Red Hat Enterprise Linux 9

ID 관리 계획

Identity Management 계획 및 액세스 제어 설정 설명서

Last Updated: 2022-07-10

Red Hat Enterprise Linux 9 ID 관리 계획

Identity Management 계획 및 액세스 제어 설정 설명서

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 문서에서는 Red Hat Enterprise Linux 9에서 Identity Management 서비스 계획에 대해 설명합니다.

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.

- ***블록 목록은*** *블랙리스트*대체
- ***허용 목록*** 교체 *허용 목록*
- 두 ***번째는*** *슬레이브*를 대체합니다.
- 컨텍스트에 따라 *master* 라는 단어가 더 정확한 언어로 대체됩니다.
  - ***IdM 서버가*** *IdM 마스터*교체
  - ***CA 갱신 서버가*** *CA 갱신 마스터*를 대체
  - ***CRL 게시자 서버가*** *CRL 마스터*를 대체
  - ***Multi-supplier*** replace *multi-master*

RED HAT 문서에 관한 피드백 제공

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

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

planning-identity-management:parent-context-of-overview-of-planning-idm-and-access-control:
planning-identity-identity-identity-management

1장. RHEL의 IDM 계획 및 액세스 제어 계획 개요

다음 섹션에서는 Red Hat Enterprise Linux의 ID 관리(IdM) 및 액세스 제어에 대한 옵션에 대한 개요를 제공합니다. 이러한 섹션을 읽은 후에는 환경의 계획 단계에 접근할 수 있습니다.

1.1. IDM 소개

이 모듈은 Red Hat Enterprise Linux의 IdM(Identity Management)의 용도를 설명합니다. 또한 도메인에 속하는 클라이언트 및 서버 시스템을 포함하여 IdM 도메인에 대한 기본 정보를 제공합니다.

Red Hat Enterprise Linux의 IdM 목표

Red Hat Enterprise Linux의 IdM은 Linux 기반 도메인에서 ID 저장소, 인증, 정책 및 권한 부여 정책을 중앙 집중식으로 관리할 수 있는 통합된 방법을 제공합니다. IdM은 다양한 서비스를 개별적으로 관리하고 다른 시스템에서 다른 툴을 사용하는 관리 오버헤드를 크게 줄입니다.

IdM은 다음을 지원하는 소수의 중앙 집중식 ID, 정책 및 권한 부여 소프트웨어 솔루션 중 하나입니다.

- Linux 운영 체제 환경의 고급 기능
- 대규모 Linux 머신 그룹 통합
- Active Directory와의 기본 통합

IdM은 Linux 기반 및 Linux 제어 도메인을 생성합니다.

- IdM은 기존 네이티브 Linux 툴 및 프로토콜을 기반으로 합니다. 자체 프로세스 및 구성이 있지만 기본 기술은 Linux 시스템에 제대로 설정되고 Linux 관리자가 신뢰합니다.
- IdM 서버 및 클라이언트는 Red Hat Enterprise Linux 시스템입니다. IdM 클라이언트는 표준 프로토콜을 지원하는 경우 다른 Linux 및 UNIX 배포판일 수도 있습니다. Windows 클라이언트는 IdM 도메인의 멤버가 될 수 없지만 AD(Active Directory)에서 관리하는 Windows 시스템에 로그인한 사용자는 Linux 클라이언트에 연결하거나 IdM에서 관리하는 서비스에 액세스할 수 있습니다. 이 작업은 AD 및 IdM 도메인 간에 크로스est 트러스트를 설정하여 수행됩니다.

여러 Linux 서버에서 ID 및 정책 관리

IdM 없음: 각 서버를 별도로 관리합니다. 모든 암호는 로컬 시스템에 저장됩니다. IT 관리자는 모든 시스템에서 사용자를 관리하고 인증 및 권한 부여 정책을 별도로 설정하고 로컬 암호를 유지 관리합니다. 그러나 사용자가 다른 중앙 집중식 솔루션에 의존하는 경우가 많을수록 AD와 직접 통합할 수 있습니다. 시스템은 여러 가지 솔루션을 사용하여 AD와 직접 통합될 수 있습니다.

- 기존 Linux 툴(사용 안 함)
- Samba winbind 기반 솔루션(특정 사용 사례에 권장)
- 타사 소프트웨어를 기반으로 하는 솔루션 (일반적으로 다른 공급 업체의 라이센스가 필요)
- SSSD를 기반으로 하는 솔루션(기본 Linux 및 대부분의 사용 사례에 권장되는)

IdM 사용: IT 관리자가 다음을 수행할 수 있습니다.

