



Red Hat Enterprise Linux 9

Ansible을 사용하여 Identity Management 설치 및 관리

Ansible Core 및 ansible-freeipa 패키지를 사용하여 Red Hat Enterprise Linux 9에서
IdM 설치, 구성, 관리 및 유지 관리

Red Hat Enterprise Linux 9 Ansible을 사용하여 Identity Management 설치 및 관리

Ansible Core 및 ansible-freeipa 패키지를 사용하여 Red Hat Enterprise Linux 9에서 IdM 설치, 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/Using_Ansible_to_install_and_manage_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 Ansible 플레이북을 사용하여 Red Hat Enterprise Linux 9에서 Identity Management를 설치, 구성, 관리 및 유지 관리하는 방법에 대한 지침을 제공합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체	7
RED HAT 문서에 관한 피드백 제공	8
1장. ANSIBLE 용어	9
2장. ANSIBLE 플레이북을 사용하여 IDENTITY MANAGEMENT 서버 설치	10
2.1. ANSIBLE 및 IDM 설치를 위한 이점	10
IdM을 설치하기 위해 Ansible을 사용할 때의 이점	10
2.2. ANSIBLE-FREEIPA 패키지 설치	10
2.3. 파일 시스템의 ANSIBLE 역할 위치	11
2.4. 통합 DNS 및 통합 CA를 루트 CA로 사용한 배포에 대한 매개변수 설정	12
2.5. 외부 DNS 및 통합 CA를 루트 CA로 사용한 배포에 대한 매개변수 설정	14
2.6. ANSIBLE 플레이북을 사용하여 통합 CA를 루트 CA로 IDM 서버 배포	16
2.7. 통합 DNS 및 외부 CA를 루트 CA로 사용하여 배포에 대한 매개변수 설정	18
2.8. 외부 DNS 및 외부 CA를 사용하여 배포에 대한 매개변수 설정	21
2.9. ANSIBLE 플레이북을 사용하여 외부 CA를 루트 CA로 IDM 서버 배포	23
2.10. 추가 리소스	24
3장. ANSIBLE 플레이북을 사용하여 IDENTITY MANAGEMENT 복제본 설치	26
3.1. IDM 복제본 설치를 위한 기본, 서버 및 클라이언트 변수 지정	26
3.2. ANSIBLE 플레이북을 사용하여 IDM 복제본을 설치하기 위한 자격 증명 지정	29
3.3. ANSIBLE 플레이북을 사용하여 IDM 복제본 배포	31
4장. ANSIBLE 플레이북을 사용하여 IDENTITY MANAGEMENT 클라이언트 설치	32
4.1. 자동 검색 클라이언트 설치 모드에 대한 인벤토리 파일의 매개변수 설정	32
4.2. 클라이언트 설치 중 자동 검색을 수행할 수 없는 경우 인벤토리 파일의 매개변수 설정	34
4.3. INSTALL-CLIENT.YML 파일에서 매개변수 확인	36
4.4. ANSIBLE PLAYBOOK을 사용하여 IDM 클라이언트 등록에 대한 권한 부여 옵션	37
4.5. ANSIBLE 플레이북을 사용하여 IDM 클라이언트 배포	38
4.6. ANSIBLE 설치 후 IDENTITY MANAGEMENT 클라이언트 테스트	39
4.7. ANSIBLE 플레이북을 사용하여 IDM 클라이언트 설치 제거	39
5장. ANSIBLE 플레이북을 사용하여 IDM을 관리하기 위한 환경 준비	41
6장. ANSIBLE 플레이북을 사용하여 글로벌 IDM 설정 구성	43
6.1. ANSIBLE 플레이북을 사용하여 IDM 구성 검색	43
6.2. ANSIBLE 플레이북을 사용하여 IDM CA 갱신 서버 구성	45
6.3. ANSIBLE 플레이북을 사용하여 IDM 사용자의 기본 셸 구성	46
6.4. 추가 리소스	48
7장. ANSIBLE 플레이북을 사용하여 사용자 계정 관리	49
7.1. 사용자 라이프 사이클	49
7.2. ANSIBLE 플레이북을 사용하여 IDM 사용자가 있는지 확인	51
7.3. ANSIBLE 플레이북을 사용하여 여러 IDM 사용자가 있는지 확인	53
7.4. ANSIBLE 플레이북을 사용하여 JSON 파일에서 여러 IDM 사용자가 있는지 확인	55
7.5. ANSIBLE 플레이북을 사용하여 사용자가 없는지 확인	57
7.6. 추가 리소스	59
8장. ANSIBLE 플레이북을 사용하여 사용자 그룹 관리	60
8.1. IDM의 다양한 그룹 유형	60
8.2. 직접 및 간접 그룹 멤버	62
8.3. ANSIBLE 플레이북을 사용하여 IDM 그룹 및 그룹 멤버가 있는지 확인	63
8.4. ANSIBLE 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 있는지 확인	65

8.5. ANSIBLE 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 없는지 확인	67
9장. ANSIBLE을 사용하여 IDM의 그룹 멤버십 자동화	69
9.1. ANSIBLE을 사용하여 IDM 사용자 그룹에 대한 AUTOMEMBER 규칙이 있는지 확인합니다.	69
9.2. ANSIBLE을 사용하여 IDM 사용자 그룹 AUTOMEMBER 규칙에 지정된 조건이 있는지 확인	71
9.3. ANSIBLE을 사용하여 IDM 사용자 그룹 AUTOMEMBER 규칙에 조건이 없는지 확인합니다.	74
9.4. ANSIBLE을 사용하여 IDM 사용자 그룹의 AUTOMEMBER 규칙이 없는지 확인합니다.	77
9.5. ANSIBLE을 사용하여 IDM 호스트 그룹 AUTOMEMBER 규칙에 조건이 있는지 확인합니다.	79
10장. ANSIBLE 플레이북을 사용하여 IDM에서 셀프 서비스 규칙 관리	83
10.1. IDM의 셀프 서비스 액세스 제어	83
10.2. ANSIBLE을 사용하여 셀프 서비스 규칙이 있는지 확인합니다.	84
10.3. ANSIBLE을 사용하여 셀프 서비스 규칙이 없는지 확인	86
10.4. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 속성이 있는지 확인합니다.	88
10.5. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 속성이 없는지 확인합니다.	90
11장. ANSIBLE 플레이북을 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한 위임	93
11.1. 위임 규칙	93
11.2. IDM용 ANSIBLE 인벤토리 파일 생성	93
11.3. ANSIBLE을 사용하여 위임 규칙이 있는지 확인합니다.	95
11.4. ANSIBLE을 사용하여 위임 규칙이 없는지 확인합니다.	97
11.5. ANSIBLE을 사용하여 위임 규칙에 특정 속성이 있는지 확인	99
11.6. ANSIBLE을 사용하여 위임 규칙에 특정 속성이 없는지 확인	102
12장. ANSIBLE 플레이북을 사용하여 IDM에서 역할 기반 액세스 제어 관리	105
12.1. IDM의 권한	106
12.2. 기본 관리 권한	107
12.3. IDM의 권한	110
12.4. IDM의 역할	110
12.5. ID 관리에서 사전 정의된 역할	111
12.6. ANSIBLE을 사용하여 권한이 있는 IDM RBAC 역할이 있는지 확인	111
12.7. ANSIBLE을 사용하여 IDM RBAC 역할이 없는지 확인	114
12.8. ANSIBLE을 사용하여 IDM RBAC 역할에 사용자 그룹이 할당되었는지 확인	116
12.9. ANSIBLE을 사용하여 특정 사용자가 IDM RBAC 역할에 할당되지 않았는지 확인	118
12.10. ANSIBLE을 사용하여 서비스가 IDM RBAC 역할의 멤버인지 확인	121
12.11. ANSIBLE을 사용하여 호스트가 IDM RBAC 역할의 멤버인지 확인	123
12.12. ANSIBLE을 사용하여 호스트 그룹이 IDM RBAC 역할의 멤버인지 확인	125
13장. ANSIBLE 플레이북을 사용하여 RBAC 권한 관리	128
13.1. ANSIBLE을 사용하여 사용자 정의 IDM RBAC 권한이 있는지 확인	128
13.2. ANSIBLE을 사용하여 사용자 정의 IDM RBAC 권한에 멤버 권한이 있는지 확인	130
13.3. ANSIBLE을 사용하여 IDM RBAC 권한에 권한이 포함되어 있지 않은지 확인	133
13.4. ANSIBLE을 사용하여 사용자 정의 IDM RBAC 권한 이름 변경	135
13.5. ANSIBLE을 사용하여 IDM RBAC 권한이 없는지 확인	137
13.6. 추가 리소스	139
14장. ANSIBLE 플레이북을 사용하여 IDM에서 RBAC 권한 관리	140
14.1. ANSIBLE을 사용하여 RBAC 권한이 있는지 확인	140
14.2. ANSIBLE을 사용하여 특성이 있는 RBAC 권한이 있는지 확인합니다.	143
14.3. ANSIBLE을 사용하여 RBAC 권한이 없는지 확인	146
14.4. ANSIBLE을 사용하여 속성이 IDM RBAC 권한의 멤버인지 확인	147
14.5. ANSIBLE을 사용하여 속성이 IDM RBAC 권한의 멤버가 아닌지 확인	149
14.6. ANSIBLE을 사용하여 IDM RBAC 권한 이름 변경	152
14.7. 추가 리소스	153

15장. ANSIBLE을 사용하여 IDM의 복제 토폴로지 관리	155
15.1. ANSIBLE을 사용하여 IDM에 복제 계약이 있는지 확인	155
15.2. ANSIBLE을 사용하여 여러 IDM 복제본 간에 복제 계약이 존재하는지 확인	158
15.3. ANSIBLE을 사용하여 두 개의 복제본 간에 복제 계약이 있는지 확인합니다.	160
15.4. ANSIBLE을 사용하여 IDM에 토폴로지 접미사가 있는지 확인합니다.	163
15.5. ANSIBLE을 사용하여 IDM 복제본 다시 초기화	165
15.6. ANSIBLE을 사용하여 IDM에 복제 주석이 없는지 확인	168
15.7. 추가 리소스	170
16장. ANSIBLE을 사용하여 IDM 서버 관리	171
16.1. ANSIBLE을 사용하여 IDM 서버가 있는지 확인	171
16.2. ANSIBLE을 사용하여 IDM 서버가 IDM 토폴로지에서 없음을 확인	173
16.3. 마지막 IDM 서버 역할을 호스팅해도 IDM 서버가 없는지 확인	175
16.4. IDM 서버가 없음을 확인하지만 다른 IDM 서버와 분리할 필요는 없습니다.	178
16.5. ANSIBLE 플레이북을 사용하여 기존 IDM 서버가 숨겨져 있는지 확인합니다.	180
16.6. ANSIBLE 플레이북을 사용하여 기존 IDM 서버가 표시되는지 확인합니다.	182
16.7. 기존 IDM 서버에 IDM DNS 위치가 할당되어 있는지 확인합니다.	184
16.8. 기존 IDM 서버에 IDM DNS 위치가 할당되지 않았는지 확인합니다.	187
17장. ANSIBLE 플레이북을 사용하여 호스트 관리	190
17.1. ANSIBLE 플레이북을 사용하여 FQDN으로 IDM 호스트 항목이 있는지 확인	190
17.2. ANSIBLE 플레이북을 사용하여 DNS 정보로 IDM 호스트 항목이 있는지 확인	192
17.3. ANSIBLE 플레이북을 사용하여 임의의 암호로 여러 IDM 호스트 항목이 있는지 확인	194
17.4. ANSIBLE 플레이북을 사용하여 여러 IP 주소로 IDM 호스트 항목이 있는지 확인	196
17.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 항목이 없는지 확인	199
17.6. 추가 리소스	200
18장. ANSIBLE 플레이북을 사용하여 호스트 그룹 관리	201
18.1. IDM의 호스트 그룹	201
18.2. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹이 있는지 확인	202
18.3. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 호스트가 있는지 확인	204
18.4. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹 중첩	206
18.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 멤버 관리자가 있는지 확인	207
18.6. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 호스트가 없는지 확인	209
18.7. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 중첩 호스트 그룹이 없는지 확인	211
18.8. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹이 없는지 확인	213
18.9. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 멤버 관리자가 없는지 확인	215
19장. IDM 암호 정책 정의	217
19.1. 암호 정책이란 무엇입니까?	217
19.2. IDM의 암호 정책	217
19.3. ANSIBLE 플레이북을 사용하여 IDM에 암호 정책이 있는지 확인	219
19.4. IDM의 추가 암호 정책 옵션	221
19.5. IDM 그룹에 추가 암호 정책 옵션 적용	222
20장. IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여	226
20.1. IDM 클라이언트의 SUDO 액세스	226
20.2. CLI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여	226
20.3. IDM 웹 UI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여	229
20.4. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 CLI에 SUDO 규칙 생성	233
20.5. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 IDM WEBUI에 SUDO 규칙 생성	236
20.6. IDM 클라이언트에서 SUDO에 대한 GSSAPI 인증 활성화	243
20.7. IDM 클라이언트에서 SUDO에 대한 GSSAPI 인증 활성화 및 적용	246
20.8. PAM 서비스의 GSSAPI 인증을 제어하는 SSSD 옵션	249

20.9. SUDO용 GSSAPI 인증 문제 해결	251
20.10. ANSIBLE 플레이북을 사용하여 IDM 클라이언트의 IDM 사용자에게 대한 SUDO 액세스 권한 확인	253
21장. ANSIBLE 플레이북을 사용하여 IDM에 호스트 기반 액세스 제어 규칙이 있는지 확인	257
21.1. IDM의 호스트 기반 액세스 제어 규칙	257
21.2. ANSIBLE 플레이북을 사용하여 IDM에 HBAC 규칙이 있는지 확인	257
22장. IDM의 자격 증명 모음	260
22.1. 자격 증명 모음 및 이점	260
22.2. VAULT 소유자, 멤버 및 관리자	261
22.3. 표준, 대칭 및 SYMMETRIC 자격 증명 모음	263
22.4. 사용자, 서비스 및 공유 자격 증명 모음	263
22.5. VAULT 컨테이너	264
22.6. 기본 IDM 자격 증명 모음 명령	264
22.7. IDM에 키 복구 기관 설치	265
23장. ANSIBLE을 사용하여 IDM 사용자 자격 증명 모음 관리: 보안 저장 및 검색	267
23.1. ANSIBLE을 사용하여 IDM에 표준 사용자 자격 증명 모음 확인	267
23.2. ANSIBLE을 사용하여 IDM의 표준 사용자 자격 증명 모음에 보안 보관	269
23.3. ANSIBLE을 사용하여 IDM의 표준 사용자 자격 증명 모음에서 시크릿 검색	271
24장. ANSIBLE을 사용하여 IDM 서비스 자격 증명 모음 관리: 보안 저장 및 검색	275
24.1. ANSIBLE을 사용하여 IDM에 SYMMETRIC 서비스 자격 증명 모음이 있는지 확인	276
24.2. ANSIBLE을 사용하여 SYMMETRIC 자격 증명 모음에 멤버 서비스 추가	279
24.3. ANSIBLE을 사용하여 SYMMETRIC 자격 증명 모음에 IDM 서비스 시크릿 저장	281
24.4. ANSIBLE을 사용하여 IDM 서비스의 서비스 시크릿 검색	283
24.5. ANSIBLE을 사용하여 손상된 경우 IDM 서비스 자격 증명 모음 시크릿 변경	286
24.6. 추가 리소스	291
25장. ANSIBLE을 사용하여 IDM에서 서비스 존재 및 서비스 존재 확인	292
25.1. ANSIBLE 플레이북을 사용하여 IDM에 HTTP 서비스가 있는지 확인	293
25.2. ANSIBLE 플레이북을 사용하여 비IDM 클라이언트에서 IDM에 HTTP 서비스가 있는지 확인	295
25.3. ANSIBLE 플레이북을 사용하여 DNS 없이 IDM 클라이언트에 HTTP 서비스가 있는지 확인	297
25.4. ANSIBLE 플레이북을 사용하여 IDM 서비스 항목에 외부 서명 인증서가 있는지 확인	299
25.5. ANSIBLE 플레이북을 사용하여 IDM 사용자, 그룹, 호스트 또는 호스트 그룹을 통해 서비스의 키탭을 생성할 수 있습니다.	302
25.6. ANSIBLE 플레이북을 사용하여 IDM 사용자, 그룹, 호스트 또는 호스트 그룹이 서비스의 키탭 검색 가능	305
25.7. ANSIBLE 플레이북을 사용하여 서비스의 KERBEROS 주체 별칭이 있는지 확인	308
25.8. ANSIBLE 플레이북을 사용하여 IDM에 HTTP 서비스가 없는지 확인	311
25.9. 추가 리소스	312
26장. ANSIBLE 플레이북을 사용하여 IDM의 글로벌 DNS 구성 관리	313
26.1. NETWORKMANAGER에서 /ETC/RESOLV.CONF의 글로벌 전달자가 제거되지 않도록 하는 방법	314
26.2. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인	315
26.3. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 없는지 확인	317
26.4. IPADNSCONFIG ANSIBLE-FREEIPA 모듈의 ACTION: MEMBER 옵션	319
26.5. IDM의 DNS 전달 정책	321
26.6. ANSIBLE 플레이북을 사용하여 IDM DNS 글로벌 구성에 첫 번째 정책이 설정되어 있는지 확인합니다.	322
26.7. ANSIBLE 플레이북을 사용하여 IDM DNS에서 글로벌 전달자가 비활성화되었는지 확인	324
26.8. ANSIBLE 플레이북을 사용하여 IDM DNS에서 정방향 및 역방향 조회 영역의 동기화가 비활성화되었는지 확인합니다.	326
27장. ANSIBLE 플레이북을 사용하여 IDM DNS 영역 관리	329
27.1. 지원되는 DNS 영역 유형	329
27.2. 기본 IDM DNS 영역의 구성 속성	331

27.3. ANSIBLE을 사용하여 IDM DNS에 기본 영역 생성	333
27.4. ANSIBLE 플레이북을 사용하여 IDM에 여러 변수가 있는 기본 DNS 영역이 있는지 확인	335
27.5. ANSIBLE 플레이북을 사용하여 IP 주소가 지정될 때 역방향 DNS 조회 영역이 있는지 확인	338
28장. ANSIBLE을 사용하여 IDM의 DNS 위치 관리	341
28.1. DNS 기반 서비스 검색	341
28.2. DNS 위치에 대한 배포 고려 사항	343
28.3. DNS 시간 (TTL)	343
28.4. ANSIBLE을 사용하여 IDM 위치 확인	343
28.5. ANSIBLE을 사용하여 IDM 위치 없음	345
28.6. 추가 리소스	347
29장. IDM에서 DNS 전달 관리	348
29.1. IDM DNS 서버의 두 가지 역할	349
29.2. IDM의 DNS 전달 정책	349
29.3. IDM 웹 UI에 글로벌 전달자 추가	350
29.4. CLI에 글로벌 전달자 추가	353
29.5. IDM 웹 UI에 DNS FORWARD ZONE 추가	354
29.6. CLI에서 DNS FORWARD ZONE 추가	358
29.7. ANSIBLE을 사용하여 IDM에서 DNS GLOBAL FORWARDER 설정	359
29.8. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인	361
29.9. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 없는지 확인	363
29.10. ANSIBLE을 사용하여 IDM에서 DNS GLOBAL FORWARDERS가 비활성화되었는지 확인	366
29.11. ANSIBLE을 사용하여 IDM에 DNS FORWARD ZONE이 있는지 확인	367
29.12. ANSIBLE을 사용하여 IDM에 DNS FORWARD ZONE이 여러 전달자가 있는지 확인	369
29.13. ANSIBLE을 사용하여 IDM에서 DNS FORWARD ZONE이 비활성화되었는지 확인	372
29.14. ANSIBLE을 사용하여 IDM의 DNS 전달 영역이 없는지 확인	374
29.15. 추가 리소스	376
30장. ANSIBLE을 사용하여 IDM의 DNS 레코드 관리	377
30.1. IDM의 DNS 레코드	377
30.2. 일반적인 IPA DNSRECORD-* 옵션	378
30.3. ANSIBLE을 사용하여 IDM에 A 및 AAAA DNS 레코드가 있는지 확인	381
30.4. ANSIBLE을 사용하여 IDM에 A 및 PTR DNS 레코드가 있는지 확인	383
30.5. ANSIBLE을 사용하여 IDM에 여러 DNS 레코드가 있는지 확인	386
30.6. ANSIBLE을 사용하여 IDM에 여러 CNAME 레코드가 있는지 확인	388
30.7. ANSIBLE을 사용하여 IDM에 SRV 레코드가 있는지 확인	391

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.

ID 관리에서 계획된 용어 교체는 다음과 같습니다.

- **블록 목록** 대체 **블랙리스트**
- **허용 목록** 대체 **허용 목록**
- **보조** 대체 **슬레이브**
- 컨텍스트에 따라 *master* 라는 단어가 보다 정확한 언어로 교체될 것입니다.
 - **IdM 서버** 대체 *IdM 마스터*
 - **CA 갱신 서버**는 *CA 갱신 마스터*로 대체
 - **CRL 게시자 서버**는 *CRL master*를 대체합니다.
 - **multi-supplier** 는 *멀티 마스터* 교체

RED HAT 문서에 관한 피드백 제공

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

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

1장. ANSIBLE 용어

이 제목의 장에서는 공식 Ansible 용어를 사용합니다. 용어에 익숙하지 않은 경우 계속하기 전에 [공식 Ansible 업스트림 문서](#)를 읽고 특히 다음 섹션을 참조하십시오.

- [Ansible 섹션의 기본 개념은 Ansible에서](#) 가장 일반적으로 사용되는 개념에 대한 개요를 제공합니다.
- [사용자 가이드에서](#) 는 명령줄 사용, 인벤토리 작업, 데이터와 상호 작용, 작업, 플레이 및 플레이북 작성, 플레이북 실행과 같은 Ansible 사용을 시작할 때 가장 일반적인 상황과 질문에 대해 간단히 설명합니다.
- [인벤토리를 구축하는 방법은 인벤토리](#) 설계 방법에 대한 팁을 제공합니다. 인벤토리는 Ansible이 인프라의 여러 관리 노드 또는 호스트에 대해 작업하는 데 사용하는 목록 또는 그룹입니다.
- [플레이북](#) 소개에는 구성 관리, 머신 배포 및 복잡한 애플리케이션 배포를 위한 반복 가능하고 재구축 가능한 시스템으로 Ansible 플레이북의 개념이 도입되었습니다.
- [Ansible roles](#) 섹션에서는 알려진 파일 구조를 기반으로 변수, 작업 및 핸들러 로드를 자동화하는 방법을 설명합니다.
- [용어집](#) 은 Ansible 설명서의 다른 위치에서 사용되는 용어를 설명합니다.

2장. ANSIBLE 플레이북을 사용하여 IDENTITY MANAGEMENT 서버 설치

다음 섹션에서는 **Ansible** 을 사용하여 시스템을 IdM 서버로 구성하는 방법을 설명합니다. 시스템을 IdM 서버로 구성하면 IdM 도메인이 설정되고 시스템에서 IdM 서비스를 IdM 클라이언트에 제공할 수 있습니다. **ipaserver** Ansible 역할을 사용하여 배포를 관리할 수 있습니다.

사전 요구 사항

- **Ansible** 및 IdM 개념을 이해할 수 있습니다.
 - Ansible 역할
 - Ansible 노트
 - Ansible 인벤토리
 - Ansible 작업
 - Ansible 모듈
 - Ansible 플레이 및 플레이북

2.1. ANSIBLE 및 IDM 설치를 위한 이점

Ansible은 시스템 구성, 소프트웨어 배포 및 롤링 업데이트를 수행하는 데 사용되는 자동화 툴입니다. Ansible에는 IdM(Identity Management) 지원이 포함되며, Ansible 모듈을 사용하여 IdM 서버, 복제본, 클라이언트 또는 전체 IdM 토폴로지 설정과 같은 설치 작업을 자동화할 수 있습니다.

IdM을 설치하기 위해 **Ansible**을 사용할 때의 이점

다음 목록은 수동 설치와 달리 Ansible을 사용하여 Identity Management를 설치하는 이점이 있습니다.

- 관리 노드에 로그인할 필요가 없습니다.
- 개별적으로 배포할 각 호스트의 설정을 구성할 필요가 없습니다. 대신 전체 클러스터를 배포하는 데 하나의 인벤토리 파일이 있을 수 있습니다.
- 나중에 관리 작업을 위해 인벤토리 파일을 재사용할 수 있습니다(예: 사용자 및 호스트 추가). IdM 과 관련이 없는 것과 같은 작업에도 인벤토리 파일을 재사용할 수 있습니다.

2.2. ANSIBLE-FREEIPA 패키지 설치

이 섹션에서는 IdM(Identity Management)을 설치 및 관리하기 위해 Ansible 역할 및 모듈을 제공하는 **ansible-freeipa** 패키지를 설치하는 방법을 설명합니다.

사전 요구 사항

- 관리형 노드에서 다음을 수행합니다.
 - 관리형 노드가 고정 IP 주소 및 작동 중인 패키지 관리자가 있는 Red Hat Enterprise Linux 8 시스템인지 확인합니다.
- 컨트롤러 의 경우:

- 컨트롤러가 유효한 서브스크립션이 있는 Red Hat Enterprise Linux 시스템인지 확인합니다. 그렇지 않은 경우 대체 설치 지침은 공식 Ansible 설명서 [설치 가이드](#)를 참조하십시오.
- 컨트롤러에서 **SSH** 프로토콜을 통해 관리형 노드에 연결할 수 있는지 확인합니다. 관리형 노드가 컨트롤러의 **/root/.ssh/known_hosts** 파일에 나열되어 있는지 확인합니다.

절차

Ansible 컨트롤러에서 다음 절차를 사용합니다.

1. 시스템이 RHEL 8.5 및 이전 버전에서 실행 중인 경우 필요한 리포지토리를 활성화합니다.

```
# subscription-manager repos --enable ansible-2.8-for-rhel-8-x86_64-rpms
```

2. 시스템이 RHEL 8.5 이하에서 실행 중인 경우 **ansible** 패키지를 설치합니다.

```
# dnf install ansible
```

3. **ansible-freeipa** 패키지를 설치합니다.

```
# dnf install ansible-freeipa
```

역할 및 모듈은 **/usr/share/ansible/roles/** 및 **/usr/share/ansible/plugins/modules** 디렉터리에 설치됩니다.

2.3. 파일 시스템의 ANSIBLE 역할 위치

기본적으로 **ansible-freeipa** 역할은 **/usr/share/ansible/roles/** 디렉터리에 설치됩니다. **ansible-freeipa** 패키지의 구조는 다음과 같습니다.

- **/usr/share/ansible/roles/** 디렉터리는 Ansible 컨트롤러에 **ipaserver**, **ipareplica** 및 **ipaclient** 역할을 저장합니다. 각 역할 디렉터리에는 **README.md** Markdown 파일의 역할에 대한 예제, 기본 개요, 라이선스 및 문서가 저장됩니다.

```
[root@server]# ls -l /usr/share/ansible/roles/
ipaclient
ipareplica
ipaserver
```

- **/usr/share/doc/ansible-freeipa/** 디렉터리는 **README.md** Markdown 파일에 개별 역할 및 토폴로지에 대한 문서를 저장합니다. 또한 **playbooks/** 하위 디렉터리를 저장합니다.

```
[root@server]# ls -l /usr/share/doc/ansible-freeipa/
playbooks
README-client.md
README.md
README-replica.md
README-server.md
README-topology.md
```

- **/usr/share/doc/ansible-freeipa/playbooks/** 디렉터리는 예제 플레이북을 저장합니다.

```
[root@server]# ls -l /usr/share/doc/ansible-freeipa/playbooks/
install-client.yml
```

```
install-cluster.yml
install-replica.yml
install-server.yml
uninstall-client.yml
uninstall-cluster.yml
uninstall-replica.yml
uninstall-server.yml
```

2.4. 통합 DNS 및 통합 CA를 루트 CA로 사용한 배포에 대한 매개변수 설정

IdM 통합 DNS 솔루션을 사용하는 환경에서 통합 CA를 루트 CA로 설치하기 위해 IdM 서버를 설치하기 위한 인벤토리 파일을 구성하려면 이 절차를 완료합니다.



참고

이 절차의 인벤토리에는 **INI** 형식을 사용합니다. 또는 **YAML** 또는 **JSON** 형식을 사용할 수 있습니다.

절차

1. 편집할 인벤토리 파일을 엽니다. IdM 서버로 사용하려는 호스트의 정규화된 도메인 이름(**FQDN**)을 지정합니다. **FQDN** 이 다음 기준을 충족하는지 확인합니다.
 - 영숫자 및 하이픈(-)만 허용됩니다. 예를 들어 밑줄은 허용되지 않으며 DNS 오류가 발생할 수 있습니다.
 - 호스트 이름은 모두 소문자여야 합니다.
2. IdM 도메인 및 영역 정보를 지정합니다.
3. 다음 옵션을 추가하여 통합 DNS를 사용할 것을 지정합니다.

```
ipaserver_setup_dns=yes
```

4. DNS 전달 설정을 지정합니다. 다음 옵션 중 하나를 선택합니다.
 - 설치 프로그램이 **/etc/resolv.conf** 파일에서 forwarders를 사용하도록 하려면 **ipaserver_auto_forwarders=yes** 옵션을 사용합니다. **/etc/resolv.conf** 파일에 지정된 이름 서버가 localhost 127.0.0.1 주소이거나 가상 사설 네트워크에 있고 사용 중인 DNS 서버는 일반적으로 공용 인터넷에서 연결할 수 없는 경우 이 옵션을 사용하지 마십시오.
 - **ipaserver_forwarders** 옵션을 사용하여 전달자를 수동으로 지정합니다. 설치 프로세스에서는 설치된 IdM 서버의 **/etc/named.conf** 파일에 전달자 IP 주소를 추가합니다.
 - **ipaserver_no_forwarders=yes** 옵션을 사용하여 대신 사용할 루트 DNS 서버를 구성합니다.



참고

DNS 전달자가 없으면 환경이 격리되고 인프라의 다른 DNS 도메인의 이름은 확인되지 않습니다.

5. DNS 역방향 레코드 및 영역 설정을 지정합니다. 다음 옵션 중에서 선택합니다.

- **ipaserver_allow_zone_overlap=yes** 옵션을 사용하여 영역이 이미 확인 가능한 경우에도 (reverse) 영역을 생성할 수 있습니다.
- **ipaserver_reverse_zones** 옵션을 사용하여 역방향 영역을 수동으로 지정합니다.
- 설치 프로그램이 역방향 DNS 영역을 생성하지 않도록 **ipaserver_no_reverse=yes** 옵션을 사용합니다.



참고

IdM을 사용하여 역방향 영역을 관리하는 것은 선택 사항입니다. 대신 외부 DNS 서비스를 사용할 수 있습니다.

6. **admin** 및 **Directory Manager**의 암호를 지정합니다. Ansible Vault를 사용하여 암호를 저장하고 플레이북 파일에서 Vault 파일을 참조합니다. 또는 덜 안전한 경우 인벤토리 파일에서 직접 암호를 지정합니다.
7. (선택 사항) IdM 서버에서 사용할 사용자 지정 **firewalld** 영역을 지정합니다. 사용자 지정 영역을 설정하지 않으면 IdM에서 서비스를 기본 **firewalld** 영역에 추가합니다. 사전 정의된 기본 영역은 **public**입니다.



중요

지정된 **firewalld** 영역이 있어야 하며 영구적으로 있어야 합니다.

필수 서버 정보가 있는 인벤토리 파일의 예(암호 제외)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
[...]
```

필수 서버 정보가 있는 인벤토리 파일의 예(암호 포함)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

사용자 지정 **firewalld** 영역이 있는 인벤토리 파일의 예

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

Ansible Vault 파일에 저장된 **admin** 및 **Directory Manager** 암호를 사용하여 **IdM** 서버를 설정하는 플레이북의 예

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true
  vars_files:
    - playbook_sensitive_data.yml

  roles:
    - role: ipaserver
      state: present
```

인벤토리 파일에서 **admin** 및 **Directory Manager** 암호를 사용하여 **IdM** 서버를 설정하는 플레이북의 예

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true

  roles:
    - role: ipaserver
      state: present
```

추가 리소스

- 전달 정책 기본 설정은 **ipa-dns-install(1)** 매뉴얼 페이지의 **--forward-policy** 설명을 참조하십시오.
- ipaserver** 역할에서 사용하는 DNS 변수에 대한 자세한 내용은 **/usr/share/doc/ansible-freeipa** 디렉터리의 **README-server.md** 파일의 DNS 변수 섹션을 참조하십시오.

2.5. 외부 DNS 및 통합 CA를 루트 CA로 사용한 배포에 대한 매개변수 설정

외부 DNS 솔루션을 사용하는 환경에서 통합 CA를 루트 CA로 설치하기 위해 IdM 서버를 설치하기 위한 인벤토리 파일을 구성하려면 이 절차를 완료합니다.

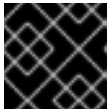


참고

이 절차의 인벤토리 파일은 **INI** 형식을 사용합니다. 또는 **YAML** 또는 **JSON** 형식을 사용할 수 있습니다.

절차

1. 편집할 인벤토리 파일을 엽니다. IdM 서버로 사용하려는 호스트의 정규화된 도메인 이름(**FQDN**)을 지정합니다. **FQDN** 이 다음 기준을 충족하는지 확인합니다.
 - 영숫자 및 하이픈(-)만 허용됩니다. 예를 들어 밑줄은 허용되지 않으며 DNS 오류가 발생할 수 있습니다.
 - 호스트 이름은 모두 소문자여야 합니다.
2. IdM 도메인 및 영역 정보를 지정합니다.
3. **ipaserver_setup_dns** 옵션이 **no** 로 설정되었거나 존재하지 않는지 확인합니다.
4. **admin** 및 **Directory Manager** 의 암호를 지정합니다. Ansible Vault를 사용하여 암호를 저장하고 플레이북 파일에서 Vault 파일을 참조합니다. 또는 덜 안전한 경우 인벤토리 파일에서 직접 암호를 지정합니다.
5. (선택 사항) IdM 서버에서 사용할 사용자 지정 **firewalld** 영역을 지정합니다. 사용자 지정 영역을 설정하지 않으면 IdM에서 서비스를 기본 **firewalld** 영역에 추가합니다. 사전 정의된 기본 영역은 **public** 입니다.



중요

지정된 **firewalld** 영역이 있어야 하며 영구적으로 있어야 합니다.

필수 서버 정보가 있는 인벤토리 파일의 예(암호 제외)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

필수 서버 정보가 있는 인벤토리 파일의 예(암호 포함)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
[...]
```

사용자 지정 **firewalld** 영역이 있는 인벤토리 파일의 예

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

Ansible Vault 파일에 저장된 **admin** 및 **Directory Manager** 암호를 사용하여 **IdM** 서버를 설정하는 플레이북의 예

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true
  vars_files:
    - playbook_sensitive_data.yml

  roles:
    - role: ipaserver
      state: present
```

인벤토리 파일에서 **admin** 및 **Directory Manager** 암호를 사용하여 **IdM** 서버를 설정하는 플레이북의 예

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true

  roles:
    - role: ipaserver
      state: present
```

2.6. ANSIBLE 플레이북을 사용하여 통합 CA를 루트 CA로 IDM 서버 배포

Ansible 플레이북을 사용하여 통합 인증 기관(CA)을 루트 CA로 사용하여 IdM 서버를 배포하려면 이 절차를 완료합니다.



참고

이 절차의 인벤토리에는 **INI** 형식을 사용합니다. 또는 **YAML** 또는 **JSON** 형식을 사용할 수 있습니다.

사전 요구 사항

- 다음 절차 중 하나를 선택하여 시나리오에 해당하는 매개변수를 설정했습니다.

• [투하 DNS 저지](#)

- 통합 DNS 절차
- 외부 DNS가 있는 프로세스

절차

1. 플레이북 파일 이름(예: **install-server.yml**)을 사용하여 **ansible-playbook** 명령을 실행합니다. **-i** 옵션으로 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i <path_to_inventory_directory>/hosts
<path_to_playbooks_directory>/install-server.yml
```

v, **-vv** 또는 **-vvv** 옵션을 사용하여 상세 정보 표시 수준을 지정합니다.

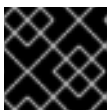
CLI(명령줄 인터페이스)에서 Ansible 플레이북 스크립트의 출력을 볼 수 있습니다. 다음 출력은 스크립트가 0개 작업으로 성공적으로 실행된 것을 보여줍니다.

```
PLAY RECAP
server.idm.example.com : ok=18  changed=10  unreachable=0  failed=0  skipped=21
rescued=0  ignored=0
```

2. 다음 옵션 중 하나를 선택합니다.

- IdM 배포에서 외부 DNS를 사용하는 경우: **/tmp/ipa.system.records.UFRBto.db** 파일에 포함된 **DNS 리소스 레코드**를 기존 외부 DNS 서버에 추가합니다. DNS 레코드를 업데이트하는 프로세스는 특정 DNS 솔루션에 따라 다릅니다.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



중요

기존 DNS 서버에 DNS 레코드를 추가할 때까지 서버 설치가 완료되지 않습니다.

- IdM 배포에서 통합 DNS를 사용하는 경우:
 - 상위 도메인의 DNS 위임을 IdM DNS 도메인에 추가합니다. 예를 들어 IdM DNS 도메인이 **idm.example.com** 인 경우 **example.com** 상위 도메인에 이름 서버(NS) 레코드를 추가합니다.



중요

IdM DNS 서버가 설치되면 이 단계를 반복합니다.

- 시간 서버의 **_ntp._udp** 서비스(SRV) 레코드를 IdM DNS에 추가합니다. IdM DNS에 새로 설치된 IdM 서버의 시간 서버에 SRV 레코드가 있으면 이 기본 IdM 서버에서 사용하는 시간 서버와 동기화되도록 향후 복제본 및 클라이언트 설치가 자동으로 구성됩니다.

추가 리소스

- 외부 CA를 사용하여 IdM 서버를 루트 CA로 배포하는 방법에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 외부 CA를 루트 CA로 IdM 서버 배포](#)를 참조하십시오.

2.7. 통합 DNS 및 외부 CA를 루트 CA로 사용하여 배포에 대한 매개변수 설정

IdM 통합 DNS 솔루션을 사용하는 환경에서 외부 CA를 루트 CA로 설치하기 위한 인벤토리 파일을 구성하려면 이 절차를 완료합니다.



참고

이 절차의 인벤토리 파일은 **INI** 형식을 사용합니다. 또는 **YAML** 또는 **JSON** 형식을 사용할 수 있습니다.

절차

1. 편집할 인벤토리 파일을 엽니다. IdM 서버로 사용하려는 호스트의 정규화된 도메인 이름(**FQDN**)을 지정합니다. **FQDN** 이 다음 기준을 충족하는지 확인합니다.
 - 영숫자 및 하이픈(-)만 허용됩니다. 예를 들어 밑줄은 허용되지 않으며 DNS 오류가 발생할 수 있습니다.
 - 호스트 이름은 모두 소문자여야 합니다.
2. IdM 도메인 및 영역 정보를 지정합니다.
3. 다음 옵션을 추가하여 통합 DNS를 사용할 것을 지정합니다.

ipaserver_setup_dns=yes

4. DNS 전달 설정을 지정합니다. 다음 옵션 중 하나를 선택합니다.
 - 설치 프로세스에서 **/etc/resolv.conf** 파일의 forwarders를 사용하도록 하려면 **ipaserver_auto_forwarders=yes** 옵션을 사용합니다. **/etc/resolv.conf** 파일에 지정된 이름 서버가 localhost 127.0.0.1 주소이거나 가상 프라이빗 네트워크에 있고 사용 중인 DNS 서버는 일반적으로 공용 인터넷에서 연결할 수 없는 경우 이 옵션을 사용하지 않는 것이 좋습니다.
 - **ipaserver_forwarders** 옵션을 사용하여 전달자를 수동으로 지정합니다. 설치 프로세스에서는 설치된 IdM 서버의 **/etc/named.conf** 파일에 전달자 IP 주소를 추가합니다.
 - **ipaserver_no_forwarders=yes** 옵션을 사용하여 대신 사용할 루트 DNS 서버를 구성합니다.



참고

DNS 전달자가 없으면 환경이 격리되고 인프라의 다른 DNS 도메인의 이름은 확인되지 않습니다.

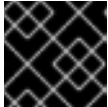
5. DNS 역방향 레코드 및 영역 설정을 지정합니다. 다음 옵션 중에서 선택합니다.
 - **ipaserver_allow_zone_overlap=yes** 옵션을 사용하여 영역이 이미 확인 가능한 경우에도 (reverse) 영역을 생성할 수 있습니다.
 - **ipaserver_reverse_zones** 옵션을 사용하여 역방향 영역을 수동으로 지정합니다.
 - 설치 프로세스가 역방향 DNS 영역을 생성하지 않도록 하려면 **ipaserver_no_reverse=yes** 옵션을 사용합니다.



참고

IdM을 사용하여 역방향 영역을 관리하는 것은 선택 사항입니다. 대신 외부 DNS 서비스를 사용할 수 있습니다.

6. **admin** 및 **Directory Manager**의 암호를 지정합니다. Ansible Vault를 사용하여 암호를 저장하고 플레이북 파일에서 Vault 파일을 참조합니다. 또는 덜 안전한 경우 인벤토리 파일에서 직접 암호를 지정합니다.
7. (선택 사항) IdM 서버에서 사용할 사용자 지정 **firewalld** 영역을 지정합니다. 사용자 지정 영역을 설정하지 않으면 IdM에서 해당 서비스를 기본 **firewalld** 영역에 추가합니다. 사전 정의된 기본 영역은 **public**입니다.



중요

지정된 **firewalld** 영역이 있어야 하며 영구적으로 있어야 합니다.

필수 서버 정보가 있는 인벤토리 파일의 예(암호 제외)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
[...]
```

필수 서버 정보가 있는 인벤토리 파일의 예(암호 포함)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

사용자 지정 **firewalld** 영역이 있는 인벤토리 파일의 예

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
```

```
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

```
[...]
```

8. 설치의 첫 번째 단계에 대한 플레이북을 생성합니다. CSR(인증서 서명 요청)을 생성하고 컨트롤러에서 관리 노드로 복사하는 지침을 입력합니다.

```
---
- name: Playbook to configure IPA server Step 1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_ca: yes

  roles:
  - role: ipaserver
    state: present

  post_tasks:
  - name: Copy CSR /root/ipa.csr from node to "{{ groups.ipaserver[0] + '-ipa.csr' }}"
    fetch:
      src: /root/ipa.csr
      dest: "{{ groups.ipaserver[0] + '-ipa.csr' }}"
      flat: yes
```

9. 설치의 마지막 단계에 대해 다른 플레이북을 생성합니다.

```
---
- name: Playbook to configure IPA server Step -1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_cert_files: "/root/chain.crt"

  pre_tasks:
  - name: Copy "{{ groups.ipaserver[0] + '-chain.crt' }}" to /root/chain.crt on node
    copy:
      src: "{{ groups.ipaserver[0] + '-chain.crt' }}"
      dest: "/root/chain.crt"
      force: yes

  roles:
  - role: ipaserver
    state: present
```

추가 리소스

- 전달 정책 기본 설정은 **ipa-dns-install(1)** 매뉴얼 페이지의 **--forward-policy** 설명을 참조하십시오.
- **ipaserver** 역할에서 사용하는 DNS 변수에 대한 자세한 내용은 **/usr/share/doc/ansible-freeipa** 디렉터리의 **README-server.md** 파일의 DNS 변수 섹션을 참조하십시오.

2.8. 외부 DNS 및 외부 CA를 사용하여 배포에 대한 매개변수 설정

외부 DNS 솔루션을 사용하는 환경의 루트 CA로 외부 CA를 사용하여 IdM 서버를 설치하기 위한 인벤토리 파일을 구성하려면 다음 절차를 완료합니다.

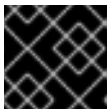


참고

이 절차의 인벤토리 파일은 **INI** 형식을 사용합니다. 또는 **YAML** 또는 **JSON** 형식을 사용할 수 있습니다.

절차

1. 편집할 인벤토리 파일을 엽니다. IdM 서버로 사용하려는 호스트의 정규화된 도메인 이름(**FQDN**)을 지정합니다. **FQDN** 이 다음 기준을 충족하는지 확인합니다.
 - 영숫자 및 하이픈(-)만 허용됩니다. 예를 들어 밑줄은 허용되지 않으며 DNS 오류가 발생할 수 있습니다.
 - 호스트 이름은 모두 소문자여야 합니다.
2. IdM 도메인 및 영역 정보를 지정합니다.
3. **ipaserver_setup_dns** 옵션이 **no** 로 설정되었거나 존재하지 않는지 확인합니다.
4. **admin** 및 **Directory Manager** 의 암호를 지정합니다. Ansible Vault를 사용하여 암호를 저장하고 플레이북 파일에서 Vault 파일을 참조합니다. 또는 덜 안전한 경우 인벤토리 파일에서 직접 암호를 지정합니다.
5. (선택 사항) IdM 서버에서 사용할 사용자 지정 **firewalld** 영역을 지정합니다. 사용자 지정 영역을 설정하지 않으면 IdM에서 서비스를 기본 **firewalld** 영역에 추가합니다. 사전 정의된 기본 영역은 **public** 입니다.



중요

지정된 **firewalld** 영역이 있어야 하며 영구적으로 있어야 합니다.

필수 서버 정보가 있는 인벤토리 파일의 예(암호 제외)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

필수 서버 정보가 있는 인벤토리 파일의 예(암호 포함)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

사용자 지정 **firewalld** 영역이 있는 인벤토리 파일의 예

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone

[...]
```

- 설치의 첫 번째 단계에 대한 플레이북을 생성합니다. CSR(인증서 서명 요청)을 생성하고 컨트롤러에서 관리 노드로 복사하는 지침을 입력합니다.

```
---
- name: Playbook to configure IPA server Step 1
  hosts: ipaserver
  become: true
  vars_files:
    - playbook_sensitive_data.yml
  vars:
    ipaserver_external_ca: yes

  roles:
    - role: ipaserver
      state: present

  post_tasks:
    - name: Copy CSR /root/ipa.csr from node to "{{ groups.ipaserver[0] + '-ipa.csr' }}"
      fetch:
        src: /root/ipa.csr
        dest: "{{ groups.ipaserver[0] + '-ipa.csr' }}"
        flat: yes
```

- 설치의 마지막 단계에 대해 다른 플레이북을 생성합니다.

```
---
- name: Playbook to configure IPA server Step -1
```

```

hosts: ipaserver
become: true
vars_files:
- playbook_sensitive_data.yml
vars:
  ipaserver_external_cert_files: "/root/chain.crt"

pre_tasks:
- name: Copy "{{ groups.ipaserver[0] + '-chain.crt' }}" to /root/chain.crt on node
  copy:
    src: "{{ groups.ipaserver[0] + '-chain.crt' }}"
    dest: "/root/chain.crt"
    force: yes

roles:
- role: ipaserver
  state: present

```

추가 리소스

- 외부 DNS 및 외부 서명된 CA를 사용하여 IdM 서버를 설치할 때 사용할 수 있는 옵션에 대한 자세한 내용은 [통합 DNS 사용과 함께 외부 CA를 루트 CA로 사용하여](#) 설치를 참조하십시오.

2.9. ANSIBLE 플레이북을 사용하여 외부 CA를 루트 CA로 IDM 서버 배포

Ansible 플레이북을 사용하여 외부 CA(인증 기관)를 루트 CA로 사용하여 IdM 서버를 배포하려면 이 절차를 완료합니다.



참고

이 절차의 인벤토리 파일은 **INI** 형식을 사용합니다. 또는 **YAML** 또는 **JSON** 형식을 사용할 수 있습니다.

사전 요구 사항

- 다음 절차 중 하나를 선택하여 시나리오에 해당하는 매개변수를 설정했습니다.
 - [통합 DNS 절차](#)
 - [외부 DNS가 있는 프로세스](#)

절차

1. 설치의 첫 번째 단계에 대한 지침이 포함된 플레이북 파일의 이름으로 **ansible-playbook** 명령을 실행합니다(예: **install-server-step1.yml**). **-i** 옵션으로 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i <path_to_inventory_directory>/host.server
<path_to_playbooks_directory>/install-server-step1.yml
```

v, **-vv** 또는 **-vvv** 옵션을 사용하여 상세 정보 표시 수준을 지정합니다.

