오류가 발생합니다.

모든 **IdM CA** 복제본에는 모든 시스템 인증서에 대한 요청 추적이 있습니다. 통합 **CA**를 사용한 **IdM** 배포에 **CA** 갱신 서버가 없는 경우 각 **IdM CA** 서버는 시스템 인증서 갱신을 독립적으로 요청합니다. 이로 인해 서로 다른 **CA** 복제본이 다양한 시스템 인증서 및 인증 실패가 발생합니다.

갱신 서버로 하나의 **CA** 복제본을 지정하면 시스템 인증서가 필요한 경우 정확히 한 번만 갱신할 수 있으므로 인증 실패를 방지할 수 있습니다.

CA 복제본에서 certmonger 서비스의 역할

모든 **IdM CA** 복제본에서 실행되는 **certmonger** 서비스는 **dogtag-ipa-ca-renew-agent** 갱신 도우미를 사용하여 **IdM** 시스템 인증서를 추적합니다. 갱신 도우미 프로그램은 **CA** 갱신 서버 구성을 읽습니다. **CA** 갱신 서버가 아닌 각 **CA** 복제본에서 갱신 도우미는 **ca_renewal LDAP** 항목에서 최신 시스템 인증서를 검색합니다. 정확히 **certmonger** 갱신 시도가 발생하는 경우 결정적이지 않은 것으로 인해 **dogtag-ipa-ca-renew-agent** 도우미는 **CA** 갱신 서버가 실제로 인증서를 갱신하기 전에 시스템 인증서를 업데이트하려고 시도하는 경우가 있습니다. 이 경우 **CA** 복제본의 **certmonger** 서비스로 이전의 곧 만료 인증서가 반환됩니다. 데이터베이스에 이미 저장된 인증서가 동일한 인증서임을 인식하는 **certmonger** 서비스는 **CA** 갱신 서버에서 업데이트된 인증서를 검색할 수 있을 때까지 개별 시도 간에 몇 가지 지연으로 인증서를 갱신합니다.

IdM CA 갱신 서버의 올바른 기능

포함된 **CA**가 포함된 **IdM** 배포는 **IdM CA**와 함께 또는 나중에 **IdM CA** 서버가 설치된 **IdM** 배포입니다. 포함된 **CA**가 포함된 **IdM** 배포에는 항상 정확히 하나의 **CA** 복제본이 갱신 서버로 구성되어 있어야 합니다. 갱신 서버는 온라인 상태이며 완전히 작동해야 하며 다른 서버와 올바르게 복제해야 합니다.

현재 **CA** 갱신 서버가 **ipa server-del**, **ipa-replica-manage del**, **ipa-csreplica-manage del** 또는 **ipa-server-install --uninstall** 명령을 사용하여 삭제되는 경우 다른 **CA** 복제본이 **CA** 갱신 서버로 자동으로 할당됩니다. 이 정책은 갱신 서버 구성이 유효한 상태로 유지됩니다.

이 정책은 다음 상황을 다루지 않습니다.

-

오프라인 갱신 서버

갱신 서버가 연장된 기간 동안 오프라인 상태인 경우 갱신 창이 누락될 수 있습니다. 이 경우 모든 업데이트되지 않은 **CA** 서버는 인증서가 만료될 때까지 현재 시스템 인증서를 다시 설치합니다. 이 경우 만료된 인증서 한 개도 다른 인증서에 대해 갱신 실패할 수 있으므로 **IdM** 배포가 중단됩니다.

이러한 상황을 방지하려면 현재 갱신 서버가 오프라인 상태이고 연장된 기간 동안 사용할 수 없는 경우 **새 CA 갱신 서버를 수동으로 할당하는** 것이 좋습니다.

-

복제 문제

갱신 서버와 기타 **CA** 복제본 간에 복제 문제가 있는 경우 갱신이 성공할 수 있지만 다른 **CA** 복제본은 만료되기 전에 업데이트된 인증서를 검색하지 못할 수 있습니다.

이러한 상황을 방지하기 위해 복제 계약이 올바르게 작동하는지 확인합니다. 자세한 내용은 **RHEL 7 Linux** 도메인 ID, 인증 및 정책 가이드의 **일반** 또는 **특정** 복제 문제 해결 지침을 참조하십시오.

58.2. IDM CA 갱신 서버 변경 및 재설정

CA(인증 기관) 갱신 서버가 해제되면 **IdM**(Identity Management)은 **IdM CA** 서버 목록에서 새 **CA** 갱신 서버를 자동으로 선택합니다. 시스템 관리자가 선택 항목에 영향을 줄 수 없습니다.

새 **IdM CA** 갱신 서버를 선택하려면 시스템 관리자가 수동으로 교체를 수행해야 합니다. 현재 갱신 서버를 해제하는 프로세스를 시작하기 전에 새 **CA** 갱신 서버를 선택합니다.

현재 **CA** 갱신 서버 구성이 유효하지 않으면 **IdM CA** 갱신 서버를 재설정합니다.

CA 갱신 서버를 변경하거나 재설정하려면 이 절차를 완료하십시오.

사전 요구 사항

-

IdM 관리자 자격 증명이 있습니다.

절차

- 1.

IdM 관리자 인증 정보를 가져옵니다.

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

- 2.

배포에서 새 **CA** 갱신 서버가 되기 위해 **CA** 역할이 필요한 **IdM** 서버를 찾으려면 선택적으로 다음을 수행합니다.

```
~]$ ipa server-role-find --role 'CA server'
-----
2 server roles matched
-----
Server name: server.idm.example.com
Role name: CA server
Role status: enabled

Server name: replica.idm.example.com
Role name: CA server
Role status: enabled
-----
Number of entries returned 2
-----
```

배포에는 두 개의 **CA** 서버가 있습니다.

3.

필요한 경우 **CA** 서버가 현재 **CA** 갱신 서버인 **CA** 서버를 찾으려면 다음을 입력합니다.

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

현재 갱신된 서버는 **server.idm.example.com** 입니다.

4.

갱신 서버 구성을 변경하려면 **--ca-renewal -master-server** 옵션과 함께 **ipa config- mod** 유틸리티를 사용합니다.

```
~]$ ipa config-mod --ca-renewal-master-server replica.idm.example.com | grep 'CA renewal'
IPA CA renewal master: replica.idm.example.com
```

중요

다음을 사용하여 새 **CA** 갱신 서버로도 전환할 수 있습니다.

- **ipa-cacert-manage --renew** 명령. 이 명령은 모두 **CA** 인증서를 갱신 하고 명령을 새 **CA** 갱신 서버를 실행하는 **CA** 서버를 만듭니다.
- **ipa-cert-fix** 명령. 이 명령은 만료된 인증서로 인해 실패할 때 배포를 복구합니다. 또한 명령을 새 **CA** 갱신 서버로 실행하는 **CA** 서버를 만듭니다.

자세한 내용은 **IdM이 오프라인 상태일 때 만료된 시스템 인증서 갱신** 을 참조하십시오.

58.3. IDM의 자체 서명된 CA로 외부에서 전환

외부 서명에서 **IdM(Identity Management)** 인증 기관(**CA**)의 자체 서명 인증서로 전환하려면 이 절차를 완료합니다. 자체 서명된 **CA**를 사용하면 **CA** 인증서 갱신이 자동으로 관리됩니다. 시스템 관리자가 **CSR**(인증서 서명 요청)을 외부 기관에 제출할 필요가 없습니다.

외부 서명에서 자체 서명된 **CA**로 전환하면 **CA** 인증서만 교체됩니다. 이전 **CA**에서 서명한 인증서는 계속 유효하며 계속 사용됩니다. 예를 들어 자체 서명된 **CA**로 이동한 후에도 **LDAP** 인증서의 인증서 체인은 변경되지 않습니다.

external_CA certificate > IdM CA certificate > LDAP certificate

사전 요구 사항

- **IdM CA** 갱신 서버에 대한 루트 액세스 권한이 있습니다.
- **IdM** 관리자 자격 증명이 있습니다.

절차

1.

IdM CA 갱신 서버에서 자체 서명으로 **CA** 인증서를 갱신합니다.

```
~]# ipa-cacert-manage renew --self-signed
Renewing CA certificate, please wait
```

```
CA certificate successfully renewed
The ipa-cacert-manage command was successful
```

2.

모든 **IdM** 서버 및 클라이언트에서 서버의 인증서를 사용하여 로컬 **IdM** 인증서 데이터베이스를 업데이트합니다.

```
[client ~]$ kinit admin
[client ~]$ ipa-certupdate
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```

3.

선택적으로 업데이트에 성공했으며 새 **CA** 인증서가 **/etc/ipa/ca.crt** 파일에 추가되었는지 확인하려면 다음을 수행합니다.

```
[client ~]$ openssl crl2pkcs7 -nocrl -certfile /etc/ipa/ca.crt | openssl pkcs7 -print_certs
-text -noout
[...]
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 39 (0x27)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: O=IDM.EXAMPLE.COM, CN=Certificate Authority
    Validity
      Not Before: Jul  1 16:32:45 2019 GMT
      Not After : Jul  1 16:32:45 2039 GMT
    Subject: O=IDM.EXAMPLE.COM, CN=Certificate Authority
  [...]

```

새 **CA** 인증서가 이전 **CA** 인증서와 함께 나열되므로 출력에 업데이트가 성공했음을 보여줍니다.

58.4. 외부 서명 인증서를 사용하여 IDM CA 갱신 서버 갱신

이 섹션에서는 **CSR**(인증서 서명 요청)에 서명하기 위해 외부 **CA**를 사용하여 **IdM**(Identity Management) **CA**(인증 기관) 인증서를 갱신하는 방법을 설명합니다. 이 구성에서 **IdM CA** 서버는 외부 **CA**의 하위 **CA**입니다. 외부 **CA**는 **Active Directory Certificate Server(ADPS)**일 수 있지만 반드시 필요하지는 않습니다.

외부 인증 기관이 **AD CS**인 경우 **CSR**에서 **IdM CA** 인증서에 사용할 템플릿을 지정할 수 있습니다. 인증서 템플릿은 인증서 요청이 수신될 때 **CA**에서 사용하는 정책과 규칙을 정의합니다. **AD**의 인증서 템플릿은 **IdM**의 인증서 프로필에 해당합니다.

OID(Object Identifier)로 특정 **ADVC** 템플릿을 정의할 수 있습니다. **OID**는 분산된 애플리케이션의 데이터 요소, 구문 및 기타 부분을 고유하게 식별하기 위해 다양한 발행 기관에서 발행한 고유한 숫자 값입니다.

또는 특정 **ADsu** 템플릿을 이름으로 정의할 수도 있습니다. 예를 들어 **IdM CA**에서 **ADVC**로 제출한 **CSR**에서 사용되는 **default** 프로파일의 이름은 **subCA**입니다.

CSR에서 **OID** 또는 이름을 지정하여 프로파일을 정의하려면 **external-ca-profile** 옵션을 사용합니다. 자세한 내용은 **ipa-cacert-manage** 도움말 페이지를 참조하십시오.

준비된 인증서 템플릿을 사용하는 것 외에도 **AD CS**에 사용자 지정 인증서 템플릿을 생성하여 **CSR**에서 사용할 수도 있습니다.

사전 요구 사항

- **IdM CA** 갱신 서버에 대한 루트 액세스 권한이 있습니다.
- **IdM** 관리자 자격 증명이 있습니다.

절차

현재 **CA** 인증서가 자체 서명되었는지 또는 외부 서명인지 여부에 관계없이 **IdM CA**의 인증서를 외부 서명으로 갱신하려면 이 절차를 완료합니다.

1.

외부 **CA**에 제출할 **CSR**을 생성합니다.

- 외부 **CA**가 **AD CS**인 경우 **--external-ca-type=ms-cs** 옵션을 사용합니다. 기본 **subCA** 템플릿과 다른 템플릿을 원하는 경우 **--external-ca-profile** 옵션을 사용하여 지정합니다.

```
~]# ipa-cacert-manage renew --external-ca --external-ca-type=ms-cs [--external-ca-profile=PROFILE]
```

Exporting CA certificate signing request, please wait

The next step is to get /var/lib/ipa/ca.csr signed by your CA and re-run ipa-cacert-manage as:

```
ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate
```

The ipa-cacert-manage command was successful

- 외부 CA가 AD가 아닌 경우:

```
~]# ipa-cacert-manage renew --external-ca
Exporting CA certificate signing request, please wait
The next step is to get /var/lib/ipa/ca.csr signed by your CA and re-run ipa-cacert-
manage as:
ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-
cert-file=/path/to/external_ca_certificate
The ipa-cacert-manage command was successful
```

출력은 CSR이 생성되었으며 /var/lib/ipa/ca.csr 파일에 저장되어 있음을 보여줍니다.

2.

/var/lib/ipa/ca.csr 에 있는 CSR을 외부 CA에 제출합니다. 프로세스는 외부 CA로 사용할 서
비스에 따라 다릅니다.

3.

다음과 같은 기본 64 인코딩 Blob에서 발급된 CA의 발급된 인증서 및 CA 인증서 체인을 검색
합니다.

- 외부 CA가 AD CS가 아닌 경우 PEM 파일.

- 외부 CA가 AD CS인 경우 Base_64 인증서입니다.

프로세스는 각 인증서 서비스에 따라 다릅니다. 일반적으로 웹 페이지 또는 알림 이메일
의 다운로드 링크를 통해 관리자는 필요한 모든 인증서를 다운로드할 수 있습니다.

외부 CA가 AD CS이고 Microsoft Windows Certification Authority 관리 창을 통해 알
려진 템플릿이 있는 CSR을 제출한 경우 ADsu는 인증서를 즉시 발급하고 Save
Certificate(인증서 저장) 대화 상자가 AD OSD 웹 인터페이스에 표시되며, 발급된 인증서를
저장할 위치를 묻는 메시지가 표시됩니다.

4.

ipa-cacert-manage renew 명령을 다시 실행하여 전체 인증서 체인을 제공하는 데 필요한 모
든 CA 인증서 파일을 추가합니다. --external-cert-file 옵션을 여러 번 사용하여 필요한 만큼 파일
을 지정합니다.

```
~]# ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-
cert-file=/path/to/external_ca_certificate_1 --external-cert-
file=/path/to/external_ca_certificate_2
```

5.

모든 **IdM** 서버 및 클라이언트에서 서버의 인증서를 사용하여 로컬 **IdM** 인증서 데이터베이스를 업데이트합니다.

```
[client ~]$ kinit admin
[client ~]$ ipa-certupdate
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```

6.

선택적으로 업데이트에 성공했으며 새 **CA** 인증서가 **/etc/ipa/ca.crt** 파일에 추가되었는지 확인하려면 다음을 수행합니다.

```
[client ~]$ openssl crl2pkcs7 -nocrl -certfile /etc/ipa/ca.crt | openssl pkcs7 -print_certs
-text -noout
[...]
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 39 (0x27)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: O=IDM.EXAMPLE.COM, CN=Certificate Authority
    Validity
      Not Before: Jul  1 16:32:45 2019 GMT
      Not After : Jul  1 16:32:45 2039 GMT
    Subject: O=IDM.EXAMPLE.COM, CN=Certificate Authority
  [...]

```

새 **CA** 인증서가 이전 **CA** 인증서와 함께 나열되므로 출력에 업데이트가 성공했음을 보여줍니다.

59장. IDM이 오프라인 상태일 때 만료된 시스템 인증서 갱신

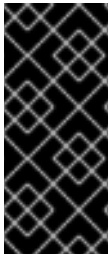
시스템 인증서가 만료되면 **IdM(Identity Management)**이 시작되지 않습니다. **IdM**은 **ipa-cert-fix** 툴을 사용하여 **IdM**이 오프라인인 경우 시스템 인증서 갱신을 지원합니다.

사전 요구 사항

- **IdM**은 **Red Hat Enterprise Linux 8.1** 이상에만 설치됩니다.

59.1. CA 갱신 서버에서 만료된 시스템 인증서 갱신

이 섹션에서는 만료된 **IdM** 인증서에 **ipa-cert-fix** 툴을 적용하는 방법에 대해 설명합니다.



중요

CA 갱신 서버가 아닌 **CA(Certificate Authority)** 호스트에서 **ipa-cert-fix** 툴을 실행하고 유틸리티에서 공유 인증서를 갱신하면 해당 호스트가 도메인의 새 **CA** 갱신 서버가 됩니다. 불일치를 방지하려면 도메인에는 항상 하나의 **CA** 갱신 서버만 있어야 합니다.

사전 요구 사항

- 관리 권한이 있는 서버에 로그인합니다.

절차

1. **ipa-cert-fix** 툴을 시작하여 시스템을 분석하고 갱신이 필요한 만료된 인증서를 나열합니다.

```
# ipa-cert-fix
```

```
...
```

```
The following certificates will be renewed:
```

```
Dogtag sslserver certificate:
```

```
Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
```

```
Serial: 13
```

```
Expires: 2019-05-12 05:55:47
```

```
...
```

```
Enter "yes" to proceed:
```

2. **yes** 를 입력하여 갱신 프로세스를 시작합니다.

```

Enter "yes" to proceed: yes
Proceeding.
Renewed Dogtag sslserver certificate:
  Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
  Serial: 268369925
  Expires: 2021-08-14 02:19:33
...

Becoming renewal master.
The ipa-cert-fix command was successful

```

ipa-cert-fix 가 완료된 모든 인증서를 갱신하는 데 최대 1분이 걸릴 수 있습니다.

3.

필요한 경우 모든 서비스가 현재 실행 중인지 확인합니다.

```

# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa: INFO: The ipactl command was successful

```

이때 인증서가 갱신되어 서비스가 실행 중입니다. 다음 단계는 **IdM** 도메인의 다른 서버를 확인하는 것입니다.

참고

여러 **CA** 서버에서 인증서를 복구해야 하는 경우:

1. **LDAP** 복제가 토폴로지 전체에서 작동하는지 확인한 후 먼저 위의 절차에 따라 하나의 **CA** 서버에서 **ipa-cert-fix** 를 실행합니다.
2. 다른 **CA** 서버에서 **ipa-cert-fix** 를 실행하기 전에 **getcert-resubmit** 을 통해 공유 인증서에 대한 **Certmonger** 갱신을 트리거하여 공유 인증서의 불필요한 갱신을 방지합니다.

59.2. 갱신 후 IDМ 도메인에서 다른 IDМ 서버 확인

ipa-cert-fix 툴을 사용하여 **CA** 갱신 서버의 인증서를 갱신한 후에는 다음을 수행해야 합니다.

- 도메인에서 다른 모든 **IdM(Identity Management)** 서버를 다시 시작합니다.
- **certmonger** 갱신된 인증서가 있는지 확인합니다.
- 만료된 시스템 인증서가 있는 다른 **CA(인증 기관)** 복제본이 있는 경우 **ipa-cert-fix** 툴로 해당 인증서를 갱신합니다.

사전 요구 사항

- 관리 권한이 있는 서버에 로그인합니다.

절차

1. **--force** 매개변수를 사용하여 **IdM**을 다시 시작합니다.

```
# ipactl restart --force
```

--force 매개변수를 사용하면 **ipactl** 유틸리티에서 개별 서비스 시작 실패를 무시합니다. 예를 들어 서버가 만료된 인증서가 있는 **CA**인 경우 **pki-tomcat** 서비스가 시작되지 않습니다. **--force** 매개 변수를 사용하기 때문에 예상되고 무시됩니다.

2. 재시작 후 **certmonger** 서비스가 인증서를 갱신했는지 확인합니다(인증 상태가 **MONITORING**으로 표시됨).

```
# getcert list | egrep '^Request/status:/subject:'
Request ID '20190522120745':
  status: MONITORING
  subject: CN=IPA RA,O=EXAMPLE.COM 201905222205
Request ID '20190522120834':
  status: MONITORING
  subject: CN=Certificate Authority,O=EXAMPLE.COM 201905222205
...
```

certmonger에서 복제본에서 공유 인증서를 갱신하는 데 시간이 걸릴 수 있습니다.

- 3.

서버가 **CA**인 경우 이전 명령에서는 **pki-tomcat** 서비스에서 사용하는 인증서에 대해 **CA_UNREACHABLE** 을 보고합니다.

```
Request ID '20190522120835':
  status: CA_UNREACHABLE
  subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
...
```

4.

이 인증서를 갱신하려면 **ipa-cert-fix** 유틸리티를 사용합니다.

```
# ipa-cert-fix
Dogtag sslserver certificate:
  Subject: CN=ca2.example.com,O=EXAMPLE.COM
  Serial: 3
  Expires: 2019-05-11 12:07:11

Enter "yes" to proceed: yes
Proceeding.
Renewed Dogtag sslserver certificate:
  Subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
  Serial: 15
  Expires: 2019-08-14 04:25:05

The ipa-cert-fix command was successful
```

이제 모든 **IdM** 인증서가 갱신되어 올바르게 작동합니다.

59.3. 웹 서버 및 **LDAP** 서버 인증서 교체

IdM(Identity Management) 시스템 관리자는 **IdM** 서버에서 실행되는 웹(또는 **httpd**) 및 **LDAP**(또는 디렉터리)서비스의 인증서를 수동으로 교체할 수 있습니다. 예를 들어 **certmonger** 유틸리티가 인증서를 자동으로 갱신하도록 구성되지 않았거나 인증서가 외부 **CA**(인증 기관)에서 서명한 경우 이 작업이 필요할 수 있습니다.

이 예제에서는 **server.idm.example.com** **IdM** 서버에서 실행되는 서비스에 대한 인증서를 설치합니다. 외부 **CA**에서 인증서를 가져옵니다.



참고

HTTP 및 **LDAP** 서비스 인증서의 키 쌍과 제목 이름은 서로 다르므로 각 **IdM** 서버에서 인증서를 개별적으로 갱신해야 합니다.

사전 요구 사항

- **IdM 서버에 대한 루트 액세스 권한이 있습니다.**
- **Directory Manager 암호를 알고 있습니다.**
- 외부 CA의 CA 인증서 체인 **ca_certificate_chain_file.crt** 를 저장하는 파일에 액세스할 수 있습니다.

절차

1. **ca_certificate_chain_file.crt** 에 포함된 인증서를 IdM에 추가 CA 인증서로 설치합니다.

```
# ipa-cacert-manage install
```

2. **ca_certificate_chain_file.crt**의 인증서로 로컬 IdM 인증서 데이터베이스를 업데이트하십시오.

```
# ipa-certupdate
```

3. OpenSSL 유틸리티를 사용하여 개인 키 및 CSR(인증서 서명 요청)을 생성합니다.

```
$ openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout new.key -out new.csr -  
addext "subjectAltName = DNS:ipa-ca.idm.example.test" -subj  
'/CN=server.idm.example.com,O=IDM.EXAMPLE.COM'
```

CSR을 외부 CA에 제출합니다. 프로세스는 외부 CA로 사용할 서비스에 따라 다릅니다. CA가 인증서에 서명하고 나면 인증서를 IdM 서버로 가져옵니다.

4. IdM 서버에서 Apache 웹 서버의 기존 개인 키와 인증서를 새 키 및 새로 서명한 인증서로 교체합니다.

```
# ipa-server-certinstall -w --pin=password new.key new.crt
```

위의 명령에서 다음을 수행합니다.

- **w** 옵션은 웹 서버에 인증서를 설치 중임을 지정합니다.

- **pin** 옵션은 개인 키를 보호하는 암호를 지정합니다.

5. 메시지가 표시되면 **Directory Manager** 암호를 입력합니다.

6. **LDAP** 서버의 기존 개인 키와 인증서를 새 키 및 새로 서명한 인증서로 교체합니다.

```
# ipa-server-certinstall -d --pin=password new.key new.cert
```

위의 명령에서 다음을 수행합니다.

- **d** 옵션은 **LDAP** 서버에 인증서를 설치하도록 지정합니다.

- **pin** 옵션은 개인 키를 보호하는 암호를 지정합니다.

7. 메시지가 표시되면 **Directory Manager** 암호를 입력합니다.

8. **httpd** 서비스를 다시 시작하십시오.

```
# systemctl restart httpd.service
```

9. **Directory** 서비스를 다시 시작하십시오.

```
# systemctl restart dirsrv@IDM.EXAMPLE.COM.service
```

추가 리소스

- [IdM과 함께 작동하도록 인증서 형식 변환](#)
- [ipa-server-certinstall\(1\) 도움말 페이지](#)

60장. IDM CA 서버에서 CRL 생성

IdM 배포에서 포함된 CA(인증 기관)를 사용하는 경우 하나의 IdM(Identity Management) 서버에서 다른 서버로 CRL(Certificate Revocation List) 생성 생성을 이동해야 할 수 있습니다. 예를 들어 서버를 다른 시스템으로 마이그레이션하려는 경우 필요할 수 있습니다.

CRL을 생성하도록 하나의 서버만 구성합니다. CRL publisher 역할을 수행하는 IdM 서버는 일반적으로 CA 갱신 서버 역할을 수행하는 서버와 동일하지만 필수는 아닙니다. CRL 게시자를 해제하기 전에 CRL Publisher 서버 역할을 수행하도록 다른 서버를 선택하고 구성합니다.

이 장에서는 다음을 설명합니다.

- IdM 서버에서 CRL 생성 중지.
- IdM 복제본에서 CRL 생성 시작.

60.1. IDM 서버에서 CRL 생성 중지

IdM CRL 게시자 서버에서 CRL(인증서 폐기 목록) 생성을 중지하려면 `ipa-crlgen-manage` 명령을 사용합니다. 생성을 비활성화하기 전에 서버가 실제로 CRL을 생성하는지 확인합니다. 그런 다음 비활성화할 수 있습니다.

사전 요구 사항

- IdM(Identity Management) 서버는 RHEL 8.1 시스템에 설치되어 있습니다.
- root로 로그인해야 합니다.

절차

1. 서버가 CRL을 생성하는지 확인합니다.

```
[root@server ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

2.

서버에서 **CRL** 생성을 중지합니다.

```
[root@server ~]# ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL
generation on another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

3.

서버가 **CRL** 생성을 중지했는지 확인합니다.

```
[root@server ~]# ipa-crlgen-manage status
```

서버가 **CRL** 생성을 중지했습니다. 다음 단계는 **IdM** 복제본에서 **CRL** 생성을 활성화하는 것입니다.

60.2. IdM 복제본 서버에서 CRL 생성 시작

ipa-crlgen-manage 명령을 사용하여 **IdM CA** 서버에서 **CRD**(Certificate Revocation List) 생성을 시작할 수 있습니다.

사전 요구 사항

- **IdM**(Identity Management) 서버는 **RHEL 8.1** 시스템에 설치되어 있습니다.
- **RHEL** 시스템은 **IdM** 인증 기관 서버여야 합니다.
- **root**로 로그인해야 합니다.

절차

1.

CRL 생성을 시작합니다.

```
[root@replica1 ~]# ipa-crlgen-manage enable
Stopping pki-tomcatd
```

```
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg  
Starting pki-tomcatd  
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf  
Restarting httpd  
Forcing CRL update  
CRL generation enabled on the local host. Please make sure to have only a single CRL  
generation master.  
The ipa-crlgen-manage command was successful
```

2.

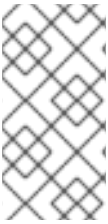
CRL이 생성되었는지 확인합니다.

```
[root@replica1 ~]# ipa-crlgen-manage status  
CRL generation: enabled  
Last CRL update: 2019-10-31 12:10:00  
Last CRL Number: 7  
The ipa-crlgen-manage command was successful
```

61장. CA 갱신 서버 및 CRL 게시자 역할을 수행하는 서버 삭제

하나의 서버가 **CA**(인증 기관) 갱신 서버 역할 및 **CRL**(인증서 폐기 목록) 게시자 역할을 모두 수행할 수 있습니다. 이 서버를 오프라인으로 사용하거나 해제해야 하는 경우 다른 **CA** 서버를 선택하고 해당 역할을 수행하도록 구성합니다.

이 예에서는 **CA** 갱신 서버 및 **CRL** 게시자 역할을 이행하는 호스트 **server.idm.example.com** 을 해제해야 합니다. 이 절차에서는 **CA** 갱신 서버 및 **CRL** 게시자 역할을 호스트 **replica.idm.example.com** 으로 전송하고 **IdM** 환경에서 **server.idm.example.com** 을 제거합니다.



참고

CA 갱신 서버와 **CRL** 게시자 역할을 모두 수행하도록 동일한 서버를 구성할 필요가 없습니다.

사전 요구 사항

- **IdM** 관리자 자격 증명이 있습니다.
- 해제할 서버의 루트 암호가 있습니다.
- **IdM** 환경에 **CA** 복제본이 두 개 이상 있습니다.

절차

1. **IdM** 관리자 인증 정보를 가져옵니다.

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

2. (선택 사항) 어떤 서버가 **CA** 갱신 서버 및 **CRL** 게시자 역할을 수행하는지 확실하지 않은 경우 다음을 수행하십시오.

- a. 현재 **CA** 갱신 서버를 표시합니다. **IdM** 서버에서 다음 명령을 실행할 수 있습니다.

```
[user@server ~]$ ipa config-show | grep 'CA renewal'
IPA CA renewal master: server.idm.example.com
```


b.

호스트가 현재 **CRL** 게시자인지 테스트합니다.

```
[user@server ~]$ ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

CRL을 생성하지 않는 **CA** 서버는 **CRL generation: disabled** 를 표시합니다.

```
[user@replica ~]$ ipa-crlgen-manage status
CRL generation: disabled
The ipa-crlgen-manage command was successful
```

CRL 게시자 서버를 찾을 때까지 **CA** 서버에서 이 명령을 계속 입력합니다.

c.

이러한 역할을 수행하기 위해 승격할 수 있는 다른 모든 **CA** 서버를 표시합니다. 이 환경에는 두 개의 **CA** 서버가 있습니다.

```
[user@server ~]$ ipa server-role-find --role 'CA server'
-----
2 server roles matched
-----
Server name: server.idm.example.com
Role name: CA server
Role status: enabled
Server name: replica.idm.example.com
Role name: CA server
Role status: enabled
-----
Number of entries returned 2
-----
```

3.

replica.idm.example.com 을 **CA** 갱신 서버로 설정합니다.

```
[user@server ~]$ ipa config-mod --ca-renewal-master-server replica.idm.example.com
```

4.

server.idm.example.com 에서 :

a.

인증서 업데이트 작업을 비활성화합니다.

```
[root@server ~]# pki-server ca-config-set ca.certStatusUpdateInterval 0
```

b.

IdM 서비스를 다시 시작하십시오.

```
[user@server ~]$ ipactl restart
```

5.

replica.idm.example.com 에서 :

a.

인증서 업데이트 작업을 활성화합니다.

```
[root@server ~]# pki-server ca-config-unset ca.certStatusUpdateInterval
```

b.

IdM 서비스를 다시 시작하십시오.

```
[user@replica ~]$ ipactl restart
```

6.

server.idm.example.com 에서 CRL 생성을 중지합니다.

```
[user@server ~]$ ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL
generation on another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

7.

replica.idm.example.com 에서 CRL 생성을 시작합니다.

```
[user@replica ~]$ ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
CRL generation enabled on the local host. Please make sure to have only a single CRL
generation master.
The ipa-crlgen-manage command was successful
```

8.

server.idm.example.com 에서 **IdM** 서비스를 중지하십시오:

```
[user@server ~]$ ipactl stop
```

9.

replica.idm.example.com 에서 **IdM** 환경에서 **server.idm.example.com** 을 삭제합니다.

```
[user@replica ~]$ ipa server-del server.idm.example.com
```

10.

server.idm.example.com 에서 **ipa-server-install --uninstall** 명령을 **root** 계정으로 사용합니다.

```
[root@server ~]# ipa-server-install --uninstall
```

```
...
```

```
Are you sure you want to continue with the uninstall procedure? [no]: yes
```

검증 단계

•

현재 **CA** 갱신 서버를 표시합니다.

```
[user@replica ~]$ ipa config-show | grep 'CA renewal'
IPA CA renewal master: replica.idm.example.com
```

•

replica.idm.example.com 호스트에서 **CRL**을 생성하고 있는지 확인합니다.

```
[user@replica ~]$ ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:10:00
Last CRL Number: 7
The ipa-crlgen-manage command was successful
```

추가 리소스

•

[IdM CA 갱신 서버 변경 및 재설정](#)

•

[IdM CA 서버에서 CRL 생성](#)

•

[IdM 복제 설치 제거](#)

62장. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기

62.1. CERTMONGER 개요

certmonger의 기능

IdM(Identity Management)은 통합 **IdM CA(인증 기관)**와 함께 설치되면 **certmonger** 서비스를 사용하여 시스템 및 서비스 인증서를 추적하고 갱신합니다. 인증서가 만료 날짜에 도달하면 **certmonger**는 다음을 통해 갱신 프로세스를 관리합니다.

- 원래 요청에 제공된 옵션을 사용하여 **CSR(인증서 서명 요청)** 다시 생성.
- **IdM API cert-request** 명령을 사용하여 **IdM CA**에 **CSR** 제출.
- **IdM CA**에서 인증서 수신.
- 원래 요청에 지정된 경우 사전 저장 명령 실행.
- 갱신 요청에 지정된 위치에 새 인증서 설치(**NSS 데이터베이스** 또는 **파일에 있음**).
- 원래 요청에 지정된 경우 **post-save** 명령 실행. 예를 들어 **post-save** 명령은 **certmonger**에 관련 서비스를 재시작하도록 지시하여 서비스에서 새 인증서를 선택하도록 할 수 있습니다.

인증서 certmonger 트랙 유형

인증서를 시스템 및 서비스 인증서로 나눌 수 있습니다.

서비스 인증서(예: **HTTP,LDAP** 및 **PKINIT**)와 달리, 서로 다른 서버에 키 쌍과 주체 이름이 다르다면 **IdM** 시스템 인증서 및 해당 키는 모든 **CA** 복제본에서 공유됩니다. **IdM** 시스템 인증서는 다음과 같습니다.

- **IdM CA 인증서**
- **OCSP 서명 인증서**

- **IdM CA 하위 시스템 인증서**
- **IdM CA 감사 서명 인증서**
- **IdM 갱신 에이전트 (RA) 인증서**
- **KRA 전송 및 스토리지 인증서**

certmonger 서비스는 통합된 **CA**를 사용하여 **IdM** 환경을 설치하는 동안 요청된 **IdM** 시스템 및 서비스 인증서를 추적합니다. 또한 **cert monger** 는 **IdM** 호스트에서 실행되는 다른 서비스에 대해 시스템 관리자가 수동으로 요청한 인증서도 추적합니다. **certmonger** 는 외부 **CA** 인증서 또는 사용자 인증서를 추적하지 않습니다.

certmonger 구성 요소

certmonger 서비스는 다음 두 가지 주요 구성 요소로 구성됩니다.

- 인증서 목록을 추적하고 갱신 명령을 시작하는 엔진인 **certmonger** 데몬
- 시스템 관리자가 **certmonger** 데몬에 적극적으로 명령을 보낼 수 있는 **CLI**(명령줄 인터페이스)에 대한 **getcert** 유틸리티입니다.

특히 시스템 관리자는 다음을 위해 **getcert** 유틸리티를 사용할 수 있습니다.

- 새 인증서 요청
- **certmonger** 가 추적하는 인증서 목록 보기
- 인증서 시작 또는 중지
- 인증서 갱신

62.2. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기

IdM(Identity Management) 클라이언트에서 실행되는 브라우저와 웹 서비스 간의 통신이 안전하고 암호화되었는지 확인하려면 **TLS** 인증서를 사용합니다. **IdM CA(인증 기관)**에서 웹 서비스의 **TLS** 인증서를 가져옵니다.

이 섹션에서는 **certmonger** 를 사용하여 **IdM** 클라이언트에서 실행되는 서비스에 대한 **IdM** 인증서 (**HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM**)를 가져오는 방법에 대해 설명합니다.

certmonger 를 사용하여 인증서를 자동으로 요청하면 **certmonger** 가 갱신될 때 인증서를 관리하고 갱신합니다.

certmonger 에서 서비스 인증서를 요청할 때 발생하는 사항을 시각적으로 표시하려면 [62.3절. “서비스 인증서를 요청하는 certmonger의 통신 흐름”](#) 을 참조하십시오.

사전 요구 사항

- 웹 서버가 **IdM** 클라이언트로 등록되었습니다.
- 프로시저를 실행 중인 **IdM** 클라이언트에 대한 루트 액세스 권한이 있습니다.
- 인증서를 요청하는 서비스는 **IdM**에 기존 존재하지 않아도 됩니다.

절차

1. **HTTP** 서비스가 실행 중인 **my_company.idm.example.com** **IdM** 클라이언트에서 **HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM** 주체에 해당하는 서비스에 대한 인증서를 요청하고 다음을 지정합니다.
 - 인증서는 로컬 **/etc/pki/tls/certs/httpd.pem** 파일에 저장해야 합니다.
 - 개인 키는 로컬 **/etc/pki/tls/private/httpd.key** 파일에 저장해야 합니다.
 - **my_company.idm.example.com**의 **DNS** 이름을 사용하여 서명 요청에 **SubjectAltName** 의 **extensionRequest**를 추가합니다.

```
# ipa-getcert request -K HTTP/my_company.idm.example.com -k
/etc/pki/tls/private/httpd.key -f /etc/pki/tls/certs/httpd.pem -g 2048 -D
my_company.idm.example.com -C "systemctl restart httpd"
New signing request "20190604065735" added.
```

위의 명령에서 다음을 수행합니다.

- **ipa-getcert request** 명령은 인증서를 **IdM CA**에서 가져오도록 지정합니다. **ipa-getcert request** 명령은 **getcert request -c IPA** 의 바로 가기입니다.
- **g** 옵션은 아직 배치되지 않은 경우 생성할 키 크기를 지정합니다.
- **D** 옵션은 요청에 추가할 **SubjectAltName DNS** 값을 지정합니다.
- **C** 옵션은 인증서를 가져온 후 **certmonger** 에 **httpd** 서비스를 다시 시작하도록 지시합니다.
- 인증서를 특정 프로필과 함께 발행하도록 지정하려면 **-T** 옵션을 사용합니다.
- 지정된 **CA**에서 명명된 발급자를 사용하여 인증서를 요청하려면 **-X ISSUER** 옵션을 사용합니다.

+



참고

RHEL 8에서는 **RHEL 7**에 사용된 모듈과 **Apache**에서 다른 **SSL** 모듈을 사용합니다. **SSL** 모듈은 **NSS** 대신 **OpenSSL**을 사용합니다. 따라서 **RHEL 8**에서는 **NSS** 데이터베이스를 사용하여 **HTTPS** 인증서 및 개인 키를 저장할 수 없습니다.

+ . 선택적으로 요청 상태를 확인하려면 다음을 수행합니다.

+

```
# ipa-getcert list -f /etc/pki/tls/certs/httpd.pem
Number of certificates and requests being tracked: 3.
Request ID '20190604065735':
  status: MONITORING
  stuck: no
  key pair storage: type=FILE,location='/etc/pki/tls/private/httpd.key'
  certificate: type=FILE,location='/etc/pki/tls/certs/httpd.crt'
  CA: IPA
[...]
```

+ 출력은 요청이 **MONITORING** 상태에 있음을 보여주며, 이는 인증서를 가져왔음을 의미합니다. 키 쌍과 인증서의 위치는 요청된 값입니다.

62.3. 서비스 인증서를 요청하는 **CERTMONGER**의 통신 흐름

이 섹션의 다이어그램은 **certmonger** 에서 **IdM(Identity Management) CA**(인증 기관) 서버에서 서비스 인증서를 요청할 때 발생하는 작업의 단계를 보여줍니다. 시퀀스는 다음 다이어그램으로 구성됩니다.

- [그림 62.1. “암호화되지 않은 통신”](#)
- [그림 62.2. “서비스 인증서 요청 **certmonger**”](#)
- [그림 62.3. “IdM CA에서 서비스 인증서 발행”](#)
- [그림 62.4. “서비스 인증서 적용 **certmonger**”](#)
- [그림 62.5. “이전 인증서가 만료될 때 새 인증서를 요청하는 **certmonger**”](#)

[그림 62.1. “암호화되지 않은 통신”](#) 는 초기 상황을 보여줍니다. **HTTPS** 인증서가 없으면 웹 서버와 브라우저 간의 통신이 암호화되지 않습니다.

그림 62.1. 암호화되지 않은 통신

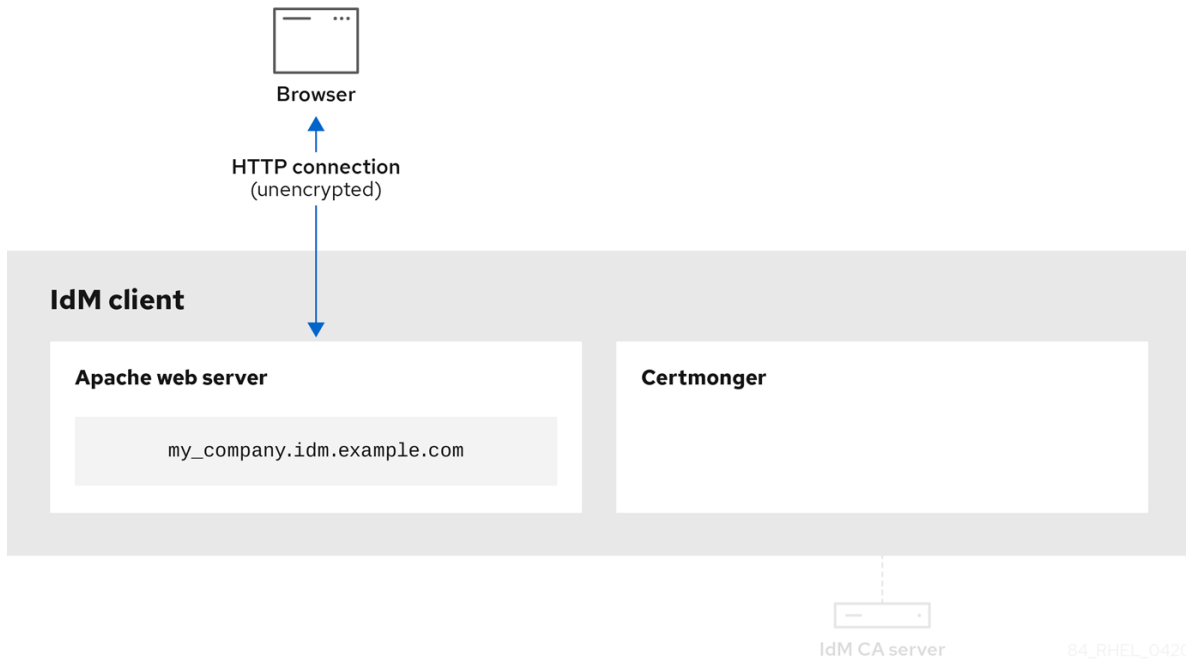


그림 62.2. “서비스 인증서 요청 certmonger” certmonger 를 사용하여 Apache 웹 서버에 대한 HTTPS 인증서를 수동으로 요청하는 시스템 관리자를 보여줍니다. 웹 서버 인증서를 요청할 때 certmonger는 CA와 직접 통신하지 않습니다. IdM을 통해 프록시됩니다.

그림 62.2. 서비스 인증서 요청 certmonger

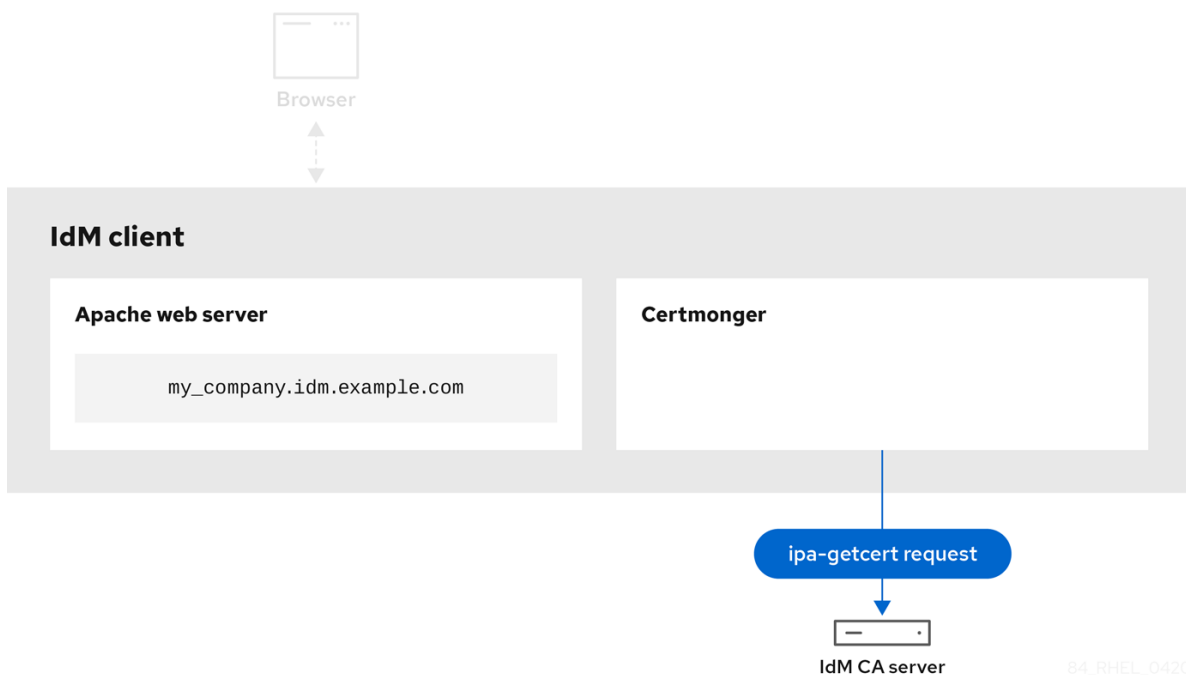


그림 62.3. “IdM CA에서 서비스 인증서 발행”에는 웹 서버의 **HTTPS** 인증서를 발급하는 **IdM CA**가 표시됩니다.

그림 62.3. IdM CA에서 서비스 인증서 발행

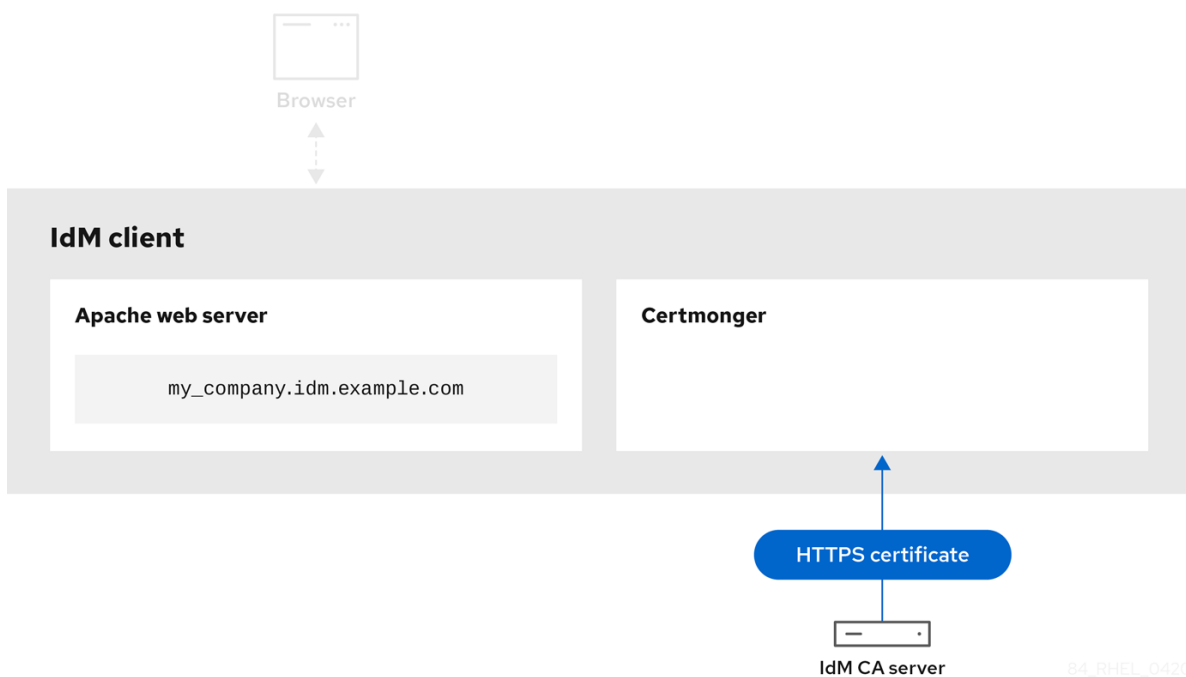


그림 62.4. “서비스 인증서 적용 certmonger”는 **HTTPS** 인증서를 **IdM** 클라이언트의 적절한 위치에 배치한 **certmonger**를 보여줍니다. 이렇게 지시한 경우 **httpd** 서비스를 다시 시작합니다. 결과적으로 **Apache** 서버는 **HTTPS** 인증서를 사용하여 자체와 브라우저 간의 트래픽을 암호화합니다.

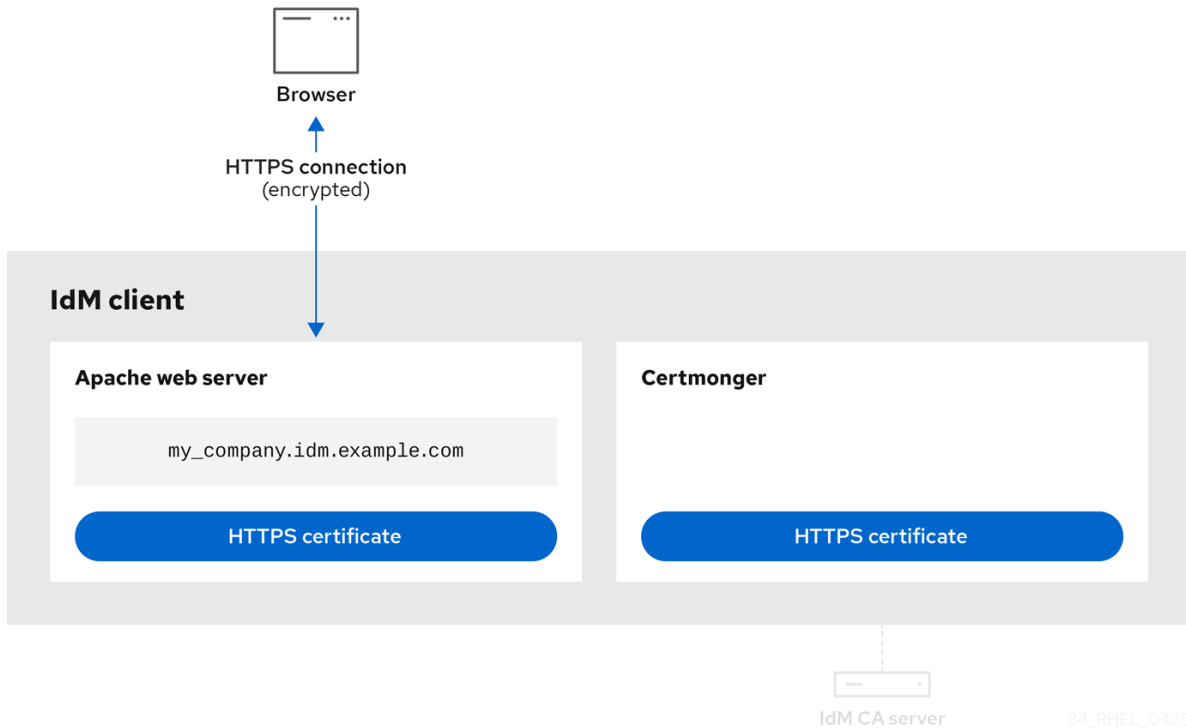
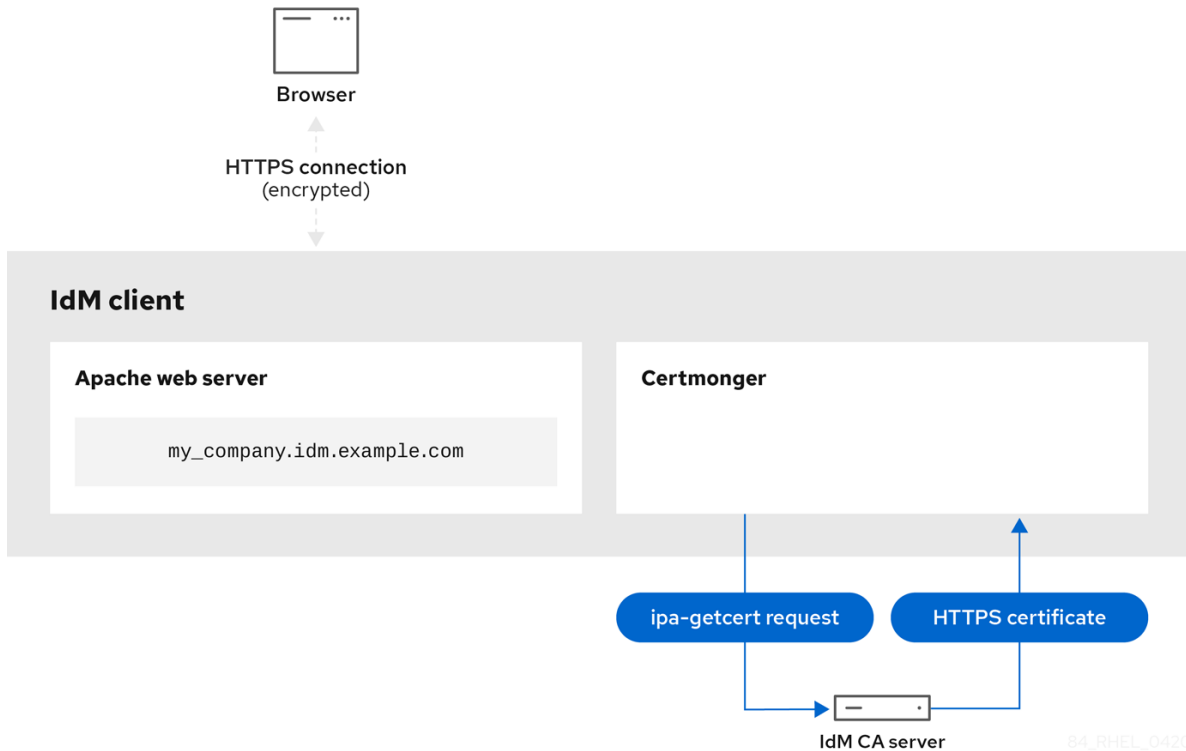
그림 62.4. 서비스 인증서 적용 **certmonger**

그림 62.5. “이전 인증서가 만료될 때 새 인증서를 요청하는 **certmonger**” **certmonger**에서 인증서 만료 전에 **IdM CA**에서 서비스 인증서 갱신을 자동으로 요청하는 방법을 보여줍니다. **IdM CA**는 새 인증서를 발행합니다.

그림 62.5. 이전 인증서가 만료될 때 새 인증서를 요청하는 **certmonger**

84_RHEL_0420

62.4. CERTMONGER에서 추적한 인증서 요청 세부 정보 보기

certmonger 서비스는 인증서 요청을 모니터링합니다. 인증서 요청이 성공적으로 서명되면 인증서가 생성됩니다. **certmonger**는 결과 인증서를 포함하여 인증서 요청을 관리합니다. 이 섹션에서는 **certmonger**에서 관리하는 특정 인증서 요청의 세부 정보를 확인하는 방법을 설명합니다.

절차

- 인증서 요청을 지정하는 방법을 알고 있는 경우 해당 특정 인증서 요청의 세부 정보만 나열합니다. 예를 들어 다음을 지정할 수 있습니다.
 - 요청 ID
 - 인증서 위치
 - 인증서 닉네임

예를 들어 인증서 요청이 성공하지 못한 경우 **-v** 옵션을 사용하여 요청 ID가 **20190408143846**인 인증서의 세부 정보를 보려면 **-v** 옵션을 사용합니다.

```
# getcert list -i 20190408143846 -v
Number of certificates and requests being tracked: 16.
Request ID '20190408143846':
status: MONITORING
stuck: no
key pair storage: type=NSSDB,location='/etc/dirsrv/slapped-IDM-EXAMPLE-
COM',nickname='Server-Cert',token='NSS Certificate DB',pinfile='/etc/dirsrv/slapped-
IDM-EXAMPLE-COM/pwdfile.txt'
certificate: type=NSSDB,location='/etc/dirsrv/slapped-IDM-EXAMPLE-
COM',nickname='Server-Cert',token='NSS Certificate DB'
CA: IPA
issuer: CN=Certificate Authority,O=IDM.EXAMPLE.COM
subject: CN=r8server.idm.example.com,O=IDM.EXAMPLE.COM
expires: 2021-04-08 16:38:47 CEST
dns: r8server.idm.example.com
principal name: ldap/server.idm.example.com@IDM.EXAMPLE.COM
key usage: digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment
eku: id-kp-serverAuth,id-kp-clientAuth
pre-save command:
post-save command: /usr/libexec/ipa/certmonger/restart_dirsrv IDM-EXAMPLE-
COM
track: yes
auto-renew: yes
```

출력에는 인증서에 대한 여러 정보가 표시됩니다. 예를 들면 다음과 같습니다.

- 인증서 위치: 위의 예에서 **/etc/dirsrv/slapped-IDM-EXAMPLE-COM** 디렉터리에 있는 **NSS** 데이터베이스입니다.
- 인증서 닉네임; 위의 예에서 이는 **Server-Cert**입니다.
- 위의 예에서 고정을 저장하는 파일은 **/etc/dirsrv/slapped-IDM-EXAMPLE-COM/pwdfile.txt**입니다.
- 인증서를 갱신하는 데 사용할 **CA**(인증 기관)입니다. 위의 예제에서는 **IPA CA**입니다.
- 만료 날짜; 위의 예에서는 **2021-04-08 16:38:47 CEST**입니다.
-

인증서의 상태입니다. 위의 예에서 **MONITORING** 상태는 인증서가 유효하며 추적 중임을 나타냅니다.

○

post-save 명령; 위의 예에서 **LDAP** 서비스를 다시 시작합니다.

●

인증서 요청을 지정하는 방법을 모르는 경우 **certmonger**가 모니터링하거나 가져오려는 모든 인증서의 세부 정보를 나열합니다.

getcert list

추가 정보

●

표시된 인증서 요청을 지정하는 방법을 다른 옵션을 보려면 **getcert list** 도움말 페이지를 참조하십시오.

62.5. 인증서 추적 시작 및 중지

이 섹션에서는 **getcert stop-tracking** 및 **getcert start-tracking** 명령을 사용하여 인증서를 모니터링하는 방법에 대해 설명합니다. 두 가지 명령은 **certmonger** 서비스에서 제공합니다. 인증서 추적 활성화는 **IdM(Identity Management)** 인증 기관(CA)에서 발급한 인증서를 다른 **IdM** 클라이언트에서 시스템으로 가져온 경우 특히 유용합니다. 인증서 추적을 활성화하는 것이 다음 프로비저닝 시나리오의 최종 단계일 수도 있습니다.

1.

IdM 서버에서 아직 존재하지 않는 시스템의 인증서를 생성합니다.

2.

새 시스템을 만듭니다.

3.

새 시스템을 **IdM** 클라이언트로 등록합니다.

4.

인증서와 키를 **IdM** 서버에서 **IdM** 클라이언트로 가져옵니다.

5.

만료될 때 갱신되는지 확인하기 위해 **certmonger**를 사용하여 인증서 추적을 시작합니다.

절차

•

Request ID가 20190408143846인 인증서 모니터링을 비활성화하려면 다음을 수행합니다.

```
# getcert stop-tracking -i 20190408143846
```

자세한 옵션은 **getcert stop-tracking** 도움말 페이지를 참조하십시오.

•

/tmp/some_cert.crt 파일에 저장된 인증서 모니터링을 활성화하려면 개인 키가 /tmp/some_key.key 파일에 저장됩니다.

```
# getcert start-tracking -c IPA -f /tmp/some_cert.crt -k /tmp/some_key.key
```

certmonger는 인증서를 발급한 **CA** 유형을 자동으로 식별할 수 없습니다. 이러한 이유로 **IdM CA**에서 인증서를 발급한 경우 **IPA** 값과 함께 **-c** 옵션을 **getcert start-tracking** 명령에 추가합니다. **c** 옵션을 추가하는 것을 생략하면 인증서가 **NEED_CA** 상태가 됩니다.

자세한 옵션은 **getcert start-tracking** 도움말 페이지를 참조하십시오.



참고

두 명령은 인증서를 조작하지 않습니다. 예를 들어 **getcert stop-tracking**은 인증서를 삭제하지 않거나 **NSS** 데이터베이스 또는 파일 시스템에서 제거하지 않고 모니터링된 인증서 목록에서 인증서를 제거합니다. 마찬가지로 **getcert start-tracking**은 모니터링된 인증서 목록에만 인증서를 추가합니다.

62.6. 인증서 수동 갱신

인증서가 만료 날짜에 가까운 경우 **certmonger** 데몬은 **CA**(인증 기관) 도우미를 사용하여 갱신 명령을 자동으로 실행하고 갱신된 인증서를 가져오고 이전 인증서를 새 인증서로 대체합니다.

getcert resubmit 명령을 사용하여 미리 인증서를 수동으로 갱신할 수도 있습니다. 이렇게 하면 인증서에 포함된 정보를 업데이트할 수 있습니다(예: **SAN**(주체 대체 이름)).

이 섹션에서는 인증서를 수동으로 갱신하는 방법에 대해 설명합니다.

절차

•

20190408143846 요청 ID가 있는 인증서를 갱신하려면 다음을 수행합니다.

```
# getcert resubmit -i 20190408143846
```

특정 인증서에 대한 요청 ID를 가져오려면 **getcert list** 명령을 사용합니다. 자세한 내용은 **getcert list** 도움말 페이지를 참조하십시오.

62.7. CERTMONGER에서 CA 복제본에서 IDM 인증서 추적을 다시 시작

다음 절차에서는 인증서 추적이 중단된 후 통합 인증 기관을 사용하여 IdM 배포에 중요한 IdM(Identity Management) 시스템 인증서 추적을 다시 시작하는 방법을 보여줍니다. 시스템 인증서 갱신 중에 IdM 호스트가 IdM에서 등록되지 않았거나 복제 토폴로지가 제대로 작동하지 않아 중단되었을 수 있습니다. 이 절차에서는 **certmonger**, 즉 **HTTP,LDAP,PKINIT** 인증서라는 IdM 서비스 인증서 추적을 다시 시작하는 방법도 표시합니다.

사전 요구 사항

•

시스템 인증서 추적을 다시 시작할 호스트는 IdM CA 갱신 서버가 아닌 IdM CA(인증 기관)이기도 합니다.

절차

1.

하위 시스템 CA 인증서에 대한 PIN을 가져옵니다.

```
# grep 'internal=' /var/lib/pki/pki-tomcat/conf/password.conf
```

2.

하위 시스템 CA 인증서에 추적을 추가하고 아래 명령에서 [내부 PIN] 을 이전 단계에서 얻은 PIN으로 바꿉니다.

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "caSigningCert cert-pki-ca" -c  
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B  
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert  
"caSigningCert cert-pki-ca"'
```

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "auditSigningCert cert-pki-ca" -c  
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B  
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert  
"auditSigningCert cert-pki-ca"'
```

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "ocspSigningCert cert-pki-ca" -c  
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B  
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
```



```
"ocspSigningCert cert-pki-ca"
```

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "subsystemCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"subsystemCert cert-pki-ca"
```

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "Server-Cert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"Server-Cert cert-pki-ca"
```

3.

나머지 **IdM** 인증서, **HTTP,LDAP,IPA** 갱신 에이전트 및 **PKINIT** 인증서에 대한 추적을 추가합니다.

```
# getcert start-tracking -f /var/lib/ipa/certs/httpd.crt -k /var/lib/ipa/private/httpd.key -p
/var/lib/ipa/passwds/idm.example.com-443-RSA -c IPA -C
/usr/libexec/ipa/certmonger/restart_httpd
```

```
# getcert start-tracking -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM -n "Server-Cert" -c IPA
-p /etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt -C
'/usr/libexec/ipa/certmonger/restart_dirsrv "IDM-EXAMPLE-COM"
```

```
# getcert start-tracking -f /var/lib/ipa/ra-agent.pem -k /var/lib/ipa/ra-agent.key -c
dogtag-ipa-ca-renew-agent -B /usr/libexec/ipa/certmonger/renew_ra_cert_pre -C
/usr/libexec/ipa/certmonger/renew_ra_cert
```

```
# getcert start-tracking -f /var/kerberos/krb5kdc/kdc.crt -k
/var/kerberos/krb5kdc/kdc.key -c dogtag-ipa-ca-renew-agent -B
/usr/libexec/ipa/certmonger/renew_ra_cert_pre -C
/usr/libexec/ipa/certmonger/renew_kdc_cert
```

4.

certmonger 를 다시 시작하십시오.

```
# systemctl restart certmonger
```

5.

certmonger 가 시작된 후 1분간 기다린 다음 새 인증서의 상태를 확인합니다.

```
# getcert list
```

추가 리소스

•

IdM 시스템 인증서가 모두 만료된 경우 이 [KCS\(지식 센터 지원\) 솔루션](#)에 설명된 절차를 수행하여 **CA** 갱신 서버 및 **CRL** 게시자 서버인 **IdM CA** 서버에서 **IdM** 시스템 인증서를 수동으로 갱신합니다. 그런 다음 이 [KCS 솔루션](#)에 설명된 절차를 수행하여 토폴로지의 다른 모든 **CA** 서버에서 **IdM** 시스템 인증서를 수동으로 갱신합니다.

63장. RHEL 시스템 역할을 사용하여 인증서 요청

Certificate System Role을 사용하면 **Red Hat Ansible Engine**을 사용하여 인증서를 발급하고 관리할 수 있습니다.

이 장에서는 다음 주제를 다룹니다.

- [인증서 시스템 역할](#)
- [인증서 시스템 역할을 사용하여 자체 서명된 새 인증서 요청](#)
- [인증서 시스템 역할을 사용하여 IdM CA에서 새 인증서 요청](#)

63.1. 인증서 시스템 역할

Certificate System Role을 사용하면 **Red Hat Ansible Engine**을 사용하여 **TLS** 및 **SSL** 인증서 발급 및 갱신을 관리할 수 있습니다.

이 역할은 인증서 프로바이더로 **certmonger**를 사용하며 현재 자체 서명된 인증서 및 **IdM** 통합 **CA**(인증 기관)를 사용하여 갱신을 지원합니다.

Certificate System 역할과 함께 **Ansible** 플레이북에서 다음 변수를 사용할 수 있습니다.

certificate_wait

작업이 인증서가 발행될 때까지 기다려야 하는지를 지정합니다.

certificate_requests

발급할 각 인증서와 해당 매개 변수를 나타냅니다.

추가 리소스

- **certificate_requests** 변수에 사용되는 매개 변수 및 인증서 시스템 역할에 대한 추가 정보는 `/usr/share/ansible/roles/rhel-system-roles.certificate/README.md` 파일을 참조하십시오.

- **RHEL 시스템 역할 및 적용 방법**에 대한 자세한 내용은 [RHEL 시스템 역할 시작하기](#)를 참조하십시오.

63.2. 인증서 시스템 역할을 사용하여 자체 서명된 새 인증서 요청

Certificate System Role을 사용하면 **Red Hat Ansible Engine**을 사용하여 자체 서명 인증서를 발급할 수 있습니다.

이 프로세스는 **certmonger** 프로바이더를 사용하고 **getcert** 명령을 통해 인증서를 요청합니다.



참고

기본적으로 **certmonger**는 만료되기 전에 자동으로 인증서를 갱신하려고 합니다. **Ansible** 플레이북에서 **auto_renew** 매개 변수를 **no**로 설정하여 이 설정을 비활성화할 수 있습니다.

사전 요구 사항

- 플레이북을 실행할 시스템에 **Red Hat Ansible Engine**이 설치되어 있어야 합니다.



참고

인증서 솔루션을 배포하려는 시스템에 **Ansible**을 설치할 필요는 없습니다.

- 플레이북을 실행할 시스템에 **rhel-system-roles** 패키지가 설치되어 있습니다.

RHEL 시스템 역할 및 적용 방법에 대한 자세한 내용은 [RHEL 시스템 역할 시작하기](#)를 참조하십시오.

절차

1. 선택 사항: 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2.

인벤토리 파일을 열고 인증서를 요청할 호스트를 정의합니다. 예를 들면 다음과 같습니다.

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

3.

플레이북 파일을 생성합니다(예: `request-certificate.yml`):

- 인증서를 요청하려는 호스트를 포함 하도록 호스트를 설정합니다(예: `webserver`).
- 다음을 포함하도록 `certificate_requests` 변수를 설정합니다.
 - `name` 매개 변수를 `mycert` 와 같이 원하는 인증서 이름으로 설정합니다.
 - `dns` 매개 변수를 인증서에 포함할 도메인(예: `*.example.com`)으로 설정합니다.
 - `ca` 매개 변수를 `self-sign` 으로 설정합니다.
- 역할에서 `rhel-system-roles.certificate` 역할을 설정합니다.

다음은 이 예제의 플레이북 파일입니다.

