Red Hat Enterprise Linux 8

IdM 사용자 그룹 호스트 및 액세스 제어 규칙 관리

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

Last Updated: 2022-07-26

Red Hat Enterprise Linux 8 IdM 사용자 그룹 호스트 및 액세스 제어 규칙 관리

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

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

법적 공지

초록

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

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체

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.

- 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.
- Bugzilla를 통해 피드백을 제출하려면 새 티켓을 생성합니다.
  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. IDM 명령줄 유틸리티 소개

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

사전 요구 사항

- 설치 및 액세스할 수 있는 IdM 서버.
  자세한 내용은 Identity Management 설치를 참조하십시오.
- IPA 명령줄 인터페이스를 사용하려면 유효한 Kerberos 티켓으로 IdM을 인증합니다.
  유효한 Kerberos 티켓을 가져오는 방법에 대한 자세한 내용은 명령줄에서 ID 관리 로그인을 참조하십시오.

1.1. IPA 명령줄 인터페이스란 무엇입니까?

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

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

IPA CLI를 사용하면 다음을 수행할 수 있습니다.

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

1.2. IPA 도움말이란 무엇입니까?

IPA 도움말은 IdM 서버를 위한 기본 제공 문서 시스템입니다.

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

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

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

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

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

- **[]** "모든 매개변수는 선택 사항이며 **ipa 도움말** 만 쓸 수 있으며 명령을 실행할 수 있습니다.

- | 파이프 문자는 **또는** 을 의미합니다. 따라서 기본 **ipa help 명령을** 사용하여 topIC , **COMMAND** 또는 **주제** 또는 명령을 지정할 수 있습니다.

  - **주제: ipa 도움말 주제를 실행하여 IPA 도움말 (예: 사용자, 인증서,서버 등)의 항목 목록을 표시할 수 있습니다.**

  - **top /Enical letters 가 있는 I/E는 변수입니다. 따라서 특정 주제(예: ipa help 사용자)를 지정할 수 있습니다.**

  - **ipa help 명령을 입력하여 IPA 도움말(예: user-add,ca-enable,server-show 등)의 명령 목록을 표시할 수 있습니다.**

  - **대 /도: 대문자로 된 Command MAND는 변수입니다. 따라서 ipa help user-add 와 같은 특정 명령을 지정할 수 있습니다.**

1.3. IPA 도움말 주제 사용

다음 절차에서는 명령줄 인터페이스에서 IPA 도움말을 사용하는 방법을 설명합니다.

절차

1. **터미널을 열고 IdM 서버에 연결합니다.**

2. **ipa 도움말 주제를 입력하여 도움말에서 다루는 주제 목록을 표시합니다.**

```
$ ipa help topics
```

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

   이 예제에서는 다음 주제를 사용합니다. user

```
$ ipa help user
```

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

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

그런 다음 아래로 스크롤하여 전체 도움을 읽을 수 있습니다.

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

1.4. IPA 도움말 명령 사용

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

절차

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

```
$ ipa help commands
```

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

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

추가 리소스

- **ipa man** 페이지.

1.5. IPA 명령 구조

IPA CLI는 다음 유형의 명령을 구분합니다.

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

- 플러그인 제공 명령

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

- 사용자,
- 호스트,
- DNS 레코드,
- 인증서,

그리고 더 많은

이러한 오브젝트 대부분에서 IPA CLI에는 다음과 같은 명령이 포함되어 있습니다.

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

명령에는 다음과 같은 구조가 있습니다.

ipa user-add, ipa user-mod, ipa user-del, ipa user-find, ipa user-show

ipa host-add, ipa host-mod, ipa host-del, ipa host-find, ipa host-show

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

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

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

1.6. IPA 명령을 사용하여 IDM에 사용자 계정 추가

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

사전 요구 사항

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

절차

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

```
$ ipa user-add
```

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

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

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

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

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

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

```
---------------------
Added user "euser"
---------------------
User login: euser
First name: Example
Last name: User
Full name: Example User
Display name: Example User
Initials: EU
Home directory: /home/euser
GECOS: Example User
Login shell: /bin/sh
Principal name: euser@IDM.EXAMPLE.COM
Principal alias: euser@IDM.EXAMPLE.COM
Email address: euser@idm.example.com
UID: 427200006
GID: 427200006
Password: False
Member of groups: ipausers
Kerberos keys available: False
```

참고

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

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

IPA CLI에서 사용자 이름과 암호를 추가하거나 확인하라는 메시지가 표시됩니다.

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

추가 리소스

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

1.7. IPA 명령을 사용하여 IDM에서 사용자 계정 수정

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

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

사전 요구 사항

- 사용자 계정을 수정하려면 관리자 권한이 있어야 합니다.

절차

1. 터미널을 열고 **IdM** 서버에 연결합니다.

2. **ipa user-mod** 명령을 입력하고 수정할 사용자를 지정하고, 암호를 추가하기 위해 **--password** 와 같은 옵션을 지정합니다.

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

이 명령은 새 암호를 추가할 수 있는 스크립트를 실행합니다.

3. 새 암호를 입력하고 **Enter** 키를 누릅니다.

IPA CLI는 다음 출력을 출력합니다.

```
----------------------
Modified user "euser"
----------------------
User login: euser
First name: Example
Last name: User
Home directory: /home/euser
Principal name: euser@IDM.EXAMPLE.COM
Principal alias: euser@IDM.EXAMPLE.COM
Email address: euser@idm.example.com
UID: 427200006
GID: 427200006
Password: True
Member of groups: ipausers
Kerberos keys available: True
```

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

추가 리소스

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

1.8. IDM 유틸리티에 값 목록을 제공하는 방법

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

IdM은 다음과 같은 다중 값 목록을 제공하는 방법을 지원합니다.

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

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

- 또는 목록을 중괄호로 묶을 수 있습니다. 이 경우 쉘이 확장을 수행합니다.

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

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

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

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

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

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

또는

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

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

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

또는

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

1.9. IDM 유틸리티와 특수 문자를 사용하는 방법

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

예를 들어 별표 (*) 를 이스케이프하려면 다음을 수행합니다.

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

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

2장. 명령줄을 사용하여 사용자 계정 관리

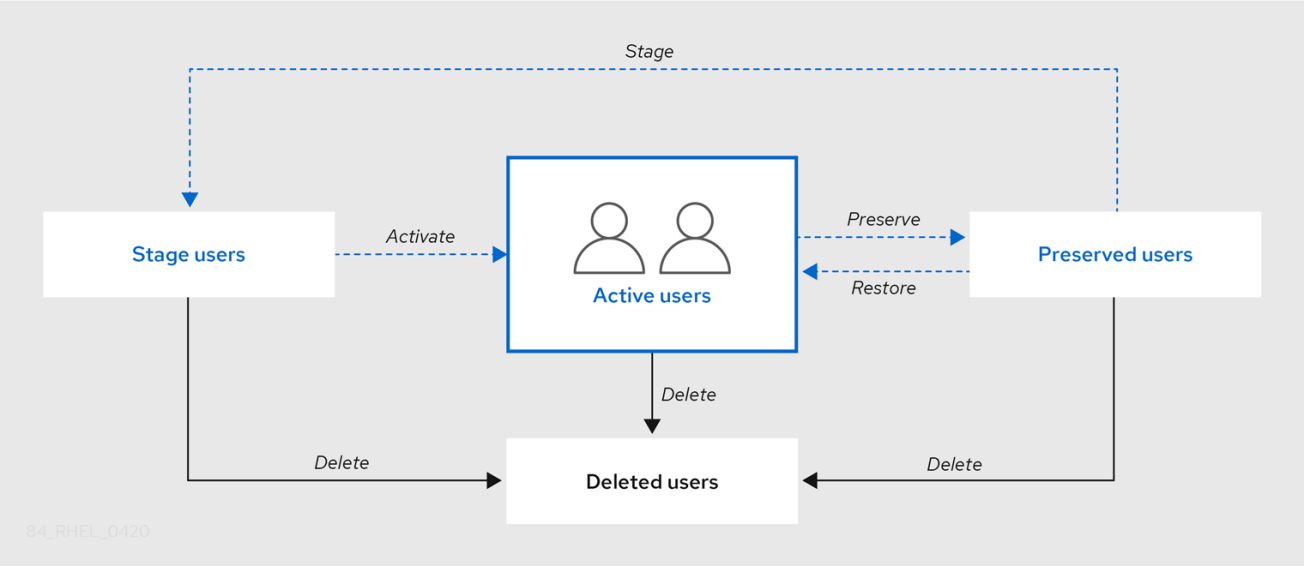
이 장에는 IdM의 사용자 라이프사이클에 대한 기본 설명(ID 관리)이 포함되어 있습니다. 다음 섹션에서는 다음을 수행하는 방법을 보여줍니다.

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

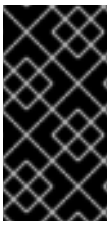
2.1. 사용자 라이프 사이클

IdM(Identity Management)은 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.

2.2. 명령줄을 사용하여 사용자 추가

다음과 같이 사용자를 추가할 수 있습니다.

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

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

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

참고

IdM은 고유한 사용자 ID(UID)를 새 사용자 계정에 자동으로 할당합니다. 이 작업을 수동으로 수행할 수도 있지만 서버에서 UID 번호가 고유한지 여부를 확인하지 않습니다. 이로 인해 여러 사용자 항목이 동일한 ID 번호가 할당될 수 있습니다. 동일한 UID가 있는 여러 항목이 없는 것을 방지하는 것이 좋습니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- Kerberos 티켓을 획득했습니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

1. 터미널을 열고 IdM 서버에 연결합니다.

2. 사용자 로그인, 사용자의 이름, 성을 추가하고 필요한 경우 이메일 주소를 추가할 수도 있습니다.

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

IdM에서는 다음 정규식으로 설명할 수 있는 사용자 이름을 지원합니다.

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

참고

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

대문자가 포함된 사용자 이름을 추가하는 경우 IdM은 자동으로 이름을 저장할 때 소문자로 변환합니다. 따라서 IdM은 로그인할 때 항상 사용자 이름을 소문자로 입력해야 합니다. 또한 사용자 및 사용자 등 문자 캐싱에서만 다른 사용자 이름을 추가할 수 없습니다 .

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

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

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

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

ipa help 명령에 대한 자세한 내용은 IPA 도움말을 참조하십시오.

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

```
$ ipa user-find
```

이 명령은 세부 정보가 있는 모든 사용자 계정을 나열합니다.

2.3. 명령줄을 사용하여 사용자 활성화

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

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- Kerberos 티켓을 획득했습니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

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

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

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

```
$ ipa user-find
```

이 명령은 세부 정보가 있는 모든 사용자 계정을 나열합니다.

2.4. 명령줄을 사용하여 사용자 보존

사용자 계정을 삭제하려면 보존할 수 있지만 나중에 복원할 수 있는 옵션을 유지합니다. 사용자 계정을 유지하려면 **ipa user-del** 또는 **ipa stageuser-del** 명령과 함께 **--preserve** 옵션을 사용합니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- Kerberos 티켓을 획득했습니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

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

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

참고

사용자 계정이 삭제되었다는 출력에도 불구하고 보존되었습니다.

2.5. 명령줄을 사용하여 사용자 삭제

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

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

- 다음 명령을 사용하여 사용자 보존: **ipa user-del**

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

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

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

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **Kerberos** 티켓을 획득했습니다. 자세한 내용은 **kinit**를 사용하여 **IdM**에 수동으로 로그인하는 방법을 참조하십시오.

절차

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

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

사용자 계정이 **IdM**에서 영구적으로 삭제되었습니다.

2.6. 명령줄을 사용하여 사용자 복원

보존된 사용자를 다음과 같이 복원할 수 있습니다.

- 활성 사용자: **ipa user-undel**

- 단계 사용자: **ipa** 사용자 단계

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

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

- Kerberos 티켓을 획득했습니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

1. 터미널을 열고 IdM 서버에 연결합니다.

2. 다음 명령으로 사용자 계정을 활성화합니다.

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

또는 사용자 계정을 스테이징된 상태로 복원할 수 있습니다.

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

검증 단계

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

```
$ ipa user-find
```

이 명령은 세부 정보가 있는 모든 사용자 계정을 나열합니다.

3장. IDM 웹 UI를 사용하여 사용자 계정 관리

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

사용자 계정 생성

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

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

사용자 계정 활성화

직원의 첫 근무일 계정 활성화.

사용자 계정 비활성화

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

사용자 계정 활성화

사용자가 반환되면 계정을 다시 활성화해야 합니다.

사용자 계정 보존

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

사용자 계정 복원

2년 후 사용자가 다시 시작되어 보존된 계정을 복원해야 합니다.

사용자 계정 삭제

직원이 차감된 경우 백업없이 계정을 삭제합니다.

3.1. 사용자 라이프 사이클

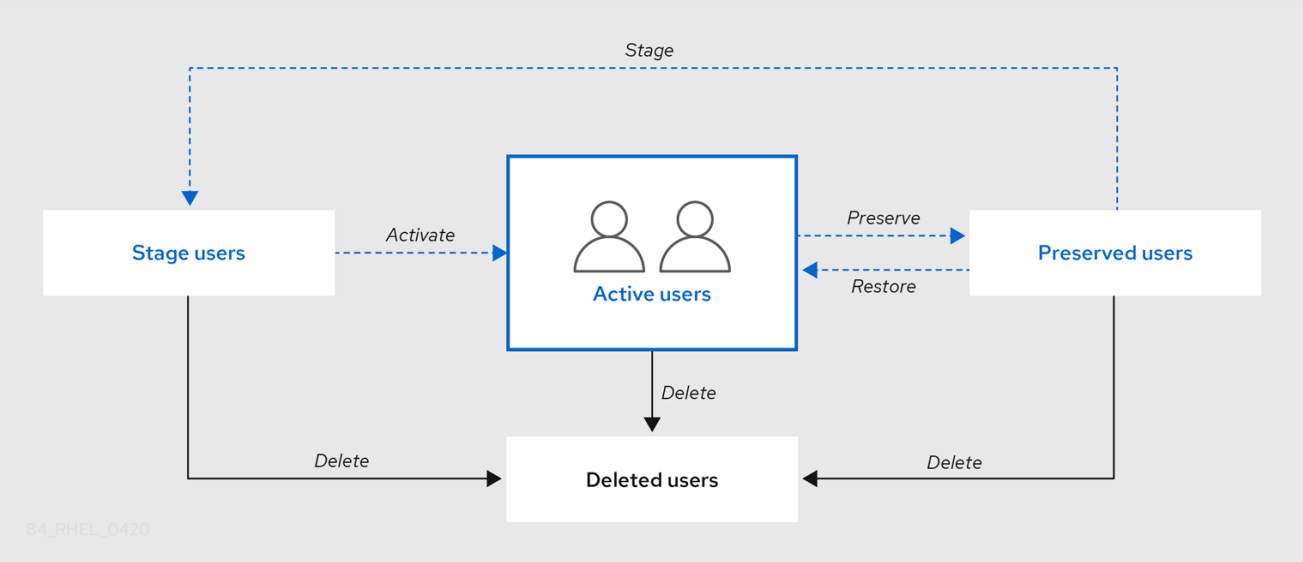
IdM(Identity Management)은 세 가지 사용자 계정 상태를 지원합니다.

-

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

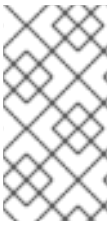
admin 사용자를 삭제하지 마십시오. admin 은 IdM에 필요한 사전 정의된 사용자이므로 이 작업으로 인해 특정 명령에서 문제가 발생합니다. 대체 admin 사용자를 정의하고 사용하려는 경우 하나 이상의 다른 사용자에게 admin 권한을 부여한 후 ipa 사용자 비활성화 admin으로 사전 정의된 admin 사용자를 비활성화합니다.

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.

3.2. 웹 UI에서 사용자 추가

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



참고

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

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

절차

1. IdM 웹 UI에 로그인합니다.

   자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

2. 사용자 → 사용자 단계 탭으로 이동합니다.

또는 사용자 → 활성 사용자에게 사용자 계정을 추가할 수도 있지만 계정에 사용자 그룹을 추가할 수는 없습니다.

3. + Add(추가) 아이콘을 클릭합니다.

4. Add stage user (단계 사용자 추가) 대화 상자에 새 사용자의 이름 및 이름을 입력합니다.

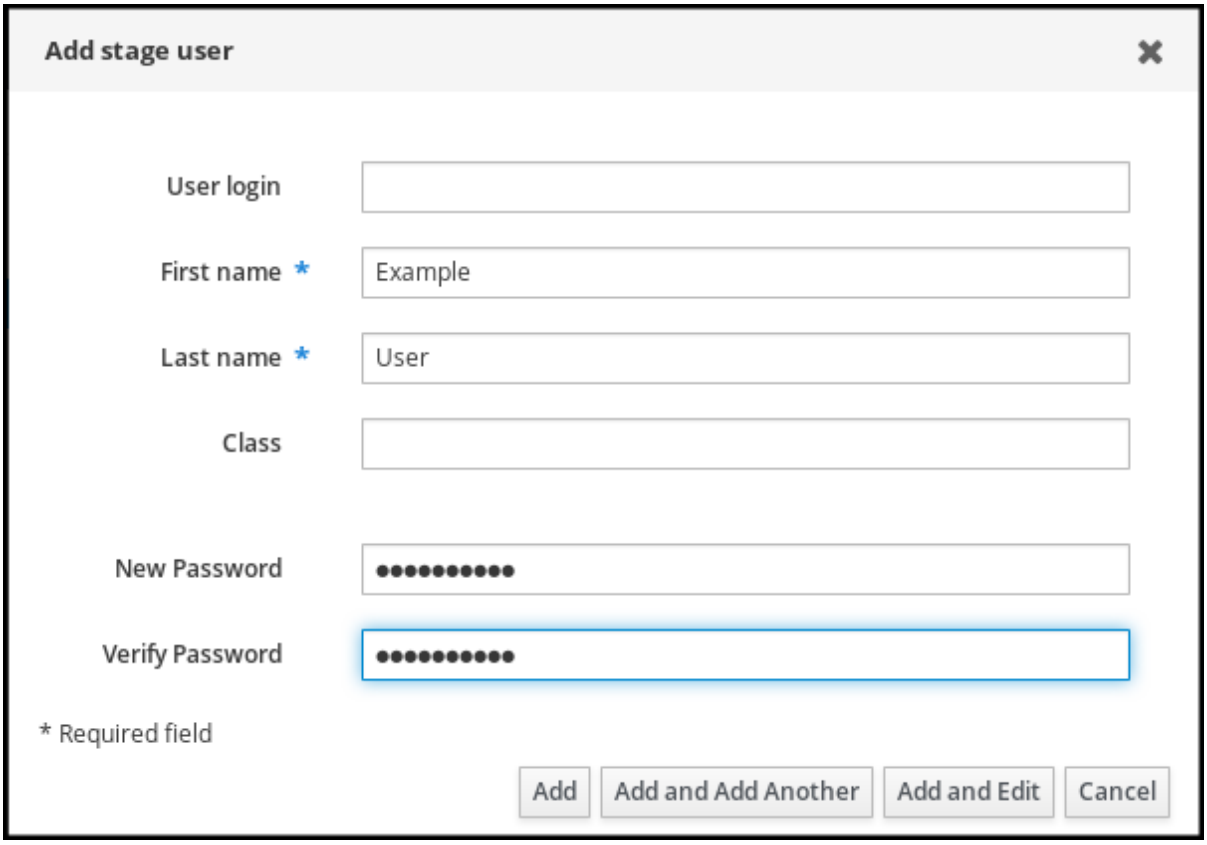
5. [선택 사항] 사용자 로그인 필드에 로그인 이름을 추가합니다.

비워 두면 IdM 서버에서 다음 패턴으로 로그인 이름을 생성합니다. 이름 및 성의 첫 글자. 전체 로그인 이름은 최대 32자를 가질 수 있습니다.

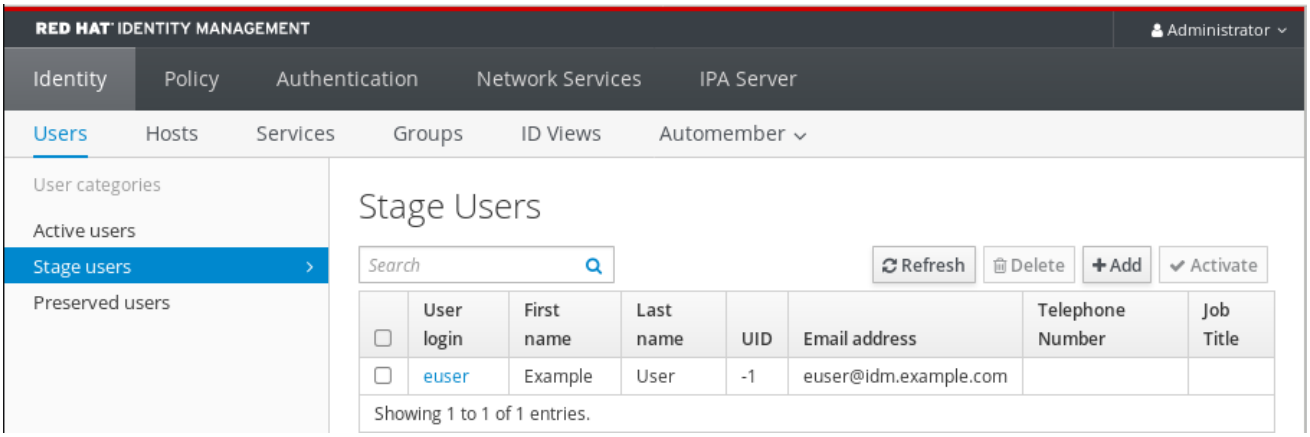
6. [선택 사항] GID 드롭다운 메뉴에서 사용자가 포함해야 하는 그룹을 선택합니다.

7. [선택 사항] Password 및 Verify password 필드에 암호를 입력하고 확인하고 둘 다 일치하는지 확인합니다.

8. Add(추가) 단추를 클릭합니다.



이때 **Stage Users** (사용자 단계) 테이블에 사용자 계정을 볼 수 있습니다.



참고

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

3.3. IDM 웹 UI에서 단계 사용자 활성화

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

사전 요구 사항

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

절차

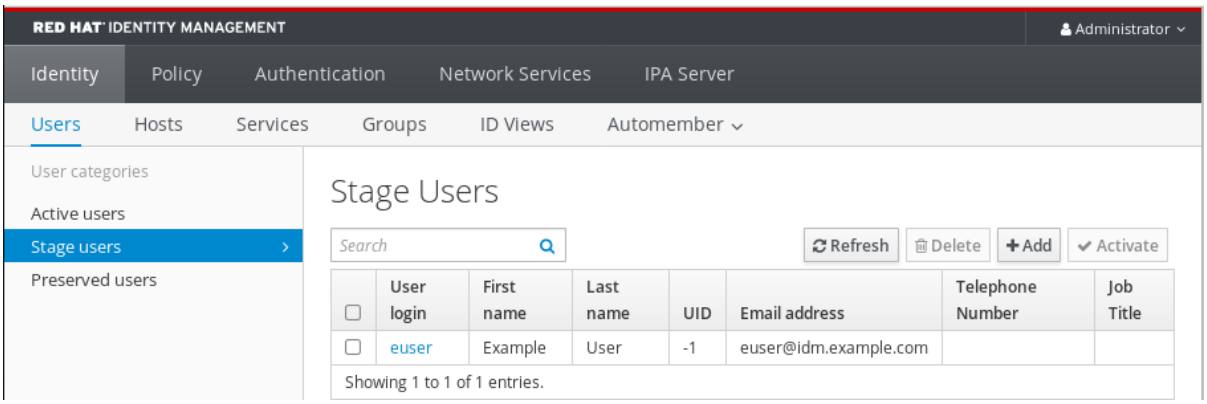
1. IdM 웹 UI에 로그인합니다.

   자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

2. 사용자 → 사용자 탭으로 이동합니다.

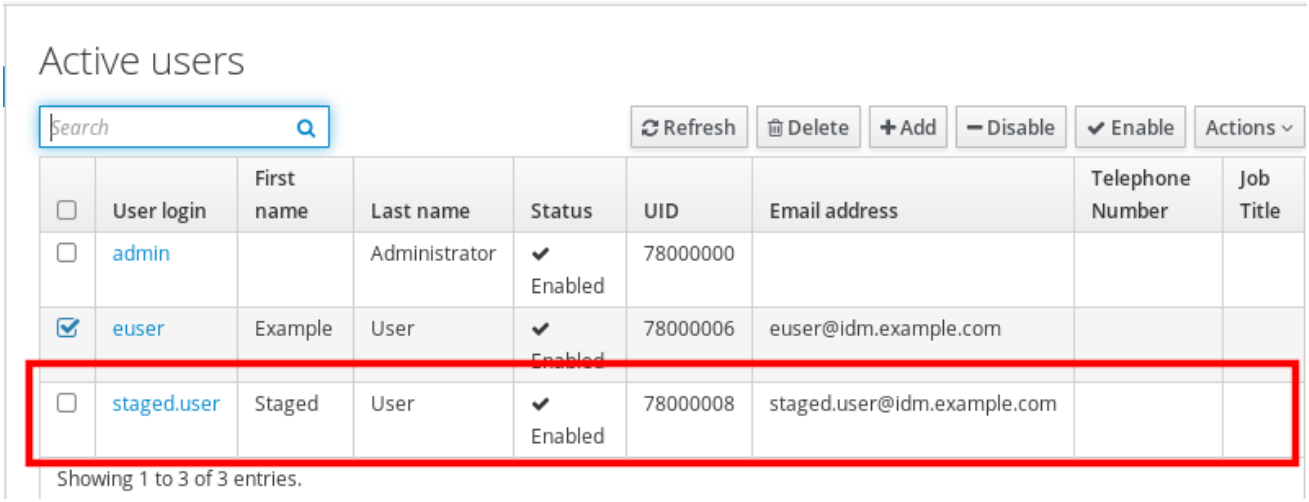
3. 활성화하려는 사용자 계정의 확인란을 클릭합니다.

4. Activate(활성화) 버튼을 클릭합니다.



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

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



참고

이 단계에서는 활성 사용자 계정을 사용자 그룹에 추가할 수 있습니다.

3.4. 웹 UI에서 사용자 계정 비활성화

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

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

참고

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

사전 요구 사항

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

절차

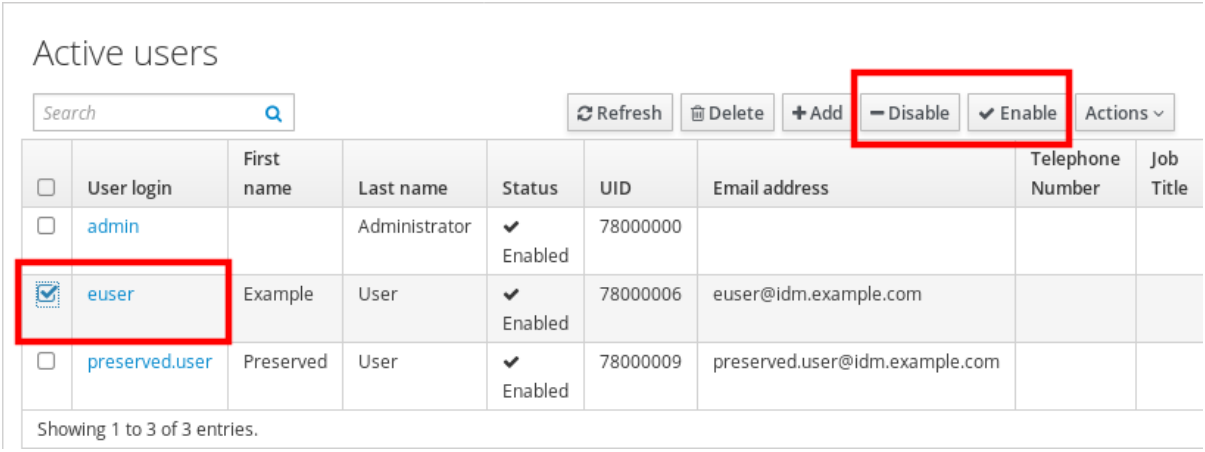
1. IdM 웹 UI에 로그인합니다.

   자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

2. 사용자 → 활성 사용자 탭으로 이동합니다.

3. 비활성화하려는 사용자 계정의 확인란을 클릭합니다.

4. Disable(비활성화) 버튼을 클릭합니다.



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

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

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

3.5. 웹 UI에서 사용자 계정 활성화

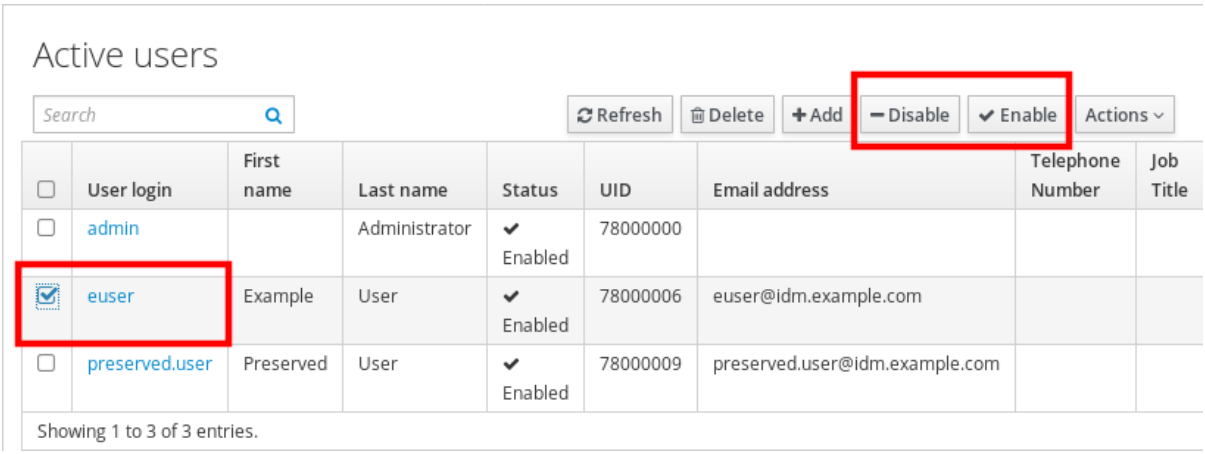
IdM을 사용하면 비활성화된 활성 사용자 계정을 활성화할 수 있습니다. 사용자 계정을 활성화하면 비활성화된 계정을 활성화합니다.

사전 요구 사항

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

절차

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



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

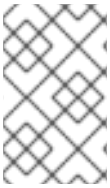
변경이 성공적으로 수행된 경우 **Active users (활성 사용자)** 테이블의 **Status(상태)** 열에서 확인할 수 있습니다.

3.6. IDM 웹 UI에서 활성 사용자 보존

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

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

복원된 계정을 활성 모드로 다시 이동할 수 있습니다.



참고

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

사전 요구 사항

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

절차

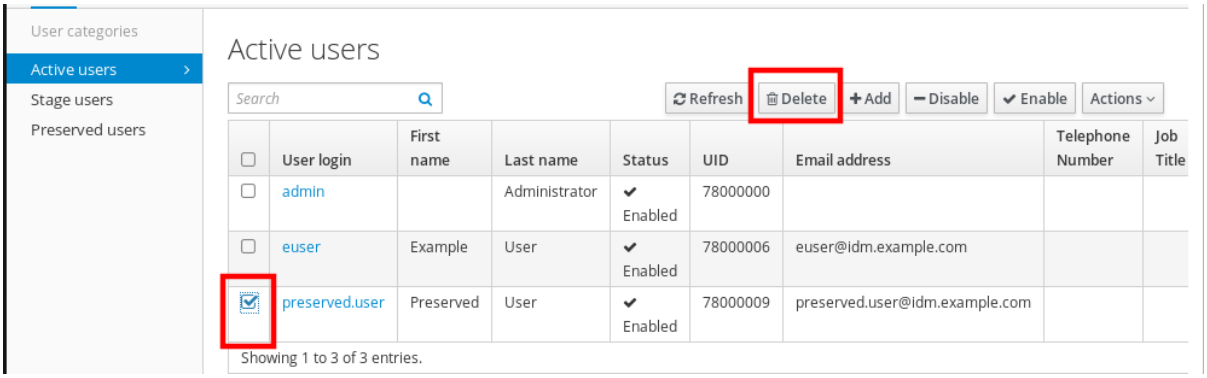
1. **IdM 웹 UI에 로그인합니다.**

   자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

2. **사용자 → 활성 사용자 탭으로 이동합니다.**

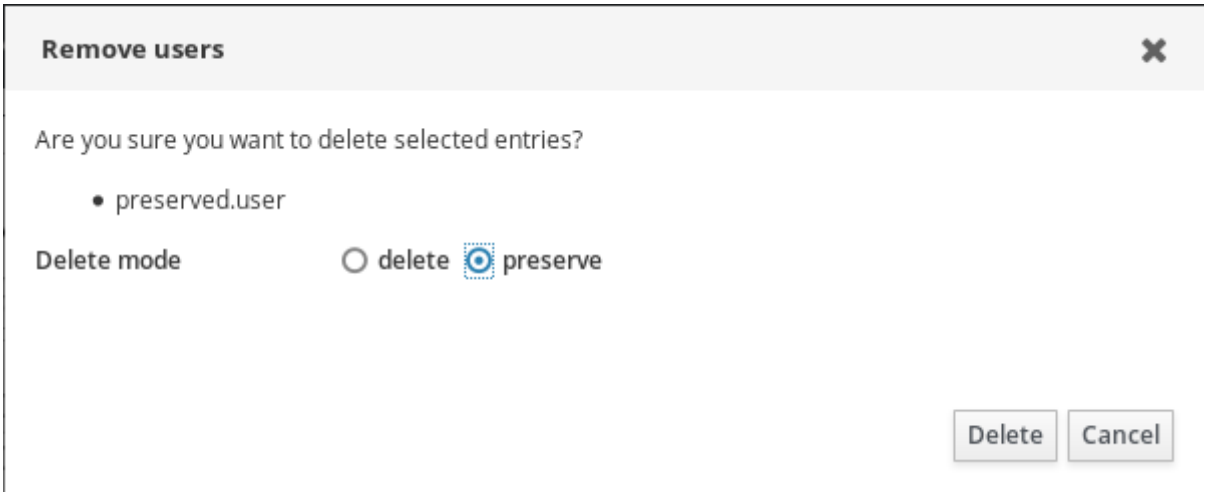
3. **보존할 사용자 계정의 확인란을 클릭합니다.**

4. **Delete(삭제) 단추를 클릭합니다.**



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

6. **Delete(삭제) 단추를 클릭합니다.**



결과적으로 사용자 계정이 보존된 사용자로 이동됩니다.

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

3.7. IDM 웹 UI에서 사용자 복원

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

사전 요구 사항

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

절차

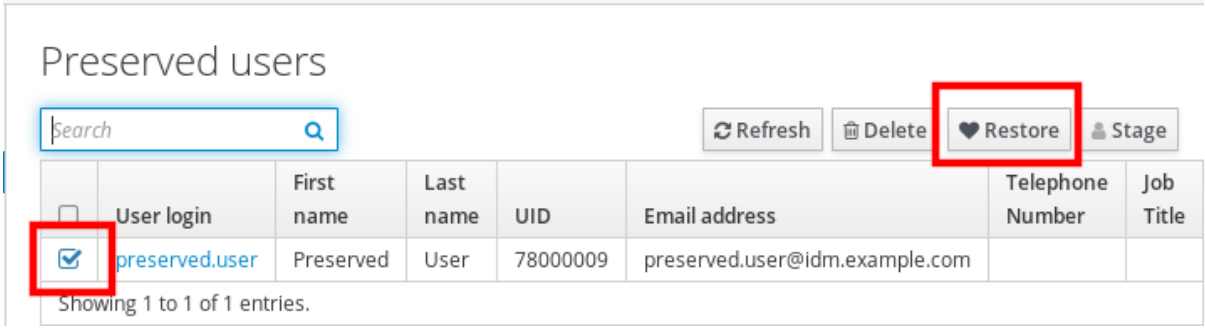
1. IdM 웹 UI에 로그인합니다.

   자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

2. 사용자 → 보존된 사용자 탭으로 이동합니다.

3. 복원할 사용자 계정에서 체크박스를 클릭합니다.

4. **Restore** (복구) 버튼을 클릭합니다.



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

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

3.8. IDM 웹 UI에서 사용자 삭제

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

다음을 삭제할 수 있습니다.

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

    자세한 내용은 **IdM** 웹 **UI**에서 활성 사용자 보존을 참조하십시오.
  - 영구적으로 삭제
- 스테이징 사용자 - 단계 사용자를 영구적으로 삭제할 수 있습니다.

- 보존된 사용자 - 보존된 사용자를 영구적으로 삭제할 수 있습니다.

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

- Stage users(단계 사용자) 탭
- 보존된 사용자 탭

사전 요구 사항

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

절차

1. IdM 웹 UI에 로그인합니다.

   자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

2. 사용자 → 활성 사용자 탭으로 이동합니다.

   또는 사용자 → 사용자 단계 또는 사용자 → 보존된 사용자의 사용자 계정을 삭제할 수 있습니다.

3. Delete(삭제) 아이콘을 클릭합니다.

4. Remove users(사용자 제거) 대화 상자에서 Delete mode(삭제 모드) 라디오 버튼을 삭제하도록 전환합니다.

5. Delete(삭제) 단추를 클릭합니다.

사용자 계정이 IdM에서 영구적으로 삭제되었습니다.

4장. ANSIBLE 플레이북을 사용하여 사용자 계정 관리

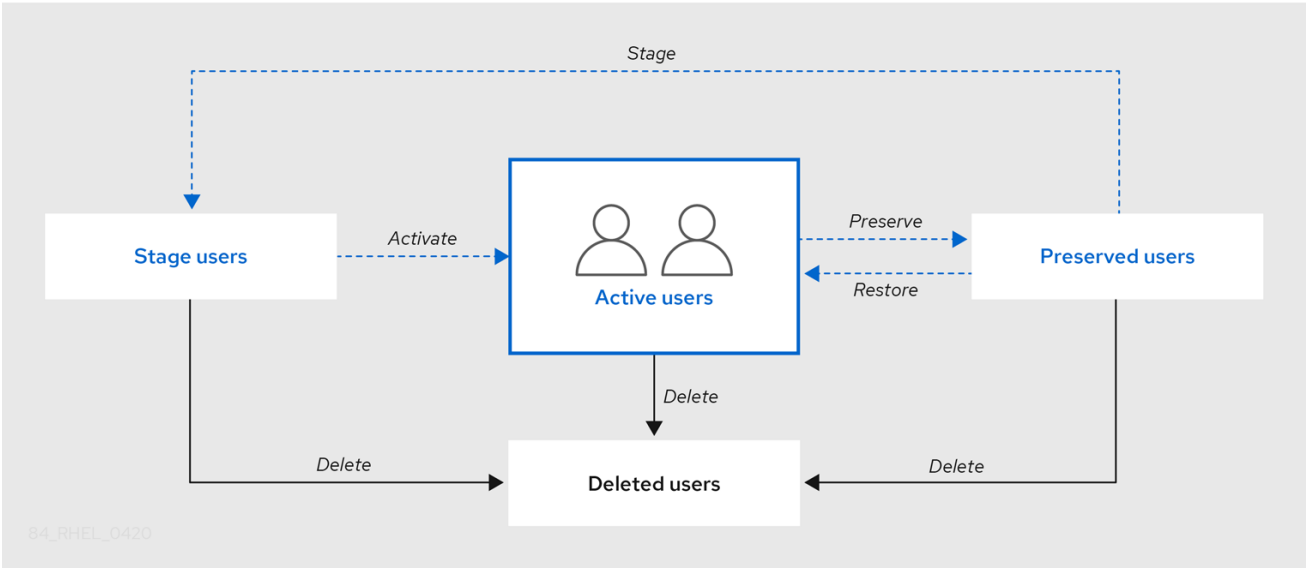
Ansible 플레이북을 사용하여 IdM에서 사용자를 관리할 수 있습니다. 사용자 라이프사이클 을 제공한 후 이 장에서는 다음 작업에 Ansible 플레이북을 사용하는 방법을 설명합니다.

- 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.

4.1. 사용자 라이프 사이클

IdM(Identity Management)은 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 Manager**에서 **Directory Server**에서 새 관리자를 수동으로 생성해야 합니다.

주의

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

주의

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

4.2. 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 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 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.

4.3. 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.

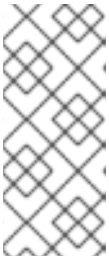
```
[ipaserver]
server.idm.example.com
```

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.

4.4. 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.

4.5. 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.

4.6. 추가 리소스

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

5장. IDM 클라이언트의 IDM 사용자에게 SUDO 액세스 권한 부여

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

5.1. IDM 클라이언트에서 SUDO 액세스

시스템 관리자는 루트가 아닌 사용자가 일반적으로 root 사용자에게 예약된 관리 명령을 실행할 수 있도록 sudo 액세스 권한을 부여할 수 있습니다. 따라서 사용자가 일반적으로 root 사용자로 예약된 관리 명령을 수행해야 하는 경우 sudo 를 사용하여 해당 명령 앞에 추가합니다. 암호를 입력한 후에는 루트 사용자인 것처럼 명령이 실행됩니다. 데이터베이스 서비스 계정과 같은 다른 사용자 또는 그룹으로 sudo 명령을 실행하려면 sudo 규칙에 대해 *RunAs 별칭* 을 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

- sudo 액세스 관리를 참조하십시오.

5.2. 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 만

듭니다.

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.
- idmclient 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:
```

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

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

idm_user_reboot sudo 규칙을 생성하여 idm _user 계정에 idm client 시스템에서 /usr/sbin/reboot 명령을 실행할 수 있는 권한을 부여하려면 이 절차를 완료합니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.
- IdM에서 idm_user 에 대한 사용자 계정을 생성하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. 명령줄 인터페이스를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄을 사용하여 사용자 추가를 참조하십시오.
-

idmclient 호스트에 로컬 **idm_user** 계정이 없습니다. **idm_user** 사용자는 로컬 **/etc/passwd** 파일에 나열되지 않습니다.

절차

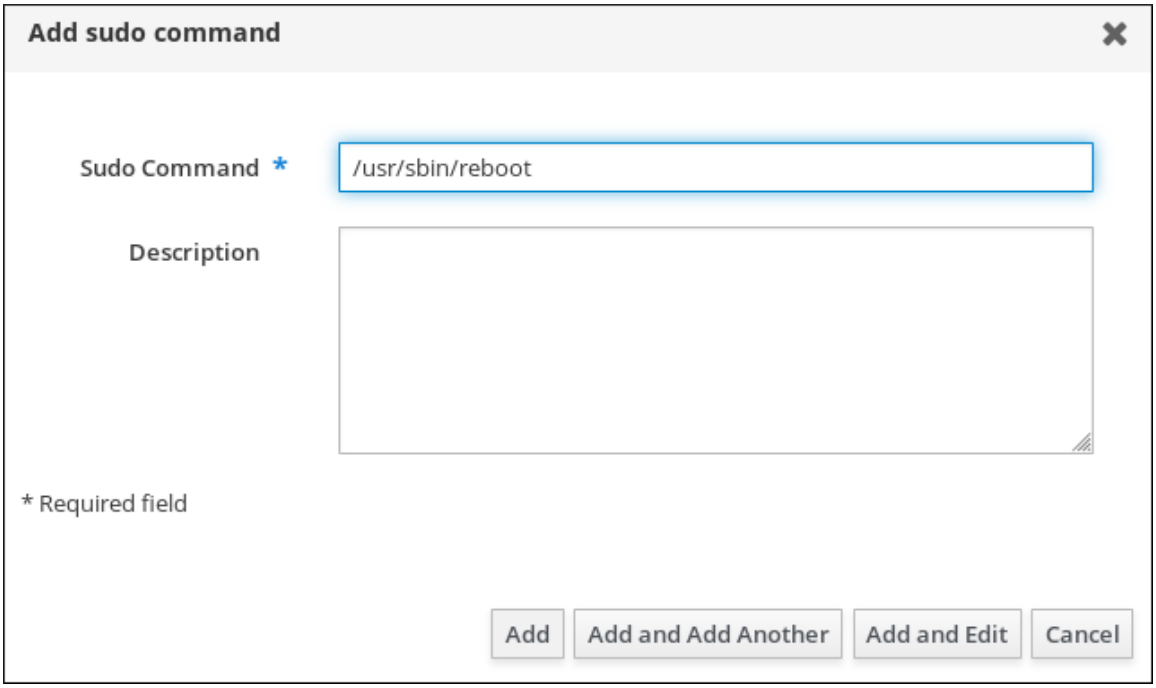
1. **/usr/sbin/reboot** 명령을 **sudo** 명령의 **IdM** 데이터베이스에 추가합니다.

   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 **5.1. IdM sudo** 명령 추가



   d. 추가를 클릭합니다.

2. 새 **sudo** 명령 항목을 사용하여 **idm_user**가 **idm client** 시스템을 재부팅할 수 있도록 **sudo** 규칙을 생성합니다.

   a. 정책 → **Sudo** → **Sudo** 규칙으로 이동합니다.

b. 오른쪽 상단 모서리에서 **Add**(추가)를 클릭하여 **Add sudo rule**(**sudo** 규칙 추가) 대화 상자를 엽니다.

c. **sudo** 규칙의 이름을 **idm_user_reboot** 로 입력합니다.

d. **Add and Edit**(추가 및 편집)를 클릭합니다.

e. 사용자를 지정합니다.

i. **who**(사용자) 섹션에서 **Specified Users and Groups**(지정된 사용자 및 그룹) 라디오 단추를 선택합니다.

ii. **User** 카테고리에서 규칙이 하위 섹션에 적용되는 경우 **Add**(추가)를 클릭하여 **Add users into sudo rule "idm_user_reboot"** 대화 상자를 엽니다.

iii. **Add users into sudo rule "idm_user_reboot"** 대화 상자의 **Available** 열에서 **idm_user** 확인란을 선택하고 **Prospective** 열로 이동합니다.

iv. 추가를 클릭합니다.

f. 호스트를 지정합니다.

i. **Access this host** (이 호스트에 액세스) 섹션에서 **Specified Hosts and Groups**(지정된 호스트 및 그룹) 라디오 단추를 선택합니다.

ii. 호스트 범주에서 이 규칙이 하위 섹션에 적용되는 경우 **Add**(추가)를 클릭하여 **Add hosts into sudo rule "idm_user_reboot"** 대화 상자를 엽니다.

iii. **Add hosts to sudo rule "idm_user_reboot"** 대화 상자의 **Available** 열에서 **idmclient.idm.example.com** 확인란을 선택하고 **Prospective** 열로 이동합니다.

.

iv. 추가를 클릭합니다.

g. 명령을 지정합니다.

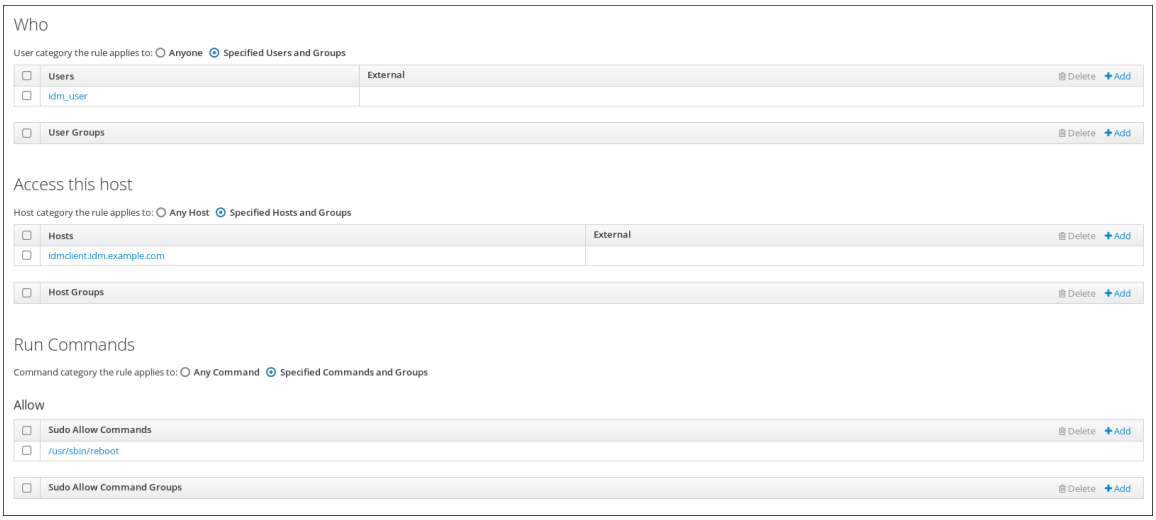
i. **Command** 카테고리에서 규칙이 **Run Commands**(명령 실행) 섹션의 하위 섹션에 적용되는 경우 **Specified Commands and Groups**(지정된 명령) 및 **Groups** (그룹) 라디오 버튼을 선택합니다.

ii. **Sudo Allow Commands** (Sudo Allow Commands) 하위 섹션에서 **Add allow sudo** 명령을 **sudo rule "idm_user_reboot"** 대화 상자에 엽니다.

iii. **Add allow sudo** 명령을 **sudo** 규칙 **"idm_user_reboot"** 대화 상자의 **Available** 열에서 **/usr/sbin/reboot** 확인란을 선택하고 **Prospective** 열로 이동합니다.

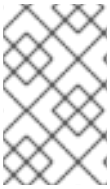
iv. **Add**(추가)를 클릭하여 **idm_sudo_reboot** 페이지로 돌아갑니다.

그림 5.2. IdM sudo 규칙 추가



h. 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. 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 규칙이 올바르게 구성되면 시스템이 재부팅됩니다.

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

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

이 예제를 사용하여 run_ third-party-app_report 라는 명령에 sudo 규칙을 생성하여 idm_user 계정이 idmclient 호스트의 thirdpartyapp 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.
- idmclient 호스트에 로컬 idm_user 계정이 없습니다. idm_user 사용자는 로컬 /etc/passwd 파일에 나열되지 않습니다.

- idmclient 호스트에 타사-app 이라는 사용자 지정 애플리케이션이 설치되어 있습니다.
- 타사 애플리케이션에 대한 report 명령이 /opt/ third-party-app /bin/report 디렉터리에 설치됩니다.
- third-party-app 애플리케이션에 대한 명령을 실행하도록 thirdpartyapp 이라는 로컬 서비스 계정을 생성했습니다.

절차

1. IdM 관리자로 Kerberos 티켓을 검색합니다.

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

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

```
[root@idmclient ~]# ipa sudocmd-add /opt/third-party-app/bin/report
----------------------------------------------------
Added Sudo Command "/opt/third-party-app/bin/report"
----------------------------------------------------
  Sudo Command: /opt/third-party-app/bin/report
```

3. 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 대해 RunAs 사용자를 지정합니다.

```
[root@idmclient ~]# ipa sudorule-add-runasuser run_third-party-app_report --
users=thirdpartyapp
  Rule name: run_third-party-app_report
  Enabled: TRUE
  RunAs External User: thirdpartyapp
```

```
-------------------------
Number of members added 1
-------------------------
```

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

5. **/opt/ third-party-app/bin/report** 명령을 **idm_user_reboot** 규칙에 추가합니다.

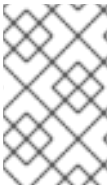
```
[root@idmclient ~]# ipa sudorule-add-allow-command run_third-party-app_report --
sudocmds '/opt/third-party-app/bin/report'
Rule name: run_third-party-app_report
Enabled: TRUE
Sudo Allow Commands: /opt/third-party-app/bin/report
RunAs External User: thirdpartyapp
-------------------------
Number of members added 1
-------------------------
```

6. **IdM idmclient** 호스트에 **run_ third-party-app_report** 규칙을 적용합니다.

```
[root@idmclient ~]# ipa sudorule-add-host run_third-party-app_report --hosts
idmclient.idm.example.com
Rule name: run_third-party-app_report
Enabled: TRUE
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /opt/third-party-app/bin/report
RunAs External User: thirdpartyapp
-------------------------
Number of members added 1
-------------------------
```

7. **idm_user** 계정을 **run_ third-party-app_report** 규칙에 추가합니다.

```
[root@idmclient ~]# ipa sudorule-add-user run_third-party-app_report --users
idm_user
Rule name: run_third-party-app_report
Enabled: TRUE
Users: idm_user
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /opt/third-party-app/bin/report
RunAs External User: thirdpartyapp
-------------------------
Number of members added 1
```



참고

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

검증 단계

1. idmclient 호스트에 idm _user 계정으로 로그인합니다.

2. 새 sudo 규칙을 테스트합니다.

a. idm_user 계정이 수행할 수 있는 sudo 규칙을 표시합니다.

