Red Hat Enterprise Linux 8

Ansible을 사용하여 Identity Management 설치 및 관리

Red Hat Ansible Engine을 사용하여 Red Hat Enterprise Linux 8에서 Identity Management 설치, 구성, 관리 및 유지 관리

Last Updated: 2022-07-25

Red Hat Enterprise Linux 8 Ansible을 사용하여 Identity Management 설치 및 관리

Red Hat Ansible Engine을 사용하여 Red Hat Enterprise Linux 8에서 Identity Management 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

초록

이 문서 컬렉션은 Ansible 플레이북을 효과적으로 사용하여 Red Hat Enterprise Linux 8에서 Identity Management를 설치, 구성, 관리 및 유지 관리하는 방법에 대한 지침을 제공합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체

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

Identity Management에서 계획된 용어 교체는 다음과 같습니다.

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

RED HAT 문서에 관한 피드백 제공

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

- 특정 문구에 대한 간단한 주석은 다음과 같습니다.
  1. 문서가 *Multi-page HTML* 형식으로 표시되는지 확인합니다. 또한 문서 오른쪽 상단에 **Feedback** (피드백) 버튼이 있는지 확인합니다.
  2. 마우스 커서를 사용하여 주석 처리하려는 텍스트 부분을 강조 표시합니다.
  3. 강조 표시된 텍스트 아래에 표시되는 **피드백 추가** 팝업을 클릭합니다.
  4. 표시된 지침을 따릅니다.
- 보다 상세하게 피드백을 제출하려면 다음과 같이 Bugzilla 티켓을 생성하십시오.
  1. Bugzilla 웹 사이트로 이동하십시오.
  2. Component로 **Documentation**을 선택하십시오.
  3. **Description** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
  4. **Submit Bug**를 클릭하십시오.

1장. ANSIBLE 용어

이 제목의 장에서는 공식 Ansible 용어를 사용합니다. 용어에 익숙하지 않은 경우 다음 섹션을 진행하기 전에 공식 Ansible 업스트림 설명서 를 읽어 보십시오.

- Ansible의 기본 개념은 Ansible에서 가장 일반적으로 사용되는 개념에 대한 개요를 제공합니다.
- 사용자 가이드는 명령줄 사용, 인벤토리 작업, 데이터 상호 작용, 작업, 플레이북 실행 등 Ansible을 시작할 때 가장 일반적인 상황과 질문을 간략하게 설명합니다.
- 인벤토리를 구축하는 방법은 재고 설계에 대한 팁을 제공합니다. 인벤토리는 Ansible이 인프라의 여러 관리 노드 또는 호스트에 대해 작업하는 데 사용하는 목록 또는 목록 그룹입니다.
- 플레이북 소개 에서는 구성 관리, 시스템 배포, 복잡한 애플리케이션 배포를 위한 반복 가능하고 재사용 가능한 시스템으로 Ansible 플레이북의 개념을 소개합니다.
- Ansible roles 섹션에서는 알려진 파일 구조를 기반으로 변수, 작업 및 핸들러 로드를 자동화하는 방법을 설명합니다.
- Glossary 는 Ansible 설명서의 다른 곳에서 사용되는 용어를 설명합니다.

2장. ANSIBLE 플레이북을 사용하여 ID 관리 서버 설치

2.1. ANSIBLE 및 IDM 설치의 이점

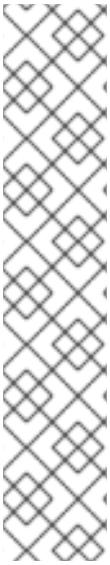
Ansible은 시스템을 구성하고, 소프트웨어를 배포하고, 롤링 업데이트를 수행하는 데 사용되는 자동화 도구입니다. Ansible에는 IdM(Identity Management) 지원이 포함되어 있으며 Ansible 모듈을 사용하여 IdM 서버, 복제본, 클라이언트 또는 전체 IdM 토폴로지 설정과 같은 설치 작업을 자동화할 수 있습니다.

Ansible을 사용하여 **IdM** 설치의 이점
다음 목록은 수동 설치와 달리 Ansible을 사용하여 Identity Management를 설치하는 경우의 장점을 보여줍니다.

- 관리 노드에 로그인할 필요가 없습니다.
- 개별적으로 배포되도록 각 호스트에서 설정을 구성할 필요는 없습니다. 대신 하나의 인벤토리 파일을 사용하여 전체 클러스터를 배포할 수 있습니다.
- 나중에 관리 작업에 대해 인벤토리 파일을 재사용할 수 있습니다(예: 사용자 및 호스트 추가). 등의 작업을 위해 IdM과 관련이 없는 경우에도 인벤토리 파일을 재사용할 수 있습니다.

2.2. ANSIBLE PLAYBOOK을 사용한 IDM 서버 설치

다음 섹션에서는 Ansible 을 사용하여 시스템을 IdM 서버로 구성하는 방법을 설명합니다. 시스템을 IdM 서버로 구성하면 IdM 도메인을 설정하고 시스템이 IdM 클라이언트에 IdM 서비스를 제공할 수 있습니다. 배포는 **ipaserver** Ansible 역할에서 관리합니다.



참고

Ansible을 사용하여 IdM 서버를 설치하기 전에 Ansible 및 IdM 개념을 이해해야 합니다. 이 장에서 사용되는 다음 용어를 이해해야 합니다.

- Ansible 역할
- Ansible 노드
- Ansible 인벤토리
- Ansible 작업
- Ansible 모듈
- Ansible 플레이 및 플레이북

개요
설치는 다음 부분으로 구성됩니다.

1. ansible-freeipa 패키지 설치
2. Ansible 플레이북을 사용하여 통합 CA가 있는 IdM 서버 배포
3. Ansible-playbook을 사용하여 외부 CA에서 IdM 서버 배포

2.3. ANSIBLE-FREEIPA 패키지 설치

사전 요구 사항

관리 노드에서:

- 관리 노드가 고정 IP 주소와 작업 패키지 관리자가 있는 Red Hat Enterprise Linux 8 시스템인지 확인합니다.

컨트롤러에서:

- 컨트롤러가 유효한 서브스크립션이 있는 Red Hat Enterprise Linux 시스템인지 확인합니다. 그렇지 않은 경우 대체 설치 지침은 공식 Ansible 설명서 설치 가이드를 참조하십시오.
- 컨트롤러의 **SSH** 프로토콜을 통해 관리 노드에 연결할 수 있는지 확인합니다. 관리 노드가 컨트롤러의 **/root/.ssh/known_hosts** 파일에 나열되어 있는지 확인합니다.

절차

Ansible 컨트롤러에서 다음 절차를 실행합니다.

1. 필요한 리포지토리를 활성화합니다.

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

2. Ansible을 설치합니다.

```
# yum install ansible
```

3. IdM Ansible 역할을 설치합니다.

```
# yum install ansible-freeipa
```

역할은 **/usr/share/ansible/roles/** 디렉터리에 설치됩니다.

2.4. 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 디렉터리는 README.md 마크다운 파일에서 예제, 기본 개요, 역할에 대한 라이센스 및 설명서를 저장합니다.

```
[root@server]# ls -1 /usr/share/ansible/roles/
ipaclient
ipareplica
ipaserver
```

- **/usr/share/doc/ansible-freeipa/** 디렉터리는 개별 역할 및 토폴로지에 대한 설명서를 README.md 마크다운 파일에 저장합니다. 또한 **playbooks/** 하위 디렉터리도 저장합니다(아래 참조).

```
[root@server]# ls -1 /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/** 디렉터리에 예제 플레이북이 저장됩니다.

```
[root@server]# ls -1 /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
```

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

2.5.1. 통합 CA를 루트 CA로 사용한 배포 매개변수 설정

통합 CA를 사용하여 IdM 서버를 루트 CA로 설치하기 위한 인벤토리 파일을 구성하려면 다음 절차를 완료합니다.

절차

1. 편집할 인벤토리 파일을 엽니다. IdM 서버로 사용할 호스트의**FQDN**(정규화된 도메인 이름)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. IdM 서버에 통합된 DNS가 포함되도록 하고 **/etc/resolv.conf** 파일의 전달자를 사용하려면 다음을 지정합니다.
4. **admin** 및 **Directory Manager** 의 암호를 지정합니다. Ansible Vault를 사용하여 암호를 저장하고 플레이북 파일에서 Vault 파일을 참조합니다. 또는 인벤토리 파일에서 직접 암호를 지정합니다.
5. (선택 사항) IdM 서버에서 사용할 사용자 지정 **firewalld** 영역을 지정합니다. 사용자 지정 영역을 설정하지 않으면 IdM이 해당 서비스를 기본 **firewalld** 영역에 추가합니다. 사전 정의된 기본 영역은 **public** 입니다.

중요

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

필요한 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
[...]
```

필요한 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** 영역을 사용하는 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 예제 플레이북

```
---
- 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 예제

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

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

IdM 서버 및 사용 가능한 옵션 설치에 대한 자세한 내용은 https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/installing_identity_management/index#installing-idm 을 참조하십시오.

2.5.2. Ansible 플레이북을 사용하여 통합 CA를 루트 CA로 사용하여 IdM 서버 배포

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

절차

- 플레이북 파일의 이름으로 **ansible-playbook** 명령을 실행합니다(예: **install-server.yml**). i **옵션을** 사용하여 인벤토리 파일을 지정합니다.

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

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

CLI(명령줄 인터페이스)에서 Ansible 플레이북 스크립트의 출력을 볼 수 있습니다. 다음 출력에서는 0개의 작업이 실패하므로 스크립트가 성공적으로 실행되었음을 보여줍니다.

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

Ansible 플레이북을 사용하여 호스트에 IdM 서버를 설치했습니다.

2.6. ANSIBLE 플레이북을 사용하여 외부 CA가 루트 CA로 있는 IDM 서버 배포

2.6.1. 외부 CA를 루트 CA로 사용하여 배포 매개 변수 설정

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

절차

1. 편집할 인벤토리 파일을 엽니다. IdM 서버로 사용할 호스트의**FQDN**(정규화된 도메인 이름)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. IdM 서버에 통합된 DNS가 포함되도록 하고 **/etc/resolv.conf** 파일의 전달자를 사용하려면 다음을 지정합니다.

4. **admin** 및 **Directory Manager** 의 암호를 지정합니다. Ansible Vault를 사용하여 암호를 저장하고 플레이북 파일에서 Vault 파일을 참조합니다. 또는 인벤토리 파일에서 직접 암호를 지정합니다.

5. (선택 사항) IdM 서버에서 사용할 사용자 지정 **firewalld** 영역을 지정합니다. 사용자 지정 영역을 설정하지 않으면 IdM이 해당 서비스를 기본 **firewalld** 영역에 추가합니다. 사전 정의된 기본 영역은 **public** 입니다.

중요

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

필요한 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
[...]
```

필요한 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** 영역을 사용하는 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

[...]
```

6. 설치 첫 번째 단계에 대한 플레이북을 만듭니다. CSR(인증서 서명 요청)을 생성하고 컨트롤러에서 관리 노드로 복사하기 위한 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
```

7. 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
```

외부 서명된 CA를 사용하여 IdM 서버를 설치할 때 사용할 수 있는 옵션에 대한 자세한 내용은 https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/installing_identity_management/index#installing-an-ipa-server-with-external-ca_installing-

identity-management 을 참조하십시오.

2.6.2. Ansible 플레이북을 사용하여 외부 CA가 루트 CA로 있는 IdM 서버 배포

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

절차

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

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

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

CLI(명령줄 인터페이스)에서 Ansible 플레이북 스크립트의 출력을 볼 수 있습니다. 다음 출력에서는 0개의 작업이 실패하므로 스크립트가 성공적으로 실행되었음을 보여줍니다.

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

2. 컨트롤러에서 **ipa.csr** 인증서 서명 요청 파일을 찾아 외부 CA에 제출합니다.

3. 외부 CA에서 서명한 IdM CA 인증서를 컨트롤러 파일 시스템에 배치하여 다음 단계의 플레이북에서 찾을 수 있도록 합니다.

4. 설치의 최종 단계에 대한 지침이 포함된 플레이북 파일의 이름으로 **ansible-playbook** 명령을 실행합니다(예: **install-server-step2.yml**). i **옵션을** 사용하여 인벤토리 파일을 지정합니다.

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

Ansible 플레이북을 사용하여 호스트에 외부 서명 CA가 있는 IdM 서버를 설치했습니다.

3장. ANSIBLE 플레이북을 사용하여 IDENTITY MANAGEMENT 복제본 설치

3.1. ANSIBLE 플레이북을 사용한 IDM 복제본 설치

다음 섹션에서는 Ansible 을 사용하여 시스템을 IdM 복제본으로 구성하는 방법을 설명합니다. 시스템을 IdM 복제본으로 구성하여 IdM 도메인에 등록하고 시스템이 도메인의 IdM 서버에서 IdM 서비스를 사용할 수 있도록 합니다.

배포는 **ipareplica** Ansible 역할로 관리합니다. 역할은 자동 검색 모드를 사용하여 IdM 서버, 도메인 및 기타 설정을 식별할 수 있습니다. 그러나 다양한 복제본 그룹이 서로 다른 시점에 배포되는 계층과 같은 모델에 여러 복제본을 배포하는 경우 각 그룹에 대해 특정 서버 또는 복제본을 정의해야 합니다.

참고

Ansible을 사용하여 IdM 복제본을 설치하기 전에 Ansible 및 IdM 개념을 이해해야 합니다. 이 장에서 사용되는 다음 용어를 이해해야 합니다.

- Ansible 역할
- Ansible 노드
- Ansible 인벤토리
- Ansible 작업
- Ansible 모듈
- Ansible 플레이 및 플레이북

개요
설치는 다음 부분으로 구성됩니다.

1. IdM 복제본 배포의 매개변수 설정
   - IdM 복제본 설치를 위한 기본, 서버 및 클라이언트 변수 지정
   - Ansible Playbook을 사용하여 IdM 복제본을 설치하기 위한 자격 증명 지정
2. Ansible 플레이북을 사용하여 IdM 복제본 배포

사전 요구 사항

- Ansible 제어 노드에 ansible-freeipa 패키지를 설치했습니다.

3.2. IDM 복제본 배포의 매개변수 설정

대상 호스트를 IdM 복제본으로 배포하기 전에 다음 설정을 구성합니다.

- IdM 복제본을 설치할 기본, 서버 및 클라이언트 변수를 지정합니다.
- Ansible 플레이북을 사용하여 IdM 복제본을 설치하기 위한 자격 증명을 지정합니다.

3.2.1. IdM 복제본 설치를 위한 기본, 서버 및 클라이언트 변수 지정

IdM 복제본을 설치하기 위한 인벤토리 파일을 구성하려면 다음 절차를 완료합니다.

절차

1. 편집할 인벤토리 파일을 엽니다. IdM 복제본이 되도록 호스트의 정규화된 도메인 이름(FQDN)을 지정합니다. **FQDN은** 유효한 DNS 이름이어야 합니다.

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

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

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

   IdM 서버가 이미 배포되어 있고 SRV 레코드가 IdM DNS 영역에 올바르게 설정된 경우 스크립트는 기타 필요한 모든 값을 자동으로 검색합니다.

2. 선택적으로 다음 시나리오 중 가장 가까운 시나리오를 기반으로 인벤토리 파일에 추가 정보를 제공합니다.

   - **시나리오 1**
     자동 검색을 방지하고 **[ipareplicas]** 섹션에 모든 복제본이 나열된 경우 특정 IdM 서버를 사용하는 경우 인벤토리 파일의 **[ipaservers]** 섹션에 서버를 설정합니다.

     IdM 서버 및 복제본의 **FQDN**이 정의된 인벤토리 호스트 파일 예

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

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

   - **시나리오 2**
     또는 자동 검색을 피하지만 특정 서버에 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 복제 소스로 사용합니다.

- **시나리오 3**
  여러 복제본을 한 배치에 배포하고 있으며 시간이 중요한 경우 다계층 복제본 배포가 유용할 수 있습니다. 인벤토리 파일에 특정 복제본 그룹(예: **[ipareplicas_tier1]** 및 **[ipareplicas_tier2]** 을 정의하고 **install-replica.yml** 플레이북에서 각 그룹에 대해 별도의 플레이를 설계합니다.

  복제본 계층이 정의된 인벤토리 파일의 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_servers 의 첫 번째 항목이 사용됩니다. 두 번째 항목은 대체 옵션으로 사용됩니다. IdM 복제본을 배포하는 데 여러 계층을 사용하는 경우, 먼저 tier1에서 복제본을 배포하기 위해 플레이북에 별도의 작업이 있어야 하고 tier2에서 복제본을 배포해야 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**
  복제본에서 기본 영역 대신 지정된 **firewalld** 영역을 사용하려면 인벤토리 파일에서 지정할 수 있습니다. 예를 들어, 기본값으로 설정된 공용 영역 대신 IdM 설치에 내부 **firewalld** 영역을 사용하려는 경우 유용할 수 있습니다.

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

중요

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

사용자 지정 **firewalld** 영역을 사용하는 간단한 인벤토리 호스트 파일 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
```

3.2.2. Ansible Playbook을 사용하여 IdM 복제본을 설치하기 위한 자격 증명 지정

IdM 복제본 설치를 위한 권한 부여를 구성하려면 다음 절차를 완료합니다.

절차

1. **복제본을 배포하도록 인증된 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 암호를 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 안전하게, 인벤토리 파일에서 직접 **admin** 의 자격 증명을 제공합니다. 인벤토리 파일의 **[ipareplicas:vars]** 섹션에 **ipaadmin_password** 옵션을 사용합니다. 그런 다음 인벤토리 파일과 **install-replica.yml** 플레이북 파일은 다음과 같이 표시됩니다.

  인벤토리 hosts.replica 파일 예

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

인벤토리 파일에서 주체 및 암호를 사용하는 플레이북의 예

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

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

- 또는 인벤토리 파일에 직접 복제본을 배포할 권한이 있는 다른 사용자의 자격 증명을 제공합니다. 권한 있는 다른 사용자를 지정하려면 사용자 이름에 **ipaadmin_principal** 옵션과 암호에 **ipaadmin_password** 옵션을 사용합니다. 그런 다음 인벤토리 파일과 **install-replica.yml** 플레이북 파일은 다음과 같이 표시됩니다.

  인벤토리 **hosts.replica** 파일 예

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

  인벤토리 파일에서 주체 및 암호를 사용하는 플레이북의 예

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

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

추가 리소스

- **ipareplica** Ansible 역할에서 수락한 옵션에 대한 자세한 내용은 **/usr/share/ansible/roles/ipareplica/README.md** Markdown 파일을 참조하십시오.

3.3. ANSIBLE 플레이북을 사용하여 IDM 복제본 배포

Ansible 플레이북을 사용하여 IdM 복제본을 배포하려면 다음 절차를 완료합니다.

절차

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

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

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

Ansible은 Ansible 플레이북 스크립트의 실행에 대해 알려줍니다. 다음 출력에서는 0개의 작업이 실패하므로 스크립트가 성공적으로 실행되었음을 보여줍니다.

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

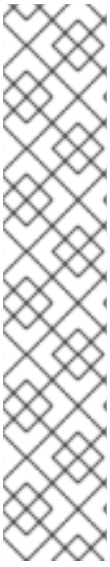
IdM 복제본을 설치했습니다.

4장. ANSIBLE 플레이북을 사용하여 IDENTITY MANAGEMENT 클라이언트 설치

4.1. ANSIBLE 플레이북을 사용한 IDM 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 역할에서 관리합니다. 기본적으로 이 역할은 IdM 서버, 도메인 및 기타 설정을 식별하기 위해 자동 검색 모드를 사용합니다. Ansible 플레이북에서 지정된 설정을 사용하도록 역할을 수정할 수 있습니다(예: 인벤토리 파일).



참고

Ansible을 사용하여 IdM 클라이언트를 설치하기 전에 Ansible 및 IdM 개념을 이해해야 합니다. 이 장에서 사용되는 다음 용어를 이해해야 합니다.

- Ansible 역할
- Ansible 노드
- Ansible 인벤토리
- Ansible 작업
- Ansible 모듈
- Ansible 플레이 및 플레이북

개요
설치는 다음 부분으로 구성됩니다.

1. 배포 시나리오에 대응하도록 IdM 클라이언트 배포의 매개 변수를 설정합니다.
   - 자동 검색 클라이언트 설치 모드에 대한 인벤토리 파일의 매개 변수 설정
   - 클라이언트 설치 중에 자동 검색을 수행할 수 없는 경우 인벤토리 파일의 매개 변수 설정
2. install-client.yml에서 매개 변수 확인
3. Ansible 플레이북을 사용하여 IdM 클라이언트 배포
4. 설치 후 Identity Management 클라이언트 테스트 .

또한 장에는 IdM 클라이언트 설치 제거 방법을 설명하는 섹션이 포함되어 있습니다 .

사전 요구 사항

- Ansible 제어 노드에 ansible-freeipa 패키지를 설치했습니다.

4.2. IDM 클라이언트 배포의 매개변수 설정

대상 호스트를 IdM 클라이언트로 배포하기 전에 제어 노드에서 배포 지침을 구성합니다. 또한 계획 중인 옵션에 따라 대상 호스트 매개변수를 구성합니다.

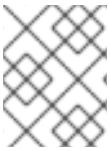
- 자동 검색 클라이언트 설치 모드 사용
- IdM 서버의 **FQDN** 및 도메인 또는 영역 정보 지정 .

4.2.1. 자동 검색 클라이언트 설치 모드에 대한 인벤토리 파일의 매개 변수 설정

Ansible 플레이북을 사용하여 Identity Management 클라이언트를 설치하려면 인벤토리 파일에 다음 정보를 제공합니다(예: **inventory/hosts**):

- 호스트에 대한 정보
- 작업에 대한 권한 부여

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



참고

RHEL에서 그래픽 사용자 인터페이스에서 스마트 카드를 사용하려면 Ansible 플레이북에 **ipaclient_mkhomedir** 변수를 포함해야 합니다.

절차

1. IdM 클라이언트가 되도록 호스트의 정규화된 호스트 이름(FQDN)을 지정합니다. 정규화된 도메인 이름은 유효한 DNS 이름이어야 합니다.
   - 숫자, 알파벳 문자 및 하이픈(-)만 허용됩니다. 예를 들어 밑줄은 허용되지 않으며 DNS 오류가 발생할 수 있습니다.
   - 호스트 이름은 모두 소문자여야 합니다. 대문자는 허용되지 않습니다.
     SRV 레코드가 IdM DNS 영역에 올바르게 설정된 경우 스크립트는 기타 필요한 모든 값을 자동으로 검색합니다.

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

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

2. 클라이언트 등록에 필요한 자격 증명을 지정합니다. 다음과 같은 인증 방법을 사용할 수 있습니다.
   - **고객 등록에 승인된 사용자의 암호입니다**. 기본 옵션입니다.
     - Red Hat은 Ansible Vault를 사용하여 암호를 저장하고 플레이북 파일에서 Vault 파일을 참조하는 것이 좋습니다(예: **install-client.yml**).

       Ansible Vault 파일의 인벤토리 파일 및 암호를 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
```

- 덜 안전하게, **inventory/hosts** 파일의 **[ipaclients:vars]** 섹션에서 **ipaadmin_password** 옵션을 사용하여 **admin** 의 자격 증명을 제공합니다. 또는 권한 있는 다른 사용자를 지정하려면 사용자 이름에 **ipaadmin_principal** 옵션을 사용하고 암호에 **ipaadmin_password** 옵션을 사용합니다. **inventory/hosts** 인벤토리 파일과 **install-client.yml** 플레이북 파일은 다음과 같이 표시될 수 있습니다.

  인벤토리 호스트 파일 예

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

  인벤토리 파일에서 주체 및 암호를 사용하는 플레이북의 예

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

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

- 이전 등록에서 계속 사용할 수 있는 경우 **client keytab** 입니다.
  - 이 옵션은 이전에 시스템이 ID 관리 클라이언트로 등록된 경우 사용할 수 있습니다. 이 인증 방법을 사용하려면 **#ipaclient_keytab** 옵션의 주석을 제거하고 keytab을 저장하는 파일의 경로를 지정합니다(예: **inventory/hosts** 의 **[ipaclient:vars]** 섹션).

- 등록 중에 생성되는 **임의의 OTP(one-time password)**입니다. 이 인증 방법을 사용하려면 인벤토리 파일에서 **ipaclient_use_otp=yes** 옵션을 사용합니다. 예를 들어 **inventory/hosts** 파일의 **[ipa clients:vars] 섹션에서 ipaclient _use_otp=yes** 옵션의 주석을 해제할 수 있습니다. OTP에서는 다음 옵션 중 하나를 지정해야 합니다.
  - 예를 들어 **inventory/hosts** 파일의 **[ipaclients:vars]** 섹션에 **ipaadmin_password** 값을 제공하여 **클라이언트를 등록하도록 인증된 사용자의 암호입니다**.
  - 예를 들어 **admin keytab** (예: 인벤토리/호스트의 **[ipaclients:vars]** 섹션에 **ipaadmin_keytab** 값을 제공하여).

추가 리소스

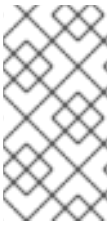
- **ipaclient Ansible 역할에서 수락한 옵션에 대한 자세한 내용은 /usr/share/ansible/roles/ipaclient/README.md README 파일을 참조하십시오.**

4.2.2. 클라이언트 설치 중에 자동 검색을 수행할 수 없는 경우 인벤토리 파일의 매개 변수 설정

Ansible 플레이북을 사용하여 **Identity Management** 클라이언트를 설치하려면 인벤토리 파일에 다음 정보를 제공합니다(예: **inventory/hosts**):

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

인벤토리 파일은 보유한 인벤토리 플러그인에 따라 다양한 형식 중 하나에 있을 수 있습니다. **INI**와 유사한 형식은 **Ansible**의 기본값 중 하나이며 아래 예제에서 사용됩니다.



참고

RHEL에서 그래픽 사용자 인터페이스에서 스마트 카드를 사용하려면 **Ansible** 플레이북에 **ipaclient_mkhomedir** 변수를 포함해야 합니다.

절차

1. **IdM** 클라이언트가 되도록 호스트의 정규화된 호스트 이름(**FQDN**)을 지정합니다. 정규화된 도메인 이름은 유효한 **DNS** 이름이어야 합니다.
   - 숫자, 알파벳 문자 및 하이픈(**-**)만 허용됩니다. 예를 들어 밑줄은 허용되지 않으며 **DNS** 오류가 발생할 수 있습니다.
   - 호스트 이름은 모두 소문자여야 합니다. 대문자는 허용되지 않습니다.
2. **inventory/hosts** 파일의 관련 섹션에 다른 옵션을 지정합니다.
   - 클라이언트가 등록할 **IdM** 서버를 나타내는 **[ipaservers]** 섹션에 있는 서버의 **FQDN**
   - 다음 두 옵션 중 하나 :
     - 클라이언트가 등록할 **IdM** 서버의 **DNS** 도메인 이름을 나타내는 **[ipaclients:vars]** 섹션의 **ipaclient_domain** 옵션

- IdM 서버에서 제어하는 Kerberos 영역의 이름을 나타내는 [ipaclients:vars] 섹션의 ipaclient_realm 옵션

클라이언트 FQDN, 서버 FQDN 및 정의된 도메인이 있는 인벤토리 호스트 파일의 예

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

[ipaservers]
server.idm.example.com

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

3. 클라이언트 등록에 필요한 자격 증명을 지정합니다. 다음과 같은 인증 방법을 사용할 수 있습니다.

- 고객 등록에 승인된 사용자의 암호입니다. 기본 옵션입니다.

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

Ansible Vault 파일의 인벤토리 파일 및 암호를 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
```

- 덜 안전하게, **inventory/hosts** 파일의 **[ipaclients:vars]** 섹션에서 **ipaadmin_password** 옵션을 사용하여 **admin** 의 자격 증명을 제공합니다. 또는 권한 있는 다른 사용자를 지정하려면 사용자 이름에 **ipaadmin_principal** 옵션을 사용하고 암호에 **ipaadmin_password** 옵션을 사용합니다. **install-client.yml** 플레이북 파일은 다음과 같이 표시됩니다.

  인벤토리 호스트 파일 예

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

  인벤토리 파일에서 주체 및 암호를 사용하는 플레이북의 예

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

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

- 이전 등록에서 계속 사용할 수 있는 경우 **client keytab** 입니다.

  - 이 옵션은 이전에 시스템이 **ID** 관리 클라이언트로 등록된 경우 사용할 수 있습니다. 이 인증 방법을 사용하려면 **ipaclient_keytab** 옵션의 주석을 제거하고 **keytab**을 저장하는 파일의 경로를 지정합니다(예: **inventory/hosts** 의 **[ipaclient:vars]** 섹션).

- 등록 중에 생성되는 임의의 **OTP(one-time password)**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 해제할 수 있습니다. **OTP**에서는 다음 옵션 중 하나를 지정해야 합니다.

  -

예를 들어 **inventory/hosts** 파일의 **[ipaclients:vars]** 섹션에 **ipaadmin_password** 값을 제공하여 클라이언트를 등록하도록 인증된 사용자의 암호입니다.

- 예를 들어 **admin keytab** (예: 인벤토리/호스트의 **[ipaclients:vars]** 섹션에 **ipaadmin_keytab** 값을 제공하여).

추가 리소스

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

4.2.3. install-client.yml 파일에서 매개변수 확인

install-client.yml 플레이북 파일에는 **IdM** 클라이언트 배포 지침이 포함되어 있습니다.

- 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 의미하는 바는 다음과 같습니다.

- **hosts** 항목은 **ansible** 스크립트에서 **ipa-client-install** 스크립트를 실행할 호스트의 **FQDN**을 검색하는 **inventory/hosts** 파일의 섹션을 지정합니다.
- **become: true** 항목은 **ipa-client-install** 스크립트를 실행하는 동안 루트 자격 증명이 호출되도록 지정합니다.
- 역할: **ipaclient** 항목은 호스트에 설치할 역할을 지정합니다. 이 경우 **ipa** 클라이언트 역할입니다.

- **state: present** 항목은 클라이언트를 설치 제거하지 않고 설치해야 함(없음)을 지정합니다.

4.2.4. Ansible Playbook을 사용하여 IdM 클라이언트 등록에 대한 권한 부여 옵션

이 참조 섹션에서는 인벤토리 및 플레이북 파일의 예와 함께 **IdM** 클라이언트 등록에 대한 개별 권한 부여 옵션을 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 vault에 저장된 암호	[ipaclients:vars] [...]	- name: Playbook to configure IPA clients with username/password hosts: ipaclients become: true vars_files: - playbook_sensitive_data.yml roles: - role: ipaclient state: present
클라이언트 등록 권한이 있는 사용자의 암호: 옵션 2	인벤토리 파일에 저장된 암호	[ipaclients:vars] ipaadmin_password=Secret123	- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true
임의의 OTP(one-time password): 옵션 1	OTP + 관리자 암호	[ipaclients:vars] ipaadmin_password=Secret123 ipaclient_use_otp=yes	- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true

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

4.3. ANSIBLE 플레이북을 사용하여 IDM 클라이언트 배포

Ansible 플레이북을 사용하여 IdM 환경에 IdM 클라이언트를 배포하려면 이 절차를 완료합니다.

절차

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

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

v, -vv 또는 -v vv 옵션을 사용하여 세부 정보 표시 수준을 지정합니다.

Ansible은 Ansible 플레이북 스크립트의 실행에 대해 알려줍니다. 다음 출력은 작업이 실패하지 않아 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 컬러를 사용하여 실행 중인 프로세스에 대한 다양한 유형의 정보를 제공합니다. /etc/ansible/ansible.cfg 파일의 [colors] 섹션에서 기본 컬러를 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.

4.4. ANSIBLE 설치 후 IDENTITY MANAGEMENT 클라이언트 테스트

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

ID 관리 클라이언트가 서버에 정의된 사용자에 대한 정보를 가져올 수 있는지 테스트하려면 서버에 정의된 사용자를 확인할 수 있는지 확인합니다. 예를 들어 기본 admin 사용자를 확인하려면 다음을 수행합니다.

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

인증이 올바르게 작동하는지 테스트하려면 su - 다른 기존 IdM 사용자로 다음을 수행합니다.

```
[user@client1 ~]$ su - idm_user
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
[idm_user@client1 ~]$
```

4.5. ANSIBLE 플레이북을 사용하여 IDM 클라이언트 설치 제거

Ansible 플레이북을 사용하여 호스트를 IdM 클라이언트로 제거하려면 다음 절차를 완료합니다.

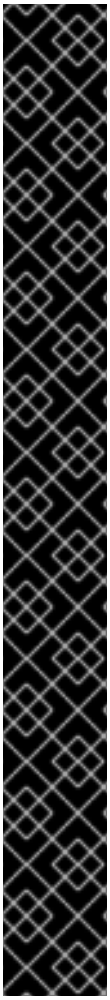
사전 요구 사항

- IdM 관리자 자격 증명.

절차

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

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



중요

클라이언트를 제거하면 호스트에서 기본 IdM 구성만 제거되지만 클라이언트를 다시 설치하기로 결정한 경우 호스트에 구성 파일이 남아 있습니다. 또한 설치 제거에는 다음과 같은 제한 사항이 있습니다.

- IdM LDAP 서버에서 클라이언트 호스트 항목을 제거하지 않습니다. 설치 제거만 호스트에 대한 등록을 취소합니다.
- IdM에서 클라이언트에 상주하는 서비스는 제거되지 않습니다.
- IdM 서버에서 클라이언트의 DNS 항목을 제거하지 않습니다.
- /etc/krb5.keytab 이외의 keytab 의 이전 주체는 제거하지 않습니다.

설치 제거는 IdM CA에서 호스트에 대해 발급한 모든 인증서가 제거됩니다.

추가 리소스

- 호스트와 IdM 환경 모두에서 IdM 클라이언트 구성을 완전히 제거하는 방법에 대한 자세한 내용은 IdM 클라이언트 제거에 대한 수동 절차를 참조하십시오.

5장. ANSIBLE 플레이북을 사용하여 IDM을 관리하기 위한 환경 준비

IdM(Identity Management)을 관리하는 시스템 관리자로서 Red Hat Ansible Engine을 사용하여 작업할 때 다음을 수행하는 것이 좋습니다.

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

이 연습을 사용하면 한 곳에서 모든 플레이북을 찾을 수 있으며 루트 권한을 호출하지 않고도 플레이북을 실행할 수 있습니다.

참고

관리 노드에서 root 권한만 있으면 ipaserver, ipareplica, ipa client 및 ipa backup ansible-freeipa 역할을 실행할 수 있습니다. 이러한 역할을 수행하려면 디렉토리 및 dnf 소프트웨어 패키지 관리자에 대한 권한 있는 액세스 권한이 필요합니다.

이 섹션에서는 ~/MyPlaybooks 디렉터리를 만들고 Ansible 플레이북을 저장하고 실행하는 데 사용할 수 있도록 구성하는 방법을 설명합니다.

사전 요구 사항

- 관리형 노드인 server.idm.example *.com 및 replica.idm.example.com에 IdM 서버를 설치했습니다.*
- 제어 노드에서 직접 관리형 노드인 *server.idm.example.com 및 replica.idm.example.com* 에 로그인할 수 있도록 DNS 및 네트워킹을 구성했습니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. 홈 디렉터리에서 **Ansible** 구성 및 플레이북의 디렉터리를 생성합니다.

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

2. **~/MyPlaybooks/** 디렉터리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

3. 다음 콘텐츠를 사용하여 **~/Myplaybooks/ansible.cfg** 파일을 만듭니다.

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

[privilege_escalation]
become=True
```

4. 다음 콘텐츠를 사용하여 **~/Myplaybooks/inventory** 파일을 만듭니다.

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

이 구성은 이러한 위치에 있는 호스트에 대한 두 개의 호스트 그룹인 **eu** 와 **us** 를 정의합니다. 또한 이 구성은 **eu** 및 **us** 그룹의 모든 호스트를 포함하는 **ipaserver** 호스트 그룹을 정의합니다.

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

```
$ ssh-keygen
```

6. **SSH** 공개 키를 각 관리 노드의 **IdM admin** 계정에 복사합니다.

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

이러한 명령을 사용하려면 IdM 관리자 암호를 입력해야 합니다.

추가 리소스

- Ansible 플레이북을 사용하여 IdM 서버 설치에 대한 자세한 내용은 Ansible 플레이북을 사용하여 ID 관리 서버 설치를 참조하십시오.
- 예제를 포함하여 Ansible 인벤토리 파일에 사용 가능한 형식 개요는 How to build your inventory(인벤토리 빌드 방법)를 참조하십시오.

6장. ANSIBLE 플레이북을 사용하여 글로벌 IDM 설정 구성

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

이 장에는 다음 섹션이 포함되어 있습니다.

- Ansible 플레이북을 사용하여 IdM 구성 검색
- Ansible 플레이북을 사용하여 IdM CA 갱신 서버 구성
- Ansible 플레이북을 사용하여 IdM 사용자에 대한 기본 쉘 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. inventory 파일(예: inventory.file)을 생성하고 [ipaserver] 섹션에서 IdM 구성을 검색하려는 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com에서 데이터를 검색하도록 Ansible에 지시하려면 다음을 입력합니다.

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

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

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

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

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

3. 다음을 변경하여 파일을 조정합니다.

   - IdM 관리자의 암호입니다.
   - 필요한 경우 기타 값입니다.

4. 파일을 저장합니다.

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

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

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


6.2. ANSIBLE 플레이북을 사용하여 IDM CA 갱신 서버 구성

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

IdM CA 갱신 서버의 역할에 대한 자세한 내용은 IdM CA 갱신 서버 사용을 참조하십시오.

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

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.

절차

1. 선택 사항: 현재 IdM CA 갱신 서버를 식별합니다.

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

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

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

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

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

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

4. 다음과 같이 변경하여 파일을 조정합니다.

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

5. 파일을 저장합니다.

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

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

검증 단계

CA 갱신 서버가 변경되었는지 확인할 수 있습니다.

1. IdM 관리자로 ipaserver 에 로그인합니다.

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

2. IdM CA 갱신 서버의 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 사용자에 대한 기본 쉘 구성

쉘은 명령을 수락하고 해석하는 프로그램입니다. bash,sh,ksh,zsh 등의 RHEL(Red Hat Enterprise Linux)에서 몇 가지 쉘을 사용할 수 있습니다. Bash 또는 /bin/bash 는 대부분의 Linux 시스템에서 널리 사용되는 쉘이며, 일반적으로 RHEL의 사용자 계정의 기본 쉘입니다.

다음 절차에서는 Ansible 플레이북을 사용하여 IdM 사용자의 기본 쉘로 대체 쉘인 sh 를 구성하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. 선택 사항: retrieve-config.yml Ansible 플레이북을 사용하여 IdM 사용자의 현재 쉘을 식별합니다. 자세한 내용은 Ansible 플레이북을 사용하여 IdM 구성 검색을 참조하십시오.

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

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

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

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

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

4. 다음을 변경하여 파일을 조정합니다.

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

5. 파일을 저장합니다.

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

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

검증 단계

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

1. IdM 관리자로 ipaserver 에 로그인합니다.

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

2. 현재 셸을 표시합니다.

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

로그인한 사용자는 sh 셸을 사용하고 있습니다.

추가 리소스

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

7장. ANSIBLE 플레이북을 사용하여 사용자 계정 관리

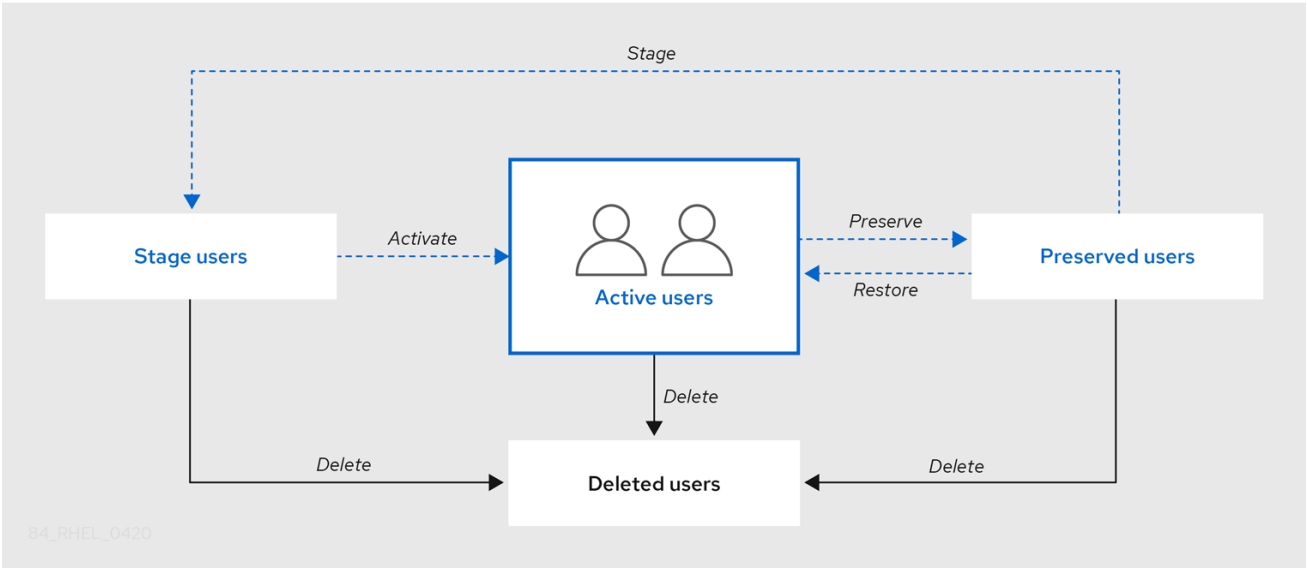
Ansible 플레이북을 사용하여 IdM에서 사용자를 관리할 수 있습니다. 사용자 라이프사이클 을 제공한 후 이 장에서는 다음 작업에 Ansible 플레이북을 사용하는 방법을 설명합니다.

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

7.1. 사용자 라이프 사이클

IdM(ID 관리)은 다음 세 가지 사용자 계정 상태를 지원합니다.

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



IdM 데이터베이스에서 영구적으로 사용자 항목을 삭제할 수 있습니다.

중요

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

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

주의

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

주의

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

7.2. ANSIBLE 플레이북을 사용하여 IDM 사용자가 있는지 확인

다음 절차에서는 Ansible 플레이북을 사용하여 IdM에 사용자가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

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

절차

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

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

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

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

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

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

사용자를 추가하려면 다음 옵션을 사용해야 합니다.

- **name:** 로그인 이름
- **first:** 첫 번째 이름 문자열
- **last:** 성 문자열

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

참고

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

3. 플레이북을 실행합니다.

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

검증 단계

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

1. **admin**으로 **ipaserver** 에 로그인합니다.

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

2. 관리자용 Kerberos 티켓을 요청합니다.

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

3. *idm_user* 에 대한 정보를 요청합니다.

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

이름이 *idm_user* 인 사용자는 IdM에 있습니다.

7.3. ANSIBLE PLAYBOOK을 사용하여 여러 IDM 사용자가 있는지 확인

다음 절차에서는 Ansible 플레이북을 사용하여 IdM에 여러 사용자가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

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

절차

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

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

2. IdM에서 확인할 사용자의 데이터로 Ansible 플레이북 파일을 생성합니다. 이 단계를 간소화

하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-users-present.yml** 파일에서 예제를 복사하고 수정할 수 있습니다. 예를 들어 ***idm_user_1, idm_user _2*** 및 ***idm_user _3*** 사용자를 생성하고, ***Password123*** 을 ***idm_user_1*** 의 암호로 추가하려면 다음을 수행합니다.

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

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

참고

update_password: on_create 옵션을 지정하지 않으면 **Ansible**은 플레이북을 실행할 때마다 사용자 암호를 다시 설정합니다. 플레이북이 마지막으로 실행된 이후 사용자가 암호를 변경한 경우 **Ansible**은 암호를 다시 설정합니다.

3. 플레이북을 실행합니다.

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

검증 단계

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

1. 관리자로 ipaserver 에 로그인합니다.

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

2. *idm_user_1* 에 대한 정보 표시:

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

IdM에 *idm_user_1* 이라는 사용자가 있습니다.

7.4. ANSIBLE 플레이북을 사용하여 JSON 파일에서 여러 IDM 사용자가 있는지 확인

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

사전 요구 사항

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

절차

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

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

2.

필요한 작업을 사용하여 **Ansible** 플레이북 파일을 생성합니다. 확인하고자 하는 사용자의 데이터로 **JSON** 파일을 참조합니다. 이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/ensure-users-present-ymlfile.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

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

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

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

3. **users.json** 파일을 생성하고 **IdM** 사용자를 추가합니다. 이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/user/users.json** 파일에서 예제를 복사하고 수정할 수 있습니다. 예를 들어 ***idm_user_1, idm_user _2*** 및 ***idm_user _3*** 사용자를 생성하고, ***Password123*** 을 ***idm_user_1*** 의 암호로 추가하려면 다음을 수행합니다.

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

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

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

검증 단계

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

1. 관리자로 ipaserver 에 로그인합니다.

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

2. *idm_user_1* 에 대한 정보 표시:

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

IdM에 *idm_user_1* 이라는 사용자가 있습니다.

7.5. ANSIBLE 플레이북을 사용하여 사용자가 없는지 확인

다음 절차에서는 Ansible 플레이북을 사용하여 특정 사용자가 IdM에 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

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

절차

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

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

2. IdM에 없는 사용자로 Ansible 플레이북 파일을 생성합니다. 이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/playbooks/user/ensure-users-present.yml 파일에서 예제를 복사하고 수정할 수 있습니다. 예를 들어 *idm_user_1, idm_user _2 및 idm_user _3* 사용자를 삭제하려면 다음을 수행합니다.

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

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

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

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

검증 단계

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

1. 관리자로 ipaserver 에 로그인합니다.

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

2. *idm_user_1* 에 대한 정보를 요청합니다.

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

이름이 *idm_user_1* 인 사용자는 IdM에 없습니다.

추가 리소스

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

- /usr/share/doc/ansible-freeipa/playbooks/user 디렉터리에 샘플 Ansible 플레이북을 볼 수도 있습니다.

8장. ANSIBLE 플레이북을 사용하여 사용자 그룹 관리

이 섹션에서는 Ansible 플레이북을 사용한 사용자 그룹 관리에 대해 소개합니다.

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

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

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

섹션에는 다음 주제가 포함됩니다.

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

8.1. IDM의 다양한 그룹 유형

IdM은 다음과 같은 유형의 그룹을 지원합니다.

POSIX 그룹(기본값)

POSIX 그룹은 구성원에 대해 Linux POSIX 특성을 지원합니다. Active Directory와 상호 작용하는 그룹은 POSIX 특성을 사용할 수 없습니다.

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

postIX 이외의 그룹

POST 이외의 그룹은 POSIX 특성을 지원하지 않습니다. 예를 들어, 이러한 그룹에는 GID가 정의되어 있지 않습니다.

이 유형의 그룹의 모든 멤버는 IdM 도메인에 속해야 합니다.

외부 그룹

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

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

외부 그룹은 POSIX 특성을 지원하지 않습니다. 예를 들어, 이러한 그룹에는 GID가 정의되어 있지 않습니다.

표 8.1. 기본적으로 생성된 사용자 그룹

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

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

주의

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

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

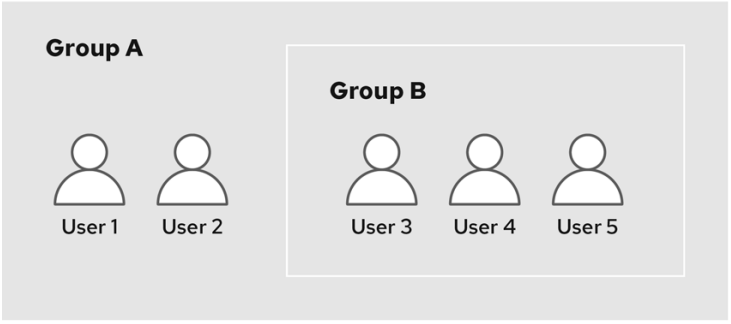
8.2. 직접 및 간접 그룹 구성원

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

예를 들어 다음 다이어그램에서 다음을 수행합니다.

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

그림 8.1. 직접 및 간접 그룹 멤버십



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

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

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

사전 요구 사항

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

절차

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

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

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

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

  tasks:
  - name: Create group ops with gid 1234
    ipagroup:
      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

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

1. 관리자로 **ipaserver** 에 로그인합니다.

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

2. ***ops*** 에 대한 정보를 표시합니다 :

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

appops 및 **sysops** 그룹 **- idm_user** 사용자를 포함하는 후자는 **IdM**에 있습니다.

추가 리소스

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

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

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

사전 요구 사항

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

절차

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

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

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

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

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

  - name: Ensure group_admins is present for group_a
    ipagroup:
```

```
    ipaadmin_password: MySecret123
    name: group_a
    membermanager_group: group_admins
```

3. 플레이북을 실행합니다.

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

검증 단계

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

1. 관리자로 ipaserver 에 로그인합니다.

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

2. *managergroup1* 에 대한 정보 표시 :

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

추가 리소스

- ipa host-add-member-manager --help를 참조하십시오.
- ipa 도움말 페이지를 참조하십시오.