```
---
- hosts: webserver

vars:
  certificate_requests:
    - name: mycert
      dns: "*.example.com"
      ca: self-sign

roles:
  - rhel-system-roles.certificate
```

4.

파일을 저장합니다.

5.

플레이북을 실행합니다.

```
$ ansible-playbook -i inventory.file request-certificate.yml
```

추가 리소스

- **certificate_requests** 변수에 사용되는 매개변수 및 인증서 시스템 역할에 대한 추가 정보는 `/usr/share/ansible/roles/rhel-system-roles.certificate/README.md` 파일을 참조하십시오.
- **ansible-playbook** 명령에 대한 자세한 내용은 **ansible-playbook(1)** 도움말 페이지를 참조하십시오.

63.3. 인증서 시스템 역할을 사용하여 IDM CA에서 새 인증서 요청

Certificate System Role을 사용하면 **Red Hat Ansible Engine**을 사용하여 **CA**(인증 기관)에서 **IdM** 서버를 사용하는 동안 인증서를 발행할 수 있습니다. 따라서 **IdM**을 **CA**로 사용할 때 여러 시스템의 인증서 신뢰 체인을 효율적이고 일관되게 관리할 수 있습니다.

이 프로세스는 **certmonger** 프로바이더를 사용하고 **getcert** 명령을 통해 인증서를 요청합니다.

참고

기본적으로 **certmonger**는 만료되기 전에 자동으로 인증서를 갱신하려고 합니다. **Ansible** 플레이북에서 **auto_renew** 매개 변수를 **no**로 설정하여 이 설정을 비활성화할 수 있습니다.

사전 요구 사항

- 플레이북을 실행할 시스템에 **Red Hat Ansible Engine**이 설치되어 있어야 합니다.

참고

인증서 솔루션을 배포하려는 시스템에 **Ansible**을 설치할 필요는 없습니다.

- 플레이북을 실행할 시스템에 **rhel-system-roles** 패키지가 설치되어 있습니다.

RHEL 시스템 역할 및 적용 방법에 대한 자세한 내용은 [RHEL 시스템 역할 시작하기](#)를 참조하십시오.

절차

1.

선택 사항: 인벤토리 파일을 생성합니다(예: `inventory.file`):

```
$ touch inventory.file
```

2.

인벤토리 파일을 열고 인증서를 요청할 호스트를 정의합니다. 예를 들면 다음과 같습니다.

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

3.

플레이북 파일을 생성합니다(예: `request-certificate.yml`):

- 인증서를 요청하려는 호스트를 포함 하도록 호스트를 설정합니다(예: `webserver`).
- 다음을 포함하도록 `certificate_requests` 변수를 설정합니다.
 - `name` 매개 변수를 `mycert` 와 같이 원하는 인증서 이름으로 설정합니다.
 - `dns` 매개 변수를 인증서에 포함할 도메인(예: `www.example.com`) 으로 설정합니다.
 - `principal` 매개 변수를 설정하여 **Kerberos** 주체(예: `HTTP/www.example.com@EXAMPLE.COM`)를 지정합니다.
 - `ca` 매개 변수를 `ipa` 로 설정합니다.
- 역할에서 `rhel-system-roles.certificate` 역할을 설정합니다.

다음은 이 예제의 플레이북 파일입니다.

```

---
- hosts: webserver
  vars:
    certificate_requests:
      - name: mycert
        dns: www.example.com
        principal: HTTP/www.example.com@EXAMPLE.COM
        ca: ipa

  roles:
    - rhel-system-roles.certificate

```

4. 파일을 저장합니다.

5. 플레이북을 실행합니다.

```
$ ansible-playbook -i inventory.file request-certificate.yml
```

추가 리소스

- **certificate_requests** 변수에 사용되는 매개변수 및 인증서 시스템 역할에 대한 추가 정보는 `/usr/share/ansible/roles/rhel-system-roles.certificate/README.md` 파일을 참조하십시오.
- **ansible-playbook** 명령에 대한 자세한 내용은 **ansible-playbook(1)** 도움말 페이지를 참조하십시오.

63.4. 인증서 시스템 역할을 사용하여 인증서 발급 전후에 실행할 명령 지정

Certificate System Role을 사용하면 인증서가 발행되거나 갱신되기 전후에 **Red Hat Ansible Engine**을 사용하여 명령을 실행할 수 있습니다.

다음 예에서 관리자는 **www.example.com**의 자체 서명된 인증서가 발급되거나 갱신되기 전에 **httpd** 서비스를 중지하고 나중에 다시 시작합니다.



참고

기본적으로 **certmonger**는 만료되기 전에 자동으로 인증서를 갱신하려고 합니다. **Ansible** 플레이북에서 **auto_renew** 매개 변수를 **no**로 설정하여 이 설정을 비활성화할 수 있습니다.

사전 요구 사항

- 플레이북을 실행할 시스템에 **Red Hat Ansible Engine**이 설치되어 있어야 합니다.



참고

인증서 솔루션을 배포하려는 시스템에 **Ansible**을 설치할 필요는 없습니다.

- 플레이북을 실행할 시스템에 **rhel-system-roles** 패키지가 설치되어 있습니다.

RHEL 시스템 역할 및 적용 방법에 대한 자세한 내용은 [RHEL 시스템 역할 시작하기](#)를 참조하십시오.

절차

1. 선택 사항: 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2. 인벤토리 파일을 열고 인증서를 요청할 호스트를 정의합니다. 예를 들면 다음과 같습니다.

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

3. 플레이북 파일을 생성합니다(예: **request-certificate.yml**):

- 인증서를 요청하려는 호스트를 포함 하도록 호스트를 설정합니다(예: **webserver**).

- 다음을 포함하도록 **certificate_requests** 변수를 설정합니다.

- **name** 매개 변수를 **mycert** 와 같이 원하는 인증서 이름으로 설정합니다.

- **dns** 매개 변수를 인증서에 포함할 도메인(예: **www.example.com**) 으로 설정합니다.

- 인증서를 발급하는 데 사용할 **ca** 매개 변수(예: 자체 서명)를 설정합니다.
- **run_before** 매개 변수를 **systemctl stop httpd.service** 와 같이 인증서가 발행되거나 갱신되기 전에 실행할 명령으로 설정합니다.
- 이 인증서가 발급되거나 갱신된 후 **systemctl start httpd.service** 와 같이 **run_after** 매개 변수를 실행할 명령으로 설정합니다.
- 역할에서 **rhel-system-roles.certificate** 역할을 설정합니다.

다음은 이 예제의 플레이북 파일입니다.

```
---
- hosts: webserver
  vars:
    certificate_requests:
      - name: mycert
        dns: www.example.com
        ca: self-sign
        run_before: systemctl stop httpd.service
        run_after: systemctl start httpd.service

  roles:
    - rhel-system-roles.certificate
```

4. 파일을 저장합니다.

5. 플레이북을 실행합니다.

```
$ ansible-playbook -i inventory.file request-certificate.yml
```

추가 리소스

- **certificate_requests** 변수에 사용되는 매개 변수 및 인증서 시스템 역할에 대한 추가 정보는 **/usr/share/ansible/roles/rhel-system-roles.certificate/README.md** 파일을 참조하십시오.
- **ansible-playbook** 명령에 대한 자세한 내용은 **ansible-playbook(1)** 도움말 페이지를 참조하십시오.

64장. 인증서의 하위 집합만 신뢰하도록 애플리케이션 제한

IdM(Identity Management) 설치기 **CSV(통합 인증서 시스템)** 인증 기관(**CA**)으로 구성된 경우 경량 하위 **CA**를 생성할 수 있습니다. 생성하는 모든 하위 **CA**는 인증서 시스템의 기본 **CA**인 **ipa CA**로 하위 설정됩니다.

이 컨텍스트의 경량 하위 **CA**는 특정 목적을 위해 인증서를 발급하는 하위 **CA**를 의미합니다. 예를 들어, 경량 하위 **CA**를 사용하면 가상 사설 네트워크(**VPN**) 게이트웨이 및 웹 브라우저와 같은 서비스를 구성하여 하위 **CA A**에서 발급한 인증서만 허용할 수 있습니다. 하위 **CA B**에서만 발급한 인증서를 수락하도록 다른 서비스를 구성하면 하위 **CA**, 기본 **CA**, **ipa CA**인 모든 중간 하위 **CA**에서 발급한 인증서를 수락하지 못하도록 합니다.

하위 **CA**의 중간 인증서를 취소하면 올바르게 구성된 클라이언트가 이 하위 **CA**에서 발급한 모든 인증서가 유효하지 않은 것으로 간주됩니다. 루트 **CA**, **ipa** 또는 다른 하위 **CA**에서 직접 발급한 기타 모든 인증서는 유효한 상태로 유지됩니다.

이 섹션에서는 **Apache** 웹 서버의 예를 사용하여 인증서의 하위 집합만 신뢰하도록 애플리케이션을 제한하는 방법을 설명합니다. 이 섹션을 완료하여 **IdM** 클라이언트에서 실행 중인 웹 서버가 **webserver-ca IdM** 하위 **CA**에서 발급한 인증서를 사용하도록 제한하고, 사용자가 **webclient-ca IdM** 하위 **CA**에서 발급한 사용자 인증서를 사용하여 웹 서버를 인증해야 합니다.

필요한 단계는 다음과 같습니다.

1. **IdM 하위 CA 생성**
2. **IdM WebUI에서 하위 CA 인증서 다운로드**
3. **사용된 사용자, 서비스 및 CA의 올바른 조합을 지정하는 CA ACL 생성**
4. **IdM 하위 CA에서 IdM 클라이언트에서 실행 중인 웹 서비스의 인증서 요청**
5. **단일 인스턴스 Apache HTTP 서버 설정**
6. **Apache HTTP 서버에 TLS 암호화 추가**

7.

Apache HTTP 서버에서 지원되는 TLS 프로토콜 버전 설정

8.

Apache HTTP 서버에서 지원되는 암호를 설정합니다.

9.

웹 서버에서 TLS 클라이언트 인증서 인증 구성

10.

IdM 하위 CA에서 사용자의 인증서를 요청하고 클라이언트로 내보냅니다.

11.

사용자 인증서를 브라우저로 가져오고 하위 CA 인증서를 신뢰하도록 브라우저를 구성합니다.

64.1. 경량 하위 CA 관리

이 섹션에서는 경량 하위 인증 기관(하위 CA)을 관리하는 방법을 설명합니다. 생성하는 모든 하위 CA는 인증서 시스템의 기본 CA인 ipa CA로 하위됩니다. 또한 하위 CA를 비활성화하고 삭제할 수 있습니다.

참고

- 하위 CA를 삭제하면 해당 하위 CA에 대한 폐기 검사가 더 이상 작동하지 않습니다. 나중에 만료 시간이 아닌 하위 CA에서 발급한 인증서가 더 이상 없는 경우에만 하위 CA를 삭제합니다.
- 해당 하위 CA에서 발급한 만료되지 않은 인증서가 있는 동안 하위 CA만 비활성화해야 합니다. 하위 CA에서 발급한 모든 인증서가 만료된 경우 해당 하위 CA를 삭제할 수 있습니다.
- IdM CA를 비활성화하거나 삭제할 수 없습니다.

하위 CA 관리에 대한 자세한 내용은 다음을 참조하십시오.

•

64.1.1절. “IdM 웹UI에서 하위 CA 생성”

- **64.1.2절. “IdM 웹UI에서 하위 CA 삭제”**
- **64.1.3절. “IdM CLI에서 하위 CA 생성”**
- **64.1.4절. “IdM CLI에서 하위 CA 비활성화”**
- **64.1.5절. “IdM CLI에서 하위 CA 삭제”**

64.1.1. IdM 웹UI에서 하위 CA 생성

이 절차에서는 IdM 웹 UI를 사용하여 **webserver-ca** 및 **webclient-ca** 라는 새 하위 CA를 생성하는 방법을 설명합니다.

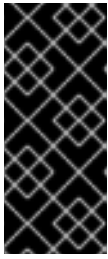
사전 요구 사항

- 관리자 자격 증명을 획득했는지 확인합니다.

절차

1. **Authentication(인증) 메뉴에서 Certificates(인증)를 클릭합니다.**
2. **Certificate Authorities(인증 기관)를 선택하고 Add(추가)를 클릭합니다.**
3. **webserver-ca 하위 CA의 이름을 입력합니다. 주체 DN(예: CN=WEBSERVER,O=IDM.EXAMPLE.COM)을 Subject DN(주체 DN) 필드에 입력합니다. 주체 DN은 IdM CA 인프라에서 고유해야 합니다.**
4. **webclient-ca 하위 CA의 이름을 입력합니다. 주체 DN CN=WEBCIENT,O=IDM.EXAMPLE.COM 을 주체 DN 필드에 입력합니다.**
5. **명령줄 인터페이스에서 ipa-certupdate 명령을 실행하여 webserver-ca 및 webclient-ca 하위 CA 인증서에 대한 certmonger 추적 요청을 생성합니다.**

```
[root@ipaserver ~]# ipa-certupdate
```



중요

sub-CA를 만든 후 **ipa-certupdate** 명령을 실행하는 것을 잊어버리면 하위 **CA** 인증서가 만료되면 엔드 엔티티 인증서가 만료되지 않았더라도 하위 **CA**에서 발급한 엔드 엔티티 인증서가 유효하지 않은 것으로 간주됩니다.

검증

- 새 하위 **CA**의 서명 인증서가 **IdM** 데이터베이스에 추가되었는지 확인합니다.

```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L
```

Certificate Nickname	Trust Attributes
	SSL,S/MIME,JAR/XPI
caSigningCert cert-pki-ca	CTu,Cu,Cu
Server-Cert cert-pki-ca	u,u,u
auditSigningCert cert-pki-ca	u,u,Pu
caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc	u,u,u
ocspSigningCert cert-pki-ca	u,u,u
subsystemCert cert-pki-ca	u,u,u



참고

새 하위 **CA** 인증서는 인증서 시스템 인스턴스가 설치된 모든 복제본으로 자동으로 전송됩니다.

64.1.2. IdM 웹UI에서 하위 CA 삭제

이 절차에서는 **IdM** 웹 UI에서 경량 하위 **CA**를 삭제하는 방법을 설명합니다.

참고

- 하위 CA를 삭제하면 해당 하위 CA에 대한 폐기 검사가 더 이상 작동하지 않습니다. 나중에 만료 시간이 아닌 하위 CA에서 발급한 인증서가 더 이상 없는 경우에만 하위 CA를 삭제합니다.
- 해당 하위 CA에서 발급한 만료되지 않은 인증서가 있는 동안 하위 CA만 비활성화해야 합니다. 하위 CA에서 발급한 모든 인증서가 만료된 경우 해당 하위 CA를 삭제할 수 있습니다.
- IdM CA를 비활성화하거나 삭제할 수 없습니다.

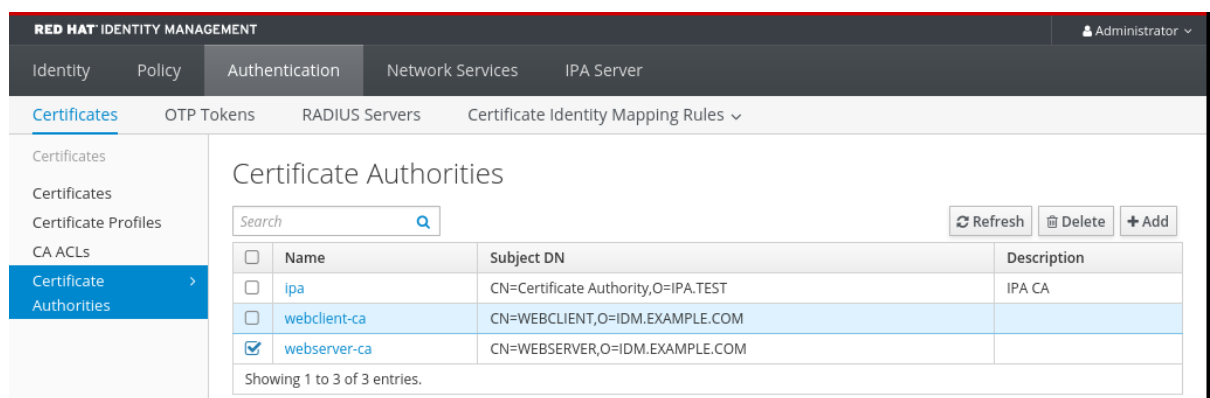
사전 요구 사항

- 관리자 자격 증명을 획득했는지 확인합니다.
- IdM CLI에서 하위 CA를 비활성화했습니다. [IdM CLI에서 하위 CA 비활성화](#)를 참조하십시오.

절차

1. IdM 웹 UI에서 **Authentication(인증)** 탭을 열고 **Certificates(인증)** 하위 탭을 선택합니다.
2. 인증 기관을 선택합니다.
3. 제거할 하위 CA를 선택하고 **Delete(삭제)**를 클릭합니다.

그림 64.1. IdM 웹 UI에서 하위 CA 삭제



4.

Delete(삭제)를 클릭하여 확인합니다.

하위 **CA**는 인증 기관 목록에서 제거됩니다.

64.1.3. IdM CLI에서 하위 CA 생성

이 절차에서는 **IdM CLI**를 사용하여 **webserver-ca** 및 **webclient-ca** 라는 새 하위 **CA**를 생성하는 방법을 설명합니다.

사전 요구 사항

- 관리자의 자격 증명을 획득했는지 확인합니다.
- **CA** 서버인 **IdM** 서버에 로그인했는지 확인합니다.

절차

1. **ipa ca-add** 명령을 입력하고 **webserver-ca** 하위 **CA**의 이름과 주체 고유 이름(**DN**)을 지정합니다.

```
[root@ipaserver ~]# ipa ca-add webserver-ca --
subject="CN=WEBSERVER,O=IDM.EXAMPLE.COM"
-----
Created CA "webserver-ca"
-----
Name: webserver-ca
Authority ID: ba83f324-5e50-4114-b109-acca05d6f1dc
Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

이름

CA의 이름입니다.

권한 **ID**

CA의 개별 **ID**가 자동으로 생성됩니다.

제목 **DN**

주체 이름(DN). 주체 DN은 IdM CA 인프라에서 고유해야 합니다.

발행자 DN

하위 CA 인증서를 발급한 상위 CA입니다. 모든 하위 CA는 IdM 루트 CA의 하위로 생성됩니다.

2.

웹 클라이언트에 인증서를 발급하기 위해 **webclient-ca** 하위 CA를 생성합니다.

```
[root@ipaserver ~]# ipa ca-add webclient-ca --
subject="CN=WEBCIENT,O=IDM.EXAMPLE.COM"
-----
Created CA "webclient-ca"
-----
Name: webclient-ca
Authority ID: 8a479f3a-0454-4a4d-8ade-fd3b5a54ab2e
Subject DN: CN=WEBCIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

3.

ipa-certupdate 명령을 실행하여 **webserver-ca** 및 **webclient-ca** 하위 CA 인증서에 대한 **certmonger** 추적 요청을 생성합니다.

```
[root@ipaserver ~]# ipa-certupdate
```



중요

하위 CA를 생성한 후 **ipa-certupdate** 명령을 실행해야 하며 하위 CA에서 발행한 엔드엔티티 인증서가 만료되지 않은 경우에도 해당 하위 CA에서 발행한 엔드엔티티 인증서가 유효하지 않은 것으로 간주됩니다.

검증 단계

•

새 하위 CA의 서명 인증서가 IdM 데이터베이스에 추가되었는지 확인합니다.

```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L
```

Certificate Nickname	Trust Attributes
	SSL,S/MIME,JAR/XPI
caSigningCert cert-pki-ca	CTu,Cu,Cu
Server-Cert cert-pki-ca	u,u,u
auditSigningCert cert-pki-ca	u,u,Pu


```
caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc u,u,u
ocspSigningCert cert-pki-ca u,u,u
subsystemCert cert-pki-ca u,u,u
```



참고

새 하위 **CA** 인증서는 인증서 시스템 인스턴스가 설치된 모든 복제본으로 자동으로 전송됩니다.

64.1.4. IdM CLI에서 하위 CA 비활성화

이 절차에서는 **IdM CLI**에서 하위 **CA**를 비활성화하는 방법을 설명합니다. 아직 하위 **CA**에서 발급하지 않은 인증서가 있는 경우 삭제하지 않아야 하지만 비활성화할 수 있습니다. 하위 **CA**를 삭제하면 해당 하위 **CA**에 대한 해지 확인이 더 이상 작동하지 않습니다.

사전 요구 사항

- 관리자 자격 증명을 획득했는지 확인합니다.

절차

1. **ipa ca-find** 명령을 실행하여 삭제 중인 하위 **CA**의 이름을 확인합니다.

```
[root@ipaserver ~]# ipa ca-find
-----
3 CAs matched
-----
Name: ipa
Description: IPA CA
Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
Subject DN: CN=Certificate Authority,O=IPA.TEST
Issuer DN: CN=Certificate Authority,O=IPA.TEST

Name: webclient-ca
Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
Subject DN: CN=WEBCIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST

Name: webserver-ca
Authority ID: 02d537f9-c178-4433-98ea-53aa92126fc3
Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST
-----
Number of entries returned 3
-----
```

2.

`ipa ca-disable` 명령을 실행하여 하위 CA를 비활성화합니다. 이 예에서는 **`webserver-ca`**:

```
ipa ca-disable webserver-ca
-----
Disabled CA "webserver-ca"
-----
```

64.1.5. IdM CLI에서 하위 CA 삭제

이 절차에서는 IdM CLI에서 경량 하위 CA를 삭제하는 방법을 설명합니다.

참고

- 하위 CA를 삭제하면 해당 하위 CA에 대한 폐기 검사가 더 이상 작동하지 않습니다. 나중에 만료 시간이 아닌 하위 CA에서 발급한 인증서가 더 이상 없는 경우에만 하위 CA를 삭제합니다.
- 해당 하위 CA에서 발급한 만료되지 않은 인증서가 있는 동안 하위 CA만 비활성화해야 합니다. 하위 CA에서 발급한 모든 인증서가 만료된 경우 해당 하위 CA를 삭제할 수 있습니다.
- IdM CA를 비활성화하거나 삭제할 수 없습니다.

사전 요구 사항

- 관리자 자격 증명을 획득했는지 확인합니다.

절차

1.

하위 CA 및 CA 목록을 표시하려면 **`ipa ca-find`** 명령을 실행합니다.

```
# ipa ca-find
-----
3 CAs matched
-----
Name: ipa
Description: IPA CA
Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
Subject DN: CN=Certificate Authority,O=IPA.TEST
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

```
Name: webclient-ca
Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

```
Name: webserver-ca
Authority ID: 02d537f9-c178-4433-98ea-53aa92126fc3
Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

```
-----
Number of entries returned 3
-----
```

2.

ipa ca-disable 명령을 실행하여 하위 CA를 비활성화합니다. 이 예에서는 **webserver-ca**:

```
# ipa ca-disable webserver-ca
```

```
-----
Disabled CA "webserver-ca"
-----
```

3.

하위 CA를 삭제합니다. 이 예에서는 **webserver-ca**:

```
# ipa ca-del webserver-ca
```

```
-----
Deleted CA "webserver-ca"
-----
```

검증

•

ipa ca-find 를 실행하여 CA 및 하위 CA 목록을 표시합니다. **webserver-ca** 가 더 이상 목록에 없습니다.

```
# ipa ca-find
```

```
-----
2 CAs matched
-----
```

```
Name: ipa
Description: IPA CA
Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
Subject DN: CN=Certificate Authority,O=IPA.TEST
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

```
Name: webclient-ca
Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

Number of entries returned 2

64.2. IDM WEBUI에서 하위 CA 인증서 다운로드

사전 요구 사항

- **IdM 관리자의 자격 증명을 획득했는지 확인합니다.**

절차

1. **인증 메뉴에서 인증서 > 인증서를 클릭합니다.**

그림 64.2. 인증서 목록의 하위 CA 인증서

☐	268173326	CN=WEBSERVER,O=IDM.EXAMPLE.COM	ipa	VALID
☐	268238849	CN=idm_user,O=IDM.EXAMPLE.COM	ipa	VALID

2. **하위 CA 인증서의 일련 번호를 클릭하여 인증서 정보 페이지를 엽니다.**
3. **인증서 정보 페이지에서 작업 > 다운로드를 클릭합니다.**
4. **CLI에서 하위 CA 인증서를 `/etc/pki/tls/private/` 디렉터리로 이동합니다.**

```
# mv path/to/the/downloaded/certificate /etc/pki/tls/private/sub-ca.crt
```

64.3. 웹 서버 및 클라이언트 인증을 위한 CA ACL 생성

CA ACL(인증 기관 액세스 제어 목록) 규칙은 어떤 사용자, 서비스 또는 호스트에 대한 인증서를 발급하는 데 사용할 수 있는 프로필을 정의합니다. **CA ACL**은 프로필, 주체 및 그룹을 연결하면 주체나 그룹이 특정 프로필을 사용하여 인증서를 요청할 수 있습니다.

예를 들어 **CA ACL**을 사용하면 관리자는 런던 사무소에 있는 사무실에서 근무하는 직원을 위해 설계된 프로필 사용을 런던 사무소 관련 그룹의 멤버인 사용자로만 제한할 수 있습니다.

64.3.1. IdM CLI에서 CA ACL 보기

IdM 배포에서 사용할 수 있는 **CA ACL**(인증 기관 액세스 제어 목록) 목록과 특정 **CA ACL**의 세부 정보를 보려면 이 섹션을 완료합니다.

절차

1.

IdM 환경의 모든 **CA ACL**을 보려면 **ipa caacl-find** 명령을 입력합니다.

```
$ ipa caacl-find
-----
1 CA ACL matched
-----
ACL name: hosts_services_calPAserviceCert
Enabled: TRUE
```

2.

CA ACL의 세부 정보를 보려면 **ipa caacl-show** 명령을 입력하고 **CA ACL** 이름을 지정합니다. 예를 들어 **hosts_services_calPAserviceCert** **CA ACL**의 세부 정보를 보려면 다음을 입력합니다.

```
$ ipa caacl-show hosts_services_calPAserviceCert
ACL name: hosts_services_calPAserviceCert
Enabled: TRUE
Host category: all
Service category: all
CAs: ipa
Profiles: calPAserviceCert
Users: admin
```

64.3.2. webserver-ca에서 발급한 인증서를 사용하여 웹 클라이언트에 인증하기 위한 **CA ACL** 생성

이 섹션에서는 시스템 관리자가 **HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM** 서비스에 대한 인증서를 요청할 때 **webserver-ca** 하위 **CA** 및 **calPAserviceCert** 프로필을 사용해야 하는 **CA ACL**을 생성하는 방법에 대해 설명합니다. 사용자가 다른 하위 **CA** 또는 다른 프로필의 인증서를 요청하면 요청이 실패합니다. 유일한 예외는 활성화된 다른 **CA ACL**이 일치하는 경우입니다. 사용 가능한 **CA ACL**을 보려면 **IdM CLI**에서 **CA ACL** 보기를 참조하십시오.

사전 요구 사항

- **HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM** 서비스가 **IdM**의 일부인지 확인합니다.
- **IdM** 관리자의 자격 증명을 획득했는지 확인하십시오.

절차

1.

ipa caacl 명령을 사용하여 **CA ACL**을 생성하고 해당 이름을 지정합니다.

```
$ ipa caacl-add TLS_web_server_authentication
-----
Added CA ACL "TLS_web_server_authentication"
-----
ACL name: TLS_web_server_authentication
Enabled: TRUE
```

2.

ipa caacl-mod 명령을 사용하여 **CA ACL**에 대한 설명을 지정합니다.

```
$ ipa caacl-mod TLS_web_server_authentication --desc="CAACL for web servers
authenticating to web clients using certificates issued by webserver-ca"
-----
Modified CA ACL "TLS_web_server_authentication"
-----
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
Enabled: TRUE
```

3.

webserver-ca 하위 **CA**를 **CA ACL**에 추가합니다.

```
$ ipa caacl-add-ca TLS_web_server_authentication --ca=webserver-ca
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
Enabled: TRUE
CAs: webserver-ca
-----
Number of members added 1
-----
```

4.

ipa caacl-add-service 를 사용하여 주체가 인증서를 요청할 수 있는 서비스를 지정합니다.

```
$ ipa caacl-add-service TLS_web_server_authentication --
service=HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
Enabled: TRUE
CAs: webserver-ca
Services: HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
-----
Number of members added 1
-----
```

5.

ipa caacl-add-profile 명령을 사용하여 요청된 인증서의 인증서 프로필을 지정합니다.

```
$ ipa caacl-add-profile TLS_web_server_authentication --
certprofiles=calPAserviceCert
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
Enabled: TRUE
CAs: webserver-ca
Profiles: calPAserviceCert
Services: HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
```

```
-----
Number of members added 1
-----
```

새로 생성된 **CA ACL**을 바로 사용할 수 있습니다. 기본적으로 생성된 후에 활성화됩니다.



참고

CA ACL의 지점은 특정 주체 또는 그룹에서 들어오는 요청에 허용되는 **CA** 및 프로필 조합을 지정하는 것입니다. **CA ACL**은 인증서 검증 또는 신뢰에 영향을 미치지 않습니다. 발급한 인증서 사용 방법에는 영향을 미치지 않습니다.

64.3.3. webclient-ca에서 발급한 인증서를 사용하여 사용자 웹 브라우저에 대한 CA ACL 생성

이 섹션에서는 시스템 관리자가 인증서를 요청할 때 **webclient-ca** 하위 **CA** 및 **UserRoles** 프로필을 사용해야 하는 **CA ACL**을 생성하는 방법에 대해 설명합니다. 사용자가 다른 하위 **CA** 또는 다른 프로필의 인증서를 요청하면 요청이 실패합니다. 유일한 예외는 활성화된 다른 **CA ACL**이 일치하는 경우입니다. 사용 가능한 **CA ACL**을 보려면 **IdM CLI**에서 **CA ACL** 보기를 참조하십시오.

사전 요구 사항

-

IdM 관리자의 자격 증명을 획득했는지 확인합니다.

절차

1.

ipa caacl 명령을 사용하여 **CA ACL**을 생성하고 해당 이름을 지정합니다.

```
$ ipa caacl-add TLS_web_client_authentication
-----
Added CA ACL "TLS_web_client_authentication"
```

```
-----
ACL name: TLS_web_client_authentication
Enabled: TRUE
```

2.

ipa caacl-mod 명령을 사용하여 CA ACL에 대한 설명을 지정합니다.

```
$ ipa caacl-mod TLS_web_client_authentication --desc="CAACL for user web
browsers authenticating to web servers using certificates issued by webclient-ca"
```

```
-----
Modified CA ACL "TLS_web_client_authentication"
```

```
-----
ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
Enabled: TRUE
```

3.

webclient-ca 하위 CA를 CA ACL에 추가합니다.

```
$ ipa caacl-add-ca TLS_web_client_authentication --ca=webclient-ca
ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
Enabled: TRUE
CAs: webclient-ca
```

```
-----
Number of members added 1
-----
```

4.

ipa caacl-add-profile 명령을 사용하여 요청된 인증서의 인증서 프로필을 지정합니다.

```
$ ipa caacl-add-profile TLS_web_client_authentication --certprofiles=IECUserRoles
ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
Enabled: TRUE
CAs: webclient-ca
Profiles: IECUserRoles
```

```
-----
Number of members added 1
-----
```

5.

ipa caacl-mod 명령을 사용하여 CA ACL을 모든 IdM 사용자에게 적용하도록 지정합니다.

```
$ ipa caacl-mod TLS_web_client_authentication --usercat=all
-----
Modified CA ACL "TLS_web_client_authentication"
```



```

-----
ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
Enabled: TRUE
User category: all
CAs: webclient-ca
Profiles: IECUserRoles

```

새로 생성된 **CA ACL**을 바로 사용할 수 있습니다. 기본적으로 생성된 후에 활성화됩니다.



참고

CA ACL의 지점은 특정 주체 또는 그룹에서 들어오는 요청에 허용되는 **CA** 및 프로파일 조합을 지정하는 것입니다. **CA ACL**은 인증서 검증 또는 신뢰에 영향을 미치지 않습니다. 발급한 인증서 사용 방법에는 영향을 미치지 않습니다.

64.4. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기

IdM 클라이언트에서 실행되는 브라우저와 웹 서비스 간의 통신이 안전하고 암호화되었는지 확인하려면 **TLS** 인증서를 사용합니다. **webserver-ca** 하위-**CA**에서 발급했지만 다른 **IdM** 하위 **CA**가 발행한 인증서를 신뢰하도록 웹 브라우저를 제한하려면 **webserver-ca** 하위 **CA**에서 웹 서비스의 **TLS** 인증서를 가져옵니다.

이 섹션에서는 **certmonger**를 사용하여 **IdM** 클라이언트에서 실행되는 서비스에 대한 **IdM** 인증서 (**HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM**)를 가져오는 방법에 대해 설명합니다.

certmonger를 사용하여 인증서를 자동으로 요청하면 **certmonger**가 갱신될 때 인증서를 관리하고 갱신합니다.

certmonger에서 서비스 인증서를 요청할 때 발생하는 사항을 시각적으로 표시하려면 [64.5절. “서비스 인증서를 요청하는 certmonger의 통신 흐름”](#)을 참조하십시오.

사전 요구 사항

- 웹 서버가 **IdM** 클라이언트로 등록되었습니다.
- 프로시저를 실행 중인 **IdM** 클라이언트에 대한 루트 액세스 권한이 있습니다.

- 인증서를 요청하는 서비스는 **IdM**에 기존 존재하지 않아도 됩니다.

절차

1.

HTTP 서비스가 실행 중인 **my_company.idm.example.com** **IdM** 클라이언트에서 **HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM** 주체에 해당하는 서비스에 대한 인증서를 요청하고 다음을 지정합니다.

- 인증서는 로컬 **/etc/pki/tls/certs/httpd.pem** 파일에 저장해야 합니다.
- 개인 키는 로컬 **/etc/pki/tls/private/httpd.key** 파일에 저장해야 합니다.
- **webserver-ca** 하위 **CA**는 인증 기관을 발급하는 것입니다.
- **my_company.idm.example.com**의 **DNS** 이름을 사용하여 서명 요청에 **SubjectAltName**의 **extensionRequest**를 추가합니다.

```
# ipa-getcert request -K HTTP/my_company.idm.example.com -k
/etc/pki/tls/private/httpd.key -f /etc/pki/tls/certs/httpd.pem -g 2048 -D
my_company.idm.example.com -X webserver-ca -C "systemctl restart httpd"
New signing request "20190604065735" added.
```

위의 명령에서 다음을 수행합니다.

- **ipa-getcert request** 명령은 인증서를 **IdM CA**에서 가져오도록 지정합니다. **ipa-getcert request** 명령은 **getcert request -c IPA**의 바로 가기입니다.
- **g** 옵션은 아직 배치되지 않은 경우 생성할 키 크기를 지정합니다.
- **D** 옵션은 요청에 추가할 **SubjectAltName DNS** 값을 지정합니다.
- **X** 옵션은 인증서의 발급자가 **ipa**가 아니라 **webserver-ca**여야 함을 지정합니다.
-

C 옵션은 인증서를 가져온 후 **certmonger** 에 **httpd** 서비스를 다시 시작하도록 지시합니다.

○

인증서를 특정 프로파일과 함께 발행하도록 지정하려면 **-T** 옵션을 사용합니다.

+



참고

RHEL 8에서는 **RHEL 7**에 사용된 모듈과 **Apache**에서 다른 **SSL** 모듈을 사용합니다. **SSL** 모듈은 **NSS** 대신 **OpenSSL**을 사용합니다. 따라서 **RHEL 8**에서는 **NSS** 데이터베이스를 사용하여 **HTTPS** 인증서 및 개인 키를 저장할 수 없습니다.

+ . 선택적으로 요청 상태를 확인하려면 다음을 수행합니다.

+

```
# ipa-getcert list -f /etc/pki/tls/certs/httpd.pem
Number of certificates and requests being tracked: 3.
Request ID '20190604065735':
  status: MONITORING
  stuck: no
  key pair storage: type=FILE,location='/etc/pki/tls/private/httpd.key'
  certificate: type=FILE,location='/etc/pki/tls/certs/httpd.crt'
  CA: IPA
  issuer: CN=WEBSERVER,O=IDM.EXAMPLE.COM
```

[...]

+ 출력은 요청이 **MONITORING** 상태에 있음을 보여주며, 이는 인증서를 가져왔음을 의미합니다. 키 쌍과 인증서의 위치는 요청된 값입니다.

64.5. 서비스 인증서를 요청하는 **CERTMONGER**의 통신 흐름

이 섹션의 다이어그램은 **certmonger** 에서 **IdM(Identity Management)** **CA**(인증 기관) 서버에서 서비스 인증서를 요청할 때 발생하는 작업의 단계를 보여줍니다. 시퀀스는 다음 다이어그램으로 구성됩니다.

- [그림 64.3. “암호화되지 않은 통신”](#)
- [그림 64.4. “서비스 인증서 요청 certmonger”](#)
- [그림 64.5. “IdM CA에서 서비스 인증서 발행”](#)
- [그림 64.6. “서비스 인증서 적용 certmonger”](#)
- [그림 64.7. “이전 인증서가 만료될 때 새 인증서를 요청하는 certmonger”](#)

다이어그램에서 **webserver-ca** 하위 **CA**는 일반 **IdM CA** 서버로 표시됩니다.

[그림 64.3. “암호화되지 않은 통신”](#)는 초기 상황을 보여줍니다. **HTTPS** 인증서가 없으면 웹 서버와 브라우저 간의 통신이 암호화되지 않습니다.

그림 64.3. 암호화되지 않은 통신

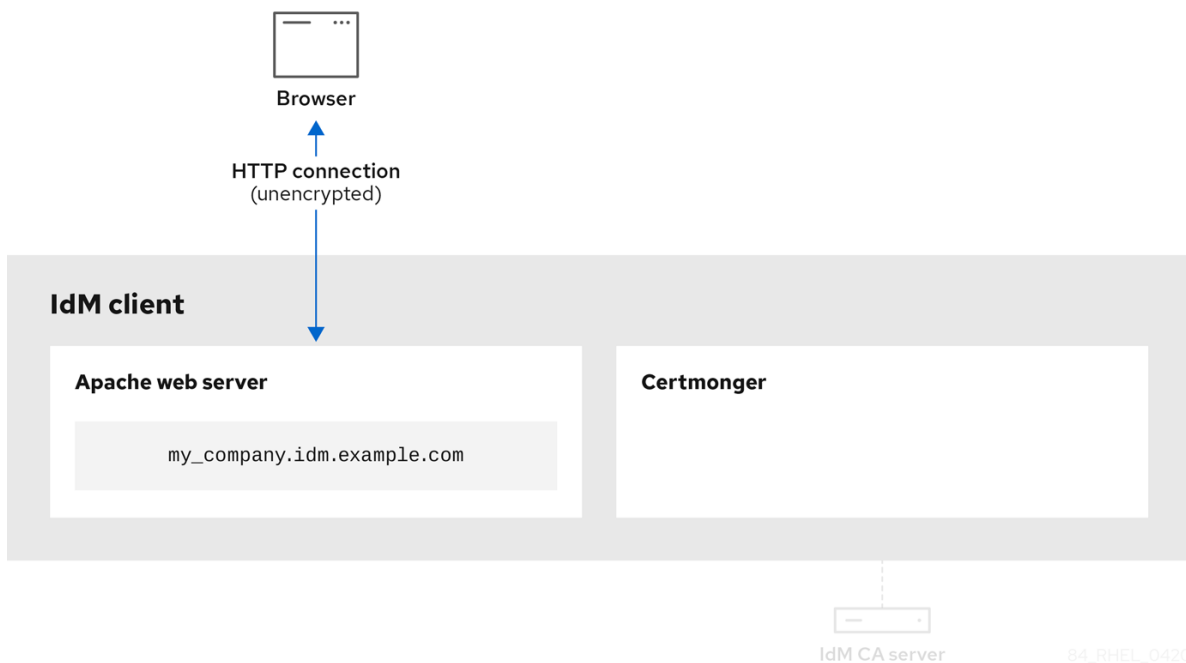


그림 64.4. “서비스 인증서 요청 certmonger” certmonger 를 사용하여 Apache 웹 서버에 대한 HTTPS 인증서를 수동으로 요청하는 시스템 관리자를 보여줍니다. 웹 서버 인증서를 요청할 때 certmonger는 CA와 직접 통신하지 않습니다. IdM을 통해 프록시됩니다.

그림 64.4. 서비스 인증서 요청 certmonger

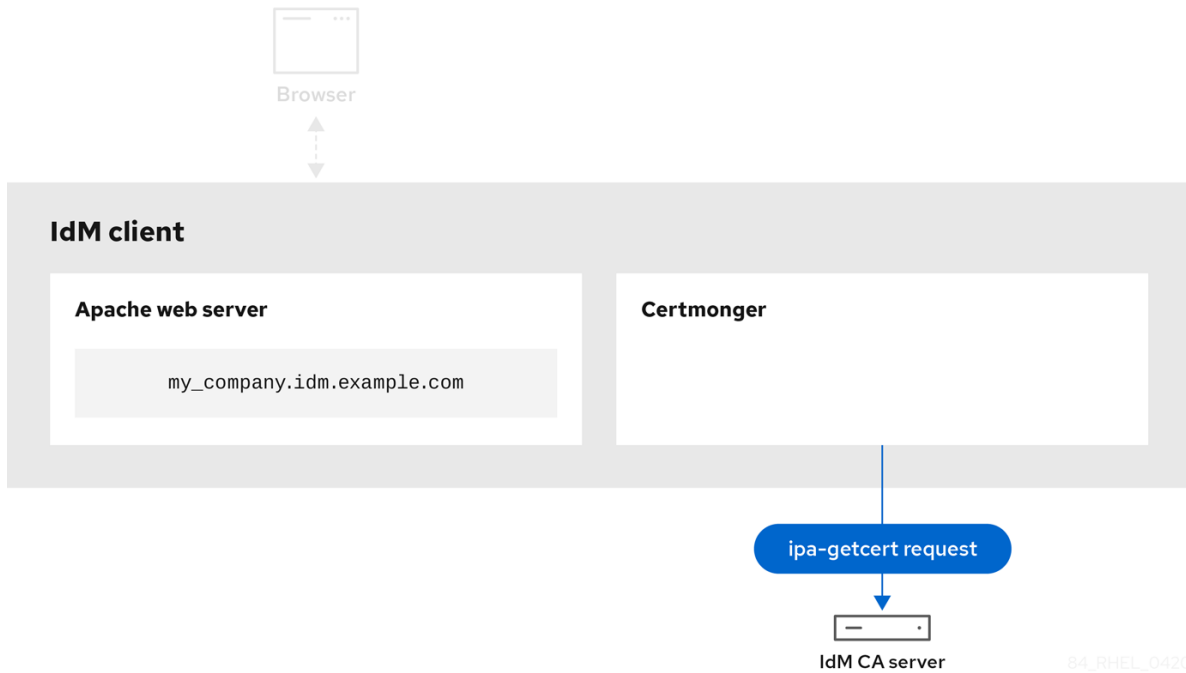


그림 64.5. “IdM CA에서 서비스 인증서 발행”에는 웹 서버의 HTTPS 인증서를 발급하는 IdM CA가 표시됩니다.

그림 64.5. IdM CA에서 서비스 인증서 발행

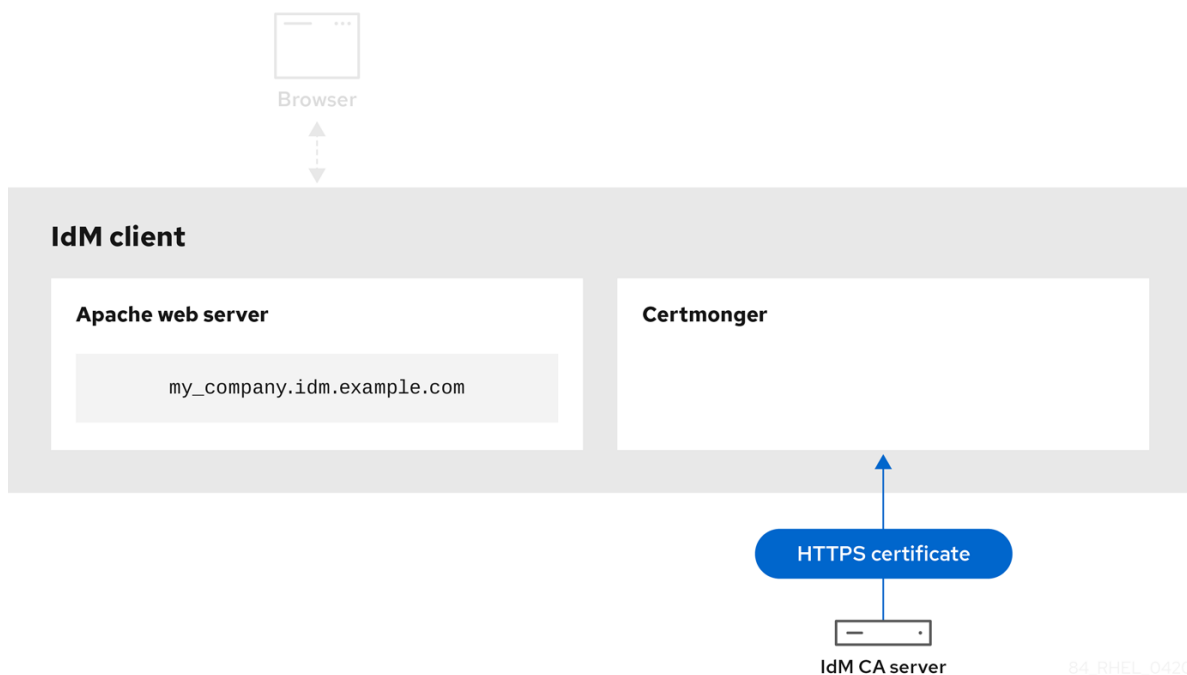


그림 64.6. “서비스 인증서 적용 **certmonger**” 는 **HTTPS** 인증서를 **IdM** 클라이언트의 적절한 위치에 배치한 **certmonger** 를 보여줍니다. 이렇게 지시한 경우 **httpd** 서비스를 다시 시작합니다. 결과적으로 **Apache** 서버는 **HTTPS** 인증서를 사용하여 자체와 브라우저 간의 트래픽을 암호화합니다.

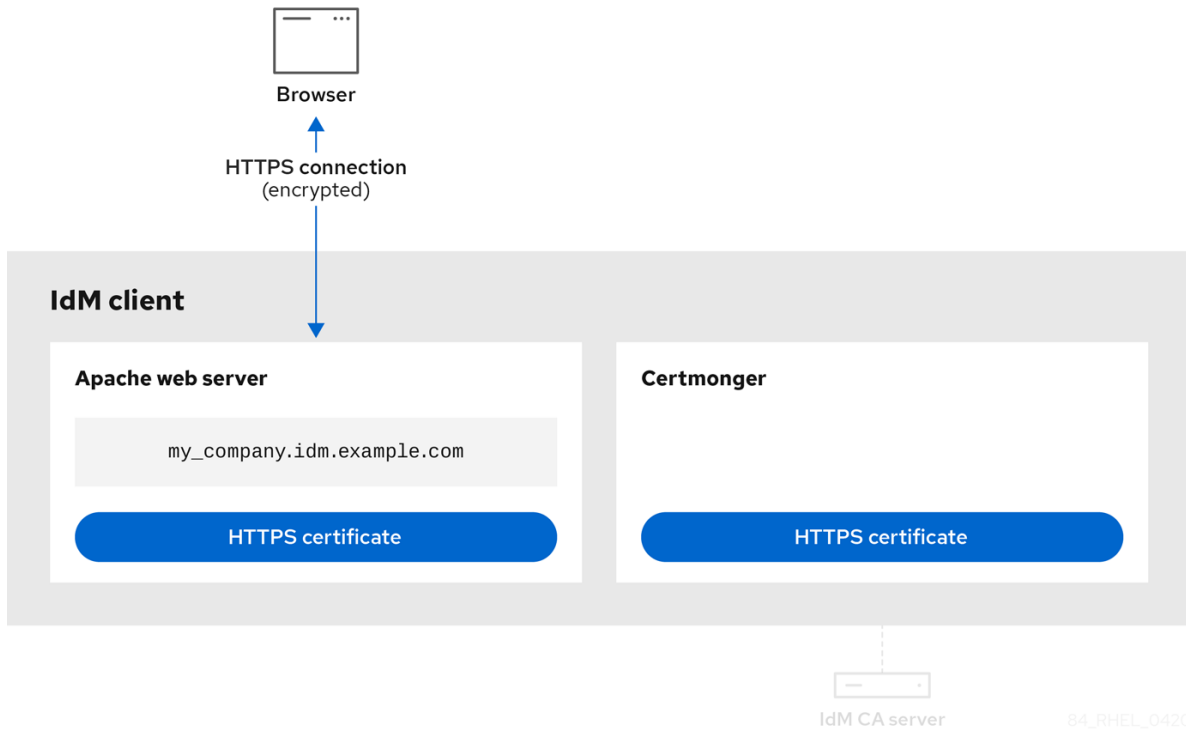
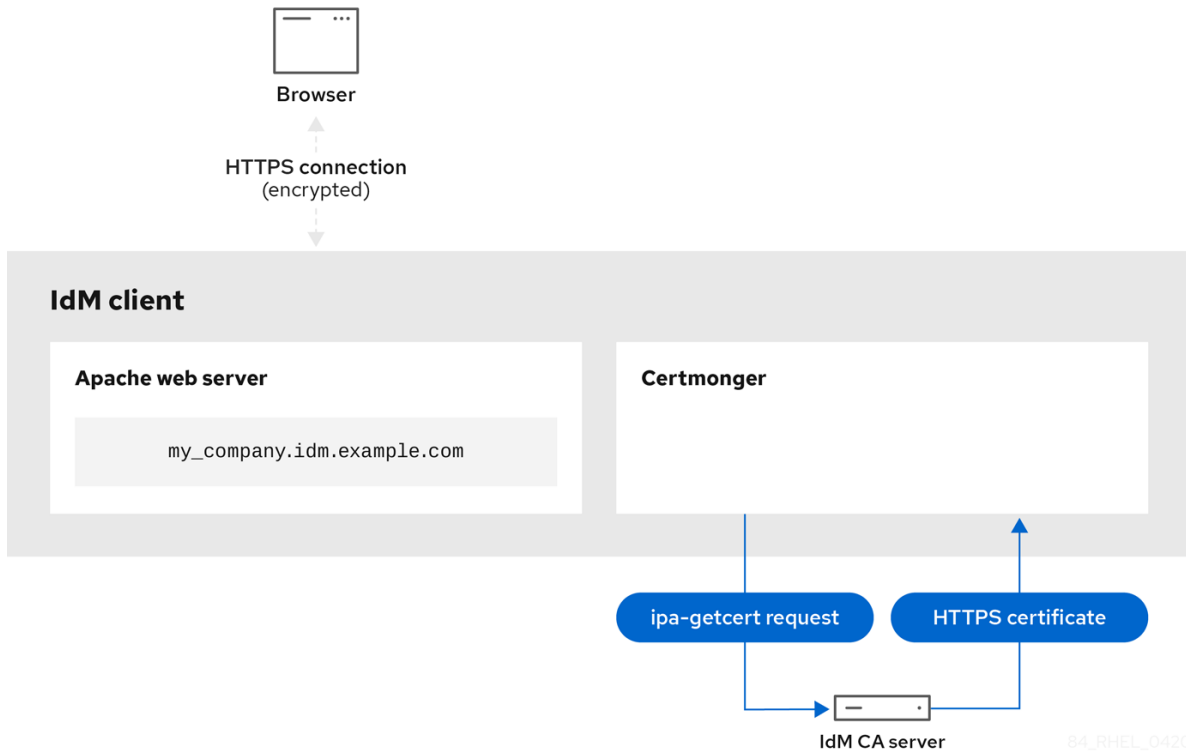
그림 64.6. 서비스 인증서 적용 **certmonger**

그림 64.7. “이전 인증서가 만료될 때 새 인증서를 요청하는 **certmonger**” **certmonger**에서 인증서 만료 전에 IdM CA에서 서비스 인증서 갱신을 자동으로 요청하는 방법을 보여줍니다. IdM CA는 새 인증서를 발행합니다.

그림 64.7. 이전 인증서가 만료될 때 새 인증서를 요청하는 **certmonger**

84_RHEL_0420

64.6. 단일 인스턴스 **APACHE HTTP** 서버 설정

이 섹션에서는 정적 **HTML** 콘텐츠를 제공하도록 단일 인스턴스 **Apache HTTP** 서버를 설정하는 방법에 대해 설명합니다.

웹 서버가 서버와 연결된 모든 도메인에 대해 동일한 콘텐츠를 제공해야 하는 경우 이 섹션의 절차를 따릅니다. 도메인마다 다른 콘텐츠를 제공하려면 이름 기반 가상 호스트를 설정합니다. 자세한 내용은 [Apache 이름 기반 가상 호스트](#) 구성을 참조하십시오.

절차

1. **httpd** 패키지를 설치합니다.

```
# yum install httpd
```

2. 로컬 방화벽에서 **TCP** 포트 **80** 을 엽니다.


```
# firewall-cmd --permanent --add-port=80/tcp
# firewall-cmd --reload
```

3.

httpd 서비스를 활성화하고 시작합니다.

```
# systemctl enable --now httpd
```

4.

선택 사항: HTML 파일을 /var/www/html/ 디렉토리에 추가합니다.



참고

콘텐츠를 /var/www/html/ 에 추가하는 경우 **httpd** 가 기본적으로 실행되는 사용자가 파일과 디렉토리를 읽을 수 있어야 합니다. 콘텐츠 소유자는 **root** 사용자 및 **root** 사용자 그룹 또는 관리자가 선택한 다른 사용자 또는 그룹일 수 있습니다. 콘텐츠 소유자가 **root** 사용자 및 **root** 사용자 그룹인 경우 다른 사용자가 파일을 읽을 수 있어야 합니다. 모든 파일과 디렉토리에 대한 **SELinux** 컨텍스트는 기본적으로 /var/www 디렉터리 내의 모든 콘텐츠에 적용되는 **httpd_sys_content_t** 여야 합니다.

검증 단계

•

웹 브라우저와 **http://my_company.idm.example.com/** 또는 **http://server_IP/** 에 연결합니다.

/var/www/html/ 디렉터리가 비어 있거나 **index.html** 또는 **index.htm** 파일이 없는 경우 **Apache**에 **Red Hat Enterprise Linux Test Page** 가 표시됩니다. /var/www/html/ 에 다른 이름의 **HTML** 파일이 포함된 경우 **http://server_IP/example.html** 또는 **http://my_company.idm.example.com/example.html** 과 같이 해당 파일에 **URL**을 입력하여 로드할 수 있습니다.

추가 리소스

•

Apache 구성 및 환경에 서비스를 조정하는 방법에 대한 자세한 내용은 **Apache** 설명서를 참조하십시오. 설명서 설치에 대한 자세한 내용은 [Apache HTTP Server 설명서 설치](#)를 참조하십시오.

•

httpd systemd 서비스 사용 또는 조정에 대한 자세한 내용은 **httpd.service(8)** 도움말 페이지를 참조하십시오.

64.7. APACHE HTTP 서버에 TLS 암호화 추가

이 섹션에서는 **my_company.idm.example.com** Apache HTTP 서버에서 **idm.example.com** 도메인에 대해 TLS 암호화를 활성화하는 방법을 설명합니다.

사전 요구 사항

- **my_company.idm.example.com** Apache HTTP 서버가 설치되어 실행 중입니다.
- **webserver-ca** 하위 CA에서 TLS 인증서를 가져와 64.4절. “**certmonger**를 사용하여 서비스에 대한 IdM 인증서 가져오기”에 설명된 대로 **/etc/pki/tls/certs/httpd.pem** 파일에 저장했습니다. 다른 경로를 사용하는 경우 해당 절차의 단계를 조정합니다.
- 해당 개인 키는 **/etc/pki/tls/private/httpd.key** 파일에 저장됩니다. 다른 경로를 사용하는 경우 해당 절차의 단계를 조정합니다.
- **webserver-ca** CA 인증서는 **/etc/pki/tls/private/sub-ca.crt** 파일에 저장됩니다. 다른 경로를 사용하는 경우 해당 절차의 단계를 조정합니다.
- 클라이언트 및 **my_company.idm.example.com** 웹 서버는 서버의 호스트 이름을 웹 서버의 IP 주소로 확인합니다.

절차

1. **mod_ssl** 패키지를 설치합니다.


```
# yum install mod_ssl
```
2. **/etc/httpd/conf.d/ssl.conf** 파일을 편집하고 <VirtualHost _default_:443> 지시문에 다음 설정을 추가합니다.
 - a. 서버 이름을 설정합니다.

```
ServerName my_company.idm.example.com
```



중요

서버 이름은 인증서의 **Common Name**(일반 이름) 필드에 있는 항목 세트와 일치해야 합니다.

b.

선택 사항: 인증서에 **SAN**(주체 **Alt Names**) 필드에 추가 호스트 이름이 포함된 경우 이러한 호스트 이름에 **TLS** 암호화도 제공하도록 **mod_ssl** 을 구성할 수 있습니다. 이를 구성하려면 해당 이름을 사용하여 **ServerAliases** 매개변수를 추가합니다.

```
ServerAlias www.my_company.idm.example.com
server.my_company.idm.example.com
```

c.

개인 키, 서버 인증서 및 **CA** 인증서에 대한 경로를 설정합니다.

```
SSLCertificateKeyFile "/etc/pki/tls/private/httpd.key"
SSLCertificateFile "/etc/pki/tls/certs/httpd.pem"
SSLCACertificateFile "/etc/pki/tls/certs/ca.crt"
```

3.

보안상의 이유로 **root** 사용자만 개인 키 파일에 액세스할 수 있도록 구성합니다.

```
# chown root:root /etc/pki/tls/private/httpd.key
# chmod 600 //etc/pki/tls/private/httpd.key
```



주의

권한 없는 사용자가 개인 키에 액세스한 경우 인증서를 취소하고 새 개인 키를 생성하고 새 인증서를 요청합니다. 그렇지 않으면 **TLS** 연결이 더 이상 안전하지 않습니다.

4.

로컬 방화벽에서 포트 **443** 을 엽니다.

```
# firewall-cmd --permanent --add-port=443/tcp
# firewall-cmd --reload
```

5.

httpd 서비스를 다시 시작하십시오.

```
# systemctl restart httpd
```



참고

개인 키 파일을 암호로 보호한 경우 **httpd** 서비스가 시작될 때마다 이 암호를 입력해야 합니다.

-

브라우저를 사용하고 연결합니다.

추가 리소스

-

TLS 구성에 대한 자세한 내용은 **Apache** 설명서의 **SSL/TLS 암호화 설명서**를 참조하십시오. 설명서 설치에 대한 자세한 내용은 [Apache HTTP Server 설명서 설치](#)를 참조하십시오.

-

[RHEL 8의 TLS 보안 고려 사항](#)

64.8. APACHE HTTP 서버에서 지원되는 TLS 프로토콜 버전 설정

기본적으로 **RHEL**의 **Apache HTTP Server**는 최신 브라우저와 호환되는 안전한 기본값을 정의하는 시스템 전체 암호화 정책을 사용합니다. 예를 들어 **DEFAULT** 정책은 **TLSv1.2** 및 **TLSv 1.3** 프로토콜 버전만 **apache**에서 사용하도록 정의합니다.

이 섹션에서는 **my_company.idm.example.com** **Apache HTTP** 서버에서 지원하는 **TLS** 프로토콜 버전을 수동으로 구성하는 방법에 대해 설명합니다. 환경에서 특정 **TLS** 프로토콜 버전만 활성화해야 하는 경우 절차를 따릅니다. 예를 들면 다음과 같습니다.

-

환경에서 클라이언트가 취약한 **TLS1(TLSv1.0)** 또는 **TLS 1.1** 프로토콜을 사용할 수 있어야 하는 경우.

-

Apache가 **TLSv1.2** 또는 **TLSv 1.3** 프로토콜만 지원하도록 구성하려는 경우.

사전 요구 사항

-

TLS 암호화는 [64.7절. “Apache HTTP 서버에 TLS 암호화 추가”](#)에 설명된 대로 **my_company.idm.example.com** 서버에서 활성화됩니다.

절차

1. **/etc/httpd/conf/httpd.conf** 파일을 편집하고 **TLS** 프로토콜 버전을 설정하려는 **<VirtualHost>** 지시문에 다음 설정을 추가합니다. 예를 들어 **TLSv1.3** 프로토콜만 활성화하려면 다음을 수행합니다.

```
SSLProtocol -All TLSv1.3
```

2. **httpd** 서비스를 다시 시작하십시오.

```
# systemctl restart httpd
```

검증 단계

1. 다음 명령을 사용하여 서버가 **TLSv1.3** 을 지원하는지 확인합니다.

```
# openssl s_client -connect example.com:443 -tls1_3
```

2. 다음 명령을 사용하여 서버가 **TLSv1.2** 를 지원하지 않는지 확인합니다.

```
# openssl s_client -connect example.com:443 -tls1_2
```

서버가 프로토콜을 지원하지 않으면 명령에서 오류를 반환합니다.

```
140111600609088:error:1409442E:SSL routines:ssl3_read_bytes:tlsv1 alert protocol version:ssl/record/rec_layer_s3.c:1543:SSL alert number 70
```

3. 선택 사항: 다른 **TLS** 프로토콜 버전에 대해 명령을 반복합니다.

추가 리소스

- 시스템 전체 암호화 정책에 대한 자세한 내용은 **update-crypto-policies(8)** 도움말 페이지 및 **시스템 전체 암호화 정책** 사용을 참조하십시오.
- **SSLProtocol** 매개변수에 대한 자세한 내용은 **Apache** 설명서의 **mod_ssl** 설명서를 참조하십시오. 설명서 설치에 대한 자세한 내용은 **Apache HTTP Server 설명서 설치**를 참조하십시오.

64.9. APACHE HTTP 서버에서 지원되는 암호 설정

기본적으로 **Apache HTTP** 서버는 최근 브라우저와 호환되는 안전한 기본값을 정의하는 시스템 전체 암호화 정책을 사용합니다. 시스템 전체 암호화에서 허용하는 암호 목록은 `/etc/crypto-policies/back-ends/openssl.config` 파일을 참조하십시오.

이 섹션에서는 `my_company.idm.example.com` **Apache HTTP** 서버에서 지원하는 암호를 수동으로 구성하는 방법에 대해 설명합니다. 환경에 특정 암호가 필요한 경우 절차를 따릅니다.

사전 요구 사항

- TLS 암호화는 [64.7절. “Apache HTTP 서버에 TLS 암호화 추가”](#)에 설명된 대로 `my_company.idm.example.com` 서버에서 활성화됩니다.

절차

1. `/etc/httpd/conf/httpd.conf` 파일을 편집하고 TLS 암호를 설정하려는 `<VirtualHost>` 지시문에 `SSLCipherSuite` 매개변수를 추가합니다.

```
SSLCipherSuite
"EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH:!SHA1:!SHA256"
```

이 예에서는 `EECDH+AESGCM`, `EDH+AESGCM`, `AES256+EECDH` 및 `AES256+EDH` 암호화만 활성화하고 `SHA1` 및 `SHA 256` 메시지 인증 코드(MAC)를 사용하는 모든 암호를 비활성화합니다.

2. `httpd` 서비스를 다시 시작하십시오.

```
# systemctl restart httpd
```

검증 단계

1. 암호 목록을 표시하려면 **Apache HTTP Server**에서 지원하는 암호 목록을 표시합니다.

- a. `nmap` 패키지를 설치합니다.

```
# yum install nmap
```

nmap 유틸리티를 사용하여 지원되는 암호를 표시합니다.

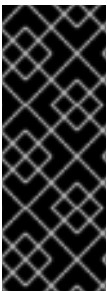
```
# nmap --script ssl-enum-ciphers -p 443 example.com
...
PORT      STATE SERVICE
443/tcp   open  https
/ ssl-enum-ciphers:
/  TLSv1.2:
/    ciphers:
/      TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdhe_x25519) - A
/      TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
/      TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdhe_x25519) - A
...
```

추가 리소스

- 시스템 전체 암호화 정책에 대한 자세한 내용은 [update-crypto-policies\(8\)](#) 도움말 페이지 및 [시스템 전체 암호화 정책](#) 사용을 참조하십시오.
- **SSLCipherSuite** 매개변수에 대한 자세한 내용은 [Apache](#) 설명서의 **mod_ssl** 설명서를 참조하십시오. 설명서 설치에 대한 자세한 내용은 [Apache HTTP Server 설명서 설치](#)를 참조하십시오.

64.10. TLS 클라이언트 인증서 인증 구성

클라이언트 인증서 인증을 통해 관리자는 인증서를 사용하여 인증한 사용자만 `my_company.idm.example.com` 웹 서버의 리소스에 액세스할 수 있습니다. 이 섹션에서는 `/var/www/html/Example/` 디렉터리에 대한 클라이언트 인증서 인증을 구성하는 방법에 대해 설명합니다.



중요

my_company.idm.example.com Apache 서버가 TLS 1.3 프로토콜을 사용하는 경우 특정 클라이언트에 추가 구성이 필요합니다. 예를 들어 Firefox에서 **about:config** 메뉴의 **security.tls.enable_post_handshake_auth** 매개 변수를 **true**로 설정합니다. 자세한 내용은 **Red Hat Enterprise Linux 8의 전송 계층 보안 버전 1.3**을 참조하십시오.

사전 요구 사항

- TLS 암호화는 **64.7절. “Apache HTTP 서버에 TLS 암호화 추가”**에 설명된 대로 `my_company.idm.example.com` 서버에서 활성화됩니다.

절차

1.

/etc/httpd/conf/httpd.conf 파일을 편집하고 클라이언트 인증을 구성하려는 **<VirtualHost>** 지시문에 다음 설정을 추가합니다.

```
<Directory "/var/www/html/Example/">
    SSLVerifyClient require
</Directory>
```

SSLVerifyClient 설정은 클라이언트가 **/var/www/html/Example/** 디렉터리의 콘텐츠에 액세스할 수 있으려면 서버가 클라이언트 인증서를 성공적으로 검증해야 함을 정의합니다.

2.

httpd 서비스를 다시 시작하십시오.

```
# systemctl restart httpd
```

검증 단계

1.

curl 유틸리티를 사용하여 클라이언트 인증없이 **https://my_company.idm.example.com/Example/** URL에 액세스합니다.

```
$ curl https://my_company.idm.example.com/Example/
curl: (56) OpenSSL SSL_read: error:1409445C:SSL routines:ssl3_read_bytes:tlsv13 **alert
certificate required**, errno 0
```

이 오류는 **my_company.idm.example.com** 웹 서버에 클라이언트 인증서 인증이 필요함을 나타냅니다.

2.

클라이언트 개인 키 및 인증서와 **CA** 인증서를 전달하여 클라이언트 인증으로 동일한 URL에 액세스합니다.

```
$ curl --cacert ca.crt --key client.key --cert client.crt
https://my_company.idm.example.com/Example/
```

요청이 성공하면 **curl** 은 **/var/www/html/Example/** 디렉터리에 저장된 **index.html** 파일을 표시합니다.