```
[idm_user@idmclient ~]$ sudo -l
Matching Defaults entries for idm_user@idm.example.com on idmclient:
    !visiblepw, always_set_home, match_group_by_gid,
always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS
LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
XAUTHORITY KRB5CCNAME",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User idm_user@idm.example.com may run the following commands on idmclient:
    (thirdpartyapp) /opt/third-party-app/bin/report
```

b. thirdpartyapp 서비스 계정으로 report 명령을 실행합니다.

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

5.5. IDM 클라이언트에서 서비스 계정으로 명령을 실행하는 IDM WEBUI에서 SUDO 규칙 생성

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

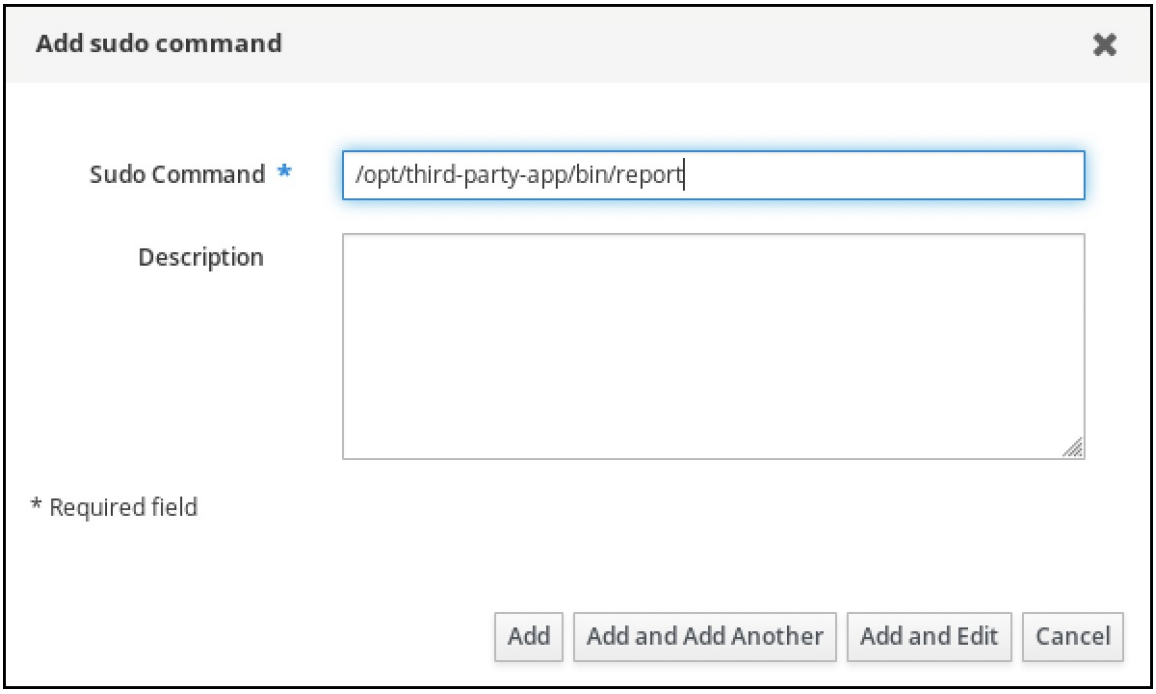
이 예제를 사용하여 **idm_user** 계정이 **idmclient** 호스트에서 타사 서비스 계정으로 **/opt/ third-party-app/report** 명령을 실행할 수 있도록 **run_ third-party- app _report** 라는 **IdM** 웹 **UI**에 **sudo** 규칙을 생성합니다.

사전 요구 사항

- **IdM** 관리자로 로그인했습니다.
- **IdM**에서 **idm_user** 에 대한 사용자 계정을 생성하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. **CLI**를 사용하여 새 **IdM** 사용자를 추가하는 방법에 대한 자세한 내용은 명령줄을 사용하여 사용자 추가를 참조하십시오.
- **idmclient** 호스트에 로컬 **idm_user** 계정이 없습니다. **idm_user** 사용자는 로컬 **/etc/passwd** 파일에 나열되지 않습니다.
- **idmclient** 호스트에 타사**-app** 이라는 사용자 지정 애플리케이션이 설치되어 있습니다.
- 타사 애플리케이션에 대한 **report** 명령이 **/opt/ third-party-app /bin/report** 디렉터리에 설치됩니다.
- **third-party-app** 애플리케이션에 대한 명령을 실행하도록 **thirdpartyapp** 이라는 로컬 서비스 계정을 생성했습니다.

절차

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



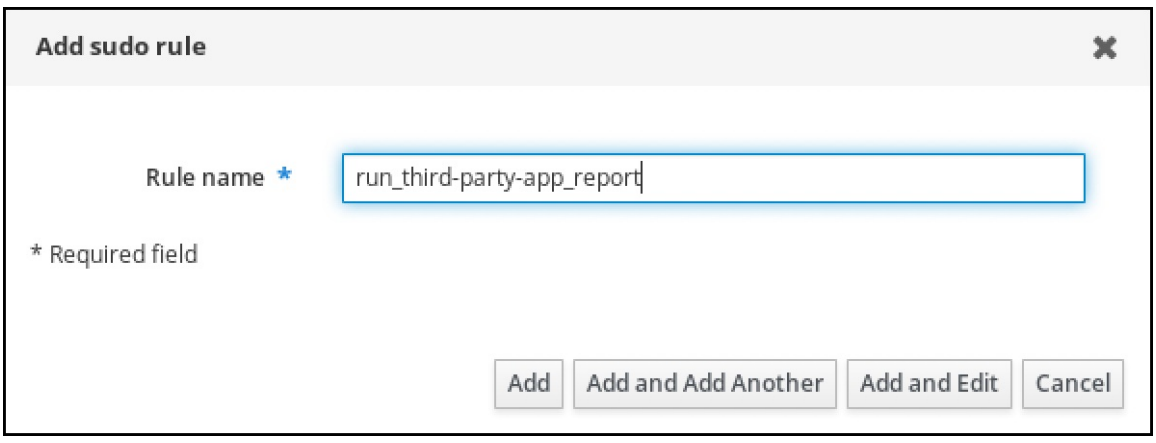
d. 추가를 클릭합니다.

2. 새 sudo 명령 항목을 사용하여 새 sudo 규칙을 만듭니다.

a. 정책 → Sudo → Sudo 규칙으로 이동합니다.

b. 오른쪽 상단 모서리에서 Add(추가)를 클릭하여 Add sudo rule(sudo 규칙 추가) 대화 상자를 엽니다.

c. sudo 규칙의 이름을 입력합니다. run_ third-party-app_report.



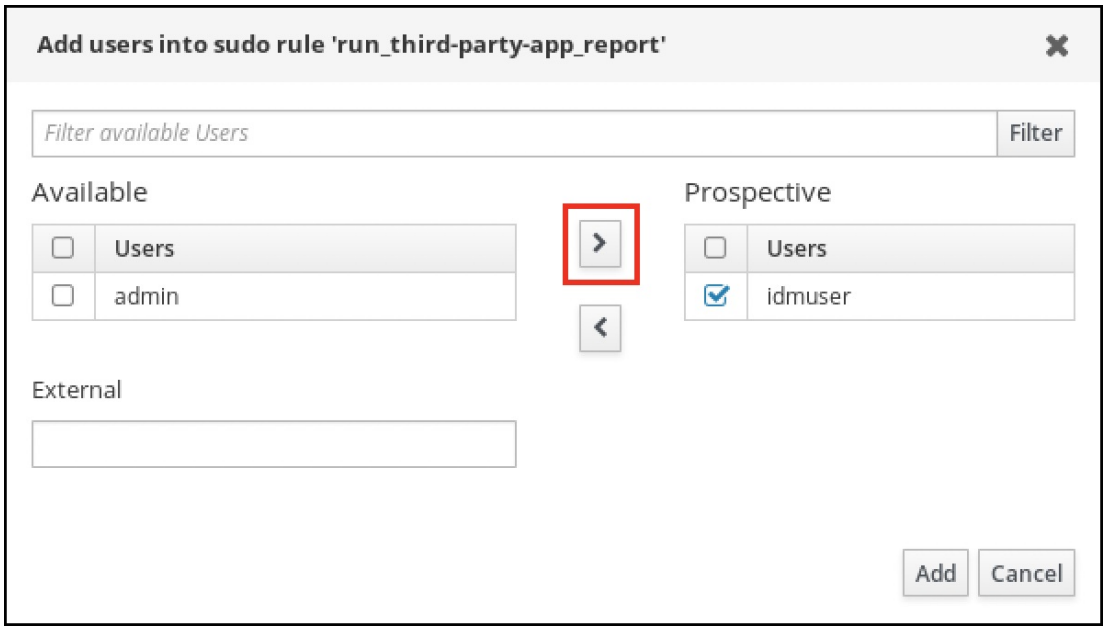
d. Add and Edit(추가 및 편집)를 클릭합니다.

e. 사용자를 지정합니다.

i. **who**(사용자) 섹션에서 **Specified Users and Groups**(지정된 사용자 및 그룹) 라디오 단추를 선택합니다.

ii. **User category the rule applies to** 하위 섹션에 적용되는 규칙 추가 를 클릭하여 **sudo** 규칙 **"run_ third-party-app_report"** 대화 상자에 사용자 추가 를 엽니다.

iii. **Add users into sudo rule "run_ third-party-app_report"** 대화 상자에서 **Available** 열의 **idm_user** 확인란을 확인하고 **Prospective** 열로 이동합니다.



iv. 추가를 클릭합니다.

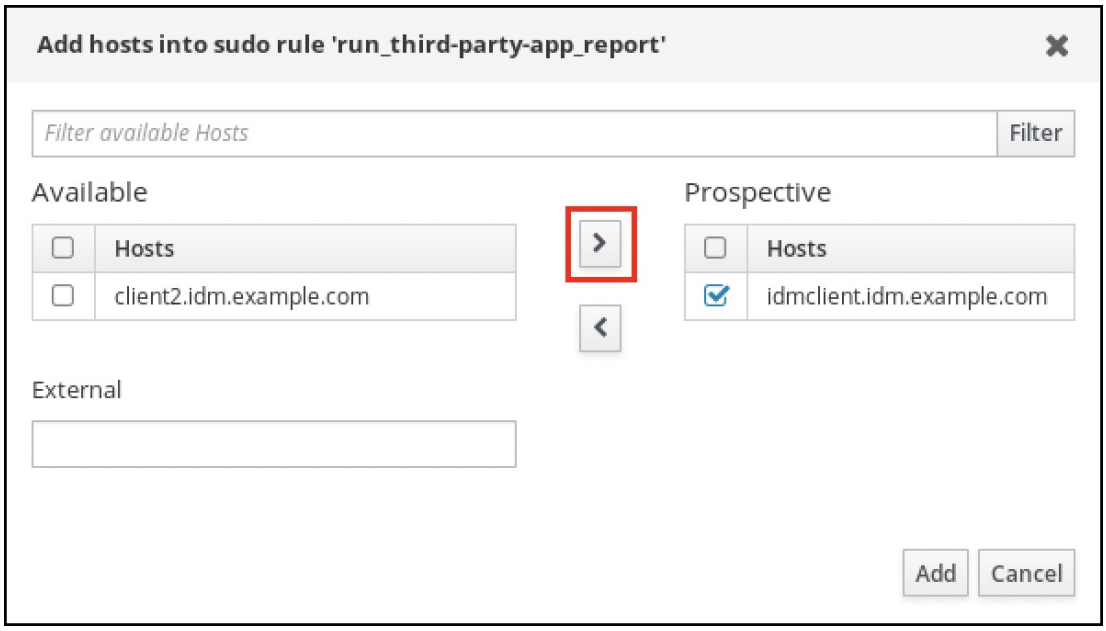
f. 호스트를 지정합니다.

i. **Access this host** (이 호스트에 액세스) 섹션에서 **Specified Hosts and Groups**(지정된 호스트 및 그룹) 라디오 단추를 선택합니다.

ii. **Host category this rule applies to** 하위 섹션에 적용되는 규칙 추가 를 클릭하여 **sudo** 규칙 **"run_ third-party-app_report"** 대화 상자에 호스트 추가 를 엽니다.

iii. **Available** 열의 **Add hosts into sudo** 규칙 **"run_ third-party-app_report"** 대화

상자에서 **idmclient.idm.example.com** 확인란을 선택하고 이를 **Prospective** 열로 이동합니다.



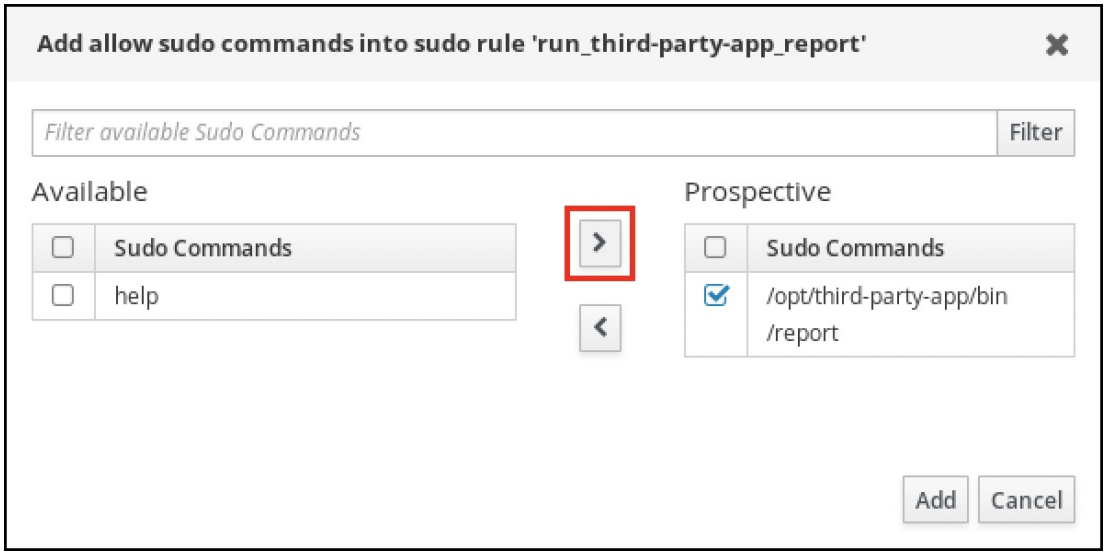
iv. 추가를 클릭합니다.

g. 명령을 지정합니다.

i. **Command** 카테고리에서 규칙이 **Run Commands**(명령 실행) 섹션의 하위 섹션에 적용되는 경우 **Specified Commands and Groups**(지정된 명령) 및 **Groups** (그룹) 라디오 버튼을 선택합니다.

ii. **Sudo Allow Commands** 하위 섹션에서 **Add** 를 클릭하여 **Add allow sudo** 명령을 **sudo** 규칙 **"run_ third-party-app_report"** 대화 상자에 엽니다.

iii. **Add allow sudo** 명령의 **Available** 열의 **sudo** 규칙 **"run_ third-party-app_report"** 대화 상자에서 **/opt/ third-party-app/bin/report** 확인란을 선택하고 이를 **Prospective** 열로 이동합니다.



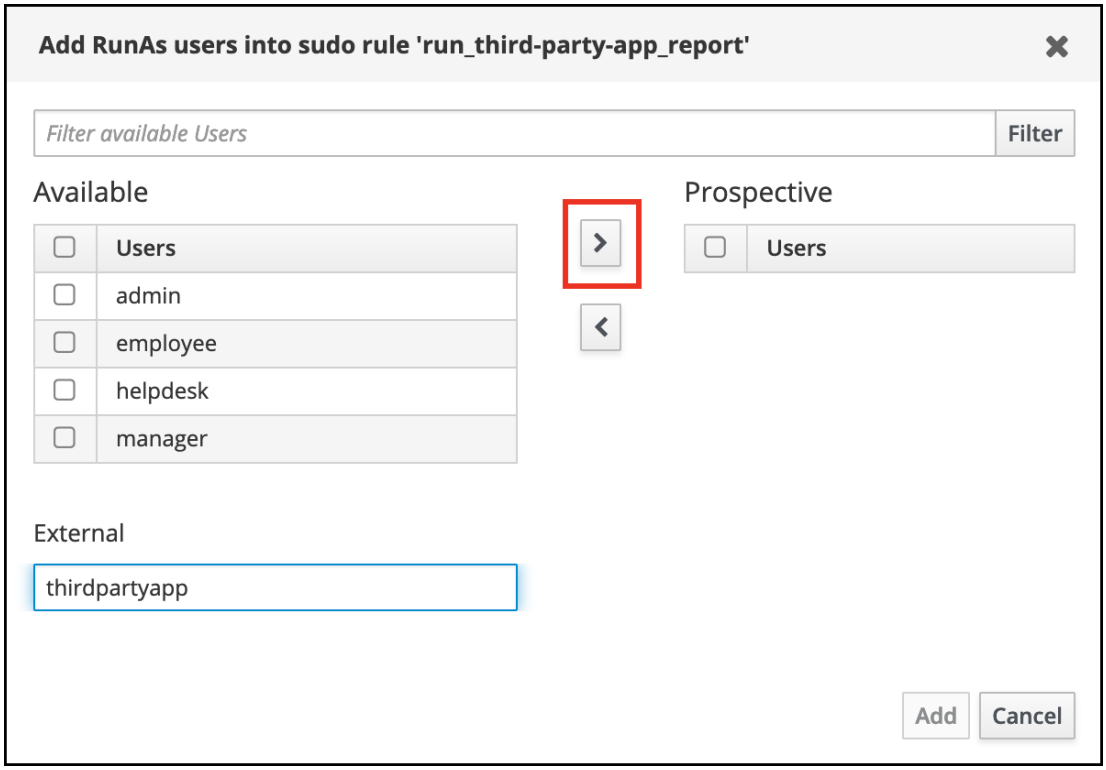
iv. **Add** 를 클릭하여 **run_ third-party-app_report** 페이지로 돌아갑니다.

h. **RunAs** 사용자를 지정합니다.

i. **As whom** 섹션에서 지정된 사용자 및 그룹 라디오 버튼을 확인합니다.

ii. **RunAs Users** 하위 섹션에서 **Add** 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 **External** 상자에 **third partyapp** 서비스 계정을 입력하고 **Prospective** 열로 이동합니다.

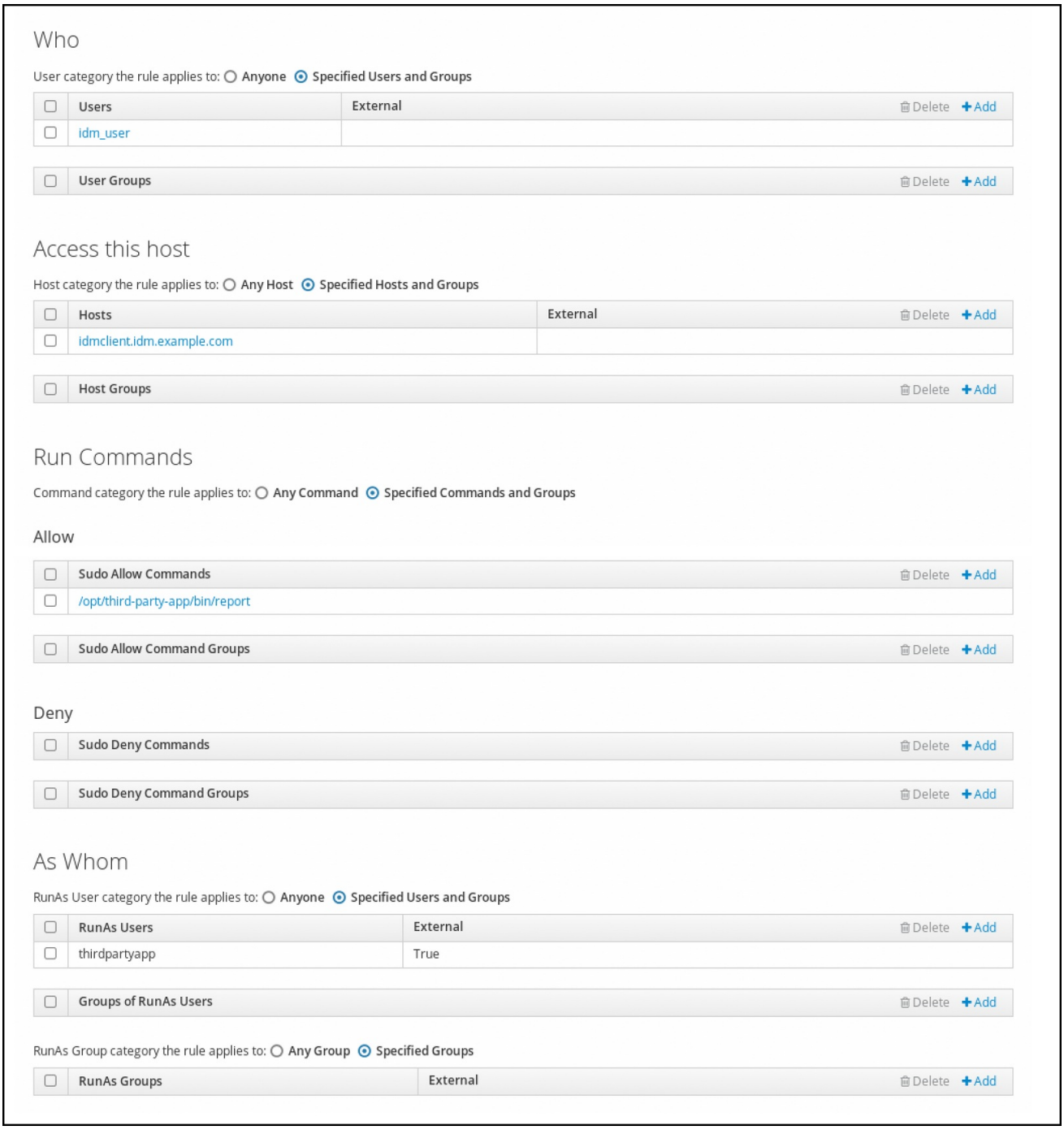


iv. **Add** 를 클릭하여 **run_ third-party-app_report** 페이지로 돌아갑니다.

i. 왼쪽 상단 모서리에서 **Save**(저장)를 클릭합니다.

새 규칙은 기본적으로 활성화되어 있습니다.

그림 5.3. sudo 규칙의 세부 사항



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. thirdpartyapp 서비스 계정으로 report 명령을 실행합니다.

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

5.6. IDM 클라이언트에서 SUDO에 대해 GSSAPI 인증 활성화

다음 절차에서는 pam_sss_gss.so PAM 모듈을 통해 sudo 및 sudo -i 명령에 대한 IdM 클라이언트에서 Generic Security Service Application Program Interface(GSSAPI) 인증을 활성화하는 방법을 설명합니다. 이 구성을 통해 IdM 사용자는 Kerberos 티켓을 사용하여 sudo 명령으로 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 용어 목록의 GSSAPI 항목
- 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

5.7. 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 대한 IdM 클라이언트에서 Generic Security Service Application Program Interface(GSSAPI) 인증을 활성화하는 방법을 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

- PAM 서비스의 GSSAPI 인증을 제어하는 SSSD 옵션
- IdM 용어 목록의 GSSAPI 항목
- 스마트 카드 인증을 위한 Identity Management 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

5.8. 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

5.9. 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
```

5.10. 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 사용

자 계정에 부여되도록 할 수 있습니다.

idm_user_reboot 라는 sudo 규칙이 있는지 확인하려면 다음 절차를 완료합니다. 규칙은 idm_user 에 idmclient 시스템에서 /usr/sbin/reboot 명령을 실행할 수 있는 권한을 부여합니다.

사전 요구 사항

- Ansible 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.
- IdM 관리자 암호를 알고 있습니다.
- IdM에 idm_user 의 사용자 계정이 있는지 확인하고 사용자의 암호를 만들어 계정의 잠금을 해제했습니다. 명령줄 인터페이스를 사용하여 새 IdM 사용자를 추가하는 방법에 대한 자세한 내용은 링크를 참조하십시오. 명령줄을 사용하여 사용자 추가.
- 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 클

라이언트에 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 리소스

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

6장. LDAPMODIFY를 사용하여 IDM 사용자 외부 관리

ldapmodify 및 ldap delete 유틸리티를 사용하여 명령줄 인터페이스(CLI)에서 직접 IdM(Identity Management) LDAP를 수정할 수 있습니다. 유틸리티는 디렉터리 내용을 추가, 편집 및 삭제하는 완전한 기능을 제공합니다. 이러한 유틸리티를 사용하여 서버의 구성 항목과 사용자 항목의 데이터를 모두 관리할 수 있습니다. 유틸리티를 사용하여 하나 이상의 디렉터리를 대량으로 관리하는 스크립트를 작성할 수도 있습니다.

6.1. 외부에서 IDM 사용자 계정을 관리하는 템플릿

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

- 새 단계 사용자 추가
- 사용자 속성 수정
- 사용자 활성화
- 사용자 비활성화
- 사용자 보존

템플릿은 LDAP(LDIF)로 포맷됩니다. LDIF는 LDAP 디렉터리 콘텐츠를 표시하고 요청을 업데이트하기 위한 표준 일반 텍스트 데이터 교환 형식입니다.

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

자세한 예제 절차는 다음 섹션을 참조하십시오.

- LDIF 파일에 정의된 IdM 단계 사용자 추가

- ldapmodify를 사용하여 CLI에서 직접 IdM 단계 사용자 추가
- ldapmodify를 사용하여 IdM 사용자 보존

새 스테이징 사용자를 추가하기 위한 템플릿

- UID 및 GID가 자동으로 할당된 사용자를 추가하는 템플릿. 생성된 항목의 고유 이름(DN)은 uid=user_login 으로 시작해야 합니다.

```
dn: uid=user_login,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
changetype: add
objectClass: top
objectClass: inetorgperson
uid: user_login
sn: surname
givenName: first_name
cn: full_name
```

- 정적으로 UID 및 GID가 할당된 사용자를 추가하는 템플릿 :

```
dn: uid=user_login,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: inetorgperson
objectClass: organizationalperson
objectClass: posixaccount
uid: user_login
uidNumber: UID_number
gidNumber: GID_number
sn: surname
givenName: first_name
cn: full_name
homeDirectory: /home/user_login
```

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

기존 사용자 수정을 위한 템플릿

- 사용자의 속성 수정:

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

- 사용자 비활성화:

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

- 사용자 활성화:

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

nssAccountLock 특성을 업데이트해도 단계 및 보존된 사용자에게는 영향을 미치지 않습니다. 업데이트 작업이 성공적으로 완료되었지만 속성 값은 **nssAccountLock**으로 유지됩니다. **TRUE.**

- 사용자 보존:

```
dn: distinguished_name
changetype: modrdn
newrdn: uid=user_login
deleteoldrdn: 0
newsuperior: cn=deleted
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
```

참고

사용자를 수정하기 전에 사용자의 로그인을 통해 검색하여 사용자의 고유 이름(DN)을 획득합니다. 다음 예에서 ***user_allowed_to_modify_user_entries*** 사용자는 활성화 기 또는 IdM 관리자와 같이 사용자 및 그룹 정보를 수정할 수 있는 사용자입니다. 예제의 암호는 이 사용자의 암호입니다.

```
[...]
# ldapsearch -LLL -x -D
"uid=user_allowed_to_modify_user_entries,cn=users,cn=accounts,dc=idm,dc=e
xample,dc=com" -w "Secret123" -H ldap://r8server.idm.example.com -b
"cn=users,cn=accounts,dc=idm,dc=example,dc=com" uid=test_user
dn: uid=test_user,cn=users,cn=accounts,dc=idm,dc=example,dc=com
memberOf: cn=ipausers,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
```

6.2. 외부에서 IDM 그룹 계정을 관리하는 템플릿

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

- 새 그룹 생성
- 기존 그룹 삭제
- 그룹에 구성원 추가
- 그룹에서 멤버 제거

템플릿은 LDAP(LDIF)로 포맷됩니다. LDIF는 LDAP 디렉터리 콘텐츠를 표시하고 요청을 업데이트하기 위한 표준 일반 텍스트 데이터 교환 형식입니다.

템플릿을 사용하여 IdM 그룹 계정을 관리하도록 프로비저닝 시스템의 LDAP 프로바이더를 구성할 수 있습니다.

새 그룹 생성

```
dn: cn=group_name,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
changetype: add
objectClass: top
objectClass: ipaobject
objectClass: ipausergroup
objectClass: groupofnames
objectClass: nestedgroup
objectClass: posixgroup
uid: group_name
cn: group_name
gidNumber: GID_number
```

그룹 수정

- 기존 그룹 삭제:

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

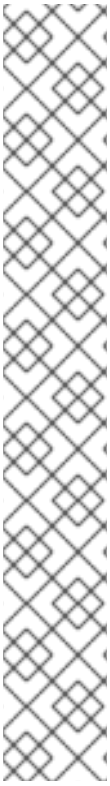
- 그룹에 멤버 추가:

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

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

- 그룹에서 멤버 제거:

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



참고

그룹을 수정하기 전에 그룹 이름을 사용하여 검색하여 그룹의 고유 이름(DN)을 가져옵니다.

```
# ldapsearch -YGSSAPI -H ldap://server.idm.example.com -b
"cn=groups,cn=accounts,dc=idm,dc=example,dc=com" "cn=group_name"
dn: cn=group_name,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
ipaNTSecurityIdentifier: S-1-5-21-1650388524-2605035987-2578146103-11017
cn: testgroup
objectClass: top
objectClass: groupofnames
objectClass: nestedgroup
objectClass: ipausergroup
objectClass: ipaobject
objectClass: posixgroup
objectClass: ipantgroupattrs
ipaUniqueID: 569bf864-9d45-11ea-bea3-525400f6f085
gidNumber: 1997010017
```

6.3. LDAPMODIFY를 사용하여 IDM 사용자 보존

이 섹션에서는 **ldapmodify** 를 사용하여 IdM 사용자를 보존하는 방법에 대해 설명합니다. 즉, 직원이 퇴사한 후 사용자 계정을 비활성화하는 방법을 설명합니다.

사전 요구 사항

- 사용자를 보존하기 위해 역할로 IdM 사용자로 인증할 수 있습니다.

절차

1. 사용자를 보존하려면 역할을 가진 IdM 사용자로 로그인합니다.

```
$ kinit admin
```

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

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

3. 보존할 사용자의 **dn** 을 입력합니다.

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

4. 수행할 변경 유형으로 **modrdn** 을 입력합니다.

```
changetype: modrdn
```

5. 사용자에 대한 **newrdn** 을 지정합니다.

```
newrdn: uid=user1
```

6. 사용자를 보존할 것을 나타냅니다.

```
deleteoldrdn: 0
```

7. 새로운 우수 **DN** 을 지정합니다.

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

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

8. **Enter** 를 다시 눌러 이것이 항목의 끝인지 확인합니다.

```
[Enter]

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

9. **Ctrl + C** 를 사용하여 연결을 종료합니다.

검증 단계

- 보존된 모든 사용자를 나열하여 사용자가 보존되었는지 확인합니다.

```
$ ipa user-find --preserved=true
--------------
1 user matched
--------------
  User login: user1
  First name: First 1
  Last name: Last 1
  Home directory: /home/user1
  Login shell: /bin/sh
  Principal name: user1@IDM.EXAMPLE.COM
  Principal alias: user1@IDM.EXAMPLE.COM
  Email address: user1@idm.example.com
  UID: 1997010003
  GID: 1997010003
  Account disabled: True
  Preserved user: True
----------------------------
Number of entries returned 1
----------------------------
```

7장. 사용자의 외부 프로비저닝을 위한 IDM 구성

시스템 관리자는 ID 관리를 위해 외부 솔루션에서 사용자의 프로비저닝을 지원하도록 IdM(Identity Management)을 구성할 수 있습니다.

ipa 유틸리티를 사용하는 대신 외부 프로비저닝 시스템 관리자는 ldapmodify 유틸리티를 사용하여 IdM LDAP에 액세스할 수 있습니다. 관리자는 ldapmodify 또는 LDIF 파일을 사용하여 CLI에서 개별 단계 사용자를 추가할 수 있습니다.

IdM 관리자는 검증된 사용자만 추가하도록 외부 프로비저닝 시스템을 완전히 신뢰한다고 가정합니다. 그러나 동시에 외부 프로비저닝 시스템의 관리자에게 사용자 관리자의 IdM 역할을 할당하여 새 활성 사용자를 직접 추가할 수 없습니다.

외부 프로비저닝 시스템에서 생성한 스테이징된 사용자를 활성 사용자로 자동으로 이동하도록 스크립트를 구성할 수 있습니다.

이 장에는 다음 섹션이 포함되어 있습니다.

1. 외부 프로비저닝 시스템을 사용하여 IdM(Identity Management) 을 준비하여 IdM에 단계 사용자를 추가합니다.
2. 외부 프로비저닝 시스템에서 추가한 사용자를 스테이지에서 활성 사용자로 이동하는 스크립트를 생성합니다.
3. 외부 프로비저닝 시스템을 사용하여 IdM 단계 사용자 추가. 다음 두 가지 방법으로 이를 수행할 수 있습니다.
   - LDIF 파일을 사용하여 IdM 스테이징 사용자 추가
   - ldapmodify를 사용하여 CLI에서 직접 IdM stage 사용자 추가

7.1. 단계 사용자 계정의 자동 활성화를 위한 IDM 계정 준비

다음 절차에서는 외부 프로비저닝 시스템에서 사용할 두 개의 IdM 사용자 계정을 구성하는 방법을 설명합니다. 적절한 암호 정책이 있는 그룹에 계정을 추가하면 외부 프로비저닝 시스템에서 IdM의 사용자

프로비저닝을 관리할 수 있습니다. 다음에서는 외부 시스템에서 단계 사용자를 추가하는 데 사용하는 사용자 계정은 **provisionator** 라고 합니다. 스테이징 사용자를 자동으로 활성화하는 데 사용할 사용자 계정은 활성화 기라고 합니다.

사전 요구 사항

- 절차를 수행하는 호스트는 **IdM**에 등록됩니다.

절차

1. **IdM** 관리자로 로그인합니다.

```
$ kinit admin
```

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

a. **provisionator** 사용자 계정을 추가합니다.

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

a. 프로비저너 사용자에게 필요한 권한을 부여합니다.

i. 사용자 지정 역할인 **System Provisioning** 을 생성하여 스테이징 사용자 추가를 관리합니다.

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

ii. 역할에 **Stage User Provisioning**(사용자 프로비저닝 단계) 권한을 추가합니다. 이 권한은 스테이징 사용자를 추가할 수 있는 기능을 제공합니다.

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

iii. **provisionator** 사용자를 역할에 추가합니다.

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

iv. 프로비저너가 IdM에 있는지 확인합니다.

```
$ ipa user-find provisionator --all --raw
--------------
1 user matched
--------------
  dn: uid=provisionator,cn=users,cn=accounts,dc=idm,dc=example,dc=com
  uid: provisionator
  [...]
```

3. 사용자 계정을 관리할 수 있는 권한을 가진 사용자 활성화 기를 만듭니다.

a. 활성화기 사용자 계정을 추가합니다.

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

b. 사용자를 기본 User Administrator 역할에 추가하여 필요한 권한을 활성화 사용자에게 부여합니다.

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

4. 애플리케이션 계정의 사용자 그룹을 생성합니다.

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

5. 그룹의 암호 정책을 업데이트합니다. 다음 정책은 계정의 암호 만료 및 잠금을 방지하지만 복잡한 암호를 요구하여 잠재적인 위험을 보상합니다.

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

6. (선택 사항) 암호 정책이 IdM에 있는지 확인합니다.

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

7. 애플리케이션 계정의 그룹에 프로비저닝 및 활성화 계정을 추가합니다.

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

8. 사용자 계정의 암호를 변경합니다.

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

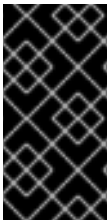
새 IdM 사용자 암호가 즉시 만료되기 때문에 암호를 변경해야 합니다.

추가 리소스:

- 명령줄을 사용하여 사용자 계정 관리를 참조하십시오.
- 사용자에 대한 권한 위임을 참조하십시오.
- IdM 암호 정책 정의를 참조하십시오.

7.2. IDM 스테이징 사용자 계정의 자동 활성화 구성

다음 절차에서는 스테이징 사용자를 활성화하는 스크립트를 생성하는 방법을 설명합니다. 시스템은 지정된 시간 간격에 따라 자동으로 스크립트를 실행합니다. 이렇게 하면 새 사용자 계정이 자동으로 활성화되고 생성된 직후 사용할 수 있습니다.



중요

이 절차에서는 외부 프로비저닝 시스템의 소유자가 이미 사용자를 검증했으며 스크립트에서 IdM에 추가 검증이 필요하지 않다고 가정합니다.

IdM 서버 중 하나에서만 활성화 프로세스를 활성화할 수 있습니다.

사전 요구 사항

- 프로비저너 및 활성 기 계정은 IdM에 있습니다. 자세한 내용은 단계 사용자 계정 자동 활성화

를 위한 IdM 계정 준비를 참조하십시오.

- 프로시저를 실행 중인 IdM 서버에 대한 root 권한이 있습니다.
- IdM 관리자로 로그인했습니다.
- 외부 프로비저닝 시스템을 신뢰합니다.

절차

1. 활성화 계정에 대한 keytab 파일을 생성합니다.

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

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

2. 다음 내용으로 /usr/local/sbin/ipa-activate-all 스크립트를 생성하여 모든 사용자를 활성화합니다.

```
#!/bin/bash

kinit -k -i activator

ipa stageuser-find --all --raw | grep "  uid:" | cut -d ":" -f 2 | while read uid; do ipa stageuser-activate ${uid}; done
```

3. ipa-activate-all 스크립트의 권한 및 소유권을 편집하여 실행 가능하게 만듭니다.

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

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

```
[Unit]
Description=Scan IdM every minute for any stage users that must be activated
```

```
[Service]
Environment=KRB5_CLIENT_KTNAME=/etc/krb5.ipa-activation.keytab
Environment=KRB5CCNAME=FILE:/tmp/krb5cc_ipa-activate-all
ExecStart=/usr/local/sbin/ipa-activate-all
```

5. 다음 내용을 사용하여 **systemd** 타이머 **/etc/systemd/system/ipa-activate-all.timer** 을 만듭니다.

```
[Unit]
Description=Scan IdM every minute for any stage users that must be activated

[Timer]
OnBootSec=15min
OnUnitActiveSec=1min

[Install]
WantedBy=multi-user.target
```

6. 새 구성을 다시 로드합니다.

```
# systemctl daemon-reload
```

7. **ipa-activate-all.timer:**

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

8. **ipa-activate-all.timer** 시작 **:**

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

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

```
# systemctl status ipa-activate-all.timer
● ipa-activate-all.timer - Scan IdM every minute for any stage users that must be
activated
   Loaded: loaded (/etc/systemd/system/ipa-activate-all.timer; enabled; vendor preset:
disabled)
   Active: active (waiting) since Wed 2020-06-10 16:34:55 CEST; 15s ago
  Trigger: Wed 2020-06-10 16:35:55 CEST; 44s left

Jun 10 16:34:55 server.idm.example.com systemd[1]: Started Scan IdM every minute
for any stage users that must be activated.
```

7.3. LDIF 파일에 정의된 IDM 단계 사용자 추가

이 섹션에서는 외부 프로비저닝 시스템의 관리자가 IdM LDAP에 액세스하고 LDIF 파일을 사용하여 스테이징 사용자를 추가하는 방법에 대해 설명합니다. 아래 예제에서는 단일 사용자를 추가하는 방법을 보여주지만, 일괄 모드로 하나의 파일에 여러 사용자를 추가할 수 있습니다.

사전 요구 사항

- IdM 관리자가 프로비저너 계정 과 암호를 생성했습니다. 자세한 내용은 단계 사용자 계정 자동 활성화를 위한 IdM 계정 준비를 참조하십시오.
- 외부 관리자가 프로비저너 계정의 암호를 알고 있습니다.
- LDAP 서버에서 IdM 서버에 SSH로 연결할 수 있습니다.
- IdM 단계 사용자가 사용자 라이프사이클의 올바른 처리를 허용해야 하는 최소한의 속성 세트를 제공할 수 있습니다.
  - 고유 이름 (dn)
  - 일반 이름 (cn)
  - 성 (sn)
  - uid

절차

1. 외부 서버에서 새 사용자에 대한 정보가 포함된 LDIF 파일을 생성합니다.

```
dn: uid=stageidmuser,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
changetype: add
objectClass: top
objectClass: inetorgperson
uid: stageidmuser
```

```
sn: surname
givenName: first_name
cn: full_name
```

2. LDIF 파일을 외부 서버에서 IdM 서버로 전송합니다.

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

3. SSH 프로토콜을 사용하여 IdM 서버에 프로비저너로 연결합니다.

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

4. IdM 서버에서 프로비저너 계정의 Kerberos 티켓을 받습니다.

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

5. f 옵션과 LDIF 파일의 이름을 사용하여 ldapadd 명령을 입력합니다. IdM 서버의 이름과 포트 번호를 지정합니다.

```
~]$ ldapadd -h server.idm.example.com -p 389 -f  add-stageidmuser.ldif
SASL/GSSAPI authentication started
SASL username: provisionator@IDM.EXAMPLE.COM
SASL SSF: 256
SASL data security layer installed.
adding the entry "uid=stageidmuser,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com"
```

7.4. LDAPMODIFY를 사용하여 CLI에서 직접 IDM 단계 사용자 추가

이 섹션에서는 외부 프로비저닝 시스템의 관리자가 IdM(Identity Management) LDAP에 액세스하고 ldapmodify 유틸리티를 사용하여 stage 사용자를 추가하는 방법에 대해 설명합니다.

사전 요구 사항

- IdM 관리자가 프로비저너 계정 과 암호를 생성했습니다. 자세한 내용은 단계 사용자 계정 자동 활성화를 위한 IdM 계정 준비를 참조하십시오.

- 외부 관리자가 프로비저너 계정의 암호를 알고 있습니다.
- LDAP 서버에서 IdM 서버에 SSH로 연결할 수 있습니다.
- IdM 단계 사용자가 사용자 라이프사이클의 올바른 처리를 허용해야 하는 최소한의 속성 세트를 제공할 수 있습니다.
  - 고유 이름 (dn)
  - 일반 이름 (cn)
  - 성 (sn)
  - uid

절차

1. IdM ID 및 자격 증명을 사용하여 SSH 프로토콜을 사용하여 IdM 서버에 연결합니다.

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

2. 새 단계 사용자를 추가하는 역할의 IdM 사용자인 프로비저너 계정의 TGT를 가져옵니다.

```
$ kinit provisionator
```

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

```
# ldapmodify -h server.idm.example.com -p 389 -Y GSSAPI
SASL/GSSAPI authentication started
SASL username: provisionator@IDM.EXAMPLE.COM
```

```
SASL SSF: 56
SASL data security layer installed.
```

4. 추가하려는 사용자의 **dn** 을 입력합니다.

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

5. 수행 중인 변경 유형으로 **add** 를 입력합니다.

```
changetype: add
```

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

```
objectClass: top
objectClass: inetorgperson
```

추가 오브젝트 클래스를 지정할 수 있습니다.

7. 사용자의 **uid** 를 입력합니다.

```
uid: stageuser
```

8. 사용자의 **cn** 을 입력합니다.

```
cn: Babs Jensen
```

9. 사용자의 성을 입력합니다.

```
sn: Jensen
```

10. **Enter** 를 다시 눌러 이것이 항목의 끝인지 확인합니다.

```
[Enter]

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

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

11. Ctrl + C 를 사용하여 연결을 종료합니다.

검증 단계

stage 항목의 콘텐츠를 확인하여 프로비저닝 시스템에서 필요한 모든 POSIX 특성을 추가하고 stage 항목을 활성화할 준비가 되었는지 확인합니다.

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

```
$ ipa stageuser-show stageuser --all --raw
 dn: uid=stageuser,cn=staged
users,cn=accounts,cn=provisioning,dc=idm,dc=example,dc=com
 uid: stageuser
 sn: Jensen
 cn: Babs Jensen
 has_password: FALSE
 has_keytab: FALSE
 nsaccountlock: TRUE
 objectClass: top
 objectClass: inetorgperson
 objectClass: organizationalPerson
 objectClass: person
```

1. 사용자는 nsaccountlock 특성에 의해 명시적으로 비활성화됩니다.

7.5. 추가 리소스

- 외부에서 IdM 사용자를 관리하기 위해 ldapmodify를 사용하여 를 참조하십시오.

8장. CLI를 사용하여 IDM에서 셀프 서비스 규칙 관리

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

8.1. 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.

8.2. CLI를 사용하여 셀프 서비스 규칙 생성

이 절차에서는 CLI(명령줄 인터페이스)를 사용하여 IdM에서 셀프 서비스 액세스 규칙을 생성하는 방법을 설명합니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

- 셀프 서비스 규칙을 추가하려면 ipa selfservice-add 명령을 사용하고 다음 두 가지 옵션을 지

정합니다.

--permissions

읽기 및 쓰기 권한을 설정합니다. ACI(액세스 제어 명령)가 부여됩니다.

--attrs

이 ACI에서 권한을 부여하는 전체 속성 목록을 설정합니다.

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

```
$ ipa selfservice-add "Users can manage their own name details" --permissions=write --
attrs=givenname --attrs=displayname --attrs=title --attrs=initials
--------------------------------------------------------
Added selfservice "Users can manage their own name details"
--------------------------------------------------------
  Self-service name: Users can manage their own name details
  Permissions: write
  Attributes: givenname, displayname, title, initials
```

8.3. CLI를 사용하여 셀프 서비스 규칙 편집

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

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

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

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

예를 들면 다음과 같습니다.

```
$ ipa selfservice-mod "Users can manage their own name details" --attrs=givenname --
attrs=displayname --attrs=title --attrs=initials --attrs=surname
-------------------------------------------------------------
Modified selfservice "Users can manage their own name details"
-------------------------------------------------------------
Self-service name: Users can manage their own name details
Permissions: write
Attributes: givenname, displayname, title, initials
```

중요

ipa selfservice-mod 명령을 사용하면 이전에 정의한 권한 및 속성을 덮어쓰므로 항상 기존 권한 및 속성의 전체 목록과 정의하고자 하는 새 권한 목록을 포함합니다.

검증 단계

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

```
$ ipa selfservice-show "Users can manage their own name details"
-------------------------------------------------------------
Self-service name: Users can manage their own name details
Permissions: write
Attributes: givenname, displayname, title, initials
```

8.4. CLI를 사용하여 셀프 서비스 규칙 삭제

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

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

-

활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

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

예를 들면 다음과 같습니다.

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

검증 단계

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

9장. IDM 웹 UI를 사용하여 셀프 서비스 규칙 관리

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

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. IDM 웹 UI를 사용하여 셀프 서비스 규칙 생성

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

사전 요구 사항

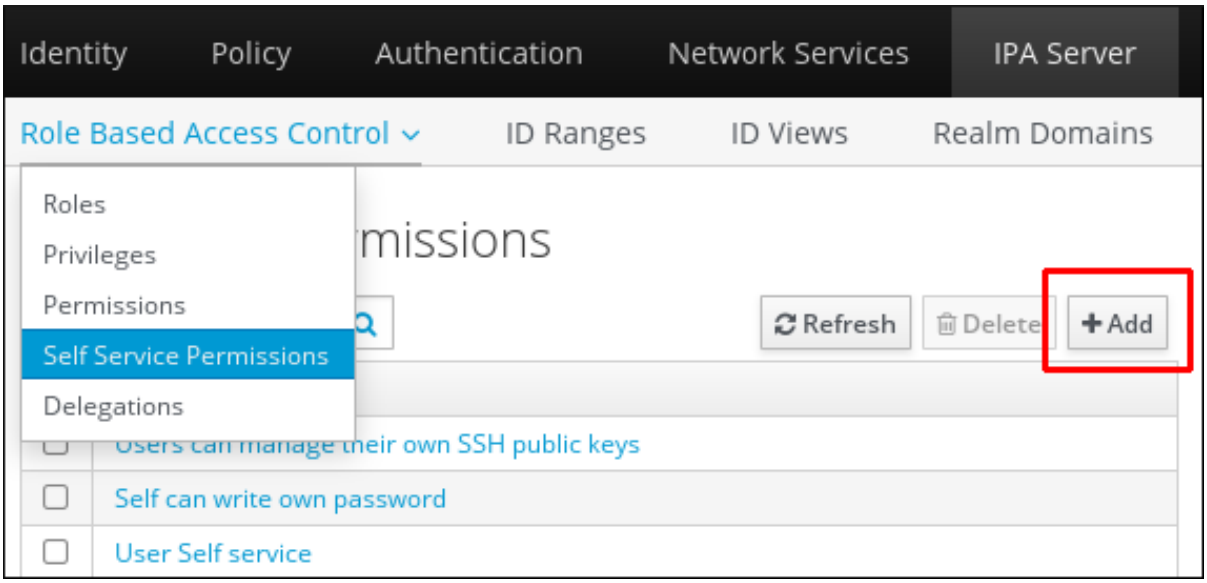
- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

절차

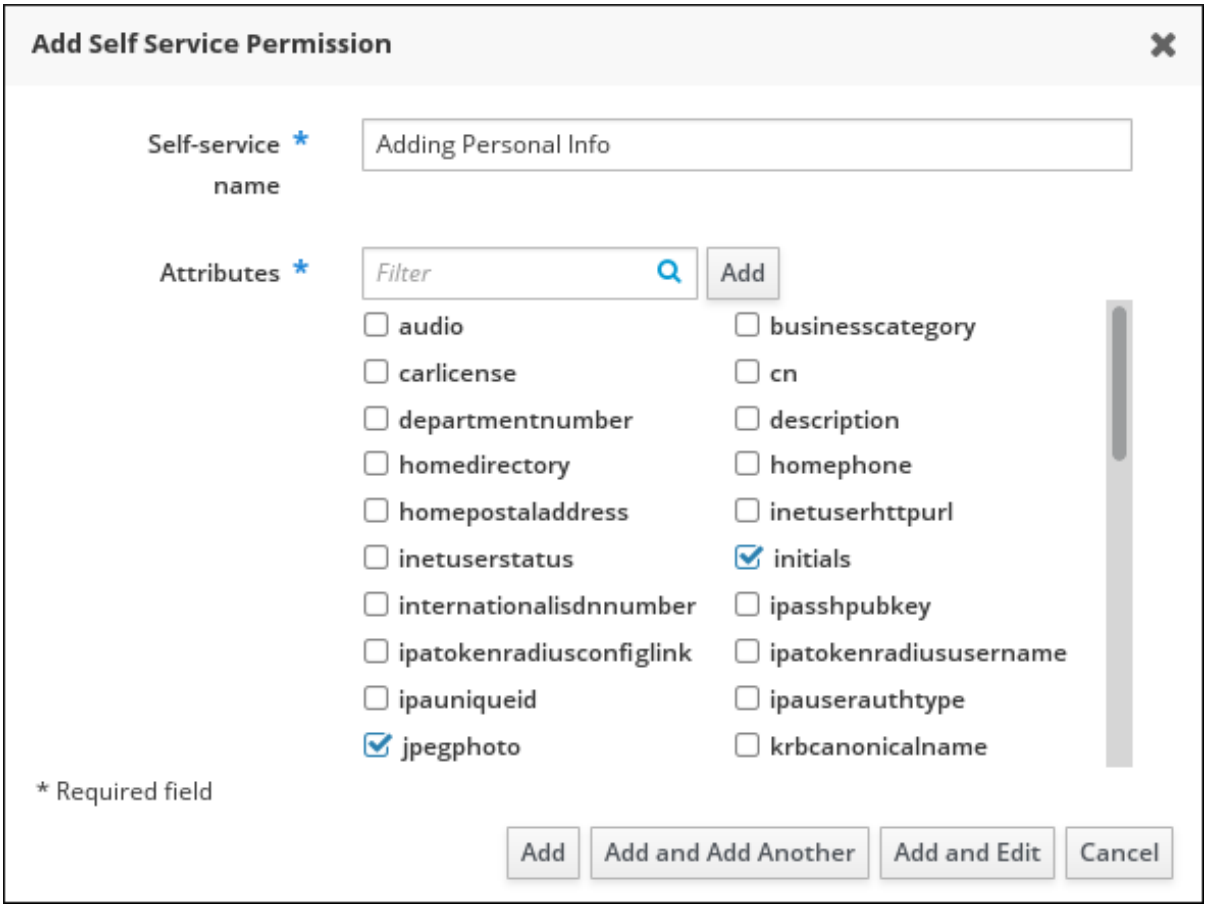
1. IPA Server 탭에서 Role-Based Access Control (역할 기반 액세스 제어) 하위 메뉴를 열고

셀프 서비스 권한을 선택합니다.

2. 셀프 서비스 액세스 규칙 목록의 오른쪽 상단에 있는 **Add** (추가)를 클릭합니다.



3. **Add self Service Permission**(셀프 서비스 권한 추가) 창이 열립니다. 셀프 서비스 이름 필드에 새 셀프 서비스 규칙의 이름을 입력합니다. 공백을 사용할 수 있습니다.



4. 사용자가 편집할 속성 옆에 있는 확인란을 선택합니다.

5. *선택 사항*: 액세스를 제공할 속성이 나열되지 않은 경우 목록을 추가할 수 있습니다.

   a. **Add**(추가) 버튼을 클릭합니다.

   b. 다음 **Add Custom Attribute**(사용자 지정 속성 추가) 창의 **Attribute** (특성) 텍스트 필드에 특성 이름을 입력합니다.

   c. **OK**(확인) 버튼을 클릭하여 속성을 추가합니다.

   d. 새 속성이 선택되었는지 확인합니다.

6. 양식 하단에서 **Add**(추가) 버튼을 클릭하여 새 셀프 서비스 규칙을 저장합니다.
또는 **Add and Edit** (추가 및 편집) 버튼을 클릭하여 셀프 서비스 규칙을 저장하고 계속 편집하거

나 추가 및 추가 버튼을 클릭하여 추가 규칙을 저장하고 추가할 수 있습니다.

9.3. IDM 웹 UI를 사용하여 셀프 서비스 규칙 편집

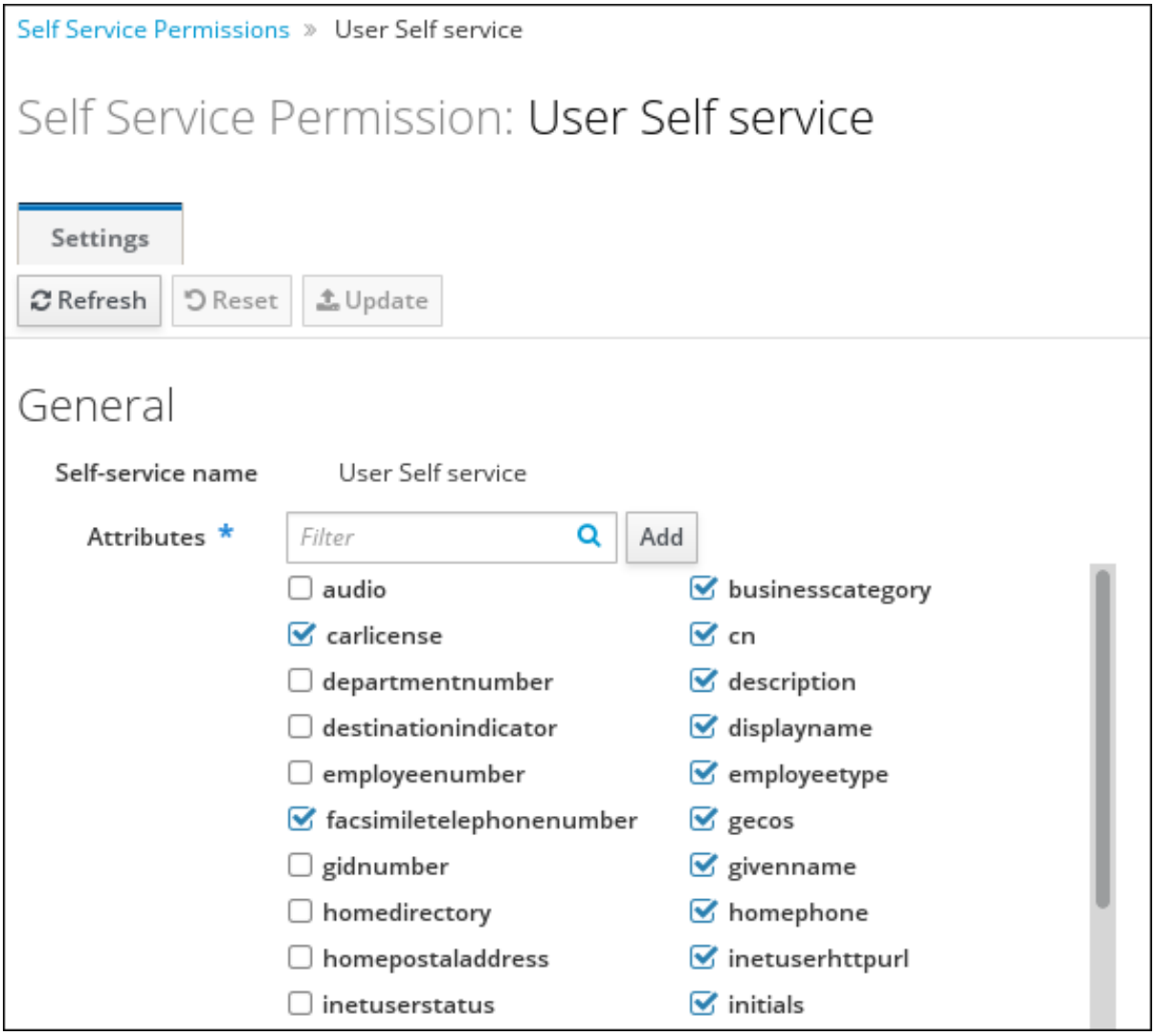
다음 절차에서는 웹 인터페이스(IdM 웹 UI)를 사용하여 IdM에서 셀프 서비스 액세스 규칙을 편집하는 방법을 설명합니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

절차

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



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

4. **Save**(저장) 버튼을 클릭하여 셀프 서비스 규칙에 변경 사항을 저장합니다.

9.4. IDM 웹 UI를 사용하여 셀프 서비스 규칙 삭제

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

사전 요구 사항

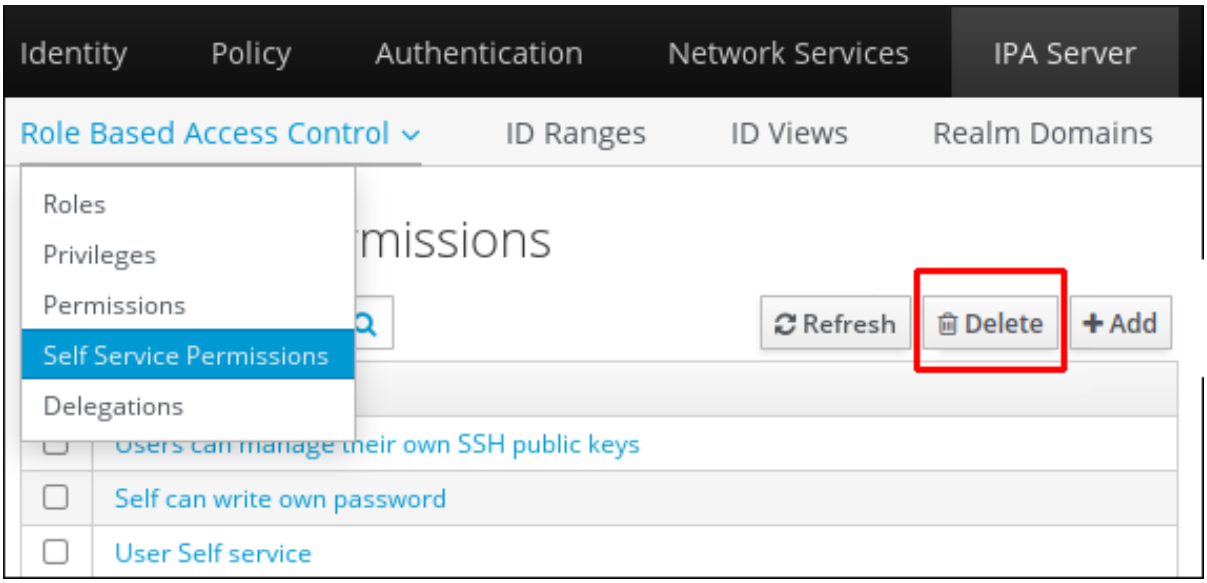
- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

절차

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

2. 삭제할 규칙 옆에 있는 확인란을 선택한 다음 목록 오른쪽에 있는 Delete (삭제) 버튼을 클릭합니다.



3. 대화 상자가 열리고 Delete(삭제)를 클릭하여 확인합니다.

10장. 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.

10.1. 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.

10.2. ANSIBLE을 사용하여 셀프 서비스 규칙이 있는지 확인합니다.

다음 절차에서는 **Ansible** 플레이북을 사용하여 셀프 서비스 규칙을 정의하고 **IdM(Identity Management)** 서버에 있는지 확인하는 방법을 설명합니다. 이 예에서 새 사용자는 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** 컨트롤러에 **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

- IdM의 셀프 서비스 액세스 제어를 참조하십시오.
- **/usr/share/doc/ansible-freeipa/** 디렉터리에서 **README-selfservice.md** 파일을 참조하십시오.
- **/usr/share/doc/ansible-freeipa/playbooks/selfservice** 디렉토리를 참조하십시오.

10.3. ANSIBLE을 사용하여 셀프 서비스 규칙이 없는지 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** 컨트롤러에 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

- IdM의 셀프 서비스 액세스 제어를 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉터리에서 **README-selfservice.md** 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/selfservice 디렉터리에서 샘플 플레이북을 참조하십시오.

10.4. ANSIBLE을 사용하여 셀프 서비스 규칙에 특정 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** 컨트롤러에 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

- IdM의 셀프 서비스 액세스 제어를 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉토리에서 사용 가능한 README-selfservice.md 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/selfservice 디렉터리에서 샘플 플레이북을 참조하십시오.

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

다음 절차에서는 Ansible 플레이북을 사용하여 셀프 서비스 규칙에 특정 설정이 없는지 확인하는 방법을 설명합니다. 이 플레이북을 사용하여 셀프 서비스 규칙이 바람직하지 않은 액세스 권한을 부여하지 않도록 할 수 있습니다. 이 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 컨트롤러에 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

- IdM의 셀프 서비스 액세스 제어를 참조하십시오.

- /usr/share/doc/ansible-freeipa/ 디렉터리에서 README-selfservice.md 파일을 참조하십시오.