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

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

사전 요구 사항

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

절차

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

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

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

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

  tasks:
  - name: Ensure member manager user and group members are absent for group_a
    ipagroup:
      ipaadmin_password: MySecret123
      name: group_a
      membermanager_user: test
      membermanager_group: group_admins
      action: member
      state: absent
```

3. 플레이북을 실행합니다.

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

검증 단계

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

1. 관리자로 **ipaserver** 에 로그인합니다.

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

2. **group_a**에 대한 정보를 표시합니다.

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

추가 리소스

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

9장. ANSIBLE 플레이북을 사용하여 IDM에서 셀프 서비스 규칙 관리

이 섹션에서는 IdM(Identity Management)의 셀프 서비스 규칙을 소개하고 Ansible 플레이북을 사용하여 셀프 서비스 액세스 규칙을 생성하고 편집하는 방법을 설명합니다. 셀프 서비스 액세스 제어 규칙을 사용하면 IdM 엔터티에서 IdM 디렉터리 서버 항목에서 지정된 작업을 수행할 수 있습니다.

이 섹션에서는 다음 주제를 다룹니다.

- IdM의 셀프 서비스 액세스 제어
- Ansible을 사용하여 셀프 서비스 규칙이 있는지 확인합니다.
- Ansible을 사용하여 셀프 서비스 규칙이 없는지 확인합니다.
- Ansible을 사용하여 셀프 서비스 규칙에 특정 특성이 있는지 확인합니다.
- Ansible을 사용하여 셀프 서비스 규칙에 특정 속성이 없는지 확인합니다.

9.1. IDM의 셀프 서비스 액세스 제어

셀프 서비스 액세스 제어 규칙은 IdM(Identity Management) 엔터티가 IdM Directory Server 항목에서 수행할 수 있는 작업을 정의합니다. 예를 들어 IdM 사용자는 자신의 암호를 업데이트할 수 있습니다.

이 제어 방법을 사용하면 인증된 IdM 엔터티에서 LDAP 항목 내에서 특정 속성을 편집할 수 있지만 전체 항목에서 작업을 추가하거나 삭제할 수는 없습니다.

주의

셀프 서비스 액세스 제어 규칙으로 작업할 때 주의하십시오. 액세스 제어 규칙을 부적절하게 구성하면 엔터티의 권한을 의도치 않게 높일 수 있습니다.

9.2. ANSIBLE을 사용하여 셀프 서비스 규칙이 있는지 확인합니다.

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

사전 요구 사항

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

절차

1. ~/***MyPlaybooks***/ 디렉터리로 이동합니다.

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

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

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

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

4. ipaselfservice 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- name 변수를 새 셀프 서비스 규칙의 이름으로 설정합니다.
- 권한 변수를 쉼표로 구분된 권한 목록(읽기 및 쓰기)으로 설정합니다.
- attribute 변수를 givenname,displayname ,title, initials 등 사용자가 자체적으로 관리할 수 있는 속성 목록으로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

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

5. 파일을 저장합니다.

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

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

추가 리소스

- 셀프 서비스 규칙 개념에 대한 자세한 내용은 IdM의 셀프 서비스 액세스 제어를 참조하십시오.
- **ipaselfservice** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
  - **/usr/share/doc/ansible -freeipa/** 디렉터리에 있는 **README- selfservice.md** 파일. 이 파일에는 **ipaselfservice** 변수 정의도 포함되어 있습니다.
  - **/usr/share/doc/ansible-freeipa/playbooks/selfservice** 디렉토리.

9.3. ANSIBLE을 사용하여 셀프 서비스 규칙이 없는지 확인합니다.

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

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 **Ansible** 제어 노드를 구성했습니다.
  - **Ansible** 버전 **2.8** 이상을 사용하고 있습니다.
  - ansible-freeipa 패키지를 설치했습니다.
  - ~/*MyPlaybooks*/ 디렉터리에 이러한 옵션을 구성하는 **IdM** 서버의 정규화된 도메인 이름

(FQDN)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

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

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

3. selfservice-absent-copy.yml Ansible 플레이북 파일을 편집하여 편집합니다.

4. ipaselfservice 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

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

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

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

5. 파일을 저장합니다.

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

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

추가 리소스

- 셀프 서비스 규칙 개념에 대한 자세한 내용은 IdM의 셀프 서비스 액세스 제어를 참조하십시오.
- ipaselfservice 모듈을 사용하는 샘플 Ansible Playbook은 다음을 참조하십시오.
  - /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README- selfservice.md 파일. 이 파일에는 ipaselfservice 변수 정의도 포함되어 있습니다.
  - /usr/share/doc/ansible-freeipa/playbooks/selfservice 디렉토리.

9.4. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 특성이 있는지 확인합니다.

다음 절차에서는 Ansible 플레이북을 사용하여 기존 셀프 서비스 규칙에 특정 설정이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 사용자가 자신의 이름 세부 정보 셀프 서비스 규칙에 성 멤버 속성도 관리할 수 있는지 확인합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - ansible-freeipa 패키지를 설치했습니다.

- ~/*MyPlaybooks*/ 디렉터리에 이러한 옵션을 구성하는 IdM 서버의 정규화된 도메인 이름(FQDN)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.

- 사용자는 IdM에 있는 고유한 이름 세부 정보 셀프 서비스 규칙을 관리할 수 있습니다.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/selfservice/ 디렉터리에 있는 selfservice-member-present.yml 파일의 복사본을 만듭니다.

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

3. selfservice-member-present-copy.yml Ansible 플레이북 파일을 편집하여 편집합니다.

4. ipaselfservice 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

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

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Self-service member present
```

```
  hosts: ipaserver
  become: true

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

5. 파일을 저장합니다.

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

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

추가 리소스

- 셀프 서비스 규칙 개념에 대한 자세한 내용은 **IdM**의 셀프 서비스 액세스 제어를 참조하십시오.
- **ipaselfservice** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
  - **/usr/share/doc/ansible -freeipa/** 디렉터리에 있는 **README- selfservice.md** 파일. 이 파일에는 **ipaselfservice** 변수 정의도 포함되어 있습니다.
  - **/usr/share/doc/ansible-freeipa/playbooks/selfservice** 디렉토리.

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

다음 절차에서는 **Ansible** 플레이북을 사용하여 셀프 서비스 규칙에 특정 설정이 없는지 확인하는 방법을 설명합니다. 이 플레이북을 사용하여 셀프 서비스 규칙이 바람직하지 않은 액세스 권한을 부여하지 않도록 할 수 있습니다. 이 예에서 사용자는 이름 세부 정보 셀프 서비스 규칙에 지정된 이름과 성 멤버 속성이 없는지 확인합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - ansible-freeipa 패키지를 설치했습니다.
  - ~/*MyPlaybooks*/ 디렉터리에 이러한 옵션을 구성하는 IdM 서버의 정규화된 도메인 이름(FQDN)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.
- 사용자는 IdM에 있는 고유한 이름 세부 정보 셀프 서비스 규칙을 관리할 수 있습니다.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

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

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

3. selfservice-member-absent-copy.yml Ansible 플레이북 파일을 편집하여 편집합니다.
4. ipaselfservice 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
   - name 변수를 수정할 셀프 서비스 규칙의 이름으로 설정합니다.

- 특성 변수를 주어진 이름 및 성으로 설정합니다.
- action 변수를 member 로 설정합니다.
- state 변수를 absent 로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

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

5. 파일을 저장합니다.

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

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

추가 리소스

- 셀프 서비스 규칙 개념에 대한 자세한 내용은 IdM의 셀프 서비스 액세스 제어를 참조하십시오.
- ipaselfservice 모듈을 사용하는 샘플 Ansible Playbook은 다음을 참조하십시오.
  -

/usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README- selfservice.md 파일. 이 파일에는 ipaselfservice 변수 정의도 포함되어 있습니다.

- **/usr/share/doc/ansible-freeipa/playbooks/selfservice 디렉토리.**

10장. ANSIBLE 플레이북을 사용하여 사용자를 관리하기 위해 사용자 그룹에 권한을 위임

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

이 섹션에서는 다음 주제를 다룹니다.

- 위임 규칙
- IdM용 Ansible 인벤토리 파일 생성
- Ansible을 사용하여 위임 규칙이 있는지 확인합니다.
- Ansible을 사용하여 위임 규칙이 없는지 확인합니다.
- 위임 규칙에 특정 속성이 있는지 확인하려면 Ansible을 사용합니다.
- 위임 규칙에 특정 속성이 없는지 확인하려면 Ansible을 사용합니다.

10.1. 위임 규칙

위임 규칙을 만들어 사용자 그룹에 권한을 위임하여 사용자를 관리할 수 있습니다.

위임 규칙을 사용하면 특정 사용자 그룹이 다른 사용자 그룹의 특정 속성에 대한 쓰기(편집) 작업을 수행할 수 있습니다. 이 형태의 액세스 제어 규칙은 위임 규칙에서 지정한 속성의 값 편집으로 제한됩니다. 전체 항목을 추가 또는 제거하는 기능을 부여하지 않습니다. 지정되지 않은 속성에 대해 전체 항목을 추가하거나 제어하는 기능을 부여하지 않습니다.

위임 규칙은 IdM의 기존 사용자 그룹에 권한을 부여합니다. 예를 들어 위임을 사용하여 managers 사용자 그룹이 employees 사용자 그룹에서 선택한 사용자 속성을 관리할 수 있습니다.

10.2. IDM용 ANSIBLE 인벤토리 파일 생성

Ansible을 사용하는 경우 홈 디렉터리에서 **/usr/share/doc/ ansible-freeipa/*** 및 **/usr/share/doc - roles/*** 하위 디렉터리를 복사하고 적응하는 **Ansible** 플레이북 전용 하위 디렉터리를 생성하는 것이 좋습니다. 이 연습에는 다음과 같은 이점이 있습니다.

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

절차

1. 홈 디렉터리에서 **Ansible** 구성 및 플레이북의 디렉터리를 생성합니다.

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

2. **~/MyPlaybooks/** 디렉터리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

3. 다음 콘텐츠를 사용하여 **~/Myplaybooks/ansible.cfg** 파일을 만듭니다.

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

[privilege_escalation]
become=True
```

4. 다음 콘텐츠를 사용하여 **~/Myplaybooks/inventory** 파일을 만듭니다.

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

이 구성은 이러한 위치에 있는 호스트에 대한 두 개의 호스트 그룹인 **eu** 와 **us** 를 정의합니다.

또한 이 구성은 **eu** 및 **us** 그룹의 모든 호스트를 포함하는 **ipaserver** 호스트 그룹을 정의합니다.

10.3. ANSIBLE을 사용하여 위임 규칙이 있는지 확인합니다.

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

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

사전 요구 사항

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

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/delegation/ 디렉토리에 있는 delegation-present.yml 파일의 복사본을 만듭니다.

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

3. 편집 할 delegation-present-copy.yml Ansible 플레이북 파일을 엽니다.

4. ipadelegation 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- name 변수를 새 위임 규칙의 이름으로 설정합니다.
- 권한 변수를 쉼표로 구분된 권한 목록(읽기 및 쓰기)으로 설정합니다.
- 속성 변수를 위임된 사용자 그룹이 관리할 수 있는 속성 목록(business category,departmentnumber, employeenumber, employeetype) 으로 설정합니다.
- 그룹 변수를 특성 보기 또는 수정에 대한 액세스 권한이 부여되는 그룹의 이름으로 설정합니다.
- membergroup 변수를 속성을 보거나 수정할 수 있는 그룹의 이름으로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

```
tasks:
- name: Ensure delegation "basic manager attributes" is present
  ipadelegation:
    ipaadmin_password: Secret123
    name: "basic manager attributes"
    permission: read, write
    attribute:
    - businesscategory
    - departmentnumber
    - employeenumber
    - employeetype
    group: managers
    membergroup: employees
```

5. 파일을 저장합니다.

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

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

추가 리소스

- 위임 규칙 개념에 대한 자세한 내용은 위임 규칙을 참조하십시오.
- **ipadelegation** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
  - **/usr/share/doc/ansible -freeipa/** 디렉터리에 있는 **README- delegation.md** 파일. 이 파일에는 **ipadelegation** 변수 정의도 포함되어 있습니다.
  - **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation** 디렉토리.

10.4. ANSIBLE을 사용하여 위임 규칙이 없는지 확인합니다.

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

사전 요구 사항

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

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/delegation/ 디렉토리에 있는 delegation-absent.yml 파일의 복사본을 만듭니다.

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

3. 편집 할 delegation-absent-copy.yml Ansible 플레이북 파일을 엽니다.
4. ipadelegation 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
   - name 변수를 위임 규칙의 이름으로 설정합니다.

- state 변수를 absent 로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

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

5. 파일을 저장합니다.

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

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

추가 리소스

- 위임 규칙 개념에 대한 자세한 내용은 위임 규칙을 참조하십시오.

- ipadelegation 모듈을 사용하는 샘플 Ansible Playbook은 다음을 참조하십시오.

  - /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README- delegation.md 파일. 이 파일에는 ipadelegation 변수 정의도 포함되어 있습니다.

  - /usr/share/doc/ansible-freeipa/playbooks/ipadelegation 디렉토리.

10.5. 위임 규칙에 특정 속성이 있는지 확인하려면 ANSIBLE을 사용합니다.

다음 절차에서는 Ansible 플레이북을 사용하여 위임 규칙에 특정 설정이 있는지 확인하는 방법을 설명합니다. 이 플레이북을 사용하여 이전에 만든 위임 역할을 수정할 수 있습니다. 이 예제에서는 기본 관리

자 속성 위임 규칙에 **departmentnumber** 멤버 속성만 있는지 확인합니다.

사전 요구 사항

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

절차

1. ~/***MyPlaybooks***/ 디렉터리로 이동합니다.

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

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

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

3. 편집 할 **delegation-member-present-copy.yml Ansible** 플레이북 파일을 엽니다.

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

   - **ipaadmin_password** 변수를 IdM 관리자의 암호로 설정합니다.
   - **name** 변수를 위임 규칙의 이름으로 설정하여 수정합니다.
   - 특성 변수를 **departmentnumber** 로 설정합니다.
   - **action** 변수를 **member** 로 설정합니다.

   이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

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

5. 파일을 저장합니다.

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

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

추가 리소스

- 위임 규칙 개념에 대한 자세한 내용은 위임 규칙을 참조하십시오.

- ipadelegation 모듈을 사용하는 샘플 Ansible Playbook은 다음을 참조하십시오.
  - /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README- delegation.md 파일. 이 파일에는 ipadelegation 변수 정의도 포함되어 있습니다.
  - /usr/share/doc/ansible-freeipa/playbooks/ipadelegation 디렉토리.

10.6. 위임 규칙에 특정 속성이 없는지 확인하려면 ANSIBLE을 사용합니다.

다음 절차에서는 Ansible 플레이북을 사용하여 위임 규칙에 특정 설정이 없는지 확인하는 방법을 설명합니다. 이 플레이북을 사용하여 위임 역할이 바람직하지 않은 액세스 권한을 부여하지 않도록 할 수 있습니다. 이 예제에서는 기본 관리자 속성 위임 규칙에 employeenumber 및 employee type 멤버 속성이 없는지 확인합니다.

사전 요구 사항

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

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/delegation/ 디렉토리에 있는 delegation-member-absent.yml 파일의 복사본을 만듭니다.

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

3. 편집 할 delegation-member-absent-copy.yml Ansible 플레이북 파일을 엽니다.

4. ipadelegation 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- name 변수를 위임 규칙의 이름으로 설정하여 수정합니다.
- 특성 변수를 employeenumber 및 employee type 으로 설정합니다.
- action 변수를 member 로 설정합니다.
- state 변수를 absent 로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
  - name: Ensure delegation "basic manager attributes" member attributes employeenumber and employeetype are absent
    ipadelegation:
      ipaadmin_password: Secret123
```

```
    name: "basic manager attributes"
    attribute:
    - employeenumber
    - employeetype
    action: member
    state: absent
```

5. 파일을 저장합니다.

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

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

추가 리소스

- 위임 규칙 개념에 대한 자세한 내용은 위임 규칙을 참조하십시오.
- ipadelegation 모듈을 사용하는 샘플 Ansible Playbook은 다음을 참조하십시오.
  - /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README- delegation.md 파일. 이 파일에는 ipadelegation 변수 정의도 포함되어 있습니다.
  - /usr/share/doc/ansible-freeipa/playbooks/ipadelegation 디렉토리.

11장. ANSIBLE 플레이북을 사용하여 IDM에서 역할 기반 액세스 제어 관리

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

- 권한은 사용자 추가 또는 삭제, 그룹 수정, 읽기 액세스 활성화 등의 특정 작업을 수행할 수 있는 권한을 부여합니다.
- 권한은 권한을 결합합니다(예: 새 사용자를 추가하는 데 필요한 모든 권한).
- 역할은 사용자, 사용자 그룹, 호스트 또는 호스트 그룹에 일련의 권한을 부여합니다.

특히 대기업에서는 RBAC를 사용하면 개별 책임을 수행하는 관리자의 계층적 시스템을 만들 수 있습니다.

이 장에서는 Ansible 플레이북을 사용하여 RBAC를 관리할 때 수행되는 다음 작업을 설명합니다.

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

- Ansible을 사용하여 사용자 그룹이 IdM RBAC 역할에 할당되었는지 확인합니다.
- Ansible을 사용하여 특정 사용자가 IdM RBAC 역할에 할당되지 않도록 합니다.
- 서비스가 IdM RBAC 역할의 멤버인지 확인하려면 Ansible을 사용합니다.
- 호스트가 IdM RBAC 역할의 멤버인지 확인하려면 Ansible을 사용합니다.
- Ansible을 사용하여 호스트 그룹이 IdM RBAC 역할의 멤버인지 확인합니다.

11.1. IDM의 권한

권한은 역할 기반 액세스 제어의 최하위 단위이며, 해당 작업이 적용되는 LDAP 항목과 함께 작업을 정의합니다. 구성 요소와 비교했을 때 필요한 수의 권한에 권한을 할당할 수 있습니다.
하나 이상의 권한은 다음을 허용하는 작업을 정의합니다.

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

이러한 작업은 세 개의 기본 대상에 적용됩니다 :

- **subtree:** 도메인 이름(DN), 이 **DN** 아래의 하위 트리
- 대상 필터**: LDAP** 필터
- 대상**:** 항목을 지정하는 데 사용할 수 있는 와일드카드가 있는 **DN**

또한 다음과 같은 편의성 옵션은 해당 특성을 설정합니다.

- **type:** 유형의 개체 (사용자, 그룹 등)는 **subtree** 및 **target filter**를 설정합니다.
- **memberOf:** 그룹의 구성원; 대상 필터를설정합니다.
- **targetGroup:** 특정 그룹을 수정할 액세스 권한을 부여합니다(예**:** 그룹 멤버십을 관리할 수 있는 권한 부여**).** 대상을설정합니다.

IdM 권한을 사용하면 어떤 사용자가 어떤 오브젝트에 대한 액세스 권한이 있는지 그리고 이러한 오브젝트의 속성까지 제어할 수 있습니다. **IdM**을 사용하면 개별 속성을 허용 또는 차단하거나 사용자, 그룹 또는 **sudo**와 같은 특정 **IdM** 기능의 전체 가시성을 모든 익명 사용자, 모든 인증된 사용자 또는 특정 권한 있는 사용자 그룹에 변경할 수 있습니다.
예를 들어 이 접근 방식의 권한의 유연성은 사용자 또는 그룹에 대한 액세스만 제한하려는 관리자에게 유용합니다. 사용자 또는 그룹이 액세스해야 하는 특정 섹션으로만 액세스하고 다른 섹션에 완전히 숨길 수 있도록 합니다.

참고

권한에는 다른 권한이 포함될 수 없습니다.

11.2. 기본 관리 권한

관리 권한은 기본적으로 **IdM**과 함께 제공되는 권한입니다. 다음과 같은 차이점을 사용하여 사용자가 생성한 다른 권한처럼 작동합니다.

- 해당 파일을 삭제하거나 이름, 위치, 대상 특성을 수정할 수 없습니다.
- 여기에는 세 가지 속성 세트가 있습니다.
  - 기본 속성은 **IdM**에서 관리하므로 사용자가 수정할 수 없습니다.
  - 사용자가 추가한 속성인 포함된 속성
  - 사용자가 제거하는 속성인 제외된 속성

관리되는 권한은 기본 및 포함된 특성 세트에 표시되지만 제외된 속성 세트에는 표시되지 않는 모든 속성에 적용됩니다.

참고

관리 권한을 삭제할 수는 없지만 바인딩 유형을 권한으로 설정하고 모든 권한에서 관리 권한을 제거하면 효과적으로 비활성화됩니다.

관리되는 모든 권한의 이름은 시스템:(예: 시스템): **Sudo rule** 또는 시스템을 추가합니다: **Services** 를 수정합니다. 이전 버전의 **IdM**에서는 기본 권한에 다른 체계를 사용했습니다. 예를 들어 사용자는 삭제할 수 없으며 권한에만 할당할 수 있었습니다. 이러한 기본 권한은 대부분 관리 권한으로 설정되었지만 다음 권한은 여전히 이전 체계를 사용합니다.

- **Automember Rebuild Membership** 작업 추가
- 구성 하위 항목 추가
- 복제 계약 추가
- 인증서 제거 보류

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

- IPA 설정 쓰기

참고

명령줄에서 관리 권한을 수정하려고 하면 시스템에서 수정할 수 없는 특성을 변경할 수 없으므로 명령이 실패합니다. 웹 UI에서 관리 권한을 수정하려고 하면 수정할 수 없는 속성이 비활성화됩니다.

11.3. IDM의 권한

권한은 역할에 적용할 수 있는 권한 그룹입니다.
권한은 단일 작업을 수행할 수 있는 권한을 제공하지만 성공하려면 여러 권한이 필요한 특정 IdM 작업이 있습니다. 따라서 권한은 특정 작업을 수행하는 데 필요한 다양한 권한을 결합합니다.
예를 들어 새 IdM 사용자의 계정을 설정하려면 다음과 같은 권한이 필요합니다.

- 새 사용자 항목 생성
- 사용자 암호 재설정
- 새 사용자를 기본 IPA 사용자 그룹에 추가

이러한 세 가지 하위 수준 작업을 이름이 지정된 사용자 지정 권한의 형태로 더 높은 수준의 작업(예: 사용자 추가) 에 결합하면 시스템 관리자가 역할을 쉽게 관리할 수 있습니다. IdM에는 이미 몇 가지 기본 권한이 포함되어 있습니다. 사용자 및 사용자 그룹 외에도 호스트 및 호스트 그룹은 네트워크 서비스에도 권한이 할당됩니다. 이 방법을 사용하면 특정 네트워크 서비스를 사용하는 호스트 집합의 사용자 집합에 의한 작업을 세부적으로 제어할 수 있습니다.

참고

권한에는 다른 권한이 없을 수 있습니다.

11.4. IDM의 역할

역할은 역할에 지정된 사용자가 보유하는 권한 목록입니다.
실제로 권한은 지정된 하위 수준 작업을 수행할 수 있는 기능을 부여합니다(예: 사용자 항목 생성, 그룹에 항목 추가) 권한은 더 높은 수준의 작업(예: 지정된 그룹에 새 사용자 만들기)에 필요한 하나 이상의 권한

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

중요

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

참고

역할에 다른 역할은 포함할 수 없습니다.

11.5. ID 관리에 사전 정의된 역할

Red Hat Identity Management는 다음과 같은 일련의 사전 정의 역할을 제공합니다.

표 11.1. ID 관리에서 사전 정의된 역할

Role	권한	설명
헬프데스크	사용자 및 암호 재설정, 그룹 멤버십 수정	간단한 사용자 관리 작업 수행 담당
IT 보안 전문가	Netgroups 관리자, HBAC 관리자, Sudo 관리자	호스트 기반 액세스 제어와 같은 보안 정책 관리, sudo 규칙
IT 전문가	호스트 관리자, 호스트 그룹 관리자, 서비스 관리자, 자동 마운트 관리자	호스트 관리 담당
보안 설계자	위임 관리자, 복제 관리자, 쓰기 IPA 구성, 암호 정책 관리자	Identity Management 환경 관리, 신뢰 생성, 복제 계약 생성
사용자 관리자	사용자 관리자, 그룹 관리자, 단계 사용자 관리자	사용자 및 그룹 생성 담당

11.6. ANSIBLE을 사용하여 권한이 있는 IDM RBAC 역할이 있는지 확인합니다.

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

다음 절차에서는 Ansible 플레이북을 사용하여 새 IdM 사용자 지정 역할에 대한 권한을 정의하고 해당

역할이 있는지 확인하는 방법을 설명합니다. 이 예제에서 새 **user_and_host_administrator** 역할에는 기본적으로 **IdM**에 있는 다음 권한의 고유한 조합이 포함되어 있습니다.

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

사전 요구 사항

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

절차

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

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

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

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

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

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

- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 새 역할의 이름으로 설정합니다.
- 권한 목록을 새 역할에 포함하려는 **IdM** 권한의 이름으로 설정합니다.
- 또는 **user** 변수를 새 역할을 부여하려는 사용자 이름으로 설정합니다.
- 필요한 경우 **group** 변수를 새 역할을 부여할 그룹 이름으로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

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

5. 파일을 저장합니다.

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

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

추가 리소스

- Ansible Vault를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오.
- IdM의 역할 개념에 대한 자세한 내용은 IdM의 역할을 참조하십시오.
- iparole 모듈을 사용하는 샘플 Ansible Playbook은 다음을 참조하십시오.
  - /usr/share/doc/ansible -freeipa/ 디렉터리에서 사용할 수 있는 README- role 파일. 이 파일에는 iparole 변수 정의도 포함되어 있습니다.
  - /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉토리.

11.7. ANSIBLE을 사용하여 IDM RBAC 역할이 없는지 확인합니다.

IdM(Identity Management)에서 역할 기반 액세스 제어(RBAC)를 관리하는 시스템 관리자는 실수로 사용자에게 할당하지 않도록 사용되지 않는 역할이 없는지 확인해야 합니다.

다음 절차에서는 Ansible 플레이북을 사용하여 역할이 없는지 확인하는 방법을 설명합니다. 아래 예제에서는 사용자 지정 user_and_host_administrator 역할이 IdM에 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

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

파일을 생성했습니다.

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

절차

1. ~/*<MyPlaybooks>*/ 디렉터리로 이동합니다.

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

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

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

3. 편집할 role-is-absent-copy.yml Ansible 플레이북 파일을 엽니다.

4. iparole 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

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

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
  - iparole:
```

```
    ipaadmin_password: Secret123
    name: user_and_host_administrator
    state: absent
```

5. 파일을 저장합니다.

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

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

추가 리소스

- Ansible Vault를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오.
- IdM의 역할 개념에 대한 자세한 내용은 IdM의 역할을 참조하십시오.
- iparole 모듈을 사용하는 샘플 Ansible Playbook은 다음을 참조하십시오.
  - /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README- role 마크다운 파일. 이 파일에는 iparole 변수 정의도 포함되어 있습니다.
  - /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉토리.

11.8. ANSIBLE을 사용하여 사용자 그룹이 IDM RBAC 역할에 할당되었는지 확인합니다.

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

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

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.

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

절차

1. ~/*<MyPlaybooks>*/ 디렉터리로 이동합니다.

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

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

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

3. 편집할 role-member-group-present-copy.yml Ansible 플레이북 파일을 엽니다.
4. iparole 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
   - name 변수를 할당할 역할의 이름으로 설정합니다.
   - 그룹 변수를 그룹 이름으로 설정합니다.
   - action 변수를 member 로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

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

5. 파일을 저장합니다.

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

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

추가 리소스

- Ansible Vault를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오.
- IdM의 역할 개념에 대한 자세한 내용은 IdM의 역할을 참조하십시오.
- iparole 모듈을 사용하는 샘플 Ansible 플레이북에 대해서는 /usr/share/doc/ansible-freeipa/ 디렉터리에 있는 README- role Markdown 파일을 참조하십시오. 파일에는 iparole 변수의 정의도 포함되어 있습니다.
- iparole 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리를 참조하십시오.

11.9. ANSIBLE을 사용하여 특정 사용자가 IDM RBAC 역할에 할당되지 않도록 합니다.

IdM(Identity Management)에서 역할 기반 액세스 제어(RBAC)를 관리하는 시스템 관리자는 특정 사용자에게 RBAC 역할이 할당되지 않도록 할 수 있습니다(예: 회사 내 다른 위치로 이동).

다음 절차에서는 **Ansible** 플레이북을 사용하여 **user_01** 및 **user_ 02** 라는 사용자가 **helpdesk** 역할에 할당되지 않도록 하는 방법을 설명합니다.

사전 요구 사항

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

절차

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

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

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

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

3. 편집할 **role-member-user-absent-copy.yml Ansible** 플레이북 파일을 엽니다.
4. **iparole** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
   - **name** 변수를 할당할 역할의 이름으로 설정합니다.

- 사용자 목록을 사용자 이름으로 설정합니다.
- action 변수를 member 로 설정합니다.
- state 변수를 absent 로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

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

5. 파일을 저장합니다.

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

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

추가 리소스

- Ansible Vault를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오.
- IdM의 역할 개념에 대한 자세한 내용은 IdM의 역할을 참조하십시오.
-

iparole 모듈을 사용하는 샘플 Ansible 플레이북에 대해서는 /usr/share/doc/ansible-freeipa/ 디렉터리에 있는 README- role Markdown 파일을 참조하십시오. 파일에는 iparole 변수의 정의도 포함되어 있습니다.

- iparole 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리를 참조하십시오.

11.10. 서비스가 IDM RBAC 역할의 멤버인지 확인하려면 ANSIBLE을 사용합니다.

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

사전 요구 사항

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

절차

1. ~/*<MyPlaybooks>*/ 디렉터리로 이동합니다.

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

2.

/usr/share/doc/ansible -freeipa/playbooks/role/ 디렉터리에 있는 role-member-service-present.yml 파일의 복사본을 만듭니다.

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

3. role-member-service-present-copy.yml Ansible 플레이북 파일을 편집하여 편집합니다.

4. iparole 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- name 변수를 할당할 역할의 이름으로 설정합니다.
- 서비스 목록을 서비스 이름으로 설정합니다.
- action 변수를 member 로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

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

5. 파일을 저장합니다.

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

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

추가 리소스

- Ansible Vault를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오.
- IdM의 역할 개념에 대한 자세한 내용은 IdM의 역할을 참조하십시오.
- iparole 모듈을 사용하는 샘플 Ansible 플레이북에 대해서는 /usr/share/doc/ansible-freeipa/ 디렉터리에 있는 README- role Markdown 파일을 참조하십시오. 파일에는 iparole 변수의 정의도 포함되어 있습니다.
- iparole 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리를 참조하십시오.

11.11. 호스트가 IDM RBAC 역할의 멤버인지 확인하려면 ANSIBLE을 사용합니다.

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

사전 요구 사항

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

- web_administrator 역할은 IdM에 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 복사본을 만듭니다.

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

3. role-member-host-present-copy.yml Ansible 플레이북 파일을 편집하여 편집합니다.

4. iparole 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
   - name 변수를 할당할 역할의 이름으로 설정합니다.
   - 호스트 목록을 호스트 이름으로 설정합니다.

   이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
  - iparole:
```

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

5. 파일을 저장합니다.

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

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

추가 리소스

- Ansible Vault를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 Ansible Vault를 사용하여 콘텐츠 암호화를 참조하십시오.
- IdM의 역할 개념에 대한 자세한 내용은 IdM의 역할을 참조하십시오.
- iparole 모듈을 사용하는 샘플 Ansible 플레이북에 대해서는 /usr/share/doc/ansible-freeipa/ 디렉터리에 있는 README- role Markdown 파일을 참조하십시오. 파일에는 iparole 변수의 정의도 포함되어 있습니다.
- iparole 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리를 참조하십시오.

11.12. ANSIBLE을 사용하여 호스트 그룹이 IDM RBAC 역할의 멤버인지 확인합니다.

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

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.

- Ansible 제어 노드에 ansible-freeipa 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 정규화된 도메인 이름(FQDN)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.
- Ansible 인벤토리 파일은 ~/*<MyPlaybooks>*/ 디렉터리에 있습니다.
- web_administrator 역할은 IdM에 있습니다.
- web_servers 호스트 그룹은 IdM에 있습니다.

절차

1. ~/*<MyPlaybooks>*/ 디렉터리로 이동합니다.

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

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

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

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

4. iparole 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
   - name 변수를 할당할 역할의 이름으로 설정합니다.
   - hostgroup 목록을 호스트 그룹의 이름으로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - iparole:
      ipaadmin_password: Secret123
      name: web_administrator
      hostgroup:
      - web_servers
      action: member
```

5. 파일을 저장합니다.

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

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

추가 리소스

- **Ansible Vault**를 사용하여 별도의 파일에 암호를 저장하거나 플레이북 파일에서 변수로 암호화하는 방법에 대한 자세한 내용은 **Ansible Vault**를 사용하여 콘텐츠 암호화를 참조하십시오.

- **IdM**의 역할 개념에 대한 자세한 내용은 **IdM**의 역할을 참조하십시오.

- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북에 대해서는 **/usr/share/doc/ansible-freeipa/** 디렉터리에 있는 **README- role Markdown** 파일을 참조하십시오. 파일에는 **iparole** 변수의 정의도 포함되어 있습니다.

- **iparole** 모듈을 사용하는 샘플 **Ansible** 플레이북은 **/usr/share/doc/ansible-freeipa/playbooks/iparole** 디렉터리를 참조하십시오.

12장. ANSIBLE 플레이북을 사용하여 RBAC 권한 관리

RBAC(역할 기반 액세스 제어)는 역할, 권한 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. 특히 대기업에서는 RBAC를 사용하면 개별 책임을 수행하는 관리자의 계층적 시스템을 만들 수 있습니다.

이 장에서는 IdM(Identity Management)에서 RBAC 권한을 관리하기 위해 Ansible 플레이북을 사용하는 작업에 대해 설명합니다.

- Ansible을 사용하여 사용자 지정 RBAC 권한이 있는지 확인합니다.
- Ansible을 사용하여 멤버 권한이 사용자 지정 IdM RBAC 권한에 있는지 확인
- Ansible을 사용하여 IdM RBAC 권한에 권한이 포함되지 않도록 합니다.
- Ansible을 사용하여 사용자 지정 IdM RBAC 권한 이름 변경
- Ansible을 사용하여 IdM RBAC 권한이 없는지 확인합니다.

사전 요구 사항

- RBAC의 개념과 원칙을 이해합니다.

12.1. ANSIBLE을 사용하여 사용자 지정 IDM RBAC 권한이 있는지 확인

IdM(Identity Management) 역할 기반 액세스 제어(RBAC)에서 사용자 지정 권한을 완전히 작동하려면 단계를 진행해야 합니다.

1. 권한이 연결되지 않고 권한을 생성합니다.
2. 선택한 권한을 권한에 추가합니다.

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

사전 요구 사항

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

절차

1. ~/***MyPlaybooks***/ 디렉터리로 이동합니다.

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

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

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

3. 편집을 위해 **privilege-present-copy.yml Ansible** 플레이북 파일을 엽니다.
4. **ipaprivilege** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
   - **name** 변수를 새 권한인 **full_host_administration** 의 이름으로 설정합니다.

- 선택적으로 **description** 변수를 사용하여 권한을 설명합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

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

5. 파일을 저장합니다.

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

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

12.2. ANSIBLE을 사용하여 멤버 권한이 사용자 지정 IDM RBAC 권한에 있는지 확인

IdM(Identity Management) 역할 기반 액세스 제어(**RBAC**)에서 사용자 지정 권한을 완전히 작동하려면 단계를 진행해야 합니다.

1. 권한이 연결되지 않고 권한을 생성합니다.

2. 선택한 권한을 권한에 추가합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 이전 단계에서 만든 권한에 권한을 추가하는 방법을 설명합니다. 이 예제에서는 호스트 관리와 관련된 모든 **IdM** 권한을 **full_host_administration** 이라는 권한에 추가하는 방법을 설명합니다. 기본적으로 권한은 **Host Enrollment**(호스트 등록), **HostAdministrators**(호스트 관리자) 및 **Host Group Administrator** (호스트 그룹 관리자) 권한 사이에 배포됩니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 ansible-freeipa 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 정규화된 도메인 이름(FQDN)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.
- Ansible 인벤토리 파일은 ~/*MyPlaybooks*/ 디렉터리에 있습니다.
- *full_host_administration* 권한이 있습니다. Ansible을 사용하여 권한을 생성하는 방법에 대한 자세한 내용은 Ansible을 사용하여 사용자 지정 IdM RBAC 권한이 있는지를 참조하십시오.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

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

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

3. 편집을 위해 privilege-member-present-copy.yml Ansible 플레이북 파일을 엽니다.
4. ipaprivilege 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - 사용 사례에 대응하도록 작업 이름을 조정합니다.
   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.

- **name** 변수를 권한 이름으로 설정합니다.
- 권한 목록을 권한에 포함할 권한의 이름으로 설정합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - name: Ensure that permissions are present for the "full_host_administration"
privilege
    ipaprivilege:
      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 파일 및 인벤토리 파일을 지정하는 Ansible 플레이북을 실행합니다.

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

12.3. ANSIBLE을 사용하여 IDM RBAC 권한에 권한이 포함되지 않도록 합니다.

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어를 사용자 지정할 수 있습니다.

다음 절차에서는 Ansible 플레이북을 사용하여 권한에서 권한을 제거하는 방법을 설명합니다. 이 예제에서는 관리자가 보안 위험이라고 간주하기 때문에 기본 Certificate Administrators 권한에서 CA ACL 권한을 무시하는 요청 인증서를 제거하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

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

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

3. 편집을 위해 **privilege-member-absent-copy.yml Ansible** 플레이북 파일을 엽니다.

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

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한 이름으로 설정합니다.
- 권한 목록을 권한에서 제거할 권한의 이름으로 설정합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
- **state** 변수가 **absent** 로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - name: Ensure that the "Request Certificate ignoring CA ACLs" permission is absent
from the "Certificate Administrators" privilege
    ipaprivilege:
      ipaadmin_password: Secret123
      name: Certificate Administrators
      permission:
      - "Request Certificate ignoring CA ACLs"
      action: member
      state: absent
```

5. 파일을 저장합니다.

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

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

12.4. ANSIBLE을 사용하여 사용자 지정 IDM RBAC 권한 이름 변경

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어를 사용자 지정할 수 있습니다.

다음 절차에서는 권한 이름을 변경하는 방법을 설명합니다. 예를 들어 몇 가지 권한을 제거했기 때문입니다. 결과적으로 권한 이름은 더 이상 정확하지 않습니다. 이 예제에서 관리자는 full_host_administration 권한의 이름을 limited_ host_administration 으로 변경합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 ansible-freeipa 패키지를 설치했습니다.
- 구성을 수행하려는 IdM 서버의 정규화된 도메인 이름(FQDN)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.
- Ansible 인벤토리 파일은 ~/*MyPlaybooks*/ 디렉터리에 있습니다.
- *full_host_administration* 권한이 있습니다. 권한을 추가하는 방법에 대한 자세한 내용은 Ansible을 사용하여 사용자 지정 IdM RBAC 권한이 있는지를 참조하십시오.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

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

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

3. 편집할 rename-privilege.yml Ansible 플레이북 파일을 엽니다.

4. ipaprivilege 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- name 변수를 권한의 현재 이름으로 설정합니다.
- rename 변수를 추가하고 권한의 새 이름으로 설정합니다.
- state 변수를 추가하고 이름을 바꿉니다.

5. 플레이북 자체의 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
---
- name: Rename a privilege
  hosts: ipaserver
  become: true
```

6. 플레이북의 작업의 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
[...]
tasks:
- name: Ensure the full_host_administration privilege is renamed to
limited_host_administration
  ipaprivilege:
  [...]
```

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

-

```
---
- name: Rename a privilege
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure the full_host_administration privilege is renamed to
limited_host_administration
    ipaprivilege:
      ipaadmin_password: Secret123
      name: full_host_administration
      rename: limited_host_administration
      state: renamed
```

7. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i inventory rename-privilege.yml
```

12.5. ANSIBLE을 사용하여 IDM RBAC 권한이 없는지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어를 사용자 지정할 수 있습니다. 다음 절차에서는 Ansible 플레이북을 사용하여 RBAC 권한이 없는지 확인하는 방법을 설명합니다. 이 예제에서는 CA 관리자 권한이 없는지 확인하는 방법을 설명합니다. 이 절차의 결과로 관리자 관리자는 IdM에서 인증 기관을 관리할 수 있는 유일한 사용자가 됩니다.

사전 요구 사항

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

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/privilege/ 디렉터리에 있는 privilege-absent.yml 파일의 복사본을 만듭니다.

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

3. 편집을 위해 privilege-absent-copy.yml Ansible 플레이북 파일을 엽니다.

4. ipaprivilege 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- name 변수를 제거할 권한의 이름으로 설정합니다.
- state 변수가 absent 로 설정되어 있는지 확인합니다.

5. 플레이북의 작업의 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
[...]
tasks:
- name: Ensure privilege "CA administrator" is absent
  ipaprivilege:
  [...]
```

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
  - name: Ensure privilege "CA administrator" is absent
```

```
  ipaprivilege:
    ipaadmin_password: Secret123
    name: CA administrator
    state: absent
```

6. 파일을 저장합니다.

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

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

추가 리소스

- IdM RBAC의 권한 개념에 대한 자세한 내용은 IdM의 권한을 참조하십시오.
- IdM RBAC의 권한 개념에 대한 자세한 내용은 Permissions in IdM을 참조하십시오.
- ipaprivilege 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README- privilege 파일을 참조하십시오. 파일에는 ipaprivilege 변수 정의도 포함되어 있습니다.
- ipaprivilege 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible-freeipa/playbooks/ipaprivilege 디렉터리를 참조하십시오.

13장. ANSIBLE 플레이북을 사용하여 IDM에서 RBAC 권한 관리

RBAC(역할 기반 액세스 제어)는 역할, 권한 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. 특히 대기업에서는 RBAC를 사용하면 개별 책임을 수행하는 관리자의 계층적 시스템을 만들 수 있습니다.

이 장에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에서 RBAC 권한을 관리할 때 수행되는 다음 작업을 설명합니다.

- Ansible을 사용하여 RBAC 권한이 있는지 확인합니다.
- Ansible을 사용하여 속성이 있는 RBAC 권한이 있는지 확인합니다.
- Ansible을 사용하여 RBAC 권한이 없는지 확인합니다.
- Ansible을 사용하여 속성이 IdM RBAC 권한의 멤버인지 확인합니다.
- Ansible을 사용하여 속성이 IdM RBAC 권한의 멤버가 아닌지 확인합니다.
- Ansible을 사용하여 IdM RBAC 권한 이름 변경

사전 요구 사항

- RBAC의 개념과 원칙을 이해합니다.

13.1. ANSIBLE을 사용하여 RBAC 권한이 있는지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어(RBAC)를 사용자 지정할 수 있습니다.

다음 절차에서는 Ansible 플레이북을 사용하여 권한에 권한을 추가할 수 있도록 IdM에 권한을 제공하는 방법을 설명합니다. 예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한은 호스트에만 적용할 수 있습니다.
- 권한이 포함된 권한이 부여된 사용자는 항목에서 다음과 같은 가능한 작업을 모두 수행할 수 있습니다.
  - 쓰기
  - 읽기
  - 검색
  - 비교
  - **add**
  - **delete**

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 ~/***MyPlaybooks***/ 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 만들고 구성 했다고 가정합니다.

절차

1. ~/***MyPlaybooks***/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/permission/ 디렉터리에 있는 permission-present.yml 파일의 복사본을 만듭니다.

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

3. 편집할 permission-present-copy.yml Ansible 플레이북 파일을 엽니다.

4. ipapermission 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- name 변수를 권한의 이름으로 설정합니다.
- object_type 변수를 host 로 설정합니다.
- 올바른 변수를 all 로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
  - name: Ensure that the "MyPermission" permission is present
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      object_type: host
      right: all
```

5. 파일을 저장합니다.

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

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

13.2. ANSIBLE을 사용하여 속성이 있는 RBAC 권한이 있는지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어(RBAC)를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 권한에 권한을 추가할 수 있도록 IdM에 권한을 제공하는 방법을 설명합니다. 예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한은 호스트를 추가하는 데만 사용할 수 있습니다.
- 권한이 포함된 권한이 부여된 사용자는 호스트 항목에서 다음과 같은 모든 작업을 수행할 수 있습니다.
  - 쓰기
  - 읽기
  - 검색
  - 비교
  - **add**

- **delete**

- **MyPermission** 권한이 포함된 권한이 부여된 사용자가 생성한 호스트 항목은 **description** 값을 가질 수 있습니다.

참고

권한을 만들거나 수정할 때 지정할 수 있는 속성 유형은 **IdM LDAP** 스키마에 의해 제한되지 않습니다. 그러나 **object_type**이 호스트 일 경우 **attrs:car_ licence** 를 지정하면 나중에 **ipa**가 됩니다. **ERROR:** 속성 **"car-license" not allowed error message when you try to a specificcar licence value to a host.**

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.

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

- 이 예제에서는 **~/*MyPlaybooks*/** 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 만들고 구성 했다고 가정합니다.

절차

1. **~/*MyPlaybooks*/** 디렉터리로 이동합니다.

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

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

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-present.yml
permission-present-with-attribute.yml
```

3. 편집을 위해 **permission-present-with-attribute.yml Ansible** 플레이북 파일을 엽니다.

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

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한의 이름으로 설정합니다.
- **object_type** 변수를 **host** 로 설정합니다.
- 올바른 변수를 **all** 로 설정합니다.
- **attrs** 변수를 **description** 으로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - name: Ensure that the "MyPermission" permission is present with an attribute
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      object_type: host
      right: all
      attrs: description
```

5. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i inventory permission-present-with-attribute.yml
```

추가 리소스

- IdM 스키마에 대한 자세한 내용은 RHEL 7의 *Linux Domain Identity, Authentication and Policy Guide*의 *사용자 및 그룹 스키마* 를 참조하십시오.

13.3. ANSIBLE을 사용하여 RBAC 권한이 없는지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어(RBAC)를 사용자 지정할 수 있습니다.

다음 절차에서는 권한에 추가할 수 없도록 Ansible 플레이북을 사용하여 IdM에 권한이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 ansible-freeipa 패키지를 설치했습니다.
- 이 예제에서는 ~/*MyPlaybooks*/ 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 만들고 구성 했다고 가정합니다.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/permission/ 디렉터리에 있는 permission-absent.yml 파일의 복사본을 만듭니다.

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

3. 편집을 위해 permission-absent-copy.yml Ansible 플레이북 파일을 엽니다.

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

   - 사용 사례에 대응하도록 작업 이름을 조정합니다.
   - **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
   - **name** 변수를 권한의 이름으로 설정합니다.

   이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission absent example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure that the "MyPermission" permission is absent
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      state: absent
```

5. 파일을 저장합니다.

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

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

13.4. ANSIBLE을 사용하여 속성이 IDM RBAC 권한의 멤버인지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어**(RBAC)**를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 속성이 **IdM**에서 **RBAC** 권한의 멤버인지 확인하는 방법을 설명합니다. 결과적으로 권한이 있는 사용자는 특성이 있는 항목을 생성할 수 있습니다.

이 예제에서는 **MyPermission** 권한을 포함하는 권한이 있는 사용자가 생성한 호스트 항목에 **gecos** 및 **description** 값을 가질 수 있도록 하는 방법을 설명합니다.

참고

권한을 만들거나 수정할 때 지정할 수 있는 속성 유형은 **IdM LDAP** 스키마에 의해 제한되지 않습니다. 그러나 **object_type**이 호스트 일 경우 **attrs:car_ licence** 를 지정하면 나중에 **ipa**가 됩니다. **ERROR: 속성 "car-license" not allowed error message when you try to a specificcar licence value to a host.**

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 ~/***MyPlaybooks***/ 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 만들고 구성 했다고 가정합니다.
- **MyPermission** 권한이 있습니다.

절차

1. ~/***MyPlaybooks***/ 디렉터리로 이동합니다.

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

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

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

3. 편집을 위해 **permission-member-present-copy.yml Ansible** 플레이북 파일을 엽니다.
4. **ipapermission** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- name 변수를 권한의 이름으로 설정합니다.
- attrs 목록을 description 및 gecos 변수로 설정합니다.
- 작업 변수가 member 로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Permission member present example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure that the "gecos" and "description" attributes are present in
"MyPermission"
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      attrs:
      - description
      - gecos
      action: member
```

5. 파일을 저장합니다.

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

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

13.5. ANSIBLE을 사용하여 속성이 IDM RBAC 권한의 멤버가 아닌지 확인합니다.

IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어(RBAC)를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 속성이 **IdM**에서 **RBAC** 권한의 멤버가 아닌지 확인하는 방법을 설명합니다. 결과적으로 권한이 있는 사용자가 **IdM LDAP**에 항목을 생성하면 해당 항목에 특성과 연결된 값이 있을 수 없습니다.

예제에서는 다음 대상 상태를 확인하는 방법을 설명합니다.

- **MyPermission** 권한이 있습니다.
- **MyPermission** 권한이 포함된 권한이 있는 사용자가 생성한 호스트 항목은 **description** 속성을 가질 수 없습니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 ansible-freeipa 패키지를 설치했습니다.
- 이 예제에서는 ~/*MyPlaybooks*/ 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 만들고 구성 했다고 가정합니다.
- **MyPermission** 권한이 있습니다.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/permission/ 디렉터리에 있는 **permission-member- absent.yml** 파일의 복사본을 만듭니다.

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

3. 편집을 위해 **permission-member-absent-copy.yml Ansible** 플레이북 파일을 엽니다.

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

- 사용 사례에 대응하도록 작업 이름을 조정합니다.
- **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
- **name** 변수를 권한의 이름으로 설정합니다.
- **attrs** 변수를 **description** 으로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.
- **state** 변수가 **absent**로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Permission absent example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure that an attribute is not a member of "MyPermission"
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      attrs: description
      action: member
      state: absent
```

5. 파일을 저장합니다.

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

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

13.6. ANSIBLE을 사용하여 IDM RBAC 권한 이름 변경

IdM(Identity Management)의 시스템 관리자는 **IdM** 역할 기반 액세스 제어를 사용자 지정할 수 있습니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 권한의 이름을 변경하는 방법을 설명합니다. 이 예제에서는 **MyPermission**의 이름을 **MyNewPermission** 으로 변경하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 제어 노드에 **ansible-freeipa** 패키지를 설치했습니다.
- 이 예제에서는 ~/***MyPlaybooks***/ 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 만들고 구성 했다고 가정합니다.
- **MyPermission** 은 **IdM**에 있습니다.
- **MyNewPermission** 은 **IdM**에 존재하지 않습니다.

절차

1. ~/***MyPlaybooks***/ 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible -freeipa/playbooks/permission/** 디렉터리에 있는 **permission-renamed.yml** 파일의 복사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-renamed.yml
permission-renamed-copy.yml
```

3. 편집을 위해 **permission-renamed-copy.yml Ansible** 플레이북 파일을 엽니다.

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

   - 사용 사례에 대응하도록 작업 이름을 조정합니다.
   - **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
   - **name** 변수를 권한의 이름으로 설정합니다.

   이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - name: Rename the "MyPermission" permission
    ipapermission:
      ipaadmin_password: Secret123
      name: MyPermission
      rename: MyNewPermission
      state: renamed
```

5. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i inventory permission-renamed-copy.yml
```

13.7. 추가 리소스

- IdM RBAC의 권한 개념에 대한 자세한 내용은 Permissions in IdM을 참조하십시오.

- IdM RBAC의 권한 개념에 대한 자세한 내용은 IdM의 권한을 참조하십시오.

- ipapermission 모듈을 사용하는 샘플 Ansible 플레이북에 대해서는 /usr/share/doc/ansible-freeipa/ 디렉터리에서 사용할 수 있는 README- permission 파일을 참조하십시오. 파일에는 ipapermission 변수 정의도 포함되어 있습니다.

- ipapermission 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible-freeipa/playbooks/ipapermission 디렉터리를 참조하십시오.

14장. ANSIBLE을 사용하여 IDM의 복제 토폴로지 관리

여러 IdM(Identity Management) 서버를 유지 관리하고 중복을 위해 서로 복제하여 서버 손실을 완화하거나 방지할 수 있습니다. 예를 들어, 한 서버가 실패하면 다른 서버는 계속 도메인에 서비스를 제공합니다. 나머지 서버 중 하나를 기반으로 새 복제본을 만들어 손실된 서버를 복구할 수도 있습니다.

IdM 서버에 저장된 데이터는 복제 계약에 따라 복제됩니다. 두 서버에 복제 계약이 구성된 경우 해당 데이터를 공유합니다. 복제된 데이터는 토폴로지 접미사 에 저장됩니다. 두 복제본에 접미사 간 복제 계약이 있는 경우 접미사는 토폴로지 세그먼트 를 형성합니다.

이 장에서는 Red Hat Ansible Engine 을 사용하여 IdM 복제 계약, 토폴로지 세그먼트 및 토폴로지 접미사를 관리하는 방법을 설명합니다. 장에는 다음 섹션이 포함되어 있습니다.