추가 리소스

•

클라이언트 인증에 대한 자세한 내용은 **Apache** 설명서의 **mod_ssl** 구성 방법 설명서를 참조하십시오. 설명서 설치에 대한 자세한 내용은 **Apache HTTP Server 설명서 설치**를 참조하십시오

오.

64.11. 새 사용자 인증서 요청 및 클라이언트로 내보내기

IdM(Identity Management) 관리자는 **IdM** 클라이언트에서 실행되는 웹 서버를 구성하여 웹 브라우저를 사용하여 서버에 액세스하여 특정 **IdM** 하위 **CA**에서 발급한 인증서로 인증하는 사용자를 요청할 수 있습니다. 특정 **IdM** 하위 **CA**에서 사용자 인증서를 요청하고 의 인증서 및 해당 개인 키를 사용자가 웹 브라우저를 사용하여 웹 서버에 액세스하려는 호스트로 내보내려면 이 섹션을 완료합니다. 그런 다음 인증서와 개인 키를 브라우저로 가져옵니다.

절차

1.

선택적으로 `~/certdb/` 와 같은 새 디렉터리를 생성하고 임시 인증서 데이터베이스로 만듭니다. 메시지가 표시되면 **NSS** 인증서 **DB** 암호를 생성하여 후속 단계에서 생성할 인증서의 키를 암호화합니다.

```
# mkdir ~/certdb/
# certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

2.

CSR(인증서 서명 요청)을 만들고 출력을 파일로 리디렉션합니다. 예를 들어 **IDM.EXAMPLE.COM** 영역에서 **idm_user** 사용자에게 대해 **certificate_request.csr** 이라는 이름으로 **CSR**을 생성하려면 찾기 쉽도록 인증서 개인 키의 닉네임을 **idm_user** 로 설정하고 제목을 **CN=idm_user,O=IDM.EXAMPLE.COM**:

```
# certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s
"CN=idm_user,O=IDM.EXAMPLE.COM" > certificate_request.csr
```

3.

메시지가 표시되면 **certutil** 을 사용하여 임시 데이터베이스를 만들 때 입력한 것과 동일한 암호를 입력합니다. 그런 다음 중지하도록 지시할 때까지 **randomly**를 계속 입력합니다.

Enter Password or Pin for "NSS Certificate DB":

A random seed must be generated that will be used in the creation of your key. One of the easiest ways to create a random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter is full. DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!

Continue typing until the progress meter is full:

4.

인증서 요청 파일을 서버에 제출합니다. 새로 발급된 인증서와 연결할 **Kerberos** 사용자, 인증서를 저장할 출력 파일 및 선택적으로 인증서 프로필을 지정합니다. 인증서를 발행하려는 **IdM** 하위 **CA**를 지정합니다. 예를 들어, **webclient-ca** 에서 **idm_user@IDM.EXAMPLE.COM** 주체에 대해 추가한 사용자 역할 확장자가 있는 프로필이 있는 **registry UserRoles** 프로필의 인증서를 가져오고 **~/idm_user.pem** 파일에 인증서를 저장하려면 다음을 수행합니다.

```
# ipa cert-request certificate_request.csr --principal=idm_user@IDM.EXAMPLE.COM --
profile-id=IECUserRoles --ca=webclient-ca --certificate-out=~/idm_user.pem
```

5.

NSS 데이터베이스에 인증서를 추가합니다. **n** 옵션을 사용하여 인증서가 **NSS** 데이터베이스의 개인 키와 일치하도록 이전에 **CSR**을 만들 때 사용한 것과 동일한 앵커를 설정합니다. **t** 옵션은 신뢰 수준을 설정합니다. 자세한 내용은 **certutil(1)** 도움말 페이지를 참조하십시오. **i** 옵션은 입력 인증서 파일을 지정합니다. 예를 들어 **NSS** 데이터베이스에 **~/certdb/** 데이터베이스의 **~/idm_user.pem** 파일에 저장된 **idm_user** 닉네임 인증서를 추가하려면 다음을 수행합니다.

```
# certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

6.

NSS 데이터베이스의 키가 쿼리로 표시되지 않는지 확인합니다. 예를 들어 **~/certdb/** 데이터베이스에 저장된 인증서가 분리되지 않았는지 확인하려면 다음을 수행합니다.

```
# certutil -K -d ~/certdb/
< 0> rsa      5ad14d41463b87a095b1896cf0068ccc467df395  NSS Certificate
DB:idm_user
```

7.

pk12util 명령을 사용하여 **NSS** 데이터베이스에서 **PKCS12** 형식으로 인증서를 내보냅니다. 예를 들어 **/root/certdb** **NSS** 데이터베이스에서 **idm_user** 닉네임이 있는 인증서를 **~/idm_user.p12** 파일로 내보내려면 다음을 수행합니다.

```
# pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

8.

idm_user에 대한 인증서 인증을 활성화할 호스트에 인증서를 전송합니다.

```
# scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

9.

인증서가 전송된 호스트에서 보안상의 이유로 **.pkcs12** 파일이 **'other'** 그룹에 액세스할 수 없

는 디렉토리를 만듭니다.

```
# chmod o-rwx /home/idm_user/
```

10.

보안상의 이유로 임시 **NSS** 데이터베이스와 **.pkcs12** 파일을 서버에서 제거합니다.

```
# rm ~/certdb/  
# rm ~/idm_user.p12
```

64.12. 인증서 인증을 활성화하도록 브라우저 구성

WebUI를 사용하여 **IdM(Identity Management)**에 로그인할 때 인증서로 인증하려면 사용자 및 관련 **CA(인증 기관)** 인증서를 **Mozilla Firefox** 또는 **Google Chrome** 브라우저로 가져와야 합니다. 브라우저가 실행 중인 호스트 자체는 **IdM** 도메인의 일부일 필요는 없습니다.

IdM은 웹 UI에 연결하기 위해 다음 브라우저를 지원합니다.

- **Mozilla Firefox 38 이상**
- **Google Chrome 46 이상**

다음 절차에서는 **Mozilla Firefox 57.0.1** 브라우저를 구성하는 방법을 보여줍니다.

사전 요구 사항

- **PKCS#12** 형식으로 브라우저에서 가져올 **사용자 인증서**가 있습니다.
- **하위 CA 인증서를 다운로드** 하여 **PEM** 형식으로 활용할 수 있습니다.

절차

1.

Firefox를 열고 환경 설정 → 개인 정보 보호 및 보안 으로 이동합니다.

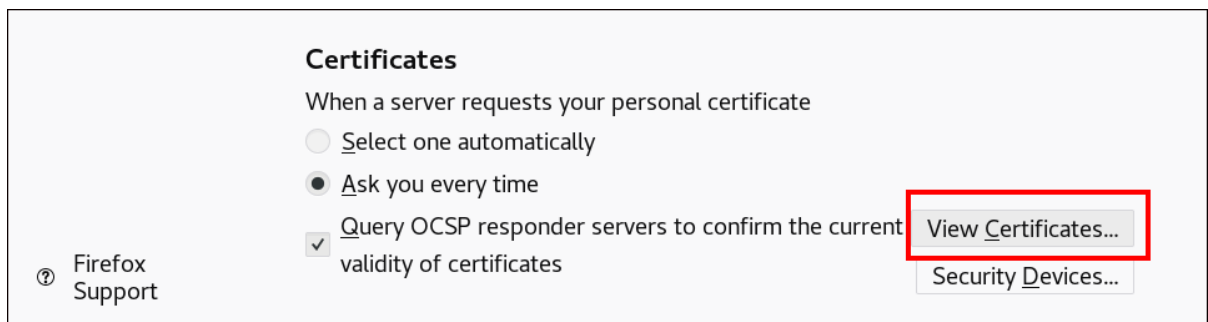
그림 64.8. 환경 설정의 개인 정보 보호 및 보안 섹션



2.

View Certificates (인증서 보기)를 클릭합니다.

그림 64.9. 개인 정보 보호 및 보안에서 인증서보기



3.

Certificates(인증서) 탭에서 **Import**(가져오기)를 클릭합니다. **PKCS12** 형식으로 사용자 인증서를 찾아 연 다음 **OK**(확인) 및 **OK**(확인)를 클릭합니다.

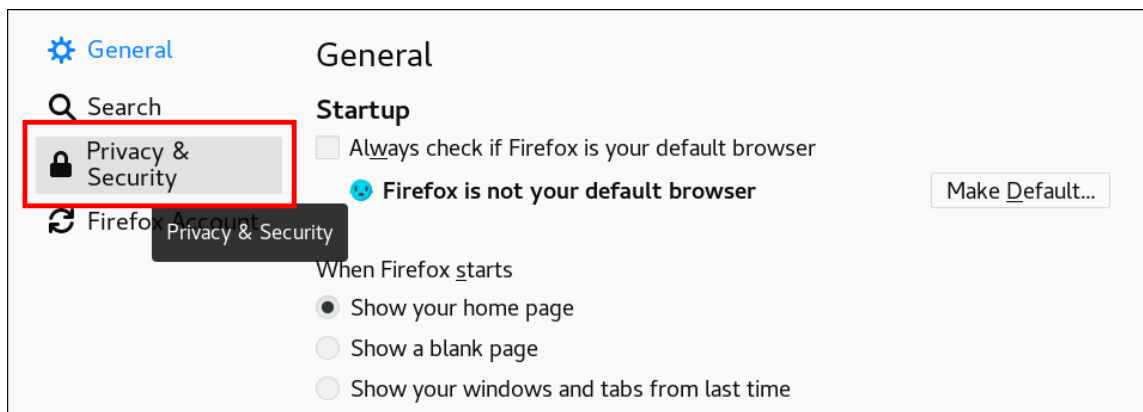
4.

Firefox에서 **IdM** 하위 **CA**가 신뢰할 수 있는 기관으로 인식되는지 확인하려면 64.2절. “**IdM WebUI**에서 하위 **CA** 인증서 다운로드”에 저장한 **IdM** 하위 **CA** 인증서를 신뢰할 수 있는 인증 기관 인증서로 가져옵니다.

a.

Firefox를 열고 **Preferences**(기본 설정)로 이동한 다음 **Privacy & Security**(개인 정보 보호 및 보안)를 클릭합니다.

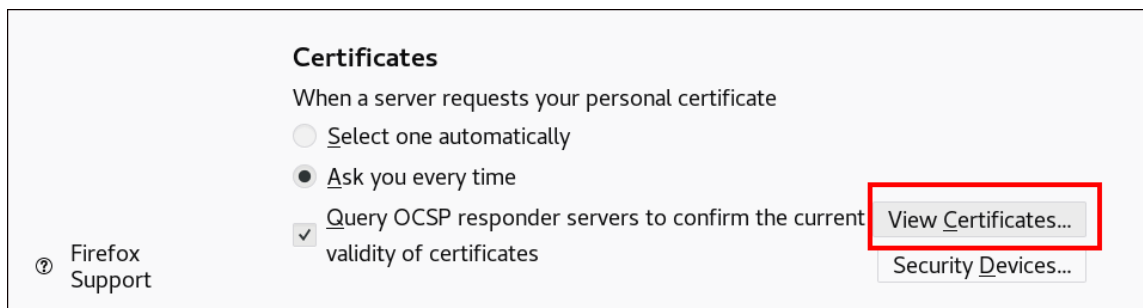
그림 64.10. 환경 설정의 개인 정보 보호 및 보안 섹션



b.

View Certificates (인증서 보기)를 클릭합니다.

그림 64.11. 개인 정보 보호 및 보안에서 인증서보기



c.

Authorities(권한) 탭에서 **Import**(가져오기)를 클릭합니다. 하위 **CA** 인증서를 찾아 엽니다. 인증서를 신뢰하여 웹사이트를 식별한 다음 **OK**(확인) 및 **OK**(확인)를 클릭합니다.

65장. 관련 인증서의 특정 그룹 무효화

시스템 관리자는 특정 인증서 그룹을 신속하게 무효화할 수 있도록 하려면 다음을 수행합니다.

- 특정 경량 **IdM(Identity Management)** 하위 **CA**에서 발급한 인증서만 신뢰하도록 애플리케이션을 설계합니다. 나중에 이러한 인증서를 발급한 **IdM(Identity Management)** 하위 **CA**의 인증서만 취소하여 이러한 모든 인증서를 무효화할 수 있습니다. **IdM**에서 경량 하위 **CA**를 생성하고 사용하는 방법에 대한 자세한 내용은 **특정 관련 인증서 그룹 인증을 빠르게 참조하십시오**.
- 호출되는 **IdM** 하위 **CA**에서 발급한 모든 인증서가 즉시 유효하지 않도록 하려면 **IdM OCSP** 응답자를 사용하도록 이러한 인증서에 의존하는 애플리케이션을 구성합니다. 예를 들어 **OCSP** 응답자를 사용하도록 **Firefox** 브라우저를 구성하려면 **Firefox** 환경 설정에서 인증서의 현재 유효성을 확인하도록 쿼리 **OCSP** 응답자 서버가 있는지 확인합니다.

IdM에서 **CRL(인증서 취소 목록)**은 4시간마다 업데이트됩니다.

IdM 하위 **CA**에서 발급한 모든 인증서를 무효화하려면 **IdM 하위 CA 인증서를 취소합니다**. 또한 **관련 CA ACL**을 비활성화하고 **IdM 하위 CA**를 비활성화하는 것이 좋습니다. 하위 **CA**를 비활성화하면 하위 **CA**가 새 인증서를 발행할 수 없지만 하위 **CA**의 서명 키가 유지되므로 이전에 발행된 인증서에 대해 **OCSP(Online Certificate Status Protocol)** 응답을 생성할 수 있습니다.

중요

환경에서 **OCSP**를 사용하는 경우 하위 **CA**를 삭제하지 마십시오. 하위 **CA**를 삭제하면 하위 **CA**의 서명 키가 삭제되므로 해당 하위 **CA**에서 발급한 인증서에 대한 **OCSP** 응답 실행을 방지할 수 있습니다.

하위 **CA**를 삭제하는 유일한 시나리오는 동일한 **DN(Subject 고유 이름)**이지만 새 서명 키로 새 하위 **CA**를 생성하려는 경우입니다.

65.1. IDm CLI에서 CA ACL 비활성화

IdM 서비스 또는 **IdM** 서비스 그룹을 폐기하려면 기존 해당 **CA ACL**을 비활성화하는 것이 좋습니다.

이 섹션을 완료하여 **IdM** 클라이언트에서 실행되는 웹 서버를 제한하는 **TLS_web_server_authentication CA ACL**을 비활성화하여 **webserver-ca IdM** 하위 **CA**에서 발행할 인증서를 요청하고, **IdM** 사용자가 **web client-ca IdM** 하위 **CA**에서 발행할 사용자 인증서를 요청하도록 제한하는 **TLS_web_client_authentication CA ACL**을 비활성화합니다.

절차

1. 선택적으로 IdM 환경의 모든 CA ACL을 보려면 **ipa caacl-find** 명령을 입력합니다.

```
$ ipa caacl-find
-----
3 CA ACLs matched
-----
ACL name: hosts_services_calPAserviceCert
Enabled: TRUE

ACL name: TLS_web_server_authentication
Enabled: TRUE

ACL name: TLS_web_client_authentication
Enabled: TRUE
```

2. 선택적으로 CA ACL의 세부 정보를 보려면 **ipa caacl-show** 명령을 입력하고 CA ACL 이름을 지정합니다.

```
$ ipa caacl-show TLS_web_server_authentication
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
Enabled: TRUE
CAs: webserver-ca
Profiles: calPAserviceCert
Services: HTTP/rhel8server.idm.example.com@IDM.EXAMPLE.COM
```

3. CA ACL을 비활성화하려면 **ipa caacl-disable** 명령을 입력하고 CA ACL 이름을 지정합니다.

- TLS_web_server_authentication CA ACL을 비활성화하려면 다음을 입력합니다.

```
$ ipa caacl-disable TLS_web_server_authentication
-----
Disabled CA ACL "TLS_web_server_authentication"
-----
```

- TLS_web_client_authentication CA ACL을 비활성화하려면 다음을 입력합니다.

```
$ ipa caacl-disable TLS_web_client_authentication
-----
Disabled CA ACL "TLS_web_client_authentication"
-----
```

이제 활성화된 유일한 CA ACL이 **hosts_services_calPAserviceCert** CA ACL입니다.



중요

hosts_services_calPAserviceCert CA ACL을 비활성화하는 데 매우 주의하십시오. **ca IPAserviceCert** 프로필과 함께 **ipa** CA를 사용하여 **IdM** 서버를 사용하여 **IdM** 서버를 부여하지 않고 **hosts_services_ca IPAserviceCert** 를 비활성화하면 **IdM HTTP** 및 **LDAP** 인증서의 인증서 갱신이 실패합니다. 만료된 **IdM HTTP** 및 **LDAP** 인증서로 인해 **IdM** 시스템이 실패합니다.

65.2. IDM 하위 CA 비활성화

해당 하위 CA에서 발급한 모든 인증서를 무효화하기 위해 **IdM** 하위 CA 인증서를 취소한 후 더 이상 필요하지 않은 경우 **IdM** 하위 CA를 비활성화하는 것이 좋습니다. 나중에 하위 CA를 다시 활성화할 수 있습니다.

하위 CA를 비활성화하면 하위 CA가 새 인증서를 발행할 수 없지만 하위 CA의 서명 키가 유지되므로 이전에 발행된 인증서에 대해 **OCSP(Online Certificate Status Protocol)** 응답을 생성할 수 있습니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.

절차

- **ipa ca-disable** 명령을 입력하고 **sub-CA**의 이름을 지정합니다.

```
$ ipa ca-disable webserver-CA
```

```
-----  
Disabled CA "webserver-CA"  
-----
```


66장. IDM의 자격 증명 모음

이 장에서는 **IdM(Identity Management)**의 자격 증명 모음에 대해 설명합니다. 다음 주제를 소개합니다.

- 자격 증명 모음의 개념.
- 자격 증명 모음과 관련된 다양한 역할.
- 보안 및 액세스 제어 수준에 따라 **IdM**에서 사용할 수 있는 다양한 유형의 자격 증명 모음입니다.
- 소유권을 기반으로 **IdM**에서 사용할 수 있는 다양한 유형의 자격 증명 모음입니다.
- 자격 증명 모음 컨테이너의 개념.
- **IdM**에서 자격 증명 모음을 관리하는 기본 명령입니다.
- **IdM**에서 자격 증명 모음을 사용하기 위한 전제 조건인 **KRA**(키 복구 기관) 설치.

66.1. 자격 증명 모음 및 이점

자격 증명 모음은 중요한 모든 데이터를 한 곳에서 안전하게 저장하지만 편리하게 저장하려는 **IdM(Identity Management)** 사용자에게 유용합니다. 이 섹션에서는 다양한 유형의 자격 증명 모음과 용도와 요구 사항에 따라 선택해야 하는 자격 증명 모음에 대해 설명합니다.

자격 증명 모음은 비밀 저장, 검색, 공유 및 복구를 위한 안전한 위치(**IdM**)입니다. 시크릿은 제한된 사용자 또는 엔터티 그룹만 액세스할 수 있는 보안에 민감한 데이터(일반적으로 인증 자격 증명)입니다. 예를 들어 보안에는 다음이 포함됩니다.

- 암호

- **PIN**

- **개인 SSH 키**

자격 증명 모음은 암호 관리자와 유사합니다. 암호 관리자와 마찬가지로 자격 증명 모음은 일반적으로 사용자가 자격 증명 모음에 저장된 정보를 잠금 해제하고 액세스하기 위해 하나의 기본 암호를 생성하고 기억해야 합니다. 그러나 사용자는 표준 자격 증명 모음을 보유하도록 결정할 수도 있습니다. 표준 자격 증명 모음에서는 사용자가 자격 증명 모음에 저장된 시크릿에 액세스하기 위해 암호를 입력할 필요가 없습니다.

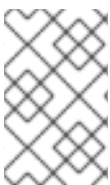


참고

IdM에서 자격 증명 모음의 목적은 외부의 **IdM**이 아닌 서비스에 인증할 수 있는 인증 자격 증명을 저장하는 것입니다.

IdM 자격 증명 모음의 기타 중요한 특성은 다음과 같습니다.

- 자격 증명 모음 소유자는 자격 증명 모음 소유자와 자격 증명 모음 소유자가 자격 증명 모음 구성원으로 선택한 해당 **IdM** 사용자만 액세스할 수 있습니다. 또한 **IdM** 관리자는 자격 증명 모음에 액세스할 수 있습니다.
- 사용자에게 자격 증명 모음을 생성하는 데 충분한 권한이 없는 경우 **IdM** 관리자는 자격 증명 모음을 생성하고 사용자를 소유자로 설정할 수 있습니다.
- 사용자 및 서비스는 **IdM** 도메인에 등록된 시스템에서 자격 증명 모음에 저장된 시크릿에 액세스할 수 있습니다.
- 하나의 자격 증명 모음은 하나의 비밀(예: 파일 한 개)만 포함할 수 있습니다. 그러나 파일 자체에는 암호, 키값 또는 인증서와 같은 여러 시크릿이 포함될 수 있습니다.



참고

Vault는 **IdM** 웹 UI가 아닌 **IdM CLI**(명령줄)에서만 사용할 수 있습니다.

66.2. VAULT 소유자, 멤버 및 관리자

IdM(Identity Management)은 다음 자격 증명 모음 사용자 유형을 구분합니다.

Vault 소유자

자격 증명 모음 소유자는 자격 증명 모음에 대한 기본 관리 권한이 있는 사용자 또는 서비스입니다. 예를 들어, 자격 증명 모음 소유자는 자격 증명 모음의 속성을 수정하거나 새 자격 증명 모음 멤버를 추가할 수 있습니다.

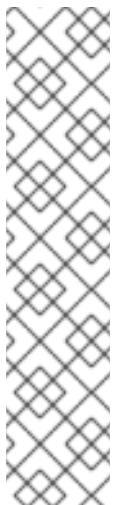
각 자격 증명 모음에는 소유자가 하나 이상 있어야 합니다. 자격 증명 모음에는 여러 소유자가 있을 수도 있습니다.

Vault 멤버

자격 증명 모음 멤버는 다른 사용자 또는 서비스에서 만든 자격 증명 모음에 액세스할 수 있는 사용자 또는 서비스입니다.

Vault 관리자

Vault 관리자는 모든 자격 증명 모음에 대한 무제한 액세스 권한을 가지며 모든 자격 증명 모음 작업을 수행할 수 있습니다.



참고

대칭 및 비대칭 자격 증명 모음은 암호 또는 키로 보호되며 특수 액세스 제어 규칙을 적용합니다([Vault 유형](#) 참조). 관리자는 다음을 위해 다음 규칙을 충족해야 합니다.

- 대칭 및 비대칭 자격 증명 모음의 액세스 보안.
- **vault** 암호 또는 키를 변경하거나 재설정합니다.

자격 증명 모음 관리자는 **Vault** 관리자 권한이 있는 모든 사용자입니다. **IdM**의 역할 기반 액세스 제어(**RBAC**)의 컨텍스트에서 권한은 역할에 적용할 수 있는 권한 그룹입니다.

Vault 사용자

자격 증명 모음 사용자는 자격 증명 모음에 있는 컨테이너에 있는 사용자를 나타냅니다. **Vault** 사용자 정보는 **ipa vault-show** 와 같은 특정 명령의 출력에 표시됩니다.

```
$ ipa vault-show my_vault
```

```

Vault name: my_vault
Type: standard
Owner users: user
Vault user: user

```

vault 컨테이너 및 사용자 자격 증명 모음에 대한 자세한 내용은 [Vault 컨테이너](#)를 참조하십시오.

추가 리소스

- 특정 소유자 및 멤버 권한은 자격 증명 모음 유형에 따라 다릅니다. 자세한 내용은 [Standard, 대칭 및 비대칭 자격 증명 모음](#) 을 참조하십시오.

66.3. 표준, 대칭 및 비대칭 자격 증명 모음

보안 및 액세스 제어 수준에 따라 **IdM**은 자격 증명 모음을 다음 유형으로 분류합니다.

표준 자격 증명 모음

Vault 소유자 및 자격 증명 모음 멤버는 암호나 키를 사용하지 않고도 시크릿을 보관하고 검색할 수 있습니다.

대칭 자격 증명 모음

자격 증명 모음의 비밀은 대칭 키로 보호됩니다. **Vault** 소유자와 멤버는 시크릿을 보관하고 검색할 수 있지만 **vault** 암호를 제공해야 합니다.

비대칭 자격 증명 모음

자격 증명 모음의 비밀은 비대칭 키로 보호됩니다. 사용자는 공개 키를 사용하여 비밀을 보관하고 개인 키를 사용하여 검색합니다. **Vault** 멤버는 비밀만 보관할 수 있지만, 자격 증명 모음 소유자는 시크릿을 모두 수행할 수 있으며, 시크릿을 보관하고 검색할 수 있습니다.

66.4. 사용자, 서비스 및 공유 자격 증명 모음

IdM은 소유권에 따라 자격 증명을 여러 유형으로 분류합니다. [아래 테이블](#)에 는 각 유형, 해당 소유자 및 사용에 대한 정보가 포함되어 있습니다.

표 66.1. 소유권 기반 **IdM** 자격 증명 모음

유형	설명	소유자	참고
사용자 자격 증명 모음	사용자의 개인 자격 증명 모음	단일 사용자	모든 사용자는 IdM 관리자가 허용한 경우 하나 이상의 사용자 자격 증명 모음을 소유할 수 있습니다.
서비스 자격 증명 모음	서비스의 개인 자격 증명 모음	단일 서비스	모든 서비스는 IdM 관리자가 허용한 경우 하나 이상의 사용자 자격 증명 모음을 소유할 수 있습니다.
공유 자격 증명 모음	여러 사용자 및 서비스에 공유하는 자격 증명 모음	자격 증명 모음을 만든 자격 증명 모음 관리자	IdM 관리자가 허용하는 경우 사용자와 서비스는 하나 이상의 사용자 자격 증명 모음을 소유할 수 있습니다. 자격 증명 모음을 만든 관리자 이외의 자격 증명 모음 관리자에게도 자격 증명 모음에 대한 전체 액세스 권한이 있습니다.

66.5. VAULT 컨테이너

자격 증명 모음 컨테이너는 자격 증명 모음입니다. [아래 표](#)에 는 IdM(Identity Management)에서 제공하는 기본 자격 증명 모음 컨테이너가 나열되어 있습니다.

표 66.2. IdM의 기본 자격 증명 모음 컨테이너

유형	설명	목적
사용자 컨테이너	사용자의 개인 컨테이너	특정 사용자의 사용자 자격 증명 모음 저장
서비스 컨테이너	서비스용 개인 컨테이너	특정 서비스에 대한 서비스 자격 증명 모음 저장
공유 컨테이너	여러 사용자 및 서비스의 컨테이너	여러 사용자 또는 서비스에서 공유할 수 있는 자격 증명 모음을 저장합니다.

IdM은 사용자 또는 서비스의 첫 번째 개인 자격 증명 모음이 생성될 때 각 사용자 또는 서비스에 대한 사용자 및 서비스 컨테이너를 자동으로 생성합니다. 사용자 또는 서비스를 삭제한 후 IdM은 컨테이너 및 해당 콘텐츠를 제거합니다.

66.6. 기본 IDM 자격 증명 모음 명령

이 섹션에서는 IdM(Identity Management) 자격 증명 모음을 관리하는 데 사용할 수 있는 기본 명령에 대해 설명합니다. [아래 표](#)에 는 목적의 설명과 함께 `ipa vault-*` 명령 목록이 포함되어 있습니다.



참고

ipa vault-* 명령을 실행하기 전에 IdM 도메인의 하나 이상의 서버에 KRA(Key Recovery Authority) 인증서 시스템 구성 요소를 설치합니다. 자세한 내용은 [IdM에 키 복구 권한](#) 설치를 참조하십시오.

표 66.3. 설명과 함께 기본 IdM 자격 증명 모음 명령

명령	목적
IPA 도움말 자격 증명 모음	IdM 자격 증명 모음 및 샘플 자격 증명 모음 명령에 대한 개념 정보를 표시합니다.
ipa vault-add --help, ipa vault-find --help	특정 ipa vault-* 명령에 --help 옵션을 추가하면 해당 명령에 사용할 수 있는 자세한 도움말과 옵션이 표시됩니다.
ipa vault-show user_vault --user idm_user	자격 증명 모음 멤버로 자격 증명 모음에 액세스하는 경우 자격 증명 모음 소유자를 지정해야 합니다. 자격 증명 모음 소유자를 지정하지 않으면 IdM에서 자격 증명 모음을 찾지 못했음을 알려줍니다. <pre>[admin@server ~]\$ ipa vault-show user_vault ipa: ERROR: user_vault: vault not found</pre>
ipa vault-show shared_vault --shared	공유 자격 증명 모음에 액세스할 때 액세스하려는 자격 증명 모음이 공유 자격 증명 모음임을 지정해야 합니다. 그러지 않으면 IdM에서 자격 증명 모음을 찾지 못했음을 알려줍니다. <pre>[admin@server ~]\$ ipa vault-show shared_vault ipa: ERROR: shared_vault: vault not found</pre>

66.7. IdM에 키 복구 권한 설치

이 섹션에서는 **KRA(키 복구 기관) 인증서 시스템(CS)** 구성 요소를 설치하여 **IdM(Identity Management)**에서 자격 증명 모음을 활성화하는 방법을 설명합니다.

사전 요구 사항

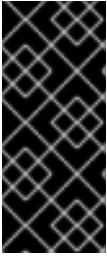
- **IdM 관리자로 로그인했습니다.**
- **IdM 클라이언트에서 root로 로그인했습니다.**

절차

•

KRA를 설치합니다.

ipa-kra-install



중요

숨겨진 복제본에 **IdM** 클러스터의 첫 번째 **KRA**를 설치할 수 있습니다. 그러나 추가 **KRA**를 설치하려면 숨기지 않은 복제본에 **KRA** 복제본을 설치하기 전에 임시로 숨겨진 복제본을 활성화해야 합니다. 그런 다음 원래 숨겨진 복제본을 다시 숨길 수 있습니다.



참고

자격 증명 모음 서비스를고가용성으로 설정하려면 두 개의 **IdM** 서버에 **KRA**를 설치합니다.

추가 리소스

•

IdM 복제본을 활성화하는 방법 및 숨기는 방법에 대한 자세한 내용은 [숨겨진 복제본 데모 또는 승격을 참조하십시오](#).

•

IdM의 숨겨진 복제본에 대한 자세한 내용은 [숨겨진 복제 모드를 참조하십시오](#).

67장. IDM 사용자 자격 증명 모음 사용: 시크릿 저장 및 검색

이 장에서는 ID 관리에서 사용자 자격 증명 모음을 사용하는 방법에 대해 설명합니다. 특히 사용자가 **IdM** 자격 증명 모음에 비밀을 저장하는 방법과 사용자가 이를 검색하는 방법을 설명합니다. 사용자는 두 개의 서로 다른 **IdM** 클라이언트에서 저장 및 검색을 수행할 수 있습니다.

사전 요구 사항

- **IdM** 도메인에 있는 하나 이상의 서버에 **KRA(Key Recovery Authority)** 인증 시스템 구성 요소가 설치되었습니다. 자세한 내용은 [IdM에 키 복구 권한 설치](#)를 참조하십시오.

67.1. 사용자 자격 증명 모음에 시크릿 저장

이 섹션에서는 사용자가 중요한 정보로 파일을 안전하게 저장하기 위해 하나 이상의 개인 자격 증명 모음으로 자격 증명 모음 컨테이너를 만드는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **idm_user** 사용자는 표준 유형의 자격 증명 모음을 생성합니다. 표준 자격 증명 모음 유형을 사용하면 파일에 액세스 할 때 **idm_user** 가 인증할 필요가 없습니다. **idm_user** 는 사용자가 로그인한 모든 **IdM** 클라이언트에서 파일을 검색할 수 있습니다.

절차에서 다음을 수행합니다.

- **idm_user** 는 자격 증명 모음을 생성하려는 사용자입니다.
- **my_vault** 는 사용자의 인증서를 저장하는 데 사용되는 자격 증명 모음입니다.
- 보관된 인증서에 액세스하기 위해 사용자가 **vault** 암호를 제공할 필요가 없도록 자격 증명 모음 유형은 표준입니다.
- **secret.txt** 는 사용자가 자격 증명 모음에 저장하려는 인증서를 포함하는 파일입니다.

사전 요구 사항

- **idm_user** 의 암호를 알고 있습니다.
- **IdM** 클라이언트인 호스트에 로그인되어 있습니다.

절차

1.

idm_user에 대한 **Kerberos** 티켓 부여 티켓(TGT)을 획득하십시오.

```
$ kinit idm_user
```

2.

ipa vault-add 명령을 **--type** 표준 옵션과 함께 사용하여 표준 자격 증명 모음을 생성합니다.

```
$ ipa vault-add my_vault --type standard
```

```
-----  
Added vault "my_vault"  
-----
```

```
Vault name: my_vault
```

```
Type: standard
```

```
Owner users: idm_user
```

```
Vault user: idm_user
```

중요

사용자에 대한 첫 번째 사용자 자격 증명 모음이 동일한 사용자가 생성되었는지 확인합니다. 사용자의 첫 번째 자격 증명 모음을 생성하면 사용자의 자격 증명 모음 컨테이너도 생성됩니다. 생성 에이전트는 자격 증명 모음 컨테이너의 소유자가 됩니다.

예를 들어 **admin**과 같은 다른 사용자가 **user1**에 대한 첫 번째 사용자 자격 증명 모음을 만드는 경우 사용자의 자격 증명 모음 컨테이너 소유자도 **admin**이 되고 **user1**은 사용자 자격 증명 모음에 액세스하거나 새 사용자 자격 증명 모음을 생성할 수 없게 됩니다.

3.

ipa vault-archive 명령을 **--in** 옵션과 함께 사용하여 **secret.txt** 파일을 자격 증명 모음에 보관합니다.

```
$ ipa vault-archive my_vault --in secret.txt
```

```
-----  
Archived data into vault "my_vault"  
-----
```

67.2. 사용자 자격 증명 모음에서 시크릿 검색

IdM(Identity Management)으로 로그인한 모든 **IdM** 클라이언트에 개인 자격 증명 모음의 시크릿을 검색할 수 있습니다.

이 섹션에서는 **idm_user**라는 IdM 사용자로 **my_vault**라는 사용자 개인 자격 증명 모음의 시크릿을 **idm_client.idm.example.com** 으로 검색하는 방법을 보여줍니다.

사전 요구 사항

- **idm_user** 는 **my_vault** 의 소유자입니다.
- **idm_user** 는 [자격 증명 모음에 비밀을 보관합니다](#).
- **my_vault** 는 표준 자격 증명 모음이므로 **idm_user** 가 자격 증명 모음의 콘텐츠에 액세스하기 위해 암호를 입력할 필요가 없음을 의미합니다.

절차

1. **idm_client**에 **idm_user**로 **ssh** 를 실행하십시오.

```
$ ssh idm_user@idm_client.idm.example.com
```

2. **idm_user** 로 로그인합니다.

```
$ kinit user
```

3. **ipa vault-retmasterve --out** 명령을 **--out** 옵션과 함께 사용하여 자격 증명 모음의 내용을 검색하고 **secret_exported.txt** 파일에 저장합니다.

```
$ ipa vault-retrieve my_vault --out secret_exported.txt
```

```
-----
Retrieved data from vault "my_vault"
-----
```

추가 리소스

- **Ansible**을 사용하여 IdM 사용자 자격 증명 모음 관리 프로세스를 자동화할 수 있습니다. 자세한 내용은 [Ansible을 사용하여 IdM 사용자 자격 증명 모음 관리: 시크릿 저장 및 검색을 참조하십시오](#).

68장. ANSIBLE을 사용하여 IDM 사용자 자격 증명 모음 관리: 시크릿 저장 및 검색

이 장에서는 **Ansible vault** 모듈을 사용하여 ID 관리에서 사용자 자격 증명 모음을 관리하는 방법을 설명합니다. 특히 사용자가 **Ansible** 플레이북을 사용하여 다음 세 가지 연속 작업을 수행하는 방법을 설명합니다.

- **IdM에서 사용자 자격 증명 모음을 생성합니다.**
- **자격 증명 모음에 비밀을 저장합니다.**
- **자격 증명 모음에서 비밀을 검색합니다.**

사용자는 두 개의 서로 다른 **IdM** 클라이언트에서 저장 및 검색을 수행할 수 있습니다.

사전 요구 사항

- **IdM 도메인에 있는 하나 이상의 서버에 KRA(Key Recovery Authority) 인증 시스템 구성 요소가 설치되었습니다.** 자세한 내용은 [IdM에 키 복구 권한 설치](#)를 참조하십시오.

68.1. ANSIBLE을 사용하여 IDM에 표준 사용자 자격 증명 모음이 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 사용자가 **Ansible** 플레이북을 사용하여 하나 이상의 개인 자격 증명 모음으로 자격 증명 모음 컨테이너를 만들어 중요한 정보를 안전하게 저장하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **idm_user** 사용자는 **my_vault** 라는 표준 유형의 자격 증명 모음을 생성합니다. 표준 자격 증명 모음 유형을 사용하면 파일에 액세스할 때 **idm_user** 가 인증할 필요가 없습니다. **idm_user** 는 사용자가 로그인한 모든 **IdM** 클라이언트에서 파일을 검색할 수 있습니다.

사전 요구 사항

- 프로세스의 단계를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **idm_user** 의 암호를 알고 있습니다.

절차

1.

`/usr/share/doc/ansible-freeipa/playbooks/vault` 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2.

인벤토리 파일을 생성합니다(예: **`inventory.file`**):

```
$ touch inventory.file
```

3.

`inventory.file` 을 열고 **`[ipaserver]`** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **`server.idm.example.com`**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

4.

`ensure-standard-vault-is-present.yml` **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-standard-vault-is-present.yml ensure-standard-vault-is-present-copy.yml
```

5.

편집을 위해 **`ensure-standard-vault-is-present-copy.yml`** 파일을 엽니다.

6.

`ipavault` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **`ipaadmin_principal`** 변수를 **`idm_user`** 로 설정합니다.
- **`ipaadmin_password`** 변수를 **`idm_user`** 의 암호로 설정합니다.
- 사용자 변수를 **`idm_user`** 로 설정합니다.
- **`name`** 변수를 **`my_vault`** 로 설정합니다.
- **`vault_type`** 변수를 **`standard`** 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
    - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      vault_type: standard
```

7. 파일을 저장합니다.

8. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-standard-vault-is-present-copy.yml
```

68.2. ANSIBLE을 사용하여 IDM의 표준 사용자 자격 증명 모음에 비밀 보관

이 섹션에서는 **IdM(Identity Management)** 사용자가 **Ansible** 플레이북을 사용하여 개인 자격 증명 모음에 중요한 정보를 저장하는 방법을 보여줍니다. 사용된 예제에서 **idm_user** 사용자는 **my_vault** 라는 자격 증명 모음에 **password.txt** 라는 중요한 정보로 파일을 보관합니다.

사전 요구 사항

- 프로세스의 단계를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- idm_user** 의 암호를 알고 있습니다.
- idm_user** 는 소유자이거나 최소 **my_vault** 멤버 사용자입니다.
- my_vault** 에 보관하려는 시크릿인 **password.txt** 에 액세스할 수 있습니다.

절차

1. **`/usr/share/doc/ansible-freeipa/playbooks/vault` 디렉터리로 이동합니다.**

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. **인벤토리 파일을 열고 구성할 IdM 서버가 `[ipaserver]` 섹션에 나열되어 있는지 확인합니다. 예를 들어 `server.idm.example.com`을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.**

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

3. **`data-archive-in-symmetric-vault.yml` Ansible 플레이북 파일의 복사본을 만들지만 "대칭"을 "표준"으로 바꿉니다. 예를 들면 다음과 같습니다.**

```
$ cp data-archive-in-symmetric-vault.yml data-archive-in-standard-vault-copy.yml
```

4. **편집을 위해 `data-archive-in-standard-vault-copy.yml` 파일을 엽니다.**

5. **`ipavault` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.**

- **`ipaadmin_principal` 변수를 `idm_user` 로 설정합니다.**
- **`ipaadmin_password` 변수를 `idm_user` 의 암호로 설정합니다.**
- **사용자 변수를 `idm_user` 로 설정합니다.**
- **`name` 변수를 `my_vault` 로 설정합니다.**
- **중요한 정보를 사용하여 변수의 전체 경로로 설정합니다.**
- **`action` 변수를 `member` 로 설정합니다.**

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
    - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      in: /usr/share/doc/ansible-freeipa/playbooks/vault/password.txt
      action: member
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file data-archive-in-standard-vault-copy.yml
```

68.3. ANSIBLE을 사용하여 IDM의 표준 사용자 자격 증명 모음에서 시크릿 검색

이 섹션에서는 **IdM(Identity Management)** 사용자가 **Ansible** 플레이북을 사용하여 사용자 개인 자격 증명 모음에서 시크릿을 검색하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **idm_user** 사용자는 **Ansible**이 설치된 **IdM** 클라이언트에 **my_vault** 라는 표준 유형의 자격 증명 모음에서 중요한 데이터가 있는 파일을 검색합니다. **idm_user** 는 파일에 액세스할 때 인증할 필요가 없습니다. **idm_user** 는 **Ansible**을 사용하여 **Ansible**이 설치된 **IdM** 클라이언트에서 파일을 검색할 수 있습니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **idm_user** 의 암호를 알고 있습니다.
- **idm_user** 는 **my_vault** 의 소유자입니다.

- *idm_user* 가 시크릿을 *my_vault*에 저장했습니다.
- *Ansible*은 비밀을 검색할 *IdM* 호스트의 디렉터리에 쓸 수 있습니다.
- *idm_user* 는 *IdM* 호스트의 디렉터리에서 시크릿을 검색할 수 있습니다.

절차

1. */usr/share/doc/ansible-freeipa/playbooks/vault* 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 인벤토리 파일을 열고 명확하게 정의된 섹션에서 시크릿을 검색하려는 *IdM* 클라이언트를 언급합니다. 예를 들어 *host01.idm.example.com*에서 비밀을 검색하도록 *Ansible*에 지시하려면 다음을 입력합니다.

```
[ipahost]
host01.idm.example.com
```

3. *retrive-data-symmetric-vault.yml* *Ansible* 플레이북 파일의 복사본을 만듭니다. "대칭"을 "표준"으로 바꿉니다. 예를 들면 다음과 같습니다.

```
$ cp retrive-data-symmetric-vault.yml retrieve-data-standard-vault.yml-copy.yml
```

4. 편집을 위해 *retrieve-data-standard-vault.yml-copy.yml* 파일을 엽니다.

5. *hosts* 변수를 *ipahost* 로 설정하여 파일을 조정합니다.

6. *ipavault* 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- *ipaadmin_principal* 변수를 *idm_user* 로 설정합니다.
- *ipaadmin_password* 변수를 *idm_user* 의 암호로 설정합니다.

- 사용자 변수를 **idm_user** 로 설정합니다.
- **name** 변수를 **my_vault** 로 설정합니다.
- 보안을 내보낼 파일의 전체 경로로 **out** 변수를 설정합니다.
- **state** 변수를 불러오도록 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Tests
  hosts: ipahost
  become: true
  gather_facts: false

  tasks:
    - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      out: /tmp/password_exported.txt
      state: retrieved
```

7. 파일을 저장합니다.

8. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file retrieve-data-standard-vault.yml-copy.yml
```

검증 단계

1. **user01**로 **host01**에 SSH로 :

```
$ ssh user01@host01.idm.example.com
```

2. **Ansible** 플레이북 파일에서 **out** 변수로 지정한 파일을 확인합니다.

```
$ vim /tmp/password_exported.txt
```

이제 내보낸 보안이 표시됩니다.

- **Ansible**을 사용하여 **IdM** 자격 증명 모음 및 사용자 시크릿 및 플레이북 변수에 대한 자세한 내용은 `/usr/share/doc/ansible-freeipa/` 디렉터리에 사용 가능한 **README-vault.md** **Markdown** 파일과 `/usr/share/doc/ansible-freeipa/playbooks/vault/` 디렉터리에 있는 샘플 플레이북을 참조하십시오.

69장. IDM 서비스 시크릿 관리: 시크릿 저장 및 검색

이 섹션에서는 관리자가 **ansible-freeipa vault** 모듈을 사용하여 서비스 시크릿을 중앙 집중식 위치에 안전하게 저장하는 방법을 보여줍니다. 예제에서 사용되는 **자격 증명 모음**은 비대칭입니다. 즉, 관리자가 다음 단계를 수행해야 함을 의미합니다.

1. 예를 들어 **openssl** 유틸리티를 사용하여 개인 키를 생성합니다.
2. 개인 키를 기반으로 공개 키를 생성합니다.

서비스 비밀은 관리자가 자격 증명 모음에 보관할 때 공개 키로 암호화됩니다. 이후 도메인의 특정 시스템에서 호스팅되는 서비스 인스턴스는 개인 키를 사용하여 비밀을 검색합니다. 서비스와 관리자만 비밀에 액세스할 수 있습니다.

보안이 손상되면 관리자는 서비스 자격 증명 모음에서 암호를 교체한 다음 손상되지 않은 개별 서비스 인스턴스에 재배포할 수 있습니다.

사전 요구 사항

- **IdM** 도메인에 있는 하나 이상의 서버에 **KRA(Key Recovery Authority)** 인증 시스템 구성 요소가 설치되었습니다. 자세한 내용은 **IdM에 키 복구 권한** 설치를 참조하십시오.

이 섹션에는 이러한 절차가 포함되어 있습니다.

1. **비대칭 자격 증명 모음에 IdM 서비스 시크릿 저장**
2. **IdM 서비스 인스턴스의 서비스 시크릿 검색**
3. **손상된 경우 IdM 서비스 자격 증명 모음 시크릿 변경**

용어 사용

절차에서 다음을 수행합니다.

- **admin** 은 서비스 암호를 관리하는 관리자입니다.
- **private-key-to-an-externally-signed-certificate.pem** 은 서비스 비밀을 포함하는 파일입니다(이 경우 외부 서명 인증서에 대한 개인 키). 자격 증명 모음에서 비밀을 검색하는 데 사용되는 개인 키와 이 개인 키를 혼동하지 마십시오.
- **secret_vault** 는 서비스에 대해 생성된 자격 증명 모음입니다.
- **HTTP/webserver.idm.example.com** 은 비밀이 보관되는 서비스입니다.
- **service-public.pem** 은 **password_vault** 에 저장된 암호를 암호화하는 데 사용되는 서비스 공개 키입니다.
- **service-private.pem** 은 **secret_vault** 에 저장된 암호를 해독하는 데 사용되는 서비스 개인 키입니다.

69.1. 비대칭 자격 증명 모음에 IDM 서비스 시크릿 저장

이 섹션에서는 비대칭 자격 증명 모음을 생성하고 이를 사용하여 서비스 시크릿을 보관하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.

절차

1. 관리자로 로그인합니다.


```
$ kinit admin
```
2. 서비스 인스턴스의 공개 키를 가져옵니다. 예를 들어 **openssl** 유틸리티를 사용합니다.

a.

service-private.pem 개인 키를 생성합니다.

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....+++
e is 65537 (0x10001)
```

b.

개인 키를 기반으로 **service-public.pem** 공개 키를 생성합니다.

```
$ openssl rsa -in service-private.pem -out service-public.pem -pubout
writing RSA key
```

3.

서비스 인스턴스 자격 증명 모음으로 비대칭 자격 증명 모음을 생성하고 공개 키를 제공합니다.

```
$ ipa vault-add secret_vault --service HTTP/webserver.idm.example.com --type
asymmetric --public-key-file service-public.pem
-----
Added vault "secret_vault"
-----
Vault name: secret_vault
Type: asymmetric
Public key: LS0tLS1C...S0tLS0tCg==
Owner users: admin
Vault service: HTTP/webserver.idm.example.com@IDM.EXAMPLE.COM
```

자격 증명 모음에 보관된 암호는 키를 사용하여 보호합니다.

4.

서비스 시크릿을 서비스 자격 증명 모음에 보관합니다.

```
$ ipa vault-archive secret_vault --service HTTP/webserver.idm.example.com --in
private-key-to-an-externally-signed-certificate.pem
-----
Archived data into vault "secret_vault"
-----
```

그러면 서비스 인스턴스 공개 키를 사용하여 비밀을 암호화합니다.

시크릿이 필요한 모든 서비스 인스턴스에 대해 이러한 단계를 반복합니다. 각 서비스 인스턴스에 대해 새 비대칭 자격 증명 모음을 만듭니다.

69.2. IDM 서비스 인스턴스의 서비스 시크릿 검색

이 섹션에서는 서비스 인스턴스가 로컬 저장소 서비스 개인 키를 사용하여 서비스 자격 증명 모음 시크릿을 검색하는 방법에 대해 설명합니다.

사전 요구 사항

- 서비스 자격 증명 모음을 소유하는 서비스 주체의 키 탭(예: **HTTP/webserver.idm.example.com**)에 액세스할 수 있습니다.
- [비대칭 자격 증명 모음을 생성하고 자격 증명 모음에 시크릿을 보관](#) 했습니다.
- 서비스 자격 증명 모음 시크릿을 검색하는 데 사용되는 개인 키에 액세스할 수 있습니다.

절차

1. 관리자로 로그인합니다.

```
$ kinit admin
```

2. 서비스에 대한 **Kerberos** 티켓을 가져옵니다.

```
# kinit HTTP/webserver.idm.example.com -k -t /etc/httpd/conf/ipa.keytab
```

3. 서비스 **vault** 암호를 검색합니다.

```
$ ipa vault-retrieve secret_vault --service HTTP/webserver.idm.example.com --private-key-file service-private.pem --out secret.txt
```

```
-----
Retrieved data from vault "secret_vault"
-----
```

69.3. 손상된 경우 IDM 서비스 자격 증명 모음 시크릿 변경

이 섹션에서는 서비스 자격 증명 모음 시크릿을 변경하여 손상된 서비스 인스턴스를 격리하는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM 관리자 암호를 알고 있습니다.**
- **서비스 시크릿 을 저장할 비대칭 자격 증명 모음을 생성했습니다.**
- **새 시크릿을 생성하고 액세스할 수 있습니다(예: `new-private-key-to-an-externally-signed-certificate.pem` 파일에서).**

절차

1. **새 시크릿을 서비스 인스턴스 자격 증명 모음에 보관합니다.**

```
$ ipa vault-archive secret_vault --service HTTP/webserver.idm.example.com --in new-private-key-to-an-externally-signed-certificate.pem
```

```
-----  
Archived data into vault "secret_vault"  
-----
```

이렇게 하면 자격 증명 모음에 저장된 현재 비밀을 덮어씁니다.

2. **비합격 서비스 인스턴스에서만 새 시크릿을 검색합니다. 자세한 내용은 [IdM 서비스 인스턴스의 서비스 시크릿 검색](#)을 참조하십시오.**

추가 리소스

- **Ansible을 사용하여 IdM 서비스 자격 증명 모음 관리 프로세스를 자동화할 수 있습니다. 자세한 내용은 [Ansible을 사용하여 IdM 서비스 자격 증명 모음 관리: 시크릿 저장 및 검색](#)을 참조하십시오.**

70장. ANSIBLE을 사용하여 IDM 서비스 자격 증명 모음 관리: 시크릿 저장 및 검색

이 섹션에서는 관리자가 **ansible-freeipa vault** 모듈을 사용하여 서비스 시크릿을 중앙 집중식 위치에 안전하게 저장하는 방법을 보여줍니다. 예제에서 사용되는 **자격 증명 모음**은 비대칭입니다. 즉, 관리자가 다음 단계를 수행해야 함을 의미합니다.

1. 예를 들어 **openssl** 유틸리티를 사용하여 개인 키를 생성합니다.
2. 개인 키를 기반으로 공개 키를 생성합니다.

서비스 비밀은 관리자가 자격 증명 모음에 보관할 때 공개 키로 암호화됩니다. 이후 도메인의 특정 시스템에서 호스팅되는 서비스 인스턴스는 개인 키를 사용하여 비밀을 검색합니다. 서비스와 관리자만 비밀에 액세스할 수 있습니다.

보안이 손상되면 관리자는 서비스 자격 증명 모음에서 암호를 교체한 다음 손상되지 않은 개별 서비스 인스턴스에 재배포할 수 있습니다.

사전 요구 사항

- **IdM 도메인에 있는 하나 이상의 서버에 KRA(Key Recovery Authority) 인증 시스템 구성 요소가 설치되었습니다.** 자세한 내용은 **IdM에 키 복구 권한 설치**를 참조하십시오.

이 섹션에는 다음 절차가 포함됩니다.

- **Ansible을 사용하여 IdM에 비대칭 서비스 자격 증명 모음이 있는지 확인**
- **Ansible을 사용하여 비대칭 자격 증명 모음에 IdM 서비스 시크릿 저장**
- **Ansible을 사용하여 IdM 서비스의 서비스 시크릿 검색**
- **Ansible을 사용하여 손상된 경우 IdM 서비스 자격 증명 모음 시크릿 변경**

절차에서 다음을 수행합니다.

- **admin** 은 서비스 암호를 관리하는 관리자입니다.
- **private-key-to-an-externally-signed-certificate.pem** 은 서비스 비밀을 포함하는 파일입니다(이 경우 외부 서명 인증서에 대한 개인 키). 자격 증명 모음에서 비밀을 검색하는 데 사용되는 개인 키와 이 개인 키를 혼동하지 마십시오.
- **secret_vault** 는 서비스 시크릿을 저장하기 위해 생성된 자격 증명 모음입니다.
- **HTTP/webserver1.idm.example.com** 은 자격 증명 모음의 소유자인 서비스입니다.
- **HTTP/webserver2.idm.example.com** 및 **HTTP/webserver3.idm.example.com** 은 자격 증명 모음 멤버 서비스입니다.
- **service-public.pem** 은 **password_vault** 에 저장된 암호를 암호화하는 데 사용되는 서비스 공개 키입니다.
- **service-private.pem** 은 **secret_vault** 에 저장된 암호를 해독하는 데 사용되는 서비스 개인 키입니다.

70.1. ANSIBLE을 사용하여 IDM에 비대칭 서비스 자격 증명 모음이 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 하나 이상의 개인 자격 증명 모음 컨테이너를 사용하여 중요한 정보를 안전하게 저장하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 관리자는 **secret_vault** 라는 비대칭 자격 증명 모음을 생성합니다. 이렇게 하면 자격 증명 모음에서 비밀을 검색하기 위해 자격 증명 모음 멤버가 개인 키를 사용하여 인증해야 합니다. 자격 증명 모음 멤버는 **IdM** 클라이언트에서 파일을 검색할 수 있습니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

- **IdM 관리자 암호를 알고 있습니다.**

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 서비스 인스턴스의 공개 키를 가져옵니다. 예를 들어 **openssl** 유틸리티를 사용합니다.

- a. **service-private.pem** 개인 키를 생성합니다.

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....+++
e is 65537 (0x10001)
```

- b. 개인 키를 기반으로 **service-public.pem** 공개 키를 생성합니다.

```
$ openssl rsa -in service-private.pem -out service-public.pem -pubout
writing RSA key
```

3. 선택 사항: 인벤토리 파일이 없는 경우(예: **inventory.file**)를 생성합니다.

```
$ touch inventory.file
```

4. 인벤토리 파일을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

5. **ensure-asymmetric-vault-is-present.yml** Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-asymmetric-vault-is-present.yml ensure-asymmetric-service-vault-is-present-copy.yml
```

6.

편집을 위해 **ensure-asymmetric-vault-is-present-copy.yml** 파일을 엽니다.

7.

service-public.pem 공개 키를 **Ansible** 컨트롤러에서 **server.idm.example.com** 서버로 복사하는 작업을 추가합니다.

8.

ipavault 작업 섹션에서 다음 변수를 설정하여 나머지 파일을 수정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 사용하여 자격 증명 모음의 이름을 정의합니다(예: **secret_vault**).
- **vault_type** 변수를 비대칭 으로 설정합니다.
- 서비스 변수를 자격 증명 모음을 소유하는 서비스의 주체(예: **HTTP/webserver1.idm.example.com**)로 설정합니다.
- **public_key_file** 을 공개 키의 위치로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Tests
  hosts: ipaserver
  become: true
  gather_facts: false
  tasks:
    - name: Copy public key to ipaserver.
      copy:
        src: /path/to/service-public.pem
        dest: /usr/share/doc/ansible-freeipa/playbooks/vault/service-public.pem
        mode: 0600
    - name: Add data to vault, from a LOCAL file.
      ipavault:
        ipaadmin_password: Secret123
        name: secret_vault
        vault_type: asymmetric
        service: HTTP/webserver1.idm.example.com
        public_key_file: /usr/share/doc/ansible-freeipa/playbooks/vault/service-public.pem
```

9.

파일을 저장합니다.

10.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-asymmetric-service-vault-is-present-copy.yml
```

70.2. ANSIBLE을 사용하여 멤버 서비스 추가

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 서비스 자격 증명 모음에 멤버 서비스를 추가하여 모두 자격 증명 모음에 저장된 비밀을 검색할 수 있는 방법을 보여줍니다. 아래 절차에서 사용되는 예제에서 **IdM** 관리자는 **HTTP/webserver2.idm.example.com** 및 **HTTP/webserver3.idm.example.com** 서비스 주체를 **HTTP/webserver1.idm.example.com** 에서 소유한 **secret_vault** 자격 증명 모음에 추가합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.
- 서비스 시크릿 을 저장할 비대칭 자격 증명 모음을 생성했습니다.

절차

1.

/usr/share/doc/ansible-freeipa/playbooks/vault 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2.

선택 사항: 인벤토리 파일이 없는 경우(예: **inventory.file**)를 생성합니다.

```
$ touch inventory.file
```

3.

인벤토리 파일을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

4.

data-archive-in-asymmetric-vault.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp data-archive-in-asymmetric-vault.yml add-services-to-an-asymmetric-vault.yml
```

5.

편집을 위해 **data-archive-in-asymmetric-vault-copy.yml** 파일을 엽니다.

6.

ipavault 작업 섹션에서 다음 변수를 설정하여 파일을 수정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 자격 증명 모음의 이름으로 설정합니다(예: **secret_vault**).
- 서비스 변수를 자격 증명 모음의 서비스 소유자(예: **HTTP/webserver1.idm.example.com**)로 설정합니다.
- **services** 변수를 사용하여 자격 증명 모음 시크릿에 액세스할 서비스를 정의합니다.
- **action** 변수를 **member** 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - ipavault:
    ipaadmin_password: Secret123
    name: secret_vault
    service: HTTP/webserver1.idm.example.com
    services:
```

```
- HTTP/webserver2.idm.example.com
- HTTP/webserver3.idm.example.com
action: member
```

7.

파일을 저장합니다.

8.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file add-services-to-an-asymmetric-vault.yml
```

70.3. ANSIBLE을 사용하여 비대칭 자격 증명 모음에 IDM 서비스 시크릿 저장

이 섹션에서는 **IdM(Identity Management)** 관리자가 나중에 서비스에서 검색할 수 있도록 **Ansible** 플레이북을 사용하여 서비스 자격 증명 모음에 시크릿을 저장할 수 있는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 관리자는 시크릿과 함께 **PEM** 파일을 **secret_vault** 라는 비대칭 자격 증명 모음에 저장합니다. 이렇게 하면 자격 증명 모음에서 비밀을 검색하기 위해 서비스가 개인 키를 사용하여 인증해야 합니다. 자격 증명 모음 멤버는 **IdM** 클라이언트에서 파일을 검색할 수 있습니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.
- 서비스 시크릿 을 저장할 비대칭 자격 증명 모음을 생성했습니다.
- 시크릿은 **Ansible** 컨트롤러에 로컬로 저장됩니다(예: **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem**).

절차

1.

/usr/share/doc/ansible-freeipa/playbooks/vault 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2.

선택 사항: 인벤토리 파일이 없는 경우(예: **inventory.file**)를 생성합니다.

```
$ touch inventory.file
```

3.

인벤토리 파일을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

4.

data-archive-in-asymmetric-vault.yml **Ansible** 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp data-archive-in-asymmetric-vault.yml data-archive-in-asymmetric-vault-copy.yml
```

5.

편집을 위해 **data-archive-in-asymmetric-vault-copy.yml** 파일을 엽니다.

6.

ipavault 작업 섹션에서 다음 변수를 설정하여 파일을 수정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 자격 증명 모음의 이름으로 설정합니다(예: **secret_vault**).
- 서비스 변수를 자격 증명 모음의 서비스 소유자(예: **HTTP/webserver1.idm.example.com**)로 설정합니다.
- 변수 의 를 **"{{ lookup('file', 'private-key-to-an-externally-signed-certificate.pem') | b64encode }}"** 로 설정합니다. 이렇게 하면 **Ansible**이 **IdM** 서버가 아닌 **Ansible** 컨트롤러의 작업 디렉터리에서 개인 키를 사용하여 파일을 검색할 수 있습니다.
- **action** 변수를 **member** 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```

---
- name: Tests
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  - ipavault:
    ipadmin_password: Secret123
    name: secret_vault
    service: HTTP/webserver1.idm.example.com
    in: "{{ lookup('file', 'private-key-to-an-externally-signed-certificate.pem') |
b64encode }}"
    action: member

```

7.

파일을 저장합니다.

8.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file data-archive-in-asymmetric-vault-copy.yml
```

70.4. ANSIBLE을 사용하여 IDM 서비스의 서비스 시크릿 검색

이 섹션에서는 **IdM(Identity Management)** 사용자가 **Ansible** 플레이북을 사용하여 서비스 대신 서비스 자격 증명 모음에서 시크릿을 검색하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 플레이북을 실행하면 **secret_vault**라는 비대칭 자격 증명 모음의 시크릿이 있는 **PEM** 파일을 검색하여 **Ansible** 인벤토리 파일에 나열된 모든 호스트의 지정된 위치에 **ipaservers**로 저장합니다.

서비스는 키맵을 사용하여 **IdM**에 인증하고 개인 키를 사용하여 자격 증명 모음에 인증합니다. **ansible-freeipa**가 설치된 **IdM** 클라이언트에서 서비스를 대신하여 파일을 검색할 수 있습니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.
- 서비스 시크릿을 저장할 비대칭 자격 증명 모음을 생성했습니다.

- 자격 증명 모음에 비밀을 보관합니다.
- Ansible 컨트롤러의 `private_key_file` 변수에서 지정한 위치에서 서비스 자격 증명 모음 시크릿을 검색하는 데 사용되는 개인 키를 저장했습니다.

절차

1. `/usr/share/doc/ansible-freeipa/playbooks/vault` 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 선택 사항: 인벤토리 파일이 없는 경우(예: `inventory.file`)를 생성합니다.

```
$ touch inventory.file
```

3. 인벤토리 파일을 열고 다음 호스트를 정의합니다.

- `[ipaserver]` 섹션에 `IdM` 서버를 정의합니다.
- `[webservers]` 섹션에서 시크릿을 검색할 호스트를 정의합니다. 예를 들어 `webserver1.idm.example.com`, `webserver2.idm.example.com` 및 `webserver3.idm.example.com`에 대한 시크릿을 검색하도록 Ansible에 지시하려면 다음을 입력합니다.

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

[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
webserver3.idm.example.com
```

4. `retrieve-data-asymmetric-vault.yml` Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp retrieve-data-asymmetric-vault.yml retrieve-data-asymmetric-vault-copy.yml
```

5.

편집을 위해 `retrieve-data-asymmetric-vault-copy.yml` 파일을 엽니다.

6.

`ipavault` 작업 섹션에서 다음 변수를 설정하여 파일을 수정합니다.

- `ipaadmin_password` 변수를 IdM 관리자 암호로 설정합니다.
- `name` 변수를 자격 증명 모음의 이름으로 설정합니다(예: `secret_vault`).
- 서비스 변수를 자격 증명 모음의 서비스 소유자(예: `HTTP/webserver1.idm.example.com`)로 설정합니다.
- `private_key_file` 변수를 서비스 자격 증명 모음 시크릿을 검색하는 데 사용되는 개인 키의 위치로 설정합니다.
- `out` 변수를 `private-key-to-an-externally-signed-certificate.pem` 시크릿(예: 현재 작업 디렉터리)을 검색하려는 IdM 서버의 위치로 설정합니다.
- `action` 변수를 `member` 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false

  tasks:
  - name: Retrieve data from the service vault
    ipavault:
      ipaadmin_password: Secret123
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      vault_type: asymmetric
      private_key: "{{ lookup('file', 'service-private.pem') | b64encode }}"
      out: private-key-to-an-externally-signed-certificate.pem
      state: retrieved
```

7.

IdM 서버에서 Ansible 컨트롤러에 데이터 파일을 검색하는 섹션을 플레이북에 추가합니다.

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false
  tasks:
[...]
```

```
- name: Retrieve data file
  fetch:
    src: private-key-to-an-externally-signed-certificate.pem
    dest: ./
    flat: yes
    mode: 0600
```

8.

의 Ansible 컨트롤러에서 인벤토리 파일의 `webserver` 섹션에 나열된 `webserver`로 검색된 `private-key-to-an-externally-signed-certificate.pem` 파일을 전송하는 플레이북에 섹션을 추가합니다.

```
---
- name: Send data file to webserver
  become: no
  gather_facts: no
  hosts: webserver
  tasks:
- name: Send data to webserver
  copy:
    src: private-key-to-an-externally-signed-certificate.pem
    dest: /etc/pki/tls/private/httpd.key
    mode: 0444
```

9.

파일을 저장합니다.

10.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file retrieve-data-asymmetric-vault-copy.yml
```

70.5. ANSIBLE을 사용하여 손상된 경우 IDM 서비스 자격 증명 모음 시크릿 변경

이 섹션에서는 **IdM(Identity Management)** 관리자가 서비스 인스턴스가 손상된 경우 **Ansible** 플레이북을 다시 사용하여 서비스 자격 증명 모음에 저장된 시크릿을 변경할 수 있는 방법을 보여줍니다. 다음 예제의 시나리오에서는 `webserver3.idm.example.com` 에서 검색된 시크릿이 손상되었지만 시크릿을 저장하는 비대칭 자격 증명 모음의 키가 손상되었다고 가정합니다. 이 예제에서 관리자는 **비대칭 자격 증명 모음에 비밀을 저장하고 IdM 호스트에 비대칭 자격 증명 모음에서 시크릿을 검색할 때 사용하는**

Ansible 플레이북을 재사용합니다. 절차 시작 시 **IdM** 관리자는 비대칭 자격 증명 모음에 새 시크릿을 사용하여 새 **PEM** 파일을 저장하고, 에서 손상된 웹 서버 **webserver3.idm.example.com** 에 새 시크릿을 검색하지 않도록 인벤토리 파일을 조정한 다음 두 절차를 다시 실행합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.
- 서비스 시크릿 을 저장할 비대칭 자격 증명 모음을 생성했습니다.
- 손상된 이전 키를 교체하기 위해 **IdM** 호스트에서 실행 중인 웹 서비스에 대한 새 **httpd** 키를 생성했습니다.
- 새 **httpd** 키는 **Ansible** 컨트롤러에 로컬로 저장됩니다(예: **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem**).

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 인벤토리 파일을 열고 다음 호스트가 올바르게 정의되었는지 확인합니다.

- **[ipaserver]** 섹션의 **IdM** 서버.
- **[webservers]** 섹션에서 시크릿을 검색할 호스트입니다. 예를 들어 **webserver1.idm.example.com** 및 **webserver2.idm.example.com** 에 대한 시크릿을 검색하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

```
[webserver1]
webserver1.idm.example.com
webserver2.idm.example.com
```



중요

현재 예제 **webserver3.idm.example.com** 에 손상된 웹 서버가 포함되어 있지 않은지 확인합니다.

3.

편집을 위해 **data-archive-in-asymmetric-vault-copy.yml** 파일을 엽니다.