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

11장. IDM CLI에서 사용자 그룹 관리

이 장에서는 IdM CLI를 사용한 사용자 그룹 관리 방법을 소개합니다.

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

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

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

11.1. 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 11.1. 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 그룹이 없는 사용자 추가를 참조하십시오.

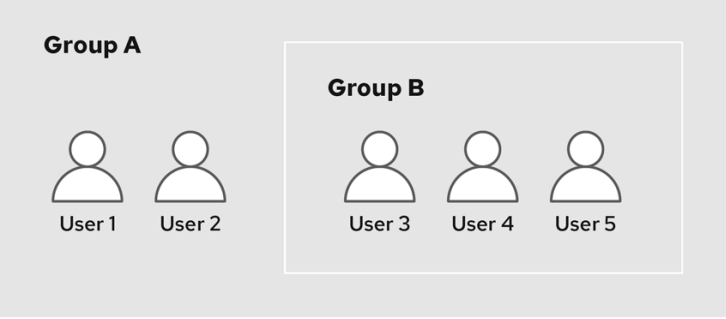
11.2. 직접 및 간접 그룹 구성원

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

예를 들어 다음 다이어그램에서 다음을 수행합니다.

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

그림 11.1. 직접 및 간접 그룹 멤버십



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

11.3. IDM CLI를 사용하여 사용자 그룹 추가

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

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

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

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

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

    그룹 유형에 대한 자세한 내용은 IdM의 다양한 그룹 유형을 참조하십시오.

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

주의

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

11.4. IDM CLI를 사용하여 사용자 그룹 검색

이 섹션에서는 IdM CLI를 사용하여 기존 사용자 그룹을 검색하는 방법을 설명합니다.

절차

- **ipa group-find** 명령을 사용하여 모든 사용자 그룹을 표시합니다. 그룹 유형을 지정하려면 **ipa group-find** 에 옵션을 추가합니다.

  - **ipa group-find --posix** 명령을 사용하여 모든 **POSIX** 그룹을 표시합니다.

  - **ipa group-find --nonposix** 명령을 사용하여 모든 비**POSIX** 그룹을 표시합니다.

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

    다양한 그룹 유형에 대한 자세한 내용은 IdM의 다양한 그룹 유형에서 참조하십시오.

11.5. IDM CLI를 사용하여 사용자 그룹 삭제

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

- **ipa group-del group _*name* 명령을 사용하여 사용자 그룹을** 삭제합니다. 예를 들어 **group_a**를 삭제하려면 다음을 수행합니다.

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

11.6. IDM CLI를 사용하여 사용자 그룹에 멤버 추가

이 섹션에서는 IdM CLI를 사용하여 사용자 그룹에 멤버를 추가하는 방법을 설명합니다. 사용자 및 사용자 그룹을 사용자 그룹의 멤버로 추가할 수 있습니다. 자세한 내용은 IdM 및 직접 및 간접 그룹 구성원의 다양한 그룹 유형을 참조하십시오.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

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

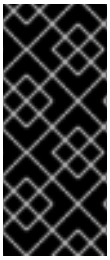
  다음 옵션을 사용하여 멤버 유형을 지정합니다.

  - --users 는 IdM 사용자 추가
  - --external 은 IdM 도메인 외부에 있는 사용자 추가: sudoAIN \user_name 또는 user_name@domain형식으로
  - --groups 는 IdM 사용자 그룹 추가

  예를 들어 group_b를 group_a의 멤버로 추가하려면 다음을 수행합니다.

```
$ ipa group-add-member group_a --groups=group_b
Group name: group_a
GID: 1133400009
Member users: user_a
Member groups: group_b
Indirect Member users: user_b
-------------------------
Number of members added 1
-------------------------
```

  group_b의 멤버는 이제 group_a의 간접 멤버입니다.



중요

그룹을 다른 그룹의 구성원으로 추가할 때 재귀 그룹을 만들지 마십시오. 예를 들어 그룹 A가 그룹 B의 구성원인 경우 그룹 A의 구성원으로 그룹 B를 추가하지 마십시오. 반복적인 그룹은 예측할 수 없는 동작이 발생할 수 있습니다.

참고

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

11.7. 사용자 개인 그룹없이 사용자 추가

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

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

그러나 **UPG**를 생성하지 않고 사용자를 추가할 수 있습니다.

11.7.1. 사용자 개인 그룹이 없는 사용자

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

다음 두 가지 방법으로 이 작업을 수행할 수 있습니다.

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

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

IdM에는 새 사용자에 대한 GID가 필요하지만 기본 사용자 그룹 ipausers 는 POSTIX 그룹이 아니므로 GID가 연결되어 있지 않기 때문입니다. 지정한 GID는 기존 그룹에 해당하지 않아도 됩니다.

참고

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

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

UPG(사용자 개인 그룹)를 시스템에서 활성화된 경우에도 사용자를 만들 수 있습니다. 이를 위해서는 새 사용자에 대한 GID를 수동으로 설정해야 합니다. 필요한 이유에 대한 자세한 내용은 사용자 개인 그룹이 없는 사용자를 참조하십시오.

절차

- IdM이 UPG를 생성하지 못하도록 하려면 ipa user-add 명령에 --noprivate 옵션을 추가합니다.

  명령이 성공하려면 사용자 지정 GID를 지정해야 합니다. 예를 들어 GID가 10000인 새 사용자를 추가하려면 다음을 수행합니다.

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

11.7.3. 모든 사용자에 대해 사용자 개인 그룹 전역 비활성화

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

절차

1. 관리자 권한을 얻습니다.

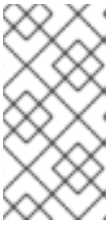
```
$ kinit admin
```

2. IdM은 Directory Server Managed Entries 플러그인을 사용하여 UPG를 관리합니다. 플러그인 인스턴스를 나열합니다.

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

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

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



참고

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

4. Directory Server를 다시 시작하여 새 구성을 로드합니다.

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

UPG를 비활성화한 다음 사용자를 추가하려면 GID를 지정해야 합니다. 자세한 내용은 사용자 개인 그룹이 전역적으로 비활성화된 경우 사용자 추가를 참조하십시오.

검증 단계

- UPG가 전역적으로 비활성화되어 있는지 확인하려면 disable 명령을 다시 사용합니다.

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

11.7.4. 사용자 개인 그룹이 전역적으로 비활성화된 경우 사용자 추가

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

전제 조건

- 모든 사용자에 대해 UPG를 전역적으로 비활성화해야 합니다. 자세한 내용은 모든 사용자에 대해 사용자 개인 그룹 전역 비활성화를 참조하십시오.

절차

-

UPG를 생성할 때 새 사용자를 추가하는 것이 성공했는지 확인하려면 다음 중 하나를 선택하십시오.

- 새 사용자를 추가할 때 사용자 지정 GID를 지정합니다. GID는 기존 사용자 그룹에 해당하지 않아도 됩니다.

  예를 들어 명령줄에서 사용자를 추가하는 경우 ipa user-add 명령에 --gid 옵션을 추가합니다.

- automember 규칙을 사용하여 GID가 있는 기존 그룹에 사용자를 추가합니다.

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

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

- 멤버 관리자로 추가하려는 사용자 또는 그룹의 이름과 관리하려는 그룹의 이름이 있어야 합니다.

절차

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

  예를 들어 사용자 test 를 group_a 의 멤버 관리자로 추가하려면 다음을 수행합니다.

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

사용자 테스트는 이제 **group_a** 의 멤버를 관리할 수 있습니다.

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

예를 들어 **group _admins** 그룹을 **group_ a** 의 멤버 관리자로 추가하려면 :

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

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

참고

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

검증 단계

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

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

추가 리소스

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

11.9. IDM CLI를 사용하여 그룹 멤버 보기

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

원을 모두 볼 수 있습니다. 자세한 내용은 직접 및 간접 그룹 구성원을 참조하십시오.

절차:

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

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

참고

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

11.10. IDM CLI를 사용하여 사용자 그룹에서 멤버 제거

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 **kinit**를 사용하여 **IdM**에 수동으로 로그인하는 방법을 참조하십시오.

절차

1. *선택 사항:* **ipa group-show** 명령을 사용하여 그룹에 제거할 멤버가 포함되어 있는지 확인합니다.

2. **ipa group-remove-member** 명령을 사용하여 사용자 그룹에서 멤버를 제거합니다.

   다음 옵션을 사용하여 제거할 멤버를 지정합니다.

- **--users** 는 IdM 사용자를 제거합니다

- **--external** 은 pamAIN \user_name 또는 user_name@domain 형식으로 IdM 도메인 외부에 있는 사용자를 제거합니다.

- **--groups** 는 IdM 사용자 그룹을 제거합니다

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

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

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

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

- 제거 중인 기존 멤버 관리자 사용자 또는 그룹의 이름과 관리 중인 그룹의 이름이 있어야 합니다.

절차

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

예를 들어 user test 를 group_a 의 멤버 관리자로 제거하려면 다음을 수행합니다.

```
$ ipa group-remove-member-manager group_a --users=test
Group name: group_a
GID: 1133400009
```

```
Membership managed by groups: group_admins
-------------------------
Number of members removed 1
-------------------------
```

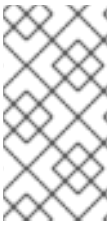
사용자 테스트는 더 이상 **group_a** 의 구성원을 관리할 수 없습니다.

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

예를 들어 **group _admins** 그룹을 **group_ a** 의 멤버 관리자로 제거하려면 다음을 수행합니다.

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

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



참고

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

검증 단계

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

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

추가 리소스

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

12장. IDM 웹 UI에서 사용자 그룹 관리

이 장에서는 IdM 웹 UI를 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

12.1. 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 12.1. 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 그룹이 없는 사용자 추가를 참조하십시오.

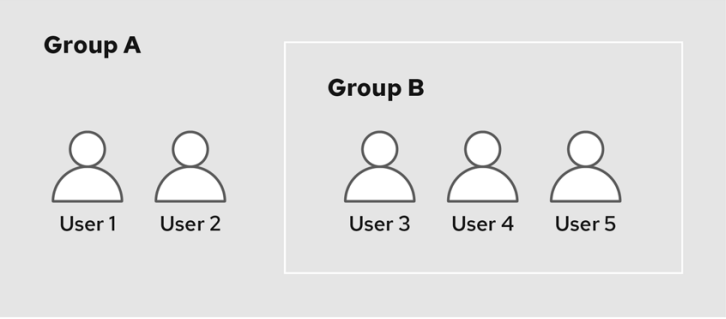
12.2. 직접 및 간접 그룹 구성원

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

예를 들어 다음 다이어그램에서 다음을 수행합니다.

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

그림 12.1. 직접 및 간접 그룹 멤버십



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

12.3. IDM 웹 UI를 사용하여 사용자 그룹 추가

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

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.

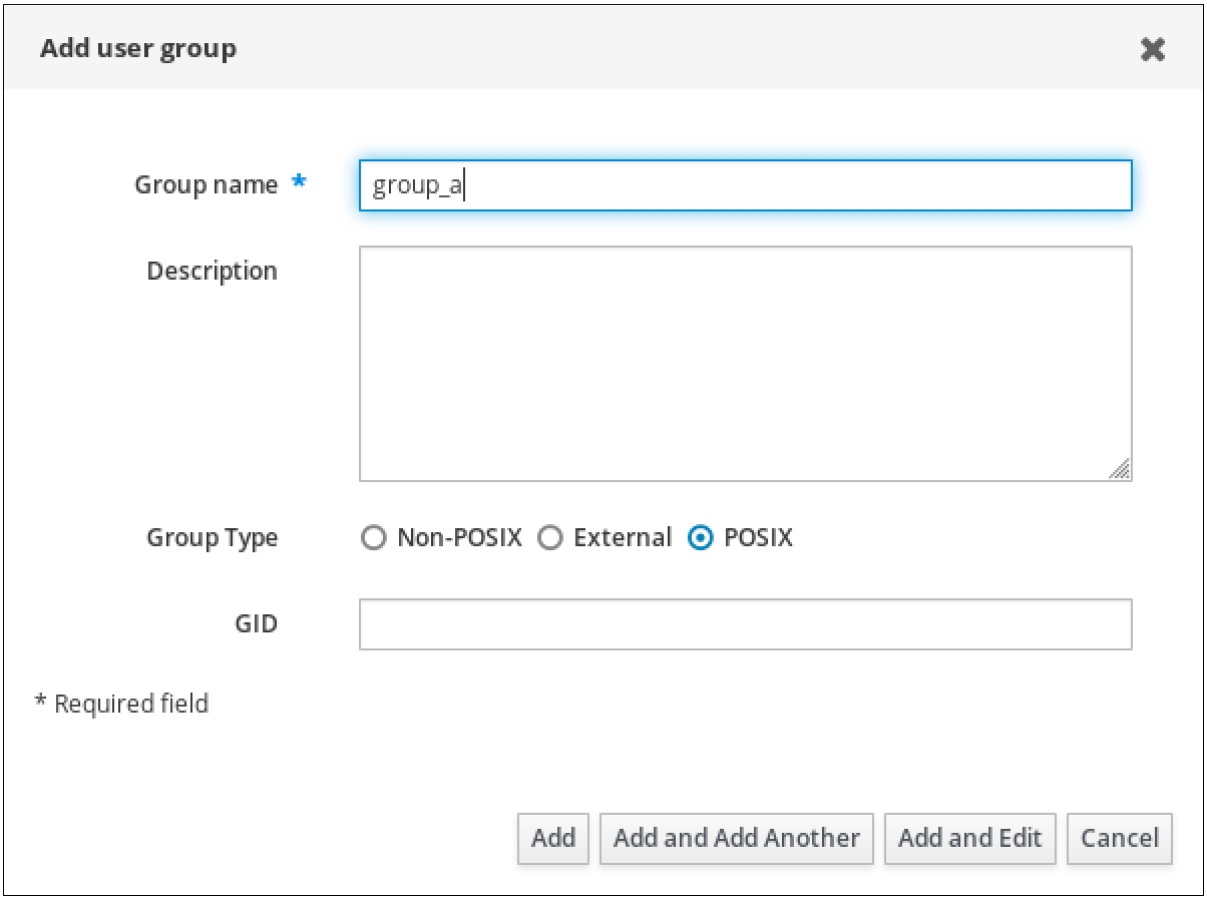
절차

1. **ID → 그룹**을 클릭하고 왼쪽 사이드바 에서 사용자 그룹을 선택합니다.

2. **Add**(추가) 를 클릭하여 그룹 추가를 시작합니다.

3. 그룹에 대한 정보를 입력합니다. 사용자 그룹 유형에 대한 자세한 내용은 IdM의 다양한 그룹 유형에서 참조하십시오.

   그룹에 대한 사용자 지정 GID를 지정할 수 있습니다. 이 작업을 수행하는 경우 ID 충돌을 피하십시오. 사용자 지정 GID를 지정하지 않으면 IdM에서 사용 가능한 ID 범위에서 GID를 자동으로 할당합니다.



4. **Add**(추가) 를 클릭하여 확인합니다.

12.4. IDM 웹 UI를 사용하여 사용자 그룹 삭제

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

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.

절차

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

12.5. IDM 웹 UI를 사용하여 사용자 그룹에 멤버 추가

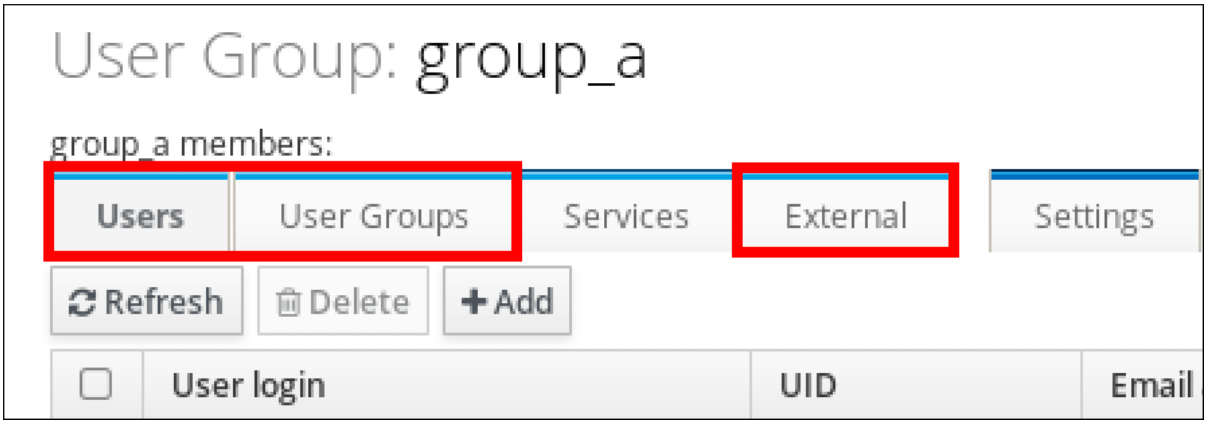
사용자 및 사용자 그룹을 사용자 그룹의 멤버로 추가할 수 있습니다. 자세한 내용은 IdM 및 직접 및 간접 그룹 구성원의 다양한 그룹 유형을 참조하십시오.

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.

절차

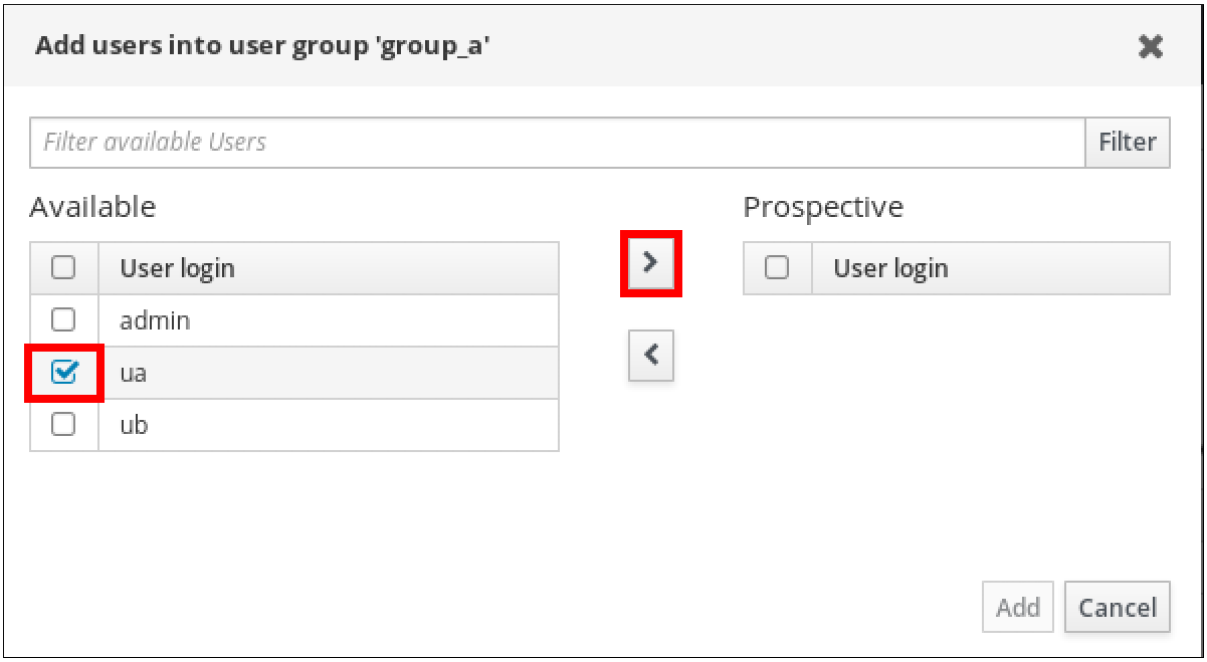
1. ID → 그룹을 클릭하고 왼쪽 사이드바 에서 사용자 그룹을 선택합니다.
2. 그룹의 이름을 클릭합니다.
3. 추가할 그룹 구성원 유형을 선택합니다. 사용자, 사용자 그룹 또는 외부.



4. 추가를 클릭합니다.

5. 추가할 구성원 하나 이상 옆에 있는 확인란을 선택합니다.

6. 오른쪽 화살표를 클릭하여 선택한 구성원을 그룹으로 이동합니다.



7. **Add(추가)** 를 클릭하여 확인합니다.

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

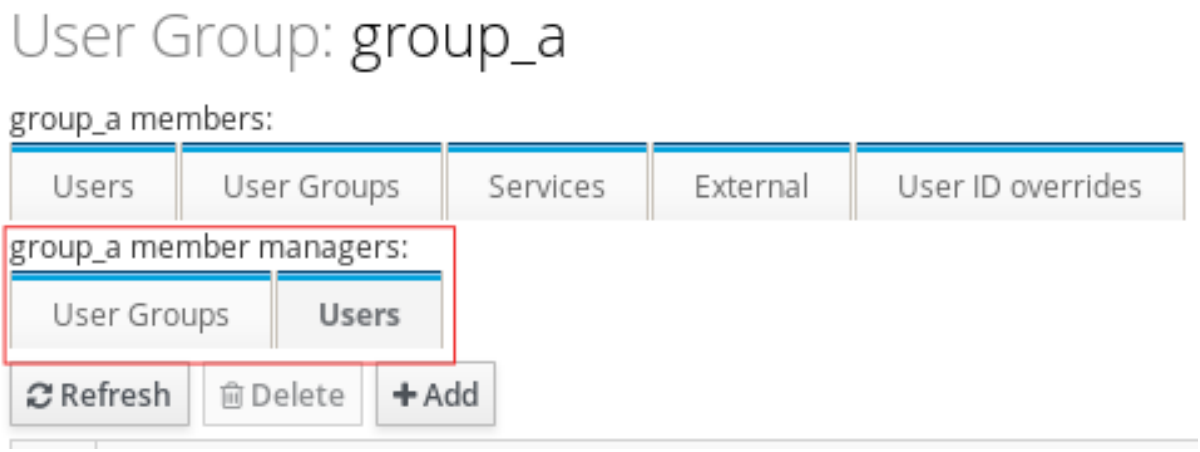
이 섹션에서는 웹 UI를 사용하여 IdM 사용자 그룹에 구성원 관리자로 사용자 또는 그룹을 추가하는 방법에 대해 설명합니다. 멤버 관리자는 IdM 사용자 그룹에 사용자 또는 그룹을 추가할 수 있지만 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

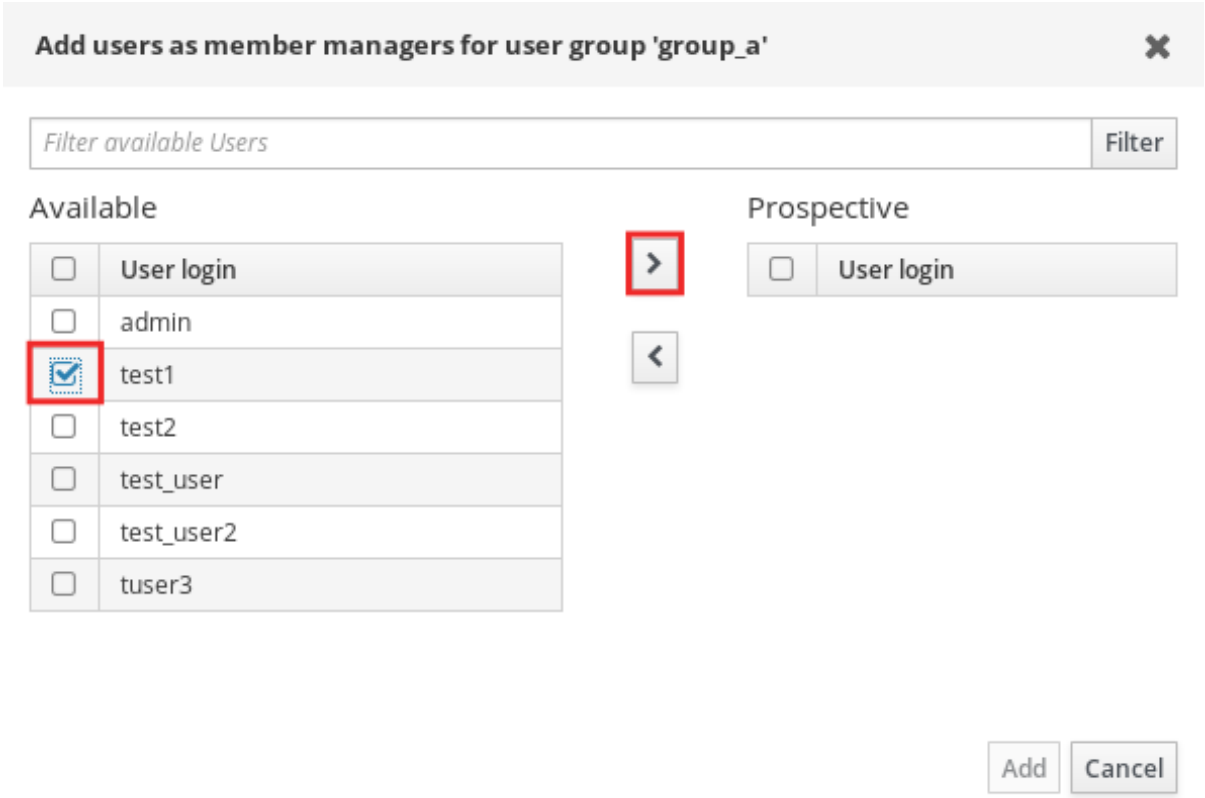
- IdM 웹 UI에 로그인되어 있습니다.
- 멤버 관리자로 추가하려는 사용자 또는 그룹의 이름과 관리하려는 그룹의 이름이 있어야 합니다.

절차

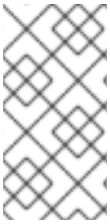
1. ID → 그룹을 클릭하고 왼쪽 사이드바 에서 사용자 그룹을 선택합니다.
2. 그룹의 이름을 클릭합니다.
3. 추가할 그룹 멤버 관리자 유형을 선택합니다. 사용자 또는 사용자 그룹.



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



7. **Add(추가)** 를 클릭하여 확인합니다.

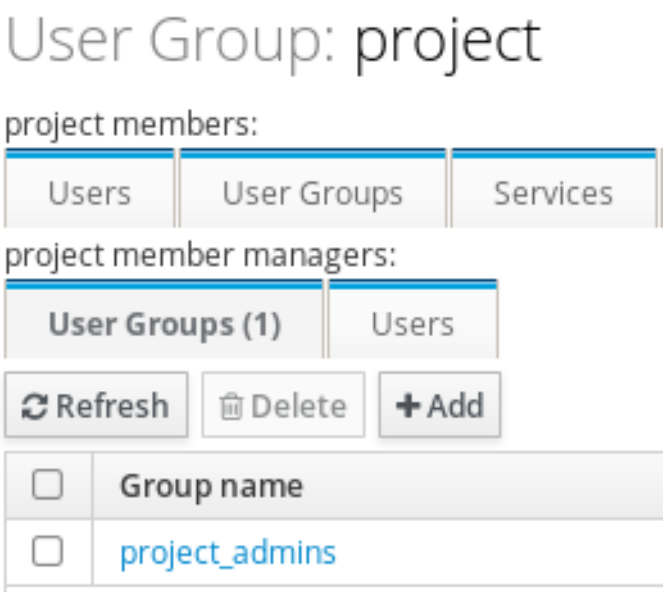


참고

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

검증 단계

- 새로 추가된 사용자 또는 사용자 그룹이 사용자 또는 사용자 그룹의 멤버 관리자 목록에 추가되었는지 확인합니다.



추가 리소스

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

12.7. IDM 웹 UI를 사용하여 그룹 구성원 보기

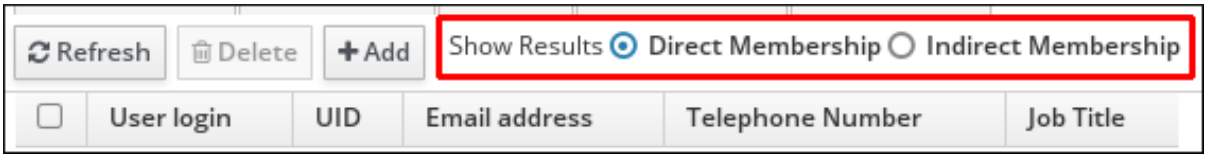
이 섹션에서는 **IdM 웹 UI**를 사용하여 그룹의 구성원을 보는 방법을 설명합니다. 직접 및 간접 그룹 구성원을 모두 볼 수 있습니다. 자세한 내용은 직접 및 간접 그룹 구성원을 참조하십시오.

사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.

절차

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



12.8. IDM 웹 UI를 사용하여 사용자 그룹에서 멤버 제거

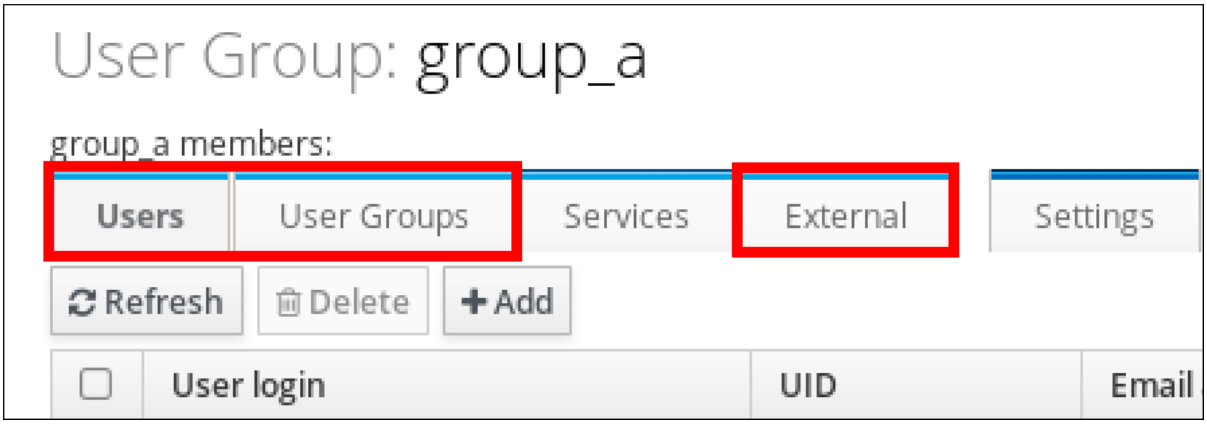
이 섹션에서는 IdM 웹 UI를 사용하여 사용자 그룹에서 멤버를 제거하는 방법을 설명합니다.

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.

절차

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



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

12.9. WEB UI를 사용하여 IDM 사용자 그룹에서 멤버 관리자로 사용자 또는 그룹 제거

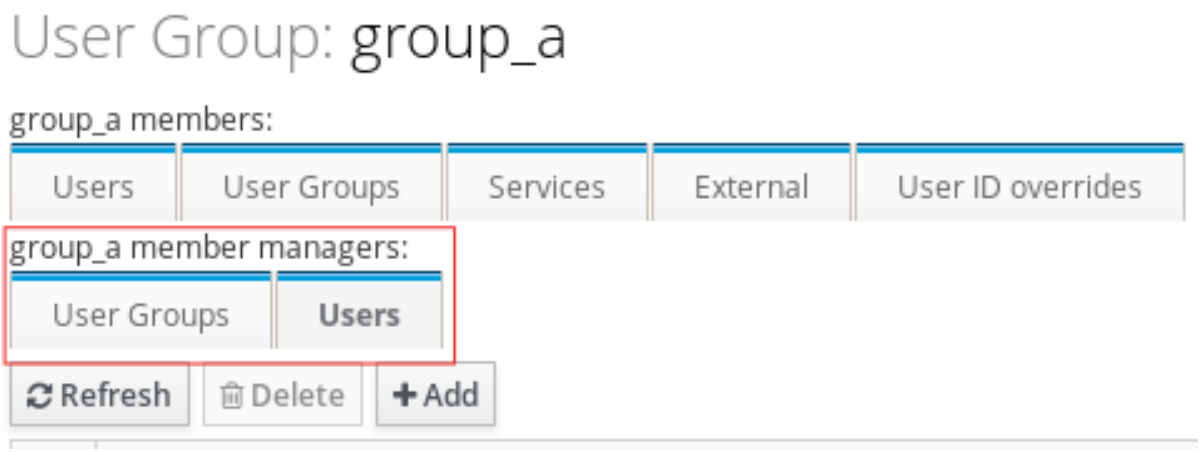
이 섹션에서는 웹 UI를 사용하여 IdM 사용자 그룹에서 사용자 또는 그룹을 멤버 관리자로 제거하는 방법을 설명합니다. 멤버 관리자는 IdM 사용자 그룹에서 사용자 또는 그룹을 제거할 수 있지만 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- 제거 중인 기존 멤버 관리자 사용자 또는 그룹의 이름과 관리 중인 그룹의 이름이 있어야 합니다.

절차

1. ID → 그룹을 클릭하고 왼쪽 사이드바 에서 사용자 그룹을 선택합니다.
2. 그룹의 이름을 클릭합니다.
3. 삭제할 멤버 관리자 유형을 선택합니다. 사용자 또는 사용자 그룹.



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

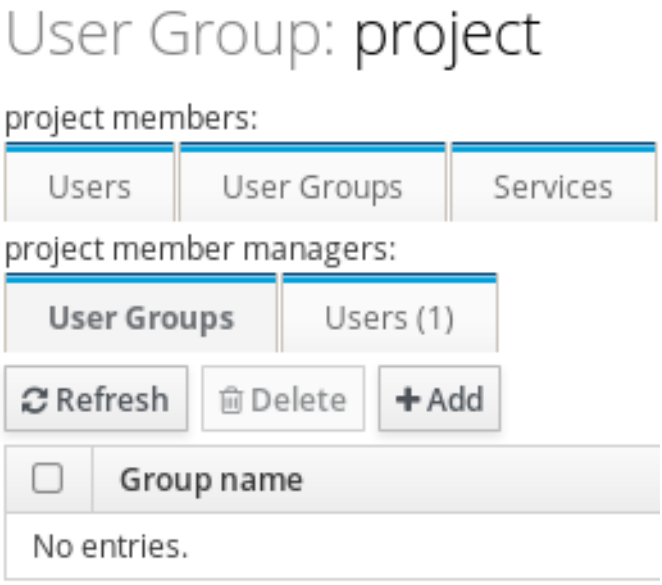
6. **Delete(삭제)** 를 클릭하여 확인합니다.

참고

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

검증 단계

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



추가 리소스

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

13장. 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.

13.1. 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 13.1. 기본적으로 생성된 사용자 그룹

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

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

주의

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

또한 IdM에서 새 *사용자가 생성될 때마다 IdM은 기본적으로 사용자 개인 그룹을 생성*합니다. 개인 그룹에 대한 자세한 내용은 개인 그룹이 없는 사용자 추가를 참조하십시오.

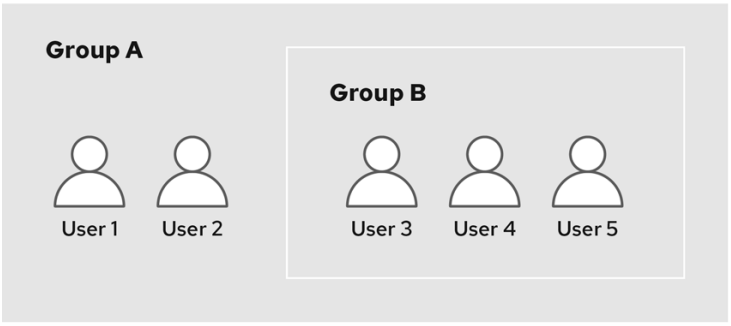
13.2. 직접 및 간접 그룹 구성원

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

예를 들어 다음 다이어그램에서 다음을 수행합니다.

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

그림 13.1. 직접 및 간접 그룹 멤버십



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

13.3. 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 IdM에 있습니다. Ansible을 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 IdM에 있습니다.

추가 리소스

- /usr/share/doc/ansible-freeipa/README-group.md Markdown 파일을 참조하십시오.

13.4. 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 --help**를 참조하십시오.
- **ipa** 도움말 페이지를 참조하십시오.

13.5. 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.

14장. IDM CLI를 사용하여 그룹 멤버십 자동화

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

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

이 장에서는 다음 주제를 다룹니다.

- IdM CLI를 사용하여 기본 자동 구성원 그룹 구성

14.1. 자동 그룹 멤버십의 이점

사용자에 대한 자동 멤버십을 사용하면 다음을 수행할 수 있습니다.

- 그룹 멤버십을 수동으로 관리하는 오버헤드 감소

  더 이상 모든 사용자와 호스트를 수동으로 그룹에 할당할 필요가 없습니다.

- 사용자 및 호스트 관리의 일관성 개선

  사용자와 호스트는 엄격하게 정의되고 자동으로 평가된 기준에 따라 그룹에 할당됩니다.

- 그룹 기반 설정 관리 간소화

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

14.2. 자동 구성원 규칙

자동 그룹 멤버십을 구성할 때 관리자는 **automember** 규칙을 정의합니다. **automember** 규칙은 특정 사용자 또는 호스트 대상 그룹에 적용됩니다. 한 번에 둘 이상의 그룹에 적용할 수 없습니다.

규칙을 생성한 후에는 관리자가 조건을 추가합니다. 대상 그룹에서 포함하거나 제외되는 사용자 또는 호스트를 지정합니다.

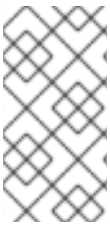
- 포함 조건

  사용자 또는 호스트 항목이 포함 조건을 충족하면 대상 그룹에 포함됩니다.

- 배타적 조건

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

조건은 Perl 호환 정규 표현식(PCRE) 형식으로 정규 표현식으로 지정됩니다. PCRE에 대한 자세한 내용은 convert rehiera(3) 매뉴얼 페이지를 참조하십시오.



참고

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

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

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

14.3. IDM CLI를 사용하여 자동 구성원 규칙 추가

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

automember 규칙을 추가한 후에는 automember 규칙에 조건 추가에 설명된 절차를 사용하여 조건을 추가할 수 있습니다.

참고

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

- 새 규칙의 대상 그룹은 IdM에 있어야 합니다.

절차

1. ipa automember-add 명령을 입력하여 automember 규칙을 추가합니다.

2. 메시지가 표시되면 다음을 지정합니다.

   - 자동 구성원 규칙. 대상 그룹 이름입니다.

   - 그룹화 유형. 이는 규칙에서 사용자 그룹을 대상으로 하는지 또는 호스트 그룹을 대상으로 하는지 여부를 지정합니다. 사용자 그룹을 대상으로 지정하려면 그룹을 입력합니다. 호스트 그룹을 대상으로 지정하려면 hostgroup 을 입력합니다.

   예를 들어 user_group 이라는 사용자 그룹에 대한 자동 구성원 규칙을 추가하려면 다음을 수행합니다.

```
$ ipa automember-add
Automember Rule: user_group
Grouping Type: group
-------------------------------
Added automember rule "user_group"
-------------------------------
  Automember Rule: user_group
```

검증 단계

- IdM CLI를 사용하여 기존 automember 규칙 보기를 사용하여 IdM에 기존 자동 구성원 규칙 및 조건을 표시할 수 있습니다.

14.4. IDM CLI를 사용하여 자동 구성원 규칙에 조건 추가

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방

법을 참조하십시오.

- 대상 규칙은 IdM에 있어야 합니다. 자세한 내용은 IdM CLI를 사용하여 자동 구성원 규칙 추가를 참조하십시오.

절차

1. ipa automember-add-condition 명령을 사용하여 하나 이상의 포함 또는 독점적 조건을 정의합니다.

2. 메시지가 표시되면 다음을 지정합니다.

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

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

```
$ ipa automember-add-condition
Automember Rule: user_group
Attribute Key: uid
Grouping Type: group
[Inclusive Regex]: .*
[Exclusive Regex]:
-------------------------------
Added condition(s) to "user_group"
-------------------------------
  Automember Rule: user_group
  Inclusive Regex: uid=.*
```

```
---------------------------
Number of conditions added 1
---------------------------
```

또 다른 예로 **automembership** 규칙을 사용하여 **AD(Active Directory)**에서 동기화된 모든 **Windows** 사용자를 대상으로 할 수 있습니다. 이를 위해 모든 **AD** 사용자가 공유하는 **objects Class** 특성 에서 모든 사용자를 대상으로 하는 모든 사용자를 대상으로 하는 조건을 생성합니다.

```
$ ipa automember-add-condition
Automember Rule: ad_users
Attribute Key: objectclass
Grouping Type: group
[Inclusive Regex]: ntUser
[Exclusive Regex]:
-----------------------------------
Added condition(s) to "ad_users"
-----------------------------------
  Automember Rule: ad_users
  Inclusive Regex: objectclass=ntUser
---------------------------
Number of conditions added 1
---------------------------
```

검증 단계

- IdM CLI를 사용하여 기존 automember 규칙 보기를 사용하여 IdM에 기존 자동 구성원 규칙 및 조건을 표시할 수 있습니다.

14.5. IDM CLI를 사용하여 기존 자동 구성원 규칙 보기

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

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

- 사용자 그룹을 대상으로 지정하려면 그룹을 입력합니다.

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

예를 들면 다음과 같습니다.

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

14.6. IDM CLI를 사용하여 자동 구성원 규칙 삭제

이 섹션에서는 IdM CLI를 사용하여 자동 구성원 규칙을 삭제하는 방법을 설명합니다.

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

1. ipa automember-del 명령을 입력합니다.

2. 메시지가 표시되면 다음을 지정합니다.

- 자동 구성원 규칙. 삭제할 규칙입니다.

-

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

14.7. IDM CLI를 사용하여 AUTOMEMBER 규칙에서 조건 제거

이 섹션에서는 automember 규칙에서 특정 조건을 제거하는 방법을 설명합니다.

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

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

     예를 들면 다음과 같습니다.

```
$ ipa automember-remove-condition
Automember Rule: user_group
Attribute Key: uid
Grouping Type: group
[Inclusive Regex]: .*
[Exclusive Regex]:
-------------------------------
```

```
Removed condition(s) from "user_group"
-----------------------------------
  Automember Rule: user_group
-----------------------------
Number of conditions removed 1
-----------------------------
```

14.8. IDM CLI를 사용하여 기존 항목에 자동 구성원 규칙 적용

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

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

참고

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 링크를 참조하십시오. **kinit**를 사용하여 **IdM**에 수동으로 로그인합니다.

절차

- 자동 멤버십을 다시 빌드하려면 **ipa automember-rebuild** 명령을 입력합니다. 다음 옵션을 사용하여 대상 항목을 지정합니다.
  - 모든 사용자에 대해 자동 멤버십을 다시 빌드하려면 **--type=group** 옵션을 사용합니다.

```
$ ipa automember-rebuild --type=group
-----------------------------------------------------
Automember rebuild task finished. Processed (9) entries.
-----------------------------------------------------
```

  - 모든 호스트에 대한 자동 멤버십을 다시 빌드하려면 **--type=hostgroup** 옵션을 사용합

니다.

- 지정된 사용자 또는 사용자의 자동 멤버십을 다시 빌드하려면 **--users=*target_user*** 옵션을 사용합니다.

```
$ ipa automember-rebuild --users=target_user1 --users=target_user2
-----------------------------------------------------
Automember rebuild task finished. Processed (2) entries.
-----------------------------------------------------
```

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

14.9. IDM CLI를 사용하여 기본 자동 구성원 그룹 구성

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

사전 요구 사항

- 관리자로 로그인해야 합니다. 자세한 내용은 **kinit**를 사용하여 **IdM**에 수동으로 로그인하는 방법을 참조하십시오.
- **IdM**에 기본값으로 설정할 대상 그룹이 있습니다.

절차

1. **ipa automember-default-group-set** 명령을 입력하여 기본 **automember** 그룹을 구성합니다.
2. 메시지가 표시되면 다음을 지정합니다.
   - 대상 그룹 이름을 지정하는 **default(fallback)** 그룹입니다.
   - 타겟이 사용자 그룹인지 또는 호스트 그룹인지 여부를 지정하는 유형 그룹화입니다. 사용자 그룹을 대상으로 지정하려면 그룹을 입력합니다. 호스트 그룹을 대상으로 지정하려면

hostgroup 을 입력합니다.

예를 들면 다음과 같습니다.

```
$ ipa automember-default-group-set
Default (fallback) Group: default_user_group
Grouping Type: group
-------------------------------------------------
Set default (fallback) group for automember "default_user_group"
-------------------------------------------------
  Default (fallback) Group:
cn=default_user_group,cn=groups,cn=accounts,dc=example,dc=com
```

참고

현재 기본 **automember** 그룹을 제거하려면 **ipa automember-default-group-remove** 명령을 입력합니다.

검증 단계

- 그룹이 올바르게 설정되었는지 확인하려면 **ipa automember-default-group-show** 명령을 입력합니다. 명령은 현재 기본 **automember** 그룹을 표시합니다. 예를 들면 다음과 같습니다.

```
$ ipa automember-default-group-show
Grouping Type: group
  Default (fallback) Group:
cn=default_user_group,cn=groups,cn=accounts,dc=example,dc=com
```

15장. IDM 웹 UI를 사용하여 그룹 멤버십 자동화

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

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

이 장에서는 다음 주제를 다룹니다.

- IdM 웹 UI를 사용하여 기본 사용자 그룹 구성
- IdM 웹 UI를 사용하여 기본 호스트 그룹 구성

15.1. 자동 그룹 멤버십의 이점

사용자에 대한 자동 멤버십을 사용하면 다음을 수행할 수 있습니다.

- 그룹 멤버십을 수동으로 관리하는 오버헤드 감소

  더 이상 모든 사용자와 호스트를 수동으로 그룹에 할당할 필요가 없습니다.

- 사용자 및 호스트 관리의 일관성 개선

  사용자와 호스트는 엄격하게 정의되고 자동으로 평가된 기준에 따라 그룹에 할당됩니다.

- 그룹 기반 설정 관리 간소화

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

15.2. 자동 구성원 규칙

자동 그룹 멤버십을 구성할 때 관리자는 **automember** 규칙을 정의합니다. **automember** 규칙은 특정 사용자 또는 호스트 대상 그룹에 적용됩니다. 한 번에 둘 이상의 그룹에 적용할 수 없습니다.

규칙을 생성한 후에는 관리자가 조건을 추가합니다. 대상 그룹에서 포함하거나 제외되는 사용자 또는 호스트를 지정합니다.

- 포함 조건

  사용자 또는 호스트 항목이 포함 조건을 충족하면 대상 그룹에 포함됩니다.

- 배타적 조건

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

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

참고

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

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

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

15.3. IDM 웹 UI를 사용하여 자동 구성원 규칙 추가

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

참고

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

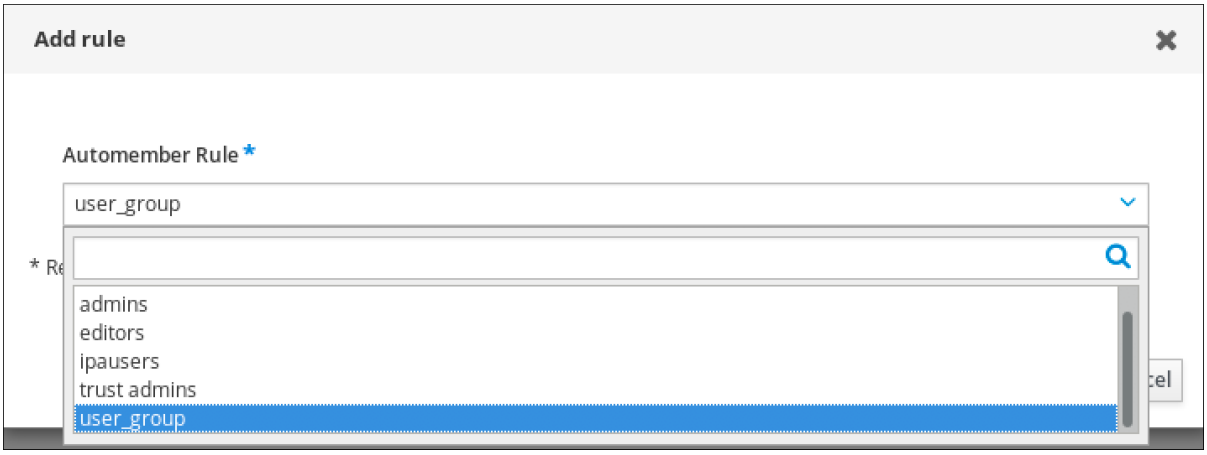
사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.

- 새 규칙의 대상 그룹은 IdM에 있습니다.

절차

1. ID → 자동 구성원을 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택합니다.
2. 추가를 클릭합니다.
3. Automember 규칙 필드에서 규칙이 적용할 그룹을 선택합니다. 대상 그룹 이름입니다.



4. Add(추가) 를 클릭하여 확인합니다.
5. 선택 사항: IdM 웹 UI를 사용하여 automember 규칙에 조건 추가에 설명된 절차를 사용하여 새 규칙에 조건을 추가할 수 있습니다.

15.4. IDM 웹 UI를 사용하여 자동 구성원 규칙에 조건 추가

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

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- admins 그룹의 멤버여야 합니다.

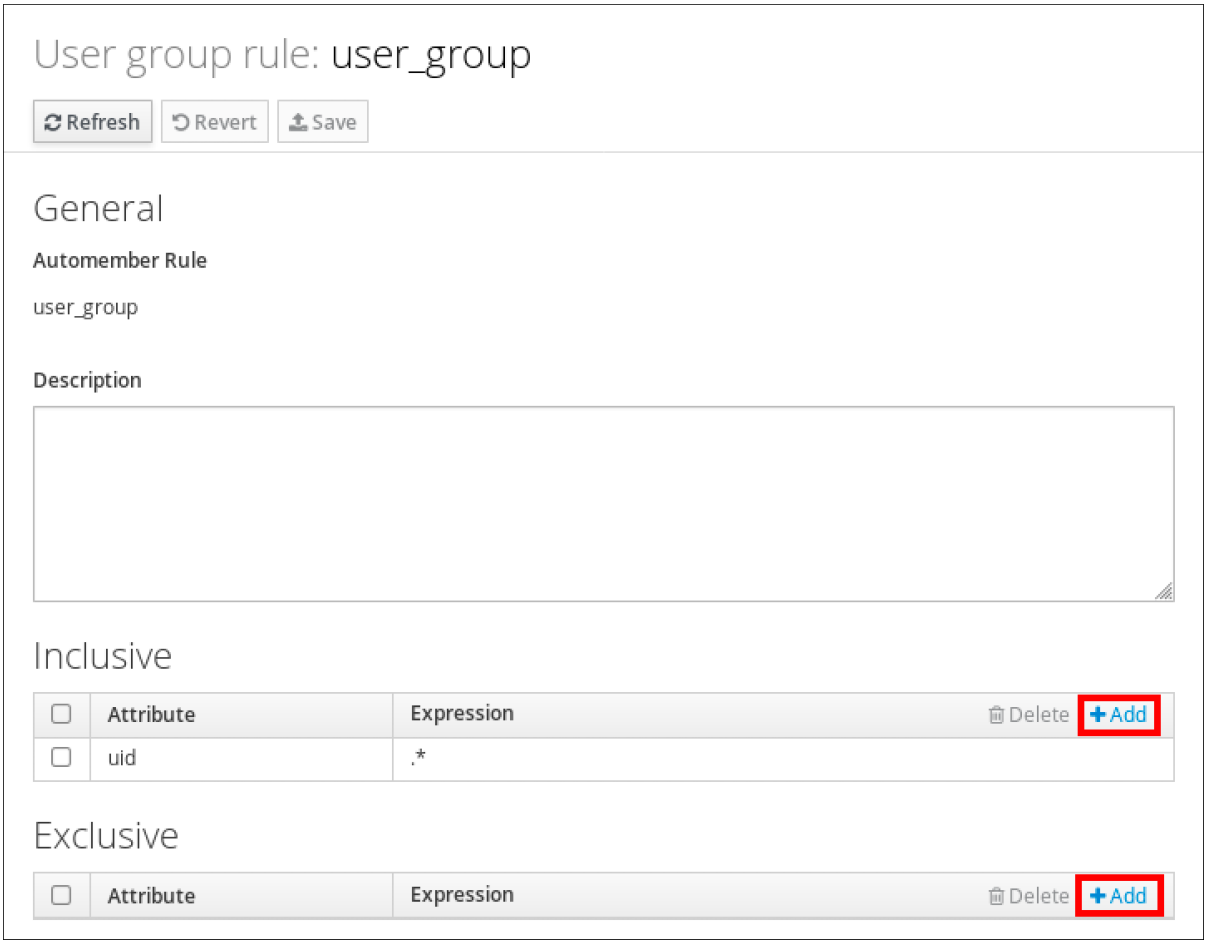
- 대상 규칙은 IdM에 있습니다.

절차

1. ID → 자동 구성원을 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택합니다.

2. 조건을 추가할 규칙을 클릭합니다.

3. Inclusive (포함) 또는 Exclusive (독점) 섹션에서 Add(추가)를 클릭합니다.

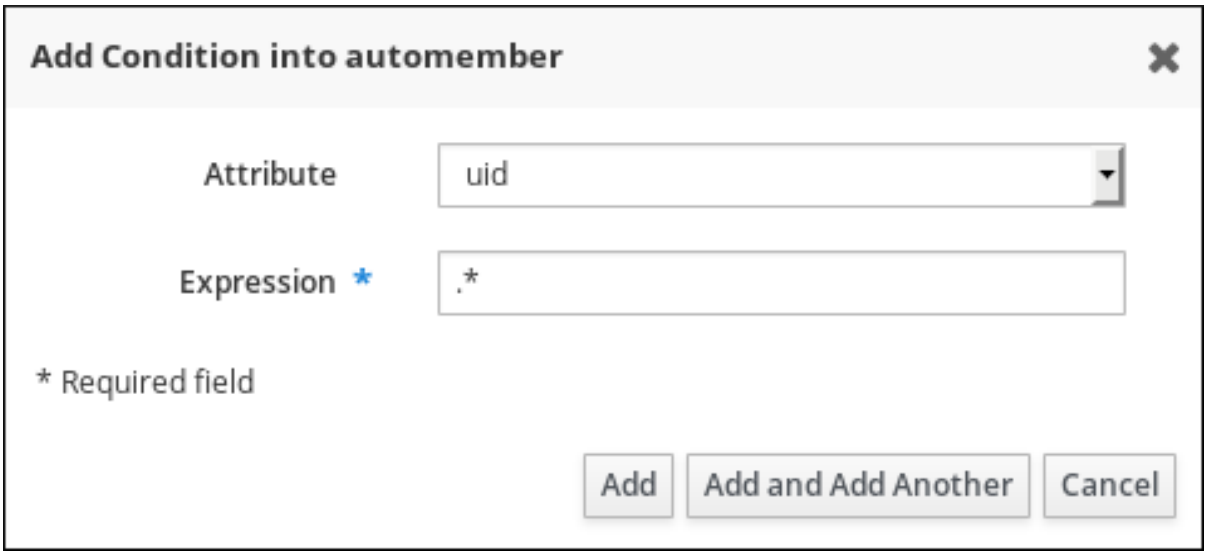


4. Attribute (특성) 필드에서 required 속성을 선택합니다(예: *uid*).

5. Expression(표현식) 필드에 정규 표현식을 정의합니다.

6. 추가를 클릭합니다.

예를 들어 다음 조건은 사용자 **ID(uid)** 속성에서 임의의 값(.*)을 가진 모든 사용자를 대상으로 합니다.



15.5. IDM 웹 UI를 사용하여 기존 자동 구성원 규칙 및 조건 보기

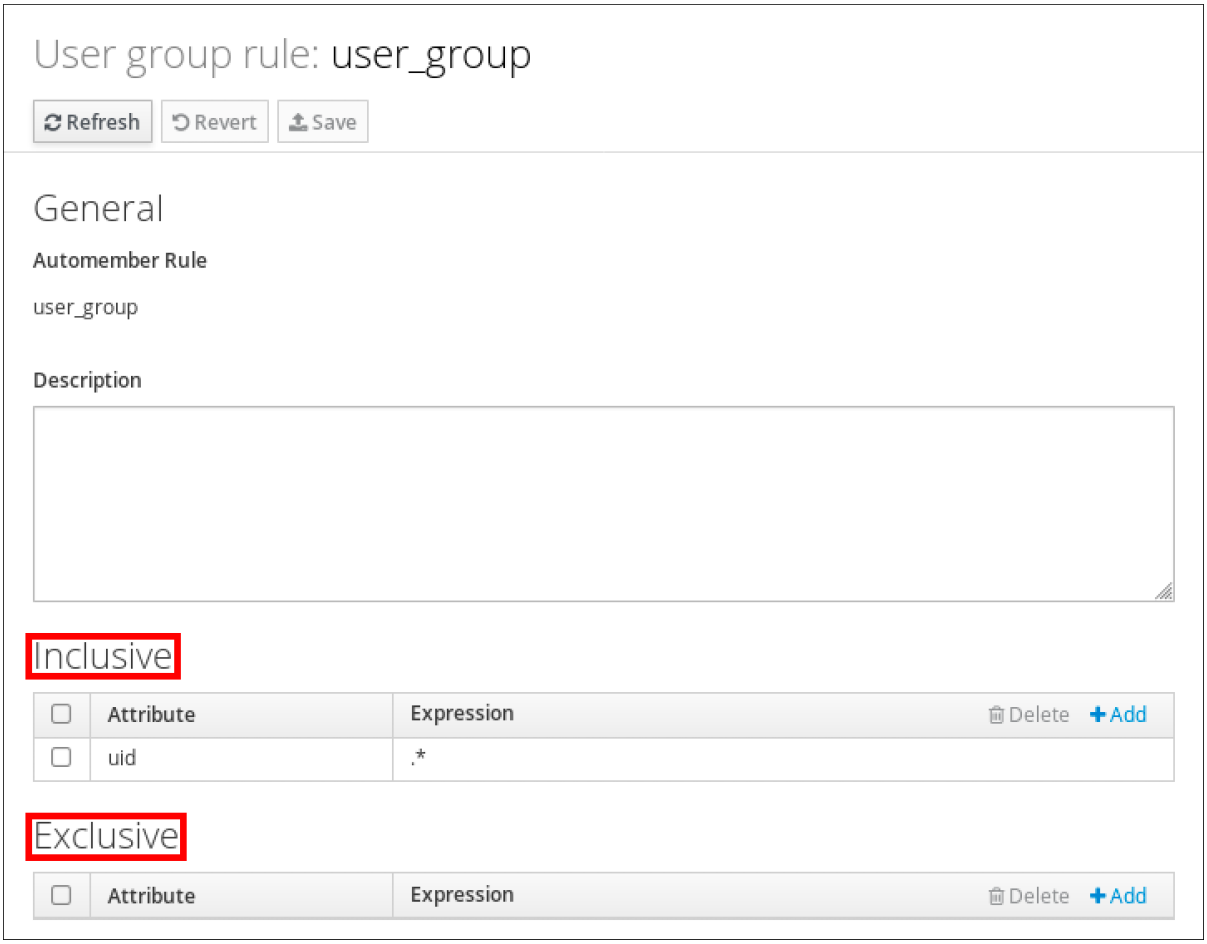
이 섹션에서는 **IdM 웹 UI**를 사용하여 기존의 자동 구성원 규칙 및 조건을 보는 방법에 대해 설명합니다.