- Ansible을 사용하여 IdM에 복제 계약이 있는지 확인
- Ansible을 사용하여 여러 IdM 복제본 간에 복제 계약이 있는지 확인
- Ansible을 사용하여 두 복제본 간에 복제 계약이 있는지 확인
- Ansible을 사용하여 토폴로지 접미사가 IdM에 있는지 확인합니다.
- Ansible을 사용하여 IdM 복제본 다시 초기화
- Ansible을 사용하여 IdM에 복제 계약이 없음

14.1. ANSIBLE을 사용하여 IDM에 복제 계약이 있는지 확인

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약에 따라 복제됩니다. 두 서버에 복제 계약이 구성된 경우 데이터를 공유합니다. 복제 계약은 항상 양방향입니다. 데이터는 첫 번째 복제본에서 다른 복제본으로는 물론 다른 복제본으로 복제됩니다.

이 섹션에서는 Ansible 플레이북을 사용하여 server.idm.example.com과 replica.idm.example.com 간에 도메인 유형의 복제 계약이 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/topology/ 디렉터리에 있는 add-topology segment.yml Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegment.yml
add-topologysegment-copy.yml
```

3. add-topologysegment-copy.yml 파일을 열어 편집합니다.
4. ipatopologysegment 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.

- 추가하려는 세그먼트 유형에 따라 접미사 변수를 **domain** 또는 **ca** 로 설정합니다.
- 복제 계약의 왼쪽 노드가 되고자 하는 **IdM** 서버의 이름으로 왼쪽 변수를 설정합니다.
- 올바른 변수를 복제 계약의 올바른 노드로 설정하고자 하는 **IdM** 서버의 이름으로 설정합니다.
- **state** 변수가 **present**로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
- name: Add topology segment
    ipatopologysegment:
      ipaadmin_password: Secret123
      suffix: domain
      left: server.idm.example.com
      right: replica.idm.example.com
      state: present
```

5. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i inventory add-topologysegment-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 설명 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트 를 참조하십시오.
- **ipatopologysegment** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.

- /usr/share/doc/ansible -freeipa/ 디렉터리의 README- topology.md 파일. 이 파일에는 ipatopologysegment 변수 정의도 포함되어 있습니다.
- /usr/share/doc/ansible-freeipa/playbooks/topology 디렉토리.

14.2. ANSIBLE을 사용하여 여러 IDM 복제본 간에 복제 계약이 있는지 확인

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약에 따라 복제됩니다. 두 서버에 복제 계약이 구성된 경우 데이터를 공유합니다. 복제 계약은 항상 양방향입니다. 데이터는 첫 번째 복제본에서 다른 복제본으로는 물론 다른 복제본으로 복제됩니다.

이 섹션에서는 IdM의 여러 복제본 쌍 간에 복제 계약이 있는지 확인하는 방법에 대해 설명합니다.

사전 요구 사항

- 토폴로지에서 복제본 연결에 나열된 IdM 토폴로지 설계 권장 사항을 이해했는지 확인하십시오. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/planning_identity_management/index#connecting-the-replicas-in-a-topology_planning-the-replica-topology
- IdM 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - ansible-freeipa 패키지를 설치했습니다.
  - ~/*MyPlaybooks*/ 디렉터리에 이러한 옵션을 구성하는 IdM 서버의 정규화된 도메인 이름(FQDN)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/topology/ 디렉터리에 있는 add-topologysegments.yml Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegments.yml
add-topologysegments-copy.yml
```

3. add-topologysegments-copy.yml 파일을 열어 편집합니다.

4. vars 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- 모든 토폴로지 세그먼트에 대해 ipatopology_segments 섹션에 행을 추가하고 다음 변수를 설정합니다.
  - 추가하려는 세그먼트 유형에 따라 접미사 변수를 domain 또는 ca 로 설정합니다.
  - 복제 계약의 왼쪽 노드가 되고자 하는 IdM 서버의 이름으로 왼쪽 변수를 설정합니다.
  - 올바른 변수를 복제 계약의 올바른 노드로 설정하고자 하는 IdM 서버의 이름으로 설정합니다.

5. add-topologysegments-copy.yml 파일의 tasks 섹션에서 state 변수가 present로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Add topology segments
  hosts: ipaserver
  become: true
  gather_facts: false

  vars:
```

```
    ipaadmin_password: Secret123
    ipatopology_segments:
    - {suffix: domain, left: replica1.idm.example.com , right: replica2.idm.example.com }
    - {suffix: domain, left: replica2.idm.example.com , right: replica3.idm.example.com }
    - {suffix: domain, left: replica3.idm.example.com , right: replica4.idm.example.com }
    - {suffix: domain+ca, left: replica4.idm.example.com , right:
replica1.idm.example.com }

  tasks:
  - name: Add topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: "{{ item.suffix }}"
      name: "{{ item.name | default(omit) }}"
      left: "{{ item.left }}"
      right: "{{ item.right }}"
      state: present
      #state: absent
      #state: checked
      #state: reinitialized
    loop: "{{ ipatopology_segments | default([]) }}"
```

6. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i inventory add-topologysegments-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 설명 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트 를 참조하십시오.
- **ipatopologysegment** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
  - **/usr/share/doc/ansible -freeipa/** 디렉터리의 **README- topology.md** 파일. 이 파일에는 **ipatopologysegment** 변수 정의도 포함되어 있습니다.
  - **/usr/share/doc/ansible-freeipa/playbooks/topology** 디렉토리.

14.3. ANSIBLE을 사용하여 두 복제본 간에 복제 계약이 있는지 확인

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약에 따라 복제됩니다. 두 서버에 복제 계약이 구성된 경우 데이터를 공유합니다. 복제 계약은 항상 양방향입니다. 데이터는 첫 번째 복제본에서 다른 복제본으로는 물론 다른 복제본으로 복제됩니다.

이 섹션에서는 IdM의 여러 복제본 쌍 간에 복제 계약이 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- 토폴로지 에서 복제본 연결에 나열된 IdM(Identity Management) 토폴로지 설계 권장 사항을 이해해야 합니다.
- IdM 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - ansible-freeipa 패키지를 설치했습니다.
  - ~/*MyPlaybooks*/ 디렉터리에 이러한 옵션을 구성하는 IdM 서버의 정규화된 도메인 이름(FQDN)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/topology/ 디렉터리에 있는 check-topologysegments.yml Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/check-topologysegments.yml
check-topologysegments-copy.yml
```

3. **check-topologysegments-copy.yml** 파일을 열어 편집합니다.

4. **vars** 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

   - **ipaadmin_password** 변수를 **IdM** 관리자의 암호로 설정합니다.
   - 모든 토폴로지 세그먼트에 대해 **ipatopology_segments** 섹션에 행을 추가하고 다음 변수를 설정합니다.
     - 추가하려는 세그먼트 유형에 따라 접미사 변수를 **domain** 또는 **ca** 로 설정합니다.
     - 복제 계약의 왼쪽 노드가 되고자 하는 **IdM** 서버의 이름으로 왼쪽 변수를 설정합니다.
     - 올바른 변수를 복제 계약의 올바른 노드로 설정하고자 하는 **IdM** 서버의 이름으로 설정합니다.

5. **check-topologysegments-copy.yml** 파일의 **tasks** 섹션에서 **state** 변수가 **present**로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Add topology segments
  hosts: ipaserver
  become: true
  gather_facts: false

  vars:
    ipaadmin_password: Secret123
    ipatopology_segments:
    - {suffix: domain, left: replica1.idm.example.com, right: replica2.idm.example.com }
    - {suffix: domain, left: replica2.idm.example.com , right: replica3.idm.example.com }
    - {suffix: domain, left: replica3.idm.example.com , right: replica4.idm.example.com }
    - {suffix: domain+ca, left: replica4.idm.example.com , right:
replica1.idm.example.com }

  tasks:
  - name: Check topology segment
    ipatopologysegment:
```

```
    ipaadmin_password: "{{ ipaadmin_password }}"
    suffix: "{{ item.suffix }}"
    name: "{{ item.name | default(omit) }}"
    left: "{{ item.left }}"
    right: "{{ item.right }}"
    state: checked
  loop: "{{ ipatopology_segments | default([]) }}"
```

6. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i inventory check-topologysegments-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 설명 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트 를 참조하십시오.
- **ipatopologysegment** 모듈을 사용하는 샘플 **Ansible Playbook**은 다음을 참조하십시오.
  - **/usr/share/doc/ansible -freeipa/** 디렉터리의 **README- topology.md** 파일. 이 파일에는 **ipatopologysegment** 변수 정의도 포함되어 있습니다.
  - **/usr/share/doc/ansible-freeipa/playbooks/topology** 디렉토리.

14.4. ANSIBLE을 사용하여 토폴로지 접미사가 IDM에 있는지 확인합니다.

IdM(Identity Management)의 복제 계약 컨텍스트에서 토폴로지 접미사는 복제되는 데이터를 저장합니다. **IdM**은 두 가지 유형의 토폴로지 접미사인 **domain** 과 **ca** 를 지원합니다. 각 접미사는 별도의 백엔드인 별도의 복제 토폴로지를 나타냅니다. 복제 계약이 구성되면 서로 다른 두 서버에서 동일한 유형의 두 토폴로지 접미사를 결합합니다.

도메인 접미사에는 사용자, 그룹 및 정책과 같은 모든 도메인 관련 데이터가 포함됩니다. **ca** 접미사에는 **Certificate System** 구성 요소에 대한 데이터가 포함되어 있습니다. **CA**(인증 기관)가 설치된 서버에만 존재합니다.

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM**에 토폴로지 접미사가 있는지 확인하는 방법을 설명

합니다. 이 예제에서는 도메인 접미사가 IdM에 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/topology/ 디렉터리에 있는 verify-topology suffix.yml Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/ verify-topologysuffix.yml
verify-topologysuffix-copy.yml
```

3. 편집을 위해 verify-topologysuffix-copy.yml Ansible 플레이북 파일을 엽니다.
4. ipatopologysuffix 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
   - 접미사 변수를 domain 로 설정합니다. ca 접미사가 있는지 확인하는 경우 변수를 ca 로

설정합니다.

- state 변수가 verify 로 설정되어 있는지 확인합니다 . 다른 옵션은 없습니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
  - name: Verify topology suffix
    ipatopologysuffix:
      ipaadmin_password: Secret123
      suffix: domain
      state: verified
```

5. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i inventory verify-topologysuffix-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 설명 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트 를 참조하십시오.

- ipatopologysegment 모듈을 사용하는 샘플 Ansible Playbook은 다음을 참조하십시오.

  - /usr/share/doc/ansible -freeipa/ 디렉터리의 README- topology.md 파일. 이 파일에는 ipatopologysuffix 변수 정의도 포함되어 있습니다.

  - /usr/share/doc/ansible-freeipa/playbooks/topology 디렉토리.

14.5. ANSIBLE을 사용하여 IDM 복제본 다시 초기화

복제본이 장기간 오프라인 상태이거나 데이터베이스가 손상된 경우 다시 초기화할 수 있습니다. 다시 초기화하면 업데이트된 데이터 세트로 복제본을 새로 고칩니다. 예를 들어 백업에서 권한 있는 복원이 필요한 경우 다시 초기화할 수 있습니다.

참고

복제 업데이트와 대조적으로 복제본이 변경된 항목만 상호 전송하는 반면 다시 초기화하면 전체 데이터베이스가 새로 고칩니다.

명령을 실행하는 로컬 호스트는 다시 초기화된 복제본입니다. 데이터를 가져오는 복제본을 지정하려면 **direction** 옵션을 사용합니다.

이 섹션에서는 **Ansible** 플레이북을 사용하여 **server.idm.example. com**에서 **replica.idm.example.com** 의 도메인 데이터를 다시 초기화하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. ***~/MyPlaybooks/*** 디렉터리로 이동합니다.

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

2.

/usr/share/doc/ansible -freeipa/playbooks/topology/ 디렉터리에 있는 reinitialize-topology segment.yml Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/reinitialize-
topologysegment.yml reinitialize-topologysegment-copy.yml
```

3. 편집을 위해 reinitialize-topologysegment-copy.yml 파일을 엽니다.

4. ipatopologysegment 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
- 접미사 변수를 domain 로 설정합니다. ca 데이터를 다시 초기화하는 경우 변수를 ca 로 설정합니다.
- 왼쪽 변수를 복제 계약의 왼쪽 노드로 설정합니다.
- 올바른 변수를 복제 계약의 올바른 노드로 설정합니다.
- 방향 변수를 다시 초기화 데이터 방향으로 설정합니다. 왼쪽에서 오른쪽 방향으로 데이터를 왼쪽 노드에서 오른쪽 노드로 이동합니다.
- state 변수가 다시 초기화 되도록 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
  - name: Reinitialize topology segment
    ipatopologysegment:
      ipaadmin_password: Secret123
      suffix: domain
      left: server.idm.example.com
```

```
right: replica.idm.example.com
direction: left-to-right
state: reinitialized
```

5. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i inventory reinitialize-topologysegment-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 설명 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트 를 참조하십시오.
- ipatopologysegment 모듈을 사용하는 샘플 Ansible Playbook은 다음을 참조하십시오.
  - /usr/share/doc/ansible -freeipa/ 디렉터리의 README- topology.md 파일. 이 파일에는 ipatopologysegment 변수 정의도 포함되어 있습니다.
  - /usr/share/doc/ansible-freeipa/playbooks/topology 디렉토리.

14.6. ANSIBLE을 사용하여 IDM에 복제 계약이 없음

IdM(Identity Management) 서버에 저장된 데이터는 복제 계약에 따라 복제됩니다. 두 서버에 복제 계약이 구성된 경우 데이터를 공유합니다. 복제 계약은 항상 양방향입니다. 데이터는 첫 번째 복제본에서 다른 복제본으로는 물론 다른 복제본으로 복제됩니다.

이 섹션에서는 두 복제본 간의 복제 연결이 IdM에 없는지 확인하는 방법에 대해 설명합니다. 이 예제에서는 replica01.idm.example.com 및 replica02.idm.example.com IdM 서버 사이에 도메인 유형의 복제 계약이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- 토폴로지에서 복제본 연결에 나열된 IdM 토폴로지 설계 권장 사항을 이해했는지 확인하십시오. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-

single/planning_identity_management/index#connecting-the-replicas-in-a-topology_planning-the-replica-topology

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

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/topology/ 디렉터리에 있는 delete-topology segment.yml Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/delete-topologysegment.yml delete-topologysegment-copy.yml
```

3. 편집을 위해 delete-topologysegment-copy.yml 파일을 엽니다.
4. ipatopologysegment 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
   - 접미사 변수를 domain 로 설정합니다. 또는 왼쪽 및 오른쪽 노드 간에 ca 데이터가 복제되지 않도록 하는 경우 변수를 ca 로 설정합니다.

- 왼쪽 변수를 복제 계약의 왼쪽 노드인 IdM 서버의 이름으로 설정합니다.
- 올바른 변수를 복제 계약의 올바른 노드인 IdM 서버의 이름으로 설정합니다.
- state 변수가 absent 로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
- name: Delete topology segment
    ipatopologysegment:
      ipaadmin_password: Secret123
      suffix: domain
      left: replica01.idm.example.com
      right: replica02.idm.example.com:
      state: absent
```

5. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i inventory delete-topologysegment-copy.yml
```

추가 리소스

- 토폴로지 계약, 접미사 및 세그먼트 개념에 대한 자세한 내용은 설명 복제 계약, 토폴로지 접미사 및 토폴로지 세그먼트 를 참조하십시오.
- ipatopologysegment 모듈을 사용하는 샘플 Ansible Playbook은 다음을 참조하십시오.
  - /usr/share/doc/ansible -freeipa/ 디렉터리의 README- topology.md 파일. 이 파일에

는 **ipatopologysegment** 변수 정의도 포함되어 있습니다.

- **/usr/share/doc/ansible-freeipa/playbooks/topology** 디렉토리.

14.7. 추가 리소스

- **IdM** 배포의 토폴로지를 계획하는 방법에 대한 자세한 내용은 복제 토폴로지 계획을 참조하십시오.

- **IdM** 복제본을 설치하는 방법에 대한 자세한 내용은 **IdM** 복제본 설치를 참조하십시오 .

15장. ANSIBLE 플레이북을 사용하여 호스트 관리

Ansible은 시스템을 구성하고, 소프트웨어를 배포하고, 롤링 업데이트를 수행하는 데 사용되는 자동화 도구입니다. Ansible에는 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.

- IdM의 호스트
- 호스트 등록
- FQDN에서만 정의된 IdM 호스트 항목이 있는지 확인
- IP 주소가 있는 IdM 호스트 항목이 있는지 확인
- 임의의 암호를 사용하여 여러 IdM 호스트 항목이 있는지 확인
- 여러 IP 주소가 있는 IdM 호스트 항목이 있는지 확인
- IdM 호스트 항목이 없는지 확인

15.1. IDM의 호스트

IdM(Identity Management)은 이러한 ID를 관리합니다.

- 사용자
- 서비스

- 호스트

호스트는 시스템을 나타냅니다. IdM ID로 호스트에 IdM LDAP에 항목이 있으며, 이는 IdM 서버의 디렉터리 서버 인스턴스 389입니다.

IdM LDAP의 host 항목은 다른 호스트와 도메인 내의 서비스 간 관계를 설정하는 데 사용됩니다. 이러한 관계는 도메인 내의 호스트에 권한 부여 및 제어 *위임* 의 일부입니다. 호스트 기반 액세스 제어(HBAC) 규칙에서 모든 호스트 를 사용할 수 있습니다.

IdM 도메인은 공통 ID 정보, 공통 정책 및 공유 서비스가 포함된 시스템 간 공통성을 설정합니다. 도메인 기능에 속한 시스템은 도메인의 클라이언트에 속하므로, 도메인에서 제공하는 서비스를 사용합니다. IdM 도메인은 시스템 전용 세 가지 기본 서비스를 제공합니다.

- DNS
- Kerberos
- 인증서 관리

IdM의 호스트는 실행 중인 서비스와 긴밀하게 연결됩니다.

- 서비스 항목은 호스트와 연결됩니다.
- 호스트는 호스트와 서비스 Kerberos 사용자를 모두 저장합니다.

15.2. 호스트 등록

이 섹션에서는 IdM 클라이언트로 호스트 등록 및 등록 중 및 등록 후에 발생하는 사항에 대해 설명합니다. 이 섹션에서는 IdM 호스트 및 IdM 사용자의 등록을 비교합니다. 또한 이 섹션에서는 호스트에서 사용할 수 있는 대체 유형의 인증에 대해 간단히 설명합니다.

호스트 등록은 다음으로 구성됩니다.

- IdM LDAP에서 호스트 항목 생성: IdM CLI에서 ipa host-add 명령을 사용하거나 이에 상응하는 IdM 웹 UI 작업을 사용할 수 있습니다.

- 호스트에서 IdM 서비스 구성(예: SSSD(시스템 보안 서비스 데몬), Kerberos, certmonger) 호스트를 IdM 도메인에 연결합니다.

두 작업은 별도로 또는 함께 수행할 수 있습니다.

개별적으로 수행되는 경우 두 가지 작업을 서로 다른 수준의 권한을 가진 두 사용자 간에 나눌 수 있습니다. 이는 대량 배포에 유용합니다.

ipa-client-install 명령은 두 작업을 함께 수행할 수 있습니다. 해당 항목이 아직 없는 경우 명령은 IdM LDAP에 호스트 항목을 생성하고 호스트에 대한 Kerberos 및 SSSD 서비스를 둘 다 구성합니다. 이 명령을 사용하면 IdM 도메인 내에 호스트를 가져와 연결할 IdM 서버를 식별할 수 있습니다. 호스트가 IdM에서 관리하는 DNS 영역에 속하는 경우 ipa-client-install 은 호스트에 대한 DNS 레코드도 추가합니다. 명령은 클라이언트에서 실행해야 합니다.

15.2.1. 호스트 등록에 필요한 사용자 권한

호스트 등록 작업을 수행하려면 권한이 없는 사용자가 원하지 않는 시스템을 IdM 도메인에 추가하지 못하도록 인증해야 합니다. 필요한 권한은 다음과 같은 여러 요인에 따라 다릅니다.

- ipa-client-install실행과 별도로 호스트 항목을 생성하는 경우

- 등록에 OTP(일회용 암호)를 사용하는 경우

IdM LDAP에서 수동으로 호스트 항목을 생성하는 데 필요한 사용자 권한

ipa host-add CLI 명령 또는 IdM 웹 UI를 사용하여 IdM LDAP에서 호스트 항목을 생성하는 데 필요한 사용자 권한은 호스트 관리자입니다. 호스트 관리자 권한은 IT 전문가 역할을 통해 얻을 수 있습니다.

클라이언트를 IdM 도메인에 가입하는 사용자 권한

호스트는 ipa-client-install 명령을 실행하는 동안 IdM 클라이언트로 구성됩니다. ipa-client-install 명령을 실행하는 데 필요한 인증 정보의 수준은 다음 중 본인이 찾은 시나리오에 따라 다릅니다.

- IdM LDAP의 호스트 항목이 존재하지 않습니다. 이 시나리오의 경우 전체 관리자 자격 증명

또는 **Host Administrators** 역할이 필요합니다. 전체 관리자는 **admins** 그룹의 구성원입니다. **Host Administrators** 역할은 호스트를 추가하고 호스트를 등록할 수 있는 권한을 제공합니다. 이 시나리오에 대한 자세한 내용은 사용자 자격 증명: 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오.

- **IdM LDAP**의 **host** 항목이 있습니다. 이 시나리오의 경우 **ipa-client-install** 을 실행하려면 제한된 관리자의 인증 정보가 필요합니다. 이 경우 제한된 관리자에게는 호스트 등록 권한을 제공하는 **Enrollment Administrator** 역할이 있습니다. 자세한 내용은 사용자 **credentials:** 대화형 설치를 사용하여 클라이언트 설치를 참조하십시오.

- **IdM LDAP**의 호스트 항목이 존재하며 전체 또는 제한된 관리자가 호스트에 대한 **OTP**가 생성되었습니다. 이 시나리오에서는 **--password** 옵션을 사용하여 **ipa-client-install** 명령을 실행하여 올바른 **OTP**를 제공하는 경우 **IdM** 클라이언트를 일반 사용자로 설치할 수 있습니다. 자세한 내용은 일회성 암호를 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치.

등록 후 **IdM** 호스트는 모든 새 세션을 인증하여 **IdM** 리소스에 액세스할 수 있도록 합니다. **IdM** 서버가 시스템을 신뢰하고 해당 시스템에 설치된 클라이언트 소프트웨어에서 **IdM** 연결을 수락하려면 시스템 인증이 필요합니다. 클라이언트를 인증한 후 **IdM** 서버는 요청에 응답할 수 있습니다.

15.2.2. IdM 호스트 및 사용자의 등록 및 인증: 비교

IdM의 사용자와 호스트 사이에는 많은 유사점이 있습니다. 이 섹션에서는 등록 단계에서 확인할 수 있는 몇 가지 유사점과 배포 단계에서 인증에 관한 몇 가지 유사 사항에 대해 설명합니다.

- 등록 단계(표 15.1. “사용자 및 호스트 등록”):
  - 관리자는 사용자 또는 호스트 모두에 대해 **IdM**에 실제로 참여하기 전에 사용자 및 호스트에 대해 **LDAP** 항목을 생성할 수 있습니다. **stage** 사용자의 경우 명령은 **ipa stageuser-add** 입니다. 호스트에 대해 명령은 **ipa host-add** 입니다.
  - *키 테이블* 또는 일부 익스텐트와 유사한 대칭 키인 **keytab**을 포함하는 파일은 호스트에서 **ipa-client-install** 명령 실행 중에 생성됩니다. 그러면 호스트가 **IdM** 영역에 결합됩니다. 비동기적으로 사용자는 계정을 활성화할 때 암호를 생성하여 **IdM** 영역에 가입해야 합니다.
  - 사용자 **password**는 사용자의 기본 인증 방법이지만 **keytab**은 호스트의 기본 인증 방법입니다. 키탭은 호스트의 파일에 저장됩니다.

표 **15.1.** 사용자 및 호스트 등록

동작	사용자	호스트
사전 등록	$ ipa stageuser-add *user_name* [--password]	$ ipa host-add *host_name* [--random]
계정 활성화	$ ipa stageuser-activate *user_name*	$ IPA-client install [--password] (호스트 자체에서 실행해야합니다)

- **배포 단계(표 15.2. "사용자 및 호스트 세션 인증"):**
  - **사용자가 새 세션을 시작하면 사용자는 암호를 사용하여 인증합니다. 마찬가지로, 콘솔이 켜질 때마다 호스트는 해당 키탭 파일을 표시하여 인증합니다. SSSD(System Security Services Daemon)는 이 프로세스를 백그라운드에서 관리합니다.**
  - **인증에 성공하면 사용자 또는 호스트에서 티켓(TGT)을 부여하는 Kerberos 티켓을 가져옵니다.**
  - **그런 다음 TGT를 사용하여 특정 서비스의 특정 티켓을 받습니다.**

표 15.2. 사용자 및 호스트 세션 인증

	사용자	호스트
기본 인증 수단	**암호**	**키탭**
세션 시작 (일반 사용자)	$ kinit *user_name*	*[호스트의 전환]*
인증에 성공한 결과	특정 서비스에 대한 액세스를 얻는 데 사용할 TGT	특정 서비스에 대한 액세스를 얻는 데 사용할 TGT

TGT 및 기타 Kerberos 티켓은 서버에서 정의한 Kerberos 서비스 및 정책의 일부로 생성됩니다. Kerberos 티켓을 처음 부여하고, Kerberos 자격 증명을 갱신하며, Kerberos 세션 삭제도 IdM 서비스에 의해 자동으로 처리됩니다.

15.2.3. IdM 호스트의 대체 인증 옵션

keytab 외에도 IdM은 두 가지 유형의 시스템 인증을 지원합니다.

-

SSH 키. 호스트의 SSH 공개 키가 생성되어 호스트 항목에 업로드됩니다. 여기에서 SSSD(시스템 보안 서비스 데몬)는 IdM을 ID 프로바이더로 사용하며 OpenSSH 및 기타 서비스와 함께 작업하여 IdM에 있는 공용 키를 참조할 수 있습니다.

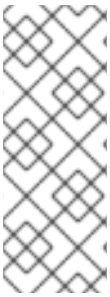
- 시스템 인증서. 이 경우 시스템은 IdM 서버의 인증 기관에서 발급한 SSL 인증서를 사용한 다음 IdM의 디렉터리 서버에 저장됩니다. 그러면 서버에 인증할 때 인증서가 존재하도록 시스템으로 전송됩니다. 클라이언트의 인증서는 certmonger 라는 서비스에서 관리합니다.

15.3. ANSIBLE PLAYBOOK을 사용하여 FQDN과 함께 IDM 호스트 항목이 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에 호스트 항목이 있는지에 대해 설명합니다. 호스트 항목은 FQDN(정규화 된 도메인 이름)으로만 정의합니다.

다음 조건 중 하나가 적용되는 경우 호스트의 FQDN 이름을 지정하는 것만으로 충분합니다.

- IdM 서버는 DNS를 관리하도록 구성되어 있지 않습니다.
- 호스트에 고정 IP 주소가 없거나 호스트가 구성된 시점에는 IP 주소를 알 수 없습니다. FQDN에서만 정의한 호스트를 추가하면 기본적으로 IdM DNS 서비스에 자리 표시자 항목이 생성됩니다. 예를 들어 랩탑은 IdM 클라이언트로 사전 구성될 수 있지만 구성 시 IP 주소는 없습니다. DNS 서비스에서 레코드를 동적으로 업데이트하면 호스트의 현재 IP 주소가 탐지되고 해당 DNS 레코드가 업데이트됩니다.



참고

Ansible이 없으면 ipa host-add 명령을 사용하여 호스트 항목이 IdM에 생성됩니다. IdM에 호스트를 추가한 결과는 IdM에 있는 호스트의 상태입니다. Ansible은 멱등에 의존하므로 Ansible을 사용하여 IdM에 호스트를 추가하려면 호스트 상태를 present: state: present 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

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

절차

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

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

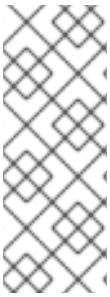
2. 확인할 IdM에 있는 호스트의 **FQDN** 으로 **Ansible** 플레이북 파일을 생성합니다. 이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/host/add-host.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Host present
  hosts: ipaserver
  become: true

  tasks:
  - name: Host host01.idm.example.com present
    ipahost:
      ipaadmin_password: MySecret123
      name: host01.idm.example.com
      state: present
      force: yes
```

3. 플레이북을 실행합니다.

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



참고

이 절차에서는 **IdM LDAP** 서버에 호스트 항목이 생성되지만 호스트를 **IdM Kerberos** 영역에 등록하지 않습니다. 따라서 호스트를 **IdM** 클라이언트로 배포해야 합니다. 자세한 내용은 **Ansible** 플레이북을 사용하여 **Identity Management** 클라이언트 설치를 참조하십시오.

검증 단계

1. **IdM** 서버에 **admin**으로 로그인합니다.

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

2. **ipa host-show** 명령을 입력하고 호스트 이름을 지정합니다.

```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Password: False
  Keytab: False
  Managed by: host01.idm.example.com
```

출력은 **host01.idm.example.com** 이 IdM에 있는지 확인합니다.

15.4. ANSIBLE PLAYBOOK을 사용하여 DNS 정보가 포함된 IDM 호스트 항목이 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**에 호스트 항목이 있는지에 대해 설명합니다. 호스트 항목은 **FQDN**(정규화 된 도메인 이름) 및 해당 **IP** 주소로 정의합니다.

참고

Ansible이 없으면 **ipa host-add** 명령을 사용하여 호스트 항목이 **IdM**에 생성됩니다. **IdM**에 호스트를 추가한 결과는 **IdM**에 있는 호스트의 상태입니다. **Ansible**은 멱등에 의존하므로 **Ansible**을 사용하여 **IdM**에 호스트를 추가하려면 호스트 상태를 **present: state: present** 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

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

절차

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

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

2. 확인하고자 하는 **IdM**에 있는 호스트의 정규화된 도메인 이름 **(FQDN)**을 사용하여 **Ansible** 플레이북 파일을 생성합니다. 또한 **IdM** 서버가 **DNS**를 관리하고 호스트의 **IP** 주소를 알고 있는 경우 **ip_address** 매개 변수의 값을 지정합니다. 호스트가 **DNS** 리소스 레코드에 있으려면 **IP** 주소가 필요합니다. 이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/host/host-**

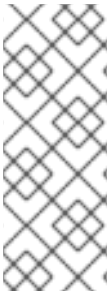
present.yml 파일에서 예제를 복사하고 수정할 수 있습니다. 다음과 같은 기타 추가 정보를 포함할 수도 있습니다.

```
---
- name: Host present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure host01.idm.example.com is present
    ipahost:
      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 절차에서는 IdM LDAP 서버에 호스트 항목이 생성되지만 호스트를 IdM Kerberos 영역에 등록하지 않습니다. 따라서 호스트를 IdM 클라이언트로 배포해야 합니다. 자세한 내용은 Ansible 플레이북을 사용하여 Identity Management 클라이언트 설치를 참조하십시오.

검증 단계

1. IdM 서버에 admin으로 로그인합니다.

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

2. ipa host-show 명령을 입력하고 호스트 이름을 지정합니다.

```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Description: Example host
  Locality: Lab
  Location: Lab
  Platform: Lenovo T61
  Operating system: CentOS 7
  Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
  MAC address: 08:00:27:E3:B1:2D, 52:54:00:BD:97:1E
  Password: False
  Keytab: False
  Managed by: host01.idm.example.com
```

출력은 IdM에 host01.idm.example.com 이 있는지 확인합니다.

15.5. ANSIBLE PLAYBOOK을 사용하여 임의의 암호와 함께 여러 IDM 호스트 항목이 있는지 확인

ipahost 모듈을 사용하면 시스템 관리자가 하나의 Ansible 작업만 사용하여 IdM에 여러 호스트 항목이 있는지 확인할 수 있습니다. 이 섹션에서는 FQDN(정규화 된 도메인 이름)으로만 정의된 여러 호스트 항목이 있는지 확인하는 방법에 대해 설명합니다. Ansible 플레이북을 실행하면 호스트에 대한 임의 암호가 생성됩니다.

참고

Ansible이 없으면 ipa host-add 명령을 사용하여 호스트 항목이 IdM에 생성됩니다. IdM에 호스트를 추가한 결과는 IdM에 있는 호스트의 상태입니다. Ansible은 멱등에 의존하므로 Ansible을 사용하여 IdM에 호스트를 추가하려면 호스트 상태를 present: state: present 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

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

절차

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

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

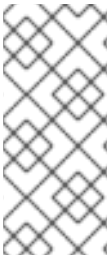
2. 확인하려는 IdM에 있는 호스트의 정규화된 도메인 이름 (FQDN)을 사용하여 Ansible 플레이북 파일을 생성합니다. 호스트가 이미 IdM에 있고 update_password가 on_ create 로 제한된 경우에도 Ansible 플레이북에서 각 호스트의 임의 암호를 생성하도록 하려면 random: yes 및 force: yes 옵션을 추가합니다. 이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/README-host.md Markdown 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- name: Ensure hosts with random password
  hosts: ipaserver
  become: true

  tasks:
  - name: Hosts host01.idm.example.com and host02.idm.example.com present with
random passwords
    ipahost:
      ipaadmin_password: MySecret123
      hosts:
      - name: host01.idm.example.com
        random: yes
        force: yes
      - name: host02.idm.example.com
        random: yes
        force: yes
    register: ipahost
```

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-are-present.yml
[...]
TASK [Hosts host01.idm.example.com and host02.idm.example.com present with
random passwords]
changed: [r8server.idm.example.com] => {"changed": true, "host":
{"host01.idm.example.com": {"randompassword": "0HolRvjUdH0Ycbf6uYdWTxH"},
"host02.idm.example.com": {"randompassword": "5VdLgrf3wvojmACdHC3uA3s"}}}
```



참고

임의의 OTP(일회 암호)를 사용하여 호스트를 IdM 클라이언트로 배포하려면 Ansible 플레이북을 사용하여 IdM 클라이언트 등록에 대한 권한 부여 옵션을 참조하거나 일회성 암호를 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치.

검증 단계

1. IdM 서버에 admin으로 로그인합니다.

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

2. ipa host-show 명령을 입력하고 호스트 중 하나의 이름을 지정합니다.

```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Password: True
  Keytab: False
  Managed by: host01.idm.example.com
```

출력은 임의의 암호가 있는 IdM에 host01.idm.example.com 이 있는지 확인합니다.

15.6. ANSIBLE PLAYBOOK을 사용하여 여러 IP 주소가 있는 IDM 호스트 항목이 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에 호스트 항목이 있는지 확인하는 방법을 설명합니다. 호스트 항목은 FQDN(정규화 된 도메인 이름) 및 여러 IP 주소로 정의합니다.

참고

ipa 호스트 유틸리티와 달리 Ansible ipahost 모듈은 호스트에 대해 여러 개의 IPv4 및 IPv6 주소가 있는지 확인할 수 있습니다. ipa host-mod 명령은 IP 주소를 처리할 수 없습니다.

사전 요구 사항

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

절차

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

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

2. Ansible 플레이북 파일을 생성합니다. 확인하려는 IdM 에 있는 호스트의 정규화된 도메인 이

름 (FQDN)을 **ipahost** 변수의 이름으로 지정합니다. **ip_address** 구문을 사용하여 별도의 행에 여러 **IPv4** 및 **IPv6** *ip_address* 값을 지정합니다. 이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/host/host-member-ipaddresses-present.yml** 파일에서 예제를 복사하고 수정할 수 있습니다. 다음과 같은 추가 정보를 포함할 수도 있습니다.

```
---
- name: Host member IP addresses present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure host101.example.com IP addresses present
    ipahost:
      ipaadmin_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-addreses-is-present.yml
```

참고

이 절차에서는 **IdM LDAP** 서버에 호스트 항목을 생성하지만 호스트를 **IdM Kerberos** 영역에 등록하지 않습니다. 따라서 호스트를 **IdM** 클라이언트로 배포해야 합니다. 자세한 내용은 **Ansible** 플레이북을 사용하여 **Identity Management** 클라이언트 설치를 참조하십시오.

검증 단계

1. **IdM** 서버에 **admin**으로 로그인합니다.

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

2. **ipa host-show** 명령을 입력하고 호스트 이름을 지정합니다.

```
$ ipa host-show host01.idm.example.com
  Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
```

```
Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
Password: False
Keytab: False
Managed by: host01.idm.example.com
```

출력은 **host01.idm.example.com** 이 IdM에 있는지 확인합니다.

3. 호스트의 여러 IP 주소가 IdM DNS 레코드에 있는지 확인하려면 **ipa dnsrecord-show** 명령을 입력하고 다음 정보를 지정합니다.

- IdM 도메인의 이름
- 호스트 이름

```
$ ipa dnsrecord-show idm.example.com host01
[...]
  Record name: host01
  A record: 192.168.0.123, 192.168.0.124
  AAAA record: fe80::20c:29ff:fe02:a1b3, fe80::20c:29ff:fe02:a1b4
```

출력은 플레이북에 지정된 모든 IPv4 및 IPv6 주소가 **host 01.idm.example.com** 호스트 항목과 올바르게 연결되어 있는지 확인합니다.

15.7. ANSIBLE 플레이북을 사용하여 IDM 호스트 항목이 없는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에 호스트 항목이 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 인증 정보

절차

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

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

2. 확인하려는 IdM에 없는 호스트의 정규화된 도메인 이름 (FQDN)을 사용하여 Ansible 플레이북 파일을 생성합니다. IdM 도메인에 통합된 DNS가 있는 경우 updatedns: yes 옵션을 사용하여 DNS에서 호스트의 연결된 레코드를 제거합니다.

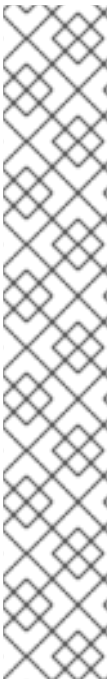
이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/playbooks/host/delete-host.yml 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
  - name: Host host01.idm.example.com absent
    ipahost:
      ipaadmin_password: MySecret123
      name: host01.idm.example.com
      updatedns: yes
      state: absent
```

3. 플레이북을 실행합니다.

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



참고

절차 결과는 다음과 같습니다.

- IdM Kerberos 영역에 존재하지 않는 호스트.
- 호스트 항목이 IdM LDAP 서버에 존재하지 않습니다.

클라이언트 호스트 자체에서 SSSD(System Security Services Daemon)와 같은 시스템 서비스의 특정 IdM 구성을 제거하려면 클라이언트에서 ipa-client-install --uninstall 명령을 실행해야 합니다. 자세한 내용은 IdM 클라이언트 설치 제거를 참조하십시오.

검증 단계

1. admin으로 ipaserver 에 로그인합니다.

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

2. *host01.idm.example.com* 에 대한 정보를 표시 :

```
$ ipa host-show host01.idm.example.com
ipa: ERROR: host01.idm.example.com: host not found
```

출력은 호스트가 IdM에 없는지 확인합니다.

추가 리소스

- ipahost 변수의 정의와 샘플 Ansible 플레이북을 통해 /usr/share/doc/ansible-freeipa/README-host.md Markdown 파일에서 호스트의 존재, 누락 및 비활성화를 확인할 수 있습니다.

- 추가 플레이북은 /usr/share/doc/ansible-freeipa/playbooks/host 디렉터리에 있습니다.

16장. ANSIBLE 플레이북을 사용하여 호스트 그룹 관리

이 장에서는 IdM(Identity Management)의 호스트 그룹을 소개하고 IdM(Identity Management)의 호스트 그룹과 관련된 다음 작업을 수행하기 위해 Ansible을 사용합니다.

- IdM의 호스트 그룹
- IdM 호스트 그룹이 있는지 확인
- IdM 호스트 그룹에 호스트가 있는지 확인
- IdM 호스트 그룹 중첩
- IdM 호스트 그룹에 멤버 관리자가 있는지 확인
- IdM 호스트 그룹에서 호스트가 없는지 확인
- IdM 호스트 그룹에서 중첩된 호스트 그룹이 없는지 확인합니다.
- IdM 호스트 그룹에서 멤버 관리자가 없는지 확인

16.1. IDM의 호스트 그룹

IdM 호스트 그룹은 중요한 관리 작업, 특히 액세스 제어에 대한 제어를 중앙 집중화하는 데 사용할 수 있습니다.

호스트 그룹의 정의

호스트 그룹은 공통 액세스 제어 규칙 및 기타 특성이 있는 IdM 호스트 집합이 포함된 엔터티입니다. 예를 들어 회사 부서, 물리적 위치 또는 액세스 제어 요구 사항을 기반으로 호스트 그룹을 정의할 수 있습니다.

IdM의 호스트 그룹에는 다음이 포함될 수 있습니다.

- IdM 서버 및 클라이언트
- 기타 IdM 호스트 그룹

기본적으로 생성된 호스트 그룹

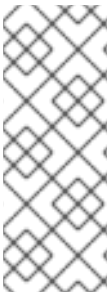
기본적으로 IdM 서버는 모든 IdM 서버 호스트에 대한 호스트 그룹 ipaservers 를 생성합니다.

직접 및 간접 그룹 구성원

IdM의 그룹 속성은 직접 및 간접 구성원 모두에 적용됩니다. 호스트 그룹 B가 호스트 그룹 A의 구성원이면 호스트 그룹 B의 모든 구성원이 호스트 그룹 A의 간접 구성원으로 간주됩니다.

16.2. ANSIBLE PLAYBOOK을 사용하여 IDM 호스트 그룹이 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에 호스트 그룹이 있는지 확인하는 방법을 설명합니다.



참고

Ansible이 없으면 ipa hostgroup-add 명령을 사용하여 호스트 그룹 항목이 IdM에 생성됩니다. IdM에 호스트 그룹을 추가한 결과는 IdM에 있는 호스트 그룹의 상태입니다. Ansible은 멱등에 의존하므로 Ansible을 사용하여 호스트 그룹을 IdM에 추가하려면 호스트 그룹의 상태를 present: state: present 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

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

절차

1. inventory .file과 같은 인벤토리 파일을 생성하고 타겟 IdM 서버 목록으로 ipaserver 를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 예를 들어 **database**라는 호스트 그룹이 있는지 확인하려면 **- ipahostgroup** 작업에 **name: database**를 지정합니다. 이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-present.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
  # Ensure host-group databases is present
  - ipahostgroup:
      ipaadmin_password: MySecret123
      name: databases
      state: present
```

플레이북에서 **state: present** 는 이미 존재하지 않는 한 **IdM**에 호스트 그룹을 추가하라는 요청을 나타냅니다.

3. 플레이북을 실행합니다.

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

검증 단계

1. **admin**으로 **ipaserver** 에 로그인합니다.

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

2. 관리자용 **Kerberos** 티켓을 요청합니다.

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

3. 확인하고자 하는 IdM에 있는 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
  Host-group: databases
```

데이터베이스 호스트 그룹은 IdM에 있습니다.

16.3. ANSIBLE PLAYBOOK을 사용하여 IDM 호스트 그룹에 호스트가 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)의 호스트 그룹에 호스트가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.
- Ansible 플레이북에서 참조할 호스트는 IdM에 있습니다. 자세한 내용은 Ansible 플레이북을 사용하여 IdM 호스트 항목이 있는지 확인하십시오.
- Ansible 플레이북 파일에서 참조하는 호스트 그룹이 IdM에 추가되었습니다. 자세한 내용은 Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 확인합니다.

절차

1. inventory .file과 같은 인벤토리 파일을 생성하고 타겟 IdM 서버 목록으로 ipaserver 를 정의합니다.

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

2. 필요한 호스트 정보를 사용하여 Ansible 플레이북 파일을 생성합니다. ipahostgroup 변수의 name 매개 변수를 사용하여 호스트 그룹의 이름을 지정합니다. ipahostgroup 변수의 host 매개 변수를 사용하여 호스트 이름을 지정합니다. 이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-present-in-hostgroup.yml 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
  # Ensure host-group databases is present
  - ipahostgroup:
      ipaadmin_password: MySecret123
      name: databases
      host:
      - db.idm.example.com
      action: member
```

이 플레이북은 **db.idm.example.com** 호스트를 **databases** 호스트 그룹에 추가합니다. **action: member** 행은 플레이북이 실행될 때 데이터베이스 그룹 자체를 추가하기 위한 시도가 수행되지 않았음을 나타냅니다. 대신 **db.idm.example.com** 을 데이터베이스에 추가하려는 시도만 수행합니다.

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-present-in-
hostgroup.yml
```

검증 단계

1. admin으로 ipaserver 에 로그인합니다.

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

2. 관리자용 Kerberos 티켓을 요청합니다.

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

3. 호스트 그룹에 대한 정보를 표시하여 어떤 호스트가 있는지 확인합니다.

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member hosts: db.idm.example.com
```

db.idm.example.com 호스트는 데이터베이스 호스트 그룹의 멤버로 있습니다.

16.4. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹 중첩

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)** 호스트 그룹에 중첩된 호스트 그룹이 있는지에 대해 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **Ansible** 플레이북 파일에서 참조하는 호스트 그룹은 **IdM**에 있습니다. 자세한 내용은 **Ansible** 플레이북을 사용하여 **IdM** 호스트 그룹이 있는지 확인합니다.

절차

1. **inventory .file**과 같은 인벤토리 파일을 생성하고 타겟 **IdM** 서버 목록으로 **ipaserver** 를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 **Ansible** 플레이북 파일을 생성합니다. 중첩된 호스트 그룹 ***A*** 가 **Ansible** 플레이북의 호스트 그룹 ***B*** 에 있는지 확인하려면 **- ipahostgroup** 변수 중 **name** 변수를 사용하여 호스트 그룹 ***B*** 의 이름을 지정합니다. **host group** 변수를 *사용하여* 중첩 호스트 그룹의 이름을 지정합니다. 이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-present-in-hostgroup.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
  # Ensure hosts and hostgroups are present in existing databases hostgroup
  - ipahostgroup:
      ipaadmin_password: MySecret123
      name: databases
```

```
    hostgroup:
    - mysql-server
    - oracle-server
    action: member
```

이 Ansible 플레이북은 databases 호스트 그룹에 myqsl-server 및 oracle-server 호스트 그룹이 있는지 확인합니다 . action: member 행은 플레이북이 실행될 때 데이터베이스 그룹 자체를 IdM에 추가하기 위한 시도가 수행되지 않았음을 나타냅니다.

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-present-in-
hostgroup.yml
```

검증 단계

1. admin으로 ipaserver 에 로그인합니다.

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

2. 관리자용 Kerberos 티켓을 요청합니다.

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

3. 중첩 호스트 그룹이 있는 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member hosts: db.idm.example.com
  Member host-groups: mysql-server, oracle-server
```

mysql-server 및 oracle-server 호스트 그룹은 databases 호스트 그룹에 있습니다.

16.5. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에 멤버 관리자가 있는지 확인

다음 절차에서는 Ansible 플레이북을 사용하여 IdM 호스트 및 호스트 그룹에 구성원 관리자가 있는지

확인하는 방법을 설명합니다.

사전 요구 사항

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

절차

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

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

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

```
---

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

  tasks:
  - name: Ensure member manager user example_member is present for group_name
      ipahostgroup:
        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. 관리자로 **ipaserver** 에 로그인합니다.

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

2. ***testhostgroup*** 에 대한 정보 표시 :

```
ipaserver]$ ipa hostgroup-show group_name
  Host-group: group_name
  Member hosts: server.idm.example.com
  Member host-groups: testhostgroup2
  Membership managed by groups: project_admins
  Membership managed by users: example_member
```

추가 리소스

- **ipa hostgroup-add-member-manager --help**를 참조하십시오.
- **ipa** 도움말 페이지를 참조하십시오.

16.6. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 호스트가 없는지 확인합니다.

이 섹션에서는 **Ansible** 플레이북을 사용하여 **IdM(Identity Management)**의 호스트 그룹에서 호스트가 없는지 확인하는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.
- Ansible 플레이북에서 참조할 호스트는 IdM에 있습니다. 자세한 내용은 Ansible 플레이북을 사용하여 IdM 호스트 항목이 있는지 확인하십시오.
- Ansible 플레이북 파일에서 참조하는 호스트 그룹은 IdM에 있습니다. 자세한 내용은 Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 확인합니다.

절차

1. inventory .file과 같은 인벤토리 파일을 생성하고 타겟 IdM 서버 목록으로 ipaserver 를 정의합니다.

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

2. 필요한 호스트 및 호스트 그룹 정보를 사용하여 Ansible 플레이북 파일을 생성합니다. ipahostgroup 변수의 name 매개 변수를 사용하여 호스트 그룹의 이름을 지정합니다. ipahostgroup 변수의 host 매개 변수를 사용하여 확인할 호스트 그룹에 없는 호스트 이름을 지정합니다. 이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-absent-in-hostgroup.yml 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
  # Ensure host-group databases is absent
  - ipahostgroup:
      ipaadmin_password: MySecret123
      name: databases
      host:
      - db.idm.example.com
      action: member
      state: absent
```

이 플레이북은 databases 호스트 그룹에서 db.idm.example.com 호스트가 없는지 확인합니다. action: member 행은 플레이북이 실행될 때 데이터베이스 그룹 자체를 제거하기 위한 시도가 수행되지 않았음을 나타냅니다.

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-absent-in-
hostgroup.yml
```

검증 단계

1. admin으로 ipaserver 에 로그인합니다.

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

2. 관리자용 Kerberos 티켓을 요청합니다.

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

3. 호스트 그룹 및 포함된 호스트에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member host-groups: mysql-server, oracle-server
```

db.idm.example.com 호스트가 databases 호스트 그룹에 존재하지 않습니다.

16.7. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 중첩된 호스트 그룹이 없는지 확인합니다.

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)의 외부 호스트 그룹에서 중첩된 호스트 그룹이 없는지 확인하는 방법에 대해 설명합니다.

사전 요구 사항

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

- Ansible 플레이북 파일에서 참조하는 호스트 그룹은 IdM에 있습니다. 자세한 내용은 Ansible 플레이북을 사용하여 IdM 호스트 그룹이 있는지 확인합니다.

절차

1. inventory .file과 같은 인벤토리 파일을 생성하고 타겟 IdM 서버 목록으로 ipaserver 를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 Ansible 플레이북 파일을 생성합니다. ipahostgroup 변수 중에 name 변수를 사용하여 외부 호스트 그룹의 이름을 지정합니다. 호스트 그룹 변수를 사용하여 중첩 호스트 그룹의 이름을 지정합니다. 이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-absent-in-hostgroup.yml 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
  # Ensure hosts and hostgroups are absent in existing databases hostgroup
  - ipahostgroup:
      ipaadmin_password: MySecret123
      name: databases
      hostgroup:
      - mysql-server
      - oracle-server
      action: member
      state: absent
```

이 플레이북은 데이터베이스 호스트 그룹에 mysql-server 및 oracle-server 호스트 그룹이 없는지 확인합니다. action: member 행은 플레이북이 실행될 때 데이터베이스 그룹 자체가 IdM 에서 삭제되었는지 확인하기 위한 시도가 수행되지 않았음을 나타냅니다.

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hosts-or-hostgroups-are-absent-in-
hostgroup.yml
```

검증 단계

1. admin으로 ipaserver 에 로그인합니다.