CLI(명령줄 인터페이스)에서 Ansible 플레이북 스크립트의 출력을 볼 수 있습니다. 다음 출력은 스크립트가 0개 작업으로 성공적으로 실행된 것을 보여줍니다.

PLAY RECAP

```
server.idm.example.com : ok=18  changed=10  unreachable=0  failed=0  skipped=21
rescued=0  ignored=0
```

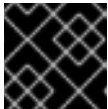
- 컨트롤러에서 **ipa.csr** 인증서 서명 요청 파일을 찾아 외부 CA에 제출합니다.
- 다음 단계의 플레이북에서 찾을 수 있도록 외부 CA가 컨트롤러 파일 시스템에 서명한 IdM CA 인증서를 배치합니다.
- 설치의 최종 단계(예: **install-server-step2.yml**)에 대한 지침이 포함된 플레이북 파일의 이름으로 **ansible-playbook** 명령을 실행합니다. **-i** 옵션으로 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i <path_to_inventory_directory>/host.server
<path_to_playbooks_directory>/install-server-step2.yml
```

- 다음 옵션 중 하나를 선택합니다.

- IdM 배포에서 외부 DNS를 사용하는 경우: **/tmp/ipa.system.recordss.UFRPto.db** 파일에 포함된 **DNS 리소스 레코드**를 기존 외부 DNS 서버에 추가합니다. DNS 레코드를 업데이트하는 프로세스는 특정 DNS 솔루션에 따라 다릅니다.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```

**중요**

기존 DNS 서버에 DNS 레코드를 추가할 때까지 서버 설치가 완료되지 않습니다.

- IdM 배포에서 통합 DNS를 사용하는 경우:
 - 상위 도메인의 DNS 위임을 IdM DNS 도메인에 추가합니다. 예를 들어 IdM DNS 도메인이 **idm.example.com**인 경우 **example.com** 상위 도메인에 이름 서버(NS) 레코드를 추가합니다.

**중요**

IdM DNS 서버가 설치되면 이 단계를 반복합니다.

- 시간 서버의 **_ntp._udp** 서비스(SRV) 레코드를 IdM DNS에 추가합니다. IdM DNS에 새로 설치된 IdM 서버의 시간 서버에 SRV 레코드가 있으면 이 기본 IdM 서버에서 사용하는 시간 서버와 동기화되도록 향후 복제본 및 클라이언트 설치가 자동으로 구성됩니다.

추가 리소스

통합 CA를 사용하여 IdM 서버를 루트 CA로 배포하는 방법에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 통합 CA가 있는 IdM 서버 배포](#)를 참조하십시오.

2.10. 추가 리소스

- 인벤토리 기본 사항: 형식, 호스트 및 그룹
- IdM 서버 설치에 필요한 샘플 Ansible 플레이북 및 **ansible-freeipa** 업스트림 설명서에서 사용 가능한 변수 목록을 확인할 수 있습니다.

3장. ANSIBLE 플레이북을 사용하여 IDENTITY MANAGEMENT 복제본 설치

Ansible 을 사용하여 시스템을 IdM 복제본으로 구성하여 IdM 도메인에 등록하고, 시스템이 도메인의 IdM 서버에서 IdM 서비스를 사용할 수 있도록 합니다.

배포는 **ipareplica** Ansible 역할에 의해 관리됩니다. 이 역할은 자동 검색 모드를 사용하여 IdM 서버, 도메인 및 기타 설정을 식별할 수 있습니다. 그러나 다양한 복제본 그룹이 서로 다른 시간에 배포되는 경우 계층형 모델에 여러 복제본을 배포하는 경우 각 그룹의 특정 서버 또는 복제본을 정의해야 합니다.

사전 요구 사항

- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- **Ansible** 및 IdM 개념을 이해할 수 있습니다.
 - Ansible 역할
 - Ansible 노드
 - Ansible 인벤토리
 - Ansible 작업
 - Ansible 모듈
 - Ansible 플레이 및 플레이북

3.1. IDM 복제본 설치를 위한 기본, 서버 및 클라이언트 변수 지정

IdM 복제본을 설치하기 위한 인벤토리 파일을 구성하려면 이 절차를 완료합니다.

사전 요구 사항

- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.

절차

1. 편집할 인벤토리 파일을 엽니다. IdM 복제본이 될 호스트의 FQDN(정규화된 도메인 이름)을 지정합니다. FQDN은 유효한 DNS 이름이어야 합니다.
 - 숫자, 영문자 및 하이픈(-)만 허용됩니다. 예를 들어 밑줄은 허용되지 않으며 DNS 오류가 발생할 수 있습니다.
 - 호스트 이름은 모두 소문자여야 합니다.

정의된 복제본의 **FQDN**만 있는 간단한 인벤토리 호스트 파일의 예

```
[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]
```

IdM 서버가 이미 배포되어 있고 IdM DNS 영역에 SRV 레코드가 올바르게 설정된 경우 스크립트에서 다른 모든 필수 값을 자동으로 검색합니다.

- 필요한 경우 다음 시나리오 중 가장 가까운 인벤토리 파일에 추가 정보를 제공하십시오.

시나리오 1

자동 검색을 방지하고 **[ipareplicas]** 섹션에 나열된 모든 복제본이 있는 경우 인벤토리 파일의 **[ipaservers]** 섹션에 서버를 설정합니다.

정의된 IdM 서버 및 복제본의 **FQDN**이 있는 인벤토리 호스트 파일의 예

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]
```

시나리오 2

또는 자동 검색을 방지하되 특정 서버로 특정 복제본을 배포하려는 경우 인벤토리 파일의 **[ipareplicas]** 섹션에서 특정 복제본의 서버를 개별적으로 설정합니다.

특정 복제본에 대해 정의된 특정 IdM 서버가 있는 인벤토리 파일의 예

```
[ipaservers]
server.idm.example.com
replica1.idm.example.com

[ipareplicas]
replica2.idm.example.com
replica3.idm.example.com ipareplica_servers=replica1.idm.example.com
```

위의 예에서 **replica3.idm.example.com** 은 이미 배포된 **replica1.idm.example.com** 을 해당 복제 소스로 사용합니다.

시나리오 3

하나의 배치에 여러 복제본을 배포하고 시간이 우려되는 경우 다중 계층 복제본 배포가 유용할 수 있습니다. 인벤토리 파일에 특정 복제본 그룹(예: **[ipareplicas_tier1]** 및 **[ipareplicas_tier2]**)을 정의하고 **install-replica.yml** 플레이북의 각 그룹에 대해 별도의 플레이를 설계합니다.

복제본 계층이 정의된 인벤토리 파일 예

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

[ipareplicas_tier1]
replica1.idm.example.com
```

```
[ipareplicas_tier2]
replica2.idm.example.com \
ipareplica_servers=replica1.idm.example.com,server.idm.example.com
```

ipareplica_servers의 첫 번째 항목이 사용됩니다. 두 번째 항목은 대체 옵션으로 사용됩니다. IdM 복제본을 배포하기 위해 여러 계층을 사용하는 경우 플레이북에서 별도의 작업을 통해 먼저 tier1에서 복제본을 배포한 다음 tier2에서 복제본을 복제해야 합니다.

다른 복제본 그룹에 대해 다른 플레이가 있는 플레이북 파일의 예

```
---
- name: Playbook to configure IPA replicas (tier1)
  hosts: ipareplicas_tier1
  become: true

  roles:
  - role: ipareplica
    state: present

- name: Playbook to configure IPA replicas (tier2)
  hosts: ipareplicas_tier2
  become: true

  roles:
  - role: ipareplica
    state: present
```

시나리오 4

복제본이 기본 영역 대신 지정된 **firewalld** 영역을 사용하도록 하려면 인벤토리 파일에 지정할 수 있습니다. 예를 들어, 기본적으로 설정된 퍼블릭 영역 대신 IdM 설치에 내부 **firewalld** 영역을 사용하려는 경우 유용할 수 있습니다.

사용자 지정 영역을 설정하지 않으면 IdM에서 서비스를 기본 **firewalld** 영역에 추가합니다. 사전 정의된 기본 영역은 **public**입니다.



중요

지정된 **firewalld** 영역이 있어야 하며 영구적으로 있어야 합니다.

사용자 지정 **firewalld** 영역이 있는 간단한 인벤토리 호스트 파일의 예

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]

[ipareplicas:vars]
ipareplica_firewalld_zone=custom zone
```

시나리오 5

복제본이 IdM DNS 서비스를 호스팅하도록 하려면 **ipareplica_setup_dns=yes** 행을 **[ipareplicas:vars]** 섹션에 추가합니다. 또한 서버당 DNS 전달자를 사용할지 여부를 지정합니다.

- 서버별 전달자를 구성하려면 **ipareplica_forwarders** 변수와 **[ipareplicas:vars]** 섹션에 문자열 목록을 추가합니다(예: **ipareplica_forwarders=192.0.2.1,192.0.2.2**).
- per-server forwarders를 구성하지 않으려면 **[ipareplicas:vars]** 섹션에 다음 행을 추가합니다. **ipareplica_no_forwarders=yes**.
- 복제본의 **/etc/resolv.conf** 파일에 나열된 forwarders에 따라 서버별 전달자를 구성하려면 **ipareplica_auto_forwarders** 변수를 **[ipareplicas:vars]** 섹션에 추가합니다.

복제본에서 **DNS** 및 서버별 전달자를 설정하는 지침이 있는 인벤토리 파일의 예

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]

[ipareplicas:vars]
ipareplica_setup_dns=yes
ipareplica_forwarders=192.0.2.1,192.0.2.2
```

추가 리소스

- **ipareplica** 변수에 대한 자세한 내용은 **/usr/share/ansible/roles/ipareplica/README.md** Markdown 파일을 참조하십시오.

3.2. ANSIBLE 플레이북을 사용하여 IDM 복제본을 설치하기 위한 자격 증명 지정

IdM 복제본 설치에 대한 권한 부여를 구성하려면 이 절차를 완료합니다.

사전 요구 사항

- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.

절차

1. 복제본을 배포할 권한이 있는 사용자의 암호를 지정합니다(예: IdM 관리자).

- Red Hat은 Ansible Vault를 사용하여 암호를 저장하고 플레이북 파일에서 Vault 파일을 참조하는 것이 좋습니다(예: **install-replica.yml**).

Ansible Vault 파일의 인벤토리 파일 및 암호의 주체를 사용하는 플레이북 파일의 예

```
- name: Playbook to configure IPA replicas
hosts: ipareplicas
```

```

become: true
vars_files:
- playbook_sensitive_data.yml

roles:
- role: ipareplica
state: present

```

Ansible Vault를 사용하는 방법에 대한 자세한 내용은 공식 [Ansible Vault](#) 설명서를 참조하십시오.

- 덜 안전한 경우, 인벤토리 파일에서 직접 **admin**의 자격 증명을 제공하십시오. 인벤토리 파일의 **[ipareplicas:vars]** 섹션에서 **ipaadmin_password** 옵션을 사용합니다. 인벤토리 파일과 **install-replica.yml** 플레이북 파일은 다음과 같습니다.

hosts.replica 파일의 예

```

[...]
[ipareplicas:vars]
ipaadmin_password=Secret123

```

인벤토리 파일의 **principal** 및 **password**를 사용하는 플레이북의 예

```

- name: Playbook to configure IPA replicas
  hosts: ipareplicas
  become: true

  roles:
  - role: ipareplica
    state: present

```

- 또는 덜 안전하지 않은 경우 인벤토리 파일에 직접 복제본을 배포할 권한이 있는 다른 사용자의 자격 증명을 제공합니다. 다른 권한 있는 사용자를 지정하려면 사용자 이름에 **ipaadmin_principal** 옵션을 사용하고 암호에 **ipaadmin_password** 옵션을 사용합니다. 인벤토리 파일과 **install-replica.yml** 플레이북 파일은 다음과 같습니다.

hosts.replica 파일의 예

```

[...]
[ipareplicas:vars]
ipaadmin_principal=my_admin
ipaadmin_password=my_admin_secret123

```

인벤토리 파일의 **principal** 및 **password**를 사용하는 플레이북의 예

```

- name: Playbook to configure IPA replicas
  hosts: ipareplicas
  become: true

  roles:
  - role: ipareplica
    state: present

```

추가 리소스

- **ipareplica** Ansible 역할에서 허용하는 옵션에 대한 자세한 내용은 `/usr/share/ansible/roles/ipareplica/README.md` 마크다운 파일을 참조하십시오.

3.3. ANSIBLE 플레이북을 사용하여 IDM 복제본 배포

Ansible 플레이북을 사용하여 IdM 복제본을 배포하려면 이 절차를 완료합니다.

사전 요구 사항

- IdM 복제본을 설치하기 위해 인벤토리 파일을 구성했습니다.
- IdM 복제본 설치를 위한 권한 부여를 구성했습니다.

절차

- Ansible 플레이북을 사용하여 IdM 복제본을 설치하려면 플레이북 파일의 이름과 함께 **ansible-playbook** 명령을 사용합니다(예: **install-replica.yml**). **-i** 옵션으로 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i <path_to_inventory_directory>/hosts.replica
<path_to_playbooks_directory>/install-replica.yml
```

v, **-vv** 또는 **-vvv** 옵션을 사용하여 상세 정보 표시 수준을 지정합니다.

Ansible은 Ansible 플레이북 스크립트 실행에 대해 알려줍니다. 다음 출력은 스크립트가 0개 작업으로 성공적으로 실행된 것을 보여줍니다.

```
PLAY RECAP
 replica.idm.example.com : ok=18  changed=10  unreachable=0  failed=0  skipped=21
 rescued=0  ignored=0
```

이제 IdM 복제본이 설치되어 있습니다.

4장. ANSIBLE 플레이북을 사용하여 IDENTITY MANAGEMENT 클라이언트 설치

다음 섹션에서는 [Ansible](#) 을 사용하여 시스템을 IdM(Identity Management) 클라이언트로 구성하는 방법을 설명합니다. 시스템을 IdM 클라이언트로 구성하여 IdM 도메인에 등록하고, 시스템이 도메인의 IdM 서버에서 IdM 서비스를 사용할 수 있도록 합니다.

배포는 **ipaclient** Ansible 역할에 의해 관리됩니다. 기본적으로 이 역할은 자동 검색 모드를 사용하여 IdM 서버, 도메인 및 기타 설정을 식별합니다. Ansible 플레이북이 인벤토리 파일에서 지정된 설정을 사용하도록 역할을 수정할 수 있습니다.

사전 요구 사항

- Ansible 제어 노드에 [ansible-freeipa](#) 패키지를 설치했습니다.
- [Ansible](#) 및 IdM 개념을 이해할 수 있습니다.
 - Ansible 역할
 - Ansible 노드
 - Ansible 인벤토리
 - Ansible 작업
 - Ansible 모듈
 - Ansible 플레이 및 플레이북

4.1. 자동 검색 클라이언트 설치 모드에 대한 인벤토리 파일의 매개변수 설정

Ansible 플레이북을 사용하여 Identity Management 클라이언트를 설치하려면 인벤토리 파일에서 대상 호스트 매개 변수를 구성합니다(예: **inventory/hosts**).

- 호스트에 대한 정보
- 작업에 대한 권한 부여

인벤토리 파일은 보유한 인벤토리 플러그인에 따라 여러 형식 중 하나일 수 있습니다. **INI와 유사한** 형식은 Ansible의 기본값 중 하나이며 아래 예제에서 사용됩니다.



참고

RHEL의 그래픽 사용자 인터페이스와 함께 스마트 카드를 사용하려면 Ansible 플레이북에 **ipaclient_mkhomedir** 변수를 포함해야 합니다.

사전 요구 사항

- 제어 노드에서 배포 지침을 확인하고 [install-client.yml](#) 파일의 [매개변수 확인](#)을 참조하십시오.

절차

1. IdM 클라이언트가 될 호스트의 정규화된 호스트 이름(FQDN)을 지정합니다. 정규화된 도메인 이름은 유효한 DNS 이름이어야 합니다.

- 숫자, 영문자 및 하이픈(-)만 허용됩니다. 예를 들어 밑줄은 허용되지 않으며 DNS 오류가 발생할 수 있습니다.
- 호스트 이름은 모두 소문자여야 합니다. 대문자는 사용할 수 없습니다.

IdM DNS 영역에 SRV 레코드가 올바르게 설정된 경우 스크립트에서 다른 모든 필수 값을 자동으로 검색합니다.

클라이언트 **FQDN**이 정의된 간단한 인벤토리 호스트 파일의 예

```
[ipaclients]
client.idm.example.com
[...]
```

2. 클라이언트 등록 자격 증명을 지정합니다. 다음 인증 방법을 사용할 수 있습니다.

- 클라이언트를 등록할 권한이 있는 사용자의 암호입니다. 이는 기본 옵션입니다.
 - Ansible Vault를 사용하여 암호를 저장하고 플레이북 파일에서 Vault 파일을 참조하는 것이 좋습니다(예: **install-client.yml**).

Ansible Vault 파일의 인벤토리 파일 및 암호의 주체를 사용하는 플레이북 파일의 예

```
- name: Playbook to configure IPA clients with username/password
  hosts: ipaclients
  become: true
  vars_files:
    - playbook_sensitive_data.yml

  roles:
    - role: ipaclient
      state: present
```

- 더 안전하게 **inventory/hosts** 파일의 **[ipaclients:vars]** 섹션에서 **ipa admin _password** 옵션을 사용하여 admin의 자격 증명을 제공하십시오. 또는 권한이 있는 다른 사용자를 지정하려면 사용자 이름에 **ipaadmin_principal** 옵션을 사용하고 암호에 **ipaadmin_password** 옵션을 사용합니다. **inventory/hosts** 인벤토리 파일 및 **install-client.yml** 플레이북 파일은 다음과 같습니다.

인벤토리 호스트 파일 예

```
[...]
[ipaclients:vars]
ipaadmin_principal=my_admin
ipaadmin_password=Secret123
```

인벤토리 파일에서 보안 주체 및 암호를 사용하는 플레이북의 예

```
- name: Playbook to unconfigure IPA clients
  hosts: ipaclients
  become: true
```

```
roles:
- role: ipaclient
state: true
```

- 이전 등록의 클라이언트 키탭이 계속 사용 가능한 경우.
이 옵션은 시스템이 이전에 ID 관리 클라이언트로 등록된 경우 사용할 수 있습니다. 이 인증 방법을 사용하려면 **#ipaclient_keytab** 옵션의 주석을 제거하여 keytab을 저장하는 파일의 경로를 지정합니다(예: **inventory/hosts**의 **[ipaclient:vars]** 섹션).
- 등록 중에 생성할 임의의 일회용 암호(OTP)입니다. 이 인증 방법을 사용하려면 인벤토리 파일에서 **ipaclient_use_otp=yes** 옵션을 사용합니다. 예를 들어 **inventory/hosts** 파일의 **[ipaclients:vars]** 섹션에서 **ipaclient_use_otp=yes** 옵션의 주석을 달 수 있습니다. OTP를 사용하면 다음 옵션 중 하나를 지정해야 합니다.
 - 예를 들어 **inventory/hosts** 파일의 **[ipaclients:vars]** 섹션에 **ipaadmin_password** 값을 제공하여 클라이언트 등록 권한이 있는 사용자의 암호입니다.
 - 예를 들어 **inventory/hosts**의 **[ipaclients:vars]** 섹션에 **ipaadmin_keytab**의 값을 제공하여 admin keytab.

추가 리소스

- **ipaclient** Ansible 역할에서 허용하는 옵션에 대한 자세한 내용은 **/usr/share/ansible/roles/ipaclient/README.md** README 파일을 참조하십시오.

4.2. 클라이언트 설치 중 자동 검색을 수행할 수 없는 경우 인벤토리 파일의 매개변수 설정

Ansible 플레이북을 사용하여 Identity Management 클라이언트를 설치하려면 인벤토리 파일에서 대상 호스트 매개 변수를 구성합니다(예: **inventory/hosts**).

- 호스트, IdM 서버, IdM 도메인 또는 IdM 영역에 대한 정보
- 작업에 대한 권한 부여

인벤토리 파일은 보유한 인벤토리 플러그인에 따라 여러 형식 중 하나일 수 있습니다. **INI와 유사한** 형식은 Ansible의 기본값 중 하나이며 아래 예제에서 사용됩니다.



참고

RHEL의 그래픽 사용자 인터페이스와 함께 스마트 카드를 사용하려면 Ansible 플레이북에 **ipaclient_mkhome** 변수를 포함해야 합니다.

사전 요구 사항

- 제어 노드에서 배포 지침을 확인하고 **install-client.yml** 파일의 **매개변수 확인**을 참조하십시오.

절차

1. IdM 클라이언트가 될 호스트의 정규화된 호스트 이름(FQDN)을 지정합니다. 정규화된 도메인 이름은 유효한 DNS 이름이어야 합니다.
 - 숫자, 영문자 및 하이픈(-)만 허용됩니다. 예를 들어 밑줄은 허용되지 않으며 DNS 오류가 발생할 수 있습니다.

- 호스트 이름은 모두 소문자여야 합니다. 대문자는 사용할 수 없습니다.

2. **inventory/hosts** 파일의 관련 섹션에 다른 옵션을 지정합니다.

- 클라이언트가 등록할 IdM 서버를 나타내는 **[ipaservers]** 섹션에 있는 서버의 FQDN
- 다음 두 가지 옵션 중 하나입니다.
 - 클라이언트가 등록할 IdM 서버의 DNS 도메인 이름을 나타내는 **[ipaclients:vars]** 섹션의 **ipaclient_domain** 옵션
 - **[ipaclients:vars]** 섹션의 **ipaclient_realm** 옵션으로 IdM 서버에서 제어하는 Kerberos 영역의 이름을 나타냅니다.

클라이언트 **FQDN**, 서버 **FQDN** 및 정의된 도메인을 사용하는 인벤토리 호스트 파일의 예

```
[ipaclients]
client.idm.example.com

[ipaservers]
server.idm.example.com

[ipaclients:vars]
ipaclient_domain=idm.example.com
[...]
```

3. 클라이언트 등록 자격 증명을 지정합니다. 다음 인증 방법을 사용할 수 있습니다.

- 클라이언트를 등록할 권한이 있는 사용자의 암호입니다. 이는 기본 옵션입니다.
 - Red Hat은 Ansible Vault를 사용하여 암호를 저장하고 플레이북 파일에서 Vault 파일을 참조하는 것이 좋습니다(예: **install-client.yml**). 직접: Ansible Vault 파일의 보안 주체 및 암호의 보안 주체를 사용하는 **.examples** 플레이북 파일입니다.

```
- name: Playbook to configure IPA clients with username/password
hosts: ipaclients
become: true
vars_files:
- *playbook_sensitive_data.yml*
```

```
roles:
- role: ipaclient
state: present
```

- 더 안전하게 **inventory/hosts** 파일의 **[ipaclients:vars]** 섹션에서 **ipa admin _password** 옵션을 사용하여 admin의 자격 증명을 제공하십시오. 또는 권한이 있는 다른 사용자를 지정하려면 사용자 이름에 **ipaadmin_principal** 옵션을 사용하고 암호에 **ipaadmin_password** 옵션을 사용합니다. **install-client.yml** Playbook 파일은 다음과 같이 표시될 수 있습니다.

인벤토리 호스트 파일 예

```
[...]
[ipaclients:vars]
ipaadmin_principal=my_admin
ipaadmin_password=Secret123
```

인벤토리 파일에서 보안 주체 및 암호를 사용하는 플레이북의 예

```
- name: Playbook to unconfigure IPA clients
  hosts: ipaclients
  become: true

  roles:
  - role: ipaclient
    state: true
```

- 이전 등록의 클라이언트 키탭이 계속 사용 가능한 경우:
이 옵션은 시스템이 이전에 ID 관리 클라이언트로 등록된 경우 사용할 수 있습니다. 이 인증 방법을 사용하려면 **ipaclient_keytab** 옵션의 주석을 제거하여 keytab을 저장하는 파일의 경로를 지정합니다(예: **inventory/hosts**의 **[ipaclient:vars]** 섹션).
- 등록 중에 생성할 임의의 일회용 암호 (OTP)입니다. 이 인증 방법을 사용하려면 인벤토리 파일에서 **ipaclient_use_otp=yes** 옵션을 사용합니다. 예를 들어 **inventory/hosts** 파일의 **[ipaclients:vars]** 섹션에서 **#ipaclient_use_otp=yes** 옵션의 주석을 달 수 있습니다. OTP를 사용하면 다음 옵션 중 하나를 지정해야 합니다.
 - 예를 들어 **inventory/hosts** 파일의 **[ipaclients:vars]** 섹션에 **ipaadmin_password** 값을 제공하여 클라이언트 등록 권한이 있는 사용자의 암호입니다.
 - 예를 들어 **inventory/hosts**의 **[ipaclients:vars]** 섹션에 **ipaadmin_keytab**의 값을 제공하여 admin keytab.

추가 리소스

- **ipaclient** Ansible 역할에서 허용하는 옵션에 대한 자세한 내용은 **/usr/share/ansible/roles/ipaclient/README.md** README 파일을 참조하십시오.

4.3. INSTALL-CLIENT.YML 파일에서 매개변수 확인

install-client.yml 플레이북 파일에는 IdM 클라이언트 배포에 대한 지침이 포함되어 있습니다.

절차

- 파일을 열고 플레이북의 지침이 배포 계획과 일치하는지 확인합니다. 일반적으로 내용은 다음과 같습니다.

```
---
- name: Playbook to configure IPA clients with username/password
  hosts: ipaclients
  become: true

  roles:
  - role: ipaclient
    state: present
```

개별 항목이 나타내는 것은 다음과 같습니다.

- hosts 항목은 ansible 스크립트가 **ipa-client-install** 스크립트를 실행할 호스트의 **FQDN**을 검색하는 **inventory/hosts** 파일의 섹션을 지정합니다.

- **become: true** 항목은 **ipa-client-install** 스크립트를 실행하는 동안 root의 자격 증명이 호출 되도록 지정합니다.
- **role: ipaclient** 항목은 호스트에 설치할 역할을 지정합니다. 이 경우 ipa 클라이언트 역할입니다.
- **state: present** 항목은 클라이언트를 설치 제거하는 대신 설치하도록 지정합니다(없음 없음).

4.4. ANSIBLE PLAYBOOK을 사용하여 IDM 클라이언트 등록에 대한 권한 부여 옵션

이 참조 섹션에서는 IdM 클라이언트 등록에 대한 개별 권한 부여 옵션에 인벤토리 및 플레이북 파일 예제를 제공합니다.

표 4.1. Ansible을 사용하여 IdM 클라이언트 등록에 대한 권한 부여 옵션

권한 부여 옵션	참고	인벤토리 파일 예	install-client.yml 플레이북 파일 예
클라이언트 등록 권한이 있는 사용자의 암호: 옵션 1	Ansible 자격 증명 모음에 저장된 암호	<pre>[ipaclients:vars] [...]</pre>	<pre>- name: Playbook to configure IPA clients with username/password hosts: ipaclients become: true vars_files: - playbook_sensitive_data.yml roles: - role: ipaclient state: present</pre>
클라이언트 등록 권한이 있는 사용자의 암호: 옵션 2	인벤토리 파일에 저장된 암호	<pre>[ipaclients:vars] ipaadmin_password=Secret123</pre>	<pre>- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true</pre>
random, one-time password (OTP): 옵션 1	OTP + 관리자 암호	<pre>[ipaclients:vars] ipaadmin_password=Secret123 ipaclient_use_otp=yes</pre>	<pre>- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true</pre>

권한 부여 옵션	참고	인벤토리 파일 예	install-client.yml 플레이북 파일 예
random, one-time password (OTP): 옵션 2	OTP + 관리자 키 탭	<pre>[ipaclients:vars] ipaadmin_keytab=/tmp/admin.keytab ipaclient_use_otp=yes</pre>	<pre>- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true</pre>
이전 등록의 클라이언트 키 탭		<pre>[ipaclients:vars] ipaclient_keytab=/tmp/krb5.keytab</pre>	<pre>- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true</pre>

4.5. ANSIBLE 플레이북을 사용하여 IDM 클라이언트 배포

IdM 환경에 IdM 클라이언트를 배포하는 Ansible 플레이북을 사용하려면 이 절차를 완료합니다.

사전 요구 사항

- IdM 클라이언트 배포의 매개변수가 배포 시나리오에 일치하도록 설정되어 있습니다.
 - 자동 검색 클라이언트 설치 모드에 대한 인벤토리 파일의 매개변수 설정
 - 클라이언트 설치 중 자동 검색을 수행할 수 없는 경우 인벤토리 파일의 매개변수 설정
- `install-client.yml`에서 매개변수를 확인했습니다.

절차

- Ansible 플레이북을 사용하여 IdM 클라이언트를 설치하려면 플레이북 파일의 이름과 함께 **ansible-playbook** 명령을 사용합니다(예: **install-client.yml**). **-i** 옵션으로 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i inventory/hosts install-client.yml
```

`v`, `-vv` 또는 `-vvv` 옵션을 사용하여 상세 정보 표시 수준을 지정합니다.

Ansible은 Ansible 플레이북 스크립트 실행에 대해 알려줍니다. 다음 출력은 작업이 실패하지 않았기 때문에 스크립트가 성공적으로 실행되었음을 보여줍니다.

PLAY RECAP

client1.idm.example.com : ok=18 changed=10 unreachable=0 failed=0 skipped=21
rescued=0 ignored=0



참고

Ansible은 다양한 색상을 사용하여 실행 중인 프로세스에 대한 다양한 유형의 정보를 제공합니다. `/etc/ansible/ansible.cfg` 파일의 **[colors]** 섹션에서 기본 색상을 수정할 수 있습니다.

```
[colors]
[...]
#error = red
#debug = dark gray
#deprecate = purple
#skip = cyan
#unreachable = red
#ok = green
#changed = yellow
[...]
```

이제 Ansible 플레이북을 사용하여 호스트에 IdM 클라이언트를 설치했습니다.

4.6. ANSIBLE 설치 후 IDENTITY MANAGEMENT 클라이언트 테스트

CLI(명령줄 인터페이스)는 **ansible-playbook** 명령이 성공했지만 자체 테스트를 수행할 수도 있음을 알려줍니다.

Identity Management 클라이언트에서 서버에 정의된 사용자에 대한 정보를 얻을 수 있는지 테스트하려면 서버에 정의된 사용자를 확인할 수 있는지 확인합니다. 예를 들어 기본 **admin** 사용자를 확인하려면 다음을 실행합니다.

```
[user@client1 ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

인증이 올바르게 작동하는지 테스트하려면 **su -** 가 이미 존재하는 다른 IdM 사용자로 작동합니다.

```
[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 ~]$
```

4.7. ANSIBLE 플레이북을 사용하여 IDM 클라이언트 설치 제거

Ansible 플레이북을 사용하여 IdM 클라이언트로 호스트를 제거하려면 이 절차를 완료합니다.

사전 요구 사항

- IdM 관리자 자격 증명.

절차

- IdM 클라이언트를 설치 제거하려면 플레이북 파일의 이름과 함께 **ansible-playbook** 명령을 사용합니다(예: **uninstall-client.yml**). **-i** 옵션으로 인벤토리 파일을 지정하고 선택적으로 **-v**, **-vv** 또는 **-vvv** 옵션을 사용하여 세부 정보 표시 수준을 지정합니다.

\$ ansible-playbook -v -i inventory/hosts uninstall-client.yml

중요

클라이언트를 다시 설치하기로 결정하는 경우 클라이언트의 기본 IdM 구성만 호스트에서 제거하지만 호스트의 구성 파일은 그대로 둡니다. 또한 설치 제거에는 다음과 같은 제한 사항이 있습니다.

- IdM LDAP 서버에서 클라이언트 호스트 항목을 제거하지 않습니다. 설치 제거는 호스트 등록을 취소만 합니다.
- IdM에서 클라이언트에 있는 서비스는 제거되지 않습니다.
- IdM 서버에서 클라이언트의 DNS 항목을 제거하지 않습니다.
- **/etc/krb5.keytab** 이 아닌 키탭의 이전 주체는 제거되지 않습니다.

제거 제거는 IdM CA에서 호스트에 대해 실행된 모든 인증서를 제거합니다.

추가 리소스

- 호스트 및 IdM 환경 모두에서 IdM 클라이언트 구성을 완전히 제거하는 방법에 대한 자세한 내용은 [IdM 클라이언트 설치 제거](#)를 위한 수동 절차를 참조하십시오.

5장. ANSIBLE 플레이북을 사용하여 IDM을 관리하기 위한 환경 준비

Red Hat Ansible Engine으로 작업할 때 IdM(Identity Management)을 관리하는 시스템 관리자는 다음을 수행하는 것이 좋습니다.

- 홈 디렉터리에서 Ansible 플레이북 전용 하위 디렉터리를 만듭니다(예: `~/MyPlaybooks`).
- `/usr/share/doc/ansible-freeipa/*` 및 `/usr/share/doc/rhel-system-roles/*` 디렉터리 및 하위 디렉터리에서 샘플 Ansible 플레이북을 `~/MyPlaybooks` 디렉터리에 복사 및 조정합니다.
- `~/MyPlaybooks` 디렉터리에 인벤토리 파일을 포함합니다.

이 방법을 사용하면 모든 플레이북을 한 곳에서 찾을 수 있으며 root 권한을 호출하지 않고도 플레이북을 실행할 수 있습니다.



참고

`ipaserver`, `ipareplica`, `ipaclient` 및 `ipabackup ansible-freeipa` 역할을 실행하려면 관리형 노드에서 **root** 권한만 있으면 됩니다. 이러한 역할을 수행하려면 디렉터리 및 **dnf** 소프트웨어 패키지 관리자에 대한 권한 있는 액세스 권한이 필요합니다.

이 섹션에서는 `~/MyPlaybooks` 디렉터리를 생성하고 이를 사용하여 Ansible 플레이북을 저장하고 실행하는 방법을 설명합니다.

사전 요구 사항

- 관리 노드, `server.idm.example.com` 및 `replica.idm.example.com`에 IdM 서버를 설치했습니다.
- 제어 노드에서 직접 관리 노드 `server.idm.example.com` 및 `replica.idm.example.com`에 로그인할 수 있도록 DNS 및 네트워킹을 구성했습니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. 홈 디렉터리에 Ansible 구성 및 플레이북의 디렉터리를 생성합니다.

```
$ mkdir ~/MyPlaybooks/
```

2. `~/MyPlaybooks/` 디렉터리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

3. 다음 콘텐츠를 사용하여 `~/MyPlaybooks/ansible.cfg` 파일을 생성합니다.

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

[privilege_escalation]
become=True
```

4. 다음 콘텐츠를 사용하여 `~/MyPlaybooks/inventory` 파일을 만듭니다.

```
[eu]
```

```
server.idm.example.com
```

```
[us]
replica.idm.example.com
```

```
[ipaserver:children]
eu
us
```

이 구성은 해당 위치에 있는 호스트에 대해 **eu** 및 **us**의 두 개의 호스트 그룹을 정의합니다. 또한 이 구성은 **eu** 및 **us** 그룹의 모든 호스트가 포함된 **ipaserver** 호스트 그룹을 정의합니다.

5. [선택 사항] SSH 공개 및 개인 키를 생성합니다. 테스트 환경에서 액세스를 단순화하려면 개인 키에 암호를 설정하지 마십시오.

```
$ ssh-keygen
```

6. SSH 공개 키를 각 관리 노드의 IdM 관리자 계정에 복사합니다.

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

이러한 명령을 실행하려면 IdM 관리자 암호를 입력해야 합니다.

추가 리소스

- [Ansible 플레이북을 사용하여 ID 관리 서버 설치를 참조하십시오.](#)
- [인벤토리 빌드 방법을 참조하십시오.](#)

6장. ANSIBLE 플레이북을 사용하여 글로벌 IDM 설정 구성

Ansible **config** 모듈을 사용하면 IdM(Identity Management)에 대한 글로벌 구성 매개변수를 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 사용자의 기본 셸 구성](#)

6.1. 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. 인벤토리 파일(예: **inventory.file**)을 생성하고 **[ipaserver]** 섹션에서 IdM 구성을 검색할 IdM 서버를 정의합니다. 예를 들어, **server.idm.example.com** 에서 데이터를 검색하도록 Ansible에 지시하려면 다음을 입력합니다.

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

2. 편집할 **/usr/share/doc/ansible-freeipa/playbooks/config/retrieve-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:
        ipadmin_password: Secret123
        register: serverconfig

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

3. 다음을 변경하여 파일을 조정합니다.

- IdM 관리자의 암호입니다.

- 필요한 경우 기타 값입니다.
4. 파일을 저장합니다.
 5. 플레이북 파일과 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
  }
}
```

6.2. ANSIBLE 플레이북을 사용하여 IDM CA 갱신 서버 구성

CA(인증 기관)를 사용하는 IdM(Identity Management) 배포에서 CA 갱신 서버는 IdM 시스템 인증서를 유지 관리하고 갱신합니다. 이를 통해 강력한 IdM 배포를 수행할 수 있습니다.

IdM CA 갱신 서버의 역할에 대한 자세한 내용은 [IdM CA 갱신 서버 사용](#)을 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 `ipaserver` 를 이 파일에 정의합니다.

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

3. 편집할 `/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:
        ipadmin_password: SomeADMINpassword
        ca_renewal_master_server: carenewal.idm.example.com
```

4. 파일을 변경하여 수정합니다.

- `ipadmin_password` 변수에서 설정한 IdM 관리자의 암호입니다.
- `ca_renewal_master_server` 변수에서 설정한 CA 갱신 서버의 이름입니다.

5. 파일을 저장합니다.

6. Ansible 플레이북을 실행합니다. 플레이북 파일 및 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 ID를 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.

6.3. ANSIBLE 플레이북을 사용하여 IDM 사용자의 기본 셸 구성

셸은 명령을 수락 및 해석하는 프로그램입니다. RHEL(Red Hat Enterprise Linux), **bash**, **ksh**, **zsh**, **zsh**, **fish** 등 다양한 셸을 사용할 수 있습니다. **Bash**, **/bin/bash** 는 대부분의 **Linux** 시스템에서 널리 사용되는 셸이며, **RHEL**의 사용자 계정의 기본 셸입니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 사용자의 기본 셸로 대체 셸을 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 **ipaserver** 를 이 파일에 정의합니다.

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

3. 편집할 **/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** 변수에서 설정한 **IdM** 관리자의 암호입니다.
- 기본 **shell** 변수로 설정한 **IdM** 사용자의 기본 셸은 **/bin/sh** 로 설정합니다.

5.

파일을 저장합니다.

6.

Ansible 플레이북을 실행합니다. 플레이북 파일 및 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 사용합니다.

6.4. 추가 리소스

- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-config.md** 를 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/config` 디렉터리에서 샘플 플레이북을 참조하십시오.

7장. ANSIBLE 플레이북을 사용하여 사용자 계정 관리

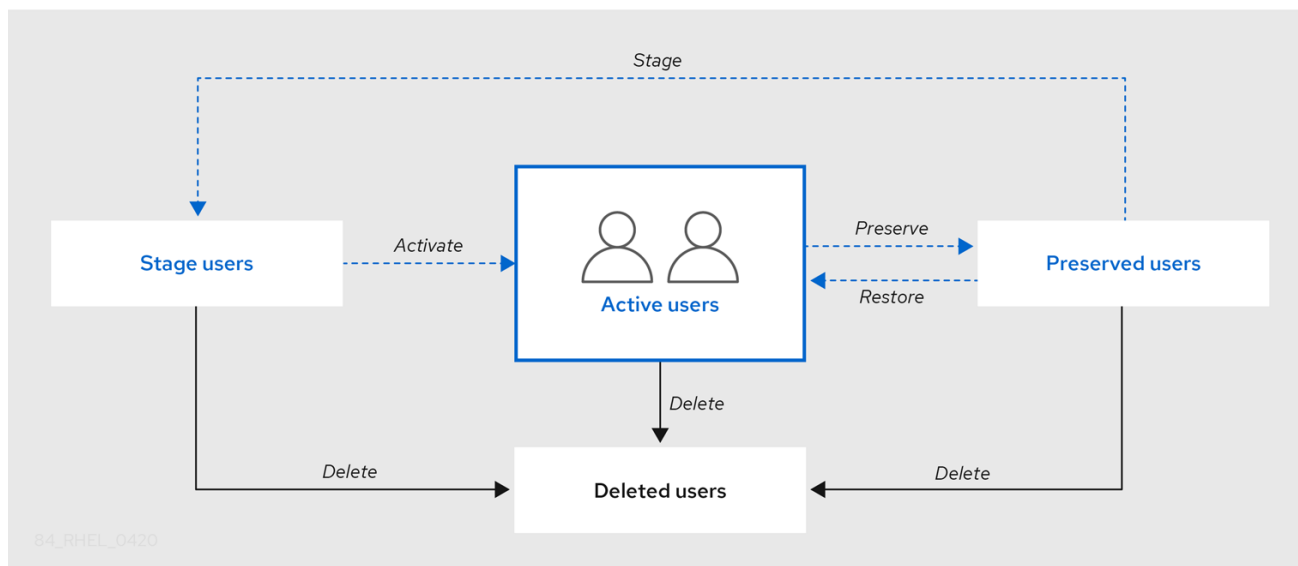
Ansible 플레이북을 사용하여 **IdM**에서 사용자를 관리할 수 있습니다. [사용자 라이프사이클](#) 을 제공한 후 이 장에서는 다음 작업에 **Ansible** 플레이북을 사용하는 방법을 설명합니다.

- **YML** 파일에 직접 나열된 [단일 사용자가](#) 있는지 확인합니다.
- **YML** 파일에 직접 나열된 [여러 사용자가](#) 있는지 확인합니다.
- **JSON** 파일에 나열된 [여러 사용자가](#) **YML** 파일에서 참조되는지 확인합니다.
- **YML** 파일에 직접 나열된 [사용자가 없는지](#) 확인합니다.

7.1. 사용자 라이프 사이클

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

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



IdM 데이터베이스에서 사용자 항목을 영구적으로 삭제할 수 있습니다.



중요

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

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



주의

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



주의

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

7.2. ANSIBLE 플레이북을 사용하여 IDM 사용자가 있는지 확인

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

사전 요구 사항

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

절차

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

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

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

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

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

```

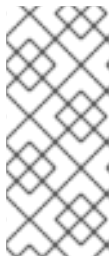
first: Alice
last: Acme
uid: 1000111
gid: 10011
phone: "+555123457"
email: idm_user@acme.com
passwordexpiration: "2023-01-19 23:59:59"
password: "Password123"
update_password: on_create

```

다음 옵션을 사용하여 사용자를 추가해야 합니다.

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

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



참고

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

3. 플레이북을 실행합니다.

```

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

```

검증 단계

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

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

2. 관리자를 위한 **Kerberos** 티켓을 요청합니다.

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

3. *idm_user*에 대한 정보 요청 :

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

idm_user 라는 사용자는 **IdM**에 있습니다.

7.3. ANSIBLE 플레이북을 사용하여 여러 IDM 사용자가 있는지 확인

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

사전 요구 사항

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

절차

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

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

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

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

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

  tasks:
  - name: Create user idm_users
    ipauser:
      ipaadmin_password: MySecret123
      users:
      - name: idm_user_1
        first: Alice
        last: Acme
        uid: 10001
        gid: 10011
        phone: "+555123457"
        email: idm_user@acme.com
        passwordexpiration: "2023-01-19 23:59:59"
        password: "Password123"
      - name: idm_user_2
        first: Bob
        last: Acme
        uid: 100011
        gid: 10011
      - name: idm_user_3
        first: Eve
        last: Acme
        uid: 1000111
        gid: 10011
```

참고

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

3.

플레이북을 실행합니다.

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

검증 단계

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

1. 관리자 권한으로 **ipaserver**에 로그인합니다.

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

2. **idm_user_1**에 대한 정보를 표시합니다.

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

idm_user_1이라는 사용자는 **IdM**에 있습니다.

7.4. ANSIBLE 플레이북을 사용하여 JSON 파일에서 여러 IDENTITY 사용자 확인

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

사전 요구 사항

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

절차

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

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

- 2.

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

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

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

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

3.

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

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

4.

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

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

검증 단계

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

1. 관리자 권한으로 **ipaserver**에 로그인합니다.

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

2. **idm_user_1**에 대한 정보를 표시합니다.

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

idm_user_1이라는 사용자는 **IdM**에 있습니다.

7.5. ANSIBLE 플레이북을 사용하여 사용자가 없는지 확인

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

사전 요구 사항

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

절차

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

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

2.

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

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

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

3.

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

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

검증 단계

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

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

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

2.

`idm_user_1` 에 대한 정보 요청:

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

`idm_user_1` 이라는 사용자는 IdM에 없습니다.

7.6. 추가 리소스

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

8장. ANSIBLE 플레이북을 사용하여 사용자 그룹 관리

이 섹션에서는 **Ansible** 플레이북을 사용하여 사용자 그룹 관리를 소개합니다.

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

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

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

섹션에는 다음 주제가 포함되어 있습니다.

- **IdM의 다양한 그룹 유형**
- **직접 및 간접 그룹 멤버**
- **Ansible 플레이북을 사용하여 IdM 그룹 및 그룹 멤버가 있는지 확인**
- **Ansible 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 있는지 확인**
- **Ansible 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 없는지 확인**

8.1. IDM의 다양한 그룹 유형

IdM은 다음 유형의 그룹을 지원합니다.

POSIX 그룹(기본값)

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

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

비POSIX 그룹

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

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

외부 그룹

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

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

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

표 8.1. 기본적으로 생성된 사용자 그룹

그룹 이름	기본 그룹 멤버
ipausers	모든 IdM 사용자
관리자	기본 admin 사용자를 포함하여 관리자 권한이 있는 사용자
editors	이는 더 이상 특별한 권한이 없는 레거시 그룹입니다.

그룹 이름	기본 그룹 멤버
신뢰 관리자	Active Directory 트러스트를 관리할 수 있는 권한이 있는 사용자

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



주의

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

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

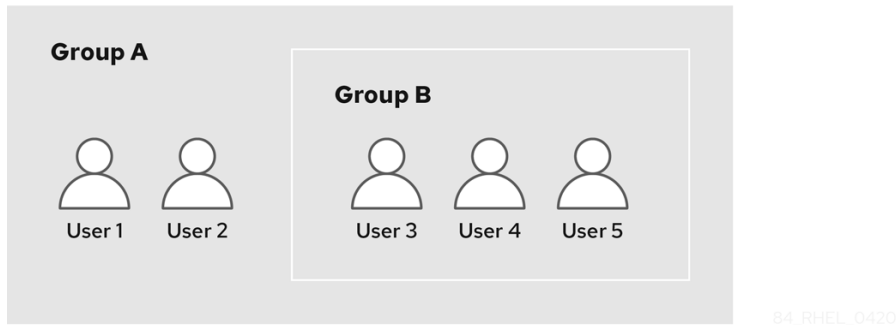
8.2. 직접 및 간접 그룹 멤버

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

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

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

그림 8.1. 직접 및 직접 그룹 멤버십



84_RHEL_0420

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

8.3. ANSIBLE 플레이북을 사용하여 IDM 그룹 및 그룹 멤버가 있는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 그룹 및 그룹 멤버(사용자 및 사용자 그룹)가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **Ansible** 플레이북에서 참조하려는 사용자는 **IdM**에 있습니다. **Ansible**을 사용하여 사용자가 있는지 확인하는 방법에 대한 자세한 내용은 **Ansible 플레이북을 사용하여 사용자 계정** 관리를 참조하십시오.