4.

ipavault 작업 섹션에서 다음 변수를 설정하여 파일을 수정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 자격 증명 모음의 이름으로 설정합니다(예: **secret_vault**).
- 서비스 변수를 자격 증명 모음의 서비스 소유자(예: **HTTP/webserver.idm.example.com**)로 설정합니다.
- 변수의 를 **"{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem') | b64encode }}"** 로 설정합니다. 이렇게 하면 **Ansible**이 **IdM** 서버가 아닌 **Ansible** 컨트롤러의 작업 디렉터리에서 개인 키를 사용하여 파일을 검색할 수 있습니다.
- **action** 변수를 **member** 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - ipavault:
    ipaadmin_password: Secret123
    name: secret_vault
```

```

    service: HTTP/webserver.idm.example.com
    in: "{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem') |
b64encode }}"
    action: member

```

5. 파일을 저장합니다.

6. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file data-archive-in-asymmetric-vault-copy.yml
```

7. 편집을 위해 `retrieve-data-asymmetric-vault-copy.yml` 파일을 엽니다.

8. `ipavault` 작업 섹션에서 다음 변수를 설정하여 파일을 수정합니다.

- `ipaadmin_password` 변수를 `IdM` 관리자 암호로 설정합니다.
- `name` 변수를 자격 증명 모음의 이름으로 설정합니다(예: `secret_vault`).
- 서비스 변수를 자격 증명 모음의 서비스 소유자(예: `HTTP/webserver1.idm.example.com`)로 설정합니다.
- `private_key_file` 변수를 서비스 자격 증명 모음 시크릿을 검색하는 데 사용되는 개인 키의 위치로 설정합니다.
- `out` 변수를 `new-private-key-to-an-externally-signed-certificate.pem` 시크릿(예: 현재 작업 디렉터리)을 검색하려는 `IdM` 서버의 위치로 설정합니다.
- `action` 변수를 `member` 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```

---
- name: Retrieve data from vault
  hosts: ipaserver

```

```

become: no
gather_facts: false

tasks:
- name: Retrieve data from the service vault
  ipavault:
    ipaadmin_password: Secret123
    name: secret_vault
    service: HTTP/webserver1.idm.example.com
    vault_type: asymmetric
    private_key: "{{ lookup('file', 'service-private.pem') | b64encode }}"
    out: new-private-key-to-an-externally-signed-certificate.pem
    state: retrieved

```

9.

IdM 서버에서 Ansible 컨트롤러에 데이터 파일을 검색하는 섹션을 플레이북에 추가합니다.

```

---
- name: Retrieve data from vault
  hosts: ipaserver
  become: yes
  gather_facts: false
  tasks:
[...]
```

```

- name: Retrieve data file
  fetch:
    src: new-private-key-to-an-externally-signed-certificate.pem
    dest: ./
    flat: yes
    mode: 0600

```

10.

의 Ansible 컨트롤러에서 인벤토리 파일의 **webserver 섹션에 나열된 **webserver**로 검색된 **new-private-key-to-an-externally-signed-certificate.pem** 파일을 전송하는 플레이북에 섹션을 추가합니다.**

```

---
- name: Send data file to webserver
  become: yes
  gather_facts: no
  hosts: webserver
  tasks:
- name: Send data to webserver
  copy:
    src: new-private-key-to-an-externally-signed-certificate.pem
    dest: /etc/pki/tls/private/httpd.key
    mode: 0444

```

11.

파일을 저장합니다.

12.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file retrieve-data-asymmetric-vault-copy.yml
```

추가 리소스

- **Ansible**을 사용하여 **IdM** 자격 증명 모음 및 서비스 시크릿 및 플레이북 변수에 대한 자세한 내용은 `/usr/share/doc/ansible-freeipa/` 디렉터리에 사용 가능한 **README-vault.md** **Markdown** 파일과 `/usr/share/doc/ansible-freeipa/playbooks/vault/` 디렉터리에 있는 샘플 플레이북을 참조하십시오.

71장. ANSIBLE을 사용하여 IDM에 서비스가 있는지 여부 확인

Ansible service 모듈을 사용하여 **IdM(Identity Management)** 관리자는 **IdM**에 고유하지 않거나 **IdM**에 없는 특정 서비스가 있는지 확인할 수 있습니다. 예를 들어 **service** 모듈을 사용하여 다음을 수행할 수 있습니다.

- 수동으로 설치한 서비스가 **IdM** 클라이언트에 있는지 확인하고 해당 서비스가 없는 경우 자동으로 설치합니다. 자세한 내용은 다음을 참조하십시오.
 - **IdM** 클라이언트에 **HTTP** 서비스가 **IdM**에 있는지 확인합니다.
 - **IdM**이 아닌 클라이언트의 **IdM**에 **HTTP** 서비스가 있는지 확인합니다.
 - **DNS** 없이 **IdM** 클라이언트에 **HTTP** 서비스가 있는지 확인합니다.
- **IdM**에 등록된 서비스에 인증서가 연결되어 있고 해당 인증서가 없는 경우 자동으로 설치되었는지 확인합니다. 자세한 내용은 다음을 참조하십시오.
 - **IdM** 서비스 항목에 외부 서명된 인증서가 있는지 확인합니다.
- **IdM** 사용자와 호스트에서 서비스 키를 검색하고 생성할 수 있도록 허용합니다. 자세한 내용은 다음을 참조하십시오.
 - **IdM** 사용자, 그룹, 호스트 또는 호스트 그룹이 서비스의 키를 생성하도록 허용.
 - **IdM** 사용자, 그룹, 호스트 또는 호스트 그룹이 서비스의 키를 검색할 수 있도록 허용.
- **IdM** 사용자와 호스트에서 **Kerberos** 별칭을 서비스에 추가할 수 있도록 허용합니다. 자세한 내용은 다음을 참조하십시오.
 - 서비스에 **Kerberos** 기본 별칭이 있는지 확인합니다.

서비스가 **IdM** 클라이언트에 없는지 확인하고 해당 서비스가 있는 경우 자동으로 제거합니다. 자세한 내용은 다음을 참조하십시오.

○

IdM 클라이언트의 IdM에 HTTP 서비스가 없는지 확인합니다.

71.1. ANSIBLE 플레이북을 사용하여 IdM에 HTTP 서비스가 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM**에 **HTTP** 서버가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **HTTP** 서비스를 호스팅하는 시스템은 **IdM** 클라이언트입니다.
- **IdM** 관리자 암호가 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2. **inventory.file** 을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present.yml** **Ansible** 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 편집을 위해 **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-copy.yml** **Ansible** 플레이북 파일을 엽니다.

```

---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
    # Ensure service is present
    - ipaservice:
        ipaadmin_password: Secret123
        name: HTTP/client.idm.example.com

```

5.

파일을 조정합니다.

•

ipaadmin_password 변수에 정의된 **IdM** 관리자 암호를 변경합니다.

•

ipaservice 작업의 **name** 변수에 정의된 대로 **HTTP** 서비스가 실행 중인 **IdM** 클라이언트의 이름을 변경합니다.

6.

파일을 저장하고 종료합니다.

7.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-copy.yml

```

검증 단계

1.

IdM 관리자로 **IdM** 웹 UI에 로그인합니다.

2.

ID → 서비스로 이동합니다.

HTTP/client.idm.example.com@IDM.EXAMPLE.COM 이 서비스 목록에 나열되면 **Ansible** 플레이북이 **IdM**에 성공적으로 추가되었습니다.

추가 리소스

•

TLS 암호화를 **Apache HTTP** 서버에 추가하여 **HTTP** 서버와 브라우저 클라이언트 간의 통신

을 보호할 수 있습니다.

- **IdM 인증 기관에서 HTTP 서비스의 인증서를 요청할 수 있습니다.** 자세한 내용은 **certmonger**를 사용하여 서비스의 **IdM 인증서 가져오기**에 설명된 절차를 참조하십시오.

71.2. ANSIBLE 플레이북을 사용하여 비IDM 클라이언트에서 IDM에 HTTP 서비스가 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하는 **IdM** 클라이언트가 아닌 호스트에서 **IdM**에 **HTTP** 서버가 있는지 확인하는 방법에 대해 설명합니다. **IdM**에 **HTTP** 서버를 추가하여 **IdM**에 호스트를 추가합니다.

사전 요구 사항

- 호스트에 **HTTP** 서비스를 설치했습니다.
- **HTTP**를 설정한 호스트는 **IdM** 클라이언트가 아닙니다. 또는 **Ansible** 플레이북을 사용하여 **IdM**에 **HTTP** 서비스가 있는지 확인하는 단계를 따르십시오.
- **IdM** 관리자 암호가 있습니다.
- 호스트에 대해 **IPv6**를 사용하는 경우 **DNS A** 레코드 또는 **AAAA** 레코드가 사용됩니다.

절차

1. 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2. **inventory.file** 을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check.yml** **Ansible** 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

■

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check-copy.yml
```

4.

복사된 파일 `/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check-copy.yml` 을 열어 편집합니다. `ipa service` 작업에서 `ipaadmin_password` 및 이름 변수를 찾습니다.

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
    # Ensure service is present
    - ipaservice:
      ipaadmin_password: MyPassword123
      name: HTTP/www2.example.com
      skip_host_check: yes
```

5.

파일을 조정합니다.

- `ipaadmin_password` 변수를 `IdM` 관리자 암호로 설정합니다.
- `name` 변수를 `HTTP` 서비스가 실행 중인 호스트의 이름으로 설정합니다.

6.

파일을 저장하고 종료합니다.

7.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check-copy.yml
```

검증 단계

1.

IdM 관리자로 **IdM** 웹 UI에 로그인합니다.

2.

ID → 서비스로 이동합니다.

이제 서비스 목록에 **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** 이 표시됩니다.

추가 리소스

•

TLS 암호화를 **Apache HTTP** 서버에 추가하여 **HTTP** 서버와 브라우저 클라이언트 간의 통신을 보호할 수 있습니다.

71.3. ANSIBLE PLAYBOOK을 사용하여 DNS 없이 IDM 클라이언트에 HTTP 서비스가 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **DNS** 항목이 없는 **IdM** 클라이언트에서 **HTTP** 서버가 실행되고 있는지 확인하는 방법에 대해 설명합니다. 시나리오는 **IdM** 호스트에 사용 가능한 **DNS A** 항목이 없으며 **IPv4** 대신 **IPv6**를 사용하는 경우 **DNS AAAA** 항목이 없다는 것입니다.

사전 요구 사항

•

HTTP 서비스를 호스트하는 시스템은 **IdM**에 등록되어 있습니다.

•

호스트의 **DNS A** 또는 **DNS AAAA** 레코드가 없을 수 있습니다. 그렇지 않으면 호스트의 **DNS** 레코드가 있는 경우 **Ansible** 플레이북을 사용하여 **IdM**에 **HTTP** 서비스가 있는지 확인하는 절차를 따르십시오.

•

IdM 관리자 암호가 있습니다.

절차

1.

인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2.

inventory.file 을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3.

/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml
```

4.

복사된 파일 **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml** 을 열어 편집합니다. **ipa service** 작업에서 **ipaadmin_password** 및 이름 변수를 찾습니다.

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
    # Ensure service is present
    - ipaservice:
      ipaadmin_password: MyPassword123
      name: HTTP/ihavenodns.info
      force: yes
```

5.

파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 **HTTP** 서비스가 실행 중인 호스트의 이름으로 설정합니다.

6.

파일을 저장하고 종료합니다.

7.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml
```

검증 단계

1. **IdM 관리자로 IdM 웹 UI에 로그인합니다.**
2. **ID → 서비스로 이동합니다.**

이제 서비스 목록에 **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** 이 표시됩니다.

추가 리소스

- **Apache HTTP 서버에 TLS 암호화를 추가하여 Apache HTTP 서버와 브라우저 클라이언트 간의 통신을 보호할 수 있습니다.**

71.4. ANSIBLE PLAYBOOK을 사용하여 IDM 서비스 항목에 외부 서명된 인증서가 있는지 확인

이 섹션에서는 **ansible-freeipa** 서비스 모듈을 사용하여 외부 **CA**(인증 기관)에서 발급한 인증서가 **HTTP** 서비스의 **IdM** 항목에 연결되어 있는지 확인하는 방법을 설명합니다. **IdM CA**가 자체 서명된 인증서를 사용하는 경우 **IdM CA**가 아닌 외부 **CA**에서 서명한 **HTTP** 서비스의 인증서를 보유하는 것이 특히 유용합니다.

사전 요구 사항

- 호스트에 **HTTP** 서비스를 설치했습니다.
- **HTTP** 서비스를 **IdM**에 등록했습니다.
- **IdM** 관리자 암호가 있습니다.
- 주체가 **HTTP** 서비스의 주체에 해당하는 외부 서명 인증서가 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

- 2.

inventory.file 을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3.

/usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present.yml 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4.

선택 사항: 인증서가 **PEM**(개인 정보 보호 강화 메일) 형식인 경우 **CLI**(명령줄 인터페이스)를 통해 더 쉽게 처리할 수 있도록 인증서를 **DER**(고급 인코딩 규칙) 형식으로 변환합니다.

```
$ openssl x509 -outform der -in cert1.pem -out cert1.der
```

5.

base64 명령을 사용하여 **DER** 파일을 표준 출력으로 디코딩합니다. **w 0** 옵션을 사용하여 래핑을 비활성화합니다.

```
$ base64 cert1.der -w0
MIIC/zCCAeegAwIBAgIUUV74O+4kXeg21o4vxfRRtyJm...
```

6.

인증서를 표준 출력에서 클립보드로 복사합니다.

7.

/usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present-copy.yml 파일을 열어 내용을 편집하고 봅니다.

```
---
- name: Service certificate present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
    # Ensure service certificate is present
    - ipaservice:
        ipaadmin_password: MyPassword123
        name: HTTP/www.example.com
        certificate: |
          - MIICBjCCAW8CFHnm32VcXaUDGfEGdDL/...
```

```
[...]
action: member
state: present
```

8.

파일을 조정합니다.

•

인증서 변수를 사용하여 정의한 인증서를 **CLI**에서 복사한 인증서로 바꿉니다. 표시된 대로 **"|"** 파이프 문자가 있는 **certificate:** 변수를 사용하는 경우 한 줄에 입력하지 않고 인증서 **THIS WAY**를 입력할 수 있습니다. 이렇게 하면 인증서를 더 쉽게 읽을 수 있습니다.

•

ipaadmin_password 변수에 정의된 **IdM** 관리자 암호를 변경합니다.

•

name 변수에서 정의한 **HTTP** 서비스가 실행 중인 **IdM** 클라이언트의 이름을 변경합니다.

•

기타 관련 변수를 변경합니다.

9.

파일을 저장하고 종료합니다.

10.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present-
copy.yml
```

검증 단계

1.

IdM 관리자로 **IdM** 웹 **UI**에 로그인합니다.

2.

ID → 서비스로 이동합니다.

3.

새로 추가된 인증서(예: **HTTP/client.idm.example.com**)를 사용하여 서비스 이름을 클릭합니다.

오른쪽에 있는 **Service Certificate**(서비스 인증서) 섹션에서 새로 추가된 인증서를 볼 수 있습니다.

71.5. ANSIBLE 플레이북을 사용하여 **IDM** 사용자, 그룹, 호스트 또는 호스트 그룹을 사용하여 서비스의 키탭을 생성할 수 있습니다.

keytab은 **Kerberos** 주체와 암호화된 키의 쌍을 포함하는 파일입니다. 키탭 파일은 일반적으로 일반 텍스트 파일에 저장된 암호나 사람의 상호 작용 없이도 **Kerberos**를 사용하여 스크립트를 자동으로 인증할 수 있도록 하는 데 사용됩니다. 그런 다음 스크립트가 가져온 자격 증명을 사용하여 원격 시스템에 저장된 파일에 액세스할 수 있습니다.

IdM(Identity Management) 관리자는 다른 사용자가 **IdM**에서 실행되는 서비스에 대한 키탭을 검색하거나 만들 수 있습니다. 특정 사용자와 사용자 그룹이 키탭을 만들 수 있도록 허용하면 **IdM** 관리자 암호를 공유하지 않고 서비스 관리를 위임할 수 있습니다. 이 위임은 보다 세부적인 시스템 관리를 제공합니다.

이 섹션에서는 특정 **IdM** 사용자, 사용자 그룹, 호스트 및 호스트 그룹을 통해 **IdM** 클라이언트에서 실행되는 **HTTP** 서비스에 대한 키탭을 생성하는 방법을 설명합니다. 특히 **user01** **IdM** 사용자가 **client**.**idm.example.com** 이라는 **IdM** 클라이언트에서 실행되는 **HTTP** 서비스에 대한 키탭을 생성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **HTTP** 서비스를 **IdM**에 등록했습니다.
- **HTTP** 서비스를 호스팅하는 시스템은 **IdM** 클라이언트입니다.
- 키탭을 생성할 수 있도록 허용하려는 **IdM** 사용자 및 사용자 그룹은 **IdM**에 있습니다.
- 키탭을 생성할 수 있도록 허용하려는 **IdM** 호스트 및 호스트 그룹은 **IdM**에 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2. **inventory.file** 을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present.yml** **Ansible** 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 편집을 위해 **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present-copy.yml** **Ansible** 플레이북 파일을 엽니다.

5. 다음을 변경하여 파일을 조정합니다.

- **ipaadmin_password** 변수에 지정된 **IdM** 관리자 암호입니다.
- **HTTP** 서비스가 실행 중인 **IdM** 클라이언트의 이름입니다. 현재 예제에서는 **HTTP/client.idm.example.com**입니다.
- **allow_create_keytab_user**: 섹션에 나열된 **IdM** 사용자의 이름입니다. 현재 예제에서는 **user01** 입니다.
- **allow_create_keytab_group**: 섹션에 나열된 **IdM** 사용자 그룹의 이름입니다.
- **allow_create_keytab_host**: 섹션에 나열된 **IdM** 호스트의 이름입니다.

- **allow_create_keytab_hostgroup**: 섹션에 나열된 **IdM** 호스트 그룹의 이름입니다.
- **tasks** 섹션의 **name** 변수에서 지정한 작업의 이름입니다.

현재 예제에 맞게 수정한 후 복사된 파일은 다음과 같습니다.

```
---
- name: Service member allow_create_keytab present
  hosts: ipaserver
  become: true

  tasks:
    - name: Service HTTP/client.idm.example.com members allow_create_keytab present
      for user01
      ipaservice:
        ipaadmin_password: Secret123
        name: HTTP/client.idm.example.com
        allow_create_keytab_user:
          - user01
        action: member
```

6.

파일을 저장합니다.

7.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-member-
allow_create_keytab-present-copy.yml
```

검증 단계

1.

특정 **HTTP** 서비스에 대한 **keytab**을 생성할 권한이 있는 **IdM** 사용자로 **IdM** 서버에 **SSH** 연결을 수행합니다.

```
$ ssh user01@server.idm.example.com
Password:
```

2.

ipa-getkeytab 명령을 사용하여 **HTTP** 서비스에 대한 새 **keytab**을 생성합니다.

```
$ ipa-getkeytab -s server.idm.example.com -p HTTP/client.idm.example.com -k
/etc/httpd/conf/krb5.keytab
```

s 옵션은 키 탭을 생성할 **KDC**(키 배포 센터) 서버를 지정합니다.

p 옵션은 생성하려는 주체를 지정합니다.

k 옵션은 새 키를 추가할 키탭 파일을 지정합니다. 파일이 없는 경우 생성됩니다.

명령이 오류가 발생하지 않으면 **user01** 로 **HTTP/client.idm.example.com** 의 키탭을 성공적으로 생성했습니다.

71.6. ANSIBLE 플레이북을 사용하여 IDM 사용자, 그룹, 호스트 또는 호스트 그룹을 통해 서비스의 키탭 검색 가능

keytab은 **Kerberos** 주체와 암호화된 키의 쌍을 포함하는 파일입니다. 키탭 파일은 일반적으로 일반 텍스트 파일에 저장된 암호나 사람의 상호 작용 없이도 스크립트를 **Kerberos**로 자동으로 인증할 수 있도록 하는 데 사용됩니다. 그런 다음 스크립트가 가져온 자격 증명을 사용하여 원격 시스템에 저장된 파일에 액세스할 수 있습니다.

IdM 관리자는 다른 사용자가 **IdM**에서 실행되는 서비스에 대한 키탭을 검색하거나 만들 수 있습니다.

이 섹션에서는 특정 **IdM** 사용자, 사용자 그룹, 호스트 및 호스트 그룹을 통해 **IdM** 클라이언트에서 실행되는 **HTTP** 서비스의 키탭을 검색하는 방법을 설명합니다. 특히 **user01** **IdM** 사용자가 **client.idm.example.com** 에서 실행되는 **HTTP** 서비스의 **keytab**을 검색할 수 있도록 하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **HTTP** 서비스를 **IdM**에 등록했습니다.
- 키탭을 검색하려는 **IdM** 사용자 및 사용자 그룹은 **IdM**에 있습니다.

- 키탭을 검색할 수 있도록 허용하려는 **IdM** 호스트 및 호스트 그룹은 **IdM**에 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2. **inventory.file** 을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present.yml** **Ansible** 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 편집을 위해 복사된 파일 **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present-copy.yml** 을 엽니다.

5. 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.

- **ipaservice** 작업의 **name** 변수를 **HTTP** 서비스의 주체로 설정합니다. 현재 예제에서는 **HTTP/client.idm.example.com**입니다.

- **allow_retrieve_keytab_group**: 섹션에서 **IdM** 사용자의 이름을 지정합니다. 현재 예제에서는 **user01** 입니다.

- **allow_retrieve_keytab_group**: 섹션에서 **IdM** 사용자 그룹의 이름을 지정합니다.
- **allow_retrieve_keytab_group**: 섹션에서 **IdM** 호스트의 이름을 지정합니다.
- **allow_retrieve_keytab_group**: 섹션에서 **IdM** 호스트 그룹의 이름을 지정합니다.
- **tasks** 섹션에서 **name** 변수를 사용하여 작업 이름을 지정합니다.

현재 예제에 맞게 수정한 후 복사된 파일은 다음과 같습니다.

```
---
- name: Service member allow_retrieve_keytab present
  hosts: ipaserver
  become: true

  tasks:
    - name: Service HTTP/client.idm.example.com members allow_retrieve_keytab
      present for user01
      ipaservice:
        ipaadmin_password: Secret123
        name: HTTP/client.idm.example.com
        allow_retrieve_keytab_user:
          - user01
        action: member
```

6. 파일을 저장합니다.

7. 플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-member-
allow_retrieve_keytab-present-copy.yml
```

검증 단계

1. **HTTP** 서비스의 **keytab**을 검색할 권한이 있는 **IdM** 사용자로 **IdM** 서버에 **SSH**로 연결합니다.

```
$ ssh user01@server.idm.example.com
Password:
```


2.

ipa-getkeytab 명령을 **-r** 옵션과 함께 사용하여 **keytab**을 검색합니다.

```
$ ipa-getkeytab -r -s server.idm.example.com -p HTTP/client.idm.example.com -k /etc/httpd/conf/krb5.keytab
```

s 옵션은 키 탭을 검색할 **KDC**(키 배포 센터) 서버를 지정합니다.

p 옵션은 검색할 키탭의 주체를 지정합니다.

k 옵션은 검색된 키를 추가할 **keytab** 파일을 지정합니다. 파일이 없는 경우 생성됩니다.

명령이 오류가 발생하지 않으면 **user01** 로 **HTTP/client.idm.example.com** 의 키탭을 성공적으로 불러 왔습니다.

71.7. ANSIBLE PLAYBOOK을 사용하여 서비스의 KERBEROS 기본 별칭이 있는지 확인

일부 시나리오에서는 **IdM** 관리자가 **Kerberos** 주체 별칭을 사용하여 **IdM** 사용자, 호스트 또는 서비스를 **Kerberos** 애플리케이션에 인증할 수 있도록 하는 것이 좋습니다. 이러한 시나리오는 다음과 같습니다.

- 사용자 이름이 변경되었지만 사용자는 이전 및 새 사용자 이름을 모두 사용하여 시스템에 로그인할 수 있어야 합니다.
- **IdM Kerberos** 영역이 이메일 도메인과 다른 경우에도 사용자는 이메일 주소를 사용하여 로그인해야 합니다.

이 섹션에서는 **client.idm.example.com**에서 실행되는 **HTTP** 서비스에 대해 **HTTP/mycompany.idm.example.com**의 주요 별칭을 생성하는 방법을 설명합니다.

사전 요구 사항

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

- 호스트에 **HTTP** 서비스를 설정했습니다.
- **HTTP** 서비스를 **IdM**에 등록했습니다.
- **HTTP**를 설정한 호스트는 **IdM** 클라이언트입니다.

절차

1. 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2. **inventory.file** 을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present.yml** **Ansible** 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 편집을 위해 **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present-copy.yml** **Ansible** 플레이북 파일을 엽니다.

5. 다음을 변경하여 파일을 조정합니다.

- **ipaadmin_password** 변수에 지정된 **IdM** 관리자 암호입니다.
- **name** 변수에서 지정한 서비스의 이름입니다. 이는 서비스의 정식 주체 이름입니다. 현재 예제에서는 **HTTP/client.idm.example.com** 입니다.

- **principal** 변수에서 지정한 **Kerberos** 주체 별칭입니다. **name** 변수로 정의한 서비스에 추가하려는 별칭입니다. 현재 예제에서는 **host/mycompany.idm.example.com** 입니다.
- **tasks** 섹션의 **name** 변수에서 지정한 작업의 이름입니다.

현재 예제에 맞게 수정한 후 복사된 파일은 다음과 같습니다.

```
---
- name: Service member principal present
  hosts: ipaserver
  become: true

  tasks:
    - name: Service HTTP/client.idm.example.com member principals
      host/mycompany.idm.exmaple.com present
      ipaservice:
        ipaadmin_password: Secret123
        name: HTTP/client.idm.example.com
        principal:
          - host/mycompany.idm.example.com
        action: member
```

6. 파일을 저장합니다.

7. 플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present-
copy.yml
```

플레이북을 실행하면 0에 연결할 수 없고 0개의 실패한 작업이 생성되면 **HTTP /client.idm.example.com** 서비스에 대한 **host/mycompany.idm.example.com Kerberos** 사용자가 성공적으로 생성되었습니다.

추가 리소스

- **Kerberos** 주체 별칭에 대한 자세한 내용은 [사용자, 호스트 및 서비스에 대한 Kerberos 주체 별칭 관리](#)를 참조하십시오.

71.8. ANSIBLE 플레이북을 사용하여 IDM에 HTTP 서비스가 있는지 확인

이 섹션에서는 **IdM**에서 서비스 등록을 취소하는 방법에 대해 설명합니다. 보다 구체적으로는 **Ansible** 플레이북을 사용하여 **IdM**에 **HTTP /client.idm.example.com**이라는 **HTTP** 서버가 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호가 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2. **inventory.file** 을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent.yml** **Ansible** 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 편집을 위해 **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent-copy.yml** **Ansible** 플레이북 파일을 엽니다.

5. 다음을 변경하여 파일을 조정합니다.

- **ipaadmin_password** 변수로 정의된 **IdM** 관리자 암호입니다.
- **ipaservice** 작업의 **name** 변수에 정의된 **HTTP** 서비스의 **Kerberos** 주체입니다.

현재 예제에 맞게 수정한 후 복사된 파일은 다음과 같습니다.

```

---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
    # Ensure service is absent
    - ipaservice:
        ipaadmin_password: Secret123
        name: HTTP/client.idm.example.com
        state: absent

```

6.

파일을 저장하고 종료합니다.

7.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent-copy.yml

```

검증 단계

1.

IdM 관리자로 **IdM** 웹 UI에 로그인합니다.

2.

ID → 서비스로 이동합니다.

서비스 목록에 **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** 서비스가 표시되지 않으면 **IdM**에 있는지 확인해야 합니다.

추가 리소스

•

/usr/share/doc/ansible-freeipa/ 디렉터리에 있는 **README-service.md** Markdown 파일에서 가능한 변수 목록을 포함하여 **IdM**에 서비스가 있는지 샘플 **Ansible** 플레이북을 볼 수 있습니다.

•

/usr/share/doc/ansible-freeipa/playbooks/config 디렉터리에서 **IdM**에 서비스가 있는지 확인하는 샘플 **Ansible** 플레이북을 확인할 수 있습니다.

72장. AD 사용자가 IDM을 관리하도록 활성화

72.1. AD 사용자의 ID 덮어쓰기

RHEL(Red Hat Enterprise Linux) 7에서 외부 그룹 멤버십을 통해 AD(Active Directory) 사용자 및 그룹이 SSSD(System Security Services Daemon)를 활용하여 POSIX 환경에서 IdM(Identity Management) 리소스에 액세스할 수 있습니다.

IdM LDAP 서버에는 액세스 제어 권한을 부여하는 자체 메커니즘이 있습니다. RHEL 8에서는 AD 사용자에게 대한 ID 사용자 재정의의 IdM 그룹의 멤버로 추가할 수 있는 업데이트가 도입되었습니다. ID 재정의는 특정 ID 보기(이 경우 기본 신뢰 보기) 내에서 특정 Active Directory 사용자 또는 그룹 속성이 표시되는 방식을 설명하는 레코드입니다. 업데이트 결과 IdM LDAP 서버에서 IdM 그룹에 대한 액세스 제어 규칙을 AD 사용자에게 적용할 수 있습니다.

이제 AD 사용자가 IdM UI의 셀프 서비스 기능을 사용하여 SSH 키를 업로드하거나 개인 데이터를 변경하는 등의 작업을 수행할 수 있습니다. AD 관리자가 두 개의 서로 다른 계정 및 암호 없이도 IdM을 완전히 관리할 수 있습니다.



참고

현재 AD 사용자가 IdM에서 선택한 기능을 사용할 수 없습니다. 예를 들어 IdM admins 그룹에서 AD 사용자로 IdM 사용자의 암호를 설정하는 데 실패할 수 있습니다.

72.2. ID 덮어쓰기를 사용하여 AD 사용자가 IDM 관리 활성화

사전 요구 사항

- **idm:DL1 스트림은 IdM(Identity Management) 서버에서 활성화되며 이 스트림을 통해 제공되는 RPM으로 전환했습니다.**

```
# yum module enable idm:DL1
# yum distro-sync
```

- **idm:DL1/adtrust 프로필이 IdM 서버에 설치되어 있습니다.**

```
# yum module install idm:DL1/adtrust
```

프로필에는 ipa-idoverride-memberof 패키지를 포함하여 AD(Active Directory)와 신뢰 계약이 있는 IdM 서버를 설치하는 데 필요한 모든 패키지가 포함되어 있습니다.

- 작동 중인 **IdM** 환경이 설정됩니다. 자세한 내용은 [Identity Management 설치](#)를 참조하십시오.
- **IdM** 환경과 **AD** 간의 작업 신뢰가 설정됩니다.

절차

이 절차에서는 **AD** 사용자가 **IdM** 사용자와 동일한 사용자 권한을 제공하기 위한 **ID** 재정의의 생성 및 사용에 대해 설명합니다. 이 절차 중에 신뢰 컨트롤러 또는 신뢰 에이전트로 구성된 **IdM** 서버를 사용합니다. 신뢰 컨트롤러 및 신뢰 에이전트에 대한 자세한 내용은 [Identity Management 계획](#)의 신뢰 컨트롤러 및 신뢰 에이전트를 참조하십시오.

1.

IdM 관리자는 **Default Trust View**에서 **AD** 사용자에 대한 **ID** 재정의의 생성합니다. 예를 들어 **ad_user@ad.example.com** 사용자에 대한 **ID** 덮어쓰기를 생성하려면 다음을 수행합니다.

```
# kinit admin
# ipa idoverrideuser-add 'default trust view' ad_user@ad.example.com
```

2.

Default Trust View의 **ID** 재정의의를 **IdM** 그룹에 멤버로 추가합니다. 문제의 그룹이 **IdM** 역할의 멤버인 경우 **ID** 재정의에서 표시하는 **AD** 사용자는 명령줄 인터페이스와 **IdM** 웹 **UI**를 모두 포함하여 **IdM API**를 사용할 때 역할에서 부여하는 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
```

72.3. IDM CLI를 AD 사용자로 관리

이 절차에서는 **AD(Active Directory)** 사용자가 **IdM(Identity Management) CLI(명령줄 인터페이스)**에 로그인하고 역할에 적합한 명령을 실행할 수 있는지 확인합니다.

1.

IdM 관리자의 현재 **Kerberos** 티켓을 삭제합니다.

```
# kdestroy -A
```



참고

MIT Kerberos의 GSSAPI 구현에서 기본 설정에 따라 대상 서비스의 자격 증명을 선택하기 때문에 Kerberos 티켓 제거가 필요합니다. 이 경우에는 **IdM** 영역입니다. 즉, 자격 증명 캐시 컬렉션, 즉 **KCM:**, **KEYRING:** 또는 **DIR:** 자격 증명 캐시 유형이 사용 중인 경우 이전에 획득한 **admin** 또는 기타 **IdM** 주체의 자격 증명이 **AD** 사용자의 자격 증명 대신 **IdM API**에 사용됩니다.

2.

ID 재정의가 생성된 **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** 그룹과 **IdM** 그룹의 멤버십에서 발생하는 동일한 권한을 사용하는지 테스트합니다. **AD** 사용자의 **ID** 재정의가 **admins** 그룹에 추가된 경우 **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
```


73장. 짧은 AD 사용자 이름 확인 순서 구성

기본적으로 `user_name@domain.com` 또는 `domain.com\user_name` 형식으로 정규화된 이름을 지정하여 AD(Active Directory) 환경에서 사용자와 그룹을 확인하고 인증해야 합니다. 다음 섹션에서는 짧은 AD 사용자 이름 및 그룹 이름을 확인하도록 IdM 서버 및 클라이언트를 구성하는 방법을 설명합니다.

- 도메인 확인 순서가 작동하는 방식
- IdM 서버에서 글로벌 도메인 확인 순서 설정
- IdM 서버에서 ID 보기에 대한 도메인 확인 순서 설정
- IdM 클라이언트의 SSSD에서 도메인 확인 순서 설정

73.1. 도메인 확인 순서가 작동하는 방식

AD(Active Directory) 신뢰가 있는 IdM(Identity Management) 환경에서는 정규화된 이름을 지정하여 사용자와 그룹을 확인하고 인증하는 것이 좋습니다. 예를 들면 다음과 같습니다.

- `<idm_username> idm.example.com` 도메인의 IdM 사용자를 위한 `@idm.example.com`
- `<ad_username>@ad.example.com` 도메인의 AD 사용자용 `@ad.example.com`

기본적으로 `ad_username` 과 같은 짧은 이름 형식을 사용하여 사용자 또는 그룹 조회를 수행하는 경우 IdM 도메인만 검색하고 AD 사용자 또는 그룹을 찾지 못합니다. 단축 이름을 사용하여 AD 사용자 또는 그룹을 확인하려면 도메인 확인 순서 옵션을 설정하여 IdM이 여러 도메인을 검색하는 순서를 변경합니다.

IdM 데이터베이스 또는 개별 클라이언트의 SSSD 구성에서 도메인 확인 순서를 중앙에서 설정할 수 있습니다. IdM은 도메인 확인 순서를 다음 순서로 평가합니다.

- 로컬 `/etc/sss/sss.conf` 구성.

- **ID 보기 구성입니다.**
- **글로벌 IdM 구성입니다.**

참고

- 호스트의 **SSSD** 구성에 **default_domain_suffix** 옵션이 포함되어 있고 이 옵션으로 지정되지 않은 도메인에 요청하려는 경우 정규화된 사용자 이름을 사용해야 합니다.
- 도메인 확인 순서 옵션을 사용하고 **compat** 트리를 쿼리하는 경우 여러 **UID(사용자 ID)**가 수신될 수 있습니다. 이것이 영향을 줄 수 있는 경우 **도메인 확인 순서가 설정된 경우 Pagure 버그 보고서 Inconsistent compat user objects for AD users** 를 참조하십시오.



중요

IdM 클라이언트 또는 IdM 서버에서 full_name_format SSSD 옵션을 사용하지 마십시오. 이 옵션에 기본값이 아닌 값을 사용하면 사용자 이름이 표시되는 방식이 변경되고 IdM 환경에서 조치가 중단될 수 있습니다.

추가 리소스

- **compat** 트리에 대한 자세한 내용은 **Active Directory Trust for Legacy Linux Clients** 를 참조하십시오.

73.2. IDM 서버에서 글로벌 도메인 확인 순서 설정

이 절차에서는 **IdM** 도메인에 있는 모든 클라이언트에 대한 도메인 확인 순서를 설정합니다. 이 예제에서는 사용자 및 그룹을 다음 순서로 검색하는 도메인 확인 순서를 설정합니다.

1. **AD(Active Directory) 루트 도메인 ad.example.com**
2. **AD 하위 도메인 1.ad.example.com**
3. **IdM domain idm.example.com**

사전 요구 사항

- **AD 환경을 사용하여 신뢰를 구성했습니다.**

절차

- **ipa config-mod --domain-resolution-order 명령을 사용하여 선호하는 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 이름만 사용하여 ad.example.com 도메인에서 사용자의 사용자 정보를 검색할 수 있는지 확인합니다.**

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

73.3. IDM 서버에서 ID 보기에 대한 도메인 확인 순서 설정

이 절차에서는 특정 IdM 서버 및 클라이언트 집합에 적용할 수 있는 ID 보기의 도메인 확인 순서를 설정합니다. 이 예제에서는 IdM 호스트 client1.idm.example.com 에 대한 ADsubdomain1_first 라는 ID 뷰를 생성하고 다음 순서로 사용자와 그룹을 검색하도록 도메인 확인 순서를 설정합니다.

1. **AD(Active Directory) 하위 도메인1.ad.example.com**
2. **AD root 도메인 ad.example.com**
3. **IdM domain idm.example.com**



참고

ID 보기에 설정된 도메인 확인 순서는 글로벌 도메인 확인 순서를 재정의하지만 SSSD 구성에서 로컬로 설정된 도메인 확인 순서는 재정의하지 않습니다.

사전 요구 사항

- AD 환경을 사용하여 신뢰를 구성했습니다.

절차

1. **domain- resolution -order** 옵션을 설정하여 ID 보기를 만듭니다.

```
[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

- *subdomain1.ad.example.com* 도메인에서 짧은 이름만 사용하여 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)
```

추가 리소스

- [ID 보기를 사용하여 IdM 클라이언트에서 사용자 속성 값을 재정의](#)

73.4. IDM 클라이언트의 SSSD에서 도메인 확인 순서 설정

이 절차에서는 IdM 클라이언트의 SSSD 구성에서 도메인 확인 순서를 설정합니다. 이 예제에서는 IdM 호스트 *client2.idm.example.com* 을 구성하여 다음 순서대로 사용자와 그룹을 검색합니다.

1. **AD(Active Directory) 하위 도메인** *1.ad.example.com*
2. **AD root 도메인** *ad.example.com*
3. **IdM domain** *idm.example.com*



참고

로컬 SSSD 구성의 도메인 확인 순서는 모든 글로벌 및 ID 보기 도메인 확인 순서를 재정의합니다.

사전 요구 사항

- AD 환경을 사용하여 신뢰를 구성했습니다.

절차

1.

텍스트 편집기에서 **/etc/sss/sssd.conf** 파일을 엽니다.

2.

파일의 **[sssd]** 섹션에서 **domain_resolution_order** 옵션을 설정합니다.

```
domain_resolution_order = subdomain1.ad.example.com, ad.example.com,
idm.example.com
```

3.

파일을 저장하고 종료합니다.

4.

SSSD 서비스를 다시 시작하여 새 구성 설정을 로드합니다.

```
[root@client2 ~]# systemctl restart sssd
```

검증 단계

•

subdomain1.ad.example.com 도메인에서 짧은 이름만 사용하여 사용자에 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)
```

74장. IDM에서 AD 사용자 주체 이름을 사용한 인증 활성화

74.1. IDM에서 신뢰하는 ADFOREST의 사용자 주요 이름

IdM(Identity Management) 관리자는 AD 사용자가 대체 UUPN(User Principal Names)을 사용하여 IdM 도메인의 리소스에 액세스할 수 있습니다. UPN은 AD 사용자가 `user_name@KERBEROS-REALM` 형식으로 인증하는 대체 사용자 로그인입니다. AD 관리자는 AD 포리스트에서 추가 Kerberos 별칭과 UPN 접미사를 둘 다 구성할 수 있으므로 `user_name` 및 `KERBEROS-REALM` 모두에 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 AD에서 `EXAMPLE.COM` 을 대체 UPN으로 구성할 수 있습니다. 대체 UPN(엔터프라이즈 UPN이라고도 함)은 회사에서 최근에 병합을 경험했으며 사용자에게 통합 로그인 네임스페이스를 제공하고자 하는 경우 특히 편리합니다.

UPN 접미사는 AD forest 루트에 정의된 경우에만 IdM에 대해 표시됩니다. AD 관리자는 Active Directory Domain and Trust 유틸리티 또는 PowerShell 명령줄 도구를 사용하여 UPN을 정의할 수 있습니다.

참고

사용자에 대한 UPN 접미사를 구성하려면 Active Directory Domain and Trust 유틸리티와 같이 오류 유효성 검사를 수행하는 툴을 사용하는 것이 좋습니다.

Active Directory는 이러한 작업을 검증하지 않기 때문에 `ldapmodify` 명령을 사용하여 사용자에 대한 `userPrincipalName` 속성을 설정하는 등 낮은 수준의 수정을 통해 UPN을 구성하는 것을 권장하고 있습니다.

AD 측에서 새 UPN을 정의한 후 IdM 서버에서 `ipa trust-fetch-domains` 명령을 실행하여 업데이트된 UPN을 검색합니다. [AD UPN이 IdM에서 최신 상태인지를 참조하십시오.](#)

IdM은 `cn=trusted_domain_name,cn=ad,cn=trusts,dc=idm,dc=example,dc=com` 의 subtree 속성 `ipaNTAdditionalSuffixes` 에 도메인의 UPN 접미사를 저장합니다.

추가 리소스



[AD forest 루트에서 UPN 접미사 설정을 스크립팅하는 방법](#)

- **AD 사용자 항목을 수동으로 수정하고 UPN 접미사 검증을 무시하는 방법**
- **신뢰 컨트롤러 및 신뢰 에이전트**

74.2. AD UPN이 IDM에서 최신 상태인지 확인

신뢰할 수 있는 **Active Directory(AD)** 포리스트에 **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. 빈 출력은 다음과 같이 예상됩니다.**

```
[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 UPN을 가져왔는지 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
```


출력에 **example.com UPN** 접미사가 이제 **ad.example.com** 영역 항목의 일부임을 보여줍니다.

74.3. AD UPN 인증 문제의 문제 해결 데이터 수집

다음 절차에서는 **AD(Active Directory)** 환경과 **IdM** 환경의 **User Principal Name(UPN)** 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 닫습니다.

4. 텍스트 편집기에서 **/etc/ipa/server.conf** 구성 파일을 엽니다. 해당 파일이 없는 경우 파일을 만듭니다.

5. 파일에 다음 콘텐츠를 추가합니다.

```
[global]
debug = True
```

6. **/etc/ipa/server.conf** 파일을 저장하고 닫습니다.

7.

Apache 웹 서버를 다시 시작하여 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

- 추가 문제 해결 단계는 **AD UPN 인증 문제에 대한 문제 해결 데이터**를 수집하는 데 **rpcclient**를 사용하여 기술 자료 문서를 참조하십시오.

75장. IDM에서 정식화된 DNS 호스트 이름 사용

잠재적인 보안 위험을 피하기 위해 **IdM(Identity Management)** 클라이언트에서 **DNS** 표준이 기본적으로 비활성화되어 있습니다. 예를 들어 공격자가 도메인의 **DNS** 서버와 호스트를 제어하는 경우 공격자는 **demo** 와 같은 짧은 호스트 이름을 유발하여 **malicious.example.com** 과 같은 손상된 호스트로 확인할 수 있습니다. 이 경우 사용자는 예상과 다른 서버에 연결합니다.

이 섹션에서는 **IdM** 클라이언트에서 정식 호스트 이름을 사용하는 방법에 대해 설명합니다.

75.1. 호스트 주체에 별칭 추가

기본적으로 **ipa-client-install** 명령을 사용하여 등록된 **IdM(Identity Management)** 클라이언트는 서비스 주체에서 짧은 호스트 이름을 사용할 수 없습니다. 예를 들어, 사용자는 서비스에 액세스할 때 **host /demo@EXAMPLE.COM** 대신 **host /demo.example.com@EXAMPLE.COM** 만 사용할 수 있습니다.

이 섹션에서는 **Kerberos** 주체에 별칭을 추가하는 방법을 설명합니다. 또는 **/etc/krb5.conf** 파일에서 호스트 이름의 정식화를 활성화할 수 있습니다. 자세한 내용은 [75.2절. “클라이언트에서 서비스 주체에서 호스트 이름의 정식화 활성화”](#)의 내용을 참조하십시오.

사전 요구 사항

- **IdM** 클라이언트가 설치되어 있습니다.
- 호스트 이름은 네트워크에서 고유합니다.

절차

1. **admin** 사용자로 **IdM**에 인증합니다.

```
$ kinit admin
```

2. 호스트 주체에 별칭을 추가합니다. 예를 들어 **demo** 별칭을 **demo.example.com** 호스트 주체에 추가하려면 다음을 수행합니다.

```
$ ipa host-add-principal demo.example.com --principal=demo
```

75.2. 클라이언트에서 서비스 주체에서 호스트 이름의 정식화 활성화

이 섹션에서는 클라이언트의 서비스 주체에서 호스트 이름의 정식화를 활성화하는 방법을 설명합니다.

75.1절. “호스트 주체에 별칭 추가”에 설명된 대로 호스트 주체 별칭을 사용하는 경우 정식화를 활성화할 필요가 없습니다.

사전 요구 사항

- **IdM(Identity Management)** 클라이언트가 설치되어 있습니다.
- **root** 사용자로 **IdM** 클라이언트에 로그인했습니다.
- 호스트 이름은 네트워크에서 고유합니다.

절차

1. **/etc/krb5.conf** 파일의 **[libdefaults]** 섹션에서 **dns_canonicalize_hostname** 매개변수를 **false**로 설정합니다.

```
[libdefaults]
...
dns_canonicalize_hostname = true
```

75.3. DNS 호스트 이름 정식화가 활성화된 호스트 이름을 사용하는 옵션

75.2절. “클라이언트에서 서비스 주체에서 호스트 이름의 정식화 활성화”에 설명된 대로 **/etc/krb5.conf** 파일에 **dns_canonicalize_hostname = true**를 설정한 경우 서비스 주체에서 호스트 이름을 사용할 때 다음 옵션이 있습니다.

- **IdM(Identity Management)** 환경에서는 서비스 주체(예: **host /demo.example.com@EXAMPLE.COM**)에서 전체 호스트 이름을 사용할 수 있습니다.
- **IdM**이 없는 환경에서는 **RHEL** 호스트를 **AD(Active Directory)** 도메인의 구성원으로 하는 경우 더 이상 고려할 필요가 없습니다. **DC(AD 도메인 컨트롤러)**는 **AD**에 등록된 시스템의 **NetBIOS** 이름에 대한 서비스 주체를 자동으로 생성하기 때문입니다.

76장. ANSIBLE 플레이북을 사용하여 IDM에서 글로벌 DNS 구성 관리

Red Hat Ansible Engine dnsconfig 모듈을 사용하면 **IdM(Identity Management) DNS**에 대한 글로벌 구성을 구성할 수 있습니다. 글로벌 **DNS** 구성에 정의된 설정은 모든 **IdM DNS** 서버에 적용됩니다. 그러나 글로벌 구성은 특정 **IdM DNS** 영역의 구성보다 우선 순위가 낮습니다.

dnsconfig 모듈은 다음 변수를 지원합니다.

- 글로벌 전달자, 특히 해당 **IP** 주소 및 통신에 사용되는 포트입니다.
- 글로벌 전달 정책: **only**, **first** 또는 **none**. 이러한 유형의 **DNS** 전달 정책에 대한 자세한 내용은 [IdM의 DNS 전달 정책을 참조하십시오](#).
- 정방향 조회 및 역방향 조회 영역의 동기화.

사전 요구 사항

- **DNS** 서비스는 **IdM** 서버에 설치되어 있습니다. 통합된 **DNS**로 **IdM** 서버를 설치하는 방법에 대한 자세한 내용은 다음 링크 중 하나를 참조하십시오.
 - [IdM 서버 설치: 통합 DNS, 루트 CA로 통합 CA 포함](#)
 - [IdM 서버 설치: 통합 DNS, 외부 CA가 루트 CA로 포함](#)
 - [IdM 서버 설치: CA 없이 통합 DNS 사용](#)

이 장에는 다음 섹션이 포함되어 있습니다.

- [IdM을 통해 NetworkManager에서 /etc/resolv.conf의 글로벌 전달자를 제거하지 않는 방법](#)

- **Ansible을 사용하여 IdM에 DNS 글로벌 전달자가 있는지 확인**
- **Ansible을 사용하여 IdM에 DNS 글로벌 전달자가 없는지 확인**
- **IdM의 DNS 전달 정책소개**
- **Ansible 플레이북을 사용하여 IdM DNS 글로벌 구성에 전달 첫 번째 정책이 설정되었는지 확인합니다.**
- **Ansible 플레이북을 사용하여 글로벌 전달자가 IdM DNS에서 비활성화되었는지 확인합니다.**
- **Ansible 플레이북을 사용하여 IdM DNS에서 정방향 및 역방향 조회 영역의 동기화가 비활성화되었는지 확인합니다.**

76.1. IDM을 통해 NETWORKMANAGER에서 /ETC/RESOLV.CONF의 글로벌 전달자를 제거하지 않는 방법

통합 DNS로 IdM(Identity Management)을 설치하면 127.0.0.1 localhost 주소를 가리키도록 /etc/resolv.conf 파일을 구성합니다.

```
# Generated by NetworkManager
search idm.example.com
nameserver 127.0.0.1
```

DHCP(Dynamic Host Configuration Protocol) 를 사용하는 네트워크와 같은 특정 환경에서 NetworkManager 서비스는 /etc/resolv.conf 파일의 변경 사항을 되돌릴 수 있습니다. DNS 구성을 영구적으로 만들기 위해 IdM DNS 설치 프로세스는 다음과 같은 방식으로 NetworkManager 서비스를 구성합니다.

1. **DNS 설치 스크립트는 검색 순서 및 DNS 서버 목록을 제어하는 /etc/NetworkManager/conf.d/zzz-ipa.conf NetworkManager 구성 파일을 생성합니다.**

```
# auto-generated by IPA installer
[main]
dns=default

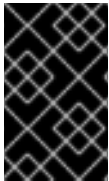
[global-dns]
```

```
searches=$DOMAIN
```

```
[global-dns-domain-*]  
servers=127.0.0.1
```

2.

NetworkManager 서비스가 다시 로드됩니다. 이 서비스는 항상 **/etc/NetworkManager/conf.d/** 디렉터리에 있는 마지막 파일에서 설정을 사용하여 **/etc/resolv.conf** 파일을 만듭니다. 이 경우 **zzz-ipa.conf** 파일입니다.



중요

/etc/resolv.conf 파일을 수동으로 수정하지 마십시오.

76.2. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에 **DNS** 글로벌 전달자가 있는지 확인하는 방법에 대해 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 **IP(인터넷 프로토콜) v4** 주소가 **7.7**이고 **IP v6** 주소 **2001:db8::1:0**에 **DNS** 글로벌 전달자가 **DNS** 서버에 있는지 확인합니다. 포트 **53**에.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

절차

1.

/usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2.

인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **forwarders-absent.yml Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. 편집을 위해 **ensure-presence-of-a-global-forwarder.yml** 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

- a. 플레이북의 **name** 변수를 플레이북으로 변경하여 **IdM DNS**에 글로벌 전달자가 있는지 확인합니다.

- b. **tasks** 섹션에서 작업 이름을 변경하여 **DNS** 글로벌 전달자가 포트 **53**에 **7.7.** 및 **2001:db8::1:0**으로 있는지 확인합니다.

- c. **ipadnsconfig** 부분의 **forwarders** 섹션에서 다음을 수행합니다.

- i. 첫 번째 **ip_address** 값을 글로벌 전달자의 **IPv4** 주소로 변경합니다. **7.7.9.9**.

- ii. 두 번째 **ip_address** 값을 글로벌 전달자의 **IPv6** 주소로 변경합니다. **2001:db8::1:0**.

- iii. **port** 값이 **53**으로 설정되어 있는지 확인합니다.

- d. 상태를 **present**로 변경합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
```

```
- name: Playbook to ensure the presence of a global forwarder in IdM DNS
  hosts: ipaserver
  become: true
```

```
tasks:
```

```
- name: Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0
```



```
on port 53
  ipadnsconfig:
    forwarders:
      - ip_address: 7.7.9.9
      - ip_address: 2001:db8::1:0
    port: 53
    state: present
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-presence-of-a-global-forwarder.yml
```

추가 리소스

- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용할 수 있는 **README-dnsconfig.md** Markdown 파일에서 **ansible-freeipa ipadnsconfig** 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 **ipadnsconfig** 변수 정의도 포함되어 있습니다.

76.3. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 없는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에 DNS 글로벌 전달자가 없는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 IP(인터넷 프로토콜) v4 주소 8.8.6.6 및 IP v6 주소 2001:4860:4860::8800의 DNS 글로벌 전달자가 없는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2.

인벤토리 파일을 열고 구성할 IdM 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **forwarders-absent.yml** Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. 편집을 위해 **ensure-absence-of-a-global-forwarder.yml** 파일을 엽니다.
5. 다음 변수를 설정하여 파일을 조정합니다.
 - a. 플레이북의 **name** 변수를 플레이북으로 변경하여 IdM DNS에서 글로벌 전달자가 없는지 확인합니다.
 - b. **tasks** 섹션에서 **DNS** 글로벌 전달자가 없는지 확인하고 포트 **53**에 **2001:4860:4860::8800**으로 작업 이름을 변경합니다.
 - c. **ipadnsconfig** 부분의 **forwarders** 섹션에서 다음을 수행합니다.
 - i. 첫 번째 **ip_address** 값을 글로벌 전달자의 **IPv4** 주소로 변경합니다. **8.8.6.6**.
 - ii. 두 번째 **ip_address** 값을 글로벌 전달자의 **IPv6** 주소로 변경합니다. **2001:4860:4860::8800**.
 - iii. **port** 값이 **53**으로 설정되어 있는지 확인합니다.
 - d. 상태가 **absent**로 설정되어 있는지 확인합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```

---
- name: Playbook to ensure the absence of a global forwarder in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure the absence of a DNS global forwarder to 8.8.6.6 and
      2001:4860:4860::8800 on port 53
      ipadnsconfig:
        forwarders:
          - ip_address: 8.8.6.6
          - ip_address: 2001:4860:4860::8800
          port: 53
          state: absent

```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-absence-of-a-global-forwarder.yml
```

추가 리소스

•

`/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용할 수 있는 **README-dnsconfig.md** Markdown 파일에서 **ansible-freeipa ipadnsconfig** 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 **ipadnsconfig** 변수 정의도 포함되어 있습니다.

76.4. IDM의 DNS 전달 정책

IdM은 첫 번째 및 표준 **BIND** 전달 정책과 **none** IdM별 전달 정책을 지원합니다.

전달 먼저 (기본값)

IdM **BIND** 서비스는 **DNS** 쿼리를 구성된 전달자로 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 **BIND**는 인터넷에서 서버를 사용하여 재귀적 해결으로 대체합니다. **forward first** 정책은 기본 정책이며 **DNS** 트래픽을 최적화하는 데 적합합니다.

앞으로만

IdM **BIND** 서비스는 **DNS** 쿼리를 구성된 전달자로 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 **BIND**에서 클라이언트에 오류를 반환합니다. **DNS** 구성이 분할된 환경에 만 전달 정책이 권장됩니다.

없음 (전달 비활성화)

DNS 쿼리는 **none** 전달 정책으로 전달되지 않습니다. 전달 비활성화는 전역 전달 구성의 영역별 재정의에서만 유용합니다. 이 옵션은 **BIND** 구성에서 비어 있는 전달자 목록을 지정하는 것과 동일한 **IdM**입니다.

참고

전달을 사용하여 **IdM**의 데이터를 다른 **DNS** 서버의 데이터와 결합할 수 없습니다. **IdM DNS**에서 기본 영역의 특정 하위 영역에 대한 쿼리만 전달할 수 있습니다.

기본적으로 쿼리된 **DNS** 이름이 **IdM** 서버에 권한이 있는 영역에 속하는 경우 **BIND** 서비스는 다른 서버로 쿼리를 전달하지 않습니다. 이러한 상황에서 쿼리된 **DNS** 이름을 **IdM** 데이터베이스에서 찾을 수 없는 경우 **NXDOMAIN** 응답이 반환됩니다. 전달은 사용되지 않습니다.

예 76.1. 시나리오 예

IdM 서버에는 **test.example**에 대한 권한이 있습니다. **DNS** 영역. **BIND**는 **192.0.2.254** IP 주소를 사용하여 **DNS** 서버로 쿼리를 전달하도록 구성됩니다.

클라이언트가 존재하지 않는 **test.example**에 대한 쿼리를 보낼 때. **DNS** 이름 **BIND**는 **IdM** 서버가 **test.example**. 영역에 대해 권한이 있음을 감지하고 **192.0.2.254**. 서버로 쿼리를 전달하지 않습니다. 결과적으로 **DNS** 클라이언트는 **NXDomain** 오류 메시지를 수신하여 쿼리된 도메인이 없음을 사용자에게 알립니다.

76.5. ANSIBLE 플레이북을 사용하여 **IDM DNS** 글로벌 구성에 전달 첫 번째 정책이 설정되었는지 확인합니다.

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM DNS**의 글로벌 전달 정책이 먼저 전달되도록 설정하는 방법에 대해 설명합니다.

전달 첫 번째 **DNS** 전달 정책을 사용하는 경우 **DNS** 쿼리가 구성된 전달자로 전달됩니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 **BIND**는 인터넷에서 서버를 사용하여 재귀적 해결로 대체합니다. **forward first** 정책은 기본 정책입니다. 트래픽 최적화에 적합합니다.

사전 요구 사항

-

절차를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 자세한 내용은 [ansible-freeipa 패키지 설치](#)를 참조하십시오.

- **IdM 관리자 암호를 알고 있습니다.**
- **IdM 환경에는 통합된 DNS 서버가 포함되어 있습니다.**

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.**

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. **인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.**

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

3. **set-configuration.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.**

```
$ cp set-configuration.yml set-forward-policy-to-first.yml
```

4. **편집을 위해 set-forward-policy-to-first.yml 파일을 엽니다.**

5. **ipadnsconfig 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.**

- **ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.**
- **forward_policy 변수를 먼저 로 설정합니다.**

원본 플레이북의 다른 모든 행을 관련성이 없는 모든 행을 삭제합니다. 이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to set global forwarding policy to first
```

```
hosts: ipaserver
become: true
```

```
tasks:
- name: Set global forwarding policy to first.
  ipadnsconfig:
    ipaadmin_password: Secret123
    forward_policy: first
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file set-forward-policy-to-first.yml
```

추가 리소스

•

IdM DNS에서 사용 가능한 전달 정책 유형에 대한 자세한 내용은 **IdM의 DNS 전달 정책을 참조하십시오.**

•

ansible-freeipa ipadnsconfig 모듈을 사용하는 샘플 **Ansible** 플레이북은 **/usr/share/doc/ansible-freeipa/** 디렉터리에 있는 **README-dnsconfig.md** Markdown 파일을 참조하십시오. 파일에는 **ipadnsconfig** 변수 정의도 포함되어 있습니다.

•

ipadnsconfig 모듈을 사용하는 샘플 **Ansible** 플레이북은 **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** 디렉터리를 참조하십시오.

76.6. ANSIBLE 플레이북을 사용하여 글로벌 전달자가 IDENTITY MANAGEMENT DNS에서 비활성화되었는지 확인합니다.

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM DNS**에서 글로벌 전달자가 비활성화되는 방법을 설명합니다. 비활성화는 **forward_policy** 변수를 **none** 으로 설정하여 수행됩니다.

전역 전달자를 비활성화하면 **DNS** 쿼리가 전달되지 않습니다. 전달 비활성화는 전역 전달 구성의 영역별 재정의에서만 유용합니다. 이 옵션은 **BIND** 구성에서 비어 있는 전달자 목록을 지정하는 것과 동일한 **IdM**입니다.

사전 요구 사항

•

절차를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 자

세한 내용은 [ansible-freeipa 패키지 설치](#)를 참조하십시오.

- **IdM 관리자 암호를 알고 있습니다.**
- **IdM 환경에는 통합된 DNS 서버가 포함되어 있습니다.**

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. **인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다.**
예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **disable-global-forwarders.yml Ansible 플레이북 파일의 사본을 만듭니다.** 예를 들면 다음과 같습니다.

```
$ cp disable-global-forwarders.yml disable-global-forwarders-copy.yml
```

4. **편집할 disable-global-forwarders-copy.yml 파일을 엽니다.**

5. **ipadnsconfig 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.**

- **ipaadmin_password** 변수를 **IdM 관리자 암호**로 설정합니다.
- **forward_policy** 변수를 **none** 으로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to disable global DNS forwarders
```

```

hosts: ipaserver
become: true

tasks:
- name: Disable global forwarders.
  ipadnsconfig:
    ipadmin_password: Secret123
    forward_policy: none

```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file disable-global-forwarders-copy.yml
```

추가 리소스

- **IdM DNS**에서 사용 가능한 전달 정책 유형에 대한 자세한 내용은 IdM의 [DNS 전달 정책을 참조하십시오](#).
- **ansible-freeipa ipadnsconfig** 모듈을 사용하는 샘플 Ansible 플레이북은 `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README- dnsconfig.md** Markdown 파일을 참조하십시오. 파일에는 **ipadnsconfig** 변수 정의도 포함되어 있습니다.
- **ipadnsconfig** 모듈을 사용하는 샘플 Ansible 플레이북은 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 디렉터리를 참조하십시오.

76.7. ANSIBLE 플레이북을 사용하여 **IDM DNS**에서 정방향 및 역방향 조회 영역의 동기화가 비활성화되었는지 확인합니다.

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 정방향 및 역방향 조회 영역이 **IdM DNS**에서 동기화되지 않는 방법을 설명합니다.

사전 요구 사항

- 절차를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 자세한 내용은 [ansible-freeipa 패키지 설치](#)를 참조하십시오.

- **IdM 관리자 암호를 알고 있습니다.**
- **IdM 환경에는 통합된 DNS 서버가 포함되어 있습니다.**

절차

1. **`/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 디렉터리로 이동합니다.**

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. **인벤토리 파일을 열고 구성할 IdM 서버가 `[ipaserver]` 섹션에 나열되어 있는지 확인합니다. 예를 들어 `server.idm.example.com`을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.**

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

3. **`disallow -reverse-sync.yml` Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.**

```
$ cp disallow-reverse-sync.yml disallow-reverse-sync-copy.yml
```

4. **편집을 위해 `disallow-reverse-sync-copy.yml` 파일을 엽니다.**

5. **`ipadnsconfig` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.**

- **`ipaadmin_password` 변수를 IdM 관리자 암호로 설정합니다.**
- **`allow_sync_ptr` 변수를 `no` 로 설정합니다.**

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to disallow reverse record synchronization
  hosts: ipaserver
  become: true
```

tasks:**- name: Disallow reverse record synchronization.****ipadnsconfig:****ipaadmin_password: Secret123****allow_sync_ptr: no**

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

\$ ansible-playbook -v -i inventory.file disallow-reverse-sync-copy.yml

추가 리소스

•

ansible-freeipa ipadnsconfig 모듈을 사용하는 샘플 **Ansible** 플레이북은 **/usr/share/doc/ansible-freeipa/** 디렉터리에 있는 **README-dnsconfig.md** Markdown 파일을 참조하십시오. 파일에는 **ipadnsconfig** 변수 정의도 포함되어 있습니다.

•

ipadnsconfig 모듈을 사용하는 샘플 **Ansible** 플레이북은 **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** 디렉터리를 참조하십시오.

77장. IDM에서 DNS 영역 관리

IdM(Identity Management) 관리자는 **IdM DNS** 영역의 작동 방식을 관리할 수 있습니다. 이 장에서는 다음 주제 및 절차를 설명합니다.

- **IdM에서 지원되는 DNS 영역 유형은 무엇입니까?**
 - **IdM 웹 UI를 사용하여 기본 IdM DNS 영역을 추가하는 방법**
 - **IdM CLI를 사용하여 기본 IdM DNS 영역을 추가하는 방법**
 - **IdM 웹 UI를 사용하여 기본 IdM DNS 영역을 제거하는 방법**
 - **IdM CLI를 사용하여 기본 IdM DNS 영역을 제거하는 방법**
- **IdM에서 구성할 수 있는 DNS 속성은 무엇입니까?**
 - **IdM 웹 UI에서 이러한 속성을 구성하는 방법**
 - **IdM CLI에서 이러한 속성을 구성하는 방법**
- **IdM에서 영역 전송이 작동하는 방법**
 - **IdM 웹 UI에서 영역 전송을 허용하는 방법**
 - **IdM CLI에서 영역 전송을 허용하는 방법**

사전 요구 사항

- **DNS 서비스는 IdM 서버에 설치되어 있습니다. 통합된 DNS로 IdM 서버를 설치하는 방법에 대한 자세한 내용은 다음 링크 중 하나를 참조하십시오.**

- **IdM 서버 설치: 통합 DNS, 루트 CA로 통합 CA 포함**
- **IdM 서버 설치: 통합 DNS, 외부 CA가 루트 CA로 포함**
- **IdM 서버 설치: CA 없이 통합 DNS 사용**

77.1. 지원되는 DNS 영역 유형

IdM(Identity Management)은 두 가지 유형의 **DNS** 영역, 즉 기본 및 전달 영역을 지원합니다. 이 섹션에서는 이러한 두 가지 유형의 영역에 대해 설명하고 **DNS** 전달 예제 시나리오를 포함합니다.



참고

이 가이드에서는 **Microsoft Windows DNS**에 사용되는 용어와 다른 영역 유형에 **BIND** 용어를 사용합니다. **BIND**의 기본 영역은 **Microsoft Windows DNS**에서 정방향 조회 영역 및 역방향 조회 영역과 동일한 목적을 제공합니다. **BIND**의 전달 영역은 **Microsoft Windows DNS**의 조건부 전달자 와 동일한 목적을 제공합니다.

기본 DNS 영역

기본 **DNS** 영역에는 권한 있는 **DNS** 데이터가 포함되어 있으며 동적 **DNS** 업데이트를 허용할 수 있습니다. 이 동작은 표준 **BIND** 구성의 유형 마스터 설정과 동일합니다. **ipa dnszone-*** 명령을 사용하여 기본 영역을 관리할 수 있습니다.

표준 **DNS** 규칙을 준수하려면 모든 기본 영역에 권한 시작 (**SOA**) 및 **NS**(이름 서버) 레코드가 포함되어야 합니다. **DNS** 영역을 생성할 때 **IdM**은 이러한 레코드를 자동으로 생성하지만 적절한 위임을 생성하려면 **NS** 레코드를 상위 영역에 수동으로 복사해야 합니다.

표준 **BIND** 동작에 따라 서버가 권한이 없는 이름에 대한 쿼리가 다른 **DNS** 서버로 전달됩니다. **forwarders**와 같은 이러한 **DNS** 서버는 쿼리에 대한 권한이 있을 수도 있고 그렇지 않을 수 있습니다.

예 77.1. DNS 전달 시나리오의 예

IdM 서버에는 **test.example.** 기본 영역이 포함되어 있습니다. 이 영역에는 **sub.test.example.** 이름에 대한 **NS** 위임 레코드가 포함됩니다. 또한 **test.example.** 영역은 **sub.test.example** 하위 영역에 대해 **192.0.2. 254** 전달자 **IP** 주소로 구성됩니다.

비존재 **.test.example.** 이름을 쿼리하는 클라이언트는 **NXDomain** 응답을 수신하고 **IdM** 서버가 이 이름에 대한 권한이 있기 때문에 전달이 발생하지 않습니다.