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

2. 관리자용 Kerberos 티켓을 요청합니다.

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

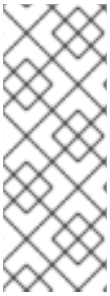
3. 중첩 호스트 그룹이 없어야 하는 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
  Host-group: databases
```

출력은 mysql-server 및 oracle- server 중첩 호스트 그룹이 외부 데이터베이스 호스트 그룹에 없는지 확인합니다.

16.8. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹이 없는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에 호스트 그룹이 없는지 확인하는 방법을 설명합니다.



참고

Ansible이 없으면 ipa hostgroup-del 명령을 사용하여 호스트 그룹 항목이 IdM에서 제거됩니다. IdM에서 호스트 그룹을 제거한 결과는 IdM에 없는 호스트 그룹의 상태입니다. Ansible은 멱등에 의존하므로 Ansible을 사용하여 IdM에서 호스트 그룹을 제거하려면 호스트 그룹의 상태를 absent: state: absent 로 정의하는 플레이북을 생성해야 합니다.

사전 요구 사항

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

절차

1. inventory .file과 같은 인벤토리 파일을 생성하고 타겟 IdM 서버 목록으로 ipaserver 를 정의합니다.

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

2. 필요한 호스트 그룹 정보를 사용하여 Ansible 플레이북 파일을 생성합니다. 이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-absent.yml 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
  - Ensure host-group databases is absent
    ipahostgroup:
      ipaadmin_password: MySecret123
      name: databases
      state: absent
```

이 플레이북은 IdM에서 데이터베이스 호스트 그룹이 없는지 확인합니다. state: absent 는 이미 삭제되지 않는 한 IdM에서 호스트 그룹을 삭제하도록 요청합니다.

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-hostgroup-is-absent.yml
```

검증 단계

1. admin으로 ipaserver 에 로그인합니다.

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

2. 관리자용 Kerberos 티켓을 요청합니다.

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

3. 확인하지 않은 호스트 그룹에 대한 정보를 표시합니다.

```
$ ipa hostgroup-show databases
ipa: ERROR: databases: host group not found
```

데이터베이스 호스트 그룹은 IdM에 존재하지 않습니다.

16.9. ANSIBLE 플레이북을 사용하여 IDM 호스트 그룹에서 멤버 관리자가 없는지 확인합니다.

다음 절차에서는 Ansible 플레이북을 사용하여 IdM 호스트 및 호스트 그룹에 구성원 관리자가 없는지 확인합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.
- 멤버 관리자로 제거하려는 사용자 또는 사용자 그룹의 이름과 관리 중인 호스트 그룹의 이름이 있어야 합니다.

절차

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

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

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

```
---

- name: Playbook to handle host group membership management
  hosts: ipaserver
```

```
  become: true

  tasks:
  - name: Ensure member manager host and host group members are absent for
group_name
    ipahostgroup:
      ipaadmin_password: MySecret123
      name: group_name
      membermanager_user: example_member
      membermanager_group: project_admins
      action: member
      state: absent
```

3. 플레이북을 실행합니다.

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

검증 단계

ipa group-show 명령을 사용하여 group_name 그룹에 example_member 또는 project_admins 가 멤버 관리자로 포함되어 있지 않은지 확인할 수 있습니다.

1. 관리자로 ipaserver 에 로그인합니다.

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

2. *testhostgroup* 에 대한 정보 표시 :

```
ipaserver]$ ipa hostgroup-show group_name
  Host-group: group_name
  Member hosts: server.idm.example.com
  Member host-groups: testhostgroup2
```

추가 리소스

- ipa hostgroup-add-member-manager --help를 참조하십시오.
- ipa 도움말 페이지를 참조하십시오.

17장. IDM 암호 정책 정의

이 장에서는 IdM(Identity Management) 암호 정책과 Ansible 플레이북을 사용하여 IdM에 새 암호 정책을 추가하는 방법을 설명합니다.

17.1. 암호 정책이란 무엇입니까

암호 정책은 암호가 충족해야 하는 규칙 집합입니다. 예를 들어 암호 정책은 최소 암호 길이와 최대 암호 기간을 정의할 수 있습니다. 이 정책의 영향을 받는 모든 사용자는 충분히 긴 암호를 설정하고 지정된 조건을 충족하기 위해 암호를 자주 변경해야 합니다. 이러한 방식으로 암호 정책은 사용자의 암호를 검색하고 오용할 위험을 줄이는 데 도움이 됩니다.

17.2. IDM의 암호 정책

IdM(Identity Management) 사용자가 IdM(Identity Management) 도메인에 인증하는 가장 일반적인 방법입니다. 암호 정책은 이러한 IdM 사용자 암호를 충족해야 하는 요구 사항을 정의합니다.

참고

IdM 암호 정책은 기본 LDAP 디렉터리에 설정되지만 Kerberos KDC(키 배포 센터)는 암호 정책을 적용합니다.

암호 정책 속성은 IdM에서 암호 정책을 정의하는 데 사용할 수 있는 속성을 나열합니다.

표 17.1. 암호 정책 속성

속성	설명	예제
최대 수명	암호가 유효한 최대 시간(사용자가 재설정해야 함).	최대 수명 = 90 사용자 암호는 90일 동안만 유효합니다. 그런 다음 IdM은 사용자에게 변경하라는 메시지를 표시합니다.
최소 수명 주기	두 개의 암호 변경 작업 간에 경과해야 하는 최소 시간(시간)입니다.	최소 수명 주기 = 1 사용자가 암호를 변경한 후 1시간 이상 기다린 후 다시 변경해야 합니다.

속성	설명	예제
기록 크기	저장된 이전 암호의 수입니다. 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 수입니다. 문자 클래스는 다음과 같습니다. * 대문자 * 소문자 * 숫자 * 쉼표 (,), 마침표 (.), 별표 (*)와 같은 특수 문자 * 기타 UTF-8 문자 행에서 세 번 이상 문자를 사용하면 문자 클래스가 1씩 줄어듭니다. 예를 들면 다음과 같습니다. * **Secret1** 에는 대문자, 소문자, 숫자 등 3개 문자 클래스가 있습니다. * **Secret111** 에는 두 개의 문자 클래스가 있습니다.: 대문자, 소문자, 숫자 및 **1** 을 반복적으로 사용할 경우 -1 위약금입니다.	문자 클래스 = 0 필요한 기본 클래스 수는 0입니다. 번호를 구성하려면 **ipa pwpolicy-mod** 명령을 **--minclasses** 옵션과 함께 실행합니다. 이 표 아래에 있는 중요 참고 사항을 참조하십시오.
최소 길이	암호의 최소 문자 수입니다. 추가 암호 정책 옵션이 설정된 경우 최소 길이 옵션이 설정된 값과 관계없이 최소 암호 길이는 6입니다.	최소 길이 = 8 사용자는 8자보다 짧은 암호를 사용할 수 없습니다.
최대 실패	IdM이 사용자 계정을 잠그기 전에 실패한 최대 로그인 횟수입니다.	최대 실패 = 6 IdM은 사용자가 잘못된 암호를 7번 입력하면 사용자 계정을 잠급니다.
실패 재설정 간격	IdM이 실패한 로그인 시도의 현재 수를 재설정한 후 시간(초)입니다.	실패 재설정 간격 = 60 사용자가 **Max 실패에** 정의된 실패한 로그인 시도 횟수가 1분 이상 기다린 후 사용자는 사용자 계정 잠금의 위험 없이 다시 로그인을 시도할 수 있습니다.

속성	설명	예제
잠금 기간	**최대 실패에** 정의된 실패한 로그인 횟수가 지난 후 사용자 계정이 잠긴 시간(초)입니다.	잠금 기간 = 600 잠긴 계정이 있는 사용자는 10분 동안 로그인할 수 없습니다.

중요

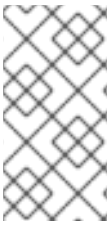
국제 문자와 기호에 액세스할 수 없는 다양한 하드웨어 세트가 있는 경우 문자 클래스 요구 사항에 영어 알파벳 및 공통 기호를 사용하십시오. 암호의 문자 클래스 정책에 대한 자세한 내용은 Red Hat Knowledgebase의 암호에 유효한 문자를 참조하십시오.

17.3. ANSIBLE 플레이북을 사용하여 IDM에 암호 정책이 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에 암호 정책이 있는지 확인하는 방법을 설명합니다.

IdM의 기본 global_policy 암호 정책에서 암호의 다양한 문자 클래스 수가 0으로 설정됩니다. 기록 크기도 0으로 설정되어 있습니다.

Ansible 플레이북을 사용하여 IdM 그룹에 대해 더 강력한 암호 정책을 적용하려면 이 절차를 완료합니다.



참고

IdM 그룹에 대한 암호 정책만 정의할 수 있습니다. 개별 사용자에 대한 암호 정책을 정의할 수 없습니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.
- IdM 관리자 암호를 알고 있습니다.
- 암호 정책이 IdM에 있는지 확인하는 그룹입니다.

절차

1. 인벤토리 파일(예: **inventory.file**)을 생성하고 **[ipaserver]** 섹션에 **IdM** 서버의 **FQDN** 을 정의합니다.

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

2. 확인할 암호 정책을 정의하는 **Ansible** 플레이북 파일을 만듭니다. 이 단계를 간소화하려면 **/usr/share/doc/ansible-freeipa/playbooks/pwpolicy/pwpolicy_present.yml** 파일에서 예제를 복사하고 수정합니다.

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

  tasks:
  - name: Ensure presence of pwpolicy for group ops
    ipapwpolicy:
      ipaadmin_password: MySecret123
      name: ops
      minlife: 7
      maxlife: 49
      history: 5
      priority: 1
      lockouttime: 300
      minlength: 8
      minclasses: 4
      maxfail: 3
      failinterval: 5
```

개별 변수의 의미에 대한 자세한 내용은 암호 정책 속성을 참조하십시오.

3. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory_/new_pwpolicy_present.yml
```

Ansible 플레이북을 사용하여 **IdM**에 **ops** 그룹의 암호 정책이 있는지 확인합니다.

중요

ops 암호 정책의 우선 순위는 ***1*** 로 설정되어 있지만 **global_policy** 암호 정책에는 우선 순위가 설정되어 있지 않습니다. 이러한 이유로 **ops** 정책은 **ops** 그룹에 대해 **global_policy** 를 자동으로 대체하며 즉시 적용됩니다.

global_policy 는 사용자에 대한 그룹 정책이 설정되지 않은 경우 대체 정책으로 작동하며 그룹 정책보다 우선할 수 없습니다.

추가 리소스

- **IdM** 및 플레이북 변수에 대한 암호 정책을 정의하는 데 **Ansible**을 사용하는 방법에 대한 자세한 내용은 **/usr/share/doc/ansible-freeipa/** 디렉터리에 있는 **README- pwpolicy.md Markdown** 파일을 참조하십시오.
- 암호 정책 우선 순위가 **IdM**에서 작동하는 방법에 대한 자세한 내용은 **RHEL 7** 설명서의 암호 정책 우선 순위를 참조하십시오.

17.4. 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 옵션

새 암호에서 동일한 연속 문자의 최대 허용 가능한 수를 지정합니다.

--maxsequence 옵션

새 암호의 최대 단일 문자 시퀀스 길이를 지정합니다. 이러한 시퀀스의 예는 **12345** 또는 **fedcb** 입니다. 이러한 암호는 대부분 단순성 검사를 통과하지 않습니다. 유일한 예외는 시퀀스가 암호의 일부일 때입니다.

--dictcheck 옵션

0이 아닌 경우 가능한 수정 가능한 암호를 사용하여 사전의 단어와 일치하는지 확인합니다. 현재 **libpwquality** 는 **the cracklib** 라이브러리를 사용하여 사전 검사를 수행합니다.

usercheck 옵션

0이 아니면 암호를 수정할 수 있는지 여부를 확인하고 일부 형식으로 사용자 이름을 포함합니다. **3** 자 미만 사용자 이름에 대해서는 수행되지 않습니다.

추가 암호 정책 옵션은 기존 암호에 적용할 수 없습니다. 추가 옵션을 적용하는 경우 IdM은 암호의 최소 문자 수인 **--minlength** 옵션을 6 자로 자동 설정합니다.

참고

RHEL 7 및 **RHEL 8** 서버와 혼합된 환경에서는 **RHEL 8.4** 이상에서 실행되는 서버에만 추가 암호 정책 설정을 적용할 수 있습니다.

추가 리소스:

- IdM 그룹에 추가 암호 정책 적용
- **pwquality(3)** 도움말 페이지

17.5. IDM 그룹에 추가 암호 정책 옵션 적용

이 섹션에서는 IdM(Identity Management)에 추가 암호 정책 옵션을 적용하는 방법을 설명합니다. 이 예제에서는 새 암호에 사용자의 각 사용자 이름이 포함되지 않고 암호에 연속해서 두 개 이상의 동일한 문자가 포함되어 있는지 확인하여 **managers** 그룹에 대한 암호 정책을 강화하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- **managers** 그룹은 IdM에 있습니다.
- **managers** 암호 정책은 IdM에 있습니다.

절차

1. **managers** 그룹의 사용자가 제안한 모든 새 암호에 사용자 이름 검사를 적용합니다.

```
$ ipa pwpolicy-mod --usercheck=True managers
```

참고

암호 정책의 이름을 지정하지 않으면 기본 **global_policy** 가 수정됩니다.

2. **managers** 암호 정책에서 동일한 연속 문자의 최대 수를 **2**로 설정합니다.

```
$ ipa pwpolicy-mod --maxrepeat=2 managers
```

2개 이상의 동일한 연속 문자가 포함된 경우 이제 암호가 허용되지 않습니다. 예를 들어 **eR873mUi111YJQ** 조합은 연속적으로 **3**개의 **1s**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. **test** 사용자를 **managers** 그룹에 추가합니다.

a. **IdM 웹 UI**에서 **Identity → Groups(ID 그룹) User Groups (사용자 그룹)** 를 클릭합니다.

b. **managers** 를 클릭합니다.

c. 추가를 클릭합니다.

d. **Add users into user group 'managers'** 페이지에서 **test_user** 를 확인합니다.

e. **>** 화살표를 클릭하여 사용자를 **Prospective** 열로 이동합니다.

f. 추가를 클릭합니다.

3. 테스트 사용자의 암호를 재설정합니다.

a. **Identity(ID) Users**(사용자) 로 이동합니다.

b. **test_user** 를 클릭합니다.

c. **Actions**(작업) 메뉴에서 **Reset Password**(암호 재설정)를 클릭합니다.

d. 사용자의 임시 암호를 입력합니다.

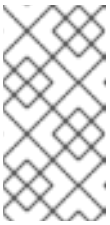
4. 명령줄에서 **test_user** 에 대한 **Kerberos** 티켓 부여 티켓(**TGT**)을 가져옵니다.

```
$ kinit test_user
```

a. 임시 암호를 입력합니다.

b. 시스템에서 암호를 변경해야 함을 알려줍니다. 사용자 이름이 **test_user** 인 암호를 입력합니다.

```
Password expired. You must change it now.
Enter new password:
Enter it again:
```



참고

Kerberos에는 세부적인 오류 암호 정책 보고가 없으며 특정 경우에는 암호가 거부된 명확한 이유를 제공하지 않습니다.

c. 시스템은 입력한 암호가 거부되었음을 알려줍니다. 연속해서 **3**개 이상의 동일한 문자를 포함하는 암호를 입력합니다.

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

d. 시스템은 입력한 암호가 거부되었음을 알려줍니다. **managers** 암호 정책의 기준을 충족하는 암호를 입력합니다.

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

5. 가져온 **TGT** 보기:

```
$ klist
Ticket cache: KCM:0:33945
Default principal: test_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
07/07/2021 12:44:44  07/08/2021 12:44:44
krbtgt@IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
```

managers 암호 정책이 **managers** 그룹의 사용자에 대해 올바르게 작동합니다.

추가 리소스

- IdM의 추가 암호 정책

18장. IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

18.1. IDM 클라이언트에서 SUDO 액세스

시스템 관리자는 루트가 아닌 사용자가 일반적으로 root 사용자에게 예약된 관리 명령을 실행할 수 있도록 sudo 액세스 권한을 부여할 수 있습니다. 따라서 사용자가 일반적으로 root 사용자로 예약된 관리 명령을 수행해야 하는 경우 sudo 를 사용하여 해당 명령 앞에 추가합니다. 암호를 입력한 후에는 루트 사용자인 것처럼 명령이 실행됩니다. 데이터베이스 서비스 계정과 같은 다른 사용자 또는 그룹으로 sudo 명령을 실행하려면 sudo 규칙에 대한 *RunAs 별칭을* 구성할 수 있습니다.

RHEL(Red Hat Enterprise Linux) 8 호스트가 IdM(Identity Management) 클라이언트에 등록된 경우 다음 방법으로 호스트에서 수행할 수 있는 IdM 사용자를 정의하는 sudo 규칙을 지정할 수 있습니다.

- 로컬로 /etc/sudoers 파일에서
- IdM에서 중앙 집중식으로

이 섹션에서는 CLI(명령줄 인터페이스) 및 IdM 웹 UI를 사용하여 IdM 클라이언트에 대한 중앙 sudo 규칙을 생성하는 방법을 설명합니다.

RHEL 8.4 이상에서는 UNIX 기반 운영 체제가 Kerberos 서비스에 액세스하고 인증하는 기본 방법인 GSSAPI(Generic Security Service Application Programming Interface)를 사용하여 sudo 에 대해 암호 없는 인증을 구성할 수도 있습니다. pam_sss_gss.so PAM(Pluggable Authentication Module)을 사용하여 SSSD 서비스를 통해 GSSAPI 인증을 호출할 수 있으므로 사용자는 유효한 Kerberos 티켓을 사용하여 sudo 명령을 인증할 수 있습니다.

추가 리소스

- RHEL 8 호스트에서 로컬 sudo 규칙 생성에 대한 자세한 내용은 sudo 액세스 관리를 참조하십시오.

18.2. CLI를 사용하여 IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

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

예를 들어 idm_user 계정에 idm_ sbin/reboot 명령을 실행할 수 있는 권한을 부여하려면 idm_user_reboot sudo 규칙을 만들려면 다음 절차를 완료합니다 .

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- IdM에서 idm_user 에 대한 사용자 계정을 생성하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. CLI를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄을 사용하여 사용자 추가를 참조하십시오.
- idm client 호스트에 로컬 idm_user 계정이 없습니다. idm_user 사용자는 로컬 /etc/passwd 파일에 나열되지 않습니다.

절차

1. IdM 관리자로 Kerberos 티켓을 검색합니다.

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

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

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

3. idm_user_reboot 라는 sudo 규칙을 만듭니다.

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

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

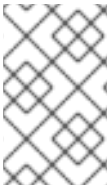
```
[root@idmclient ~]# ipa sudorule-add-allow-command idm_user_reboot --sudocmds
'/usr/sbin/reboot'
  Rule name: idm_user_reboot
  Enabled: TRUE
  Sudo Allow Commands: /usr/sbin/reboot
-----------------------
Number of members added 1
-----------------------
```

5. IdM idm client 호스트에 idm_user_reboot 규칙을 적용합니다.

```
[root@idmclient ~]# ipa sudorule-add-host idm_user_reboot --hosts
idmclient.idm.example.com
Rule name: idm_user_reboot
Enabled: TRUE
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-----------------------
Number of members added 1
-----------------------
```

6. idm_user 계정을 idm_user_reboot 규칙에 추가합니다.

```
[root@idmclient ~]# ipa sudorule-add-user idm_user_reboot --users idm_user
Rule name: idm_user_reboot
Enabled: TRUE
Users: idm_user
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-----------------------
Number of members added 1
-----------------------
```



참고

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

검증 단계

1. idmclient 호스트에 idm _user 계정으로 로그인합니다.

2. idm_user 계정이 수행할 수 있는 sudo 규칙을 표시합니다.

```
[idm_user@idmclient ~]$ sudo -l
```

```
Matching Defaults entries for idm_user on idmclient:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
XAUTHORITY KRB5CCNAME",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User idm_user may run the following commands on idmclient:
    (root) /usr/sbin/reboot
```

3. sudo 를 사용하여 시스템을 재부팅합니다. 메시지가 표시되면 idm_user 의 암호를 입력합니다.

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

18.3. IDM 웹 UI를 사용하여 IDM 클라이언트에서 IDM 사용자에게 SUDO 액세스 권한 부여

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

idm_user_reboot sudo 규칙을 생성하여 idm _user 계정에 idm client 시스템에서 /usr/sbin/reboot 명령을 실행할 수 있는 권한을 부여하려면 이 절차를 완료합니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- IdM에서 idm_user 에 대한 사용자 계정을 생성하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. 명령줄 인터페이스를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄 을 사용하여 사용자 추가를 참조하십시오.
- idm client 호스트에 로컬 idm_user 계정이 없습니다. idm_user 사용자는 로컬 /etc/passwd 파일에 나열되지 않습니다.

절차

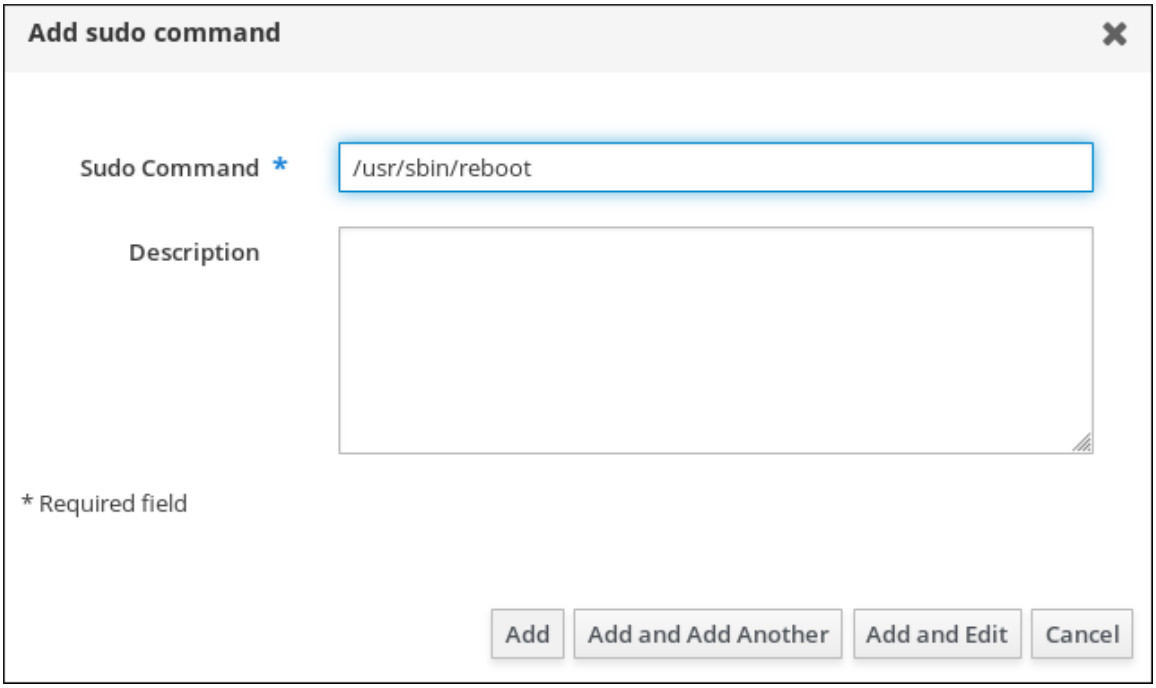
1. /usr/sbin/reboot 명령을 sudo 명령의 IdM 데이터베이스에 추가합니다.

   a. 정책 → Sudo → Sudo 명령으로 이동합니다.

   b. 오른쪽 상단 모서리에서 Add (추가)를 클릭하여 Add sudo 명령 대화 상자를 엽니다.

   c. sudo:/usr/sbin/reboot 를 사용하여 사용자가 수행할 수 있는 명령을 입력합니다.

   그림 18.1. IdM sudo 명령 추가



   d. 추가를 클릭합니다.

2. 새 sudo 명령 항목을 사용하여 idm_user가 idm client 시스템을 재부팅할 수 있도록 sudo 규칙을 생성합니다.

   a. 정책 → Sudo → Sudo 규칙으로 이동합니다.

   b. 오른쪽 상단 모서리에서 Add(추가)를 클릭하여 Add sudo rule(sudo 규칙 추가) 대화 상자를 엽니다.

c. **sudo 규칙의 이름을 idm_user_reboot 로 입력합니다.**

d. **Add and Edit(추가 및 편집)를 클릭합니다.**

e. **사용자를 지정합니다.**

i. **who(사용자) 섹션에서 Specified Users and Groups(지정된 사용자 및 그룹) 라디오 단추를 선택합니다.**

ii. **User 카테고리에서 규칙이 하위 섹션에 적용되는 경우 Add(추가)를 클릭하여 Add users into sudo rule "idm_user_reboot" 대화 상자를 엽니다.**

iii. **Add users into sudo rule "idm_user_reboot" 대화 상자의 Available 열에서 idm_user 확인란을 선택하고 Prospective 열로 이동합니다.**

iv. **추가를 클릭합니다.**

f. **호스트를 지정합니다.**

i. **Access this host (이 호스트에 액세스) 섹션에서 Specified Hosts and Groups(지정된 호스트 및 그룹) 라디오 단추를 선택합니다.**

ii. **호스트 범주에서 이 규칙이 하위 섹션에 적용되는 경우 Add(추가)를 클릭하여 Add hosts into sudo rule "idm_user_reboot" 대화 상자를 엽니다.**

iii. **Add hosts to sudo rule "idm_user_reboot" 대화 상자의 Available 열에서 idmclient.idm.example.com 확인란을 선택하고 Prospective 열로 이동합니다.**

iv. **추가를 클릭합니다.**

g. **명령을 지정합니다.**

i. **Command** 카테고리에서 규칙이 **Run Commands(**명령 실행**)** 섹션의 하위 섹션에 적용되는 경우 **Specified Commands and Groups(**지정된 명령**)** 및 **Groups (**그룹**)** 라디오 버튼을 선택합니다.

ii. **Sudo Allow Commands (Sudo Allow Commands)** 하위 섹션에서 **Add allow sudo** 명령을 **sudo rule "idm_user_reboot"** 대화 상자에 엽니다.

iii. **Add allow sudo** 명령을 **sudo** 규칙 **"idm_user_reboot"** 대화 상자의 **Available** 열에서 **/usr/sbin/reboot** 확인란을 선택하고 **Prospective** 열로 이동합니다.

iv. **Add(**추가 **)**를 클릭하여 **idm_sudo_reboot** 페이지로 돌아갑니다.

IdM sudo 규칙 추가

+ image::IdM-sudo-rule-WebUI.png[추가된 **sudo** 규칙 개요의 스크린샷입니다. 규칙이 적용되는 사용자 테이블이 있는 **"Who"** 섹션이 있습니다. 규칙이 적용되는 호스트 테이블이 있는 **"**이 호스트 액세스**"** 섹션이 있습니다. 규칙에 관련된 명령 테이블이 있는 **"Run Commands"** 섹션이 있습니다.**]**

a. 왼쪽 상단 모서리에서 **Save(**저장 **)**를 클릭합니다.

새 규칙은 기본적으로 활성화되어 있습니다.

참고

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

검증 단계

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

2. **sudo** 를 사용하여 시스템을 재부팅합니다. 메시지가 표시되면 **idm_user** 의 암호를 입력합니다.

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

sudo 규칙이 올바르게 구성되면 시스템이 재부팅됩니다.

18.4. IDM 클라이언트에서 명령을 서비스 계정으로 실행하는 CLI에서 SUDO 규칙 생성

IdM에서 *RunAs 별칭으로* 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 할 수 있습니다.

이 예제를 사용하여 run_third-party-app_report 라는 명령줄에 sudo 규칙을 생성하여 idm_user 계정이 idm client 호스트에서 third partyapp 서비스 계정으로 /opt/third-party-app /bin/report 명령을 실행할 수 있습니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- IdM에서 idm_user 에 대한 사용자 계정을 생성하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. CLI를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄을 사용하여 사용자 추가를 참조하십시오.
- idm client 호스트에 로컬 idm_user 계정이 없습니다. idm_user 사용자는 로컬 /etc/passwd 파일에 나열되지 않습니다.
- idmclient 호스트에 third-party-app 이라는 사용자 지정 애플리케이션이 설치되어 있어야 합니다.
- 타사-app 애플리케이션의 report 명령은 /opt/third-party-app/bin/report 디렉터리에 설치됩니다.
- 타사 -app 애플리케이션에 대한 명령을 실행하기 위해 thirdpartyapp 이라는 로컬 서비스 계정을 생성했습니다.

절차

1. IdM 관리자로 Kerberos 티켓을 검색합니다.

-

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

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

```
[root@idmclient ~]# ipa sudocmd-add /opt/third-party-app/bin/report
----------------------------------------------------
Added Sudo Command "/opt/third-party-app/bin/report"
----------------------------------------------------
  Sudo Command: /opt/third-party-app/bin/report
```

3. **run_third-party-app_report** 라는 **sudo** 규칙을 만듭니다.

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

4. **users =<*user*>** 옵션을 사용하여 **sudorule-add-runasuser** 명령에 **RunAs** 사용자를 지정합니다.

```
[root@idmclient ~]# ipa sudorule-add-runasuser run_third-party-app_report --
users=thirdpartyapp
  Rule name: run_third-party-app_report
  Enabled: TRUE
  RunAs External User: thirdpartyapp
------------------------
Number of members added 1
------------------------
```

로컬 서비스 계정 또는 **Active Directory** 사용자와 같은 사용자(또는 **--groups=*** 옵션으로 지정된 그룹)는 **IdM** 외부에 있을 수 있습니다. 그룹 이름에 **%** 접두사를 추가하지 마십시오.

5. **/opt/third-party-app/bin/report** 명령을 **idm_user_reboot** 규칙에 추가합니다.

```
[root@idmclient ~]# ipa sudorule-add-allow-command run_third-party-app_report --
sudocmds '/opt/third-party-app/bin/report'
Rule name: run_third-party-app_report
Enabled: TRUE
Sudo Allow Commands: /opt/third-party-app/bin/report
RunAs External User: thirdpartyapp
```

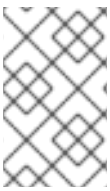
```
------------------------
Number of members added 1
------------------------
```

6. run_third-party-app_report 규칙을 IdM idmclient 호스트에 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.
```

18.5. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 IDM WEBUI에서 SUDO 규칙 생성

IdM에서 *RunAs 별칭으로* 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 할 수 있습니다.

이 예제를 사용하여 IdM WebUI에서 run_third-party-app_report 라는 sudo 규칙을 생성하여 idm_user 계정이 idmclient 호스트에서 third partyapp 서비스 계정으로 /opt/third-party-app /bin/report 명령을 실행할 수 있도록 합니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- IdM에서 idm_user 에 대한 사용자 계정을 생성하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. CLI를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄을 사용하여 사용자 추가를 참조하십시오.

- idm client 호스트에 로컬 idm_user 계정이 없습니다. idm_user 사용자는 로컬 /etc/passwd 파일에 나열되지 않습니다.

- idmclient 호스트에 third-party-app 이라는 사용자 지정 애플리케이션이 설치되어 있어야 합니다.

- 타사-app 애플리케이션의 report 명령은 /opt/third-party-app/bin/report 디렉터리에 설치됩니다.

- 타사 -app 애플리케이션에 대한 명령을 실행하기 위해 thirdpartyapp 이라는 로컬 서비스 계정을 생성했습니다.

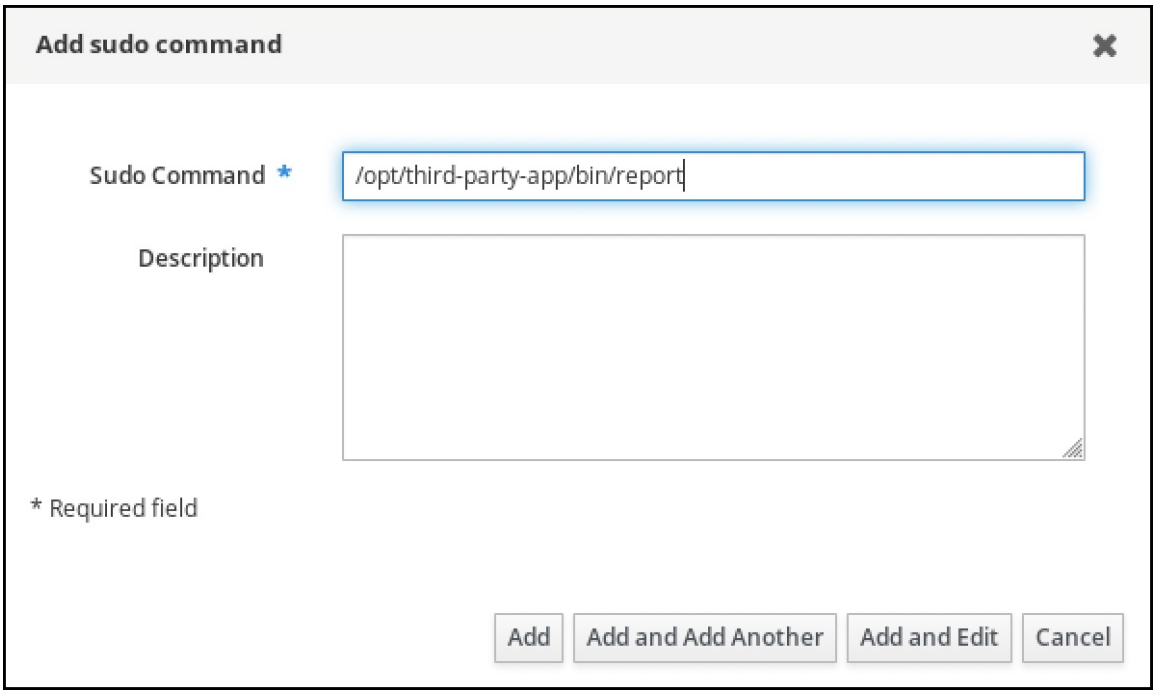
절차

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

   a. 정책 → Sudo → Sudo 명령으로 이동합니다.

   b. 오른쪽 상단 모서리에서 Add (추가)를 클릭하여 Add sudo 명령 대화 상자를 엽니다.

   c. /opt/third-party-app/bin/report 명령을 입력합니다.



d.

추가를 클릭합니다.

2.

새 sudo 명령 항목을 사용하여 새 sudo 규칙을 생성합니다.

a.

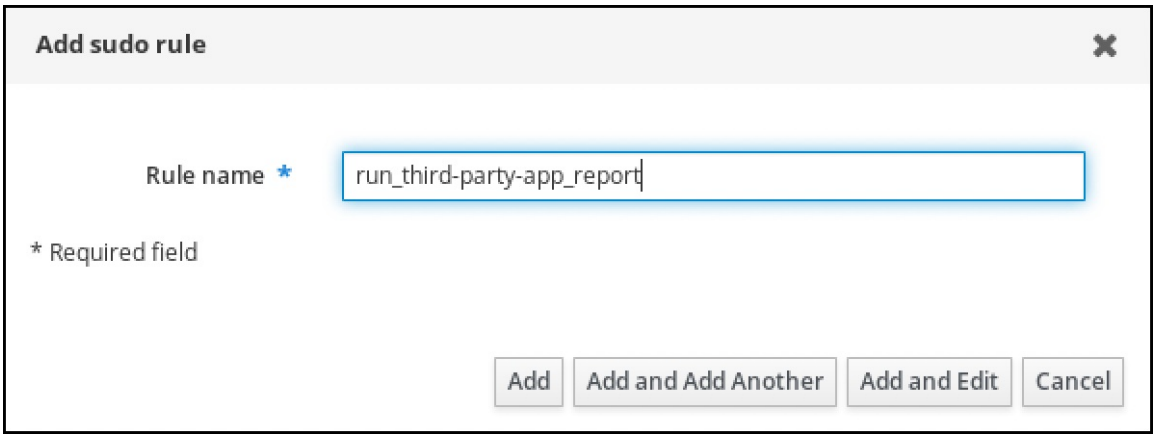
정책 → Sudo → Sudo 규칙으로 이동합니다.

b.

오른쪽 상단 모서리에서 Add(추가)를 클릭하여 Add sudo rule(sudo 규칙 추가) 대화 상자를 엽니다.

c.

sudo 규칙의 이름을 run_third-party-app_report 로 입력합니다.



d.

Add and Edit(추가 및 편집)를 클릭합니다.

e.

사용자를 지정합니다.

i.

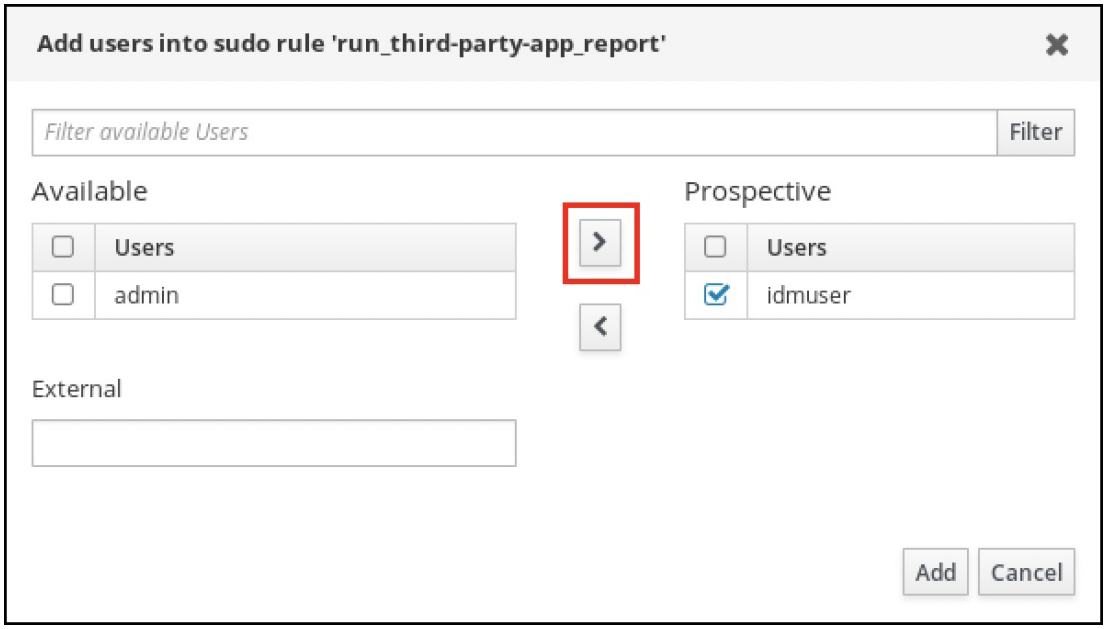
who(사용자) 섹션에서 Specified Users and Groups(지정된 사용자 및 그룹) 라디오 단추를 선택합니다.

ii.

사용자 범주에서 규칙이 하위 섹션에 적용되는 경우 Add(추가)를 클릭하여 Add users into sudo rule "run_ third-party-app_report" 대화 상자를 엽니다.

iii.

Add users into sudo rule "run_ third-party-app_report" 대화 상자의 Available 열에서 idm_user 확인란을 선택하고 Prospective 열로 이동합니다.



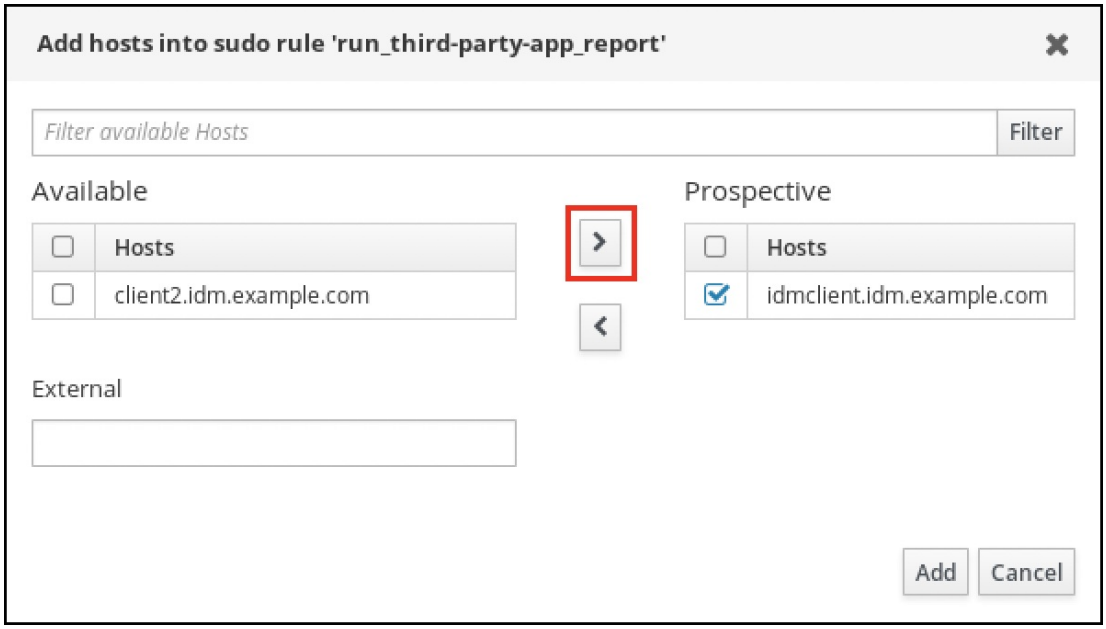
iv. 추가를 클릭합니다.

f. 호스트를 지정합니다.

i. **Access this host** (이 호스트에 액세스) 섹션에서 **Specified Hosts and Groups**(지정된 호스트 및 그룹) 라디오 단추를 선택합니다.

ii. 이 규칙이 하위 섹션에 적용되는 호스트 카테고리에서 **Add**(추가)를 클릭하여 **Add hosts into sudo rule "run_ third-party-app_report"** 대화 상자를 엽니다.

iii. **Add hosts into sudo rule "run_ third-party-app_report"** 대화 상자의 **Available** 열에서 **idmclient.idm.example.com** 확인란을 선택하고 **Prospective** 열로 이동합니다.



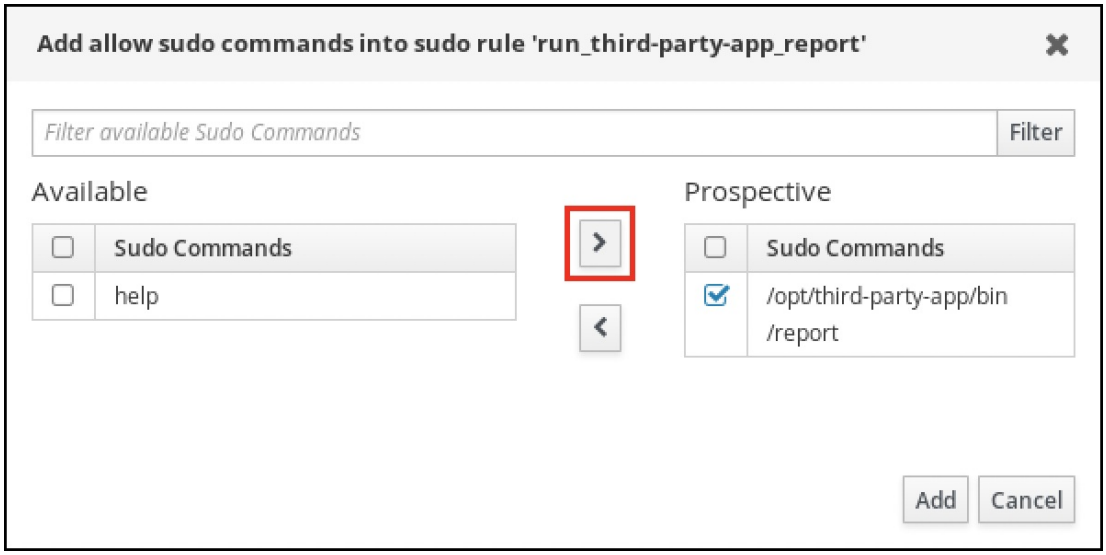
iv. 추가를 클릭합니다.

g. 명령을 지정합니다.

i. **Command** 카테고리에서 규칙이 **Run Commands**(명령 실행) 섹션의 하위 섹션에 적용되는 경우 **Specified Commands and Groups**(지정된 명령) 및 **Groups** (그룹) 라디오 버튼을 선택합니다.

ii. **Sudo Allow Commands** (Sudo Allow Commands) 하위 섹션에서 **Add allow sudo** 명령을 **sudo** 규칙 **"run_ third-party-app_report"** 대화 상자에 엽니다.

iii. **Add allow sudo command to sudo rule "run_ third-party-app_report"** 대화 상자의 **Available** 열에서 **/opt/third-party-app/bin/report** 확인란을 선택하고 **Prospective** 열로 이동합니다.



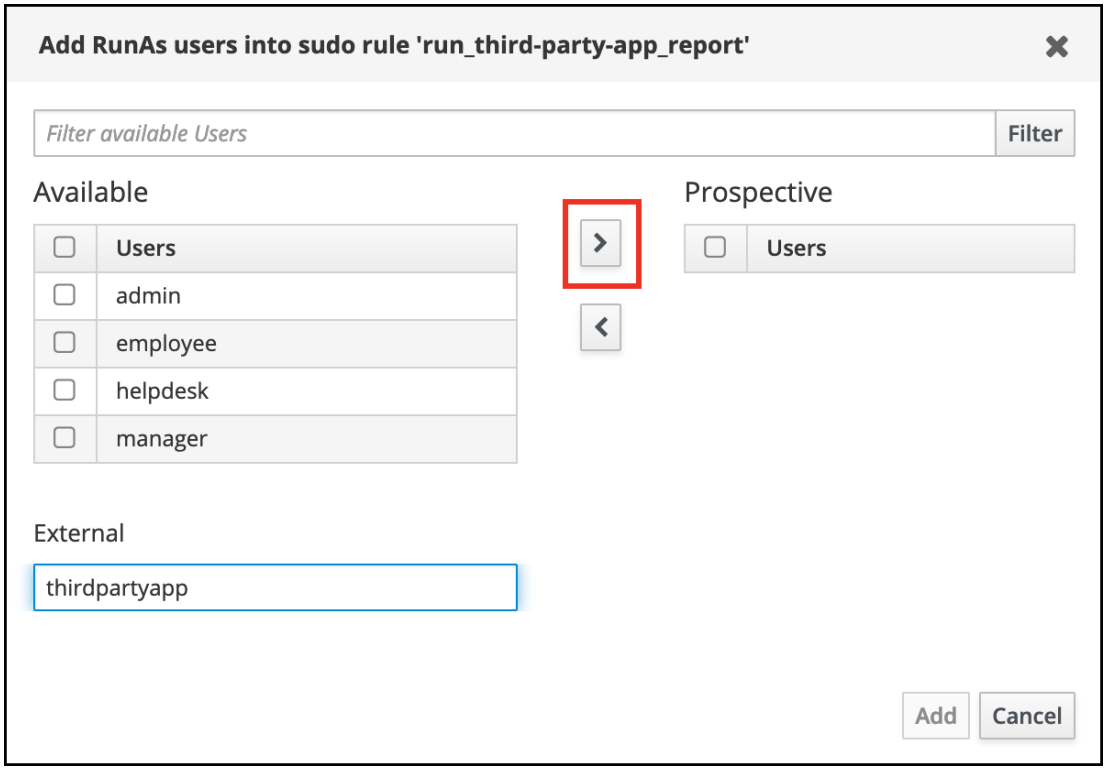
iv. **Add(추가)**를 클릭하여 **run_third-party-app_report** 페이지로 돌아갑니다.

h. **RunAs** 사용자를 지정합니다.

i. **As whom**(사용자 및 그룹 지정) 섹션에서 **Specified Users and Groups**(지정된 사용자 및 그룹) 라디오 단추를 선택합니다.

ii. **RunAs Users** (사용자 실행) 하위 섹션에서 **Add**(추가)를 클릭하여 **Add RunAs** 사용자를 **sudo** 규칙 **"run_ third-party-app_report"** 대화 상자에 엽니다.

iii. **Add RunAs users into sudo rule "run_ third-party-app_report"** 대화 상자에서 외부 상자에 **thirdpartyapp** 서비스 계정을 입력하고 **Prospective** 열로 이동합니다.

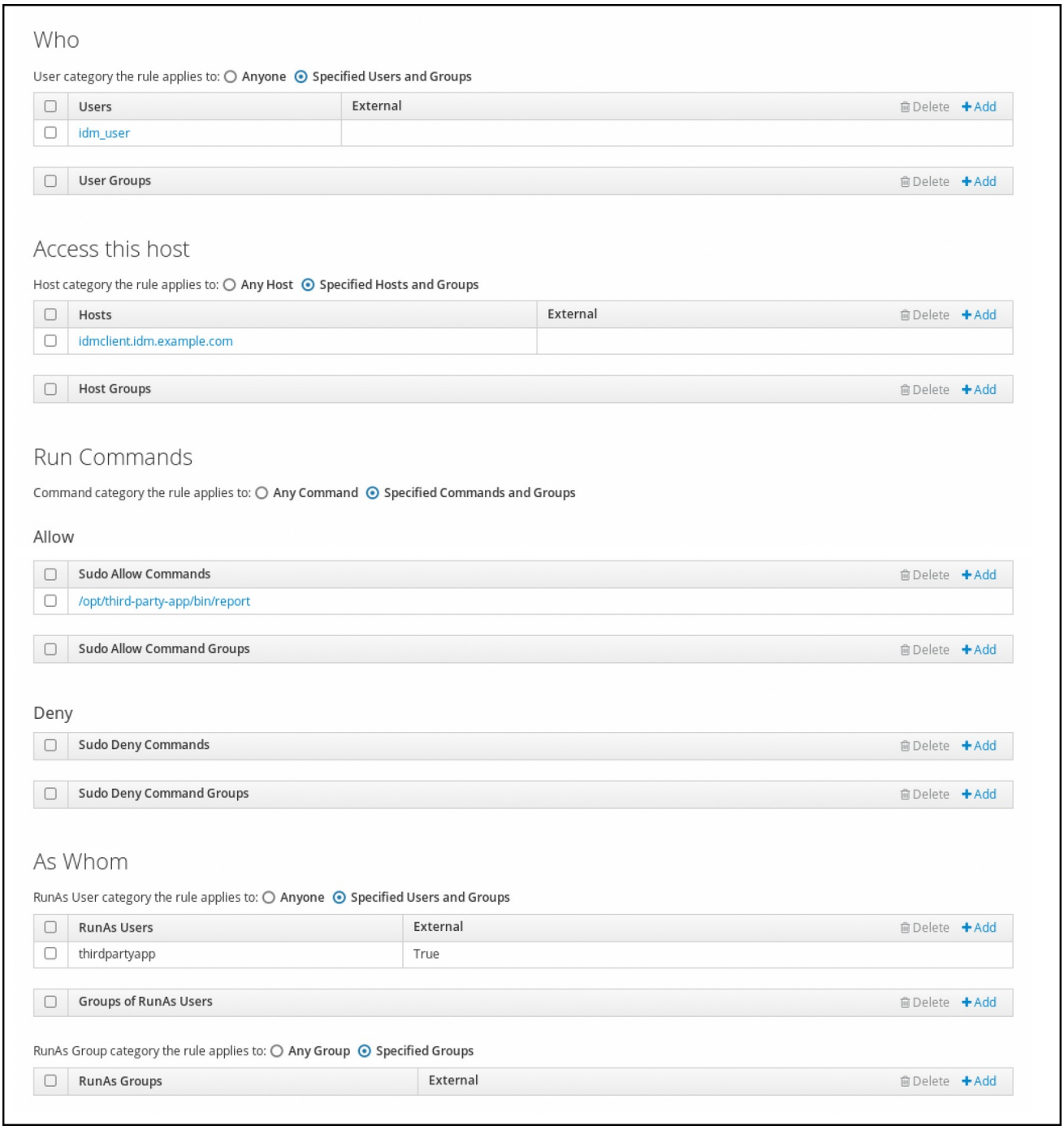


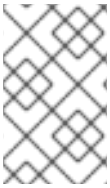
iv. **Add(추가)를 클릭하여 run_third-party-app_report 페이지로 돌아갑니다.**

i. **왼쪽 상단 모서리에서 Save(저장)를 클릭합니다.**

새 규칙은 기본적으로 활성화되어 있습니다.