절차

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

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

2. 필요한 사용자 및 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다.

```

- name: Playbook to handle groups
  hosts: ipaserver
  become: true

  tasks:
    - name: Create group ops with gid 1234
      ipagroup:
        ipaadmin_password: MySecret123
        name: ops
        gidnumber: 1234

    - name: Create group sysops
      ipagroup:
        ipaadmin_password: MySecret123
        name: sysops
        user:
          - idm_user

    - name: Create group appops
      ipagroup:
        ipaadmin_password: MySecret123
        name: appops

    - name: Add group members sysops and appops to group ops
      ipagroup:
        ipaadmin_password: MySecret123
        name: ops
        group:
          - sysops
          - appops

```

3.

플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/add-group-members.yml

```

검증 단계

ops 그룹에 **sysops** 및 **app ops** 가 직접 멤버로 포함되어 있는지, **ipa group-show** 명령을 사용하여 **idm_user** 를 간접 멤버로 포함하고 있는지 확인할 수 있습니다.

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

```

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

```

2.

ops에 대한 정보를 표시합니다:

```
ipaserver]$ ipa group-show ops
Group name: ops
GID: 1234
Member groups: sysops, appops
Indirect Member users: idm_user
```

appops 및 **sysops** 그룹 - 후자의 경우 **idm_user** 사용자를 포함하여 **IdM**에 있습니다.

추가 리소스

- [/usr/share/doc/ansible-freeipa/README-group.md](#) Markdown 파일을 참조하십시오.

8.4. ANSIBLE 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 있는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 멤버 관리자(사용자 및 사용자 그룹 모두)가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

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

절차

1.

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

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

2.

필요한 사용자 및 그룹 멤버 관리 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다.

```

---
- name: Playbook to handle membership management
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure user test is present for group_a
      ipagroup:
        ipadmin_password: MySecret123
        name: group_a
        membermanager_user: test

    - name: Ensure group_admins is present for group_a
      ipagroup:
        ipadmin_password: MySecret123
        name: group_a
        membermanager_group: group_admins

```

3.

플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/add-member-managers-user-groups.yml

```

검증 단계

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

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

```

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

```

2.

managergroup1 에 대한 정보를 표시합니다.

```

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

```

추가 리소스

- `ipa host-add-member-manager --help` 를 참조하십시오.
- `ipa man` 페이지를 참조하십시오.

8.5. ANSIBLE 플레이북을 사용하여 IDM 사용자 그룹에 멤버 관리자가 없는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 멤버 관리자(사용자 및 사용자 그룹 모두)가 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.
- 제거 중인 기존 멤버 관리자 사용자 또는 그룹의 이름과 관리 중인 그룹의 이름이 있어야 합니다.

절차

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

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

2. 필요한 사용자 및 그룹 멤버 관리 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다.

```
---
- name: Playbook to handle membership management
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure member manager user and group members are absent for group_a
      ipagroup:
        ipadmin_password: MySecret123
        name: group_a
        membermanager_user: test
```

```
membermanager_group: group_admins
action: member
state: absent
```

3.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-member-managers-are-absent.yml
```

검증 단계

ipa group-show 명령을 사용하여 **group_a** 그룹에 **test** 가 멤버 관리자로, **group_admins** 를 **group_a** 의 멤버 관리자로 포함하고 있지 않은지 확인할 수 있습니다.

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

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

2.

group_a에 대한 정보를 표시합니다.

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

추가 리소스

- **ipa host-remove-member-manager --help** 를 참조하십시오.
- **ipa man** 페이지를 참조하십시오.

9장. ANSIBLE을 사용하여 IDM의 그룹 멤버십 자동화

자동 그룹 멤버십을 사용하여 속성에 따라 사용자 및 호스트 사용자 그룹 및 호스트 그룹을 자동으로 할당할 수 있습니다. 예를 들면 다음을 수행할 수 있습니다.

- 직원의 사용자 항목을 직원의 관리자, 위치, 위치 또는 기타 속성에 따라 그룹으로 나눕니다. 명령줄에서 `ipa user-add --help` 를 입력하여 모든 속성을 나열할 수 있습니다.
- 클래스, 위치 또는 기타 속성에 따라 호스트를 그룹으로 나눕니다. 명령줄에서 `ipa host-add --help` 를 입력하여 모든 속성을 나열할 수 있습니다.
- 단일 글로벌 그룹에 모든 사용자 또는 모든 호스트를 추가합니다.

Red Hat Ansible Engine을 사용하여 IdM(Identity Management)에서 자동 그룹 멤버십 관리를 자동화할 수 있습니다.

이 섹션에서는 다음 주제를 다룹니다.

- [Ansible을 사용하여 IdM 사용자 그룹에 대한 automember 규칙이 있는지 확인합니다.](#)
- [Ansible을 사용하여 IdM 사용자 그룹 automember 규칙에 조건이 있는지 확인합니다.](#)
- [Ansible을 사용하여 IdM 사용자 그룹 automember 규칙에 조건이 없는지 확인합니다.](#)
- [Ansible을 사용하여 IdM 그룹의 automember 규칙이 없는지 확인합니다.](#)
- [Ansible을 사용하여 IdM 호스트 그룹 automember 규칙에 조건이 있는지 확인합니다.](#)

9.1. ANSIBLE을 사용하여 IDM 사용자 그룹에 대한 AUTOMEMBER 규칙이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 IdM(Identity Management) 그룹에 대한 자동 멤버십 규칙이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 `testing_group` 사용자 그룹에 대해

automember 규칙이 있는지 확인합니다.

사전 요구 사항

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

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/automember/` 디렉터리에 있는 **automember-group-present.yml** Ansible 플레이북 파일을 복사합니다.

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

3. 편집할 **automember-group-present-copy.yml** 파일을 엽니다.
4. **ipaautomember** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
 - **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.

- **name** 변수를 **testing_group** 으로 설정합니다.
- **automember_type** 변수를 그룹으로 설정합니다.
- **state** 변수가 **present** 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Automember group present example
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure group automember rule admins is present
      ipaautomember:
        ipaadmin_password: Secret123
        name: testing_group
        automember_type: group
        state: present
```

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

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

9.2. ANSIBLE을 사용하여 IDM 사용자 그룹 AUTOMEMBER 규칙에 지정된 조건이 있는지 확인

추가 리소스

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 그룹의 **automember** 규칙에 지정된 조건이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 **automember** 규칙에 **UID** 관련 조건이 있는 경우 **testing_group** 그룹에 대해 확인됩니다. * 조건을 지정하면 향후 모든 **IdM** 사용자가 자동으로 **testing_group**의 멤버가 되도록 합니다.

사전 요구 사항

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

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/automember/` directory(예: `automember-usergroup-rule-present.yml`)에 있는 `automember-hostgroup-rule-present.yml` Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-present.yml automember-usergroup-rule-present.yml
```

3. 편집할 `automember-usergroup-rule-present.yml` 파일을 엽니다.
4. 다음 매개 변수를 수정하여 파일을 조정합니다.

- 플레이북의 이름을 사용 사례에 맞게 변경합니다(예: **Automember** 사용자 그룹 규칙 멤버).
-

사용 사례에 맞게 작업 이름을 변경합니다. 예를 들어 사용자 그룹의 **automember** 조건이 있는지 확인하십시오.

- **ipaautomember** 작업 섹션에서 다음 변수를 설정합니다.
 - **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
 - **name** 변수를 **testing_group** 으로 설정합니다.
 - **automember_type** 변수를 그룹으로 설정합니다.
 - **state** 변수가 **present** 로 설정되어 있는지 확인합니다.
 - **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
 - **include** 키 변수를 **UID** 로 설정합니다.
 - 포함 식 변수를 **.***로 설정합니다.*

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Automember user group rule member present
  hosts: ipaserver
  become: true
  tasks:
  - name: Ensure an automember condition for a user group is present
    ipaautomember:
      ipaadmin_password: Secret123
      name: testing_group
      automember_type: group
      state: present
      action: member
      inclusive:
        - key: UID
          expression: .*
```

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

```
$ ansible-playbook -v -i inventory automember-usergroup-rule-present.yml
```

검증 단계

1. **IdM** 관리자로 로그인합니다.

```
$ kinit admin
```

2. 다음과 같은 사용자를 추가합니다.

```
$ ipa user-add user101 --first user --last 101
-----
Added user "user101"
-----
User login: user101
First name: user
Last name: 101
...
Member of groups: ipausers, testing_group
...
```

9.3. ANSIBLE을 사용하여 IDM 사용자 그룹 AUTOMEMBER 규칙에 조건이 없는지 확인합니다.

추가 리소스

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 그룹에 대한 자동 멤버 규칙에서 조건이 없는지 확인하는 방법을 설명합니다. 이 예제에서는 자동 멤버 규칙에 조건이 없으면 초기 사용자가 **dp** 여야 함을 지정하는지 확인합니다. **automember** 규칙이 **testing_group** 그룹에 적용됩니다. 조건을 적용하면 초기 단계가 **dp** 인 향후 **IdM** 사용자가 **testing_group**의 멤버가 되지 않도록 합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **IdM**에 **testing_group** 사용자 그룹 및 **automember** 사용자 그룹 규칙이 있습니다.

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

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/automember/` 디렉터리(예: `automember-usergroup-rule-absent.yml`)에 있는 `automember-hostgroup-rule-absent.yml` **Ansible** 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-absent.yml automember-usergroup-rule-absent.yml
```

3. 편집할 `automember-usergroup-rule-absent.yml` 파일을 엽니다.

4. 다음 매개 변수를 수정하여 파일을 조정합니다.

- 플레이북의 이름을 사용 사례에 맞게 변경합니다(예: **Automember** 사용자 그룹 규칙 멤버 **absent**).
- 사용 사례에 해당하는 작업 이름 변경(예: 사용자 그룹의 **automember** 조건 확인)이 없습니다.
- `ipaautomember` 작업 섹션에서 다음 변수를 설정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 **testing_group** 으로 설정합니다.
- **automember_type** 변수를 그룹으로 설정합니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
- **include** 키 변수 를 **initial** 로 설정합니다.
- 포함 식 변수를 **dp** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Automember user group rule member absent
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure an automember condition for a user group is absent
      ipaautomember:
        ipaadmin_password: Secret123
        name: testing_group
        automember_type: group
        state: absent
        action: member
        inclusive:
          - key: initials
            expression: dp
```

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

```
$ ansible-playbook -v -i inventory automember-usergroup-rule-absent.yml
```

검증 단계

1. **IdM** 관리자로 로그인합니다.

```
$ kinit admin
```

2. **automember** 그룹을 확인합니다.

```
$ ipa automember-show --type=group testing_group
Automember Rule: testing_group
```

출력에 결정된 **Regex: initials=dp** 항목이 없으면 **testing_group automember** 규칙에 지정된 조건이 포함되어 있지 않습니다.

9.4. ANSIBLE을 사용하여 IDM 사용자 그룹의 AUTOMEMBER 규칙이 없는지 확인합니다.

추가 리소스

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 그룹에 대한 **automember** 규칙이 없는지 확인하는 방법을 설명합니다. 이 예제에서는 **testing_group** 그룹에 대해 **automember** 규칙이 없는지 확인합니다.



참고

automember 규칙을 삭제하면 규칙과 관련된 모든 조건도 삭제됩니다. 규칙에서 특정 조건만 제거하려면 **Ansible을 사용하여 IdM 사용자 그룹 automember 규칙에 조건이 없는지 확인합니다.**

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
 - **Ansible** 버전 2.8 이상을 사용하고 있습니다.

- **ansible-freeipa** 패키지를 설치했습니다.
- `~/MyPlaybooks/` 디렉터리에서 이러한 옵션을 구성하는 IdM 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 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. 편집할 `automember-group-absent-copy.yml` 파일을 엽니다.

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

- `ipaadmin_password` 변수를 IdM 관리자 암호로 설정합니다.
- `name` 변수를 `testing_group` 으로 설정합니다.
- `automember_type` 변수를 그룹으로 설정합니다.
- `state` 변수가 `absent` 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Automember group absent example
```

```

hosts: ipaserver
become: true
tasks:
- name: Ensure group automember rule admins is absent
  ipaautomember:
    ipaadmin_password: Secret123
    name: testing_group
    automember_type: group
    state: absent

```

5.

파일을 저장합니다.

6.

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

```
$ ansible-playbook -v -i inventory automember-group-absent.yml
```

추가 리소스

- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-automember.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/automember` 디렉토리를 참조하십시오.

9.5. 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 멤버이고 **FQDN** 이 `*.example.org` 가 **primary_dns_domain_hosts** 호스트 그룹의 멤버가 아닌지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **IdM**에 **primary_dns_domain_hosts** 호스트 그룹과 **automember** 호스트 그룹 규칙이 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.

- **Ansible 버전 2.8 이상을 사용하고 있습니다.**
- **ansible-freeipa** 패키지를 설치했습니다.
- **~/MyPlaybooks/** 디렉터리에서 이러한 옵션을 구성하는 **IdM 서버의 FQDN(정규화된 도메인 이름)**을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/automember/** 디렉터리에 있는 **automember-hostgroup-rule-present.yml** Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automember/automember-hostgroup-rule-present.yml automember-hostgroup-rule-present-copy.yml
```

3. 편집할 **automember-hostgroup-rule-present-copy.yml** 파일을 엽니다.

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

- **ipadmin_password** 변수를 **IdM 관리자 암호**로 설정합니다.
- **name** 변수를 **primary_dns_domain_hosts** 로 설정합니다.
- **automember_type** 변수를 **hostgroup** 으로 설정합니다.
- **state** 변수가 **present** 로 설정되어 있는지 확인합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.

- **inclusive** 키 변수가 **fqdn;**으로 설정되어 있는지 확인합니다.
- 해당하는 **include** 표현식 변수를 ***.idm.example.com** 으로 설정합니다.
- 전용 키 변수를 **fqdn;**로 설정합니다.
- 해당 전용 표현식 변수를 ***.example.org** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Automember user group rule member present
  hosts: ipaserver
  become: true
  tasks:
    - name: Ensure an automember condition for a user group is present
      ipaautomember:
        ipaadmin_password: Secret123
        name: primary_dns_domain_hosts
        automember_type: hostgroup
        state: present
        action: member
        inclusive:
          - key: fqdn
            expression: *.idm.example.com
        exclusive:
          - key: fqdn
            expression: *.example.org
```

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

```
$ ansible-playbook -v -i inventory automember-hostgroup-rule-present-copy.yml
```

추가 리소스

- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-automember.md](#) 파일을 참조하십시오.

- **/usr/share/doc/ansible-freeipa/playbooks/automember** 디렉토리를 참조하십시오.

10장. ANSIBLE 플레이북을 사용하여 IDM에서 셀프 서비스 규칙 관리

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

이 섹션에서는 다음 주제를 다룹니다.

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

10.1. IDM의 셀프 서비스 액세스 제어

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

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



주의

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

10.2. ANSIBLE을 사용하여 셀프 서비스 규칙이 있는지 확인합니다.

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

사전 요구 사항

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

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

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

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

3.

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

4.

ipaselfservice 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

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

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

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

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

5.

파일을 저장합니다.

6.

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

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

추가 리소스

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

10.3. ANSIBLE을 사용하여 셀프 서비스 규칙이 없는지 확인

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

사전 요구 사항

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

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

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

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

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

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

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

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

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

5. 파일을 저장합니다.

6.

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

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

추가 리소스

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

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

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

사전 요구 사항

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

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

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

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

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

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

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

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

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

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

```

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

```

5.

파일을 저장합니다.

6.

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

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

추가 리소스

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

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

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

사전 요구 사항

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

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

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

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

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

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

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

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 수정하려는 셀프 서비스 규칙의 이름으로 설정합니다.
- 특성 변수를 **givenname** 및 **surname** 으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

- **state** 변수를 **absent** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

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

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

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

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

추가 리소스

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

11장. ANSIBLE 플레이북을 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한 위임

위임은 셀프 서비스 규칙 및 **RBAC**(역할 기반 액세스 제어)와 함께 **IdM**의 액세스 제어 방법 중 하나입니다. 위임을 사용하여 한 사용자 그룹에 권한을 할당하여 다른 사용자 그룹의 항목을 관리할 수 있습니다.

이 섹션에서는 다음 주제를 다룹니다.

- [위임 규칙](#)
- [IdM용 Ansible 인벤토리 파일 생성](#)
- [Ansible을 사용하여 위임 규칙이 있는지 확인합니다.](#)
- [Ansible을 사용하여 위임 규칙이 없는지 확인합니다.](#)
- [Ansible을 사용하여 위임 규칙에 특정 속성이 있는지 확인](#)
- [Ansible을 사용하여 위임 규칙에 특정 속성이 없는지 확인](#)

11.1. 위임 규칙

위임 규칙을 만들어 사용자를 관리하기 위해 사용자 그룹에 권한을 위임할 수 있습니다.

위임 규칙을 사용하면 특정 사용자 그룹에서 다른 사용자 그룹의 사용자에게 대한 특정 속성에 대한 쓰기 (편집) 작업을 수행할 수 있습니다. 이 형식의 액세스 제어 규칙은 위임 규칙에 지정한 속성의 하위 집합 값을 편집하도록 제한됩니다. 전체 항목을 추가하거나 제거하지 않은 속성을 제어하는 기능을 부여하지 않습니다.

위임 규칙은 **IdM**의 기존 사용자 그룹에 권한을 부여합니다. 예를 들어 **managers** 사용자 그룹이 **employees** 사용자 그룹에 있는 사용자의 선택된 특성을 관리할 수 있도록 위임을 사용할 수 있습니다.

11.2. IDm용 ANSIBLE 인벤토리 파일 생성

Ansible을 사용하여 작업하는 경우 홈 디렉토리에서 `/usr/share/doc/ansible-freeipa/*` 및 `/usr/share/doc/rhel-system-roles/*` 하위 디렉토리에서 복사하여 조정하는 하위 디렉토리를 생성하는 것이 좋습니다. 이 방법은 다음과 같은 장점이 있습니다.

- 모든 플레이북을 한 곳에서 찾을 수 있습니다.
- **root** 권한을 호출하지 않고 플레이북을 실행할 수 있습니다.

절차

1. 홈 디렉토리에 **Ansible** 구성 및 플레이북의 디렉토리를 생성합니다.

```
$ mkdir ~/MyPlaybooks/
```

2. `~/MyPlaybooks/` 디렉토리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

3. 다음 콘텐츠를 사용하여 `~/MyPlaybooks/ansible.cfg` 파일을 생성합니다.

```
[defaults]
inventory = /home/<username>/MyPlaybooks/inventory

[privilege_escalation]
become=True
```

4. 다음 콘텐츠를 사용하여 `~/MyPlaybooks/inventory` 파일을 만듭니다.

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

이 구성은 해당 위치에 있는 호스트에 대해 **eu** 및 **us**의 두 개의 호스트 그룹을 정의합니다. 또

한 이 구성은 **eu** 및 **us** 그룹의 모든 호스트가 포함된 **ipaserver** 호스트 그룹을 정의합니다.

11.3. ANSIBLE을 사용하여 위임 규칙이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 새 **IdM** 위임 규칙에 대한 권한을 정의하고 있는지 확인하는 방법을 설명합니다. 이 예제에서 새 기본 관리자 속성 위임 규칙은 **managers** 그룹에 **staff** 그룹의 멤버에 대해 다음 속성을 읽고 쓸 수 있는 기능을 부여합니다.

- **businesscategory**
- **departmentnumber**
- **employeenumber**
- **employeetype**

사전 요구 사항

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

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

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

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

3. 편집할 `delegation-present-copy.yml` Ansible 플레이북 파일을 엽니다.

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

- `ipaadmin_password` 변수를 `IdM` 관리자의 암호로 설정합니다.
- `name` 변수를 새 위임 규칙의 이름으로 설정합니다.
- 권한 변수를 쉼표로 구분된 권한 목록으로 설정하여 부여할 권한(읽기 및 쓰기)을 부여합니다.
- 특성 변수를 위임된 사용자 그룹이 관리할 수 있는 속성 목록으로 설정합니다. `businesscategory, departmentnumber, staffnumber, and employeetype`.
- 특성을 보거나 수정할 수 있도록 액세스 권한이 부여되는 그룹의 이름으로 그룹 변수를 설정합니다.
- `membergroup` 변수를 보거나 수정할 수 있는 그룹의 이름으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage a delegation rule
  hosts: ipaserver
```

```
become: true
```

```
tasks:
```

```
- name: Ensure delegation "basic manager attributes" is present
```

```
  ipadelegation:
```

```
    ipadmin_password: Secret123
```

```
    name: "basic manager attributes"
```

```
    permission: read, write
```

```
    attribute:
```

```
      - businesscategory
```

```
      - departmentnumber
```

```
      - employeenumber
```

```
      - employeetype
```

```
    group: managers
```

```
    membergroup: employees
```

5.

파일을 저장합니다.

6.

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

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-present-copy.yml
```

추가 리소스

-

분류 [규칙](#)을 참조하십시오.

-

`/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-delegation.md** 파일을 참조하십시오.

-

`/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉터리에서 샘플 플레이북을 참조하십시오.

11.4. ANSIBLE을 사용하여 위임 규칙이 없는지 확인합니다.

다음 절차에서는 **IdM** 구성에 지정된 위임 규칙이 없는지 확인하는 데 **Ansible** 플레이북을 사용하는 방법을 설명합니다. 아래 예제에서는 **IdM**에 사용자 정의 기본 관리자 속성 위임 규칙이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

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

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

3. 편집할 **delegation-absent-copy.yml** **Ansible** 플레이북 파일을 엽니다.
4. **ipadelegation** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.
- **name** 변수를 위임 규칙의 이름으로 설정합니다.

- **state** 변수를 **absent** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Delegation absent
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure delegation "basic manager attributes" is absent
      ipadelegation:
        ipadmin_password: Secret123
        name: "basic manager attributes"
        state: absent
```

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

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-absent-copy.yml
```

추가 리소스

- 분류 [규칙](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-delegation.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉터리에서 샘플 플레이북을 참조하십시오.

11.5. ANSIBLE을 사용하여 위임 규칙에 특정 속성이 있는지 확인

다음 절차에서는 위임 규칙에 특정 설정이 있도록 **Ansible** 플레이북을 사용하는 방법을 설명합니다. 이 플레이북을 사용하여 이전에 만든 위임 역할을 수정할 수 있습니다. 이 예제에서는 기본 관리자 속성 위임 규칙만 **departmentnumber member** 특성을 갖도록 합니다. In the example, you ensure the basic manager attributes delegation rule only has the departmentnumber member attribute.

사전 요구 사항

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

절차

1. **~/MyPlaybooks/ 디렉터리로 이동합니다.**

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

2. **/usr/share/doc/ansible-freeipa/playbooks/delegation/ 디렉터리에 있는 delegation-member-present.yml 파일의 사본을 만듭니다.**

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

3. 편집할 **delegation-member-present-copy.yml Ansible 플레이북 파일**을 엽니다.
4. **ipadelegation** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 수정할 위임 규칙의 이름으로 설정합니다.
- 특정 변수를 **departmentnumber** 로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Delegation member present
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure delegation "basic manager attributes" member attribute
      departmentnumber is present
      ipadelegation:
        ipaadmin_password: Secret123
        name: "basic manager attributes"
        attribute:
          - departmentnumber
        action: member
```

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

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-member-present-copy.yml
```

추가 리소스

- 분류 [규칙](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-delegation.md** 파일을 참조하십시오.

- `/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉토리에서 샘플 플레이북을 참조하십시오.

11.6. ANSIBLE을 사용하여 위임 규칙에 특정 속성이 없는지 확인

다음 절차에서는 위임 규칙에 특정 설정이 없도록 **Ansible** 플레이북을 사용하는 방법을 설명합니다. 이 플레이북을 사용하여 위임 역할에 바람직하지 않은 액세스 권한이 부여되지 않도록 할 수 있습니다. 이 예제에서는 기본 관리자 속성 의 위임 규칙에 **staff number** 및 **employee type** 멤버 속성이 없는지 확인합니다.

사전 요구 사항

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

절차

1. `~/MyPlaybooks/` 디렉토리로 이동합니다.

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

2.

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

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

3.

편집할 `delegation-member-absent-copy.yml` Ansible 플레이북 파일을 엽니다.

4.

`ipadelegation` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 `IdM` 관리자의 암호로 설정합니다.
- `name` 변수를 수정할 위임 규칙의 이름으로 설정합니다.
- 특정 변수를 `worker number` 및 `employee type` 로 설정합니다.
- `action` 변수를 `member` 로 설정합니다.
- `state` 변수를 `absent` 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Delegation member absent
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure delegation "basic manager attributes" member attributes
      employee number and employee type are absent
      ipadelegation:
        ipaadmin_password: Secret123
        name: "basic manager attributes"
        attribute:
          - employee number
          - employee type
        action: member
        state: absent
```

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

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory delegation-member-absent-copy.yml
```

추가 리소스

- 분류 [규칙](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-delegation.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/ipadelegation` 디렉터리에서 샘플 플레이북을 참조하십시오.

12장. ANSIBLE 플레이북을 사용하여 IDM에서 역할 기반 액세스 제어 관리

RBAC(역할 기반 액세스 제어)는 역할 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. **IdM(Identity Management)**의 **RBAC** 구성 요소는 역할, 권한 및 권한입니다.

- 권한은 사용자 추가 또는 삭제, 그룹 수정, 읽기 액세스 등의 특정 작업을 수행할 수 있는 권한을 부여합니다.
- 권한은 예를 들어 새 사용자를 추가하는 데 필요한 모든 권한을 결합합니다.
- 역할에 따라 사용자, 사용자 그룹, 호스트 또는 호스트 그룹에 일련의 권한이 부여됩니다.

특히 대규모 회사에서 **RBAC**를 사용하면 개별 책임 영역이 있는 관리자의 계층적 시스템을 만드는 데 도움이 될 수 있습니다.

이 장에서는 **Ansible** 플레이북을 사용하여 **RBAC**를 관리할 때 수행되는 다음 작업을 설명합니다.

- **IdM의 권한**
- **기본 관리 권한**
- **IdM의 권한**
- **IdM의 역할**
- **IdM에서 사전 정의된 역할**
- **Ansible을 사용하여 권한이 있는 IdM RBAC 역할이 있는지 확인**
- **Ansible을 사용하여 IdM RBAC 역할이 없는지 확인**

- **Ansible을 사용하여 IdM RBAC 역할에 사용자 그룹이 할당되었는지 확인**
- **Ansible을 사용하여 특정 사용자가 IdM RBAC 역할에 할당되지 않았는지 확인**
- **Ansible을 사용하여 서비스가 IdM RBAC 역할의 멤버인지 확인**
- **Ansible을 사용하여 호스트가 IdM RBAC 역할의 멤버인지 확인**
- **Ansible을 사용하여 호스트 그룹이 IdM RBAC 역할의 멤버인지 확인**

12.1. IDM의 권한

권한은 역할 기반 액세스 제어의 가장 낮은 수준 단위이며 해당 작업이 적용되는 **LDAP** 항목과 함께 작업을 정의합니다. 빌딩 블록에 비해 필요한 만큼의 권한에 권한을 할당할 수 있습니다. 하나 이상의 권한은 허용되는 작업을 정의합니다.

- **write**
- **read**
- **search**
- **비교**
- **add**
- **delete**
- **all**

이러한 작업은 다음 세 가지 기본 대상에 적용됩니다.

- **subtree:** 도메인 이름(DN); 이 DN 아래의 하위 트리
- 대상 필터: **LDAP** 필터
- **target:** 가능한 와일드카드가 있는 DN을 사용하여 항목을 지정

또한 다음 편의 옵션에서 해당 속성을 설정합니다.

- **type:** 개체 유형 (사용자, 그룹 등)은 하위 트리와 대상 필터를 설정합니다.
- **memberOf:** 그룹의 멤버이며, 대상 필터를 설정합니다.
- **TargetGroup :** 특정 그룹을 수정할 수 있는 액세스 권한을 부여합니다(예: 그룹 멤버십을 관리할 수 있는 권한 부여)

IdM 권한을 사용하면 어떤 오브젝트에 액세스할 수 있는 사용자와 이러한 오브젝트의 속성을 제어할 수 있습니다. **IdM**을 사용하면 개별 속성을 허용 또는 차단하거나 사용자, 그룹 또는 **sudo**와 같은 특정 **IdM** 기능의 전체 가시성을 모든 익명 사용자, 인증된 모든 사용자 또는 특정 권한 있는 사용자 그룹에 변경할 수 있습니다.

예를 들어, 권한에 대한 이 접근 방식의 유연성은 사용자 또는 그룹의 액세스 권한을 이러한 사용자 또는 그룹에 대한 액세스만 제한하려는 관리자에게 유용합니다. 이러한 사용자 또는 그룹에 액세스해야 하며 다른 섹션을 완전히 숨겨야 합니다.



참고

권한은 다른 권한을 포함할 수 없습니다.

12.2. 기본 관리 권한

관리 권한은 기본적으로 **IdM**과 함께 제공되는 권한입니다. 이러한 권한은 사용자가 생성한 다른 권한처럼 작동하며 다음과 같은 차이점이 있습니다.

- 해당 항목을 삭제하거나 이름, 위치 및 대상 속성을 수정할 수 없습니다.
- 세 가지 속성 세트가 있습니다.
 - 기본 속성, 사용자가 **IdM**에서 관리하므로 수정할 수 없습니다.
 - 사용자가 추가한 추가 속성인 포함 특성
 - 사용자가 제거된 속성인 제외된 특성

관리 권한은 **default** 및 **included** 특성 세트에 표시되지만 제외 세트에는 표시되지 않는 모든 특성에 적용됩니다.



참고

관리 권한을 삭제할 수는 없지만 해당 **bind** 유형을 권한으로 설정하고 모든 권한에서 관리 권한을 제거하면 이를 효과적으로 비활성화합니다.

모든 관리 권한의 이름은 **System:** (예: **System: Add Sudo rule** or **System: Modify Services**)로 시작합니다. 이전 버전의 **IdM**에서는 기본 권한에 다른 스키마를 사용했습니다. 예를 들어 사용자는 해당 항목을 삭제할 수 없으며 해당 사용자를 권한에만 할당할 수 있었습니다. 이러한 기본 권한의 대부분은 관리 되는 권한으로 전환되었지만 다음 권한에서는 여전히 이전 스키마를 사용합니다.

- 멤버십 자동 작성 작업 추가
- 설정 하위 항목 추가
- 복제 계약 추가
- 인증서 제거 **Hold**

- **CA에서 인증서 상태 가져오기**
- **DNA 범위 읽기**
- **DNA 범위 수정**
- **PassSync Managers 설정 읽기**
- **PassSync Managers 설정 수정**
- **복제 계약 읽기**
- **복제 계약 수정**
- **복제 계약 제거**
- **LDBM 데이터베이스 구성 읽기**
- **요청 인증서**
- **CA ACL을 무시하는 인증서 요청**
- **다른 호스트의 인증서 요청**
- **CA에서 인증서 검색**
- **인증서 해지**

•

IPA 설정 쓰기



참고

명령줄에서 관리 권한을 수정하려고 하면 시스템에서 수정할 수 없는 속성을 변경할 수 없으므로 명령이 실패합니다. 웹 UI에서 관리되는 권한을 수정하려고 하면 수정할 수 없는 특성이 비활성화됩니다.

12.3. IDM의 권한

권한은 역할에 적용 가능한 권한 그룹입니다.

권한은 단일 작업을 수행할 수 있는 권한을 제공하지만 성공하려면 여러 권한이 필요한 특정 **IdM** 작업이 있습니다. 따라서 권한은 특정 작업을 수행하는 데 필요한 다양한 권한을 결합합니다.

예를 들어 새 **IdM** 사용자에게 대한 계정을 설정하려면 다음 권한이 필요합니다.

•

새 사용자 항목 만들기

•

사용자 암호 재설정

•

기본 **IPA** 사용자 그룹에 새 사용자 추가

이러한 세 가지 하위 수준 작업을 라는 사용자 지정 권한 형식의 상위 수준 작업으로 결합하면 시스템 관리자가 역할을 보다 쉽게 관리할 수 있습니다. **IdM**에는 이미 여러 기본 권한이 포함되어 있습니다. 사용자 및 사용자 그룹 외에도 호스트 및 호스트 그룹과 네트워크 서비스에 권한이 할당됩니다. 이 방법을 사용하면 특정 네트워크 서비스를 사용하는 호스트 집합에서 일련의 사용자별로 작업을 세부적으로 제어할 수 있습니다.



참고

권한에는 다른 권한이 포함되지 않을 수 있습니다.

12.4. IDM의 역할

역할은 역할에 지정된 사용자가 보유한 권한 목록입니다.

실제로 권한은 낮은 수준의 작업(사용자 항목 생성, 그룹에 항목을 추가 등)을 수행할 수 있는 기능을 부여합니다. 권한은 상위 수준의 작업(예: 지정된 그룹에서 새 사용자 생성)에 필요한 이러한 권한 중 하나

이상을 결합합니다. 역할은 필요에 따라 권한을 함께 수집합니다. 예를 들어 사용자 관리자 역할은 사용자를 추가, 수정, 삭제할 수 있습니다.



중요

역할은 허용된 작업을 분류하는 데 사용됩니다. 권한 분리를 구현하거나 권한 에스컬레이션으로부터 보호하는 도구로는 사용되지 않습니다.



참고

역할에 다른 역할을 포함할 수 없습니다.

12.5. ID 관리에서 사전 정의된 역할

Red Hat Identity Management는 다음과 같은 다양한 사전 정의된 역할을 제공합니다.

표 12.1. ID 관리에서 사전 정의된 역할

Role	권한	설명
등록 관리자	호스트 등록	클라이언트 또는 호스트, 등록
helpdesk	사용자 수정 및 암호 재설정, 그룹 멤버십 수정	간단한 사용자 관리 작업 수행
IT 보안 전문가	Netgroups 관리자, HBAC 관리자, Sudo 관리자	호스트 기반 액세스 제어와 같은 보안 정책 관리 sudo 규칙
IT 전문가	호스트 관리자, 호스트 그룹 관리자, 서비스 관리자, 자동 마운트 관리자	호스트 관리
보안 아키텍트	위임 관리자, 복제 관리자, 쓰기 IPA 구성, 암호 정책 관리자	ID 관리 환경 관리, 신뢰 생성, 복제 계약 생성
사용자 관리자	사용자 관리자, 그룹 관리자, 사용자 관리자 단계	사용자 및 그룹 생성

12.6. ANSIBLE을 사용하여 권한이 있는 IDM RBAC 역할이 있는지 확인

기본 역할보다 IdM(Identity Management) 리소스에 대한 역할 기반 액세스(RBAC)를 보다 세밀하게 제어하려면 사용자 지정 역할을 생성합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 새 **IdM** 사용자 지정 역할에 대한 권한을 정의하고 해당 역할이 있는지 확인하는 방법을 설명합니다. 이 예제에서 새 **user_and_host_administrator** 역할에는 기본적으로 **IdM**에 있는 다음 권한의 고유한 조합이 포함되어 있습니다.

- 그룹 관리자
- 사용자 관리자
- 사용자 관리자 단계
- 그룹 관리자

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 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-present.yml** 파일의 사본을 만듭니다.

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

- 3. 편집할 **role-member-user-present-copy.yml Ansible** 플레이북 파일을 엽니다.

- 4. **iparole** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 새 역할의 이름으로 설정합니다.
- 새 역할에 포함하려는 **IdM** 권한 이름으로 권한 목록을 설정합니다.
- 선택적으로 사용자 변수를 새 역할을 부여하려는 사용자 이름으로 설정합니다.
- 선택적으로 그룹 변수를 새 역할을 부여하려는 그룹 이름으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
  - iparole:
    ipaadmin_password: Secret123
    name: user_and_host_administrator
    user: idm_user01
    group: idm_group01
    privilege:
      - Group Administrators
      - User Administrators
      - Stage User Administrators
      - Group Administrators
```

- 5. 파일을 저장합니다.

6.

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

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-user-present-copy.yml
```

추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.
- [IdM의 역할](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 **README-role** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉터리에서 샘플 플레이북을 참조하십시오.

12.7. ANSIBLE을 사용하여 IDM RBAC 역할이 없는지 확인

IdM(Identity Management)에서 **RBAC(역할 기반 액세스 제어)**를 관리하는 시스템 관리자는 관리자가 실수로 사용자에게 이를 할당하지 않도록 더 이상 사용되지 않도록 할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 역할이 없는지 확인하는 방법을 설명합니다. 아래 예제에서는 사용자 지정 `user_and_host_administrator` 역할이 **IdM**에 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 [ansible-freeipa](#) 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

- **Ansible** 인벤토리 파일은 `~/<MyPlaybooks>/` 디렉터리에 있습니다.

절차

1. `~/<MyPlaybooks>/` 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

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

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

3. 편집할 `role-is-absent-copy.yml` **Ansible** 플레이북 파일을 엽니다.
4. `iparole` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 **IdM** 관리자의 암호로 설정합니다.
- `name` 변수를 역할의 이름으로 설정합니다.
- `state` 변수가 `absent` 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
  - iparole:
    ipaadmin_password: Secret123
    name: user_and_host_administrator
    state: absent
```

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

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-is-absent-copy.yml
```

추가 리소스

- [Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오.](#)
- [IdM의 역할을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-role Markdown 파일을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리에서 샘플 플레이북을 참조하십시오.](#)

12.8. ANSIBLE을 사용하여 IDENTITY RBAC 역할에 사용자 그룹이 할당되었는지 확인

IdM(Identity Management)에서 **RBAC(역할 기반 액세스 제어)**를 관리하는 시스템 관리자는 특정 사용자 그룹(예: **junior** 관리자)에 역할을 할당할 수 있습니다.

다음 예제에서는 **Ansible** 플레이북을 사용하여 기본 제공 **IdM RBAC** 지원 **datacenter** 역할이 **junior_sysadmins**에 할당되도록 하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** [인벤토리 파일](#)을 생성했습니다.

- **Ansible** 인벤토리 파일은 `~/ <MyPlaybooks>/` 디렉터리에 있습니다.

절차

1. `~/ <MyPlaybooks> /` 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

2. `/usr/share/doc/ansible-freeipa/playbooks/role/` 디렉터리에 있는 `role-member-group-present.yml` 파일의 사본을 만듭니다.

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

3. 편집할 `role-member-group-present-copy.yml` **Ansible** 플레이북 파일을 엽니다.
4. `iparole` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 **IdM** 관리자의 암호로 설정합니다.
- 이름 변수를 할당하려는 역할의 이름으로 설정합니다.
- 그룹 변수를 그룹 이름으로 설정합니다.
- `action` 변수를 `member` 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no
```

```
tasks:
- iparole:
  ipaadmin_password: Secret123
  name: helpdesk
  group: junior_sysadmins
  action: member
```

5.

파일을 저장합니다.

6.

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

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-group-present-copy.yml
```

추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.
- [IdM의 역할](#)을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-role Markdown](#) 파일을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/playbooks/iparole](#) 디렉터리에서 샘플 플레이북을 참조하십시오.

12.9. ANSIBLE을 사용하여 특정 사용자가 IDM RBAC 역할에 할당되지 않았는지 확인

IdM(Identity Management)에서 RBAC(역할 기반 액세스 제어)를 관리하는 시스템 관리자는 RBAC 역할이 회사 내의 다른 위치로 이동된 후 특정 사용자에게 할당되지 않도록 할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **user_01** 및 **user_02** 라는 사용자가 **helpRuntimeConfig** 역할에 할당되지 않도록 하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 ~/ <MyPlaybooks>/ 디렉터리에 있습니다.

절차

1. ~/ <MyPlaybooks> / 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

2. /usr/share/doc/ansible-freeipa/playbooks/role/ 디렉터리에 있는 role-member-user-absent.yml 파일의 사본을 만듭니다.

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

3. 편집할 role-member-user-absent-copy.yml Ansible 플레이북 파일을 엽니다.
4. iparole 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- 이름 변수를 할당하려는 역할의 이름으로 설정합니다.
- 사용자 목록을 사용자 이름으로 설정합니다.
- action 변수를 member 로 설정합니다.

- **state** 변수를 **absent** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
    - iparole:
      ipaadmin_password: Secret123
      name: helpdesk
      user
      - user_01
      - user_02
      action: member
      state: absent
```

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

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-user-absent-copy.yml
```

추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.
- [IdM의 역할](#)을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/](#) 디렉토리의 **README-role Markdown** 파일을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/playbooks/iparole](#) 디렉터리에서 샘플 플레이북을 참조하십시오.

12.10. ANSIBLE을 사용하여 서비스가 IDM RBAC 역할의 멤버인지 확인

IdM(Identity Management)에서 RBAC(역할 기반 액세스 제어)를 관리하는 시스템 관리자는 IdM에 등록된 특정 서비스가 특정 역할의 멤버인지 확인할 수 있습니다. 다음 예제에서는 사용자 지정 `web_administrator` 역할이 `client01.idm.example.com` 서버에서 실행 중인 HTTP 서비스를 관리할 수 있도록 하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 `ansible-freeipa` 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 `~/ <MyPlaybooks>/` 디렉터리에 있습니다.
- IdM에 `web_administrator` 역할이 있습니다.
- IdM에 `HTTP/client01.idm.example.com@IDM.EXAMPLE.COM` 서비스가 있습니다.

절차

1. `~/ <MyPlaybooks> /` 디렉터리로 이동합니다.

```
$ cd ~/<MyPlaybooks>/
```

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

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

3. 편집할 `role-member-service-present-copy.yml` Ansible 플레이북 파일을 엽니다.

4.

iparole 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.
- 이름 변수를 할당하려는 역할의 이름으로 설정합니다.
- 서비스 목록을 서비스 이름으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
  - iparole:
    ipaadmin_password: Secret123
    name: web_administrator
    service:
    - HTTP/client01.idm.example.com
    action: member
```

5.

파일을 저장합니다.

6.

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

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-service-present-copy.yml
```

추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.

- **IdM의 역할을** 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-role Markdown** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉터리에서 샘플 플레이북을 참조하십시오.

12.11. ANSIBLE을 사용하여 호스트가 IDM RBAC 역할의 멤버인지 확인

IdM(Identity Management)에서 역할 기반 액세스 제어를 관리하는 시스템 관리자는 특정 호스트 또는 호스트 그룹이 특정 역할과 연결되도록 할 수 있습니다. 다음 예제에서는 사용자 지정 `web_administrator` 역할에서 HTTP 서비스가 실행 중인 `client01.idm.example.com` IdM 호스트를 관리할 수 있도록 하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일을** 생성했습니다.
- Ansible 인벤토리 파일은 `~/ <MyPlaybooks>/` 디렉터리에 있습니다.
- IdM에 `web_administrator` 역할이 있습니다.
- `client01.idm.example.com` 호스트는 IdM에 있습니다.

절차

1. `~/ <MyPlaybooks> /` 디렉터리로 이동합니다.

■

```
$ cd ~/<MyPlaybooks>/
```

2.

/usr/share/doc/ansible-freeipa/playbooks/role/ 디렉터리에 있는 **role-member-host-present.yml** 파일의 사본을 만듭니다.

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

3.

편집할 **role-member-host-present-copy.yml** Ansible 플레이북 파일을 엽니다.

4.

iparole 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.
- 이름 변수를 할당하려는 역할의 이름으로 설정합니다.
- 호스트 목록을 호스트 이름으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
  - iparole:
    ipaadmin_password: Secret123
    name: web_administrator
    host:
    - client01.idm.example.com
    action: member
```

5.

파일을 저장합니다.

6.

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

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-host-present-copy.yml
```

추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.
- [IdM의 역할](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-role** Markdown 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/iparole` 디렉터리에서 샘플 플레이북을 참조하십시오.

12.12. ANSIBLE을 사용하여 호스트 그룹이 IDM RBAC 역할의 멤버인지 확인

IdM(Identity Management)에서 역할 기반 액세스 제어를 관리하는 시스템 관리자는 특정 호스트 또는 호스트 그룹이 특정 역할과 연결되도록 할 수 있습니다. 다음 예제에서는 사용자 지정 `web_administrator` 역할이 HTTP 서비스가 실행 중인 IdM 호스트의 `web_servers` 그룹을 관리할 수 있도록 하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 `ansible-freeipa` 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 [Ansible 인벤토리 파일](#)을 생성했습니다.
- Ansible 인벤토리 파일은 `~/<MyPlaybooks>/` 디렉터리에 있습니다.
- IdM에 `web_administrator` 역할이 있습니다.

- **IdM에 web_servers 호스트 그룹이 있습니다.**

절차

1. **~/ <MyPlaybooks> / 디렉터리로 이동합니다.**

```
$ cd ~/<MyPlaybooks>/
```

2. **/usr/share/doc/ansible-freeipa/playbooks/role/ 디렉터리에 있는 role-member-hostgroup-present.yml 파일의 사본을 만듭니다.**

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

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

- **ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.**
- **이름 변수를 할당하려는 역할의 이름으로 설정합니다.**
- **hostgroup 목록을 호스트 그룹의 이름으로 설정합니다.**

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  tasks:
  - iparole:
    ipaadmin_password: Secret123
    name: web_administrator
```

```
hostgroup:
- web_servers
action: member
```

5.

파일을 저장합니다.

6.

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

```
$ ansible-playbook -v -i ~/<MyPlaybooks>/inventory role-member-hostgroup-present-copy.yml
```

추가 리소스

- [Ansible Vault](#)를 사용하여 콘텐츠 암호화를 참조하십시오.
- [IdM의 역할](#)을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-role Markdown](#) 파일을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/playbooks/iparole](#) 디렉터리에서 샘플 플레이북을 참조하십시오.

13장. ANSIBLE 플레이북을 사용하여 RBAC 권한 관리

RBAC(역할 기반 액세스 제어)는 역할, 권한 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. 특히 대규모 회사에서 **RBAC**를 사용하면 개별 책임 영역이 있는 관리자의 계층적 시스템을 만드는 데 도움이 될 수 있습니다.

이 장에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에서 **RBAC** 권한을 관리하기 위한 다음 작업을 설명합니다.

- **Ansible을 사용하여 사용자 정의 RBAC 권한이 있는지 확인**
- **Ansible을 사용하여 사용자 정의 IdM RBAC 권한에 멤버 권한이 있는지 확인**
- **Ansible을 사용하여 IdM RBAC 권한에 권한이 포함되어 있지 않은지 확인**
- **Ansible을 사용하여 사용자 정의 IdM RBAC 권한 이름 변경**
- **Ansible을 사용하여 IdM RBAC 권한이 없는지 확인**

사전 요구 사항

- **RBAC의 개념과 원칙을 이해합니다.**

13.1. ANSIBLE을 사용하여 사용자 정의 IdM RBAC 권한이 있는지 확인

IdM(Identity Management) 역할 기반 액세스 제어(**RBAC**)에서 완전한 사용자 정의 권한을 얻으려면 단계를 진행해야 합니다.

1. 권한이 연결되어 있지 않은 권한을 생성합니다.
2. 선택한 권한을 권한에 추가합니다.

다음 절차에서는 나중에 권한을 추가할 수 있도록 **Ansible** 플레이북을 사용하여 빈 권한을 생성하는 방법을 설명합니다. 이 예제에서는 호스트 관리와 관련된 모든 **IdM** 권한을 결합하는 데 사용되는 **full_host_administration** 이라는 권한을 생성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
- **Ansible** 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/privilege/** 디렉터리에 있는 **privilege-present.yml** 파일의 사본을 만듭니다.

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

3. 편집할 **privilege-present-copy.yml** **Ansible** 플레이북 파일을 엽니다.

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

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 새 권한의 이름으로 설정합니다. **full_host_administration**.