사전 요구 사항

- **IdM 웹 UI**에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.

절차

1. **ID** → 자동 구성원을 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택하여 해당 자동 구성원 규칙을 확인합니다.
2. 선택 사항: 규칙을 클릭하여 **Inclusive** 또는 **Exclusive** 섹션에서 해당 규칙의 조건을 확인합니다.



15.6. IDM 웹 UI를 사용하여 자동 구성원 규칙 삭제

이 섹션에서는 IdM 웹 UI를 사용하여 자동 구성원 규칙을 삭제하는 방법을 설명합니다.

자동 구성원 규칙을 삭제하면 규칙과 연결된 모든 조건도 삭제됩니다. 규칙에서 특정 조건만 제거하려면 IdM 웹 UI를 사용하여 automember 규칙에서 조건 제거를 참조하십시오.

사전 요구 사항

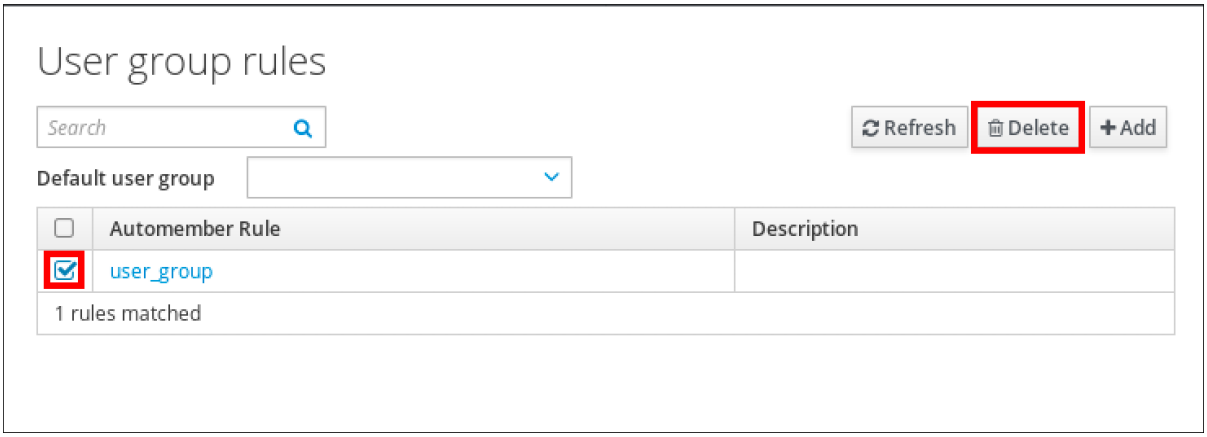
- IdM 웹 UI에 로그인되어 있습니다.
- admins 그룹의 멤버여야 합니다.

절차

1. ID → 자동 구성원을 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택하여 해당 자동 구성원 규칙을 확인합니다.

2. 제거할 규칙 옆에 있는 확인란을 선택합니다.

3. 삭제를 클릭합니다.



4. **Delete**(삭제) 를 클릭하여 확인합니다.

15.7. IDM 웹 UI를 사용하여 AUTOMEMBER 규칙에서 조건 제거

이 섹션에서는 IdM 웹 UI를 사용하여 automember 규칙에서 특정 조건을 제거하는 방법을 설명합니다.

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- admins 그룹의 멤버여야 합니다.

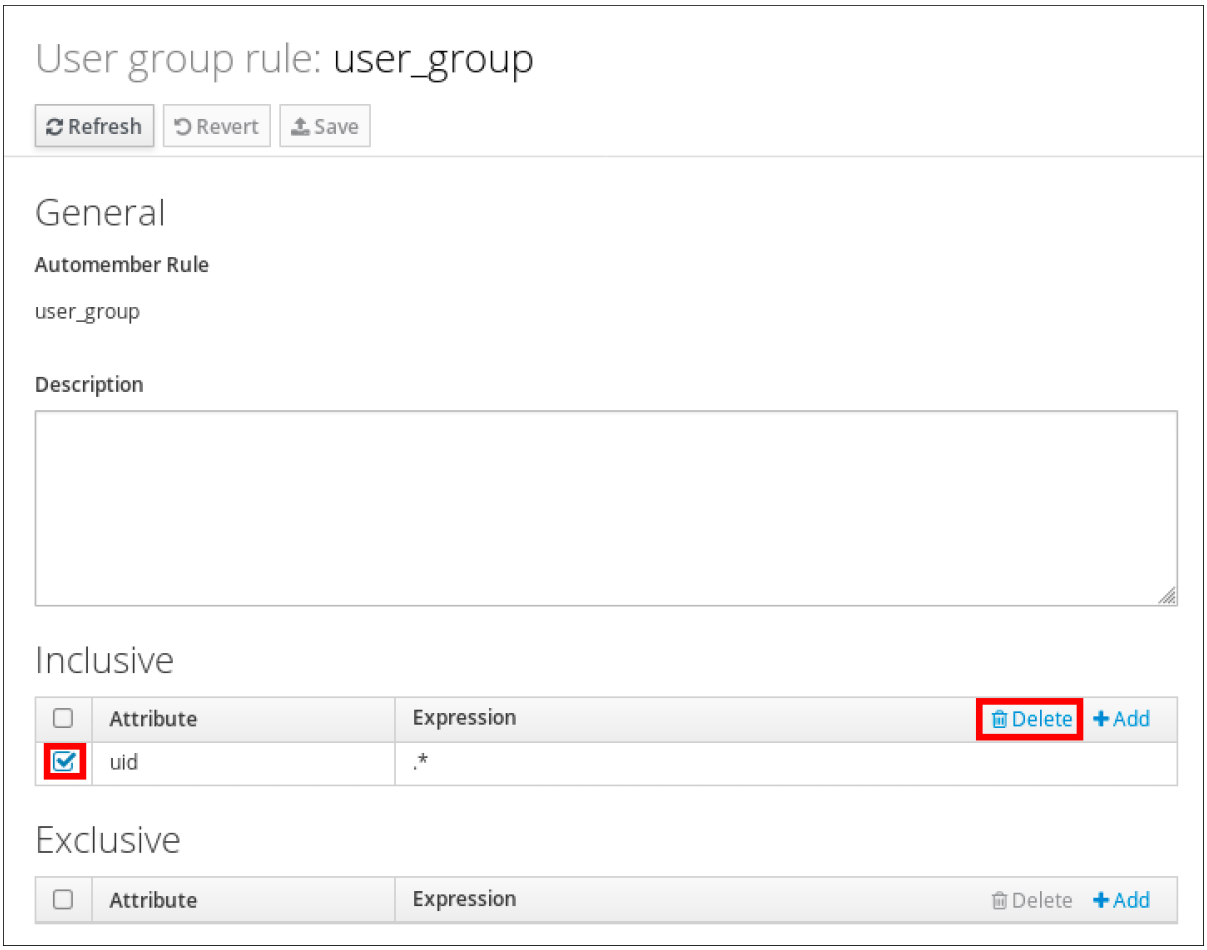
절차

1. ID → 자동 구성원을 클릭하고 사용자 그룹 규칙 또는 호스트 그룹 규칙을 선택하여 해당 자동 구성원 규칙을 확인합니다.

2. 규칙을 클릭하여 Inclusive 또는 Exclusive 섹션에서 해당 규칙의 조건을 확인합니다.

3. 제거할 조건 옆에 있는 확인란을 선택합니다.

4. **삭제**를 클릭합니다.



5. **Delete(삭제)** 를 클릭하여 확인합니다.

15.8. IDM 웹 UI를 사용하여 기존 항목에 자동 구성원 규칙 적용

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

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

참고

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

15.8.1. 모든 사용자 또는 호스트에 대한 자동 멤버십 다시 빌드

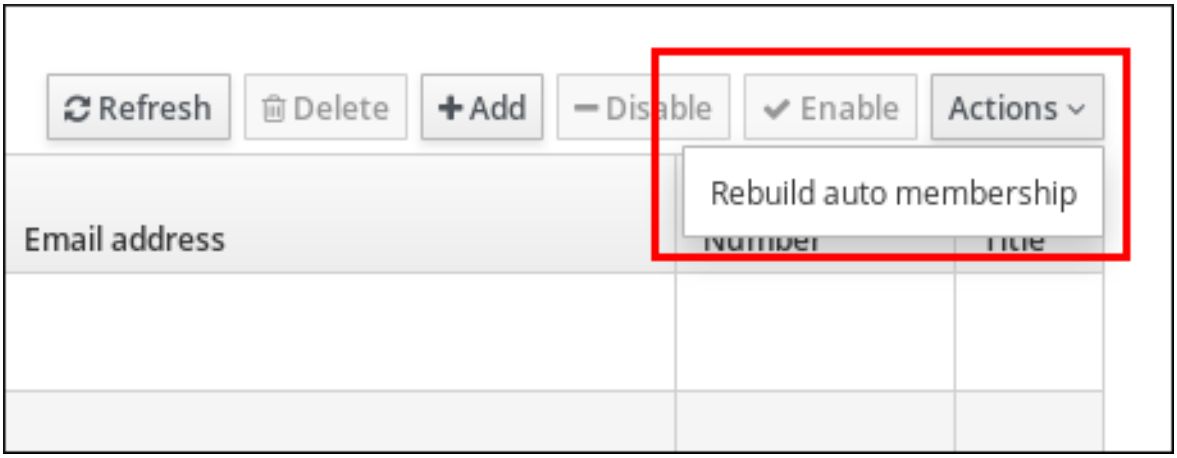
이 섹션에서는 모든 사용자 또는 호스트 항목에 대한 자동 멤버십을 다시 빌드하는 방법을 설명합니다.

사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- admins 그룹의 멤버여야 합니다.

절차

1. ID → 사용자 또는 호스트 를 선택합니다.
2. 작업 → 자동 멤버십 다시 빌드를 클릭합니다.



15.8.2. 단일 사용자 또는 호스트에 대한 자동 멤버십 다시 빌드

이 섹션에서는 특정 사용자 또는 호스트 항목에 대한 자동 멤버십을 다시 빌드하는 방법을 설명합니다.

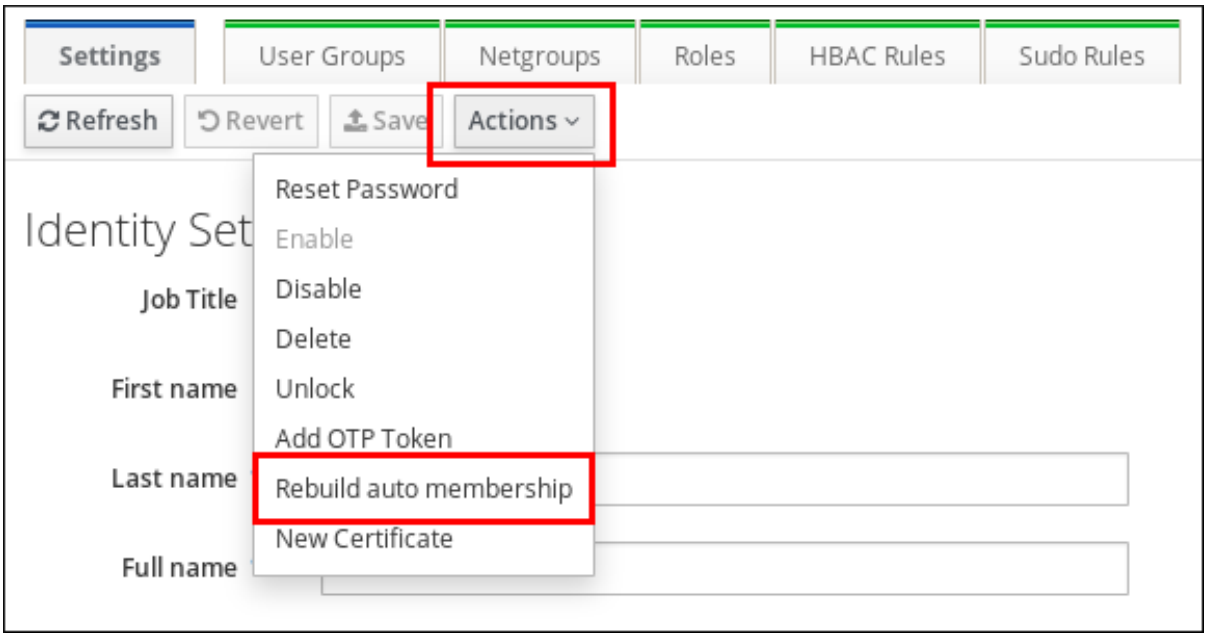
사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- admins 그룹의 멤버여야 합니다.

절차

절차

1. **ID → 사용자** 또는 **호스트** 를 선택합니다.
2. 필요한 사용자 또는 호스트 이름을 클릭합니다.
3. **작업 → 자동 멤버십 다시 빌드**를 클릭합니다.



15.9. IDM 웹 UI를 사용하여 기본 사용자 그룹 구성

기본 사용자 그룹을 구성하면 **automember** 규칙과 일치하지 않는 새 사용자 항목이 이 기본 그룹에 자동으로 추가됩니다.

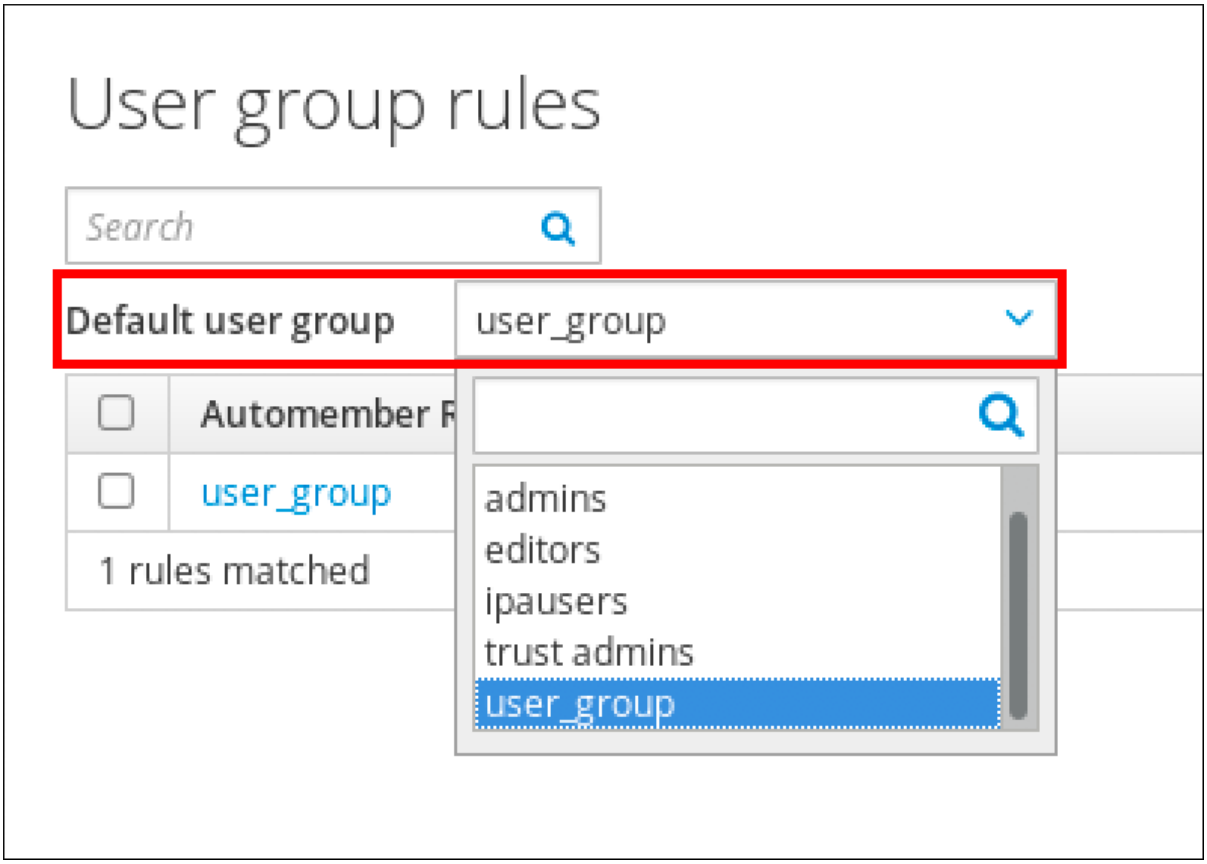
사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.
- IdM에 기본값으로 설정할 대상 사용자 그룹이 있습니다.

절차

1. **Identity → Automember** 를 클릭하고 사용자 그룹 규칙을 선택합니다.

2. **Default** 사용자 그룹 필드에서 기본 사용자 그룹으로 설정할 그룹을 선택합니다.



15.10. IDM 웹 UI를 사용하여 기본 호스트 그룹 구성

기본 호스트 그룹을 구성하면 **automember** 규칙과 일치하지 않는 새 호스트 항목이 이 기본 그룹에 자동으로 추가됩니다.

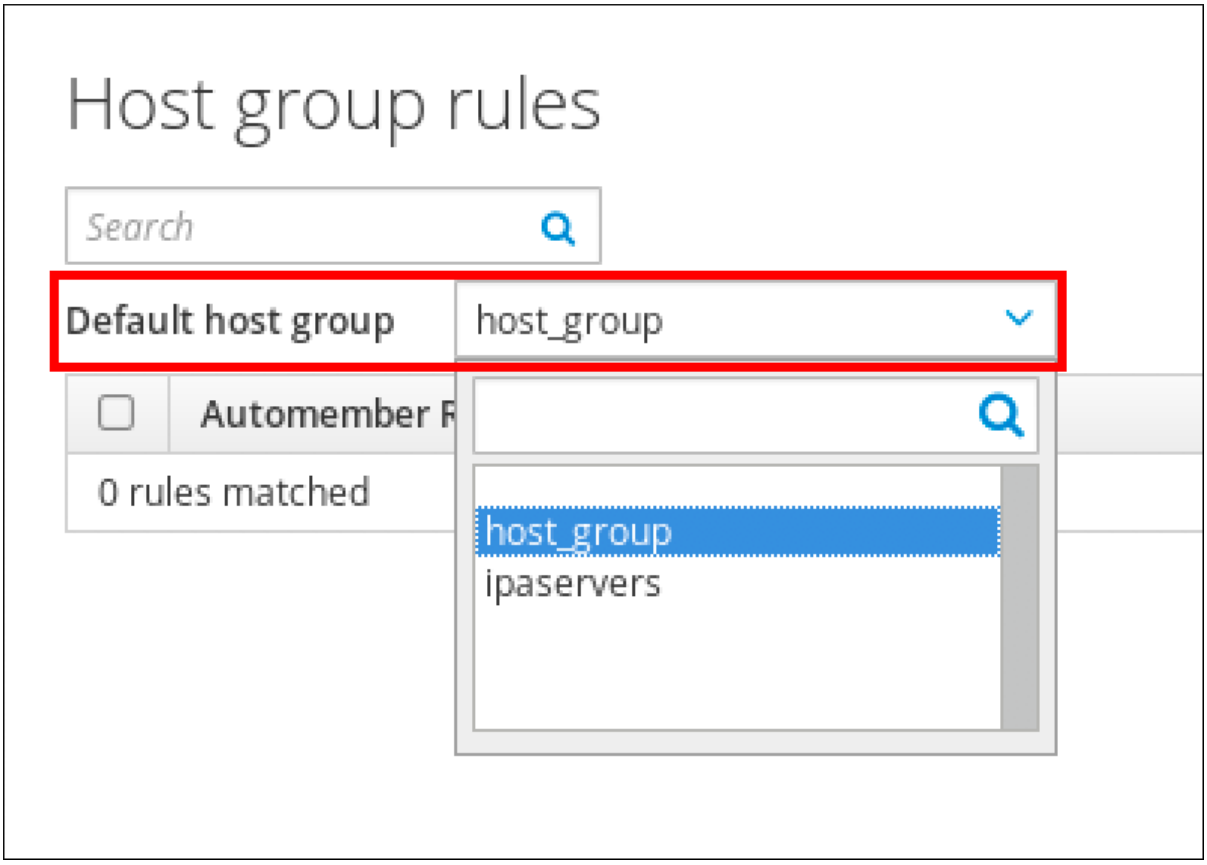
사전 요구 사항

- IdM 웹 UI에 로그인되어 있습니다.
- **admins** 그룹의 멤버여야 합니다.
- IdM에 기본값으로 설정할 대상 호스트 그룹이 있습니다.

절차

1. **ID → 자동 구성원**을 클릭하고 **호스트 그룹 규칙**을 선택합니다.

2. **Default** 호스트 그룹 필드에서 기본 호스트 그룹으로 설정할 그룹을 선택합니다.



16장. ANSIBLE을 사용하여 IDM의 그룹 멤버십 자동화

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

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

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

이 섹션에서는 다음 주제를 다룹니다.

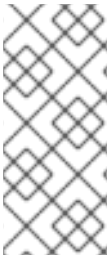
- IdM 관리를 위한 Ansible 제어 노드 준비
- Ansible을 사용하여 IdM 사용자 그룹에 대한 automember 규칙이 있는지 확인합니다.
- Ansible을 사용하여 IdM 사용자 그룹 automember 규칙에 조건이 있는지 확인합니다.
- Ansible을 사용하여 IdM 사용자 그룹 automember 규칙에 조건이 없는지 확인합니다.
- Ansible을 사용하여 IdM 그룹의 automember 규칙이 없는지 확인합니다.
- Ansible을 사용하여 IdM 호스트 그룹 automember 규칙에 조건이 있는지 확인합니다.

16.1. IDM 관리를 위한 ANSIBLE 제어 노드 준비

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 방법을 사용하면 모든 플레이북을 한 곳에서 찾을 수 있으며 root 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 입력하면 IdM 관리자 암호를 입력해야 합니다.

추가 리소스

- Ansible 플레이북을 사용하여 Identity Management 서버 설치.
- 인벤토리를 빌드하는 방법.

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

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

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- IdM에 testing_group 사용자 그룹이 있습니다.
- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - Ansible 컨트롤러에 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/automember/ 디렉터리에 있는 automember-group-present.yml Ansible 플레이북 파일을 복사합니다.

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

3. 편집할 automember-group-present-copy.yml 파일을 엽니다.

4. ipaautomember 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

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

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

5. 파일을 저장합니다.

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

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

추가 리소스

- 자동 그룹 멤버십 및 Automember 규칙의 이점 을 참조하십시오.
- Ansible을 사용하여 IdM 사용자 그룹 automember 규칙에 조건이 있는지 확인합니다.
- /usr/share/doc/ansible-freeipa/ 디렉토리의 README-automember.md 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/automember 디렉토리를 참조하십시오.

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

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

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- IdM에 testing_group 사용자 그룹 및 automember 사용자 그룹 규칙이 있습니다.
- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - Ansible 컨트롤러에 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/automember/ directory(예: automember-usergroup-rule-present.yml)에 있는 automember-hostgroup-rule-present.yml Ansible 플레이북 파일을 복사합니다.

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

3. 편집할 automember-usergroup-rule-present.yml 파일을 엽니다.

4. 다음 매개 변수를 수정하여 파일을 조정합니다.

- 플레이북 이름을 사용 사례에 해당하도록 변경합니다. 예를 들면 다음과 같습니다. automember 사용자 그룹 규칙 멤버가.
- 작업의 이름을 사용 사례에 맞게 변경합니다. 예를 들면 다음과 같습니다. 사용자 그룹의 automember 조건이 있는지 확인합니다.
- ipaautomember 작업 섹션에서 다음 변수를 설정합니다.
  - ipaadmin_password 변수를 IdM 관리자 암호로 설정합니다.
  - name 변수를 testing_group 으로 설정합니다.
  - automember_type 변수를 그룹으로 설정합니다.

- state 변수가 present 로 설정되어 있는지 확인합니다.
- action 변수가 member 로 설정되어 있는지 확인합니다.
- include 키 변수 를 UID 로 설정합니다.
- 포함 식 변수를 . *로 설정합니다 .*

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

5. 파일을 저장합니다.

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

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

검증 단계

1. IdM 관리자로 로그인합니다.

```
$ kinit admin
```

2. 다음과 같은 사용자를 추가합니다.

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

추가 리소스

- IdM CLI를 사용하여 기존 항목에 자동 멤버십 규칙 적용을 참조하십시오.
- 자동 그룹 멤버십 및 Automember 규칙의 이점 을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉토리의 README-automember.md 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/automember 디렉토리를 참조하십시오.

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

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

사전 요구 사항

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

- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - Ansible 컨트롤러에 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/automember/ 디렉터리(예: automember-usergroup-rule-absent.yml)에 있는 automember-hostgroup-rule-absent.yml Ansible 플레이북 파일을 복사합니다.

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

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

4. 다음 매개 변수를 수정하여 파일을 조정합니다.
   - 플레이북 이름을 사용 사례에 해당하도록 변경합니다. 예를 들면 다음과 같습니다. 사용자 그룹 규칙 member absent.
   - 작업의 이름을 사용 사례에 맞게 변경합니다. 예를 들면 다음과 같습니다. 사용자 그룹의 automember 조건이 없는지 확인합니다.
   - ipaautomember 작업 섹션에서 다음 변수를 설정합니다.

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

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

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

5. 파일을 저장합니다.

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

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

검증 단계

1. IdM 관리자로 로그인합니다.

```
$ kinit admin
```

2. automember 그룹을 확인합니다.

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

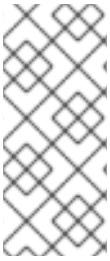
출력에 결정된 Regex: initials=dp 항목이 없으면 testing_group automember 규칙에 지정된 조건이 포함되어 있지 않습니다.

추가 리소스

- IdM CLI를 사용하여 기존 항목에 자동 멤버십 규칙 적용을 참조하십시오.
- 자동 그룹 멤버십 및 Automember 규칙의 이점 을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉토리의 README-automember.md 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/automember 디렉토리를 참조하십시오.

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

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



참고

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

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - Ansible 컨트롤러에 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/automember/ 디렉터리에 있는 automember-group-absent.yml Ansible 플레이북 파일을 복사합니다.

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

3. 편집할 automember-group-absent-copy.yml 파일을 엽니다.

4. ipaautomember 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

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

이는 현재 예제에 대한 수정된 Ansible 플레이북 파일입니다.

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

5. 파일을 저장합니다.

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

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

추가 리소스

- 자동 그룹 멤버십 및 Automember 규칙의 이점 을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉토리의 README-automember.md 파일을 참조하십시오.

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

16.6. ANSIBLE을 사용하여 IDM 호스트 그룹 AUTOMEMBER 규칙에 조건이 있는지 확인합니다.

이 섹션에서는 Ansible을 사용하여 IdM 호스트 그룹 automember 규칙에 조건이 있는지 확인하는 방법을 설명합니다. 이 예제에서는 FQDN 이 .*.idm.example.com 인 호스트가 primary_dns_domain_hosts 호스트 그룹의 멤버이고 FQDN 이 .*.example.org 가 primary_dns_domain_hosts 호스트 그룹의 멤버가 아닌지 확인하는 방법을 설명합니다.

사전 요구 사항

- IdM 관리자 암호를 알고 있습니다.
- IdM에 primary_dns_domain_hosts 호스트 그룹과 automember 호스트 그룹 규칙이 있습니다.
- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - Ansible 컨트롤러에 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/automember/ 디렉터리에 있는 automember-hostgroup-rule-present.yml Ansible 플레이북 파일을 복사합니다.

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

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

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

- **ipaadmin_password** 변수를 **IdM** 관리자 암호로 설정합니다.
- **name** 변수를 **primary_dns_domain_hosts** 로 설정합니다.
- **automember_type** 변수를 **hostgroup** 으로 설정합니다.
- **state** 변수가 **present** 로 설정되어 있는지 확인합니다.
- **action** 변수가 **member** 로 설정되어 있는지 확인합니다.
- **inclusive** 키 변수가 **fqdn;**으로 설정되어 있는지 확인합니다.
- 해당하는 **include** 표현식 변수 를 **.*.idm.example.com** 으로 설정합니다.
- 전용 키 변수를 **fqdn;**로 설정합니다.
- 해당 전용 표현식 변수를 **.*.example.org** 로 설정합니다.

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

```
---
- name: Automember user group rule member present
  hosts: ipaserver
  become: true
  tasks:
```

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

5. 파일을 저장합니다.

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

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

추가 리소스

- IdM CLI를 사용하여 기존 항목에 자동 멤버십 규칙 적용을 참조하십시오.
- 자동 그룹 멤버십 및 Automember 규칙의 이점 을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉토리의 README-automember.md 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/automember 디렉토리를 참조하십시오.

16.7. 추가 리소스

- Ansible 플레이북을 사용하여 사용자 계정 관리
- Ansible 플레이북을 사용하여 호스트 관리

- Ansible 플레이북을 사용하여 사용자 그룹 관리
- IdM CLI를 사용하여 호스트 그룹 관리

17장. IDM CLI를 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
- IdM CLI를 사용하여 위임 규칙 생성
- IdM CLI를 사용하여 기존 위임 규칙 보기
- IdM CLI를 사용하여 위임 규칙 수정
- IdM CLI를 사용하여 위임 규칙 삭제

17.1. 위임 규칙

위임 규칙을 만들어 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.

17.2. IDM CLI를 사용하여 위임 규칙 생성

이 섹션에서는 IdM CLI를 사용하여 위임 규칙을 생성하는 방법에 대해 설명합니다.

사전 요구 사항

- admins 그룹의 멤버로 로그인했습니다.

절차

- ipa delegation-add 명령을 입력합니다. 다음 옵션을 지정합니다.
  - --group: 사용자 그룹의 사용자 항목에 *권한을 부여하는* 그룹입니다.
  - --membergroup: 위임 그룹의 멤버가 *항목을 편집할 수* 있는 그룹입니다.
  - --permissions: 사용자가 지정된 속성을 보고(*읽기*) 지정된 속성을 보고 주어진 속성을 추가하거나 변경할 수 있는지(*쓰기*). 권한을 지정하지 않으면 *쓰기* 권한만 추가됩니다.
  - --attrs: 멤버 그룹의 사용자가 보거나 편집할 수 있는 속성입니다.

  예를 들면 다음과 같습니다.

```
$ ipa delegation-add "basic manager attributes" --permissions=read --permissions=write --
attrs=businesscategory --attrs=departmentnumber --attrs=employeetype --
attrs=employeenumber --group=managers --membergroup=employees
-----------------------------------------
Added delegation "basic manager attributes"
-----------------------------------------
  Delegation name: basic manager attributes
  Permissions: read, write
  Attributes: businesscategory, departmentnumber, employeetype, employeenumber
  Member user group: employees
  User group: managers
```

17.3. IDM CLI를 사용하여 기존 위임 규칙 보기

이 섹션에서는 IdM CLI를 사용하여 기존 위임 규칙을 보는 방법에 대해 설명합니다.

사전 요구 사항

- admins 그룹의 멤버로 로그인했습니다.

절차

- ipa delegation-find 명령을 입력합니다.

```
$ ipa delegation-find
-------------------
1 delegation matched
-------------------
  Delegation name: basic manager attributes
  Permissions: read, write
  Attributes: businesscategory, departmentnumber, employeenumber, employeetype
  Member user group: employees
  User group: managers
----------------------------
Number of entries returned 1
----------------------------
```

17.4. IDM CLI를 사용하여 위임 규칙 수정

이 섹션에서는 IdM CLI를 사용하여 기존 위임 규칙을 수정하는 방법을 설명합니다.

중요

--attrs 옵션은 지원되는 속성의 이전 목록을 덮어쓰므로 항상 새 속성과 함께 전체 속성 목록을 포함합니다. 이는 --permissions 옵션에도 적용됩니다.

사전 요구 사항

- admins 그룹의 멤버로 로그인했습니다.

절차

- 원하는 변경 사항을 사용하여 ipa delegation-mod 명령을 입력합니다. 예를 들어 기본 관리자 속성 예제 규칙에 displayname 속성을 추가하려면 다음을 수행합니다.

```
$ ipa delegation-mod "basic manager attributes" --attrs=businesscategory --
attrs=departmentnumber --attrs=employeetype --attrs=employeenumber --
attrs=displayname
---------------------------------------------
Modified delegation "basic manager attributes"
```

```
-----------------------------------------
  Delegation name: basic manager attributes
  Permissions: read, write
  Attributes: businesscategory, departmentnumber, employeetype, employeenumber,
displayname
  Member user group: employees
  User group: managers
```

17.5. IDM CLI를 사용하여 위임 규칙 삭제

이 섹션에서는 IdM CLI를 사용하여 기존 위임 규칙을 삭제하는 방법을 설명합니다.

사전 요구 사항

- admins 그룹의 멤버로 로그인했습니다.

절차

- ipa delegation-del 명령을 입력합니다.
- 메시지가 표시되면 삭제할 위임 규칙의 이름을 입력합니다.

```
$ ipa delegation-del
Delegation name: basic manager attributes
-----------------------------------------
Deleted delegation "basic manager attributes"
-----------------------------------------
```

18장. IDM WEBUI를 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
- IdM WebUI를 사용하여 위임 규칙 생성
- IdM WebUI를 사용하여 기존 위임 규칙 보기
- IdM WebUI를 사용하여 위임 규칙 수정
- IdM WebUI를 사용하여 위임 규칙 삭제

18.1. 위임 규칙

위임 규칙을 만들어 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.

18.2. IDM WEBUI를 사용하여 위임 규칙 생성

이 섹션에서는 IdM WebUI를 사용하여 위임 규칙을 생성하는 방법에 대해 설명합니다.

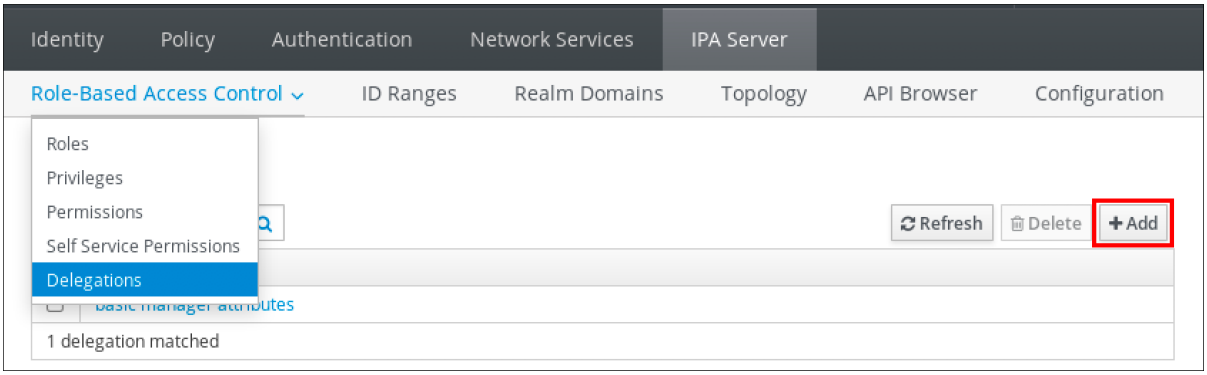
사전 요구 사항

- IdM 웹 UI에 admins 그룹의 멤버로 로그인했습니다.

절차

1. IPA 서버 메뉴에서 역할 기반 액세스 제어 → 위임을 클릭합니다.

2. 추가를 클릭합니다.



3. Add delegation(디렉토리 추가) 창에서 다음을 수행합니다.

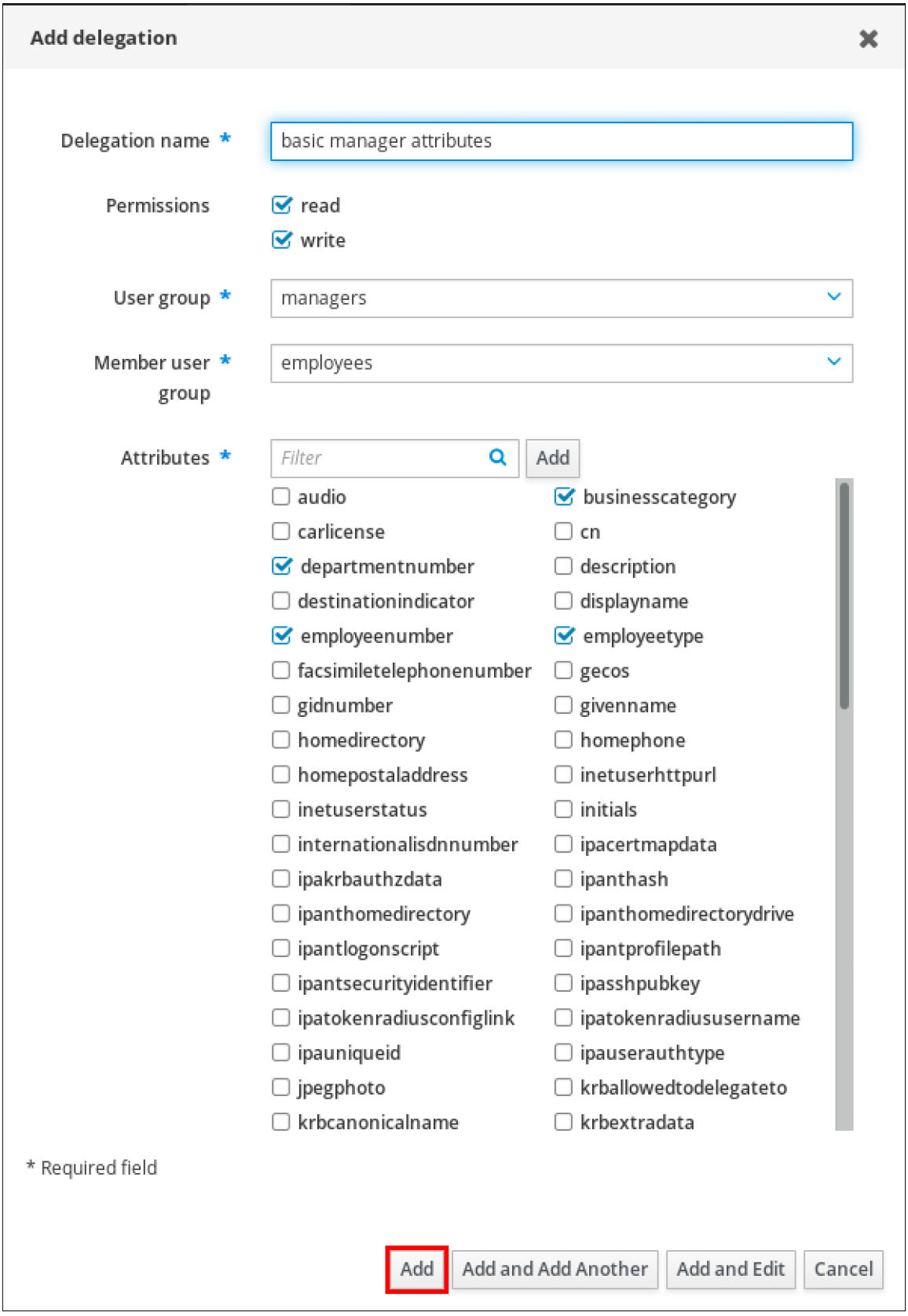
a. 새 위임 규칙의 이름을 지정합니다.

b. 사용자가 지정된 속성(*읽기*)을 보고 지정된 속성을 추가하거나 변경할 수 있는지 여부를 나타내는 확인란을 선택하여 권한을 설정합니다.

c. User group(사용자 그룹) 드롭다운 메뉴에서 *권한을 부여 받고 있는* 그룹을 선택하여 멤버 그룹에서 사용자 항목을 보거나 편집할 수 있습니다.

d. Member 사용자 그룹 드롭다운 메뉴에서 위임 그룹의 구성원이 *편집할 수 있는* 그룹을 선택합니다.

e. attributes(속성) 상자에서 권한을 부여할 속성별로 확인란을 선택합니다.



f.

Add(추가) 버튼을 클릭하여 새 위임 규칙을 저장합니다.

18.3. IDM WEBUI를 사용하여 기존 위임 규칙 보기

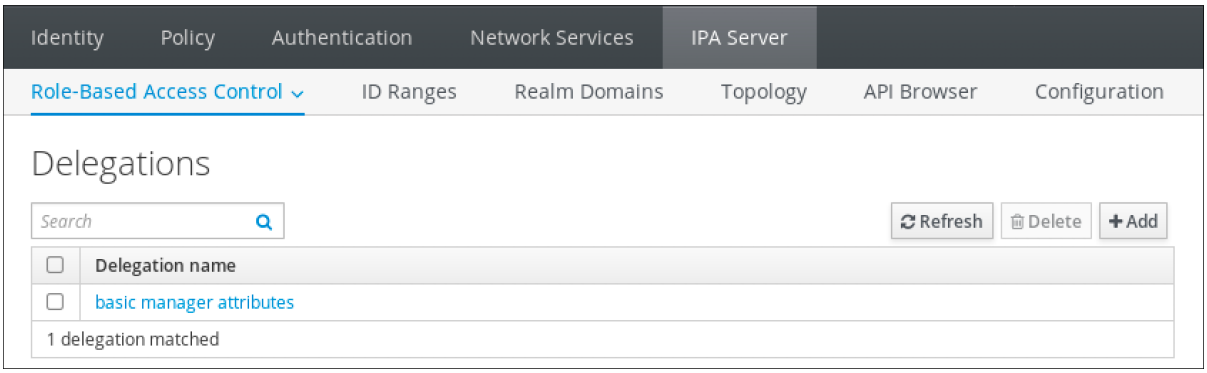
이 섹션에서는 IdM WebUI를 사용하여 기존 위임 규칙을 보는 방법에 대해 설명합니다.

사전 요구 사항

- IdM 웹 UI에 admins 그룹의 멤버로 로그인했습니다.

절차

- IPA 서버 메뉴에서 역할 기반 액세스 제어 → 위임을 클릭합니다.



18.4. IDM WEBUI를 사용하여 위임 규칙 수정

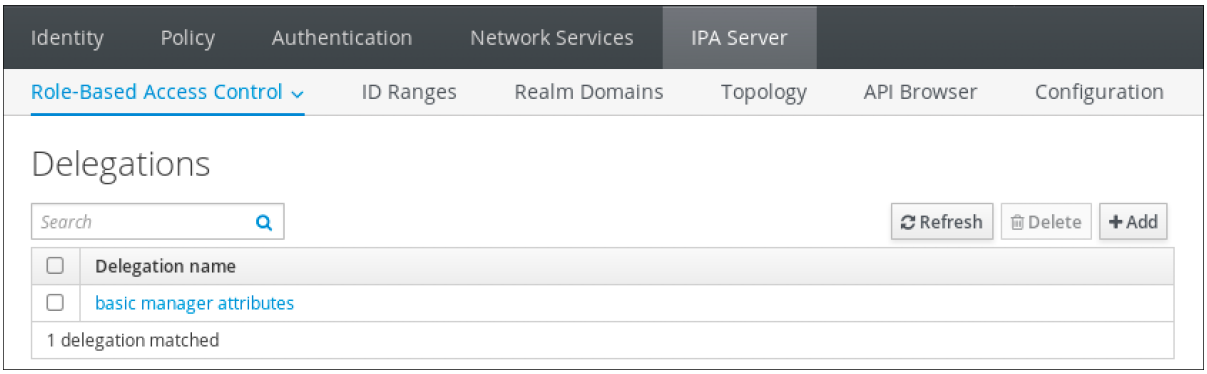
이 섹션에서는 IdM WebUI를 사용하여 기존 위임 규칙을 수정하는 방법에 대해 설명합니다.

사전 요구 사항

- IdM 웹 UI에 admins 그룹의 멤버로 로그인했습니다.

절차

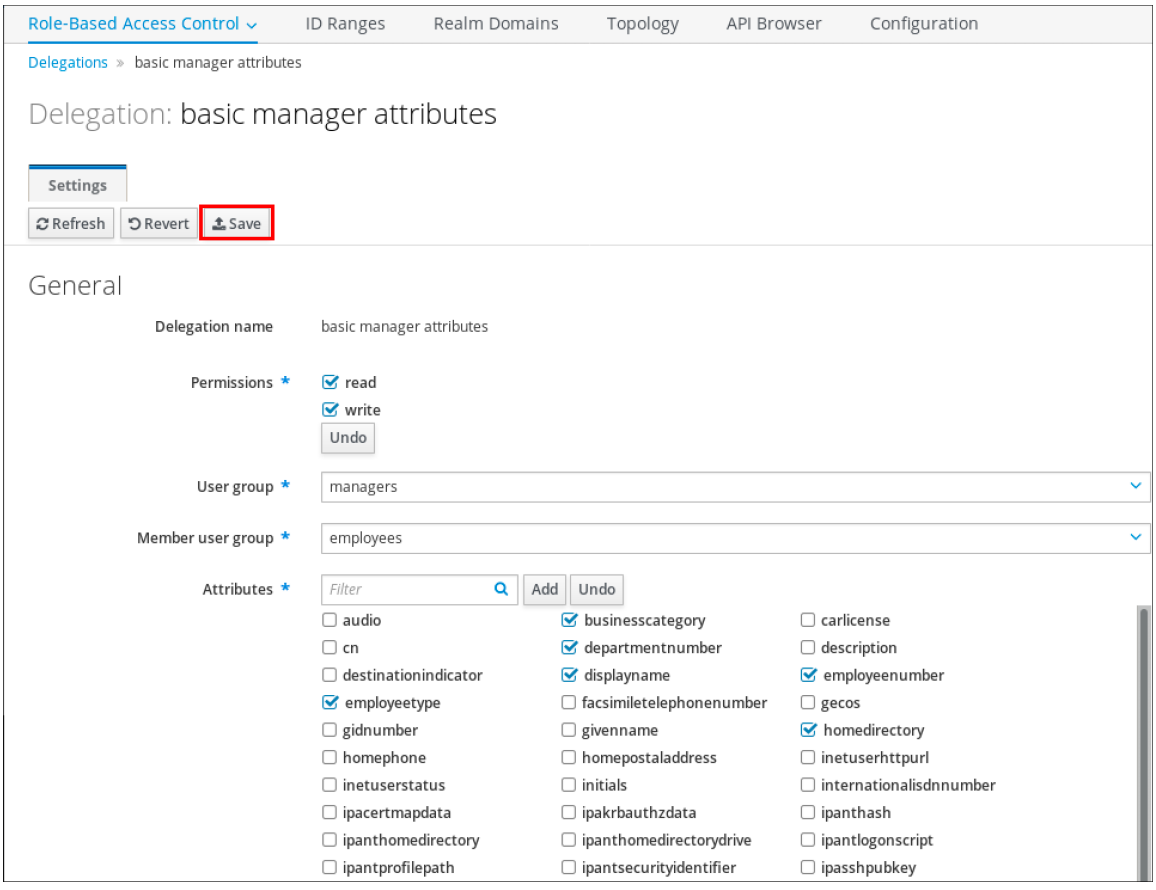
1. IPA 서버 메뉴에서 역할 기반 액세스 제어 → 위임을 클릭합니다.



2. 수정할 규칙을 클릭합니다.

3. 원하는 대로 변경합니다.

- 규칙의 이름을 변경합니다.
- 사용자가 지정된 속성(*읽기*)을 보고 지정된 속성을 추가하거나 변경할 수 있는지 여부를 나타내는 확인란을 선택하여 부여된 권한을 변경합니다.
- **User group**(사용자 그룹) 드롭다운 메뉴에서 *권한을 부여 받고 있는* 그룹을 선택하여 멤버 그룹에서 사용자 항목을 보거나 편집할 수 있습니다.
- **Member** 사용자 그룹 드롭다운 메뉴에서 위임 그룹의 구성원이 *편집할 수 있는* 그룹을 선택합니다.
- **attributes**(속성) 상자에서 권한을 부여할 속성별로 확인란을 선택합니다. 속성에 대한 권한을 제거하려면 관련 확인란을 선택 취소합니다.



- **Save(저장)** 버튼을 클릭하여 변경 사항을 저장합니다.

18.5. IDM WEBUI를 사용하여 위임 규칙 삭제

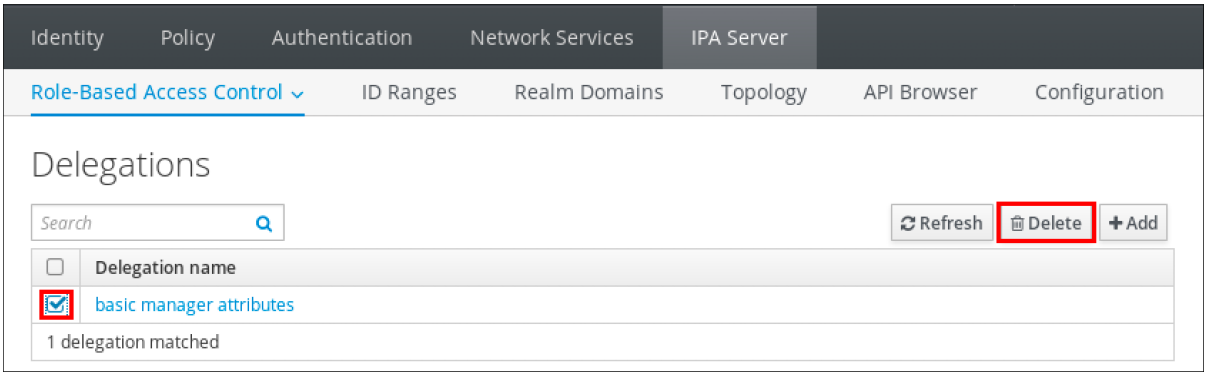
이 섹션에서는 **IdM WebUI**를 사용하여 기존 위임 규칙을 삭제하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 웹 **UI**에 **admins** 그룹의 멤버로 로그인했습니다.

절차

1. **IPA** 서버 메뉴에서 역할 기반 액세스 제어 → 위임을 클릭합니다.
2. 제거할 규칙 옆에 있는 확인란을 선택합니다.
3. 삭제를 클릭합니다.



4. **Delete(삭제)** 를 클릭하여 확인합니다.

19장. 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.

19.1. 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.

19.2. 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.

19.3. 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** 컨트롤러에 **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

- 분류 규칙을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉터리에서 README-delegation.md 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/ipadelegation 디렉토리에서 샘플 플레이북을 참조하십시오.

19.4. 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 IdM에 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 컨트롤러에 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

- 분류 규칙을 참조하십시오.
- **/usr/share/doc/ansible-freeipa/** 디렉터리에서 **README-delegation.md** 파일을 참조하십시오.
- **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation** 디렉토리에서 샘플 플레이북을 참조하십시오.

19.5. 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 관리자 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 컨트롤러에 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/ 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

- 분류 규칙을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉터리에서 README-delegation.md 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/ipadelegation 디렉토리에서 샘플 플레이북을

참조하십시오.

19.6. 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** 컨트롤러에 **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/** 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

- 분류 규칙을 참조하십시오.
- **/usr/share/doc/ansible-freeipa/** 디렉터리에서 **README-delegation.md** 파일을 참조하십시오.
- **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation** 디렉토리에서 샘플 플레이북을 참조하십시오.

20장. CLI를 사용하여 IDM에서 역할 기반 액세스 제어 관리

이 장에서는 IdM(Identity Management)의 역할 기반 액세스 제어를 소개하고 명령줄 인터페이스(CLI)에서 다음 작업을 설명합니다.

- 권한 관리
- 권한 관리
- 역할 관리

20.1. IDM의 역할 기반 액세스 제어

IdM의 역할 기반 액세스 제어(RBAC)는 셀프 서비스 및 위임 액세스 제어와 비교하여 사용자에게 매우 다른 종류의 권한을 부여합니다.

역할 기반 액세스 제어는 다음 세 부분으로 구성됩니다.

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

20.1.1. IdM의 권한

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

- write

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

이러한 작업은 세 개의 기본 대상에 적용됩니다 :

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

또한 다음과 같은 편의성 옵션은 해당 특성을 설정합니다.

- **type:** 유형의 개체 (사용자, 그룹 등)는 **subtree** 및 **target filter**를 설정합니다.
- **memberOf:** 그룹의 구성원; 대상 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. IdM을 사용하면 개별 속성을 허용 또는 차단하거나 사용자, 그룹 또는 sudo와 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.

20.1.2. 기본 관리 권한

관리 권한은 기본적으로 IdM과 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 IdM에서 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 IdM에서는 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.

20.1.3. IdM의 권한

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

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

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

참고

권한에는 다른 권한이 없을 수 있습니다.

20.1.4. IdM의 역할

역할은 역할에 지정된 사용자가 보유하는 권한 목록입니다.
실제로 권한은 지정된 하위 수준 작업을 수행할 수 있는 기능을 부여합니다(예: 사용자 항목 생성, 그룹에 항목 추가) 권한은 더 높은 수준의 작업(예: 지정된 그룹에 새 사용자 만들기)에 필요한 하나 이상의 권한을 결합합니다. 역할은 필요에 따라 권한을 함께 수집합니다. 예를 들어 사용자 관리자 역할은 사용자를 추가, 수정, 삭제할 수 있습니다.

중요

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

참고

역할에 다른 역할은 포함할 수 없습니다.

20.1.5. ID 관리에 사전 정의된 역할

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

표 20.1. ID 관리에서 사전 정의된 역할

Role	권한	설명
등록 관리자	호스트 등록	클라이언트 또는 호스트 등록 담당자
helpdesk	사용자 및 암호 재설정, 그룹 멤버십 수정	간단한 사용자 관리 작업 수행 담당

Role	권한	설명
IT 보안 전문가	Netgroups 관리자, HBAC 관리자, Sudo 관리자	호스트 기반 액세스 제어와 같은 보안 정책 관리, sudo 규칙
IT 전문가	호스트 관리자, 호스트 그룹 관리자, 서비스 관리자, 자동 마운트 관리자	호스트 관리 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

20.2. CLI에서 IDM 권한 관리

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 IdM(Identity Management) 권한을 관리하는 방법을 설명합니다.

사전 요구 사항

- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**
- **활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.**

절차

1. **ipa permission-add 명령을 사용하여 새 권한 항목을 생성합니다.**
예를 들어 *dns admin* 이라는 권한을 추가하려면 :

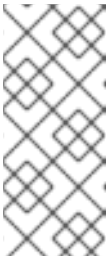
```
$ ipa permission-add "dns admin"
```

2. **다음 옵션을 사용하여 권한의 속성을 지정합니다.**

- **--bindtype 은 바인드 규칙 유형을 지정합니다. 이 옵션은 모든,익명 및 권한 인수를 허용합니다. 권한 바인드 유형은 역할을 통해 이 권한을 부여 받은 사용자만 수행할 수 있음을 나타냅니다.**
예를 들면 다음과 같습니다.

```
$ ipa permission-add "dns admin" --bindtype=all
```

--bindtype 을 지정하지 않으면 **permission** 이 기본값입니다.



참고

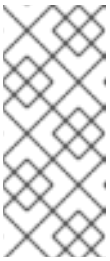
기본이 아닌 바인드 규칙 유형의 권한을 권한에 추가할 수 없습니다. 또한 권한이 이미 권한에 있는 권한을 기본이 아닌 바인드 규칙 유형으로 설정할 수 없습니다.

- **--right** 는 권한에 의해 부여된 권한을 나열하고 더 이상 사용되지 않는 **--permissions** 옵션을 대체합니다. 사용 가능한 값은 **add,delete,read,search,compare,write,all** 입니다.

여러 **--right** 옵션을 사용하거나 중괄호 안에 쉼표로 구분된 목록을 사용하여 여러 특성을 설정할 수 있습니다. 예를 들면 다음과 같습니다.

```
$ ipa permission-add "dns admin" --right=read --right=write
```

```
$ ipa permission-add "dns admin" --right={read,write}
```



참고

추가 및 삭제 는 읽기,검색,비교, 쓰기 등의 항목 수준의 작업(예: 사용자 삭제, 그룹 추가 등)입니다. **userCertificate**에 쓸 수는 있지만 **user Password** 는 읽을 수 없습니다.

- **--attrs** 는 권한이 부여되는 속성 목록을 제공합니다.
여러 **--attrs** 옵션을 사용하거나 중괄호 안의 쉼표로 구분된 목록에 옵션을 나열하여 여러 특성을 설정할 수 있습니다. 예를 들면 다음과 같습니다.

```
$ ipa permission-add "dns admin" --attrs=description --attrs=automountKey
```

```
$ ipa permission-add "dns admin" --attrs={description,automountKey}
```

--attrs 로 제공되는 속성은 존재해야 하며 지정된 오브젝트 유형에 대해 허용된 속성이 있어야 합니다. 그렇지 않으면 스키마 구문 오류로 인해 명령이 실패합니다.