그림 18.2. sudo 규칙의 세부 정보





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.
```

18.6. IDM 클라이언트에서 SUDO에 대해 GSSAPI 인증 활성화

다음 절차에서는 pam_sss_gss.so PAM 모듈을 통해 sudo 및 sudo -i 명령에 대해 IdM 클라이언트에서 GSSAPI 인증 활성화를 설명합니다. 이 구성을 사용하면 IdM 사용자가 Kerberos 티켓을 사용하여 sudo 명령에 인증할 수 있습니다.

사전 요구 사항

- IdM 호스트에 적용되는 IdM 사용자에 대한 sudo 규칙을 생성했습니다. 이 예제에서는 idm_user 계정에 idm_ sbin/reboot 명령을 실행할 수 있는 권한을 부여하는 idm _user_ reboot sudo 규칙을 생성했습니다.

- idmclient 호스트는 RHEL 8.4 이상을 실행하고 있습니다.

- /etc/ pam.d / 디렉토리에서 /etc/sssd/sssd.conf 파일과 PAM 파일을 수정하려면 root 권한이 필요합니다.

절차

1. /etc/sssd/sssd.conf 구성 파일을 엽니다.

2. 다음 항목을 [domain/*<domain_name>*] 섹션에 추가합니다.

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

3. /etc/sssd/sssd.conf 파일을 저장하고 닫습니다.

4. SSSD 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.

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

5. /etc/pam.d/sudo PAM 구성 파일을 엽니다.

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

```
#%PAM-1.0
auth sufficient pam_sss_gss.so
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    include      system-auth
```

7. /etc/pam.d/sudo 파일을 저장하고 닫습니다.

8. /etc/pam.d/sudo-i PAM 구성 파일을 엽니다.

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

```
#%PAM-1.0
auth sufficient pam_sss_gss.so
auth       include      sudo
account    include      sudo
password   include      sudo
session    optional     pam_keyinit.so force revoke
session    include      sudo
```

10. **/etc/pam.d/sudo-i** 파일을 저장하고 닫습니다.

검증 단계

1. **idm_user** 계정으로 호스트에 로그인합니다.

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

2. 티켓이 **idm_user** 계정으로 티켓을 부여했는지 확인합니다.

```
[idmuser@idmclient ~]$ klist
Ticket cache: KCM:1366201107
Default principal: idm_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
01/08/2021 09:11:48  01/08/2021 19:11:48
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
 renew until 01/15/2021 09:11:44
```

3. *(선택 사항)* **idm_user** 계정에 대한 **Kerberos** 자격 증명이 없는 경우 현재 **Kerberos** 자격 증명을 삭제하고 올바른 자격 증명을 요청하십시오.

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

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

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

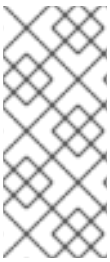
```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
```

추가 리소스

- IdM 웹 UI를 사용하여 IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한 부여
- CLI를 사용하여 IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한 부여.
- pam_sss_gss (8) 도움말 페이지
- sssd.conf (5) 도움말 페이지

18.7. GSSAPI 인증 활성화 및 IDM 클라이언트에서 SUDO에 대한 KERBEROS 인증 표시기 적용

다음 절차에서는 pam_sss_gss.so PAM 모듈을 통해 sudo 및 sudo -i 명령에 대해 IdM 클라이언트에서 GSSAPI 인증 활성화를 설명합니다. 또한 스마트 카드로 로그인한 사용자만 Kerberos 티켓을 사용하여 해당 명령에 인증됩니다.



참고

이 절차를 템플릿으로 사용하여 다른 PAM 인식 서비스에 대해 SSSD를 사용하여 GSSAPI 인증을 구성하고, Kerberos 티켓에 특정 인증 표시기가 연결된 사용자에게만 액세스를 제한할 수 있습니다.

사전 요구 사항

- IdM 호스트에 적용되는 IdM 사용자에 대한 sudo 규칙을 생성했습니다. 이 예제에서는 idm_user 계정에 idm_ sbin/reboot 명령을 실행할 수 있는 권한을 부여하는 idm _user_ reboot sudo 규칙을 생성했습니다.
- idmclient 호스트에 대해 스마트 카드 인증을 구성했습니다.
- idmclient 호스트는 RHEL 8.4 이상을 실행하고 있습니다.
- /etc/ pam.d / 디렉토리에서 /etc/sssd/sssd.conf 파일과 PAM 파일을 수정하려면 root 권한이 필요합니다.

절차

1. **/etc/sssd/sssd.conf** 구성 파일을 엽니다.

2. 다음 항목을 **[domain/*<domain_name>*]** 섹션에 추가합니다.

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

3. **/etc/sssd/sssd.conf** 파일을 저장하고 닫습니다.

4. **SSSD** 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.

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

5. **/etc/pam.d/sudo PAM** 구성 파일을 엽니다.

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

```
#%PAM-1.0
auth sufficient pam_sss_gss.so
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    include      system-auth
```

7. **/etc/pam.d/sudo** 파일을 저장하고 닫습니다.

8. **/etc/pam.d/sudo-i PAM** 구성 파일을 엽니다.

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

```
#%PAM-1.0
auth sufficient pam_sss_gss.so
auth       include      sudo
account    include      sudo
```

```
password   include     sudo
session    optional    pam_keyinit.so force revoke
session    include     sudo
```

10. **/etc/pam.d/sudo-i** 파일을 저장하고 닫습니다.

검증 단계

1. **idm_user** 계정으로 호스트에 로그인하고 스마트 카드로 인증합니다.

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

2. 스마트 카드 사용자로 티켓이 제공된지 확인합니다.

```
[idm_user@idmclient ~]$ klist
Ticket cache: KEYRING:persistent:1358900015:krb_cache_TObtNMd
Default principal: idm_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
02/15/2021 16:29:48  02/16/2021 02:29:48
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
 renew until 02/22/2021 16:29:44
```

3. **idm_user** 계정이 수행할 수 있는 **sudo** 규칙을 표시합니다.

```
[idm_user@idmclient ~]$ sudo -l
Matching Defaults entries for idmuser on idmclient:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
XAUTHORITY KRB5CCNAME",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User idm_user may run the following commands on idmclient:
    (root) /usr/sbin/reboot
```

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

-

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

추가 리소스

- 스마트 카드 인증을 위한 ID 관리 구성
- Kerberos 인증 표시기
- IdM 웹 UI를 사용하여 IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한 부여
- CLI를 사용하여 IdM 클라이언트의 IdM 사용자에게 sudo 액세스 권한 부여.
- pam_sss_gss (8) 도움말 페이지
- sssd.conf (5) 도움말 페이지

18.8. PAM 서비스에 대한 GSSAPI 인증을 제어하는 SSSD 옵션

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

pam_gssapi_services

SSSD를 사용한 GSSAPI 인증은 기본적으로 비활성화되어 있습니다. 이 옵션을 사용하여 pam_ss_gss.so PAM 모듈을 사용하여 GSSAPI 인증을 시도할 수 있는 PAM 서비스의 쉼표로 구분된 목록을 지정할 수 있습니다. GSSAPI 인증을 명시적으로 비활성화하려면 이 옵션을 - 로 설정합니다.

pam_gssapi_indicators_map

이 옵션은 IdM(Identity Management) 도메인에만 적용됩니다. 이 옵션을 사용하여 서비스에 대한 PAM 액세스 권한을 부여하는 데 필요한 Kerberos 인증 표시기를 나열합니다. 쌍은 *<PAM_service> :_<* required_authentication_indicator>_ 형식이어야 합니다.

유효한 인증 표시기는 다음과 같습니다.

- 이중 인증을 위한 **OTP**
- **RADIUS** 인증을 위한 준비
- **PKINIT**, 스마트 카드 또는 인증서 인증을 위한 **Pk init**
- 강화된 암호를 위한 강화

pam_gssapi_check_upn

이 옵션은 활성화되어 있으며 기본적으로 **true** 로 설정됩니다. 이 옵션을 활성화하면 **SSSD** 서비스에 사용자 이름이 **Kerberos** 자격 증명과 일치해야 합니다. **false**인 경우 **pam_sss_gss.so PAM** 모듈은 필수 서비스 티켓을 가져올 수 있는 모든 사용자를 인증합니다.

예

다음 옵션을 사용하면 **sudo** 및 **sudo -i** 서비스에 **Kerberos** 인증을 사용하려면 **sudo** 사용자가 일회성 암호로 인증해야 하며 사용자 이름은 **Kerberos** 주체와 일치해야 합니다. 이러한 설정은 **[pam]** 섹션에 있으므로 모든 도메인에 적용됩니다.

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

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

idm.example.com 도메인의 경우

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

- 일치하는 사용자 이름과 **Kerberos** 주체 적용.

ad.example.com 도메인의 경우

- **sudo** 서비스에 대해서만 **GSSAPI** 인증을 활성화합니다.

- 일치하는 사용자 이름과 주체를 적용하지 마십시오.

```
[domain/idm.example.com]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:pkinit, sudo-i:otp
pam_gssapi_check_upn = true
...

[domain/ad.example.com]
pam_gssapi_services = sudo
pam_gssapi_check_upn = false
...
```

추가 리소스

- Kerberos 인증 표시기

18.9. SUDO에 대한 GSSAPI 인증 문제 해결

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

사전 요구 사항

- **sudo** 서비스에 대해 **GSSAPI** 인증을 활성화했습니다. IdM 클라이언트에서 sudo에 대한 GSSAPI 인증 활성화를 참조하십시오.

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

절차

-

다음 오류가 표시되면 **Kerberos** 서비스에서 호스트 이름을 기반으로 서비스 티켓에 대한 올바른 영역을 확인할 수 없습니다.

```
Server not found in Kerberos database
```

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

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

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

- 다음 오류가 표시되면 **Kerberos** 인증 정보가 없습니다.

```
No Kerberos credentials available
```

이 경우 **kinit** 유틸리티를 사용하여 **Kerberos** 인증 정보를 검색하거나 **SSSD**로 인증합니다.

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

- **/var/log/sssd/sssd_pam.log** 로그 파일에 다음 중 하나가 표시되면 **Kerberos** 자격 증명이 현재 로그인한 사용자의 사용자 이름과 일치하지 않습니다.

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

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

이 경우 **SSSD**로 인증했는지 또는 **/etc/sssd/sssd.conf** 파일에서 **pam_gssapi_check_upn** 옵션을 비활성화하는 것이 좋습니다.

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

pam_gssapi_check_upn = false
```

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

- PAM 파일의 모든 pam_sss_gss.so 항목(예: /etc/pam.d/ sudo 및 /etc/pam.d/sudo -i)의 끝에 debug 옵션을 추가합니다.

```
[root@idm-client ~]# cat /etc/pam.d/sudo
#%PAM-1.0
auth       sufficient   pam_sss_gss.so   debug
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    include      system-auth
```

```
[root@idm-client ~]# cat /etc/pam.d/sudo-i
#%PAM-1.0
auth       sufficient   pam_sss_gss.so   debug
auth       include      sudo
account    include      sudo
password   include      sudo
session    optional     pam_keyinit.so force revoke
session    include      sudo
```

- pam_sss_gss.so 모듈로 인증을 시도하고 콘솔 출력을 검토합니다. 이 예제에서 사용자에게 Kerberos 자격 증명이 없습니다.

```
[idm-user@idm-client ~]$ sudo ls -l /etc/sssd/sssd.conf
pam_sss_gss: Initializing GSSAPI authentication with SSSD
pam_sss_gss: Switching euid from 0 to 1366201107
pam_sss_gss: Trying to establish security context
pam_sss_gss: SSSD User name: idm-user@idm.example.com
pam_sss_gss: User domain: idm.example.com
pam_sss_gss: User principal:
pam_sss_gss: Target name: host@idm.example.com
pam_sss_gss: Using ccache: KCM:
pam_sss_gss: Acquiring credentials, principal name will be derived
pam_sss_gss: Unable to read credentials from [KCM:] [maj:0xd0000,
min:0x96c73ac3]
pam_sss_gss: GSSAPI: Unspecified GSS failure.  Minor code may provide more
information
pam_sss_gss: GSSAPI: No credentials cache found
pam_sss_gss: Switching euid from 1366200907 to 0
pam_sss_gss: System error [5]: Input/output error
```

18.10. ANSIBLE 플레이북을 사용하여 IDM 클라이언트에서 IDM 사용자에 대한 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 다음 절차를 완료합니다. 규칙은 idm_user 에 idmclient 시스템에서 /usr/sbin/reboot 명령을 실행할 수 있는 권한을 부여합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.
- IdM 관리자 암호를 알고 있습니다.
- IdM에 idm_user 의 사용자 계정이 있는지 확인하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. 명령줄 인터페이스를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄 을 사용하여 사용자 추가를 참조하십시오.
- idm client 에 로컬 idm_user 계정이 없습니다. idm_user 사용자는 idmclient 의 /etc/passwd 파일에 나열되지 않습니다.

절차

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

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

2. 하나 이상의 sudo 명령을 추가합니다.

   a. sudo 명령의 IdM 데이터베이스에 /usr/sbin/ reboot 명령이 있는지 확인하는 ensure-reboot -sudocmd-is-present.yml Ansible 플레이북을 생성합니다. 이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/playbooks/sudocmd/ensure-sudocmd-is-present.yml 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
  # Ensure sudo command is present
  - ipasudocmd:
```

```
    ipaadmin_password: MySecret123
    name: /usr/sbin/reboot
    state: present
```

b. 플레이북을 실행합니다.

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

3. 명령을 참조하는 **sudo** 규칙을 생성합니다.

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

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

  tasks:
  # Ensure a sudorule is present granting idm_user the permission to run
/usr/sbin/reboot on idmclient
  - ipasudorule:
      ipaadmin_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** 서버에서 **IdM** 서버가 작동하는지 확인하는 **sudo** 규칙이 **idmclient** 에서 작동하는지 테스트합니다. 서버에서 변경한 사항이 클라이언트에 적용되는 데 몇 분이 걸릴 수 있습니다.

1. idmclient에 idm _user 로 로그인합니다.

2. sudo 를 사용하여 시스템을 재부팅합니다. 메시지가 표시되면 idm_user 의 암호를 입력합니다.

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

sudo 가 올바르게 구성되면 시스템이 재부팅됩니다.

추가 자료

- 플레이북 변수에 대한 설명을 포함하는 Ansible 플레이북을 사용하여 sudo 명령, 명령 그룹 및 규칙을 적용하는 방법에 대한 자세한 내용은 README-sudocmd.md, README-sudocmdgroup.md 및 /usr/share/doc/ansible-freeipa/ 디렉터리에 있는 README-sudorule.md Markdown 파일을 참조하십시오.

19장. ANSIBLE 플레이북을 사용하여 IDM에 호스트 기반 액세스 제어 규칙이 있는지 확인

이 장에서는 IdM(Identity Management) 호스트 기반 액세스 정책과 Ansible 을 사용하여 정의하는 방법에 대해 설명합니다.

Ansible은 시스템을 구성하고, 소프트웨어를 배포하고, 롤링 업데이트를 수행하는 데 사용되는 자동화 도구입니다. IdM(Identity Management) 지원이 포함되어 있습니다.

19.1. IDM의 호스트 기반 액세스 제어 규칙

호스트 기반 액세스 제어(HBAC) 규칙은 서비스 그룹의 서비스 또는 서비스를 사용하여 어떤 호스트나 호스트 그룹에 액세스할 수 있는 사용자 또는 사용자 그룹을 정의합니다. 시스템 관리자는 HBAC 규칙을 사용하여 다음 목표를 달성할 수 있습니다.

- 도메인의 지정된 시스템에 대한 액세스를 특정 사용자 그룹의 멤버로 제한합니다.
- 특정 서비스만 도메인의 시스템에 액세스하도록 허용합니다.

기본적으로 IdM은 allow_all 이라는 기본 HBAC 규칙으로 구성되며, 이는 전체 IdM 도메인의 모든 관련 서비스를 통해 모든 사용자에 대한 모든 호스트에 대한 범용 액세스를 의미합니다.

기본 allow_all 규칙을 자체 HBAC 규칙 세트로 교체하여 다른 호스트에 대한 액세스를 미세 조정할 수 있습니다. 중앙 집중적이고 단순화된 액세스 제어 관리를 위해 개별 사용자, 호스트 또는 서비스 대신 사용자 그룹, 호스트 그룹 또는 서비스 그룹에 HBAC 규칙을 적용할 수 있습니다.

19.2. ANSIBLE 플레이북을 사용하여 IDM에 HBAC 규칙이 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에 HBAC(호스트 기반 액세스 제어) 규칙이 있는지 확인하는 방법에 대해 설명합니다.

사전 요구 사항

- ansible-freeipa 패키지는 Ansible 컨트롤러에 설치됩니다.

- IdM 관리자 암호를 알고 있습니다.
- HBAC 규칙에 사용할 사용자 및 사용자 그룹이 IdM에 있습니다. 자세한 내용은 Ansible 플레이북을 사용하여 사용자 계정 관리 및 Ansible 플레이북을 사용하여 IdM 그룹 및 그룹 구성원이 있는지 확인하십시오.
- HBAC 규칙을 적용하려는 호스트 및 호스트 그룹은 IdM에 있습니다. 자세한 내용은 Ansible 플레이북을 사용하여 호스트 관리 및 Ansible 플레이북을 사용하여 호스트 관리를 참조하십시오.

절차

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

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

2. 확인할 HBAC 정책을 정의하는 Ansible 플레이북 파일을 만듭니다. 이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/playbooks/hbacrule/ensure-hbacrule-allhosts-present.yml 파일에서 예제를 복사하고 수정할 수 있습니다.

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

  tasks:
  # Ensure idm_user can access client.idm.example.com via the sshd service
  - ipahbacrule:
      ipaadmin_password: MySecret123
      name: login
      user: idm_user
      host: client.idm.example.com
      hbacsvc:
      - sshd
      state: present
```

3. 플레이북을 실행합니다.

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

검증 단계

1. IdM 웹 UI에 관리자로 로그인합니다.

2. 정책 → 호스트 기반 액세스 제어 → HBAC 테스트로 이동합니다.

3. who(사용자) 탭에서 idm_user를 선택합니다.

4. Accessing(액세스) 탭에서 client.idm.example.com 을 선택합니다.

5. Via service 탭에서 sshd 를 선택합니다.

6. Rules(규칙) 탭에서 로그인을 선택합니다.

7. Run test(테스트 실행) 탭에서 Run test(테스트 실행) 단추를 클릭합니다. ACCESS GRANTED가 표시되면 HBAC 규칙이 성공적으로 구현됩니다.

추가 리소스

- Ansible을 사용하여 HBAC 서비스, 서비스 그룹 및 규칙을 구성하는 방법에 대한 자세한 내용은 README-hbacsvc.md, README-hbacsvcgroup.md, README-hbacrule.md Markdown 파일을 참조하십시오. 이러한 파일은 /usr/share/doc/ansible-freeipa 디렉토리에서 사용할 수 있습니다. /usr/share/doc/ansible-freeipa/playbooks 디렉토리의 관련 하위 디렉터리에서 사용할 수 있는 플레이북도 참조하십시오.

20장. IDM의 자격 증명 모음

이 장에서는 IdM(Identity Management)의 자격 증명 모음에 대해 설명합니다. 다음 주제를 소개합니다.

- 자격 증명 모음의 개념.
- 자격 증명 모음과 관련된 다양한 역할.
- 보안 및 액세스 제어 수준에 따라 IdM에서 사용할 수 있는 다양한 유형의 자격 증명 모음입니다.
- 소유권을 기반으로 IdM에서 사용할 수 있는 다양한 유형의 자격 증명 모음입니다.
- 자격 증명 모음 컨테이너의 개념.
- IdM에서 자격 증명 모음을 관리하는 기본 명령입니다.
- IdM에서 자격 증명 모음을 사용하기 위한 전제 조건인 KRA(키 복구 기관) 설치.

20.1. 자격 증명 모음 및 이점

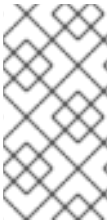
자격 증명 모음은 중요한 모든 데이터를 한 곳에서 안전하게 저장하지만 편리하게 저장하려는 IdM(Identity Management) 사용자에게 유용합니다. 이 섹션에서는 다양한 유형의 자격 증명 모음과 용도와 요구 사항에 따라 선택해야 하는 자격 증명 모음에 대해 설명합니다.

자격 증명 모음은 비밀 저장, 검색, 공유 및 복구를 위한 안전한 위치(IdM)입니다. 시크릿은 제한된 사용자 또는 엔터티 그룹만 액세스할 수 있는 보안에 민감한 데이터(일반적으로 인증 자격 증명)입니다. 예를 들어 보안에는 다음이 포함됩니다.

- 암호

- PIN
- 개인 SSH 키

자격 증명 모음은 암호 관리자와 유사합니다. 암호 관리자와 마찬가지로 자격 증명 모음은 일반적으로 사용자가 자격 증명 모음에 저장된 정보를 잠금 해제하고 액세스하기 위해 하나의 기본 암호를 생성하고 기억해야 합니다. 그러나 사용자는 표준 자격 증명 모음을 보유하도록 결정할 수도 있습니다. 표준 자격 증명 모음에서는 사용자가 자격 증명 모음에 저장된 시크릿에 액세스하기 위해 암호를 입력할 필요가 없습니다.

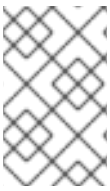


참고

IdM에서 자격 증명 모음의 목적은 외부의 IdM이 아닌 서비스에 인증할 수 있는 인증 자격 증명을 저장하는 것입니다.

IdM 자격 증명 모음의 기타 중요한 특성은 다음과 같습니다.

- 자격 증명 모음 소유자는 자격 증명 모음 소유자와 자격 증명 모음 소유자가 자격 증명 모음 구성원으로 선택한 해당 IdM 사용자만 액세스할 수 있습니다. 또한 IdM 관리자는 자격 증명 모음에 액세스할 수 있습니다.
- 사용자에게 자격 증명 모음을 생성하는 데 충분한 권한이 없는 경우 IdM 관리자는 자격 증명 모음을 생성하고 사용자를 소유자로 설정할 수 있습니다.
- 사용자 및 서비스는 IdM 도메인에 등록된 시스템에서 자격 증명 모음에 저장된 시크릿에 액세스할 수 있습니다.
- 하나의 자격 증명 모음은 하나의 비밀(예: 파일 한 개)만 포함할 수 있습니다. 그러나 파일 자체에는 암호, 키탭 또는 인증서와 같은 여러 시크릿이 포함될 수 있습니다.



참고

Vault는 IdM 웹 UI가 아닌 IdM CLI(명령줄)에서만 사용할 수 있습니다.

20.2. VAULT 소유자, 멤버 및 관리자

IdM(Identity Management)은 다음 자격 증명 모음 사용자 유형을 구분합니다.

Vault 소유자

자격 증명 모음 소유자는 자격 증명 모음에 대한 기본 관리 권한이 있는 사용자 또는 서비스입니다. 예를 들어, 자격 증명 모음 소유자는 자격 증명 모음의 속성을 수정하거나 새 자격 증명 모음 멤버를 추가할 수 있습니다.

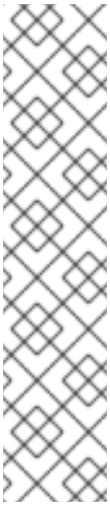
각 자격 증명 모음에는 소유자가 하나 이상 있어야 합니다. 자격 증명 모음에는 여러 소유자가 있을 수도 있습니다.

Vault 멤버

자격 증명 모음 멤버는 다른 사용자 또는 서비스에서 만든 자격 증명 모음에 액세스할 수 있는 사용자 또는 서비스입니다.

Vault 관리자

Vault 관리자는 모든 자격 증명 모음에 대한 무제한 액세스 권한을 가지며 모든 자격 증명 모음 작업을 수행할 수 있습니다.



참고

대칭 및 비대칭 자격 증명 모음은 암호 또는 키로 보호되며 특수 액세스 제어 규칙을 적용합니다(Vault 유형 참조). 관리자는 다음을 위해 다음 규칙을 충족해야 합니다.

- 대칭 및 비대칭 자격 증명 모음의 액세스 보안.
- vault 암호 또는 키를 변경하거나 재설정합니다.

자격 증명 모음 관리자는 Vault 관리자 권한이 있는 모든 사용자입니다. IdM의 역할 기반 액세스 제어(RBAC)의 컨텍스트에서 권한은 역할에 적용할 수 있는 권한 그룹입니다.

Vault 사용자

자격 증명 모음 사용자는 자격 증명 모음에 있는 컨테이너에 있는 사용자를 나타냅니다. Vault 사용자 정보는 ipa vault-show 와 같은 특정 명령의 출력에 표시됩니다.

```
$ ipa vault-show my_vault
```

```
Vault name: my_vault
Type: standard
Owner users: user
Vault user: user
```

vault 컨테이너 및 사용자 자격 증명 모음에 대한 자세한 내용은 Vault 컨테이너를 참조하십시오.

추가 리소스

- 특정 소유자 및 멤버 권한은 자격 증명 모음 유형에 따라 다릅니다. 자세한 내용은 Standard, 대칭 및 비대칭 자격 증명 모음 을 참조하십시오.

20.3. 표준, 대칭 및 비대칭 자격 증명 모음

보안 및 액세스 제어 수준에 따라 IdM은 자격 증명 모음을 다음 유형으로 분류합니다.

표준 자격 증명 모음

Vault 소유자 및 자격 증명 모음 멤버는 암호나 키를 사용하지 않고도 시크릿을 보관하고 검색할 수 있습니다.

대칭 자격 증명 모음

자격 증명 모음의 비밀은 대칭 키로 보호됩니다. Vault 소유자와 멤버는 시크릿을 보관하고 검색할 수 있지만 vault 암호를 제공해야 합니다.

비대칭 자격 증명 모음

자격 증명 모음의 비밀은 비대칭 키로 보호됩니다. 사용자는 공개 키를 사용하여 비밀을 보관하고 개인 키를 사용하여 검색합니다. Vault 멤버는 비밀만 보관할 수 있지만, 자격 증명 모음 소유자는 시크릿을 모두 수행할 수 있으며, 시크릿을 보관하고 검색할 수 있습니다.

20.4. 사용자, 서비스 및 공유 자격 증명 모음

IdM은 소유권에 따라 자격 증명을 여러 유형으로 분류합니다. 아래 테이블에 는 각 유형, 해당 소유자 및 사용에 대한 정보가 포함되어 있습니다.

표 20.1. 소유권 기반 IdM 자격 증명 모음

유형	설명	소유자	참고
사용자 자격 증명 모음	사용자의 개인 자격 증명 모음	단일 사용자	모든 사용자는 IdM 관리자가 허용한 경우 하나 이상의 사용자 자격 증명 모음을 소유할 수 있습니다.
서비스 자격 증명 모음	서비스의 개인 자격 증명 모음	단일 서비스	모든 서비스는 IdM 관리자가 허용한 경우 하나 이상의 사용자 자격 증명 모음을 소유할 수 있습니다.
공유 자격 증명 모음	여러 사용자 및 서비스에서 공유하는 자격 증명 모음	자격 증명 모음을 만든 자격 증명 모음 관리자	IdM 관리자가 허용하는 경우 사용자와 서비스는 하나 이상의 사용자 자격 증명 모음을 소유할 수 있습니다. 자격 증명 모음을 만든 관리자 이외의 자격 증명 모음 관리자에게도 자격 증명 모음에 대한 전체 액세스 권한이 있습니다.

20.5. VAULT 컨테이너

자격 증명 모음 컨테이너는 자격 증명 모음입니다. 아래 표에 는 IdM(Identity Management)에서 제공하는 기본 자격 증명 모음 컨테이너가 나열되어 있습니다.

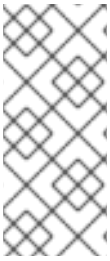
표 20.2. IdM의 기본 자격 증명 모음 컨테이너

유형	설명	목적
사용자 컨테이너	사용자의 개인 컨테이너	특정 사용자의 사용자 자격 증명 모음 저장
서비스 컨테이너	서비스용 개인 컨테이너	특정 서비스에 대한 서비스 자격 증명 모음 저장
공유 컨테이너	여러 사용자 및 서비스의 컨테이너	여러 사용자 또는 서비스에서 공유할 수 있는 자격 증명 모음을 저장합니다.

IdM은 사용자 또는 서비스의 첫 번째 개인 자격 증명 모음이 생성될 때 각 사용자 또는 서비스에 대한 사용자 및 서비스 컨테이너를 자동으로 생성합니다. 사용자 또는 서비스를 삭제한 후 IdM은 컨테이너 및 해당 콘텐츠를 제거합니다.

20.6. 기본 IDM 자격 증명 모음 명령

이 섹션에서는 IdM(Identity Management) 자격 증명 모음을 관리하는 데 사용할 수 있는 기본 명령에 대해 설명합니다. 아래 표에 는 목적의 설명과 함께 ipa vault-* 명령 목록이 포함되어 있습니다.



참고

ipa vault-* 명령을 실행하기 전에 **IdM** 도메인의 하나 이상의 서버에 **KRA(Key Recovery Authority)** 인증서 시스템 구성 요소를 설치합니다. 자세한 내용은 **IdM에 키 복구 권한** 설치를 참조하십시오.

표 20.3. 설명과 함께 기본 IdM 자격 증명 모음 명령

명령	목적
IPA 도움말 자격 증명 모음	IdM 자격 증명 모음 및 샘플 자격 증명 모음 명령에 대한 개념 정보를 표시합니다.
ipa vault-add --help, **ipa vault-find --help**	특정 **ipa vault-*** 명령에 **--help** 옵션을 추가하면 해당 명령에 사용할 수 있는 자세한 도움말과 옵션이 표시됩니다.

ipa vault-show user_vault --user idm_user	자격 증명 모음 멤버로 자격 증명 모음에 액세스하는 경우 자격 증명 모음 소유자를 지정해야 합니다. 자격 증명 모음 소유자를 지정하지 않으면 IdM에서 자격 증명 모음을 찾지 못했음을 알려줍니다.
	`[admin@server ~]$ ipa vault-show user_vault`
	`ipa: ERROR: user_vault: vault not found`

ipa vault-show shared_vault --shared	공유 자격 증명 모음에 액세스할 때 액세스하려는 자격 증명 모음이 공유 자격 증명 모음임을 지정해야 합니다. 그러지 않으면 IdM에서 자격 증명 모음을 찾지 못했음을 알려줍니다.
	`[admin@server ~]$ ipa vault-show shared_vault`
	`ipa: ERROR: shared_vault: vault not found`

20.7. IDM에 키 복구 권한 설치

이 섹션에서는 **KRA(키 복구 기관) 인증서 시스템(CS) 구성 요소를 설치하여 IdM(Identity Management)**에서 자격 증명 모음을 활성화하는 방법을 설명합니다.

사전 요구 사항

- **IdM 관리자로 로그인했습니다.**
- **IdM 클라이언트에서 root로 로그인했습니다.**

절차

- KRA를 설치합니다.

```
# ipa-kra-install
```

중요

숨겨진 복제본에 IdM 클러스터의 첫 번째 KRA를 설치할 수 있습니다. 그러나 추가 KRA를 설치하려면 숨기지 않은 복제본에 KRA 복제본을 설치하기 전에 임시로 숨겨진 복제본을 활성화해야 합니다. 그런 다음 원래 숨겨진 복제본을 다시 숨길 수 있습니다.

참고

자격 증명 모음 서비스를 고가용성으로 설정하려면 두 개의 IdM 서버에 KRA를 설치합니다.

추가 리소스

- IdM 복제본을 활성화하는 방법 및 숨기는 방법에 대한 자세한 내용은 숨겨진 복제본 데모 또는 승격을 참조하십시오.

- IdM의 숨겨진 복제본에 대한 자세한 내용은 숨겨진 복제본 모드를 참조하십시오.

21장. ANSIBLE을 사용하여 IDM 사용자 자격 증명 모음 관리: 시크릿 저장 및 검색

이 장에서는 Ansible vault 모듈을 사용하여 ID 관리에서 사용자 자격 증명 모음을 관리하는 방법을 설명합니다. 특히 사용자가 Ansible 플레이북을 사용하여 다음 세 가지 연속 작업을 수행하는 방법을 설명합니다.

- IdM에서 사용자 자격 증명 모음을 생성합니다.
- 자격 증명 모음에 비밀을 저장합니다.
- 자격 증명 모음에서 비밀을 검색합니다.

사용자는 두 개의 서로 다른 IdM 클라이언트에서 저장 및 검색을 수행할 수 있습니다.

사전 요구 사항

- IdM 도메인에 있는 하나 이상의 서버에 KRA(Key Recovery Authority) 인증 시스템 구성 요소가 설치되었습니다. 자세한 내용은 IdM에 키 복구 권한 설치를 참조하십시오.

21.1. ANSIBLE을 사용하여 IDM에 표준 사용자 자격 증명 모음이 있는지 확인

이 섹션에서는 IdM(Identity Management) 사용자가 Ansible 플레이북을 사용하여 하나 이상의 개인 자격 증명 모음으로 자격 증명 모음 컨테이너를 만들어 중요한 정보를 안전하게 저장하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 idm_user 사용자는 my_ vault 라는 표준 유형의 자격 증명 모음을 생성합니다. 표준 자격 증명 모음 유형을 사용하면 파일에 액세스할 때 idm_user 가 인증할 필요가 없습니다. idm_user 는 사용자가 로그인한 모든 IdM 클라이언트에서 파일을 검색할 수 있습니다.

사전 요구 사항

- 프로세스의 단계를 실행하는 호스트인 Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.
- idm_user 의 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/vault 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 인벤토리 파일을 생성합니다(예: inventory.file:

```
$ touch inventory.file
```

3. inventory.file 을 열고 [ipaserver] 섹션에서 구성할 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

4. ensure-standard-vault-is-present.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-standard-vault-is-present.yml ensure-standard-vault-is-present-copy.yml
```

5. 편집을 위해 ensure-standard-vault-is-present-copy.yml 파일을 엽니다.

6. ipavault 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_principal 변수를 idm_user 로 설정합니다.
- ipaadmin_password 변수를 idm_user 의 암호로 설정합니다.
- 사용자 변수를 idm_user 로 설정합니다.
- name 변수를 my_vault 로 설정합니다.
- vault_type 변수를 standard 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      vault_type: standard
```

7. 파일을 저장합니다.

8. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-standard-vault-is-present-copy.yml
```

21.2. ANSIBLE을 사용하여 IDM의 표준 사용자 자격 증명 모음에 비밀 보관

이 섹션에서는 **IdM(Identity Management)** 사용자가 **Ansible** 플레이북을 사용하여 개인 자격 증명 모음에 중요한 정보를 저장하는 방법을 보여줍니다. 사용된 예제에서 **idm_user** 사용자는 **my_ vault** 라는 자격 증명 모음에 **password.txt** 라는 중요한 정보로 파일을 보관합니다.

사전 요구 사항

- 프로세스의 단계를 실행하는 호스트인 **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **idm_user** 의 암호를 알고 있습니다.
- **idm_user** 는 소유자이거나 최소 **my_vault** 멤버 사용자입니다.
- **my_vault** 에 보관하려는 시크릿인 **password.txt** 에 액세스할 수 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/vault 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. data-archive-in-symmetric-vault.yml Ansible 플레이북 파일의 복사본을 만들지만 "대칭"을 "표준"으로 바꿉니다. 예를 들면 다음과 같습니다.

```
$ cp data-archive-in-symmetric-vault.yml data-archive-in-standard-vault-copy.yml
```

4. 편집을 위해 data-archive-in-standard-vault-copy.yml 파일을 엽니다.

5. ipavault 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_principal 변수를 idm_user 로 설정합니다.
- ipaadmin_password 변수를 idm_user 의 암호로 설정합니다.
- 사용자 변수를 idm_user 로 설정합니다.
- name 변수를 my_vault 로 설정합니다.
- 중요한 정보를 사용하여 변수 의 을 파일의 전체 경로로 설정합니다.
- action 변수를 member 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      in: /usr/share/doc/ansible-freeipa/playbooks/vault/password.txt
      action: member
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file data-archive-in-standard-vault-copy.yml
```

21.3. ANSIBLE을 사용하여 IDM의 표준 사용자 자격 증명 모음에서 시크릿 검색

이 섹션에서는 **IdM(Identity Management)** 사용자가 **Ansible** 플레이북을 사용하여 사용자 개인 자격 증명 모음에서 시크릿을 검색하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **idm_user** 사용자는 **Ansible**이 설치된 **IdM** 클라이언트에 **my_vault** 라는 표준 유형의 자격 증명 모음에서 중요한 데이터가 있는 파일을 검색합니다 . **idm_user** 는 파일에 액세스할 때 인증할 필요가 없습니다. **idm_user** 는 **Ansible**을 사용하여 **Ansible**이 설치된 **IdM** 클라이언트에서 파일을 검색할 수 있습니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- **idm_user** 의 암호를 알고 있습니다.
- **idm_user** 는 **my_vault** 의 소유자입니다.

- idm_user 가 시크릿을 my_vault에 저장했습니다.
- Ansible은 비밀을 검색할 IdM 호스트의 디렉터리에 쓸 수 있습니다.
- idm_user 는 IdM 호스트의 디렉터리에서 시크릿을 검색할 수 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/vault 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 인벤토리 파일을 열고 명확하게 정의된 섹션에서 시크릿을 검색하려는 IdM 클라이언트를 언급합니다. 예를 들어 host01.idm.example.com에서 비밀을 검색하도록 Ansible에 지시하려면 다음을 입력합니다.

```
[ipahost]
host01.idm.example.com
```

3. retrive-data-symmetric-vault.yml Ansible 플레이북 파일의 복사본을 만듭니다. "대칭"을 "표준"으로 바꿉니다. 예를 들면 다음과 같습니다.

```
$ cp retrive-data-symmetric-vault.yml retrieve-data-standard-vault.yml-copy.yml
```

4. 편집을 위해 retrieve-data-standard-vault.yml-copy.yml 파일을 엽니다.
5. hosts 변수를 ipahost 로 설정하여 파일을 조정합니다.
6. ipavault 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - ipaadmin_principal 변수를 idm_user 로 설정합니다.
   - ipaadmin_password 변수를 idm_user 의 암호로 설정합니다.

- 사용자 변수를 **idm_user** 로 설정합니다.
- **name** 변수를 **my_vault** 로 설정합니다.
- 보안을 내보낼 파일의 전체 경로로 **out** 변수를 설정합니다.
- **state** 변수를 불러오도록 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Tests
  hosts: ipahost
  become: true
  gather_facts: false

  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      out: /tmp/password_exported.txt
      state: retrieved
```

7. 파일을 저장합니다.

8. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file retrieve-data-standard-vault.yml-copy.yml
```

검증 단계

1. **user01**로 **host01** 에 **SSH** 로 :

```
$ ssh user01@host01.idm.example.com
```

2. **Ansible** 플레이북 파일에서 **out** 변수로 지정한 파일을 확인합니다.

```
$ vim /tmp/password_exported.txt
```

이제 내보낸 보안이 표시됩니다.

- Ansible을 사용하여 IdM 자격 증명 모음 및 사용자 시크릿 및 플레이북 변수에 대한 자세한 내용은 /usr/share/doc/ansible-freeipa/ 디렉터리에 사용 가능한 README-vault.md Markdown 파일과 /usr/ share/doc/ansible-freeipa/playbooks/vault/ 디렉터리에 있는 샘플 플레이북을 참조하십시오.

22장. ANSIBLE을 사용하여 IDM 서비스 자격 증명 모음 관리: 시크릿 저장 및 검색

이 섹션에서는 관리자가 **ansible-freeipa vault** 모듈을 사용하여 서비스 시크릿을 중앙 집중식 위치에 안전하게 저장하는 방법을 보여줍니다.

예제에서 사용되는 자격 증명 모음 은 비대칭입니다. 즉, 관리자가 다음 단계를 수행해야 함을 의미합니다.

1. 예를 들어 **openssl** 유틸리티를 사용하여 개인 키를 생성합니다.
2. 개인 키를 기반으로 공개 키를 생성합니다.

서비스 비밀은 관리자가 자격 증명 모음에 보관할 때 공개 키로 암호화됩니다. 이후 도메인의 특정 시스템에서 호스팅되는 서비스 인스턴스는 개인 키를 사용하여 비밀을 검색합니다. 서비스와 관리자만 비밀에 액세스할 수 있습니다.

보안이 손상되면 관리자는 서비스 자격 증명 모음에서 암호를 교체한 다음 손상되지 않은 개별 서비스 인스턴스에 재배포할 수 있습니다.

사전 요구 사항

- **IdM** 도메인에 있는 하나 이상의 서버에 **KRA(Key Recovery Authority)** 인증 시스템 구성 요소가 설치되었습니다. 자세한 내용은 **IdM**에 키 복구 권한 설치를 참조하십시오.

이 섹션에는 다음 절차가 포함됩니다.

- **Ansible**을 사용하여 **IdM**에 비대칭 서비스 자격 증명 모음이 있는지 확인
- **Ansible**을 사용하여 비대칭 자격 증명 모음에 **IdM** 서비스 시크릿 저장
- **Ansible**을 사용하여 **IdM** 서비스의 서비스 시크릿 검색

- Ansible을 사용하여 손상된 경우 IdM 서비스 자격 증명 모음 시크릿 변경

절차에서 다음을 수행합니다.

- admin 은 서비스 암호를 관리하는 관리자입니다.
- private-key-to-an-externally-signed-certificate.pem 은 서비스 비밀을 포함하는 파일입니다(이 경우 외부 서명 인증서에 대한 개인 키). 자격 증명 모음에서 비밀을 검색하는 데 사용되는 개인 키와 이 개인 키를 혼동하지 마십시오.
- secret_vault 는 서비스 시크릿을 저장하기 위해 생성된 자격 증명 모음입니다.
- HTTP/webserver1.idm.example.com 은 자격 증명 모음의 소유자인 서비스입니다.
- HTTP/webserver2.idm.example.com 및 HTTP/webserver3.idm.example.com 은 자격 증명 모음 멤버 서비스입니다.
- service-public.pem 은 password _vault 에 저장된 암호를 암호화하는 데 사용되는 서비스 공개 키입니다.
- service-private.pem 은 secret_vault 에 저장된 암호를 해독하는 데 사용되는 서비스 개인 키입니다.

22.1. ANSIBLE을 사용하여 IDM에 비대칭 서비스 자격 증명 모음이 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 하나 이상의 개인 자격 증명 모음 컨테이너를 사용하여 중요한 정보를 안전하게 저장하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 관리자는 secret_vault 라는 비대칭 자격 증명 모음을 생성합니다. 이렇게 하면 자격 증명 모음에서 비밀을 검색하기 위해 자격 증명 모음 멤버가 개인 키를 사용하여 인증해야 합니다. 자격 증명 모음 멤버는 IdM 클라이언트에서 파일을 검색할 수 있습니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/vault 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 서비스 인스턴스의 공개 키를 가져옵니다. 예를 들어 openssl 유틸리티를 사용합니다.

   a. service-private.pem 개인 키를 생성합니다.

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....................................+++
e is 65537 (0x10001)
```

   b. 개인 키를 기반으로 service-public.pem 공개 키를 생성합니다.

```
$ openssl rsa -in service-private.pem -out service-public.pem -pubout
writing RSA key
```

3. 선택 사항: 인벤토리 파일이 없는 경우(예: inventory.file)를 생성합니다.

```
$ touch inventory.file
```

4. 인벤토리 파일을 열고 [ipaserver] 섹션에서 구성할 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

5. ensure-asymmetric-vault-is-present.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-asymmetric-vault-is-present.yml ensure-asymmetric-service-vault-is-present-copy.yml
```

6. 편집을 위해 **ensure-asymmetric-vault-is-present-copy.yml** 파일을 엽니다.

7. **service-public.pem** 공개 키를 **Ansible** 컨트롤러에서 **server. idm.example.com** 서버로 복사하는 작업을 추가합니다.

8. **ipavault** 작업 섹션에서 다음 변수를 설정하여 나머지 파일을 수정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 사용하여 자격 증명 모음의 이름을 정의합니다(예: **secret_vault**).
- **vault_type** 변수를 비대칭 으로 설정합니다.
- 서비스 변수를 자격 증명 모음을 소유하는 서비스의 주체(예: **HTTP/webserver1.idm.example.com**)로 설정합니다.
- **public_key_file** 을 공개 키의 위치로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Tests
  hosts: ipaserver
  become: true
  gather_facts: false
  tasks:
  - name: Copy public key to ipaserver.
    copy:
      src: /path/to/service-public.pem
      dest: /usr/share/doc/ansible-freeipa/playbooks/vault/service-public.pem
      mode: 0600
  - name: Add data to vault, from a LOCAL file.
    ipavault:
      ipaadmin_password: Secret123
      name: secret_vault
```

```
    vault_type: asymmetric
    service: HTTP/webserver1.idm.example.com
    public_key_file: /usr/share/doc/ansible-freeipa/playbooks/vault/service-public.pem
```

9. 파일을 저장합니다.

10. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-asymmetric-service-vault-is-present-copy.yml
```

22.2. ANSIBLE을 사용하여 멤버 서비스 추가

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 서비스 자격 증명 모음에 멤버 서비스를 추가하여 모두 자격 증명 모음에 저장된 비밀을 검색할 수 있는 방법을 보여줍니다. 아래 절차에서 사용되는 예제에서 IdM 관리자는 HTTP/webserver2.idm.example.com 및 HTTP/webserver3. idm.example.com 서비스 주체를 HTTP/webserver1.idm.example.com 에서 소유한 secret_vault 자격 증명 모음에 추가합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- 서비스 시크릿 을 저장할 비대칭 자격 증명 모음을 생성했습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/vault 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 선택 사항: 인벤토리 파일이 없는 경우(예: inventory.file)를 생성합니다.

```
$ touch inventory.file
```

3. 인벤토리 파일을 열고 **[ipaserver]** 섹션에서 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

4. **data-archive-in-asymmetric-vault.yml Ansible** 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp data-archive-in-asymmetric-vault.yml add-services-to-an-asymmetric-vault.yml
```

5. 편집을 위해 **data-archive-in-asymmetric-vault-copy.yml** 파일을 엽니다.

6. **ipavault** 작업 섹션에서 다음 변수를 설정하여 파일을 수정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 자격 증명 모음의 이름으로 설정합니다(예: **secret_vault**).
- 서비스 변수를 자격 증명 모음의 서비스 소유자(예: **HTTP/webserver1.idm.example.com**)로 설정합니다.
- **services** 변수를 사용하여 자격 증명 모음 시크릿에 액세스할 서비스를 정의합니다.
- **action** 변수를 **member** 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - ipavault:
      ipaadmin_password: Secret123
```

```
    name: secret_vault
    service: HTTP/webserver1.idm.example.com
    services:
    - HTTP/webserver2.idm.example.com
    - HTTP/webserver3.idm.example.com
    action: member
```

7. 파일을 저장합니다.

8. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file add-services-to-an-asymmetric-vault.yml
```

22.3. ANSIBLE을 사용하여 비대칭 자격 증명 모음에 IDM 서비스 시크릿 저장

이 섹션에서는 IdM(Identity Management) 관리자가 나중에 서비스에서 검색할 수 있도록 Ansible 플레이북을 사용하여 서비스 자격 증명 모음에 시크릿을 저장할 수 있는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 관리자는 시크릿과 함께 PEM 파일을 secret_vault 라는 비대칭 자격 증명 모음에 저장합니다. 이렇게 하면 자격 증명 모음에서 비밀을 검색하기 위해 서비스가 개인 키를 사용하여 인증해야 합니다. 자격 증명 모음 멤버는 IdM 클라이언트에서 파일을 검색할 수 있습니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- 서비스 시크릿 을 저장할 비대칭 자격 증명 모음을 생성했습니다.
- 시크릿은 Ansible 컨트롤러에 로컬로 저장됩니다(예: /usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem).

절차

1. /usr/share/doc/ansible-freeipa/playbooks/vault 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 선택 사항: 인벤토리 파일이 없는 경우(예: inventory.file)를 생성합니다.

```
$ touch inventory.file
```

3. 인벤토리 파일을 열고 [ipaserver] 섹션에서 구성할 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

4. data-archive-in-asymmetric-vault.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp data-archive-in-asymmetric-vault.yml data-archive-in-asymmetric-vault-copy.yml
```

5. 편집을 위해 data-archive-in-asymmetric-vault-copy.yml 파일을 엽니다.

6. ipavault 작업 섹션에서 다음 변수를 설정하여 파일을 수정합니다.

- ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.
- name 변수를 자격 증명 모음의 이름으로 설정합니다(예: secret_vault).
- 서비스 변수를 자격 증명 모음의 서비스 소유자(예: HTTP/webserver1.idm.example.com)로 설정합니다.
- 변수 의 를 "{{ lookup('file', 'private-key-to-an-externally-signed-certificate.pem') | b64encode }}" 로 설정합니다. 이렇게 하면 Ansible이 IdM 서버가 아닌 Ansible 컨트롤러의 작업 디렉터리에서 개인 키를 사용하여 파일을 검색할 수 있습니다.
- action 변수를 member 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
  - ipavault:
      ipaadmin_password: Secret123
      name: secret_vault
      service: HTTP/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
```

22.4. ANSIBLE을 사용하여 IDM 서비스의 서비스 시크릿 검색

이 섹션에서는 IdM(Identity Management) 사용자가 Ansible 플레이북을 사용하여 서비스 대신 서비스 자격 증명 모음에서 시크릿을 검색하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 플레이북을 실행하면 secret _vault라는 비대칭 자격 증명 모음의 시크릿 이 있는 PEM 파일을 검색하여 Ansible 인벤토리 파일에 나열된 모든 호스트의 지정된 위치에 ipaservers 로 저장합니다.

서비스는 키탭을 사용하여 IdM에 인증하고 개인 키를 사용하여 자격 증명 모음에 인증합니다. ansible-freeipa 가 설치된 IdM 클라이언트에서 서비스를 대신하여 파일을 검색할 수 있습니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- 서비스 시크릿 을 저장할 비대칭 자격 증명 모음을 생성했습니다.

- 자격 증명 모음에 비밀을 보관합니다.

- Ansible 컨트롤러의 private_key_file 변수에서 지정한 위치에서 서비스 자격 증명 모음 시크릿을 검색하는 데 사용되는 개인 키를 저장했습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/vault 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 선택 사항: 인벤토리 파일이 없는 경우(예: inventory.file)를 생성합니다.