•

필요한 경우 **description** 변수를 사용하여 권한을 설명합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Privilege present example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure privilege full_host_administration is present
      ipaprivilege:
        ipaadmin_password: Secret123
        name: full_host_administration
        description: This privilege combines all IdM permissions related to host
                     administration
```

5.

파일을 저장합니다.

6.

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

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

13.2. ANSIBLE을 사용하여 사용자 정의 IDM RBAC 권한에 멤버 권한이 있는지 확인

IdM(Identity Management) 역할 기반 액세스 제어(**RBAC**)에서 완전한 사용자 정의 권한을 얻으려면 단계를 진행해야 합니다.

1.

권한이 연결되어 있지 않은 권한을 생성합니다.

2.

선택한 권한을 권한에 추가합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 이전 단계에서 생성된 권한에 권한을 추가하는 방법을 설명합니다. 이 예제에서는 호스트 관리와 관련된 모든 **IdM** 권한을 **full_host_administration** 이라는 권한에 추가하는 방법을 설명합니다. 기본적으로 권한은 호스트 등록, 호스트 관리자 및 호스트 그룹 관리자 권한 간에 배포됩니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.
- **full_host_administration** 권한이 있습니다. Ansible을 사용하여 권한을 생성하는 방법에 대한 자세한 내용은 **Using Ansible to ensure a custom IdM RBAC 권한이 있는지** 확인하십시오.

절차

1. ~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible-freeipa/playbooks/privilege/ 디렉터리에 있는 **privilege-member-present.yml** 파일의 사본을 만듭니다.

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

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

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.

- **name** 변수를 권한 이름으로 설정합니다.
- 권한 목록을 권한에 포함하려는 권한 이름으로 설정합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Privilege member present example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that permissions are present for the "full_host_administration"
      privilege
      ipaprivilege:
        ipaadmin_password: Secret123
        name: full_host_administration
        permission:
          - "System: Add krbPrincipalName to a Host"
          - "System: Enroll a Host"
          - "System: Manage Host Certificates"
          - "System: Manage Host Enrollment Password"
          - "System: Manage Host Keytab"
          - "System: Manage Host Principals"
          - "Retrieve Certificates from the CA"
          - "Revoke Certificate"
          - "System: Add Hosts"
          - "System: Add krbPrincipalName to a Host"
          - "System: Enroll a Host"
          - "System: Manage Host Certificates"
          - "System: Manage Host Enrollment Password"
          - "System: Manage Host Keytab"
          - "System: Manage Host Keytab Permissions"
          - "System: Manage Host Principals"
          - "System: Manage Host SSH Public Keys"
          - "System: Manage Service Keytab"
          - "System: Manage Service Keytab Permissions"
          - "System: Modify Hosts"
          - "System: Remove Hosts"
          - "System: Add Hostgroups"
          - "System: Modify Hostgroup Membership"
          - "System: Modify Hostgroups"
          - "System: Remove Hostgroups"
```

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

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

13.3. ANSIBLE을 사용하여 IDM RBAC 권한에 권한이 포함되어 있지 않은지 확인

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 권한에서 권한을 제거하는 방법을 설명합니다. 이 예제에서는 기본 인증서 관리자 권한에서 **CA ACL** 권한을 무시하는 요청 인증서 제거 방법을 설명합니다. 예를 들어 관리자는 보안 위험을 고려합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 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 사본을 만듭니다.

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

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

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

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.
- 권한 에서 제거할 권한의 이름으로 권한 목록을 설정합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Privilege absent example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that the "Request Certificate ignoring CA ACLs" permission is absent
      from the "Certificate Administrators" privilege
      ipaprivilege:
        ipaadmin_password: Secret123
        name: Certificate Administrators
        permission:
          - "Request Certificate ignoring CA ACLs"
        action: member
        state: absent
```

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

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

13.4. ANSIBLE을 사용하여 사용자 정의 IDM RBAC 권한 이름 변경

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어를 사용자 지정할 수 있습니다.

다음 절차에서는 예를 들어 권한에서 몇 가지 권한을 제거했기 때문에 권한 이름 변경 방법을 설명합니다. 결과적으로 권한 이름이 더 이상 정확하지 않습니다. 이 예제에서 관리자는 **full_host_administration** 권한의 이름을 **limited_host_administration** 으로 변경합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
- Ansible 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.
- **full_host_administration** 권한이 있습니다. 권한을 추가하는 방법에 대한 자세한 내용은 **Using Ansible to ensure a custom IdM RBAC 권한이 있는지** 확인하십시오.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2.

`/usr/share/doc/ansible-freeipa/playbooks/privilege/` 디렉터리에 있는 `privilege-present.yml` 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-present.yml rename-privilege.yml
```

3.

편집할 `rename-privilege.yml` Ansible 플레이북 파일을 엽니다.

4.

`ipaprivilege` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.
- `name` 변수를 현재 권한 이름으로 설정합니다.
- `rename` 변수를 추가하고 권한의 새 이름으로 설정합니다.
- `state` 변수를 추가하고 이름이 로 설정합니다.

5.

플레이북 자체의 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
---
- name: Rename a privilege
  hosts: ipaserver
  become: true
```

6.

플레이북의 작업 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
[...]
tasks:
- name: Ensure the full_host_administration privilege is renamed to
  limited_host_administration
  ipaprivilege:
  [...]
```

현재 예제에 대해 수정된 Ansible 플레이북 파일입니다.

■

```

---
- name: Rename a privilege
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure the full_host_administration privilege is renamed to
      limited_host_administration
      ipaprivilege:
        ipaadmin_password: Secret123
        name: full_host_administration
        rename: limited_host_administration
        state: renamed

```

7.

파일을 저장합니다.

8.

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

```
$ ansible-playbook -v -i inventory rename-privilege.yml
```

13.5. ANSIBLE을 사용하여 IDM RBAC 권한이 없는지 확인

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어를 사용자 지정할 수 있습니다. 다음 절차에서는 **Ansible** 플레이북을 사용하여 **RBAC** 권한이 없는지 확인하는 방법을 설명합니다. 이 예제에서는 **CA** 관리자 권한이 없는지 확인하는 방법을 설명합니다. 이 절차를 통해 관리자 관리자는 **IdM**에서 인증 기관을 관리할 수 있는 유일한 사용자가 됩니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 구성을 수행하려는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
- **Ansible** 인벤토리 파일은 **~/MyPlaybooks/** 디렉터리에 있습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

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

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

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

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

- `ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.
- 이름 변수를 제거하려는 권한의 이름으로 설정합니다.
- `state` 변수가 `absent` 로 설정되어 있는지 확인합니다.

5. 플레이북의 작업 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
[...]
tasks:
- name: Ensure privilege "CA administrator" is absent
  ipaprivilege:
  [...]
```

현재 예제에 대해 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Privilege absent example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure privilege "CA administrator" is absent
```

```

ipaprivilege:
  ipaadmin_password: Secret123
  name: CA administrator
  state: absent

```

6.

파일을 저장합니다.

7.

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

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

13.6. 추가 리소스

•

[IdM의 권한 참조.](#)

•

[IdM의 권한을 참조하십시오.](#)

•

[/usr/share/doc/ansible-freeipa/](#) 디렉터리에서 사용 가능한 **README-privilege** 파일을 참조하십시오.

•

[/usr/share/doc/ansible-freeipa/playbooks/ipaprivilege](#) 디렉터리에서 샘플 플레이북을 참조하십시오.

14장. ANSIBLE 플레이북을 사용하여 IDM에서 RBAC 권한 관리

RBAC(역할 기반 액세스 제어)는 역할, 권한 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. 특히 대규모 회사에서 **RBAC**를 사용하면 개별 책임 영역이 있는 관리자의 계층적 시스템을 만드는 데 도움이 될 수 있습니다.

이 장에서는 **Ansible** 플레이북을 사용하여 **IdM**(Identity Management)에서 **RBAC** 권한을 관리할 때 수행되는 다음 작업을 설명합니다.

- **Ansible**을 사용하여 **RBAC** 권한이 있는지 확인
- **Ansible**을 사용하여 특성이 있는 **RBAC** 권한이 있는지 확인합니다.
- **Ansible**을 사용하여 **RBAC** 권한이 없는지 확인
- **Ansible**을 사용하여 속성이 **IdM RBAC** 권한의 멤버인지 확인
- **Ansible**을 사용하여 속성이 **IdM RBAC** 권한의 멤버가 아닌지 확인
- **Ansible**을 사용하여 **IdM RBAC** 권한 이름 변경

사전 요구 사항

- **RBAC**의 개념과 원칙을 이해합니다.

14.1. ANSIBLE을 사용하여 RBAC 권한이 있는지 확인

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 권한에 추가할 수 있도록 **Ansible** 플레이북을 사용하여 **IdM**에 권한이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한은 호스트에만 적용할 수 있습니다.
- 권한이 포함된 권한이 부여된 사용자에게는 해당 항목에 대해 가능한 다음 모든 작업을 수행할 수 있습니다.
 - **write**
 - **read**
 - 검색
 - 비교
 - **add**
 - **delete**

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2.

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

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

3.

편집할 `permission-present-copy.yml` Ansible 플레이북 파일을 엽니다.

4.

`ipapermission` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- `ipaadmin_password` 변수를 IdM 관리자의 암호로 설정합니다.
- `name` 변수를 권한 이름으로 설정합니다.
- `object_type` 변수를 `host` 로 설정합니다.
- 올바른 변수를 모든 으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission present example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure that the "MyPermission" permission is present
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      object_type: host
      right: all
```

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

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

14.2. ANSIBLE을 사용하여 특성이 있는 RBAC 권한이 있는지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 권한에 추가할 수 있도록 **Ansible** 플레이북을 사용하여 **IdM**에 권한이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한은 호스트를 추가하는 데만 사용할 수 있습니다.
- 권한이 포함된 권한이 부여된 사용자에게는 호스트 항목에서 가능한 다음 모든 작업을 수행할 수 있습니다.
 - **write**
 - **read**
 - 검색
 - 비교
 - **add**

○

delete

●

MyPermission 권한이 포함된 권한이 부여된 사용자가 생성한 호스트 항목은 설명 값을 가질 수 있습니다.



참고

권한을 생성할 때 지정할 수 있는 속성 유형은 **IdM LDAP** 스키마에 의해 제한되지 않습니다. 그러나 예를 들어 **object_type** 이 호스트가 나중에 **host** 인 경우 **attrs: car_licence** 를 지정하면 권한을 수행하고 호스트에 특정 자동차 라이선스 값을 추가하려고 할 때 **ipa: ERROR: attribute "car-license"가 오류 메시지를 허용하지 않습니다.**

사전 요구 사항

●

IdM 관리자 암호를 알고 있습니다.

●

Ansible 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.

●

이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.

절차

1.

~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2.

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

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-present.yml
permission-present-with-attribute.yml
```

3.

편집할 **permission-present-present-with-attribute.yml** **Ansible** 플레이북 파일을 엽니다.

4.

ipapermission 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.
- **object_type** 변수를 **host** 로 설정합니다.
- 올바른 변수를 모든 으로 설정합니다.
- **attrs** 변수를 설명으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission present example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure that the "MyPermission" permission is present with an attribute
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      object_type: host
      right: all
      attrs: description
```

5.

파일을 저장합니다.

6.

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

```
$ ansible-playbook -v -i inventory permission-present-with-attribute.yml
```

추가 리소스

- **RHEL 7의 *Linux* 도메인 ID, 인증 및 정책 가이드의 사용자 및 그룹 스키마**를 참조하십시오.

14.3. ANSIBLE을 사용하여 RBAC 권한이 없는지 확인

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 권한에 추가할 수 없도록 **Ansible** 플레이북을 사용하여 **IdM**에 권한이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

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

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

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

4.

ipapermission 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission absent example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that the "MyPermission" permission is absent
      ipapermission:
        ipaadmin_password: Secret123
        name: MyPermission
        state: absent
```

5.

파일을 저장합니다.

6.

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

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

14.4. ANSIBLE을 사용하여 속성이 IDM RBAC 권한의 멤버인지 확인

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 속성이 **IdM**의 **RBAC** 권한 멤버인지 확인하는 방법을 설명합니다. 결과적으로 권한이 있는 사용자는 특성이 있는 항목을 생성할 수 있습니다.

이 예제에서는 **MyPermission** 권한이 포함된 사용자가 생성한 호스트 항목에 **gecos** 및 **description** 값을 가질 수 있는지 확인하는 방법을 설명합니다.



참고

권한을 생성할 때 지정할 수 있는 속성 유형은 **IdM LDAP** 스키마에 의해 제한되지 않습니다. 그러나 예를 들어 **object_type** 이 호스트가 나중에 **host** 인 경우 **attrs: car_licence** 를 지정하면 권한을 수행하고 호스트에 특정 자동차 라이선스 값을 추가하려고 할 때 **ipa: ERROR: attribute "car-license"가 오류 메시지를 허용하지 않습니다.**

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.
- **MyPermission** 권한이 있습니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

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

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

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

- 사용 사례에 맞게 작업 이름을 조정합니다.
- `ipaadmin_password` 변수를 **IdM** 관리자의 암호로 설정합니다.
- `name` 변수를 권한 이름으로 설정합니다.
- `attrs` 목록을 `description` 및 `gecos` 변수로 설정합니다.
- `action` 변수가 `member` 로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission member present example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that the "gecos" and "description" attributes are present in
      "MyPermission"
      ipapermission:
        ipaadmin_password: Secret123
        name: MyPermission
        attrs:
          - description
          - geccos
        action: member
```

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

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

14.5. ANSIBLE을 사용하여 속성이 **IDM RBAC** 권한의 멤버가 아닌지 확인

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어(**RBAC**)를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 속성이 **IdM**의 **RBAC** 권한 멤버가 아닌지 확인하는 방법을 설명합니다. 결과적으로 권한이 있는 사용자가 **IdM LDAP**에 항목을 생성하면 해당 항목에 속성과 연결된 값을 가질 수 없습니다.

이 예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한이 포함된 권한이 있는 사용자가 생성한 호스트 항목은 **description** 속성을 가질 수 없습니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.
- **MyPermission** 권한이 있습니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

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

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

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

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

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.
- **attrs** 변수를 설명으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.
- **state** 변수가 **absent**로 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission absent example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure that an attribute is not a member of "MyPermission"
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      attrs: description
      action: member
      state: absent
```

5. 파일을 저장합니다.

6.

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

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

14.6. ANSIBLE을 사용하여 IDM RBAC 권한 이름 변경

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 권한의 이름을 바꾸는 방법을 설명합니다. 이 예제에서는 **MyPermission**의 이름을 **MyNewPermission**으로 변경하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.
- **IdM**에 **MyPermission**이 있습니다.
- **IdM**에는 **MyNewPermission**이 없습니다.

절차

1.

~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2.

/usr/share/doc/ansible-freeipa/playbooks/permission/ 디렉터리에 있는 **permission-renamed.yml** 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-renamed.yml
permission-renamed-copy.yml
```

3. 편집할 **permission-renamed-copy.yml** Ansible 플레이북 파일을 엽니다.

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

- 사용 사례에 맞게 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission present example
  hosts: ipaserver
  become: true

  tasks:
  - name: Rename the "MyPermission" permission
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      rename: MyNewPermission
      state: renamed
```

5. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i inventory permission-renamed-copy.yml
```

14.7. 추가 리소스

- [IdM의 권한](#)을 참조하십시오.
- [IdM의 권한](#) 참조.
- `/usr/share/doc/ansible-freeipa/` 디렉터리에서 사용 가능한 **README-permission** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/ipapermission` 디렉터리에서 샘플 플레이북을 참조하십시오.

15장. ANSIBLE을 사용하여 IDM의 복제 토폴로지 관리

여러 IdM(Identity Management) 서버를 유지 관리하고 중복을 위해 서버 손실을 완화하거나 방지하기 위해 서로 복제할 수 있습니다. 예를 들어, 한 서버가 실패하면 다른 서버에서 도메인에 서비스를 계속 제공할 수 있습니다. 나머지 서버 중 하나를 기반으로 새 복제본을 생성하여 손실된 서버를 복구할 수도 있습니다.

IdM 서버에 저장된 데이터는 복제 계약을 기반으로 복제됩니다. 두 서버에 복제 계약이 구성된 경우 해당 데이터를 공유합니다. 복제된 데이터는 토폴로지 접미사에 저장됩니다. 두 복제본의 접미사 간에 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. 이 장에는 다음 섹션이 포함되어 있습니다.

- [Ansible을 사용하여 IdM에 복제 계약이 있는지 확인](#)
- [Ansible을 사용하여 여러 IdM 복제본 간에 복제 계약이 존재하는지 확인](#)
- [Ansible을 사용하여 두 개의 복제본 간에 복제 계약이 있는지 확인합니다.](#)
- [Ansible을 사용하여 IdM에 토폴로지 접미사가 있는지 확인합니다.](#)
- [Ansible을 사용하여 IdM 복제본 다시 초기화](#)
- [Ansible을 사용하여 IdM에 복제 주석이 없는지 확인](#)

15.1. ANSIBLE을 사용하여 IDM에 복제 계약이 있는지 확인

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약을 기반으로 복제됩니다. 두 서버에 복제 계약이 구성된 경우 해당 데이터가 공유됩니다. 복제 계약은 항상 서로 구분되며 데이터는 첫 번째 복제본에서 다른 복제본으로 복제되고 다른 복제본에서 첫 번째 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

- 토폴로지의 복제본 연결에 나열된 **IdM** 토폴로지 설계에 대한 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/` 디렉터리에서 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/topology/` 디렉터리에 있는 `add-topologysegment.yml` Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegment.yml  
add-topologysegment-copy.yml
```

3. 편집할 `add-topologysegment-copy.yml` 파일을 엽니다.
4. `ipatopologysegment` 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
 - `ipaadmin_password` 변수를 **IdM** 관리자 암호로 설정합니다.

- 추가할 세그먼트 유형에 따라 접미사 변수를 **domain** 또는 **ca** 로 설정합니다.
- 왼쪽 변수를 복제 계약의 왼쪽 노드가 될 **IdM** 서버의 이름으로 설정합니다.
- 올바른 변수를 복제 계약의 올바른 노드가 될 **IdM** 서버의 이름으로 설정합니다.
- **state** 변수가 **present** 로 설정되어 있는지 확인합니다.

현재 예제에 대해 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 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory add-topologysegment-copy.yml
```

추가 리소스

- [복제 계약](#), [Topology Suffixes](#), [Topology Segments](#)를 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리에 있는 **README-topology.md** 파일을 참조하십시오.
-

`/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉터리에서 샘플 플레이북을 참조하십시오.

15.2. ANSIBLE을 사용하여 여러 IDM 복제본 간에 복제 계약이 존재하는지 확인

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약을 기반으로 복제됩니다. 두 서버에 복제 계약이 구성된 경우 해당 데이터가 공유됩니다. 복제 계약은 항상 서로 구분되며 데이터는 첫 번째 복제본에서 다른 복제본으로 복제되고 다른 복제본에서 첫 번째 복제본으로 복제됩니다.

이 섹션에서는 **IdM**의 여러 복제본 쌍 간에 복제 계약이 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/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.

편집하기 위해 **add-topologysegments-copy.yml** 파일을 엽니다.

4.

vars 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- 모든 토폴로지 세그먼트에서 **ipatopology_segments** 섹션에 행을 추가하고 다음 변수를 설정합니다.
 - 추가할 세그먼트 유형에 따라 접미사 변수를 **domain** 또는 **ca** 로 설정합니다.
 - 왼쪽 변수를 복제 계약의 왼쪽 노드가 될 **IdM** 서버의 이름으로 설정합니다.
 - 올바른 변수를 복제 계약의 올바른 노드가 될 **IdM** 서버의 이름으로 설정합니다.

5.

add-topologysegments-copy.yml 파일의 **tasks** 섹션에서 **state** 변수가 **present** 로 설정되어 있는지 확인합니다.

현재 예제에 대해 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 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory add-topologysegments-copy.yml
```

추가 리소스

- [복제 계약](#), [Topology Suffixes](#), [Topology Segments](#)를 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리에 있는 `README-topology.md` 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉토리에서 샘플 플레이북을 참조하십시오.

15.3. ANSIBLE을 사용하여 두 개의 복제본 간에 복제 계약이 있는지 확인합니다.

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약을 기반으로 복제됩니다. 두 서버에 복제 계약이 구성된 경우 해당 데이터가 공유됩니다. 복제 계약은 항상 서로 구분되며 데이터는 첫 번째 복제본에서 다른 복제본으로 복제되고 다른 복제본에서 첫 번째 복제본으로 복제됩니다.

이 섹션에서는 **IdM**의 여러 복제본 쌍 간에 복제 계약이 존재하는지 확인하는 방법을 설명합니다.

사전 요구 사항

- 토폴로지의 복제본 연결에 나열된 **IdM(Identity Management)** 토폴로지를 설계하기 위한 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/` 디렉터리에서 이러한 옵션을 구성하는 **IdM** 서버의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/topology/` 디렉터리에 있는 **check-topology.yml** Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/check-topologysegments.yml  
check-topologysegments-copy.yml
```

3. 편집을 위해 **check-topologysegments-copy.yml** 파일을 엽니다.
4. **vars** 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- 모든 토폴로지 세그먼트에서 **ipatopology_segments** 섹션에 행을 추가하고 다음 변수

를 설정합니다.

- 추가하는 세그먼트 유형에 따라 접미사 변수를 **domain** 또는 **ca** 로 설정합니다.
- 왼쪽 변수를 복제 계약의 왼쪽 노드가 될 **IdM** 서버의 이름으로 설정합니다.
- 올바른 변수를 복제 계약의 올바른 노드가 될 **IdM** 서버의 이름으로 설정합니다.

5.

check-topologysegments-copy.yml 파일의 **tasks** 섹션에서 **state** 변수가 **present** 로 설정되어 있는지 확인합니다.

현재 예제에 대해 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 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory check-topologysegments-copy.yml
```

추가 리소스

- 복제 계약, [Topology Suffixes](#), [Topology Segments](#)를 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리에 있는 `README-topology.md` 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉터리에서 샘플 플레이북을 참조하십시오.

15.4. ANSIBLE을 사용하여 IDM에 토폴로지 접미사가 있는지 확인합니다.

IdM(Identity Management)의 복제 계약 컨텍스트에서 토폴로지 접미사는 복제된 데이터를 저장합니다. **IdM**은 **domain** 및 **ca**의 두 가지 유형의 토폴로지 접미사를 지원합니다. 각 접미사는 별도의 백엔드인 별도의 복제 토폴로지를 나타냅니다. 복제 계약이 구성되면 두 개의 다른 서버에 동일한 유형의 토폴로지 접미사가 사용됩니다.

도메인 접미사에는 사용자, 그룹 및 정책과 같은 모든 도메인 관련 데이터가 포함됩니다. **ca** 접미사에는 인증서 시스템 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 설명합니다. 이 예제에서는 **IdM**에 도메인 접미사가 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/ 디렉터리에서 이러한 옵션을 구성하는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

절차

1.

~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2.

/usr/share/doc/ansible-freeipa/playbooks/topology/ 디렉터리에 있는 verify-topologysuffix.yml Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/ verify-topologysuffix.yml
verify-topologysuffix-copy.yml
```

3.

편집할 verify-topologysuffix-copy.yml Ansible 플레이북 파일을 엽니다.

4.

ipatologysuffix 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

•

ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.

•

접미사 변수를 domain 으로 설정합니다. ca 접미사가 있는지 확인하는 경우 변수를 ca 로 설정합니다.

•

state 변수가 검증 되도록 설정되어 있는지 확인합니다. 다른 옵션은 불가능합니다.

현재 예제에 대해 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
    - name: Verify topology suffix
      ipatologysuffix:
```

```
ipaadmin_password: Secret123
suffix: domain
state: verified
```

5.

파일을 저장합니다.

6.

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

```
$ ansible-playbook -v -i inventory verify-topologysuffix-copy.yml
```

추가 리소스

•

[복제 계약](#), [Topology Suffixes](#), [Topology Segments](#)를 참조하십시오.

•

/usr/share/doc/ansible-freeipa/ 디렉토리에 있는 **README-topology.md** 파일을 참조하십시오.

•

/usr/share/doc/ansible-freeipa/playbooks/topology 디렉터리에서 샘플 플레이북을 참조하십시오.

15.5. ANSIBLE을 사용하여 IDM 복제본 다시 초기화

복제본이 오랜 기간 동안 오프라인된 경우 또는 해당 데이터베이스가 손상된 경우 다시 초기화할 수 있습니다. 업데이트된 데이터 세트로 복제본을 새로 고칩니다. 예를 들어 백업에서 신뢰할 수 있는 복원이 필요한 경우 다시 초기화할 수 있습니다.



참고

복제 업데이트와 달리 복제본에서 변경된 항목만 서로 다시 보내면 전체 데이터베이스를 새로 고칩니다. **In contrast to replication updates, during which replicas only send changed entries to each other, reinitialization refreshes the whole database.**

명령을 실행하는 로컬 호스트는 **reinitialized** 복제본입니다. 데이터를 가져올 복제본을 지정하려면 **direction** 옵션을 사용합니다.

이 섹션에서는 **Ansible** 플레이북을 사용하여 **server.idm.example.com** 에서 **replica.idm.example.com** 에서 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/` 디렉터리에서 이러한 옵션을 구성하는 **IdM 서버의 FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/topology/` 디렉터리에 있는 `reinitialize-topologys.yml` Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/reinitialize-topologysegment.yml reinitialize-topologysegment-copy.yml
```

3. 편집할 `reinitialize-topologysegment-copy.yml` 파일을 엽니다.
4. `ipatopologysegment` 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM 관리자 암호**로 설정합니다.
- 접미사 변수를 **domain** 으로 설정합니다. **ca** 데이터를 다시 시작하는 경우 변수를 **ca** 로 설정합니다.

- 왼쪽 변수를 복제 계약의 왼쪽 노드로 설정합니다.
- 올바른 변수를 복제 계약의 올바른 노드로 설정합니다.
- **direction** 변수를 **reinitializing** 데이터의 방향으로 설정합니다. 왼쪽에서 오른쪽 방향은 데이터가 왼쪽 노드에서 오른쪽 노드로 전송됨을 의미합니다.
- **state** 변수가 다시 초기화 되도록 설정되어 있는지 확인합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - name: Reinitialize topology segment
    ipatopologysegment:
      ipadmin_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 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory reinitialize-topologysegment-copy.yml
```

추가 리소스

- 복제 계약, [Topology Suffixes](#), [Topology Segments](#)를 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리에 있는 **README-topology.md** 파일을 참조하십시오

시오.

- `/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉터리에서 샘플 플레이북을 참조하십시오.

15.6. ANSIBLE을 사용하여 IDM에 복제 주석이 없는지 확인

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약을 기반으로 복제됩니다. 두 서버에 복제 계약이 구성된 경우 해당 데이터가 공유됩니다. 복제 계약은 항상 서로 구분되며 데이터는 첫 번째 복제본에서 다른 복제본으로 복제되고 다른 복제본에서 첫 번째 복제본으로 복제됩니다.

이 섹션에서는 두 복제본 간의 복제 계약이 **IdM**에 없는지 확인하는 방법에 대해 설명합니다. 이 예제에서는 `replica01.idm.example.com` 과 `replica02.idm.example.com` **IdM** 서버 간에 도메인 유형의 복제 계약이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. `~/MyPlaybooks/` 디렉터리로 이동합니다.

■

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

2.

/usr/share/doc/ansible-freeipa/playbooks/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.

편집할 **delete-topologysegment-copy.yml** 파일을 엽니다.

4.

ipatopologysegment 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- 접미사 변수를 **domain** 으로 설정합니다. 또는 왼쪽과 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.

현재 예제에 대해 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 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory delete-topologysegment-copy.yml
```

추가 리소스

- [복제 계약, Topology Suffixes, Topology Segments](#)를 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리에 있는 **README-topology.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/topology` 디렉터리에서 샘플 플레이북을 참조하십시오.

15.7. 추가 리소스

- [복제본 토폴로지 계획에서](#) 참조하십시오.
- [IdM 복제본 설치](#)를 참조하십시오.

16장. ANSIBLE을 사용하여 IDM 서버 관리

Red Hat Ansible Engine 을 사용하여 IdM(Identity Management) 토폴로지의 서버를 관리할 수 있습니다. **ansible-freeipa** 패키지에서 **server** 모듈을 사용하여 IdM 토폴로지에 서버가 있는지 확인할 수 있습니다. 복제본을 숨기거나 복제본을 볼 수도 있습니다.

섹션에는 다음 항목이 포함되어 있습니다.

- [Ansible을 사용하여 IdM 서버가 있는지 확인](#)
- [Ansible을 사용하여 IdM 서버가 IdM 토폴로지에서 없음을 확인](#)
- [마지막 IdM 서버 역할을 호스팅해도 IdM 서버가 없는지 확인](#)
- [IdM 서버가 없음을 확인하지만 다른 IdM 서버와 분리할 필요는 없습니다.](#)
- [Ansible 플레이북을 사용하여 기존 IdM 서버가 숨겨져 있는지 확인합니다.](#)
- [Ansible 플레이북을 사용하여 기존 IdM 서버가 표시되는지 확인합니다.](#)
- [기존 IdM 서버에 IdM DNS 위치가 할당되어 있는지 확인합니다.](#)
- [기존 IdM 서버에 IdM DNS 위치가 할당되지 않았는지 확인합니다.](#)

16.1. ANSIBLE을 사용하여 IDM 서버가 있는지 확인

Ansible 플레이북에서 **ipaserver ansible-freeipa** 모듈을 사용하여 IdM(Identity Management) 서버가 있는지 확인할 수 있습니다.



참고

ipaserver Ansible 모듈은 **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/` 디렉터리에 참조하려는 **LDAP** 데이터베이스의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 제대로 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. 편집할 **server-present-copy.yml** 파일을 엽니다.

4.

ipaserver 작업 섹션에서 다음 변수를 설정하고 파일을 저장하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 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 플레이북을 사용하여 ID 관리 서버 설치를 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-server.md 파일을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/playbooks/server 디렉터리에서 샘플 플레이북을 참조하십시오.](#)

16.2. ANSIBLE을 사용하여 IDM 서버가 IDM 토폴로지에서 없음을 확인

Ansible 플레이북을 사용하여 **IdM(Identity Management)** 서버가 호스트로도 **IdM** 토폴로지에 없는지 확인합니다.

ansible-freeipa ipaserver 역할과 달리 이 플레이북에서 사용되는 **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/` 디렉토리에는 참조 및 수정하려는 **LDAP** 데이터베이스의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 제대로 작동합니다.

절차

1. `~/MyPlaybooks/` 디렉토리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/server/` 디렉토리에 있는 `server-absent.yml` **Ansible** 플레이북 파일을 복사합니다.

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

3. 편집할 `server-absent-copy.yml` 파일을 엽니다.

4. `ipaserver` 작업 섹션에서 다음 변수를 설정하고 파일을 저장하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 **IdM** 관리자 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:
      ipadmin_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. **server123.idm.example.com** 을 가리키는 모든 이름 서버(NS) **DNS** 레코드가 **DNS** 영역에서 삭제되었는지 확인합니다. 이는 **IdM** 또는 외부 **DNS**에서 관리하는 통합 **DNS**를 사용하는지 여부에 관계없이 적용됩니다.

추가 리소스

- **IdM 서버 제거**를 참조하십시오.
- **/usr/share/doc/ansible-freeipa/** 디렉토리의 **README-server.md** 파일을 참조하십시오.
- **/usr/share/doc/ansible-freeipa/playbooks/server** 디렉터리에서 샘플 플레이북을 참조하십시오.

16.3. 마지막 IDM 서버 역할을 호스팅해도 IDM 서버가 없는지 확인

Ansible을 사용하여 서버에서 마지막 **IdM** 서비스 인스턴스가 실행 중인 경우에도 **IdM(Identity Management)** 서버가 없는지 확인할 수 있습니다. **CA**(인증 기관), 키 복구 기관(**KRA**) 또는 **DNS** 서버는 모두 **IdM** 서비스의 예입니다.



주의

CA, KRA 또는 DNS 서버로 사용되는 마지막 서버를 제거하면 **IdM** 기능이 크게 중단됩니다. **ipa service-find** 명령을 사용하여 어떤 **IdM** 서버에서 실행 중인지 수동으로 확인할 수 있습니다. **CA** 서버의 기본 이름은 **dogtag/server_name/tektonM_NAME** 입니다.

ansible-freeipa ipaserver 역할과 달리 이 플레이북에서 사용되는 **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/** 디렉터리에는 참조 및 수정하려는 **LDAP** 데이터베이스의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible 인벤토리 파일**을 생성했습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 제대로 작동합니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/server/** 디렉터리에 있는 **server-absent-**

ignore-last-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.

편집할 **server-absent-ignore-last-role-copy.yml** 파일을 엽니다.

4.

ipaserver 작업 섹션에서 다음 변수를 설정하고 파일을 저장하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 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.

server123.idm.example.com 을 가리키는 모든 이름 서버(**NS**) **DNS** 레코드가 **DNS** 영역에서 삭제되었는지 확인합니다. 이는 **IdM** 또는 외부 **DNS**에서 관리하는 통합 **DNS**를 사용하는지 여부에 관계없이 적용됩니다.

추가 리소스

- [IdM 서버 제거](#)를 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 `README-server.md` 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/server` 디렉터리에서 샘플 플레이북을 참조하십시오.

16.4. IDM 서버가 없음을 확인하지만 다른 IDM 서버와 분리할 필요는 없습니다.

토폴로지에서 **IdM(Identity Management)** 서버를 제거하는 경우 **Ansible** 플레이북을 사용하여 복제 계약을 그대로 유지할 수 있습니다. 또한 이 플레이북은 **IdM** 서버가 호스트로도 **IdM**에 존재하지 않도록 합니다.



중요

제거하려는 다른 서버가 제대로 작동하는 경우에만 서버의 복제 계약을 무시하는 것이 좋습니다. 토폴로지의 중앙 지점 역할을 하는 서버를 제거하면 토폴로지를 두 개의 연결이 끊긴 클러스터로 분할할 수 있습니다.

`ipa server-del` 명령을 사용하여 토폴로지에서 일관성 있는 서버를 제거할 수 있습니다.



참고

CA(인증 기관), **키 복구 기관(KRA)** 또는 **DNS** 서버 역할을 하는 마지막 서버를 제거하면 **IdM(Identity Management)** 기능이 크게 손상됩니다. 이 문제를 방지하기 위해 플레이북에서는 **CA**, **KRA** 또는 **DNS** 서버 역할을 하는 서버를 제거하기 전에 이러한 서비스가 도메인의 다른 서버에서 실행 중인지 확인합니다.

`ansible-freeipa ipaserver` 역할과 달리 이 플레이북에서 사용되는 `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/` 디렉터리에는 참조 및 수정하려는 **LDAP** 데이터베이스의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** **인벤토리 파일**을 생성했습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 제대로 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. 편집할 **server-absent-ignore_topology_disconnect-copy.yml** 파일을 엽니다.
4. **ipaserver** 작업 섹션에서 다음 변수를 설정하고 파일을 저장하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 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:
        ipadmin_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] **server123.idm.example.com** 을 가리키는 모든 이름 서버(NS) DNS 레코드가 DNS 영역에서 삭제되었는지 확인합니다. 이는 IdM 또는 외부 DNS에서 관리하는 통합 DNS를 사용하는지 여부에 관계없이 적용됩니다.

추가 리소스

- [IdM 서버 제거](#)를 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-server.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/server` 디렉터리에서 샘플 플레이북을 참조하십시오.

16.5. ANSIBLE 플레이북을 사용하여 기존 IDM 서버가 숨겨져 있는지 확인합니다.

Ansible 플레이북에서 **ipaserver ansible-freeipa** 모듈을 사용하여 기존 **IdM(Identity Management)** 서버가 숨겨졌는지 확인합니다. 이 **Playbook**은 **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/` 디렉토리에는 참조 및 수정하려는 **LDAP** 데이터베이스의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 제대로 작동합니다.

절차

1. `~/MyPlaybooks/` 디렉토리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/server/` 디렉토리에 있는 `server-hidden.yml` **Ansible** 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-hidden.yml server-hidden-copy.yml
```

3. 편집할 `server-hidden-copy.yml` 파일을 엽니다.

4. `ipaserver` 작업 섹션에서 다음 변수를 설정하고 파일을 저장하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 **IdM** 관리자 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

- [Ansible 플레이북을 사용하여 ID 관리 서버 설치를 참조하십시오.](#)
- [숨겨진 복제본 모드를 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-server.md 파일을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/playbooks/server 디렉터리에서 샘플 플레이북을 참조하십시오.](#)

16.6. ANSIBLE 플레이북을 사용하여 기존 IDM 서버가 표시되는지 확인합니다.

Ansible 플레이북에서 **ipaserver ansible-freeipa** 모듈을 사용하여 기존 **IdM(Identity Management)** 서버가 표시되는지 확인합니다. 이 **Playbook**은 **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/` 디렉토리에는 참조 및 수정하려는 **LDAP** 데이터베이스의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 제대로 작동합니다.

절차

1. `~/MyPlaybooks/` 디렉토리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/server/` 디렉토리에 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. 편집할 **server-not-hidden-copy.yml** 파일을 엽니다.
4. **ipaserver** 작업 섹션에서 다음 변수를 설정하고 파일을 저장하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 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
```

추가 리소스

- [Ansible 플레이북을 사용하여 ID 관리 서버 설치를 참조하십시오.](#)
- [숨겨진 복제본 모드를 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-server.md 파일을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/playbooks/server 디렉터리에서 샘플 플레이북을 참조하십시오.](#)

16.7. 기존 IDM 서버에 IDM DNS 위치가 할당되어 있는지 확인합니다.

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. 예를 들어 **location**은 **germany** 입니다.
- 서버에 대한 루트 액세스 권한이 있어야 합니다. 예제 서버는 **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/` 디렉토리에는 참조 및 수정하려는 **LDAP** 데이터베이스의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 인벤토리 파일을 생성했습니다.
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 제대로 작동합니다.

절차

1. `~/MyPlaybooks/` 디렉토리로 이동합니다.

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

2. `/usr/share/doc/ansible-freeipa/playbooks/server/` 디렉토리에 있는 **server-location.yml** **Ansible** 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-location.yml server-location-copy.yml
```

3. 편집할 **server-location-copy.yml** 파일을 엽니다.

4.

ipaserver 작업 섹션에서 다음 변수를 설정하고 파일을 저장하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 **server123.idm.example.com** 으로 설정합니다.
- 위치 변수를 **germany** 로 설정합니다.

현재 예제에 대해 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.

업데이트가 즉시 적용되도록 서버에서 **named-pkcs11** 서비스를 다시 시작하십시오.

```
[root@server123.idm.example.com ~]# systemctl restart named-pkcs11
```

추가 리소스

- [Ansible 플레이북을 사용하여 ID 관리 서버 설치를 참조하십시오.](#)

- **IdM 위치가 있는지 확인하려면 Ansible 사용을 참조하십시오.**
- **/usr/share/doc/ansible-freeipa/ 디렉토리의 README-server.md 파일을 참조하십시오.**
- **/usr/share/doc/ansible-freeipa/playbooks/server 디렉터리에서 샘플 플레이북을 참조하십시오.**

16.8. 기존 IDM 서버에 IDM DNS 위치가 할당되지 않았는지 확인합니다.

Ansible 플레이북에서 **ipaserver ansible-freeipa** 모듈을 사용하여 기존 **IdM(Identity Management)** 서버에 **IdM DNS** 위치가 할당되어 있지 않은지 확인합니다. 지리적 위치를 자주 변경하는 서버에 **DNS** 위치를 할당하지 마십시오. **Playbook**은 **IdM** 서버를 설치하지 않습니다.

사전 요구 사항

- **IdM 관리자 암호를 알고 있습니다.**
- 서버에 대한 루트 액세스 권한이 있어야 합니다. 예제 서버는 **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/ 디렉터리에는 참조 및 수정하려는 LDAP 데이터베이스의 FQDN(정규화된 도메인 이름)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.**
 - 제어 노드에서 인벤토리 파일에 정의된 **IdM** 서버로의 **SSH** 연결이 제대로 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 설정하고 파일을 저장하여 파일을 조정합니다.

- `ipaadmin_password` 변수를 IdM 관리자 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.

업데이트가 즉시 적용되도록 서버에서 **named-pkcs11** 서비스를 다시 시작하십시오.

```
[root@server123.idm.example.com ~]# systemctl restart named-pkcs11
```

추가 리소스

- [Ansible 플레이북을 사용하여 ID 관리 서버 설치를 참조하십시오.](#)
- [Ansible을 사용하여 IdM의 DNS 위치 관리를 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-server.md 파일을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/playbooks/server 디렉터리에서 샘플 플레이북을 참조하십시오.](#)

17장. ANSIBLE 플레이북을 사용하여 호스트 관리

Ansible은 시스템 구성, 소프트웨어 배포 및 롤링 업데이트를 수행하는 데 사용되는 자동화 틀입니다. **Ansible**에는 **IdM(Identity Management)** 지원이 포함되어 있으며 **Ansible** 모듈을 사용하여 호스트 관리를 자동화할 수 있습니다.

이 장에서는 **Ansible** 플레이북을 사용하여 호스트 및 호스트 항목을 관리할 때 수행되는 다음 개념과 작업을 설명합니다.

- **FQDN**으로만 정의된 **IdM** 호스트 항목이 있는지 확인
- **IP** 주소를 사용하여 **IdM** 호스트 항목이 있는지 확인
- 임의의 암호로 여러 **IdM** 호스트 항목이 있는지 확인
- 여러 **IP** 주소로 **IdM** 호스트 항목이 있는지 확인
- **IdM** 호스트 항목이 없는지 확인

17.1. ANSIBLE 플레이북을 사용하여 FQDN으로 IDENTITY 호스트 항목이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 호스트 항목이 있는지 확인하는 방법을 설명합니다. 호스트 항목은 **FQDN**(정규화된 도메인 이름)으로만 정의됩니다.

다음 조건 중 하나가 적용되는 경우 호스트의 **FQDN** 이름을 지정하면 충분합니다.

- **IdM** 서버는 **DNS**를 관리하도록 구성되지 않았습니다.
- 호스트에 고정 **IP** 주소가 없거나 호스트가 구성된 시점에 **IP** 주소를 알 수 없습니다. **FQDN** 으로만 정의된 호스트를 추가하면 기본적으로 **IdM DNS** 서비스에 자리 표시자 항목이 생성됩니다. 예를 들어 랩탑은 **IdM** 클라이언트로 사전 구성될 수 있지만, 구성된 시점에는 **IP** 주소가 없습니다. **DNS** 서비스가 레코드를 동적으로 업데이트하면 호스트의 현재 **IP** 주소가 감지되고 해당 **DNS** 레코드가 업데이트됩니다.



참고

Ansible이 없으면 `ipa host-add` 명령을 사용하여 IdM에 호스트 항목이 생성됩니다. IdM에 호스트를 추가하는 결과는 IdM에 호스트 상태가 됩니다. Ansible의 **idempotence on idempotence** 때문에 Ansible을 사용하여 호스트를 IdM에 추가하려면 호스트 상태를 **present: state: present** 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

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

절차

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

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

2. 확인하려는 IdM에 있는 호스트의 FQDN 으로 Ansible 플레이북 파일을 생성합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/host/add-host.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Host present
  hosts: ipaserver
  become: true

  tasks:
    - name: Host host01.idm.example.com present
      ipahost:
        ipadmin_password: MySecret123
        name: host01.idm.example.com
        state: present
        force: yes
```

3. 플레이북을 실행합니다.

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



참고

이 절차를 수행하면 **IdM LDAP** 서버의 호스트 항목이 생성되지만 **IdM Kerberos** 영역에 호스트를 등록하지 않습니다. 이를 위해 호스트를 **IdM 클라이언트**로 배포해야 합니다. 자세한 내용은 [Ansible 플레이북을 사용하여 ID 관리 클라이언트 설치](#)를 참조하십시오.

검증 단계

1.

IdM 서버에 **admin**으로 로그인합니다.

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

2.

ipa host-show 명령을 입력하고 호스트 이름을 지정합니다.

```
$ ipa host-show host01.idm.example.com
Host name: host01.idm.example.com
Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
Password: False
Keytab: False
Managed by: host01.idm.example.com
```

출력은 **IdM**에 **host01.idm.example.com** 이 있는지 확인합니다.

17.2. ANSIBLE 플레이북을 사용하여 DNS 정보로 IDM 호스트 항목이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 호스트 항목이 있는지 확인하는 방법을 설명합니다. 호스트 항목은 **FQDN**(정규화된 도메인 이름) 및 해당 **IP** 주소로 정의됩니다.



참고

Ansible이 없으면 **ipa host-add** 명령을 사용하여 **IdM**에 호스트 항목이 생성됩니다. **IdM**에 호스트를 추가하는 결과는 **IdM**에 호스트 상태가 됩니다. **Ansible**의 **idempotence on idempotence** 때문에 **Ansible**을 사용하여 호스트를 **IdM**에 추가하려면 호스트 상태를 **present: state: present** 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

•

IdM 관리자 암호를 알고 있습니다.

•

ansible-freeipa 패키지는 **Ansible** 컨트롤러에 설치됩니다.

절차

1.

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

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

2.

보장하려는 **IdM**에 있는 호스트의 **FQDN**(정규화된 도메인 이름)을 사용하여 **Ansible** 플레이북 파일을 생성합니다. 또한 **IdM** 서버가 **DNS**를 관리하도록 구성되어 호스트의 **IP** 주소를 알고 있는 경우 **ip_address** 매개 변수의 값을 지정합니다. 호스트가 **DNS** 리소스 레코드에 존재하려면 **IP** 주소가 필요합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/host/host-present.yml** 파일에서 예제를 복사하고 수정할 수 있습니다. 다른 추가 정보를 포함할 수도 있습니다.

```
---
- name: Host present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure host01.idm.example.com is present
    ipahost:
      ipaadmin_password: MySecret123
      name: host01.idm.example.com
      description: Example host
      ip_address: 192.168.0.123
      locality: Lab
      ns_host_location: Lab
      ns_os_version: CentOS 7
      ns_hardware_platform: Lenovo T61
      mac_address:
        - "08:00:27:E3:B1:2D"
        - "52:54:00:BD:97:1E"
      state: present
```

3.

플레이북을 실행합니다.

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



참고

이 절차를 수행하면 **IdM LDAP** 서버의 호스트 항목이 생성되지만 **IdM Kerberos** 영역에 호스트를 등록하지 않습니다. 이를 위해 호스트를 **IdM** 클라이언트로 배포해야 합니다. 자세한 내용은 [Ansible 플레이북을 사용하여 ID 관리 클라이언트 설치](#)를 참조하십시오.

검증 단계

1.

IdM 서버에 **admin**으로 로그인합니다.

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

2.

ipa host-show 명령을 입력하고 호스트 이름을 지정합니다.

```
$ ipa host-show host01.idm.example.com
Host name: host01.idm.example.com
Description: Example host
Locality: Lab
Location: Lab
Platform: Lenovo T61
Operating system: CentOS 7
Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
MAC address: 08:00:27:E3:B1:2D, 52:54:00:BD:97:1E
Password: False
Keytab: False
Managed by: host01.idm.example.com
```

출력은 **IdM**에 **host01.idm.example.com** 이 있는지 확인합니다.

17.3. ANSIBLE 플레이북을 사용하여 임의의 암호로 여러 ID 호스트 항목이 있는지 확인

ipahost 모듈을 사용하면 시스템 관리자가 하나의 **Ansible** 작업만 사용하여 **IdM**에 여러 호스트 항목이 있거나 있는지 확인할 수 있습니다. 이 섹션에서는 **FQDN**(정규화된 도메인 이름)에서만 정의한 여러 호스트 항목이 있는지 확인하는 방법을 설명합니다. **Ansible** 플레이북을 실행하면 호스트에 대한 임의의 암호가 생성됩니다.



참고

Ansible이 없으면 `ipa host-add` 명령을 사용하여 IdM에 호스트 항목이 생성됩니다. IdM에 호스트를 추가하는 결과는 IdM에 호스트 상태가 됩니다. Ansible의 **idempotence on idempotence** 때문에 Ansible을 사용하여 호스트를 IdM에 추가하려면 호스트 상태를 **present: state: present** 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

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

절차

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

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

2. 보장하려는 IdM에 있는 호스트의 **FQDN**(정규화된 도메인 이름)을 사용하여 Ansible 플레이북 파일을 생성합니다. Ansible 플레이북에서 호스트가 IdM에 이미 있고 `update_password` 가 `on_create` 로 제한되더라도 각 호스트에 대해 임의의 암호를 생성하도록 하려면 **random: yes** 및 **force: yes** 옵션을 추가합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/README-host.md` Markdown 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Ensure hosts with random password
  hosts: ipaserver
  become: true

  tasks:
    - name: Hosts host01.idm.example.com and host02.idm.example.com present with
      random passwords
      ipahost:
        ipaadmin_password: MySecret123
        hosts:
          - name: host01.idm.example.com
            random: yes
            force: yes
          - name: host02.idm.example.com
            random: yes
            force: yes
        register: ipahost
```

3.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-are-present.yml
[...]
TASK [Hosts host01.idm.example.com and host02.idm.example.com present with
random passwords]
changed: [r8server.idm.example.com] => {"changed": true, "host":
{"host01.idm.example.com": {"randompassword": "0HolRvjUdH0Ycbf6uYdWTxH"},
"host02.idm.example.com": {"randompassword": "5VdLgrf3wvojmACdHC3uA3s"}}
```



참고

임의 일회성 암호(OTP)를 사용하여 호스트를 IdM 클라이언트로 배포하려면 [Ansible 플레이북을 사용하여 IdM 클라이언트 등록에 대한 권한 부여 옵션](#)을 참조하거나 일회성 암호: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오.

검증 단계

1.

IdM 서버에 **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
```

출력에서는 임의의 암호를 사용하여 IdM에 **host01.idm.example.com** 이 있는지 확인합니다.