- **--type** 은 권한이 적용되는 항목 오브젝트 유형(예: 사용자, 호스트 또는 서비스)을 정의

합니다. 각 유형에는 허용된 속성 집합이 있습니다.
예를 들면 다음과 같습니다.

```
$ ipa permission-add "manage service" --right=all --type=service --
attrs=krbprincipalkey --attrs=krbprincipalname --attrs=managedby
```

- **--subtree** 는 하위 트리 항목을 제공합니다. 필터는 이 하위 트리 항목 아래의 모든 항목을 대상으로 합니다. 기존 하위 트리 항목을 제공합니다 . **--subtree** 는 와일드카드 또는 존재하지 않는 도메인 이름(DN)을 허용하지 않습니다. 디렉터리에 DN을 포함합니다.
IdM은 간소화된 플랫 디렉터리 트리 구조를 사용하기 때문에 **--subtree** 를 사용하여 다른 구성의 컨테이너 또는 상위 항목인 자동 마운트 위치와 같은 일부 항목을 대상으로 지정할 수 있습니다. 예를 들면 다음과 같습니다.

```
$ ipa permission-add "manage automount locations" --
subtree="ldap://ldap.example.com:389/cn=automount,dc=example,dc=com" --
right=write --attrs=automountmapname --attrs=automountkey --
attrs=automountInformation
```

참고

type 및 **--subtree** 옵션은 상호 배타적입니다. **--type**에 대한 필터가 **--subtree** 의 간소화로 표시되어 관리자가 쉽게 사용할 수 있도록 합니다.

- **--filter** 는 LDAP 필터를 사용하여 권한이 적용되는 항목을 식별합니다.
IdM은 지정된 필터의 유효성을 자동으로 확인합니다. 필터는 유효한 LDAP 필터일 수 있습니다. 예를 들면 다음과 같습니다.

```
$ ipa permission-add "manage Windows groups" --filter="(!
(objectclass=posixgroup))" --right=write --attrs=description
```

- **--memberOf**는 그룹이 존재하는지 확인한 후 대상 필터를 지정된 그룹의 멤버로 설정합니다. 예를 들어 이 권한이 있는 사용자가 **engineer** 그룹의 멤버의 로그인 쉘을 수정하도록 하려면 다음을 수행합니다.

```
$ ipa permission-add ManageShell --right="write" --type=user --attr=loginshell --
memberof=engineers
```

- **--targetGroup** 은 그룹이 존재하는지 확인한 후 지정된 사용자 그룹으로 **target**을 설정합니다. 예를 들어 권한이 있는 사용자가 **engineers** 그룹에 **member** 속성을 쓰도록 하려면 (구성원 추가 또는 제거 가능).

```
$ ipa permission-add ManageMembers --right="write" --
subtree=cn=groups,cn=accounts,dc=example,dc=test --attr=member --
targetgroup=engineers
```

- 선택적으로 **DN**(대상 도메인 이름)을 지정할 수 있습니다.
  - **--target** 은 권한을 적용할 **DN**을 지정합니다. 와일드카드가 허용됩니다.
  - **--targetto** 는 항목을 이동할 수 있는 **DN** 하위 트리를 지정합니다.
  - **--targetfrom** 은 항목을 이동할 수 있는 **DN** 하위 트리를 지정합니다.

20.3. 기존 권한의 명령 옵션

필요에 따라 다음 변형을 사용하여 기존 권한을 수정합니다.

- 기존 권한을 편집하려면 **ipa permission-mod** 명령을 사용합니다. 권한을 추가할 때 와 동일한 명령 옵션을 사용할 수 있습니다.
- 기존 권한을 찾으려면 **ipa permission-find** 명령을 사용합니다. 권한을 추가할 때 와 동일한 명령 옵션을 사용할 수 있습니다.
- 특정 권한을 보려면 **ipa permission-show** 명령을 사용합니다.
--raw 인수는 생성된 원시 **389-ds ACI**를 보여줍니다. 예를 들면 다음과 같습니다.

```
$ ipa permission-show <permission> --raw
```

- **ipa permission-del** 명령은 권한을 완전히 삭제합니다.

추가 리소스

- **ipa** 도움말 페이지를 참조하십시오.

- ipa help 명령을 참조하십시오.

20.4. CLI에서 IDM 권한 관리

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 IdM(Identity Management) 권한을 관리하는 방법을 설명합니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 링크를 참조하십시오. kinit를 사용하여 IdM에 수동으로 로그인합니다.
- 기존 권한. 권한에 대한 자세한 내용은 CLI에서 IdM 권한 관리를 참조하십시오.

절차

1. ipa privilege-add 명령을(를) 사용하여 권한 항목을 추가합니다. 예: 설명으로 *파일 시스템 관리* 라는 권한을 추가하려면 다음을 실행합니다.

```
$ ipa privilege-add "managing filesystems" --desc="for filesystems"
```

2. 예를 들어, privilege-add-permission 명령을 사용하여 권한 그룹에 필요한 권한을 할당합니다. 예를 들어 *자동 마운트 관리 및 ftp 서비스 관리* 라는 권한을 *파일 시스템 권한 관리에* 추가하려면 다음을 수행합니다.

```
$ ipa privilege-add-permission "managing filesystems" --permissions="managing automount" --permissions="managing ftp services"
```

20.5. 기존 권한에 대한 명령 옵션

필요에 따라 기존 권한을 수정하려면 다음 변형을 사용합니다.

- 기존 권한을 수정하려면 **ipa privilege-mod** 명령을 사용합니다.
- 기존 권한을 찾으려면 **ipa privilege-find** 명령을 사용합니다.
- 특정 권한을 보려면 **ipa privilege-show** 명령을 사용합니다.
- **ipa privilege-remove-permission** 명령은 권한에서 하나 이상의 권한을 제거합니다.
- **ipa privilege-del** 명령은 권한을 완전히 삭제합니다.

추가 리소스

- **ipa** 도움말 페이지를 참조하십시오.
- **ipa help** 명령을 참조하십시오.

20.6. CLI에서 IDM 역할 관리

이 섹션에서는 **CLI(명령줄 인터페이스)**를 사용하여 **IdM(Identity Management)** 역할을 관리하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 **Kerberos** 티켓. 자세한 내용은 **kinit**를 사용하여 **IdM**에 수동으로 로그인하는 방법을 참조하십시오.
- 기존 권한. 권한에 대한 자세한 내용은 **CLI**에서 **IdM** 권한 관리를 참조하십시오.

절차

1. **ipa role-add** 명령을 사용하여 새 역할 항목을 추가합니다.

```
$ ipa role-add --desc="User Administrator" useradmin
----------------------
Added role "useradmin"
----------------------
Role name: useradmin
Description: User Administrator
```

2. **ipa role-add-privilege** 명령을 사용하여 필요한 권한을 역할에 추가합니다.

```
$ ipa role-add-privilege --privileges="user administrators" useradmin
Role name: useradmin
Description: User Administrator
Privileges: user administrators
----------------------------
Number of privileges added 1
----------------------------
```

3. **ipa role-add-member** 명령을 사용하여 필요한 멤버를 역할에 추가합니다. 허용되는 멤버 유형은 사용자, 그룹, 호스트 및 호스트 그룹입니다.
예를 들어 ***useradmins****라는 그룹을 이전에 생성한* ***useradmin*** 역할에 추가하려면 다음을 수행합니다.

```
$ ipa role-add-member --groups=useradmins useradmin
Role name: useradmin
Description: User Administrator
Member groups: useradmins
Privileges: user administrators
------------------------
Number of members added 1
------------------------
```

20.7. 기존 역할에 대한 명령 옵션

필요에 따라 다음 변형을 사용하여 기존 역할을 수정합니다.

- 기존 역할을 수정하려면 **ipa role-mod** 명령을 사용합니다.
- 기존 역할을 찾으려면 **ipa role-find** 명령을 사용합니다.

- 특정 역할을 보려면 **ipa role-show** 명령을 사용합니다.
- 역할에서 멤버를 제거하려면 **ipa role-remove-member** 명령을 사용합니다.
- **ipa role-remove-privilege** 명령은 역할에서 하나 이상의 권한을 제거합니다.
- **ipa role-del** 명령은 역할을 완전히 삭제합니다.

추가 리소스

- **ipa man** 페이지 참조
- **ipa help** 명령을 참조하십시오.

21장. IDM 웹 UI를 사용하여 역할 기반 액세스 제어 관리

이 장에서는 IdM(Identity Management)의 역할 기반 액세스 제어를 소개하고 웹 인터페이스(웹 UI)에서 다음 작업을 설명합니다.

- 권한 관리
- 권한 관리
- 역할 관리

21.1. IDM의 역할 기반 액세스 제어

IdM의 역할 기반 액세스 제어(RBAC)는 셀프 서비스 및 위임 액세스 제어와 비교하여 사용자에게 매우 다른 종류의 권한을 부여합니다.

역할 기반 액세스 제어는 다음 세 부분으로 구성됩니다.

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

21.1.1. IdM의 권한

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

- write

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

이러한 작업은 세 개의 기본 대상에 적용됩니다 :

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

또한 다음과 같은 편의성 옵션은 해당 특성을 설정합니다.

- **type:** 유형의 개체 (사용자, 그룹 등)는 **subtree** 및 **target filter**를 설정합니다.
- **memberOf:** 그룹의 구성원; 대상 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. IdM을 사용하면 개별 속성을 허용 또는 차단하거나 사용자, 그룹 또는 sudo와 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.

21.1.2. 기본 관리 권한

관리 권한은 기본적으로 IdM과 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 IdM에서 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 IdM에서는 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.

21.1.3. IdM의 권한

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

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

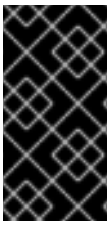
이러한 세 가지 하위 수준 작업을 이름이 지정된 사용자 지정 권한의 형태로 더 높은 수준의 작업(예: 사용자 추가) 에 결합하면 시스템 관리자가 역할을 쉽게 관리할 수 있습니다. IdM에는 이미 몇 가지 기본 권한이 포함되어 있습니다. 사용자 및 사용자 그룹 외에도 호스트 및 호스트 그룹은 네트워크 서비스에도 권한이 할당됩니다. 이 방법을 사용하면 특정 네트워크 서비스를 사용하는 호스트 집합의 사용자 집합에 의한 작업을 세부적으로 제어할 수 있습니다.

참고

권한에는 다른 권한이 없을 수 있습니다.

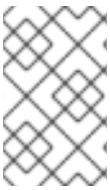
21.1.4. IdM의 역할

역할은 역할에 지정된 사용자가 보유하는 권한 목록입니다.
실제로 권한은 지정된 하위 수준 작업을 수행할 수 있는 기능을 부여합니다(예: 사용자 항목 생성, 그룹에 항목 추가) 권한은 더 높은 수준의 작업(예: 지정된 그룹에 새 사용자 만들기)에 필요한 하나 이상의 권한을 결합합니다. 역할은 필요에 따라 권한을 함께 수집합니다. 예를 들어 사용자 관리자 역할은 사용자를 추가, 수정, 삭제할 수 있습니다.



중요

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



참고

역할에 다른 역할은 포함할 수 없습니다.

21.1.5. ID 관리에 사전 정의된 역할

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

표 21.1. ID 관리에서 사전 정의된 역할

Role	권한	설명
등록 관리자	호스트 등록	클라이언트 또는 호스트 등록 담당자
helpdesk	사용자 및 암호 재설정, 그룹 멤버십 수정	간단한 사용자 관리 작업 수행 담당

Role	권한	설명
IT 보안 전문가	Netgroups 관리자, HBAC 관리자, Sudo 관리자	호스트 기반 액세스 제어와 같은 보안 정책 관리, sudo 규칙
IT 전문가	호스트 관리자, 호스트 그룹 관리자, 서비스 관리자, 자동 마운트 관리자	호스트 관리 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

21.2. IDM 웹 UI에서 권한 관리

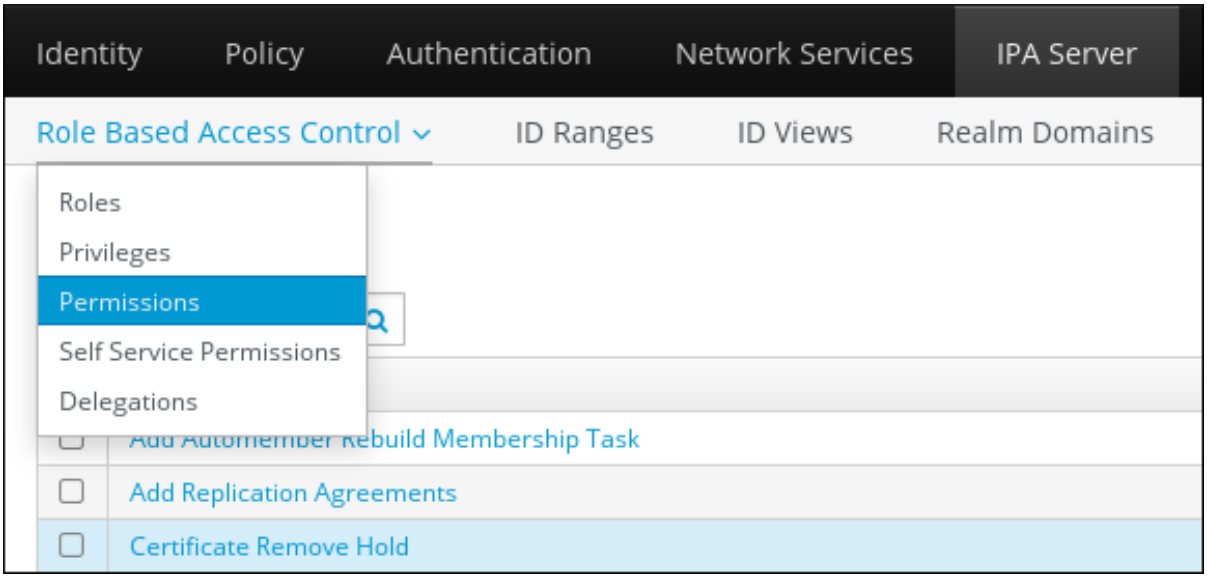
이 섹션에서는 웹 인터페이스(IdM 웹 UI)를 사용하여 IdM(Identity Management)에서 권한을 관리하는 방법을 설명합니다.

사전 요구 사항

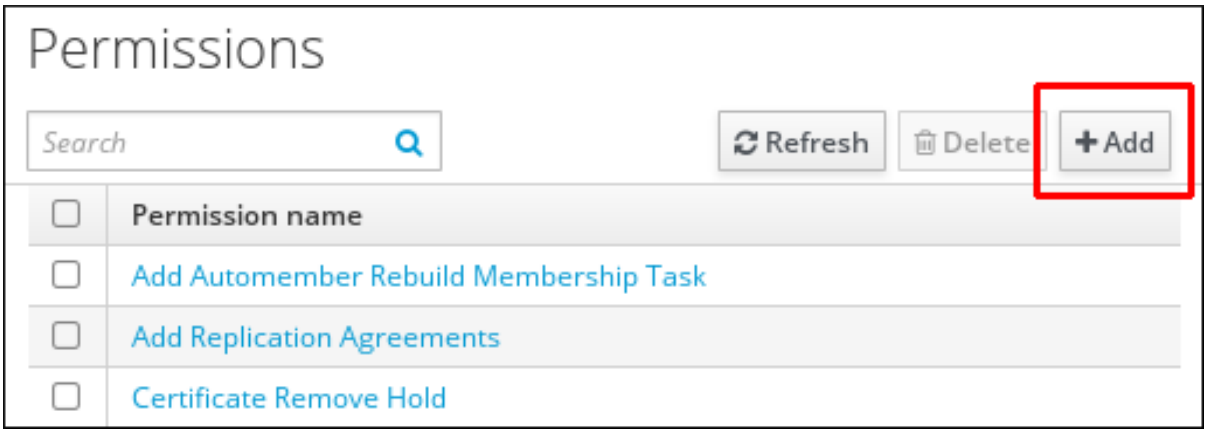
- **IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.**
- **IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.**

절차

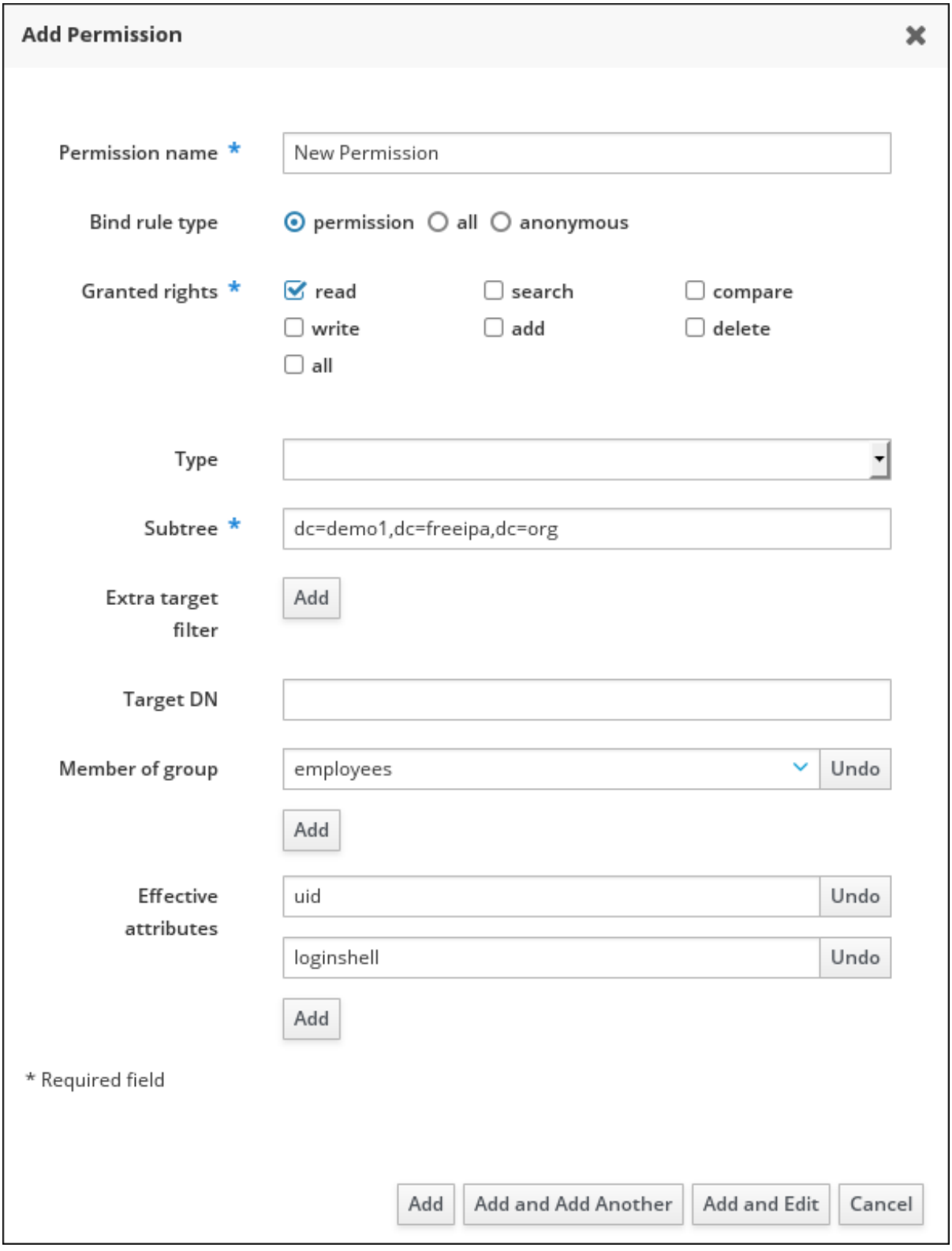
1. **새 권한을 추가하려면 IPA Server 탭에서 Role-Based Access Control (역할 기반 액세스 제어) 하위 메뉴를 열고 Permissions:을 선택합니다.**



2. 권한 목록이 열립니다. 권한 목록 상단에 있는 **Add** 단추를 클릭합니다.



3. **Add Permission**(권한 추가) 양식이 열립니다. 새 권한의 이름을 지정하고 그에 따라 속성을 정의합니다.

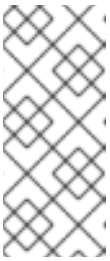


4. **적절한 Bind 규칙 유형을 선택합니다.**

- **권한은 기본 권한 유형으로, 권한 및 역할을 통해 액세스 권한을 부여합니다.**

- **all** 은 권한이 인증된 모든 사용자에게 적용되도록 지정합니다.

- **anonymous** 는 인증되지 않은 사용자를 포함하여 모든 사용자에게 권한이 적용되도록 지정합니다.



참고

기본이 아닌 바인드 규칙 유형의 권한을 권한에 추가할 수 없습니다. 또한 권한이 이미 권한에 있는 권한을 기본이 아닌 바인드 규칙 유형으로 설정할 수 없습니다.

5. 부여된 권리에서 이 권한에 대해 부여할 권한을 선택하십시오 .

6. 권한에 대한 대상 항목을 식별하는 메서드를 정의합니다.

- **type**은 사용자, 호스트 또는 서비스와 같은 항목 유형을 지정합니다. **Type** (유형) 설정에 값을 선택하면 해당 항목 유형에 대해 이 **ACI**를 통해 액세스할 수 있는 모든 가능한 속성 목록이 유효 속성 아래에 표시됩니다. **Type(** 유형 **)**을 정의하면 사전 정의된 값 중 하나로 **Subtree** 및 **Target DN** 이 설정됩니다.

- **subtree** (필수)는 **subtree** 항목을 지정합니다. 그러면 이 **subtree** 항목 아래의 모든 항목을 대상으로 지정합니다. **Subtree** 는 와일드카드 또는 존재하지 않는 도메인 이름**(DN)**을 허용하지 않으므로 기존 하위 트리 항목을 제공합니다. 예**:** **cn=automount,dc=example,dc=com**

- 추가 대상 필터는 **LDAP** 필터를 사용하여 권한이 적용되는 항목을 식별합니다. 필터는 유효한 **LDAP** 필터일 수 있습니다(예**: (!(objectclass=posixgroup))**
IdM은 지정된 필터의 유효성을 자동으로 확인합니다. 잘못된 필터를 입력하면 **IdM**에서 권한을 저장하려고 할 때 이 문제에 대해 경고합니다.

- 대상 **DN** 은 **DN(**도메인 이름**)**을 지정하고 와일드카드를 허용합니다. 예**:** **uid=*,cn=users,cn=accounts,dc=com**

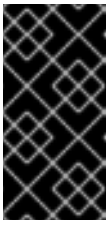
- 그룹의 멤버 는 대상 필터를 지정된 그룹의 멤버로 설정합니다. 필터 설정을 지정하고 **Add** 를 클릭하면 **IdM**에서 필터를 검증합니다. 모든 권한 설정이 올바르면 **IdM**에서 검색을

수행합니다. 일부 권한 설정이 올바르지 않으면 IdM은 어떤 설정이 잘못 설정되었는지 알려주는 메시지를 표시합니다.

7. 권한에 속성을 추가합니다.

- 유형을 설정하는 경우 사용 가능한 ACI 속성 목록에서 유효한 속성을 선택합니다.

- Type (유형)을 사용하지 않은 경우 Effective attributes(유효한 속성) 필드에 수동으로 속성을 추가하여 속성을 추가합니다. 한 번에 단일 속성을 추가합니다. 여러 특성을 추가하려면 Add(추가)를 클릭하여 다른 입력 필드를 추가합니다.



중요

권한에 대한 속성을 설정하지 않으면 권한에는 기본적으로 모든 속성이 포함됩니다.

8. 양식 하단에 Add(추가) 버튼을 사용하여 권한 추가를 완료합니다.

- Add(추가) 버튼을 클릭하여 권한을 저장하고 권한 목록으로 돌아갑니다.

- 또는 권한을 저장하고 Add(추가) 버튼을 클릭하여 동일한 양식에 추가 권한을 추가할 수 있습니다.

- Add and Edit(추가 및 편집) 버튼을 사용하면 새로 만든 권한을 저장하고 계속 편집할 수 있습니다.

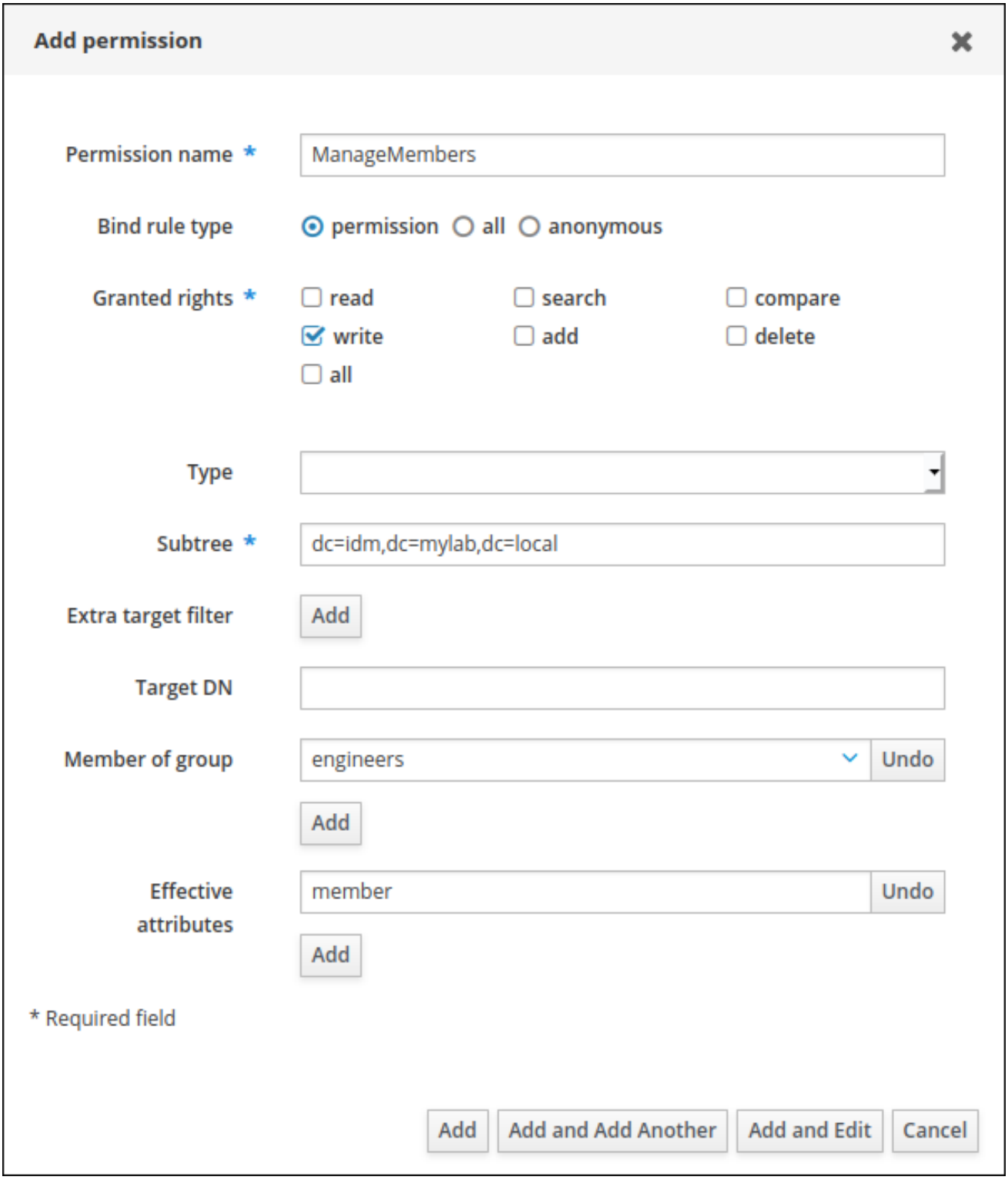
9. *선택 사항:* 권한 목록에서 해당 이름을 클릭하여 권한 설정 페이지를 표시하여 기존 권한의 속성을 편집할 수도 있습니다.

10. *선택 사항:* 기존 권한을 제거해야 하는 경우 목록의 이름 옆에 있는 확인란을 선택한 후 삭제 버튼을 클릭하여 The Remove permissions (권한 제거) 대화 상자를 표시합니다.

참고

기본 관리 권한에 대한 작업은 제한됩니다. 수정할 수 없는 속성은 IdM 웹 UI 에서 비활성화되며 관리 권한을 완전히 삭제할 수 없습니다.
그러나 모든 권한에서 관리되는 권한을 제거하여 권한으로 바인드 유형이 설정된 관리 권한을 효과적으로 비활성화할 수 있습니다.

예를 들어 권한이 있는 경우 **engineer** 그룹의 **member** 속성을 쓰도록 하려면 멤버를 추가하거나 제거할 수 있습니다.



21.3. IDM WEBUI에서 권한 관리

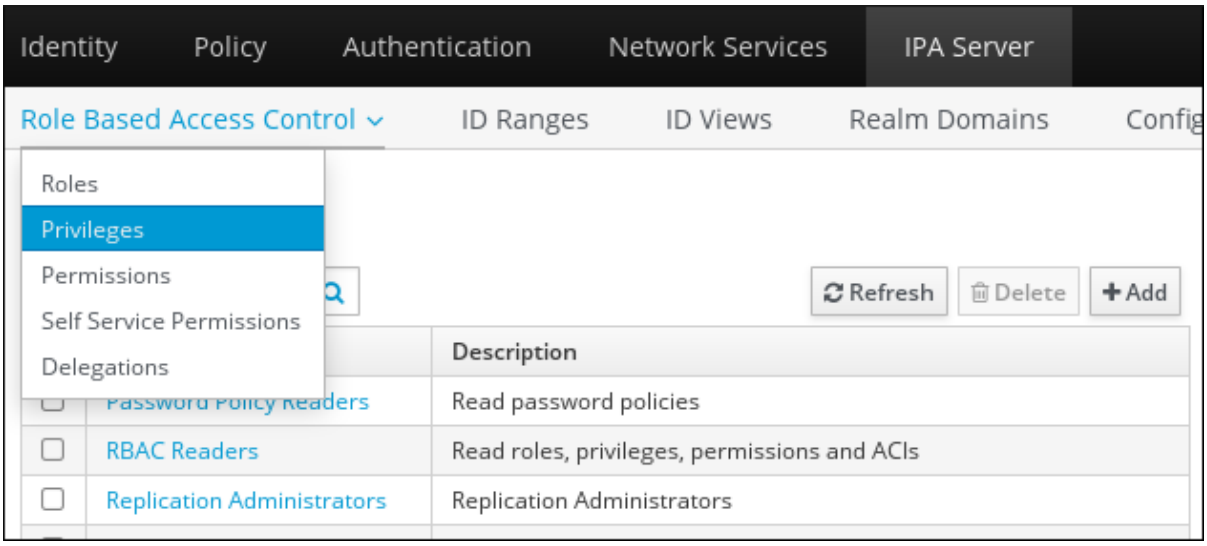
이 섹션에서는 웹 인터페이스(IdM 웹 UI)를 사용하여 IdM에서 권한을 관리하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **IdM** 웹 **UI**에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 **IdM** 웹 **UI** 액세스를 참조하십시오.
- 기존 권한. 권한에 대한 자세한 내용은 **IdM** 웹 **UI**에서 권한 관리를 참조하십시오.

절차

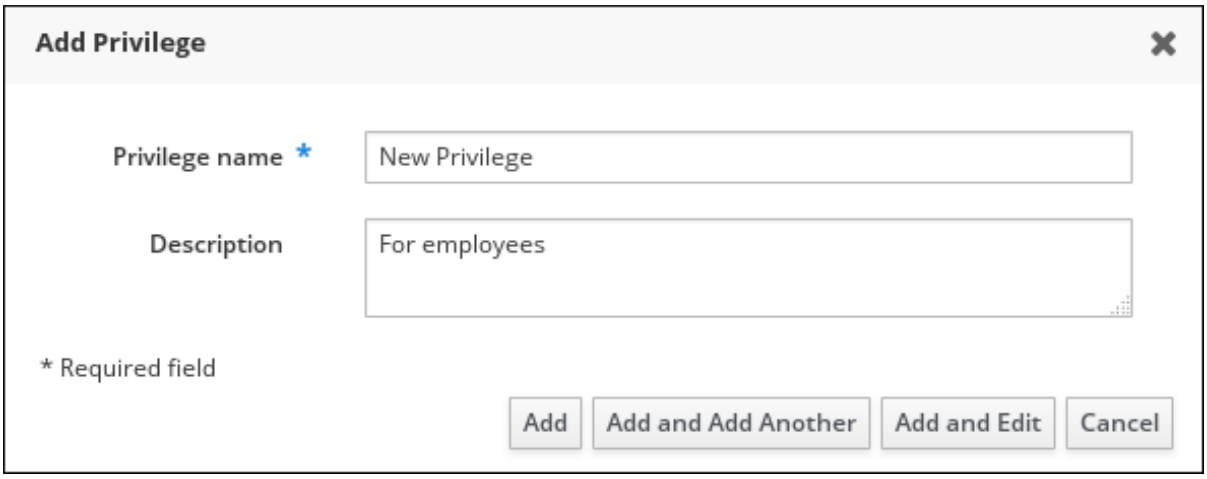
1. 새 권한을 추가하려면 **IPA Server** 탭에서 **Role-Based Access Control** (역할 기반 액세스 제어) 하위 메뉴를 열고 **Privileges:**을 선택합니다.



2. 권한 목록이 열립니다. 권한 목록 상단에 있는 **Add** 단추를 클릭합니다.

3.

Add Privilege(권한 추가) 양식이 열립니다. 권한의 이름과 설명을 입력합니다.



4.

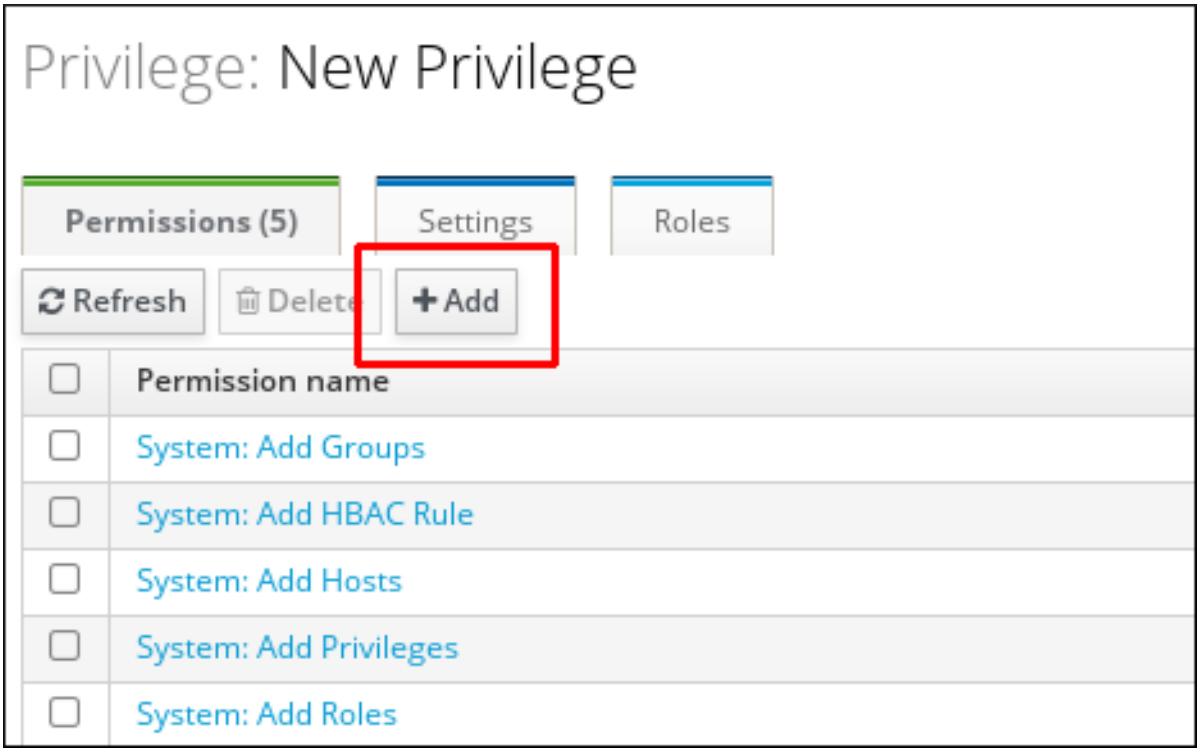
Add and Edit(추가 및 편집) 버튼을 클릭하여 새 권한을 저장하고 권한 구성 페이지로 계속하여 권한을 추가합니다.

5.

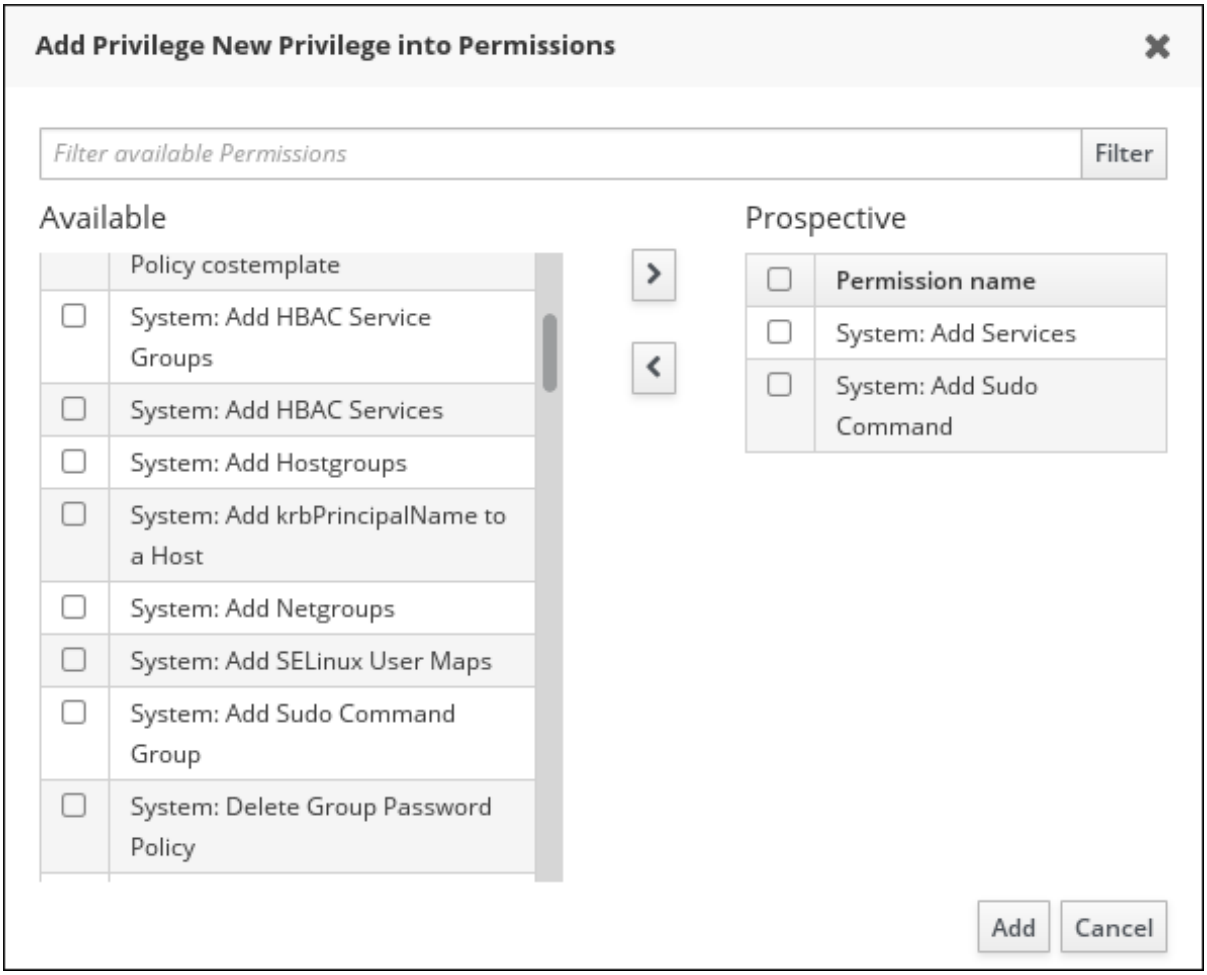
permissions 목록에서 권한 이름을 클릭하여 권한 속성을 편집합니다. 권한 구성 페이지가 열립니다.

6.

Permissions(권한) 탭에는 선택한 권한에 포함된 권한 목록이 표시됩니다. 목록 상단에 있는 Add(추가) 버튼을 클릭하여 권한을 권한에 추가합니다.



7. 추가할 각 권한의 이름 옆에 있는 확인란을 선택하고 **>** 버튼을 사용하여 권한을 **Prospective** 열로 이동합니다.



8. **Add (추가) 버튼을 클릭하여 확인합니다.**

9. ***선택 사항:*** **권한을 제거해야 하는 경우 관련 권한 옆에 있는 확인란을 선택한 후 삭제 버튼을 클릭합니다. 권한 에서 권한 제거 대화 상자가 열립니다.**

10. ***선택 사항:*** **기존 권한을 삭제해야 하는 경우 목록의 이름 옆에 있는 확인란을 선택한 후 삭제 버튼을 클릭합니다. 권한 제거 대화 상자가 열립니다.**

21.4. IDM 웹 UI에서 역할 관리

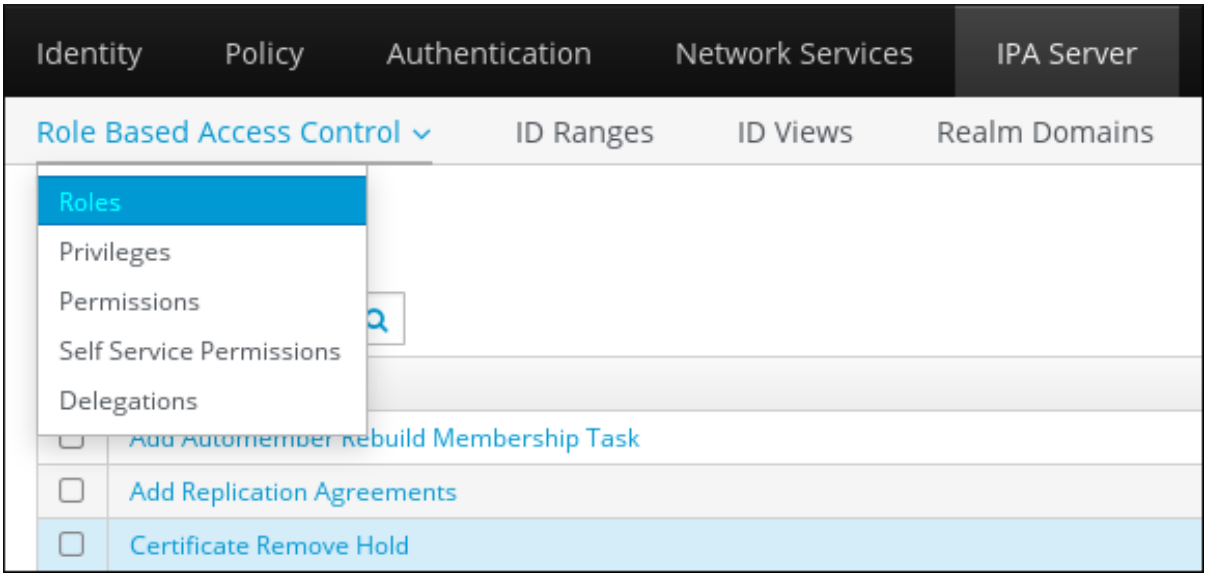
이 섹션에서는 웹 인터페이스(IdM 웹 UI)를 사용하여 IdM(Identity Management)에서 역할을 관리하는 방법을 설명합니다.

사전 요구 사항

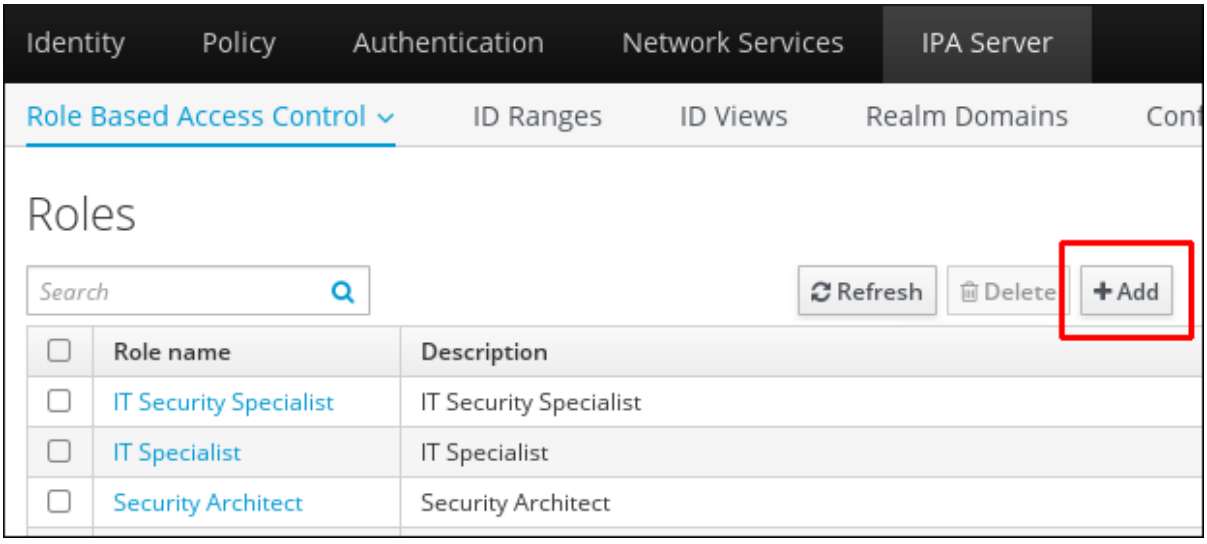
- **IdM** 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- **IdM** 웹 **UI**에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 **IdM** 웹 **UI** 액세스를 참조하십시오.
- 기존 권한. 권한에 대한 자세한 내용은 **IdM** 웹 **UI**에서 권한 관리를 참조하십시오.

절차

1. 새 역할을 추가하려면 **IPA Server** 탭에서 **Role-Based Access Control** (역할 기반 액세스 제어) 하위 메뉴를 열고 **Roles** (역할)를 선택합니다.



2. 역할 목록이 열립니다. 역할 기반 액세스 제어 지침 목록 상단에 있는 **Add** (추가) 버튼을 클릭합니다.

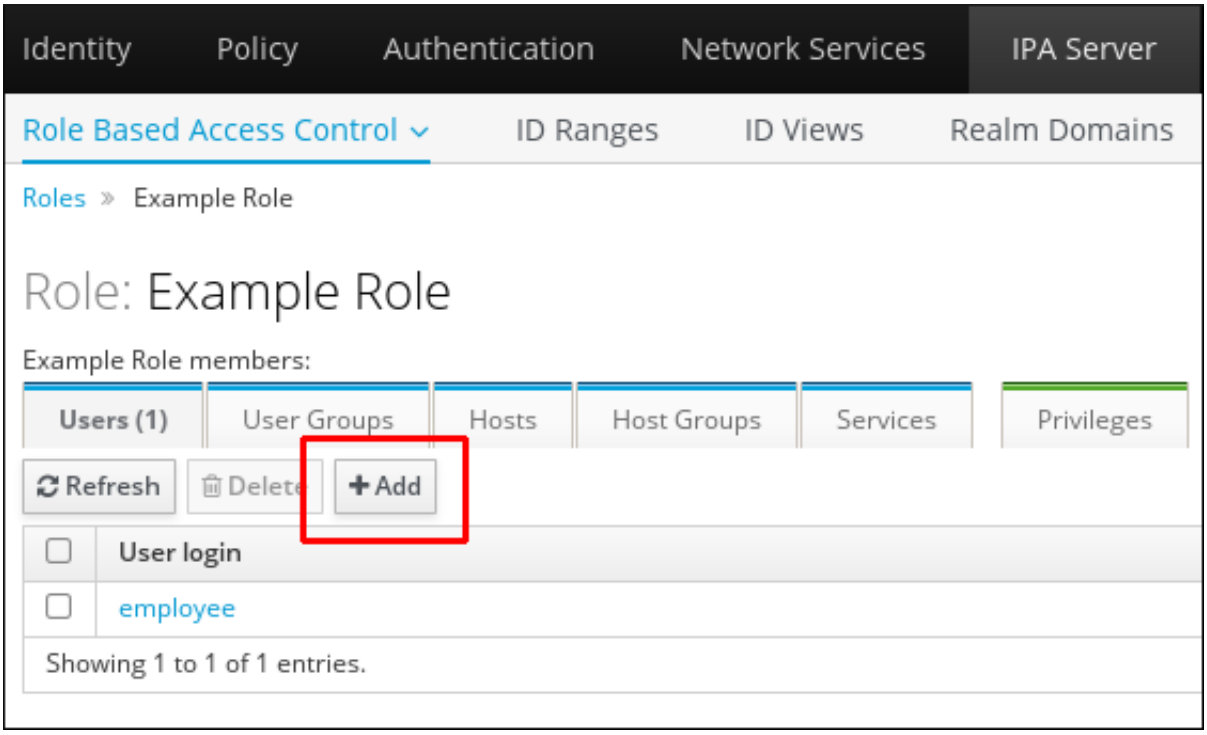


3. **Add Role**(역할 추가) 형식이 열립니다. 역할 이름과 설명을 입력합니다.

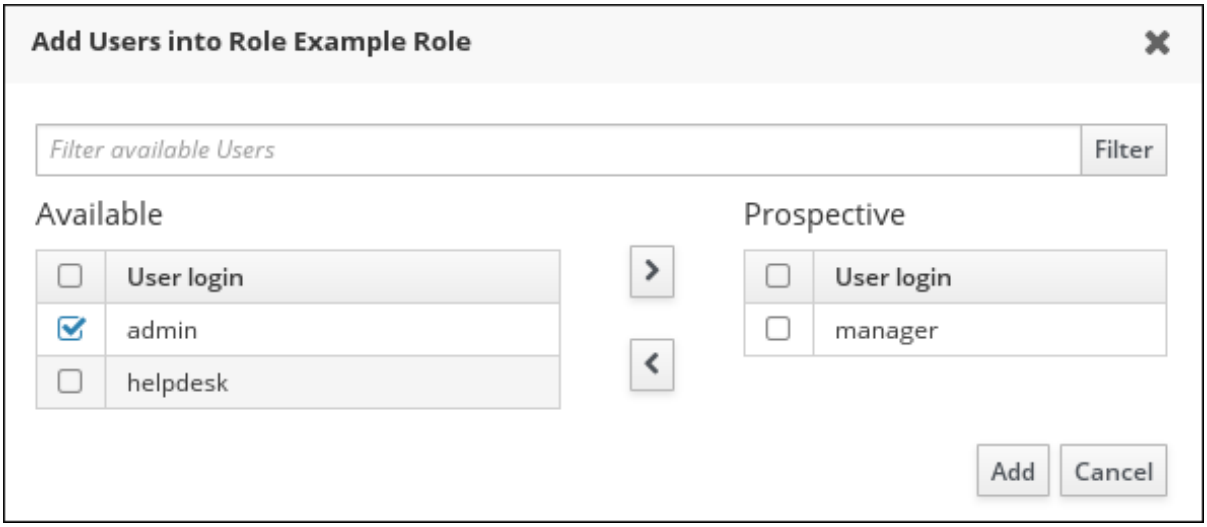
4. **Add and Edit**(추가 및 편집) 버튼을 클릭하여 새 역할을 저장하고 역할 구성 페이지로 이동하여 권한과 사용자를 추가합니다.

5. 역할 목록에서 역할 이름을 클릭하여 역할 속성을 편집합니다. 역할 구성 페이지가 열립니다.

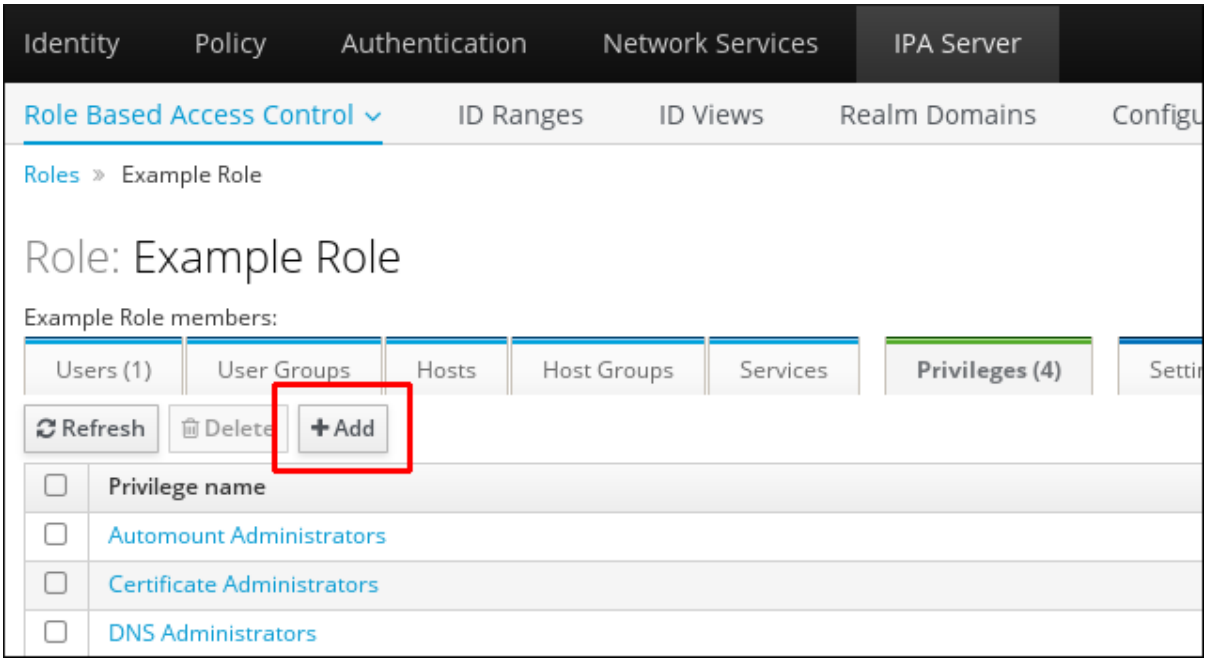
6. 관련 목록 상단에 있는 **Add**(추가) 버튼을 클릭하여 **Users** , **Users Groups** (사용자, 사용자 그룹),**Hosts** (호스트), **Host Groups** (호스트 그룹) 또는 **Services** (서비스) 탭을 사용하여 멤버를 추가합니다.



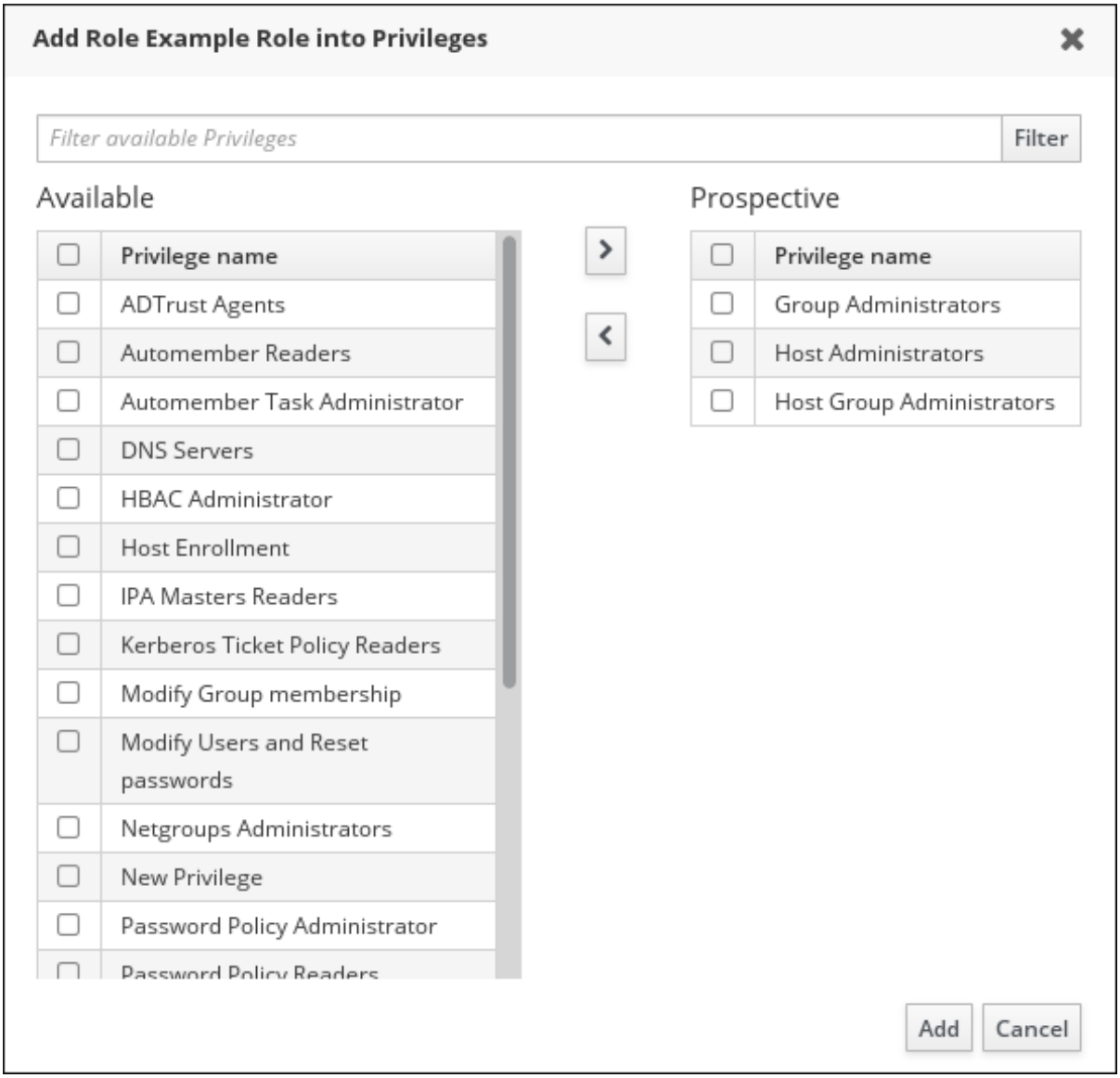
7. 열리는 창에서 왼쪽의 구성원을 선택하고 > 버튼을 사용하여 **Prospective** 열로 이동합니다.