```
$ touch inventory.file
```

3. 인벤토리 파일을 열고 다음 호스트를 정의합니다.

   - [ipaserver] 섹션에 IdM 서버를 정의합니다.

   - [webservers] 섹션에서 시크릿을 검색할 호스트를 정의합니다. 예를 들어 webserver1.idm.example.com, webserver2.idm.example.com 및 webserver3.idm.example.com 에 대한 시크릿을 검색하도록 Ansible에 지시하려면 다음을 입력합니다.

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

[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
webserver3.idm.example.com
```

4. retrieve-data-asymmetric-vault.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp retrieve-data-asymmetric-vault.yml retrieve-data-asymmetric-vault-copy.yml
```

5. 편집을 위해 **retrieve-data-asymmetric-vault-copy.yml** 파일을 엽니다.

6. **ipavault** 작업 섹션에서 다음 변수를 설정하여 파일을 수정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 자격 증명 모음의 이름으로 설정합니다(예: **secret_vault**).
- 서비스 변수를 자격 증명 모음의 서비스 소유자(예: **HTTP/webserver1.idm.example.com**)로 설정합니다.
- **private_key_file** 변수를 서비스 자격 증명 모음 시크릿을 검색하는 데 사용되는 개인 키의 위치로 설정합니다.
- **out** 변수를 **private-key-to-an-externally-signed-certificate.pem** 시크릿(예: 현재 작업 디렉터리)을 검색하려는 **IdM** 서버의 위치로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false

  tasks:
  - name: Retrieve data from the service vault
    ipavault:
      ipaadmin_password: Secret123
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      vault_type: asymmetric
      private_key: "{{ lookup('file', 'service-private.pem') | b64encode }}"
      out: private-key-to-an-externally-signed-certificate.pem
      state: retrieved
```

7. IdM 서버에서 Ansible 컨트롤러에 데이터 파일을 검색하는 섹션을 플레이북에 추가합니다.

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false
  tasks:
[...]
  - name: Retrieve data file
    fetch:
      src: private-key-to-an-externally-signed-certificate.pem
      dest: ./
      flat: yes
      mode: 0600
```

8. 의 Ansible 컨트롤러에서 인벤토리 파일의 webservers 섹션에 나열된 webservers로 검색된 private-key-to-an-externally-signed-certificate.pem 파일을 전송하는 플레이북에 섹션을 추가합니다.

```
---
- name: Send data file to webservers
  become: no
  gather_facts: no
  hosts: webservers
  tasks:
  - name: Send data to webservers
    copy:
      src: private-key-to-an-externally-signed-certificate.pem
      dest: /etc/pki/tls/private/httpd.key
      mode: 0444
```

9. 파일을 저장합니다.

10. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file retrieve-data-asymmetric-vault-copy.yml
```

22.5. ANSIBLE을 사용하여 손상된 경우 IDM 서비스 자격 증명 모음 시크릿 변경

이 섹션에서는 IdM(Identity Management) 관리자가 서비스 인스턴스가 손상된 경우 Ansible 플레이북을 다시 사용하여 서비스 자격 증명 모음에 저장된 시크릿을 변경할 수 있는 방법을 보여줍니다. 다음 예제의 시나리오에서는 webserver3.idm.example.com 에서 검색된 시크릿이 손상되었지만 시크릿을 저장하는 비대칭 자격 증명 모음의 키가 손상되었다고 가정합니다. 이 예제에서 관리자는 비대칭 자격 증명 모음에 비밀을 저장하고 IdM 호스트에 비대칭 자격 증명 모음에서 시크릿 을 검색할 때 사용하는

Ansible 플레이북을 재사용합니다. 절차 시작 시 IdM 관리자는 비대칭 자격 증명 모음에 새 시크릿을 사용하여 새 PEM 파일을 저장하고, 에서 손상된 웹 서버 webserver3.idm.example.com 에 새 시크릿을 검색하지 않도록 인벤토리 파일을 조정한 다음 두 절차를 다시 실행합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- 서비스 시크릿 을 저장할 비대칭 자격 증명 모음을 생성했습니다.
- 손상된 이전 키를 교체하기 위해 IdM 호스트에서 실행 중인 웹 서비스에 대한 새 httpd 키를 생성했습니다.
- 새 httpd 키는 Ansible 컨트롤러에 로컬로 저장됩니다(예: /usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem).

절차

1. /usr/share/doc/ansible-freeipa/playbooks/vault 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. 인벤토리 파일을 열고 다음 호스트가 올바르게 정의되었는지 확인합니다.
   - [ipaserver] 섹션의 IdM 서버.
   - [webservers] 섹션에서 시크릿을 검색할 호스트입니다. 예를 들어 webserver1.idm.example.com 및 webserver2.idm.example.com 에 대한 시크릿을 검색하도록 Ansible에 지시하려면 다음을 입력합니다.

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

```
[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
```

중요

현재 예제 **webserver3.idm.example.com** 에 손상된 웹 서버가 포함되어 있지 않은지 확인합니다.

3. 편집을 위해 **data-archive-in-asymmetric-vault-copy.yml** 파일을 엽니다.

4. **ipavault** 작업 섹션에서 다음 변수를 설정하여 파일을 수정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 자격 증명 모음의 이름으로 설정합니다(예: **secret_vault**).
- 서비스 변수를 자격 증명 모음의 서비스 소유자(예: **HTTP/webserver.idm.example.com**)로 설정합니다.
- 변수 의 를 **"{{ lookup('file'), 'new-private-key-to-an-externally-signed-certificate.pem') | b64encode }}"** 로 설정합니다. 이렇게 하면 **Ansible**이 **IdM** 서버가 아닌 **Ansible** 컨트롤러의 작업 디렉터리에서 개인 키를 사용하여 파일을 검색할 수 있습니다.
- **action** 변수를 **member** 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

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

  tasks:
  - ipavault:
      ipaadmin_password: Secret123
      name: secret_vault
```

```
    service: HTTP/webserver.idm.example.com
    in: "{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem') |
b64encode }}"
    action: member
```

5. 파일을 저장합니다.

6. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file data-archive-in-asymmetric-vault-copy.yml
```

7. 편집을 위해 **retrieve-data-asymmetric-vault-copy.yml** 파일을 엽니다.

8. **ipavault** 작업 섹션에서 다음 변수를 설정하여 파일을 수정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 자격 증명 모음의 이름으로 설정합니다(예: **secret_vault**).
- 서비스 변수를 자격 증명 모음의 서비스 소유자(예: **HTTP/webserver1.idm.example.com**)로 설정합니다.
- **private_key_file** 변수를 서비스 자격 증명 모음 시크릿을 검색하는 데 사용되는 개인 키의 위치로 설정합니다.
- **out** 변수를 **new-private-key-to-an-externally-signed-certificate.pem** 시크릿(예: 현재 작업 디렉터리)을 검색하려는 **IdM** 서버의 위치로 설정합니다.
- **action** 변수를 **member** 로 설정합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Retrieve data from vault
  hosts: ipaserver
```

```
  become: no
  gather_facts: false

  tasks:
  - name: Retrieve data from the service vault
    ipavault:
      ipaadmin_password: Secret123
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      vault_type: asymmetric
      private_key: "{{ lookup('file', 'service-private.pem') | b64encode }}"
      out: new-private-key-to-an-externally-signed-certificate.pem
      state: retrieved
```

9. **IdM** 서버에서 **Ansible** 컨트롤러에 데이터 파일을 검색하는 섹션을 플레이북에 추가합니다.

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: yes
  gather_facts: false
  tasks:
[...]
  - name: Retrieve data file
    fetch:
      src: new-private-key-to-an-externally-signed-certificate.pem
      dest: ./
      flat: yes
      mode: 0600
```

10. 의 **Ansible** 컨트롤러에서 인벤토리 파일의 **webservers** 섹션에 나열된 **webservers**로 검색된 **new-private-key-to-an-externally-signed-certificate.pem** 파일을 전송하는 플레이북에 섹션을 추가합니다.

```
---
- name: Send data file to webservers
  become: yes
  gather_facts: no
  hosts: webservers
  tasks:
  - name: Send data to webservers
    copy:
      src: new-private-key-to-an-externally-signed-certificate.pem
      dest: /etc/pki/tls/private/httpd.key
      mode: 0444
```

11. 파일을 저장합니다.

12. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file retrieve-data-asymmetric-vault-copy.yml
```

추가 리소스

- Ansible을 사용하여 IdM 자격 증명 모음 및 서비스 시크릿 및 플레이북 변수에 대한 자세한 내용은 /usr/share/doc/ansible-freeipa/ 디렉터리에 사용 가능한 README-vault.md Markdown 파일과 /usr/ share/doc/ansible-freeipa/playbooks/vault/ 디렉터리에 있는 샘플 플레이북을 참조하십시오.

23장. ANSIBLE을 사용하여 IDM에 서비스가 있는지 여부 확인

Ansible service 모듈을 사용하여 IdM(Identity Management) 관리자는 IdM에 고유하지 않거나 IdM에 없는 특정 서비스가 있는지 확인할 수 있습니다. 예를 들어 service 모듈을 사용하여 다음을 수행할 수 있습니다.

- 수동으로 설치한 서비스가 IdM 클라이언트에 있는지 확인하고 해당 서비스가 없는 경우 자동으로 설치합니다. 자세한 내용은 다음을 참조하십시오.
  - IdM 클라이언트에 HTTP 서비스가 IdM에 있는지 확인합니다.
  - IdM이 아닌 클라이언트의 IdM에 HTTP 서비스가 있는지 확인합니다.
  - DNS 없이 IdM 클라이언트에 HTTP 서비스가 있는지 확인합니다.
- IdM에 등록된 서비스에 인증서가 연결되어 있고 해당 인증서가 없는 경우 자동으로 설치되었는지 확인합니다. 자세한 내용은 다음을 참조하십시오.
- IdM 서비스 항목에 외부 서명된 인증서가 있는지 확인합니다.
- IdM 사용자와 호스트에서 서비스 키탭을 검색하고 생성할 수 있도록 허용합니다. 자세한 내용은 다음을 참조하십시오.
  - IdM 사용자, 그룹, 호스트 또는 호스트 그룹이 서비스의 키탭을 생성하도록 허용.
  - IdM 사용자, 그룹, 호스트 또는 호스트 그룹이 서비스의 키탭을 검색할 수 있도록 허용.
- IdM 사용자와 호스트에서 Kerberos 별칭을 서비스에 추가할 수 있도록 허용합니다. 자세한 내용은 다음을 참조하십시오.
  - 서비스에 Kerberos 기본 별칭이 있는지 확인합니다.
-

서비스가 IdM 클라이언트에 없는지 확인하고 해당 서비스가 있는 경우 자동으로 제거합니다. 자세한 내용은 다음을 참조하십시오.

- IdM 클라이언트의 IdM에 HTTP 서비스가 없는지 확인합니다.

23.1. ANSIBLE 플레이북을 사용하여 IDM에 HTTP 서비스가 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하여 IdM에 HTTP 서버가 있는지 확인하는 방법을 설명합니다.

사전 요구 사항

- HTTP 서비스를 호스팅하는 시스템은 IdM 클라이언트입니다.
- IdM 관리자 암호가 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: inventory.file:

```
$ touch inventory.file
```

2. inventory.file 을 열고 [ipaserver] 섹션에서 구성할 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 편집을 위해 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-copy.yml Ansible 플레이북 파일을 엽니다.

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure service is present
  - ipaservice:
      ipaadmin_password: Secret123
      name: HTTP/client.idm.example.com
```

5. 파일을 조정합니다.

   - **ipaadmin_password** 변수에 정의된 IdM 관리자 암호를 변경합니다.
   - **ipaservice** 작업의 **name** 변수에 정의된 대로 **HTTP** 서비스가 실행 중인 IdM 클라이언트의 이름을 변경합니다.

6. 파일을 저장하고 종료합니다.

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

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-copy.yml
```

검증 단계

1. IdM 관리자로 IdM 웹 UI에 로그인합니다.
2. ID → 서비스로 이동합니다.

HTTP/client.idm.example.com@IDM.EXAMPLE.COM 이 서비스 목록에 나열되면 Ansible 플레이북이 IdM에 성공적으로 추가되었습니다.

추가 리소스

- TLS 암호화를 Apache HTTP 서버에 추가하여 HTTP 서버와 브라우저 클라이언트 간의 통신

을 보호할 수 있습니다.

- IdM 인증 기관에서 HTTP 서비스의 인증서를 요청할 수 있습니다. 자세한 내용은 certmonger를 사용하여 서비스의 IdM 인증서 가져오기에 설명된 절차를 참조하십시오.

23.2. ANSIBLE 플레이북을 사용하여 비IDM 클라이언트에서 IDM에 HTTP 서비스가 있는지 확인

이 섹션에서는 Ansible 플레이북을 사용하는 IdM 클라이언트가 아닌 호스트에서 IdM에 HTTP 서버가 있는지 확인하는 방법에 대해 설명합니다. IdM에 HTTP 서버를 추가하여 IdM에 호스트를 추가합니다.

사전 요구 사항

- 호스트에 HTTP 서비스를 설치했습니다.
- HTTP를 설정한 호스트는 IdM 클라이언트가 아닙니다. 또는 Ansible 플레이북을 사용하여 IdM에 HTTP 서비스가 있는지 확인하는 단계를 따르십시오.
- IdM 관리자 암호가 있습니다.
- 호스트에 대해 IPv6를 사용하는 경우 DNS A 레코드 또는 AAAA 레코드가 사용됩니다.

절차

1. 인벤토리 파일을 생성합니다(예: inventory.file:

```
$ touch inventory.file
```

2. inventory.file 을 열고 [ipaserver] 섹션에서 구성할 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

-

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check-copy.yml
```

4. 복사된 파일 **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check-copy.yml** 을 열어 편집합니다. **ipa service** 작업에서 **ipaadmin_password** 및 이름 변수를 찾습니다.

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure service is present
  - ipaservice:
      ipaadmin_password: MyPassword123
      name: HTTP/www2.example.com
      skip_host_check: yes
```

5. 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 **HTTP** 서비스가 실행 중인 호스트의 이름으로 설정합니다.

6. 파일을 저장하고 종료합니다.

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

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check-copy.yml
```

검증 단계

1. **IdM** 관리자로 **IdM** 웹 **UI**에 로그인합니다.

2. **ID → 서비스**로 이동합니다.

이제 서비스 목록에 **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** 이 표시됩니다.

추가 리소스

- **TLS 암호화를 Apache HTTP 서버에 추가하여 HTTP 서버와 브라우저 클라이언트 간의 통신을 보호할 수 있습니다.**

23.3. ANSIBLE PLAYBOOK을 사용하여 DNS 없이 IDM 클라이언트에 HTTP 서비스가 있는지 확인

이 섹션에서는 **Ansible** 플레이북을 사용하여 **DNS** 항목이 없는 **IdM** 클라이언트에서 **HTTP** 서버가 실행되고 있는지 확인하는 방법에 대해 설명합니다. 시나리오는 **IdM** 호스트에 사용 가능한 **DNS A** 항목이 없으며 **IPv4** 대신 **IPv6**를 사용하는 경우 **DNS AAAA** 항목이 없다는 것입니다.

사전 요구 사항

- **HTTP** 서비스를 호스트하는 시스템은 **IdM**에 등록되어 있습니다.
- 호스트의 **DNS A** 또는 **DNS AAAA** 레코드가 없을 수 있습니다. 그렇지 않으면 호스트의 **DNS** 레코드가 있는 경우 **Ansible 플레이북을 사용하여 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 구성할 **IdM** 서버를 정의합니다. 예를 들어 **server.idm.example.com**을 구성하도록 **Ansible**에 지시하려면 다음을 입력합니다.

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

3. **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.**

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml
```

4. **복사된 파일 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml 을 열어 편집합니다. ipa service 작업에서 ipaadmin_password 및 이름 변수를 찾습니다.**

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure service is present
  - ipaservice:
      ipaadmin_password: MyPassword123
      name: HTTP/ihavenodns.info
      force: yes
```

5. **파일을 조정합니다.**

   - **ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.**
   - **name 변수를 HTTP 서비스가 실행 중인 호스트의 이름으로 설정합니다.**

6. **파일을 저장하고 종료합니다.**

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

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml
```

검증 단계

1. IdM 관리자로 IdM 웹 UI에 로그인합니다.

2. ID → 서비스로 이동합니다.

이제 서비스 목록에 HTTP/client.idm.example.com@IDM.EXAMPLE.COM 이 표시됩니다.

추가 리소스

- Apache HTTP 서버에 TLS 암호화를 추가하여 Apache HTTP 서버와 브라우저 클라이언트 간의 통신을 보호할 수 있습니다.

23.4. ANSIBLE PLAYBOOK을 사용하여 IDM 서비스 항목에 외부 서명된 인증서가 있는지 확인

이 섹션에서는 ansible-freeipa 서비스 모듈을 사용하여 외부 CA(인증 기관)에서 발급한 인증서가 HTTP 서비스의 IdM 항목에 연결되어 있는지 확인하는 방법을 설명합니다. IdM CA가 자체 서명된 인증서를 사용하는 경우 IdM CA가 아닌 외부 CA에서 서명한 HTTP 서비스의 인증서를 보유하는 것이 특히 유용합니다.

사전 요구 사항

- 호스트에 HTTP 서비스를 설치했습니다.
- HTTP 서비스를 IdM에 등록했습니다.
- IdM 관리자 암호가 있습니다.
- 주체가 HTTP 서비스의 주체에 해당하는 외부 서명 인증서가 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: inventory.file:

```
$ touch inventory.file
```

2.

inventory.file 을 열고 [ipaserver] 섹션에서 구성할 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. /usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present.yml 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 선택 사항: 인증서가 PEM(개인 정보 보호 강화 메일) 형식인 경우 CLI(명령줄 인터페이스)를 통해 더 쉽게 처리할 수 있도록 인증서를 DER(고급 인코딩 규칙) 형식으로 변환합니다.

```
$ openssl x509 -outform der -in cert1.pem -out cert1.der
```

5. base64 명령을 사용하여 DER 파일을 표준 출력으로 디코딩합니다. w 0 옵션을 사용하여 래핑을 비활성화합니다.

```
$ base64 cert1.der -w0
MIIC/zCCAeegAwIBAgIUV74O+4kXeg21o4vxfRRtyJm...
```

6. 인증서를 표준 출력에서 클립보드로 복사합니다.

7. /usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present-copy.yml 파일을 열어 내용을 편집하고 봅니다.

```
---
- name: Service certificate present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure service certificate is present
  - ipaservice:
      ipaadmin_password: MyPassword123
      name: HTTP/www.example.com
      certificate: |
        - MIICBjCCAW8CFHnm32VcXaUDGfEGdDL/...
```

```
    [...]
    action: member
    state: present
```

8. 파일을 조정합니다.

- 인증서 변수를 사용하여 정의한 인증서를 **CLI**에서 복사한 인증서로 바꿉니다. 표시된 대로 "|" 파이프 문자가 있는 **certificate:** 변수를 사용하는 경우 한 줄에 입력하지 않고 인증서 **THIS WAY**를 입력할 수 있습니다. 이렇게 하면 인증서를 더 쉽게 읽을 수 있습니다.
- **ipaadmin_password** 변수에 정의된 **IdM** 관리자 암호를 변경합니다.
- **name** 변수에서 정의한 **HTTP** 서비스가 실행 중인 **IdM** 클라이언트의 이름을 변경합니다.
- 기타 관련 변수를 변경합니다.

9. 파일을 저장하고 종료합니다.

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

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present-copy.yml
```

검증 단계

1. **IdM** 관리자로 **IdM** 웹 **UI**에 로그인합니다.

2. **ID** → 서비스로 이동합니다.

3. 새로 추가된 인증서(예: **HTTP/client.idm.example.com**)를 사용하여 서비스 이름을 클릭합니다.

오른쪽에 있는 **Service Certificate**(서비스 인증서) 섹션에서 새로 추가된 인증서를 볼 수 있습니다.

23.5. ANSIBLE 플레이북을 사용하여 IDM 사용자, 그룹, 호스트 또는 호스트 그룹을 사용하여 서비스의 키탭을 생성할 수 있습니다.

keytab은 **Kerberos** 주체와 암호화된 키의 쌍을 포함하는 파일입니다. 키탭 파일은 일반적으로 일반 텍스트 파일에 저장된 암호나 사람의 상호 작용 없이도 **Kerberos**를 사용하여 스크립트를 자동으로 인증할 수 있도록 하는 데 사용됩니다. 그런 다음 스크립트가 가져온 자격 증명을 사용하여 원격 시스템에 저장된 파일에 액세스할 수 있습니다.

IdM(Identity Management) 관리자는 다른 사용자가 **IdM**에서 실행되는 서비스에 대한 키탭을 검색하거나 만들 수 있습니다. 특정 사용자와 사용자 그룹이 키탭을 만들 수 있도록 허용하면 **IdM** 관리자 암호를 공유하지 않고 서비스 관리를 위임할 수 있습니다. 이 위임은 보다 세부적인 시스템 관리를 제공합니다.

이 섹션에서는 특정 **IdM** 사용자, 사용자 그룹, 호스트 및 호스트 그룹을 통해 **IdM** 클라이언트에서 실행되는 **HTTP** 서비스에 대한 키탭을 생성하는 방법을 설명합니다. 특히 **user01 IdM** 사용자가 **client.idm.example.com** 이라는 **IdM** 클라이언트에서 실행되는 **HTTP** 서비스에 대한 키탭을 생성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **HTTP** 서비스를 **IdM**에 등록했습니다.
- **HTTP** 서비스를 호스팅하는 시스템은 **IdM** 클라이언트입니다.
- 키탭을 생성할 수 있도록 허용하려는 **IdM** 사용자 및 사용자 그룹은 **IdM**에 있습니다.
- 키탭을 생성할 수 있도록 허용하려는 **IdM** 호스트 및 호스트 그룹은 **IdM**에 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: inventory.file:

```
$ touch inventory.file
```

2. inventory.file 을 열고 [ipaserver] 섹션에서 구성할 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 편집을 위해 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present-copy.yml Ansible 플레이북 파일을 엽니다.

5. 다음을 변경하여 파일을 조정합니다.

- ipaadmin_password 변수에 지정된 IdM 관리자 암호입니다.
- HTTP 서비스가 실행 중인 IdM 클라이언트의 이름입니다. 현재 예제에서는 HTTP/client.idm.example.com입니다.
- allow_create_keytab_user: 섹션에 나열된 IdM 사용자의 이름입니다. 현재 예제에서는 user01 입니다.
- allow_create_keytab_group: 섹션에 나열된 IdM 사용자 그룹의 이름입니다.
- allow_create_keytab_host: 섹션에 나열된 IdM 호스트의 이름입니다.

- **allow_create_keytab_hostgroup:** 섹션에 나열된 **IdM** 호스트 그룹의 이름입니다.

- **tasks** 섹션의 **name** 변수에서 지정한 작업의 이름입니다.

  현재 예제에 맞게 수정한 후 복사된 파일은 다음과 같습니다.

```
---
- name: Service member allow_create_keytab present
  hosts: ipaserver
  become: true

  tasks:
  - name: Service HTTP/client.idm.example.com members allow_create_keytab present
for user01
    ipaservice:
      ipaadmin_password: Secret123
      name: HTTP/client.idm.example.com
      allow_create_keytab_user:
      - user01
      action: member
```

6. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-member-
allow_create_keytab-present-copy.yml
```

검증 단계

1. 특정 **HTTP** 서비스에 대한 **keytab**을 생성할 권한이 있는 **IdM** 사용자로 **IdM** 서버에 **SSH** 연결을 수행합니다.

```
$ ssh user01@server.idm.example.com
Password:
```

2. **ipa-getkeytab** 명령을 사용하여 **HTTP** 서비스에 대한 새 **keytab**을 생성합니다.

```
$ ipa-getkeytab -s server.idm.example.com -p HTTP/client.idm.example.com -k
/etc/httpd/conf/krb5.keytab
```

s 옵션은 키 탭을 생성할 **KDC**(키 배포 센터) 서버를 지정합니다.

p 옵션은 생성하려는 주체를 지정합니다.

k 옵션은 새 키를 추가할 키탭 파일을 지정합니다. 파일이 없는 경우 생성됩니다.

명령이 오류가 발생하지 않으면 **user01** 로 **HTTP/client.idm.example.com** 의 키탭을 성공적으로 생성했습니다.

23.6. ANSIBLE 플레이북을 사용하여 IDM 사용자, 그룹, 호스트 또는 호스트 그룹을 통해 서비스의 키탭 검색 가능

keytab은 **Kerberos** 주체와 암호화된 키의 쌍을 포함하는 파일입니다. 키탭 파일은 일반적으로 일반 텍스트 파일에 저장된 암호나 사람의 상호 작용 없이도 스크립트를 **Kerberos**로 자동으로 인증할 수 있도록 하는 데 사용됩니다. 그런 다음 스크립트가 가져온 자격 증명을 사용하여 원격 시스템에 저장된 파일에 액세스할 수 있습니다.

IdM 관리자는 다른 사용자가 **IdM**에서 실행되는 서비스에 대한 키탭을 검색하거나 만들 수 있습니다.

이 섹션에서는 특정 **IdM** 사용자, 사용자 그룹, 호스트 및 호스트 그룹을 통해 **IdM** 클라이언트에서 실행되는 **HTTP** 서비스의 키탭을 검색하는 방법을 설명합니다. 특히 **user01 IdM** 사용자가 **client.idm.example.com** 에서 실행되는 **HTTP** 서비스의 **keytab**을 검색할 수 있도록 하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다.
- **HTTP** 서비스를 **IdM**에 등록했습니다.
- 키탭을 검색하려는 **IdM** 사용자 및 사용자 그룹은 **IdM**에 있습니다.

- 키탭을 검색할 수 있도록 허용하려는 IdM 호스트 및 호스트 그룹은 IdM에 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: inventory.file:

```
$ touch inventory.file
```

2. inventory.file 을 열고 [ipaserver] 섹션에서 구성할 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retmasterve_keytab-present.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 편집을 위해 복사된 파일 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retmasterve_keytab-present-copy.yml 을 엽니다.

5. 파일을 조정합니다.

   - ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.
   - ipaservice 작업의 name 변수를 HTTP 서비스의 주체로 설정합니다. 현재 예제에서는 HTTP/client.idm.example.com입니다.
   - allow_retmasterve_keytab_group: 섹션에서 IdM 사용자의 이름을 지정합니다. 현재 예제에서는 user01 입니다.

- **allow_retmasterve_keytab_group:** 섹션에서 **IdM** 사용자 그룹의 이름을 지정합니다.
- **allow_retmasterve_keytab_group:** 섹션에서 **IdM** 호스트의 이름을 지정합니다.
- **allow_retmasterve_keytab_group:** 섹션에서 **IdM** 호스트 그룹의 이름을 지정합니다.
- **tasks** 섹션에서 **name** 변수를 사용하여 작업 이름을 지정합니다.

  현재 예제에 맞게 수정한 후 복사된 파일은 다음과 같습니다.

```
---
- name: Service member allow_retrieve_keytab present
  hosts: ipaserver
  become: true

  tasks:
  - name: Service HTTP/client.idm.example.com members allow_retrieve_keytab
present for user01
    ipaservice:
      ipaadmin_password: Secret123
      name: HTTP/client.idm.example.com
      allow_retrieve_keytab_user:
      - user01
      action: member
```

6. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-member-
allow_retrieve_keytab-present-copy.yml
```

검증 단계

1. **HTTP** 서비스의 **keytab**을 검색할 권한이 있는 **IdM** 사용자로 **IdM** 서버에 **SSH**로 연결합니다.

```
$ ssh user01@server.idm.example.com
Password:
```

2. **ipa-getkeytab** 명령을 **-r** 옵션과 함께 사용하여 **keytab**을 검색합니다.

```
$ ipa-getkeytab -r -s server.idm.example.com -p HTTP/client.idm.example.com -k
/etc/httpd/conf/krb5.keytab
```

s 옵션은 키 탭을 검색할 **KDC**(키 배포 센터) 서버를 지정합니다.

p 옵션은 검색할 키탭의 주체를 지정합니다.

k 옵션은 검색된 키를 추가할 **keytab** 파일을 지정합니다. 파일이 없는 경우 생성됩니다.

명령이 오류가 발생하지 않으면 **user01** 로 **HTTP/client.idm.example.com** 의 키탭을 성공적으로 불러왔습니다.

23.7. ANSIBLE PLAYBOOK을 사용하여 서비스의 KERBEROS 기본 별칭이 있는지 확인

일부 시나리오에서는 **IdM** 관리자가 **Kerberos** 주체 별칭을 사용하여 **IdM** 사용자, 호스트 또는 서비스를 **Kerberos** 애플리케이션에 인증할 수 있도록 하는 것이 좋습니다. 이러한 시나리오는 다음과 같습니다.

- 사용자 이름이 변경되었지만 사용자는 이전 및 새 사용자 이름을 모두 사용하여 시스템에 로그인할 수 있어야 합니다.
- **IdM Kerberos** 영역이 이메일 도메인과 다른 경우에도 사용자는 이메일 주소를 사용하여 로그인해야 합니다.

이 섹션에서는 **client.idm.example .com**에서 실행되는 **HTTP** 서비스에 대해 **HTTP/mycompany.idm.example.com** 의 주요 별칭을 생성하는 방법을 설명합니다.

사전 요구 사항

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

- 호스트에 HTTP 서비스를 설정했습니다.
- HTTP 서비스를 IdM에 등록했습니다.
- HTTP를 설정한 호스트는 IdM 클라이언트입니다.

절차

1. 인벤토리 파일을 생성합니다(예: inventory.file:

```
$ touch inventory.file
```

2. inventory.file 을 열고 [ipaserver] 섹션에서 구성할 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. /usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 편집을 위해 /usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present-copy.yml Ansible 플레이북 파일을 엽니다.
5. 다음을 변경하여 파일을 조정합니다.
   - ipaadmin_password 변수에 지정된 IdM 관리자 암호입니다.
   - name 변수에서 지정한 서비스의 이름입니다. 이는 서비스의 정식 주체 이름입니다. 현재 예제에서는 HTTP/client.idm.example.com 입니다.

- **principal** 변수에서 지정한 **Kerberos** 주체 별칭입니다. **name** 변수로 정의한 서비스에 추가하려는 별칭입니다. 현재 예제에서는 **host/mycompany.idm.example.com** 입니다.

- **tasks** 섹션의 **name** 변수에서 지정한 작업의 이름입니다.

  현재 예제에 맞게 수정한 후 복사된 파일은 다음과 같습니다.

```
---
- name: Service member principal present
  hosts: ipaserver
  become: true

  tasks:
  - name: Service HTTP/client.idm.example.com member principals
host/mycompany.idm.exmaple.com present
    ipaservice:
      ipaadmin_password: Secret123
      name: HTTP/client.idm.example.com
      principal:
        - host/mycompany.idm.example.com
      action: member
```

6. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present-
copy.yml
```

플레이북을 실행하면 **0**에 연결할 수 없고 **0**개의 실패한 작업이 생성되면 **HTTP /client.idm.example.com** 서비스에 대한 **host/mycompany.idm.example.com Kerberos** 사용자가 성공적으로 생성되었습니다.

추가 리소스

- **Kerberos** 주체 별칭에 대한 자세한 내용은 사용자, 호스트 및 서비스에 대한 **Kerberos** 주체 별칭 관리를 참조하십시오.

23.8. ANSIBLE 플레이북을 사용하여 IDM에 HTTP 서비스가 없는지 확인

이 섹션에서는 IdM에서 서비스 등록을 취소하는 방법에 대해 설명합니다. 보다 구체적으로는 Ansible 플레이북을 사용하여 IdM에 HTTP /client.idm.example.com이라는 HTTP 서버가 없는지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호가 있습니다.

절차

1. 인벤토리 파일을 생성합니다(예: inventory.file:

```
$ touch inventory.file
```

2. inventory.file 을 열고 [ipaserver] 섹션에서 구성할 IdM 서버를 정의합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. /usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

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

4. 편집을 위해 /usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent-copy.yml Ansible 플레이북 파일을 엽니다.

5. 다음을 변경하여 파일을 조정합니다.

- ipaadmin_password 변수로 정의된 IdM 관리자 암호입니다.
- ipaservice 작업의 name 변수에 정의된 HTTP 서비스의 Kerberos 주체입니다.

현재 예제에 맞게 수정한 후 복사된 파일은 다음과 같습니다.

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure service is absent
  - ipaservice:
      ipaadmin_password: Secret123
      name: HTTP/client.idm.example.com
      state: absent
```

6. 파일을 저장하고 종료합니다.

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

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
/usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent-copy.yml
```

검증 단계

1. **IdM** 관리자로 **IdM** 웹 **UI**에 로그인합니다.

2. **ID** → 서비스로 이동합니다.

서비스 목록에 **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** 서비스가 표시되지 않으면 **IdM**에 없는지 확인해야 합니다.

추가 리소스

- **/usr/share/doc/ansible-freeipa/** 디렉터리에 있는 **README-service.md Markdown** 파일에서 가능한 변수 목록을 포함하여 **IdM**에 서비스가 있는지 샘플 **Ansible** 플레이북을 볼 수 있습니다.

- **/usr/share/doc/ansible-freeipa/playbooks/config** 디렉터리에서 **IdM**에 서비스가 있는지 확인하는 샘플 **Ansible** 플레이북을 확인할 수 있습니다.

24장. ANSIBLE 플레이북을 사용하여 IDM에서 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 해당 IP 주소 및 통신에 사용되는 포트입니다.
- 글로벌 전달 정책: only, first 또는 none. 이러한 유형의 DNS 전달 정책에 대한 자세한 내용은 IdM의 DNS 전달 정책을 참조하십시오.
- 정방향 조회 및 역방향 조회 영역의 동기화.

사전 요구 사항

- DNS 서비스는 IdM 서버에 설치되어 있습니다. 통합된 DNS로 IdM 서버를 설치하는 방법에 대한 자세한 내용은 다음 링크 중 하나를 참조하십시오.
  - IdM 서버 설치: 통합 DNS, 루트 CA로 통합 CA 포함
  - IdM 서버 설치: 통합 DNS, 외부 CA가 루트 CA로 포함
  - IdM 서버 설치: CA 없이 통합 DNS 사용

이 장에는 다음 섹션이 포함되어 있습니다.

- IdM을 통해 NetworkManager에서 /etc/resolv.conf의 글로벌 전달자를 제거하지 않는 방법

- Ansible을 사용하여 IdM에 DNS 글로벌 전달자가 있는지 확인
- Ansible을 사용하여 IdM에 DNS 글로벌 전달자가 없는지 확인
- IdM의 DNS 전달 정책소개
- Ansible 플레이북을 사용하여 IdM DNS 글로벌 구성에 전달 첫 번째 정책이 설정되었는지 확인합니다.
- Ansible 플레이북을 사용하여 글로벌 전달자가 IdM DNS에서 비활성화되었는지 확인합니다.
- Ansible 플레이북을 사용하여 IdM DNS에서 정방향 및 역방향 조회 영역의 동기화가 비활성화되었는지 확인합니다.

24.1. IDM을 통해 NETWORKMANAGER에서 /ETC/RESOLV.CONF의 글로벌 전달자를 제거하지 않는 방법

통합 DNS로 IdM(Identity Management)을 설치하면 127.0.0.1 localhost 주소를 가리키도록 /etc/resolv.conf 파일을 구성합니다.

```
# Generated by NetworkManager
search idm.example.com
nameserver 127.0.0.1
```

DHCP(Dynamic Host Configuration Protocol) 를 사용하는 네트워크와 같은 특정 환경에서 NetworkManager 서비스는 /etc/resolv.conf 파일의 변경 사항을 되돌릴 수 있습니다. DNS 구성을 영구적으로 만들기 위해 IdM DNS 설치 프로세스는 다음과 같은 방식으로 NetworkManager 서비스를 구성합니다.

1. DNS 설치 스크립트는 검색 순서 및 DNS 서버 목록을 제어하는 /etc/NetworkManager/conf.d/zzz-ipa.conf NetworkManager 구성 파일을 생성합니다.

```
# auto-generated by IPA installer
[main]
dns=default

[global-dns]
```

```
searches=$DOMAIN

[global-dns-domain-*]
servers=127.0.0.1
```

2. NetworkManager 서비스가 다시 로드됩니다. 이 서비스는 항상 /etc/NetworkManager/conf.d/ 디렉터리에 있는 마지막 파일에서 설정을 사용하여 /etc/resolv.conf 파일을 만듭니다. 이 경우 zzz-ipa.conf 파일입니다.

중요

/etc/resolv.conf 파일을 수동으로 수정하지 마십시오.

24.2. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에 DNS 글로벌 전달자가 있는지 확인하는 방법에 대해 설명합니다. 아래 예제 절차에서 IdM 관리자는 IP(인터넷 프로토콜) v4 주소가 7.7이고 IP v6 주소 2001:db8::1:0 에 DNS 글로벌 전달자가 DNS 서버에 있는지 확인합니다. 포트 53 에.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. forwarders-absent.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. 편집을 위해 ensure-presence-of-a-global-forwarder.yml 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

a. 플레이북의 name 변수를 플레이북으로 변경하여 IdM DNS에 글로벌 전달자가 있는지 확인합니다.

b. tasks 섹션에서 작업 이름을 변경하여 DNS 글로벌 전달자가 포트 53에 7.7. 및 2001:db8::1:0으로 있는지 확인합니다.

c. ipadnsconfig 부분의 forwarders 섹션에서 다음을 수행합니다.

i. 첫 번째 ip_address 값을 글로벌 전달자의 IPv4 주소로 변경합니다. 7.7.9.9.

ii. 두 번째 ip_address 값을 글로벌 전달자의 IPv6 주소로 변경합니다. 2001:db8::1:0.

iii. port 값이 53 으로 설정되어 있는지 확인합니다.

d. 상태를 present 로 변경합니다.

이 파일은 현재 예제에 맞게 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to ensure the presence of a global forwarder in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0
```

```
on port 53
   ipadnsconfig:
     forwarders:
       - ip_address: 7.7.9.9
       - ip_address: 2001:db8::1:0
         port: 53
     state: present
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-presence-of-a-global-forwarder.yml
```

추가 리소스

- /usr/share/doc/ansible -freeipa/ 디렉터리에서 사용할 수 있는 README-dnsconfig.md Markdown 파일에서 ansible -freeipa ipadnsconfig 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 ipadnsconfig 변수 정의도 포함되어 있습니다.

24.3. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 없는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에 DNS 글로벌 전달자가 없는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 IP(인터넷 프로토콜) v4 주소 8.8.6.6 및 IP v6 주소 2001:4860:4860::8800 의 DNS 글로벌 전달자가 없는지 확인합니다 .

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2.

인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. forwarders-absent.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. 편집을 위해 ensure-absence-of-a-global-forwarder.yml 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

   a. 플레이북의 name 변수를 플레이북으로 변경하여 IdM DNS에서 글로벌 전달자가 없는지 확인합니다.

   b. tasks 섹션에서 DNS 글로벌 전달자가 없는지 확인하고 포트 53에 2001:4860:4860::8800 으로 작업 이름을 변경합니다.

   c. ipadnsconfig 부분의 forwarders 섹션에서 다음을 수행합니다.

      i. 첫 번째 ip_address 값을 글로벌 전달자의 IPv4 주소로 변경합니다. 8.8.6.6.

      ii. 두 번째 ip_address 값을 글로벌 전달자의 IPv6 주소로 변경합니다. 2001:4860:4860::8800.

      iii. port 값이 53 으로 설정되어 있는지 확인합니다.

   d. 상태가 absent 로 설정되어 있는지 확인합니다.

   이 파일은 현재 예제에 맞게 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to ensure the absence of a global forwarder in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure the absence of a DNS global forwarder to 8.8.6.6 and
2001:4860:4860::8800 on port 53
    ipadnsconfig:
      forwarders:
        - ip_address: 8.8.6.6
        - ip_address: 2001:4860:4860::8800
          port: 53
      state: absent
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-absence-of-a-global-forwarder.yml
```

추가 리소스

- /usr/share/doc/ansible -freeipa/ 디렉터리에서 사용할 수 있는 README-dnsconfig.md Markdown 파일에서 ansible -freeipa ipadnsconfig 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 ipadnsconfig 변수 정의도 포함되어 있습니다.

24.4. IDM의 DNS 전달 정책

IdM은 첫 번째 및 표준 BIND 전달 정책과 none IdM별 전달 정책을 지원합니다.

전달 먼저 *(기본값)*

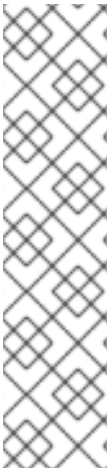
IdM BIND 서비스는 DNS 쿼리를 구성된 전달자로 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 BIND는 인터넷에서 서버를 사용하여 재귀적 해결으로 대체합니다. forward first 정책은 기본 정책이며 DNS 트래픽을 최적화하는 데 적합합니다.

앞으로만

IdM BIND 서비스는 DNS 쿼리를 구성된 전달자로 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 BIND에서 클라이언트에 오류를 반환합니다. DNS 구성이 분할된 환경에 만 전달 정책이 권장됩니다.

없음 *(전달 비활성화)*

DNS 쿼리는 none 전달 정책으로 전달되지 않습니다. 전달 비활성화는 전역 전달 구성의 영역별 재정의에서만 유용합니다. 이 옵션은 BIND 구성에서 비어 있는 전달자 목록을 지정하는 것과 동일한 IdM입니다.



참고

전달을 사용하여 IdM의 데이터를 다른 DNS 서버의 데이터와 결합할 수 없습니다. IdM DNS에서 기본 영역의 특정 하위 영역에 대한 쿼리만 전달할 수 있습니다.

기본적으로 쿼리된 DNS 이름이 IdM 서버에 권한이 있는 영역에 속하는 경우 BIND 서비스는 다른 서버로 쿼리를 전달하지 않습니다. 이러한 상황에서 쿼리된 DNS 이름을 IdM 데이터베이스에서 찾을 수 없는 경우 NXDOMAIN 응답이 반환됩니다. 전달은 사용되지 않습니다.

예 24.1. 시나리오 예

IdM 서버에는 test.example에 대한 권한이 있습니다. DNS 영역. BIND는 192.0.2.254 IP 주소를 사용하여 DNS 서버로 쿼리를 전달하도록 구성됩니다.

클라이언트가 존재하지 않는.test.example에 대한 쿼리를 보낼 때. DNS 이름 BIND는 IdM 서버가 test.example. 영역에 대해 권한이 있음을 감지하고 192. 0.2.254. 서버로 쿼리를 전달하지 않습니다. 결과적으로 DNS 클라이언트는 NXDomain 오류 메시지를 수신하여 쿼리된 도메인이 없음을 사용자에게 알립니다.

24.5. ANSIBLE 플레이북을 사용하여 IDM DNS 글로벌 구성에 전달 첫 번째 정책이 설정되었는지 확인합니다.

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM DNS의 글로벌 전달 정책이 먼저 전달되도록 설정하는 방법에 대해 설명합니다.

전달 첫 번째 DNS 전달 정책을 사용하는 경우 DNS 쿼리가 구성된 전달자로 전달됩니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 BIND는 인터넷에서 서버를 사용하여 재귀적 해결으로 대체합니다. forward first 정책은 기본 정책입니다. 트래픽 최적화에 적합합니다.

사전 요구 사항

- 절차를 실행하는 호스트인 Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 자세한 내용은 ansible-freeipa 패키지 설치를 참조하십시오.

- IdM 관리자 암호를 알고 있습니다.
- IdM 환경에는 통합된 DNS 서버가 포함되어 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. set-configuration.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp set-configuration.yml set-forward-policy-to-first.yml
```

4. 편집을 위해 set-forward-policy-to-first.yml 파일을 엽니다.

5. ipadnsconfig 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.
- forward_policy 변수를 먼저 로 설정합니다.

원본 플레이북의 다른 모든 행을 관련성이 없는 모든 행을 삭제합니다. 이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to set global forwarding policy to first
```

```
hosts: ipaserver
become: true

tasks:
- name: Set global forwarding policy to first.
  ipadnsconfig:
    ipaadmin_password: Secret123
    forward_policy: first
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file set-forward-policy-to-first.yml
```

추가 리소스

- IdM DNS에서 사용 가능한 전달 정책 유형에 대한 자세한 내용은 IdM의 DNS 전달 정책을 참조하십시오.
- ansible-freeipa ipadnsconfig 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README- dnsconfig.md Markdown 파일을 참조하십시오. 파일에는 ipadnsconfig 변수 정의도 포함되어 있습니다.
- ipadnsconfig 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리를 참조하십시오.

24.6. ANSIBLE 플레이북을 사용하여 글로벌 전달자가 IDM DNS에서 비활성화되었는지 확인합니다.

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM DNS에서 글로벌 전달자가 비활성화되는 방법을 설명합니다. 비활성화는 forward_policy 변수를 none 으로 설정하여 수행됩니다.

전역 전달자를 비활성화하면 DNS 쿼리가 전달되지 않습니다. 전달 비활성화는 전역 전달 구성의 영역별 재정의에서만 유용합니다. 이 옵션은 BIND 구성에서 비어 있는 전달자 목록을 지정하는 것과 동일한 IdM입니다.

사전 요구 사항

- 절차를 실행하는 호스트인 Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 자

세한 내용은 ansible-freeipa 패키지 설치를 참조하십시오.

- IdM 관리자 암호를 알고 있습니다.
- IdM 환경에는 통합된 DNS 서버가 포함되어 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. disable-global-forwarders.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp disable-global-forwarders.yml disable-global-forwarders-copy.yml
```

4. 편집할 disable-global-forwarders-copy.yml 파일을 엽니다.

5. ipadnsconfig 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

   - ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.
   - forward_policy 변수를 none 으로 설정합니다.

   이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to disable global DNS forwarders
```

```
hosts: ipaserver
become: true

tasks:
- name: Disable global forwarders.
  ipadnsconfig:
    ipaadmin_password: Secret123
    forward_policy: none
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file disable-global-forwarders-copy.yml
```

추가 리소스

- IdM DNS에서 사용 가능한 전달 정책 유형에 대한 자세한 내용은 IdM의 DNS 전달 정책을 참조하십시오.
- ansible-freeipa ipadnsconfig 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README- dnsconfig.md Markdown 파일을 참조하십시오. 파일에는 ipadnsconfig 변수 정의도 포함되어 있습니다.
- ipadnsconfig 모듈을 사용하는 샘플 Ansible 플레이북은 /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리를 참조하십시오.

24.7. ANSIBLE 플레이북을 사용하여 IDM DNS에서 정방향 및 역방향 조회 영역의 동기화가 비활성화되었는지 확인합니다.

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 정방향 및 역방향 조회 영역이 IdM DNS에서 동기화되지 않는 방법을 설명합니다.

사전 요구 사항

- 절차를 실행하는 호스트인 Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 자세한 내용은 ansible-freeipa 패키지 설치를 참조하십시오.

- IdM 관리자 암호를 알고 있습니다.
- IdM 환경에는 통합된 DNS 서버가 포함되어 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. disallow -reverse-sync.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp disallow-reverse-sync.yml disallow-reverse-sync-copy.yml
```

4. 편집을 위해 disallow-reverse-sync-copy.yml 파일을 엽니다.

5. ipadnsconfig 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

   - ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.
   - allow_sync_ptr 변수를 no 로 설정합니다.