반면, **IdM** 서버가 이 이름에 대한 권한이 없기 때문에 **host1.sub.test.example .** 이름을 쿼리하는 것은 구성된 **forwarder 192.0.2.254** 로 전달됩니다.

DNS 영역 전달

IdM의 관점에서, 전달 **DNS** 영역에는 권한 있는 데이터가 포함되어 있지 않습니다. 실제로 **forward "zone"**에는 일반적으로 두 가지 정보만 포함됩니다.

- 도메인 이름
- 도메인과 연결된 **DNS** 서버의 **IP** 주소

정의된 도메인에 속하는 이름에 대한 모든 쿼리는 지정된 **IP** 주소로 전달됩니다. 이 동작은 표준 **BIND** 구성의 **type forward** 설정과 동일합니다. **ipa dnsforwardzone-*** 명령을 사용하여 전달 영역을 관리할 수 있습니다.

정방향 **DNS** 영역은 **IdM-AD(Active Directory)** 신뢰 컨텍스트에서 특히 유용합니다. **IdM DNS** 서버에 **idm.example.com** 영역에 대한 권한이 있고 **AD DNS** 서버가 **ad.example.com** 영역에 대해 권한이 있는 경우 **ad.example.com** 은 **idm.example.com** 기본 영역의 **DNS** 전달 영역입니다. 즉, 쿼리가 **somehost.ad.example.com**의 **IP** 주소에 대한 **IdM** 클라이언트에서 제공되면 쿼리가 **ad.example.com** **IdM DNS** 전달 영역에 지정된 **AD** 도메인 컨트롤러로 전달됩니다.

77.2. IDM 웹 UI에서 기본 DNS 영역 추가

이 섹션에서는 **IdM(Identity Management)** 웹 UI를 사용하여 기본 **DNS** 영역을 추가하는 방법을 설명합니다.

사전 요구 사항

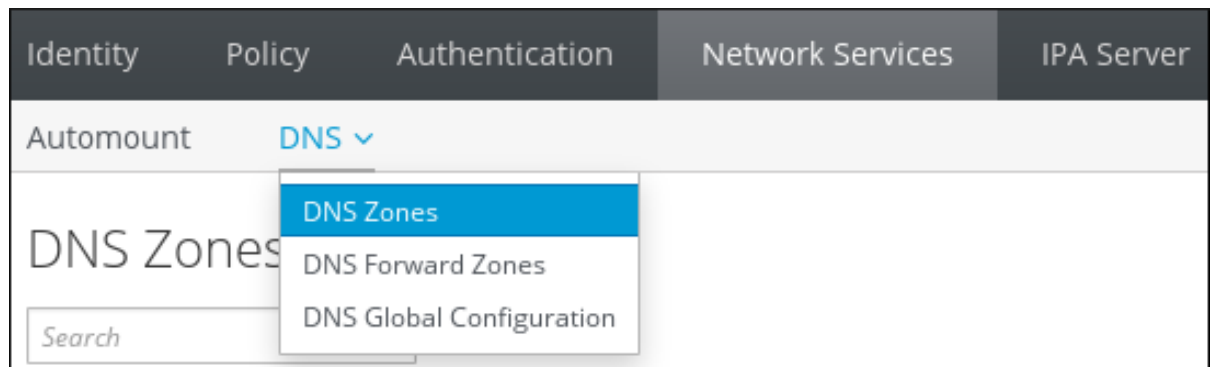
- **IdM** 관리자로 로그인했습니다.

절차

1.

IdM 웹 UI에서 네트워크 서비스 → DNS → DNS 영역을 클릭합니다.

그림 77.1. IdM DNS 기본 영역 관리



2.

모든 영역 목록 상단에 있는 **Add(추가)**를 클릭합니다.

3.

영역 이름을 입력합니다.

그림 77.2. 새 IdM 기본 영역 입력

4.

추가를 클릭합니다.

77.3. IDM CLI에서 기본 DNS 영역 추가

이 섹션에서는 **IdM(Identity Management) CLI(명령줄 인터페이스)**에 기본 **DNS** 영역을 추가하는 방법을 설명합니다.

사전 요구 사항

- **IdM 관리자로 로그인했습니다.**

절차

- **ipa dnszone-add 명령은 DNS 도메인에 새 영역을 추가합니다. 새 영역을 추가하려면 새 하위 도메인의 이름을 지정해야 합니다. 명령을 사용하여 하위 도메인 이름을 직접 전달할 수 있습니다.**

```
$ ipa dnszone-add newzone.idm.example.com
```

이름을 **ipa dnszone-add** 에 전달하지 않으면 스크립트에서 자동으로 프롬프트를 표시합니다.

추가 리소스

- **ipa dnszone-add 명령은 다양한 명령줄 옵션도 허용합니다. 이러한 옵션의 전체 목록을 보려면 **ipa dnszone-add --help** 명령을 실행합니다.**

77.4. IDM 웹 UI에서 기본 DNS 영역 제거

이 섹션에서는 **IdM(Identity Management)**에서 **IdM(Identity Web UI)**에서 기본 **DNS** 영역을 제거하는 방법을 설명합니다.

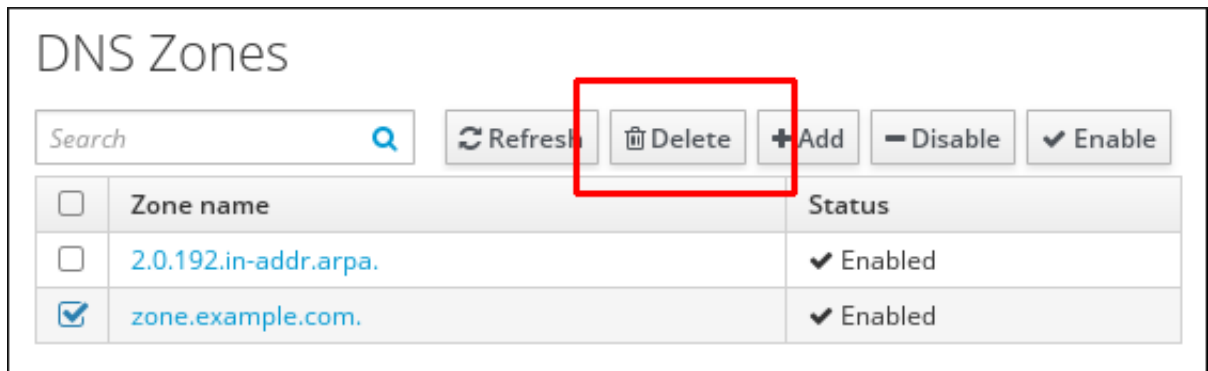
사전 요구 사항

- **IdM 관리자로 로그인했습니다.**

절차

1. **IdM 웹 UI에서 네트워크 서비스 → DNS → DNS 영역을 클릭합니다.**
2. **영역 이름별로 확인란을 선택하고 **Delete(삭제)**를 클릭합니다.**

그림 77.3. 기본 DNS 영역 제거



3.

Remove DNS zones(DNS 영역 제거) 대화 상자 창에서 선택한 영역을 삭제하도록 확인합니다.

77.5. IDM CLI에서 기본 DNS 영역 제거

이 섹션에서는 **IdM(명령줄 인터페이스)**을 사용하여 **IdM(Identity Management)**에서 기본 **DNS** 영역을 제거하는 방법을 설명합니다.

사전 요구 사항

- **IdM 관리자**로 로그인했습니다.

절차

- 기본 **DNS** 영역을 제거하려면 **ipa dnszone-del** 명령을 입력한 다음 제거할 영역의 이름을 입력합니다. 예를 들면 다음과 같습니다.

```
$ ipa dnszone-del idm.example.com
```

77.6. DNS 구성 우선순위

다음 수준에서 많은 **DNS** 구성 옵션을 구성할 수 있습니다. 각 수준은 다른 우선 순위를 갖습니다.

영역별 구성

IdM에 정의된 특정 영역에 고유한 구성 수준이 우선 순위가 가장 높습니다. **ipa dnszone-*** 및 **ipa dnsforwardzone-*** 명령을 사용하여 영역별 구성을 관리할 수 있습니다.

서버당 구성

IdM 서버를 설치하는 동안 서버별 전달자를 정의해야 합니다. **ipa dnsserver-*** 명령을 사용하여 서버당 전달자를 관리할 수 있습니다. 복제본을 설치할 때 서버별 전달자를 설정하지 않으려면 **--no-forwarder** 옵션을 사용할 수 있습니다.

글로벌 DNS 구성

영역별 구성이 정의되지 않은 경우 **IdM**은 **LDAP**에 저장된 글로벌 **DNS** 구성을 사용합니다. **ipa dnsconfig-*** 명령을 사용하여 글로벌 **DNS** 구성을 관리할 수 있습니다. 글로벌 **DNS** 구성에 정의된 설정은 모든 **IdM DNS** 서버에 적용됩니다.

/etc/named.conf의 설정

각 **IdM DNS** 서버의 **/etc/named.conf** 파일에 정의된 구성은 우선 순위가 가장 낮습니다. 각 서버에 따라 다르며 수동으로 편집해야 합니다.

/etc/named.conf 파일은 일반적으로 로컬 **DNS** 캐시에 대한 **DNS** 전달을 지정하는 데만 사용됩니다. 다른 옵션은 위에서 언급한 영역별 글로벌 **DNS** 구성에 명령을 사용하여 관리됩니다.

여러 수준에서 동시에 **DNS** 옵션을 구성할 수 있습니다. 이 경우 우선 순위가 가장 높은 구성이 하위 수준에서 정의된 구성보다 우선합니다.

추가 리소스

- [LDAP의 서버 구성당 설정 섹션의 우선 순위 순서](#)

77.7. 기본 IDM DNS 영역의 구성 특성

IdM(Identity Management)은 새로 고침 기간, 전송 설정 또는 캐시 설정과 같은 특정 기본 구성으로 새 영역을 생성합니다. **IdM DNS 영역 속성**에서는 다음 옵션 중 하나를 사용하여 수정할 수 있는 기본 영역 구성의 특성을 찾을 수 있습니다.

- **CLI(명령줄 인터페이스)의 dnszone-mod 명령.** 자세한 내용은 [IdM CLI에서 기본 DNS 영역 구성 편집](#)을 참조하십시오.
- **IdM 웹 UI.** 자세한 내용은 [IdM 웹 UI에서 기본 DNS 영역 구성 편집](#)을 참조하십시오.
- **ipadnszone** 모듈을 사용하는 **Ansible** 플레이북. 자세한 내용은 [IdM에서 DNS 영역 관리](#)를 참조하십시오.

영역에 대한 실제 정보를 설정하는 것과 함께 설정은 **DNS** 서버가 권한 시작 (**SOA**) 레코드 항목을 처리하는 방법과 **DNS** 이름 서버에서 레코드를 업데이트하는 방법을 정의합니다.

표 77.1. IdM DNS 영역 속성

속성	명령줄 옵션	설명
인가된 이름 서버	--name-server	SOA MNAME이라고도 하는 기본 DNS 이름 서버의 도메인 이름을 설정합니다. 기본적으로 각 IdM 서버는 SOA MNAME 필드에 자체적으로 알립니다. 결과적으로 --name-server 를 사용하여 LDAP에 저장된 값은 무시됩니다.
관리자 이메일 주소	--admin-email	영역 관리자에게 사용할 이메일 주소를 설정합니다. 기본값은 호스트의 root 계정으로 설정됩니다.
SOA 직렬	--serial	SOA 레코드에 일련 번호를 설정합니다. IdM은 버전 번호를 자동으로 설정하고 사용자가 수정하지 않아야 합니다.
SOA 새로 고침	--refresh	보조 DNS 서버가 기본 DNS 서버에서 업데이트를 요청하기 전에 대기할 간격(초)을 설정합니다.
SOA 재시도	--retry	실패한 새로 고침 작업을 재시도하기 전에 대기할 시간(초)을 설정합니다.
SOA 만료	--expire	보조 DNS 서버가 작업 시도를 종료하기 전에 새로 고침 업데이트를 수행하는 시간을 초 단위로 설정합니다.
SOA 최소	--minimum	RFC 2308 에 따라 음수 캐싱의 TTL(Time to Live) 값을 초 단위로 설정합니다.
SOA 수명	--ttl	zone apex에서 레코드에 TTL(초)을 설정합니다. 예를 들어 zone example.com 에서는 example.com 이름의 모든 레코드(A, NS 또는 SOA)가 구성되지만 test.example.com 과 같은 다른 도메인 이름은 영향을 받지 않습니다.
기본 수명	--default-ttl	개별 TTL 값이 없는 영역의 모든 값에 대해 음수 캐싱을 위해 기본 TTL(Time to Live) 값을 초 단위로 설정합니다. 변경 사항을 적용한 후 모든 IdM DNS 서버에서 named-pkcs11 서비스를 다시 시작해야 합니다.
BIND 업데이트 정책	--update-policy	DNS 영역에서 허용된 권한을 클라이언트에 설정합니다.
동적 업데이트	--dynamic-update=TRUE FALSE	클라이언트의 DNS 레코드에 대한 동적 업데이트를 활성화합니다. 이 값을 false로 설정하면 IdM 클라이언트 시스템에서 IP 주소를 추가하거나 업데이트할 수 없습니다.

속성	명령줄 옵션	설명
전송 허용	--allow-transfer=string	세미콜론(;)으로 구분된 지정된 영역을 전송할 수 있는 IP 주소 또는 네트워크 이름 목록을 제공합니다. 영역 전송은 기본적으로 비활성화되어 있습니다. 기본 --allow-transfer 값은 none 입니다.
쿼리 허용	--allow-query	세미콜론(;)으로 구분된 DNS 쿼리를 실행할 수 있는 IP 주소 또는 네트워크 이름 목록을 제공합니다.
PTR 동기화 허용	--allow-sync-ptr=1 0	영역에 대한 A 또는 AAAA 레코드(전달 레코드)가 PTR(역방향) 레코드와 자동으로 동기화되는지 여부를 설정합니다.
영역 전달기	--forwarder=IP_address	DNS 영역에 구체적으로 구성된 전달자를 지정합니다. IdM 도메인에서 사용되는 모든 글로벌 전달자와는 별개입니다. 여러 전달자를 지정하려면 옵션을 여러 번 사용합니다.
전달 정책	--forward-policy=none only first	전달 정책을 지정합니다. 지원되는 정책에 대한 자세한 내용은 IdM의 DNS 전달 정책 을 참조하십시오.

77.8. IDM 웹 UI에서 기본 DNS 영역 구성 편집

이 섹션에서는 **IdM 웹 UI**를 사용하여 기본 **IdM(Identity Management) DNS**의 구성 속성을 편집하는 방법을 설명합니다.

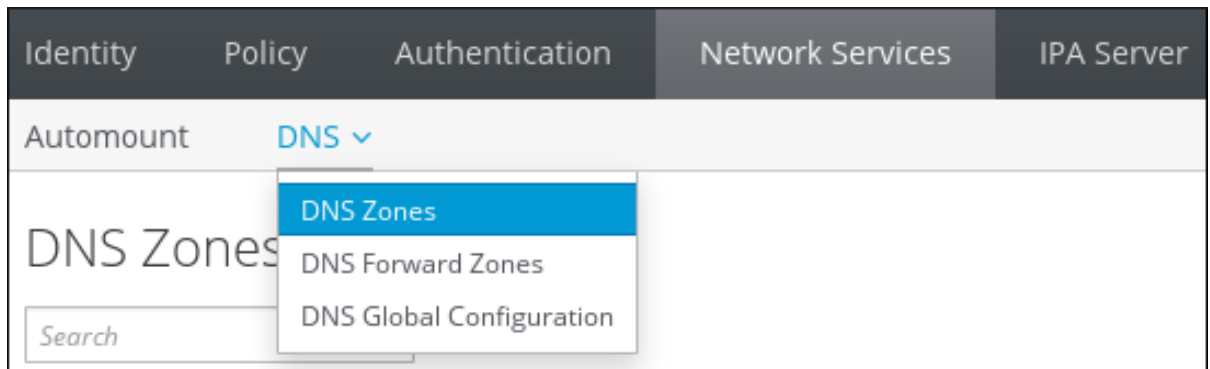
사전 요구 사항

- **IdM 관리자**로 로그인했습니다.

절차

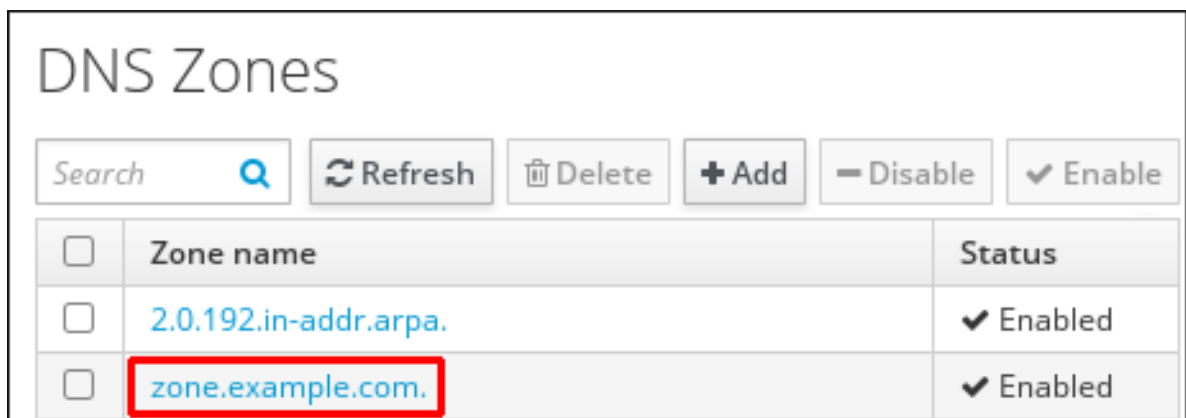
1. **IdM 웹 UI**에서 **네트워크 서비스** → **DNS** → **DNS 영역**을 클릭합니다.

그림 77.4. DNS 기본 영역 관리



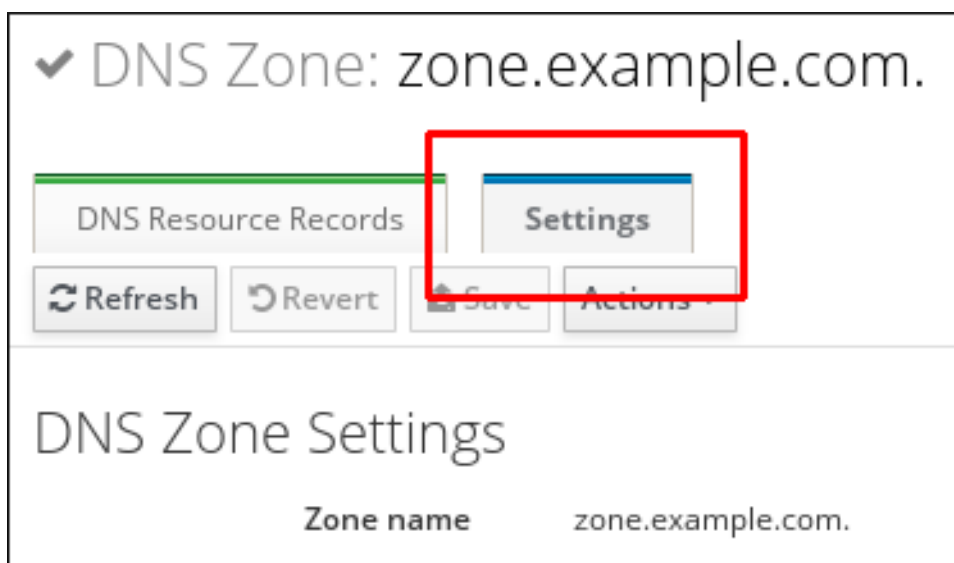
2. **DNS 영역 섹션의 모든 영역 목록에서 영역 이름을 클릭하여 DNS 영역 페이지를 엽니다.**

그림 77.5. 기본 영역 편집



3. **설정을 클릭합니다.**

그림 77.6. 기본 영역 편집 페이지의 Settings(설정) 탭



4. **필요에 따라 영역 구성을 변경합니다.**

사용 가능한 설정에 대한 자세한 내용은 [IdM DNS 영역 속성](#)을 참조하십시오.

5.

Save(저장)를 클릭하여 새 구성을 확인합니다.



참고

영역의 **TTL(Live to Live)**을 변경하는 경우 모든 **IdM DNS** 서버에서 **named-pkcs11** 서비스를 다시 시작하여 변경 사항을 적용합니다. 다른 모든 설정은 자동으로 즉시 활성화됩니다.

77.9. IDM CLI에서 기본 DNS 영역 구성 편집

이 섹션에서는 **IdM(Identity Management) CLI(명령줄 인터페이스)**를 사용하여 기본 **DNS** 영역 구성을 편집하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.

절차

- 기존 기본 **DNS** 영역을 수정하려면 **ipa dnszone-mod** 명령을 사용합니다. 예를 들어 실패한 새로 고침 작업을 다시 시도하기 전에 대기할 시간을 **1800**초로 설정하려면 다음을 수행합니다.

```
$ ipa dnszone-mod --retry 1800
```

사용 가능한 설정 및 해당 **CLI** 옵션에 대한 자세한 내용은 [IdM DNS 영역 속성](#)을 참조하십시오.

특정 설정에 수정할 **DNS** 영역 항목에 값이 없는 경우 **ipa dnszone-mod** 명령은 값을 추가합니다. 설정이 값이 없는 경우 명령은 현재 값을 지정된 값으로 덮어씁니다.



참고

영역의 **TTL(Live to Live)**을 변경하는 경우 모든 **IdM DNS** 서버에서 **named-pkcs11** 서비스를 다시 시작하여 변경 사항을 적용합니다. 다른 모든 설정은 자동으로 즉시 활성화됩니다.

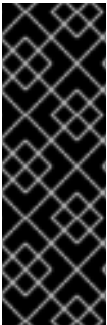
추가 리소스

- **ipa dnszone-mod** 및 해당 옵션에 대한 자세한 내용을 보려면 **ipa dnszone-mod --help** 명령을 실행합니다.

77.10. IDM의 영역 전송

이 섹션에서는 **DNS**가 통합된 **IdM(Identity Management)** 배포에서 영역 전송이 작동하는 방법에 대해 설명합니다.

이름 서버는 해당 영역에 대해 권한 있는 데이터를 유지합니다. 영역 **A DNS** 영역에 대해 권한이 있는 **DNS** 서버의 영역을 변경하는 경우 **A** 외부 영역인 **IdM DNS** 도메인의 다른 이름 서버에 변경 사항을 배포해야 합니다. 영역 전송은 이름 서버에서 다른 이름 서버로 모든 리소스 레코드를 복사합니다.



중요

IdM 통합 **DNS**는 다른 서버에서 동시에 쓸 수 있습니다. **IdM** 영역에서 권한 시작(**SOA**) 일련 번호는 개별 **IdM DNS** 서버 간에 동기화되지 않습니다. 따라서 **to-be-transferred**(전환할 예정) 영역 내에서 하나의 특정 **DNS** 서버만 사용하도록 **DNS** 서버를 전송할 수 있도록 구성합니다. 이로 인해 동기화되지 않은 **SOA** 일련 번호로 인한 영역 전송 실패가 방지됩니다.

IdM은 **RFC 5936(AXFR)** 및 **RFC 1995 (IXFR)** 표준에 따라 영역 전송을 지원합니다.

추가 리소스

- **IdM** 웹 UI에서 영역 전송을 활성화하는 방법에 대한 자세한 내용은 **IdM 웹 UI 에서 영역 전송 활성화를 참조하십시오.**
- **IdM CLI**에서 영역 전송을 활성화하는 방법에 대한 자세한 내용은 **IdM CLI 에서 영역 전송 활성화를 참조하십시오.**

77.11. IDM 웹 UI에서 영역 전송 활성화

이 섹션에서는 **IdM 웹 UI**를 사용하여 **IdM(Identity Management)**에서 영역 전송을 활성화하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.

절차

1. **IdM 웹 UI**에서 네트워크 서비스 → **DNS** → **DNS** 영역을 클릭합니다.
2. **설정을 클릭합니다.**
3. **Allow transfer (전송 허용)**에서 영역 레코드를 전송할 이름 서버를 지정합니다.

그림 77.7. 영역 전송 활성화

Allow transfer	
192.0.2.1	Undo
198.51.100.1	Undo
203.0.113.1	Undo
Add	Undo All

4. **DNS** 영역 페이지 상단에 있는 **Save(저장)**를 클릭하여 새 구성을 확인합니다.

77.12. IDM CLI에서 영역 전송 활성화

이 섹션에서는 **IdM CLI(명령줄 인터페이스)**를 사용하여 **IdM(Identity Management)**에서 영역 전송을 활성화하는 방법을 설명합니다.

사전 요구 사항

- **IdM 관리자로 로그인했습니다.**
- **보조 DNS 서버에 대한 루트 액세스 권한이 있습니다.**

절차

- **BIND 서비스에서 영역 전송을 활성화하려면 `ipa dnszone-mod` 명령을 입력하고 `--allow-transferred` 옵션을 사용하여 영역 레코드를 전송할 `to-be-transferred` 영역 외부에 있는 이름 서버 목록을 지정합니다. 예를 들면 다음과 같습니다.**

```
$ ipa dnszone-mod --allow-transfer=192.0.2.1;198.51.100.1;203.0.113.1
idm.example.com
```

검증 단계

1. **영역 전송이 활성화된 DNS 서버 중 하나에 SSH로 연결합니다.**

```
$ ssh 192.0.2.1
```

2. **`dig` 유틸리티와 같은 도구를 사용하여 IdM DNS 영역을 전송합니다.**

```
# dig @ipa-server zone_name AXFR
```

명령에서 오류를 반환하지 않으면 `zone_name` 에 대해 영역 전송을 성공적으로 활성화했습니다.

77.13. 추가 리소스

- **Red Hat Ansible Engine을 사용하여 IdM DNS 영역을 관리하는 방법에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 IdM DNS 영역 관리](#) 방법을 참조하십시오.**

78장. ANSIBLE 플레이북을 사용하여 IDM DNS 영역 관리

IdM(Identity Management) 관리자는 **ansible-freeipa** 패키지에서 사용할 수 있는 **dnszone** 모듈을 사용하여 **IdM DNS** 영역이 작동하는 방식을 관리할 수 있습니다. 이 장에서는 다음 주제 및 절차를 설명합니다.

- **IdM에서 지원되는 DNS 영역 유형은 무엇입니까?**
- **IdM에서 구성할 수 있는 DNS 속성은 무엇입니까?**
- **Ansible 플레이북을 사용하여 IdM DNS에서 기본 영역을 생성하는 방법**
- **Ansible 플레이북을 사용하여 여러 변수가 있는 기본 IdM DNS 영역이 있는지 확인하는 방법**
- **IP 주소가 제공될 때 Ansible 플레이북을 사용하여 역방향 DNS 조회 영역이 있는지 확인하는 방법**

사전 요구 사항

- **DNS 서비스는 IdM 서버에 설치되어 있습니다. Red Hat Ansible Engine을 사용하여 통합된 DNS로 IdM 서버를 설치하는 방법에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 ID 관리 서버 설치](#)를 참조하십시오.**

78.1. 지원되는 DNS 영역 유형

IdM(Identity Management)은 두 가지 유형의 **DNS** 영역, 즉 기본 및 전달 영역을 지원합니다. 이 섹션에서는 이러한 두 가지 유형의 영역에 대해 설명하고 **DNS** 전달 예제 시나리오를 포함합니다.



참고

이 가이드에서는 **Microsoft Windows DNS**에 사용되는 용어와 다른 영역 유형에 **BIND** 용어를 사용합니다. **BIND**의 기본 영역은 **Microsoft Windows DNS**에서 정방향 조회 영역 및 역방향 조회 영역과 동일한 목적을 제공합니다. **BIND**의 전달 영역은 **Microsoft Windows DNS**의 조건부 전달자 와 동일한 목적을 제공합니다.

기본 DNS 영역

기본 **DNS** 영역에는 권한 있는 **DNS** 데이터가 포함되어 있으며 동적 **DNS** 업데이트를 허용할 수 있습니다. 이 동작은 표준 **BIND** 구성의 유형 마스터 설정과 동일합니다. `ipa dnszone-*` 명령을 사용하여 기본 영역을 관리할 수 있습니다.

표준 **DNS** 규칙을 준수하려면 모든 기본 영역에 권한 시작 (**SOA**) 및 **NS**(이름 서버) 레코드가 포함되어야 합니다. **DNS** 영역을 생성할 때 **IdM**은 이러한 레코드를 자동으로 생성하지만 적절한 위임을 생성하려면 **NS** 레코드를 상위 영역에 수동으로 복사해야 합니다.

표준 **BIND** 동작에 따라 서버가 권한이 없는 이름에 대한 쿼리가 다른 **DNS** 서버로 전달됩니다. **forwarders**와 같은 이러한 **DNS** 서버는 쿼리에 대한 권한이 있을 수도 있고 그렇지 않을 수 있습니다.

예 78.1. DNS 전달 시나리오의 예

IdM 서버에는 **test.example.** 기본 영역이 포함되어 있습니다. 이 영역에는 **sub.test.example.** 이름에 대한 **NS** 위임 레코드가 포함됩니다. 또한 **test.example.** 영역은 **sub.test.example** 하위 영역에 대해 **192.0.2. 254** 전달자 **IP** 주소로 구성됩니다.

비존재 **.test.example.** 이름을 쿼리하는 클라이언트는 **NXDomain** 응답을 수신하고 **IdM** 서버가 이 이름에 대한 권한이 있기 때문에 전달이 발생하지 않습니다.

반면, **IdM** 서버가 이 이름에 대한 권한이 없기 때문에 **host1.sub.test.example .** 이름을 쿼리하는 것은 구성된 **forwarder 192.0.2.254** 로 전달됩니다.

DNS 영역 전달

IdM의 관점에서, 전달 **DNS** 영역에는 권한 있는 데이터가 포함되어 있지 않습니다. 실제로 **forward "zone"**에는 일반적으로 두 가지 정보만 포함됩니다.

- 도메인 이름
- 도메인과 연결된 **DNS** 서버의 **IP** 주소

정의된 도메인에 속하는 이름에 대한 모든 쿼리는 지정된 **IP** 주소로 전달됩니다. 이 동작은 표준 **BIND** 구성의 **type forward** 설정과 동일합니다. `ipa dnsforwardzone-*` 명령을 사용하여 전달 영역을 관리할 수 있습니다.

정방향 DNS 영역은 IdM-AD(Active Directory) 신뢰 컨텍스트에서 특히 유용합니다. IdM DNS 서버에 `idm.example.com` 영역에 대한 권한이 있고 AD DNS 서버가 `ad.example.com` 영역에 대해 권한이 있는 경우 `ad.example.com` 은 `idm.example.com` 기본 영역의 DNS 전달 영역입니다. 즉, 쿼리가 `somehost.ad.example.com`의 IP 주소에 대한 IdM 클라이언트에서 제공되면 쿼리가 `ad.example.com` IdM DNS 전달 영역에 지정된 AD 도메인 컨트롤러로 전달됩니다.

78.2. 기본 IDM DNS 영역의 구성 특성

IdM(Identity Management)은 새로 고침 기간, 전송 설정 또는 캐시 설정과 같은 특정 기본 구성으로 새 영역을 생성합니다. [IdM DNS 영역 속성](#)에서는 다음 옵션 중 하나를 사용하여 수정할 수 있는 기본 영역 구성의 특성을 찾을 수 있습니다.

- CLI(명령줄 인터페이스)의 `dnszone-mod` 명령. 자세한 내용은 [IdM CLI에서 기본 DNS 영역 구성 편집](#)을 참조하십시오.
- IdM 웹 UI. 자세한 내용은 [IdM 웹 UI에서 기본 DNS 영역 구성 편집](#)을 참조하십시오.
- `ipadnszone` 모듈을 사용하는 Ansible 플레이북. 자세한 내용은 [IdM에서 DNS 영역 관리](#)를 참조하십시오.

영역에 대한 실제 정보를 설정하는 것과 함께 설정은 DNS 서버가 권한 시작 (SOA) 레코드 항목을 처리하는 방법과 DNS 이름 서버에서 레코드를 업데이트하는 방법을 정의합니다.

표 78.1. IdM DNS 영역 속성

속성	ansible-freeipa 변수	설명
인가된 이름 서버	<code>name_server</code>	SOA MNAME이라고도 하는 기본 DNS 이름 서버의 도메인 이름을 설정합니다. 기본적으로 각 IdM 서버는 SOA MNAME 필드에 자체적으로 알립니다. 결과적으로 <code>--name-server</code> 를 사용하여 LDAP에 저장된 값은 무시됩니다.
관리자 이메일 주소	<code>admin_email</code>	영역 관리자에게 사용할 이메일 주소를 설정합니다. 기본값은 호스트의 root 계정으로 설정됩니다.
SOA 직렬	시리얼	SOA 레코드에 일련 번호를 설정합니다. IdM은 버전 번호를 자동으로 설정하고 사용자가 수정하지 않아야 합니다.

속성	ansible-freeipa 변수	설명
SOA 새로 고침	새로 고침	보조 DNS 서버가 기본 DNS 서버에서 업데이트를 요청하기 전에 대기할 간격(초)을 설정합니다.
SOA 재시도	재시도	실패한 새로 고침 작업을 재시도하기 전에 대기할 시간(초)을 설정합니다.
SOA 만료	expire	보조 DNS 서버가 작업 시도를 종료하기 전에 새로 고침 업데이트를 수행하는 시간을 초 단위로 설정합니다.
SOA 최소	최소	RFC 2308 에 따라 음수 캐싱의 TTL(Time to Live) 값을 초 단위로 설정합니다.
SOA 수명	ttl	zone apex에서 레코드에 TTL(초)을 설정합니다. 예를 들어 zone example.com 에서는 example.com 이름의 모든 레코드(A , NS 또는 SOA)가 구성되지만 test.example.com 과 같은 다른 도메인 이름은 영향을 받지 않습니다.
기본 수명	default_ttl	개별 TTL 값이 없는 영역의 모든 값에 대해 음수 캐싱을 위해 기본 TTL(Time to Live) 값을 초 단위로 설정합니다. 변경 사항을 적용한 후 모든 IdM DNS 서버에서 named-pkcs11 서비스를 다시 시작해야 합니다.
BIND 업데이트 정책	update_policy	DNS 영역에서 허용된 권한을 클라이언트에 설정합니다.
동적 업데이트	dynamic_update=TRUE FALSE	클라이언트의 DNS 레코드에 대한 동적 업데이트를 활성화합니다. 이 값을 false 로 설정하면 IdM 클라이언트 시스템에서 IP 주소를 추가하거나 업데이트할 수 없습니다.
전송 허용	allow_transfer=string	세미콜론(;)으로 구분된 지정된 영역을 전송할 수 있는 IP 주소 또는 네트워크 이름 목록을 제공합니다. 영역 전송은 기본적으로 비활성화되어 있습니다. 기본 allow_transfer 값은 none 입니다.
쿼리 허용	allow_query	세미콜론(;)으로 구분된 DNS 쿼리를 실행할 수 있는 IP 주소 또는 네트워크 이름 목록을 제공합니다.
PTR 동기화 허용	allow_sync_ptr=1 0	영역에 대한 A 또는 AAAA 레코드(전달 레코드)가 PTR(역방향) 레코드와 자동으로 동기화되는지 여부를 설정합니다.
영역 전달기	forwarder=IP_address	DNS 영역에 구체적으로 구성된 전달자를 지정합니다. IdM 도메인에서 사용되는 모든 글로벌 전달자와는 별개입니다. 여러 전달자를 지정하려면 옵션을 여러 번 사용합니다.

속성	ansible-freeipa 변수	설명
전달 정책	forward_policy= none only first	전달 정책을 지정합니다. 지원되는 정책에 대한 자세한 내용은 IdM의 DNS 전달 정책 을 참조하십시오.

추가 리소스

- **/usr/share/doc/ansible-freeipa/ 디렉토리에서 사용할 수 있는 README-dnszone.md Markdown 파일에서 ansible-freeipa ipadsnzone 모듈의 속성 정의가 더 많은 것을 확인할 수 있습니다.**

78.3. ANSIBLE을 사용하여 IDM DNS에서 기본 영역 생성

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 기본 **DNS** 영역이 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **IdM** 관리자는 **zone.idm.example.com DNS** 영역이 있는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

절차

1.

/usr/share/doc/ansible-freeipa/playbooks/dnszone 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2.

인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3.

dnszone-present.yml **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp dnszone-present.yml dnszone-present-copy.yml
```

4.

편집할 **dnszone-present-copy.yml** 파일을 엽니다.

5.

ipadnszone 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 IdM 관리자 암호로 설정합니다.
- **zone_name** 변수를 **zone.idm.example.com** 으로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure zone is present.
      ipadnszone:
        ipaadmin_password: Secret123
        zone_name: zone.idm.example.com
        state: present
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

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

추가 리소스

- DNS 영역에 대한 자세한 내용은 [지원되는 DNS 영역 유형](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa-freeipa/` 디렉터리에서 사용할 수 있는 **README-dnszone.md** Markdown 파일에서 **ansible-freeipa ipadnszone** 모듈에 대한 더 많은 샘플

Ansible 플레이북을 볼 수 있습니다. 파일에는 **ipadnszone** 변수의 정의도 포함되어 있습니다.

- **/usr/share/doc/ansible-freeipa/playbooks/dns zone** 디렉터리에서 **ipadns zone** 모듈에 대한 샘플 **Ansible** 플레이북을 볼 수 있습니다.

78.4. ANSIBLE 플레이북을 사용하여 여러 변수와 함께 IDM에 기본 DNS 영역이 있는지 확인합니다.

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 기본 **DNS** 영역이 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **IdM** 관리자는 **zone.idm.example.com** **DNS** 영역이 있는지 확인합니다. **Ansible** 플레이북은 영역의 여러 매개 변수를 구성합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

절차

1.

/usr/share/doc/ansible-freeipa/playbooks/dnszone 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2.

인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3.

dnszone-all-params.yml **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp dnszone-all-params.yml dnszone-all-params-copy.yml
```

4.

편집할 **dnszone-all-params-copy.yml** 파일을 엽니다.

5.

ipadnszone 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **zone_name** 변수를 **zone.idm.example.com** 으로 설정합니다.
- **PTR** 레코드와 **A** 및 **AAAA** 레코드의 동기화인 정방향 및 역방향 레코드의 동기화를 허용하려면 **allow_sync_ptr** 변수를 **true**로 설정합니다.
- **IdM** 클라이언트 시스템에서 **IP** 주소를 추가하거나 업데이트할 수 있도록 **dynamic_update** 변수를 **true**로 설정합니다.
- 영역에 있는 레코드의 인라인 **DNSSEC** 서명을 허용하려면 **dnssec** 변수를 **true**로 설정합니다.
- **allow_transfer** 변수를 영역의 보조 이름 서버의 **IP** 주소로 설정합니다.
- **allow_query** 변수를 쿼리를 발행할 수 있는 **IP** 주소 또는 네트워크로 설정합니다.
- **forwarders** 변수를 글로벌 전달자의 **IP** 주소로 설정합니다.
- **serial** 변수를 **SOA** 레코드 일련 번호로 설정합니다.
- 영역의 **DNS** 레코드에 대한 **refresh, retry, expire, minimum, ttl** 및 **default_ttl** 값을 정의합니다.
- **nsec3param_rec** 변수를 사용하여 영역에 대한 **NSEC3 PARAM** 레코드를 정의합니다.
- 기존 영역과 겹치는 경우에도 **DNS** 생성을 강제 하려면 **skip_overlap_check** 변수를 **true**로 설정합니다.

이름 서버를 확인할 수 없는 경우에도 **DNS** 영역을 강제로 생성하려면 **skip_nameserver_check** 를 **true**로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure zone is present.
      ipadnszone:
        ipadmin_password: Secret123
        zone_name: zone.idm.example.com
        allow_sync_ptr: true
        dynamic_update: true
        dnssec: true
        allow_transfer:
          - 1.1.1.1
          - 2.2.2.2
        allow_query:
          - 1.1.1.1
          - 2.2.2.2
        forwarders:
          - ip_address: 8.8.8.8
          - ip_address: 8.8.4.4
          port: 52
        serial: 1234
        refresh: 3600
        retry: 900
        expire: 1209600
        minimum: 3600
        ttl: 60
        default_ttl: 90
        name_server: server.idm.example.com.
        admin_email: admin.admin@idm.example.com
        nsec3param_rec: "1 7 100 0123456789abcdef"
        skip_overlap_check: true
        skip_nameserver_check: true
        state: present
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file dnszone-all-params-copy.yml
```

추가 리소스

- **DNS 영역에 대한 자세한 내용은 지원되는 DNS 영역 유형을 참조하십시오.**
- **IdM에서 구성할 수 있는 DNS 영역 속성에 대한 자세한 내용은 기본 IdM DNS 영역의 구성 속성을 참조하십시오.**
- **`/usr/share/doc/ansible-freeipa-freeipa/` 디렉터리에서 사용할 수 있는 **README-dnszone.md** Markdown 파일에서 **ansible-freeipa ipadnszone** 모듈에 대한 더 많은 샘플 Ansible 플레이북을 볼 수 있습니다. 파일에는 **ipadnszone** 변수의 정의도 포함되어 있습니다.**
- **`/usr/share/doc/ansible-freeipa/playbooks/dns zone` 디렉터리에서 **ipadns zone** 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.**

78.5. IP 주소가 제공될 때 역방향 DNS 조회에 영역이 있는지 확인하는 데 ANSIBLE 플레이북을 사용합니다.

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 역방향 DNS 영역이 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 IdM 관리자는 IdM 호스트의 IP 주소 및 접두사 길이를 사용하여 역방향 DNS 조회 영역을 확인합니다.

`name_from_ip` 변수를 사용하여 DNS 서버의 IP 주소 접두사 길이를 제공하면 영역 이름을 제어할 수 있습니다. 접두사 길이를 설명하지 않으면 시스템은 DNS 서버에 영역에 대해 쿼리하고 **`name_from_ip`** 값 **192.168.1.2** 를 기준으로 쿼리에서 다음 DNS 영역을 반환할 수 있습니다.

- **1.168.192.in-addr.arpa.**
- **168.192.in-addr.arpa.**
- **192.in-addr.arpa.**

쿼리에서 반환한 영역이 예상과 다를 수 있으므로 **`name_from_ip`** 는 실수로 영역이 제거되지 않도록 **`state`** 옵션과 함께만 사용할 수 있습니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/dnszone** 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. 인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **dnszone-reverse-from-ip.yml** **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp dnszone-reverse-from-ip.yml dnszone-reverse-from-ip-copy.yml
```

4. 편집을 위해 **dnszone-reverse-from-ip-copy.yml** 파일을 엽니다.

5. **ipadnszone** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
 - **name_from_ip** 변수를 **IdM** 이름 서버의 **IP**로 설정하고 해당 접두사 길이를 제공합니다.
- 이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true
```

tasks:

- name: Ensure zone for reverse DNS lookup is present.

ipadnszone:

ipaadmin_password: Secret123

name_from_ip: 192.168.1.2/24

state: present

register: result

- name: Display inferred zone name.

debug:

msg: "Zone name: {{ result.dnszone.name }}"

플레이북은 192.168.1.2 IP 주소 및 접두사 길이 24에서 역방향 DNS 조회 영역을 생성합니다. 그런 다음 플레이북에 결과 영역 이름이 표시됩니다.

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file dnszone-reverse-from-ip-copy.yml
```

추가 리소스

- DNS 영역에 대한 자세한 내용은 [지원되는 DNS 영역 유형](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa-freeipa/` 디렉터리에서 사용할 수 있는 **README-dnszone.md** Markdown 파일에서 **ansible-freeipa ipadnszone** 모듈에 대한 더 많은 샘플 Ansible 플레이북을 볼 수 있습니다. 파일에는 **ipadnszone** 변수의 정의도 포함되어 있습니다.
- `/usr/share/doc/ansible-freeipa/playbooks/dns zone` 디렉터리에서 **ipadns zone** 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.

79장. IDM에서 DNS 위치 관리

IdM(Identity Management) 관리자는 **IdM 웹 UI** 및 **IdM CLI(명령줄 인터페이스)**를 사용하여 **IdM(Identity Management) DNS** 위치를 관리할 수 있습니다. 이 장에서는 다음 주제 및 절차를 설명합니다.

- **DNS 기반 서비스 검색**
- **DNS 위치에 대한 배포 고려 사항**
- **DNS Time to Live (TTL)**
- **IdM 웹 UI를 사용하여 DNS 위치 생성**
- **IdM CLI를 사용하여 DNS 위치 생성**
- **IdM 웹 UI를 사용하여 DNS 위치에 IdM 서버 할당**
- **IdM 웹 UI를 사용하여 DNS 위치에 IdM 서버 할당**
- **동일한 위치에서 IdM 서버를 사용하도록 IdM 클라이언트 구성**

79.1. DNS 기반 서비스 검색

DNS 기반 서비스 검색은 클라이언트에서 **DNS** 프로토콜을 사용하여 **LDAP** 또는 **Kerberos**와 같은 특정 서비스를 제공하는 네트워크에서 서버를 찾는 프로세스입니다. 일반적인 작업 유형 중 하나는 클라이언트가 가장 가까운 네트워크 인프라 내에서 인증 서버를 찾을 수 있도록 허용하는 것입니다. 이는 처리량이 증가하고 네트워크 대기 시간이 단축되어 전체 비용이 절감되기 때문입니다.

서비스 검색의 주요 장점은 다음과 같습니다.

- 가까운 서버의 이름으로 클라이언트를 명시적으로 구성할 필요가 없습니다.

•

DNS 서버는 정책의 중앙 공급업체로 사용됩니다. 동일한 **DNS** 서버를 사용하는 클라이언트는 서비스 프로바이더 및 해당 기본 순서에 대해 동일한 정책에 액세스할 수 있습니다.

IdM(Identity Management) 도메인에서는 **LDAP, Kerberos** 및 기타 서비스에 대한 **SRV 레코드(DNS 서비스 레코드)**가 있습니다. 예를 들어 다음 명령은 **IdM DNS** 도메인에서 **TCP 기반 Kerberos** 서비스를 제공하는 호스트를 **DNS** 서버에 쿼리합니다.

예 79.1. DNS 위치 독립 결과

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
0 100 88 idmserver-01.idm.example.com.
0 100 88 idmserver-02.idm.example.com.
```

출력에는 다음 정보가 포함됩니다.

•

0 (우선 순위): 대상 호스트의 우선 순위입니다. 더 낮은 값이 우선합니다.

•

100 (가벼움). 우선순위가 동일한 항목의 상대 가중치를 지정합니다. 자세한 내용은 [RFC 2782, 섹션 3](#)을 참조하십시오.

•

88 (포트 번호): 서비스의 포트 번호입니다.

•

서비스를 제공하는 호스트의 정식 이름입니다.

이 예에서 반환된 두 호스트 이름은 우선 순위와 가중치가 동일합니다. 이 경우 클라이언트는 결과 목록의 임의 항목을 사용합니다.

대신 클라이언트가 **DNS** 위치에 구성된 **DNS** 서버를 쿼리하도록 구성된 경우 출력은 다릅니다. 위치에 할당된 **IdM** 서버의 경우 맞춤형 값이 반환됩니다. 아래 예제에서 클라이언트는 위치 **germany** 에서 **DNS** 서버를 쿼리하도록 구성됩니다.

예 79.2. DNS 위치 기반 결과

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
_kerberos._tcp.germany.locations.idm.example.com.
0 100 88 idmserver-01.idm.example.com.
```

50 100 88 idmserver-02.idm.example.com.

IdM DNS 서버는 로컬 서버를 선호하는 DNS 위치별 SRV 레코드를 가리키는 DNS 별칭(CNAME)을 자동으로 반환합니다. 이 CNAME 레코드는 출력의 첫 번째 줄에 표시됩니다. 이 예제에서 호스트 idmserver-01.idm.example.com 은 우선 순위 값이 가장 낮으므로 선호됩니다. idmserver-02.idm.example.com 은 우선 순위가 높으므로 기본 호스트를 사용할 수 없는 경우에 대해서만 백업으로 사용됩니다.

79.2. DNS 위치에 대한 배포 고려 사항

IdM(Identity Management)은 통합 DNS를 사용할 때 SRV(위치별 서비스) 레코드를 생성할 수 있습니다. 각 IdM DNS 서버는 위치별 SRV 레코드를 생성하므로 각 DNS 위치에 하나 이상의 IdM DNS 서버를 설치해야 합니다.

DNS 위치에 대한 클라이언트의 선호도는 클라이언트가 수신한 DNS 레코드에서만 정의합니다. 이러한 이유로 IdM DNS 서버를 비IdM DNS 소비자 서버와 결합하고 DNS 서비스 검색을 수행하는 클라이언트가 IdM DNS 서버의 위치별 레코드를 분석하는 경우 재귀할 수 있습니다.

혼합 IdM 및 비IdM DNS 서비스가 있는 대부분의 배포에서 DNS 재귀는 왕복 시간 지표를 사용하여 가장 가까운 IdM DNS 서버를 자동으로 선택합니다. 일반적으로, IdM이 아닌 DNS 서버를 사용하는 클라이언트가 가장 가까운 DNS 위치에 대한 레코드를 확보하므로 최적의 IdM 서버 세트를 사용할 수 있습니다.

79.3. DNS TIME TO LIVE (TTL)

클라이언트는 영역 구성에 설정된 기간 동안 DNS 리소스 레코드를 캐시할 수 있습니다. 이 캐싱으로 인해 클라이언트는 TTL(Time To Live) 값이 만료될 때까지 클라이언트에서 변경 사항을 받지 못할 수 있습니다. IdM(Identity Management)의 기본 TTL 값은 1일입니다.

클라이언트 컴퓨터가 사이트 간에 로밍되는 경우 IdM DNS 영역의 TTL 값을 조정해야 합니다. 클라이언트가 사이트 간에 로밍해야 하는 시간보다 낮은 값으로 값을 설정합니다. 이렇게 하면 클라이언트에 캐시된 DNS 항목이 다른 사이트에 다시 연결되기 전에 만료되도록 하므로 DNS 서버를 쿼리하여 위치별 SRV 레코드를 새로 고칩니다.

추가 리소스



DNS 영역의 기본 TTL을 수정하는 방법에 대한 자세한 내용은 [기본 IdM DNS 영역의 구성 속성](#)을 참조하십시오.

79.4. IDM 웹 UI를 사용하여 DNS 위치 생성

DNS 위치를 사용하여 IdM(Identity Management) 클라이언트와 서버 간의 통신 속도를 높일 수 있습니다. 이 섹션에서는 **IdM 웹 UI를 사용하여 DNS 위치를 생성하는 방법을 설명합니다.**

사전 요구 사항

- **IdM 배포에는 통합된 DNS가 있습니다.**
- **IdM에 DNS 위치를 생성할 수 있는 권한이 있습니다.** 예를 들어 **IdM 관리자**로 로그인되어 있습니다.

절차

1. **IPA Server(IPA 서버)** 탭을 엽니다.
2. **Topology(토폴로지)** 하위 탭을 선택합니다.
3. 네비게이션 바에서 **IPA Locations (IPA 위치)**를 클릭합니다.
4. 위치 목록 상단에 있는 **Add(추가)**를 클릭합니다.
5. 위치 이름을 입력합니다.
6. **Add(추가)** 버튼을 클릭하여 위치를 저장합니다.
7. 선택 사항: 단계를 반복하여 위치를 추가합니다.

추가 리소스

- 추가한 **IdM 위치**에 대한 특정 서버를 구성하려면 **IdM 웹 UI를 사용하여 DNS 위치에 IdM 서버 할당을 참조하십시오.**
-

Ansible 플레이북을 사용하여 절차를 실행하는 방법에 대한 자세한 내용은 [Ansible을 사용하여 IdM 위치가 있는지 확인하십시오](#).

79.5. IDM CLI를 사용하여 DNS 위치 생성

DNS 위치를 사용하여 **IdM(Identity Management)** 클라이언트와 서버 간의 통신 속도를 높일 수 있습니다. 이 섹션에서는 **IdM CLI(명령줄 인터페이스)**에서 **ipa location-add** 명령을 사용하여 **DNS** 위치를 생성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 배포에는 통합된 **DNS**가 있습니다.
- **IdM**에 **DNS** 위치를 생성할 수 있는 권한이 있습니다. 예를 들어 **IdM** 관리자로 로그인되어 있습니다.

절차

1. 예를 들어 새 위치 **germany** 를 생성하려면 다음을 입력합니다.

```
$ ipa location-add germany
-----
Added IPA location "germany"
-----
Location name: germany
```

2. 선택 사항: 단계를 반복하여 위치를 추가합니다.

추가 리소스

- 추가한 **IdM** 위치에 대한 특정 서버를 구성하려면 **IdM CLI를 사용하여 DNS 위치에 IdM 서버 할당을 참조하십시오**.
- **Ansible** 플레이북을 사용하여 절차를 실행하는 방법에 대한 자세한 내용은 [Ansible을 사용하여 IdM 위치가 있는지 확인하십시오](#).

79.6. IDM 웹 UI를 사용하여 DNS 위치에 IDM 서버 할당

IdM(Identity Management) DNS 위치를 사용하여 **IdM** 클라이언트와 서버 간의 통신 속도를 높일 수 있습니다. 이 섹션에서는 **IdM** 웹 UI를 사용하여 **DNS** 위치에 **IdM** 서버를 할당하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 배포에는 통합된 **DNS**가 있습니다.
- 서버를 **DNS** 위치에 할당할 권한이 있는 사용자로 로그인했습니다(예: **IdM** 관리자).
- **DNS** 위치를 할당할 호스트에 대한 루트 액세스 권한이 있습니다.
- 서버를 할당 [할 IdM DNS 위치를 생성했습니다](#).

절차

1. **IPA Server(IPA 서버)** 탭을 엽니다.
2. **Topology(토폴로지)** 하위 탭을 선택합니다.
3. 탐색에서 **IPA Servers (IPA 서버)**를 클릭합니다.
4. **IdM** 서버 이름을 클릭합니다.
5. **DNS** 위치를 선택하고 서비스 가중치를 선택적으로 설정합니다.

그림 79.1. DNS 위치에 서버 할당

IPA Server: idmserver-01.idm.example.com

Refresh Revert Save

Server name	idmserver-01.idm.example.com.
Min domain level	0
Max domain level	1
Managed suffixes	domain ca
Location	germany
Service weight	100

6.

저장을 클릭합니다.

7.

이전 단계에서 할당한 호스트의 **CLI**(명령줄 인터페이스)에서 **DNS** 위치를 에 배치한 후 **named-pkcs11** 서비스를 다시 시작합니다.

```
[root@idmserver-01 ~]# systemctl restart named-pkcs11
```

8.

선택 사항: 단계를 반복하여 **DNS** 위치를 추가 **IdM** 서버에 할당합니다.

추가 리소스

•

계속하려면 동일한 위치에서 **IdM** 서버를 사용하도록 **IdM** 클라이언트 구성을 참조하십시오.

79.7. IDM CLI를 사용하여 DNS 위치에 IDM 서버 할당

IdM(Identity Management) **DNS** 위치를 사용하여 **IdM** 클라이언트와 서버 간의 통신 속도를 높일 수 있습니다. 이 섹션에서는 **IdM CLI**(명령줄 인터페이스)를 사용하여 **IdM** 서버를 **DNS** 위치에 할당하는 방법을 설명합니다.

사전 요구 사항

•

IdM 배포에는 통합된 **DNS**가 있습니다.

- 서버를 **DNS** 위치에 할당할 권한이 있는 사용자로 로그인했습니다(예: **IdM** 관리자).
- **DNS** 위치를 할당할 호스트에 대한 루트 액세스 권한이 있습니다.
- 서버를 할당 **할 IdM DNS 위치를 생성했습니다**.

절차

1. 선택 사항: 구성된 모든 **DNS** 위치를 나열합니다.

```
[root@server ~]# ipa location-find
-----
2 IPA locations matched
-----
Location name: australia
Location name: germany
-----
Number of entries returned: 2
-----
```

2. 서버를 **DNS** 위치에 할당합니다. 예를 들어 서버 **idmserver-01.idm.example.com**에 위치 **germany**를 할당하려면 다음을 실행합니다.

```
# ipa server-mod idmserver-01.idm.example.com --location=germany
ipa: WARNING: Service named-pkcs11.service requires restart on IPA server
idmserver-01.idm.example.com to apply configuration changes.
-----
Modified IPA server "idmserver-01.idm.example.com"
-----
Servername: idmserver-01.idm.example.com
Min domain level: 0
Max domain level: 1
Location: germany
Enabled server roles: DNS server, NTP server
```

3. 이전 단계에서 할당한 호스트에서 **named-pkcs11** 서비스를 다시 시작하십시오.

```
# systemctl restart named-pkcs11
```

4. 선택 사항: 단계를 반복하여 **DNS** 위치를 추가 **IdM** 서버에 할당합니다.

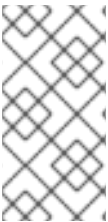
추가 리소스

- 계속하려면 동일한 위치에서 **IdM** 서버를 사용하도록 **IdM** 클라이언트 구성을 참조하십시오.

79.8. 동일한 위치에서 **IDM** 서버를 사용하도록 **IDM** 클라이언트 구성

IdM(Identity Management) 서버는 **IdM** 웹 UI를 사용하여 **DNS** 위치에 **IdM** 서버 할당에 설명된 대로 **DNS** 위치에 할당됩니다. 이제 **IdM** 서버와 동일한 위치에 있는 **DNS** 서버를 사용하도록 클라이언트를 구성할 수 있습니다.

- **DHCP** 서버가 클라이언트에 **DNS** 서버 IP 주소를 할당하는 경우 **DHCP** 서비스를 구성합니다. **DHCP** 서비스에서 **DNS** 서버를 할당하는 방법에 대한 자세한 내용은 **DHCP** 서비스 설명서를 참조하십시오.
- 클라이언트가 **DHCP** 서버에서 **DNS** 서버 IP 주소를 받지 못하는 경우 클라이언트의 네트워크 구성에서 IP를 수동으로 설정합니다. **Red Hat Enterprise Linux**에서 네트워크 구성에 대한 자세한 내용은 **Red Hat Enterprise Linux** 네트워킹 가이드의 **네트워크 연결 설정 구성** 섹션을 참조하십시오.



참고

다른 위치에 할당된 **DNS** 서버를 사용하도록 클라이언트를 구성하는 경우 클라이언트는 두 위치의 **IdM** 서버에 연결합니다.

예 79.3. 클라이언트 위치에 따라 다른 이름 서버 항목

다음 예제에서는 다양한 위치에 있는 클라이언트의 **/etc/resolv.conf** 파일의 다양한 이름 서버 항목을 보여줍니다.

플래티넘 고객:

```
nameserver 10.10.0.1
nameserver 10.10.0.2
```

플래티넘 고객:

```
nameserver 10.50.0.1
nameserver 10.50.0.3
```

Oslo의 클라이언트:

nameserver 10.30.0.1

덴마크의 고객:

nameserver 10.30.0.1

각 **DNS** 서버가 **IdM**의 위치에 할당되면 클라이언트는 해당 위치에서 **IdM** 서버를 사용합니다.

79.9. 추가 리소스

- **Red Hat Ansible Engine**을 사용하여 **IdM DNS** 위치를 관리하는 방법에 대한 자세한 내용은 [Ansible을 사용하여 IdM의 DNS 위치 관리 방법을 참조하십시오.](#)

80장. ANSIBLE을 사용하여 IDM에서 DNS 위치 관리

IdM(Identity Management) 관리자는 **ansible-freeipa** 패키지에서 사용할 수 있는 위치 모듈을 사용하여 **IdM DNS** 위치를 관리할 수 있습니다. 이 장에서는 다음 주제 및 절차를 설명합니다.

- **DNS 기반 서비스 검색**
- **DNS 위치에 대한 배포 고려 사항**
- **DNS Time to Live (TTL)**
- **Ansible을 사용하여 IdM 위치가 있는지 확인**
- **Ansible을 사용하여 IdM 위치가 없는지 확인**

80.1. DNS 기반 서비스 검색

DNS 기반 서비스 검색은 클라이언트에서 **DNS** 프로토콜을 사용하여 **LDAP** 또는 **Kerberos** 와 같은 특정 서비스를 제공하는 네트워크에서 서버를 찾는 프로세스입니다. 일반적인 작업 유형 중 하나는 클라이언트가 가장 가까운 네트워크 인프라 내에서 인증 서버를 찾을 수 있도록 허용하는 것입니다. 이는 처리량이 증가하고 네트워크 대기 시간이 단축되어 전체 비용이 절감되기 때문입니다.

서비스 검색의 주요 장점은 다음과 같습니다.

- 가까운 서버의 이름으로 클라이언트를 명시적으로 구성할 필요가 없습니다.
- **DNS** 서버는 정책의 중앙 공급업체로 사용됩니다. 동일한 **DNS** 서버를 사용하는 클라이언트는 서비스 프로바이더 및 해당 기본 순서에 대해 동일한 정책에 액세스할 수 있습니다.

IdM(Identity Management) 도메인에서는 **LDAP**, **Kerberos** 및 기타 서비스에 대한 **SRV 레코드(DNS 서비스 레코드)**가 있습니다. 예를 들어 다음 명령은 **IdM DNS** 도메인에서 **TCP** 기반 **Kerberos** 서비스를 제공하는 호스트를 **DNS** 서버에 쿼리합니다.

예 80.1. DNS 위치 독립 결과

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
0 100 88 idmserver-01.idm.example.com.
0 100 88 idmserver-02.idm.example.com.
```

출력에는 다음 정보가 포함됩니다.

- **0 (우선 순위):** 대상 호스트의 우선 순위입니다. 더 낮은 값이 우선합니다.
- **100 (가벼움):** 우선순위가 동일한 항목의 상대 가중치를 지정합니다. 자세한 내용은 [RFC 2782](#), [섹션 3](#)을 참조하십시오.
- **88 (포트 번호):** 서비스의 포트 번호입니다.
- **서비스를 제공하는 호스트의 정식 이름입니다.**

이 예에서 반환된 두 호스트 이름은 우선 순위와 가중치가 동일합니다. 이 경우 클라이언트는 결과 목록의 임의 항목을 사용합니다.

대신 클라이언트가 **DNS** 위치에 구성된 **DNS** 서버를 쿼리하도록 구성된 경우 출력은 다릅니다. 위치에 할당된 **IdM** 서버의 경우 맞춤형 값이 반환됩니다. 아래 예제에서 클라이언트는 위치 **germany** 에서 **DNS** 서버를 쿼리하도록 구성됩니다.

예 80.2. DNS 위치 기반 결과

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
_kerberos._tcp.germany._locations.idm.example.com.
0 100 88 idmserver-01.idm.example.com.
50 100 88 idmserver-02.idm.example.com.
```

IdM DNS 서버는 로컬 서버를 선호하는 **DNS** 위치별 **SRV** 레코드를 가리키는 **DNS** 별칭(**CNAME**)을 자동으로 반환합니다. 이 **CNAME** 레코드는 출력의 첫 번째 줄에 표시됩니다. 이 예제에서 호스트 **idmserver-01.idm.example.com** 은 우선 순위 값이 가장 낮으므로 선호됩니다. **idmserver-02.idm.example.com** 은 우선 순위가 높으므로 기본 호스트를 사용할 수 없는 경우에 대해서만 백업으로 사용됩니다.

80.2. DNS 위치에 대한 배포 고려 사항

IdM(Identity Management)은 통합 **DNS**를 사용할 때 **SRV(위치별 서비스)** 레코드를 생성할 수 있습니다. 각 **IdM DNS** 서버는 위치별 **SRV** 레코드를 생성하므로 각 **DNS** 위치에 하나 이상의 **IdM DNS** 서버를 설치해야 합니다.

DNS 위치에 대한 클라이언트의 선호도는 클라이언트가 수신한 **DNS** 레코드에서만 정의합니다. 이러한 이유로 **IdM DNS** 서버를 비**IdM DNS** 소비자 서버와 결합하고 **DNS** 서비스 검색을 수행하는 클라이언트가 **IdM DNS** 서버의 위치별 레코드를 분석하는 경우 재귀할 수 있습니다.

혼합 **IdM** 및 비**IdM DNS** 서비스가 있는 대부분의 배포에서 **DNS** 재귀는 왕복 시간 지표를 사용하여 가장 가까운 **IdM DNS** 서버를 자동으로 선택합니다. 일반적으로, **IdM**이 아닌 **DNS** 서버를 사용하는 클라이언트가 가장 가까운 **DNS** 위치에 대한 레코드를 확보하므로 최적의 **IdM** 서버 세트를 사용할 수 있습니다.

80.3. DNS TIME TO LIVE (TTL)

클라이언트는 영역 구성에 설정된 기간 동안 **DNS** 리소스 레코드를 캐시할 수 있습니다. 이 캐싱으로 인해 클라이언트는 **TTL(Time To Live)** 값이 만료될 때까지 클라이언트에서 변경 사항을 받지 못할 수 있습니다. **IdM(Identity Management)**의 기본 **TTL** 값은 1일입니다.

클라이언트 컴퓨터가 사이트 간에 로밍되는 경우 **IdM DNS** 영역의 **TTL** 값을 조정해야 합니다. 클라이언트가 사이트 간에 로밍해야 하는 시간보다 낮은 값으로 값을 설정합니다. 이렇게 하면 클라이언트에 캐시된 **DNS** 항목이 다른 사이트에 다시 연결되기 전에 만료되도록 하므로 **DNS** 서버를 쿼리하여 위치별 **SRV** 레코드를 새로 고칩니다.

추가 리소스

-

DNS 영역의 기본 **TTL**을 수정하는 방법에 대한 자세한 내용은 [기본 IdM DNS 영역의 구성 속성](#)을 참조하십시오.

80.4. ANSIBLE을 사용하여 IDM 위치가 있는지 확인

IdM(Identity Management)의 시스템 관리자는 클라이언트가 가장 가까운 네트워크 인프라 내에서 인증 서버를 찾을 수 있도록 **IdM DNS** 위치를 구성할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **DNS** 위치가 **IdM**에 있는지 확인하는 방법을 설명합니다. 이 예제에서는 **IdM**에 **germany DNS** 위치가 있는지 확인하는 방법을 설명합니다. 결과적으로 로컬

IdM 클라이언트가 이를 사용하여 서버 응답 시간을 줄일 수 있도록 특정 IdM 서버를 이 위치에 할당할 수 있습니다.

사전 요구 사항

- **IdM 관리자 암호를 알고 있습니다.**
- **Ansible 제어 노드에 `ansible-freeipa` 패키지를 설치했습니다.**
- **이 예제에서는 `~/MyPlaybooks/` 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 **만들고 구성** 했다고 가정합니다.**
- **DNS 위치에 대한 배포 고려 사항을 이해합니다.**

절차

1. **`~/MyPlaybooks/` 디렉터리로 이동합니다.**

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

2. **`/usr/share/doc/ansible-freeipa/playbooks/location/` 디렉터리에 있는 `location-present.yml` 파일의 복사본을 만듭니다.**

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

3. **편집할 `location-present-copy.yml` Ansible 플레이북 파일을 엽니다.**
4. **`ipalocation` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.**

- **사용 사례에 대응하도록 작업 이름을 조정합니다.**
- **`ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.**

- **name** 변수를 위치 이름으로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: location present example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure that the "germany" location is present
    ipalocation:
      ipaadmin_password: Secret123
      name: germany
```

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

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

추가 리소스

- **IdM**에 현재 존재하는 **IdM** 위치에 대한 특정 서버를 구성하려면 **IdM 웹 UI**를 사용하여 **IdM 서버 할당** 또는 **IdM CLI**를 사용하여 **DNS** 위치에 **IdM 서버 할당**을 참조하십시오.

80.5. ANSIBLE을 사용하여 IDM 위치가 없는지 확인

IdM(Identity Management)의 시스템 관리자는 클라이언트가 가장 가까운 네트워크 인프라 내에서 인증 서버를 찾을 수 있도록 **IdM DNS** 위치를 구성할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM**에 **DNS** 위치가 없는지 확인하는 방법을 설명합니다. 이 예제에서는 **IdM**에 **germany DNS** 위치가 없는지 확인하는 방법을 설명합니다. 따라서 특정 **IdM** 서버를 이 위치에 할당할 수 없으며 로컬 **IdM** 클라이언트는 사용할 수 없습니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.

- **IdM 서버는 *germany* DNS 위치에 할당되지 않습니다.**
- **Ansible 제어 노드에 *ansible-freeipa* 패키지를 설치했습니다.**
- 이 예제에서는 **~/MyPlaybooks/** 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 **만들고 구성** 했다고 가정합니다.

절차

1. **~/MyPlaybooks/ 디렉터리로 이동합니다.**

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

2. **/usr/share/doc/ansible-freeipa/playbooks/location/ 디렉터리에 있는 *location-absent.yml* 파일의 복사본을 만듭니다.**

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

3. **편집할 *location-absent-copy.yml* Ansible 플레이북 파일을 엽니다.**
4. ***ipalocation* 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.**

- **사용 사례에 대응하도록 작업 이름을 조정합니다.**
- ***ipaadmin_password* 변수를 IdM 관리자의 암호로 설정합니다.**
- ***name* 변수를 DNS 위치의 이름으로 설정합니다.**
- ***state* 변수가 *absent* 로 설정되어 있는지 확인합니다.**

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```

---
- name: location absent example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that the "germany" location is absent
      ipalocation:
        ipaadmin_password: Secret123
        name: germany
        state: absent

```

5.