17.4. ANSIBLE 플레이북을 사용하여 여러 IP 주소로 IDM 호스트 항목이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 호스트 항목이 있는지 확인하는 방법을 설명합니다. 호스트 항목은 **FQDN**(정규화된 도메인 이름)과 해당 다중 **IP** 주소로 정의됩니다.



참고

ipa host 유틸리티와 달리 **Ansible ipahost** 모듈은 호스트에 여러 **IPv4** 및 **IPv6** 주소가 있는지 확인할 수 있습니다. **ipa host-mod** 명령은 **IP** 주소를 처리할 수 없습니다.

사전 요구 사항

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

절차

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

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

2. **Ansible** 플레이북 파일을 생성합니다. **ipahost** 변수의 이름으로, 보장하려는 **IdM**에 있는 호스트의 **FQDN**(정규화된 도메인 이름)을 지정합니다. **ip_address** 구문을 사용하여 별도의 행에 여러 **IPv4** 및 **IPv6** **ip_address** 값을 지정합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/host/host-member-ipaddresses-present.yml** 파일에서 예제를 복사하고 수정할 수 있습니다. 추가 정보를 포함할 수도 있습니다.

```
---
- name: Host member IP addresses present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure host101.example.com IP addresses present
    ipahost:
      ipadmin_password: MySecret123
      name: host01.idm.example.com
      ip_address:
        - 192.168.0.123
        - fe80::20c:29ff:fe02:a1b3
        - 192.168.0.124
        - fe80::20c:29ff:fe02:a1b4
      force: yes
```

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-host-with-multiple-IP-addresses-is-present.yml
```



참고

이 절차에서는 **IdM LDAP** 서버에 호스트 항목을 생성하지만, **IdM Kerberos** 영역에 호스트를 등록하지 않습니다. 이를 위해 호스트를 **IdM** 클라이언트로 배포해야 합니다. 자세한 내용은 [Ansible 플레이북을 사용하여 ID 관리 클라이언트 설치](#)를 참조하십시오.

검증 단계

1.

IdM 서버에 **admin**으로 로그인합니다.

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

2.

ipa host-show 명령을 입력하고 호스트 이름을 지정합니다.

```
$ ipa host-show host01.idm.example.com
Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
Password: False
Keytab: False
Managed by: host01.idm.example.com
```

출력은 **IdM**에 **host01.idm.example.com** 이 있는지 확인합니다.

3.

IdM DNS 레코드에 호스트의 여러 **IP** 주소가 있는지 확인하려면 **ipa dnsHistory-show** 명령을 입력하고 다음 정보를 지정합니다.

- **IdM** 도메인의 이름
- 호스트 이름

```
$ ipa dnsrecord-show idm.example.com host01
[...]
Record name: host01
A record: 192.168.0.123, 192.168.0.124
AAAA record: fe80::20c:29ff:fe02:a1b3, fe80::20c:29ff:fe02:a1b4
```

출력에서 플레이북에 지정된 모든 IPv4 및 IPv6 주소가 `host01.idm.example.com` 호스트 항목과 올바르게 연결되었는지 확인합니다.

17.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 항목이 없는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에서 호스트 항목이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 인증 정보

절차

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

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

2. 보장하려는 IdM에 없는 호스트의 FQDN(정규화된 도메인 이름)을 사용하여 Ansible 플레이북 파일을 생성합니다. IdM 도메인에 DNS가 통합된 경우 `updatedns: yes` 옵션을 사용하여 DNS에서 호스트에 대한 모든 종류의 연결된 레코드를 제거합니다.

이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/host/delete-host.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Host absent
  hosts: ipaserver
  become: true

  tasks:
  - name: Host host01.idm.example.com absent
    ipahost:
      ipadmin_password: MySecret123
      name: host01.idm.example.com
      updatedns: yes
      state: absent
```

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-host-absent.yml
```

참고

절차는 다음과 같습니다.

- **IdM Kerberos** 영역에는 호스트가 존재하지 않습니다.
- **IdM LDAP** 서버에 호스트 항목이 없습니다.

클라이언트 호스트 자체에서 **SSSD(System Security Services Daemon)**와 같은 시스템 서비스의 특정 **IdM** 구성을 제거하려면 클라이언트에서 **ipa-client-install --uninstall** 명령을 실행해야 합니다. 자세한 내용은 [IdM 클라이언트 설치 제거](#)를 참조하십시오.

검증 단계

1. **admin**으로 **ipaserver**에 로그인합니다.

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

2. **host01.idm.example.com**에 대한 정보를 표시합니다.

```
$ ipa host-show host01.idm.example.com
ipa: ERROR: host01.idm.example.com: host not found
```

출력은 호스트가 **IdM**에 없음을 확인합니다.

17.6. 추가 리소스

- **/usr/share/doc/ansible-freeipa/README-host.md** Markdown 파일을 참조하십시오.
- **/usr/share/doc/ansible-freeipa/playbooks/host** 디렉터리에서 추가 플레이북을 참조하십시오.

18장. 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 중첩 호스트 그룹이 없는지 확인**
- **IdM 호스트 그룹에서 멤버 관리자가 없는지 확인**

18.1. IDm의 호스트 그룹

IdM 호스트 그룹은 중요한 관리 작업, 특히 액세스 제어에 대한 제어를 중앙 집중화하는 데 사용할 수 있습니다.

호스트 그룹의 정의

호스트 그룹은 공통 액세스 제어 규칙 및 기타 특성을 가진 **IdM 호스트 세트**가 포함된 엔티티입니다. 예를 들어 회사 부서, 물리적 위치 또는 액세스 제어 요구 사항에 따라 호스트 그룹을 정의할 수 있습니다.

IdM의 호스트 그룹에는 다음이 포함될 수 있습니다.

- **IdM 서버 및 클라이언트**
- **기타 IdM 호스트 그룹**

기본적으로 생성된 호스트 그룹

기본적으로 **IdM** 서버는 모든 **IdM** 서버 호스트에 대한 호스트 그룹 **ipaservers** 를 생성합니다.

직접 및 간접 그룹 멤버

IdM의 그룹 속성은 직접 및 간접 멤버 모두에 적용됩니다. 호스트 그룹 **B**가 호스트 그룹 **A**의 멤버인 경우 호스트 그룹 **B**의 모든 멤버가 호스트 그룹 **A**의 간접 멤버로 간주됩니다.

18.2. ANSIBLE 플레이북을 사용하여 IDENTITY MANAGEMENT 호스트 그룹이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에서 호스트 그룹이 있는지 확인하는 방법을 설명합니다.



참고

Ansible이 없으면 **ipa hostgroup-add** 명령을 사용하여 **IdM**에 호스트 그룹 항목이 생성됩니다. **IdM**에 호스트 그룹을 추가하는 결과는 **IdM**에 호스트 그룹 상태가 됩니다. **Ansible**의 **idempotence on idempotence** 때문에 **Ansible**을 사용하여 호스트 그룹을 **IdM**에 추가하려면 호스트 그룹의 상태를 **present: state: present** 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

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

절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 타겟할 **IdM** 서버 목록으로 **ipaserver** 를 정의합니다.

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

2.

필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 예를 들어 **databases** 라는 호스트 그룹이 있는지 확인하려면 - **ipahostgroup** 작업에 **name: databases** 를 지정합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-present.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
    # Ensure host-group databases is present
    - ipahostgroup:
        ipaadmin_password: MySecret123
        name: databases
        state: present
```

플레이북에서 **state: present** 는 이미 존재하지 않는 한 호스트 그룹을 **IdM**에 추가하라는 요청을 나타냅니다.

3.

플레이북을 실행합니다.

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

검증 단계

1.

admin으로 **ipaserver** 에 로그인합니다.

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

2.

관리자를 위한 **Kerberos** 티켓을 요청합니다.

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

3.

다음은 확인하려는 IdM에 있는 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
Host-group: databases
```

IdM에 데이터베이스 호스트 그룹이 있습니다.

18.3. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 호스트가 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**의 호스트 그룹에 호스트가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- Ansible 플레이북에 참조하려는 호스트는 IdM에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 항목이 있는지 여부](#) 를 참조하십시오.
- Ansible 플레이북 파일에서 참조하는 호스트 그룹이 IdM에 추가되었습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 여부](#) 를 참조하십시오.

절차

1.

인벤토리 파일(예: **inventory.file**)을 생성하고 타겟할 IdM 서버 목록으로 **ipaserver** 를 정의합니다.

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

2.

필요한 호스트 정보로 **Ansible** 플레이북 파일을 생성합니다. **ipahostgroup** 변수의 **name** 매개 변수를 사용하여 호스트 그룹의 이름을 지정합니다. **ipahostgroup** 변수의 **host** 매개 변수를 사용하여 호스트 이름을 지정합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-are-hostgroups-are-in-hostgroup.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

```

---
- name: Playbook to handle hostgroups
  hosts: ipaserver
  become: true

  tasks:
    # Ensure host-group databases is present
    - ipahostgroup:
        ipaadmin_password: MySecret123
        name: databases
        host:
          - db.idm.example.com
        action: member

```

이 플레이북은 **db.idm.example.com** 호스트를 **databases** 호스트 그룹에 추가합니다. **action: member** 행은 플레이북이 실행되면 **databases** 그룹 자체를 추가하려고 시도하지 않음을 나타냅니다. 대신 **db.idm.example.com** 을 데이터베이스에 추가하려는 시도만 하면 됩니다.

3. 플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-present-in-
hostgroup.yml

```

검증 단계

1. **admin**으로 **ipaserver** 에 로그인합니다.

```

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

```

2. 관리자를 위한 **Kerberos** 티켓을 요청합니다.

```

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

```

3. 호스트 그룹에 대한 정보를 표시하여 호스트 그룹에 있는 호스트를 확인합니다.

```

$ ipa hostgroup-show databases
Host-group: databases
Member hosts: db.idm.example.com

```

db.idm.example.com 호스트는 **databases** 호스트 그룹의 멤버로 제공됩니다.

18.4. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹 중첩

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 호스트 그룹에 중첩 호스트 그룹이 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **Ansible** 플레이북 파일에서 참조하는 호스트 그룹은 **IdM**에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 여부](#)를 참조하십시오.

절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 타겟할 **IdM** 서버 목록으로 **ipaserver**를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. **Ansible** 플레이북에서 호스트 그룹 **A**가 호스트 그룹 **B**에 있는지 확인하려면 **name** 변수를 사용하여 호스트 그룹 **B**의 이름을 지정합니다. **hostgroup** 변수를 사용하여 중첩 호스트 그룹 **A**의 이름을 지정합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-are-hostgroups-are-in-hostgroup.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
    # Ensure hosts and hostgroups are present in existing databases hostgroup
    - ipahostgroup:
        ipaadmin_password: MySecret123
        name: databases
```

```

hostgroup:
- mysql-server
- oracle-server
action: member

```

이 Ansible 플레이북은 **databases** 호스트 그룹에 **mysql-server** 및 **oracle-server** 호스트 그룹이 있는지 확인합니다. **action: member** 행은 플레이북이 실행되면 IdM에 데이터베이스 그룹 자체를 추가하려고 시도하지 않음을 나타냅니다.

3. 플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-present-in-
hostgroup.yml

```

검증 단계

1. **admin**으로 **ipaserver** 에 로그인합니다.

```

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

```

2. 관리자를 위한 **Kerberos** 티켓을 요청합니다.

```

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

```

3. 중첩된 호스트 그룹이 있는 호스트 그룹에 대한 정보를 표시합니다.

```

$ ipa hostgroup-show databases
Host-group: databases
Member hosts: db.idm.example.com
Member host-groups: mysql-server, oracle-server

```

mysql-server 및 **oracle-server** 호스트 그룹은 **databases** 호스트 그룹에 있습니다.

18.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 멤버 관리자가 있는지 확인

다음 절차에서는 Ansible 플레이북을 사용하여 IdM 호스트 및 호스트 그룹에 멤버 관리자가 있는지 확

인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- 멤버 관리자로 추가하는 호스트 또는 호스트 그룹의 이름과 관리할 호스트 그룹의 이름이 있어야 합니다.

절차

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

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

2. 필요한 호스트 및 호스트 그룹 멤버 관리 정보로 **Ansible** 플레이북 파일을 생성합니다.

```
---
- name: Playbook to handle host group membership management
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure member manager user example_member is present for group_name
      ipahostgroup:
        ipaadmin_password: MySecret123
        name: group_name
        membermanager_user: example_member

    - name: Ensure member manager group project_admins is present for group_name
      ipahostgroup:
        ipaadmin_password: MySecret123
        name: group_name
        membermanager_group: project_admins
```

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/add-member-managers-host-groups.yml
```

검증 단계

ipa group-show 명령을 사용하여 **group_name** 그룹에 **example_member** 및 **project_admins** 가 멤버 관리자로 포함되어 있는지 확인할 수 있습니다.

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

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

2.

testhostgroup 에 대한 정보를 표시합니다.

```
ipaserver]$ ipa hostgroup-show group_name
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: testhostgroup2
Membership managed by groups: project_admins
Membership managed by users: example_member
```

추가 리소스

- **ipa hostgroup-add-member-manager --help** 를 참조하십시오.
- **ipa man** 페이지를 참조하십시오.

18.6. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 호스트가 없는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**의 호스트 그룹에서 호스트가 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- Ansible 플레이북에 참조하려는 호스트는 IdM에 있습니다. 자세한 내용은 **Ansible 플레이북을 사용하여 IdM 호스트 항목이 있는지 여부** 를 참조하십시오.
- Ansible 플레이북 파일에서 참조하는 호스트 그룹은 IdM에 있습니다. 자세한 내용은 **Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 여부** 를 참조하십시오.

절차

1.

인벤토리 파일(예: `inventory.file`)을 생성하고 타겟할 IdM 서버 목록으로 `ipaserver` 를 정의합니다.

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

2.

필요한 호스트 및 호스트 그룹 정보를 사용하여 Ansible 플레이북 파일을 생성합니다. `ipahostgroup` 변수의 `name` 매개 변수를 사용하여 호스트 그룹의 이름을 지정합니다. `ipahostgroup` 변수의 호스트 매개 변수를 사용하여 확인하려는 호스트 그룹이 없는 호스트의 이름을 지정합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-are-hostgroups-are-in-hostgroup.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
    # Ensure host-group databases is absent
    - ipahostgroup:
        ipadmin_password: MySecret123
        name: databases
        host:
          - db.idm.example.com
        action: member
        state: absent
```

이 플레이북은 `databases` 호스트 그룹에서 `db.idm.example.com` 호스트가 없는지 확인합니다. `action: member` 행은 플레이북이 실행되면 `databases` 그룹 자체를 제거하려고 시도하지 않음을 나타냅니다.

3.

플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-absent-in-
hostgroup.yml
```

검증 단계

1.

admin으로 ipaserver 에 로그인합니다.

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

2.

관리자를 위한 Kerberos 티켓을 요청합니다.

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

3.

호스트 그룹 및 호스트에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
Host-group: databases
Member host-groups: mysql-server, oracle-server
```

db.idm.example.com 호스트는 databases 호스트 그룹에 존재하지 않습니다.

18.7. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 중복 호스트 그룹이 없는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)의 외부 호스트 그룹에서 중복된 호스트 그룹이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

•

IdM 관리자 암호를 알고 있습니다.

•

Ansible 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.

- **Ansible** 플레이북 파일에서 참조하는 호스트 그룹은 **IdM**에 있습니다. 자세한 내용은 [Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 여부](#)를 참조하십시오.

절차

1. 인벤토리 파일(예: `inventory.file`)을 생성하고 타겟할 **IdM** 서버 목록으로 **ipaserver**를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. - **ipahostgroup** 변수 중에서 **name** 변수를 사용하여 외부 호스트 그룹의 이름을 지정합니다. **hostgroup** 변수를 사용하여 중첩 호스트 그룹의 이름을 지정합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-are-hostgroups-are-in-hostgroup.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
    # Ensure hosts and hostgroups are absent in existing databases hostgroup
    - ipahostgroup:
        ipaadmin_password: MySecret123
        name: databases
        hostgroup:
          - mysql-server
          - oracle-server
        action: member
        state: absent
```

이 플레이북은 **mysql-server** 및 **oracle-server** 호스트 그룹이 **databases** 호스트 그룹에 없는지 확인합니다. **action: member** 행은 플레이북이 실행되면 **IdM**에서 데이터베이스 그룹 자체가 삭제되도록 시도하지 않음을 나타냅니다.

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-absent-in-
hostgroup.yml
```

검증 단계

1. admin으로 ipaserver 에 로그인합니다.

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

2. 관리자를 위한 Kerberos 티켓을 요청합니다.

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

3. 중첩 호스트 그룹이 없어야 하는 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
Host-group: databases
```

출력은 **mysql-server** 및 **oracle-server** 중첩 호스트 그룹이 외부 데이터베이스 호스트 그룹에 없음을 확인합니다.

18.8. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹이 없는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에서 호스트 그룹이 없는지 확인하는 방법을 설명합니다.

참고

Ansible이 없으면 **ipa hostgroup-del** 명령을 사용하여 **IdM**에서 호스트 그룹 항목이 제거됩니다. **IdM**에서 호스트 그룹을 제거하는 결과는 **IdM**에 호스트 그룹이 없는 상태입니다. **Ansible**의 **idempotence on idempotence** 때문에 **Ansible**을 사용하여 **IdM**에서 호스트 그룹을 제거하려면 호스트 그룹의 상태를 **absent: state: absent** 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

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

절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 타겟할 **IdM** 서버 목록으로 **ipaserver**를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 단순화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-absent.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
  - Ensure host-group databases is absent
    ipahostgroup:
      ipaadmin_password: MySecret123
      name: databases
      state: absent
```

이 플레이북은 **IdM**에서 **databases** 호스트 그룹이 없는지 확인합니다. **state: absent**는 이미 삭제되지 않는 한 **IdM**에서 호스트 그룹을 삭제하라는 요청을 의미합니다.

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hostgroup-is-absent.yml
```

검증 단계

1. **admin**으로 **ipaserver**에 로그인합니다.

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

2. 관리자를 위한 **Kerberos** 티켓을 요청합니다.

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

3. 확인하지 않은 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
ipa: ERROR: databases: host group not found
```

IdM에는 데이터베이스 호스트 그룹이 없습니다.

18.9. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 멤버 관리자가 없는지 확인

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 호스트 및 호스트 그룹에 멤버 관리자가 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- 멤버 관리자로 제거하고 관리하는 호스트 그룹의 이름을 제거하는 사용자 또는 사용자 그룹의 이름이 있어야 합니다.

절차

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

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

2. 필요한 호스트 및 호스트 그룹 멤버 관리 정보로 **Ansible** 플레이북 파일을 생성합니다.

```
---
- name: Playbook to handle host group membership management
  hosts: ipaserver
  become: true
```

```

tasks:
- name: Ensure member manager host and host group members are absent for
  group_name
  ipahostgroup:
    ipaadmin_password: MySecret123
    name: group_name
    membermanager_user: example_member
    membermanager_group: project_admins
    action: member
    state: absent

```

3.

플레이북을 실행합니다.

```

$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-member-managers-host-groups-are-absent.yml

```

검증 단계

ipa group-show 명령을 사용하여 **group_name** 그룹에 **example_member** 또는 **project_admins** 가 멤버 관리자로 포함되어 있지 않은지 확인할 수 있습니다.

1.

관리자 권한으로 **ipaserver** 에 로그인합니다.

```

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

```

2.

testhostgroup 에 대한 정보를 표시합니다.

```

ipaserver]$ ipa hostgroup-show group_name
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: testhostgroup2

```

추가 리소스

- **ipa hostgroup-add-member-manager --help** 를 참조하십시오.
- **ipa man** 페이지를 참조하십시오.

19장. IDM 암호 정책 정의

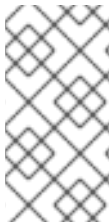
이 장에서는 **IdM(Identity Management)** 암호 정책 및 **Ansible** 플레이북을 사용하여 **IdM**에 새 암호 정책을 추가하는 방법을 설명합니다.

19.1. 암호 정책이란 무엇입니까?

암호 정책은 암호가 충족해야 하는 일련의 규칙입니다. 예를 들어 암호 정책은 최소 암호 길이 및 최대 암호 수명을 정의할 수 있습니다. 이 정책의 영향을 받는 모든 사용자는 충분히 긴 암호를 설정하고 지정된 조건을 충족하기에 충분히 자주 변경해야 합니다. 이렇게 하면 암호 정책을 사용하면 누군가가 사용자의 암호를 검색하고 잘못 사용할 위험을 줄일 수 있습니다.

19.2. IDM의 암호 정책

암호는 **IdM(Identity Management)** 사용자가 **IdM Kerberos** 도메인에 인증하는 가장 일반적인 방법입니다. 암호 정책은 이러한 **IdM** 사용자 암호가 충족해야 하는 요구 사항을 정의합니다.



참고

IdM 암호 정책은 기본 **LDAP** 디렉터리에 설정되어 있지만 **Kerberos KDC(Key Distribution Center)**는 암호 정책을 적용합니다.

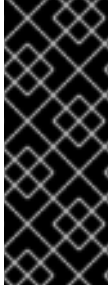
암호 정책 속성은 **IdM**에서 암호 정책을 정의하는 데 사용할 수 있는 속성이 나열됩니다.

표 19.1. 암호 정책 속성

속성	설명	예제
최대 수명	사용자가 암호를 재설정하기 전에 암호가 유효한 최대 시간(일)입니다.	최대 수명 = 90 사용자 암호는 90일 동안만 유효합니다. 그러면 IdM에서 사용자에게 변경하라는 메시지를 표시합니다.
최소 수명	두 암호 변경 작업 간에 전달해야 하는 최소 시간(시간)입니다.	최소 수명 = 1 사용자가 암호를 변경한 후에는 암호를 변경하기 전에 1시간 이상 기다려야 합니다.

속성	설명	예제
기록 크기	저장된 이전 암호 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 * comma (,), 마침표(.), 별표 (*) 와 같은 특수 문자 * 다른 UTF-8 문자 행에서 문자를 세 번 이상 사용하면 문자 클래스가 하나씩 감소합니다. 예를 들어 다음과 같습니다. * Secret1에는 대문자, 소문자, 숫자 3개가 있습니다. * Secret111에는 대문자, 소문자, 숫자 및 1을 반복적으로 사용하기 위한 -1의 문자 클래스가 있습니다.	문자 클래스 = 0 기본 클래스 수는 0입니다. 번호를 구성하려면 --minclasses 옵션과 함께 ipa pwpolicy-mod 명령을 실행합니다. 이 표 아래에 있는 중요한 참고 사항도 참조하십시오.
최소 길이	암호의 최소 문자 수입입니다. 추가 암호 정책 옵션을 설정하는 경우 최소 암호 길이는 Min length 옵션이 설정된 값과 관계없이 6입니다.	최소 길이 = 8 사용자는 8자 미만의 암호를 사용할 수 없습니다.
최대 실패	IdM이 사용자 계정을 잠기 전에 실패한 로그인 시도 횟수입니다.	최대 실패 = 6 사용자가 행에 잘못된 암호 7번을 입력하면 IdM에서 사용자 계정을 잠급니다.
실패 재설정 간격	IdM이 현재 실패한 로그인 시도 횟수를 재설정 한 후 시간(초)입니다.	실패 재설정 간격 = 60 사용자가 최대 실패에 정의된 로그인 시도 횟수가 1분 이상 대기하는 경우 사용자는 사용자 계정 잠금에 위험을 주지 않고 다시 로그인을 시도할 수 있습니다.

속성	설명	예제
Lockout 기간	최대 실패 시 정의된 로그인 시도 횟수 후 사용자 계정이 잠겼는 시간(초)입니다.	Lockout 기간 = 600 연결된 계정이 있는 사용자는 10분 동안 로그인할 수 없습니다.



중요

국제 문자 및 기호에 액세스할 수 없는 다양한 하드웨어 세트가 있는 경우 문자 클래스 요구 사항에 대한 영어 알파벳과 공통 기호를 사용하십시오. 암호의 문자 클래스 정책에 대한 자세한 내용은 [Red Hat Knowledgebase의 암호에서 유효한 문자는 무엇입니까?](#) 를 참조하십시오.

19.3. ANSIBLE 플레이북을 사용하여 IDM에 암호 정책이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 암호 정책이 있는지 확인하는 방법에 대해 설명합니다.

IdM의 기본 **global_policy** 암호 정책에서 암호에 있는 다른 문자 클래스의 수는 **0**으로 설정됩니다. 또한 기록 크기는 **0**으로 설정됩니다.

Ansible 플레이북을 사용하여 **IdM** 그룹에 대해 강력한 암호 정책을 적용하려면 이 절차를 완료합니다.



참고

IdM 그룹에 대한 암호 정책만 정의할 수 있습니다. 개별 사용자의 암호 정책을 정의할 수 없습니다.

사전 요구 사항

- **Ansible** 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다.
- **IdM** 관리자 암호를 알고 있습니다.
- **IdM**에 암호 정책이 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. 확인하려는 암호 정책을 정의하는 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 단순화하려면 `/usr/share/doc/ansible-freeipa/playbooks/pwpolicy/pwpolicy_present.yml` 파일의 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 변수가 무엇을 의미하는지에 대한 자세한 내용은 [Password policy attributes](#) 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 성공적으로 사용하여 IdM에 대한 암호 정책이 IdM에 있는지 확인합니다.

중요

controlPlane 암호 정책의 우선 순위는 **1**로 설정되지만 **global_policy** 암호 정책에는 우선 순위가 설정되지 않습니다. 이러한 이유로 **EgressIP** 정책은 **ActiveDirectory** 그룹의 전역 **_policy**를 자동으로 대체하며 즉시 적용됩니다.

global_policy는 사용자에게 대해 그룹 정책이 설정되지 않은 경우 대체 정책 역할을 하며 그룹 정책보다 우선할 수 없습니다.

추가 리소스

- [/usr/share/doc/ansible-freeipa/ 디렉토리에서 README-pwpolicy.md 파일을 참조하십시오.](#)
- [암호 정책 우선 순위를 참조하십시오.](#)

19.4. 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 허용되는 최대 연속 문자 수를 지정합니다.

--maxsequence

새 암호에서 **monotonic** 문자 시퀀스의 최대 길이를 지정합니다. 이러한 시퀀스의 예로는 **12345** 또는 **fedcb**가 있습니다. 이러한 암호의 대부분은 단순성 검사를 통과하지 않습니다. 유일한 예외는 시퀀스가 암호의 마이너 부분인 경우입니다.

--dictcheck

0이 아닌 경우 가능한 수정 사항이 있는 암호가 사전의 단어와 일치하는지 확인합니다. 현재 **libpwquality**는 **Cracklib** 라이브러리를 사용하여 사전 검사를 수행합니다.

--usercheck

0이 아닌 경우 가능한 수정 가능한 암호에 일부 형식의 사용자 이름이 포함되어 있는지 확인합니다. 3자 미만의 사용자 이름에는 적용되지 않습니다.

추가 암호 정책 옵션을 기존 암호에 적용할 수 없습니다. 추가 옵션을 적용하면 **IdM**에서 **--minlength** 옵션, 최소 문자 수를 **6** 자로 설정합니다.



참고

RHEL 7, RHEL 8 및 RHEL 9 서버와 혼합된 환경에서는 **RHEL 8.4** 이상에서 실행되는 서버에만 추가 암호 정책 설정을 적용할 수 있습니다. 사용자가 **IdM** 클라이언트에 로그인되어 있고 **IdM** 클라이언트가 **RHEL 8.3** 이상에서 실행되는 **IdM** 서버와 통신하는 경우 시스템 관리자가 설정한 새로운 암호 정책 요구 사항이 적용되지 않습니다. 일관된 동작을 위해 모든 서버를 **RHEL 8.4** 이상으로 업그레이드하거나 업데이트합니다.

추가 리소스:

- [IdM 그룹에 추가 암호 정책 적용](#)
- [pwquality\(3\) 매뉴얼 페이지](#)

19.5. IDM 그룹에 추가 암호 정책 옵션 적용

이 섹션에서는 **IdM(Identity Management)**에서 추가 암호 정책 옵션을 적용하는 방법을 설명합니다. 이 예제에서는 새 암호에 사용자의 각 사용자 이름이 포함되지 않고 암호에 두 개 이상의 동일한 문자가 포함되어 있는지 확인하여 **managers** 그룹에 대한 암호 정책을 적용하는 방법을 설명합니다.

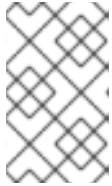
사전 요구 사항

- **IdM** 관리자로 로그인되어 있습니다.
- **managers** 그룹은 **IdM**에 있습니다.
- **managers** 암호 정책은 **IdM**에 있습니다.

절차

1. **managers** 그룹의 사용자가 지정한 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 연속 문자가 포함된 경우 암호를 사용할 수 없습니다. 예를 들어 **eR873mUi111YJQ** 조합은 연속 **3** 개 **s**가 포함되어 있기 때문에 허용되지 않습니다.

검증

1.

test_user 라는 테스트 사용자를 추가합니다.

```
$ ipa user-add test_user
First name: test
Last name: user
-----
Added user "test_user"
-----
```

2.

managers 그룹에 **test** 사용자를 추가합니다.

a.

IdM 웹 UI에서 **Identity** → **Groups** → **User Groups** (사용자 그룹) 를 클릭합니다.

b.

관리자.

c.

추가를 클릭합니다.

d.

사용자 그룹 '**managers**'에 사용자 사용자 추가 페이지에서 **test_user** 를 확인합니다.

e.

> ; 화살표를 클릭하여 사용자를 **Prospective** 열로 이동합니다.

- f.
 - a. **추가**를 클릭합니다.
3. **test** 사용자의 암호를 재설정합니다.
 - a. **Identity** → **Users** 로 이동합니다.
 - b. **test_user** 를 클릭합니다.
 - c. 작업 메뉴에서 암호 재설정 을 클릭합니다.
 - d. 사용자에게 대한 임시 암호를 입력합니다.
4. 명령줄에서 **test_user** 에 대한 Kerberos TGT(ticket-granting ticket)를 받으십시오.

```
$ kinit test_user
```

- a. 임시 암호를 입력합니다.
- b. 시스템에서 암호를 변경해야 함을 알려줍니다. **test_user** 의 사용자 이름이 포함된 암호를 입력합니다.

```
Password expired. You must change it now.
Enter new password:
Enter it again:
```



참고

Kerberos는 세분화된 오류 암호 정책 보고가 없으며, 경우에 따라 암호가 거부된 명확한 이유를 제공하지 않습니다.

- c. 시스템에서 입력한 암호가 거부되었음을 알려줍니다. **3**개 이상의 동일한 문자가 포함된 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.

시스템에서 입력한 암호가 거부되었음을 알려줍니다. 관리자 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 TGT를 확인합니다.

```

$ 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

```

이제 **managers** 암호 정책이 **managers** 그룹의 사용자에게 올바르게 작동합니다.

추가 리소스

•

[IdM의 추가 암호 정책](#)

20장. IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

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

20.1. IDM 클라이언트의 SUDO 액세스

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

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

- 로컬로 **/etc/sudoers** 파일
- IdM 중앙

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

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

추가 리소스

- [sudo 액세스 관리를 참조하십시오.](#)

20.2. CLI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

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

됩니다.

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

사전 요구 사항

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

절차

1. IdM 관리자로 Kerberos 티켓을 검색합니다.

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

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

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

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

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

4.

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

```
[root@idmclient ~]# ipa sudorule-add-allow-command idm_user_reboot --sudocmds
'/usr/sbin/reboot'
Rule name: idm_user_reboot
Enabled: TRUE
Sudo Allow Commands: /usr/sbin/reboot
-----
Number of members added 1
-----
```

5.

IdM `idmclient` 호스트에 `idm_user_reboot` 규칙을 적용합니다.

```
[root@idmclient ~]# ipa sudorule-add-host idm_user_reboot --hosts
idmclient.idm.example.com
Rule name: idm_user_reboot
Enabled: TRUE
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-----
Number of members added 1
-----
```

6.

`idm_user` 계정을 `idm_user_reboot` 규칙에 추가합니다.

```
[root@idmclient ~]# ipa sudorule-add-user idm_user_reboot --users idm_user
Rule name: idm_user_reboot
Enabled: TRUE
Users: idm_user
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-----
Number of members added 1
-----
```



참고

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

검증 단계

1.

`idmclient` 호스트에 `idm_user` 계정으로 로그인합니다.

2.

idm_user 계정이 수행할 수 있는 **sudo** 규칙을 표시합니다.

```
[idm_user@idmclient ~]$ sudo -l
Matching Defaults entries for idm_user on idmclient:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
    LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
    LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
    LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
    XAUTHORITY KRB5CCNAME",
    secure_path="/sbin\:/bin\:/usr/sbin\:/usr/bin

User idm_user may run the following commands on idmclient:
    (root) /usr/sbin/reboot
```

3.

sudo 를 사용하여 시스템을 재부팅합니다. 메시지가 표시되면 **idm_user** 의 암호를 입력합니다.

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

20.3. IDM 웹 UI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

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

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

사전 요구 사항

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

idmclient 호스트에 로컬 **idm_user** 계정이 없습니다. **idm_user** 사용자는 로컬 **/etc/passwd** 파일에 나열되어 있지 않습니다.

절차

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

그림 20.1. 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. **idm_user** 가 **idmclient** 시스템을 재부팅할 수 있도록 새 **sudo** 명령 항목을 사용하여 **sudo** 규칙을 생성합니다.
 - a. 정책 → **Sudo** → **Sudo** 규칙으로 이동합니다.

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

g.

명령을 지정합니다.

i.

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

ii.

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

iii.

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

iv.

Add 를 클릭하여 **idm_sudo_reboot** 페이지로 돌아갑니다.

그림 20.2. IdM sudo 규칙 추가

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
☐ /usr/sbin/reboot	
☐ Sudo Allow Command Groups	
Delete Add	

h.

왼쪽 상단 모서리에서 저장을 클릭합니다.

새 규칙은 기본적으로 활성화되어 있습니다.



참고

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

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

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

sudo 규칙이 올바르게 구성된 경우 시스템이 재부팅됩니다.

20.4. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 CLI에 SUDO 규칙 생성

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

이 예제를 사용하여 **run_third-party-app_report** 라는 명령 줄에 **sudo** 규칙을 생성하여 **idmclient** 호스트의 **thirdpartyapp** 서비스 계정으로 **idm_user** 계정에서 **/opt/ third-party-app/bin/report** 명령을 실행할 수 있습니다.

사전 요구 사항

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

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

절차

1. IdM 관리자로 **Kerberos** 티켓을 검색합니다.

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

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

```
[root@idmclient ~]# ipa sudocmd-add /opt/third-party-app/bin/report
-----
Added Sudo Command "/opt/third-party-app/bin/report"
-----
Sudo Command: /opt/third-party-app/bin/report
```

3. 이름이 **run_third-party-app_report** 인 **sudo** 규칙을 만듭니다.

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

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

```
[root@idmclient ~]# ipa sudorule-add-runasuser run_third-party-app_report --
users=thirdpartyapp
Rule name: run_third-party-app_report
Enabled: TRUE
RunAs External User: thirdpartyapp
-----
Number of members added 1
-----
```

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

5.

idm_user_reboot 규칙에 /opt/third-party-app/bin/report 명령을 추가합니다.

```
[root@idmclient ~]# ipa sudorule-add-allow-command run_third-party-app_report --
sudocmds '/opt/third-party-app/bin/report'
Rule name: run_third-party-app_report
Enabled: TRUE
Sudo Allow Commands: /opt/third-party-app/bin/report
RunAs External User: thirdpartyapp
-----
Number of members added 1
-----
```

6.

IdM idmclient 호스트에 run_third-party-app_report 규칙을 적용합니다.

```
[root@idmclient ~]# ipa sudorule-add-host run_third-party-app_report --hosts
idmclient.idm.example.com
Rule name: run_third-party-app_report
Enabled: TRUE
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /opt/third-party-app/bin/report
RunAs External User: thirdpartyapp
-----
Number of members added 1
-----
```

7.

idm_user 계정을 run_third-party-app_report 규칙에 추가합니다.

```
[root@idmclient ~]# ipa sudorule-add-user run_third-party-app_report --users
idm_user
Rule name: run_third-party-app_report
Enabled: TRUE
Users: idm_user
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /opt/third-party-app/bin/report
RunAs External User: thirdpartyapp
-----
Number of members added 1
```



참고

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

검증 단계

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

```
[idm_user@idmclient ~]$ sudo -l
Matching Defaults entries for idm_user@idm.example.com on idmclient:
    !visiblepw, always_set_home, match_group_by_gid,
    always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
    LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS
    LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
    LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
    LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
    XAUTHORITY KRB5CCNAME",
    secure_path="/sbin:/bin:/usr/sbin:/usr/bin

User idm_user@idm.example.com may run the following commands on idmclient:
    (thirdpartyapp) /opt/third-party-app/bin/report
```

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

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

20.5. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 IDM WEBUI에 SUDO 규칙 생성

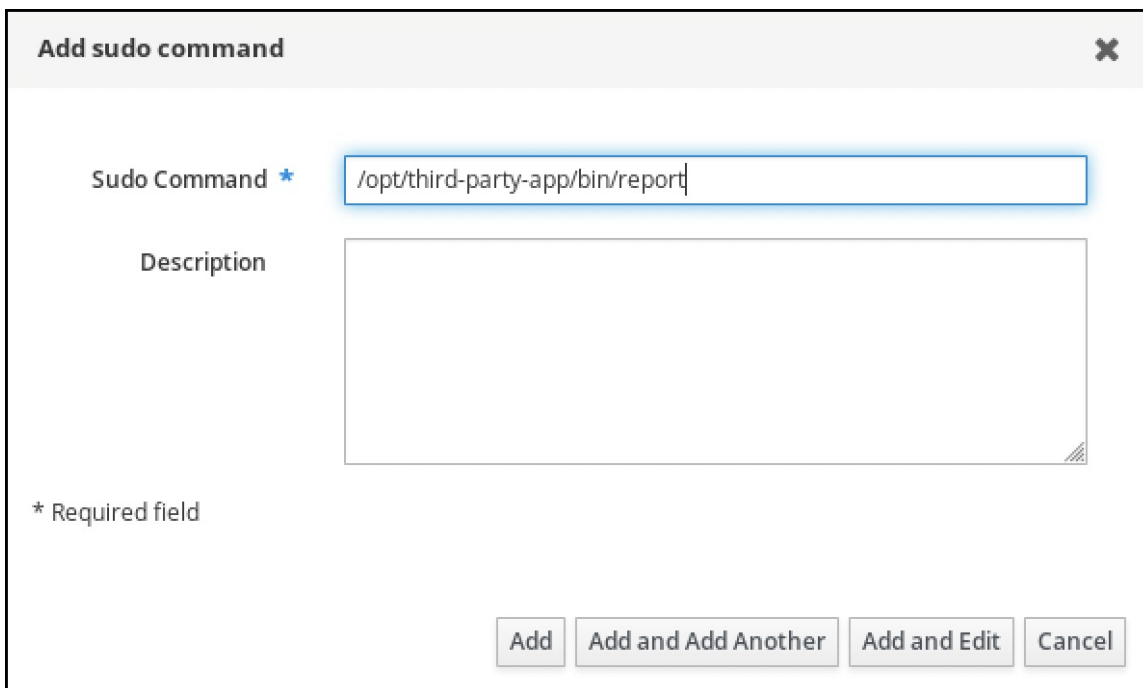
IdM에서는 **RunAs** 별칭을 사용하여 **sudo** 규칙을 구성하여 다른 사용자 또는 그룹으로 **sudo** 명령을 실행할 수 있습니다. 예를 들어 데이터베이스 애플리케이션을 호스팅하는 IdM 클라이언트가 있을 수 있으며 해당 애플리케이션에 해당하는 로컬 서비스 계정으로 명령을 실행해야 합니다.

이 예제를 사용하여 **run_third-party-app_report** 라는 IdM WebUI에 **sudo** 규칙을 생성하여 **idmclient** 호스트의 **thirdpartyapp** 서비스 계정으로 **idm_user** 계정에서 **/opt/third-party-app/bin/report** 명령을 실행할 수 있습니다.

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

절차

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



Add sudo command [X]

Sudo Command *

Description

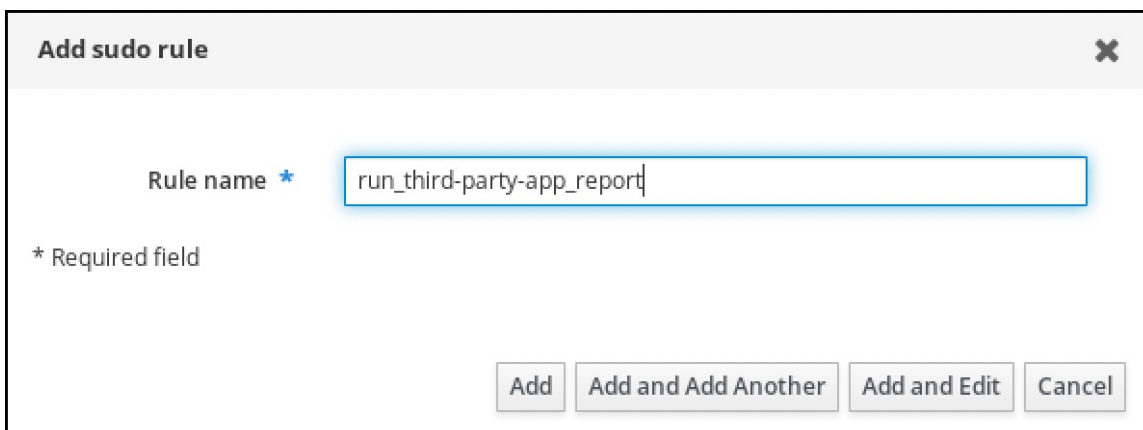
* Required field

Add Add and Add Another Add and Edit Cancel

- d. 추가를 클릭합니다.

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

- a. 정책 → **Sudo** → **Sudo** 규칙으로 이동합니다.
- b. 오른쪽 상단에서 **Add** 를 클릭하여 **sudo** 규칙 추가 대화 상자를 엽니다.
- c. **sudo** 규칙의 이름을 입력합니다. **run_third-party-app_report**.



Add sudo rule [X]

Rule name *

* Required field

Add Add and Add Another Add and Edit Cancel

- d. 추가를 클릭하고 편집 을 클릭합니다.

e.

사용자를 지정합니다.

i.

who 섹션에서 지정된 사용자 및 그룹 라디오 버튼을 선택합니다.

ii.

사용자 카테고리에서 규칙이 하위 섹션에 적용되는 경우 **Add users into sudo rule "run_third-party-app_report"** 대화 상자를 클릭합니다.

iii.

Add users into sudo rule "run_third-party-app_report" 대화 상자에서 **Available** 열에 **idm_user** 확인란을 선택한 후 **Prospective** 열로 이동합니다.

iv.

추가를 클릭합니다.

f.

호스트를 지정합니다.

i.

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

ii.

호스트 카테고리에서 이 규칙이 하위 섹션에 적용되면 **Add hosts**를 클릭하여 **sudo** 규칙 **"run_third-party-app_report"** 대화 상자에 호스트 추가를 엽니다.

iii.

Add hosts into sudo rule "run_third-party-app_report" 대화 상자의 **Available** 열에 **idm.idm.example.com** 확인란을 선택하고 **Prospective** 열로 이동합니다.

Add hosts into sudo rule 'run_third-party-app_report'

Filter available Hosts Filter

Available			Prospective	
☐	Hosts	> <	☐	Hosts
☐	client2.idm.example.com		☒	idmclient.idm.example.com

External

Add Cancel

iv. 추가를 클릭합니다.

g. 명령을 지정합니다.

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

ii. **Sudo Allow Commands (명령 허용)** 하위 섹션에서 **Add allow sudo command into sudo rule "run_third-party-app_report"** 대화 상자를 엽니다.

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

Add allow sudo commands into sudo rule 'run_third-party-app_report'

Filter available Sudo Commands Filter

Available			Prospective	
☐	Sudo Commands	> <	☐	Sudo Commands
☐	help		☒	/opt/third-party-app/bin/report

Add Cancel

iv.

Add (추가)를 클릭하여 **run_third-party-app_report** 페이지로 돌아갑니다.

h.

Runas 사용자를 지정합니다.

i.

As whom 섹션에서 지정된 사용자 및 그룹 라디오 버튼을 선택합니다.

ii.

RunAs Users 하위 섹션에서 **Add** 를 클릭하여 **Add RunAs users into sudo rule "run_third-app_report"** 대화 상자를 엽니다.

iii.

Add RunAs users into sudo rule "run_third-party-app_report" 대화 상자에서 **External** 상자에 **thirdpartyapp** 서비스 계정을 입력하고 **Prospective** 열로 이동합니다.

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 저장을 클릭합니다.

새 규칙은 기본적으로 활성화되어 있습니다.

그림 20.3. sudo 규칙에 대한 세부 정보

Who

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

☐ Users	External	Delete + Add
☐ <code>idm_user</code>		

☐ User Groups [Delete](#) [+ Add](#)

Access this host

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

☐ Hosts	External	Delete + Add
☐ <code>idmclient.idm.example.com</code>		

☐ Host Groups [Delete](#) [+ Add](#)

Run Commands

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

Allow

☐ Sudo Allow Commands	Delete + Add
☐ <code>/opt/third-party-app/bin/report</code>	

☐ 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
☐ <code>thirdpartyapp</code>	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.

report 명령을 **thirdpartyapp** 서비스 계정으로 실행합니다.

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

20.6. IDM 클라이언트에서 SUDO에 대한 GSSAPI 인증 활성화

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

사전 요구 사항

•

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

•

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

절차

1. `/etc/sss/sss.conf` 설정 파일을 엽니다.

2. `[domain/ <domain_name>; ]` 섹션에 다음 항목을 추가합니다.

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

3. `/etc/sss/sss.conf` 파일을 저장하고 닫습니다.

4. SSSD 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.

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

5. `/etc/pam.d/sudo` PAM 구성 파일을 엽니다.

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

```
##PAM-1.0
auth sufficient pam_sss_gss.so
auth    include    system-auth
account include    system-auth
password include    system-auth
session include    system-auth
```

7. `/etc/pam.d/sudo` 파일을 저장하고 종료합니다.

8. `/etc/pam.d/sudo-i` PAM 구성 파일을 엽니다.

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

```
##PAM-1.0
auth sufficient pam_sss_gss.so
auth    include    sudo
```

```
account include sudo
password include sudo
session optional pam_keyinit.so force revoke
session include sudo
```

10.

/etc/pam.d/sudo-i 파일을 저장하고 닫습니다.

검증 단계

1.

idm_user 계정으로 호스트에 로그인합니다.