8. **Privileges** (권한) 탭의 상단에서 **Add**(추가)를 클릭합니다.



9. **왼쪽에서 권한을 선택하고 > 버튼을 사용하여 Prospective 열로 이동합니다.**



10. 저장하려면 **Add**(추가) 단추를 클릭합니다.

11. *선택 사항:* 역할에서 권한 또는 멤버를 제거해야 하는 경우 제거할 엔터티 이름 옆에 있는 확인란을 선택한 후 **Delete** (삭제) 단추를 클릭합니다. 대화 상자가 열립니다.

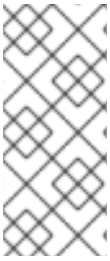
12. *선택 사항:* 기존 역할을 제거해야 하는 경우 목록의 이름 옆에 있는 확인란을 선택한 후 **Delete** (삭제) 버튼을 클릭하여 **Remove roles** (역할 제거) 대화 상자를 표시합니다.

22장. 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 Identity Management 서버 설치를 참조하십시오.
- 인벤토리 빌드 방법을 참조하십시오.

23장. ANSIBLE 플레이북을 사용하여 IDM에서 역할 기반 액세스 제어 관리

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

- 권한은 사용자 추가 또는 삭제, 그룹 수정, 읽기 액세스 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.

23.1. 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

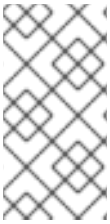
권한에는 다른 권한이 포함될 수 없습니다.

23.2. 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 IdM에서 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

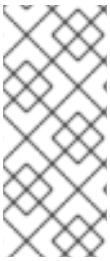
관리 권한을 삭제할 수는 없지만 바인딩 유형을 권한으로 설정하고 모든 권한에서 관리 권한을 제거하면 효과적으로 비활성화됩니다.

관리되는 모든 권한의 이름은 시스템:(예: 시스템): Sudo rule 또는 시스템을 추가합니다: Services 를 수정합니다. 이전 버전의 IdM에서는 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.

23.3. 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.

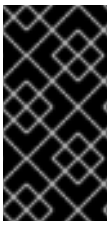
참고

권한에는 다른 권한이 없을 수 있습니다.

23.4. IDM의 역할

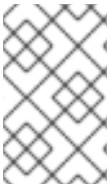
역할은 역할에 지정된 사용자가 보유하는 권한 목록입니다.
실제로 권한은 지정된 하위 수준 작업을 수행할 수 있는 기능을 부여합니다(예: 사용자 항목 생성, 그룹에 항목 추가) 권한은 더 높은 수준의 작업(예: 지정된 그룹에 새 사용자 만들기)에 필요한 하나 이상의 권한

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.

23.5. ID 관리에 사전 정의된 역할

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

표 23.1. ID 관리에서 사전 정의된 역할

Role	권한	설명
등록 관리자	호스트 등록	클라이언트 또는 호스트 등록 담당자
helpdesk	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

23.6. 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** 컨트롤러에 **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 콘텐츠 암호화를 참조하십시오.
- IdM의 역할을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉터리에서 README-role 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리에서 샘플 플레이북을 참조하십시오.

23.7. ANSIBLE을 사용하여 IDM RBAC 역할이 없는지 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 컨트롤러에 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 콘텐츠 암호화를 참조하십시오.
- IdM의 역할을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉토리의 README-role Markdown 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리에서 샘플 플레이북을 참조하십시오.

23.8. ANSIBLE을 사용하여 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 컨트롤러에 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 콘텐츠 암호화를 참조하십시오.
- IdM의 역할을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉토리의 README-role Markdown 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리에서 샘플 플레이북을 참조하십시오.

23.9. 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 컨트롤러에 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 콘텐츠 암호화를 참조하십시오.
- IdM의 역할을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉토리의 README-role Markdown 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리에서 샘플 플레이북을 참조하십시오.

23.10. 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 컨트롤러에 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 콘텐츠 암호화를 참조하십시오.

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

23.11. 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 컨트롤러에 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 콘텐츠 암호화를 참조하십시오.
- IdM의 역할을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉토리의 README-role Markdown 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리에서 샘플 플레이북을 참조하십시오.

23.12. 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 컨트롤러에 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 Vault를 사용하여 콘텐츠 암호화를 참조하십시오.
- IdM의 역할을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉토리의 README-role Markdown 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/iparole 디렉터리에서 샘플 플레이북을 참조하십시오.

24장. ANSIBLE 플레이북을 사용하여 RBAC 권한 관리

RBAC(역할 기반 액세스 제어)는 역할, 권한 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. 특히 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.

24.1. 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 컨트롤러에 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
```

24.2. 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 컨트롤러에 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
```

24.3. 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 컨트롤러에 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
```

24.4. 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 컨트롤러에 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
```

24.5. 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** 컨트롤러에 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
```

24.6. 추가 리소스

- IdM의 권한 참조.
- IdM의 권한을 참조하십시오.
- /usr/share/doc/ansible-freeipa/ 디렉터리에서 사용 가능한 README-privilege 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/ipaprivilege 디렉터리에서 샘플 플레이북을 참조하십시오.

25장. ANSIBLE 플레이북을 사용하여 IDM에서 RBAC 권한 관리

RBAC(역할 기반 액세스 제어)는 역할, 권한 및 권한에 대해 정의된 정책 중립 액세스 제어 메커니즘입니다. 특히 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.

25.1. 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** 컨트롤러에 **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
```

25.2. ANSIBLE을 사용하여 속성이 있는 RBAC 권한이 있는지 확인합니다.

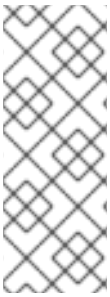
IdM(Identity Management)의 시스템 관리자는 IdM 역할 기반 액세스 제어(RBAC)를 사용자 지정할 수 있습니다.

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** 컨트롤러에 **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

- RHEL 7의 *Linux 도메인 ID, 인증 및 정책 가이드의 사용자 및 그룹 스키마* 를 참조하십시오.

25.3. 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 컨트롤러에 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
```

25.4. 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** 컨트롤러에 **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
```

25.5. 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** 컨트롤러에 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
```

25.6. ANSIBLE을 사용하여 IDM RBAC 권한 이름 변경

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

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

사전 요구 사항

- **IdM** 관리자 암호를 알고 있습니다.
- **Ansible** 컨트롤러에 **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
```

25.7. 추가 리소스

- IdM의 권한을 참조하십시오.
- IdM의 권한 참조.
- /usr/share/doc/ansible-freeipa/ 디렉터리에서 사용 가능한 README-permission 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/ipapermission 디렉터리에서 샘플 플레이북을 참조하십시오.

26장. ID 보기를 사용하여 IDM 클라이언트에서 사용자 속성 값을 재정의

IdM(Identity Management) 사용자가 IdM 서버에 저장된 일부 사용자 또는 그룹 속성(예: 로그인 이름, 홈 디렉터리, 인증 또는 SSH 키)을 재정의하려는 경우 IdM 관리자가 IdM ID 보기를 사용하여 특정 IdM 클라이언트에 대해 이러한 값을 재정의할 수 있습니다. 예를 들어 사용자가 IdM에 로그인하는 데 가장 일반적으로 사용하는 IdM 클라이언트에서 사용자에 대해 다른 홈 디렉터리를 지정할 수 있습니다.

이 장에서는 IdM에 클라이언트로 등록된 호스트에서 IdM 사용자와 관련된 POSIX 특성 값을 재정의하는 방법을 설명합니다. 특히 장에서는 사용자 로그인 이름과 홈 디렉토리를 재정의하는 방법을 설명합니다.

이 장에는 다음 섹션이 포함되어 있습니다.

26.1. ID 보기

IdM(Identity Management)의 ID 보기는 다음 정보를 지정하는 IdM 클라이언트 측 보기입니다.

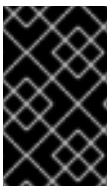
- 중앙에서 정의된 **POSIX** 사용자 또는 그룹 속성의 새 값
- 새 값이 적용되는 클라이언트 호스트 또는 호스트입니다.

ID 보기에는 하나 이상의 재정의가 포함되어 있습니다. 재정의는 중앙에서 정의된 **POSIX** 특성 값을 특정 대체합니다.

IdM 서버에서 중앙에서 **IdM** 클라이언트에 대한 **ID** 보기만 정의할 수 있습니다. **IdM** 클라이언트의 클라이언트 측 덮어쓰기를 로컬로 구성할 수 없습니다.

예를 들어 **ID** 뷰를 사용하여 다음 목표를 달성할 수 있습니다.

- 다양한 환경에 대한 다른 특성 값을 정의합니다. 예를 들어 **IdM** 관리자 또는 다른 **IdM** 사용자가 다른 **IdM** 클라이언트에 다른 홈 디렉토리를 가질 수 있습니다. **/home/encrypted/username** 을 한 **IdM** 클라이언트 및 **/dropbox/username** 에서 이 사용자의 홈 디렉토리로 구성할 수 있습니다. 이 상황에서 **ID** 보기를 사용하는 것은 대체 방법으로 사용할 수 있습니다. 예를 들어 **fallback_homedir,override_homedir** 또는 클라이언트의 **/etc/sssd/sssd.conf** 파일의 기타 홈 디렉터리 변수는 모든 사용자에게 영향을 미칩니다. 예제 절차는 **IdM** 클라이언트에서 **IdM** 사용자 홈 디렉터리를 덮어쓰려면 **ID** 보기 추가를 참조하십시오.
- 이전에 생성한 특성 값을 사용자의 **UID** 재정의와 같은 다른 값으로 바꿉니다. 이 기능은 **LDAP** 측에서 수행하기 어려운 시스템 전체 변경을 수행하려는 경우 유용할 수 있습니다(예: **IdM** 사용자의 **UID 1009**). **IdM** 사용자 **UID**를 생성하는 데 사용되는 **IdM ID** 범위는 **1000** 또는 **10000**으로 시작하지 않습니다. **IdM** 사용자가 모든 **IdM** 클라이언트에서 **UID 1009**를 사용하여 로컬 사용자를 가장하는 이유가 있는 경우 **ID** 보기를 사용하여 **IdM**에서 사용자를 생성할 때 생성된 이 **IdM** 사용자의 **UID**를 재정의할 수 있습니다.



중요

IdM 서버가 아닌 **IdM** 클라이언트에만 **ID** 뷰를 적용할 수 있습니다.

추가 리소스

- **Active Directory** 사용자를 위한 **ID** 보기 사용
- **SSSD** 클라이언트 측 보기

26.2. SSSD 성능에 대한 ID 뷰의 잠재적 영향

ID 뷰를 정의할 때 IdM은 IdM 서버의 SSSD(System Security Services Daemon) 캐시에 원하는 덮어쓰기 값을 배치합니다. 그러면 IdM 클라이언트에서 실행 중인 SSSD가 서버 캐시에서 덮어쓰기 값을 검색합니다.

특정 최적화 및 ID 보기를 동시에 실행할 수 없기 때문에 ID 보기를 적용하면 SSSD(System Security Services Daemon) 성능에 부정적인 영향을 미칠 수 있습니다. 예를 들어 ID 뷰는 SSSD가 서버에서 그룹을 조회하는 프로세스를 최적화하지 못하도록 합니다.

- ID 보기를 사용하면 그룹 이름이 재정의된 경우 SSSD에서 그룹 구성원 이름에서 반환된 모든 멤버를 확인해야 합니다.
- ID 보기가 없으면 SSSD는 그룹 오브젝트의 member 속성에서만 사용자 이름만 수집할 수 있습니다.

이러한 부정적인 영향은 SSSD 캐시가 비어 있거나 캐시를 지운 후 모든 항목을 잘못 만드는 경우 가장 드러납니다.

26.3. ID 보기가 재정의할 수 있는 속성

ID 보기는 사용자 및 그룹 ID 재정의로 구성됩니다. 재정의는 새로운 POSIX 특성 값을 정의합니다.

사용자 및 그룹 ID 재정의는 다음 POSIX 속성에 대한 새 값을 정의할 수 있습니다.

사용자 속성

- 로그인 이름(uid)
- GECOS 항목(gecos)
- UID 번호(uidNumber)
- GID 번호(gidNumber)

- 로그인 쉘 (loginShell)
- 홈 디렉토리(homeDirectory)
- SSH 공개 키(ipaSshPubkey)
- 인증서(userCertificate)

그룹 속성

- 그룹 이름(cn)
- 그룹 GID 번호(gidNumber)

26.4. ID 보기 명령에 대한 도움말 가져오기

IdM CLI(명령줄 인터페이스)에서 IdM(Identity Management) ID 뷰와 관련된 명령에 대한 도움말을 얻을 수 있습니다.

사전 요구 사항

- IdM 사용자의 Kerberos 티켓을 획득했습니다.

절차

- ID 보기 및 재정의를 관리하는 데 사용되는 모든 명령을 표시하려면 다음을 수행합니다.

```
$ ipa help idviews
ID Views

Manage ID Views

IPA allows to override certain properties of users and groups[...]
[...]
Topic commands:
```

```
  idoverridegroup-add          Add a new Group ID override
  idoverridegroup-del          Delete a Group ID override
[...]
```

- 특정 명령에 대한 자세한 도움말을 표시하려면 명령에 **--help** 옵션을 추가합니다.

```
$ ipa idview-add --help
Usage: ipa [global-options] idview-add NAME [options]

Add a new ID View.
Options:
  -h, --help      show this help message and exit
  --desc=STR      Description
[...]
```

26.5. ID 보기를 사용하여 특정 호스트의 IDM 사용자 로그인 이름 덮어쓰기

이 섹션에서는 IdM(Identity Management) 시스템 관리자로서 특정 IdM 사용자와 관련된 POSIX 특성 값을 재정의하는 특정 IdM 클라이언트에 대한 ID 보기를 생성하는 방법을 설명합니다. 이 절차에서는 idm_user라는 IdM 사용자가 user_ 1234 로그인 이름을 사용하여 host1 이라는 IdM 클라이언트에 로그인할 수 있는 ID 보기의 예를 사용합니다.

사전 요구 사항

- IdM 관리자로 로그인했습니다.

절차

1. 새 ID 보기 생성. 예를 들어 example_for_host1 이라는 ID 보기를 생성하려면 다음을 실행합니다.

```
$ ipa idview-add example_for_host1
-------------------------
Added ID View "example_for_host1"
-------------------------
  ID View Name: example_for_host1
```

2. example_for_host1 ID 보기에 사용자 재정의를 추가합니다. 사용자 로그인을 재정의하려면 다음을 수행합니다.

   - ipa idoverrideuser-add 명령을 입력합니다.

- ID 보기의 이름 추가
- 앵커라고도 하는 사용자 이름 추가
- login 옵션을 추가합니다.

```
$ ipa idoverrideuser-add example_for_host1 idm_user --login=user_1234
---------------------------
Added User ID override "idm_user"
---------------------------
  Anchor to override: idm_user
  User login: user_1234
```

사용 가능한 옵션 목록은 ipa idoverrideuser-add --help를 실행합니다.

참고

ipa idoverrideuser-add --certificate 명령은 지정된 ID 보기에 있는 계정의 기존 인증서를 모두 대체합니다. 추가 인증서를 추가하려면 대신 ipa idoverrideuser-add-cert 명령을 사용하십시오.

```
$ ipa idoverrideuser-add-cert example_for_host1 user --
certificate="MIIEATCC..."
```

3. 선택 사항: ipa idoverrideuser-mod 명령을 사용하여 기존 사용자 재정의에 대한 새 속성 값을 지정할 수 있습니다.

4. example_for_host1 을 host1.idm.example.com 호스트에 적용합니다.

```
$ ipa idview-apply example_for_host1 --hosts=host1.idm.example.com
---------------------------
Applied ID View "example_for_host1"
---------------------------
hosts: host1.idm.example.com
-----------------------------------------
Number of hosts the ID View was applied to: 1
-----------------------------------------
```

참고

ipa idview-apply 명령은 **--hostgroups** 옵션도 허용합니다. 옵션은 지정된 호스트 그룹에 속하는 호스트에 **ID** 보기를 적용하지만 **ID** 보기를 호스트 그룹 자체와 연결하지 않습니다. 대신, **--hostgroups** 옵션은 지정된 호스트 그룹의 멤버를 확장하고 **--hosts** 옵션을 각 호스트 그룹에 개별적으로 적용합니다.

즉, 나중에 호스트가 호스트 그룹에 추가되면 **ID** 보기가 새 호스트에 적용되지 않습니다.

5. 새 구성을 **host1.idm.example.com** 시스템에 즉시 적용하려면 다음을 수행합니다.

a. 시스템에 **root**로 **SSH** 연결을 수행합니다.

```
$ ssh root@host1
Password:
```

b. **SSSD** 캐시를 지웁니다.

```
root@host1 ~]# sss_cache -E
```

c. **SSSD** 데몬을 다시 시작합니다.

```
root@host1 ~]# systemctl restart sssd
```

검증 단계

- **user_1234** 의 자격 증명이 있는 경우 이를 사용하여 **host1** 의 **IdM**에 로그인할 수 있습니다.

1. **user_1234** 를 로그인 이름으로 사용하여 **host1** 에 **SSH**로 연결합니다.

```
[root@r8server ~]# ssh user_1234@host1.idm.example.com
Password:

Last login: Sun Jun 21 22:34:25 2020 from 192.168.122.229
[user_1234@host1 ~]$
```

2.

2. 작업 디렉터리를 표시합니다.

```
[user_1234@host1 ~]$ pwd
/home/idm_user/
```

- 또는 **host1** 에 **root** 자격 증명이 있는 경우 이를 사용하여 **id** 명령의 출력을 **idm_user** 및 **user_ 1234** 에 대해 확인할 수 있습니다.

```
[root@host1 ~]# id idm_user
uid=779800003(user_1234) gid=779800003(idm_user) groups=779800003(idm_user)
[root@host1 ~]# user_1234
uid=779800003(user_1234) gid=779800003(idm_user) groups=779800003(idm_user)
```

26.6. IDM ID 보기 수정

IdM(Identity Management)의 ID 보기는 특정 IdM 사용자와 관련된 POSIX 특성 값을 재정의합니다. 이 섹션에서는 기존 ID 보기를 수정하는 방법에 대해 설명합니다. 특히 ID 보기를 수정하여 idm_user 라는 사용자가 host1. idm.example.com IdM 클라이언트에서 /home/user _1234/ 디렉터리를 사용자 홈 디렉터리로 사용하도록 설정하는 방법을 설명합니다.

사전 요구 사항

- host1.idm.example.com 에 대한 루트 액세스 권한이 있습니다.
- 필요한 권한(예: admin)이 있는 사용자로 로그인했습니다.
- host1 IdM 클라이언트에 적용되는 idm_user 에 대해 ID 보기가 구성되어 있습니다.

절차

1. root로 idm_user 가 host1.idm.example.com 에서 사용자 홈 디렉터리로 사용할 디렉터리를 생성합니다.

```
[root@host1 /]# mkdir /home/user_1234/
```

2. 디렉터리의 소유권을 변경합니다.

```
[root@host1 /]# chown idm_user:idm_user /home/user_1234/
```

3.

ID 보기가 현재 적용되는 호스트를 포함하여 ID 보기를 표시합니다. example_for_host1 이라는 ID 보기를 표시하려면 다음을 수행합니다.

```
$ ipa idview-show example_for_host1 --all
  dn: cn=example_for_host1,cn=views,cn=accounts,dc=idm,dc=example,dc=com
  ID View Name: example_for_host1
  User object override: idm_user
  Hosts the view applies to: host1.idm.example.com
  objectclass: ipaIDView, top, nsContainer
```

출력에 ID 보기가 현재 host1.idm.example.com 에 적용됨을 보여줍니다.

4.

example_for_host1 ID 보기의 사용자 재정의를 수정합니다. 사용자 홈 디렉터리를 재정의하려면 다음을 수행합니다.

- ipa idoverrideuser-add 명령을 입력합니다.
- ID 보기의 이름 추가
- 앵커라고도 하는 사용자 이름 추가
- home dir 옵션을 추가합니다.

```
$ ipa idoverrideuser-mod example_for_host1 idm_user --
homedir=/home/user_1234
----------------------------
Modified a User ID override "idm_user"
----------------------------
  Anchor to override: idm_user
  User login: user_1234
  Home directory: /home/user_1234/
```

사용 가능한 옵션 목록은 ipa idoverrideuser-mod --help 를 실행합니다.

5.

새 구성을 host1.idm.example.com 시스템에 즉시 적용하려면 다음을 수행합니다.

a.

시스템에 root로 SSH 연결을 수행합니다.

```
$ ssh root@host1
Password:
```

b. SSSD 캐시를 지웁니다.

```
root@host1 ~]# sss_cache -E
```

c. SSSD 데몬을 다시 시작합니다.

```
root@host1 ~]# systemctl restart sssd
```

검증 단계

1. idm_user 로 host1 에 SSH 를 수행합니다.

```
[root@r8server ~]# ssh idm_user@host1.idm.example.com
Password:

Last login: Sun Jun 21 22:34:25 2020 from 192.168.122.229
[user_1234@host1 ~]$
```

2. 작업 디렉터리를 출력합니다.

```
[user_1234@host1 ~]$ pwd
/home/user_1234/
```

추가 리소스

- 기본 신뢰 뷰를 수정하여 AD 사용자의 글로벌 속성 정의

26.7. IDM 클라이언트에서 IDM 사용자 홈 디렉터리 재정의를 위해 ID 보기 추가

IdM(Identity Management)의 ID 보기는 특정 IdM 사용자와 관련된 POSIX 특성 값을 재정의합니다. 이 섹션에서는 사용자가 /home/ user_ 1234/ 디렉터리를 /home/ idm_user/ 대신 사용자 홈 디렉터리로 사용할 수 있도록 host1 이라는 IdM 클라이언트에서 idm _user 에 적용되는 ID 보기를 생성하는 방법을 설명합니다.

사전 요구 사항

- host1.idm.example.com 에 대한 루트 액세스 권한이 있습니다.
- 필요한 권한(예: admin)이 있는 사용자로 로그인했습니다.

절차

1. root로 idm_user 가 host1.idm.example.com 에서 사용자 홈 디렉터리로 사용할 디렉터리를 생성합니다.

```
[root@host1 /]# mkdir /home/user_1234/
```

2. 디렉터리의 소유권을 변경합니다.

```
[root@host1 /]# chown idm_user:idm_user /home/user_1234/
```

3. ID 뷰를 생성합니다. 예를 들어 example_for_host1 이라는 ID 보기를 생성하려면 다음을 실행합니다.

```
$ ipa idview-add example_for_host1
---------------------------
Added ID View "example_for_host1"
---------------------------
  ID View Name: example_for_host1
```

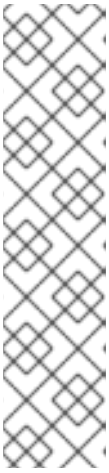
4. example_for_host1 ID 보기에 사용자 재정의를 추가합니다. 사용자 홈 디렉터리를 재정의하려면 다음을 수행합니다.

- ipa idoverrideuser-add 명령을 입력합니다.
- ID 보기의 이름 추가
- 앵커라고도 하는 사용자 이름 추가
- home dir 옵션을 추가합니다.

```
$ ipa idoverrideuser-add example_for_host1 idm_user --homedir=/home/user_1234
---------------------------
Added User ID override "idm_user"
---------------------------
  Anchor to override: idm_user
  Home directory: /home/user_1234/
```

5. example_for_host1 을 host1.idm.example.com 호스트에 적용합니다.

```
$ ipa idview-apply example_for_host1 --hosts=host1.idm.example.com
---------------------------
Applied ID View "example_for_host1"
---------------------------
hosts: host1.idm.example.com
-----------------------------------------
Number of hosts the ID View was applied to: 1
-----------------------------------------
```



참고

ipa idview-apply 명령은 --hostgroups 옵션도 허용합니다. 옵션은 지정된 호스트 그룹에 속하는 호스트에 ID 보기를 적용하지만 ID 보기를 호스트 그룹 자체와 연결하지 않습니다. 대신, --hostgroups 옵션은 지정된 호스트 그룹의 멤버를 확장하고 --hosts 옵션을 각 호스트 그룹에 개별적으로 적용합니다.

즉, 나중에 호스트가 호스트 그룹에 추가되면 ID 보기가 새 호스트에 적용되지 않습니다.

6. 새 구성을 host1.idm.example.com 시스템에 즉시 적용하려면 다음을 수행합니다.

a. 시스템에 root로 SSH 연결을 수행합니다.

```
$ ssh root@host1
Password:
```

b. SSSD 캐시를 지웁니다.

```
root@host1 ~]# sss_cache -E
```

c. SSSD 데몬을 다시 시작합니다.

```
root@host1 ~]# systemctl restart sssd
```

검증 단계

1. idm_user 로 host1 에 SSH 를 수행합니다.

```
[root@r8server ~]# ssh idm_user@host1.idm.example.com
Password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sun Jun 21 22:34:25 2020 from 192.168.122.229
[idm_user@host1 /]$
```

2. 작업 디렉터리를 출력합니다.

```
[idm_user@host1 /]$ pwd
/home/user_1234/
```

추가 리소스

- ID 보기를 사용하여 IdM 클라이언트의 AD 사용자의 기본 신뢰 보기 속성 덮어쓰기

26.8. IDM 호스트 그룹에 ID 보기 적용

ipa idview-apply 명령은 --hostgroups 옵션을 허용합니다. 그러나 옵션은 현재 지정된 호스트 그룹에 속해 있지만 ID 보기를 호스트 그룹 자체와 동적으로 연결하지 않는 호스트에 ID 뷰를 적용하는 일회성 작업 역할을 합니다. hostgroups 옵션은 지정된 호스트 그룹의 멤버를 확장하고 --hosts 옵션을 모든 호스트에 개별적으로 적용합니다.

나중에 호스트 그룹에 새 호스트를 추가하는 경우 --hosts 옵션과 함께 ipa idview-apply 명령을 사용하여 새 호스트에 ID 보기를 수동으로 적용해야 합니다.

마찬가지로 호스트 그룹에서 호스트를 제거해도 제거 후에도 ID 뷰가 여전히 호스트에 할당됩니다. 제거된 호스트에서 ID 보기를 적용 취소하려면 ipa idview-unapply *id_view_name --hosts=name_of_the_removed_host* 명령을 실행해야 합니다.

이 섹션에서는 다음 목표를 달성하는 방법에 대해 설명합니다.

1. 호스트 그룹을 생성하고 호스트를 추가하는 방법.

2. 호스트 그룹에 ID 보기를 적용하는 방법.

3. 새 호스트를 호스트 그룹에 추가하고 ID 보기를 새 호스트에 적용하는 방법.

사전 요구 사항

- 호스트 그룹에 적용할 ID 보기가 IdM에 있는지 확인합니다. 예를 들어 AD 사용자의 GID를 재정의하는 ID 보기를 생성하려면 ID 뷰를 사용하여 IdM 클라이언트의 AD 사용자에 대한 기본 신뢰 보기 속성 덮어쓰기를참조하십시오.

절차

1. 호스트 그룹을 생성하고 호스트를 추가합니다.

   a. 호스트 그룹을 만듭니다. 예를 들어 호스트 그룹을 생성하려면 다음을 실행합니다.

```
[root@server ~]# ipa hostgroup-add --desc="Baltimore hosts" baltimore
--------------------------
Added hostgroup "baltimore"
--------------------------
Host-group: baltimore
Description: Baltimore hosts
```

   b. 호스트 그룹에 호스트를 추가합니다. 예를 들어 host 102 및 host 103 을 hub timore 호스트 그룹에 추가하려면 다음을 수행합니다.

```
[root@server ~]# ipa hostgroup-add-member --hosts={host102,host103} baltimore
Host-group: baltimore
Description: Baltimore hosts
Member hosts: host102.idm.example.com, host103.idm.example.com
-----------------------
Number of members added 2
-----------------------
```

2. 호스트 그룹의 호스트에 ID 보기를 적용합니다. 예를 들어 example_for_host1 ID 보기를 hub timore 호스트 그룹에 적용하려면 다음을 수행합니다.

```
[root@server ~]# ipa idview-apply --hostgroups=baltimore
ID View Name: example_for_host1
-----------------------------------
Applied ID View "example_for_host1"
-----------------------------------
  hosts: host102.idm.example.com, host103.idm.example.com
-------------------------------------------
Number of hosts the ID View was applied to: 2
-------------------------------------------
```

3. 새 호스트를 호스트 그룹에 추가하고 새 호스트에 ID 보기를 적용합니다.

a. 새 호스트를 호스트 그룹에 추가합니다. 예를 들어 somehost.idm.example.com 호스트를 parent timore 호스트 그룹에 추가하려면 다음을 수행합니다.

```
[root@server ~]# ipa hostgroup-add-member --hosts=somehost.idm.example.com
baltimore
  Host-group: baltimore
  Description: Baltimore hosts
  Member hosts:  host102.idm.example.com,
host103.idm.example.com,somehost.idm.example.com
------------------------
Number of members added 1
------------------------
```

b. 선택적으로 ID 보기 정보를 표시합니다. 예를 들어 example_for_host1 ID 보기에 대한 세부 정보를 표시하려면 다음을 수행합니다.

```
[root@server ~]# ipa idview-show example_for_host1 --all
  dn: cn=example_for_host1,cn=views,cn=accounts,dc=idm,dc=example,dc=com
  ID View Name: example_for_host1
[...]
  Hosts the view applies to: host102.idm.example.com, host103.idm.example.com
  objectclass: ipaIDView, top, nsContainer
```

출력에서 ID 보기가 catalog timore 호스트 그룹의 새로 추가된 호스트인 somehost.idm.example.com 에 적용되지 않았음을 보여줍니다.

c. ID 보기를 새 호스트에 적용합니다. 예를 들어 example_for_host1 ID 보기를 somehost.idm.example.com에 적용하려면 다음을 수행합니다.

```
[root@server ~]# ipa idview-apply --host=somehost.idm.example.com
ID View Name: example_for_host1
-----------------------------------
Applied ID View "example_for_host1"
```

```
-------------------------------------
  hosts: somehost.idm.example.com
-------------------------------------------
Number of hosts the ID View was applied to: 1
-------------------------------------------
```

검증 단계

- ID 보기 정보를 다시 표시합니다.

```
[root@server ~]# ipa idview-show example_for_host1 --all
  dn: cn=example_for_host1,cn=views,cn=accounts,dc=idm,dc=example,dc=com
  ID View Name: example_for_host1
[...]
  Hosts the view applies to: host102.idm.example.com, host103.idm.example.com,
somehost.idm.example.com
  objectclass: ipaIDView, top, nsContainer
```

출력에서 ID 보기가 이제 satellite timore 호스트 그룹의 새로 추가된 호스트인 somehost.idm.example.com 에 적용됨을 보여줍니다.

26.9. NIS 도메인을 IDENTITY MANAGEMENT로 마이그레이션

ID 뷰를 사용하여 기존 호스트에 대한 호스트 특정 UID 및 GID를 설정하여 NIS 도메인을 IdM으로 마이그레이션할 때 파일 및 디렉터리의 권한 변경을 방지할 수 있습니다.

사전 요구 사항

- kinit admin 명령을 사용하여 자신을 관리자로 인증했습니다.

절차

1. IdM 도메인에 사용자 및 그룹을 추가합니다.

   a. ipa user-add 명령을 사용하여 사용자를 생성합니다. 자세한 내용은 다음을 참조하십시오. IdM에 사용자 추가

   b. ipa group-add 명령을 사용하여 그룹을 생성합니다. 자세한 내용은 다음을 참조하십시오. IdM에 그룹 추가

2. 사용자 생성 중에 생성된 ID를 덮어씁니다.

   a. ipa idview-add 명령을 사용하여 새 ID 보기를 생성합니다. 자세한 내용은 다음을 참조하십시오. ID 보기 명령에 대한 도움말 가져오기.

   b. 각각 ipa idoverrideuser-add 및 idoverridegroup-add를 사용하여 ID 보기에 사용자 및 그룹의 ID 재정의를 추가합니다.

3. ipa idview-apply 명령을 사용하여 ID 보기를 특정 호스트에 할당합니다.

4. NIS 도메인 해제.

검증

1. 모든 사용자와 그룹이 ID 보기에 올바르게 추가되었는지 확인하려면 ipa idview-show 명령을 사용합니다.

```
$ ipa idview-show example-view
  ID View Name: example-view
  User object overrides: example-user1
  Group object overrides: example-group
```

27장. ACTIVE DIRECTORY 사용자를 위한 ID 보기 사용

ID 보기를 사용하여 IdM-AD 신뢰 환경에서 AD(Active Directory) 사용자의 POSIX 속성에 대한 새 값을 지정할 수 있습니다.

기본적으로 IdM은 기본 신뢰 보기를 모든 AD 사용자에게 적용합니다. 개별 IdM 클라이언트에 추가 ID 뷰를 구성하여 수신하는 POSIX 속성별 사용자를 추가로 조정할 수 있습니다.

27.1. 기본 신뢰 보기의 작동 방식

Default Trust View 는 항상 신뢰 기반 설정의 AD 사용자 및 그룹에 적용되는 기본 ID 보기입니다. ipa-adtrust-install 명령을 사용하여 신뢰를 설정할 때 자동으로 생성되며 삭제할 수 없습니다.

참고

Default Trust View는 IdM 사용자 및 그룹이 아닌 AD 사용자 및 그룹에 대한 덮어쓰기만 허용합니다.

기본 신뢰 보기를 사용하여 AD 사용자 및 그룹에 대해 사용자 지정 POSIX 특성을 정의하여 AD에 정의된 값을 재정의할 수 있습니다.

표 27.1. 기본 신뢰 보기 적용

	AD의 값	기본 신뢰 보기	결과
login	ad_user	ad_user	ad_user
UID	111	222	222
GID	111	(값 없음)	111

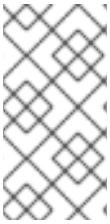
IdM 클라이언트의 기본 신뢰 보기를 재정의하도록 추가 ID 뷰를 구성할 수도 있습니다. IdM은 기본 신뢰 보기 상단에 호스트별 ID 보기의 값을 적용합니다.

- 속성이 호스트별 ID 보기에 정의된 경우 IdM은 이 ID 보기의 값을 적용합니다.

- 속성이 호스트별 ID 보기에 정의되지 않은 경우 IdM은 기본 신뢰 보기의 값을 적용합니다.

표 27.2. 기본 신뢰 보기 상단에 호스트별 ID 보기 적용

	AD의 값	기본 신뢰 보기	호스트별 ID 보기	결과
login	ad_user	ad_user	(값 없음)	ad_user
UID	111	222	333	333
GID	111	(값 없음)	333	333



참고

호스트별 ID 보기만 적용하여 IdM 클라이언트의 기본 신뢰 보기를 덮어쓸 수 있습니다. IdM 서버와 복제본은 항상 Default Trust View의 값을 적용합니다.

추가 리소스

- ID 보기를 사용하여 IdM 클라이언트에서 사용자 속성 값을 재정의

27.2. 기본 신뢰 뷰를 수정하여 AD 사용자의 글로벌 속성 정의

전체 IdM 배포 전반에 걸쳐 AD(Active Directory) 사용자에 대한 POSIX 속성을 재정의하려면 Default Trust View에서 해당 사용자의 항목을 수정하십시오. 이 절차에서는 AD 사용자 ad_user@ad.example.com 의 GID를 732000006으로 설정합니다.

사전 요구 사항

- 신뢰할 수 있는 컨트롤러 또는 신뢰 에이전트인 IdM 서버와 작업하고 있습니다.
- IdM 관리자가 인증되었습니다.

절차

1. IdM 관리자로서 GID 번호를 732000006으로 변경하는 기본 신뢰 보기에서 AD 사용자에 대한 ID를 생성합니다.

```
# ipa idoverrideuser-add 'Default Trust View' ad_user@ad.example.com --
gidnumber=732000006
```

2. 모든 IdM 서버 및 클라이언트의 SSSD 캐시에서 ad_user@ad.example.com 사용자의 항목을 지웁니다. 이렇게 하면 오래된 데이터가 제거되고 새로운 덮어쓰기 값을 적용할 수 있습니다.

```
# sssctl cache-expire -u ad_user@ad.example.com
```

검증

- ad_user@ad.example.com 사용자의 정보를 검색하여 GID가 업데이트된 값을 반영하는지 확인합니다.

```
# id ad_user@ad.example.com
uid=702801456(ad_user@ad.example.com) gid=732000006(ad_admins)
groups=732000006(ad_admins),702800513(domain users@ad.example.com)
```

27.3. ID 보기를 사용하여 IDM 클라이언트의 AD 사용자의 기본 신뢰 보기 속성 덮어쓰기

AD(Active Directory) 사용자에 대한 기본 신뢰 보기에서 일부 POSIX 속성을 재정의할 수 있습니다. 예를 들어 특정 IdM 클라이언트에 AD 사용자에게 다른 GID를 제공해야 할 수 있습니다. ID 보기를 사용하여 AD 사용자에 대한 기본 신뢰 보기의 값을 재정의하여 단일 호스트에 적용할 수 있습니다. 이 절차에서는 host1.idm.example.com IdM 클라이언트에 있는 ad_user@ad.example.com AD 사용자의 GID를 732001337으로 설정하는 방법을 설명합니다.

사전 요구 사항

- host1.idm.example.com IdM 클라이언트에 대한 루트 액세스 권한이 있습니다.
- 필요한 권한이 있는 사용자로 로그인했습니다(예: admin 사용자).

절차

1. ID 뷰를 생성합니다. 예를 들어 example_for_host1 이라는 ID 보기를 생성하려면 다음을 실행합니다.

```
$ ipa idview-add example_for_host1
----------------------------
Added ID View "example_for_host1"
----------------------------
  ID View Name: example_for_host1
```

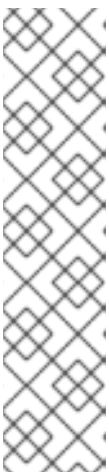
2. example_for_host1 ID 보기에 사용자 재정의를 추가합니다. 사용자의 GID를 덮어쓰려면 다음을 수행합니다.

- ipa idoverrideuser-add 명령을 입력합니다.
- ID 보기의 이름 추가
- 앵커라고도 하는 사용자 이름 추가
- --gidnumber= 옵션을 추가합니다.

```
$ ipa idoverrideuser-add example_for_host1 ad_user@ad.example.com --
gidnumber=732001337
----------------------------
Added User ID override "ad_user@ad.example.com"
----------------------------
  Anchor to override: ad_user@ad.example.com
  GID: 732001337
```

3. example_for_host1 을 host1.idm.example.com IdM 클라이언트에 적용합니다.

```
$ ipa idview-apply example_for_host1 --hosts=host1.idm.example.com
----------------------------
Applied ID View "example_for_host1"
----------------------------
hosts: host1.idm.example.com
-----------------------------------------
Number of hosts the ID View was applied to: 1
-----------------------------------------
```



참고

ipa idview-apply 명령은 --hostgroups 옵션도 허용합니다. 옵션은 지정된 호스트 그룹에 속하는 호스트에 ID 보기를 적용하지만 ID 보기를 호스트 그룹 자체와 연결하지 않습니다. 대신, --hostgroups 옵션은 지정된 호스트 그룹의 멤버를 확장하고 --hosts 옵션을 각 호스트 그룹에 개별적으로 적용합니다.

즉, 나중에 호스트가 호스트 그룹에 추가되면 ID 보기가 새 호스트에 적용되지 않습니다.

4. host1.idm.example.com IdM 클라이언트의 SSSD 캐시에서 ad_user@ad.example.com 사용자의 항목을 지웁니다. 이렇게 하면 오래된 데이터가 제거되고 새로운 덮어쓰기 값을 적용할 수 있습니다.

```
[root@host1 ~]# sssctl cache-expire -u ad_user@ad.example.com
```

검증 단계

1. SSH to host1 as ad_user@ad.example.com:

```
[root@r8server ~]# ssh ad_user@ad.example.com@host1.idm.example.com
```

2. ad_user@ad.example.com 사용자의 정보를 검색하여 GID가 업데이트된 값을 반영하는지 확인합니다.

```
[ad_user@ad.example.com@host1 ~]$ id ad_user@ad.example.com
uid=702801456(ad_user@ad.example.com) gid=732001337(admins2)
groups=732001337(admins2),702800513(domain users@ad.example.com)
```

27.4. IDM 호스트 그룹에 ID 보기 적용

ipa idview-apply 명령은 --hostgroups 옵션을 허용합니다. 그러나 옵션은 현재 지정된 호스트 그룹에 속해 있지만 ID 보기를 호스트 그룹 자체와 동적으로 연결하지 않는 호스트에 ID 뷰를 적용하는 일회성 작업 역할을 합니다. hostgroups 옵션은 지정된 호스트 그룹의 멤버를 확장하고 --hosts 옵션을 모든 호스트에 개별적으로 적용합니다.

나중에 호스트 그룹에 새 호스트를 추가하는 경우 --hosts 옵션과 함께 ipa idview-apply 명령을 사용하여 새 호스트에 ID 보기를 수동으로 적용해야 합니다.

마찬가지로 호스트 그룹에서 호스트를 제거해도 제거 후에도 ID 뷰가 여전히 호스트에 할당됩니다. 제거된 호스트에서 ID 보기를 적용 취소하려면 ipa idview-unapply *id_view_name --hosts=name_of_the_removed_host* 명령을 실행해야 합니다.

이 섹션에서는 다음 목표를 달성하는 방법에 대해 설명합니다.

1. 호스트 그룹을 생성하고 호스트를 추가하는 방법.

2. 호스트 그룹에 ID 보기를 적용하는 방법.

3. 새 호스트를 호스트 그룹에 추가하고 ID 보기를 새 호스트에 적용하는 방법.

사전 요구 사항

- 호스트 그룹에 적용할 ID 보기가 IdM에 있는지 확인합니다. 예를 들어 AD 사용자의 GID를 재정의하는 ID 보기를 생성하려면 ID 뷰를 사용하여 IdM 클라이언트의 AD 사용자에 대한 기본 신뢰 보기 속성 덮어쓰기를참조하십시오.

절차

1. 호스트 그룹을 생성하고 호스트를 추가합니다.

   a. 호스트 그룹을 만듭니다. 예를 들어 호스트 그룹을 생성하려면 다음을 실행합니다.

```
[root@server ~]# ipa hostgroup-add --desc="Baltimore hosts" baltimore
-------------------------
Added hostgroup "baltimore"
-------------------------
Host-group: baltimore
Description: Baltimore hosts
```

   b. 호스트 그룹에 호스트를 추가합니다. 예를 들어 host 102 및 host 103 을 hub timore 호스트 그룹에 추가하려면 다음을 수행합니다.

```
[root@server ~]# ipa hostgroup-add-member --hosts={host102,host103} baltimore
Host-group: baltimore
Description: Baltimore hosts
Member hosts: host102.idm.example.com, host103.idm.example.com
-----------------------
Number of members added 2
-----------------------
```

2. 호스트 그룹의 호스트에 ID 보기를 적용합니다. 예를 들어 example_for_host1 ID 보기를 hub timore 호스트 그룹에 적용하려면 다음을 수행합니다.

```
[root@server ~]# ipa idview-apply --hostgroups=baltimore
ID View Name: example_for_host1
---------------------------------------
Applied ID View "example_for_host1"
```

```
----------------------------------------
 hosts: host102.idm.example.com, host103.idm.example.com
----------------------------------------
Number of hosts the ID View was applied to: 2
----------------------------------------
```

3. 새 호스트를 호스트 그룹에 추가하고 새 호스트에 ID 보기를 적용합니다.

a. 새 호스트를 호스트 그룹에 추가합니다. 예를 들어 somehost.idm.example.com 호스트를 parent timore 호스트 그룹에 추가하려면 다음을 수행합니다.

```
[root@server ~]# ipa hostgroup-add-member --hosts=somehost.idm.example.com
baltimore
  Host-group: baltimore
  Description: Baltimore hosts
  Member hosts:  host102.idm.example.com,
host103.idm.example.com,somehost.idm.example.com
-----------------------
Number of members added 1
-----------------------
```

b. 선택적으로 ID 보기 정보를 표시합니다. 예를 들어 example_for_host1 ID 보기에 대한 세부 정보를 표시하려면 다음을 수행합니다.

```
[root@server ~]# ipa idview-show example_for_host1 --all
  dn: cn=example_for_host1,cn=views,cn=accounts,dc=idm,dc=example,dc=com
  ID View Name: example_for_host1
[...]
  Hosts the view applies to: host102.idm.example.com, host103.idm.example.com
  objectclass: ipaIDView, top, nsContainer
```

출력에서 ID 보기가 catalog timore 호스트 그룹의 새로 추가된 호스트인 somehost.idm.example.com 에 적용되지 않았음을 보여줍니다.

c. ID 보기를 새 호스트에 적용합니다. 예를 들어 example_for_host1 ID 보기를 somehost.idm.example.com에 적용하려면 다음을 수행합니다.

```
[root@server ~]# ipa idview-apply --host=somehost.idm.example.com
ID View Name: example_for_host1
----------------------------------------
Applied ID View "example_for_host1"
----------------------------------------
  hosts: somehost.idm.example.com
```

```
-----------------------------------------
Number of hosts the ID View was applied to: 1
-----------------------------------------
```

검증 단계

- ID 보기 정보를 다시 표시합니다.

```
[root@server ~]# ipa idview-show example_for_host1 --all
  dn: cn=example_for_host1,cn=views,cn=accounts,dc=idm,dc=example,dc=com
  ID View Name: example_for_host1
[...]
  Hosts the view applies to: host102.idm.example.com, host103.idm.example.com,
somehost.idm.example.com
  objectclass: ipaIDView, top, nsContainer
```

출력에서 ID 보기가 이제 **satellite timore** 호스트 그룹의 새로 추가된 호스트인 **somehost.idm.example.com** 에 적용됨을 보여줍니다.

28장. 수동으로 ID 범위 조정

IdM 서버는 고유한 사용자 ID(UID) 및 그룹 ID(GID) 번호를 생성합니다. 복제본에 다양한 ID 범위를 만들고 할당하면 동일한 ID 번호를 생성하지 않도록 합니다. 기본적으로 이 프로세스는 자동입니다. 그러나 IdM 서버를 설치하는 동안 IdM ID 범위를 수동으로 조정하거나 복제본의 DNA ID 범위를 수동으로 정의할 수 있습니다.

28.1. ID 범위

ID 번호는 *ID 범위*로 구분됩니다. 개별 서버 및 복제본에 대해 별도의 숫자 범위를 유지하면 항목에 대해 발행된 ID 번호를 다른 서버 또는 복제본의 다른 항목에서 이미 사용할 가능성이 제거됩니다.

두 가지 유형의 ID 범위가 있습니다.

- 첫 번째 서버를 설치하는 동안 할당되는 IdM *ID 범위*. 이 범위는 생성된 후에는 수정할 수 없습니다. 그러나 원래 ID 범위 외에 새 IdM ID 범위를 생성할 수 있습니다. 자세한 내용은 자동 ID 범위 할당 및 새 IdM ID 범위 추가를 참조하십시오.

- 사용자가 수정할 수 있는 DNA(*Distributed Numeric Assignment*) ID 범위입니다. 이러한 값은 기존 IdM ID 범위에 적합해야 합니다. 자세한 내용은 DNA ID 범위가 수동으로 할당을 참조하십시오.

  복제본에 다음 DNA ID 범위가 할당될 수도 있습니다. 복제본은 현재 범위의 ID가 부족할 때 다음 범위를 사용합니다. 복제본이 삭제되면 다음 범위는 자동으로 할당되지 않으며 수동으로 할당해야 합니다.

도메인의 백엔드 389 Directory Server 인스턴스의 일부로 DNA 플러그인에 의해 서버와 복제본 간에 범위가 업데이트되고 공유됩니다.

DNA 범위 정의는 다음 두 가지 속성으로 설정됩니다.

- 서버의 사용 가능한 다음 번호: DNA 범위의 낮은 끝

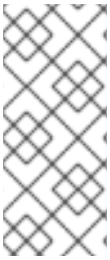
- 범위 크기: DNA 범위의 ID 수

초기 하단 범위는 플러그인 인스턴스 구성 중에 설정됩니다. 그런 다음 플러그인은 하단 값을 업데이트 합니다. 사용 가능한 숫자를 범위로 분할하면 서버는 서로 겹치지 않고 번호를 계속 할당할 수 있습니다.

28.2. 자동 ID 범위 할당

IdM ID 범위

기본적으로 IdM ID 범위는 IdM 서버 설치 중에 자동으로 할당됩니다. ipa-server-install 명령은 총 10,000개 범위의 ID 범위를 임의로 선택하고 할당합니다. 이와 같이 임의의 범위를 선택하면 향후 두 개의 별도의 IdM 도메인을 병합하기로 결정한 경우 ID 충돌 가능성이 크게 줄어듭니다.



참고

IdM ID 범위를 생성한 후에는 수정할 수 없습니다. DNA ID 범위 할당에 설명된 명령을 사용하여DNA(Distributed Numeric Assignment) 범위만 수동으로 조정할 수 있습니다. IdM ID 범위와 일치하는 DNA 범위가 설치 중에 자동으로 생성됩니다.

DNA ID 범위

IdM 서버가 1개 설치되어 있으면 전체 DNA ID 범위를 제어합니다. 새 복제본을 설치하고 복제본이 고유한 DNA ID 범위를 요청하면 서버 분할의 초기 ID 범위가 분리되고 서버와 복제본 간에 배포됩니다. 복제본은 초기 서버에서 사용할 수 있는 나머지 DNA ID 범위의 절반을 수신합니다. 그런 다음 서버 및 복제본은 새 사용자 또는 그룹 항목에 대해 원래 ID 범위의 각 부분을 사용합니다. 또한 복제본이 할당된 ID 범위를 제거하고 100개 미만의 ID를 유지하는 경우에도 복제본은 사용 가능한 다른 서버에 연결하여 새 DNA ID 범위를 요청합니다.

중요

복제본을 설치하면 ID 범위가 즉시 수신 되지 않습니다. 복제본은 DNA 플러그인이 처음 사용되는 경우(예: 사용자를 처음 추가할 때) ID 범위를 수신합니다.

복제본에서 DNA ID 범위를 요청하기 전에 초기 서버가 작동을 중지하면 복제본이 서버에 연결하여 ID 범위를 요청할 수 없습니다. 복제본에 새 사용자를 추가하려고 하면 실패합니다. 이러한 상황에서는 비활성화된 서버에 할당된 ID 범위를 확인하고 복제본에 ID 범위를 수동으로 할당할 수 있습니다.

28.3. 서버 설치 중 IDM ID 범위 수동 할당

임의로 할당하는 대신 기본 동작을 재정의하고 IdM ID 범위를 수동으로 설정할 수 있습니다.

중요

UID 값이 1000 이하인 ID 범위를 설정하지 마십시오. 이러한 값은 시스템용으로 예약되어 있습니다. 또한 0 값을 포함할 ID 범위를 설정하지 마십시오. SSSD 서비스에서 0 ID 값을 처리하지 않습니다.

절차

- ipa-server-install 과 함께 다음 두 옵션을 사용하여 서버 설치 중에 IdM ID 범위를 수동으로 정의할 수 있습니다.
  - --idstart 는 UID 및 GID 번호의 시작 값을 제공합니다.
  - --idmax 는 최대 UID 및 GID 번호를 제공합니다. 기본적으로 값은 --idstart 시작 값과 199,999입니다.

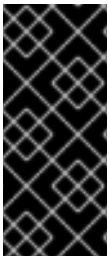
검증 단계

- ID 범위가 올바르게 할당되었는지 확인하려면 ipa idrange-find 명령을 사용하여 할당된 IdM ID 범위를 표시할 수 있습니다.

```
# ipa idrange-find
----------------
1 range matched
----------------
  Range name: IDM.EXAMPLE.COM_id_range
  First Posix ID of the range: 882200000
  Number of IDs in the range: 200000
  Range type: local domain range
----------------------------
Number of entries returned 1
----------------------------
```

28.4. 새 IDM ID 범위 추가

경우에 따라 원래 ID 외에도 새 IdM ID 범위를 생성할 수 있습니다(예: 복제본의 ID가 부족하고 원래 IdM ID 범위가 고갈된 경우).



중요

새 IdM ID 범위를 추가하면 새 DNA ID 범위가 자동으로 생성되지 않습니다. 필요에 따라 복제본에 새 DNA ID 범위를 수동으로 할당해야 합니다. 이 작업을 수행하는 방법에 대한 자세한 내용은 DNA ID 범위를 수동으로 할당을 참조하십시오.

절차

1. 새 IdM ID 범위를 생성하려면 ipa idrange-add 명령을 사용합니다. 새 범위 이름, 범위의 첫 번째 ID 번호 및 범위 크기를 지정해야 합니다.

```
# ipa idrange-add IDM.EXAMPLE.COM_new_range --base-id=1000000 --range-
size=200000
----------------------------------------
Added ID range "IDM.EXAMPLE.COM_new_range"
----------------------------------------
  Range name: IDM.EXAMPLE.COM_new_range
  First Posix ID of the range: 1000000
  Number of IDs in the range: 200000
  Range type: local domain range
```

2. 선택 사항: ID 범위를 즉시 업데이트합니다.

   a. SSSD(System Security Services Daemon) 캐시를 지웁니다.

   ```
   # sss_cache -E
   ```

   b. SSSD 데몬을 다시 시작합니다.

   ```
   # systemctl restart sssd
   ```

참고

SSSD 캐시를 지우지 않고 서비스를 다시 시작하지 않으면 SSSD에서 도메인 목록 및 IdM 서버에 저장된 기타 구성 데이터를 업데이트할 때만 새 ID 범위를 탐지합니다.

검증 단계

- ipa idrange-find 명령을 사용하여 새 범위가 올바르게 설정되었는지 확인할 수 있습니다.

```
# ipa idrange-find
----------------
2 ranges matched
----------------
  Range name: IDM.EXAMPLE.COM_id_range
  First Posix ID of the range: 882200000
  Number of IDs in the range: 200000
  Range type: local domain range

  Range name: IDM.EXAMPLE.COM_new_range
  First Posix ID of the range: 1000000
  Number of IDs in the range: 200000
  Range type: local domain range
----------------------------
Number of entries returned 2
----------------------------
```

28.5. IDM ID 범위에서 보안 및 상대 식별자의 역할

IdM(Identity Management) ID 범위는 여러 매개변수로 정의됩니다.

- 범위 이름
- 범위의 첫 번째 POSIX ID
- 범위 크기: 범위의 ID 수
- 해당 RID 범위의 첫 번째 상대 식별자 (RID)
- 보조 RID 범위의 첫 번째 RID

ipa idrange-show 명령을 사용하여 이러한 값을 볼 수 있습니다.

```
$ ipa idrange-show IDM.EXAMPLE.COM_id_range
  Range name: IDM.EXAMPLE.COM_id_range
  First Posix ID of the range: 196600000
  Number of IDs in the range: 200000
  First RID of the corresponding RID range: 1000
  First RID of the secondary RID range: 1000000
  Range type: local domain range
```

보안 식별자

로컬 도메인의 ID 범위 데이터는 IdM 서버에서 IdM 사용자 및 그룹에 고유한 보안 식별자 (SID)를 할당하는 데 사용됩니다. STS는 사용자 및 그룹 개체에 저장됩니다. 사용자의 SID는 다음으로 구성됩니다.

- 도메인 SID
- 사용자 상대 식별자 (RID)는 도메인 STS에 추가된 네 자리 32비트 값입니다.

예를 들어 도메인 SID가 S-1-5-21-123-456-789이고 이 도메인의 사용자 RID가 1008인 경우, 사용자에게 S-1-5-21-123-456-789-1008의 HEAD가 있습니다.

상대 식별자

RID 자체는 다음과 같은 방식으로 계산됩니다.

범위의 첫 번째 POSIX ID를 사용자의 POSIX UID에서 빼고 해당 RID 범위의 첫 번째 RID를 결과에 추가합니다. 예를 들어 *idmuser* 의 UID가 196600008인 경우 첫 번째 POSIX ID는 jenkinsfile600000이고 첫 번째 RID는 1000이고 *idmuser* 's RID는 1008입니다.

참고

사용자의 RID 알고리즘을 계산하는 알고리즘은 지정된 POSIX ID가 해당 RID를 계산하기 전에 할당된 ID 범위에 속하는지 확인합니다. 예를 들어 첫 번째 ID가 jenkinsfile600000이고 범위 크기가 200000인 경우, 이 범위의 POSIX ID는 ID 범위 외부에 있고 알고리즘은 RID를 계산하지 않습니다.