파일을 저장합니다.

6.

플레이북 파일 및 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

80.6. 추가 리소스

•

`/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README-location.md** 파일에서 **ansible-freeipa ipalocation** 모듈에 대한 샘플 **Ansible** 플레이북을 확인할 수 있습니다. 파일에는 **ipalocation** 변수의 정의도 포함되어 있습니다.

•

`/usr/share/doc/ansible-freeipa/playbooks/` `location` 디렉터리에서 **ipa location** 모듈에 대한 샘플 **Ansible** 플레이북을 볼 수 있습니다.

81장. IDM에서 DNS 전달 관리

다음 절차에서는 **IdM(Identity Management)** 웹 UI, **IdM CLI** 및 **Ansible**을 사용하여 **DNS** 글로벌 전달자 및 **DNS** 전달 영역을 구성하는 방법을 설명합니다.

- **81.1절. “IdM DNS 서버의 두 가지 역할”**
- **81.2절. “IdM의 DNS 전달 정책”**
- **81.3절. “IdM 웹 UI에서 글로벌 전달자 추가”**
- **81.4절. “CLI에서 글로벌 전달자 추가”**
- **81.5절. “IdM 웹 UI에서 DNS 전달 영역 추가”**
- **81.6절. “CLI에서 DNS 전달 영역 추가”**
- **81.7절. “Ansible을 사용하여 IdM에 DNS Global Forwarder 설정”**
- **81.8절. “Ansible을 사용하여 IdM에 DNS 글로벌 전달자가 있는지 확인”**
- **81.9절. “Ansible을 사용하여 IdM에 DNS 글로벌 전달자가 없는지 확인”**
- **81.10절. “Ansible을 사용하여 DNS Global Forwarders가 IdM에서 비활성화되었는지 확인”**
- **81.11절. “Ansible을 사용하여 IdM에 DNS 전달 영역이 있는지 확인”**
- **81.12절. “Ansible을 사용하여 DNS Forward Zone에 IdM에 여러 개의 전달자가 있는지 확인”**

- **81.13절. “Ansible을 사용하여 IdM에서 DNS 전달 영역이 비활성화되어 있는지 확인”**
- **81.14절. “Ansible을 사용하여 IdM에 DNS 전달 영역이 없는지 확인”**

추가 리소스

- **DNS 전달**

81.1. IDM DNS 서버의 두 가지 역할

DNS 전달은 DNS 서비스가 DNS 쿼리에 응답하는 방법에 영향을 미칩니다. 기본적으로 IdM과 통합된 Berkeley Internet Name Domain(BIND) 서비스는 권한 있는 DNS 서버 및 재귀적 DNS 서버 역할을 합니다.

인가된 DNS 서버

DNS 클라이언트가 IdM 서버에 권한이 있는 DNS 영역에 속하는 이름을 쿼리하면 BIND에서 구성된 영역에 포함된 데이터로 응답합니다. 신뢰할 수 있는 데이터는 항상 다른 데이터보다 우선합니다.

재귀 DNS 서버

DNS 클라이언트가 IdM 서버에 권한이 없는 이름을 쿼리하면 BIND에서 다른 DNS 서버를 사용하여 쿼리를 확인합니다. 전달자가 정의되지 않은 경우 BIND는 인터넷의 루트 서버에 요청하고 재귀적 확인 알고리즘을 사용하여 DNS 쿼리에 응답합니다.

경우에 따라 BIND가 다른 DNS 서버에 직접 연락하고 인터넷에서 사용 가능한 데이터를 기반으로 재귀를 수행하는 것이 바람직하지 않습니다. 쿼리를 확인하기 위해 다른 DNS 서버인 forwarder 를 사용하도록 BIND를 구성할 수 있습니다.

전달자를 사용하도록 BIND를 구성하면 IdM 서버와 전달자 간에 쿼리 및 응답을 앞뒤로 전달하고 IdM 서버는 권한이 없는 데이터의 DNS 캐시 역할을 합니다.

81.2. IDM의 DNS 전달 정책

IdM은 첫 번째 및 표준 BIND 전달 정책과 none IdM별 전달 정책을 지원합니다.

전달 먼저 (기본값)

IdM BIND 서비스는 **DNS** 쿼리를 구성된 전달자로 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 **BIND**는 인터넷에서 서버를 사용하여 재귀적 해결로 대체합니다. **forward first** 정책은 기본 정책이며 **DNS** 트래픽을 최적화하는 데 적합합니다.

앞으로만

IdM BIND 서비스는 **DNS** 쿼리를 구성된 전달자로 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 **BIND**에서 클라이언트에 오류를 반환합니다. **DNS** 구성이 분할된 환경에 만 전달 정책이 권장됩니다.

없음 (전달 비활성화)

DNS 쿼리는 **none** 전달 정책으로 전달되지 않습니다. 전달 비활성화는 전역 전달 구성의 영역별 제정의에서만 유용합니다. 이 옵션은 **BIND** 구성에서 비어 있는 전달자 목록을 지정하는 것과 동일한 **IdM**입니다.

참고

전달을 사용하여 **IdM**의 데이터를 다른 **DNS** 서버의 데이터와 결합할 수 없습니다. **IdM DNS**에서 기본 영역의 특정 하위 영역에 대한 쿼리만 전달할 수 있습니다.

기본적으로 쿼리된 **DNS** 이름이 **IdM** 서버에 권한이 있는 영역에 속하는 경우 **BIND** 서비스는 다른 서버로 쿼리를 전달하지 않습니다. 이러한 상황에서 쿼리된 **DNS** 이름을 **IdM** 데이터베이스에서 찾을 수 없는 경우 **NXDOMAIN** 응답이 반환됩니다. 전달은 사용되지 않습니다.

예 81.1. 시나리오 예

IdM 서버에는 **test.example**에 대한 권한이 있습니다. **DNS** 영역. **BIND**는 **192.0.2.254** IP 주소를 사용하여 **DNS** 서버로 쿼리를 전달하도록 구성됩니다.

클라이언트가 존재하지 않는 **test.example**에 대한 쿼리를 보낼 때. **DNS** 이름 **BIND**는 **IdM** 서버가 **test.example**. 영역에 대해 권한이 있음을 감지하고 **192.0.2.254**. 서버로 쿼리를 전달하지 않습니다. 결과적으로 **DNS** 클라이언트는 **NXDomain** 오류 메시지를 수신하여 쿼리된 도메인이 없음을 사용자에게 알립니다.

81.3. IDМ 웹 UI에서 글로벌 전달자 추가

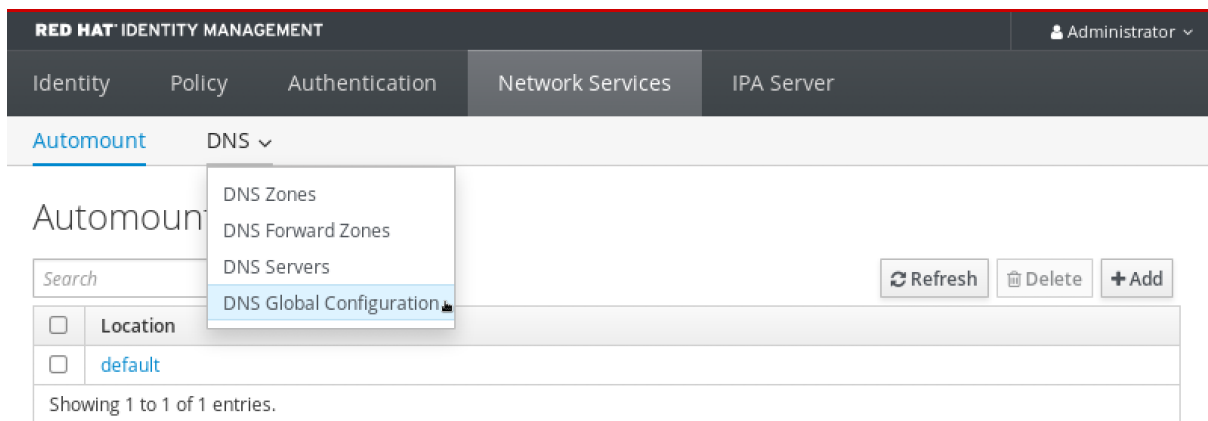
이 섹션에서는 **IdM(Identity Management)** 웹 UI에서 글로벌 **DNS** 전달자를 추가하는 방법에 대해 설명합니다.

사전 요구 사항

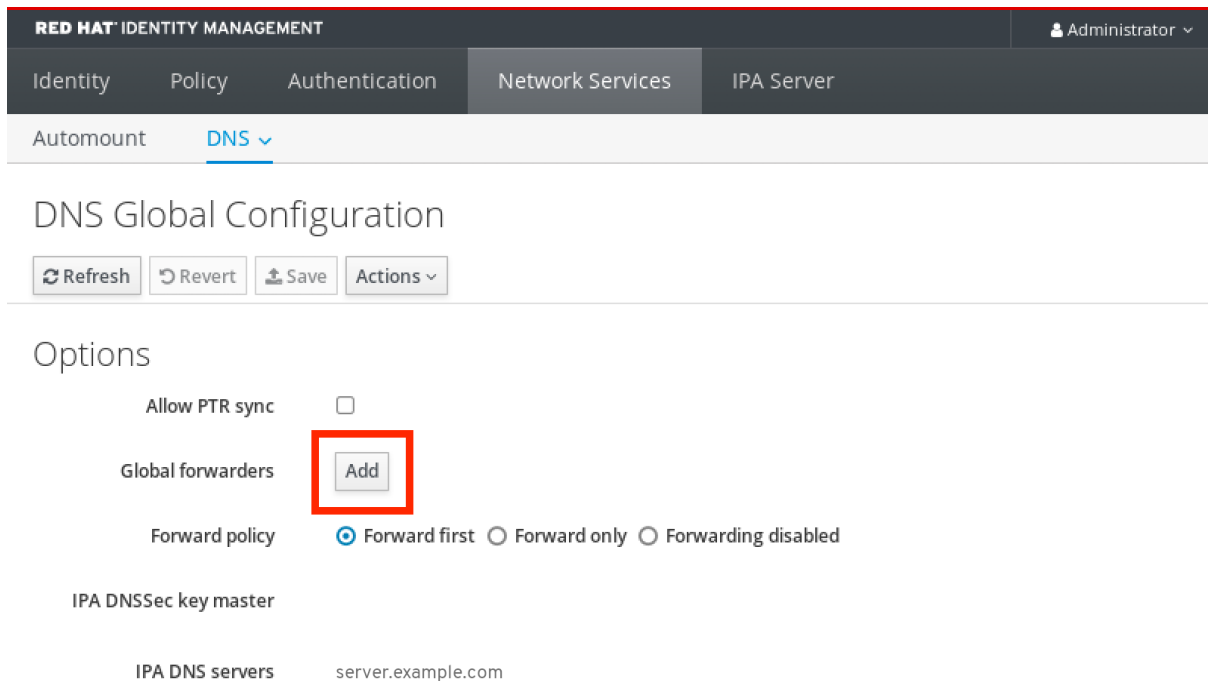
- **IdM 관리자로 IdM WebUI에 로그인했습니다.**
- **쿼리를 전달할 DNS 서버의 IP(Internet Protocol) 주소를 알고 있습니다.**

절차

1. **IdM 웹 UI에서 네트워크 서비스 → DNS 글로벌 구성 → DNS 를 선택합니다.**



2. **DNS Global Configuration(DNS 글로벌 구성) 섹션에서 Add(추가)를 클릭합니다.**



3. **전달된 DNS 쿼리를 수신할 DNS 서버의 IP 주소를 지정합니다.**

RED HAT IDENTITY MANAGEMENT Administrator ▾

Identity Policy Authentication **Network Services** IPA Server

Automount **DNS ▾**

DNS Global Configuration

Options

Allow PTR sync ☐

Global forwarders

Forward policy ☒ Forward first ☐ Forward only ☐ Forwarding disabled

IPA DNSSec key master

IPA DNS servers server.example.com

4.

Forward policy(전달 정책을 선택합니다.

RED HAT IDENTITY MANAGEMENT Administrator ▾

Identity Policy Authentication **Network Services** IPA Server

Automount **DNS ▾**

DNS Global Configuration

Options

Allow PTR sync ☐

Global forwarders

Forward policy ☒ Forward first ☐ Forward only ☐ Forwarding disabled

IPA DNSSec key master

IPA DNS servers server.example.com

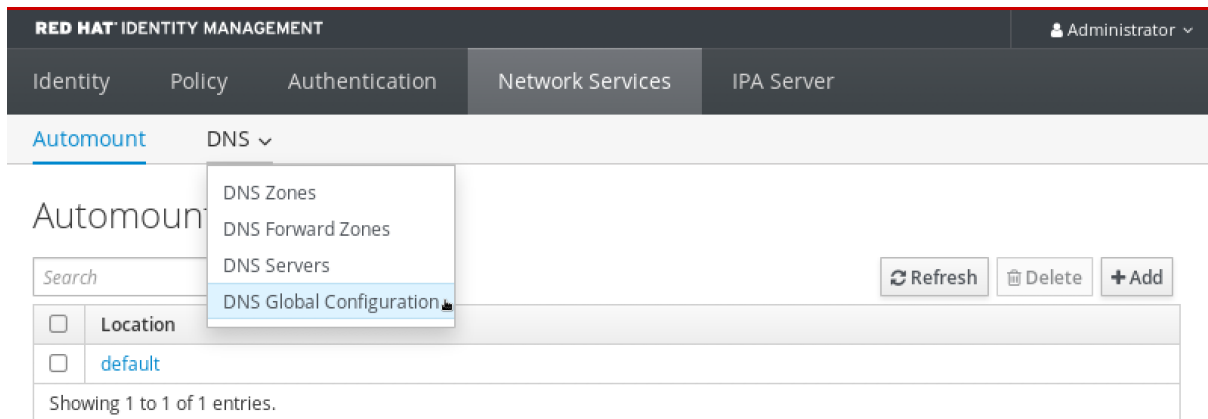
5.

창 맨 위에 있는 **Save**(저장)를 클릭합니다.

검증 단계

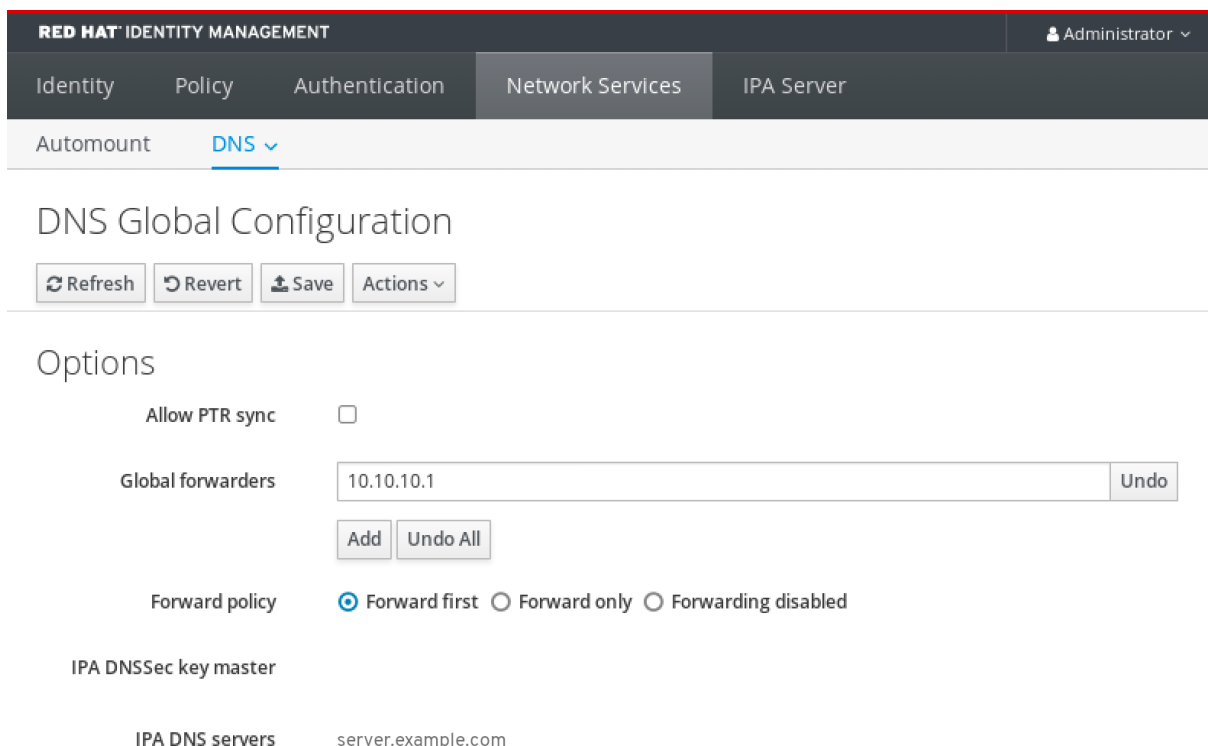
1.

네트워크 서비스 → **DNS** 글로벌 구성 → **DNS** 를 선택합니다.



2.

지정한 전달 정책이 있는 글로벌 전달자가 IdM 웹 UI에 있고 활성화되어 있는지 확인합니다.



81.4. CLI에서 글로벌 전달자 추가

이 섹션에서는 **CLI**(명령줄 인터페이스)에서 글로벌 **DNS** 전달자를 추가하는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.

- 쿼리를 전달할 **DNS 서버의 IP(Internet Protocol) 주소**를 알고 있습니다.

절차

- **ipa dnsconfig-mod** 명령을 사용하여 새 글로벌 전달자를 추가합니다. **--forwarder** 옵션을 사용하여 **DNS 전달자의 IP 주소**를 지정합니다.

```
[user@server ~]$ ipa dnsconfig-mod --forwarder=10.10.0.1
Server will check DNS forwarder(s).
This may take some time, please wait ...
Global forwarders: 10.10.0.1
IPA DNS servers: server.example.com
```

검증 단계

- **dnsconfig-show** 명령을 사용하여 글로벌 전달자를 표시합니다.

```
[user@server ~]$ ipa dnsconfig-show
Global forwarders: 10.10.0.1
IPA DNS servers: server.example.com
```

81.5. IDM 웹 UI에서 DNS 전달 영역 추가

이 섹션에서는 **IdM(Identity Management) 웹 UI**에 **DNS 전달 영역**을 추가하는 방법을 설명합니다.

중요

필요하지 않은 한 전달 영역을 사용하지 마십시오. 전달 영역은 표준 솔루션이 아니며 이를 사용하면 예기치 않고 문제가 있는 동작이 발생할 수 있습니다. 전달 영역을 사용해야 하는 경우 글로벌 전달 구성을 재정의하도록 사용을 제한합니다.

새 **DNS 영역**을 생성하는 경우 **Red Hat**은 **NS(이름 서버) 레코드**를 사용하여 표준 **DNS** 위임을 항상 사용하고 전달 영역을 방지하는 것이 좋습니다. 대부분의 경우 글로벌 전달자를 사용하는 것이 충분하며 전달 영역은 필요하지 않습니다.

사전 요구 사항

- **IdM 관리자**로 **IdM WebUI**에 로그인했습니다.

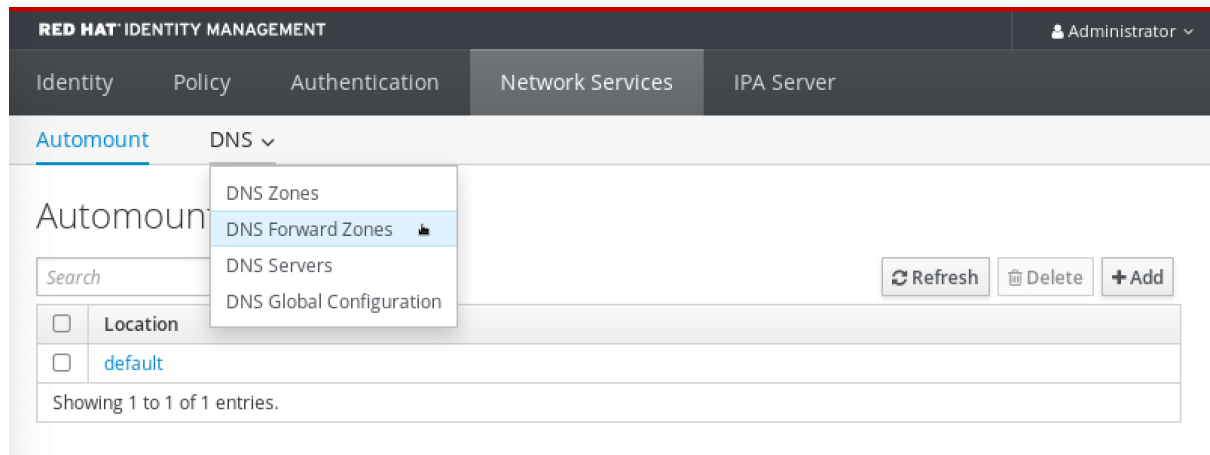
•

쿼리를 전달할 **DNS 서버의 IP(Internet Protocol) 주소**를 알고 있습니다.

절차

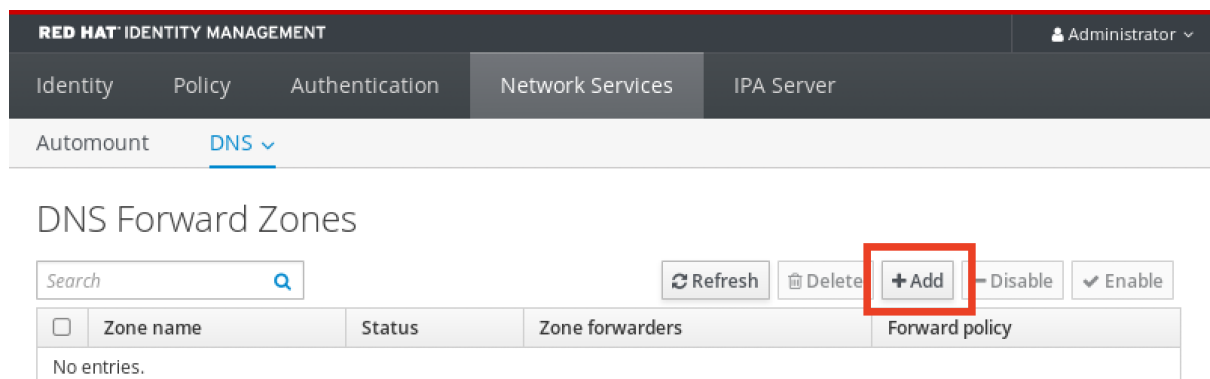
1.

IdM 웹 UI에서 네트워크 서비스 → DNS 전달 영역 → DNS 를 선택합니다.



2.

DNS Forward Zones (DNS 전달 영역) 섹션에서 Add(추가)를 클릭합니다.



3.

Add DNS forward zone (DNS 전달 영역 추가) 창에서 전달 영역 이름을 지정합니다.

RED HAT IDENTITY MANAGEMENT

Administrator

Identity

Automount

DNS Forwarding

Search

☐ Zone forwarding

No entries.

Add DNS forward zone

☒ Zone name * forward.example.com.

☐ Reverse zone IP network

Zone forwarders * Add

Forward policy ☒ Forward first ☐ Forward only ☐ Forwarding disabled

Skip overlap check ☐

* Required field

Add Add and Add Another Add and Edit Cancel

4.

Add(추가) 버튼을 클릭하고 **DNS 서버의 IP** 주소를 지정하여 전달 요청을 받습니다. 전달 영역당 여러 개의 전달자를 지정할 수 있습니다.

RED HAT IDENTITY MANAGEMENT

Administrator

Identity

Automount

DNS Forwarding

Search

☐ Zone forwarding

No entries.

Add DNS forward zone

☒ Zone name * forward.example.com.

☐ Reverse zone IP network

Zone forwarders * 10.10.0.14 Undo

Add

Forward policy ☒ Forward first ☐ Forward only ☐ Forwarding disabled

Skip overlap check ☐

* Required field

Add Add and Add Another Add and Edit Cancel

5.

Forward policy(전달 정책을 선택합니다.

RED HAT IDENTITY MANAGEMENT

Administrator

Identity

Automount

DNS Forward Zones

Search

☐ Zone name

No entries.

☒ Zone name *

forward.example.com

☐ Reverse zone

IP network

Zone forwarders *

10.10.0.14

Undo

Add

Forward policy

☒ Forward first ☐ Forward only ☐ Forwarding disabled

Skip overlap check ☐

* Required field

Add Add and Add Another Add and Edit Cancel

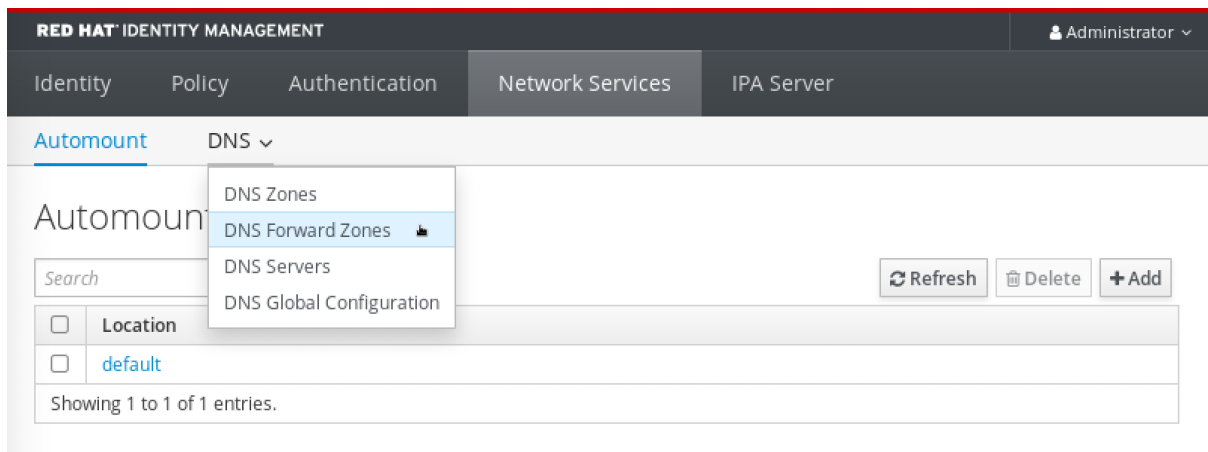
6.

창 하단에서 **Add(추가)**를 클릭하여 새 전달 영역을 추가합니다.

검증 단계

1.

IdM 웹 UI에서 네트워크 서비스 → DNS 전달 영역 → DNS 를 선택합니다.



2.

지정한 전달자 및 전달 정책을 사용하여 생성한 전달 영역이 IdM 웹 UI에 있는지 확인합니다.

RED HAT IDENTITY MANAGEMENT Administrator

Identity Policy Authentication Network Services IPA Server

Automount DNS

DNS Forward Zones

Search Refresh Delete Add Disable Enable

☐	Zone name	Status	Zone forwarders	Forward policy
☐	forward.example.com.	✓ Enabled	10.10.0.14	first

Showing 1 to 1 of 1 entries.

81.6. CLI에서 DNS 전달 영역 추가

이 섹션에서는 CLI(명령줄 인터페이스)에서 DNS 전달 영역을 추가하는 방법을 설명합니다.

중요

필요하지 않은 한 전달 영역을 사용하지 마십시오. 전달 영역은 표준 솔루션이 아니며 이를 사용하면 예기치 않고 문제가 있는 동작이 발생할 수 있습니다. 전달 영역을 사용해야 하는 경우 글로벌 전달 구성을 재정의하도록 사용을 제한합니다.

새 DNS 영역을 생성하는 경우 Red Hat은 NS(이름 서버) 레코드를 사용하여 표준 DNS 위임을 항상 사용하고 전달 영역을 방지하는 것이 좋습니다. 대부분의 경우 글로벌 전달자를 사용하는 것이 충분하며 전달 영역은 필요하지 않습니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- 쿼리를 전달할 DNS 서버의 IP(Internet Protocol) 주소를 알고 있습니다.

절차

- **dnsforwardzone-add** 명령을 사용하여 새 전달 영역을 추가합니다. **forward** 정책이 없음 인 경우 **--forwarder** 옵션을 사용하여 하나 이상의 전달자를 지정하고 **--forward-policy** 옵션을 사용하여 전달 정책을 지정합니다.

```
[user@server ~]$ ipa dnsforwardzone-add forward.example.com. --
forwarder=10.10.0.14 --forwarder=10.10.1.15 --forward-policy=first
```



```
Zone name: forward.example.com.
Zone forwarders: 10.10.0.14, 10.10.1.15
Forward policy: first
```

검증 단계

- **dnsforwardzone-show** 명령을 사용하여 방금 생성한 **DNS** 전달 영역을 표시합니다.

```
[user@server ~]$ ipa dnsforwardzone-show forward.example.com.
```

```
Zone name: forward.example.com.
Zone forwarders: 10.10.0.14, 10.10.1.15
Forward policy: first
```

81.7. ANSIBLE을 사용하여 IDM에 DNS GLOBAL FORWARDER 설정

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에 **DNS** 글로벌 전달자를 설정하는 방법에 대해 설명합니다.

아래 예제 절차에서 **IdM** 관리자는 **IP(인터넷 프로토콜) v4** 주소 **8.8.6.6** 및 **IPv6** 주소 **2001:4860:4860:4860::8800**의 **DNS** 글로벌 전달자를 **DNS** 서버로 생성합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

- 3. **set-configuration.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.**

```
$ cp set-configuration.yml establish-global-forwarder.yml
```

- 4. 편집을 위해 **establish-global-forwarder.yml** 파일을 엽니다.

- 5. 다음 변수를 설정하여 파일을 조정합니다.

- a. 플레이북의 **name** 변수를 플레이북으로 변경하여 **IdM DNS**에서 글로벌 전달자를 설정합니다.

- b. **tasks** 섹션에서 작업 이름을 **Create a DNS global forwarder(DNS 글로벌 전달자를 8.8.6.6 및 2001:4860:4860::8800)**로 변경합니다.

- c. **ipadnsconfig** 부분의 **forwarders** 섹션에서 다음을 수행합니다.

- i. 첫 번째 **ip_address** 값을 글로벌 전달자의 **IPv4** 주소로 변경합니다. **8.8.6.6**.

- ii. 두 번째 **ip_address** 값을 글로벌 전달자의 **IPv6** 주소로 변경합니다. **2001:4860:4860::8800**.

- iii. **port** 값이 **53**으로 설정되어 있는지 확인합니다.

- d. **forward_policy**를 먼저 **로** 변경합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to establish a global forwarder in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
```

```
- name: Create a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800
  ipadnsconfig:
    forwarders:
      - ip_address: 8.8.6.6
      - ip_address: 2001:4860:4860::8800
      port: 53
    forward_policy: first
    allow_sync_ptr: yes
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file establish-global-forwarder.yml
```

추가 리소스

•

`/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용할 수 있는 **README-dnsconfig.md** Markdown 파일에서 **ansible-freeipa ipadnsconfig** 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 **ipadnsconfig** 변수 정의도 포함되어 있습니다.

81.8. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에 **DNS** 글로벌 전달자가 있는지 확인하는 방법에 대해 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 **IP(인터넷 프로토콜) v4** 주소가 **7.7**이고 **IP v6** 주소 **2001:db8::1:0**에 **DNS** 글로벌 전달자가 **DNS** 서버에 있는지 확인합니다. 포트 **53**에.

사전 요구 사항

•

Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

•

IdM 관리자 암호를 알고 있습니다.

절차

1.

`/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2.

인벤토리 파일을 열고 구성할 IdM 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3.

forwarders-absent.yml **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4.

편집을 위해 **ensure-presence-of-a-global-forwarder.yml** 파일을 엽니다.

5.

다음 변수를 설정하여 파일을 조정합니다.

a.

플레이북의 **name** 변수를 플레이북으로 변경하여 **IdM DNS**에 글로벌 전달자가 있는지 확인합니다.

b.

tasks 섹션에서 작업 이름을 변경하여 **DNS** 글로벌 전달자가 포트 **53**에 **7.7. 및 2001:db8::1:0**으로 있는지 확인합니다.

c.

ipadnsconfig 부분의 **forwarders** 섹션에서 다음을 수행합니다.

i.

첫 번째 **ip_address** 값을 글로벌 전달자의 **IPv4** 주소로 변경합니다. **7.7.9.9**.

ii.

두 번째 **ip_address** 값을 글로벌 전달자의 **IPv6** 주소로 변경합니다. **2001:db8::1:0**.

iii.

port 값이 **53**으로 설정되어 있는지 확인합니다.

d.

상태를 **present**로 변경합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```

---
- name: Playbook to ensure the presence of a global forwarder in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0
      on port 53
      ipadnsconfig:
        forwarders:
          - ip_address: 7.7.9.9
          - ip_address: 2001:db8::1:0
          port: 53
          state: present

```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-presence-of-a-global-forwarder.yml
```

추가 리소스

•

`/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용할 수 있는 **README-dnsconfig.md** Markdown 파일에서 **ansible-freeipa** **ipadnsconfig** 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 **ipadnsconfig** 변수 정의도 포함되어 있습니다.

81.9. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 없는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에 **DNS** 글로벌 전달자가 없는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 **IP**(인터넷 프로토콜) **v4** 주소 **8.8.6.6** 및 **IP v6** 주소 **2001:4860:4860::8800**의 **DNS** 글로벌 전달자가 없는지 확인합니다.

사전 요구 사항

•

Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

•

IdM 관리자 암호를 알고 있습니다.

절차

1.

`/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2.

인벤토리 파일을 열고 구성할 IdM 서버가 `[ipaserver]` 섹션에 나열되어 있는지 확인합니다. 예를 들어 `server.idm.example.com`을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3.

`forwarders-absent.yml` Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4.

편집을 위해 `ensure-absence-of-a-global-forwarder.yml` 파일을 엽니다.

5.

다음 변수를 설정하여 파일을 조정합니다.

a.

플레이북의 `name` 변수를 플레이북으로 변경하여 IdM DNS에서 글로벌 전달자가 없는지 확인합니다.

b.

`tasks` 섹션에서 DNS 글로벌 전달자가 없는지 확인하고 포트 53에 `2001:4860:4860::8800`으로 작업 이름을 변경합니다.

c.

`ipadnsconfig` 부분의 `forwarders` 섹션에서 다음을 수행합니다.

i.

첫 번째 `ip_address` 값을 글로벌 전달자의 IPv4 주소로 변경합니다. `8.8.6.6`.

ii.

두 번째 `ip_address` 값을 글로벌 전달자의 IPv6 주소로 변경합니다. `2001:4860:4860::8800`.

iii.

`port` 값이 53으로 설정되어 있는지 확인합니다.

d.

상태가 **absent** 로 설정되어 있는지 확인합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to ensure the absence of a global forwarder in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure the absence of a DNS global forwarder to 8.8.6.6 and
      2001:4860:4860::8800 on port 53
      ipadnsconfig:
        forwarders:
          - ip_address: 8.8.6.6
          - ip_address: 2001:4860:4860::8800
        port: 53
        state: absent
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-absence-of-a-global-forwarder.yml
```

추가 리소스

•

`/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용할 수 있는 **README-dnsconfig.md** Markdown 파일에서 **ansible-freeipa** **ipadnsconfig** 모듈에 대한 샘플 **Ansible** 플레이북을 확인할 수 있습니다. 파일에는 **ipadnsconfig** 변수 정의도 포함되어 있습니다.

81.10. ANSIBLE을 사용하여 DNS GLOBAL FORWARDERS가 IDM에서 비활성화되었는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에서 **DNS** 글로벌 전달자를 비활성화하는 방법에 대해 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 글로벌 전달자의 전달 정책이 **none** 으로 설정되어 글로벌 전달자를 효과적으로 비활성화합니다.

사전 요구 사항

•

Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

- **IdM 관리자 암호를 알고 있습니다.**

절차

1. **`/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 디렉터리로 이동합니다.**

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. **인벤토리 파일을 열고 구성할 IdM 서버가 `[ipaserver]` 섹션에 나열되어 있는지 확인합니다. 예를 들어 `server.idm.example.com`을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.**

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

3. **모든 DNS 글로벌 전달자를 비활성화하도록 이미 구성된 `disable-global-forwarders.yml` Ansible 플레이북 파일의 내용을 확인합니다. 예를 들면 다음과 같습니다.**

```
$ cat disable-global-forwarders.yml
---
- name: Playbook to disable global DNS forwarders
  hosts: ipaserver
  become: true

  tasks:
    - name: Disable global forwarders.
      ipadnsconfig:
        forward_policy: none
```

4. **플레이북을 실행합니다.**

```
$ ansible-playbook -v -i inventory.file disable-global-forwarders.yml
```

추가 리소스

- **`/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용할 수 있는 `README-dnsconfig.md` Markdown 파일에서 `ansible-freeipa ipadnsconfig` 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 `ipadnsconfig` 변수 정의도 포함되어 있습니다.**

81.11. ANSIBLE을 사용하여 IDM에 DNS 전달 영역이 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에 **DNS** 전달 영역이 있는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 **IP(인터넷 프로토콜)** 주소가 **8.8.8.8** 인 **DNS** 서버에 **example.com** 의 **DNS** 전달 영역이 있는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **forwarders-absent.yml** **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-forwardzone.yml
```

4. 편집을 위해 **ensure-presence-forwardzone.yml** 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

- a. 플레이북의 **name** 변수를 플레이북으로 변경하여 **IdM DNS**에 **dnsforwardzone**이 있는지 확인합니다.

- b. **tasks** 섹션에서 **example.com**에 대한 **dnsforwardzone**이 있는지 확인하도록 작업 이름을 **8.8.8.8**로 변경합니다.

- c. **tasks** 섹션에서 **ipadnsconfig** 제목을 **ipadns forwardzone** 으로 변경합니다.
- d. **ipadnsforwardzone** 섹션에서 다음을 수행합니다.
 - i. **ipaadmin_password** 변수를 추가하고 **IdM** 관리자 암호로 설정합니다.
 - ii. **name** 변수를 추가하고 **example.com** 으로 설정합니다.
 - iii. **forwarders** 섹션에서 다음을 수행합니다.
 - A. **ip_address** 및 포트 행을 제거합니다.
 - B. 대시 다음에 지정하여 전달된 요청을 받기 위해 **DNS** 서버의 **IP** 주소를 추가합니다.


```

- 8.8.8.8

```
 - iv. **forwardpolicy** 변수를 추가하고 **first** 로 설정합니다.
 - v. **skip_overlap_check** 변수를 추가하고 **true** 로 설정합니다.
 - vi. **state** 변수를 **present** 로 변경합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```

---
- name: Playbook to ensure the presence of a dnsforwardzone in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure the presence of a dnsforwardzone for example.com to 8.8.8.8

```

```

ipadnsforwardzone:
  ipaadmin_password: password01
  name: example.com
  forwarders:
    - 8.8.8.8
  forwardpolicy: first
  skip_overlap_check: true
  state: present

```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-presence-forwardzone.yml
```

추가 리소스

- **/usr/share/doc/ansible-freeipa/ 디렉터리에서 사용할 수 있는 README-dnsforwardzone.md Markdown 파일에서 ansible-freeipa ipadnsforwardzone 모듈에 대한 더 많은 샘플 Ansible 플레이북을 볼 수 있습니다. 파일에는 ipadnsforwardzone 변수의 정의도 포함되어 있습니다.**

81.12. ANSIBLE을 사용하여 DNS FORWARD ZONE에 IDM에 여러 개의 전달자가 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM의 DNS Forward Zone**에 여러 전달자가 있는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 **example.com**의 **DNS** 전달 영역이 **8.8.8.8** 및 **4.4.4.4**로 전달되고 있는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

절차

1.

```
/usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.
```

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2.

인벤토리 파일을 열고 구성할 IdM 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3.

forwarders-absent.yml **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-multiple-forwarders.yml
```

4.

편집을 위해 **ensure-presence-multiple-forwarders.yml** 파일을 엽니다.

5.

다음 변수를 설정하여 파일을 조정합니다.

a.

플레이북의 **name** 변수를 플레이북으로 변경하여 IdM DNS의 **dnsforwardzone**에 여러 전달자가 있는지 확인합니다.

b.

tasks 섹션에서 **example.com**의 **dnsforwardzone**에서 **8.8.8.8** 및 **4.4.4.4** 전달자가 있는지 확인하도록 작업 이름을 변경합니다.

c.

tasks 섹션에서 **ipadnsconfig** 제목을 **ipadns forwardzone** 으로 변경합니다.

d.

ipadnsforwardzone 섹션에서 다음을 수행합니다.

i.

ipaadmin_password 변수를 추가하고 IdM 관리자 암호로 설정합니다.

ii.

name 변수를 추가하고 **example.com** 으로 설정합니다.

iii.

forwarders 섹션에서 다음을 수행합니다.

A.

ip_address 및 포트 행을 제거합니다.

B.

확인하려는 **DNS** 서버의 **IP** 주소를 추가합니다. 앞에 대시가 있습니다.

```
- 8.8.8.8
- 4.4.4.4
```

iv.

state 변수를 **present**로 변경합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: name: Playbook to ensure the presence of multiple forwarders in a
  dnsforwardzone in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure presence of 8.8.8.8 and 4.4.4.4 forwarders in dnsforwardzone for
      example.com
      ipadnsforwardzone:
        ipadmin_password: password01
        name: example.com
        forwarders:
          - 8.8.8.8
          - 4.4.4.4
        state: present
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-presence-multiple-forwarders.yml
```

추가 리소스

•

`/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용할 수 있는 **README-dnsforwardzone.md** Markdown 파일에서 **ansible-freeipa ipadnsforwardzone** 모듈에 대한 더 많은 샘플 **Ansible** 플레이북을 볼 수 있습니다. 파일에는 **ipadnsforwardzone** 변수의 정의도 포함되어 있습니다.

81.13. ANSIBLE을 사용하여 **IDM**에서 **DNS** 전달 영역이 비활성화되어 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에서 **DNS Forward Zone**을 비활성화하는 방법을 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 **example.com**의 **DNS** 전달 영역이 비활성화되었는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **forwarders-absent.yml** **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-disabled-forwardzone.yml
```

4. 편집을 위해 **ensure-disabled-forwardzone.yml** 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

- a. 플레이북의 **name** 변수를 플레이북으로 변경하여 **IdM DNS**에서 **dnsforwardzone**이 비활성화되었는지 확인합니다.

- b. **tasks** 섹션에서 **example.com**의 **dnsforwardzone**이 비활성화되어 있는지 확인하도록 작업 이름을 변경합니다.

- c. **tasks** 섹션에서 **ipadnsconfig** 제목을 **ipadns forwardzone** 으로 변경합니다.
- d. **ipadnsforwardzone** 섹션에서 다음을 수행합니다.
 - i. **ipaadmin_password** 변수를 추가하고 **IdM** 관리자 암호로 설정합니다.
 - ii. **name** 변수를 추가하고 **example.com** 으로 설정합니다.
 - iii. 전체 **forwarders** 섹션을 제거합니다.
 - iv. **state** 변수를 **disabled** 로 변경합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to ensure a dnsforwardzone is disabled in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure a dnsforwardzone for example.com is disabled
      ipadnsforwardzone:
        ipaadmin_password: password01
        name: example.com
        state: disabled
```

- 6. 파일을 저장합니다.
- 7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-disabled-forwardzone.yml
```

추가 리소스



/usr/share/doc/ansible-freeipa/ 디렉터리에서 사용할 수 있는 **README-**

dnsforwardzone.md Markdown 파일에서 **ansible-freeipa ipadnsforwardzone** 모듈에 대한 더 많은 샘플 **Ansible** 플레이북을 볼 수 있습니다. 파일에는 **ipadnsforwardzone** 변수의 정의도 포함되어 있습니다.

81.14. ANSIBLE을 사용하여 IDM에 DNS 전달 영역이 없는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에 **DNS** 전달 영역이 없는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 **example.com**에 대한 **DNS** 전달 영역이 없는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

절차

1.

/usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2.

인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3.

forwarders-absent.yml **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-absence-forwardzone.yml
```

4.

편집을 위해 **ensure-absence-forwardzone.yml** 파일을 엽니다.

5.

다음 변수를 설정하여 파일을 조정합니다.

- a. 플레이북의 **name** 변수를 플레이북으로 변경하여 **IdM DNS**에 **dnsforwardzone**이 없는지 확인합니다.
- b. **tasks** 섹션에서 **example.com**에 대한 **dnsforwardzone**이 없으면 작업 이름을 변경합니다.
- c. **tasks** 섹션에서 **ipadnsconfig** 제목을 **ipadns forwardzone** 으로 변경합니다.
- d. **ipadnsforwardzone** 섹션에서 다음을 수행합니다.
 - i. **ipaadmin_password** 변수를 추가하고 **IdM** 관리자 암호로 설정합니다.
 - ii. **name** 변수를 추가하고 **example.com** 으로 설정합니다.
 - iii. 전체 **forwarders** 섹션을 제거합니다.
 - iv. **state** 변수를 **absent** 로 둡니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to ensure the absence of a dnsforwardzone in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure the absence of a dnsforwardzone for example.com
      ipadnsforwardzone:
        ipaadmin_password: password01
        name: example.com
        state: absent
```

6. 파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-absence-forwardzone.yml
```

추가 리소스

- **`/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용할 수 있는 `README-dnsforwardzone.md` Markdown 파일에서 `ansible-freeipa ipadnsforwardzone` 모듈에 대한 더 많은 샘플 Ansible 플레이북을 볼 수 있습니다. 파일에는 `ipadnsforwardzone` 변수의 정의도 포함되어 있습니다.**

82장. IDM에서 DNS 레코드 관리

이 장에서는 **IdM(Identity Management)**에서 **DNS** 레코드를 관리하는 방법을 설명합니다. **IdM** 관리자는 **IdM**에서 **DNS** 레코드를 추가, 수정 및 삭제할 수 있습니다. 장에는 다음 섹션이 포함되어 있습니다.

- [IdM의 DNS 레코드](#)
- [IdM 웹 UI에서 DNS 리소스 레코드 추가](#)
- [IdM CLI에서 DNS 리소스 레코드 추가](#)
- [일반적인 ipa dnsrecord-add 옵션](#)
- [IdM 웹 UI에서 DNS 레코드 삭제](#)
- [IdM 웹 UI에서 전체 DNS 레코드 삭제](#)
- [IdM CLI에서 DNS 레코드 삭제](#)

사전 요구 사항

- **IdM** 배포에 통합된 **DNS** 서버가 포함되어 있습니다. 통합 **DNS**를 사용하여 **IdM**을 설치하는 방법에 대한 자세한 내용은 다음 링크 중 하나를 참조하십시오.
 - [IdM 서버 설치: 통합 DNS를 사용하면 루트 CA로 통합 CA가 포함됩니다.](#)
 - [IdM 서버 설치: 통합 DNS를 사용하면 외부 CA가 루트 CA로 설정되어 있습니다.](#)

82.1. IDM의 DNS 레코드

IdM(Identity Management)은 다양한 **DNS** 레코드 유형을 지원합니다. 다음 4가지가 가장 자주 사용됩니다.

A

호스트 이름과 IPv4 주소에 대한 기본 맵입니다. **A** 레코드의 레코드 이름은 **www** 와 같은 호스트 이름입니다. **A** 레코드의 IP 주소 값은 **192.0.2.1** 과 같은 IPv4 주소입니다.

A 레코드에 대한 자세한 내용은 [RFC 1035](#) 를 참조하십시오.

AAAA

호스트 이름과 IPv6 주소에 대한 기본 맵입니다. **AAAA** 레코드의 레코드 이름은 **www** 와 같은 호스트 이름입니다. IP 주소 값은 **2001:DB8::1111** 과 같은 IPv6 주소입니다.

AAAA 레코드에 대한 자세한 내용은 [RFC 3596](#) 을 참조하십시오.

SRV

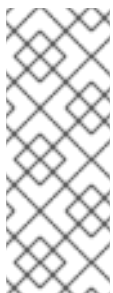
서비스(**SRV**) 리소스 레코드는 서비스 이름을 특정 서비스를 제공하고 있는 서버의 **DNS** 이름에 매핑합니다. 예를 들어 이 레코드 유형은 **LDAP** 디렉터리와 같은 서비스를 관리하는 서버에 매핑할 수 있습니다.

SRV 레코드의 레코드 이름은 **_service** 형식을 갖습니다. **_ldap._tcp**와 같은 **_protocol**. **SRV** 레코드의 구성 옵션에는 대상 서비스의 우선 순위, 가중치, 포트 번호 및 호스트 이름이 포함됩니다.

SRV 레코드에 대한 자세한 내용은 [RFC 2782](#) 을 참조하십시오.

PTR

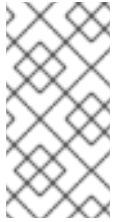
PTR(포인터 레코드)은 IP 주소를 도메인 이름에 매핑하는 역방향 **DNS** 레코드를 추가합니다.



참고

IPv4 주소에 대한 모든 역방향 **DNS** 조회는 **in-addr.arpa**. 도메인에 정의된 역방향 항목을 사용합니다. 역방향 주소는 사람이 읽을 수 있는 형식으로, 일반 IP 주소의 정확한 역순이며 **in-addr.arpa**. 도메인이 추가됩니다. 예를 들어 네트워크 주소 **192.0.2.0/24** 의 경우 역방향 영역은 **2.0.192.in-addr.arpa** 입니다.

PTR의 레코드 이름은 [RFC 1035](#)에 지정된 표준 형식이어야 하며 [RFC 2317](#) 및 [RFC 3596](#) 에서 확장되어야 합니다. 호스트 이름 값은 레코드를 만들 호스트의 정식 호스트 이름이어야 합니다.



참고

또한, **ip6.arpa** 도메인의 영역을 사용하여 **IPv6** 주소에 대해 역방향 영역을 구성할 수 있습니다. **IPv6** 역방향 영역에 대한 자세한 내용은 [RFC 3596](#) 을 참조하십시오.

DNS 리소스 레코드를 추가할 때 많은 레코드에 다른 데이터가 필요합니다. 예를 들어 **CNAME** 레코드에는 호스트 이름이 필요하지만 **A** 레코드에는 **IP** 주소가 필요합니다. **IdM** 웹 **UI**에서 현재 선택한 레코드 유형에 필요한 데이터를 반영하도록 새 레코드를 추가하기 위한 양식의 필드가 자동으로 업데이트됩니다.

82.2. IDM 웹 UI에서 DNS 리소스 레코드 추가

이 섹션에서는 **IdM(Identity Management)** 웹 **UI**에 **DNS** 리소스 레코드를 추가하는 방법을 설명합니다.

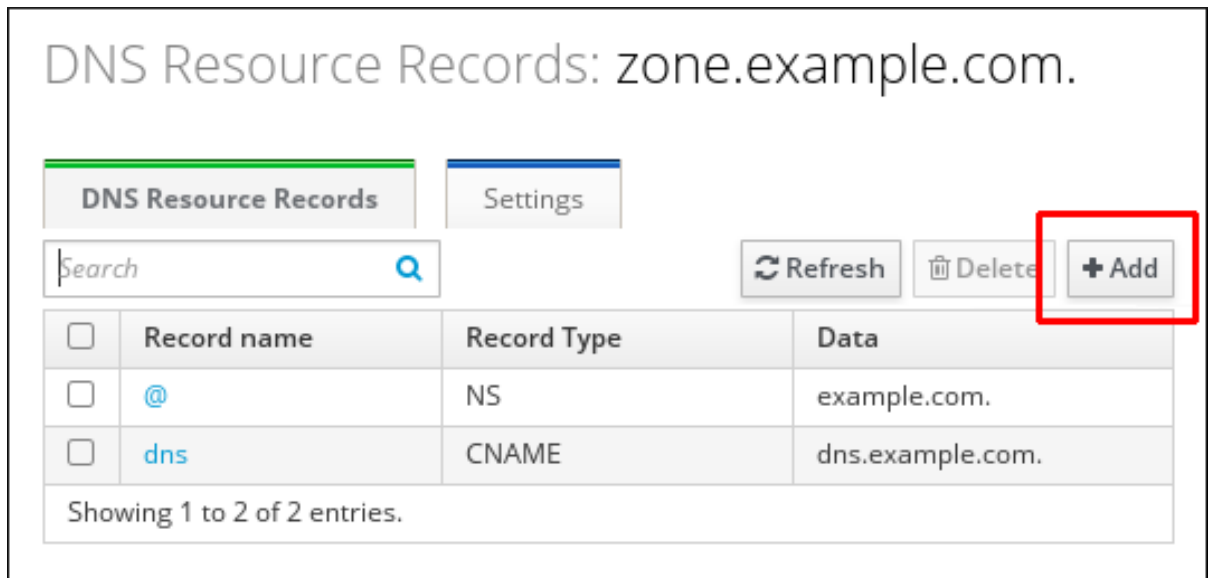
사전 요구 사항

- **DNS** 레코드를 추가하려는 **DNS** 영역이 존재하며 **IdM**에서 관리합니다. **IdM DNS**에서 **DNS** 영역 생성에 대한 자세한 내용은 **IdM의 DNS 영역 관리**를 참조하십시오.
- **IdM** 관리자로 로그인했습니다.

절차

1. **IdM** 웹 **UI**에서 네트워크 서비스 → **DNS** → **DNS** 영역을 클릭합니다.
2. **DNS** 레코드를 추가할 **DNS** 영역을 클릭합니다.
3. **DNS** 리소스 레코드 섹션에서 **Add(추가)**를 클릭하여 새 레코드를 추가합니다.

그림 82.1. 새 DNS 리소스 레코드 추가



4.

필요에 따라 다른 필드를 만들고 채울 레코드 유형을 선택합니다.

그림 82.2. 새 DNS 리소스 레코드 정의

Add DNS Resource Record ✕

Record name *

Record Type

Hostname *

* Required field

Add Add and Add Another Add and Edit Cancel

5.

Add(추가) 를 클릭하여 새 레코드를 확인합니다.

82.3. IDM CLI에서 DNS 리소스 레코드 추가

이 섹션에서는 **CLI**(명령줄 인터페이스)에서 모든 유형의 **DNS** 리소스 레코드를 추가하는 방법을 설명합니다.

사전 요구 사항

- **DNS 레코드를 추가할 DNS 영역입니다. IdM DNS에서 DNS 영역 생성에 대한 자세한 내용은 IdM의 [DNS 영역 관리](#)를 참조하십시오.**
- **IdM 관리자로 로그인했습니다.**

절차

1. **DNS 리소스 레코드를 추가하려면 `ipa dnsrecord-add` 명령을 사용합니다. 명령은 다음 구문을 따릅니다.**

```
$ ipa dnsrecord-add zone_name record_name --record_type_option=data
```

위의 명령에서 다음을 수행합니다.

- **zone_name** 은 레코드를 추가할 **DNS 영역의 이름**입니다.
- **record_name** 은 새 **DNS 리소스 레코드**의 식별자입니다.

예를 들어 **host1** 의 **A** 유형 **DNS 레코드**를 **idm.example.com** 영역에 추가하려면 다음을 입력합니다.

```
$ ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123
```

82.4. 일반적인 IPA DNSRECORD-* 옵션

이 섹션에서는 **IdM(Identity Management)**에 가장 일반적인 **DNS 리소스 레코드 유형**을 추가, 수정 및 삭제할 때 사용할 수 있는 옵션에 대해 설명합니다.

- **A (IPv4)**
- **AAAA (IPv6)**

- **SRV**
- **PTR**

Bash에서는 중괄호 안의 쉼표로 구분된 목록에 값을 나열하여 여러 항목을 정의할 수 있습니다(예: `--option={val1,val2,val3}`).

표 82.1. 일반 레코드 옵션

옵션	설명
<code>--ttl=number</code>	레코드에 사용할 시간을 설정합니다.
<code>--structured</code>	원시 DNS 레코드를 구문 분석하고 구조화된 형식으로 반환합니다.

표 82.2. "a" 레코드 옵션

옵션	설명	예
<code>--a-rec=ARECORD</code>	하나의 A 레코드 또는 하나의 A 레코드 목록을 전달합니다.	<code>ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123</code>
	지정된 IP 주소가 있는 와일드카드 A 레코드를 만들 수 있습니다.	<code>ipa dnsrecord-add idm.example.com "*" --a-rec=192.168.122.123^[a]</code>
<code>--a-ip-address=string</code>	레코드의 IP 주소를 제공합니다. 레코드를 만들 때 A 레코드 값을 지정하는 옵션은 <code>--a-rec</code> 입니다. 그러나 A 레코드를 수정할 때 <code>--a-rec</code> 옵션은 A 레코드의 현재 값을 지정하는 데 사용됩니다. 새 값은 <code>--a-ip-address</code> 옵션으로 설정됩니다.	<code>ipa dnsrecord-mod idm.example.com --a-rec 192.168.122.123 --a-ip-address 192.168.122.124</code>
[a] 이 예제에서는 IP 주소가 192.0.2.123인 와일드카드 A 레코드를 생성합니다.		

표 82.3. "AAAA" 레코드 옵션

옵션	설명	예제
<code>--aaaa-rec=AAAARECORD</code>	단일 AAAA(IPv6) 레코드 또는 AAAA 레코드 목록을 전달합니다.	<code>ipa dnsrecord-add idm.example.com www --aaaa-rec 2001:db8::1231:5675</code>

옵션	설명	예제
--aaaa-ip-address=string	레코드에 IPv6 주소를 제공합니다. 레코드를 만들 때 A 레코드 값을 지정하는 옵션은 --aaa-rec 입니다. 그러나 A 레코드를 수정할 때 --aaaa-rec 옵션은 A 레코드의 현재 값을 지정하는 데 사용됩니다. 새 값은 --a-ip-address 옵션으로 설정됩니다.	IPA dnsrecord-mod idm.example.com --aaaa-rec 2001:db8::1231:5675 --aaa- ip-address 2001:db8::1231:5676

표 82.4. "PTR" 레코드 옵션

옵션	설명	예제
--ptr-rec=PTRREC ORD	단일 PTR 레코드 또는 PTR 레코드 목록을 전달합니다. 역방향 DNS 레코드를 추가할 때 다른 DNS 레코드를 추가하는 사용과 비교하여 ipa dnsrecord-add 명령과 함께 사용되는 영역 이름이 되돌립니다. 일반적으로 호스트 IP 주소는 지정된 네트워크에 있는 IP 주소의 마지막 옥텟입니다. 오른쪽의 첫 번째 예에서는 server4.idm.example.com 에 IPv4 주소 192.168.122.4 인 PTR 레코드를 추가합니다. 두 번째 예제는 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa 에 역방향 DNS 항목을 추가합니다. IP 주소가 2001:DB8::1111 인 호스트 server2.example.com 의 IPv6 역방향 영역입니다.	ipa dnsrecord-add 122.168.192.in-addr.arpa 4 -- ptr-rec server4.idm.example.com. \$ ipa dnsrecord-add 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.i p6.arpa. 1.1.1.0.0.0.0.0.0.0.0.0.0.0.0 -- ptr-rec server2.idm.example.com.
--ptr-hostname=string	레코드의 호스트 이름을 제공합니다.	

표 82.5. "SRV" 레코드 옵션

옵션	설명	예제
--srv-rec=SRVRECORD	단일 SRV 레코드 또는 SRV 레코드 목록을 전달합니다. 오른쪽 예제에서 _ldap.tcp 는 SRV 레코드의 서비스 유형 및 연결 프로토콜을 정의합니다. srv-rec 옵션은 우선순위, 가중치, 포트, 대상 값을 정의합니다. 예에서 57 및 49의 가중치 값은 100까지 추가하고 특정 레코드가 사용되는 백분율로 가중성을 나타냅니다.	# ipa dnsrecord-add idm.example.com _ldap.tcp --srv- rec="0 51 389 server1.idm.example.com." # IPA dnsrecord-add server.idm.example.com _ldap.tcp --srv-rec 49 389 server2.idm.example.com".
--srv-priority=number	레코드의 우선 순위를 설정합니다. 서비스 유형에 대한 SRV 레코드가 여러 개 있을 수 있습니다. 우선 순위 (0 - 65535)는 레코드의 순위를 설정합니다. 숫자가 작을수록 우선 순위가 높습니다. 서비스는 우선 순위가 가장 높은 레코드를 먼저 사용해야 합니다.	# IPA dnsrecord-mod server.idm.example.com _ldap.tcp --srv-rec 49 389 server2.idm.example.com." --srv- priority=0

옵션	설명	예제
--srv-weight=number	레코드의 가중치를 설정합니다. 이렇게 하면 동일한 우선 순위의 SRV 레코드 순서를 결정하는 데 도움이 됩니다. 설정 가중치는 특정 레코드가 사용되는 가능성(백분율)을 나타내는 최대 100개까지 추가해야 합니다.	# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec 49 389 server2.idm.example.com." --srv-weight=60
--srv-port=number	대상 호스트에서 서비스에 대한 포트를 제공합니다.	# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec 60 389 server2.idm.example.com." --srv-port=636
--srv-target=string	대상 호스트의 도메인 이름을 제공합니다. 도메인에서 서비스를 사용할 수 없는 경우 단일 마침표(.)일 수 있습니다.	

추가 리소스

-

ipa dnsrecord-add 및 **IdM**에서 지원하는 **DNS** 레코드 유형을 사용하는 방법에 대한 자세한 내용을 보려면 **ipa dnsrecord-add --help** 명령을 실행합니다.

82.5. IDM 웹 UI에서 DNS 레코드 삭제

이 섹션에서는 **IdM** 웹 UI를 사용하여 **IdM(Identity Management)**에서 **DNS** 레코드를 삭제하는 방법을 설명합니다.

사전 요구 사항

-

IdM 관리자로 로그인했습니다.

절차

- 1.

IdM 웹 UI에서 네트워크 서비스 → **DNS** → **DNS** 영역을 클릭합니다.

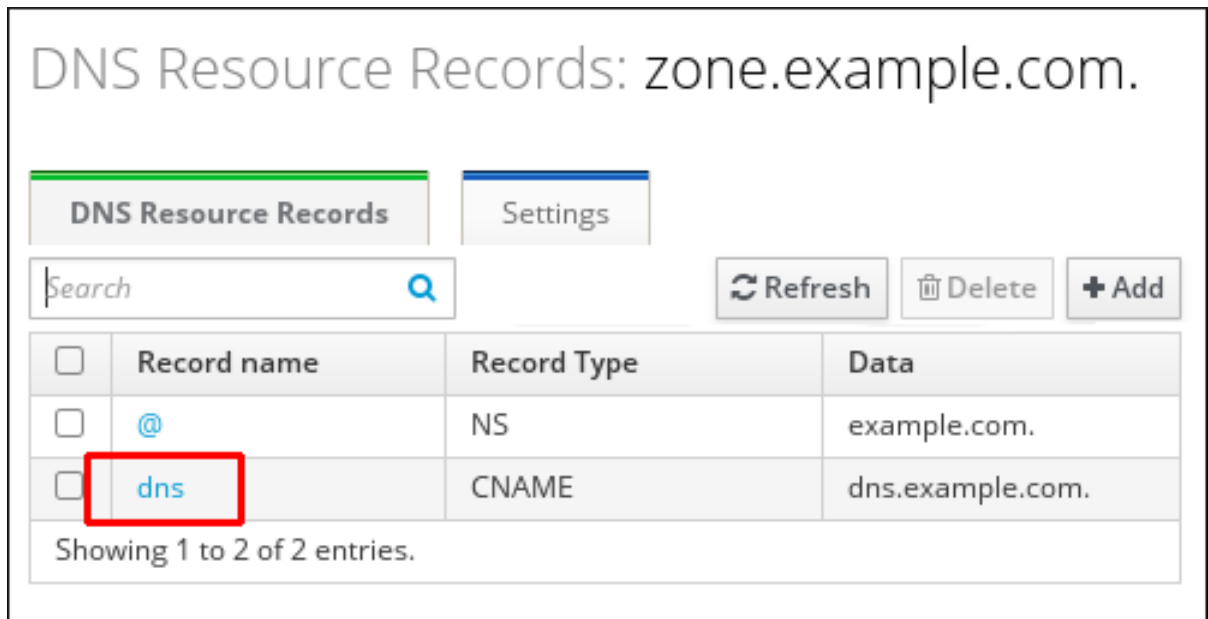
- 2.

DNS 레코드를 삭제할 영역을 클릭합니다(예: **example.com.**).

3.

DNS 리소스 레코드 섹션에서 리소스 레코드의 이름을 클릭합니다.

그림 82.3. DNS 리소스 레코드 선택



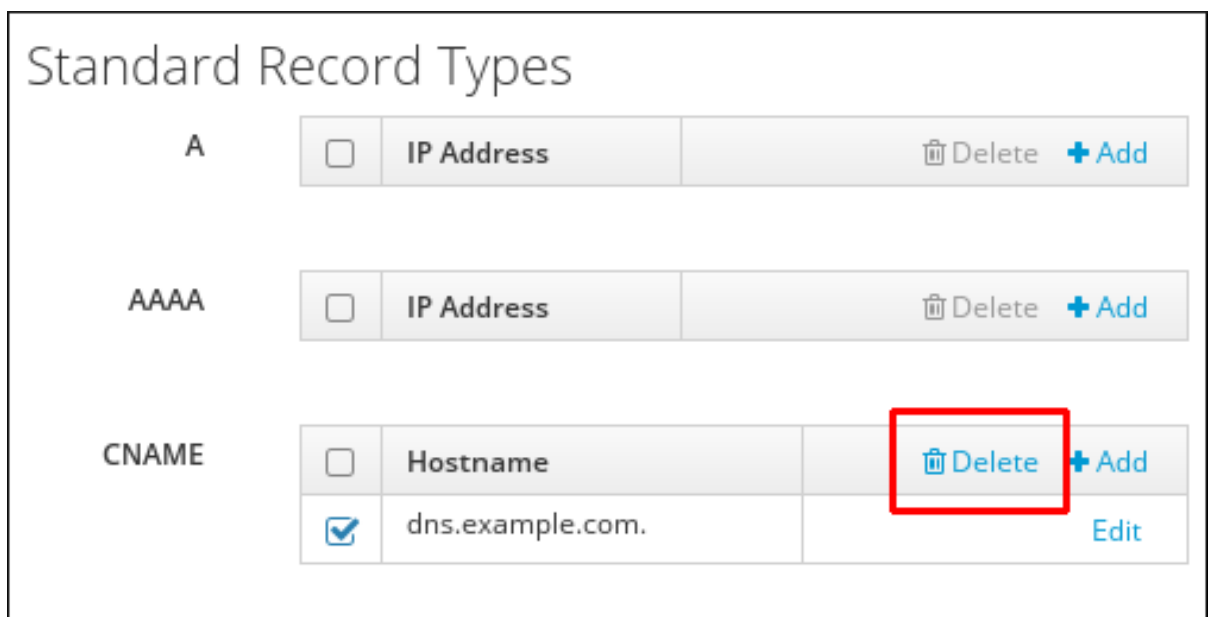
4.

삭제할 레코드 유형의 이름으로 확인란을 선택합니다.

5.

삭제를 클릭합니다.

그림 82.4. DNS 리소스 레코드 삭제



이제 선택한 레코드 유형이 삭제됩니다. 리소스 레코드의 다른 구성은 그대로 유지됩니다.

- 전체 DNS 레코드 삭제에 대한 자세한 내용은 **IdM 웹 UI에서 전체 DNS 레코드 삭제**를 참조하십시오.

82.6. IdM 웹 UI에서 전체 DNS 레코드 삭제

이 섹션에서는 **IdM(Identity Management)** 웹 UI를 사용하여 영역의 특정 리소스에 대한 모든 레코드를 삭제하는 방법을 설명합니다.

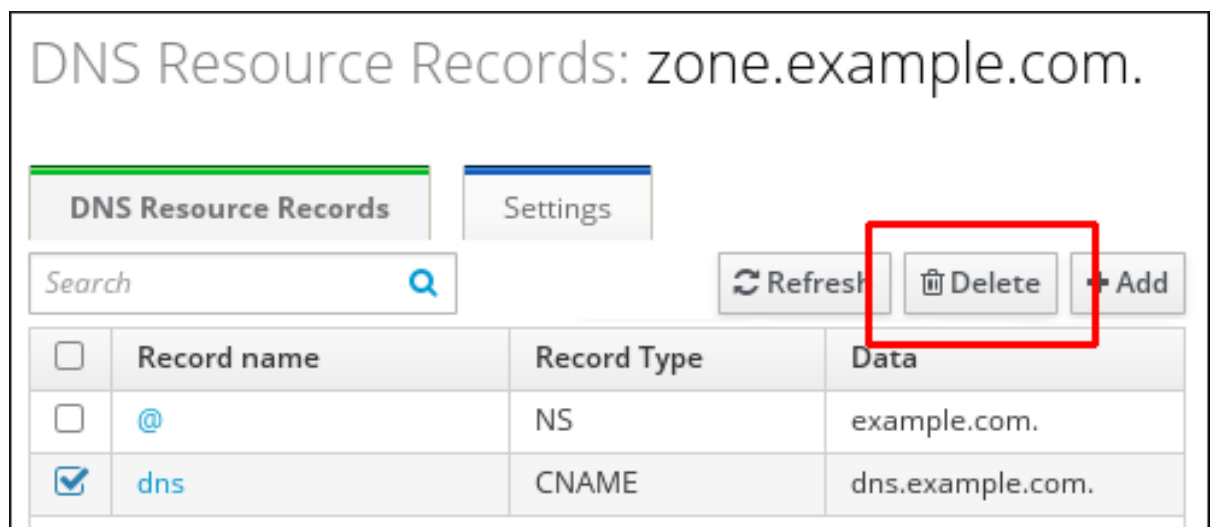
사전 요구 사항

- **IdM 관리자**로 로그인했습니다.

절차

1. **IdM 웹 UI에서 네트워크 서비스 → DNS → DNS 영역**을 클릭합니다.
2. **DNS 레코드를 삭제할 영역**을 클릭합니다(예: **zone.example.com.**).
3. **DNS 리소스 레코드 섹션에서 삭제할 리소스 레코드의 확인란**을 선택합니다.
4. **삭제**를 클릭합니다.

그림 82.5. 전체 리소스 레코드 삭제



이제 전체 리소스 레코드가 삭제됩니다.

82.7. IDM CLI에서 DNS 레코드 삭제

이 섹션에서는 **IdM(Identity Management) DNS**에서 관리하는 영역에서 **DNS** 레코드를 제거하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.

절차

- 영역에서 레코드를 제거하려면 **ipa dnsrecord-del** 명령을 사용하고 레코드 값과 함께 **-recordType-rec** 옵션을 추가합니다. 예를 들어 **A** 유형 레코드를 제거하려면 다음을 수행합니다.