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

2.

idm_user 계정으로 티켓이 통합되었는지 확인합니다.

```
[idmuser@idmclient ~]$ klist
Ticket cache: KCM:1366201107
Default principal: idm_user@IDM.EXAMPLE.COM

Valid starting    Expires          Service principal
01/08/2021 09:11:48 01/08/2021 19:11:48
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
renew until 01/15/2021 09:11:44
```

3.

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

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

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

4.

암호를 지정하지 않고 sudo 를 사용하여 시스템을 재부팅합니다.

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

추가 리소스

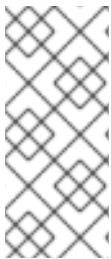
•

IdM 용어 목록의 GSSAPI 항목

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

20.7. IDENTITY MANAGEMENT 클라이언트에서 SUDO에 대한 GSSAPI 인증 활성화 및 적용

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



참고

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

사전 요구 사항

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

절차

1. **/etc/sss/sss.conf** 설정 파일을 엽니다.

2.

[domain/ <domain_name>,] 섹션에 다음 항목을 추가합니다.

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

3.

/etc/sss/sss.conf 파일을 저장하고 닫습니다.

4.

SSSD 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.

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

5.

/etc/pam.d/sudo PAM 구성 파일을 엽니다.

6.

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

```
##PAM-1.0
auth sufficient pam_sss_gss.so
auth include system-auth
account include system-auth
password include system-auth
session include system-auth
```

7.

/etc/pam.d/sudo 파일을 저장하고 종료합니다.

8.

/etc/pam.d/sudo-i PAM 구성 파일을 엽니다.

9.

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

```
##PAM-1.0
auth sufficient pam_sss_gss.so
auth include sudo
account include sudo
password include sudo
session optional pam_keyinit.so force revoke
session include sudo
```

10.

/etc/pam.d/sudo-i 파일을 저장하고 닫습니다.

검증 단계

1.

idm_user 계정으로 호스트에 로그인하고 스마트 카드로 인증합니다.

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

2.

스마트 카드 사용자로 티켓팅 티켓이 있는지 확인합니다.

```
[idm_user@idmclient ~]$ klist
Ticket cache: KEYRING:persistent:1358900015:krb_cache_TObtNMd
Default principal: idm_user@IDM.EXAMPLE.COM

Valid starting    Expires          Service principal
02/15/2021 16:29:48 02/16/2021 02:29:48
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
renew until 02/22/2021 16:29:44
```

3.

idm_user 계정이 수행할 수 있는 **sudo** 규칙을 표시합니다.

```
[idm_user@idmclient ~]$ sudo -l
Matching Defaults entries for idmuser on idmclient:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
    LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
    LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
    LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
    XAUTHORITY KRB5CCNAME",
    secure_path="/sbin\:/bin\:/usr/sbin\:/usr/bin

User idm_user may run the following commands on idmclient:
    (root) /usr/sbin/reboot
```

4.

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

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

추가 리소스

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

20.8. PAM 서비스의 GSSAPI 인증을 제어하는 SSSD 옵션

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

pam_gssapi_services

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

pam_gssapi_indicators_map

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

유효한 인증 지표는 다음과 같습니다.

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

pam_gssapi_check_upn

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

예제

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

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

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

idm.example.com 도메인의 경우

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

- 일치하는 사용자 이름과 **Kerberos** 주체 적용.

ad.example.com 도메인의 경우

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

```
[domain/idm.example.com]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:pkinit, sudo-i:otp
pam_gssapi_check_upn = true
...

[domain/ad.example.com]
pam_gssapi_services = sudo
pam_gssapi_check_upn = false
...
```

추가 리소스

- [Kerberos 인증 표시](#)

20.9. SUDO용 GSSAPI 인증 문제 해결

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

사전 요구 사항

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

절차

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

바른 영역을 확인할 수 없을 수 있습니다.

Server not found in Kerberos database

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

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

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

- 다음 오류가 표시되면 **Kerberos** 자격 증명이 없습니다.

No Kerberos credentials available

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

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

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

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

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

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

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

pam_gssapi_check_upn = false
```

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

○

`/etc/pam.d/ sudo` 및 `/etc/pam.d/sudo-i`와 같이 PAM 파일의 모든 `pam_sss_gss.so` 항목 끝에 `debug` 옵션을 추가합니다.

```
[root@idm-client ~]# cat /etc/pam.d/sudo
#%PAM-1.0
auth    sufficient pam_sss_gss.so  debug
auth    include    system-auth
account include    system-auth
password include    system-auth
session include    system-auth
```

```
[root@idm-client ~]# cat /etc/pam.d/sudo-i
#%PAM-1.0
auth    sufficient pam_sss_gss.so  debug
auth    include    sudo
account include    sudo
password include    sudo
session optional   pam_keyinit.so force revoke
session include    sudo
```

○

`pam_sss_gss.so` 모듈로 인증을 시도하고 콘솔 출력을 검토합니다. 이 예에서는 사용자에게 Kerberos 자격 증명이 없습니다.

```
[idm-user@idm-client ~]$ sudo ls -l /etc/sss/sss.conf
pam_sss_gss: Initializing GSSAPI authentication with SSSD
pam_sss_gss: Switching euid from 0 to 1366201107
pam_sss_gss: Trying to establish security context
pam_sss_gss: SSSD User name: idm-user@idm.example.com
pam_sss_gss: User domain: idm.example.com
pam_sss_gss: User principal:
pam_sss_gss: Target name: host@idm.example.com
pam_sss_gss: Using ccache: KCM:
pam_sss_gss: Acquiring credentials, principal name will be derived
pam_sss_gss: Unable to read credentials from [KCM:] [maj:0xd0000,
min:0x96c73ac3]
pam_sss_gss: GSSAPI: Unspecified GSS failure. Minor code may provide more
information
pam_sss_gss: GSSAPI: No credentials cache found
pam_sss_gss: Switching euid from 1366200907 to 0
pam_sss_gss: System error [5]: Input/output error
```

20.10. ANSIBLE 플레이북을 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 확인

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

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

사전 요구 사항

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

절차

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


```
[ipaservers]
server.idm.example.com
```
2. 하나 이상의 **sudo** 명령을 추가합니다.
 - a. **sudo** 명령의 IdM 데이터베이스에 **/usr/sbin/reboot** 명령이 있는지 확인하는 **ensure-reboot-sudocmd-is-present.yml** Ansible 플레이북을 생성합니다. 이 단계를 단순화하기

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

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

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

b.

플레이북을 실행합니다.

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

3.

명령을 참조하는 **sudo** 규칙을 생성합니다.

a.

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

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

  tasks:
    # Ensure a sudorule is present granting idm_user the permission to run
    /usr/sbin/reboot on idmclient
    - ipasudorule:
        ipadmin_password: MySecret123
        name: idm_user_reboot
        description: A test sudo rule.
        allow_sudocmd: /usr/sbin/reboot
        host: idmclient.idm.example.com
        user: idm_user
        state: present
```

b.

플레이북을 실행합니다.

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

검증 단계

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

1.

idmclient 에 **idm_user** 로 로그인합니다.

2.

sudo 를 사용하여 시스템을 재부팅합니다. 메시지가 표시되면 **idm_user** 의 암호를 입력합니다.

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

sudo 가 올바르게 구성된 경우 시스템이 재부팅됩니다.

추가 리소스

-

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

21장. ANSIBLE 플레이북을 사용하여 IDM에 호스트 기반 액세스 제어 규칙이 있는지 확인

이 장에서는 **IdM(Identity Management)** 호스트 기반 액세스 정책과 **Ansible** 을 사용하여 정의하는 방법을 설명합니다.

Ansible은 시스템 구성, 소프트웨어 배포 및 롤링 업데이트를 수행하는 데 사용되는 자동화 틀입니다. **IdM(Identity Management)** 지원이 포함됩니다.

21.1. IDM의 호스트 기반 액세스 제어 규칙

호스트 기반 액세스 제어(**HBAC**) 규칙은 서비스 그룹의 서비스 또는 서비스를 사용하여 호스트 또는 호스트 그룹에 액세스할 수 있는 사용자 또는 사용자 그룹을 정의합니다. 시스템 관리자는 **HBAC** 규칙을 사용하여 다음 목표를 달성할 수 있습니다.

- 도메인의 지정된 시스템에 대한 액세스 권한을 특정 사용자 그룹의 멤버로 제한합니다.
- 특정 서비스만 도메인의 시스템에 액세스하는 데 사용할 수 있도록 허용합니다.

기본적으로 **IdM**은 전체 **IdM** 도메인의 모든 관련 서비스를 통해 모든 사용자에게 대해 모든 호스트에 대한 범용 액세스를 의미하는 **allow_all** 이라는 기본 **HBAC** 규칙으로 구성됩니다.

기본 **allow_all** 규칙을 고유한 **HBAC** 규칙 세트로 대체하여 다른 호스트에 대한 액세스를 미세 조정할 수 있습니다. 중앙 집중식으로 간소화된 액세스 제어 관리의 경우 **HBAC** 규칙을 개별 사용자, 호스트 또는 서비스 대신 사용자 그룹, 호스트 그룹 또는 서비스 그룹에 적용할 수 있습니다.

21.2. ANSIBLE 플레이북을 사용하여 IDM에 HBAC 규칙이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 호스트 기반 액세스 제어 (**HBAC**) 규칙이 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- **ansible-freeipa** 패키지는 **Ansible** 컨트롤러에 설치됩니다.

- **IdM 관리자 암호를 알고 있습니다.**
- **HBAC 규칙에 사용할 사용자 및 사용자 그룹은 IdM에 있습니다.** 자세한 내용은 [Ansible 플레이북을 사용하여 사용자 계정 관리](#) 및 [Ansible 플레이북을 사용하여 IdM 그룹 및 그룹 멤버가 있는지 확인합니다.](#)
- **HBAC 규칙을 적용할 호스트 및 호스트 그룹이 IdM에 있습니다.** 자세한 내용은 [Ansible 플레이북을 사용하여 호스트 관리](#) 및 [Ansible 플레이북을 사용하여 호스트 그룹을 관리하십시오.](#)

절차

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

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

2. 확인하려는 **HBAC 정책**을 정의하는 **Ansible 플레이북** 파일을 생성합니다. 이 단계를 단순화하기 위해 `/usr/share/doc/ansible-freeipa/playbooks/hbacrule/ensure-hbacrule-allhosts-present.yml` 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
    # Ensure idm_user can access client.idm.example.com via the sshd service
    - ipahbacrule:
        ipaadmin_password: MySecret123
        name: login
        user: idm_user
        host: client.idm.example.com
        hbacsvc:
          - sshd
        state: present
```

3. 플레이북을 실행합니다.

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

검증 단계

1. 관리자 권한으로 **IdM 웹 UI**에 로그인합니다.
2. 정책 → 호스트 기반 액세스 제어 → **HBAC** 테스트 로 이동합니다.
3. **who** 탭에서 **idm_user**를 선택합니다.
4. 액세스 탭에서 **client.idm.example.com** 을 선택합니다.
5. **Via** 서비스 탭에서 **sshd** 를 선택합니다.
6. 규칙 탭에서 로그인 을 선택합니다.
7. **Run test (테스트 실행)** 탭에서 **Run test(테스트 실행)** 버튼을 클릭합니다. **ACCESS GRANTED**가 표시되면 **HBAC** 규칙이 성공적으로 구현됩니다.

추가 리소스

- **/usr/share/doc/ansible-freeipa** 디렉터리의 **README-hbacsvgroup.md**, **README-hbacrule.md** 파일을 참조하십시오.
- **/usr/share/doc/ansible-freeipa/playbooks** 디렉터리의 하위 디렉터리에 있는 플레이북을 참조하십시오.

22장. IDM의 자격 증명 모음

이 장에서는 **IdM(Identity Management)**의 자격 증명 모음에 대해 설명합니다. 다음 주제를 소개합니다.

- 자격 증명 모음의 개념입니다.
- 자격 증명 모음과 관련된 다양한 역할입니다.
- 보안 및 액세스 제어 수준에 따라 **IdM**에서 사용할 수 있는 다양한 유형의 자격 증명 모음.
- 소유권을 기반으로 **IdM**에서 사용할 수 있는 다양한 유형의 자격 증명 모음.
- 자격 증명 모음 컨테이너의 개념.
- **IdM**에서 자격 증명 모음을 관리하는 기본 명령입니다.
- **IdM**에서 자격 증명 모음을 사용하기 위한 사전 요구 사항인 **KRA**(키 복구 기관) 설치.

22.1. 자격 증명 모음 및 이점

자격 증명 모음은 중요한 모든 데이터를 한 곳에서 안전하게 저장하려는 **IdM(Identity Management)** 사용자에게 유용한 기능입니다. 이 섹션에서는 다양한 유형의 자격 증명 모음과 용도와 요구 사항에 따라 선택해야 하는 자격 증명 모음에 대해 설명합니다.

자격 증명 모음은 시크릿을 저장, 검색, 공유 및 복구하는 데 필요한 보안 위치(**IdM**)입니다. 보안은 보안에 민감한 데이터(일반적으로 인증 자격 증명)이며 제한된 사용자 또는 엔티티 그룹만 액세스할 수 있습니다. 예를 들어 시크릿은 다음과 같습니다.

- 암호
- pins

•

개인 SSH 키

자격 증명 모음은 암호 관리자와 비교할 수 없습니다. 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 설정하도록 선택할 수도 있습니다. 표준 자격 증명 모음은 사용자가 자격 증명 모음에 저장된 시크릿에 액세스하기 위해 암호를 입력할 필요가 없습니다.



참고

IdM의 자격 증명 모음은 외부 비 **IdM** 관련 서비스에 인증할 수 있는 인증 자격 증명을 저장하는 것입니다.

IdM 자격 증명 모음의 기타 중요한 특성은 다음과 같습니다.

•

자격 증명 모음은 자격 증명 모음 소유자와 자격 증명 모음 소유자가 자격 증명 모음 멤버로 선택하는 **IdM** 사용자만 액세스할 수 있습니다. 또한 **IdM** 관리자는 자격 증명 모음에 액세스할 수 있습니다.

•

사용자에게 자격 증명 모음을 생성할 수 있는 충분한 권한이 없는 경우 **IdM** 관리자는 자격 증명 모음을 생성하고 사용자를 소유자로 설정할 수 있습니다.

•

사용자 및 서비스는 **IdM** 도메인에 등록된 모든 시스템에서 자격 증명 모음에 저장된 시크릿에 액세스할 수 있습니다.

•

하나의 자격 증명 모음에는 하나의 시크릿(예: 하나의 파일)만 포함할 수 있습니다. 그러나 파일 자체는 암호, 키값 또는 인증서와 같은 여러 시크릿을 포함할 수 있습니다.



참고

Vault는 **IdM** 웹 UI가 아닌 **IdM CLI**(명령줄)에서만 사용할 수 있습니다.

22.2. VAULT 소유자, 멤버 및 관리자

IdM(Identity Management)은 다음과 같은 **vault** 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.

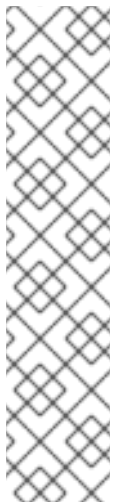
각 자격 증명 모음에는 소유자가 한 명 이상 있어야 합니다. 자격 증명 모음에는 여러 개의 소유자가 있을 수도 있습니다.

Vault 멤버

자격 증명 모음 멤버는 다른 사용자 또는 서비스에서 생성한 자격 증명 모음에 액세스할 수 있는 사용자 또는 서비스입니다.

Vault 관리자

Vault 관리자는 모든 자격 증명 모음에 대한 무제한 액세스 권한이 있으며 모든 자격 증명 모음 작업을 수행할 수 있습니다.



참고

대칭 및 **symmetric** 자격 증명 모음은 암호 또는 키로 보호되며 특수 액세스 제어 규칙을 적용합니다([Vault 유형](#) 참조). 관리자는 다음 규칙을 충족해야 합니다.

- 대칭 및 **symmetric** 자격 증명 모음의 시크릿에 액세스합니다.
- 자격 증명 모음 암호 또는 키를 변경하거나 재설정합니다.

자격 증명 모음 관리자는 **Vault** 관리자 권한이 있는 모든 사용자입니다. **IdM**의 역할 기반 액세스 제어(**RBAC**) 컨텍스트에서 권한은 역할에 적용할 수 있는 권한 그룹입니다.

Vault 사용자

vault 사용자는 자격 증명 모음이 있는 컨테이너가 있는 사용자를 나타냅니다. **ipa vault-show** 와 같은 특정 명령의 출력에 **Vault** 사용자 정보가 표시됩니다.

```
$ ipa vault-show my_vault
```

```
Vault name: my_vault
Type: standard
Owner users: user
Vault user: user
```

자격 증명 모음 컨테이너 및 사용자 자격 증명 모음에 대한 자세한 내용은 [Vault containers](#) 를 참조하십시오.

추가 리소스

- [자격 증명 모음에 대한 자세한 내용은 표준, 대칭 및 symmetric 자격 증명 모음을 참조하십시오.](#)

22.3. 표준, 대칭 및 SYMMETRIC 자격 증명 모음

IdM은 보안 및 액세스 제어 수준에 따라 IdM에 다음 유형으로 자격 증명 모음을 부여합니다.

표준 자격 증명 모음

Vault 소유자와 자격 증명 모음 멤버는 암호 또는 키를 사용하지 않고도 시크릿을 보관하고 검색할 수 있습니다.

대칭 자격 증명 모음

자격 증명 모음의 보안은 대칭 키로 보호됩니다. **Vault** 소유자와 멤버는 시크릿을 보관 및 검색할 수 있지만 자격 증명 모음 암호를 제공해야 합니다.

GCE 자격 증명 모음

자격 증명 모음의 보안은 **symmetric** 키로 보호됩니다. **secrets in the vault are protected with an symmetric key.** 사용자는 공개 키를 사용하여 시크릿을 보관하고 개인 키를 사용하여 검색합니다. **Vault** 멤버는 시크릿만 보관할 수 있지만 자격 증명 모음 소유자는 시크릿을 보관하고 검색할 수 있습니다.

22.4. 사용자, 서비스 및 공유 자격 증명 모음

IdM은 소유권을 기반으로 자격 증명 모음을 여러 유형으로 분류합니다. [아래 표](#)에 는 각 유형, 소유자 및 사용에 대한 정보가 포함되어 있습니다.

표 22.1. 소유권을 기반으로 하는 IdM 자격 증명 모음

유형	설명	소유자	참고
사용자 자격 증명 모음	사용자의 개인 자격 증명 모음	단일 사용자	IdM 관리자가 허용하는 경우 사용자는 하나 이상의 사용자 자격 증명 모음을 보유할 수 있습니다.
서비스 자격 증명 모음	서비스의 개인 자격 증명 모음	단일 서비스	IdM 관리자가 허용하는 경우 모든 서비스에서 하나 이상의 사용자 자격 증명 모음을 보유할 수 있습니다.
공유 자격 증명 모음	여러 사용자 및 서비스에 공유하는 자격 증명 모음	자격 증명 모음을 생성한 자격 증명 모음 관리자	IdM 관리자가 허용하는 경우 사용자 및 서비스에서 하나 이상의 사용자 자격 증명 모음을 소유할 수 있습니다. 자격 증명 모음을 생성한 자격 증명 모음 이외의 관리자는 자격 증명 모음에 대한 전체 액세스 권한도 있습니다.

22.5. VAULT 컨테이너

자격 증명 모음 컨테이너는 자격 증명 모음의 컬렉션입니다. [아래 표](#)에 는 IdM(Identity Management)에서 제공하는 기본 자격 증명 모음 컨테이너가 나열되어 있습니다.

표 22.2. IdM의 기본 자격 증명 모음 컨테이너

유형	설명	목적
사용자 컨테이너	사용자의 개인 컨테이너	특정 사용자의 사용자 자격 증명 모음 저장
서비스 컨테이너	서비스의 개인 컨테이너	특정 서비스에 대한 서비스 자격 증명 모음 저장
공유 컨테이너	여러 사용자 및 서비스를 위한 컨테이너	여러 사용자 또는 서비스에서 공유할 수 있는 자격 증명 모음 저장

IdM은 사용자 또는 서비스의 첫 번째 개인 자격 증명 모음이 생성될 때 각 사용자 또는 서비스에 대해 사용자 및 서비스 컨테이너를 자동으로 생성합니다. 사용자 또는 서비스가 삭제되면 IdM은 컨테이너 및 해당 콘텐츠를 제거합니다.

22.6. 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 용도의 설명이 포함된 `ipa vault-*` 명령 목록이 포함되어 있습니다.



참고

ipa vault-* 명령을 실행하기 전에 **IdM** 도메인의 서버 중 하나 이상에 키 복구 기관 (**KRA**) 인증서 시스템 구성 요소를 설치합니다. 자세한 내용은 **IdM에 키 복구 기관 설치**를 참조하십시오.

표 22.3. 설명이 포함된 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 옵션 및 해당 명령에 사용 가능한 자세한 도움말이 표시됩니다.
ipa vault-show user_vault --user idm_user	자격 증명 모음 멤버로 자격 증명 모음 구성원으로 자격 증명 모음에 액세스할 때 자격 증명 모음 소유자를 지정해야 합니다. vault 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 모음인지 지정해야 합니다. 그렇지 않으면 IdM에서 자격 증명 모음을 찾을 수 없다고 알립니다. <pre>[admin@server ~]\$ ipa vault-show shared_vault ipa: ERROR: shared_vault: vault not found</pre>

22.7. IDM에 키 복구 기관 설치

이 섹션에서는 특정 **IdM** 서버에서 **KRA(Key recovery Authority)** 구성 요소를 설치하여 **IdM(Identity Management)**에서 자격 증명 모음을 활성화하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 서버에 **root** 로 로그인되어 있습니다.
- **IdM** 인증 기관이 **IdM** 서버에 설치되어 있습니다.

- **Directory Manager** 자격 증명이 있습니다.

절차

- **KRA**를 설치합니다.

ipa-kra-install



중요

숨겨진 복제본에 **IdM** 클러스터의 첫 번째 **KRA**를 설치할 수 있습니다. 그러나 추가 **KRAs**를 설치하려면 금지된 복제본에 **KRA** 복제본을 설치하기 전에 숨겨진 복제본을 임시로 활성화해야 합니다. 그런 다음 원래 숨겨진 복제본을 다시 숨길 수 있습니다.



참고

자격 증명 모음 서비스를 사용하고 탄력적으로 사용하려면 두 개의 **IdM** 서버에 **KRA**를 설치합니다. 여러 개의 **KRA** 서버를 유지 관리하면 데이터 손실이 발생하지 않습니다.

추가 리소스

- [숨겨진 복제본 데모 또는 승격을 참조하십시오.](#)
- [숨겨진 복제본 모드를 참조하십시오.](#)

23장. ANSIBLE을 사용하여 IDM 사용자 자격 증명 모음 관리: 보안 저장 및 검색

이 장에서는 **Ansible** 자격 증명 모음 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 시크릿을 저장합니다.**
- **자격 증명 모음에서 시크릿을 검색합니다.**

사용자는 두 개의 다른 **IdM** 클라이언트에서 저장하고 검색할 수 있습니다.

사전 요구 사항

- 키 복구 기관(**KRA**) 인증서 시스템 구성 요소가 **IdM** 도메인의 서버 중 하나 이상에 설치되어 있습니다. 자세한 내용은 **IdM에 키 복구 기관 설치**를 참조하십시오.

23.1. ANSIBLE을 사용하여 IDM에 표준 사용자 자격 증명 모음 확인

이 섹션에서는 **IdM(Identity Management)** 사용자가 **Ansible** 플레이북을 사용하여 하나 이상의 개인 자격 증명 모음을 사용하여 중요한 정보를 안전하게 저장하는 방법을 보여줍니다. 아래 절차에 사용된 예에서 **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

- 절차의 단계를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **idm_user** 의 암호를 알고 있습니다.

절차

1. `/usr/share/doc/ansible-freeipa/playbooks/vault` 디렉토리로 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 설정할 `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 들어 다음과 같습니다.

```
$ cp ensure-standard-vault-is-present.yml ensure-standard-vault-is-present-copy.yml
```

5. 편집할 수 있도록 `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 표준 으로 설정합니다.

이는 현재 예에 대해 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
```

23.2. ANSIBLE을 사용하여 IDM의 표준 사용자 자격 증명 모음에 보안 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 민감한 정보가 있는 파일을 아카이브합니다.

사전 요구 사항

- 절차의 단계를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **idm_user** 의 암호를 알고 있습니다.
- **idm_user** 는 소유자이거나 **my_vault** 의 멤버 사용자입니다.
- **password.txt.txt**에 액세스할 수 있습니다. **my_vault** 에 아카이브할 시크릿입니다.

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** 디렉토리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 인벤토리 파일을 열고 구성하려는 **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. **data-archive-in-symmetric-vault.yml** **Ansible** 플레이북 파일의 사본을 만들지만 **"symmetric"**을 **"standard"**로 바꿉니다. 예를 들어 다음과 같습니다.

```
$ cp data-archive-in-symmetric-vault.yml data-archive-in-standard-vault-copy.yml
```

4. 편집할 수 있도록 **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.
- **in** 변수를 중요한 정보가 있는 파일의 전체 경로로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

이는 현재 예에 대해 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
```

23.3. ANSIBLE을 사용하여 IDM의 표준 사용자 자격 증명 모음에서 시크릿 검색

이 섹션에서는 **IdM(Identity Management)** 사용자가 **Ansible** 플레이북을 사용하여 사용자 개인 자격 증명에서 시크릿을 검색하는 방법을 보여줍니다. 아래 절차에 사용된 예에서 **idm_user** 사용자는 **my_vault** 라는 표준 유형의 자격 증명 모음에서 **host01** 이라는 **IdM** 클라이언트에 대한 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. 프로세스의 단계를 실행하는 호스트입니다.
- **idm_user** 의 암호를 알고 있습니다.
- **idm_user** 는 **my_vault** 의 소유자입니다.

- **idm_user** 는 **my_vault** 에 시크릿을 저장했습니다.
- **Ansible**은 시크릿을 검색하려는 **IdM** 호스트의 디렉터리에 쓸 수 있습니다.
- **idm_user** 는 시크릿을 검색할 **IdM** 호스트의 디렉터리에서 읽을 수 있습니다.

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** 디렉토리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 인벤토리 파일을 열고, 시크릿을 검색하려는 **IdM** 클라이언트라는 명확하게 정의된 섹션에서 언급합니다. 예를 들어, **Ansible**에 시크릿을 **host01.idm.example.com** 에 검색하도록 지시하려면 다음을 입력합니다.

```
[ipahost]
host01.idm.example.com
```

3. **retrive-data-symmetric-vault.yml** **Ansible** 플레이북 파일의 사본을 만듭니다. **"symmetric"**을 **"standard"**로 바꿉니다. 예를 들어 다음과 같습니다.

```
$ cp retrive-data-symmetric-vault.yml retrieve-data-standard-vault.yml-copy.yml
```

4. 편집하기 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.
- 시크릿을 내보낼 파일의 전체 경로로 **out** 변수를 설정합니다.
- **state** 변수를 **retrieved** 로 설정합니다.

이는 현재 예에 대해 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. **SSH to host01 as user01:**

```
$ ssh user01@host01.idm.example.com
```

2. **Ansible** 플레이북 파일에서 **out** 변수에서 지정한 파일을 확인합니다.

```
$ vim /tmp/password_exported.txt
```

이제 내보낸 시크릿을 볼 수 있습니다.

- **Ansible**을 사용하여 **IdM** 자격 증명 모음 및 사용자 보안 관리 및 플레이북 변수에 대한 자세한 내용은 **/usr/share/doc/ansible-freeipa/** 디렉터리에서 사용 가능한 **README-vault.md** 마크 파일과 **/usr/share/doc/ansible-freeipa/playbooks/vault/** 디렉터리에 있는 샘플 플레이북을 참조하십시오.

24장. ANSIBLE을 사용하여 IDM 서비스 자격 증명 모음 관리: 보안 저장 및 검색

이 섹션에서는 관리자가 **ansible-freeipa vault** 모듈을 사용하여 서비스 보안을 중앙 집중식 위치에 안전하게 저장하는 방법을 보여줍니다.

예제에서 사용되는 **자격 증명 모음**은 **symmetric**이며, 이는 이를 사용하기 위해 관리자가 다음 단계를 수행해야 함을 의미합니다.

1. **openssl** 유틸리티와 같이 **을** 사용하여 개인 키를 생성합니다.
2. 개인 키를 기반으로 공개 키를 생성합니다.

관리자가 자격 증명 모음에 아카이브할 때 서비스 시크릿은 공개 키로 암호화됩니다. 이후 도메인의 특정 시스템에서 호스팅되는 서비스 인스턴스는 개인 키를 사용하여 시크릿을 검색합니다. 서비스 및 관리자만 보안에 액세스할 수 있습니다.

보안이 손상된 경우 관리자는 서비스 자격 증명 모음에서 이를 교체한 다음 손상되지 않은 개별 서비스 인스턴스에 다시 배포할 수 있습니다.

사전 요구 사항

- 키 복구 기관(**KRA**) 인증서 시스템 구성 요소가 **IdM** 도메인의 서버 중 하나 이상에 설치되어 있습니다. 자세한 내용은 **IdM에 키 복구 기관 설치**를 참조하십시오.

이 섹션에는 다음 절차가 포함됩니다.

- **Ansible을 사용하여 IdM에 symmetric 서비스 자격 증명 모음이 있는지 확인**
- **Ansible을 사용하여 symmetric 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

절차의 경우:

-

관리자 는 서비스 암호를 관리하는 관리자입니다.

-

private-key-to-externally-signed-certificate.pem은 서비스 보안이 포함된 파일입니다(이 경우 외부 서명된 인증서에 대한 개인 키). 이 개인 키를 자격 증명 모음에서 시크릿을 검색하는 데 사용되는 개인 키와 혼동하지 마십시오.

-

secret_vault 는 서비스 보안을 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.

24.1. ANSIBLE을 사용하여 IDM에 SYMMETRIC 서비스 자격 증명 모음이 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 하나 이상의 개인 자격 증명 모음을 사용하여 중요한 정보를 안전하게 저장하는 방법을 보여줍니다. 아래 절차에 사용되는 예에서는 관리자가 **secret_vault** 라는 **symmetric** 자격 증명 모음을 생성합니다. 이렇게 하면 자격 증명 모음에서 시크릿을 검색하려면 자격 증명 모음 멤버가 개인 키를 사용하여 인증해야 합니다. 자격 증명 모음 멤버는 모든 **IdM** 클라이언트에서 파일을 검색할 수 있습니다.

사전 요구 사항

-

Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 단계를 실행하는 호스트입니다.

- **IdM 관리자 암호를 알고 있습니다.**

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** 디렉토리로 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: **inventory.file**이 없는 경우 인벤토리 파일을 생성합니다(예: **inventory.file**).

```
$ touch inventory.file
```

4. 인벤토리 파일을 열고 **[ipaserver]** 섹션에서 설정할 **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 들어 다음과 같습니다.

```
$ cp ensure-asymmetric-vault-is-present.yml ensure-asymmetric-service-vault-is-present-copy.yml
```

6. 편집할 수 있도록 **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 파일의 나머지 부분을 수정합니다.
 - **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
 - **name** 변수를 사용하여 자격 증명 모음의 이름을 정의합니다(예: **secret_vault**).
 - **vault_type** 변수를 **symmetric** 으로 설정합니다.
 - 서비스 변수를 자격 증명 모음을 소유한 서비스 주체(예: **HTTP/webserver1.idm.example.com**)로 설정합니다.
 - **public_key_file** 을 공개 키의 위치로 설정합니다.

현재 예제에 대해 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

```

24.2. ANSIBLE을 사용하여 SYMMETRIC 자격 증명 모음에 멤버 서비스 추가

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 서비스 자격 증명 모음에 멤버 서비스를 추가하는 방법을 보여줍니다. 모두 자격 증명 모음에 저장된 시크릿을 검색할 수 있습니다. 아래 절차에 사용된 예제에서 IdM 관리자는 HTTP/webserver2.idm.example.com 및 HTTP/webserver3.idm.example.com 서비스 주체를 HTTP/webserver1.idm.example.com에서 소유한 secret_vault vault.com 에 추가합니다.

사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- 서비스 보안을 저장하기 위해 **symmetric** 자격 증명 모음을 생성 했습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/vault 디렉토리로 이동합니다.

```

$ cd /usr/share/doc/ansible-freeipa/playbooks/vault

```

2. 선택 사항: inventory.file이 없는 경우 인벤토리 파일을 생성합니다(예: inventory.file).

```

$ touch inventory.file

```

3.

인벤토리 파일을 열고 **[ipaserver]** 섹션에서 설정할 **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 파일을 복사합니다. 예를 들어 다음과 같습니다.

```
$ cp data-archive-in-asymmetric-vault.yml add-services-to-an-asymmetric-vault.yml
```

5.

편집하기 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 수 있는 서비스를 정의합니다.
- **action** 변수를 **member** 로 설정합니다.

이는 현재 예에 대해 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
```

24.3. ANSIBLE을 사용하여 SYMMETRIC 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 저장하는 방법을 보여줍니다. 아래 절차에 사용된 예에서는 관리자가 **secret_vault** 라는 **symmetric** 자격 증명 모음에 시크릿이 포함된 **PEM** 파일을 저장합니다. 이렇게 하면 자격 증명 모음에서 시크릿을 검색하려면 개인 키를 사용하여 서비스가 인증해야 합니다. 자격 증명 모음 멤버는 모든 **IdM** 클라이언트에서 파일을 검색할 수 있습니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.
- 서비스를 보안을 저장하기 위해 **symmetric** 자격 증명 모음을 생성 했습니다.
- 보안은 **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` 디렉토리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2.

선택 사항: **inventory.file**이 없는 경우 인벤토리 파일을 생성합니다(예: **inventory.file**).

```
$ touch inventory.file
```

3.

인벤토리 파일을 열고 **[ipaserver]** 섹션에서 설정할 **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 파일을 복사합니다. 예를 들어 다음과 같습니다.

```
$ cp data-archive-in-asymmetric-vault.yml data-archive-in-asymmetric-vault-copy.yml
```

5.

편집하기 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.
- 변수의 **in** 변수를 **"{{ lookup('file', 'private-key-to-an-externally-certificate.pem') | b64encode }}"** 로 설정합니다. 이렇게 하면 **Ansible**에서 **IdM** 서버가 아닌 **Ansible** 컨트롤러의 작업 디렉터리에서 개인 키를 사용하여 파일을 검색합니다.
- **action** 변수를 **member** 로 설정합니다.

이는 현재 예에 대해 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
    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
```

24.4. ANSIBLE을 사용하여 IDM 서비스의 서비스 시크릿 검색

이 섹션에서는 **IdM(Identity Management)** 사용자가 서비스를 대신하여 서비스 자격 증명 모음에서 시크릿을 검색하도록 **Ansible** 플레이북을 사용하는 방법을 보여줍니다. 아래 절차에 사용된 예제에서 플레이북을 실행하면 **secret_vault** 라는 **symmetric** 자격 증명 모음에서 시크릿을 사용하여 **PEM** 파일을 검색하고, **Ansible** 인벤토리 파일에 **ipaservers** 로 나열된 모든 호스트의 지정된 위치에 저장합니다.

서비스는 키탭을 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. 프로세스의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

- 서비스 보안을 저장하기 위해 **symmetric** 자격 증명 모음을 생성 했습니다.
- **자격 증명 모음에 시크릿을 보관** 했습니다.
- **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** 디렉토리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 선택 사항: **inventory.file**이 없는 경우 인벤토리 파일을 생성합니다(예: **inventory.file**).

```
$ touch inventory.file
```

3. 인벤토리 파일을 열고 다음 호스트를 정의합니다.

- **[ipaserver]** 섹션에 **IdM** 서버를 정의합니다.
- **[webservers]** 섹션에서 시크릿을 검색할 호스트를 정의합니다. 예를 들어, **Ansible**에 시크릿을 **webserver1.idm.example.com**, **webserver2.idm.example.com** 및 **webserver3.idm.example.com** 에 검색하도록 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 들어 다음과 같습니다.

```
$ cp retrieve-data-asymmetric-vault.yml retrieve-data-asymmetric-vault-copy.yml
```

5. 편집하기 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.
- **private-key-to-an-externally-certificate.pem** 시크릿(예: 현재 작업 디렉터리)을 검색하려는 **IdM** 서버의 위치로 **아웃** 변수를 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

이는 현재 예에 대해 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** 컨트롤러로 데이터 파일을 검색하는 플레이북에 섹션을 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.

검색된 **private-key-to-an-externally-signed-certificate.pem** 파일을 **Ansible** 컨트롤러의 **Ansible** 컨트롤러의 **Playbook**에 전송하는 섹션을 플레이북에 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
```

24.5. ANSIBLE을 사용하여 손상된 경우 IDENTITY 서비스 자격 증명 모음 시크릿 변경

이 섹션에서는 **IdM(Identity Management)** 관리자가 서비스 인스턴스가 손상되었을 때 **Ansible** 플레이북을 재사용하여 서비스 자격 증명 모음에 저장된 시크릿을 변경하는 방법을 보여줍니다. 다음 예제의 시나리오에서는 검색된 시크릿이 손상된 **webserver3.idm.example.com** 에서 해당 시크릿을 저장하는 **symmetric** 자격 증명 모음의 키가 아닌 것으로 가정합니다. 이 예제에서 관리자는 **symmetric** 자격 증명 모음에 시크릿을 저장하고 **IdM** 호스트에 대한 **symmetric** 자격 증명 모음에서 시크릿을 검색할 때 사용되는 **Ansible** 플레이북을 재사용합니다. 이 절차를 시작할 때 **IdM** 관리자는 새 시크릿을 사용하여 새 **PEM**

파일을 **anyone** 자격 증명 모음에 저장하고, 손상된 웹 서버 **webserver3.idm.example.com**에 대한 새 시크릿을 검색하지 않도록 인벤토리 파일을 조정한 다음 두 절차를 다시 실행합니다.

사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- 서비스 보안을 저장하기 위해 **symmetric** 자격 증명 모음을 생성했습니다.
- IdM 호스트에서 실행되는 웹 서비스에 대한 새 **httpd** 키를 생성하여 손상된 이전 키를 교체했습니다.
- 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` 디렉토리로 이동합니다.

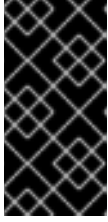
```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 인벤토리 파일을 열고 다음 호스트가 올바르게 정의되어 있는지 확인합니다.

- **[ipaserver]** 섹션의 IdM 서버입니다.
- **[webservers]** 섹션에서 시크릿을 검색할 호스트입니다. 예를 들어, Ansible에 시크릿을 **webserver1.idm.example.com** 및 **webserver2.idm.example.com**에 검색하도록 지시하려면 다음을 입력합니다.

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

```
[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
```



중요

목록에 손상된 웹 서버가 포함되어 있지 않은지 확인합니다. 현재 예제 **webserver3.idm.example.com**.

3. 편집하기 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 **in** 변수를 `"{{ lookup('file', 'new-private-key-to-externally-signed-certificate.pem) | b64encode }}"` 로 설정합니다. 이렇게 하면 Ansible에서 IdM 서버가 아닌 Ansible 컨트롤러의 작업 디렉터리에서 개인 키를 사용하여 파일을 검색합니다.
- **action** 변수를 **member** 로 설정합니다.

이는 현재 예에 대해 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. 편집하기 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.
- 기존 작업 디렉터리와 같이 **new-private-key-to-an-externally-certificate.pem** 시크릿을 검색하려는 **IdM** 서버의 위치로 아웃 변수를 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

이는 현재 예에 대해 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** 컨트롤러로 데이터 파일을 검색하는 플레이북에 섹션을 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.

검색된 **new-private-key-an-externally-signed-certificate.pem** 파일을 **Ansible** 컨트롤러의 **Ansible** 컨트롤러의 **Playbook**에 전송하는 섹션을 플레이북에 추가합니다.

```

---
- name: Send data file to web servers
  become: yes
  gather_facts: no
  hosts: web servers
  tasks:
- name: Send data to web servers
  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
```

24.6. 추가 리소스

- `/usr/share/doc/ansible-freeipa/` 디렉토리에서 **README-vault.md** 마크다운 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/vault/` 디렉토리에서 샘플 플레이북을 참조하십시오.

25장. ANSIBLE을 사용하여 IDM에서 서비스 존재 및 서비스 존재 확인

IdM(Identity Management) 관리자는 **Ansible service** 모듈을 사용하면 **IdM**에 네이티브하지 않은 특정 서비스가 **IdM**에 존재하지 않거나 누락되어 있는지 확인할 수 있습니다. 예를 들어, **service** 모듈을 사용하여 다음을 수행할 수 있습니다.

- 수동으로 설치된 서비스가 **IdM** 클라이언트에 있는지 확인하고 없는 경우 해당 서비스를 자동으로 설치합니다. 자세한 내용은 다음을 참조하십시오.
 - **IdM** 클라이언트의 **IdM**에 **HTTP** 서비스가 있는지 확인합니다.
 - **IdM** 이외의 클라이언트의 **IdM**에 **HTTP** 서비스가 있는지 확인합니다.
 - **DNS** 없이 **IdM** 클라이언트에 **HTTP** 서비스가 있는지 확인합니다.
- **IdM**에 등록된 서비스에 인증서가 연결되어 있는지 확인하고 없는 경우 해당 인증서를 자동으로 설치합니다. 자세한 내용은 다음을 참조하십시오.
- **IdM** 서비스 항목에 외부 서명 인증서가 있는지 확인합니다.
- **IdM** 사용자와 호스트가 서비스 키를 검색하고 생성하도록 허용합니다. 자세한 내용은 다음을 참조하십시오.
 - **IdM** 사용자, 그룹, 호스트 또는 호스트 그룹이 서비스의 키를 생성하도록 허용합니다.
 - **IdM** 사용자, 그룹, 호스트 또는 호스트 그룹을 허용하여 서비스의 키를 검색할 수 있습니다.
- **IdM** 사용자와 호스트가 서비스에 **Kerberos** 별칭을 추가할 수 있도록 허용합니다. 자세한 내용은 다음을 참조하십시오.
 - 서비스에 대한 **Kerberos** 보안 주체 별칭이 있는지 확인합니다.

- 서비스가 **IdM** 클라이언트에 없는지 확인하고 있는 경우 해당 서비스를 자동으로 제거합니다. 자세한 내용은 다음을 참조하십시오.
- **IdM 클라이언트의 IdM에 HTTP 서비스가 없는지 확인합니다.**

25.1. 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 호스트하는 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 설정할 **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 파일을 복사합니다. 예를 들어 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. 편집할 **/usr/share/doc/ansible-freeipa/playbooks/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 수정합니다.

•

ipaadmin_password 변수에서 정의한 **IdM** 관리자 암호를 변경합니다.

•

ipaservice 작업의 **name** 변수에 정의된 대로 **HTTP** 서비스가 실행 중인 **IdM** 클라이언트의 이름을 변경합니다.

6.

파일을 저장하고 종료합니다.

7.

플레이북 파일과 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.

Identity → **Services** 로 이동합니다.

HTTP/client.idm.example.com@IDM.EXAMPLE.COM 가 서비스 목록에 표시되면 **Ansible** 플레이북이 **IdM**에 성공적으로 추가되었습니다.

추가 리소스

•

HTTP 서버와 브라우저 클라이언트 간의 통신을 보호하려면 [Apache HTTP 서버에 TLS 암호](#)

화 추가 를 참조하십시오.

- HTTP 서비스의 인증서를 요청하려면 **certmonger**를 사용하여 서비스에 대한 **IdM** 인증서 가져오기 에 설명된 절차를 참조하십시오.

25.2. ANSIBLE 플레이북을 사용하여 비IDM 클라이언트에서 IDM에 HTTP 서비스가 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM** 클라이언트가 아닌 호스트에서 **IdM**에 **HTTP** 서버가 있는지 확인하는 방법을 설명합니다. **HTTP** 서버를 **IdM**에 추가하여 **IdM**에 호스트를 추가합니다.

사전 요구 사항

- 호스트에 **HTTP** 서비스를 설치했습니다.
- **HTTP**를 설정한 호스트는 **IdM** 클라이언트가 아닙니다. 그렇지 않으면 **Ansible** 플레이북을 사용하여 **IdM**에 **HTTP** 서비스가 있는지 확인하는 단계를 수행합니다.
- **IdM** 관리자 암호가 있습니다.
- **IPv6**를 사용하는 경우 **DNS A** 레코드 - 또는 **AAAA** 레코드 - 호스트를 사용할 수 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2. **inventory.file** 을 열고 **[ipaserver]** 섹션에서 설정할 **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 파일을 복사합니다. 예를 들어 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 편집하여 편집합니다. `ipaservice` 작업에서 `ipaadmin_password` 및 `name` 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 수정합니다.

- `ipaadmin_password` 변수를 **IdM** 관리자 암호로 설정합니다.
- `name` 변수를 **HTTP** 서비스가 실행 중인 호스트의 이름으로 설정합니다.

6.

파일을 저장하고 종료합니다.

7.

플레이북 파일과 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.

Identity → Services 로 이동합니다.

이제 서비스 목록에 `HTTP/client.idm.example.com@IDM.EXAMPLE.COM` 가 표시됩니다.

추가 리소스

•

통신을 보호하려면 [Apache HTTP Server](#)에 [TLS 암호화 추가](#)를 참조하십시오.

25.3. ANSIBLE 플레이북을 사용하여 DNS 없이 IDM 클라이언트에 HTTP 서비스가 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **DNS** 항목이 없는 **IdM** 클라이언트에서 실행 중인 **HTTP** 서버가 있는지 확인하는 방법을 설명합니다. **IPv6**를 사용하는 경우 **IdM** 호스트에 **DNS A** 항목이 없거나 **DNS AAAA** 항목이 없음을 의미합니다.

사전 요구 사항

•

HTTP 서비스를 호스트하는 시스템이 **IdM**에 등록되어 있습니다.

•

호스트의 **DNS A** 또는 **DNS AAAA** 레코드가 존재하지 않을 수 있습니다. 그렇지 않으면 호스트의 **DNS** 레코드가 있는 경우 [Ansible 플레이북을 사용하여 IdM에 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 설정할 **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 파일을 복사합니다. 예를 들어 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. `ipaservice` 작업에서 `ipaadmin_password` 및 `name` 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 수정합니다.

- `ipaadmin_password` 변수를 IdM 관리자 암호로 설정합니다.
- `name` 변수를 HTTP 서비스가 실행 중인 호스트의 이름으로 설정합니다.

6.

파일을 저장하고 종료합니다.

7.

플레이북 파일과 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. **Identity → Services 로 이동합니다.**

이제 서비스 목록에 **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** 가 표시됩니다.

추가 리소스

- 통신을 보호하려면 **Apache HTTP Server에 TLS 암호화 추가**를 참조하십시오.

25.4. ANSIBLE 플레이북을 사용하여 IDM 서비스 항목에 외부 서명 인증서가 있는지 확인

이 섹션에서는 **ansible-freeipa service** 모듈을 사용하여 외부 **CA**(인증 기관)에서 발급한 인증서가 **HTTP** 서비스의 **IdM** 항목에 연결되었는지 확인하는 방법을 설명합니다. **IdM CA**가 아닌 외부 **CA**가 서명한 **HTTP** 서비스의 인증서가 있으면 **IdM CA**에서 자체 서명된 인증서를 사용하는 경우 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 설정할 **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 사본을 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 보안 메일) 형식인 경우 **CLI**(명령줄 인터페이스)를 통해 더 쉽게 처리할 수 있도록 인증서가 **Distinguished Encoding Rules(DER)** 형식으로 변환합니다.

```
$ openssl x509 -outform der -in cert1.pem -out cert1.der
```

5.

base64 명령을 사용하여 **DER** 파일을 표준 출력으로 디코딩합니다. 래핑을 비활성화하려면 **-w0** 옵션을 사용합니다.

```
$ 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 확인합니다.

```
---
- 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 수정합니다.