   이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to disallow reverse record synchronization
  hosts: ipaserver
  become: true
```

```
tasks:
- name: Disallow reverse record synchronization.
  ipadnsconfig:
    ipaadmin_password: Secret123
    allow_sync_ptr: no
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file disallow-reverse-sync-copy.yml
```

추가 리소스

- **ansible-freeipa ipadnsconfig** 모듈을 사용하는 샘플 **Ansible** 플레이북은 **/usr/share/doc/ansible -freeipa/** 디렉터리에 있는 **README- dnsconfig.md Markdown** 파일을 참조하십시오. 파일에는 **ipadnsconfig** 변수 정의도 포함되어 있습니다.

- **ipadnsconfig** 모듈을 사용하는 샘플 **Ansible** 플레이북은 **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** 디렉터리를 참조하십시오.

25장. ANSIBLE 플레이북을 사용하여 IDM DNS 영역 관리

IdM(Identity Management) 관리자는 ansible-freeipa 패키지에서 사용할 수 있는 dnszone 모듈을 사용하여 IdM DNS 영역이 작동하는 방식을 관리할 수 있습니다. 이 장에서는 다음 주제 및 절차를 설명합니다.

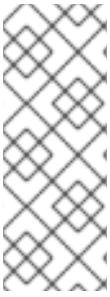
- IdM에서 지원되는 DNS 영역 유형은 무엇입니까?
- IdM에서 구성할 수 있는 DNS 속성은 무엇입니까?
- Ansible 플레이북을 사용하여 IdM DNS에서 기본 영역을 생성하는 방법
- Ansible 플레이북을 사용하여 여러 변수가 있는 기본 IdM DNS 영역이 있는지 확인하는 방법
- IP 주소가 제공될 때 Ansible 플레이북을 사용하여 역방향 DNS 조회 영역이 있는지 확인하는 방법

사전 요구 사항

- DNS 서비스는 IdM 서버에 설치되어 있습니다. Red Hat Ansible Engine을 사용하여 통합된 DNS로 IdM 서버를 설치하는 방법에 대한 자세한 내용은 Ansible 플레이북을 사용하여 ID 관리 서버 설치를 참조하십시오.

25.1. 지원되는 DNS 영역 유형

IdM(Identity Management)은 두 가지 유형의 DNS 영역, 즉 *기본* 및 *전달* 영역을 지원합니다. 이 섹션에서는 이러한 두 가지 유형의 영역에 대해 설명하고 DNS 전달 예제 시나리오를 포함합니다.



참고

이 가이드에서는 Microsoft Windows DNS에 사용되는 용어와 다른 영역 유형에 BIND 용어를 사용합니다. BIND의 기본 영역은 Microsoft Windows DNS에서 *정방향 조회 영역* 및 *역방향 조회 영역과* 동일한 목적을 제공합니다. BIND의 전달 영역은 Microsoft Windows DNS의 *조건부 전달자* 와 동일한 목적을 제공합니다.

기본 DNS 영역

기본 DNS 영역에는 권한 있는 DNS 데이터가 포함되어 있으며 동적 DNS 업데이트를 허용할 수 있습니다. 이 동작은 표준 BIND 구성의 유형 마스터 설정과 동일합니다. ipa dnszone-* 명령을 사용하여 기본 영역을 관리할 수 있습니다.

표준 DNS 규칙을 준수하려면 모든 기본 영역에 권한 시작 (SOA) 및 NS(이름 서버) 레코드가 포함되어야 합니다. DNS 영역을 생성할 때 IdM은 이러한 레코드를 자동으로 생성하지만 적절한 위임을 생성하려면 NS 레코드를 상위 영역에 수동으로 복사해야 합니다.

표준 BIND 동작에 따라 서버가 권한이 없는 이름에 대한 쿼리가 다른 DNS 서버로 전달됩니다. forwarders와 같은 이러한 DNS 서버는 쿼리에 대한 권한이 있을 수도 있고 그렇지 않을 수 있습니다.

> 예 25.1. DNS 전달 시나리오의 예
>
> IdM 서버에는 test.example. 기본 영역이 포함되어 있습니다. 이 영역에는 sub.test.example. 이름에 대한 NS 위임 레코드가 포함됩니다. 또한 test.example. 영역은 sub.text .example 하위 영역에 대해 192.0.2. 254 전달자 IP 주소로 구성됩니다.
>
> 비존재 .test.example. 이름을 쿼리하는 클라이언트는 NXDomain 응답을 수신하고 IdM 서버가 이 이름에 대한 권한이 있기 때문에 전달이 발생하지 않습니다.
>
> 반면, IdM 서버가 이 이름에 대한 권한이 없기 때문에 host1.sub.test.example . 이름을 쿼리하는 것은 구성된 forwarder 192.0.2.254 로 전달됩니다.

DNS 영역 전달

IdM의 관점에서, 전달 DNS 영역에는 권한 있는 데이터가 포함되어 있지 않습니다. 실제로 forward "zone"에는 일반적으로 두 가지 정보만 포함됩니다.

- 도메인 이름
- 도메인과 연결된 DNS 서버의 IP 주소

정의된 도메인에 속하는 이름에 대한 모든 쿼리는 지정된 IP 주소로 전달됩니다. 이 동작은 표준 BIND 구성의 type forward 설정과 동일합니다. ipa dnsforwardzone-* 명령을 사용하여 전달 영역을 관리할 수 있습니다.

정방향 DNS 영역은 IdM-AD(Active Directory) 신뢰 컨텍스트에서 특히 유용합니다. IdM DNS 서버에 idm.example.com 영역에 대한 권한이 있고 AD DNS 서버가 ad.example.com 영역에 대해 권한이 있는 경우 ad.example.com 은 idm.example.com 기본 영역의 DNS 전달 영역입니다. 즉, 쿼리가 somehost.ad.example.com의 IP 주소에 대한 IdM 클라이언트에서 제공되면 쿼리가 ad.example.com IdM DNS 전달 영역에 지정된 AD 도메인 컨트롤러로 전달됩니다.

25.2. 기본 IDM DNS 영역의 구성 특성

IdM(Identity Management)은 새로 고침 기간, 전송 설정 또는 캐시 설정과 같은 특정 기본 구성으로 새 영역을 생성합니다. IdM DNS 영역 속성 에서는 다음 옵션 중 하나를 사용하여 수정할 수 있는 기본 영역 구성의 특성을 찾을 수 있습니다.

- CLI(명령줄 인터페이스)의 dnszone-mod 명령. 자세한 내용은 IdM CLI에서 기본 DNS 영역 구성 편집을 참조하십시오.
- IdM 웹 UI. 자세한 내용은 IdM 웹 UI에서 기본 DNS 영역 구성 편집을 참조하십시오.
- ipadnszone 모듈을 사용하는 Ansible 플레이북. 자세한 내용은 IdM에서 DNS 영역 관리를 참조하십시오.

영역에 대한 실제 정보를 설정하는 것과 함께 설정은 DNS 서버가 *권한 시작* (SOA) 레코드 항목을 처리하는 방법과 DNS 이름 서버에서 레코드를 업데이트하는 방법을 정의합니다.

표 25.1. IdM DNS 영역 속성

속성	ansible-freeipa 변수	설명
인가된 이름 서버	**name_server**	SOA MNAME이라고도 하는 기본 DNS 이름 서버의 도메인 이름을 설정합니다. 기본적으로 각 IdM 서버는 SOA MNAME 필드에 자체적으로 알립니다. 결과적으로 **--name-server** 를 사용하여 LDAP에 저장된 값은 무시됩니다.
관리자 이메일 주소	**admin_email**	영역 관리자에게 사용할 이메일 주소를 설정합니다. 기본값은 호스트의 root 계정으로 설정됩니다.
SOA 직렬	**시리얼**	SOA 레코드에 일련 번호를 설정합니다. IdM은 버전 번호를 자동으로 설정하고 사용자가 수정하지 않아야 합니다.

속성	ansible-freeipa 변수	설명
SOA 새로 고침	**새로 고침**	보조 DNS 서버가 기본 DNS 서버에서 업데이트를 요청하기 전에 대기할 간격(초)을 설정합니다.
SOA 재시도	**재시도**	실패한 새로 고침 작업을 재시도하기 전에 대기할 시간(초)을 설정합니다.
SOA 만료	**expire**	보조 DNS 서버가 작업 시도를 종료하기 전에 새로 고침 업데이트를 수행하는 시간을 초 단위로 설정합니다.
SOA 최소	**최소**	RFC 2308 에 따라 음수 캐싱의 TTL(Time to Live) 값을 초 단위로 설정합니다.
SOA 수명	**ttl**	zone apex에서 레코드에 TTL(초)을 설정합니다. **예를 들어 zone example.com** 에서는 **example.com 이름의 모든 레코드(A, NS 또는 SOA)가 구성되지만 test. example.com** 과 같은 다른 도메인 이름은 영향을 받지 않습니다.
기본 수명	**default_ttl**	개별 TTL 값이 없는 영역의 모든 값에 대해 음수 캐싱을 위해 기본 TTL(Time to Live) 값을 초 단위로 설정합니다. 변경 사항을 적용한 후 모든 IdM DNS 서버에서 **named-pkcs11** 서비스를 다시 시작해야 합니다.
BIND 업데이트 정책	**update_policy**	DNS 영역에서 허용된 권한을 클라이언트에 설정합니다.
동적 업데이트	**dynamic_update**=TRUE|FALSE	클라이언트의 DNS 레코드에 대한 동적 업데이트를 활성화합니다. 이 값을 false로 설정하면 IdM 클라이언트 시스템에서 IP 주소를 추가하거나 업데이트할 수 없습니다.
전송 허용	**allow_transfer**=*string*	세미콜론(;)으로 구분된 지정된 영역을 전송할 수 있는 IP 주소 또는 네트워크 이름 목록을 제공합니다. 영역 전송은 기본적으로 비활성화되어 있습니다. 기본 **allow_transfer** 값은 **none** 입니다.
쿼리 허용	**allow_query**	세미콜론(;)으로 구분된 DNS 쿼리를 실행할 수 있는 IP 주소 또는 네트워크 이름 목록을 제공합니다.
PTR 동기화 허용	**allow_sync_ptr**=1|0	영역에 대한 A 또는 AAAA 레코드(전달 레코드)가 PTR(역방향) 레코드와 자동으로 동기화되는지 여부를 설정합니다.
영역 전달기	**forwarder**=*IP_address*	DNS 영역에 구체적으로 구성된 전달자를 지정합니다. IdM 도메인에서 사용되는 모든 글로벌 전달자와는 별개입니다. 여러 전달자를 지정하려면 옵션을 여러 번 사용합니다.

속성	ansible-freeipa 변수	설명
전달 정책	**forward_policy=** none|only|first	전달 정책을 지정합니다. 지원되는 정책에 대한 자세한 내용은 IdM의 DNS 전달 정책을 참조하십시오.

추가 리소스

- /usr/share/doc/ansible -freeipa/ 디렉토리에서 사용할 수 있는 **README-dnszone.md** Markdown 파일에서 **ansible -freeipa ipadnszone** 모듈 의 속성 정의가 더 많은 것을 확인할 수 있습니다.

25.3. ANSIBLE을 사용하여 IDM DNS에서 기본 영역 생성

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 기본 DNS 영역이 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 IdM 관리자는 zone.idm.example.com DNS 영역이 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnszone 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. dnszone-present.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp dnszone-present.yml dnszone-present-copy.yml
```

4. 편집할 **dnszone-present-copy.yml** 파일을 엽니다.

5. **ipadnszone** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

   - **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
   - **zone_name** 변수를 **zone.idm.example.com** 으로 설정합니다.

   이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure zone is present.
    ipadnszone:
      ipaadmin_password: Secret123
      zone_name: zone.idm.example.com
      state: present
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

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

추가 리소스

- **DNS** 영역에 대한 자세한 내용은 지원되는 **DNS** 영역 유형을 참조하십시오.
- **/usr/share/doc/ansible-freeipa -freeipa/** 디렉터리에서 사용할 수 있는 **README-dnszone.md Markdown** 파일에서 **ansible -freeipa ipadnszone** 모듈에 대한 더 많은 샘플

Ansible 플레이북을 볼 수 있습니다. 파일에는 ipadnszone 변수의 정의도 포함되어 있습니다.

- /usr/share/doc/ansible-freeipa/playbooks/dns zone 디렉터리에서 ipadns zone 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.

25.4. ANSIBLE 플레이북을 사용하여 여러 변수와 함께 IDM에 기본 DNS 영역이 있는지 확인합니다.

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 기본 DNS 영역이 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 IdM 관리자는 zone.idm.example.com DNS 영역이 있는지 확인합니다. Ansible 플레이북은 영역의 여러 매개 변수를 구성합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnszone 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. dnszone-all-params.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp dnszone-all-params.yml dnszone-all-params-copy.yml
```

4. 편집할 dnszone-all-params-copy.yml 파일을 엽니다.

5. **ipadnszone** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **zone_name** 변수를 **zone.idm.example.com** 으로 설정합니다.
- **PTR** 레코드와 **A** 및 **AAAA** 레코드의 동기화인 정방향 및 역방향 레코드의 동기화를 허용하려면 **allow_sync_ptr** 변수를 **true**로 설정합니다.
- **IdM** 클라이언트 시스템에서 **IP** 주소를 추가하거나 업데이트할 수 있도록 **dynamic_update** 변수를 **true**로 설정합니다.
- 영역에 있는 레코드의 인라인 **DNSSEC** 서명을 허용하려면 **dnssec** 변수를 **true**로 설정합니다.
- **allow_transfer** 변수를 영역의 보조 이름 서버의 **IP** 주소로 설정합니다.
- **allow_query** 변수를 쿼리를 발행할 수 있는 **IP** 주소 또는 네트워크로 설정합니다.
- **forwarders** 변수를 글로벌 전달자의 **IP** 주소로 설정합니다.
- **serial** 변수를 **SOA** 레코드 일련 번호로 설정합니다.
- 영역의 **DNS** 레코드에 대한 **refresh,retry,expire , minimum, ttl** 및 **default_ttl** 값을 정의합니다.
- **nsec3param_rec** 변수를 사용하여 영역에 대한 **NSEC3 PARAM** 레코드를 정의합니다.
- 기존 영역과 겹치는 경우에도 **DNS** 생성을 강제 하려면 **skip_overlap_check** 변수를 **true**로 설정합니다.

- 이름 서버를 확인할 수 없는 경우에도 **DNS** 영역을 강제로 생성하려면 **skip_nameserver_check** 를 **true**로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure zone is present.
    ipadnszone:
      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

- DNS 영역에 대한 자세한 내용은 지원되는 DNS 영역 유형을 참조하십시오.
- IdM에서 구성할 수 있는 DNS 영역 속성에 대한 자세한 내용은 기본 IdM DNS 영역의 구성 속성을 참조하십시오.
- /usr/share/doc/ansible-freeipa -freeipa/ 디렉터리에서 사용할 수 있는 README-dnszone.md Markdown 파일에서 ansible -freeipa ipadnszone 모듈에 대한 더 많은 샘플 Ansible 플레이북을 볼 수 있습니다. 파일에는 ipadnszone 변수의 정의도 포함되어 있습니다.
- /usr/share/doc/ansible-freeipa/playbooks/dns zone 디렉터리에서 ipadns zone 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.

25.5. IP 주소가 제공될 때 역방향 DNS 조회에 영역이 있는지 확인하는 데 ANSIBLE 플레이북을 사용합니다.

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 역방향 DNS 영역이 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 IdM 관리자는 IdM 호스트의 IP 주소 및 접두사 길이를 사용하여 역방향 DNS 조회 영역을 확인합니다.

name_from_ip 변수를 사용하여 DNS 서버의 IP 주소 접두사 길이를 제공하면 영역 이름을 제어할 수 있습니다. 접두사 길이를 설명하지 않으면 시스템은 DNS 서버에 영역에 대해 쿼리하고 name_from_ip 값 192.168.1.2 를 기준으로 쿼리에서 다음 DNS 영역을 반환할 수 있습니다.

- 1.168.192.in-addr.arpa.
- 168.192.in-addr.arpa.
- 192.in-addr.arpa.

쿼리에서 반환한 영역이 예상과 다를 수 있으므로 name_from_ip 는 실수로 영역이 제거되지 않도록 state 옵션과 함께만 사용할 수 있습니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnszone 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. dnszone-reverse-from-ip.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp dnszone-reverse-from-ip.yml dnszone-reverse-from-ip-copy.yml
```

4. 편집을 위해 dnszone-reverse-from-ip-copy.yml 파일을 엽니다.

5. ipadnszone 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

   - ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.
   - name_from_ip 변수를 IdM 이름 서버의 IP로 설정하고 해당 접두사 길이를 제공합니다.

   이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true
```

```
tasks:
- name: Ensure zone for reverse DNS lookup is present.
  ipadnszone:
    ipaadmin_password: Secret123
    name_from_ip: 192.168.1.2/24
    state: present
  register: result
- name: Display inferred zone name.
  debug:
    msg: "Zone name: {{ result.dnszone.name }}"
```

플레이북은 **192.168.1.2 IP** 주소 및 접두사 길이 **24**에서 역방향 **DNS** 조회 영역을 생성합니다. 그런 다음 플레이북에 결과 영역 이름이 표시됩니다.

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file dnszone-reverse-from-ip-copy.yml
```

추가 리소스

- **DNS** 영역에 대한 자세한 내용은 지원되는 **DNS** 영역 유형을 참조하십시오.

- **/usr/share/doc/ansible-freeipa -freeipa/** 디렉터리에서 사용할 수 있는 **README-dnszone.md Markdown** 파일에서 **ansible -freeipa ipadnszone** 모듈에 대한 더 많은 샘플 **Ansible** 플레이북을 볼 수 있습니다. 파일에는 **ipadnszone** 변수의 정의도 포함되어 있습니다.

- **/usr/share/doc/ansible-freeipa/playbooks/dns zone** 디렉터리에서 **ipadns zone** 모듈에 대한 샘플 **Ansible** 플레이북을 볼 수 있습니다.

26장. ANSIBLE을 사용하여 IDM에서 DNS 위치 관리

IdM(Identity Management) 관리자는 ansible-freeipa 패키지에서 사용할 수 있는 위치 모듈을 사용하여 IdM DNS 위치를 관리할 수 있습니다. 이 장에서는 다음 주제 및 절차를 설명합니다.

- DNS 기반 서비스 검색
- DNS 위치에 대한 배포 고려 사항
- DNS Time to Live (TTL)
- Ansible을 사용하여 IdM 위치가 있는지 확인
- Ansible을 사용하여 IdM 위치가 없는지 확인

26.1. DNS 기반 서비스 검색

DNS 기반 서비스 검색은 클라이언트에서 DNS 프로토콜을 사용하여 LDAP 또는 Kerberos 와 같은 특정 서비스를 제공하는 네트워크에서 서버를 찾는 프로세스입니다. 일반적인 작업 유형 중 하나는 클라이언트가 가장 가까운 네트워크 인프라 내에서 인증 서버를 찾을 수 있도록 허용하는 것입니다. 이는 처리량이 증가하고 네트워크 대기 시간이 단축되어 전체 비용이 절감되기 때문입니다.

서비스 검색의 주요 장점은 다음과 같습니다.

- 가까운 서버의 이름으로 클라이언트를 명시적으로 구성할 필요가 없습니다.
- DNS 서버는 정책의 중앙 공급업체로 사용됩니다. 동일한 DNS 서버를 사용하는 클라이언트는 서비스 프로바이더 및 해당 기본 순서에 대해 동일한 정책에 액세스할 수 있습니다.

IdM(Identity Management) 도메인에서는 LDAP,Kerberos 및 기타 서비스에 대한 SRV 레코드(DNS 서비스 레코드)가 있습니다. 예를 들어 다음 명령은 IdM DNS 도메인에서 TCP 기반 Kerberos 서비스를 제공하는 호스트를 DNS 서버에 쿼리합니다.

예 26.1. DNS 위치 독립 결과

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
0 100 88 idmserver-01.idm.example.com.
0 100 88 idmserver-02.idm.example.com.
```

출력에는 다음 정보가 포함됩니다.

- **0** (우선 순위): 대상 호스트의 우선 순위입니다. 더 낮은 값이 우선합니다.
- **100** (가벼움). 우선순위가 동일한 항목의 상대 가중치를 지정합니다. 자세한 내용은 RFC 2782, 섹션 3을 참조하십시오.
- **88** (포트 번호): 서비스의 포트 번호입니다.
- 서비스를 제공하는 호스트의 정식 이름입니다.

이 예에서 반환된 두 호스트 이름은 우선 순위와 가중치가 동일합니다. 이 경우 클라이언트는 결과 목록의 임의 항목을 사용합니다.

대신 클라이언트가 **DNS** 위치에 구성된 **DNS** 서버를 쿼리하도록 구성된 경우 출력은 다릅니다. 위치에 할당된 **IdM** 서버의 경우 맞춤형 값이 반환됩니다. 아래 예제에서 클라이언트는 위치 **germany** 에서 **DNS** 서버를 쿼리하도록 구성됩니다.

예 26.2. DNS 위치 기반 결과

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
_kerberos._tcp.germany._locations.idm.example.com.
0 100 88 idmserver-01.idm.example.com.
50 100 88 idmserver-02.idm.example.com.
```

IdM DNS 서버는 로컬 서버를 선호하는 **DNS** 위치별 **SRV** 레코드를 가리키는 **DNS** 별칭(**CNAME**)을 자동으로 반환합니다. 이 **CNAME** 레코드는 출력의 첫 번째 줄에 표시됩니다. 이 예제에서 호스트 **idmserver-01.idm.example.com** 은 우선 순위 값이 가장 낮으므로 선호됩니다. **idmserver-02.idm.example.com** 은 우선 순위가 높으므로 기본 호스트를 사용할 수 없는 경우에 대해서만 백업으로 사용됩니다.

26.2. DNS 위치에 대한 배포 고려 사항

IdM(Identity Management)은 통합 DNS를 사용할 때 SRV(위치별 서비스) 레코드를 생성할 수 있습니다. 각 IdM DNS 서버는 위치별 SRV 레코드를 생성하므로 각 DNS 위치에 하나 이상의 IdM DNS 서버를 설치해야 합니다.

DNS 위치에 대한 클라이언트의 선호도는 클라이언트가 수신한 DNS 레코드에서만 정의합니다. 이러한 이유로 IdM DNS 서버를 비IdM DNS 소비자 서버와 결합하고 DNS 서비스 검색을 수행하는 클라이언트가 IdM DNS 서버의 위치별 레코드를 분석하는 경우 재귀할 수 있습니다.

혼합 IdM 및 비IdM DNS 서비스가 있는 대부분의 배포에서 DNS 재귀는 왕복 시간 지표를 사용하여 가장 가까운 IdM DNS 서버를 자동으로 선택합니다. 일반적으로, IdM이 아닌 DNS 서버를 사용하는 클라이언트가 가장 가까운 DNS 위치에 대한 레코드를 확보하므로 최적의 IdM 서버 세트를 사용할 수 있습니다.

26.3. DNS TIME TO LIVE (TTL)

클라이언트는 영역 구성에 설정된 기간 동안 DNS 리소스 레코드를 캐시할 수 있습니다. 이 캐싱으로 인해 클라이언트는 TTL(Time To Live) 값이 만료될 때까지 클라이언트에서 변경 사항을 받지 못할 수 있습니다. IdM(Identity Management)의 기본 TTL 값은 1일입니다.

클라이언트 컴퓨터가 사이트 간에 로밍되는 경우 IdM DNS 영역의 TTL 값을 조정해야 합니다. 클라이언트가 사이트 간에 로밍해야 하는 시간보다 낮은 값으로 값을 설정합니다. 이렇게 하면 클라이언트에 캐시된 DNS 항목이 다른 사이트에 다시 연결되기 전에 만료되도록 하므로 DNS 서버를 쿼리하여 위치별 SRV 레코드를 새로 고칩니다.

추가 리소스

- DNS 영역의 기본 TTL을 수정하는 방법에 대한 자세한 내용은 기본 IdM DNS 영역의 구성 속성을 참조하십시오.

26.4. ANSIBLE을 사용하여 IDM 위치가 있는지 확인

IdM(Identity Management)의 시스템 관리자는 클라이언트가 가장 가까운 네트워크 인프라 내에서 인증 서버를 찾을 수 있도록 IdM DNS 위치를 구성할 수 있습니다.

다음 절차에서는 Ansible 플레이북을 사용하여 DNS 위치가 IdM에 있는지 확인하는 방법을 설명합니다. 이 예제에서는 IdM에 germany DNS 위치가 있는지 확인하는 방법을 설명합니다. 결과적으로 로컬

IdM 클라이언트가 이를 사용하여 서버 응답 시간을 줄일 수 있도록 특정 IdM 서버를 이 위치에 할당할 수 있습니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- Ansible 제어 노드에 ansible-freeipa 패키지를 설치했습니다.
- 이 예제에서는 ~/*MyPlaybooks*/ 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 만들고 구성 했다고 가정합니다.
- DNS 위치에 대한 배포 고려 사항을 이해합니다.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/location/ 디렉터리에 있는 location-present.yml 파일의 복사본을 만듭니다.

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

3. 편집할 location-present-copy.yml Ansible 플레이북 파일을 엽니다.
4. ipalocation 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - 사용 사례에 대응하도록 작업 이름을 조정합니다.
   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.

- name 변수를 위치 이름으로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

  tasks:
  - name: Ensure that the "germany" location is present
    ipalocation:
      ipaadmin_password: Secret123
      name: germany
```

5. 파일을 저장합니다.

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

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

추가 리소스

- IdM에 현재 존재하는 IdM 위치에 대한 특정 서버를 구성하려면 IdM 웹 UI를 사용하여 IdM 서버 할당 또는 IdM CLI를 사용하여 DNS 위치에 IdM 서버 할당을 참조하십시오.

26.5. ANSIBLE을 사용하여 IDM 위치가 없는지 확인

IdM(Identity Management)의 시스템 관리자는 클라이언트가 가장 가까운 네트워크 인프라 내에서 인증 서버를 찾을 수 있도록 IdM DNS 위치를 구성할 수 있습니다.

다음 절차에서는 Ansible 플레이북을 사용하여 IdM에 DNS 위치가 없는지 확인하는 방법을 설명합니다. 이 예제에서는 IdM에 germany DNS 위치가 없는지 확인하는 방법을 설명합니다. 따라서 특정 IdM 서버를 이 위치에 할당할 수 없으며 로컬 IdM 클라이언트는 사용할 수 없습니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.

- IdM 서버가 germany DNS 위치에 할당되지 않습니다.
- Ansible 제어 노드에 ansible-freeipa 패키지를 설치했습니다.
- 이 예제에서는 ~/*MyPlaybooks*/ 디렉터리를 샘플 플레이북 복사본을 저장할 중앙 위치로 만들고 구성 했다고 가정합니다.

절차

1. ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/location/ 디렉터리에 있는 location-absent.yml 파일의 복사본을 만듭니다.

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

3. 편집할 location-absent-copy.yml Ansible 플레이북 파일을 엽니다.
4. ipalocation 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - 사용 사례에 대응하도록 작업 이름을 조정합니다.
   - ipaadmin_password 변수를 IdM 관리자의 암호로 설정합니다.
   - name 변수를 DNS 위치의 이름으로 설정합니다.
   - state 변수가 absent 로 설정되어 있는지 확인합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: location absent example
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure that the "germany" location is absent
    ipalocation:
      ipaadmin_password: Secret123
      name: germany
      state: absent
```

5. 파일을 저장합니다.

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

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

26.6. 추가 리소스

- /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README-location.md 파일에서 ansible -freeipa ipalocation 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 ipalocation 변수의 정의도 포함되어 있습니다.

- /usr/share/doc/ansible-freeipa/playbooks/ location 디렉터리에서 ipa location 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.

27장. IDM에서 DNS 전달 관리

다음 절차에서는 IdM(Identity Management) 웹 UI, IdM CLI 및 Ansible을 사용하여 DNS 글로벌 전달자 및 DNS 전달 영역을 구성하는 방법을 설명합니다.

- 27.13절. “Ansible을 사용하여 IdM에서 DNS 전달 영역이 비활성화되어 있는지 확인”
- 27.14절. “Ansible을 사용하여 IdM에 DNS 전달 영역이 없는지 확인”

27.1. IDM DNS 서버의 두 가지 역할

DNS 전달은 DNS 서비스가 DNS 쿼리에 응답하는 방법에 영향을 미칩니다. 기본적으로 IdM과 통합된 Berkeley Internet Name Domain(BIND) 서비스는 *권한 있는* DNS 서버 및 *재귀적* DNS 서버 역할을 합니다.

인가된 DNS 서버

DNS 클라이언트가 IdM 서버에 권한이 있는 DNS 영역에 속하는 이름을 쿼리하면 BIND에서 구성된 영역에 포함된 데이터로 응답합니다. 신뢰할 수 있는 데이터는 항상 다른 데이터보다 우선합니다.

재귀 DNS 서버

DNS 클라이언트가 IdM 서버에 권한이 없는 이름을 쿼리하면 BIND에서 다른 DNS 서버를 사용하여 쿼리를 확인합니다. 전달자가 정의되지 않은 경우 BIND는 인터넷의 루트 서버에 요청하고 재귀적 확인 알고리즘을 사용하여 DNS 쿼리에 응답합니다.

경우에 따라 BIND가 다른 DNS 서버에 직접 연락하고 인터넷에서 사용 가능한 데이터를 기반으로 재귀를 수행하는 것이 바람직하지 않습니다. 쿼리를 확인하기 위해 다른 DNS 서버인 ***forwarder*** 를 사용하도록 BIND를 구성할 수 있습니다.

전달자를 사용하도록 BIND를 구성하면 IdM 서버와 전달자 간에 쿼리 및 응답을 앞뒤로 전달하고 IdM 서버는 권한이 없는 데이터의 DNS 캐시 역할을 합니다.

27.2. IDM의 DNS 전달 정책

IdM은 첫 번째 및 표준 BIND 전달 정책과 none IdM별 전달 정책을 지원합니다.

전달 먼저 *(기본값)*

IdM BIND 서비스는 DNS 쿼리를 구성된 전달자로 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 BIND는 인터넷에서 서버를 사용하여 재귀적 해결으로 대체합니다. forward first 정책은 기본 정책이며 DNS 트래픽을 최적화하는 데 적합합니다.

앞으로만

IdM BIND 서비스는 DNS 쿼리를 구성된 전달자로 전달합니다. 서버 오류 또는 시간 초과로 인해 쿼리가 실패하면 BIND에서 클라이언트에 오류를 반환합니다. DNS 구성이 분할된 환경에 만 전달 정책이 권장됩니다.

없음 *(전달 비활성화)*

DNS 쿼리는 none 전달 정책으로 전달되지 않습니다. 전달 비활성화는 전역 전달 구성의 영역별 재정의에서만 유용합니다. 이 옵션은 BIND 구성에서 비어 있는 전달자 목록을 지정하는 것과 동일한 IdM입니다.

참고

전달을 사용하여 IdM의 데이터를 다른 DNS 서버의 데이터와 결합할 수 없습니다. IdM DNS에서 기본 영역의 특정 하위 영역에 대한 쿼리만 전달할 수 있습니다.

기본적으로 쿼리된 DNS 이름이 IdM 서버에 권한이 있는 영역에 속하는 경우 BIND 서비스는 다른 서버로 쿼리를 전달하지 않습니다. 이러한 상황에서 쿼리된 DNS 이름을 IdM 데이터베이스에서 찾을 수 없는 경우 NXDOMAIN 응답이 반환됩니다. 전달은 사용되지 않습니다.

예 27.1. 시나리오 예

IdM 서버에는 test.example에 대한 권한이 있습니다. DNS 영역. BIND는 192.0.2.254 IP 주소를 사용하여 DNS 서버로 쿼리를 전달하도록 구성됩니다.

클라이언트가 존재하지 않는.test.example에 대한 쿼리를 보낼 때. DNS 이름 BIND는 IdM 서버가 test.example. 영역에 대해 권한이 있음을 감지하고 192. 0.2.254. 서버로 쿼리를 전달하지 않습니다. 결과적으로 DNS 클라이언트는 NXDomain 오류 메시지를 수신하여 쿼리된 도메인이 없음을 사용자에게 알립니다.

27.3. IDM 웹 UI에서 글로벌 전달자 추가

이 섹션에서는 IdM(Identity Management) 웹 UI에서 글로벌 DNS 전달자를 추가하는 방법에 대해 설명합니다.

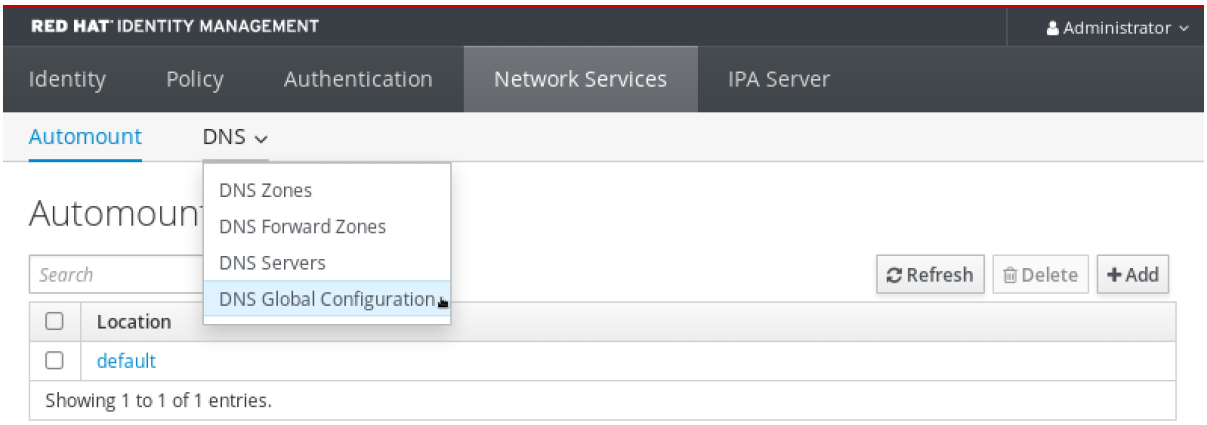
사전 요구 사항

- IdM 관리자로 IdM WebUI에 로그인했습니다.

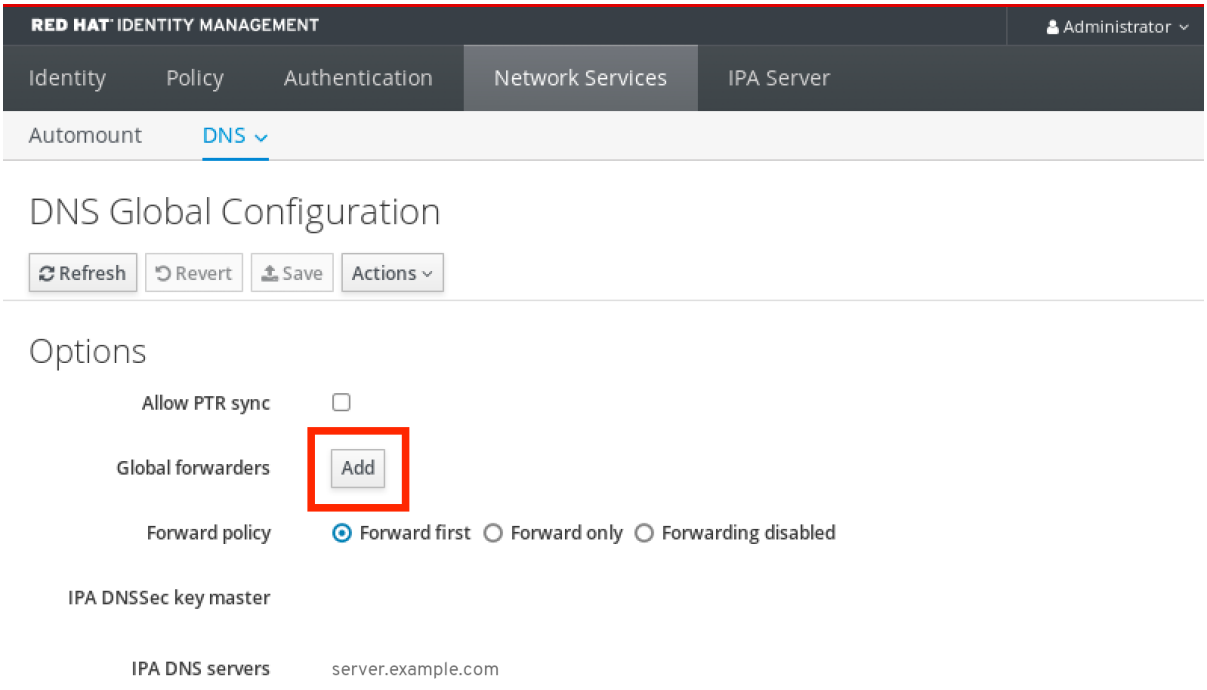
- 쿼리를 전달할 **DNS** 서버의 **IP(Internet Protocol)** 주소를 알고 있습니다.

절차

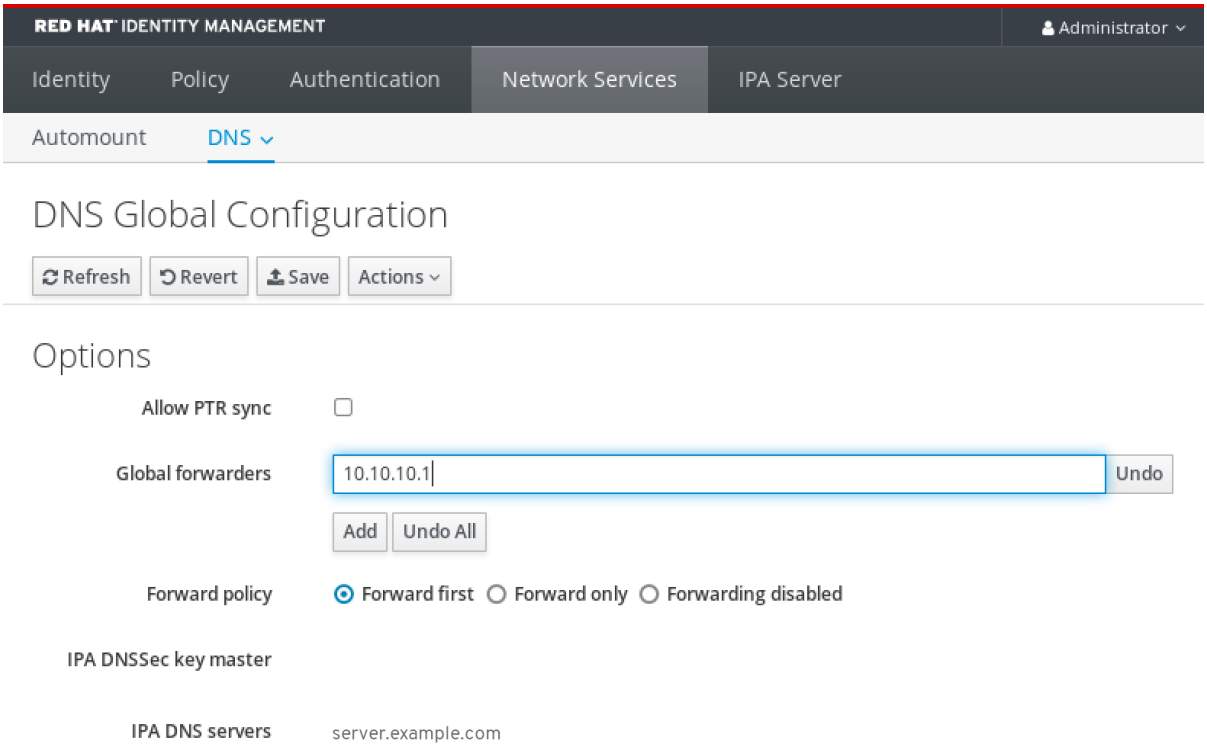
1. **IdM** 웹 **UI**에서 네트워크 서비스 → **DNS** 글로벌 구성 → **DNS** 를 선택합니다.



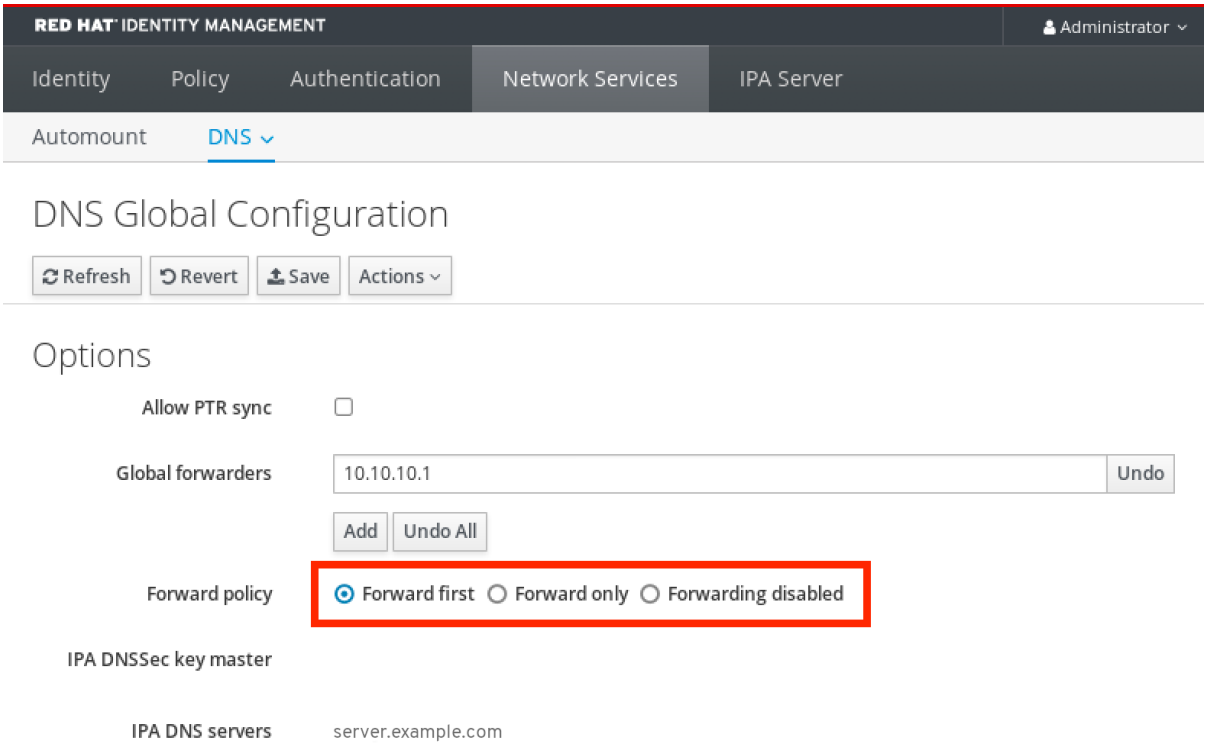
2. **DNS Global Configuration(DNS** 글로벌 구성) 섹션에서 **Add(**추가 **)**를 클릭합니다.



3. 전달된 **DNS** 쿼리를 수신할 **DNS** 서버의 **IP** 주소를 지정합니다.



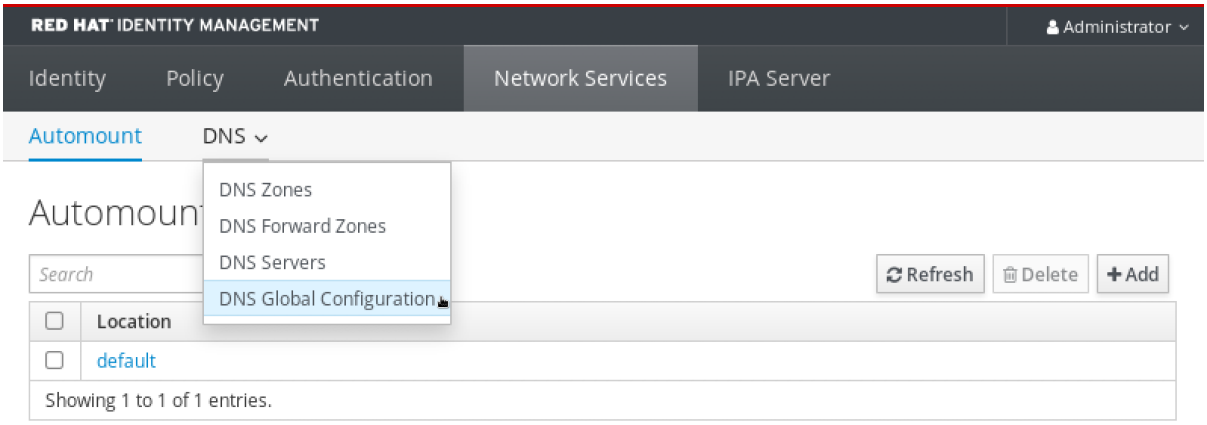
4. **Forward policy(전달 정책을 선택합니다.**



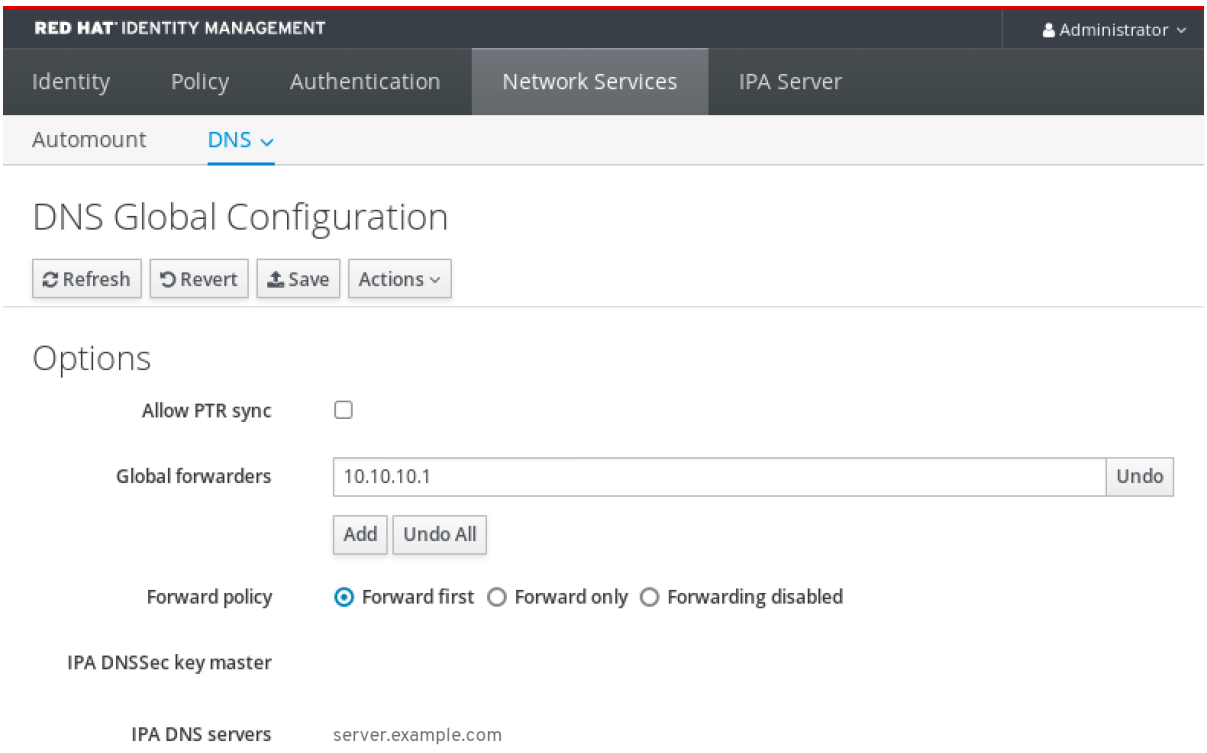
5. **창 맨 위에 있는 Save(저장)를 클릭합니다.**

검증 단계

1. 네트워크 서비스 → **DNS** 글로벌 구성 → **DNS** 를 선택합니다.



2. 지정한 전달 정책이 있는 글로벌 전달자가 **IdM** 웹 **UI**에 있고 활성화되어 있는지 확인합니다.



27.4. CLI에서 글로벌 전달자 추가

이 섹션에서는 **CLI(명령줄 인터페이스)**에서 글로벌 **DNS** 전달자를 추가하는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.

- 쿼리를 전달할 DNS 서버의 IP(Internet Protocol) 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
```

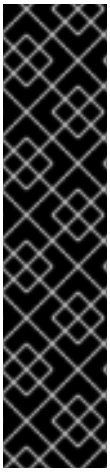
검증 단계

- dnsconfig-show 명령을 사용하여 글로벌 전달자를 표시합니다.