```
$ ipa dnsrecord-del example.com www --a-rec 192.0.2.1
```

옵션 없이 **ipa dnsrecord-del** 을 실행하면 삭제할 레코드에 대한 정보를 입력하라는 메시지가 표시됩니다. 명령에 **--del-all** 옵션을 전달하면 영역에 대한 연결된 모든 레코드가 제거됩니다.

추가 리소스

- **ipa dnsrecord-del** 및 명령에서 수락한 전체 옵션 목록을 사용하는 방법에 대한 자세한 내용을 보려면 **ipa dnsrecord-del --help** 명령을 실행합니다.

82.8. 추가 리소스

- **ansible-freeipa ipadnsrecord** 모듈을 사용하여 **IdM DNS**에서 레코드를 추가, 수정 및 삭제할 수 있습니다. 자세한 내용은 **IdM에서 Ansible을 사용하여 DNS 레코드 관리**를 참조하십시오.

83장. ANSIBLE을 사용하여 IDM에서 DNS 레코드 관리

이 장에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에서 **DNS** 레코드를 관리하는 방법을 설명합니다. **IdM** 관리자는 **IdM**에서 **DNS** 레코드를 추가, 수정, 삭제할 수 있습니다. 장에는 다음 섹션이 포함되어 있습니다.

- [Ansible을 사용하여 IdM에 A 및 AAAA DNS 레코드가 있는지 확인](#)
- [Ansible을 사용하여 IdM에 A 및 PTR DNS 레코드가 있는지 확인](#)
- [Ansible을 사용하여 IdM에 여러 DNS 레코드가 있는지 확인](#)
- [Ansible을 사용하여 IdM에 여러 CNAME 레코드가 있는지 확인](#)
- [Ansible을 사용하여 IdM에 SRV 레코드가 있는지 확인](#)

83.1. IDM의 DNS 레코드

IdM(Identity Management)은 다양한 **DNS** 레코드 유형을 지원합니다. 다음 4가지가 가장 자주 사용됩니다.

A

호스트 이름과 **IPv4** 주소에 대한 기본 맵입니다. **A** 레코드의 레코드 이름은 **www** 와 같은 호스트 이름입니다. **A** 레코드의 IP 주소 값은 **192.0.2.1** 과 같은 **IPv4** 주소입니다.

A 레코드에 대한 자세한 내용은 [RFC 1035](#) 를 참조하십시오.

AAAA

호스트 이름과 **IPv6** 주소에 대한 기본 맵입니다. **AAAA** 레코드의 레코드 이름은 **www** 와 같은 호스트 이름입니다. IP 주소 값은 **2001:DB8::1111** 과 같은 **IPv6** 주소입니다.

AAAA 레코드에 대한 자세한 내용은 [RFC 3596](#) 을 참조하십시오.

SRV

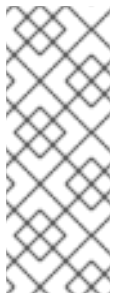
서비스(SRV) 리소스 레코드는 서비스 이름을 특정 서비스를 제공하고 있는 서버의 DNS 이름에 매핑합니다. 예를 들어 이 레코드 유형은 LDAP 디렉터리와 같은 서비스를 관리하는 서버에 매핑할 수 있습니다.

SRV 레코드의 레코드 이름은 `_service` 형식을 갖습니다. `_ldap._tcp`와 같은 `_protocol`. SRV 레코드의 구성 옵션에는 대상 서비스의 우선 순위, 가중치, 포트 번호 및 호스트 이름이 포함됩니다.

SRV 레코드에 대한 자세한 내용은 [RFC 2782](#) 을 참조하십시오.

PTR

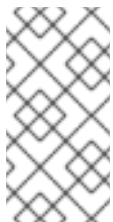
PTR(포인터 레코드)은 IP 주소를 도메인 이름에 매핑하는 역방향 DNS 레코드를 추가합니다.



참고

IPv4 주소에 대한 모든 역방향 DNS 조회는 `in-addr.arpa`. 도메인에 정의된 역방향 항목을 사용합니다. 역방향 주소는 사람이 읽을 수 있는 형식으로, 일반 IP 주소의 정확한 역순이며 `in-addr.arpa`. 도메인이 추가됩니다. 예를 들어 네트워크 주소 `192.0.2.0/24`의 경우 역방향 영역은 `2.0.192.in-addr.arpa` 입니다.

PTR의 레코드 이름은 [RFC 1035](#)에 지정된 표준 형식이어야 하며 [RFC 2317](#) 및 [RFC 3596](#)에서 확장되어야 합니다. 호스트 이름 값은 레코드를 만들 호스트의 정식 호스트 이름이어야 합니다.



참고

또한, `ip6.arpa`. 도메인의 영역을 사용하여 IPv6 주소에 대해 역방향 영역을 구성할 수 있습니다. IPv6 역방향 영역에 대한 자세한 내용은 [RFC 3596](#) 을 참조하십시오.

DNS 리소스 레코드를 추가할 때 많은 레코드에 다른 데이터가 필요합니다. 예를 들어 CNAME 레코드에는 호스트 이름이 필요하지만 A 레코드에는 IP 주소가 필요합니다. IdM 웹 UI에서 현재 선택한 레코드 유형에 필요한 데이터를 반영하도록 새 레코드를 추가하기 위한 양식의 필드가 자동으로 업데이트됩니다.

83.2. 일반적인 IPA DNSRECORD-* 옵션

이 섹션에서는 IdM(Identity Management)에 가장 일반적인 DNS 리소스 레코드 유형을 추가, 수정 및 삭제할 때 사용할 수 있는 옵션에 대해 설명합니다.

- **A (IPv4)**
- **AAAA (IPv6)**
- **SRV**
- **PTR**

Bash에서는 중괄호 안의 쉼표로 구분된 목록에 값을 나열하여 여러 항목을 정의할 수 있습니다(예: `--option={val1,val2,val3}`).

표 83.1. 일반 레코드 옵션

옵션	설명
<code>--ttl=number</code>	레코드에 사용할 시간을 설정합니다.
<code>--structured</code>	원시 DNS 레코드를 구문 분석하고 구조화된 형식으로 반환합니다.

표 83.2. "a" 레코드 옵션

옵션	설명	예
<code>--a-rec=ARECORD</code>	하나의 A 레코드 또는 하나의 A 레코드 목록을 전달합니다.	<code>ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123</code>
	지정된 IP 주소가 있는 와일드카드 A 레코드를 만들 수 있습니다.	<code>ipa dnsrecord-add idm.example.com "*" --a-rec=192.168.122.123^[a]</code>
<code>--a-ip-address=string</code>	레코드의 IP 주소를 제공합니다. 레코드를 만들 때 A 레코드 값을 지정하는 옵션은 <code>--a-rec</code> 입니다. 그러나 A 레코드를 수정할 때 <code>--a-rec</code> 옵션은 A 레코드의 현재 값을 지정하는 데 사용됩니다. 새 값은 <code>--a-ip-address</code> 옵션으로 설정됩니다.	<code>ipa dnsrecord-mod idm.example.com --a-rec 192.168.122.123 --a-ip-address 192.168.122.124</code>
[a] 이 예제에서는 IP 주소가 192.0.2.123인 와일드카드 A 레코드를 생성합니다.		

표 83.3. "AAAA" 레코드 옵션

옵션	설명	예제
--aaaa-rec=AAAARECORD	단일 AAAA(IPv6) 레코드 또는 AAAA 레코드 목록을 전달합니다.	ipa dnsrecord-add idm.example.com www -- aaaa-rec 2001:db8::1231:5675
--aaaa-ip-address=string	레코드에 IPv6 주소를 제공합니다. 레코드를 만들 때 A 레코드 값을 지정하는 옵션은 --aaa-rec 입니다. 그러나 A 레코드를 수정할 때 --aaaa-rec 옵션은 A 레코드의 현재 값을 지정하는 데 사용됩니다. 새 값은 --a-ip-address 옵션으로 설정됩니다.	IPA dnsrecord-mod idm.example.com --aaaa-rec 2001:db8::1231:5675 --aaa- ip-address 2001:db8::1231:5676

표 83.4. "PTR" 레코드 옵션

옵션	설명	예제
--ptr-rec=PTRRECORD	단일 PTR 레코드 또는 PTR 레코드 목록을 전달합니다. 역방향 DNS 레코드를 추가할 때 다른 DNS 레코드를 추가하는 사용과 비교하여 ipa dnsrecord-add 명령과 함께 사용되는 영역 이름이 되돌립니다. 일반적으로 호스트 IP 주소는 지정된 네트워크에 있는 IP 주소의 마지막 옥텟입니다. 오른쪽의 첫 번째 예에서는 server4.idm.example.com 에 IPv4 주소 192.168.122.4 인 PTR 레코드를 추가합니다. 두 번째 예제는 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa 에 역방향 DNS 항목을 추가합니다. IP 주소가 2001:DB8::1111 인 호스트 server2.example.com 의 IPv6 역방향 영역입니다.	ipa dnsrecord-add 122.168.192.in-addr.arpa 4 -- ptr-rec server4.idm.example.com. \$ ipa dnsrecord-add 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.i p6.arpa. 1.1.1.0.0.0.0.0.0.0.0.0.0.0.0 -- ptr-rec server2.idm.example.com.
--ptr-hostname=string	레코드의 호스트 이름을 제공합니다.	

표 83.5. "SRV" 레코드 옵션

옵션	설명	예제
--srv-rec=SRVRECORD	단일 SRV 레코드 또는 SRV 레코드 목록을 전달합니다. 오른쪽 예제에서 _ldap._tcp 는 SRV 레코드의 서비스 유형 및 연결 프로토콜을 정의합니다. srv-rec 옵션은 우선순위, 가중치, 포트, 대상 값을 정의합니다. 예에서 57 및 49의 가중치 값은 100까지 추가하고 특정 레코드가 사용되는 백분율로 가중성을 나타냅니다.	# ipa dnsrecord-add idm.example.com _ldap._tcp --srv- rec="0 51 389 server1.idm.example.com." # IPA dnsrecord-add server.idm.example.com _ldap._tcp --srv-rec 49 389 server2.idm.example.com".

옵션	설명	예제
--srv-priority=number	레코드의 우선 순위를 설정합니다. 서비스 유형에 대한 SRV 레코드가 여러 개 있을 수 있습니다. 우선 순위 (0 - 65535)는 레코드의 순위를 설정합니다. 숫자가 작을수록 우선 순위가 높습니다. 서비스는 우선 순위가 가장 높은 레코드를 먼저 사용해야 합니다.	# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec 49 389 server2.idm.example.com." --srv-priority=0
--srv-weight=number	레코드의 가중치를 설정합니다. 이렇게 하면 동일한 우선 순위의 SRV 레코드 순서를 결정하는 데 도움이 됩니다. 설정 가중치는 특정 레코드가 사용되는 가능성(백분율)을 나타내는 최대 100개까지 추가해야 합니다.	# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec 49 389 server2.idm.example.com." --srv-weight=60
--srv-port=number	대상 호스트에서 서비스에 대한 포트를 제공합니다.	# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec 60 389 server2.idm.example.com." --srv-port=636
--srv-target=string	대상 호스트의 도메인 이름을 제공합니다. 도메인에서 서비스를 사용할 수 없는 경우 단일 마침표(.)일 수 있습니다.	

추가 리소스

•

ipa dnsrecord-add 및 **IdM**에서 지원하는 **DNS** 레코드 유형을 사용하는 방법에 대한 자세한 내용을 보려면 **ipa dnsrecord-add --help** 명령을 실행합니다.

83.3. ANSIBLE을 사용하여 IDM에 A 및 AAAA DNS 레코드가 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 특정 **IdM** 호스트에 대한 **A** 및 **AAAA** 레코드가 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **IdM** 관리자는 **idm.example.com DNS** 영역에 **host1**에 대한 **A** 및 **AAAA** 레코드가 있는지 확인합니다.

사전 요구 사항

•

Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

•

IdM 관리자 암호를 알고 있습니다.

- **idm.example.com** 영역이 존재하며 **IdM DNS**에 의해 관리됩니다. **IdM DNS**에서 기본 **DNS** 영역 추가에 대한 자세한 내용은 **Ansible 플레이북을 사용하여 IdM DNS 영역 관리에 대한 내용**을 참조하십시오.

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **ensure-A-and-AAAA-records-are-present.yml** **Ansible** 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-A-and-AAAA-records-are-present.yml ensure-A-and-AAAA-records-are-present-copy.yml
```

4. 편집을 위해 **ensure-A-and-AAAA-records-are-present-copy.yml** 파일을 엽니다.

5. **ipadnsrecord** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **zone_name** 변수를 **idm.example.com** 으로 설정합니다.
- **records** 변수에서 **name** 변수를 **host1** 로 설정하고 **a_ip_address** 변수를 **192.168.122.123** 으로 설정합니다.
- **records** 변수에서 **name** 변수를 **host1** 로 설정하고 **aaaa_ip_address** 변수를 **::1** 로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Ensure A and AAAA records are present
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
    # Ensure A and AAAA records are present
    - name: Ensure that 'host1' has A and AAAA records.
      ipadnsrecord:
        ipaadmin_password: Secret123
        zone_name: idm.example.com
        records:
          - name: host1
            a_ip_address: 192.168.122.123
          - name: host1
            aaaa_ip_address: ::1
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-A-and-AAAA-records-are-present-copy.yml
```

추가 리소스

- **A 및 AAAA 레코드에 대한 자세한 내용은 [IdM의 DNS 레코드를 참조하십시오.](#)**
- **`/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 `README-dnsrecord.md` Markdown 파일에서 `ansible-freeipa ipadnsrecord` 모듈에 대한 샘플 **Ansible** 플레이북을 확인할 수 있습니다. 파일에는 `ipadnsrecord` 변수의 정의도 포함되어 있습니다.**
- **`/usr/share/doc/ansible-freeipa/playbooks/dns record` 디렉터리에서 `ipadns record` 모듈에 대한 샘플 **Ansible** 플레이북을 볼 수 있습니다.**

83.4. ANSIBLE을 사용하여 IDM에 A 및 PTR DNS 레코드가 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 특정 **IdM** 호스트의 **A** 레코드가 해당 **PTR** 레코드가 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **IdM**

관리자는 IP 주소가 192.168.122.45 인 host1 에 대해 **idm.example.com** 영역에 **A** 및 **PTR** 레코드가 있는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.
- **idm.example.com** DNS 영역이 존재하며 **IdM DNS**에서 관리합니다. **IdM DNS**에서 기본 DNS 영역 추가에 대한 자세한 내용은 **Ansible** 플레이북을 사용하여 **IdM DNS** 영역 관리에 대한 내용을 참조하십시오.

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **ensure-dnsrecord-with-reverse-is-present.yml** **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-dnsrecord-with-reverse-is-present.yml ensure-dnsrecord-with-reverse-is-present-copy.yml
```

4. 편집을 위해 **ensure-dnsrecord-with-reverse-is-present-copy.yml** 파일을 엽니다.

5. **ipadnsrecord** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.

- **`name` 변수를 `host 1` 로 설정합니다.**
- **`zone_name` 변수를 `idm.example.com` 으로 설정합니다.**
- **`ip_address` 변수를 `192.168.122.45` 로 설정합니다.**
- **`create_reverse` 변수를 `yes` 로 설정합니다.**

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Ensure DNS Record is present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
    # Ensure that dns record is present
    - ipadsrecord:
        ipaadmin_password: Secret123
        name: host1
        zone_name: idm.example.com
        ip_address: 192.168.122.45
        create_reverse: yes
        state: present
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-dnsrecord-with-reverse-is-present-copy.yml
```

추가 리소스

- **A 및 PTR DNS 레코드에 대한 자세한 내용은 [IdM의 DNS 레코드](#)를 참조하십시오.**

`/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README-dnsrecord.md** Markdown 파일에서 **ansible-freeipa ipadsnsrecord** 모듈에 대한 샘플 **Ansible** 플레이북을 확인할 수 있습니다. 파일에는 **ipadsnsrecord** 변수의 정의도 포함되어 있습니다.

•

`/usr/share/doc/ansible-freeipa/playbooks/dns record` 디렉터리에서 **ipadsns record** 모듈에 대한 샘플 **Ansible** 플레이북을 볼 수 있습니다.

83.5. ANSIBLE을 사용하여 IDM에 여러 DNS 레코드가 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 여러 값이 특정 **IdM DNS** 레코드와 연결되어 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **IdM** 관리자는 **idm.example.com** **DNS** 영역에 **host1**에 대해 여러 **A** 레코드가 있는지 확인합니다.

사전 요구 사항

•

Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

•

IdM 관리자 암호를 알고 있습니다.

•

idm.example.com 영역이 존재하며 **IdM DNS**에 의해 관리됩니다. **IdM DNS**에서 기본 **DNS** 영역 추가에 대한 자세한 내용은 **Ansible** 플레이북을 사용하여 **IdM DNS** 영역 관리에 대한 내용을 참조하십시오.

절차

1.

`/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2.

인벤토리 파일을 열고 구성할 **IdM** 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3.

ensure-presence-multiple-records.yml **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-presence-multiple-records.yml ensure-presence-multiple-records-copy.yml
```

4.

편집을 위해 **ensure-presence-multiple-records-copy.yml** 파일을 엽니다.

5.

ipadnsrecord 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 IdM 관리자 암호로 설정합니다.
- **records** 섹션에서 **name** 변수를 **host1** 로 설정합니다.
- **records** 섹션에서 **zone_name** 변수를 **idm.example.com** 으로 설정합니다.
- **records** 섹션에서 **a_rec** 변수를 **192.168.122.112**로 설정하고 **192.168.122. 122** 로 설정합니다.
- **records** 섹션에 두 번째 레코드 를 정의합니다.
 - **name** 변수를 **host 1** 로 설정합니다.
 - **zone_name** 변수를 **idm.example.com** 으로 설정합니다.
 - **aaaa_rec** 변수를 **::1** 로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Test multiple DNS Records are present.
  hosts: ipaserver
  become: true
  gather_facts: false
```



```
tasks:
# Ensure that multiple dns records are present
- ipadnsrecord:
  ipadmin_password: Secret123
  records:
    - name: host1
      zone_name: idm.example.com
      a_rec: 192.168.122.112
      a_rec: 192.168.122.122
    - name: host1
      zone_name: idm.example.com
      aaaa_rec: ::1
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-presence-multiple-records-copy.yml
```

추가 리소스

- DNS의 A 레코드에 대한 자세한 내용은 [IdM의 DNS 레코드를 참조하십시오](#).
- `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 `README-dnsrecord.md` Markdown 파일에서 `ansible-freeipa ipadnsrecord` 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 `ipadnsrecord` 변수의 정의도 포함되어 있습니다.
- `/usr/share/doc/ansible-freeipa/playbooks/dns record` 디렉터리에서 `ipadns record` 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.

83.6. ANSIBLE을 사용하여 IDM에 여러 CNAME 레코드가 있는지 확인

CNAME 레코드(CNAME 레코드)는 하나의 도메인 이름, 별칭을 다른 이름인 정식 이름에 매핑하는 DNS(Domain Name System)의 리소스 레코드 유형입니다.

CNAME 레코드는 단일 IP 주소에서 여러 서비스를 실행할 때 유용합니다(예: FTP 서비스 및 웹 서비스는 각각 다른 포트에서 실행됨).

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 여러 CNAME 레코드가 IdM DNS에 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 `host03` 은 HTTP

서버와 FTP 서버 둘 다입니다. IdM 관리자는 `idm.example.com` 영역에 `host03 A` 레코드에 대한 `www` 및 `ftp CNAME` 레코드가 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 `ansible-freeipa` 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- `idm.example.com` 영역이 존재하며 IdM DNS에 의해 관리됩니다. IdM DNS에서 기본 DNS 영역 추가에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 IdM DNS 영역 관리에 대한 내용](#)을 참조하십시오.
- `host03 A` 레코드는 `idm.example.com` 영역에 있습니다.

절차

1. `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 `[ipaserver]` 섹션에 나열되어 있는지 확인합니다. 예를 들어 `server.idm.example.com`을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. `ensure-CNAME-record-is-present.yml` Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-CNAME-record-is-present.yml ensure-CNAME-record-is-present-copy.yml
```

4. 편집을 위해 `ensure-CNAME-record-is-present-copy.yml` 파일을 엽니다.
5. `ipadnsrecord` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- (선택 사항) 플레이 이름에 제공된 설명을 설명합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **zone_name** 변수를 **idm.example.com** 으로 설정합니다.
- **records** 변수 섹션에서 다음 변수와 값을 설정합니다.
 - **name** 변수를 **www** 로 설정합니다.
 - **cname_hostname** 변수를 **host03** 으로 설정합니다.
 - **name** 변수를 **ftp** 로 설정합니다.
 - **cname_hostname** 변수를 **host03** 으로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Ensure that 'www.idm.example.com' and 'ftp.idm.example.com' CNAME
  records point to 'host03.idm.example.com'.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  - ipadsrecord:
    ipaadmin_password: Secret123
    zone_name: idm.example.com
    records:
    - name: www
      cname_hostname: host03
    - name: ftp
      cname_hostname: host03
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-CNAME-record-is-present.yml
```

추가 리소스

- `/usr/share/doc/ansible-freeipa/` 디렉터리에 있는 **README-dnsrecord.md** Markdown 파일에서 **ansible-freeipa ipadnsrecord** 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 **ipadnsrecord** 변수의 정의도 포함되어 있습니다.
- `/usr/share/doc/ansible-freeipa/playbooks/dns record` 디렉터리에서 **ipadns record** 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.

83.7. ANSIBLE을 사용하여 IDM에 SRV 레코드가 있는지 확인

SRV(DNS 서비스) 레코드는 도메인에서 사용할 수 있는 서비스의 호스트 이름, 포트 번호, 전송 프로토콜, 우선 순위 및 가중치를 정의합니다. **IdM(Identity Management)**에서는 **SRV** 레코드를 사용하여 **IdM** 서버 및 복제본을 찾을 수 있습니다.

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **SRV** 레코드가 **IdM DNS**에 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **IdM** 관리자는 값이 **10 50 88 idm.example.com**인 **_kerberos._udp.idm.example.com** **SRV** 레코드가 있는지 확인합니다. 이렇게 하면 다음 값이 설정됩니다.

- 서비스의 우선 순위를 **10**으로 설정합니다.
- 서비스의 **weight**를 **50**으로 설정합니다.
- 서비스에서 사용할 포트를 **88**으로 설정합니다.

사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

- **IdM 관리자 암호를 알고 있습니다.**
- **idm.example.com 영역이 존재하며 IdM DNS에 의해 관리됩니다. IdM DNS에서 기본 DNS 영역 추가에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 IdM DNS 영역 관리에 대한 내용](#)을 참조하십시오.**

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord 디렉터리로 이동합니다.**

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. **인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.**

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

3. **ensure-SRV-record-is-present.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.**

```
$ cp ensure-SRV-record-is-present.yml ensure-SRV-record-is-present-copy.yml
```

4. **편집을 위해 ensure-SRV-record-is-present-copy.yml 파일을 엽니다.**
5. **ipadnsrecord 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.**

- **ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.**
- **name 변수를 _kerberos._udp.idm.example.com 으로 설정합니다.**
- **srv_rec 변수를 '10 50 88 idm.example.com' 으로 설정합니다.**

•

zone_name 변수를 **idm.example.com** 으로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Test multiple DNS Records are present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
    # Ensure a SRV record is present
    - ipadnsrecord:
        ipaadmin_password: Secret123
        name: _kerberos._udp.idm.example.com
        srv_rec: '10 50 88 idm.example.com'
        zone_name: idm.example.com
        state: present
```

6.

파일을 저장합니다.

7.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-SRV-record-is-present.yml
```

추가 리소스

•

SRV 레코드에 대한 자세한 내용은 **IdM의 DNS 레코드를** 참조하십시오.

•

/usr/share/doc/ansible-freeipa/ 디렉터리에 있는 **README-dnsrecord.md** Markdown 파일에서 **ansible-freeipa ipadnsrecord** 모듈에 대한 샘플 **Ansible** 플레이북을 확인할 수 있습니다. 파일에는 **ipadnsrecord** 변수의 정의도 포함되어 있습니다.

•

/usr/share/doc/ansible-freeipa/playbooks/dns record 디렉터리에서 **ipadns record** 모듈에 대한 샘플 **Ansible** 플레이북을 볼 수 있습니다.

84장. ANSIBLE을 사용하여 IDM 서버 관리

Red Hat Ansible Engine 을 사용하여 **IdM(Identity Management)** 토폴로지에서 서버를 관리할 수 있습니다. **ansible-freeipa** 패키지에서 **server** 모듈을 사용하여 **IdM** 토폴로지에 서버가 있거나 없는지 확인할 수 있습니다. 복제본을 숨기거나 복제본을 볼 수도 있습니다.

섹션에는 다음 항목이 포함되어 있습니다.

- [Ansible을 사용하여 IdM 서버가 있는지 확인](#)
- [Ansible을 사용하여 IdM 토폴로지에 IdM 서버가 없는지 확인](#)
- [마지막 IdM 서버 역할을 호스팅해도 IdM 서버가 없는지 확인](#)
- [IdM 서버가 없지만 다른 IdM 서버와 반드시 연결이 끊어지지 않도록 확인](#)
- [Ansible 플레이북을 사용하여 기존 IdM 서버가 숨겨진지 확인](#)
- [Ansible 플레이북을 사용하여 기존 IdM 서버를 볼 수 있는지 확인](#)
- [기존 IdM 서버에 IdM DNS 위치가 할당되었는지 확인](#)
- [기존 IdM 서버에 IdM DNS 위치가 할당되지 않았는지 확인](#)

84.1. ANSIBLE을 사용하여 IDM 서버가 있는지 확인

Ansible 플레이북에서 **ipaserver ansible-freeipa** 모듈을 사용하여 **IdM(Identity Management)** 서버가 있는지 확인할 수 있습니다.



참고

ipaserver Ansible 모듈에서는 **IdM** 서버를 설치하지 않습니다.

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 참조하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)으로 **Ansible** 인벤토리 파일을 만들었습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 올바르게 작동합니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

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

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

3. 편집을 위해 **server-present-copy.yml** 파일을 엽니다.

4.

ipaserver 작업 섹션에서 다음 변수를 설정하여 파일을 수정하고 파일을 저장합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 서버의 **FQDN** 으로 설정합니다. 예제 서버의 **FQDN** 은 **server123.idm.example.com** 입니다.

```
---
- name: Server present example
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure server server123.idm.example.com is present
      ipaserver:
        ipaadmin_password: Secret123
        name: server123.idm.example.com
```

5.

Ansible 플레이북을 실행하고 플레이북 파일 및 인벤토리 파일을 지정합니다.

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

추가 리소스

- **Ansible**을 사용하여 **IdM** 서버를 설치하는 방법에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 ID 관리 서버 설치](#)를 참조하십시오.
- **ipaserver** 모듈을 사용하는 더 많은 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
 - **/usr/share/doc/ansible-freeipa/** 디렉토리의 **README- server.md** 파일. 이 파일에는 **ipaserver** 변수 정의도 포함되어 있습니다.
 - **/usr/share/doc/ansible-freeipa/playbooks/server** 디렉토리.

84.2. ANSIBLE을 사용하여 IDM 토폴로지에 IDM 서버가 없는지 확인

Ansible 플레이북을 사용하여 호스트로도 **IdM** 토폴로지에 **IdM(Identity Management)** 서버가 없는지 확인합니다.

ansible-freeipa ipaserver 역할과 달리 이 플레이북에서 사용된 **ipaserver** 모듈은 서버에서 **IdM** 서비스를 제거하지 않습니다.

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 참조하고 수정할 수 있는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)으로 **Ansible** 인벤토리 파일을 만들었습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 올바르게 작동합니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/server/` 디렉터리에 있는 **server-absent.yml** **Ansible** 플레이북 파일을 복사합니다.

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

3. 편집을 위해 **server-absent-copy.yml** 파일을 엽니다.

4.

ipaserver 작업 섹션에서 다음 변수를 설정하여 파일을 수정하고 파일을 저장합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 서버의 **FQDN** 으로 설정합니다. 예제 서버의 **FQDN** 은 **server123.idm.example.com** 입니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.

```
---
- name: Server absent example
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure server server123.idm.example.com is absent
      ipaserver:
        ipaadmin_password: Secret123
        name: server123.idm.example.com
        state: absent
```

5.

Ansible 플레이북을 실행하고 플레이북 파일 및 인벤토리 파일을 지정합니다.

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

6.

server 123.idm.example.com을 가리키는 모든 이름 서버 (**NS**) **DNS** 레코드가 **DNS** 영역에서 삭제되었는지 확인합니다. 이는 **IdM** 또는 외부 **DNS**에서 관리하는 통합 **DNS**를 사용하든 상관없이 적용됩니다.

추가 리소스

- **IdM** 서버를 제거하는 전체 절차는 **IdM** 서버 [제거](#)를 참조하십시오.
- **ipaserver** 모듈을 사용하는 더 많은 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
 - **/usr/share/doc/ansible-freeipa/ 디렉토리**의 **README- server.md** 파일. 이 파일에는 **ipaserver** 변수 정의도 포함되어 있습니다.

-

`/usr/share/doc/ansible-freeipa/playbooks/server` 디렉토리.

84.3. 마지막 IDM 서버 역할을 호스팅해도 IDM 서버가 없는지 확인

마지막 **IdM** 서비스 인스턴스가 서버에서 실행 중이더라도 **Ansible**을 사용하여 **IdM(Identity Management)** 서버가 없는지 확인할 수 있습니다. **CA**(인증 기관), **KRA**(키 복구 기관) 또는 **DNS** 서버는 모두 **IdM** 서비스의 예입니다.



주의

CA, **KRA** 또는 **DNS** 서버 역할을 하는 마지막 서버를 제거하면 **IdM** 기능이 크게 중단됩니다. `ipa service-find` 명령을 사용하여 **IdM** 서버에서 실행 중인 서비스를 수동으로 확인할 수 있습니다. **CA** 서버의 기본 이름은 `dogtag/server_name/REALM_NAME` 입니다.

ansible-freeipa ipaserver 역할과 달리 이 플레이북에서 사용된 **ipaserver** 모듈은 서버에서 **IdM** 서비스를 제거하지 않습니다.

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 참조하고 수정할 수 있는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)으로 **Ansible** 인벤토리 파일을 만들었습니다.

-

제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 올바르게 작동합니

다.

절차

1. *~/MyPlaybooks/ 디렉터리로 이동합니다.*

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

2. */usr/share/doc/ansible-freeipa/playbooks/server/ 디렉터리에 있는 **server-absent-ignore-last-of-role.yml** Ansible 플레이북 파일을 복사합니다.*

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-absent-ignore-last-of-role.yml server-absent-ignore-last-of-role-copy.yml
```

3. *편집을 위해 **server-absent-ignore-last-of-role-copy.yml** 파일을 엽니다.*

4. ***ipaserver** 작업 섹션에서 다음 변수를 설정하여 파일을 수정하고 파일을 저장합니다.*

- ***ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.*
- ***name** 변수를 서버의 **FQDN** 으로 설정합니다. 예제 서버의 **FQDN** 은 **server123.idm.example.com** 입니다.*
- ***ignore_last_of_role** 변수가 **yes** 로 설정되어 있는지 확인합니다.*
- ***state** 변수를 **absent** 로 설정합니다.*

```
---
- name: Server absent with last of role skip example
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure server "server123.idm.example.com" is absent with last of role skip
      ipaserver:
        ipaadmin_password: Secret123
        name: server123.idm.example.com
        ignore_last_of_role: yes
        state: absent
```

5.

Ansible 플레이북을 실행하고 플레이북 파일 및 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i inventory server-absent-ignore-last-of-role-copy.yml
```

6.

server 123.idm.example.com을 가리키는 모든 이름 서버 (NS) DNS 레코드가 DNS 영역에서 삭제되었는지 확인합니다. 이는 IdM 또는 외부 DNS에서 관리하는 통합 DNS를 사용하든 상관없이 적용됩니다.

추가 리소스

•

IdM 서버를 제거하는 전체 절차는 **IdM 서버 제거**를 참조하십시오.

•

ipaserver 모듈을 사용하는 더 많은 샘플 **Ansible** 플레이북은 다음을 참조하십시오.

◦

/usr/share/doc/ansible-freeipa/ 디렉토리의 **README- server.md** 파일. 이 파일에는 **ipaserver** 변수 정의도 포함되어 있습니다.

◦

/usr/share/doc/ansible-freeipa/playbooks/server 디렉토리.

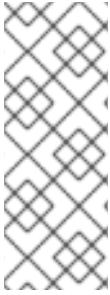
84.4. IDM 서버가 없지만 다른 IDM 서버와 반드시 연결이 끊어지지 않도록 확인

토폴로지에서 IdM(Identity Management) 서버를 제거하는 경우 **Ansible** 플레이북을 사용하여 해당 복제 계약을 그대로 유지할 수 있습니다. 또한 플레이북은 호스트로서도 IdM 서버가 IdM에 존재하지 않는지 확인합니다.

중요

삭제할 때 서버의 복제 계약을 무시하는 것은 다른 서버가 제대로 작동하지 않는 서버인 경우에만 무시하는 것이 좋습니다. 토폴로지의 중심점 역할을 하는 서버를 제거하면 토폴로지를 두 개의 연결이 끊긴 클러스터로 분할할 수 있습니다.

ipa server-del 명령을 사용하여 토폴로지에서 비기능 서버를 제거할 수 있습니다.



참고

CA(인증 기관), **KRA**(키 복구 기관) 또는 **DNS** 서버 역할을 하는 마지막 서버를 제거하면 **IdM**(Identity Management) 기능이 심각하게 중단됩니다. 이 문제를 방지하기 위해 플레이북은 **CA**, **KRA** 또는 **DNS** 서버 역할을 하는 서버를 제거하기 전에 이러한 서비스가 도메인의 다른 서버에서 실행되고 있는지 확인합니다.

ansible-freeipa ipaserver 역할과 달리 이 플레이북에서 사용된 **ipaserver** 모듈은 서버에서 **IdM** 서비스를 제거하지 않습니다.

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 참조하고 수정할 수 있는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)으로 **Ansible** 인벤토리 파일을 만들었습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 올바르게 작동합니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/server/` 디렉터리에 있는 **server-absent-ignore_topology_disconnect.yml** **Ansible** 플레이북 파일을 복사합니다.

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

3. 편집을 위해 **server-absent-ignore_topology_disconnect-copy.yml** 파일을 엽니다.

4. **ipaserver** 작업 섹션에서 다음 변수를 설정하여 파일을 수정하고 파일을 저장합니다.

- **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.
- **name** 변수를 서버의 FQDN 으로 설정합니다. 예제 서버의 FQDN 은 **server123.idm.example.com** 입니다.
- **ignore_topology_disconnect** 변수가 **yes** 로 설정되어 있는지 확인합니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.

```
---
- name: Server absent with ignoring topology disconnects example
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure server "server123.idm.example.com" with ignoring topology
      disconnects
      ipaserver:
        ipaadmin_password: Secret123
        name: server123.idm.example.com
        ignore_topology_disconnect: yes
        state: absent
```

5. Ansible 플레이북을 실행하고 플레이북 파일 및 인벤토리 파일을 지정합니다.

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

6. [선택 사항] **server 123.idm.example.com**을 가리키는 모든 이름 서버 (NS) DNS 레코드가 DNS 영역에서 삭제되었는지 확인합니다. 이는 IdM 또는 외부 DNS에서 관리하는 통합 DNS를 사용하는 상관없이 적용됩니다.

- **IdM 서버를 제거하는 전체 절차는 IdM 서버 제거를 참조하십시오.**
- **ipaserver 모듈을 사용하는 더 많은 샘플 Ansible 플레이북은 다음을 참조하십시오.**
 - **/usr/share/doc/ansible-freeipa/ 디렉토리의 README- server.md 파일. 이 파일에는 ipaserver 변수 정의도 포함되어 있습니다.**
 - **/usr/share/doc/ansible-freeipa/playbooks/server 디렉토리.**

84.5. ANSIBLE 플레이북을 사용하여 기존 IDM 서버가 숨겨진지 확인

Ansible 플레이북에서 ipaserver ansible-freeipa 모듈을 사용하여 기존 IdM(Identity Management) 서버를 숨깁니다. 이 플레이북은 IdM 서버를 설치하지 않습니다.

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 참조하고 수정할 수 있는 IdM 서버의 FQDN(정규화된 도메인 이름)으로 Ansible 인벤토리 파일을 만들었습니다.**
 - **제어 노드에서 인벤토리 파일에 정의된 IdM 서버로의 SSH 연결이 올바르게 작동합니다.**

절차

1.

~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2.

*/usr/share/doc/ansible-freeipa/playbooks/server/ 디렉터리에 있는 **server- hidden.yml** Ansible 플레이북 파일을 복사합니다.*

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-hidden.yml server-hidden-copy.yml
```

3.

*편집을 위해 **server-hidden-copy.yml** 파일을 엽니다.*

4.

***ipaserver** 작업 섹션에서 다음 변수를 설정하여 파일을 수정하고 파일을 저장합니다.*

- ***ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.*
- ***name** 변수를 서버의 **FQDN** 으로 설정합니다. 예제 서버의 **FQDN** 은 **server123.idm.example.com** 입니다.*
- *숨겨진 변수가 **True** 로 설정되어 있는지 확인합니다.*

```
---
- name: Server hidden example
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure server server123.idm.example.com is hidden
      ipaserver:
        ipaadmin_password: Secret123
        name: server123.idm.example.com
        hidden: True
```

5.

Ansible 플레이북을 실행하고 플레이북 파일 및 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i inventory server-hidden-copy.yml
```

추가 리소스

-

Ansible을 사용하여 **IdM** 서버를 설치하는 방법에 대한 자세한 내용은 **Ansible** 플레이북을 사용하여 **ID** 관리 서버 설치를 참조하십시오.

- 숨겨진 복제본에 대한 자세한 내용은 **숨겨진 복제본 모드**를 참조하십시오.
- **ipaserver** 모듈을 사용하는 더 많은 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
 - **/usr/share/doc/ansible-freeipa/** 디렉토리의 **README- server.md** 파일. 이 파일에는 **ipaserver** 변수 정의도 포함되어 있습니다.
 - **/usr/share/doc/ansible-freeipa/playbooks/server** 디렉토리.

84.6. ANSIBLE 플레이북을 사용하여 기존 IDM 서버를 볼 수 있는지 확인

Ansible 플레이북에서 **ipaserver ansible-freeipa** 모듈을 사용하여 기존 **IdM**(Identity Management) 서버가 표시되는지 확인합니다. 이 플레이북은 **IdM** 서버를 설치하지 않습니다.

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/** 디렉토리에서 참조하고 수정할 수 있는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)으로 **Ansible** 인벤토리 파일을 만들었습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 올바르게 작동합니다.

절차

1. *~/MyPlaybooks/ 디렉터리로 이동합니다.*

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

2. */usr/share/doc/ansible-freeipa/playbooks/server/ 디렉터리에 있는 server-not-hidden.yml Ansible 플레이북 파일을 복사합니다.*

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-not-hidden.yml server-not-hidden-copy.yml
```

3. *편집을 위해 server-not-hidden-copy.yml 파일을 엽니다.*

4. *ipaserver 작업 섹션에서 다음 변수를 설정하여 파일을 수정하고 파일을 저장합니다.*

- *ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.*
- *name 변수를 서버의 FQDN 으로 설정합니다. 예제 서버의 FQDN 은 server123.idm.example.com 입니다.*
- *숨겨진 변수가 no 로 설정되어 있는지 확인합니다.*

```
---
- name: Server not hidden example
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure server server123.idm.example.com is not hidden
      ipaserver:
        ipaadmin_password: Secret123
        name: server123.idm.example.com
        hidden: no
```

5. *Ansible 플레이북을 실행하고 플레이북 파일 및 인벤토리 파일을 지정합니다.*

```
$ ansible-playbook -v -i inventory server-not-hidden-copy.yml
```

- **Ansible**을 사용하여 **IdM** 서버를 설치하는 방법에 대한 자세한 내용은 **Ansible** 플레이북을 사용하여 **ID 관리 서버** 설치를 참조하십시오.
- 숨겨진 복제본에 대한 자세한 내용은 **숨겨진 복제본 모드**를 참조하십시오.
- **ipaserver** 모듈을 사용하는 더 많은 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
 - **/usr/share/doc/ansible-freeipa/** 디렉토리의 **README- server.md** 파일. 이 파일에는 **ipaserver** 변수 정의도 포함되어 있습니다.
 - **/usr/share/doc/ansible-freeipa/playbooks/server** 디렉토리.

84.7. 기존 IDM 서버에 IDM DNS 위치가 할당되었는지 확인

Ansible 플레이북에서 ipaserver ansible-freeipa 모듈을 사용하여 기존 IdM(Identity Management) 서버에 특정 IdM DNS 위치가 할당되었는지 확인합니다.

ipaserver Ansible 모듈은 **IdM** 서버를 설치하지 않습니다.

사전 요구 사항

- **IdM 관리자 암호를 알고 있습니다.**
- **IdM DNS 위치가 있습니다. 예제 위치는 *germany* 입니다.**
- **서버에 대한 루트 액세스 권한이 있습니다. 예제 *server*는 *server123.idm.example.com* 입니다.**
- **다음 요구 사항을 충족하는 *Ansible* 제어 노드를 구성했습니다.**
- ***Ansible* 버전 2.8 이상을 사용하고 있습니다.**

- **ansible-freeipa** 패키지를 설치했습니다.
- `~/MyPlaybooks/` 디렉터리에서 참조하고 수정할 수 있는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)으로 **Ansible 인벤토리 파일**을 만들었습니다.
- 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 올바르게 작동합니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/server/` 디렉터리에 있는 **server- location.yml** **Ansible** 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-location.yml server-location-copy.yml
```

3. 편집을 위해 **server-location-copy.yml** 파일을 엽니다.

4. **ipaserver** 작업 섹션에서 다음 변수를 설정하여 파일을 수정하고 파일을 저장합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 **server123.idm.example.com** 으로 설정합니다.
- 위치 변수를 **germany** 로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
```

```

- name: Server enabled example
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure server server123.idm.example.com with location "germany" is
      present
      ipaserver:
        ipaadmin_password: Secret123
        name: server123.idm.example.com
        location: germany

```

5.

Ansible 플레이북을 실행하고 플레이북 파일 및 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i inventory server-location-copy.yml
```

6.

SSH 를 사용하여 **server123.idm.example.com** 에 **root** 로 연결합니다.

```
ssh root@server123.idm.example.com
```

7.

서버에서 **named-pkcs11** 서비스를 재시작하여 업데이트를 즉시 적용합니다.

```
[root@server123.idm.example.com ~]# systemctl restart named-pkcs11
```

추가 리소스

- **Ansible**을 사용하여 **IdM** 서버를 설치하는 방법에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 ID 관리 서버 설치를 참조하십시오](#).
- **IdM DNS** 위치가 있는지 확인하는 방법에 대한 자세한 내용은 **IdM** 위치가 있는지 [확인하기](#) 위해 **Ansible** 사용을 참조하십시오.
- **ipaserver** 모듈을 사용하는 더 많은 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
 - **/usr/share/doc/ansible-freeipa/** 디렉토리의 **README- server.md** 파일. 이 파일에는 **ipaserver** 변수 정의도 포함되어 있습니다.
 - **/usr/share/doc/ansible-freeipa/playbooks/server** 디렉토리.

84.8. 기존 IDM 서버에 IDM DNS 위치가 할당되지 않았는지 확인

Ansible 플레이북에서 **ipaserver ansible-freeipa** 모듈을 사용하여 기존 **IdM(Identity Management)** 서버에 **IdM DNS** 위치가 할당되지 않았는지 확인합니다. 지리적 위치를 자주 변경하는 서버에 **DNS** 위치를 할당하지 마십시오. 플레이북은 **IdM** 서버를 설치하지 않습니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- 서버에 대한 루트 액세스 권한이 있습니다. 예제 **server**는 **server123.idm.example.com** 입니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
 - **Ansible** 버전 2.8 이상을 사용하고 있습니다.
 - **ansible-freeipa** 패키지를 설치했습니다.
 - **~/MyPlaybooks/** 디렉터리에서 참조하고 수정할 수 있는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)으로 **Ansible** 인벤토리 파일을 만들었습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 올바르게 작동합니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/server/** 디렉터리에 있는 **server-no-location.yml** **Ansible** 플레이북 파일을 복사합니다.


```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-no-location.yml server-no-location-copy.yml
```

3.

편집할 **server-no-location-copy.yml** 파일을 엽니다.

4.

ipaserver 작업 섹션에서 다음 변수를 설정하여 파일을 수정하고 파일을 저장합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 **server123.idm.example.com** 으로 설정합니다.
- 위치 변수가 "" 로 설정되어 있는지 확인합니다.

```
---
- name: Server no location example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure server server123.idm.example.com is present with no location
    ipaserver:
      ipaadmin_password: Secret123
      name: server123.idm.example.com
      location: ""
```

5.

Ansible 플레이북을 실행하고 플레이북 파일 및 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i inventory server-no-location-copy.yml
```

6.

SSH 를 사용하여 **server123.idm.example.com** 에 **root** 로 연결합니다.

```
ssh root@server123.idm.example.com
```

7.

서버에서 **named-pkcs11** 서비스를 재시작하여 업데이트를 즉시 적용합니다.

```
[root@server123.idm.example.com ~]# systemctl restart named-pkcs11
```

- **Ansible**을 사용하여 **IdM** 서버를 설치하는 방법에 대한 자세한 내용은 **Ansible** 플레이북을 사용하여 **ID 관리 서버** 설치를 참조하십시오.
- **IdM DNS** 위치의 개념 및 **IdM DNS** 위치가 없는지 확인하는 방법에 대한 자세한 내용은 **IdM**에서 **Ansible**을 사용하여 **DNS** 위치를 관리하는 방법을 참조하십시오.
- **ipaserver** 모듈을 사용하는 더 많은 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
 - `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-server.md** 파일. 이 파일에는 **ipaserver** 변수 정의도 포함되어 있습니다.
 - `/usr/share/doc/ansible-freeipa/playbooks/server` 디렉토리.

85장. IDM 상태 점검 정보 수집

Healthcheck는 **IdM(Identity Management)**에서 가능한 문제를 식별하는 데 도움이 되는 수동 명령줄 도구로 설계되었습니다.

이 장에서는 **30일** 순환된 상태 점검 출력을 기반으로 로그 컬렉션을 생성하는 방법을 설명합니다.

사전 요구 사항

- **Healthcheck** 도구는 **RHEL 8.1** 이상에서만 사용할 수 있습니다.

85.1. IDM의 상태 점검

IdM(Identity Management)의 상태 점검 도구는 **IdM** 환경의 상태에 영향을 줄 수 있는 문제를 찾는 데 도움이 됩니다.



참고

Healthcheck 도구는 **Kerberos** 인증 없이 사용할 수 있는 명령줄 도구입니다.

85.1.1. 모듈은 독립적입니다

Healthcheck는 다음을 테스트하는 독립 모듈로 구성됩니다.

- 복제 문제
- 인증서 유효
- 인증 기관 인프라 문제
- **IdM** 및 **Active Directory** 신뢰 문제

- 올바른 파일 권한 및 소유권 설정

85.1.2. 두 가지 출력 형식

Healthcheck는 **output -type** 옵션을 사용하여 설정할 수 있는 다음 출력을 생성합니다.

- **json: JSON** 형식의 머신에서 읽을 수 있는 출력 (기본값)
- **사람: 사람이 읽을 수 있는 출력**

output-file 옵션을 사용하여 다른 파일 대상을 지정할 수 있습니다.

85.1.3. 결과

각 **Healthcheck** 모듈은 다음 결과 중 하나를 반환합니다.

성공

예상대로 구성됨

경고

오류는 아니지만 계속 감시하거나 평가할 수 있습니다.

ERROR

예상대로 구성되지 않음

심각

제대로 구성되지 않았습니다. 영향을 미칠 가능성이 높습니다.

85.2. 로그 회전

로그 회전은 매일 새 로그 파일을 생성하고 파일은 날짜별로 구성됩니다. 로그 파일은 동일한 디렉터리에 저장되므로 날짜에 따라 특정 로그 파일을 선택할 수 있습니다.

순환은 최대 로그 파일 수가 구성되었음을 의미하며, 숫자가 초과된 경우 최신 파일이 다시 작성되고 가장 오래된 파일 이름을 바꿉니다. 예를 들어 순환 번호가 30인 경우 30초 로그 파일은 첫 번째(오래된) 로그 파일을 대체합니다.

로그 회전은 엄청난 로그 파일을 줄이고 이를 구성하므로 로그 분석에 도움이 될 수 있습니다.

85.3. IDM 상태 점검을 사용하여 로그 회전 구성

이 섹션에서는 다음을 사용하여 로그 회전을 구성하는 방법에 대해 설명합니다.

- **systemd 타이머**
- **crond 서비스**

systemd 타이머는 **Healthcheck** 도구를 정기적으로 실행하고 로그를 생성합니다. 기본값은 매일 4am으로 설정됩니다.

crond 서비스는 로그 회전에 사용됩니다.

기본 로그 이름은 **healthcheck.log** 이고 순환된 로그는 **healthcheck.log-YYYYMMDD** 형식을 사용합니다.

사전 요구 사항

- **root**로 명령을 실행해야 합니다.

절차

1. **systemd** 타이머를 활성화합니다.

```
# systemctl enable ipa-healthcheck.timer
Created symlink /etc/systemd/system/multi-user.target.wants/ipa-healthcheck.timer ->
/usr/lib/systemd/system/ipa-healthcheck.timer.
```

2.

systemd 타이머를 시작합니다.

```
# systemctl start ipa-healthcheck.timer
```

3.

/etc/logrotate.d/ipahealthcheck 파일을 열어 저장해야 하는 로그 수를 구성합니다.

기본적으로 로그 회전은 **30일** 동안 설정됩니다.

4.

/etc/logrotate.d/ipahealthcheck 파일에서 로그 경로를 구성합니다.

기본적으로 로그는 **/var/log/ipa/healthcheck/** 디렉터리에 저장됩니다.

5.

/etc/logrotate.d/ipahealthcheck 파일에서 로그 생성 시간을 구성합니다.

기본적으로 로그는 매일 오전 **4시**에 생성됩니다.

6.

로그 회전을 사용하려면 **crond** 서비스가 활성화되어 실행 중인지 확인합니다.

```
# systemctl enable crond
# systemctl start crond
```

로그를 생성하려면 **IPA** 상태 점검 서비스를 시작합니다.

```
# systemctl start ipa-healthcheck
```

결과를 확인하려면 **/var/log/ipa/healthcheck/** 로 이동하여 로그가 올바르게 생성되었는지 확인합니다.

86장. IDM 상태 점검을 사용하여 서비스 확인

이 섹션에서는 **Healthcheck** 툴을 사용하여 **IdM(Identity Management)** 서버에서 사용하는 모니터링 서비스에 대해 설명합니다.

자세한 내용은 **IdM의 상태 점검**을 참조하십시오.

사전 요구 사항

- **Healthcheck** 도구는 **RHEL 8.1** 이상에서만 사용할 수 있습니다.

86.1. 서비스 상태 점검 테스트

상태 점검 도구에는 **IdM** 서비스가 실행 중인지 확인하는 테스트가 포함되어 있습니다. 이 테스트는 실행 중이지 않은 서비스가 다른 테스트에서 실패할 수 있기 때문에 중요합니다. 따라서 모든 서비스가 먼저 실행되고 있는지 확인합니다. 그런 다음 다른 모든 테스트 결과를 확인할 수 있습니다.

모든 서비스 테스트를 보려면 **--list -sources** 옵션을 사용하여 **ipa-healthcheck** 를 실행합니다.

```
# ipa-healthcheck --list-sources
```

ipahealthcheck.meta.services 소스에서 **Healthcheck**로 테스트된 모든 서비스를 찾을 수 있습니다.

- **certmonger**
- **dirsrv**
- **gssproxy**
- **httpd**
- **ipa_custodia**

- **`ipa_dnssyncd`**
- **`ipa_otpd`**
- **`kadmin`**
- **`krb5kdc`**
- **명명됨**
- **`pki_tomcatd`**
- **`sssd`**



참고

문제를 발견하려고 할 때 모든 IdM 서버에서 이러한 테스트를 실행합니다.

86.2. 상태 점검을 사용하는 폴링 서비스

이 섹션에서는 상태 점검 도구를 사용하여 IdM(Identity Management) 서버에서 실행되는 서비스의 독립 실행형 수동 테스트에 대해 설명합니다.

Healthcheck 도구에는 다음을 사용하여 결과를 단축할 수 있는 많은 테스트가 포함되어 있습니다.

- 성공적인 모든 테스트 제외: **`--failures-only`**
- 서비스 테스트만 포함: **`--source=ipahealthcheck.meta.services`**

절차

-

경고, 오류 및 서비스와 관련된 중요한 문제를 사용하여 상태 점검을 실행하려면 다음을 입력합니다.

```
# ipa-healthcheck --source=ipahealthcheck.meta.services --failures-only
```

성공적인 테스트에 빈 괄호가 표시됩니다.

[]

서비스 중 하나가 실패하면 결과가 다음 예와 유사하게 표시됩니다.

```
{
  "source": "ipahealthcheck.meta.services",
  "check": "httpd",
  "result": "ERROR",
  "kw": {
    "status": false,
    "msg": "httpd: not running"
  }
}
```

추가 리소스

-

자세한 참조를 검토하려면 명령줄에 `man ipa-healthcheck` 를 입력합니다.

87장. IDM 상태 점검을 사용하여 IDM 및 AD 신뢰 구성 확인

이 섹션에서는 IdM(Identity Management)의 Healthcheck 툴을 이해하고 사용하여 IdM 및 Active Directory 신뢰 문제를 식별하는 데 도움이 됩니다.

사전 요구 사항

- Healthcheck 도구는 RHEL 8.1 이상에서만 사용할 수 있습니다.

87.1. IDM 및 AD 신뢰 상태 점검 테스트

Healthcheck 도구에는 IdM(Identity Management) 및 AD(Active Directory) 신뢰 상태를 테스트하기 위한 여러 테스트가 포함되어 있습니다.

모든 신뢰 테스트를 보려면 **--list -sources** 옵션을 사용하여 **ipa-healthcheck** 를 실행합니다.

```
# ipa-healthcheck --list-sources
```

ipahealthcheck.ipa.trust 소스에서 모든 테스트를 찾을 수 있습니다.

IPATrustAgentCheck

이 테스트에서는 시스템이 신뢰 에이전트로 구성된 경우 **SSSD** 구성을 확인합니다. **/etc/sss/sss.conf** 의 각 도메인에 대해 **id_provider=ipa** 는 **ipa_server_mode** 가 **True** 인지 확인합니다.

IPATrustDomainsCheck

이 테스트에서는 **sssctl** 도메인 목록의 도메인 목록을 **IPA** 도메인을 제외하고 **ipa trust-find** 의 도메인 목록과 비교하여 신뢰 도메인이 **SSSD** 도메인과 일치하는지 확인합니다.

IPATrustCatalogCheck

이 테스트에서는 **AD** 사용자 **Administrator@REALM** 이 해결됩니다. 그러면 **sssctl domain-status** 출력에서 **AD Global** 카탈로그 및 **AD** 도메인 컨트롤러 값이 채워집니다.

신뢰할 수 있는 각 도메인에 대해 **SID + 500(관리자)**의 **id**를 사용하여 사용자를 검색한 다음 **sssctl domain-status <domain> --active-server** 의 출력을 확인하여 도메인이 활성 상태인지 확인합니다.

IPAsidgenpluginCheck

이 테스트에서는 IPA 389-ds 인스턴스에서 **sidgen** 플러그인이 활성화되었는지 확인합니다. 또한 테스트에서는 **cn=plugins,cn=config** 의 **IPA SIDGEN** 및 **ipa-sidgen-task** 플러그인에 **nsslapd-pluginEnabled** 옵션이 포함되어 있는지 확인합니다.

IPATrustAgentMemberCheck

이 테스트에서는 현재 호스트가 **cn=adtrust** 에이전트인 **cn=sysaccounts,cn=etc,SUFFIX** 의 멤버인지 확인합니다.

IPATrustControllerPrincipalCheck

이 테스트에서는 현재 호스트가 **cn=adtrust** 에이전트인 **cn=sysaccounts,cn=etc,SUFFIX** 의 멤버인지 확인합니다.

IPATrustControllerServiceCheck

이 테스트에서는 **ipactl**에서 현재 호스트가 **ADTRUST** 서비스를 시작하는지 확인합니다.

IPATrustControllerConfCheck

이 테스트에서는 **net conf** 목록의 **passdb** 백엔드에 대해 **ldapi** 가 활성화되어 있는지 확인합니다.

IPATrustControllerGroupSIDCheck

이 테스트에서는 **admins** 그룹의 **SID**가 **512**(도메인 관리자 **RID**)로 끝나는지 확인합니다.

IPATrustPackageCheck

이 테스트에서는 신뢰 컨트롤러 및 **AD** 신뢰가 활성화되지 않은 경우 **trust-ad** 패키지가 설치되었는지 확인합니다.



참고

문제를 찾으려고 할 때 모든 **IdM** 서버에서 이러한 테스트를 실행합니다.

87.2. HEALTHCHECK 도구를 사용하여 신뢰를 건너뛰니다.

이 섹션에서는 **Healthcheck** 툴을 사용하여 **IdM**(Identity Management) 및 **AD**(Active Directory) 신뢰 상태 점검의 독립 실행형 수동 테스트에 대해 설명합니다.

Healthcheck 도구에는 많은 테스트가 포함되어 있으므로 다음과 같이 결과를 단축할 수 있습니다.

- 성공적인 모든 테스트 제외: **--failures-only**
- 신뢰 테스트만 포함: **--source=ipahealthcheck.ipa.trust**

절차

- 경고, 오류 및 중요한 문제로 인해 상태 점검을 실행하려면 다음을 입력합니다.

```
# ipa-healthcheck --source=ipahealthcheck.ipa.trust --failures-only
```

성공적인 테스트에 빈 괄호가 표시됩니다.

```
# ipa-healthcheck --source=ipahealthcheck.ipa.trust --failures-only
[]
```

추가 리소스

- 자세한 참조를 검토하려면 명령줄에 **man ipa-healthcheck** 를 입력합니다.

88장. IDM 상태 점검을 사용하여 인증서 확인

이 섹션에서는 **IdM(Identity Management)**의 **Healthcheck** 툴을 이해하고 사용하여 **certmonger**에서 유지 관리하는 **IPA** 인증서 문제를 식별하는 데 도움이 됩니다.

자세한 내용은 **IdM의 상태 점검**을 참조하십시오.

사전 요구 사항

- **Healthcheck** 도구는 **RHEL 8.1** 이상에서만 사용할 수 있습니다.

88.1. IDM 인증서 상태 점검 테스트

Healthcheck 도구에는 **IdM(Identity Management)**에서 **certmonger**가 유지 관리하는 인증서 상태를 확인하기 위한 여러 테스트가 포함되어 있습니다. **certmonger**에 대한 자세한 내용은 **certmonger를 사용하여 서비스의 IdM 인증서 가져오기를 참조하십시오**.

이 테스트 제품군은 만료, 검증, 신뢰 및 기타 문제를 확인합니다. 동일한 기본 문제에 대해 여러 오류가 발생할 수 있습니다.

모든 인증서 테스트를 보려면 **--list -sources** 옵션을 사용하여 **ipa-healthcheck** 를 실행합니다.

```
# ipa-healthcheck --list-sources
```

ipahealthcheck.ipa.certs 소스에서 모든 테스트를 찾을 수 있습니다.

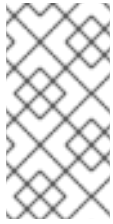
IPACertmongerExpirationCheck

이 테스트에서는 **certmonger** 의 만료를 확인합니다.

오류가 보고되면 인증서가 만료됩니다.

경고가 나타나면 인증서가 곧 만료됩니다. 기본적으로 이 테스트는 인증서 만료일 전 **28일** 이내에 적용됩니다.

`/etc/ipahealthcheck/ipahealthcheck.conf` 파일에서 일 수를 구성할 수 있습니다. 파일을 열면 **default** 섹션에 있는 **cert_expiration_days** 옵션을 변경합니다.

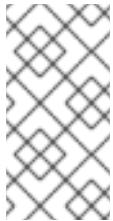


참고

certmonger는 인증서 만료의 자체 보기를 로드하고 유지합니다. 이 검사는 디스크에 있는 인증서의 유효성을 검사하지 않습니다.

IPACertfileExpirationCheck

이 테스트에서는 인증서 파일 또는 **NSS** 데이터베이스를 열 수 있는지 확인합니다. 이 테스트도 만료를 확인합니다. 따라서 오류 또는 경고 출력에서 **msg** 특성을 주의 깊게 읽습니다. 메시지는 문제를 지정합니다.



참고

이 테스트는 디스크상의 인증서를 확인합니다. 인증서가 없으면 읽을 수 없음 등 별도의 오류도 발생할 수 있습니다.

IPACertNSSTrust

이 테스트는 **NSS** 데이터베이스에 저장된 인증서의 신뢰성을 비교합니다. **NSS** 데이터베이스에서 예상되는 추적된 인증서의 경우 신뢰는 예상 값과 일치하지 않는 경우 발생한 오류와 비교됩니다.

IPANSSChainValidation

이 테스트에서는 **NSS** 인증서의 인증서 체인의 유효성을 검사합니다. 테스트 실행: **certutil -V -u V -e -d [dbdir] -n [nickname]**

IPAOpenSSLChainValidation

이 테스트는 **OpenSSL** 인증서의 인증서 체인의 유효성을 검사합니다. 여기에서 **NSS** 검증과 유사한 것은 **OpenSSL** 명령입니다.

```
openssl verify -verbose -show_chain -CAfile /etc/ipa/ca.crt [cert file]
```

IPARAAGent

이 테스트에서는 디스크의 인증서를 **uid=ipara,ou=People,o=ipaca**의 **LDAP**에 있는 동등한 레코드와 비교합니다.

IPACertRevocation

이 테스트에서는 **certmonger**를 사용하여 인증서가 취소되지 않았는지 확인합니다. 따라서 테스트는 **certmonger**에서만 유지 관리하는 인증서와 연결된 문제를 찾을 수 있습니다.

IPACertmongerCA

이 테스트는 **certmonger CA**(인증 기관) 구성을 확인합니다. **IdM**은 **CA** 없이 인증서를 발행할 수 없습니다.

certmonger는 일련의 **CA** 도우미를 유지 관리합니다. **IdM**에는 호스트 또는 서비스 인증서에 대해 호스트 또는 사용자 주체로 인증, **IdM**을 통한 인증서를 발행하는 **IPA**라는 **CA**가 있습니다.

CA 하위 시스템 인증서를 갱신 하는 **dogtag-ipa-ca-renew -agent** 및 **dogtag-ipa-ca-renew-agent-reuse**도 있습니다.



참고

문제를 확인하려고 할 때 모든 **IdM** 서버에서 이러한 테스트를 실행합니다.

88.2. 상태 점검 툴을 사용한 암호 인증서

이 섹션에서는 **Healthcheck** 툴을 사용한 **IdM**(Identity Management) 인증서 상태 점검의 독립 실행형 수동 테스트에 대해 설명합니다.

Healthcheck 도구에는 많은 테스트가 포함되어 있으므로 다음을 사용하여 결과를 단축할 수 있습니다.

- 성공적인 모든 테스트 제외: **--failures-only**
- 인증서 테스트만 포함: **--source=ipahealthcheck.ipa.certs**

사전 요구 사항

- 상태 점검 테스트를 **root** 사용자로 수행해야 합니다.

절차

- 경고, 오류 및 인증서와 관련된 중요한 문제를 사용하여 상태 점검을 실행하려면 다음을 입력

합니다.

```
# ipa-healthcheck --source=ipahealthcheck.ipa.certs --failures-only
```

성공적인 테스트에 빈 괄호가 표시됩니다.

```
[]
```

실패한 테스트는 다음 출력을 보여줍니다.

```
{
  "source": "ipahealthcheck.ipa.certs",
  "check": "IPACertfileExpirationCheck",
  "result": "ERROR",
  "kw": {
    "key": 1234,
    "dbdir": "/path/to/nssdb",
    "error": [error],
    "msg": "Unable to open NSS database '/path/to/nssdb': [error]"
  }
}
```

이 `IPACertfileExpirationCheck` 테스트는 `NSS` 데이터베이스를 열 때 실패했습니다.