•

인증서 변수를 사용하여 정의한 인증서 를 **CLI**에서 복사한 인증서로 교체합니다. 표시된 대로 **"|"** pipe 문자와 함께 **certificate:** 변수를 사용하는 경우 한 줄에 입력하도록 하는 대신 인증서 **THIS WAY**를 입력할 수 있습니다. 이렇게 하면 인증서 읽기가 더 쉬워집니다.

•

ipaadmin_password 변수에서 정의한 **IdM** 관리자 암호를 변경합니다.

•

name 변수로 정의된 **HTTP** 서비스가 실행 중인 **IdM** 클라이언트의 이름을 변경합니다.

•

다른 관련 변수를 변경합니다.

9.

파일을 저장하고 종료합니다.

10.

플레이북 파일과 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 UI에 로그인합니다.

2.

Identity → **Services** 로 이동합니다.

3.

새로 추가된 인증서(예: **HTTP/client.idm.example.com**)가 있는 서비스 이름을 클릭합니다.

오른쪽의 **Service Certificate** 섹션에서 새로 추가된 인증서를 볼 수 있습니다.

25.5. ANSIBLE 플레이북을 사용하여 **IDM** 사용자, 그룹, 호스트 또는 호스트 그룹을 통해 서비스의 키탭을 생성할 수 있습니다.

keytab은 **Kerberos** 보안 주체와 암호화된 키 쌍을 포함하는 파일입니다. 키탭 파일은 일반적으로 사용자가 조작하거나 일반 텍스트 파일에 저장된 암호에 액세스하지 않고도 **Kerberos**를 사용하여 스크립트가 자동으로 인증할 수 있도록 하는 데 사용됩니다. 그런 다음 스크립트는 인수된 자격 증명을 사용하여 원격 시스템에 저장된 파일에 액세스할 수 있습니다.

IdM(Identity Management) 관리자는 다른 사용자가 **IdM**에서 실행되는 서비스에 대한 키탭을 검색하거나 생성하도록 허용할 수 있습니다. 특정 사용자 및 사용자 그룹이 키탭을 생성할 수 있도록 허용하면 **IdM** 관리자 암호를 공유하지 않고 서비스 관리를 사용자에게 위임할 수 있습니다. 이 위임은 보다 세분화된 시스템 관리를 제공합니다.

이 섹션에서는 특정 **IdM** 사용자, 사용자 그룹, 호스트 및 호스트 그룹을 허용하여 **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 생성하도록 허용하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **HTTP** 서비스를 **IdM**에 등록했습니다.
- **HTTP** 서비스를 호스트하는 시스템은 **IdM** 클라이언트입니다.
- **IdM**에 키탭을 생성할 수 있도록 허용하는 **IdM** 사용자 및 사용자 그룹입니다.
- **IdM**에 키탭을 생성할 수 있는 **IdM** 호스트 및 호스트 그룹입니다.

절차

1. 인벤토리 파일을 생성합니다(예: **inventory.file**):

—

```
$ touch inventory.file
```

2.

inventory.file 을 열고 **[ipaserver]** 섹션에서 설정할 **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 파일을 복사합니다. 예를 들어 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.

편집할 **/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** 변수에서 지정한 **IdM** 관리자 암호입니다.
- **HTTP** 서비스가 실행 중인 **IdM** 클라이언트의 이름입니다. 현재 예에서는 **HTTP/client.idm.example.com**입니다.
- **allow_create_keytab_user**: 섹션에 나열된 **IdM** 사용자의 이름입니다. 현재 예에서는 **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** 변수로 지정된 작업의 이름입니다.

현재 예제를 위해 조정된 후에는 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 파일과 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 키탭을 생성할 수 있는 권한이 있는 **IdM** 사용자로 **IdM** 서버에 **SSH**를 수행합니다.

```
$ ssh user01@server.idm.example.com
Password:
```

2. **ipa-getkeytab** 명령을 사용하여 **HTTP** 서비스의 새 키탭을 생성합니다.

```
$ ipa-getkeytab -s server.idm.example.com -p HTTP/client.idm.example.com -k
/etc/httpd/conf/krb5.keytab
```

s 옵션은 키탭을 생성하기 위해 **KDC(Key Distribution Center)** 서버를 지정합니다.

p 옵션은 생성하려는 **keytab**의 보안 주체를 지정합니다.

k 옵션은 새 키를 추가할 **keytab** 파일을 지정합니다. 파일이 없는 경우 생성됩니다.

명령에서 오류가 발생하지 않으면 **user01** 로 **HTTP/client.idm.example.com** 의 키탭을 성공적으로 생성했습니다.

25.6. ANSIBLE 플레이북을 사용하여 IDM 사용자, 그룹, 호스트 또는 호스트 그룹이 서비스의 키탭 검색 가능

keytab은 **Kerberos** 보안 주체와 암호화된 키 쌍을 포함하는 파일입니다. 키탭 파일은 일반적으로 사용자가 조작하거나 일반 텍스트 파일에 저장된 암호에 액세스하지 않고도 **Kerberos**를 사용하여 스크립트가 자동으로 인증할 수 있도록 하는 데 사용됩니다. 그런 다음 스크립트는 인수된 자격 증명을 사용하여 원격 시스템에 저장된 파일에 액세스할 수 있습니다.

IdM 관리자는 다른 사용자가 **IdM**에서 실행되는 서비스에 대한 키탭을 검색하거나 생성하도록 허용할 수 있습니다.

이 섹션에서는 특정 **IdM** 사용자, 사용자 그룹, 호스트 및 호스트 그룹을 허용하여 **IdM** 클라이언트에서 실행되는 **HTTP** 서비스의 키탭을 검색하는 방법을 설명합니다. 특히 **user01** **IdM** 사용자가 **client.idm.example.com** 에서 실행되는 **HTTP** 서비스의 키탭을 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.
- **IdM**에 키탭을 검색할 수 있는 **IdM** 사용자 및 사용자 그룹입니다.

- **IdM에 키탭을 검색할 수 있는 IdM 호스트 및 호스트 그룹입니다.**

절차

1. 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2. **inventory.file** 을 열고 **[ipaserver]** 섹션에서 설정할 **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 파일을 복사합니다. 예를 들어 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. 복사된 파일 **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present-copy.yml** 을 엽니다.

5. 파일을 수정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.

- **ipaservice** 작업의 **name** 변수를 **HTTP** 서비스 주체로 설정합니다. 현재 예에서는 **HTTP/client.idm.example.com**입니다.

- **allow_retrieve_keytab_group:** 섹션에 **IdM** 사용자의 이름을 지정합니다. 현재 예에서는 **user01** 입니다.

- **allow_retrieve_keytab_group**: 섹션에 **IdM** 사용자 그룹의 이름을 지정합니다.
- **allow_retrieve_keytab_group**: 섹션에 **IdM** 호스트의 이름을 지정합니다.
- **allow_retrieve_keytab_group**: 섹션에 **IdM** 호스트 그룹의 이름을 지정합니다.
- **tasks** 섹션의 **name** 변수를 사용하여 작업 이름을 지정합니다.

현재 예제를 위해 조정된 후에는 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 파일과 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 키탭을 검색할 수 있는 권한이 있는 **IdM** 사용자로 **IdM** 서버에 **SSH**를 수행합니다.

```
$ 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 키탭을 검색하려는 **KDC(Key Distribution Center)** 서버를 지정합니다.

p 옵션은 검색할 **keytab**의 보안 주체를 지정합니다.

-k 옵션은 검색된 키를 추가할 키탭 파일을 지정합니다. 파일이 없는 경우 생성됩니다.

명령에서 오류가 발생하지 않으면 **user01** 로 **HTTP/client.idm.example.com** 의 키탭을 성공적으로 검색했습니다.

25.7. ANSIBLE 플레이북을 사용하여 서비스의 KERBEROS 주체 별칭이 있는지 확인

일부 시나리오에서는 **IdM** 관리자가 **Kerberos** 주체 별칭을 사용하여 **IdM** 사용자, 호스트 또는 서비스에서 **Kerberos** 애플리케이션에 인증할 수 있도록 하는 것이 좋습니다. 이러한 시나리오에는 다음이 포함됩니다.

- 사용자 이름이 변경되었지만 사용자는 이전 사용자 이름과 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 보안 별칭을 만드는 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 설정할 **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 파일을 복사합니다. 예를 들어 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. 편집할 **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present-copy.yml** Ansible 플레이북 파일을 엽니다.

5. 다음을 변경하여 파일을 조정합니다.

- **ipaadmin_password** 변수에서 지정한 **IdM** 관리자 암호입니다.

-

name 변수로 지정된 서비스의 이름입니다. 이는 서비스의 정식 주체 이름입니다. 현재 예에서는 **HTTP/client.idm.example.com** 입니다.

- 보안 주체 변수에서 지정한 **Kerberos** 사용자 별칭입니다. 이는 **name** 변수로 정의된 서비스에 추가할 별칭입니다. 현재 예에서는 **host/mycompany.idm.example.com** 입니다.
- **tasks** 섹션의 **name** 변수로 지정된 작업의 이름입니다.

현재 예제를 위해 조정된 후에는 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.example.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 파일과 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 없으며 0 failed 작업이 발생하면 **HTTP/client.idm.example.com** 서비스에 대한 **host/mycompany.idm.example.com** Kerberos 주체를 성공적으로 생성했습니다.

추가 리소스

- [사용자, 호스트 및 서비스의 Kerberos 보안 주체 별칭 관리를 참조하십시오.](#)

25.8. ANSIBLE 플레이북을 사용하여 IDM에 HTTP 서비스가 없는지 확인

이 섹션에서는 **IdM**에서 서비스 등록 해제 방법에 대해 설명합니다. 특히, **Ansible** 플레이북을 사용하여 **IdM**에서 **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 설정할 **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 파일을 복사합니다. 예를 들어 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. 편집할 **/usr/share/doc/ansible-freeipa/playbooks/service-is-absent-copy.yml** **Ansible** 플레이북 파일을 엽니다.

5. 다음을 변경하여 파일을 조정합니다.

- **ipaadmin_password** 변수에서 정의한 **IdM** 관리자 암호입니다.
- **ipaservice** 작업의 **name** 변수에 정의된 **HTTP** 서비스의 **Kerberos** 사용자입니다.

현재 예제를 위해 조정된 후에는 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 파일과 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.

Identity → Services 로 이동합니다.

서비스 목록에 HTTP/client.idm.example.com@IDM.EXAMPLE.COM 서비스가 표시되지 않으면 IdM에 없는 것으로 확인되었습니다.

25.9. 추가 리소스

-

/usr/share/doc/ansible-freeipa/ 디렉터리의 README-service.md 마크다운 파일을 참조하십시오.

-

/usr/share/doc/ansible-freeipa/playbooks/config 디렉터리에서 샘플 플레이북을 참조하십시오.

26장. ANSIBLE 플레이북을 사용하여 IDM의 글로벌 DNS 구성 관리

Red Hat Ansible Engine `dnsconfig` 모듈을 사용하면 `IdM`(Identity Management) DNS에 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 IP 주소 및 통신에 사용되는 포트입니다.
- 글로벌 전달 정책: `only`, `first` 또는 `none`입니다. 이러한 유형의 DNS 전달 정책에 대한 자세한 내용은 [IdM의 DNS 전달 정책](#)을 참조하십시오.
- 정방향 조회 및 역방향 조회 영역의 동기화입니다.

사전 요구 사항

- DNS 서비스가 `IdM` 서버에 설치되어 있습니다. 통합 DNS를 사용하여 `IdM` 서버를 설치하는 방법에 대한 자세한 내용은 다음 링크 중 하나를 참조하십시오.
 - [IdM 서버 설치: 통합 DNS와 함께 통합 CA를 루트 CA로 설치](#)
 - [IdM 서버 설치: 루트 CA로 외부 CA를 사용하는 통합 DNS 사용](#)
 - [IdM 서버 설치: CA없이 통합 DNS 사용](#)

이 장에는 다음 섹션이 포함되어 있습니다.

- [NetworkManager에서 /etc/resolv.conf의 글로벌 전달자가 제거되지 않도록 하는 방법](#)

- **Ansible을 사용하여 IdM에 DNS 글로벌 전달자가 있는지 확인**
- **Ansible을 사용하여 IdM에 DNS 글로벌 전달자가 없는지 확인**
- **ipadnsconfig ansible-freeipa 모듈의 action: member 옵션**
- **IdM의 DNS 전달 정책소개**
- **Ansible 플레이북을 사용하여 IdM DNS 글로벌 구성에 첫 번째 정책이 설정되어 있는지 확인합니다.**
- **Ansible 플레이북을 사용하여 IdM DNS에서 글로벌 전달자가 비활성화되었는지 확인**
- **Ansible 플레이북을 사용하여 IdM DNS에서 정방향 및 역방향 조회 영역의 동기화가 비활성화되었는지 확인합니다.**

26.1. NETWORKMANAGER에서 /ETC/RESOLV.CONF의 글로벌 전달자가 제거되지 않도록 하는 방법

통합 DNS를 사용하여 IdM(Identity Management)을 설치하면 127.0.0.1 localhost 주소를 가리키도록 /etc/resolv.conf 파일이 설정됩니다.

```
# Generated by NetworkManager
search idm.example.com
nameserver 127.0.0.1
```

DHCP(Dynamic Host Configuration Protocol)를 사용하는 네트워크와 같은 특정 환경에서는 NetworkManager 서비스가 /etc/resolv.conf 파일에 대한 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 제어하기 위해 /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 파일은 항상 `/etc/NetworkManager/conf.d/` 디렉터리에 마지막 파일의 설정을 사용하여 `/etc/resolv.conf` 파일을 생성합니다. 이 경우 `zzz-ipa.conf` 파일입니다.



중요

`/etc/resolv.conf` 파일을 수동으로 수정하지 마십시오.

26.2. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에서 **DNS** 글로벌 전달자가 있는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 포트 **53**에 있는 **IP(Internet Protocol) v4** 주소가 **7.7.9.9**인 **DNS** 서버로 **DNS** 글로벌 전달자가 있는지, 포트 **53**에서 **2001:db8::1:0**의 **IP v6** 주소가 있는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 **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 들어 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. 편집할 수 있도록 **ensure-presence-of-a-global-forwarder.yml** 파일을 엽니다.
5. 다음 변수를 설정하여 파일을 조정합니다.
 - a. **IdM DNS**에 글로벌 전달자가 있는지 확인하기 위해 플레이북의 **name** 변수를 **Playbook**로 변경합니다.
 - b. **tasks** 섹션에서 작업의 이름을 변경하여 **DNS** 글로벌 전달자가 포트 **53**의 **7.7.9.9** 및 **2001:db8::1:0**으로 있는지 확인합니다.
 - c. **ipadnsconfig** 부분의 **forwarders** 섹션에서 다음을 수행합니다.
 - i. 첫 번째 **ip_address** 값을 **global forwarder: 7.7.9.9**의 IPv4 주소로 변경합니다.
 - ii. 두 번째 **ip_address** 값을 **global forwarder: 2001:db8::1:0**의 IPv6 주소로 변경합니다.
 - iii. 포트 값이 **53**으로 설정되어 있는지 확인합니다.
 - d. **state**를 **present**로 변경합니다.

이는 현재 예에 대해 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/` 디렉토리의 **README-dnsconfig.md** 파일을 참조하십시오.

26.3. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 없는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에서 **DNS** 글로벌 전달자가 없는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 인터넷 프로토콜 (**IP**) **v4** 주소가 **8.8.6.6** 이고 포트 **53** 에서 **2001:4860:4860::8800** 의 **IP v6** 주소가 있는 **DNS** 글로벌 전달자가 없는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 **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 들어 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4.

편집할 수 있도록 **ensure-absence-of-a-global-forwarder.yml** 파일을 엽니다.

5.

다음 변수를 설정하여 파일을 조정합니다.

a.

IdM DNS에 글로벌 전달자가 없도록 플레이북의 **name** 변수를 **Playbook**로 변경합니다.

b.

tasks 섹션에서 작업의 이름을 변경하여 **DNS** 글로벌 전달자가 **8.8.6.6** 및 **2001:4860:4860::8800**이 없는 것을 확인하도록 합니다.

c.

ipadnsconfig 부분의 **forwarders** 섹션에서 다음을 수행합니다.

i.

첫 번째 **ip_address** 값을 글로벌 **forwarder**의 **IPv4** 주소로 변경합니다. **8.8.6.6**.

ii.

두 번째 **ip_address** 값을 **global forwarder**의 **IPv6** 주소로 변경합니다. **2001:4860:4860::8800**.

iii.

포트 값이 **53** 으로 설정되어 있는지 확인합니다.

d.

action 변수를 **member** 로 설정합니다.

e.

상태가 **absent** 로 설정되어 있는지 확인합니다.

이는 현재 예에 대해 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
        action: member
        state: absent
```



중요

action: member 를 사용하지 않고 플레이북에서 **state: absent** 옵션만 사용하면 플레이북이 실패합니다.

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/ 디렉토리의 README-dnsconfig.md 파일](#)
- [ipadnsconfig ansible-freeipa 모듈의 action: member 옵션](#)

26.4. IPADNSCONFIG ANSIBLE-FREEIPA 모듈의 ACTION: MEMBER 옵션

ansible-freeipa ipadnsconfig 모듈을 사용하여 IdM(Identity Management)에서 글로벌 전달자를 제외하려면 **state: absent** 옵션 외에 **action: member** 옵션을 사용해야 합니다. **action: member** 를 사용하지 않고 플레이북에서 **state: absent** 만 사용하면 플레이북이 실패합니다. 결과적으로 모든 글로벌 전달자를 제거하려면 플레이북에서 모두 개별적으로 지정해야 합니다. 반면 **state:** 현재 옵션에는 **action: member** 가 필요하지 않습니다.

다음 표에서 **action: member** 옵션의 올바른 사용을 보여주는 **DNS** 글로벌 전달자를 추가 및 제거하는 구성 예제를 제공합니다. 표는 각 줄에 표시됩니다.

- 플레이북을 실행하기 전에 구성된 글로벌 전달자
- 플레이북의 발췌 내용
- 플레이북을 실행한 후 구성된 글로벌 전달자

표 26.1. 글로벌 전달자의 **ipadnsconfig** 관리

이전 버전과 의 전달자	플레이북 발췌 내용	이후의 forwarders
8.8.6.6	<pre>[...] tasks: - name: Ensure the presence of DNS global forwarder 8.8.6.7 ipadnsconfig: forwarders: - ip_address: 8.8.6.7 state: present</pre>	8.8.6.7
8.8.6.6	<pre>[...] tasks: - name: Ensure the presence of DNS global forwarder 8.8.6.7 ipadnsconfig: forwarders: - ip_address: 8.8.6.7 action: member state: present</pre>	8.8.6.6, 8.8.6.7

이전 버전과 의 전달자	플레이북 발췌 내용	이후의 forwarders
8.8.6.6, 8.8.6.7	<pre>[...] tasks: - name: Ensure the absence of DNS global forwarder 8.8.6.7 ipadnsconfig: forwarders: - ip_address: 8.8.6.7 state: absent</pre>	플레이북을 실행하면 오류가 발생합니다. 원래 설정 - 8.8.6.6, 8.8.6.7은 변경되지 않습니다.
8.8.6.6, 8.8.6.7	<pre>[...] tasks: - name: Ensure the absence of DNS global forwarder 8.8.6.7 ipadnsconfig: forwarders: - ip_address: 8.8.6.7 action: member state: absent</pre>	8.8.6.6

26.5. IDM의 DNS 전달 정책

IdM은 첫 번째 및 유일한 표준 **BIND** 전달 정책은 물론 **none IdM** 관련 전달 정책을 지원합니다.

forward first (기본값)

IdM BIND 서비스는 **DNS** 쿼리를 구성된 전달자에게 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 **BIND**는 인터넷의 서버를 사용하여 재귀 해결으로 돌아갑니다. 첫 번째 정책은 기본 정책이며 **DNS** 트래픽을 최적화하는 데 적합합니다.

forward only

IdM BIND 서비스는 **DNS** 쿼리를 구성된 전달자에게 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 **BIND**에서 클라이언트에 오류를 반환합니다. 정 방향 전용 정책은 분할 **DNS** 구성에 있는 환경에 권장됩니다.

none (forwarding disabled)

DNS 쿼리는 **none** 전달 정책으로 전달되지 않습니다. 전달 비활성화는 글로벌 전달 구성에 대한 영역별 재정의로만 유용합니다. 이 옵션은 **BIND** 구성에서 빈 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 **BIND** 서비스는 쿼리된 **DNS** 이름이 **IdM** 서버에 권한이 있는 영역에 속하는 경우 쿼리를 다른 서버로 전달하지 않습니다. 이러한 상황에서 **IdM** 데이터베이스에서 쿼리된 **DNS** 이름을 찾을 수 없으면 **NXDOMAIN** 응답이 반환됩니다. 포워딩은 사용되지 않습니다.

예 26.1. 예를 들면 Scenario

IdM 서버는 **test.example**에 대한 권한이 있습니다. **DNS** 영역. **BIND**는 **192.0.2.254** IP 주소가 있는 **DNS** 서버로 쿼리를 전달하도록 구성됩니다.

클라이언트가 존재하지 않는 **test.example**에 대한 쿼리를 전송하는 경우 **DNS** 이름인 **BIND**에서는 **IdM** 서버가 **test.example.** 영역에 대한 권한이 있고 쿼리를 **192.0.2.254.** 서버로 전달하지 않음을 탐지합니다. 결과적으로 **DNS** 클라이언트는 **NXDomain** 오류 메시지를 수신하여 사용자에게 쿼리된 도메인이 존재하지 않음을 알립니다.

26.6. ANSIBLE 플레이북을 사용하여 **IDM DNS** 글로벌 구성에 첫 번째 정책이 설정되어 있는지 확인합니다.

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM DNS**의 글로벌 전달 정책이 먼저 전달되도록 설정하는 방법을 설명합니다.

전달 첫 번째 **DNS** 전달 정책을 사용하는 경우 **DNS** 쿼리가 구성된 전달자에게 전달됩니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 **BIND**는 인터넷의 서버를 사용하여 재귀 해결로 돌아갑니다. **forward first policy**는 기본 정책입니다. 이는 트래픽 최적화에 적합합니다.

사전 요구 사항

- 프로시저를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 자세한 내용은 [ansible-freeipa 패키지 설치](#)를 참조하십시오.
- **IdM** 관리자 암호를 알고 있습니다.

- **IdM 환경에는 통합 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 구성하려는 **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 들어 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.

관련이 없는 원래 플레이북의 다른 모든 행을 삭제합니다. 현재 예제에 대해 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

- [IdM의 DNS 전달 정책을](#) 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 `README-dnsconfig.md` 파일을 참조하십시오.
- 자세한 샘플 플레이북은 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 디렉토리를 참조하십시오.

26.7. ANSIBLE 플레이북을 사용하여 IDM DNS에서 글로벌 전달자가 비활성화되었는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM DNS**에서 글로벌 전달자가 비활성화되었는지 확인하는 방법을 설명합니다. **disable**는 **forward_policy** 변수를 **none**으로 설정하여 수행됩니다.

글로벌 전달자를 비활성화하면 **DNS** 쿼리가 전달되지 않습니다. 전달 비활성화는 글로벌 전달 구성에 대한 영역별 재정의로만 유용합니다. 이 옵션은 **BIND** 구성에서 빈 전달자 목록을 지정하는 것과 동일한 **IdM**입니다.

사전 요구 사항

- 프로시저를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 자세한 내용은 [ansible-freeipa 패키지](#) 설치를 참조하십시오.
- **IdM** 관리자 암호를 알고 있습니다.

- **IdM 환경에는 통합 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 구성하려는 **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 들어 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 암호로 설정합니다.

- **forward_policy** 변수를 **none** 으로 설정합니다.

현재 예제에 대해 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

- [IdM의 DNS 전달 정책을](#) 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 `README-dnsconfig.md` 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 디렉터리에서 더 많은 샘플 플레이북을 참조하십시오.

26.8. ANSIBLE 플레이북을 사용하여 **IDM DNS**에서 정방향 및 역방향 조회 영역의 동기화가 비활성화되었는지 확인합니다.

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM DNS**에서 정방향 및 역방향 조회 영역이 동기화되지 않도록 하는 방법을 설명합니다.

사전 요구 사항

- 프로시저를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 자세한 내용은 [ansible-freeipa 패키지](#) 설치를 참조하십시오.
- **IdM** 관리자 암호를 알고 있습니다.
- **IdM** 환경에는 통합 **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 구성하려는 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 파일을 복사합니다. 예를 들어 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.

현재 예제에 대해 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

- [/usr/share/doc/ansible-freeipa/](#) 디렉토리의 **README-dnsconfig.md** 파일을 참조하십시오.
- 자세한 샘플 플레이북은 [/usr/share/doc/ansible-freeipa/playbooks/dnsconfig](#) 디렉토리를 참조하십시오.

27장. 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 절차에 대해 설명합니다.

- **IdM에서 지원되는 DNS 영역 유형**
- **IdM에서 구성할 수 있는 DNS 속성**
- **Ansible 플레이북을 사용하여 IdM DNS에 기본 영역을 생성하는 방법**
- **Ansible 플레이북을 사용하여 여러 변수가 있는 기본 IdM DNS 영역이 있는지 확인하는 방법**
- **Ansible 플레이북을 사용하여 IP 주소가 지정될 때 역방향 DNS 조회 영역이 있는지 확인하는 방법**

사전 요구 사항

- DNS 서비스가 IdM 서버에 설치되어 있습니다. **Red Hat Ansible Engine**을 사용하여 통합 **DNS**로 IdM 서버를 설치하는 방법에 대한 자세한 내용은 **Ansible 플레이북을 사용하여 ID 관리 서버** 설치를 참조하십시오.

27.1. 지원되는 DNS 영역 유형

IdM(Identity Management)은 기본 및 전달 영역의 두 가지 유형의 **DNS** 영역을 지원합니다. 이 섹션에서는 이러한 두 가지 유형의 영역에 대해 설명하고 **DNS** 전달을 위한 시나리오 예제를 포함합니다.



참고

이 가이드에서는 **Microsoft Windows DNS**에 사용되는 용어와 다른 영역 유형에 대해 **BIND** 용어를 사용합니다. **BIND**의 기본 영역은 **Microsoft Windows DNS**의 정방향 **조회 영역** 및 **역방향 조회 영역**과 동일한 용도로 사용됩니다. **BIND**의 전달 영역은 **Microsoft Windows DNS**의 **조건부 전달자**와 동일한 용도로 사용됩니다.

기본 DNS 영역

기본 **DNS** 영역에는 권한 있는 **DNS** 데이터가 포함되어 있으며 동적 **DNS** 업데이트를 허용할 수 있습니다. 이 동작은 표준 **BIND** 구성의 유형 **master** 설정과 동일합니다. **ipa dnszone-*** 명령을 사용하여 기본 영역을 관리할 수 있습니다.

표준 **DNS** 규칙에 따라 모든 기본 영역에는 권한 시작 (**SOA**) 및 네임서버 (**NS**) 레코드가 포함되어야 합니다. **IdM**은 **DNS** 영역을 생성할 때 이러한 레코드를 자동으로 생성하지만 적절한 위임을 생성하려면 **NS** 레코드를 상위 영역에 수동으로 복사해야 합니다.

표준 **BIND** 동작에 따라 서버에 권한이 없는 이름에 대한 쿼리가 다른 **DNS** 서버로 전달됩니다. 이러한 **DNS** 서버는 전달자라고 하며 쿼리에 대해 권한이 없을 수도 있습니다.

예 27.1. DNS 전달 시나리오 예

IdM 서버에는 **test.example.** 기본 영역이 포함되어 있습니다. 이 영역에는 **sub.test.example.** 이름에 대한 **NS** 위임 레코드가 포함되어 있습니다. 또한 **test.example.** 영역은 **sub.test.example** 하위 영역에 대한 **192.0.2.254 forwarder IP** 주소로 구성됩니다.

존재하지 않는 이름을 쿼리하는 클라이언트는 **NXDomain** 응답을 수신하며, **IdM** 서버가 이 이름에 대해 권한이 있기 때문에 전달이 수행되지 않습니다.

반면, **IdM** 서버가 이 이름에 대해 권한이 없으므로 **host1.sub.test.example.** 이름에 대한 쿼리가 구성된 **forwarder 192.0.2.254** 로 전달됩니다.

DNS 영역 전달

IdM의 관점에서는 전달 **DNS** 영역에 권한 있는 데이터가 포함되어 있지 않습니다. 사실, 앞으로 "**zone**"은 일반적으로 다음 두 가지 정보만 포함합니다.

- 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.

전달 DNS 영역은 **IdM-Active Directory (AD)** 신뢰의 컨텍스트에서 특히 유용합니다. **IdM DNS** 서버에서 **idm.example.com** 영역에 대한 권한이 있고 **AD DNS** 서버에 **ad.example.com** 영역에 대한 권한이 있는 경우 **ad.example.com**은 **idm.example.com** 기본 영역의 **DNS** 전달 영역입니다. 즉, **somehost.ad.example.com**의 IP 주소에 대한 쿼리가 **IdM** 클라이언트에서 제공되면 쿼리가 **ad.example.com** **IdM DNS** 전달 영역에 지정된 **AD** 도메인 컨트롤러로 전달됩니다.

27.2. 기본 IDM DNS 영역의 구성 속성

IdM(Identity Management)은 새로 고침 기간, 전송 설정 또는 캐시 설정과 같은 특정 기본 구성으로 새 영역을 생성합니다. **IdM DNS 영역 속성**에서 다음 옵션 중 하나를 사용하여 수정할 수 있는 기본 영역 구성의 속성을 찾을 수 있습니다.

- **CLI(명령줄 인터페이스)의 `dnszone-mod` 명령.** 자세한 내용은 **IdM CLI에서 기본 DNS 영역의 구성 편집**을 참조하십시오.
- **IdM 웹 UI.** 자세한 내용은 **IdM 웹 UI에서 기본 DNS 영역의 구성 편집**을 참조하십시오.
- **ipadnszone** 모듈을 사용하는 **Ansible** 플레이북입니다. 자세한 내용은 **IdM의 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 해당 레코드를 업데이트하는 방법을 정의합니다.

표 27.1. IdM DNS 영역 속성

속성	Ansible-freeipa 변수	설명
권한 있는 이름 서버	name_server	SOA MNAME이라고도 하는 기본 DNS 이름 서버의 도메인 이름을 설정합니다. 기본적으로 각 IdM 서버는 SOA MNAME 필드에 자신을 알립니다. 결과적으로 --name-server 를 사용하여 LDAP에 저장된 값이 무시됩니다.
관리자 이메일 주소	admin_email	영역 관리자에게 사용할 이메일 주소를 설정합니다. 이 기본값은 호스트의 root 계정입니다.
SOA 직렬	serial	SOA 레코드의 일련 번호를 설정합니다. IdM은 버전 번호를 자동으로 설정하며 사용자는 수정할 필요가 없습니다.

속성	Ansible-freeipa 변수	설명
SOA 새로 고침	새로 고침	기본 DNS 서버에서 업데이트를 요청하기 전에 대기할 간격(초)을 설정합니다.
SOA 재시도	Retry	실패한 새로 고침 작업을 다시 시도하기 전에 대기하는 시간(초)을 설정합니다.
SOA 만료	expire	작업을 종료하기 전에 보조 DNS 서버에서 새로 고침 업데이트를 수행하려고 하는 시간(초)을 설정합니다.
SOA 최소	minimum	RFC 2308 에 따라 음수 캐시의 시간(TTL) 값을 초 단위로 설정합니다.
SOA 지원 시간	tll	영역 apex에서 레코드의 TTL을 초 단위로 설정합니다. example.com 영역의 경우 example.com이라는 이름의 모든 레코드(A, NS 또는 SOA)가 구성되어 있지만 test.example.com 과 같은 다른 도메인 이름은 영향을 받지 않습니다.
기본 라이브 시간	default_ttl	이전에 개별 TTL 값이 설정되지 않은 영역의 모든 값에 대한 음수 캐싱을 위해 기본 시간을 TTL(TTL)으로 설정합니다. 변경 사항을 적용하려면 모든 IdM DNS 서버에서 named-pkcs11 서비스를 다시 시작해야 합니다.
BIND 업데이트 정책	update_policy	DNS 영역의 클라이언트에 허용되는 권한을 설정합니다.
동적 업데이트	dynamic_update=TRUE FALSE	클라이언트의 DNS 레코드에 대한 동적 업데이트를 활성화합니다. false로 설정하면 IdM 클라이언트 시스템에서 IP 주소를 추가하거나 업데이트할 수 없습니다.
전송 허용	allow_transfer=string	지정된 영역을 부분(;)으로 구분하여 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	Semicolons (;)로 구분된 DNS 쿼리를 실행할 수 있는 IP 주소 또는 네트워크 이름 목록을 제공합니다.
PTR 동기화 허용	allow_sync_ptr=1 0	영역의 A 또는 AAAA 레코드(전달 레코드)가 PTR 레코드(reverse) 레코드와 자동으로 동기화되는지 여부를 설정합니다.
영역 전달자	forwarder=IP_address	DNS 영역을 위해 특별히 구성된 전달자를 지정합니다. IdM 도메인에 사용된 글로벌 전달자와는 별개입니다. 여러 전달자를 지정하려면 옵션을 여러 번 사용합니다.

속성	Ansible-freeipa 변수	설명
forward policy	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/](#) 디렉토리의 **README-dnszone.md** 파일을 참조하십시오.

27.3. ANSIBLE을 사용하여 IDM DNS에 기본 영역 생성

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 기본 **DNS** 영역이 있는지 확인하는 방법을 보여줍니다. 아래 절차에 사용된 예에서 **IdM** 관리자는 **zone.idm.example.com** **DNS** 영역이 있는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 **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 파일을 복사합니다. 예를 들어 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.

현재 예제에 대해 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

- [지원되는 DNS 영역 유형](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-dnszone.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/dnszone` 디렉터리에서 샘플 **Ansible** 플레이북

을 참조하십시오.

27.4. ANSIBLE 플레이북을 사용하여 IDM에 여러 변수가 있는 기본 DNS 영역이 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 기본 **DNS** 영역이 있는지 확인하는 방법을 보여줍니다. 아래 절차에 사용된 예에서 **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. 프로세스의 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 구성하려는 **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 들어 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.
- **A** 및 **AAAA** 레코드의 동기화를 **PTR** 레코드와 **AAAA** 레코드의 동기화를 허용하려면 **allow_sync_ptr** 변수를 **true**로 설정합니다.
- **IdM** 클라이언트 시스템에서 **IP** 주소를 추가하거나 업데이트하는 데 **dynamic_update** 변수를 **true**로 설정합니다.
- 영역에 레코드의 인라인 **DNSSEC** 서명을 허용하려면 **dnssec** 변수를 **true**로 설정합니다.
- **allow_transfer** 변수를 영역에 있는 보조 이름 서버의 **IP** 주소로 설정합니다.
- **allow_query** 변수를 쿼리를 실행할 수 있는 **IP** 주소 또는 네트워크로 설정합니다.
- 전달자 변수를 글로벌 전달자의 **IP** 주소로 설정합니다.
- **serial** 변수를 **SOA** 레코드 일련 번호로 설정합니다.
- 영역의 **DNS** 레코드에 대한 새로 고침,재시도,만료,최소,**ttl** 및 **default_ttl** 값을 정의합니다.
- **nsec3param_rec** 변수를 사용하여 영역에 대한 **NSEC3PARAM** 레코드를 정의합니다.
- 기존 영역과 겹치는 경우에도 **skip_overlap_check** 변수를 **true**로 설정합니다.
- 이름 서버가 확인할 수 없는 경우에도 **DNS** 영역 생성을 강제 적용하려면 **skip_nameserver_check** 를 **true**로 설정합니다.

현재 예제에 대해 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
        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

- 지원되는 **DNS 영역 유형**을 참조하십시오.
- 기본 **IdM DNS 영역의 구성 속성**을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-dnszone.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/dnszone` 디렉터리에서 샘플 **Ansible** 플레이북을 참조하십시오.

27.5. ANSIBLE 플레이북을 사용하여 IP 주소가 지정될 때 역방향 DNS 조회 영역이 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 역방향 **DNS** 영역이 있는지 확인하는 방법을 보여줍니다. 아래 절차에 사용된 예에서 **IdM** 관리자는 **IdM** 호스트의 **IP** 주소 및 접두사 길이를 사용하여 역방향 **DNS** 조회 영역이 있는지 확인합니다.

`name_from_ip` 변수를 사용하여 **DNS** 서버의 **IP** 주소 접두사 길이를 제공하면 영역 이름을 제어할 수 있습니다. 접두사 길이를 지정하지 않으면 시스템은 **DNS** 서버를 영역을 쿼리하고 `name_from_ip` 값 **192.168.1.2**를 기반으로 쿼리는 다음 **DNS** 영역 중 하나를 반환할 수 있습니다.

- **1.168.192.in-addr.arpa.**
- **168.192.in-addr.arpa.**
- **192.in-addr.arpa.**

쿼리에서 반환된 영역이 예상하지 않을 수 있기 때문에 `name_from_ip`는 실수로 영역 제거를 방지하기 위해 **present**로 설정된 **state** 옵션에서만 사용할 수 있습니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 **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 들어 다음과 같습니다.

```
$ cp dnszone-reverse-from-ip.yml dnszone-reverse-from-ip-copy.yml
```

4. 편집할 **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 접두사 길이를 제공합니다.

현재 예제에 대해 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 }}"

```

Playbook은 192.168.1.2 IP 주소 및 해당 접두사 길이 24에서 역방향 DNS 조회 영역을 생성합니다. 다음으로 플레이북은 결과 영역 이름을 표시합니다.

6. 파일을 저장합니다.
7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file dnszone-reverse-from-ip-copy.yml
```

추가 리소스

- 지원되는 [DNS 영역 유형](#)을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 `README-dnszone.md` 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/dnszone` 디렉터리에서 샘플 Ansible 플레이북을 참조하십시오.

28장. ANSIBLE을 사용하여 IDM의 DNS 위치 관리

IdM(Identity Management) 관리자는 **ansible-freeipa** 패키지에서 사용할 수 있는 **location** 모듈을 사용하여 IdM DNS 위치를 관리할 수 있습니다. 이 장에서는 다음 주제 및 절차에 대해 설명합니다.

- [DNS 기반 서비스 검색](#)
- [DNS 위치에 대한 배포 고려 사항](#)
- [DNS 시간 \(TTL\)](#)
- [Ansible을 사용하여 IdM 위치 확인](#)
- [Ansible을 사용하여 IdM 위치 없음](#)

28.1. DNS 기반 서비스 검색

DNS 기반 서비스 검색은 클라이언트가 DNS 프로토콜을 사용하여 LDAP 또는 Kerberos 와 같은 특정 서비스를 제공하는 네트워크의 서버를 찾는 프로세스입니다. 일반적인 작업 유형 중 하나는 클라이언트가 더 높은 처리량과 네트워크 대기 시간을 줄이며 전체 비용을 절감하므로 가장 가까운 네트워크 인프라 내에서 인증 서버를 찾을 수 있도록 하는 것입니다.

서비스 검색의 주요 장점은 다음과 같습니다.

- 가까운 서버의 이름으로 명시적으로 구성할 필요는 없습니다.
- DNS 서버는 정책의 중앙 공급자로 사용됩니다. 동일한 DNS 서버를 사용하는 클라이언트는 서비스 공급자 및 기본 순서에 대한 동일한 정책에 액세스할 수 있습니다.

IdM(Identity Management) 도메인에서 LDAP, Kerberos 및 기타 서비스에 대한 DNS 서비스 레코드 (SRV 레코드)가 있습니다. 예를 들어 다음 명령은 IdM DNS 도메인에 TCP 기반 Kerberos 서비스를 제공하는 호스트의 DNS 서버를 쿼리합니다.

예 28.1. 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 선호됩니다.
- **100 (가장).** 동일한 우선순위가 있는 항목의 상대적 가중치를 지정합니다. 자세한 내용은 [RFC 2782](#), [섹션 3](#)에서 참조하십시오.
- **88 (포트 번호):** 서비스의 포트 번호.
- **서비스를 제공하는 호스트의 정식 이름입니다.**

이 예제에서 반환된 두 호스트 이름은 동일한 우선 순위와 **weight**를 갖습니다. 이 경우 클라이언트는 결과 목록의 임의의 항목을 사용합니다.

클라이언트가 **DNS** 위치에 구성된 **DNS** 서버를 쿼리하도록 구성된 경우 출력이 달라집니다. 위치에 할당된 **IdM** 서버의 경우 맞춤형 값이 반환됩니다. 아래 예제에서 클라이언트는 위치 **germany**에서 **DNS** 서버를 쿼리하도록 구성되어 있습니다.

예 28.2. 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 순위가 가장 낮은 값이 있으므로 우선합니다. **idmserver-02.idm.example.com**은 우선 순위가 높습니다. 따라서 기본 호스트를 사용할 수 없는 경우 백업으로만 사용됩니다.

28.2. DNS 위치에 대한 배포 고려 사항

IdM(Identity Management)은 통합 **DNS**를 사용할 때 위치별 서비스(**SRV**) 레코드를 생성할 수 있습니다. **IdM DNS** 서버는 위치별 **SRV** 레코드를 생성하기 때문에 각 **DNS** 위치에 적어도 하나의 **IdM DNS** 서버를 설치해야 합니다.

DNS 위치에 대한 클라이언트의 선호도는 클라이언트에서 수신한 **DNS** 레코드에서만 정의됩니다. 이러한 이유로 **IdM DNS** 서버를 비**IdM DNS** 소비자 서버와 결합하고 **DNS** 서비스 검색을 수행하는 클라이언트가 **IdM DNS** 서버에서 위치별 레코드를 확인하면 재구성할 수 있습니다.

IdM 및 비**IdM DNS** 서비스가 혼합된 대부분의 배포에서 **DNS** 재귀자는 왕복 시간 메트릭을 사용하여 가장 가까운 **IdM DNS** 서버를 자동으로 선택합니다. 일반적으로 **IdM**이 아닌 **DNS** 서버를 사용하는 클라이언트가 가장 가까운 **DNS** 위치에 대한 레코드를 수신하므로 최적의 **IdM** 서버 세트를 사용할 수 있습니다.

28.3. DNS 시간 (TTL)

클라이언트는 영역의 구성에 설정된 시간 동안 **DNS** 리소스 레코드를 캐시할 수 있습니다. 이러한 캐싱으로 인해 클라이언트는 **TTL(Time to live)** 값이 만료될 때까지 변경 사항을 수신하지 못할 수 있습니다. **IdM(Identity Management)**의 기본 **TTL** 값은 1일입니다.

클라이언트 컴퓨터가 사이트 간에 로밍하는 경우 **IdM DNS** 영역의 **TTL** 값을 조정해야 합니다. 클라이언트가 사이트 간에 로밍하는 데 필요한 시간보다 값을 더 낮은 값으로 설정합니다. 이렇게 하면 다른 사이트에 다시 연결하기 전에 클라이언트의 캐시된 **DNS** 항목이 만료되므로 **DNS** 서버를 쿼리하여 위치별 **SRV** 레코드를 새로 고칩니다.

추가 리소스

- [기본 IdM DNS 영역의 구성 속성](#) 을 참조하십시오.

28.4. ANSIBLE을 사용하여 IDM 위치 확인

IdM(Identity Management)의 시스템 관리자로서 **IdM DNS** 위치를 구성하여 클라이언트가 가장 가까운 네트워크 인프라 내의 인증 서버를 찾을 수 있도록 할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM**에 **DNS** 위치가 있는지 확인하는 방법을 설명합니다. 이 예제에서는 **IdM**에 **germany DNS** 위치가 있는지 확인하는 방법을 설명합니다. 결과적으로 로컬

IdM 클라이언트가 서버 응답 시간을 줄이는 데 사용할 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 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 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 사본을 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 맞게 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.

- **name** 변수를 위치 이름으로 설정합니다.