```
[user@server ~]$ ipa dnsconfig-show
  Global forwarders: 10.10.0.1
  IPA DNS servers: server.example.com
```

27.5. IDM 웹 UI에서 DNS 전달 영역 추가

이 섹션에서는 IdM(Identity Management) 웹 UI에 DNS 전달 영역을 추가하는 방법을 설명합니다.



중요

필요하지 않은 한 전달 영역을 사용하지 마십시오. 전달 영역은 표준 솔루션이 아니며 이를 사용하면 예기치 않고 문제가 있는 동작이 발생할 수 있습니다. 전달 영역을 사용해야 하는 경우 글로벌 전달 구성을 재정의하도록 사용을 제한합니다.

새 DNS 영역을 생성하는 경우 Red Hat은 NS(이름 서버) 레코드를 사용하여 표준 DNS 위임을 항상 사용하고 전달 영역을 방지하는 것이 좋습니다. 대부분의 경우 글로벌 전달자를 사용하는 것이 충분하며 전달 영역은 필요하지 않습니다.

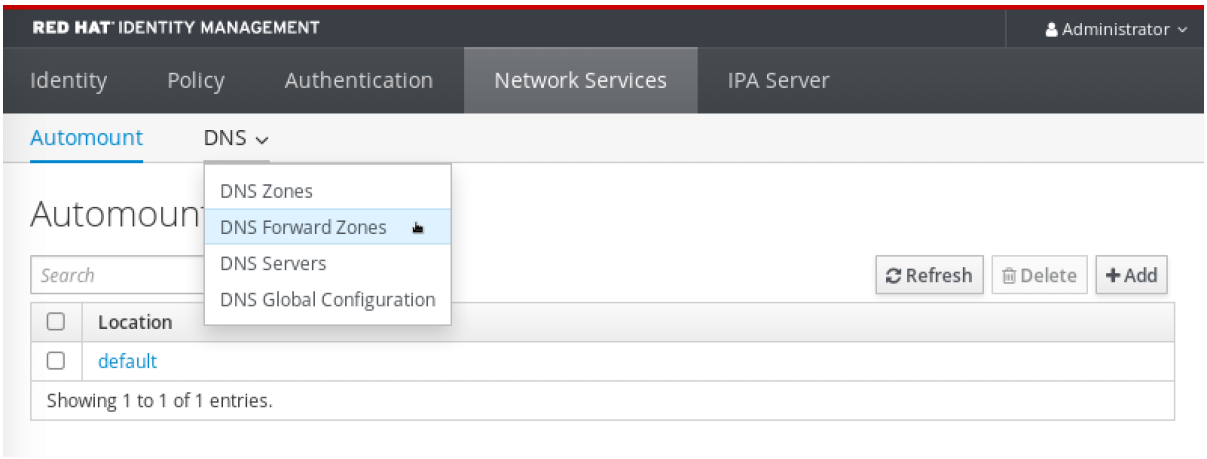
사전 요구 사항

- IdM 관리자로 IdM WebUI에 로그인했습니다.

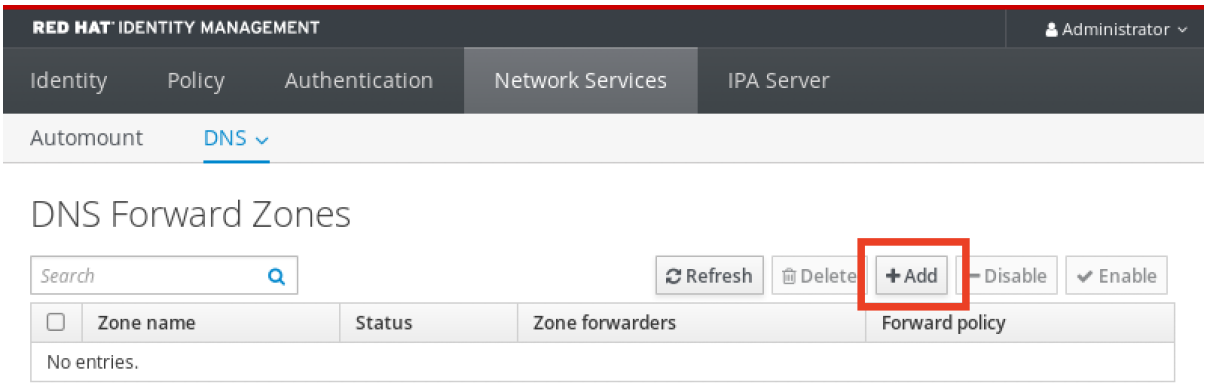
- 쿼리를 전달할 **DNS** 서버의 **IP(Internet Protocol)** 주소를 알고 있습니다.

절차

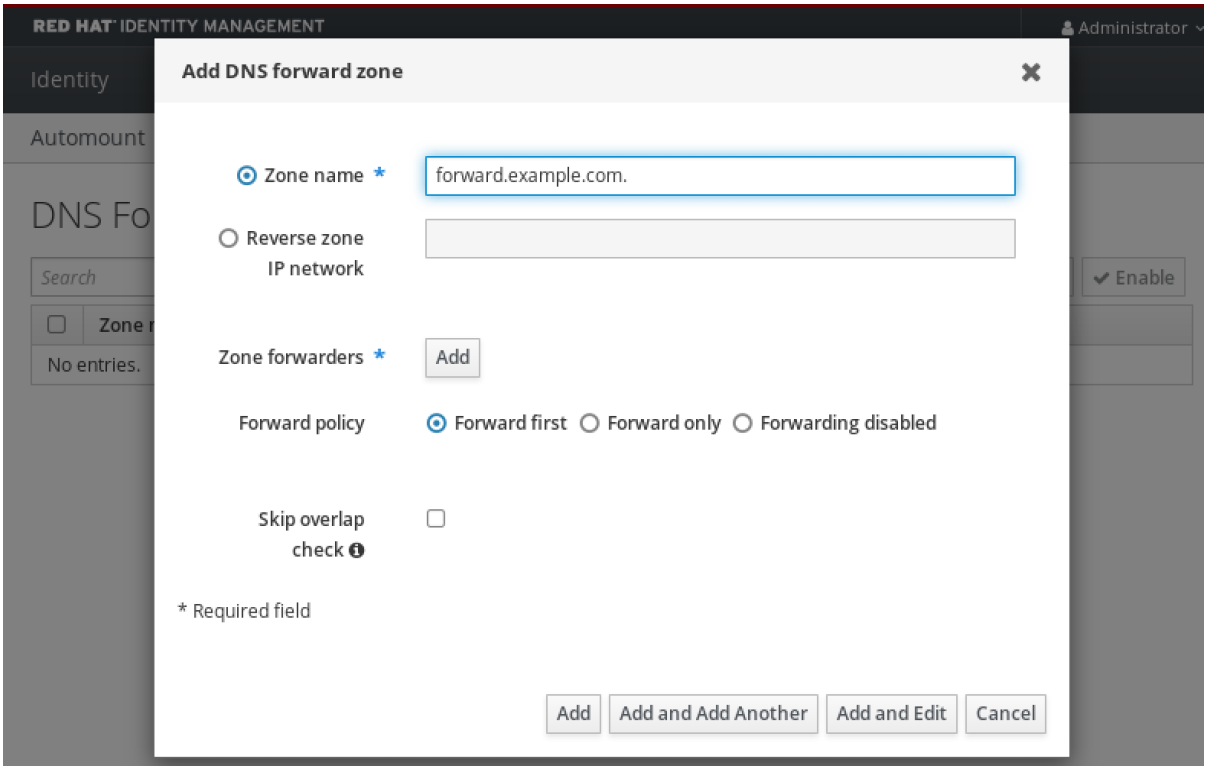
1. **IdM 웹 UI**에서 **네트워크 서비스 → DNS 전달 영역 → DNS** 를 선택합니다.



2. **DNS Forward Zones (DNS 전달 영역)** 섹션에서 **Add(추가)**를 클릭합니다.

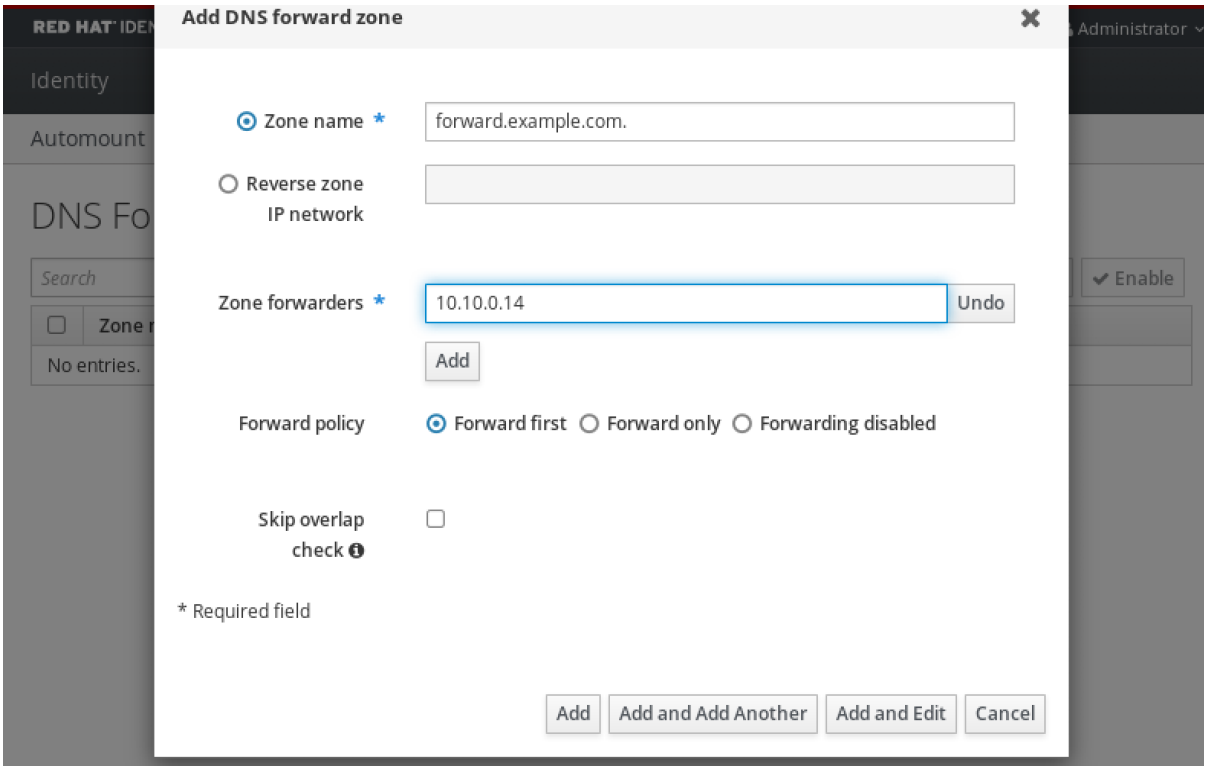


3. **Add DNS forward zone (DNS 전달 영역 추가)** 창에서 전달 영역 이름을 지정합니다.



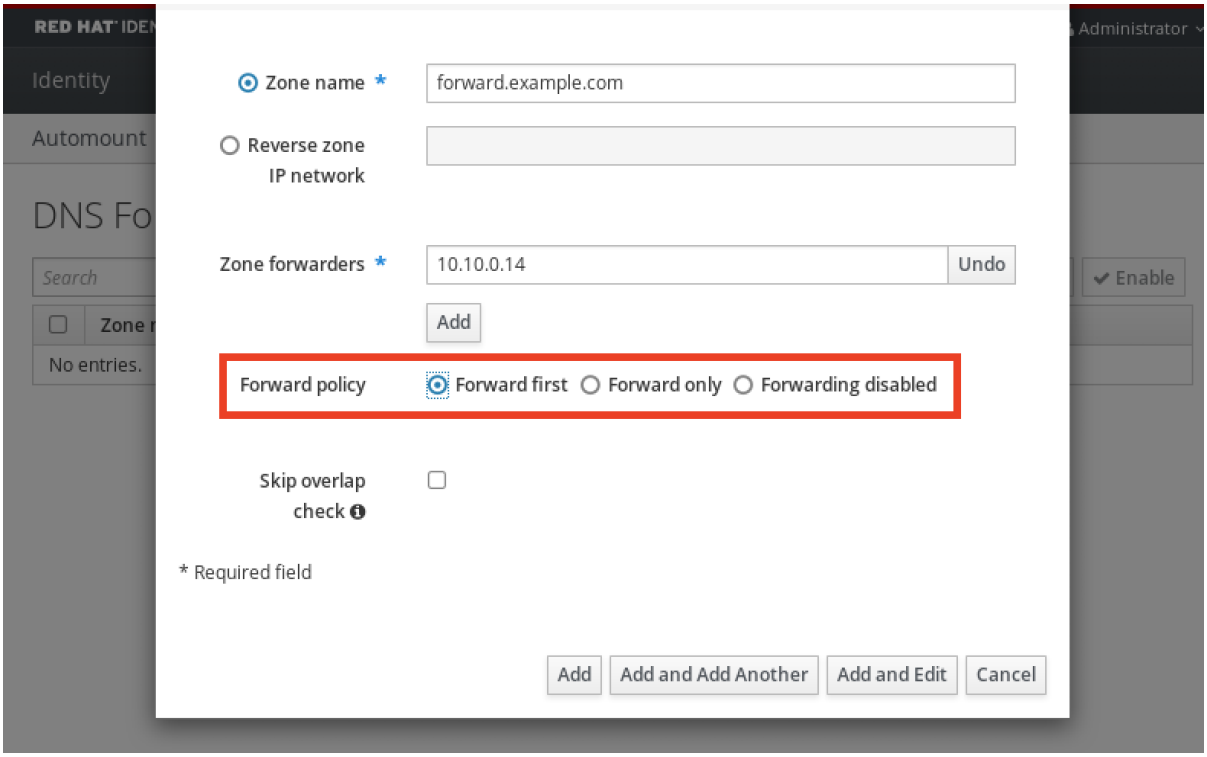
4.

Add(추가) 버튼을 클릭하고 DNS 서버의 IP 주소를 지정하여 전달 요청을 받습니다. 전달 영역당 여러 개의 전달자를 지정할 수 있습니다.



5.

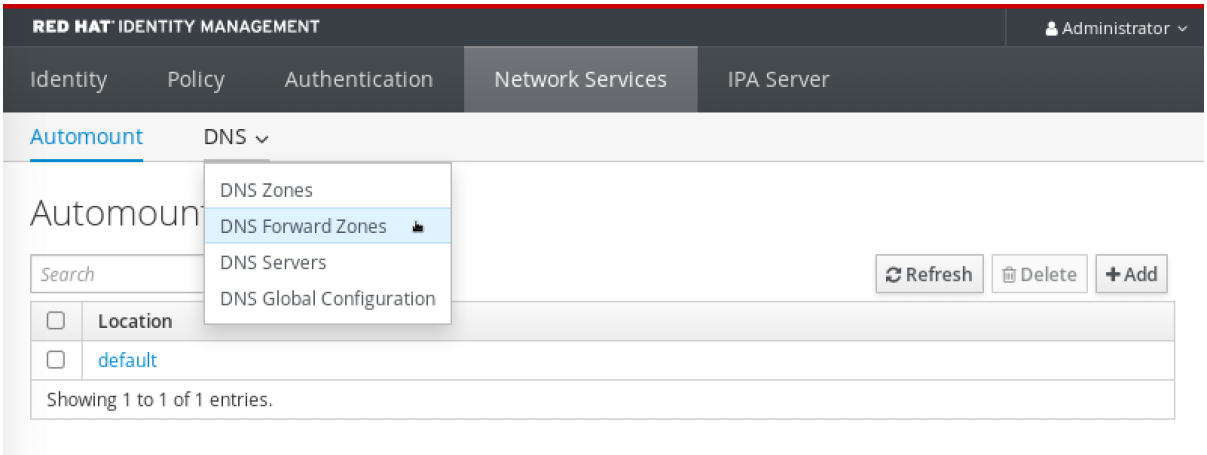
Forward policy(전달 정책을 선택합니다.



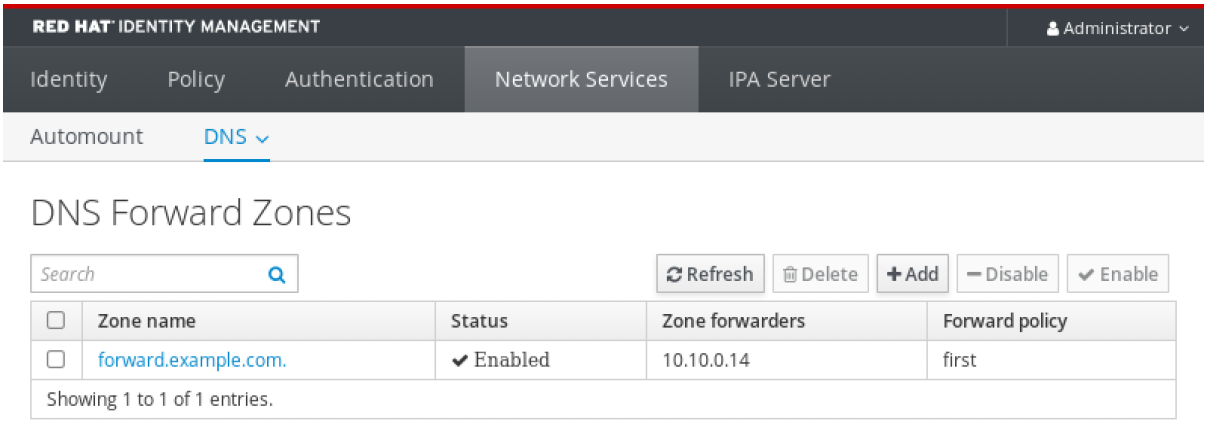
6. 창 하단에서 **Add**(추가)를 클릭하여 새 전달 영역을 추가합니다.

검증 단계

1. **IdM** 웹 **UI**에서 네트워크 서비스 → **DNS** 전달 영역 → **DNS** 를 선택합니다.

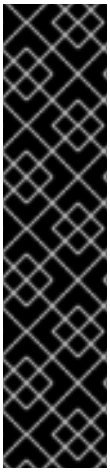


2. 지정한 전달자 및 전달 정책을 사용하여 생성한 전달 영역이 **IdM** 웹 **UI**에 있는지 확인합니다.



27.6. CLI에서 DNS 전달 영역 추가

이 섹션에서는 CLI(명령줄 인터페이스)에서 DNS 전달 영역을 추가하는 방법을 설명합니다.



중요

필요하지 않은 한 전달 영역을 사용하지 마십시오. 전달 영역은 표준 솔루션이 아니며 이를 사용하면 예기치 않고 문제가 있는 동작이 발생할 수 있습니다. 전달 영역을 사용해야 하는 경우 글로벌 전달 구성을 재정의하도록 사용을 제한합니다.

새 DNS 영역을 생성하는 경우 Red Hat은 NS(이름 서버) 레코드를 사용하여 표준 DNS 위임을 항상 사용하고 전달 영역을 방지하는 것이 좋습니다. 대부분의 경우 글로벌 전달자를 사용하는 것이 충분하며 전달 영역은 필요하지 않습니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- 쿼리를 전달할 DNS 서버의 IP(Internet Protocol) 주소를 알고 있습니다.

절차

- dnsforwardzone-add 명령을 사용하여 새 전달 영역을 추가합니다. forward 정책이 없음 인 경우 --forwarder 옵션을 사용하여 하나 이상의 전달자를 지정하고 --forward-policy 옵션을 사용하여 전달 정책을 지정합니다.

```
[user@server ~]$ ipa dnsforwardzone-add forward.example.com. --
forwarder=10.10.0.14 --forwarder=10.10.1.15 --forward-policy=first
```

```
Zone name: forward.example.com.
Zone forwarders: 10.10.0.14, 10.10.1.15
Forward policy: first
```

검증 단계

- dnsforwardzone-show 명령을 사용하여 방금 생성한 DNS 전달 영역을 표시합니다.

```
[user@server ~]$ ipa dnsforwardzone-show forward.example.com.

Zone name: forward.example.com.
Zone forwarders: 10.10.0.14, 10.10.1.15
Forward policy: first
```

27.7. ANSIBLE을 사용하여 IDM에 DNS GLOBAL FORWARDER 설정

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에 DNS 글로벌 전달자를 설정하는 방법에 대해 설명합니다.

아래 예제 절차에서 IdM 관리자는 IP(인터넷 프로토콜) v4 주소 8.8.6.6 및 IPv6 주소 2001:4860:4860:4860::8800 의 DNS 글로벌 전달자를 DNS 서버로 생성합니다 .

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. set-configuration.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp set-configuration.yml establish-global-forwarder.yml
```

4. 편집을 위해 establish-global-forwarder.yml 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

a. 플레이북의 name 변수를 플레이북으로 변경하여 IdM DNS에서 글로벌 전달자를 설정합니다.

b. tasks 섹션에서 작업 이름을 Create a DNS global forwarder(DNS 글로벌 전달자를 8.8.6.6 및 2001:4860:4860::8800)로 변경합니다.

c. ipadnsconfig 부분의 forwarders 섹션에서 다음을 수행합니다.

i. 첫 번째 ip_address 값을 글로벌 전달자의 IPv4 주소로 변경합니다. 8.8.6.6.

ii. 두 번째 ip_address 값을 글로벌 전달자의 IPv6 주소로 변경합니다. 2001:4860:4860::8800.

iii. port 값이 53 으로 설정되어 있는지 확인합니다.

d. forward_policy 를 먼저 로 변경합니다.

이 파일은 현재 예제에 맞게 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to establish a global forwarder in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
```

```
- name: Create a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800
  ipadnsconfig:
    forwarders:
      - ip_address: 8.8.6.6
      - ip_address: 2001:4860:4860::8800
        port: 53
    forward_policy: first
    allow_sync_ptr: yes
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file establish-global-forwarder.yml
```

추가 리소스

- /usr/share/doc/ansible -freeipa/ 디렉터리에서 사용할 수 있는 README-dnsconfig.md Markdown 파일에서 ansible -freeipa ipadnsconfig 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 ipadnsconfig 변수 정의도 포함되어 있습니다.

27.8. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에 DNS 글로벌 전달자가 있는지 확인하는 방법에 대해 설명합니다. 아래 예제 절차에서 IdM 관리자는 IP(인터넷 프로토콜) v4 주소가 7.7이고 IP v6 주소 2001:db8::1:0 에 DNS 글로벌 전달자가 DNS 서버에 있는지 확인합니다. 포트 53 에.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. forwarders-absent.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. 편집을 위해 ensure-presence-of-a-global-forwarder.yml 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

   a. 플레이북의 name 변수를 플레이북으로 변경하여 IdM DNS에 글로벌 전달자가 있는지 확인합니다.

   b. tasks 섹션에서 작업 이름을 변경하여 DNS 글로벌 전달자가 포트 53에 7.7. 및 2001:db8::1:0으로 있는지 확인합니다.

   c. ipadnsconfig 부분의 forwarders 섹션에서 다음을 수행합니다.

      i. 첫 번째 ip_address 값을 글로벌 전달자의 IPv4 주소로 변경합니다. 7.7.9.9.

      ii. 두 번째 ip_address 값을 글로벌 전달자의 IPv6 주소로 변경합니다. 2001:db8::1:0.

      iii. port 값이 53 으로 설정되어 있는지 확인합니다.

   d. 상태를 present 로 변경합니다.

   이 파일은 현재 예제에 맞게 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to ensure the presence of a global forwarder in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0
on port 53
    ipadnsconfig:
      forwarders:
        - ip_address: 7.7.9.9
        - ip_address: 2001:db8::1:0
          port: 53
      state: present
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-presence-of-a-global-forwarder.yml
```

추가 리소스

- /usr/share/doc/ansible -freeipa/ 디렉터리에서 사용할 수 있는 README-dnsconfig.md Markdown 파일에서 ansible -freeipa ipadnsconfig 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 ipadnsconfig 변수 정의도 포함되어 있습니다.

27.9. ANSIBLE을 사용하여 IDM에 DNS 글로벌 전달자가 없는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에 DNS 글로벌 전달자가 없는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 IP(인터넷 프로토콜) v4 주소 8.8.6.6 및 IP v6 주소 2001:4860:4860::8800 의 DNS 글로벌 전달자가 없는지 확인합니다 .

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. forwarders-absent.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. 편집을 위해 ensure-absence-of-a-global-forwarder.yml 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

   a. 플레이북의 name 변수를 플레이북으로 변경하여 IdM DNS에서 글로벌 전달자가 없는지 확인합니다.

   b. tasks 섹션에서 DNS 글로벌 전달자가 없는지 확인하고 포트 53에 2001:4860:4860::8800 으로 작업 이름을 변경합니다.

   c. ipadnsconfig 부분의 forwarders 섹션에서 다음을 수행합니다.

      i. 첫 번째 ip_address 값을 글로벌 전달자의 IPv4 주소로 변경합니다. 8.8.6.6.

      ii. 두 번째 ip_address 값을 글로벌 전달자의 IPv6 주소로 변경합니다. 2001:4860:4860::8800.

      iii. port 값이 53 으로 설정되어 있는지 확인합니다.

d. 상태가 **absent** 로 설정되어 있는지 확인합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to ensure the absence of a global forwarder in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure the absence of a DNS global forwarder to 8.8.6.6 and
2001:4860:4860::8800 on port 53
    ipadnsconfig:
      forwarders:
        - ip_address: 8.8.6.6
        - ip_address: 2001:4860:4860::8800
          port: 53
      state: absent
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-absence-of-a-global-forwarder.yml
```

추가 리소스

- **/usr/share/doc/ansible -freeipa/** 디렉터리에서 사용할 수 있는 **README-dnsconfig.md Markdown** 파일에서 **ansible -freeipa ipadnsconfig** 모듈에 대한 샘플 **Ansible** 플레이북을 확인할 수 있습니다. 파일에는 **ipadnsconfig** 변수 정의도 포함되어 있습니다.

27.10. ANSIBLE을 사용하여 DNS GLOBAL FORWARDERS가 IDM에서 비활성화되었는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 **IdM**에서 **DNS** 글로벌 전달자를 비활성화하는 방법에 대해 설명합니다. 아래 예제 절차에서 **IdM** 관리자는 글로벌 전달자의 전달 정책이 **none** 으로 설정되어 글로벌 전달자를 효과적으로 비활성화합니다.

사전 요구 사항

- **Ansible** 컨트롤러에 **ansible-freeipa** 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. 모든 DNS 글로벌 전달자를 비활성화하도록 이미 구성된 disable-global-forwarders.yml Ansible 플레이북 파일의 내용을 확인합니다. 예를 들면 다음과 같습니다.

```
$ cat disable-global-forwarders.yml
---
- name: Playbook to disable global DNS forwarders
  hosts: ipaserver
  become: true

  tasks:
  - name: Disable global forwarders.
    ipadnsconfig:
      forward_policy: none
```

4. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file disable-global-forwarders.yml
```

추가 리소스

- /usr/share/doc/ansible -freeipa/ 디렉터리에서 사용할 수 있는 README-dnsconfig.md Markdown 파일에서 ansible -freeipa ipadnsconfig 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 ipadnsconfig 변수 정의도 포함되어 있습니다.

27.11. ANSIBLE을 사용하여 IDM에 DNS 전달 영역이 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에 DNS 전달 영역이 있는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 IP(인터넷 프로토콜) 주소가 8.8.8.8 인 DNS 서버에 example.com 의 DNS 전달 영역이 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. forwarders-absent.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-forwardzone.yml
```

4. 편집을 위해 ensure-presence-forwardzone.yml 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

   a. 플레이북의 name 변수를 플레이북으로 변경하여 IdM DNS에 dnsforwardzone이 있는지 확인합니다.

   b. tasks 섹션에서 example.com에 대한 dnsforwardzone이 있는지 확인하도록 작업 이름을 8.8.8.8로 변경합니다.

c. **tasks 섹션에서 ipadnsconfig 제목을 ipadns forwardzone 으로 변경합니다.**

d. **ipadnsforwardzone 섹션에서 다음을 수행합니다.**

i. **ipaadmin_password 변수를 추가하고 IdM 관리자 암호로 설정합니다.**

ii. **name 변수를 추가하고 example.com 으로 설정합니다.**

iii. **forwarders 섹션에서 다음을 수행합니다.**

A. **ip_address 및 포트 행을 제거합니다.**

B. **대시 다음에 지정하여 전달된 요청을 받기 위해 DNS 서버의 IP 주소를 추가합니다.**

```
- 8.8.8.8
```

iv. **forwardpolicy 변수를 추가하고 first 로 설정합니다.**

v. **skip_overlap_check 변수를 추가하고 true 로 설정합니다.**

vi. **state 변수를 present 로 변경합니다.**

이 파일은 현재 예제에 맞게 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Playbook to ensure the presence of a dnsforwardzone in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure the presence of a dnsforwardzone for example.com to 8.8.8.8
```

```
ipadnsforwardzone:
  ipaadmin_password: password01
  name: example.com
  forwarders:
    - 8.8.8.8
  forwardpolicy: first
  skip_overlap_check: true
  state: present
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-presence-forwardzone.yml
```

추가 리소스

- /usr/share/doc/ansible -freeipa/ 디렉터리에서 사용할 수 있는 README-dnsforwardzone.md Markdown 파일에서 ansible -freeipa ipadnsforwardzone 모듈에 대한 더 많은 샘플 Ansible 플레이북을 볼 수 있습니다. 파일에는 ipadnsforwardzone 변수의 정의도 포함되어 있습니다.

27.12. ANSIBLE을 사용하여 DNS FORWARD ZONE에 IDM에 여러 개의 전달자가 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM의 DNS Forward Zone에 여러 전달자가 있는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 example.com의 DNS 전달 영역이 8.8.8.8 및 4.4.4. 4 로 전달되고 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. forwarders-absent.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-presence-multiple-forwarders.yml
```

4. 편집을 위해 ensure-presence-multiple-forwarders.yml 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

a. 플레이북의 name 변수를 플레이북으로 변경하여 IdM DNS의 dnsforwardzone에 여러 전달자가 있는지 확인합니다.

b. tasks 섹션에서 example.com의 dnsforwardzone에서 8.8.8.8 및 4.4.4.4 전달자가 있는지 확인하도록 작업 이름을 변경합니다.

c. tasks 섹션에서 ipadnsconfig 제목을 ipadns forwardzone 으로 변경합니다.

d. ipadnsforwardzone 섹션에서 다음을 수행합니다.

i. ipaadmin_password 변수를 추가하고 IdM 관리자 암호로 설정합니다.

ii. name 변수를 추가하고 example.com 으로 설정합니다.

iii. forwarders 섹션에서 다음을 수행합니다.

A. ip_address 및 포트 행을 제거합니다.

B.

확인하려는 **DNS** 서버의 **IP** 주소를 추가합니다. 앞에 대시가 있습니다.

```
- 8.8.8.8
- 4.4.4.4
```

iv.

state 변수를 **present**로 변경합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: name: Playbook to ensure the presence of multiple forwarders in a
dnsforwardzone in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure presence of 8.8.8.8 and 4.4.4.4 forwarders in dnsforwardzone for
example.com
  ipadnsforwardzone:
      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/ 디렉터리에서 사용할 수 있는 **README-dnsforwardzone.md Markdown** 파일에서 **ansible -freeipa ipadnsforwardzone** 모듈에 대한 더 많은 샘플 **Ansible** 플레이북을 볼 수 있습니다. 파일에는 **ipadnsforwardzone** 변수의 정의도 포함되어 있습니다.

27.13. ANSIBLE을 사용하여 IDM에서 DNS 전달 영역이 비활성화되어 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에서 DNS Forward Zone을 비활성화하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 example.com 의 DNS 전달 영역이 비활성화되었는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. forwarders-absent.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-disabled-forwardzone.yml
```

4. 편집을 위해 ensure-disabled-forwardzone.yml 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

   a. 플레이북의 name 변수를 플레이북으로 변경하여 IdM DNS에서 dnsforwardzone이 비활성화되었는지 확인합니다.

   b. tasks 섹션에서 example.com의 dnsforwardzone이 비활성화되어 있는지 확인하도록 작업 이름을 변경합니다.

c. **tasks** 섹션에서 **ipadnsconfig** 제목을 **ipadns forwardzone** 으로 변경합니다.

d. **ipadnsforwardzone** 섹션에서 다음을 수행합니다.

i. **ipaadmin_password** 변수를 추가하고 **IdM** 관리자 암호로 설정합니다.

ii. **name** 변수를 추가하고 **example.com** 으로 설정합니다.

iii. 전체 **forwarders** 섹션을 제거합니다.

iv. **state** 변수를 **disabled** 로 변경합니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to ensure a dnsforwardzone is disabled in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure a dnsforwardzone for example.com is disabled
  ipadnsforwardzone:
     ipaadmin_password: password01
     name: example.com
     state: disabled
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-disabled-forwardzone.yml
```

추가 리소스

- **/usr/share/doc/ansible -freeipa/** 디렉터리에서 사용할 수 있는 **README-**

dnsforwardzone.md Markdown 파일에서 ansible -freeipa ipadnsforwardzone 모듈에 대한 더 많은 샘플 Ansible 플레이북을 볼 수 있습니다. 파일에는 ipadnsforwardzone 변수의 정의도 포함되어 있습니다.

27.14. ANSIBLE을 사용하여 IDM에 DNS 전달 영역이 없는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 IdM에 DNS 전달 영역이 없는지 확인하는 방법을 설명합니다. 아래 예제 절차에서 IdM 관리자는 example.com 에 대한 DNS 전달 영역이 없는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsconfig 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. forwarders-absent.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp forwarders-absent.yml ensure-absence-forwardzone.yml
```

4. 편집을 위해 ensure-absence-forwardzone.yml 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

a. 플레이북의 **name** 변수를 플레이북으로 변경하여 **IdM DNS**에 **dnsforwardzone**이 없는지 확인합니다.

b. **tasks** 섹션에서 **example.com**에 대한 **dnsforwardzone**이 없으면 작업 이름을 변경합니다.

c. **tasks** 섹션에서 **ipadnsconfig** 제목을 **ipadns forwardzone** 으로 변경합니다.

d. **ipadnsforwardzone** 섹션에서 다음을 수행합니다.

i. **ipaadmin_password** 변수를 추가하고 **IdM** 관리자 암호로 설정합니다.

ii. **name** 변수를 추가하고 **example.com** 으로 설정합니다.

iii. 전체 **forwarders** 섹션을 제거합니다.

iv. **state** 변수를 **absent** 로 둡니다.

이 파일은 현재 예제에 맞게 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Playbook to ensure the absence of a dnsforwardzone in IdM DNS
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure the absence of a dnsforwardzone for example.com
    ipadnsforwardzone:
      ipaadmin_password: password01
      name: example.com
      state: absent
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-absence-forwardzone.yml
```

추가 리소스

- /usr/share/doc/ansible -freeipa/ 디렉터리에서 사용할 수 있는 README-dnsforwardzone.md Markdown 파일에서 ansible -freeipa ipadnsforwardzone 모듈에 대한 더 많은 샘플 Ansible 플레이북을 볼 수 있습니다. 파일에는 ipadnsforwardzone 변수의 정의도 포함되어 있습니다.

28장. ANSIBLE을 사용하여 IDM에서 DNS 레코드 관리

이 장에서는 Ansible 플레이북을 사용하여 IdM(Identity Management)에서 DNS 레코드를 관리하는 방법을 설명합니다. IdM 관리자는 IdM에서 DNS 레코드를 추가, 수정, 삭제할 수 있습니다. 장에는 다음 섹션이 포함되어 있습니다.

- Ansible을 사용하여 IdM에 A 및 AAAA DNS 레코드가 있는지 확인
- Ansible을 사용하여 IdM에 A 및 PTR DNS 레코드가 있는지 확인
- Ansible을 사용하여 IdM에 여러 DNS 레코드가 있는지 확인
- Ansible을 사용하여 IdM에 여러 CNAME 레코드가 있는지 확인
- Ansible을 사용하여 IdM에 SRV 레코드가 있는지 확인

28.1. IDM의 DNS 레코드

IdM(Identity Management)은 다양한 DNS 레코드 유형을 지원합니다. 다음 4가지가 가장 자주 사용됩니다.

A

호스트 이름과 IPv4 주소에 대한 기본 맵입니다. A 레코드의 레코드 이름은 www 와 같은 호스트 이름입니다. A 레코드의 IP 주소 값은 192.0.2.1 과 같은 IPv4 주소입니다.

A 레코드에 대한 자세한 내용은 RFC 1035 를 참조하십시오.

AAAA

호스트 이름과 IPv6 주소에 대한 기본 맵입니다. AAAA 레코드의 레코드 이름은 www 와 같은 호스트 이름입니다. IP 주소 값은 2001:DB8::1111 과 같은 IPv6 주소입니다.

AAAA 레코드에 대한 자세한 내용은 RFC 3596 을 참조하십시오.

SRV

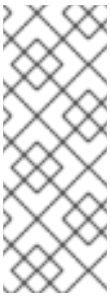
서비스(SRV) 리소스 레코드 는 서비스 이름을 특정 서비스를 제공하고 있는 서버의 **DNS** 이름에 매핑합니다. 예를 들어 이 레코드 유형은 **LDAP** 디렉터리와 같은 서비스를 관리하는 서버에 매핑할 수 있습니다.

SRV 레코드의 레코드 이름은 *_service* 형식을 갖습니다. *_ldap._tcp와 같은 _ protocol*. **SRV** 레코드의 구성 옵션에는 대상 서비스의 우선 순위, 가중치, 포트 번호 및 호스트 이름이 포함됩니다.

SRV 레코드에 대한 자세한 내용은 RFC 2782 을 참조하십시오.

PTR

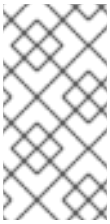
PTR(포인터 레코드)은 **IP** 주소를 도메인 이름에 매핑하는 역방향 **DNS** 레코드를 추가합니다.



참고

IPv4 주소에 대한 모든 역방향 **DNS** 조회는 **in-addr.arpa.** 도메인에 정의된 역방향 항목을 사용합니다. 역방향 주소는 사람이 읽을 수 있는 형식으로, 일반 **IP** 주소의 정확한 역순이며 **in-addr.arpa.** 도메인이 추가됩니다. 예를 들어 네트워크 주소 **192.0.2.0/24** 의 경우 역방향 영역은 **2.0.192.in-addr.arpa** 입니다.

PTR의 레코드 이름은 RFC 1035에 지정된 표준 형식이어야 하며 RFC 2317 및 RFC 3596 에서 확장되어야 합니다. 호스트 이름 값은 레코드를 만들 호스트의 정식 호스트 이름이어야 합니다.



참고

또한. **ip6.arpa.** 도메인의 영역을 사용하여 **IPv6** 주소에 대해 역방향 영역을 구성할 수 있습니다. **IPv6** 역방향 영역에 대한 자세한 내용은 RFC 3596 을 참조하십시오.

DNS 리소스 레코드를 추가할 때 많은 레코드에 다른 데이터가 필요합니다. 예를 들어 **CNAME** 레코드에는 호스트 이름이 필요하지만 **A** 레코드에는 **IP** 주소가 필요합니다. **IdM** 웹 **UI**에서 현재 선택한 레코드 유형에 필요한 데이터를 반영하도록 새 레코드를 추가하기 위한 양식의 필드가 자동으로 업데이트됩니다.

28.2. 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**

Bash 에서는 중괄호 안의 쉼표로 구분된 목록에 값을 나열하여 여러 항목을 정의할 수 있습니다(예: --option={val1,val2,val3}).

표 28.1. 일반 레코드 옵션

옵션	*설명*
--ttl=*number*	레코드에 사용할 시간을 설정합니다.
--structured	원시 DNS 레코드를 구문 분석하고 구조화된 형식으로 반환합니다.

표 28.2. "a" 레코드 옵션

옵션	*설명*	*예*
--a-rec=*ARECORD*	하나의 A 레코드 또는 하나의 A 레코드 목록을 전달합니다.	**ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123**
	지정된 IP 주소가 있는 와일드카드 A 레코드를 만들 수 있습니다.	**ipa dnsrecord-add idm.example.com "*" --a-rec=192.168.122.123** [a]
--a-ip-address=*string*	레코드의 IP 주소를 제공합니다. 레코드를 만들 때 **A** 레코드 값을 지정하는 옵션은 **--a-rec** 입니다. 그러나 **A** 레코드를 수정할 때 **--a-rec** 옵션은 **A** 레코드의 현재 값을 지정하는 데 사용됩니다. 새 값은 **--a-ip-address** 옵션으로 설정됩니다.	**ipa dnsrecord-mod idm.example.com --a-rec 192.168.122.123 --a-ip-address 192.168.122.124**

[a] 이 예제에서는 IP 주소가 192.0.2.123인 와일드카드 **A** 레코드를 생성합니다.

표 28.3. "AAAA" 레코드 옵션

옵션	설명	예제
--aaaa-rec=*AAAARECORD*	단일 AAAA(IPv6) 레코드 또는 AAAA 레코드 목록을 전달합니다.	**ipa dnsrecord-add idm.example.com www --aaaa-rec 2001:db8::1231:5675**
--aaaa-ip-address=*string*	레코드에 IPv6 주소를 제공합니다. 레코드를 만들 때 **A** 레코드 값을 지정하는 옵션은 **--aaa-rec** 입니다. 그러나 **A** 레코드를 수정할 때 **--aaaa-rec** 옵션은 **A** 레코드의 현재 값을 지정하는 데 사용됩니다. 새 값은 **--a-ip-address** 옵션으로 설정됩니다.	**IPA dnsrecord-mod idm.example.com --aaaa-rec 2001:db8::1231:5675 --aaa-ip-address 2001:db8::1231:5676**

표 28.4. "PTR" 레코드 옵션

옵션	설명	예제
--ptr-rec=*PTRRECORD*	단일 PTR 레코드 또는 PTR 레코드 목록을 전달합니다. 역방향 DNS 레코드를 추가할 때 다른 DNS 레코드를 추가하는 사용과 비교하여 **ipa dnsrecord-add** 명령과 함께 사용되는 영역 이름이 되돌립니다. 일반적으로 호스트 IP 주소는 지정된 네트워크에 있는 IP 주소의 마지막 옥텟입니다. 오른쪽의 첫 번째 예에서는 **server4.idm.example.com에 IPv4** 주소 **192.168.122.4** 인 **PTR 레코드를 추가합니다.** 두 번째 예제는 **0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa에 역방향 DNS 항목을 추가합니다.** IP 주소가 **2001:DB8::1111** 인 호스트 **server2.example.com** 의 IPv6 역방향 영역입니다.	**ipa dnsrecord-add 122.168.192.in-addr.arpa 4 --ptr-rec server4.idm.example.com.** **$ ipa dnsrecord-add 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. 1.1.1.0.0.0.0.0.0.0.0.0.0.0.0.0 --ptr-rec server2.idm.example.com.**
--ptr-hostname=*string*	레코드의 호스트 이름을 제공합니다.	

표 28.5. "SRV" 레코드 옵션

옵션	설명	예제
--srv-rec=*SRVRECORD*	단일 SRV 레코드 또는 SRV 레코드 목록을 전달합니다. 오른쪽 예제에서 _ldap._tcp 는 SRV 레코드의 서비스 유형 및 연결 프로토콜을 정의합니다. srv **-rec** 옵션은 우선순위, 가중치, 포트, 대상 값을 정의합니다. 예에서 57 및 49의 가중치 값은 100까지 추가하고 특정 레코드가 사용되는 백분율로 가능성을 나타냅니다.	**# ipa dnsrecord-add idm.example.com _ldap._tcp --srv-rec="0 51 389 server1.idm.example.com."** **# IPA dnsrecord-add server.idm.example.com _ldap._tcp --srv-rec 49 389 server2.idm.example.com".**

옵션	*설명*	*예제*
--srv-priority=*number*	레코드의 우선 순위를 설정합니다. 서비스 유형에 대한 SRV 레코드가 여러 개 있을 수 있습니다. 우선 순위 (0 - 65535)는 레코드의 순위를 설정합니다. 숫자가 작을수록 우선 순위가 높습니다. 서비스는 우선 순위가 가장 높은 레코드를 먼저 사용해야 합니다.	**# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec 49 389 server2.idm.example.com." --srv-priority=0**
--srv-weight=*number*	레코드의 가중치를 설정합니다. 이렇게 하면 동일한 우선 순위의 SRV 레코드 순서를 결정하는 데 도움이 됩니다. 설정 가중치는 특정 레코드가 사용되는 가능성(백분율)을 나타내는 최대 100개까지 추가해야 합니다.	**# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec 49 389 server2.idm.example.com." --srv-weight=60**
--srv-port=*number*	대상 호스트에서 서비스에 대한 포트를 제공합니다.	**# IPA dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec 60 389 server2.idm.example.com." --srv-port=636**
--srv-target=*string*	대상 호스트의 도메인 이름을 제공합니다. 도메인에서 서비스를 사용할 수 없는 경우 단일 마침표(.)일 수 있습니다.	

추가 리소스

- **ipa dnsrecord-add** 및 IdM에서 지원하는 DNS 레코드 유형을 사용하는 방법에 대한 자세한 내용을 보려면 **ipa dnsrecord-add --help** 명령을 실행합니다.

28.3. ANSIBLE을 사용하여 IDM에 A 및 AAAA DNS 레코드가 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 특정 IdM 호스트에 대한 A 및 AAAA 레코드가 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 IdM 관리자는 idm.example.com DNS 영역에 host1 에 대한 A 및 AAAA 레코드가 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.

- idm.example.com 영역이 존재하며 IdM DNS에 의해 관리됩니다. IdM DNS에서 기본 DNS 영역 추가에 대한 자세한 내용은 Ansible 플레이북을 사용하여 IdM DNS 영역 관리에 대한 내용을 참조하십시오.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. ensure-A-and-AAAA-records-are-present.yml Ansible 플레이북 파일의 복사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-A-and-AAAA-records-are-present.yml ensure-A-and-AAAA-records-are-present-copy.yml
```

4. 편집을 위해 ensure-A-and-AAAA-records-are-present-copy.yml 파일을 엽니다.

5. ipadnsrecord 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.
- zone_name 변수를 idm.example.com 으로 설정합니다.
- records 변수에서 name 변수를 host1 로 설정하고 a_ip_address 변수를 192.168.122.123 으로 설정합니다.
- records 변수에서 name 변수를 host1 로 설정하고 aaaa_ip_address 변수를 ::1 로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Ensure A and AAAA records are present
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure A and AAAA records are present
  - name: Ensure that 'host1' has A and AAAA records.
    ipadnsrecord:
      ipaadmin_password: Secret123
      zone_name: idm.example.com
      records:
      - name: host1
        a_ip_address: 192.168.122.123
      - name: host1
        aaaa_ip_address: ::1
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-A-and-AAAA-records-are-present-copy.yml
```

추가 리소스

- **A** 및 **AAAA** 레코드에 대한 자세한 내용은 **IdM**의 **DNS** 레코드를 참조하십시오.
- **/usr/share/doc/ansible -freeipa/** 디렉터리에 있는 **README-dnsrecord.md Markdown** 파일에서 **ansible -freeipa ipadnsrecord** 모듈에 대한 샘플 **Ansible** 플레이북을 확인할 수 있습니다. 파일에는 **ipadnsrecord** 변수의 정의도 포함되어 있습니다.
- **/usr/share/doc/ansible-freeipa/playbooks/dns record** 디렉터리에서 **ipadns record** 모듈에 대한 샘플 **Ansible** 플레이북을 볼 수 있습니다.

28.4. ANSIBLE을 사용하여 IDM에 A 및 PTR DNS 레코드가 있는지 확인

이 섹션에서는 **IdM(Identity Management)** 관리자가 **Ansible** 플레이북을 사용하여 특정 **IdM** 호스트의 **A** 레코드가 해당 **PTR** 레코드가 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 **IdM**

관리자는 IP 주소가 192.168.122.45 인 host1 에 대해 idm.example.com 영역에 A 및 PTR 레코드가 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- idm.example.com DNS 영역이 존재하며 IdM DNS에서 관리합니다. IdM DNS에서 기본 DNS 영역 추가에 대한 자세한 내용은 Ansible 플레이북을 사용하여 IdM DNS 영역 관리에 대한 내용을 참조하십시오.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. ensure-dnsrecord-with-reverse-is-present.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-dnsrecord-with-reverse-is-present.yml ensure-dnsrecord-with-reverse-is-present-copy.yml
```

4. 편집을 위해 ensure-dnsrecord-with-reverse-is-present-copy.yml 파일을 엽니다.

5. ipadnsrecord 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.

- name 변수를 host 1 로 설정합니다.
- zone_name 변수를 idm.example.com 으로 설정합니다.
- ip_address 변수를 192.168.122.45 로 설정합니다.
- create_reverse 변수를 yes 로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Ensure DNS Record is present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure that dns record is present
  - ipadnsrecord:
      ipaadmin_password: Secret123
      name: host1
      zone_name: idm.example.com
      ip_address: 192.168.122.45
      create_reverse: yes
      state: present
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-dnsrecord-with-reverse-is-present-copy.yml
```

추가 리소스

- A 및 PTR DNS 레코드에 대한 자세한 내용은 IdM의 DNS 레코드를 참조하십시오.
-

/usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README-dnsrecord.md Markdown 파일에서 ansible -freeipa ipadnsrecord 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 ipadnsrecord 변수의 정의도 포함되어 있습니다.

- /usr/share/doc/ansible-freeipa/playbooks/dns record 디렉터리에서 ipadns record 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.

28.5. ANSIBLE을 사용하여 IDM에 여러 DNS 레코드가 있는지 확인

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 여러 값이 특정 IdM DNS 레코드와 연결되어 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 IdM 관리자는 idm.example.com DNS 영역에 host1 에 대해 여러 A 레코드가 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- idm.example.com 영역이 존재하며 IdM DNS에 의해 관리됩니다. IdM DNS에서 기본 DNS 영역 추가에 대한 자세한 내용은 Ansible 플레이북을 사용하여 IdM DNS 영역 관리에 대한 내용을 참조하십시오.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. ensure-presence-multiple-records.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-presence-multiple-records.yml ensure-presence-multiple-records-copy.yml
```

4. 편집을 위해 **ensure-presence-multiple-records-copy.yml** 파일을 엽니다.

5. **ipadnsrecord** 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **records** 섹션에서 **name** 변수를 **host1** 로 설정합니다.
- **records** 섹션에서 **zone_name** 변수를 **idm.example.com** 으로 설정합니다.
- **records** 섹션에서 **a_rec** 변수를 **192.168.122.112**로 설정하고 **192.168.122. 122** 로 설정합니다.
- **records** 섹션에 두 번째 레코드 를 정의합니다.
  - **name** 변수를 **host 1** 로 설정합니다.
  - **zone_name** 변수를 **idm.example.com** 으로 설정합니다.
  - **aaaa_rec** 변수를 **::1** 로 설정합니다.

이는 현재 예제에 대한 수정된 **Ansible** 플레이북 파일입니다.

```
---
- name: Test multiple DNS Records are present.
  hosts: ipaserver
  become: true
  gather_facts: false
```

```
tasks:
# Ensure that multiple dns records are present
- ipadnsrecord:
    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

- DNS의 A 레코드에 대한 자세한 내용은 IdM의 DNS 레코드를 참조하십시오.
- /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README-dnsrecord.md Markdown 파일에서 ansible -freeipa ipadnsrecord 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 ipadnsrecord 변수의 정의도 포함되어 있습니다.
- /usr/share/doc/ansible-freeipa/playbooks/dns record 디렉터리에서 ipadns record 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.

28.6. ANSIBLE을 사용하여 IDM에 여러 CNAME 레코드가 있는지 확인

CNAME 레코드(CNAME 레코드)는 하나의 도메인 이름, 별칭을 다른 이름인 정식 이름에 매핑하는 DNS(Domain Name System)의 리소스 레코드 유형입니다.

CNAME 레코드는 단일 IP 주소에서 여러 서비스를 실행할 때 유용합니다(예: FTP 서비스 및 웹 서비스는 각각 다른 포트에서 실행됨).

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 여러 CNAME 레코드가 IdM DNS에 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 host03 은 HTTP

서버와 FTP 서버 둘 다입니다. IdM 관리자는 idm.example.com 영역에 host03 A 레코드에 대한 www 및 ftp CNAME 레코드가 있는지 확인합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.
- IdM 관리자 암호를 알고 있습니다.
- idm.example.com 영역이 존재하며 IdM DNS에 의해 관리됩니다. IdM DNS에서 기본 DNS 영역 추가에 대한 자세한 내용은 Ansible 플레이북을 사용하여 IdM DNS 영역 관리에 대한 내용을 참조하십시오.
- host03 A 레코드는 idm.example.com 영역에 있습니다.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. ensure-CNAME-record-is-present.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-CNAME-record-is-present.yml ensure-CNAME-record-is-present-copy.yml
```

4. 편집을 위해 ensure-CNAME-record-is-present-copy.yml 파일을 엽니다.

5. ipadnsrecord 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- (선택 사항) 플레이 이름에 제공된 설명을 설명합니다.
- ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.
- zone_name 변수를 idm.example.com 으로 설정합니다.
- records 변수 섹션에서 다음 변수와 값을 설정합니다.
  - name 변수를 www 로 설정합니다.
  - cname_hostname 변수를 host03 으로 설정합니다.
  - name 변수를 ftp 로 설정합니다.
  - cname_hostname 변수를 host03 으로 설정합니다.

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Ensure that 'www.idm.example.com' and 'ftp.idm.example.com' CNAME
records point to 'host03.idm.example.com'.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  - 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/ 디렉터리에 있는 README-dnsrecord.md Markdown 파일에서 ansible -freeipa ipadnsrecord 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 ipadnsrecord 변수의 정의도 포함되어 있습니다.

- /usr/share/doc/ansible-freeipa/playbooks/dns record 디렉터리에서 ipadns record 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.

28.7. ANSIBLE을 사용하여 IDM에 SRV 레코드가 있는지 확인

SRV(DNS 서비스) 레코드는 도메인에서 사용할 수 있는 서비스의 호스트 이름, 포트 번호, 전송 프로토콜, 우선 순위 및 가중치를 정의합니다. IdM(Identity Management)에서는 SRV 레코드를 사용하여 IdM 서버 및 복제본을 찾을 수 있습니다.

이 섹션에서는 IdM(Identity Management) 관리자가 Ansible 플레이북을 사용하여 SRV 레코드가 IdM DNS에 있는지 확인하는 방법을 보여줍니다. 아래 절차에서 사용된 예제에서 IdM 관리자는 값이 10 50 88 idm. example.com인 _kerberos._udp.idm.example.com SRV 레코드가 있는지 확인합니다. 이렇게 하면 다음 값이 설정됩니다.

- 서비스의 우선 순위를 10으로 설정합니다.

- 서비스의 weight를 50으로 설정합니다.

- 서비스에서 사용할 포트를 88으로 설정합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다. 이는 절차의 단계를 실행하는 호스트입니다.

- IdM 관리자 암호를 알고 있습니다.
- idm.example.com 영역이 존재하며 IdM DNS에 의해 관리됩니다. IdM DNS에서 기본 DNS 영역 추가에 대한 자세한 내용은 Ansible 플레이북을 사용하여 IdM DNS 영역 관리에 대한 내용을 참조하십시오.

절차

1. /usr/share/doc/ansible-freeipa/playbooks/dnsrecord 디렉터리로 이동합니다.

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 인벤토리 파일을 열고 구성할 IdM 서버가 [ipaserver] 섹션에 나열되어 있는지 확인합니다. 예를 들어 server.idm.example.com을 구성하도록 Ansible에 지시하려면 다음을 입력합니다.

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

3. ensure-SRV-record-is-present.yml Ansible 플레이북 파일의 사본을 만듭니다. 예를 들면 다음과 같습니다.

```
$ cp ensure-SRV-record-is-present.yml ensure-SRV-record-is-present-copy.yml
```

4. 편집을 위해 ensure-SRV-record-is-present-copy.yml 파일을 엽니다.
5. ipadnsrecord 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.
   - ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.
   - name 변수를 _kerberos._udp.idm.example.com 으로 설정합니다.
   - srv_rec 변수를 '10 50 88 idm.example.com' 으로 설정합니다.

- zone_name 변수를 idm.example.com 으로 설정합니다.

  이 파일은 현재 예제에 맞게 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Test multiple DNS Records are present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure a SRV record is present
  - ipadnsrecord:
      ipaadmin_password: Secret123
      name: _kerberos._udp.idm.example.com
      srv_rec: '10 50 88 idm.example.com'
      zone_name: idm.example.com
      state: present
```

6. 파일을 저장합니다.

7. 플레이북을 실행합니다.

```
$ ansible-playbook -v -i inventory.file ensure-SRV-record-is-present.yml
```

추가 리소스

- SRV 레코드에 대한 자세한 내용은 IdM의 DNS 레코드를 참조하십시오.

- /usr/share/doc/ansible -freeipa/ 디렉터리에 있는 README-dnsrecord.md Markdown 파일에서 ansible -freeipa ipadnsrecord 모듈에 대한 샘플 Ansible 플레이북을 확인할 수 있습니다. 파일에는 ipadnsrecord 변수의 정의도 포함되어 있습니다.

- /usr/share/doc/ansible-freeipa/playbooks/dns record 디렉터리에서 ipadns record 모듈에 대한 샘플 Ansible 플레이북을 볼 수 있습니다.