보조 상대 식별자

IdM에서 POSIX UID는 POSIX GID와 같을 수 있습니다. 즉, *idmuser* 가 196600008 UID가 있는 경우 GID가 196600008인 새 *idmgroup* 그룹을 계속 생성할 수 있습니다.

그러나 STS는 사용자 또는 그룹 한 개만 정의할 수 있습니다. *idmuser* 용으로 이미 생성된 S-1-5-21-123-456-789-1008의 SID는 *idmgroup* 과 공유할 수 없습니다. *idmgroup* 에 대해 대체 STS를 생성해야 합니다.

IdM은 보조 상대 식별자 (또는 보조 RID)를 사용하여 conflict replacess가 발생하지 않습니다. 이 보조 RID는 다음과 같이 구성됩니다.

- 2차 RID 기반
- 범위 크기(기본 범위 크기와 기본적으로 동일)

위의 예에서 보조 RID 베이스는 1000000으로 설정됩니다. 새로 생성된 *idmgroup* 의 RID를 계산하려면 : 사용자의 POSIX UID에서 범위의 첫 번째 POSIX ID를 제거하고 보조 RID 범위의 첫 번째 RID를 결과에 추가합니다. *idmgroup* 은 RID 1000008이 할당됩니다. 그 결과 *idmgroup* 의 SID는 S-1-5-21-123-456-789-1000008입니다.

IdM은 보조 RID를 사용하여 사용자 또는 그룹 오브젝트가 이전에 수동으로 설정된 POSIX ID를 사용하여 생성한 경우에만 STS를 계산합니다. 그렇지 않으면 자동 할당으로 동일한 ID를 두 번 할당하지 않습니다.

28.6. 신뢰할 수 있는 AD를 제거한 후 ID 범위 제거

IdM 및 AD(Active Directory) 환경 간에 신뢰가 제거된 경우 연결된 ID 범위를 제거할 수 있습니다.

주의

신뢰할 수 있는 도메인과 연결된 ID 범위에 할당된 ID는 여전히 IdM에 등록된 시스템의 파일 및 디렉터리 소유권에 사용될 수 있습니다.

제거한 AD 트러스트에 해당하는 ID 범위를 제거하면 AD 사용자가 소유한 파일과 디렉터리의 소유권을 확인할 수 없습니다.

사전 요구 사항

- AD 환경에 대한 신뢰성을 제거했습니다.

절차

1. 현재 사용 중인 모든 ID 범위를 표시합니다.

```
[root@server ~]# ipa idrange-find
```

2. 제거한 신뢰와 연결된 ID 범위의 이름을 식별합니다. ID 범위의 첫 번째 부분은 신뢰의 이름입니다(예: AD.EXAMPLE.COM_id_range).

3. 범위를 제거합니다.

```
[root@server ~]# ipa idrange-del AD.EXAMPLE.COM_id_range
```

4. SSSD 서비스를 다시 시작하여 제거한 ID 범위에 대한 참조를 제거합니다.

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

추가 리소스

- 명령줄을 사용하여 신뢰 제거를 참조하십시오.
- IdM 웹 UI를 사용하여 신뢰 제거를 참조하십시오.

28.7. 현재 할당된 DNA ID 범위 표시

현재 활성화된 DNA(Distributed Numeric Assignment) ID 범위와 서버에 모두 DNA(Distributed Numeric Assignment) ID 범위를 할당한 경우 해당 다음 DNA 범위를 표시할 수 있습니다.

절차

- 토폴로지의 서버에 구성된 DNA ID 범위를 표시하려면 다음 명령을 사용합니다.
  - ipa-replica-manage dnarange-show 는 모든 서버에 설정된 현재 DNA ID 범위를 표시하거나 지정된 서버에만 서버를 지정하는 경우 다음을 수행합니다.

```
# ipa-replica-manage dnarange-show
serverA.example.com: 1001-1500
serverB.example.com: 1501-2000
```

```
serverC.example.com: No range set

# ipa-replica-manage dnarange-show serverA.example.com
serverA.example.com: 1001-1500
```

- **ipa-replica-manage dnanextrange-show** 는 현재 모든 서버에 설정된 다음 **DNA ID** 범위를 표시하거나 지정된 서버에만 서버를 지정하는 경우 다음을 표시합니다.

```
# ipa-replica-manage dnanextrange-show
serverA.example.com: 2001-2500
serverB.example.com: No on-deck range set
serverC.example.com: No on-deck range set

# ipa-replica-manage dnanextrange-show serverA.example.com
serverA.example.com: 2001-2500
```

28.8. 수동 ID 범위 할당

특정 상황에서는 다음과 같이 **DNA(Distributed Numeric Assignment) ID** 범위를 수동으로 할당해야 합니다.

- 복제본에 **ID**가 부족하고 **IdM ID** 범위가 제거되었습니다.

  복제본은 할당된 **DNA ID** 범위를 고갈시키고 **IdM** 범위에서 사용 가능한 **ID**를 더 이상 사용할 수 없기 때문에 추가 **ID**를 요청하지 못했습니다.

  이러한 상황을 해결하려면 복제본에 할당된 **DNA ID** 범위를 확장합니다. 다음 두 가지 방법으로 이 작업을 수행할 수 있습니다.

  - 다른 복제본에 할당된 **DNA ID** 범위를 단축한 다음 새로 사용 가능한 값을 고갈된 복제본에 할당합니다.
  - 새 **IdM ID** 범위를 생성한 다음 이 **IdM** 범위 내에서 복제본의 새 **DNA ID** 범위를 설정합니다.

    새 **IdM ID** 범위를 생성하는 방법에 대한 자세한 내용은 새 IdM ID 범위 추가를 참조하십시오.

- 복제본의 작동이 중지되었습니다

복제본의 DNA ID 범위는 복제본이 작동하지 않고 삭제해야 할 때 자동으로 검색되지 않으므로 복제본에 이전에 할당된 DNA ID 범위를 사용할 수 없게 됩니다. DNA ID 범위를 복구하고 다른 복제본에서 사용할 수 있도록 합니다.

이렇게 하려면 해당 범위를 다른 서버에 수동으로 할당하기 전에 ID 범위 값이 무엇인지 확인합니다. 또한 중복된 UID 또는 GID를 방지하려면 복구된 범위의 ID 값이 사용자 또는 그룹에 할당되지 않았는지 확인합니다. 기존 사용자와 그룹의 UID 및 GID를 검사하여 이 작업을 수행할 수 있습니다.

DNA ID 범위를 수동으로 할당하는 명령을 사용하여 DNA ID 범위를 복제본에 수동으로 할당할 수 있습니다.

참고

새 DNA ID 범위를 할당하면 서버 또는 복제본에 있는 기존 항목의 UID가 동일하게 유지됩니다. 현재 DNA ID 범위를 변경해도 IdM은 과거에 할당된 범위에 대한 레코드를 유지하므로 문제가 발생하지 않습니다.

28.9. 수동으로 DNA ID 범위 할당

경우에 따라 기존 복제본에 DNA(Distributed Numeric Assignment) ID 범위를 수동으로 할당하여 작동하지 않는 복제본에 할당된 DNA ID 범위를 다시 할당해야 할 수 있습니다. 자세한 내용은 수동 ID 범위 할당 을 참조하십시오.

DNA ID 범위를 수동으로 조정할 때 새로 조정된 범위가 IdM ID 범위에 포함되어 있는지 확인합니다. ipa idrange-find 명령을 사용하여 확인할 수 있습니다. 그러지 않으면 명령이 실패합니다.

중요

중복되는 ID 범위를 만들지 않도록 주의하십시오. 서버 또는 복제본에 할당한 ID 범위가 겹치는 경우 서로 다른 두 개의 서버에서 동일한 ID 값을 다른 항목에 할당할 수 있습니다.

사전 요구 사항

- *선택 사항:* 작동하지 않는 복제본에서 DNA ID 범위를 복구하는 경우 먼저 현재 할당된 DNA ID 범위 표시에 설명된 명령을 사용하여 ID 범위를 찾습니다.

절차

- 지정된 서버에 대한 현재 **DNA ID** 범위를 정의하려면 **ipa-replica-manage dnarange-set** 을 사용합니다.

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

- 지정된 서버에 대한 다음 **DNA ID** 범위를 정의하려면 **ipa-replica-manage dnanextrange-set** 를 사용합니다.

```
# ipa-replica-manage dnanextrange-set serverB.example.com 1500-5000
```

검증 단계

- 현재 할당된 **DNA ID** 범위 표시에 설명된 명령을 사용하여 새로운 **DNA** 범위가 올바르게 설정되었는지 확인할 수 있습니다.

29장. 하위 ID 범위 수동 관리

컨테이너화된 환경에서는 IdM 사용자가 하위 ID 범위를 수동으로 할당해야 하는 경우가 있습니다. 다음 지침은 subID 범위를 관리하는 데 도움이 됩니다.

29.1. IDM CLI를 사용하여 SUBID 범위 생성

하위 ID 범위를 생성하여 수동으로 사용자에게 할당할 수 있습니다. 사용자 이름 *jsmith* 가 ipa 서버에 있다고 가정합니다.

사전 요구 사항

- IdM 사용자가 있습니다.
- 유효한 Kerberos 티켓을 받습니다. 웹 UI에서 Logging to IdM을 참조하십시오. 자세한 내용은 Kerberos 티켓 사용
- 루트 권한.

절차

1. 기존 하위 ID 범위를 확인합니다.

```
# ipa subid-find
```

2. subID 범위가 없는 경우 다음 명령을 입력하여 새 하위 ID 범위를 사용자에게 생성하고 할당합니다.

```
# ipa subid-generate --owner=jsmith

Added subordinate id "359dfcef-6b76-4911-bd37-bb5b66b8c418"

  Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
  Description: auto-assigned subid
  Owner: jsmith
  SubUID range start: 2147483648
  SubUID range size: 65536
  SubGID range start: 2147483648
  SubGID range size: 65536
```

3. 또는 모든 사용자에게 새 하위 ID 범위를 생성하고 할당합니다.

```
# /usr/libexec/ipa/ipa-subids --all-users

Found 2 user(s) without subordinate ids
  Processing user 'user4' (1/2)
  Processing user 'user5' (2/2)
Updated 2 user(s)
The ipa-subids command was successful
```

기본적으로 새 IdM 사용자에게 하위 ID 범위를 할당하려면 다음 옵션을 활성화합니다.

```
# ipa config-mod --user-default-subid=True
```

검증

1. 사용자에게 하위 ID 범위가 할당되어 있는지 확인하려면 다음 명령을 입력합니다.

```
# ipa subid-find --owner=jsmith

1 subordinate id matched

  Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
  Owner: jsmith
  SubUID range start: 2147483648
  SubUID range size: 65536
  SubGID range start: 2147483648
  SubGID range size: 65536

Number of entries returned 1
```

29.2. IDM WEBUI 인터페이스를 사용하여 하위 ID 범위 생성

하위 ID 범위를 생성하여 IdM WebUI 인터페이스의 사용자에게 할당할 수 있습니다.

사전 요구 사항

- IdM 사용자가 있습니다.
- 유효한 Kerberos 티켓을 받습니다. 웹 UI에서 Logging to IdM을 참조하십시오. 자세한 내용은 Kerberos 티켓 사용

- 루트 권한.

절차

1. IdM WebUI 인터페이스에서는 ID 탭을 확장하고 ID 대체 옵션을 선택합니다.

2. Subordinate ID 인터페이스가 표시되면 인터페이스 오른쪽 상단에 있는 Add (추가) 버튼을 클릭합니다. "Add subid" 창이 표시됩니다.

3. "Add subid" 창에서 subID 범위를 할당하려는 사용자인 소유자를 선택합니다.

4. 추가 버튼을 클릭합니다.

검증

1. Subordinate ID 탭에서 테이블을 확인합니다. 새 레코드가 표시되고 소유자는 하위 ID 범위를 할당하는 사용자입니다.

29.3. IDM CLI를 사용하여 기존 SUBID 범위 관리

subID 범위를 검색하고 필요한 경우 특정 하나에 대한 정보를 표시할 수 있습니다. 사용자 이름 *jsmith* 가 ipa 서버에 있다고 가정합니다.

사전 요구 사항

- IdM 사용자가 있습니다.

절차

1. 고유 ID 해시를 알고 있는 경우 subID 범위에 대한 세부 정보를 표시하려면 다음 명령을 입력합니다.

```
# ipa subid-show 359dfcef-6b76-4911-bd37-bb5b66b8c418

 Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
 Owner: jsmith
 SubUID range start: 2147483648
```

```
SubUID range size: 65536
SubGID range start: 2147483648
SubGID range size: 65536
```

2. 해당 범위의 하위 ID가 있는 경우 하위 ID 범위에 대한 세부 정보를 찾으려면 다음 명령을 사용합니다.

```
# ipa subid-match --subuid=2147483648

1 subordinate id matched

 Unique ID: 359dfcef-6b76-4911-bd37-bb5b66b8c418
 Owner: uid=jsmith
 SubUID range start: 2147483648
 SubUID range size: 65536
 SubGID range start: 2147483648
 SubGID range size: 65536

Number of entries returned 1
```

29.4. GETSUBID 명령을 사용하여 하위 ID 범위 나열

하위 ID 범위(예: IdM 환경의 user 1)를 나열하려면 아래 지침을 따르십시오.

사전 요구 사항

- user1 은 IdM에 있습니다.
- shadow-utils-subid 패키지가 설치됩니다.

절차

1. subid: sss 레코드를 /etc/nsswitch.conf 파일에 포함합니다.

   하위 필드에 대해 하나의 값만 제공할 수 있습니다. subid 필드를 sss 값으로 설정하면 utils에 IdM 설정에서 subID 범위를 사용하도록 지시합니다. file 값 또는 no 값은 /etc/subuid 및 /etc/subgid 파일의 하위 ID 범위를 사용하도록 utils를 설정합니다.

2. 사용자의 하위 ID 범위를 나열합니다.

```
# getsubids user1
0: user1 2147483648 65536
```

30장. IDM CLI에서 호스트 관리

이 장에서는 IdM(Identity Management)의 호스트 및 호스트 항목 과 IdM CLI에서 호스트 및 호스트 항목을 관리할 때 수행되는 다음 작업을 소개합니다.

- 호스트 등록
- IdM 호스트 항목 추가
- IdM 호스트 항목 삭제
- 호스트 재등록
- 호스트 이름 변경
- 호스트 비활성화
- 호스트 다시 활성화

또한 장에는 사전 요구 사항, 컨텍스트 및 이러한 작업 결과에 대한 개요 테이블이 포함되어 있습니다.

30.1. 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.

30.2. 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.

30.3. 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 증명을 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치.

- IdM LDAP의 host 항목이 있습니다. 이 시나리오의 경우 ipa-client-install 을 실행하려면 제한된 관리자의 인증 정보가 필요합니다. 이 경우 제한된 관리자에게는 호스트 등록 권한을 제공하는 Enrollment Administrator 역할이 있습니다. 자세한 내용은 사용자 자격 증명을 사용하여 클라이언트 설치: 대화식 설치.

- IdM LDAP의 호스트 항목이 존재하며 전체 또는 제한된 관리자가 호스트에 대한 OTP가 생성되었습니다. 이 시나리오에서는 --password 옵션을 사용하여 ipa-client-install 명령을 실행하여 올바른 OTP를 제공하는 경우 IdM 클라이언트를 일반 사용자로 설치할 수 있습니다. 자세한 내용은 회용 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.

30.4. 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(사용자 및 호스트 등록):
  - 관리자는 사용자 또는 호스트 모두에 대해 IdM에 실제로 참여하기 전에 사용자 및 호스트에 대해 LDAP 항목을 생성할 수 있습니다. stage 사용자의 경우 명령은 ipa stageuser-add 입니다. 호스트에 대해 명령은 ipa host-add 입니다.
  - *키 테이블* 또는 일부 익스텐트와 유사한 대칭 키인 keytab을 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 password는 사용자의 기본 인증 방법이지만 keytab은 호스트의 기본 인증 방법입니다. 키탭은 호스트의 파일에 저장됩니다.

표 30.1. 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(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 30.2. 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.

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. 클라이언트에서 인증서는 certmonger 라는 서비스에서 관리합니다.**

30.5. 호스트 작업

이 섹션에는 호스트 등록 및 활성화와 관련된 가장 일반적인 작업과 전제 조건, 컨텍스트 및 실행 결과에 대해 설명합니다.

표 30.3. 호스트 운영 파트 1

동작	작업의 전제 조건은 무엇입니까?	명령을 실행하는 것은 언제 적합합니까?	시스템 관리자가 작업을 어떻게 수행합니까? 실행 중인 명령은 무엇입니까?
고객 등록	*Installing_Identity_Management*에서 Identity Management 클라이언트 설치를 위한 시스템 준비를 참조하십시오.	호스트가 IdM 영역에 가입하려는 경우.	IdM 도메인에서 클라이언트로 시스템을 등록하는 작업은 2부로 구성된 프로세스입니다. **ipa host-add 명령이 실행되는 경우 클라이언트에 대한 호스트** 항목이 생성되고(389 Directory Server 인스턴스에 저장되고) 클라이언트를 프로비저닝하기 위해 keytab이 생성됩니다. 두 부분 모두 **ipa-client-install** 명령으로 자동으로 수행됩니다. 이러한 단계를 별도로 수행할 수도 있습니다. 이를 통해 관리자는 실제로 클라이언트를 구성하기 전에 시스템과 IdM을 준비할 수 있습니다. 따라서 대규모 배포를 비롯한 보다 유연한 설정 시나리오가 가능합니다.
클라이언트 비활성화	호스트에 IdM에 항목이 있어야 합니다. 호스트에 활성 키탭이 있어야 합니다.	유지 관리를 위해 IdM 영역에서 임시로 호스트를 제거하려는 경우.	**ipa host-disable host_name**
클라이언트 활성화	호스트에 IdM에 항목이 있어야 합니다.	일시적으로 비활성화된 호스트를 다시 활성화하려면 다음을 수행합니다.	**ipa-getkeytab**

동작	작업의 전제 조건은 무엇입니까?	명령을 실행하는 것은 언제 적합합니까?	시스템 관리자가 작업을 어떻게 수행합니까? 실행 중인 명령은 무엇입니까?
클라이언트 재등록	호스트에는 IdM에 en 항목이 있어야 합니다.	원래 호스트가 손실되었지만 동일한 호스트 이름을 가진 호스트를 설치한 경우.	**ipa-client-install --keytab** 또는 **ipa-client-install --force-join**
클라이언트 등록 해제	호스트에 IdM에 항목이 있어야 합니다.	IdM 영역에서 호스트를 영구적으로 제거하려는 경우.	**ipa-client-install --uninstall**

표 30.4. 호스트 작업 2부

동작	관리자가 어떤 시스템에서 명령을 실행할 수 있습니까?	작업이 수행되면 어떻게 됩니까? IdM에서 호스트의 기능에 대한 결과는 무엇입니까? 도입/제거된 제한 사항은 무엇입니까?
고객 등록	2단계 등록의 경우: **ipa host-add** 는 IdM 클라이언트에서 실행될 수 있습니다. **ipa-client-install** 의 두 번째 단계는 클라이언트 자체에서 실행해야 합니다.	기본적으로 인증 및 권한 부여를 위해 IdM 서버에 연결하도록 SSSD를 구성합니다. 선택적으로 Kerberos 및 LDAP를 통해 IdM 서버에서 작동하도록 PAM(Pluggable Authentication Module) 및 NSS(Name Switching Service)를 구성할 수 있습니다.
클라이언트 비활성화	IdM의 모든 시스템, 호스트 자체	호스트의 Kerberos 키와 SSL 인증서가 무효화되고 호스트에서 실행 중인 모든 서비스가 비활성화됩니다.
클라이언트 활성화	IdM의 모든 시스템. 비활성화 호스트에서 실행되는 경우 LDAP 자격 증명을 제공해야 합니다.	호스트의 Kerberos 키와 SSL 인증서가 다시 유효해지며 호스트에서 실행되는 모든 IdM 서비스가 다시 활성화됩니다.
클라이언트 재등록	다시 등록할 호스트입니다. LDAP 자격 증명을 제공해야 합니다.	호스트에 대해 새 Kerberos 키가 생성되어 이전 키가 교체됩니다.

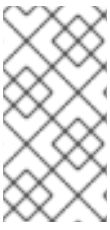
동작	관리자가 어떤 시스템에서 명령을 실행할 수 있습니까?	작업이 수행되면 어떻게 됩니까? IdM에서 호스트의 기능에 대한 결과는 무엇입니까? 도입/제거된 제한 사항은 무엇입니까?
클라이언트 등록 해제	등록 취소할 호스트입니다.	명령은 IdM을 구성 해제하고 시스템을 이전 상태로 되돌리려고 시도합니다. 이 프로세스의 일부는 IdM 서버에서 호스트 등록을 취소하는 것입니다. Unenrollment는 IdM 서버에서 키 키를 비활성화하도록 구성됩니다. **/etc/krb5.keytab** 의 시스템 주체(host/**<fqdn>@REALM**)는 IdM 서버에 인증하여 등록 취소합니다. 이 주체가 없으면 등록 취소가 실패하고 관리자가 호스트 주체(ipa**host-disable <fqdn>**)를 비활성화해야 합니다.

30.6. IDM LDAP의 호스트 항목

이 섹션에서는 IdM(Identity Management)의 호스트 항목과 어떤 속성을 포함할 수 있는지에 대해 설명합니다.

LDAP 호스트 항목에는 IdM 내의 클라이언트에 대한 모든 관련 정보가 포함되어 있습니다.

- **호스트와 관련된 서비스 항목**
- **호스트 및 서비스 주체**
- **액세스 제어 규칙**
- **물리적 위치 및 운영 체제와 같은 시스템 정보**



참고

IdM 웹 UI ID → 호스트 탭에는 IdM LDAP에 저장된 특정 호스트에 대한 모든 정보가 표시되지 않습니다.

호스트 항목 구성 속성

호스트 항목은 실제 위치, MAC 주소, 키 및 인증서와 같이 시스템 구성 외부에 있는 호스트에 대한 정보를 포함할 수 있습니다.

이 정보는 수동으로 만드는 경우 호스트 항목을 생성할 때 설정할 수 있습니다. 또는 호스트가 도메인에 등록된 후 대부분의 이 정보를 host 항목에 추가할 수 있습니다.

표 30.5. 호스트 구성 속성

UI 필드	명령줄 옵션	설명
설명	**--desc**=*description*	호스트에 대한 설명입니다.
지역	**--locality**=*locality*	호스트의 지리적 위치.
위치	**--location**=*location*	호스트의 실제 위치(예: 데이터 센터 랙).
플랫폼	**--platform**=*string*	호스트 하드웨어 또는 아키텍처.
운영 체제	**--os**=*string*	호스트의 운영 체제 및 버전.
MAC 주소	**--macaddress**=*address*	호스트의 MAC 주소입니다. 이것은 다중 값 속성입니다. MAC 주소는 NIS 플러그인에서 호스트에 대한 NIS **ethers** 맵을 만드는 데 사용됩니다.
SSH 공개 키	**--sshpubkey**=*string*	호스트에 대한 전체 SSH 공개 키. 이것은 다중 값 속성이므로 여러 키를 설정할 수 있습니다.
주체 이름 (편집할 수 없음)	**--principalname**=*principal*	호스트의 Kerberos 주체 이름입니다. 다른 주체가 **-p**;에 명시적으로 설정되지 않는 한 클라이언트 설치 중에 호스트 이름이 기본값입니다. 이 설정은 명령줄 도구를 사용하여 변경할 수 있지만 UI에서는 변경할 수 없습니다.
일회성 암호 설정	**--password**=*string*	이 옵션은 대규모 등록에 사용할 수 있는 호스트의 암호를 설정합니다.
-	**--random**	이 옵션은 대량 등록에 사용할 임의 암호를 생성합니다.
-	**--certificate**=*string*	호스트에 대한 인증서 Blob.
-	**--updatedns**	이렇게 하면 IP 주소가 변경되면 호스트가 DNS 항목을 동적으로 업데이트할 수 있는지 여부를 설정합니다.

30.7. IDM CLI에서 IDM 호스트 항목 추가

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 IdM(Identity Management)에 호스트 항목을 추가하는 방법을 설명합니다.

호스트 항목은 host-add 명령을 사용하여 생성합니다. 이 명령은 IdM 디렉터리 서버에 호스트 항목을 추가합니다. CLI에 ipa help host 를 입력하여 host -add에 사용할 수 있는 전체 옵션 목록을 가져와 ipa host manpage를 참조합니다.

IdM에 호스트를 추가할 때 몇 가지 시나리오가 있습니다.

- 가장 기본적인 경우 클라이언트 호스트 이름만 지정하여 Kerberos 영역에 클라이언트를 추가하고 IdM LDAP 서버에서 항목을 생성합니다.

```
$ ipa host-add client1.example.com
```

- IdM 서버가 DNS를 관리하도록 구성된 경우 --ip-address 옵션을 사용하여 호스트를 DNS 리소스 레코드에 추가합니다.

예 30.1. 정적 IP 주소를 사용하여 호스트 항목 생성

```
$ ipa host-add --ip-address=192.168.166.31 client1.example.com
```

- 추가할 호스트에 고정 IP 주소가 없거나 클라이언트가 구성될 때 IP 주소를 알 수 없는 경우 ipa host-add 명령과 함께 --force 옵션을 사용합니다.

예 30.2. DHCP를 사용하여 호스트 항목 생성

```
$ ipa host-add --force client1.example.com
```

예를 들어 랩탑은 IdM 클라이언트로 사전 구성될 수 있지만 구성 시 IP 주소는 없습니다. --force 를 사용하면 기본적으로 IdM DNS 서비스에 자리 표시자 항목이 생성됩니다. DNS 서비스에서 레코드를 동적으로 업데이트하면 호스트의 현재 IP 주소가 탐지되고 해당 DNS 레코드가 업데이트됩니다.

30.8. IDM CLI에서 호스트 항목 삭제

- host-del 명령을 사용하여 호스트 레코드를 삭제합니다. IdM 도메인에 통합된 DNS가 있는 경

우 **--updatedns** 옵션을 사용하여 **DNS**에서 호스트에 대한 모든 종류의 관련 레코드를 제거합니다.

```
$ ipa host-del --updatedns client1.example.com
```

30.9. IDENTITY MANAGEMENT 클라이언트 다시 등록

이 섹션에서는 **ID** 관리 클라이언트를 다시 등록할 수 있는 다양한 방법에 대해 설명합니다.

30.9.1. IdM의 클라이언트 등록

이 섹션에서는 **IdM(Identity Management)** 클라이언트를 다시 등록하는 방법을 설명합니다.

클라이언트 시스템이 삭제되고 **IdM** 서버와 연결이 끊어진 경우(예: 클라이언트의 하드웨어 실패로 인해) 키탭이 있는 경우 클라이언트를 다시 등록할 수 있습니다. 이 시나리오에서는 동일한 호스트 이름이 있는 **IdM** 환경에서 클라이언트를 다시 가져오려고 합니다.

다시 등록하는 동안 클라이언트는 새 **Kerberos** 키와 **SSH** 키를 생성하지만 **LDAP** 데이터베이스의 클라이언트 **ID**는 변경되지 않습니다. 다시 등록한 후 호스트에 **IdM** 서버와의 연결이 손실되기 전에 이전과 동일한 **FQDN** 을 사용하는 동일한 **LDAP** 오브젝트에 키 및 기타 정보가 있습니다.

중요

도메인 항목이 아직 활성 상태인 클라이언트만 다시 등록할 수 있습니다. 클라이언트를 설치 제거하거나(**ipa -client-install --uninstall**사용) 호스트 항목(**ipa host-disable**사용)을 비활성화한 경우 다시 등록할 수 없습니다.

이름을 변경한 후에는 클라이언트를 다시 등록할 수 없습니다. 이는 **ID** 관리에서 **LDAP**에서 클라이언트 항목의 키 특성이 클라이언트의 호스트 이름인 해당 **FQDN** 이기 때문입니다. 클라이언트의 **LDAP** 오브젝트가 변경되지 않은 상태로 남아 있는 클라이언트를 다시 등록하는 것과 달리, 클라이언트의 이름을 변경한 결과는 새로운 **FQDN** 을 사용하여 다른 **LDAP** 오브젝트에 키와 기타 정보가 있다는 것입니다. 따라서 클라이언트 이름을 변경할 수 있는 유일한 방법은 **IdM**에서 호스트를 제거하고 호스트의 호스트 이름을 변경한 후 새 이름으로 **IdM** 클라이언트로 설치하는 것입니다. 클라이언트의 이름을 바꾸는 방법에 대한 자세한 내용은 **Identity Management** 클라이언트 시스템 복원을 참조하십시오.

클라이언트를 다시 등록하는 동안 발생하는 사항

다시 등록하는 동안 **ID** 관리:

- 원래 호스트 인증서 취소
- 새 SSH 키 생성
- 새 키탭 생성

30.9.2. 사용자 인증 정보를 사용하여 클라이언트 재등록: 대화형 등록

이 절차에서는 권한 있는 사용자의 자격 증명을 사용하여 대화식으로 ID 관리 클라이언트를 다시 등록하는 방법을 설명합니다.

1. 동일한 호스트 이름으로 클라이언트 시스템을 다시 생성합니다.
2. 클라이언트 시스템에서 **ipa-client-install --force-join** 명령을 실행합니다.

```
# ipa-client-install --force-join
```

3. 이 스크립트는 클라이언트를 다시 등록하는 데 ID를 사용하는 사용자를 묻습니다. 예를 들어 **Enrollment Administrator** 역할이 있는 **hostadmin** 사용자는 다음과 같습니다.

```
User authorized to enroll computers: hostadmin
Password for hostadmin@EXAMPLE.COM:
```

추가 리소스

- 사용자 인증 정보를 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치 *ID 관리 설치 중*.

30.9.3. 클라이언트 키탭을 사용하여 클라이언트를 다시 등록합니다. 비대화형 재등록

사전 요구 사항

- 원래 클라이언트 키탭 파일을 백업합니다(예: /tmp 또는 /root 디렉토리).

절차

이 절차에서는 클라이언트 시스템의 키탭을 사용하여 IdM(Identity Management) 클라이언트를 비대화식으로 등록하는 방법을 설명합니다. 예를 들어 클라이언트 keytab을 사용하여 다시 등록하는 것은 자동화된 설치에 적합합니다.

1. 동일한 호스트 이름으로 클라이언트 시스템을 다시 생성합니다.

2. 백업 위치의 keytab 파일을 다시 생성된 클라이언트 시스템의 /etc/ 디렉토리로 복사합니다.

3. ipa-client-install 유틸리티를 사용하여 클라이언트를 다시 등록하고 --keytab 옵션으로 keytab 위치를 지정합니다.

```
# ipa-client-install --keytab /etc/krb5.keytab
```

참고

keytab 옵션에 지정된 key tab 은 인증을 시작하여 등록을 시작할 때만 사용됩니다. 다시 등록하는 동안 IdM은 클라이언트에 대한 새 키탭을 생성합니다.

30.9.4. 설치 후 Identity Management 클라이언트 테스트

명령줄 인터페이스는 ipa-client-install 이 성공적으로 수행되었지만 자체 테스트를 수행할 수도 있음을 알려줍니다.

ID 관리 클라이언트가 서버에 정의된 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 - 를 다른 IdM 사용자로 실행하십시오.

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

30.10. ID 관리 클라이언트 시스템 이름 변경

다음 섹션에서는 ID 관리 클라이언트 시스템의 호스트 이름을 변경하는 방법을 설명합니다.

주의

클라이언트 이름 바꾸기는 수동 절차입니다. 호스트 이름을 변경할 필요가 없는 한 수행하지 마십시오.

Identity Management 클라이언트의 이름 변경에는 다음이 포함됩니다.

1. 호스트 준비. 자세한 내용은 IdM 클라이언트의 이름 변경 준비를 참조하십시오.
2. 호스트에서 IdM 클라이언트 설치 제거. 자세한 내용은 ID 관리 클라이언트 설치 제거를 참조하십시오.
3. 호스트 이름 바꾸기. 자세한 내용은 호스트 시스템 복원을 참조하십시오.
4. 호스트에 새 이름으로 IdM 클라이언트 설치. 자세한 내용은 ID 관리 클라이언트 설치를 참조하십시오. 기본 시나리오 를 사용하여 *ID 관리 설치*.
5. IdM 클라이언트 설치 후 호스트 구성. 자세한 내용은 서비스 다시 생성, 인증서 재 생성 및 호스트 그룹 다시 추가를 참조하십시오.

30.10.1. 변경 사항을 위해 IdM 클라이언트 준비

현재 클라이언트를 제거하기 전에 클라이언트의 특정 설정을 기록해 두십시오. 새 호스트 이름으로 시스템을 다시 등록한 후 이 구성을 적용합니다.

- 시스템에서 실행 중인 서비스를 확인합니다.
  - ipa service-find 명령을 사용하고 출력에서 인증서가 있는 서비스를 식별합니다.

```
$ ipa service-find old-client-name.example.com
```

- 또한 각 호스트에는 **ipa** ***service-find*** *출력에 표시되지 않는 기본 호스트* 서비스가 있습니다. **host** 서비스의 서비스 주체(**host principal**라고도 함)는 ***host*/*old-client-name.example.com*** 입니다.

- **ipa service-find** ***old-client-name.example.com*** 에서 표시되는 모든 서비스 주체에 대해 ***old-client-name.example.com*** *시스템에서 해당 키 탭의 위치를 확인합니다.*

```
# find / -name "*.keytab"
```

클라이언트 시스템의 각 서비스에는 **ldap/*old-client-name.example.com@EXAMPLE.COM*** 과 같은 ***service_name/host_name@REALM*** 형식의 **Kerberos** 주체가 있습니다.

- 시스템이 속한 모든 호스트 그룹을 식별합니다.

```
# ipa hostgroup-find old-client-name.example.com
```

30.10.2. ID 관리 클라이언트 설치 제거

클라이언트를 설치 제거하면 **ID** 관리 도메인에서 클라이언트가 제거되며 **SSSD(System Security Services Daemon)**와 같은 시스템 서비스의 모든 특정 **ID** 관리 구성도 제거됩니다. 이렇게 하면 클라이언트 시스템의 이전 구성이 복원됩니다.

절차

1. **ipa-client-install --uninstall** 명령을 실행합니다.

```
[root@client]# ipa-client-install --uninstall
```

2. 서버에서 클라이언트 호스트의 **DNS** 항목을 수동으로 제거합니다.

```
[root@server]# ipa dnsrecord-del
Record name: old-client-client
Zone name: idm.example.com
No option to delete specific record provided.
```

```
Delete all? Yes/No (default No): yes
----------------------
Deleted record "old-client-name"
```

3. /etc/krb5.keytab 이외의 식별된 각 키탭에 대해 이전 주체를 제거하십시오.

```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

4. IdM 서버에서 호스트 항목을 제거합니다. 이렇게 하면 모든 서비스가 제거되고 해당 호스트에 대해 발급된 모든 인증서가 취소됩니다.

```
[root@server ~]# ipa host-del client.example.com
```

30.10.3. 호스트 시스템 이름 변경

필요에 따라 시스템의 이름을 변경합니다. 예를 들면 다음과 같습니다.

```
[root@client]# hostnamectl set-hostname new-client-name.example.com
```

이제 새 호스트 이름을 사용하여 Identity Management 클라이언트를 Identity Management 도메인에 다시 설치할 수 있습니다.

30.10.4. 서비스 다시 추가, 인증서 재생, 호스트 그룹 다시 추가

IdM(Identity Management) 서버에서 이름 변경을 위해 IdM 클라이언트 준비에서 확인된 모든 서비스에 대해 새 키탭을 추가합니다.

+

```
[root@server ~]# ipa service-add service_name/new-client-name
```

1. 이름을 변경하기 위해 IdM 클라이언트 준비에 할당된 인증서가 있는 서비스에 대한 인증서를 생성합니다. 다음을 수행할 수 있습니다.

   - IdM 관리 도구 사용

- certmonger 유틸리티 사용

2. IdM 클라이언트의 이름 변경을 위해 IdM 클라이언트 준비에서 확인된 호스트 그룹에 클라이언트를 다시 추가합니다.

30.11. 호스트 항목 비활성화 및 재활성화

이 섹션에서는 IdM(Identity Management)에서 호스트를 비활성화하고 다시 활성화하는 방법에 대해 설명합니다.

30.11.1. 호스트 비활성화

IdM에서 호스트 항목을 비활성화하려면 다음 절차를 완료합니다.

도메인 서비스, 호스트 및 사용자가 활성 호스트에 액세스할 수 있습니다. 유지 관리상의 이유로 활성 호스트를 일시적으로 제거해야 하는 상황이 있을 수 있습니다. 예를 들면 다음과 같습니다. 이러한 상황에서 호스트를 삭제하는 것은 호스트 항목 및 연결된 모든 구성을 영구적으로 제거하므로 바람직하지 않습니다. 대신 호스트를 비활성화하는 옵션을 선택합니다.

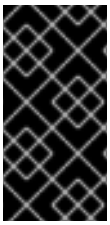
호스트를 비활성화하면 도메인 사용자가 도메인에서 영구적으로 제거하지 않고 액세스할 수 없습니다.

절차

- host-disable 명령을 사용하여 호스트를 비활성화합니다. 호스트를 비활성화하면 현재 활성 키탭이 종료됩니다. 예를 들면 다음과 같습니다.

```
$ kinit admin
$ ipa host-disable client.example.com
```

호스트를 비활성화하면 모든 IdM 사용자, 호스트 및 서비스에서 호스트를 사용할 수 없게 됩니다.



중요

호스트 항목을 비활성화하면 해당 호스트가 비활성화될 뿐만 아니라. 해당 호스트에서 구성된 모든 서비스도 비활성화합니다.

30.11.2. 호스트 재활성화

이 섹션에서는 비활성화된 IdM 호스트를 다시 활성화하는 방법에 대해 설명합니다.

호스트를 비활성화하면 활성 키탭이 종료되어 구성 항목을 사용하지 않고 IdM 도메인에서 호스트가 제거되었습니다.

절차

- 호스트를 다시 활성화하려면 ipa-getkeytab 명령을 사용하여 다음을 추가합니다.
  - 키탭을 요청할 IdM 서버를 지정하는 -s 옵션
  - 주체 이름을 지정하는 -p 옵션
  - keytab을 저장할 파일을 지정하는 -k 옵션.

예를 들어 client.example.com에 대해 server.example.com 에서 새 호스트 키탭을 요청하고 /etc/krb5.keytab 파일에 키탭을 저장하려면 다음을 수행합니다.

```
$  ipa-getkeytab -s server.example.com -p host/client.example.com -k /etc/krb5.keytab -D
"cn=directory manager" -w password
```

참고

관리자 자격 증명을 사용하여 -D "uid=admin,cn=users,cn=accounts,dc=example,dc=com" 을 지정할 수도 있습니다. 자격 증명은 호스트에 대한 키탭을 만들 수 있는 사용자에게 대응하는 것이 중요합니다.

ipa-getkeytab 명령이 활성 IdM 클라이언트 또는 서버에서 실행되면 사용자에게 TGT를 사용하여 얻은 경우 LDAP 자격 증명(예: kinit admin) 없이 실행할 수 있습니다. 비활성화된 호스트에서 직접 명령을 실행하려면 IdM 서버에 인증할 LDAP 자격 증명을 제공합니다.

31장. IDM 웹 UI에서 호스트 항목 추가

이 장에서는 IdM(Identity Management)의 호스트와 IdM 웹 UI에 호스트 항목을 추가하는 작업을 소개합니다.

31.1. 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.

31.2. 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.

31.3. 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 증명을 사용하여 클라이언트 설치를 참조하십시오. 대화식 설치.
- IdM LDAP의 host 항목이 있습니다. 이 시나리오의 경우 ipa-client-install 을 실행하려면 제한된 관리자의 인증 정보가 필요합니다. 이 경우 제한된 관리자에게는 호스트 등록 권한을 제공하는 Enrollment Administrator 역할이 있습니다. 자세한 내용은 사용자 자격 증명을 사용하여 클라이언트 설치: 대화식 설치.
- IdM LDAP의 호스트 항목이 존재하며 전체 또는 제한된 관리자가 호스트에 대한 OTP가 생성되었습니다. 이 시나리오에서는 --password 옵션을 사용하여 ipa-client-install 명령을 실행하여 올바른 OTP를 제공하는 경우 IdM 클라이언트를 일반 사용자로 설치할 수 있습니다. 자세한 내용은 회용 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.

31.4. IDM 호스트 및 사용자의 등록 및 인증: 비교

IdM의 사용자와 호스트 사이에는 많은 유사점이 있습니다. 이 섹션에서는 등록 단계에서 확인할 수 있

는 몇 가지 유사점과 배포 단계에서 인증에 관한 몇 가지 유사 사항에 대해 설명합니다.

- 등록 단계(사용자 및 호스트 등록):
  - 관리자는 사용자 또는 호스트 모두에 대해 IdM에 실제로 참여하기 전에 사용자 및 호스트에 대해 LDAP 항목을 생성할 수 있습니다. stage 사용자의 경우 명령은 ipa stageuser-add 입니다. 호스트에 대해 명령은 ipa host-add 입니다.
  - *키 테이블* 또는 일부 익스텐트와 유사한 대칭 키인 keytab을 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 password는 사용자의 기본 인증 방법이지만 keytab은 호스트의 기본 인증 방법입니다. 키탭은 호스트의 파일에 저장됩니다.

표 31.1. 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(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 31.2. 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.

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. 클라이언트에서 인증서는 certmonger 라는 서비스에서 관리합니다.**

31.5. IDM LDAP의 호스트 항목

이 섹션에서는 IdM(Identity Management)의 호스트 항목과 어떤 속성을 포함할 수 있는지에 대해 설명합니다.

LDAP 호스트 항목에는 IdM 내의 클라이언트에 대한 모든 관련 정보가 포함되어 있습니다.

- **호스트와 관련된 서비스 항목**

- **호스트 및 서비스 주체**

- 액세스 제어 규칙
- 물리적 위치 및 운영 체제와 같은 시스템 정보

참고

IdM 웹 UI ID → 호스트 탭에는 IdM LDAP에 저장된 특정 호스트에 대한 모든 정보가 표시되지 않습니다.

호스트 항목 구성 속성

호스트 항목은 실제 위치, MAC 주소, 키 및 인증서와 같이 시스템 구성 외부에 있는 호스트에 대한 정보를 포함할 수 있습니다.

이 정보는 수동으로 만드는 경우 호스트 항목을 생성할 때 설정할 수 있습니다. 또는 호스트가 도메인에 등록된 후 대부분의 이 정보를 host 항목에 추가할 수 있습니다.

표 31.3. 호스트 구성 속성

UI 필드	명령줄 옵션	설명
설명	**--desc**=*description*	호스트에 대한 설명입니다.
지역	**--locality**=*locality*	호스트의 지리적 위치.
위치	**--location**=*location*	호스트의 실제 위치(예: 데이터 센터 랙).
플랫폼	**--platform**=*string*	호스트 하드웨어 또는 아키텍처.
운영 체제	**--os**=*string*	호스트의 운영 체제 및 버전.
MAC 주소	**--macaddress**=*address*	호스트의 MAC 주소입니다. 이것은 다중 값 속성입니다. MAC 주소는 NIS 플러그인에서 호스트에 대한 NIS **ethers** 맵을 만드는 데 사용됩니다.
SSH 공개 키	**--sshpubkey**=*string*	호스트에 대한 전체 SSH 공개 키. 이것은 다중 값 속성이므로 여러 키를 설정할 수 있습니다.

UI 필드	명령줄 옵션	설명
주체 이름 (편집할 수 없음)	**--principalname**=*principal*	호스트의 Kerberos 주체 이름입니다. 다른 주체가 **-p**;에 명시적으로 설정되지 않는 한 클라이언트 설치 중에 호스트 이름이 기본값입니다. 이 설정은 명령줄 도구를 사용하여 변경할 수 있지만 UI에서는 변경할 수 없습니다.
일회성 암호 설정	**--password**=*string*	이 옵션은 대규모 등록에 사용할 수 있는 호스트의 암호를 설정합니다.
-	**--random**	이 옵션은 대량 등록에 사용할 임의 암호를 생성합니다.
-	**--certificate**=*string*	호스트에 대한 인증서 Blob.
-	**--updatedns**	이렇게 하면 IP 주소가 변경되면 호스트가 DNS 항목을 동적으로 업데이트할 수 있는지 여부를 설정합니다.

31.6. 웹 UI에서 호스트 항목 추가

1. **Identity(ID) 탭을 열고 Hosts(호스트) 하위 탭을 선택합니다.**

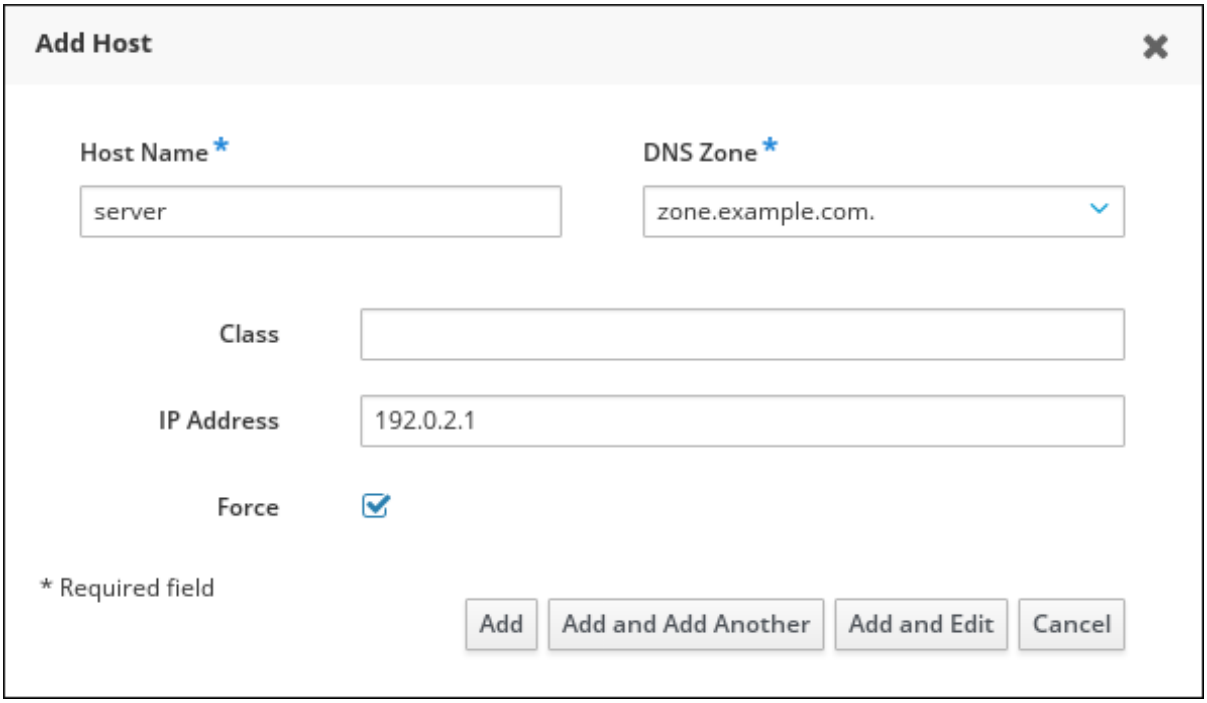
2. **hosts 목록 상단에 있는 Add(추가)를 클릭합니다.**

그림 31.1. 호스트 항목 추가

3. **시스템 이름을 입력하고 드롭다운 목록에서 구성된 영역에서 도메인을 선택합니다. 호스트에 이미 정적 IP 주소가 할당된 경우 DNS 항목이 완전히 생성되도록 host 항목으로 해당 주소를 포함합니다.**

Class 필드는 현재 특별한 용도가 없습니다.

그림 **31.2.** 호스트 마법사 추가



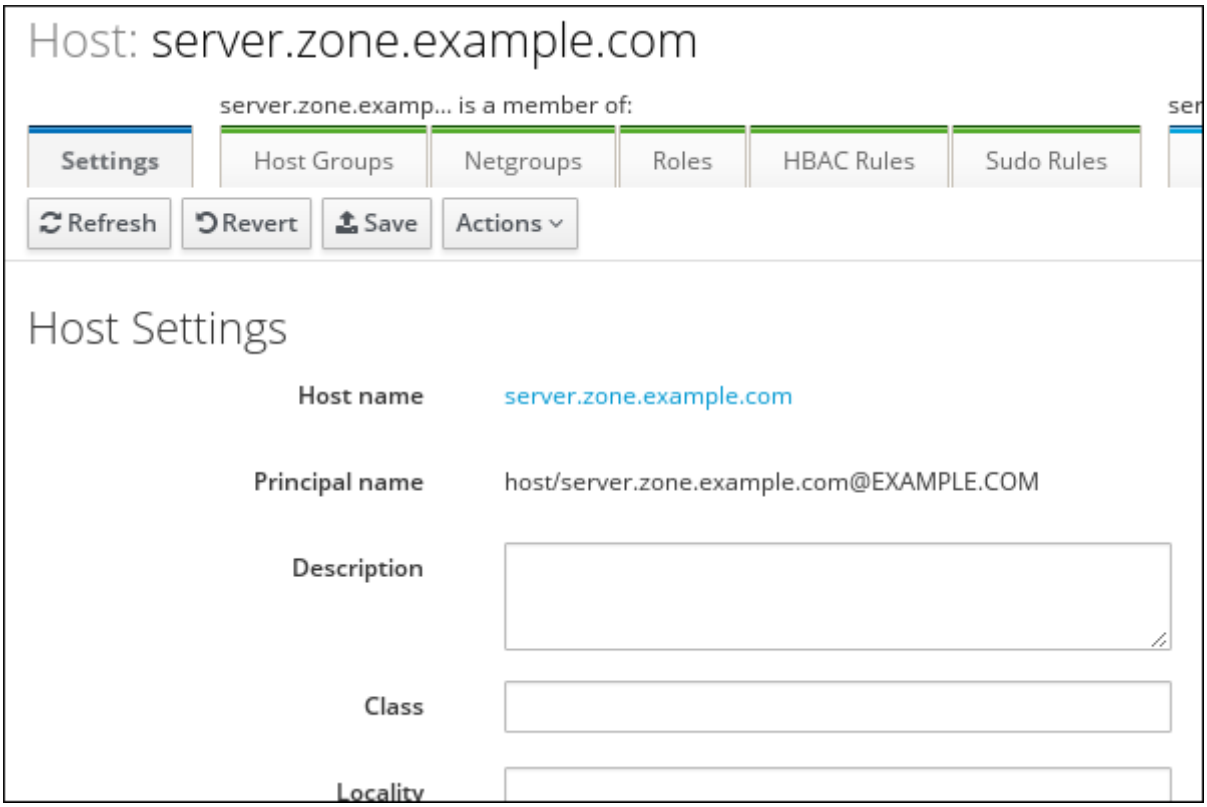
DNS 영역은 **IdM**에서 생성할 수 있습니다. **IdM** 서버가 **DNS** 서버를 관리하지 않으면 일반 텍스트 필드와 같이 메뉴 영역에 수동으로 영역을 입력할 수 있습니다.

참고

DNS를 통해 호스트를 확인할 수 있는지 여부를 건너뛰려면 **Force** (강제) 확인란을 선택합니다.

4. **Add and Edit**(추가 및 편집) 버튼을 클릭하여 확장된 항목 페이지로 바로 이동하고 추가 속성 정보를 입력합니다. 호스트 하드웨어 및 물리적 위치에 대한 정보는 **host** 항목에 포함할 수 있습니다.

그림 31.3. 항목 페이지 확장



32장. 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.

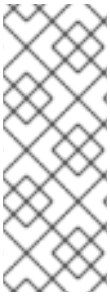
- FQDN에서만 정의된 IdM 호스트 항목이 있는지 확인
- IP 주소가 있는 IdM 호스트 항목이 있는지 확인
- 임의의 암호를 사용하여 여러 IdM 호스트 항목이 있는지 확인
- 여러 IP 주소가 있는 IdM 호스트 항목이 있는지 확인
- IdM 호스트 항목이 없는지 확인

32.1. 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 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.

절차

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

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

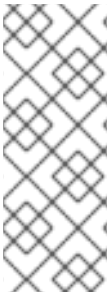
2. 확인할 IdM에 있는 호스트의 FQDN 으로 Ansible 플레이북 파일을 생성합니다. 이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/playbooks/host/add-host.yml 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- 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 **IdM**에 있는지 확인합니다.

32.2. 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 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.

절차

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

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

2. 확인하고자 하는 IdM에 있는 호스트의 정규화된 도메인 이름 (FQDN)을 사용하여 Ansible 플레이북 파일을 생성합니다. 또한 IdM 서버가 DNS를 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 **admin**으로 로그인합니다.

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

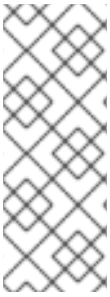
2. **ipa host-show** 명령을 입력하고 호스트 이름을 지정합니다.

```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Description: Example host
  Locality: Lab
  Location: Lab
  Platform: Lenovo T61
  Operating system: CentOS 7
  Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
  MAC address: 08:00:27:E3:B1:2D, 52:54:00:BD:97:1E
  Password: False
  Keytab: False
  Managed by: host01.idm.example.com
```

출력은 **IdM**에 **host01.idm.example.com** 이 있는지 확인합니다.

32.3. ANSIBLE PLAYBOOK을 사용하여 임의의 암호와 함께 여러 IDM 호스트 항목이 있는지 확인

ipahost 모듈을 사용하면 시스템 관리자가 하나의 **Ansible** 작업만 사용하여 **IdM**에 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 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.

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": "0HoIRvjUdH0Ycbf6uYdWTxH"},
"host02.idm.example.com": {"randompassword": "5VdLgrf3wvojmACdHC3uA3s"}}}
```

참고

임의의 일회성 암호(one-time passwords)를 사용하여 호스트를 IdM 클라이언트로 배포하려면 Ansible 플레이북 또는 회용 클라이언트를 한 번 사용하여 IdM 클라이언트 등록에 대한 권한 부여 옵션을 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.

32.4. 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 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.

절차

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

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

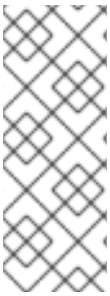
2. Ansible 플레이북 파일을 생성합니다. 확인하려는 IdM 에 있는 호스트의 정규화된 도메인 이름 (FQDN)을 ipahost 변수의 이름으로 지정합니다. ip_address 구문을 사용하여 별도의 행에 여러 IPv4 및 IPv6 *ip_address* 값을 지정합니다. 이 단계를 간소화하기 위해 /usr/share/doc/ansible-freeipa/playbooks/host/host-member-ipaddresses-present.yml 파일에서 예제를 복사하고 수정할 수 있습니다. 다음과 같은 추가 정보를 포함할 수도 있습니다.

```
---
- 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 **IdM**에 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.

32.5. 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 **IdM**에 없는 호스트의 정규화된 도메인 이름 **(FQDN)**을 사용하여 **Ansible** 플레이북 파일을 생성합니다. **IdM** 도메인에 통합된 **DNS**가 있는 경우 **updatedns: yes** 옵션을 사용하여 **DNS**에서 호스트의 연결된 레코드를 제거합니다.

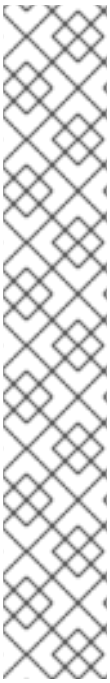
이 단계를 간소화하기 위해 **/usr/share/doc/ansible-freeipa/playbooks/host/delete-host.yml** 파일에서 예제를 복사하고 수정할 수 있습니다.

```
---
- 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.

32.6. 추가 리소스

- /usr/share/doc/ansible-freeipa/README-host.md Markdown 파일을 참조하십시오.
- /usr/share/doc/ansible-freeipa/playbooks/host 디렉터리에서 추가 플레이북을 참조하십시오.

33장. IDM CLI를 사용하여 호스트 그룹 관리

이 장에서는 IdM(Identity Management)의 호스트 그룹을 소개하고 CLI(명령줄 인터페이스)에서 호스트 그룹과 멤버를 관리하기 위한 다음 작업을 설명합니다.

- 호스트 그룹 및 해당 구성원 보기
- 호스트 그룹 생성
- 호스트 그룹 삭제
- 호스트 그룹 구성원 추가
- 호스트 그룹 구성원 제거
- 호스트 그룹 멤버 관리자 추가
- 호스트 그룹 구성원 관리자 제거

33.1. IDM의 호스트 그룹

IdM 호스트 그룹은 중요한 관리 작업, 특히 액세스 제어에 대한 제어를 중앙 집중화하는 데 사용할 수 있습니다.

호스트 그룹의 정의

호스트 그룹은 공통 액세스 제어 규칙 및 기타 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.

33.2. CLI를 사용하여 IDM 호스트 그룹 보기

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 IdM 호스트 그룹을 보는 방법에 대해 설명합니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

1. ipa hostgroup-find 명령을 사용하여 모든 호스트 그룹을 찾습니다.

```
$ ipa hostgroup-find
------------------
1 hostgroup matched
------------------
  Host-group: ipaservers
  Description: IPA server hosts
----------------------------
Number of entries returned 1
----------------------------
```

호스트 그룹의 모든 특성을 표시하려면 **--all** 옵션을 추가합니다. 예를 들면 다음과 같습니다.

```
$ ipa hostgroup-find --all
------------------
1 hostgroup matched
------------------
  dn: cn=ipaservers,cn=hostgroups,cn=accounts,dc=idm,dc=local
  Host-group: ipaservers
  Description: IPA server hosts
  Member hosts: xxx.xxx.xxx.xxx
  ipauniqueid: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
  objectclass: top, groupOfNames, nestedGroup, ipaobject, ipahostgroup
----------------------------
Number of entries returned 1
----------------------------
```

33.3. CLI를 사용하여 IDM 호스트 그룹 생성

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 IdM 호스트 그룹을 생성하는 방법에 대해 설명합니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

1. ipa hostgroup-add 명령을 사용하여 호스트 그룹을 추가합니다.
예를 들어 group *_name이라는 IdM 호스트 그룹을* 생성하고 설명을 제공하려면 다음을 수행합니다.

```
$ ipa hostgroup-add --desc 'My new host group' group_name
--------------------
Added hostgroup "group_name"
--------------------
  Host-group: group_name
  Description: My new host group
--------------------
```

33.4. CLI를 사용하여 IDM 호스트 그룹 삭제

이 섹션에서는 CLI(명령줄 인터페이스)를 사용하여 IdM 호스트 그룹을 삭제하는 방법을 설명합니다.