현재 예제에 대해 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: location present example
  hosts: ipaserver
  become: true

  tasks:
    - name: Ensure that the "germany" location is present
      ipalocation:
        ipadmin_password: Secret123
        name: germany
```

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

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

추가 리소스

- **IdM 웹 UI를 사용하여 DNS 위치에 IdM 서버 할당 또는 IdM CLI를 사용하여 DNS 위치에 IdM 서버 할당을 참조하십시오.**

28.5. ANSIBLE을 사용하여 IDM 위치 없음

IdM(Identity Management)의 시스템 관리자로서 **IdM DNS** 위치를 구성하여 클라이언트가 가장 가까운 네트워크 인프라 내의 인증 서버를 찾을 수 있도록 할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM**에서 **DNS** 위치가 없는지 확인하는 방법을 설명합니다. 이 예제에서는 **IdM**에서 **germany DNS** 위치가 없는지 확인하는 방법을 설명합니다. 결과적으로 특정 **IdM** 서버를 이 위치에 할당할 수 없으며 로컬 **IdM** 클라이언트에서 이 서버를 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 샘플 플레이북의 복사본을 저장할 중앙 위치로 **~/MyPlaybooks/** 디렉터리를 **생성하고 구성** 했다고 가정합니다.

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks/location/** 디렉터리에 있는 **location-absent.yml** 파일의 사본을 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 맞게 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.
- **name** 변수를 DNS 위치 이름으로 설정합니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.

현재 예제에 대해 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 파일과 인벤토리 파일을 지정하는 **Ansible** 플레이북을 실행합니다.

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

28.6. 추가 리소스

- `/usr/share/doc/ansible-freeipa/` 디렉토리에 있는 **README-location.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/location` 디렉터리에서 샘플 **Ansible** 플레이북을 참조하십시오.

29장. IDM에서 DNS 전달 관리

다음 절차에서는 **IdM(Identity Management)** 웹 UI에서 **DNS** 글로벌 전달자 및 **DNS** 전달 영역을 구성하는 방법, **IdM CLI** 및 **Ansible**을 사용하는 방법을 설명합니다.

- [IdM DNS 서버의 두 가지 역할](#)
- [IdM의 DNS 전달 정책](#)
- [IdM 웹 UI에 글로벌 전달자 추가](#)
- [CLI에 글로벌 전달자 추가](#)
- [IdM 웹 UI에 DNS Forward Zone 추가](#)
- [CLI에서 DNS Forward Zone 추가](#)
- [Ansible을 사용하여 IdM에서 DNS Global Forwarder 설정](#)
- [Ansible을 사용하여 IdM에 DNS 글로벌 전달자가 있는지 확인](#)
- [Ansible을 사용하여 IdM에 DNS 글로벌 전달자가 없는지 확인](#)
- [Ansible을 사용하여 IdM에서 DNS Global Forwarders가 비활성화되었는지 확인](#)
- [Ansible을 사용하여 IdM에 DNS Forward Zone이 있는지 확인](#)
- [Ansible을 사용하여 IdM에 DNS Forward Zone이 여러 전달자가 있는지 확인](#)

- **Ansible을 사용하여 IdM에서 DNS Forward Zone이 비활성화되었는지 확인**
- **Ansible을 사용하여 IdM의 DNS 전달 영역이 없는지 확인**

29.1. 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**와 **재귀 DNS 서버** 역할을 합니다.

권한 있는 DNS 서버

DNS 클라이언트에서 IdM 서버에 속하는 이름을 쿼리하면 BIND에서 구성된 영역에 포함된 데이터로 응답합니다. 권한 있는 데이터가 항상 다른 데이터보다 우선합니다.

재귀 DNS 서버

DNS 클라이언트에서 IdM 서버에 권한이 없는 이름을 쿼리하면 BIND에서 다른 DNS 서버를 사용하여 쿼리를 확인하려고 합니다. 전달자가 정의되지 않은 경우 BIND는 인터넷의 루트 서버를 요청하고 재귀 해결 알고리즘을 사용하여 DNS 쿼리에 응답합니다.

경우에 따라 BIND에서 다른 DNS 서버에 직접 연결하고 인터넷에서 사용 가능한 데이터를 기반으로 재귀를 수행하는 것이 바람직하지 않습니다. 다른 DNS 서버인 전달자를 사용하여 쿼리를 확인하도록 BIND를 구성할 수 있습니다.

전달자를 사용하도록 BIND를 구성하면 IdM 서버와 전달자 간에 쿼리 및 응답이 다시 전달되고 IdM 서버는 신뢰할 수 없는 데이터의 DNS 캐시 역할을 합니다.

29.2. IDM의 DNS 전달 정책

IdM은 첫 번째 및 유일한 표준 BIND 전달 정책은 물론 none IdM 관련 전달 정책을 지원합니다.

forward first (기본값)

IdM BIND 서비스는 DNS 쿼리를 구성된 전달자에게 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 BIND는 인터넷의 서버를 사용하여 재귀 해결으로 돌아갑니다. 첫 번째 정책은 기본 정책이며 DNS 트래픽을 최적화하는 데 적합합니다.

forward only

IdM BIND 서비스는 **DNS** 쿼리를 구성된 전달자에게 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 **BIND**에서 클라이언트에 오류를 반환합니다. 정 방향 전용 정책은 분할 **DNS** 구성에 있는 환경에 권장됩니다.

none (forwarding disabled)

DNS 쿼리는 **none** 전달 정책으로 전달되지 않습니다. 전달 비활성화는 글로벌 전달 구성에 대한 영역별 재정의로만 유용합니다. 이 옵션은 **BIND** 구성에서 빈 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 **BIND** 서비스는 쿼리된 **DNS** 이름이 **IdM** 서버에 권한이 있는 영역에 속하는 경우 쿼리를 다른 서버로 전달하지 않습니다. 이러한 상황에서 **IdM** 데이터베이스에서 쿼리된 **DNS** 이름을 찾을 수 없으면 **NXDOMAIN** 응답이 반환됩니다. 포워딩은 사용되지 않습니다.

예 29.1. 예를 들면 Scenario

IdM 서버는 **test.example**에 대한 권한이 있습니다. **DNS** 영역. **BIND**는 **192.0.2.254** IP 주소가 있는 **DNS** 서버로 쿼리를 전달하도록 구성됩니다.

클라이언트가 존재하지 않는 **test.example**에 대한 쿼리를 전송하는 경우 **DNS** 이름인 **BIND**에서는 **IdM** 서버가 **test.example**. 영역에 대한 권한이 있고 쿼리를 **192.0.2.254**. 서버로 전달하지 않음을 탐지합니다. 결과적으로 **DNS** 클라이언트는 **NXDomain** 오류 메시지를 수신하여 사용자에게 쿼리된 도메인이 존재하지 않음을 알립니다.

29.3. IDМ 웹 UI에 글로벌 전달자 추가

이 섹션에서는 **IdM(Identity Management)** 웹 UI에 글로벌 **DNS** 전달자를 추가하는 방법을 설명합니다.

사전 요구 사항

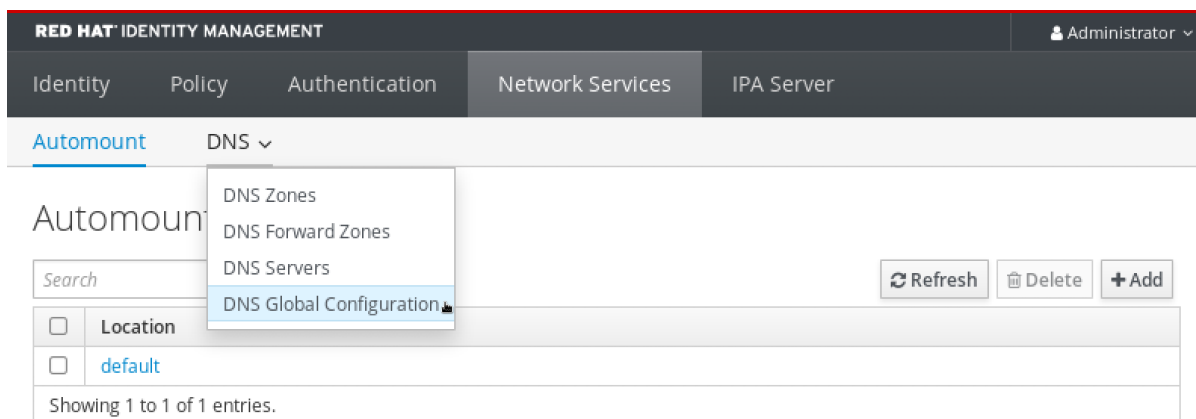
- **IdM** 관리자로 **IdM** 웹 UI에 로그인되어 있습니다.

• 쿼리를 전달할 **DNS** 서버의 **IP(Internet Protocol)** 주소를 알고 있습니다.

절차

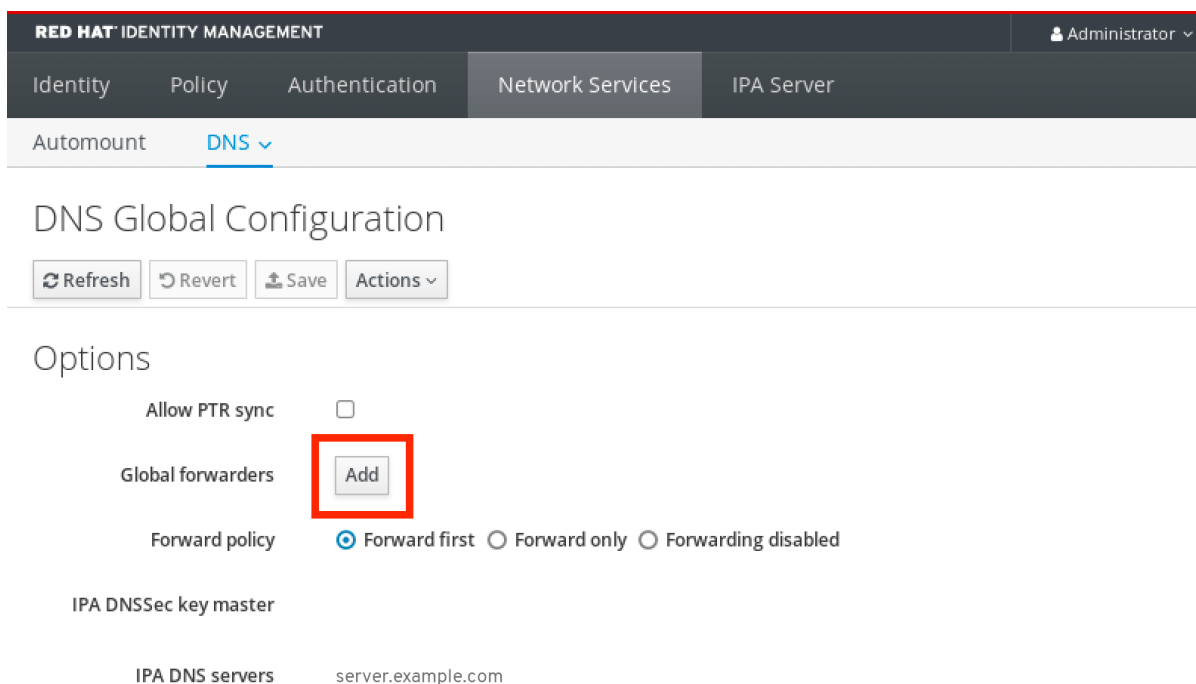
1.

IdM 웹 UI에서 네트워크 서비스 → **DNS** 글로벌 구성 → **DNS** 를 선택합니다.



2.

DNS 글로벌 구성 섹션에서 추가를 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 를 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

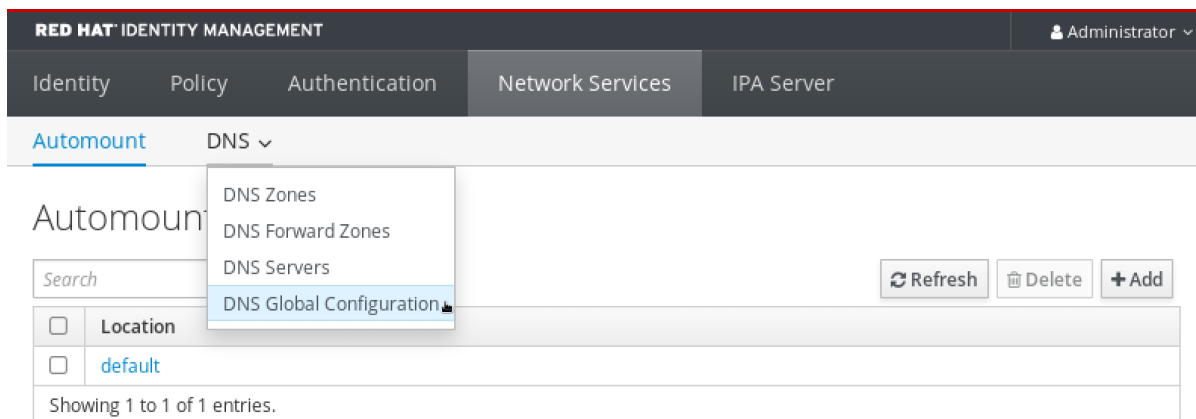
IPA DNS servers server.example.com

5.

창 위쪽의 저장을 클릭합니다.**Click Save at the top of the window.**

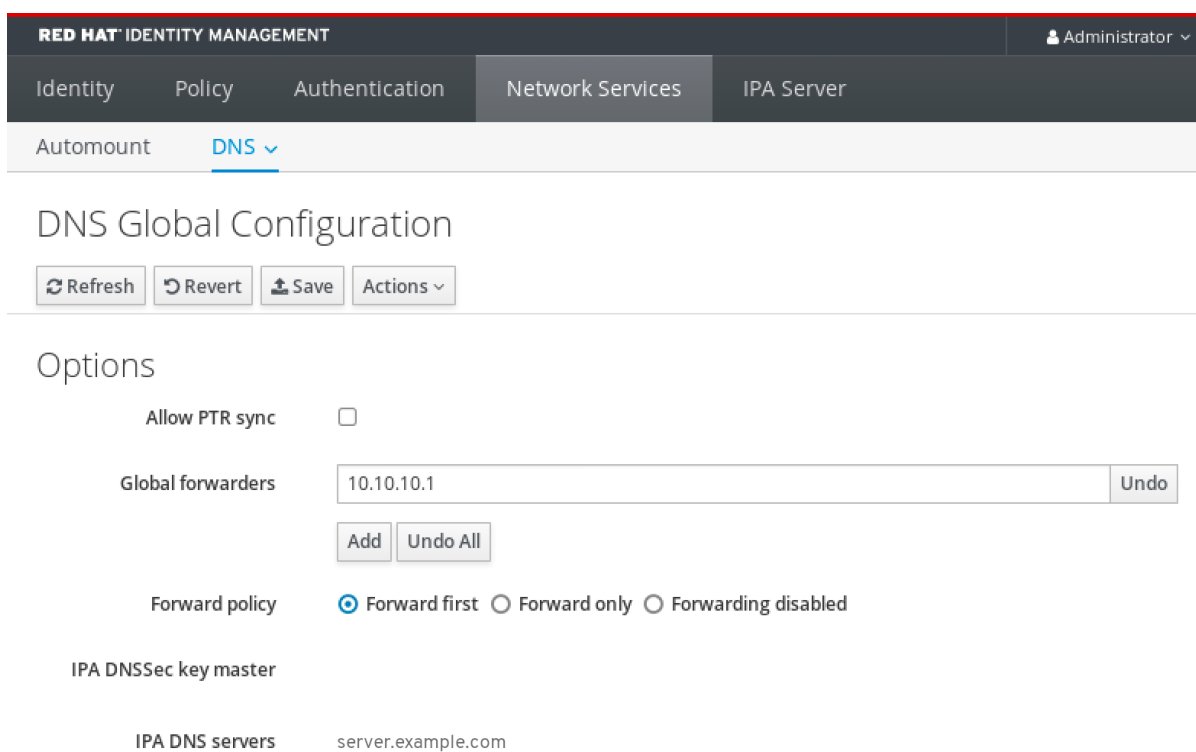
검증 단계

1.

네트워크 서비스 → **DNS** 글로벌 구성 → **DNS** 를 선택합니다.

2.

IdM 웹 UI에 지정한 전달 정책이 있고 활성화되어 있는 글로벌 전달자가 있는지 확인합니다.



29.4. CLI에 글로벌 전달자 추가

이 섹션에서는 **CLI**(명령줄 인터페이스)에서 글로벌 **DNS** 전달자를 추가하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자로 로그인되어 있습니다.

- 쿼리를 전달할 **DNS** 서버의 **IP(Internet Protocol)** 주소를 알고 있습니다.

절차

- **ipa dnsconfig-mod** 명령을 사용하여 새 글로벌 전달자를 추가합니다. **--forwarder** 옵션을 사용하여 **DNS** 전달자의 **IP** 주소를 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
```

29.5. IDM 웹 UI에 DNS FORWARD ZONE 추가

이 섹션에서는 **IdM(Identity Management)** 웹 UI에 **DNS** 전달 영역을 추가하는 방법을 설명합니다.

중요

반드시 필요한 경우를 제외하고 전달 영역을 사용하지 마십시오. 전달 영역은 표준 솔루션이 아니며 이를 사용하면 예기치 않은 문제가 발생할 수 있습니다. 전달 영역을 사용해야 하는 경우 글로벌 전달 구성을 재정의하는 데 사용을 제한합니다.

새 **DNS** 영역을 생성할 때 **Red Hat**은 항상 네임서버(**NS**) 레코드를 사용하여 표준 **DNS** 위임을 사용하고 정방향 영역을 방지하는 것이 좋습니다. 대부분의 경우 글로벌 전달자를 사용하는 것으로 충분하며 전달 영역이 필요하지 않습니다.

사전 요구 사항

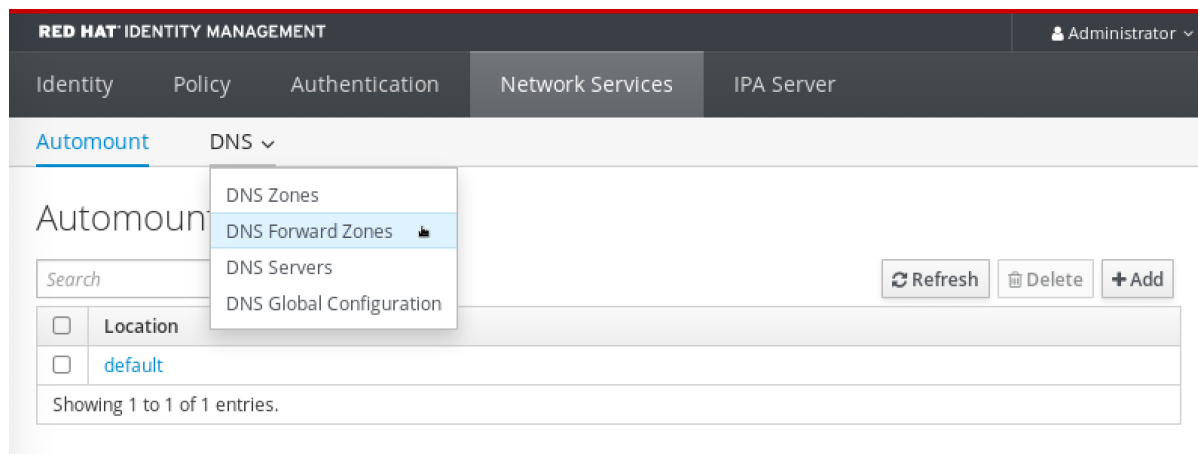
- **IdM** 관리자로 **IdM** 웹 UI에 로그인되어 있습니다.

쿼리를 전달할 **DNS** 서버의 **IP(Internet Protocol)** 주소를 알고 있습니다.

절차

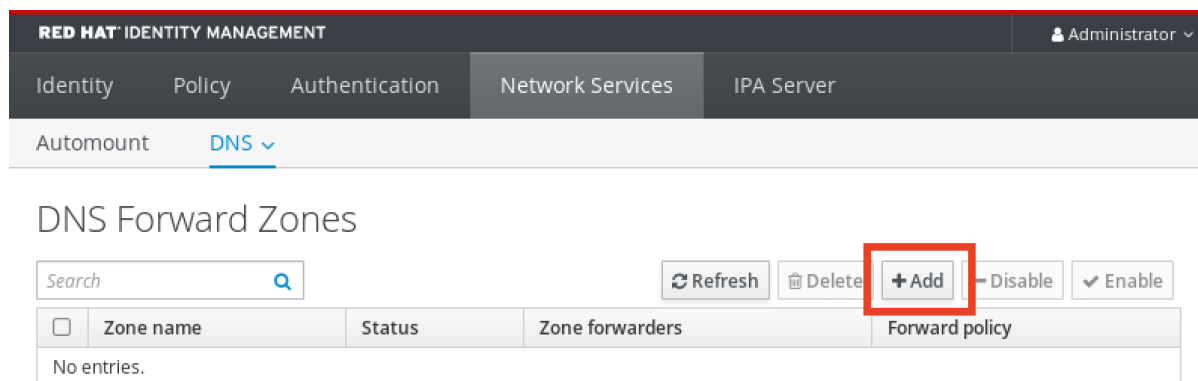
1.

IdM 웹 UI에서 네트워크 서비스 → **DNS Forward Zones** → **DNS** 를 선택합니다.



2.

DNS Forward Zones 섹션에서 추가를 클릭합니다.



3.

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.

추가 버튼을 클릭하고 **DNS** 서버의 **IP** 주소를 지정하여 전달 요청을 받습니다. 정방향 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 를 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

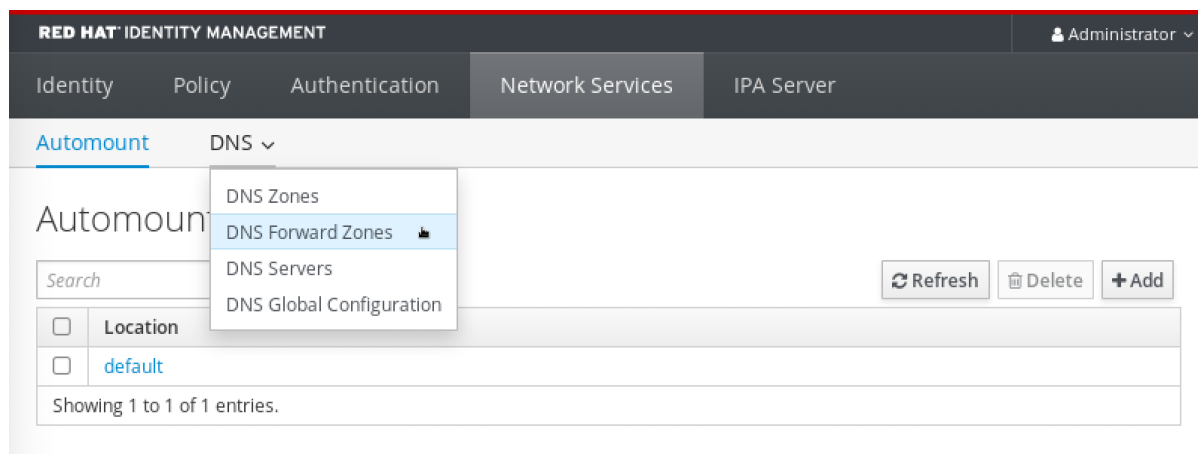
6.

창 하단에서 **Add** (추가)를 클릭하여 새 전달 영역을 추가합니다.

검증 단계

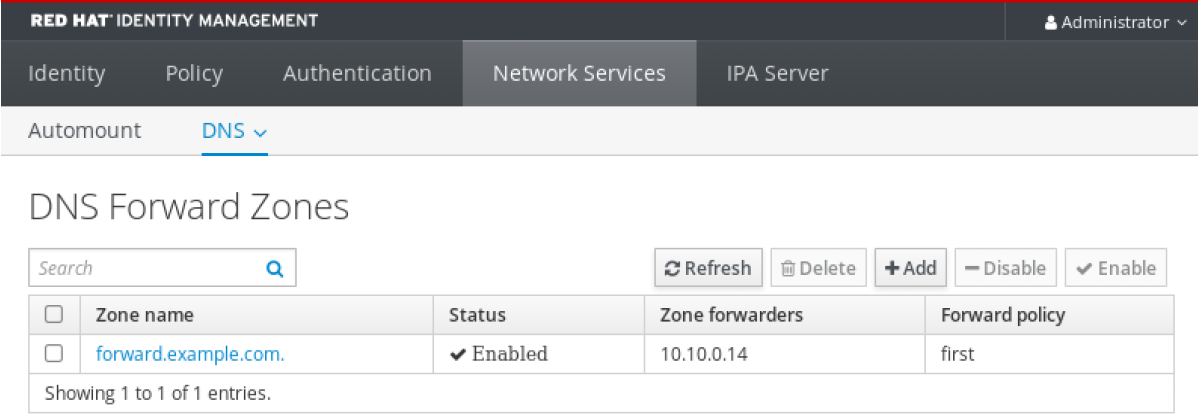
1.

IdM 웹 UI에서 네트워크 서비스 → **DNS Forward Zones** → **DNS** 를 선택합니다.



2.

IdM 웹 UI에 지정한 **forwarders** 및 **forward policy**를 사용하여 생성한 정방향 영역이 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.

29.6. CLI에서 DNS FORWARD ZONE 추가

이 섹션에서는 CLI(명령줄 인터페이스)에서 DNS 전달 영역을 추가하는 방법을 설명합니다.

중요

반드시 필요한 경우를 제외하고 전달 영역을 사용하지 마십시오. 전달 영역은 표준 솔루션이 아니며 이를 사용하면 예기치 않은 문제가 발생할 수 있습니다. 전달 영역을 사용해야 하는 경우 글로벌 전달 구성을 재정의하는 데 사용을 제한합니다.

새 DNS 영역을 생성할 때 Red Hat은 항상 네임서버(NS) 레코드를 사용하여 표준 DNS 위임을 사용하고 정방향 영역을 방지하는 것이 좋습니다. 대부분의 경우 글로벌 전달자를 사용하는 것으로 충분하며 전달 영역이 필요하지 않습니다.

사전 요구 사항

- IdM 관리자로 로그인되어 있습니다.
- 쿼리를 전달할 DNS 서버의 IP(Internet Protocol) 주소를 알고 있습니다.

절차

- **dnsforwardzone-add** 명령을 사용하여 새 전달 영역을 추가합니다. 전달 정책이 없는 경우 **--forwarder** 옵션으로 하나 이상의 전달자를 지정하고 **--forward-policy** 옵션으로 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
```

29.7. ANSIBLE을 사용하여 IDM에서 DNS GLOBAL FORWARDER 설정

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에서 DNS Global Forwarder를 설정하는 방법을 설명합니다.

아래 예제 절차에서 IdM 관리자는 포트 53에서 인터넷 프로토콜(IP) v4 주소가 8.8.6.6 이고 IPv6 주소 2001:4860:4860::8800 이 있는 DNS 서버에 DNS를 생성합니다.

사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 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. **set-configuration.yml** Ansible 플레이북 파일의 사본을 만듭니다. 예를 들어 다음과 같습니다.

```
$ cp set-configuration.yml establish-global-forwarder.yml
```

4. 편집할 **establish-global-forwarder.yml** 파일을 엽니다.
5. 다음 변수를 설정하여 파일을 조정합니다.
 - a. **IdM DNS**에서 글로벌 전달자를 설정하도록 플레이북의 **name** 변수를 **Playbook**로 변경합니다.
 - b. **tasks** 섹션에서 **DNS** 글로벌 전달자를 **8.8.6.6** 및 **2001:4860:4860::8800**으로 만들기 위해 작업 이름을 변경합니다.
 - c. **ipadnsconfig** 부분의 **forwarders** 섹션에서 다음을 수행합니다.
 - i. 첫 번째 **ip_address** 값을 글로벌 **forwarder**의 **IPv4** 주소로 변경합니다. **8.8.6.6**.
 - ii. 두 번째 **ip_address** 값을 **global forwarder**의 **IPv6** 주소로 변경합니다. **2001:4860:4860::8800**.
 - iii. 포트 값이 **53**으로 설정되어 있는지 확인합니다.
 - d. **forward_policy**를 먼저 **로** 변경합니다.

이는 현재 예에 대해 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/` 디렉토리의 **README-dnsconfig.md** 파일을 참조하십시오.

29.8. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에서 DNS 글로벌 전달자가 있는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 포트 53에 있는 IP(Internet Protocol) v4 주소가 7.7.9.9 인 DNS 서버로 DNS 글로벌 전달자가 있는지, 포트 53에서 2001:db8::1:0 의 IP v6 주소가 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 **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 들어 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4.

편집할 수 있도록 **ensure-presence-of-a-global-forwarder.yml** 파일을 엽니다.

5.

다음 변수를 설정하여 파일을 조정합니다.

a.

IdM DNS에 글로벌 전달자가 있는지 확인하기 위해 플레이북의 **name** 변수를 **Playbook**로 변경합니다.

b.

tasks 섹션에서 작업의 이름을 변경하여 **DNS** 글로벌 전달자가 포트 **53**의 **7.7.9.9** 및 **2001:db8::1:0**으로 있는지 확인합니다.

c.

ipadnsconfig 부분의 **forwarders** 섹션에서 다음을 수행합니다.

i.

첫 번째 **ip_address** 값을 **global forwarder: 7.7.9.9**의 **IPv4** 주소로 변경합니다.

ii.

두 번째 **ip_address** 값을 **global forwarder: 2001:db8::1:0**의 **IPv6** 주소로 변경합니다.

iii.

포트 값이 **53**으로 설정되어 있는지 확인합니다.

d.

state를 **present**로 변경합니다.

이는 현재 예에 대해 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/ 디렉토리의 **README-dnsconfig.md** 파일을 참조하십시오.

29.9. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에서 **DNS** 글로벌 전달자가 있는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 인터넷 프로토콜 (**IP**) **v4** 주소가 **8.8.6.6** 이고 포트 **53** 에서 **2001:4860:4860::8800** 의 **IP v6** 주소가 있는 **DNS** 글로벌 전달자가 있는지 확인합니다.

사전 요구 사항

-

Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 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 들어 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. 편집할 수 있도록 `ensure-absence-of-a-global-forwarder.yml` 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

- a. IdM DNS에 글로벌 전달자가 없도록 플레이북의 `name` 변수를 **Playbook**로 변경합니다.

- b. `tasks` 섹션에서 작업의 이름을 변경하여 **DNS** 글로벌 전달자가 `8.8.6.6` 및 `2001:4860:4860:4860::8800`이 없는 것을 확인하도록 합니다.

- c. `ipadnsconfig` 부분의 `forwarders` 섹션에서 다음을 수행합니다.

- i. 첫 번째 `ip_address` 값을 글로벌 `forwarder`의 IPv4 주소로 변경합니다. `8.8.6.6`.

- ii. 두 번째 `ip_address` 값을 `global forwarder`의 IPv6 주소로 변경합니다. `2001:4860:4860::8800`.

...

III.

포트 값이 **53** 으로 설정되어 있는지 확인합니다.

d.

action 변수를 **member** 로 설정합니다.

e.

상태가 **absent** 로 설정되어 있는지 확인합니다.

이는 현재 예에 대해 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
        action: member
        state: absent
```

중요

action: member 를 사용하지 않고 플레이북에서 **state: absent** 옵션만 사용하면 플레이북이 실패합니다.

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/ 디렉토리의 **README-dnsconfig.md** 파일

•

ipadnsconfig ansible-freeipa 모듈의 **action: member** 옵션**29.10. ANSIBLE을 사용하여 IDM에서 DNS GLOBAL FORWARDERS가 비활성화되었는지 확인**

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에서 DNS Global Forwarders가 비활성화되었는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 글로벌 전달자의 전달 정책이 **none** 으로 설정되어 전역 전달자를 효과적으로 비활성화합니다.

사전 요구 사항

•

Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 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 들어 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/ 디렉토리의 README-dnsconfig.md 파일을 참조하십시오.

29.11. ANSIBLE을 사용하여 IDM에 DNS FORWARD ZONE이 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에서 DNS Forward Zone이 있는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 example.com 의 DNS 전달 영역이 8. 8.8.8.8 인 IP(Internet Protocol) 주소가 있는 DNS 서버로 있는지 확인합니다.

사전 요구 사항

•

Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 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 들어 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-forwardzone.yml
```

4.

편집할 수 있도록 **ensure-presence-forwardzone.yml** 파일을 엽니다.

5.

다음 변수를 설정하여 파일을 조정합니다.

a.

IdM DNS에 **dnsforwardzone**이 있는지 확인하기 위해 플레이북의 **name** 변수를 **Playbook**로 변경합니다.

b.

tasks 섹션에서 **example.com**의 **dnsforwardzone**이 **8.8.8.8**로 있는지 확인하도록 작업 이름을 변경합니다.

c.

tasks 섹션에서 **ipadnsconfig** 제목을 **ipadnsforwardzone** 로 변경합니다.

d.

ipadnsforwardzone 섹션에서 다음을 수행합니다.

i.

ipaadmin_password 변수를 추가하고 **IdM** 관리자 암호로 설정합니다.

ii.

name 변수를 추가하고 **example.com** 으로 설정합니다.

iii.

전달자 섹션에서 다음을 수행합니다.

A.

ip_address 및 **port** 행을 제거합니다.

B.

대시 뒤에 지정하여 전달된 요청을 수신하도록 **DNS** 서버의 **IP** 주소를 추가합니다.

```
- 8.8.8.8
```

iv.

forwardpolicy 변수를 추가하고 먼저 **로** 설정합니다.

v.

skip_overlap_check 변수를 추가하고 **true** 로 설정합니다.

vi.

state 변수를 **present** 로 변경합니다.

이는 현재 예에 대해 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:
      ipadmin_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/ 디렉토리에서 **README-dnsforwardzone.md** 파일을 참조하십시오.

29.12. ANSIBLE을 사용하여 IDM에 DNS FORWARD ZONE이 여러 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.com의 DNS 전달 영역이 **8.8.8** 및 **4.4.4.4**로 전달되어 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 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 들어 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-multiple-forwarders.yml
```

4. 편집할 수 있도록 **ensure-presence-multiple-forwarders.yml** 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

- a. IdM DNS의 **dnsforwardzone**에 여러 개의 전달자가 있는지 확인하기 위해 플레이북의 **name** 변수를 Playbook로 변경합니다.

- b. **tasks** 섹션에서 **dnsforwardzone**의 **dnsforwardzone**의 **8.8.8.8** 및 **4.4.4.4** **forwarders**가 있는지 확인하도록 작업 이름을 변경합니다.

- c. **tasks** 섹션에서 **ipadnsconfig** 제목을 **ipadnsforwardzone** 로 변경합니다.
- d. **ipadnsforwardzone** 섹션에서 다음을 수행합니다.
 - i. **ipaadmin_password** 변수를 추가하고 **IdM** 관리자 암호로 설정합니다.
 - ii. **name** 변수를 추가하고 **example.com** 으로 설정합니다.
 - iii. 전달자 섹션에서 다음을 수행합니다.
 - A. **ip_address** 및 **port** 행을 제거합니다.
 - B. 확인하려는 **DNS** 서버의 **IP** 주소를 대시 앞에 추가합니다.

- 8.8.8.8
 - 4.4.4.4
 - iv. **state** 변수를 **present**로 변경합니다.

이는 현재 예에 대해 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:
        ipaadmin_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/](#) 디렉토리에서 **README-dnsforwardzone.md** 파일을 참조하십시오.

29.13. ANSIBLE을 사용하여 IDM에서 DNS FORWARD ZONE이 비활성화되었는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에서 DNS Forward Zone이 비활성화되었는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 **example.com**의 DNS 전달 영역이 비활성화되었는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 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 구성하려는 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 들어 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-disabled-forwardzone.yml
```

4.

편집할 **ensure-disabled-forwardzone.yml** 파일을 엽니다.

5.

다음 변수를 설정하여 파일을 조정합니다.

a.

IdM DNS에서 **dnsforwardzone**이 비활성화되도록 플레이북의 **name** 변수를 **Playbook**로 변경합니다.

b.

tasks 섹션에서 **example.com**의 **dnsforwardzone**이 비활성화되었는지 확인 하도록 작업 이름을 변경합니다.

c.

tasks 섹션에서 **ipadnsconfig** 제목을 **ipadnsforwardzone**로 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 전달자 섹션을 제거합니다.

iv.

state 변수를 **disabled**로 변경합니다.

이는 현재 예에 대해 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/](#) 디렉토리에서 **README-dnsforwardzone.md** 파일을 참조하십시오.

29.14. ANSIBLE을 사용하여 IDM의 DNS 전달 영역이 없는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에서 DNS Forward Zone이 없는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 **example.com**의 DNS 전달 영역이 없음을 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 [ansible-freeipa](#) 패키지를 설치했습니다. 프로세스의 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 구성하려는 IdM 서버가 **[ipaserver]** 섹션에 나열되어 있는지 확인합니

다. 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 들어 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-absence-forwardzone.yml
```

4. 편집할 수 있도록 **ensure-absence-forwardzone.yml** 파일을 엽니다.
5. 다음 변수를 설정하여 파일을 조정합니다.
 - a. **IdM DNS**에 **dnsforwardzone**이 없는 것을 확인하도록 플레이북의 **name** 변수를 **Playbook**로 변경합니다.
 - b. **tasks** 섹션에서 작업 이름을 변경하여 **example.com**의 **dnsforwardzone**이 없는 것을 확인합니다.
 - c. **tasks** 섹션에서 **ipadnsconfig** 제목을 **ipadnsforwardzone** 로 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 전달자 섹션을 제거합니다.
 - iv. **state** 변수를 **absent** 로 둡니다.

이는 현재 예에 대해 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:
      ipadmin_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/ 디렉토리에서 README-dnsforwardzone.md 파일을 참조 하십시오.](#)

29.15. 추가 리소스

- [DNS 전달](#)

30장. ANSIBLE을 사용하여 IDM의 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. 이 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](#)

30.1. IDM의 DNS 레코드

IdM(Identity Management)은 다양한 **DNS** 레코드 유형을 지원합니다. 다음 4개가 가장 자주 사용됩니다.

A

호스트 이름과 **IPv4** 주소에 대한 기본 맵입니다. **A** 레코드의 레코드 이름은 과 같은 호스트 이름입니다. **A** 레코드의 **IP** 주소 값은 **192.0.2.1** 과 같은 **IPv4** 주소입니다.

A 레코드에 대한 자세한 내용은 [RFC 1035](#) 에서 참조하십시오.

AAAA

호스트 이름과 **IPv6** 주소에 대한 기본 맵입니다. **AAAA** 레코드의 레코드 이름은 과 같은 호스트 이름입니다. **IP** 주소 값은 **2001:DB8::1111** 과 같은 **IPv6** 주소입니다.

AAAA 레코드에 대한 자세한 내용은 [RFC 3596](#) 을 참조하십시오.

SRV

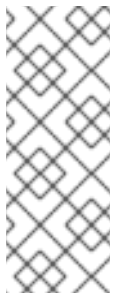
서비스(SRV) 리소스 레코드는 서비스 이름을 특정 서비스를 제공하는 서버의 **DNS** 이름에 매핑합니다. 예를 들어 이 레코드 유형은 **LDAP** 디렉터리와 같은 서비스를 관리하는 서버에 매핑할 수 있습니다.

SRV 레코드의 레코드 이름은 **_service._protocol**(예: **_ldap._tcp**)입니다. **SRV** 레코드에 대한 구성 옵션에는 대상 서비스의 우선순위, 가중치, 포트 번호, 호스트 이름이 포함됩니다.

SRV 레코드에 대한 자세한 내용은 [RFC 2782](#)를 참조하십시오.

PTR

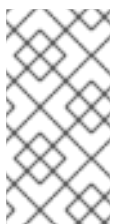
포인터 레코드(PTR)는 **IP** 주소를 도메인 이름에 매핑하는 역방향 **DNS** 레코드를 추가합니다.



참고

IPv4 주소에 대한 역방향 **DNS** 조회는 모두 **in-addr.arpa**. 도메인에 정의된 역방향 항목을 사용합니다. 사람이 읽을 수 있는 형식의 역방향 주소는 **in-addr.arpa**. 도메인이 추가된 일반 **IP** 주소와 정확하게 역방향입니다. 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 생성하려는 호스트의 정식 호스트 이름이어야 합니다.



참고

또한 **.ip6.arpa**. 도메인의 영역을 사용하여 **IPv6** 주소에 대해 역방향 영역을 구성할 수도 있습니다. **IPv6** 역방향 영역에 대한 자세한 내용은 [RFC 3596](#)을 참조하십시오.

DNS 리소스 레코드를 추가할 때 많은 레코드에 다른 데이터가 필요하다는 점에 유의하십시오. 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 추가하는 양식의 필드가 자동으로 업데이트됩니다.

30.2. 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 `--tektonoption={val1,val2,val3}`와 같이 **curly braces** 내부의 쉼표로 구분된 목록에 값을 나열하여 여러 항목을 정의할 수 있습니다.

표 30.1. 일반 레코드 옵션

옵션	설명
<code>--ttl=number</code>	레코드를 저장할 시간을 설정합니다.
<code>--structured</code>	원시 DNS 레코드를 구문 분석하고 구조화된 형식으로 반환합니다.

표 30.2. "A" 레코드 옵션

옵션	설명	예제
<code>--a-rec=ARECORD</code>	단일 A 레코드 또는 A 레코드 목록을 전달합니다.	<code>ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123</code>
	지정된 IP 주소를 사용하여 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 지정합니다. 레코드를 만들 때 A 레코드 값을 지정하는 옵션은 <code>--a-rec</code> 입니다. 그러나 A 레코드를 수정할 때 <code>--a-rec</code> 옵션을 사용하여 A 레코드의 현재 값을 지정합니다. 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 30.3. "AAAA" 레코드 옵션

옵션	설명	예제
--aaaa-rec=AAAAREC ORD	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	레코드의 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 --aaaa- ip-address 2001:db8::1231:5676

표 30.4. "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 리버스됩니다. 일반적으로 호스트 IP 주소는 지정된 네트워크에서 IP 주소의 마지막 노출입니다. 오른쪽에 있는 첫 번째 예제에서는 IPv4 주소가 192.168.122.4인 server4.idm.example.com 의 PTR 레코드를 추가합니다. 두 번째 예제에서는 reverse DNS 항목을 0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa 에 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 0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.i p6.arpa. 1.1.1.0.0.0.0.0.0.0.0.0.0.0 -- ptr-rec server2.idm.example.com.
--ptr-hostname=string	레코드의 호스트 이름을 지정합니다.	

표 30.5. "SRV" 레코드 옵션

옵션	설명	예제
--srv-rec=SRVRECORD	단일 SRV 레코드 또는 SRV 레코드 목록을 전달합니다. 오른쪽 예제에서 _ldap._tcp 는 SRV 레코드에 대한 서비스 유형 및 연결 프로토콜을 정의합니다. srv-rec 옵션은 priority, weight, port 및 target 값을 정의합니다. 예에서 51 및 49의 weight 값은 100까지 추가하고 특정 레코드가 사용되는 확률을 백분율로 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="1 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 설정하며, 우선 순위가 낮을 수 있습니다. 서비스는 우선 순위가 가장 높은 레코드를 먼저 사용해야 합니다.	# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="1 49 389 server2.idm.example.com" --srv-priority=0
--srv-weight=number	레코드의 가중치를 설정합니다. 동일한 우선 순위의 SRV 레코드 순서를 결정하는 데 도움이 됩니다. 설정 가중치는 100까지 추가해야 하며 특정 레코드가 사용되는 확률(분율)을 나타냅니다.	# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 49 389 server2.idm.example.com." --srv-weight=60
--srv-port=number	대상 호스트에서 서비스의 포트를 지정합니다.	# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 60 389 server2.idm.example.com." --srv-port=636
--srv-target=string	대상 호스트의 도메인 이름을 지정합니다. 도메인에서 서비스를 사용할 수 없는 경우 이 기간은 단일 기간(.)일 수 있습니다.	

추가 리소스

- **ipa dnsrecord-add --help** 를 실행합니다.

30.3. ANSIBLE을 사용하여 IDM에 A 및 AAAA DNS 레코드가 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 특정 IdM 호스트의 A 및 AAAA 레코드가 있는지 확인하는 방법을 보여줍니다. 아래 절차에 사용된 예제에서 IdM 관리자는 idm.example.com DNS 영역에 host1 의 A 및 AAAA 레코드가 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
-

idm.example.com 영역이 존재하며 **IdM DNS**에서 관리합니다. **IdM DNS**에서 기본 **DNS** 영역을 추가하는 방법에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 IdM DNS 영역 관리를 참조하십시오](#).

절차

1. `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성하려는 **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 파일을 복사합니다. 예를 들어 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** 로 설정합니다.
- **records** 변수에서 **name** 변수를 **host1** 로 설정하고 **aaaa_ip_address** 변수를 **::1** 로 설정합니다.

현재 예제에 대해 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

- [IdM의 DNS 레코드를](#) 참조하십시오.
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-dns history.md](#) 파일을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/playbooks/dnsrecord](#) 디렉토리의 샘플 **Ansible** 플레이북을 참조하십시오.

30.4. ANSIBLE을 사용하여 IDM에 A 및 PTR DNS 레코드가 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 해당 **PTR** 레코드와 함께 특정 **IdM** 호스트에 대한 **A** 레코드가 있는지 확인하는 방법을 보여줍니다. 아래 절차에 사용된 예

에서 **IdM** 관리자는 **idm.example.com** 영역에 IP 주소가 **192.168.122.45** 인 **host1** 의 **A** 및 **PTR** 레코드가 있는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 단계를 실행하는 호스트입니다.
- **IdM** 관리자 암호를 알고 있습니다.
- **idm.example.com** DNS 영역이 존재하며 **IdM DNS**에서 관리합니다. **IdM DNS**에서 기본 DNS 영역을 추가하는 방법에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 IdM DNS 영역 관리](#)를 참조하십시오.

절차

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성하려는 **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-dns history-with-reverse-is-present.yml** **Ansible** 플레이북 파일의 사본을 만듭니다. 예를 들어 다음과 같습니다.

```
$ cp ensure-dnsrecord-with-reverse-is-present.yml ensure-dnsrecord-with-reverse-is-present-copy.yml
```

4. 편집을 위해 **ensure-dns history-with-reverse-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 **host1** 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.

현재 예제에 대해 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

- [IdM의 DNS 레코드를 참조하십시오.](#)

- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-dns history.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 디렉토리의 샘플 **Ansible** 플레이북을 참조하십시오.

30.5. ANSIBLE을 사용하여 IDM에 여러 DNS 레코드가 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 **Ansible** 플레이북을 사용하여 여러 값이 특정 IdM DNS 레코드에 연결되어 있는지 확인하는 방법을 보여줍니다. 아래 절차에 사용된 예에서 IdM 관리자는 `idm.example.com` DNS 영역에 `host1`에 대한 여러 **A** 레코드가 있는지 확인합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- `idm.example.com` 영역이 존재하며 IdM DNS에서 관리합니다. IdM DNS에서 기본 DNS 영역을 추가하는 방법에 대한 자세한 내용은 **Ansible** 플레이북을 사용하여 IdM DNS 영역 관리를 참조하십시오.

절차

1. `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성하려는 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 들

어 다음과 같습니다.

```
$ cp ensure-presence-multiple-records.yml ensure-presence-multiple-records-copy.yml
```

4.

편집할 수 있도록 **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** 섹션에서 두 번째 레코드를 정의합니다.
 - **name** 변수를 **host1** 로 설정합니다.
 - **zone_name** 변수를 **idm.example.com** 으로 설정합니다.
 - **aaa_rec** 변수를 **::1** 로 설정합니다.

현재 예제에 대해 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:
  ipaadmin_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

- [IdM의 DNS 레코드를 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/ 디렉토리의 README-dns history.md 파일을 참조하십시오.](#)
- [/usr/share/doc/ansible-freeipa/playbooks/dnsrecord 디렉토리의 샘플 Ansible 플레이북을 참조하십시오.](#)

30.6. ANSIBLE을 사용하여 IDM에 여러 CNAME 레코드가 있는지 확인

Canonical Name 레코드(CNAME 레코드)는 하나의 도메인 이름인 별칭을 다른 이름인 다른 이름으로 매핑하는 **DNS(Domain Name System)**의 리소스 레코드 유형입니다.

단일 IP 주소에서 여러 서비스를 실행할 때 **CNAME 레코드(예: FTP 서비스 및 웹 서비스)**가 각각 다른 포트에서 실행되는 것을 확인할 수 있습니다.

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM DNS**에 여러 **CNAME 레코드**가 있는지 확인하는 방법을 보여줍니다. 아래 절차에 사용된 예에서 **host03**은 **HTTP** 서버

와 FTP 서버입니다. IdM 관리자는 **idm.example.com** 영역에 **host03 A** 레코드의 및 **ftp CNAME** 레코드가 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- **idm.example.com** 영역이 존재하며 IdM DNS에서 관리합니다. IdM DNS에서 기본 DNS 영역을 추가하는 방법에 대한 자세한 내용은 **Ansible 플레이북을 사용하여 IdM DNS 영역 관리를 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 구성하려는 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 들어 다음과 같습니다.

```
$ cp ensure-CNAME-record-is-present.yml ensure-CNAME-record-is-present-copy.yml
```

4. 편집할 수 있도록 **ensure-CNAME- records-is-present-copy.yml** 파일을 엽니다.

5.

ipadnsrecord 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- (선택 사항) 플레이 이름으로 제공한 설명을 지정합니다.
- **ipaadmin_password** 변수를 IdM 관리자 암호로 설정합니다.
- **zone_name** 변수를 **idm.example.com** 으로 설정합니다.
- **records** 변수 섹션에서 다음 변수 및 값을 설정합니다.
 - **name** 변수를 **www** 로 설정합니다.
 - **cname_hostname** 변수를 **host03** 으로 설정합니다.
 - **name** 변수를 **ftp** 로 설정합니다.
 - **cname_hostname** 변수를 **host03** 으로 설정합니다.

현재 예제에 대해 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:
  - ipadnsrecord:
    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/](#) 디렉토리의 **README-dns history.md** 파일을 참조하십시오.
- [/usr/share/doc/ansible-freeipa/playbooks/dnsrecord](#) 디렉토리의 샘플 **Ansible** 플레이북을 참조하십시오.

30.7. ANSIBLE을 사용하여 IDM에 SRV 레코드가 있는지 확인

DNS 서비스(SRV) 레코드는 도메인에서 사용할 수 있는 서비스의 호스트 이름, 포트 번호, 전송 프로토콜, 우선 순위 및 가중치를 정의합니다. **IdM(Identity Management)**에서 **SRV** 레코드를 사용하여 **IdM** 서버 및 복제본을 찾을 수 있습니다.

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM DNS**에 **SRV** 레코드가 있는지 확인하는 방법을 보여줍니다. 아래 절차에 사용된 예에서 **IdM** 관리자는 **_kerberos._udp.idm.example.com SRV** 레코드가 **10 50 88 idm.example.com**의 값으로 있는지 확인합니다. 이렇게 하면 다음 값이 설정됩니다.

- 서비스의 우선 순위를 **10**으로 설정합니다.
- 서비스의 가중치를 **50**으로 설정합니다.
- 서비스에서 사용할 포트를 **88**으로 설정합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 프로세스의 단계를 실행하는 호스트입니다.

- IdM 관리자 암호를 알고 있습니다.
- **idm.example.com** 영역이 존재하며 IdM DNS에서 관리합니다. IdM DNS에서 기본 DNS 영역을 추가하는 방법에 대한 자세한 내용은 [Ansible 플레이북을 사용하여 IdM DNS 영역 관리를 참조하십시오](#).

절차

1. `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성하려는 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-SRV- history-is-present.yml** Ansible 플레이북 파일의 사본을 만듭니다. 예를 들어 다음과 같습니다.

```
$ cp ensure-SRV-record-is-present.yml ensure-SRV-record-is-present-copy.yml
```

4. 편집할 수 있도록 **ensure-SRV- history-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.

이는 현재 예에 대해 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
    - ipadsnrecord:
        ipadmin_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

- [IdM의 DNS 레코드를 참조하십시오.](#)
- `/usr/share/doc/ansible-freeipa/` 디렉토리의 **README-dns history.md** 파일을 참조하십시오.
- `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 디렉토리의 샘플 **Ansible** 플레이북을 참조하십시오.