추가 리소스

- 자세한 참조를 검토하려면 명령줄에 `man ipa-healthcheck` 를 입력합니다.

89장. IDM 상태 점검을 사용하여 시스템 인증서 확인

이 섹션에서는 시스템 인증서 문제를 식별하기 위한 **IdM(Identity Management)**의 상태 점검 틀에 대해 설명합니다.

자세한 내용은 **IdM의 상태 점검**을 참조하십시오.

사전 요구 사항

- **Healthcheck** 도구는 **RHEL 8.1** 이상에서만 사용할 수 있습니다.

89.1. 시스템 인증서 상태 점검 테스트

Healthcheck 도구에는 시스템 확인(**DogTag**) 인증서에 대한 여러 테스트가 포함되어 있습니다.

모든 테스트를 보려면 **--list -sources** 옵션을 사용하여 **ipa-healthcheck** 를 실행합니다.

```
# ipa-healthcheck --list-sources
```

ipahealthcheck.dogtag.ca 소스에서 모든 테스트를 찾을 수 있습니다.

DogtagCertsConfigCheck

이 테스트에서는 **NSS** 데이터베이스의 **CA(Certificate Authority)** 인증서를 **su.cfg** 에 저장된 동일한 값과 비교합니다. 일치하지 않으면 **CA**가 시작되지 않습니다.

특히, 다음을 확인합니다.

- **auditSigningCert cert-pki-ca against ca.audit_signing.cert**
- **ocspSigningCert cert-pki-ca against ca.ocsp_signing.cert**
- **caSigningCert cert-pki-ca against ca.signing.cert**

- **subsystemCert cert-pki-ca against ca.subsystem.cert**
- **ca.sslserver.cert에 대한 server-Cert cert-pki-ca**

KRA(키 복구 권한)가 설치된 경우:

- **transportCert cert-pki-kra against ca.connector.KRA.transportCert**

DogtagCertsConnectivityCheck

이 테스트에서는 연결을 확인합니다. 이 테스트는 다음을 확인하는 **ipa cert-show 1** 명령과 동일합니다.

- **Apache의 PKI 프록시 설정**
- **IdM에서 CA를 찾을 수 있음**
- **RA 에이전트 클라이언트 인증서**
- **요청에 대한 CA 응답의 정확성**

cert-show 를 실행하고 **CA**(인증서 또는 찾을 수 없음)에서 예상 결과를 가져올 수 있는지 확인하려고 하므로 테스트에서 **serial #1**을 사용하여 인증서를 점검합니다.



참고

문제를 찾으려고 할 때 모든 **IdM** 서버에서 이러한 테스트를 실행합니다.

89.2. HEALTHCHECK를 사용한 시스템 인증서

이 섹션에서는 **Healthcheck** 툴을 사용한 **IdM(Identity Management)** 인증서의 독립 실행형 수동 테스트에 대해 설명합니다.

상태 점검 도구에는 많은 테스트가 포함되어 있으므로 **DogTag** 테스트만 포함하여 결과를 줄 수 있습니다. **--source=ipahealthcheck.dogtag.ca**

절차

- **Healthcheck restricted**를 **DogTag** 인증서로 실행하려면 다음을 입력합니다.

```
# ipa-healthcheck --source=ipahealthcheck.dogtag.ca
```

성공적인 테스트의 예는 다음과 같습니다.

```
{
  "source: ipahealthcheck.dogtag.ca",
  "check: DogtagCertsConfigCheck",
  "result: SUCCESS",
  "uuid: 9b366200-9ec8-4bd9-bb5e-9a280c803a9c",
  "when: 20191008135826Z",
  "duration: 0.252280",
  "kw:" {
    "key": "Server-Cert cert-pki-ca",
    "configfile": "/var/lib/pki/pki-tomcat/conf/ca/CS.cfg"
  }
}
```

실패한 테스트의 예는 다음과 같습니다.

```
{
  "source: ipahealthcheck.dogtag.ca",
  "check: DogtagCertsConfigCheck",
  "result: CRITICAL",
  "uuid: 59d66200-1447-4b3b-be01-89810c803a98",
  "when: 20191008135912Z",
  "duration: 0.002022",
  "kw:" {
    "exception": "NSDB /etc/pki/pki-tomcat/alias not initialized",
  }
}
```

추가 리소스

- 자세한 참조를 검토하려면 명령줄에 **man ipa-healthcheck** 를 입력합니다.

90장. IDM 상태 점검을 사용하여 디스크 공간 확인

이 섹션에서는 **Healthcheck** 도구를 사용하여 ID 관리 서버의 여유 디스크 공간을 모니터링하는 방법을 설명합니다.

자세한 내용은 [IdM의 상태 점검](#)을 참조하십시오.

사전 요구 사항

- **Healthcheck** 도구는 **RHEL 8.1** 이상에서만 사용할 수 있습니다.

90.1. 디스크 공간 상태 점검 테스트

Healthcheck 도구에는 사용 가능한 디스크 공간을 확인하는 테스트가 포함되어 있습니다. 사용 가능한 디스크 공간이 충분하지 않으면 다음과 같은 문제가 발생할 수 있습니다.

- 로깅
- 실행
- 백업

테스트에서는 다음 경로를 확인합니다.

표 90.1. 테스트된 경로

테스트에서 확인한 경로	최소 디스크 공간(MB)
<code>/var/lib/dirsrv/</code>	1024
<code>/var/lib/ipa/backup/</code>	512
<code>/var/log/</code>	1024
<code>var/log/audit/</code>	512

테스트에서 확인한 경로	최소 디스크 공간(MB)
/var/tmp/	512
/tmp/	512

모든 테스트를 나열하려면 **--list -sources** 옵션을 사용하여 **ipa-healthcheck** 를 실행합니다.

```
# ipa-healthcheck --list-sources
```

파일 시스템 공간 점검 테스트는 **ipahealthcheck.system.filesystemspace** 소스 아래에 배치됩니다.

FileSystemSpaceCheck

이 테스트에서는 다음과 같은 방법으로 사용 가능한 디스크 공간을 확인합니다.

- 필요한 최소 원시 가용 바이트.
- 백분율 - **cinder** 최소 사용 가능한 디스크 공간이 20%로 하드 코딩됩니다.

90.2. 상태 점검 도구를 사용하여 디스크 공간

이 섹션에서는 상태 점검 도구를 사용하여 **IdM(Identity Management)** 서버에서 사용 가능한 디스크 공간을 독립 실행형 수동 테스트에 대해 설명합니다.

Healthcheck에는 많은 테스트가 포함되어 있으므로 다음을 통해 결과를 좁힐 수 있습니다.

- 성공적인 모든 테스트 제외: **--failures-only**
- 공간 점검 테스트만 포함: **--source=ipahealthcheck.system.filesystemspace**

절차

- 경고, 오류 및 사용 가능한 디스크 공간과 관련된 중요한 문제를 사용하여 상태 점검을 실행하

려면 다음을 입력합니다.

```
# ipa-healthcheck --source=ipahealthcheck.system.filesystemspace --failures-only
```

성공적인 테스트에 빈 괄호가 표시됩니다.

```
[]
```

예를 들어 실패한 테스트는 다음을 표시할 수 있습니다.

```
{
  "source": "ipahealthcheck.system.filesystemspace",
  "check": "FileSystemSpaceCheck",
  "result": "ERROR",
  "kw": {
    "msg": "/var/lib/dirsrv: free space under threshold: 0 MiB < 1024 MiB",
    "store": "/var/lib/dirsrv",
    "free_space": 0,
    "threshold": 1024
  }
}
```

실패한 테스트에서는 `/var/lib/dirsrv` 디렉터리에 공간이 부족했음을 알려줍니다.

추가 리소스

- 자세한 참조를 검토하려면 명령줄에 `man ipa-healthcheck` 를 입력합니다.

91장. HEALTHCHECK를 사용하여 IDM 구성 파일에 대한 권한 확인

이 섹션에서는 상태 점검 도구를 사용하여 **IdM(Identity Management)** 구성 파일을 테스트하는 방법을 설명합니다.

자세한 내용은 **IdM의 상태 점검**을 참조하십시오.

사전 요구 사항

- **Healthcheck** 도구는 **RHEL 8.1** 이상의 시스템에서만 사용할 수 있습니다.

91.1. 파일 권한 상태 점검 테스트

Healthcheck 툴은 **IdM(Identity Management)**에서 설치하거나 구성한 몇 가지 중요한 파일의 소유권 및 권한을 테스트합니다.

테스트된 파일의 소유권 또는 권한을 변경하면 테스트에서 결과 섹션에 경고를 반환합니다. 구성이 작동하지 않을 필요는 없지만 파일이 기본 구성과 다릅니다.

모든 테스트를 보려면 **--list -sources** 옵션을 사용하여 **ipa-healthcheck** 를 실행합니다.

```
# ipa-healthcheck --list-sources
```

파일 권한 테스트는 **ipahealthcheck.ipa.files** 소스 아래에 배치됩니다.

IPAFileNSSDBCheck

이 테스트는 **389-ds NSS** 데이터베이스와 **CA(인증 기관)** 데이터베이스를 확인합니다. **389-ds** 데이터베이스는 **/etc/dirsrv/slapd-**<dashed-REALM>****에 있으며 **CA** 데이터베이스는 **/etc/pki/pki-tomcat/alias/**에 있습니다.

IPAFileCheck

이 테스트에서는 다음 파일을 확인합니다.

- **/var/lib/ipa/ra-agent.{key|pem}**

- `/var/lib/ipa/certs/httpd.pem`
- `/var/lib/ipa/private/httpd.key`
- `/etc/httpd/alias/ipasession.key`
- `/etc/dirsrv/ds.keytab`
- `/etc/ipa/ca.crt`
- `/etc/ipa/custodia/server.keys`

PKINIT가 활성화된 경우:

- `/var/lib/ipa/certs/kdc.pem`
- `/var/lib/ipa/private/kdc.key`

DNS가 구성된 경우 다음을 수행합니다.

- `/etc/named.keytab`
- `/etc/ipa/dnssec/ipa-dnskeysyncd.keytab`

TomcatFileCheck

이 테스트에서는 **CA**가 구성된 경우 일부 **tomcat** 관련 파일을 확인합니다.

- `/etc/pki/pki-tomcat/password.conf`

- `/var/lib/pki/pki-tomcat/conf/ca/CS.cfg`
- `/etc/pki/pki-tomcat/server.xml`



참고

문제를 찾으려고 할 때 모든 **IdM** 서버에서 이러한 테스트를 실행합니다.

91.2. HEALTHCHECK를 사용하는 구성 파일

이 섹션에서는 상태 점검 도구를 사용하여 **IdM(Identity Management)** 서버의 구성 파일의 독립 실행형 수동 테스트에 대해 설명합니다.

Healthcheck 도구에는 많은 테스트가 포함되어 있습니다. 결과는 다음과 같이 줄힐 수 있습니다.

- 성공적인 모든 테스트 제외: `--failures-only`
- 소유권 및 권한 테스트만 포함: `--source=ipahealthcheck.ipa.files`

절차

1.

경고, 오류 및 심각한 문제만 표시하는 동시에 **IdM** 구성 파일 소유권 및 권한에 대해 상태 점검 테스트를 실행하려면 다음을 입력합니다.

```
# ipa-healthcheck --source=ipahealthcheck.ipa.files --failures-only
```

성공적인 테스트에 빈 괄호가 표시됩니다.

```
# ipa-healthcheck --source=ipahealthcheck.ipa.files --failures-only
[]
```

실패한 테스트에 다음 경고와 유사한 결과가 표시됩니다.

```
{
  "source": "ipahealthcheck.ipa.files",
  "check": "IPAFileNSSDBCheck",
  "result": "WARNING",
  "kw": {
    "key": "_etc_dirsrv_slapd-EXAMPLE-TEST_pkcs11.txt_mode",
    "path": "/etc/dirsrv/slapd-EXAMPLE-TEST/pkcs11.txt",
    "type": "mode",
    "expected": "0640",
    "got": "0666",
    "msg": "Permissions of /etc/dirsrv/slapd-EXAMPLE-TEST/pkcs11.txt are 0666 and should be 0640"
  }
}
```

추가 리소스

- 자세한 참조 자료를 검토하려면 명령줄에서 **man ipa-healthcheck** 를 엽니다.

92장. 상태 점검을 사용하여 IDM 복제 확인

이 섹션에서는 상태 점검 도구를 사용하여 IdM(Identity Management) 복제를 테스트하는 방법을 설명합니다.

자세한 내용은 [IdM의 상태 점검](#)을 참조하십시오.

사전 요구 사항

- **Healthcheck** 도구는 **RHEL 8.1** 이상에서만 사용할 수 있습니다.

92.1. 복제 상태 점검 테스트

Healthcheck 툴은 IdM(Identity Management) 토폴로지 구성을 테스트하고 복제 충돌 문제를 검색합니다.

모든 테스트를 나열하려면 **--list -sources** 옵션을 사용하여 **ipa-healthcheck** 를 실행합니다.

```
# ipa-healthcheck --list-sources
```

토폴로지 테스트는 **ipahealthcheck.ipa.topology** 및 **ipahealthcheck.ds.replication** 소스 아래에 배치됩니다.

IPATopologyDomainCheck

이 테스트는 다음을 확인합니다.

- 토폴로지 연결이 끊어지지 않고 모든 서버 간에 복제 경로가 있는지 여부.
- 서버에 권장되는 복제 계약 수 이상 없는 경우.

테스트에 실패하면 테스트에서 연결 오류 또는 너무 많은 복제 계약과 같은 오류를 반환합니다.

테스트가 성공하면 테스트에서 구성된 도메인을 반환합니다.



참고

테스트에서는 도메인 및 **ca** 접미사 모두에 대해 **ipa topologysuffix-verify** 명령을 실행합니다(이 서버에서 인증 기관이 구성되었다고 가정).

ReplicationConflictCheck

테스트에서 **LDAP** 일치 항목 검색(&!(objectclass=nstombstone))(nsds5ReplConflict=*).



참고

문제를 확인하려고 할 때 모든 **IdM** 서버에서 이러한 테스트를 실행합니다.

92.2. 상태 점검을 사용하여 복제 복제

이 섹션에서는 상태 점검 툴을 사용한 **IdM**(Identity Management) 복제 토폴로지 및 구성에 대한 독립 실행형 수동 테스트에 대해 설명합니다.

Healthcheck 도구에는 많은 테스트가 포함되어 있으므로 다음을 사용하여 결과를 단축할 수 있습니다.

- 복제 충돌 테스트: **--source=ipahealthcheck.ds.replication**
- 올바른 토폴로지 테스트: **--source=ipahealthcheck.ipa.topology**

사전 요구 사항

- 상태 점검 테스트를 **root** 사용자로 수행해야 합니다.

절차

- 상태 점검 복제 충돌 및 토폴로지 검사를 실행하려면 다음을 입력합니다.

```
# ipa-healthcheck --source=ipahealthcheck.ds.replication --
source=ipahealthcheck.ipa.topology
```

4가지 다른 결과가 가능합니다.

- **SUCCESS**-|테스트가 성공적으로 통과되었습니다.

```
{
  "source": "ipahealthcheck.ipa.topology",
  "check": "IPATopologyDomainCheck",
  "result": "SUCCESS",
  "kw": {
    "suffix": "domain"
  }
}
```

- 경고 메시지- 테스트가 통과되었지만 문제가 있을 수 있습니다.

- 오류 메시지- 테스트에 실패했습니다.

```
{
  "source": "ipahealthcheck.ipa.topology",
  "check": "IPATopologyDomainCheck",
  "result": "ERROR",
  "uuid": d6ce3332-92da-423d-9818-e79f49ed321f
  "when": 20191007115449Z
  "duration": 0.005943
  "kw": {
    "msg": "topologysuffix-verify domain failed, server2 is not connected
(server2_139664377356472 in MainThread)"
  }
}
```

- **CRITICAL**-|테스트에 실패했습니다. **IdM** 서버 기능에 영향을 줍니다.

추가 리소스

- 자세한 참조를 검토하려면 명령줄에서 **man ipa-healthcheck** 를 엽니다.

93장. IDM 상태 점검을 사용하여 DNS 레코드 확인

이 섹션에서는 **DNS 레코드** 문제를 식별하기 위한 **IdM(Identity Management)**의 상태 점검 툴에 대해 설명합니다.

사전 요구 사항

- **DNS 레코드 Healthcheck 툴은 RHEL 8.2 이상에서만 사용할 수 있습니다.**

93.1. DNS 레코드 상태 점검 테스트

Healthcheck 도구에는 자동 검색에 필요한 예상 **DNS 레코드**를 확인할 수 있는지 확인하는 테스트가 포함되어 있습니다.

모든 테스트를 나열하려면 **--list -sources** 옵션을 사용하여 **ipa-healthcheck** 를 실행합니다.

```
# ipa-healthcheck --list-sources
```

DNS 레코드 확인 테스트는 **ipahealthcheck.ipa.idns** 소스에 배치됩니다.

IPADNSSystemRecordsCheck

이 테스트에서는 **/etc/resolv.conf** 파일에 지정된 첫 번째 확인자를 사용하여 **ipa dns-update-system-records --dry-run** 명령에서 **DNS 레코드**를 확인합니다. **IPA 서버**에서 레코드가 테스트됩니다.

93.2. 상태 점검 툴을 사용하여 DNS 레코드 표시

이 섹션에서는 상태 점검 도구를 사용하여 **IdM(Identity Management)** 서버에서 **DNS 레코드**를 독립 실행형 수동 테스트에 대해 설명합니다.

Healthcheck 도구에는 많은 테스트가 포함되어 있습니다. **--source ipahealthcheck.ipa.idns** 옵션을 추가하여 **DNS 레코드 테스트**만 포함하여 결과를 줄일 수 있습니다.

사전 요구 사항

- 상태 점검 테스트를 **root** 사용자로 수행해야 합니다.

절차

- DNS 레코드 확인을 실행하려면 다음을 입력합니다.

```
# ipa-healthcheck --source ipahealthcheck.ipa.idns
```

레코드를 확인할 수 있는 경우 테스트에서 **SUCCESS** 를 결과적으로 반환합니다.

```
{
  "source": "ipahealthcheck.ipa.idns",
  "check": "IPADNSSystemRecordsCheck",
  "result": "SUCCESS",
  "uuid": "eb7a3b68-f6b2-4631-af01-798cac0eb018",
  "when": "20200415143339Z",
  "duration": "0.210471",
  "kw": {
    "key": "_ldap._tcp.idm.example.com.:server1.idm.example.com."
  }
}
```

예를 들어 레코드 수가 예상 수와 일치하지 않는 경우 테스트에서 **WARNING**을 반환합니다.

```
{
  "source": "ipahealthcheck.ipa.idns",
  "check": "IPADNSSystemRecordsCheck",
  "result": "WARNING",
  "uuid": "972b7782-1616-48e0-bd5c-49a80c257895",
  "when": "20200409100614Z",
  "duration": "0.203049",
  "kw": {
    "msg": "Got {count} ipa-ca A records, expected {expected}",
    "count": 2,
    "expected": 1
  }
}
```

추가 리소스

- 자세한 참조를 검토하려면 명령줄에 **man ipa-healthcheck** 를 입력합니다.

94장. 숨겨진 복제본 데모 또는 승격

복제본을 설치한 후에는 복제본이 숨겨져 있는지 또는 표시되는지 여부를 구성할 수 있습니다.

숨겨진 복제본에 대한 자세한 내용은 [숨겨진 복제본 모드를](#) 참조하십시오.

복제본이 **CA** 갱신 서버인 경우 이 복제본을 숨기기 전에 서비스를 다른 복제본으로 이동합니다. 자세한 내용은 [IdM CA 갱신 서버 변경 및 재설정을](#) 참조하십시오.



참고

RHEL 8.1에서 기술 프리뷰로 도입된 숨겨진 복제본 기능은 **RHEL 8.2**부터 완전히 지원됩니다.

절차

•

복제본을 숨기려면 다음을 입력합니다.

```
# ipa server-state replica.idm.example.com --state=hidden
```

또는 다음 명령을 사용하여 복제본을 볼 수 있습니다.

```
# ipa server-state replica.idm.example.com --state=enabled
```


95장. IDENTITY MANAGEMENT 보안 설정

이 섹션에서는 ID 관리의 보안 관련 기능에 대해 설명합니다.

95.1. IDENTITY MANAGEMENT가 기본 보안 설정을 적용하는 방법

기본적으로 **RHEL 8의 IdM(Identity Management)**에서는 시스템 전체 암호화 정책을 사용합니다. 이 정책의 이점은 개별 IdM 구성 요소를 수동으로 강화할 필요가 없다는 것입니다.



중요

시스템 전체의 암호화 정책을 사용하는 것이 좋습니다. 개별 보안 설정을 변경하면 IdM의 구성 요소가 손상될 수 있습니다. 예를 들어 **RHEL 8의 Java**는 **TLS 1.3** 프로토콜을 완벽하게 지원하지 않습니다. 따라서 이 프로토콜을 사용하면 IdM에서 오류가 발생할 수 있습니다.

추가 리소스

- 시스템 전체 암호화 정책에 대한 자세한 내용은 **crypto-policies(7)** 도움말 페이지를 참조하십시오.

95.2. IDENTITY MANAGEMENT에서 익명 LDAP 바인딩

기본적으로 익명이 **IdM(Identity Management) LDAP** 서버에 바인딩됩니다. 익명 바인드는 특정 구성 설정 또는 디렉토리 값을 노출할 수 있습니다. 그러나 **realmd** 또는 이전 **RHEL** 클라이언트와 같은 일부 유틸리티에서는 클라이언트를 등록할 때 도메인 설정을 검색하기 위해 익명 바인드가 활성화되어야 합니다.

추가 리소스

- **IdM LDAP** 서버에서 익명 바인딩 비활성화에 대한 자세한 내용은 **Red Hat Directory Server 11** 관리 가이드의 **익명 바인딩 비활성화** 섹션을 참조하십시오.

96장. IDM 도메인 멤버에서 SAMBA 설정

이 섹션에서는 **Red Hat IdM(Identity Management)** 도메인에 연결된 호스트에서 **Samba**를 설정하는 방법에 대해 설명합니다. **IdM**의 사용자는 신뢰할 수 있는 **AD(Active Directory)** 도메인의 사용자도 **Samba**에서 제공하는 공유 및 프린터 서비스에 액세스할 수 있습니다.

중요

IdM 도메인 멤버에서 **Samba**를 사용하는 것은 지원되지 않는 기술 프리뷰 기능이며 특정 제한 사항이 포함됩니다. 예를 들어 **IdM** 신뢰 컨트롤러가 글로벌 카탈로그 서비스를 지원하지 않기 때문에 **AD-Enrolled Windows** 호스트는 **Windows**에서 **IdM** 사용자 및 그룹을 찾을 수 없습니다. 또한 **IdM Trust** 컨트롤러는 분산 컴퓨팅 환경 / **DCE/RPC**(원격 프로시저 호출) 프로토콜을 사용하여 **IdM** 그룹 확인을 지원하지 않습니다. 결과적으로 **AD** 사용자는 **IdM** 클라이언트의 **Samba** 공유 및 프린터에만 액세스할 수 있습니다.

IdM 도메인 구성원에 **Samba**를 배포하는 고객은 **Red Hat**에 피드백을 제공하는 것이 좋습니다.

사전 요구 사항

- 호스트가 **IdM** 도메인에 클라이언트로 연결됩니다.
- **IdM** 서버와 클라이언트는 모두 **RHEL 8.1** 이상에서 실행해야 합니다.

96.1. 도메인 구성원에 SAMBA 설치를 위한 IDM 도메인 준비

IdM 클라이언트에 **Samba**를 설정하려면 **IdM** 서버에서 **ipa-adtrust-install** 유틸리티를 사용하여 **IdM** 도메인을 준비해야 합니다.

참고

ipa-adtrust-install 명령을 실행하는 시스템은 자동으로 **AD** 신뢰 컨트롤러가 됩니다. 그러나 **IdM** 서버에서 **ipa-adtrust-install** 을 한 번만 실행해야 합니다.

사전 요구 사항

- **IdM** 서버가 설치되어 있습니다.

- 패키지를 설치하고 **IdM** 서비스를 다시 시작하려면 **root** 권한이 필요합니다.

절차

1. 필수 패키지를 설치합니다.

```
[root@ipaserver ~]# yum install ipa-server-trust-ad samba-client
```

2. **IdM** 관리자로 인증합니다.

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

3. **ipa-adtrust-install** 유틸리티를 실행합니다.

```
[root@ipaserver ~]# ipa-adtrust-install
```

IdM이 통합된 **DNS** 서버와 함께 설치된 경우 **DNS** 서비스 레코드가 자동으로 생성됩니다.

통합된 **DNS** 서버 없이 **IdM**을 설치한 경우 **ipa-adtrust-install** 은 계속하기 전에 수동으로 **DNS**에 추가해야 하는 서비스 레코드 목록을 출력합니다.

4. 이 스크립트는 **/etc/samba/smb.conf** 가 이미 있으며 다시 작성하라는 메시지를 표시합니다.

WARNING: The smb.conf already exists. Running ipa-adtrust-install will break your existing Samba configuration.

Do you wish to continue? [no]: yes

5. 스크립트에서는 이전 **Linux** 클라이언트가 신뢰할 수 있는 사용자와 작업할 수 있는 호환성 플러그인인 **slapi-nis** 플러그인을 구성하라는 메시지를 표시합니다.

Do you want to enable support for trusted domains in Schema Compatibility plugin? This will allow clients older than SSSD 1.9 and non-Linux clients to work with trusted users.

Enable trusted domains support in slapi-nis? [no]: yes

- 6.

메시지가 표시되면 **IdM** 도메인의 **NetBIOS** 이름을 입력하거나 **Enter** 키를 눌러 제안된 이름을 수락합니다.

*Trust is configured but no NetBIOS domain name found, setting it now.
Enter the NetBIOS name for the IPA domain.
Only up to 15 uppercase ASCII letters, digits and dashes are allowed.
Example: EXAMPLE.*

NetBIOS domain name [IDM]:

7.

SID 생성 작업을 실행하여 기존 사용자에게 대한 **SID**를 생성하라는 메시지가 표시됩니다.

Do you want to run the ipa-sidgen task? [no]: yes

리소스 집약적인 작업이므로 사용자가 많으면 다른 시간에 이 작업을 실행할 수 있습니다.

8.

(선택 사항) 기본적으로 **Dynamic RPC** 포트 범위는 **Windows Server 2008** 이상에서 **49152-65535**로 정의됩니다. 환경에 대해 다른 동적 **RPC** 포트 범위를 정의해야 하는 경우 다른 포트를 사용하고 방화벽 설정에서 해당 포트를 열도록 **Samba**를 구성합니다. 다음 예제에서는 포트 범위를 **55000-65000**으로 설정합니다.

```
[root@ipaserver ~]# net conf setparm global 'rpc server dynamic port range' 55000-65000
[root@ipaserver ~]# firewall-cmd --add-port=55000-65000/tcp
[root@ipaserver ~]# firewall-cmd --runtime-to-permanent
```

9.

ipa 서비스를 다시 시작하십시오.

```
[root@ipaserver ~]# ipactl restart
```

10.

smbclient 유틸리티를 사용하여 **Samba**가 **IdM** 측에서 **Kerberos** 인증에 응답하는지 확인합니다.

```
[root@ipaserver ~]# smbclient -L server.idm.example.com -k
lp_load_ex: changing to config backend registry
Sharename      Type      Comment
-----
IPC$           IPC       IPC Service (Samba 4.12.3)
...
```

96.2. GPO를 사용하여 ACTIVE DIRECTORY에서 AES 암호화 유형 활성화

이 섹션에서는 **GPO(그룹 정책 오브젝트)**를 사용하여 **AD(Active Directory)**에서 **AES** 암호화 유형을 활성화하는 방법에 대해 설명합니다. **IdM** 클라이언트에서 **Samba** 서버를 실행하는 등의 **RHEL**의 특정 기능에는 이 암호화 유형이 필요합니다.

RHEL은 더 이상 약한 **DES** 및 **RC4** 암호화 유형을 지원하지 않습니다.

사전 요구 사항

- 그룹 정책을 편집할 수 있는 사용자로 **AD**에 로그인했습니다.
- **Group Policy Management Console**(그룹 정책 관리 콘솔)은 컴퓨터에 설치됩니다.

절차

1. 그룹 정책 관리 콘솔을 엽니다.
2. **Default Domain Policy** (기본 도메인 정책)를 마우스 오른쪽 버튼으로 클릭하고 **Edit**(편집)를 선택합니다. **Group Policy Management Editor**(그룹 정책 관리 편집기)가 열립니다.
3. 컴퓨터 구성 → 정책 → **Windows** 설정 → 보안 설정 → 로컬 정책 → 보안 옵션으로 이동합니다.
4. **Network** 보안을 두 번 클릭합니다. **Kerberos** 정책에 허용되는 암호화 유형을 구성합니다.
5. **AES256_HMAC_SHA1**을 선택하고 선택적으로 향후 암호화 유형을 선택합니다.
6. **OK**(확인)를 클릭합니다.
7. 그룹 정책 관리 편집기를 닫습니다.
8. 기본 도메인 컨트롤러 정책에 대한 단계를 반복합니다.

9.

Windows 도메인 컨트롤러(DC)가 그룹 정책을 자동으로 적용할 때까지 기다립니다. 또는 **DC**에 **GPO**를 수동으로 적용하려면 관리자 권한이 있는 계정을 사용하여 다음 명령을 입력합니다.

```
C:\> gpupdate /force /target:computer
```

96.3. IDM 클라이언트에 SAMBA 서버 설치 및 구성

이 섹션에서는 **IdM** 도메인에 등록된 클라이언트에 **Samba**를 설치하고 구성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 서버와 클라이언트는 모두 **RHEL 8.1** 이상에서 실행해야 합니다.
- **IdM** 도메인은 다양한 유형의 서버 문서 [배포의 도메인 구성원에 Samba를 설치하기 위한 IdM 도메인 준비](#)에 설명된 대로 준비됩니다.
- **IdM**에 **AD**로 신뢰가 설정된 경우 **Kerberos**에 대해 **AES** 암호화 유형을 활성화합니다. 예를 들어 그룹 정책 오브젝트(**GPO**)를 사용하여 **AES** 암호화 유형을 활성화합니다. 자세한 내용은 [Deploying different type of servers](#) 설명서에서 **GPO**를 사용하여 **Active Directory**에서 **AES** 암호화 활성화 를 참조하십시오.

절차

1.

ipa-client-samba 패키지를 설치합니다.

```
[root@idm_client]# yum install ipa-client-samba
```

2.

ipa-client-samba 유틸리티를 사용하여 클라이언트를 준비하고 초기 **Samba** 구성을 생성합니다.

```
[root@idm_client]# ipa-client-samba
Searching for IPA server...
IPA server: DNS discovery
Chosen IPA master: idm_server.idm.example.com
SMB principal to be created: cifs/idm_client.idm.example.com@IDM.EXAMPLE.COM
NetBIOS name to be used: IDM_CLIENT
Discovered domains to use:

Domain name: idm.example.com
NetBIOS name: IDM
```

```
SID: S-1-5-21-525930803-952335037-206501584
ID range: 212000000 - 212199999
```

```
Domain name: ad.example.com
NetBIOS name: AD
```

```
SID: None
ID range: 1918400000 - 1918599999
```

```
Continue to configure the system with these values? [no]: yes
Samba domain member is configured. Please check configuration at
/etc/samba/smb.conf and start smb and winbind services
```

3.

기본적으로 **ipa-client-samba** 는 사용자가 연결할 때 사용자의 홈 디렉토리를 동적으로 공유하는 **/etc/samba/smb.conf** 파일에 **[homes]** 섹션을 자동으로 추가합니다. 사용자에게 이 서버에 홈 디렉토리가 없거나 공유하지 않으려면 **/etc/samba/smb.conf**에서 다음 행을 제거하십시오.

```
[homes]
read only = no
```

4.

디렉토리 및 프린터 공유. 자세한 내용은 다양한 유형의 서버 배포 설명서의 다음 섹션을 참조하십시오.

•

[POSIX ACL을 사용하는 Samba 파일 공유 설정](#)

•

[Windows ACL을 사용하는 공유 설정](#)

•

[인쇄 서버로 Samba 설정](#)

5.

로컬 방화벽에서 **Samba** 클라이언트에 필요한 포트를 엽니다.

```
[root@idm_client]# firewall-cmd --permanent --add-service=samba-client
[root@idm_client]# firewall-cmd --reload
```

6.

smb 및 **winbind** 서비스를 활성화하고 시작합니다.

```
[root@idm_client]# systemctl enable --now smb winbind
```

검증 단계

samba-client 패키지가 설치된 다른 **IdM** 도메인 멤버에서 다음 확인 단계를 실행합니다.

1. **Kerberos** 티켓 부여 티켓을 인증하고 받습니다.

```
$ kinit example_user
```

2. **Kerberos** 인증을 사용하여 **Samba** 서버의 공유를 나열합니다.

```
$ smbclient -L idm_client.idm.example.com -k
lp_load_ex: changing to config backend registry
```

Sharename	Type	Comment
example	Disk	
IPC\$	IPC	IPC Service (Samba 4.12.3)

...

추가 리소스

- 구성 중에 **ipa-client-samba** 가 수행하는 단계에 대한 자세한 내용은 **ipa-client-samba(1)** 도움말 페이지를 참조하십시오.

96.4. IDM에서 새 도메인을 신뢰하는 경우 수동으로 ID 매핑 구성 추가

Samba에는 사용자가 리소스에 액세스하는 각 도메인에 대한 **ID** 매핑 구성이 필요합니다. **IdM** 클라이언트에서 실행되는 기존 **Samba** 서버의 경우 관리자가 새 신뢰성을 **AD(Active Directory)** 도메인에 추가한 후 수동으로 **ID** 매핑 구성을 추가해야 합니다.

사전 요구 사항

- **IdM** 클라이언트에 **Samba**를 구성했습니다. 이후 **IdM**에 새로운 신뢰가 추가되었습니다.
- **Kerberos**의 **DES** 및 **RC4** 암호화 유형은 신뢰할 수 있는 **AD** 도메인에서 비활성화해야 합니다. 보안상의 이유로 **RHEL 8**은 이러한 약한 암호화 유형을 지원하지 않습니다.

절차

1. 호스트의 **keytab**을 사용하여 인증합니다.


```
[root@idm_client]# kinit -k
```

2.

ipa idrange-find 명령을 사용하여 새 도메인의 기본 ID 및 ID 범위 크기를 둘 다 표시합니다. 예를 들어 다음 명령은 **ad.example.com** 도메인의 값을 표시합니다.

```
[root@idm_client]# ipa idrange-find --name="AD.EXAMPLE.COM_id_range" --raw
-----
1 range matched
-----
cn: AD.EXAMPLE.COM_id_range
ipabaseid: 1918400000
ipaidrangesize: 200000
ipabaserid: 0
ipantrusteddomainsid: S-1-5-21-968346183-862388825-1738313271
iparangetype: ipa-ad-trust
-----
Number of entries returned 1
-----
```

다음 단계의 **ipabaseid** 및 **ipaidrangesize** 속성의 값이 필요합니다.

3.

사용 가능한 가장 높은 ID를 계산하려면 다음 공식을 사용하십시오.

```
maximum_range = ipabaseid + ipaidrangesize - 1
```

이전 단계의 값을 사용할 때 **ad.example.com** 도메인에 사용할 수 있는 가장 높은 ID는 **1918599999** (**1918400000 + 200000 - 1**)입니다.

4.

/etc/samba/smb.conf 파일을 편집하고 도메인의 ID 매핑 구성을 **[global]** 섹션에 추가합니다.

```
idmap config AD : range = 1918400000 - 1918599999
idmap config AD : backend = sss
```

ipabaseid 특성의 값을 가장 낮은 값으로 지정하고 이전 단계의 계산된 값을 가장 높은 범위 값으로 지정합니다.

5.

smb 및 **winbind** 서비스를 다시 시작하십시오.

```
[root@idm_client]# systemctl restart smb winbind
```

검증 단계

1. 새 도메인에서 사용자로 인증하고 **Kerberos** 티켓을 받습니다.

```
$ kinit example_user
```

2. **Kerberos** 인증을 사용하여 **Samba** 서버의 공유를 나열합니다.

```
$ smbclient -L idm_client.idm.example.com -k
lp_load_ex: changing to config backend registry
```

Sharename	Type	Comment
example	Disk	
IPC\$	IPC	IPC Service (Samba 4.12.3)
...		

96.5. 추가 리소스

- **RHEL** 호스트를 **IdM** 도메인에 연결하는 방법에 대한 자세한 내용은 **Identity Management 설치 가이드**의 **Identity Management 클라이언트 설치** 섹션을 참조하십시오.

97장. 다른 RED HAT 제품과 IDM 통합

이 섹션에서는 **IdM**과 통합된 다른 **Red Hat** 제품에 대한 설명서를 제공합니다. **IdM** 사용자가 서비스에 액세스할 수 있도록 이러한 제품을 구성할 수 있습니다.

Ansible Automation Platform

LDAP 인증 설정

OpenShift Container Platform

LDAP ID 공급자 구성

OpenStack Platform

OpenStack Identity(keystone)와 Red Hat IdM(Identity Manager) 통합

Satellite

Red Hat Identity Management 사용

SSO(Single Sign-On)

SSSD 및 FreeIPA ID 관리 통합

가상화

외부 LDAP 공급자 구성

98장. NIS에서 IDENTITY MANAGEMENT로 마이그레이션

NIS(네트워크 정보 서비스) 서버에는 사용자, 그룹, 호스트, 넷그룹 및 자동 마운트 맵에 대한 정보가 포함될 수 있습니다. 시스템 관리자는 이러한 항목 유형, 인증 및 권한 부여를 **NIS** 서버에서 **IdM**(Identity Management) 서버로 마이그레이션하여 모든 사용자 관리 작업이 **IdM** 서버에서 수행되도록 할 수 있습니다. **NIS**에서 **IdM**으로 마이그레이션하면 **Kerberos**와 같은 보안 프로토콜에도 액세스할 수 있습니다.

98.1. IDM에서 NIS 활성화

NIS와 **IdM**(Identity Management) 서버 간 통신을 허용하려면 **IdM** 서버에서 **NIS** 호환성 옵션을 활성화해야 합니다.

사전 요구 사항

- **IdM** 서버에 루트 액세스 권한이 있습니다.

절차

1. **IdM** 서버에서 **NIS** 리스너 및 호환성 플러그인을 활성화합니다.

```
[root@ipaserver ~]# ipa-nis-manage enable
[root@ipaserver ~]# ipa-compatible-manage enable
```

2. 선택 사항: 보다 엄격한 방화벽 구성의 경우 고정 포트를 설정합니다.

예를 들어 포트를 사용하지 않는 포트 **514**로 설정하려면 다음을 수행합니다.

```
[root@ipaserver ~]# ldapmodify -x -D 'cn=directory manager' -W
dn: cn=NIS Server,cn=plugins,cn=config
changetype: modify
add: nsslapd-pluginarg0
nsslapd-pluginarg0: 514
```



주의

다른 서비스와 충돌하지 않도록 하려면 1024 이상의 포트 번호를 사용하지 않습니다.

3.

포트 매핑 서비스를 활성화하고 시작합니다.

```
[root@ipaserver ~]# systemctl enable rpcbind.service
[root@ipaserver ~]# systemctl start rpcbind.service
```

4.

디렉터리 서버를 다시 시작하십시오.

```
[root@ipaserver ~]# systemctl restart dirsrv.target
```

98.2. NIS에서 IDM으로 사용자 항목 마이그레이션

NIS `passwd` 맵에는 이름, UID, 기본 그룹, GECOS, 셸 및 홈 디렉터리와 같은 사용자에 대한 정보가 포함되어 있습니다. 이 데이터를 사용하여 NIS 사용자 계정을 IdM(Identity Management)으로 마이그레이션합니다.

사전 요구 사항

- NIS 서버에 루트 액세스 권한이 있습니다.
- NIS가 IdM에서 활성화되어 있습니다.
- NIS 서버가 IdM에 등록되었습니다.

절차

1.

`yp-tools` 패키지를 설치합니다.

```
[root@nis-server ~]# yum install yp-tools -y
```

2.

NIS 서버에서 다음 콘텐츠를 사용하여 `/root/nis-users.sh` 스크립트를 만듭니다.

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -d $1 -h $2 passwd > /dev/shm/nis-map.passwd 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.passwd) ; do
  IFS=' '
  username=$(echo $line | cut -f1 -d:)
  # Not collecting encrypted password because we need cleartext password
  # to create kerberos key
  uid=$(echo $line | cut -f3 -d:)
  gid=$(echo $line | cut -f4 -d:)
  gecos=$(echo $line | cut -f5 -d:)
  homedir=$(echo $line | cut -f6 -d:)
  shell=$(echo $line | cut -f7 -d:)

  # Now create this entry
  echo passwd0rd1 | ipa user-add $username --first=NIS --last=USER \
    --password --gidnumber=$gid --uid=$uid --gecos='$gecos' --homedir=$homedir \
    --shell=$shell
  ipa user-show $username
done
```

3.

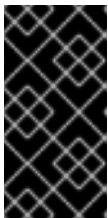
IdM 관리자로 인증합니다.

```
[root@nis-server ~]# kinit admin
```

4.

스크립트를 실행합니다. 예를 들면 다음과 같습니다.

```
[root@nis-server ~]# sh /root/nis-users.sh nisdomain nis-server.example.com
```



중요

이 스크립트는 이름, 성에 하드 코딩된 값을 사용하고 암호를 `passwd0rd1` 로 설정합니다. 사용자는 다음 로그인 시 임시 암호를 변경해야 합니다.

98.3. NIS에서 IDM으로 사용자 그룹 마이그레이션

NIS 그룹 맵에는 그룹 이름, GID 또는 그룹 구성원과 같은 그룹에 대한 정보가 포함되어 있습니다. 이 데이터를 사용하여 NIS 그룹을 IdM(Identity Management)으로 마이그레이션합니다.

- **NIS 서버에 루트 액세스 권한이 있습니다.**
- **NIS가 IdM에서 활성화되어 있습니다.**
- **NIS 서버가 IdM에 등록되었습니다.**

절차

1. **yp-tools 패키지를 설치합니다.**

```
[root@nis-server ~]# yum install yp-tools -y
```

2. **NIS 서버에서 다음 콘텐츠를 사용하여 /root/nis-groups.sh 스크립트를 만듭니다.**

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -d $1 -h $2 group > /dev/shm/nis-map.group 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.group); do
  IFS=' '
  groupname=$(echo $line | cut -f1 -d:)
  # Not collecting encrypted password because we need cleartext password
  # to create kerberos key
  gid=$(echo $line | cut -f3 -d:)
  members=$(echo $line | cut -f4 -d:)

  # Now create this entry
  ipa group-add $groupname --desc=NIS_GROUP_$groupname --gid=$gid
  if [ -n "$members" ]; then
    ipa group-add-member $groupname --users=${members}
  fi
  ipa group-show $groupname
done
```

3. **IdM 관리자로 인증합니다.**

```
[root@nis-server ~]# kinit admin
```

4. **스크립트를 실행합니다. 예를 들면 다음과 같습니다.**

```
[root@nis-server ~]# sh /root/nis-groups.sh nisdomain nis-server.example.com
```

98.4. NIS에서 IDM으로 호스트 항목 마이그레이션

NIS 호스트 맵에는 호스트 이름 및 IP 주소와 같은 호스트에 대한 정보가 포함되어 있습니다. 이 데이터를 사용하여 NIS 호스트 항목을 IdM(Identity Management)으로 마이그레이션합니다.



참고

IdM에서 호스트 그룹을 생성하면 해당 새도우 NIS 그룹이 자동으로 생성됩니다. 이러한 새도우 NIS 그룹에서 `ipa netgroup-*` 명령을 사용하지 마십시오. `ipa netgroup-*` 명령을 사용하여 `netgroup-add` 명령을 통해 생성된 기본 넷 그룹을 관리합니다.

사전 요구 사항

- NIS 서버에 루트 액세스 권한이 있습니다.
- NIS가 IdM에서 활성화되어 있습니다.
- NIS 서버가 IdM에 등록되었습니다.

절차

1. `yp-tools` 패키지를 설치합니다.

```
[root@nis-server ~]# yum install yp-tools -y
```

2. NIS 서버에서 다음 콘텐츠를 사용하여 `/root/nis-hosts.sh` 스크립트를 만듭니다.

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -d $1 -h $2 hosts | egrep -v "localhost|127.0.0.1" > /dev/shm/nis-map.hosts 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.hosts); do
    IFS=' '
    ipaddress=$(echo $line | awk '{print $1}')
    hostname=$(echo $line | awk '{print $2}')
    master=$(ipa env xmlrpc_uri | tr -d '[:space:]' | cut -f3 -d: | cut -f3 -d/)
```



```

domain=$(ipa env domain | tr -d '[:space:]' | cut -f2 -d:)
if [ $(echo $hostname | grep "\." | wc -l) -eq 0 ] ; then
    hostname=$(echo $hostname.$domain)
fi
zone=$(echo $hostname | cut -f2- -d.)
if [ $(ipa dnszone-show $zone 2>/dev/null | wc -l) -eq 0 ] ; then
    ipa dnszone-add --name-server=$master --admin-email=root.$master
fi
ptrzone=$(echo $ipaddress | awk -F. '{print $3 "." $2 "." $1 ".in-addr.arpa."}')
if [ $(ipa dnszone-show $ptrzone 2>/dev/null | wc -l) -eq 0 ] ; then
    ipa dnszone-add $ptrzone --name-server=$master --admin-email=root.$master
fi
# Now create this entry
ipa host-add $hostname --ip-address=$ipaddress
ipa host-show $hostname
done

```

3.

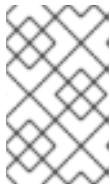
IdM 관리자로 인증합니다.

```
[root@nis-server ~]# kinit admin
```

4.

스크립트를 실행합니다. 예를 들면 다음과 같습니다.

```
[root@nis-server ~]# sh /root/nis-hosts.sh nisdomain nis-server.example.com
```



참고

이 스크립트는 별칭과 같은 특수 호스트 구성을 마이그레이션하지 않습니다.

98.5. NIS에서 IDM으로 넷그룹 항목 마이그레이션

NIS netgroup 맵에는 넷 그룹에 대한 정보가 포함되어 있습니다. 이 데이터를 사용하여 **NIS** 넷 그룹을 **IdM(Identity Management)**으로 마이그레이션합니다.

사전 요구 사항

- **NIS** 서버에 루트 액세스 권한이 있습니다.
- **NIS**가 **IdM**에서 활성화되어 있습니다.

- **NIS 서버가 IdM에 등록되었습니다.**

절차

1.

yp-tools 패키지를 설치합니다.

```
[root@nis-server ~]# yum install yp-tools -y
```

2.

NIS 서버에서 다음 콘텐츠를 사용하여 /root/nis-netgroups.sh 스크립트를 만듭니다.

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -k -d $1 -h $2 netgroup > /dev/shm/nis-map.netgroup 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.netgroup); do
  IFS=' '
  netgroupname=$(echo $line | awk '{print $1}')
  triples=$(echo $line | sed "s/^$netgroupname //" )
  echo "ipa netgroup-add $netgroupname --desc=NIS_NG_$netgroupname"
  if [ $(echo $line | grep "(," | wc -l) -gt 0 ]; then
    echo "ipa netgroup-mod $netgroupname --hostcat=all"
  fi
  if [ $(echo $line | grep ",," | wc -l) -gt 0 ]; then
    echo "ipa netgroup-mod $netgroupname --usercat=all"
  fi

  for triple in $triples; do
    triple=$(echo $triple | sed -e 's/-//g' -e 's/(// -e 's/)//')
    if [ $(echo $triple | grep ",.*," | wc -l) -gt 0 ]; then
      hostname=$(echo $triple | cut -f1 -d,)
      username=$(echo $triple | cut -f2 -d,)
      domain=$(echo $triple | cut -f3 -d,)
      hosts=""; users=""; doms="";
      [ -n "$hostname" ] && hosts="--hosts=$hostname"
      [ -n "$username" ] && users="--users=$username"
      [ -n "$domain" ] && doms="--nisdomain=$domain"
      echo "ipa netgroup-add-member $hosts $users $doms"
    else
      netgroup=$triple
      echo "ipa netgroup-add $netgroup --desc=NIS_NG_$netgroup"
    fi
  done
done
```

3.

IdM 관리자로 인증합니다.

```
[root@nis-server ~]# kinit admin
```

4.

스크립트를 실행합니다. 예를 들면 다음과 같습니다.

```
[root@nis-server ~]# sh /root/nis-netgroups.sh nisdomain nis-server.example.com
```

98.6. NIS에서 IDM으로 자동 마운트 맵 마이그레이션

자동 마운트 맵은 위치(상위 항목), 관련 키 및 맵을 정의하는 일련의 중첩 및 상호 관련 항목입니다. NIS 자동 마운트를 IdM(Identity Management)으로 마이그레이션하려면 다음을 수행합니다.

사전 요구 사항

- NIS 서버에 루트 액세스 권한이 있습니다.
- NIS가 IdM에서 활성화되어 있습니다.
- NIS 서버가 IdM에 등록되었습니다.

절차

1.

yp-tools 패키지를 설치합니다.

```
[root@nis-server ~]# yum install yp-tools -y
```

2.

NIS 서버에서 다음 콘텐츠를 사용하여 /root/nis-automounts.sh 스크립트를 만듭니다.

```
#!/bin/sh
# $1 is for the automount entry in ipa

ipa automountlocation-add $1

# $2 is the NIS domain, $3 is the NIS master server, $4 is the map name

ypcat -k -d $2 -h $3 $4 > /dev/shm/nis-map.$4 2>&1

ipa automountmap-add $1 $4

basedn=$(ipa env basedn | tr -d '[:space:]' | cut -f2 -d:)
```

```

cat > /tmp/imap.ldif <<EOF
dn: nis-domain=$2+nis-map=$4,cn=NIS Server,cn=plugins,cn=config
objectClass: extensibleObject
nis-domain: $2
nis-map: $4
nis-base: automountmapname=$4,cn=$1,cn=automount,$basedn
nis-filter: (objectclass=*)
nis-key-format: %{automountKey}
nis-value-format: %{automountInformation}
EOF
ldapadd -x -h $3 -D "cn=Directory Manager" -W -f /tmp/imap.ldif

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.$4); do
  IFS=" "
  key=$(echo "$line" | awk '{print $1}')
  info=$(echo "$line" | sed -e "s^$key[ \t]*")
  ipa automountkey-add nis $4 --key="$key" --info="$info"
done

```



참고

스크립트는 **NIS** 자동 마운트 정보를 내보내고 자동 마운트 위치 및 관련 맵에 대한 **LDAP(LDIF)**를 생성하고, **LDIF** 파일을 **IdM** 디렉터리 서버로 가져옵니다.

3.

IdM 관리자로 인증합니다.

```
[root@nis-server ~]# kinit admin
```

4.

스크립트를 실행합니다. 예를 들면 다음과 같습니다.

```
[root@nis-server ~]# sh /root/nis-automounts.sh location nisdomain
nis-server.example.com map_name
```

99장. IDM에서 자동 마운트 사용

자동 마운트는 여러 시스템에서 디렉토리를 관리, 구성 및 액세스하는 방법입니다. 자동 마운트는 액세스 권한이 요청될 때마다 디렉토리를 자동으로 마운트합니다. **IdM(Identity Management)** 도메인에서는 도메인 내의 클라이언트에서 손쉽게 디렉토리를 공유할 수 있으므로 잘 작동합니다.

예에서는 다음 시나리오를 사용합니다.

- **NFS-server.idm.example.com** 은 **NFS**(네트워크 파일 시스템) 서버의 **FQDN**(정규화된 도메인 이름)입니다.
- **nfs-server.idm.example.com** 은 **raleigh** 자동 마운트 위치에 있는 **IdM** 클라이언트입니다.
- **NFS** 서버는 **/exports/project** 디렉토리를 읽기-쓰기로 내보냅니다.
- **developers** 그룹에 속하는 **IdM** 사용자는 동일한 **raleigh** 자동 마운트 위치에 있는 **IdM** 클라이언트에서 내보낸 디렉토리의 **/devel/project/** 로 **NFS** 서버와 동일한 **raleigh** 자동 마운트 위치에 있는 콘텐츠에 액세스할 수 있습니다.
- **IdM-client.idm.example.com** 은 **raleigh** 자동 마운트 위치에 있는 **IdM** 클라이언트입니다.

중요

NFS 서버 대신 **Samba** 서버를 사용하여 **IdM** 클라이언트의 공유를 제공하려면 **IPA** 환경에서 **Autofs**를 사용하여 **Kerberos CIFS** 마운트를 구성하는 방법을 참조하십시오.

<https://access.redhat.com/solutions/6596071> KCS 솔루션.

장에는 다음 섹션이 포함되어 있습니다.

1. **IdM에서 autofs 및 자동 마운트**

2. **NFS 서버에 대한 IdM 키맵 구성**
3. **IdM에서 NFS 공유 내보내기**
4. **IdM CLI를 사용하여 IdM에서 자동 마운트 위치 및 맵 구성**
5. **자동 마운트 위치에 IdM 클라이언트 추가**
6. **IdM 클라이언트에 자동 마운트 설정**
7. **IdM 사용자가 IdM 클라이언트의 NFS 공유에 액세스할 수 있는지 확인**

99.1. IDM에서 AUTOFS 및 자동 마운트

autofs 서비스는 액세스할 때 디렉토리를 마운트하도록 자동 마운트 하여 필요에 따라 자동 마운트를 자동화합니다. 또한 기간 동안 사용하지 않으면 **autofs** 는 자동 마운트된 디렉토리 를 마운트 해제하도록 자동 마운트를 지시합니다. 정적 마운팅과 달리 온디맨드 마운트는 시스템 리소스를 절약합니다.

자동 마운트 맵

autofs 를 사용하는 시스템에서는 자동 마운트 구성이 여러 다른 파일에 저장됩니다. 기본 자동 마운트 구성 파일은 자동 마운트 지점의 마스터 매핑과 관련 리소스가 포함된 **/etc/auto.master** 입니다. 이 매핑을 자동 마운트 맵 이라고 합니다.

/etc/auto.master 구성 파일에는 마스터 맵이 포함되어 있습니다. 다른 맵에 대한 참조를 포함할 수 있습니다. 이 맵은 직접 또는 간접일 수 있습니다. 직접 맵은 마운트 지점에 절대 경로 이름을 사용하지만 간접 맵은 상대 경로 이름을 사용합니다.

IdM에서 구성 자동 마운트

자동 마운트 는 일반적으로 로컬 **/etc/auto.master** 및 관련 파일에서 맵 데이터를 검색하지만 다른 소스에서 맵 데이터를 검색할 수도 있습니다. 가장 일반적인 소스는 **LDAP** 서버입니다. **IdM(Identity Management)** 컨텍스트에서 이는 **389 Directory Server**입니다.

autofs 를 사용하는 시스템이 **IdM** 도메인의 클라이언트인 경우 자동 마운트 구성은 로컬 구성 파일에 저장되지 않습니다. 대신 맵, 위치 및 키와 같은 **autofs** 구성이 **IdM** 디렉터리에 **LDAP** 항목으로

저장됩니다. 예를 들어 **idm.example.com** IdM 도메인의 경우 기본 마스터 맵 은 다음과 같이 저장됩니다.

```
dn:
automountmapname=auto.master,cn=default,cn=automount,dc=idm,dc=example,dc=com
objectClass: automountMap
objectClass: top
automountMapName: auto.master
```

추가 리소스

- [필요에 따라 파일 시스템 마운트](#)

99.2. NFS 서버에 대한 IDM 키맵 구성

다른 IdM(Identity Management) 클라이언트에 로그인한 사용자가 이 NFS 서버의 디렉터리 및 파일에 액세스할 수 있도록 Kerberos 인식 NFS 서버를 구성합니다.

이 예제에서는 **nfs-server.idm.example.com** 에서 실행되는 NFS 서비스를 구성하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- NFS 서버에 대한 루트 액세스 권한이 있습니다.
- NFS 공유를 내보내는 데 필요한 패키지가 설치되어 있어야 합니다.
- [선택 사항] 방화벽 뒤에서 실행되도록 NFS 서버를 구성했습니다.

절차

1. IdM이 등록된 호스트에서 IdM에 NFS 서비스를 추가합니다.

```
$ ipa service-add nfs/nfs-server.idm.example.com
```

```
-----
Added service "nfs/nfs-server.idm.example.com@IDM.EXAMPLE.COM"
-----
```

```
Principal name: nfs/nfs-server.idm.example.com@IDM.EXAMPLE.COM
Principal alias: nfs/nfs-server.idm.example.com@IDM.EXAMPLE.COM
Managed by: nfs-server.idm.example.com
```

2.

NFS 서버에서 NFS 서비스의 keytab을 가져옵니다.

```
# ipa-getkeytab -p nfs/nfs-server.idm.example.com -k /etc/krb5.keytab
Keytab successfully retrieved and stored in: /etc/krb5.keytab
```

3.

NFS 서버에서 NFS 서비스를 다시 시작합니다.

```
# systemctl restart nfs-server
```

4.

NFS 서버에서 NFS 서비스를 활성화합니다.

```
# systemctl enable nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service →
/usr/lib/systemd/system/nfs-server.service.
```

99.3. IDM에서 NFS 공유 내보내기

IdM(Identity Management) 시스템 관리자는 **NFS** 서버를 사용하여 네트워크를 통해 디렉터리를 **IdM** 사용자와 공유할 수 있습니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.
- **NFS** 서버에 대한 루트 액세스 권한이 있습니다.
- **NFS** 공유를 내보내는 데 필요한 패키지가 설치되어 있어야 합니다.
- [선택 사항] 방화벽 뒤에서 실행되도록 **NFS** 서버를 구성했습니다.

절차

1. 내보낼 디렉토리를 만듭니다.

```
# mkdir -p /exports/project
```

2. 소유자와 그룹에 디렉토리를 읽고 쓰고 실행할 수 있는 권한을 부여합니다.

```
# chmod 770 /exports/project
```

3. 디렉토리에 생성된 모든 파일의 그룹 소유권이 디렉토리 소유자의 그룹 소유권이 설정되도록 **GSID Sticky bit**를 추가합니다.

```
# chmod g+s /exports_ro/documentation /exports/project
```

4. 디렉토리에 액세스할 수 있는 멤버가 있는 **IdM** 그룹을 만듭니다. 예제 **IdM** 그룹은 **developers**입니다.

```
# ipa group-add developers
```

5. 그룹의 모든 **IdM** 사용자가 액세스할 수 있도록 두 디렉토리의 그룹 소유권을 **developers**로 변경합니다.

```
# chgrp developers /exports/project
```

1. **IdM** 사용자를 그룹에 추가합니다. 예제 사용자는 **idm_user**:

```
# ipa group-add-member developers --users=idm_user
```

2. 다음과 같은 내용이 포함된 디렉토리에 파일을 생성합니다.

```
# echo "this is a read-write file" > /exports/project/rw_file
```

3. **/etc/exports.d/** 디렉토리의 파일에 다음 정보를 추가합니다.

- 내보낼 디렉터리

- 사용자가 디렉터리의 파일에 액세스하기 위해 인증하는 방법
- 사용자가 디렉터리의 파일에 대해 보유할 수 있는 권한

```
# echo "/exports/project *(sec=krb5p,rw)" > /etc/exports.d/project.exports
```

sec=krb5 는 로컬 **UNIX UID** 및 **GID** 대신 **Kerberos V5** 프로토콜을 사용하여 사용자를 인증합니다.

또는 **sec=krb5i** 또는 **sec=krb5p** 를 사용합니다.

sec=krb5i

사용자 인증에 **Kerberos V5**를 사용하고 보안 체크섬을 사용하여 **NFS** 작업을 무결성 검사를 수행하여 데이터 조작을 방지합니다.

sec=krb5p

사용자 인증, 무결성 검사 및 **NFS** 트래픽을 암호화하여 트래픽 스니핑을 방지합니다. 이 설정은 가장 안전한 설정이지만 가장 많은 성능 오버헤드가 포함됩니다.

4.

/var/lib/nfs/etab에 보관된 마스터 내보내기 테이블을 **/etc/exports.d** 아래의 **/etc/exports.d**와 동기화하여 모든 디렉토리를 다시 내보낼 수 있습니다.

```
# exportfs -r
```

5.

/etc/exports에 적합한 현재 내보내기 목록을 표시하십시오 :

```
# exportfs -s
/exports/project *
(sync,wdelay,hide,no_subtree_check,sec=krb5p,rw,secure,root_squash,no_all_squas
h)
```

추가 리소스

- **the krb5,krb5 i 및 krb5 p** 방법에 대한 자세한 내용은 **nfs** 도움말 페이지를 참조하십시오.

99.4. IDM CLI를 사용하여 IDM에서 자동 마운트 위치 및 맵 구성

위치는 모두 **auto.master**에 저장되는 맵 집합입니다. 위치는 여러 개의 맵을 저장할 수 있습니다. 위치 항목은 맵 항목의 컨테이너로만 작동하며, 에서 자동 마운트 구성이 아닙니다.

IdM(Identity Management)의 시스템 관리자는 **IdM**에서 자동 마운트 위치와 맵을 구성하여 지정된 위치의 **IdM** 사용자가 호스트의 특정 마운트 지점으로 이동하여 **NFS** 서버에서 내보낸 공유에 액세스할 수 있습니다. 내보낸 **NFS** 서버 디렉터리와 마운트 지점 모두 맵에 지정됩니다. 이 예제에서는 **IdM** 클라이언트의 **/devel/** 마운트 지점에 **nfs-server.idm.example.com:/exports/project** 공유를 읽기-쓰기 디렉터리로 마운트 하는 맵과 **raleigh** 위치를 구성하는 방법을 설명합니다.

사전 요구 사항

- **IdM**이 등록한 호스트에 **IdM** 관리자로 로그인되어 있습니다.

절차

1. **raleigh** 자동 마운트 위치를 생성합니다.

```
$ ipa automountlocation-add raleigh
-----
Added automount location "raleigh"
-----
Location: raleigh
```

2. **raleigh** 위치에 **auto.devel** 자동 마운트 맵을 생성합니다.

```
$ ipa automountmap-add raleigh auto.devel
-----
Added automount map "auto.devel"
-----
Map: auto.devel
```

3. 내보내기/ 공유에 대한 키 및 마운트 정보를 추가합니다.

- a. **auto.devel** 맵의 키 및 마운트 정보를 추가합니다.

```
$ ipa automountkey-add raleigh auto.devel --key='*' --info='-sec=krb5p,vers=4 nfs-
server.idm.example.com:/exports/&'
-----
Added automount key ""
```

```
-----
Key: *
```

```
Mount information: -sec=krb5p,vers=4 nfs-server.idm.example.com:/exports/&
```

b.

auto.master 맵의 키 및 마운트 정보를 추가합니다.

```
$ ipa automountkey-add raleigh auto.master --key=/devel --info=auto.devel
```

```
-----
Added automount key "/devel"
```

```
-----
Key: /devel
```

```
Mount information: auto.devel
```

99.5. 자동 마운트 위치에 IDM 클라이언트 추가

IdM(Identity Management) 시스템 관리자는 **IdM** 사용자가 이 **IdM** 클라이언트에 로그인한 경우 위치에 구성된 **NFS** 공유를 **IdM** 사용자가 액세스할 수 있도록 자동 마운트 위치에 **IdM** 클라이언트를 추가할 수 있습니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.
- 호스트는 **IdM**에 있습니다. 예제 호스트는 **idm-client.idm.example.com**입니다.
- 자동 마운트 위치가 있습니다. 예제 위치는 **raleigh**입니다.

절차

- **IdM**이 등록된 호스트에서 자동 마운트 위치에 호스트를 추가합니다.

```
$ ipa host-mod idm-client.idm.example.com --location raleigh
```

```
-----
Modified host "idm-client.idm.example.com"
```

```
-----
Host name: idm-client.idm.example.com
```

```
Location: raleigh
```

```
Platform: x86_64
```

```
[...]
```

99.6. IDM 클라이언트에 자동 마운트 설정

IdM(Identity Management) 시스템 관리자는 **IdM** 클라이언트에서 자동 마운트 서비스를 구성하여 사용자가 클라이언트에 로그인할 때 클라이언트가 추가된 위치에 대해 구성된 **NFS** 공유를 **IdM** 사용자가 자동으로 액세스할 수 있도록 할 수 있습니다. 이 예제에서는 **raleigh** 위치에서 사용할 수 있는 자동 마운트 서비스를 사용하도록 **IdM** 클라이언트를 구성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 클라이언트에 대한 루트 액세스 권한이 있습니다.
- **IdM** 관리자로 로그인했습니다.
- 자동 마운트 위치가 있습니다. 예제 위치는 **raleigh** 입니다.

절차

1. **IdM** 클라이언트에서 **ipa-client-automount** 명령을 입력하고 위치를 지정합니다. 스크립트를 무인으로 실행하려면 **-U** 옵션을 사용합니다.

```
# ipa-client-automount --location raleigh -U
```

2. **autofs** 서비스를 중지하고 **SSSD** 캐시를 지우고 **autofs** 서비스를 시작하여 새 구성 설정을 로드합니다.

```
# systemctl stop autofs ; sss_cache -E ; systemctl start autofs
```

99.7. IDM 사용자가 IDM 클라이언트의 NFS 공유에 액세스할 수 있는지 확인

IdM(Identity Management) 시스템 관리자는 특정 그룹의 멤버인 **IdM** 사용자가 특정 **IdM** 클라이언트에 로그인할 때 **NFS** 공유에 액세스할 수 있는지 테스트할 수 있습니다.

예에서는 다음 시나리오가 테스트되었습니다.

- **developers** 그룹에 속하는 **IdM** 사용자는 구성된 **IdM** 클라이언트의 **/devel/project** 디렉터리에 있는 파일의 내용을 읽고 쓸 수 있습니다.

- **IdM-client.idm.example.com** 은 **NFS 클라이언트로 구성된 IdM 클라이언트**입니다.
- **IdM 사용자는 NFS 서버에서 공유 디렉터리를 소유한 그룹의 멤버**입니다. **예제 사용자는 idm_user** 이고 **예제 그룹은 developers** 입니다.

사전 요구 사항

- **NFS 서버에 대해 IdM 키맵을 구성하고 NFS 공유를 내보내** 있습니다.
- **IdM 사용자가 NFS 공유에 액세스할 수 있도록 지정한 IdM에서 자동 마운트 위치 및 맵을 구성**했습니다.
- **IdM 클라이언트를 관련 자동 마운트 위치에 추가**했습니다.
- **IdM 클라이언트에 자동 마운트를 구성**했습니다.

절차

1. **IdM 사용자가 읽기-쓰기 디렉터리에 액세스할 수 있는지 확인**합니다.

- a. **IdM 사용자로 IdM 클라이언트에 연결**합니다.

```
$ ssh idm_user@idm-client.idm.example.com
Password:
```

- b. **IdM 사용자의 TGT(권장 티켓)를 받습니다.**

```
$ kinit idm_user
```

- c. **[선택 사항] IdM 사용자의 그룹 구성원을 확인**합니다.

```
$ ipa user-show idm_user
User login: idm_user
[...]
Member of groups: developers, ipausers
```

d.

/devel/project 디렉터리로 이동합니다.

```
$ cd /devel/project
```

e.

디렉터리 내용을 나열합니다.

```
$ ls  
rw_file
```

f.

쓰기 권한을 테스트하려면 디렉터리의 파일에 행을 추가합니다.

```
$ echo "idm_user can write into the file" > rw_file
```

g.

[선택 사항] 파일의 업데이트된 내용을 확인합니다.

```
$ cat rw_file  
this is a read-write file  
idm_user can write into the file
```

*출력은 **idm_user** 가 파일에 쓸 수 있음을 확인합니다.*