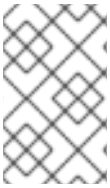
사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

절차

1. ipa hostgroup-del 명령을 사용하여 호스트 그룹을 삭제합니다.
예를 들어 group _*name* 이라는 IdM 호스트 그룹을 삭제하려면 다음을 수행합니다.

```
$ ipa hostgroup-del group_name
-------------------------
Deleted hostgroup "group_name"
-------------------------
```



참고

그룹을 제거해도 IdM에서 그룹 구성원이 삭제되지 않습니다.

33.5. CLI를 사용하여 IDM 호스트 그룹 멤버 추가

단일 명령을 사용하여 호스트 및 호스트 그룹을 IdM 호스트 그룹에 구성원을 추가할 수 있습니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.
- *선택사항*입니다. ipa hostgroup-find 명령을 사용하여 호스트 및 호스트 그룹을 찾습니다.

절차

1. 멤버를 호스트 그룹에 추가하려면 **ipa hostgroup-add-member** 를 사용하고 관련 정보를 제공합니다. 다음 옵션을 사용하여 추가할 멤버 유형을 지정할 수 있습니다.

- **hosts** 옵션을 사용하여 **IdM** 호스트 그룹에 하나 이상의 호스트를 추가합니다.
예를 들어 *example_member라는 호스트를 group_ name 그룹에* 추가하려면 다음을 실행합니다.

```
$ ipa hostgroup-add-member group_name --hosts example_member
Host-group: group_name
Description: My host group
Member hosts: example_member
-------------------------
Number of members added 1
-------------------------
```

- **host groups** 옵션을 사용하여 **IdM** 호스트 그룹에 하나 이상의 호스트 그룹을 추가합니다.
예를 들어 *nested_group이라는 호스트 그룹을 group_ name이라는 그룹에* 추가하려면 :

```
$ ipa hostgroup-add-member group_name --hostgroups nested_group
Host-group: group_name
Description: My host group
Member host-groups: nested_group
-------------------------
Number of members added 1
-------------------------
```

- 다음 구문을 사용하여 하나의 단일 명령으로 여러 호스트와 여러 호스트 그룹을 **IdM** 호스트 그룹에 추가할 수 있습니다.

```
$ ipa hostgroup-add-member group_name --hosts={host1,host2} --hostgroups=
{group1,group2}
```

중요

호스트 그룹을 다른 호스트 그룹의 구성원으로 추가할 때 재귀 그룹을 생성하지 마십시오. 예를 들어 그룹 **A**가 그룹 **B**의 구성원인 경우 그룹 **A**의 구성원으로 그룹 **B**를 추가하지 마십시오. 반복적인 그룹은 예측할 수 없는 동작이 발생할 수 있습니다.

33.6. CLI를 사용하여 IDM 호스트 그룹 멤버 제거

단일 명령을 사용하여 IdM 호스트 그룹에서 호스트 및 호스트 그룹을 제거할 수 있습니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.
- *선택사항*입니다. ipa hostgroup-find 명령을 사용하여 그룹에 제거할 멤버가 포함되어 있는지 확인합니다.

절차

1. 호스트 그룹 구성원을 제거하려면 ipa hostgroup-remove-member 명령을 사용하고 관련 정보를 제공합니다. 다음 옵션을 사용하여 제거할 멤버 유형을 지정할 수 있습니다.

   - IdM 호스트 그룹에서 하나 이상의 호스트를 제거하려면 --hosts 옵션을 사용합니다.
     예를 들어 이름이 group_ *name* 인 그룹에서 *example_member* 라는 호스트를 제거하려면 다음을 수행하십시오.

     ```
     $ ipa hostgroup-remove-member group_name --hosts example_member
     Host-group: group_name
     Description: My host group
     ----------------------
     Number of members removed 1
     ----------------------
     ```

   - hostnames 옵션을 사용하여 IdM 호스트 그룹에서 하나 이상의 호스트 그룹을 제거합니다.
     예를 들어, *nested_group* 이라는 호스트 그룹을 *group_name*:

     ```
     $ ipa hostgroup-remove-member group_name --hostgroups example_member
     Host-group: group_name
     Description: My host group
     ----------------------
     Number of members removed 1
     ----------------------
     ```

참고

그룹을 제거해도 IdM에서 그룹 구성원이 삭제되지 않습니다.

- 다음 구문을 사용하여 하나의 단일 명령으로 IdM 호스트 그룹에서 여러 호스트와 여러 호스트 그룹을 제거할 수 있습니다.

```
$ ipa hostgroup-remove-member group_name --hosts={host1,host2} --hostgroups={group1,group2}
```

33.7. CLI를 사용하여 IDM 호스트 그룹 멤버 관리자 추가

단일 명령을 사용하여 호스트 및 호스트 그룹과 구성원 관리자를 IdM 호스트 그룹에 추가할 수 있습니다. 멤버 관리자는 IdM 호스트 그룹에 호스트 또는 호스트 그룹을 추가할 수 있지만 호스트 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- 활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.
- 멤버 관리자로 추가하려는 호스트 또는 호스트 그룹의 이름과 관리할 호스트 그룹의 이름이 있어야 합니다.

절차

1. *선택사항*입니다. ipa hostgroup-find 명령을 사용하여 호스트 및 호스트 그룹을 찾습니다.
2. 멤버 관리자를 호스트 그룹에 추가하려면 ipa hostgroup-add-member-manager 를 사용합니다.

예를 들어, *example_member* 라는 사용자를 멤버 관리자로 group *_name이라는 그룹에* 추가하려면 다음을 수행합니다.

```
$ ipa hostgroup-add-member-manager group_name --user example_member
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: project_admins
Member of netgroups: group_name
Membership managed by users: example_member
-------------------------
Number of members added 1
-------------------------
```

3. 하나 이상의 호스트 그룹을 **IdM** 호스트 그룹에 멤버 관리자로 추가하려면 **--groups** 옵션을 사용합니다.

예를 들어 *admin_group이라는 호스트 그룹을 멤버 관리자로 group_ name이라는 그룹에* 추가하려면 다음을 수행합니다.

```
$ ipa hostgroup-add-member-manager group_name --groups admin_group
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: project_admins
Member of netgroups: group_name
Membership managed by groups: admin_group
Membership managed by users: example_member
-------------------------
Number of members added 1
-------------------------
```

참고

호스트 그룹에 멤버 관리자를 추가한 후 업데이트에 **Identity Management** 환경의 모든 클라이언트에 분배하는 데 다소 시간이 걸릴 수 있습니다.

검증 단계

- **ipa group-show** 명령을 사용하여 호스트 사용자 및 호스트 그룹이 멤버 관리자로 추가되었는지 확인합니다.

```
$ ipa hostgroup-show group_name
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: project_admins
Membership managed by groups: admin_group
Membership managed by users: example_member
```

추가 리소스

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

- 자세한 내용은 **ipa hostgroup-show --help** 를 참조하십시오.

33.8. CLI를 사용하여 IDM 호스트 그룹 멤버 관리자 제거

단일 명령을 사용하여 IdM 호스트 그룹에서 멤버 관리자뿐만 아니라 호스트 그룹을 제거할 수 있습니다. 멤버 관리자는 IdM 호스트 그룹에서 호스트 그룹 구성원 관리자를 제거할 수 있지만 호스트 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

- 활성 Kerberos 티켓. 자세한 내용은 kinit를 사용하여 IdM에 수동으로 로그인하는 방법을 참조하십시오.

- 제거 중인 기존 멤버 관리자 호스트 그룹의 이름과 관리 중인 호스트 그룹의 이름이 있어야 합니다.

절차

1. *선택사항*입니다. **ipa hostgroup-find** 명령을 사용하여 호스트 및 호스트 그룹을 찾습니다.

2. 호스트 그룹에서 멤버 관리자를 제거하려면 **ipa hostgroup-remove-member-manager** 명령을 사용합니다.

   예를 들어 **group_** ***name*** 이라는 그룹에서 멤버 관리자로 ***example_member*** 라는 사용자를 제거하려면 다음을 수행하십시오.

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

```
-------------------------
Number of members removed 1
-------------------------
```

3. IdM 호스트 그룹에서 멤버 관리자로 하나 이상의 호스트 그룹을 제거하려면 **--groups** 옵션을 사용합니다.

예를 들어 group_ *name* 이라는 그룹에서 멤버 관리자로 *nested_group* 이라는 호스트 그룹을 제거하려면 다음을 수행하십시오.

```
$ ipa hostgroup-remove-member-manager group_name --groups nested_group
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: project_admins
Member of netgroups: group_name
-------------------------
Number of members removed 1
-------------------------
```

참고

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

검증 단계

- ipa group-show 명령을 사용하여 호스트 사용자 및 호스트 그룹이 멤버 관리자로 제거되었는지 확인합니다.

```
$ ipa hostgroup-show group_name
Host-group: group_name
Member hosts: server.idm.example.com
Member host-groups: project_admins
```

추가 리소스

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

34장. IDM 웹 UI를 사용하여 호스트 그룹 관리

이 장에서는 IdM(Identity Management)의 호스트 그룹을 소개하고 웹 인터페이스(웹 UI)에서 호스트 그룹과 멤버를 관리하기 위한 다음 작업을 설명합니다.

- 호스트 그룹 및 해당 구성원 보기
- 호스트 그룹 생성
- 호스트 그룹 삭제
- 호스트 그룹 구성원 추가
- 호스트 그룹 구성원 제거
- 호스트 그룹 멤버 관리자 추가
- 호스트 그룹 구성원 관리자 제거

34.1. IDM의 호스트 그룹

IdM 호스트 그룹은 중요한 관리 작업, 특히 액세스 제어에 대한 제어를 중앙 집중화하는 데 사용할 수 있습니다.

호스트 그룹의 정의

호스트 그룹은 공통 액세스 제어 규칙 및 기타 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.

34.2. IDM 웹 UI에서 호스트 그룹 보기

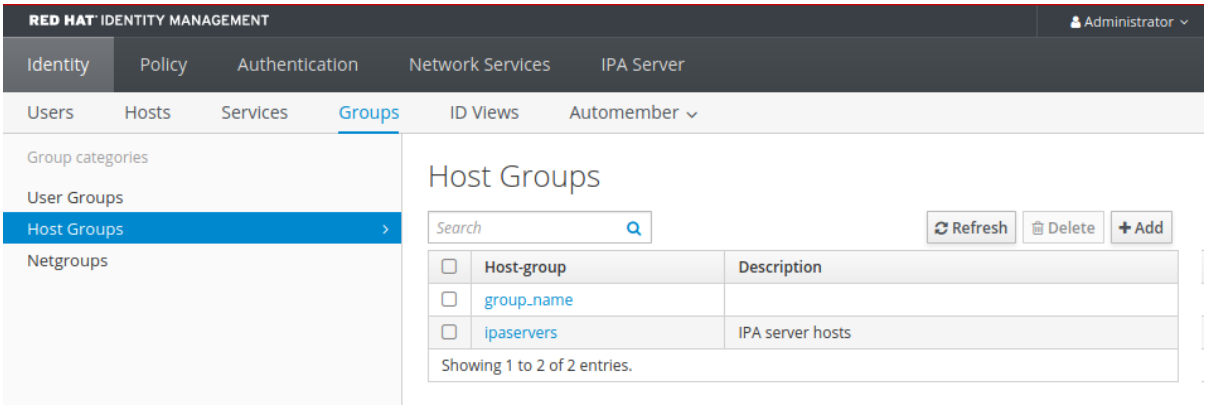
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM 호스트 그룹을 보는 방법에 대해 설명합니다.

사전 요구 사항

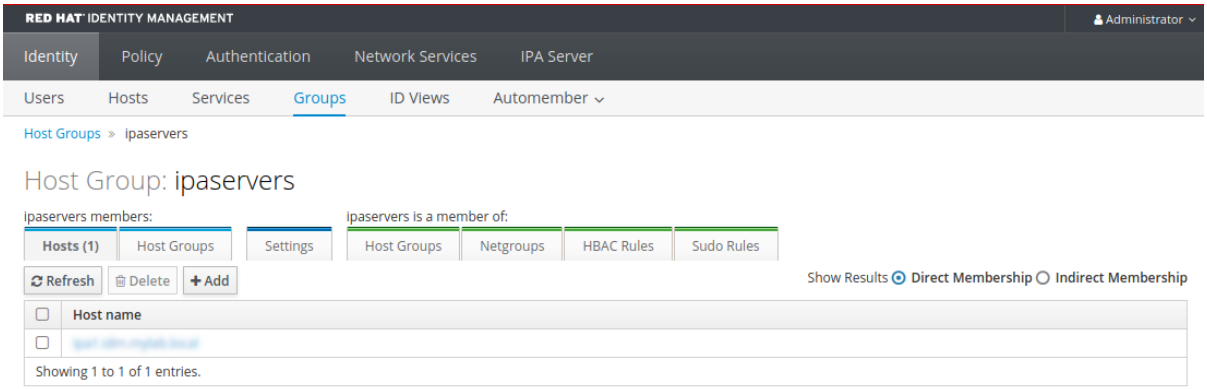
- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

절차

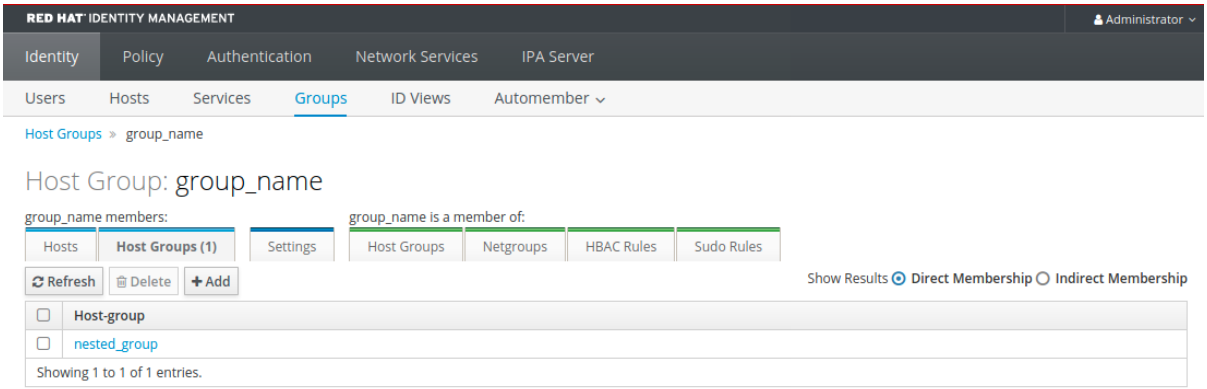
1. Identity → Groups(ID → 그룹)를 클릭하고 Host Groups(호스트 그룹) 탭을 선택합니다.
   - 페이지에는 기존 호스트 그룹과 해당 설명이 나열됩니다.
   - 특정 호스트 그룹을 검색할 수 있습니다.



2. 목록에서 그룹을 클릭하여 이 그룹에 속하는 호스트를 표시합니다. 직접 또는 간접 구성원으로 결과를 제한할 수 있습니다.



3. **Host Groups**(호스트 그룹) 탭을 선택하여 이 그룹(**nested** 호스트 그룹)에 속하는 호스트 그룹을 표시합니다. 직접 또는 간접 구성원으로 결과를 제한할 수 있습니다.



34.3. IDM 웹 UI에서 호스트 그룹 생성

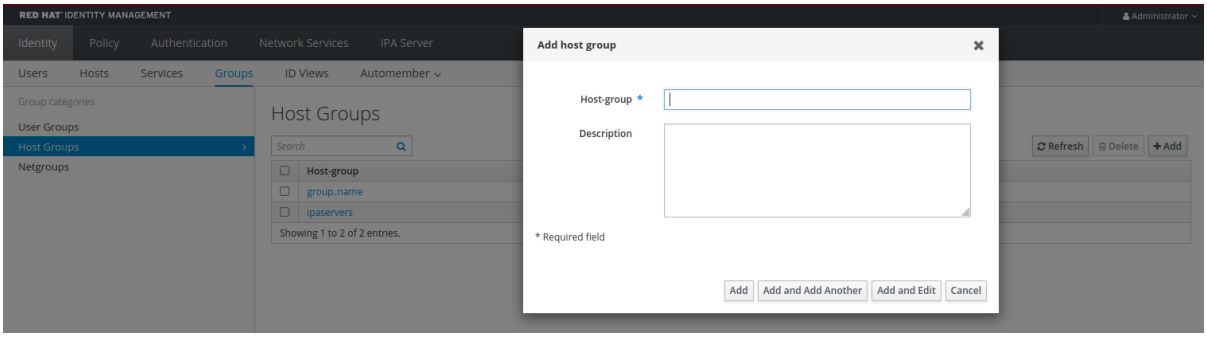
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM 호스트 그룹을 생성하는 방법에 대해 설명합니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

절차

1. Identity → Groups(ID → 그룹)를 클릭하고 Host Groups(호스트 그룹) 탭을 선택합니다.
2. 추가를 클릭합니다. Add host group (호스트 그룹 추가) 대화 상자가 표시됩니다.
3. 그룹에 대한 정보 제공: name(필수) 및 설명(선택 사항).
4. Add(추가) 를 클릭하여 확인합니다.



34.4. IDM 웹 UI에서 호스트 그룹 삭제

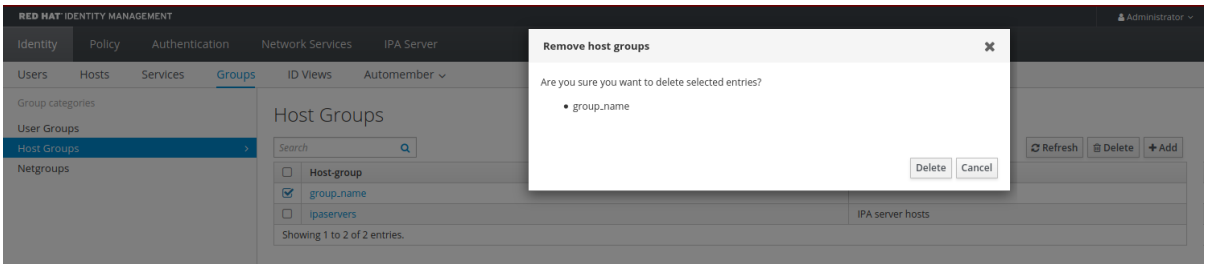
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM 호스트 그룹을 삭제하는 방법을 설명합니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

절차

1. ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.
2. 제거할 IdM 호스트 그룹을 선택하고 Delete(삭제)를 클릭합니다. 확인 대화 상자가 나타납니다.
3. 삭제를 클릭하여 확인합니다.



참고

호스트 그룹을 제거해도 IdM의 그룹 구성원이 삭제되지 않습니다.

34.5. IDM 웹 UI에서 호스트 그룹 구성원 추가

이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM에 호스트 그룹 구성원을 추가하는 방법을 설명합니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

절차

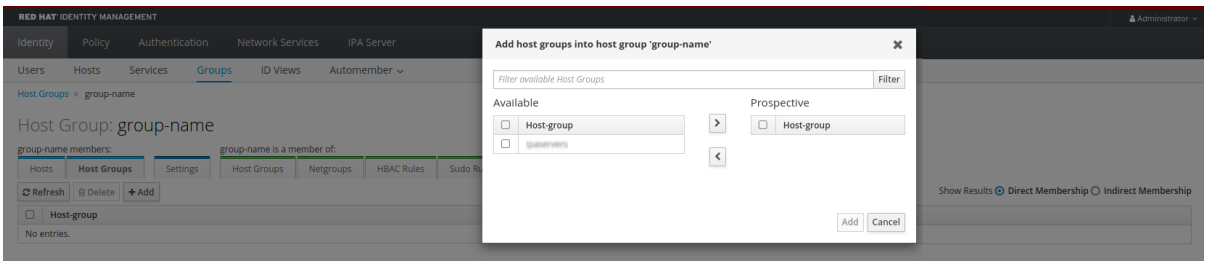
1. ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.

2. 멤버를 추가할 그룹의 이름을 클릭합니다.

3. 추가할 구성원 유형에 따라 Hosts(호스트) 또는 Host groups (호스트 그룹) 탭을 클릭합니다. 해당 대화 상자가 나타납니다.

4. 추가할 호스트 또는 호스트 그룹을 선택하고 > 화살표 버튼을 클릭하여 Prospective 열로 이동합니다.

5. Add(추가) 를 클릭하여 확인합니다.



34.6. IDM 웹 UI에서 호스트 그룹 구성원 제거

이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM에서 호스트 그룹 구성원을 제거하는 방법을 설명합니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.

- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.

절차

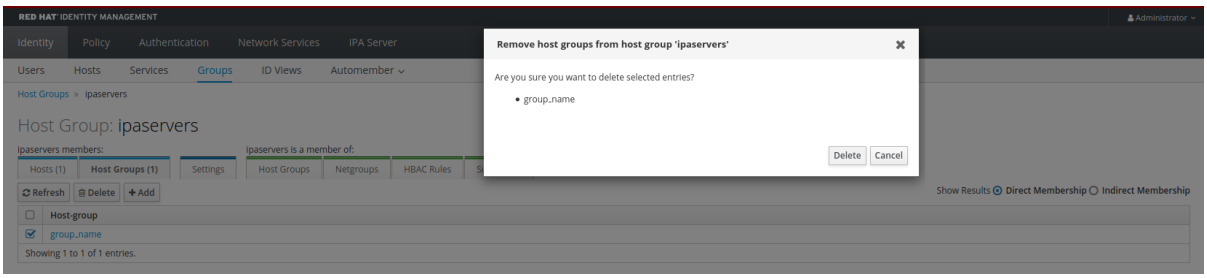
1. ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.

2. 멤버를 제거할 그룹의 이름을 클릭합니다.

3. 제거할 구성원 유형에 따라 Hosts(호스트) 또는 Host groups (호스트 그룹) 탭을 클릭합니다.

4. 제거할 멤버 옆에 있는 확인란을 선택합니다.

5. Delete(삭제)를 클릭합니다. 확인 대화 상자가 나타납니다.



6. Delete(삭제)를 클릭하여 확인합니다. 선택한 멤버가 삭제됩니다.

34.7. 웹 UI를 사용하여 IDM 호스트 그룹 멤버 관리자 추가

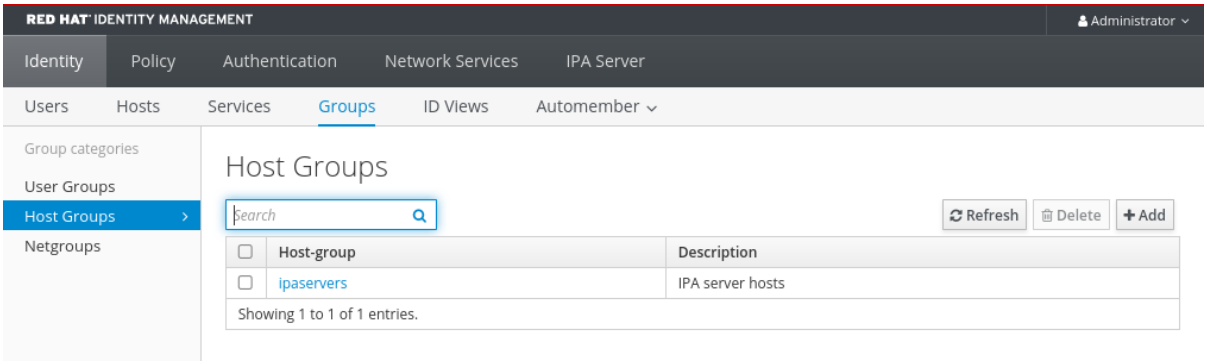
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM에서 사용자 또는 사용자 그룹을 호스트 그룹 멤버 관리자로 추가하는 방법에 대해 설명합니다. 멤버 관리자는 IdM 호스트 그룹에 호스트 그룹 구성원 관리자를 추가할 수 있지만 호스트 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

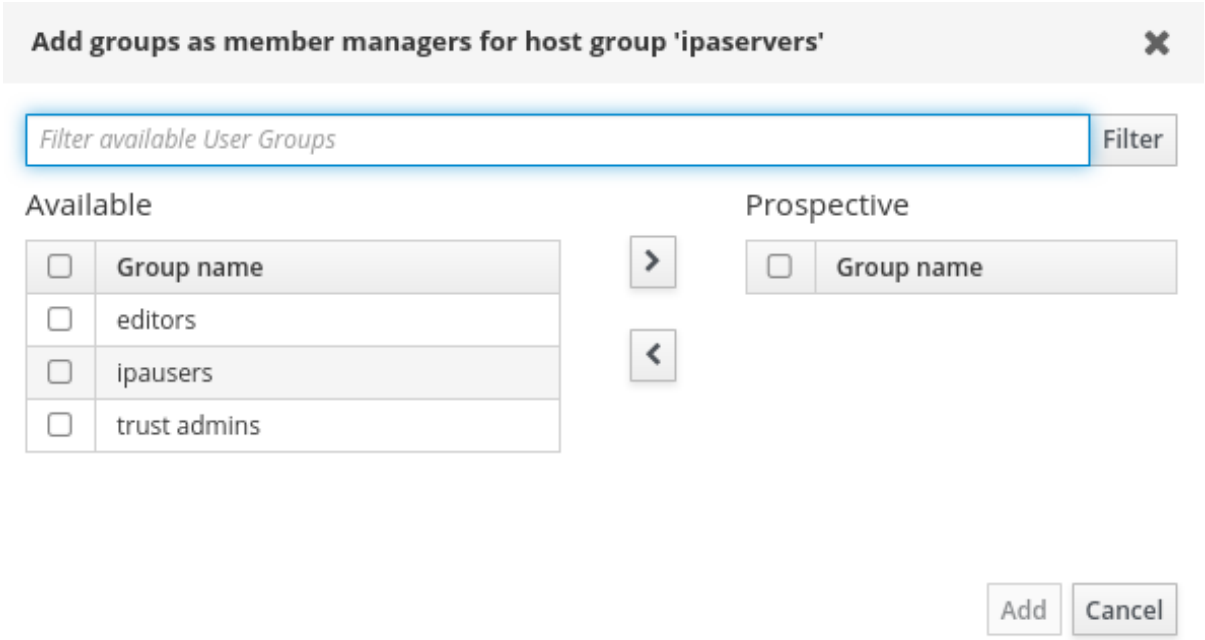
- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.
- 멤버 관리자로 추가하려는 호스트 그룹 이름과 관리할 호스트 그룹의 이름이 있어야 합니다.

절차

1. ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.

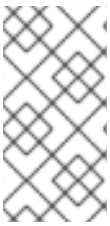


2. 멤버 관리자를 추가할 그룹의 이름을 클릭합니다.
3. 추가하려는 멤버 관리자 유형에 따라 멤버 managerss(사용자 그룹) 또는 Users (사용자) 탭을 클릭합니다. 해당 대화 상자가 나타납니다.
4. 추가를 클릭합니다.



5. 추가할 사용자 또는 사용자 그룹을 선택하고 **>** 화살표 버튼을 클릭하여 **Prospective** 열로 이동합니다.

6. **Add(추가)** 를 클릭하여 확인합니다.

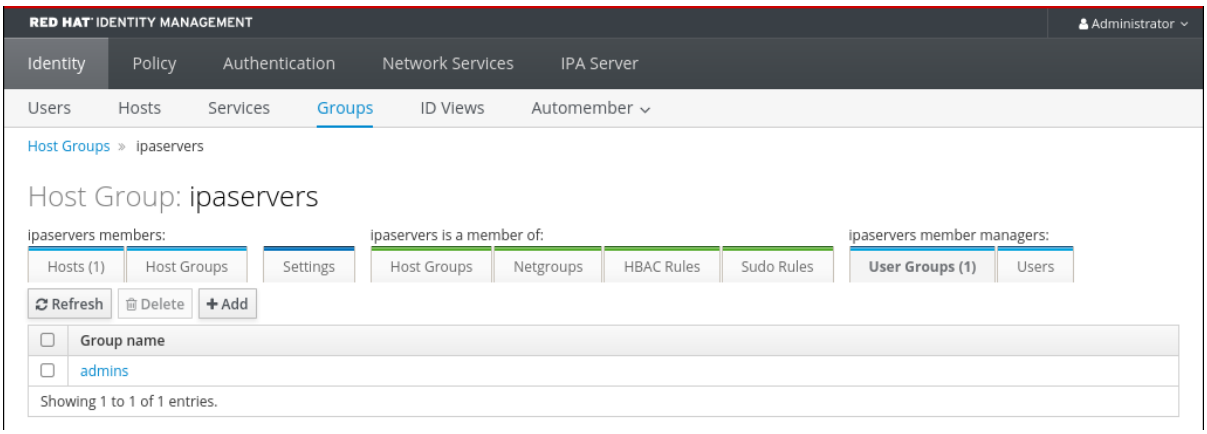


참고

호스트 그룹에 멤버 관리자를 추가한 후 업데이트에 **Identity Management** 환경의 모든 클라이언트에 분배하는 데 다소 시간이 걸릴 수 있습니다.

검증 단계

- **Host Group(호스트 그룹)** 대화 상자에서 사용자 그룹 또는 사용자가 그룹 또는 사용자의 멤버 관리자 목록에 추가되었는지 확인합니다.



34.8. 웹 UI를 사용하여 IDM 호스트 그룹 멤버 관리자 제거

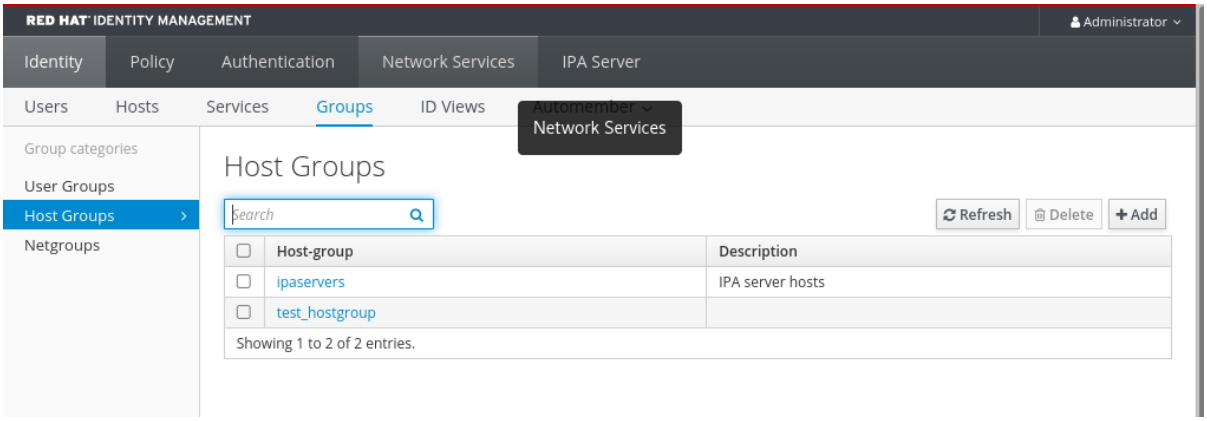
이 섹션에서는 웹 인터페이스(웹 UI)를 사용하여 IdM에서 사용자 또는 사용자 그룹을 호스트 그룹 멤버 관리자로 제거하는 방법을 설명합니다. 멤버 관리자는 IdM 호스트 그룹에서 호스트 그룹 구성원 관리자를 제거할 수 있지만 호스트 그룹의 특성을 변경할 수는 없습니다.

사전 요구 사항

- IdM 또는 사용자 관리자 역할을 관리하기 위한 관리자 권한.
- IdM 웹 UI에 로그인되어 있습니다. 자세한 내용은 웹 브라우저에서 IdM 웹 UI 액세스를 참조하십시오.
- 제거 중인 기존 멤버 관리자 호스트 그룹의 이름과 관리 중인 호스트 그룹의 이름이 있어야 합니다.

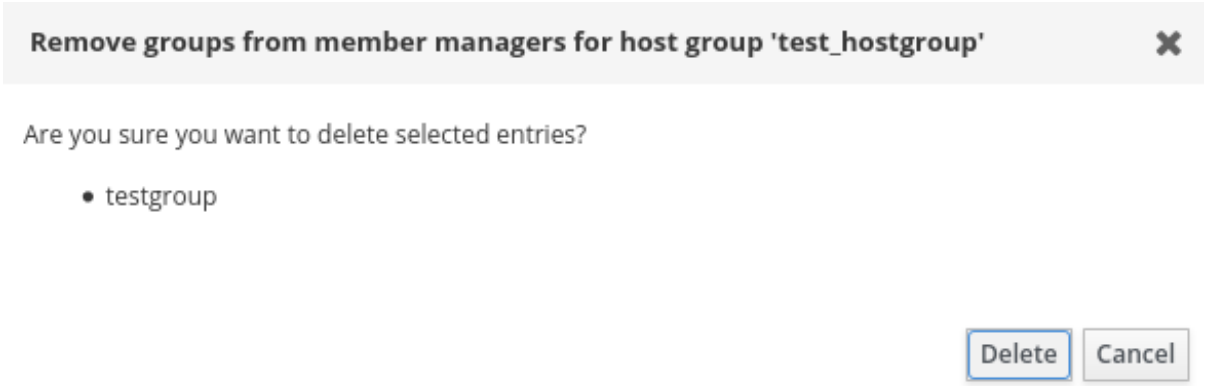
절차

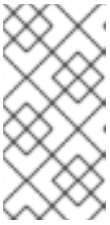
1. ID → 그룹을 클릭하고 호스트 그룹 탭을 선택합니다.



2. 멤버 관리자를 제거할 그룹의 이름을 클릭합니다.
3. 제거할 멤버 관리자의 유형에 따라 멤버 managerss(사용자 그룹) 또는 Users (사용자) 탭을 클릭합니다. 해당 대화 상자가 나타납니다.
4. 제거할 사용자 또는 사용자 그룹을 선택하고 Delete(삭제)를 클릭합니다.

5. **Delete(삭제)** 를 클릭하여 확인합니다.



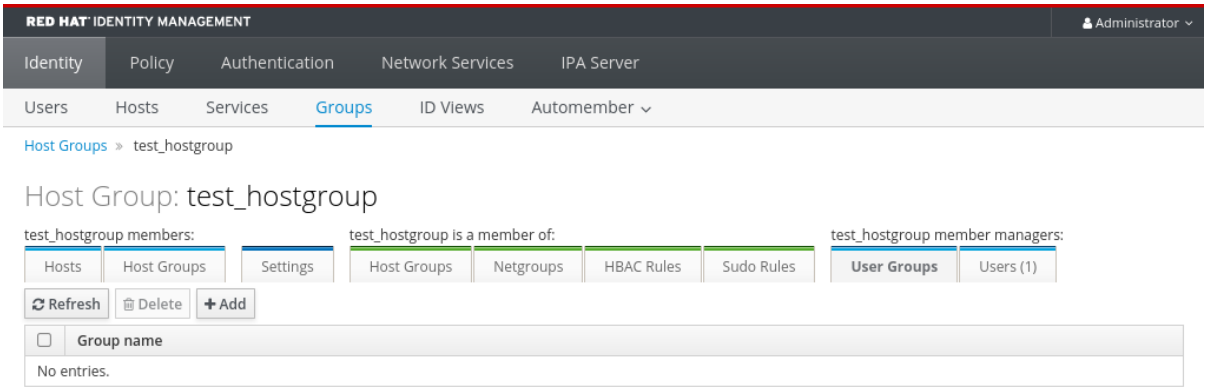


참고

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

검증 단계

- **Host Group(호스트 그룹)** 대화 상자에서 사용자 그룹 또는 사용자가 그룹 또는 사용자의 멤버 관리자 목록에서 제거되었는지 확인합니다.



35장. 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

35.1. IDM의 호스트 그룹

IdM 호스트 그룹은 중요한 관리 작업, 특히 액세스 제어에 대한 제어를 중앙 집중화하는 데 사용할 수 있습니다.

호스트 그룹의 정의

호스트 그룹은 공통 액세스 제어 규칙 및 기타 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.

35.2. 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.

35.3. 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.

35.4. 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 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. 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.

35.5. 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 --help를 참조하십시오.
- ipa 도움말 페이지를 참조하십시오.

35.6. 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.

35.7. 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 IdM에서 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.

35.8. 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.

35.9. 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.

36장. ANSIBLE 플레이북을 사용하여 IDM에 호스트 기반 액세스 제어 규칙이 있는지 확인

이 장에서는 IdM(Identity Management) 호스트 기반 액세스 정책과 Ansible 을 사용하여 정의하는 방법에 대해 설명합니다.

Ansible은 시스템을 구성하고, 소프트웨어를 배포하고, 롤링 업데이트를 수행하는 데 사용되는 자동화 도구입니다. IdM(Identity Management) 지원이 포함되어 있습니다.

36.1. 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.

36.2. 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 컨트롤러에 ansible-freeipa 패키지를 설치했습니다.

- 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

- /usr/share/doc/ansible-freeipa 디렉터리의README-hbacsvcgroup.md, README-hbacrule.md 파일을 참조하십시오.

- /usr/share/doc/ansible-freeipa/playbooks 디렉터리의 하위 디렉터리에 있는 플레이북을 참조하십시오.

37장. 짧은 AD 사용자 이름 확인 순서 구성

기본적으로 user_name@domain.com 또는 domain.com\user_name 형식으로 정규화된 이름을 지정하여 AD(Active Directory) 환경에서 사용자와 그룹을 확인하고 인증해야 합니다. 다음 섹션에서는 짧은 AD 사용자 이름 및 그룹 이름을 확인하도록 IdM 서버 및 클라이언트를 구성하는 방법을 설명합니다.

- 도메인 확인 순서가 작동하는 방식
- IdM 서버에서 글로벌 도메인 확인 순서 설정
- IdM 서버에서 ID 보기에 대한 도메인 확인 순서 설정
- IdM 클라이언트의 SSSD에서 도메인 확인 순서 설정

37.1. 도메인 확인 순서가 작동하는 방식

AD(Active Directory) 신뢰가 있는 IdM(Identity Management) 환경에서는 정규화된 이름을 지정하여 사용자와 그룹을 확인하고 인증하는 것이 좋습니다. 예를 들면 다음과 같습니다.

- *<idm_username>* idm.example.com 도메인의 IdM 사용자를 위한 @idm.example.com
- *<ad_username>*@ad.example.com 도메인의 AD 사용자용 @ad .example.com

기본적으로 ad_username 과 같은 *짧은 이름* 형식을 사용하여 사용자 또는 그룹 조회를 수행하는 경우 IdM 도메인만 검색하고 AD 사용자 또는 그룹을 찾지 못합니다. 단축 이름을 사용하여 AD 사용자 또는 그룹을 확인하려면 도메인 확인 순서 옵션을 설정하여 IdM이 여러 도메인을 검색하는 순서를 변경합니다.

IdM 데이터베이스 또는 개별 클라이언트의 SSSD 구성에서 도메인 확인 순서를 중앙에서 설정할 수 있습니다. IdM은 도메인 확인 순서를 다음 순서로 평가합니다.

- 로컬 /etc/sssd/sssd.conf 구성.

- **ID** 보기 구성입니다.
- 글로벌 **IdM** 구성입니다.

참고

- 호스트의 **SSSD** 구성에 **default_domain_suffix** 옵션이 포함되어 있고 이 옵션으로 지정되지 않은 도메인에 요청하려는 경우 정규화된 사용자 이름을 사용해야 합니다.
- 도메인 확인 순서 옵션을 사용하고 **compat** 트리를 쿼리하는 경우 여러 **UID**(사용자 **ID**)가 수신될 수 있습니다. 이것이 영향을 줄 수 있는 경우 도메인 확인 순서가 설정된 경우 **Pagure** 버그 보고서 **Inconsistent compat user objects for AD users** 를 참조하십시오.

중요

IdM 클라이언트 또는 **IdM** 서버에서 **full_name_format SSSD** 옵션을 사용하지 마십시오. 이 옵션에 기본값이 아닌 값을 사용하면 사용자 이름이 표시되는 방식이 변경되고 **IdM** 환경에서 조회가 중단될 수 있습니다.

추가 리소스

- 레거시 **Linux** 클라이언트를 위한 **Active Directory** 트러스트.

37.2. IDM 서버에서 글로벌 도메인 확인 순서 설정

이 절차에서는 **IdM** 도메인에 있는 모든 클라이언트에 대한 도메인 확인 순서를 설정합니다. 이 예제에서는 사용자 및 그룹을 다음 순서로 검색하는 도메인 확인 순서를 설정합니다.

1. **AD(Active Directory)** 루트 도메인 **ad.example.com**
2. **AD** 하위 도메인**1.ad.example.com**
3. **IdM domain idm.example.com**

사전 요구 사항

- AD 환경을 사용하여 신뢰를 구성했습니다.

절차

- ipa config-mod --domain- resolution-order 명령을 사용하여 선호하는 순서로 검색할 도메인을 나열합니다. 도메인을 콜론(:)으로 구분합니다.

```
[user@server ~]$ ipa config-mod --domain-resolution-
order='ad.example.com:subdomain1.ad.example.com:idm.example.com'
Maximum username length: 32
Home directory base: /home
...
  Domain Resolution Order:
ad.example.com:subdomain1.ad.example.com:idm.example.com
...
```

검증 단계

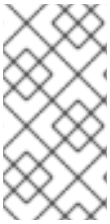
- 짧은 이름만 사용하여 ad.example.com 도메인에서 사용자의 사용자 정보를 검색할 수 있는지 확인합니다.

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

37.3. IDM 서버에서 ID 보기에 대한 도메인 확인 순서 설정

이 절차에서는 특정 IdM 서버 및 클라이언트 집합에 적용할 수 있는 ID 보기의 도메인 확인 순서를 설정합니다. 이 예제에서는 IdM 호스트 client1.idm.example.com 에 대한 ADsubdomain1_first 라는 ID 뷰를 생성하고 다음 순서로 사용자와 그룹을 검색하도록 도메인 확인 순서를 설정합니다.

1. AD(Active Directory) 하위 도메인1.ad.example.com
2. AD root 도메인 ad.example.com
3. IdM domain idm.example.com



참고

ID 보기에 설정된 도메인 확인 순서는 글로벌 도메인 확인 순서를 재정의하지만 SSSD 구성에서 로컬로 설정된 도메인 확인 순서는 재정의하지 않습니다.

사전 요구 사항

- AD 환경을 사용하여 신뢰를 구성했습니다.

절차

1. domain- resolution -order 옵션을 설정하여 ID 보기를 만듭니다.

```
[user@server ~]$ ipa idview-add ADsubdomain1_first --desc "ID view for resolving AD
subdomain1 first on client1.idm.example.com" --domain-resolution-order
subdomain1.ad.example.com:ad.example.com:idm.example.com
---------------------------------
Added ID View "ADsubdomain1_first"
---------------------------------
ID View Name: ADsubdomain1_first
Description: ID view for resolving AD subdomain1 first on client1.idm.example.com
Domain Resolution Order:
subdomain1.ad.example.com:ad.example.com:idm.example.com
```

2. IdM 호스트에 ID 보기를 적용합니다.

```
[user@server ~]$ ipa idview-apply ADsubdomain1_first --hosts
client1.idm.example.com
---------------------------------
Applied ID View "ADsubdomain1_first"
---------------------------------
  hosts: client1.idm.example.com
-----------------------------------------
Number of hosts the ID View was applied to: 1
-----------------------------------------
```

검증 단계

- ID 보기의 세부 정보를 표시합니다.

```
[user@server ~]$ ipa idview-show ADsubdomain1_first --show-hosts
  ID View Name: ADsubdomain1_first
  Description: ID view for resolving AD subdomain1 first on client1.idm.example.com
```

```
 Hosts the view applies to: client1.idm.example.com
 Domain resolution order:
subdomain1.ad.example.com:ad.example.com:idm.example.com
```

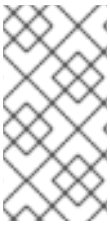
- subdomain1.ad.example.com 도메인에서 짧은 이름만 사용하여 사용자에 대한 사용자 정보를 검색할 수 있는지 확인합니다.

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

37.4. IDM 클라이언트의 SSSD에서 도메인 확인 순서 설정

이 절차에서는 IdM 클라이언트의 SSSD 구성에서 도메인 확인 순서를 설정합니다. 이 예제에서는 IdM 호스트 client2.idm.example.com 을 구성하여 다음 순서대로 사용자와 그룹을 검색합니다.

1. AD(Active Directory) 하위 도메인1.ad.example.com
2. AD root 도메인 ad.example.com
3. IdM domain idm.example.com



참고

로컬 SSSD 구성의 도메인 확인 순서는 모든 글로벌 및 ID 보기 도메인 확인 순서를 재정의합니다.

사전 요구 사항

- AD 환경을 사용하여 신뢰를 구성했습니다.

절차

1. 텍스트 편집기에서 /etc/sssd/sssd.conf 파일을 엽니다.
2. 파일의 [sssd] 섹션에서 domain_ resolution_order 옵션을 설정합니다.

```
domain_resolution_order = subdomain1.ad.example.com, ad.example.com,
idm.example.com
```

3. 파일을 저장하고 종료합니다.

4. SSSD 서비스를 다시 시작하여 새 구성 설정을 로드합니다.

```
[root@client2 ~]# systemctl restart sssd
```

검증 단계

- subdomain1.ad.example.com 도메인에서 짧은 이름만 사용하여 사용자에 대한 사용자 정보를 검색할 수 있는지 확인합니다.

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

37.5. 추가 리소스

- ID 보기를 사용하여 IdM 클라이언트에서 사용자 속성 값을 재정의

38장. IDM에서 AD 사용자 주체 이름을 사용한 인증 활성화

38.1. IDM에서 신뢰하는 ADFOREST의 사용자 주요 이름

IdM(Identity Management) 관리자는 AD 사용자가 대체 UUPN(User Principal Names)을 사용하여 IdM 도메인의 리소스에 액세스할 수 있습니다. UPN은 AD 사용자가 user_name@KERBEROS-REALM 형식으로 인증하는 대체 사용자 로그인입니다. AD 관리자는 AD 포리스트에서 추가 Kerberos 별칭과 UPN 접미사를 둘 다 구성할 수 있으므로 user_name 및 KERBEROS-REALM 모두에 대한 대체 값을 설정할 수 있습니다.

예를 들어 회사에서 Kerberos 영역 AD.EXAMPLE.COM 을 사용하는 경우 사용자의 기본 UPN은 user@ad.example.com 입니다. 사용자가 이메일 주소(예: user@example.com)를 사용하여 로그인할 수 있도록 하려면 AD에서 EXAMPLE.COM 을 대체 UPN으로 구성할 수 있습니다. 대체 UPN(*엔터프라이즈 UPN이라고도 함*)은 회사에서 최근에 병합을 경험했으며 사용자에게 통합 로그온 네임스페이스를 제공하고자 하는 경우 특히 편리합니다.

UPN 접미사는 AD forest 루트에 정의된 경우에만 IdM에 대해 표시됩니다. AD 관리자는 Active Directory Domain and Trust 유틸리티 또는 PowerShell 명령줄 도구를 사용하여 UPN을 정의할 수 있습니다.

참고

사용자에 대한 UPN 접미사를 구성하려면 Active Directory Domain and Trust 유틸리티와 같이 오류 유효성 검사를 수행하는 툴을 사용하는 것이 좋습니다.

Active Directory는 이러한 작업을 검증하지 않기 때문에 ldapmodify 명령을 사용하여 사용자에 대한 userPrincipalName 속성을 설정하는 등 낮은 수준의 수정을 통해 UPN을 구성하는 것을 권장하고 있습니다.

AD 측에서 새 UPN을 정의한 후 IdM 서버에서 ipa trust-fetch-domains 명령을 실행하여 업데이트된 UPN을 검색합니다. AD UPN이 IdM에서 최신 상태인지를 참조하십시오.

IdM은 cn=trusted_domain_name,cn=ad,cn=trusts,dc=idm,dc=example,dc=com 의 subtree 속성 ipaNTAdditionalSuffixes 에 도메인의 UPN 접미사를 저장합니다.

추가 리소스

- AD forest 루트에서 UPN 접미사 설정을 스크립팅하는 방법

- AD 사용자 항목을 수동으로 수정하고 UPN 접미사 검증을 무시하는 방법
- 신뢰 컨트롤러 및 신뢰 에이전트

38.2. AD UPN이 IDM에서 최신 상태인지 확인

신뢰할 수 있는 Active Directory(AD) 포리스트에 UPN(User Principal Name) 접미사를 추가하거나 제거한 후 IdM 서버에서 신뢰할 수 있는 포리스트에 대한 정보를 새로 고칩니다.

사전 요구 사항

- IdM 관리자 자격 증명.

절차

- ipa trust-fetch-domains 명령을 입력합니다. 빈 출력은 다음과 같이 예상됩니다.

```
[root@ipaserver ~]# ipa trust-fetch-domains
Realm-Name: ad.example.com
-----------------------------
No new trust domains were found
-----------------------------
----------------------------
Number of entries returned 0
----------------------------
```

검증 단계

- ipa trust-show 명령을 입력하여 서버가 새 UPN을 가져왔는지 확인합니다. 메시지가 표시되면 AD 영역의 이름을 지정합니다.

```
[root@ipaserver ~]# ipa trust-show
Realm-Name: ad.example.com
  Realm-Name: ad.example.com
  Domain NetBIOS name: AD
  Domain Security Identifier: S-1-5-21-796215754-1239681026-23416912
  Trust direction: One-way trust
  Trust type: Active Directory domain
  UPN suffixes: example.com
```

출력에 **example.com UPN** 접미사가 이제 **ad.example.com** 영역 항목의 일부임을 보여줍니다.

38.3. AD UPN 인증 문제의 문제 해결 데이터 수집

다음 절차에서는 **AD(Active Directory)** 환경과 **IdM** 환경의 **User Principal Name(UPN)** 구성에 대한 문제 해결 데이터를 수집하는 방법을 설명합니다. **AD** 사용자가 대체 **UPN**을 사용하여 로그인할 수 없는 경우 이 정보를 사용하여 문제 해결 작업을 줄일 수 있습니다.

사전 요구 사항

- **AD** 도메인 컨트롤러에서 정보를 검색하려면 **IdM** 신뢰 컨트롤러 또는 신뢰 에이전트에 로그인해야 합니다.
- 다음 구성 파일을 수정하고 **IdM** 서비스를 다시 시작하려면 **root** 권한이 필요합니다.

절차

1. 텍스트 편집기에서 **/usr/share/ipa/smb.conf.empty** 구성 파일을 엽니다.
2. 파일에 다음 콘텐츠를 추가합니다.

```
[global]
log level = 10
```

3. **/usr/share/ipa/smb.conf.empty** 파일을 저장하고 닫습니다.
4. 텍스트 편집기에서 **/etc/ipa/server.conf** 구성 파일을 엽니다. 해당 파일이 없는 경우 파일을 만듭니다.
5. 파일에 다음 콘텐츠를 추가합니다.

```
[global]
debug = True
```

6. **/etc/ipa/server.conf** 파일을 저장하고 닫습니다.

7. Apache 웹 서버를 다시 시작하여 구성 변경 사항을 적용합니다.

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

8. AD 도메인에서 신뢰 정보를 검색합니다.

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

9. 다음 로그 파일에서 디버깅 출력 및 문제 해결 정보를 검토합니다.

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

추가 리소스

- AD UPN 인증 문제에 대한 문제 해결 데이터를 수집하려면 Using-02-client 를 참조하십시오.

39장. AD 사용자가 IDM을 관리하도록 활성화

39.1. AD 사용자의 ID 덮어쓰기

RHEL(Red Hat Enterprise Linux) 7에서 외부 그룹 멤버십을 통해 AD(Active Directory) 사용자 및 그룹이 SSSD(System Security Services Daemon)를 활용하여 POSIX 환경에서 IdM(Identity Management) 리소스에 액세스할 수 있습니다.

IdM LDAP 서버에는 액세스 제어 권한을 부여하는 자체 메커니즘이 있습니다. RHEL 8에서는 AD 사용자에 대한 ID 사용자 재정의를 IdM 그룹의 멤버로 추가할 수 있는 업데이트가 도입되었습니다. ID 재정의는 특정 ID 보기(이 경우 Default Trust View) 내에서 특정 Active Directory 사용자 또는 그룹 속성이 어떻게 보이는지를 설명하는 레코드입니다. 업데이트 결과 IdM LDAP 서버에서 IdM 그룹에 대한 액세스 제어 규칙을 AD 사용자에게 적용할 수 있습니다.

이제 AD 사용자가 IdM UI의 셀프 서비스 기능을 사용하여 SSH 키를 업로드하거나 개인 데이터를 변경하는 등의 작업을 수행할 수 있습니다. AD 관리자가 두 개의 서로 다른 계정 및 암호 없이도 IdM을 완전히 관리할 수 있습니다.

참고

현재 AD 사용자가 IdM에서 선택한 기능을 사용할 수 없습니다. 예를 들어 IdM admins 그룹에서 AD 사용자로 IdM 사용자의 암호를 설정하는 데 실패할 수 있습니다.

추가 리소스

- Active Directory 사용자를 위한 ID 보기 사용

39.2. ID 덮어쓰기를 사용하여 AD 사용자가 IDM 관리 활성화

이 절차에서는 AD 사용자가 IdM 사용자와 동일한 사용자 권한을 제공하기 위한 ID 재정의 생성 및 사용에 대해 설명합니다. 이 절차 중에 신뢰 컨트롤러 또는 신뢰 에이전트로 구성된 IdM 서버를 사용합니다.

사전 요구 사항

- idm:DL1 스트림은 IdM(Identity Management) 서버에서 활성화되며 이 스트림을 통해 제공되는 RPM으로 전환했습니다.

```
# yum module enable idm:DL1
# yum distro-sync
```

- idm:DL1/adtrust 프로필이 IdM 서버에 설치되어 있습니다.

```
# yum module install idm:DL1/adtrust
```

프로필에는 ipa-idoverride-memberof 패키지를 포함하여 AD(Active Directory)와 신뢰 계약이 있는 IdM 서버를 설치하는 데 필요한 모든 패키지가 포함되어 있습니다.

- 작동 중인 IdM 환경이 설정됩니다. 자세한 내용은 Identity Management 설치를 참조하십시오.

- IdM 환경과 AD 간의 작업 신뢰가 설정됩니다.

절차

1. IdM 관리자는 Default Trust View에서 AD 사용자에 대한 ID 재정의를 생성합니다. 예를 들어 ad_user@ad.example.com 사용자에 대한 ID 덮어쓰기를 생성하려면 다음을 수행합니다.

```
# kinit admin
# ipa idoverrideuser-add 'default trust view' ad_user@ad.example.com
```

2. Default Trust View의 ID 재정의를 IdM 그룹에 멤버로 추가합니다. 문제의 그룹이 IdM 역할의 멤버인 경우 ID 재정의에서 표시하는 AD 사용자는 명령줄 인터페이스와 IdM 웹 UI를 모두 포함하여 IdM API를 사용할 때 역할에서 부여하는 모든 권한을 얻습니다. 예를 들어 ad_user@ad.example.com 사용자의 ID 덮어쓰기를 admins 그룹에 추가하려면 다음을 수행합니다.

```
# ipa group-add-member admins --idoverrideusers=ad_user@ad.example.com
```

추가 리소스

- Active Directory 사용자를 위한 ID 보기 사용

39.3. IDM CLI를 AD 사용자로 관리

이 절차에서는 AD(Active Directory) 사용자가 IdM(Identity Management) CLI(명령줄 인터페이스)에 로그인하고 역할에 적합한 명령을 실행할 수 있는지 확인합니다.

1.

IdM 관리자의 현재 Kerberos 티켓을 삭제합니다.

```
# kdestroy -A
```

참고

MIT Kerberos의 GSSAPI 구현에서 기본 설정에 따라 대상 서비스의 자격 증명을 선택하기 때문에 Kerberos 티켓 제거가 필요합니다. 이 경우에는 IdM 영역입니다. 즉, 인증 정보 수집, 즉 KCM:, KEYRING:, 또는 DIR: 인증 정보 캐시 유형이 사용 중인 경우 이전에 받은 admin 또는 기타 IdM 주체의 자격 증명이 AD 사용자 자격 증명 대신 IdM API에 액세스하는 데 사용됩니다.

2.

ID 재정의가 생성된 AD 사용자의 Kerberos 자격 증명을 가져옵니다.

```
# kinit ad_user@AD.EXAMPLE.COM
Password for ad_user@AD.EXAMPLE.COM:
```

3.

AD 사용자의 ID 재정의가 해당 그룹의 IdM 그룹과 IdM 그룹의 멤버십에서 발생하는 동일한 권한을 사용하는지 테스트합니다. AD 사용자의 ID 재정의가 admins 그룹에 추가된 경우 AD 사용자는 IdM에 그룹을 생성할 수 있습니다.

```
# ipa group-add some-new-group
---------------------------
Added group "some-new-group"
---------------------------
  Group name: some-new-group
  GID: 1997000011
```