- IdM 서버 한 곳에서 ID를 유지 관리
- 여러 대의 컴퓨터에 동시에 정책을 일관되게 적용하십시오.

- 호스트 기반 액세스 제어, 위임 및 기타 규칙을 사용하여 사용자에 대해 다른 액세스 수준을 설정합니다.
- 권한 에스컬레이션 규칙 중앙 관리
- 홈 디렉터리 마운트 방법 정의

Enterprise SSO

IdM Enterprise의 경우 SSO(Single Sign-On)는 Kerberos 프로토콜을 활용하여 구현됩니다. 이 프로토콜은 인프라 수준에서 널리 사용되고 있으며 SSH, LDAP, NFS, CUPS 또는 DNS와 같은 서비스를 사용하여 SSO를 활성화합니다. 다양한 웹 스택을 사용하는 웹 서비스(Apache, EAP, Django 등)도 SSO에 Kerberos를 사용하도록 활성화할 수 있습니다. 그러나 실제로 SSO 기반 OpenID Connect 또는 SAML을 사용하는 것이 웹 애플리케이션에 더 편리하다는 것을 보여줍니다. 두 계층을 브리지하려면 Kerberos 인증을 OpenID Connect 티켓 또는 SAML 어설션으로 변환할 수 있는 IdM(Identity Provider) 솔루션을 배포하는 것이 좋습니다. Keycloak 오픈 소스 프로젝트를 기반으로 하는 Red Hat SSO 기술은 이러한 IdP의 예입니다.

IdM 없음: 사용자가 서비스 또는 애플리케이션에 액세스할 때마다 시스템에 로그인하고 암호를 입력하라는 메시지가 표시됩니다. 이러한 암호는 다를 수 있으며 사용자는 어떤 애플리케이션에 사용할 인증 정보를 기억해야 합니다.

Idm: 사용자가 시스템에 로그인한 후 자격 증명을 반복적으로 요청하지 않고 여러 서비스 및 애플리케이션에 액세스할 수 있습니다. 이 도움말은 다음과 같습니다.

- 유용성 개선
- 암호 쓰기가 중단되거나 안전하지 않게 저장된 보안 위험을 줄일 수 있습니다.
- 사용자 생산성 향상

혼합 Linux 및 Windows 환경 관리

IdM 없음: Windows 시스템은 AD forest에서 관리되지만 개발, 프로덕션 및 기타 팀에는 많은 Linux 시스템이 있습니다. Linux 시스템은 AD 환경에서 제외됩니다.

IdM 사용: IT 관리자가 다음을 수행할 수 있습니다.

- 기본 Linux 툴을 사용하여 Linux 시스템 관리
- Linux 시스템을 Active Directory에서 관리하는 환경에 통합하므로 중앙 집중식 사용자 저장소를 유지합니다.
- 새로운 Linux 시스템을 확장하거나 필요에 따라 쉽게 배포할 수 있습니다.
- 비즈니스 요구에 신속하게 대응하고 다른 팀에 종속되지 않고 Linux 인프라 관리와 관련된 결정을 내릴 수 있습니다.

IdM과 표준 LDAP 디렉터리 비교

Red Hat Directory Server와 같은 표준 LDAP 디렉토리는 범용 디렉터리입니다. 다양한 사용 사례에 맞게 사용자 지정할 수 있습니다.

- 스키마: 사용자, 시스템, 네트워크 엔티티, 물리적 장비 또는 빌딩과 같은 광범위한 항목에 대해 사용자 지정할 수 있는 유연한 스키마입니다.
- 일반적으로 인터넷에서 서비스를 제공하는 비즈니스 애플리케이션과 같은 다른 애플리케이션의 데이터를 저장하는 백엔드 디렉터리입니다.

IdM은 이러한 ID와 관련된 내부 및 엔터프라이즈 ID와 인증 및 권한 부여 정책 등의 특정 목적을 가지고 있습니다.

- 스키마: 사용자 또는 시스템 ID 항목에 대한 항목 등 해당 용도와 관련된 특정 항목 세트를 정의하는 특정 스키마입니다.
- 일반적으로 엔터프라이즈 또는 프로젝트의 경계 내에서 ID를 관리하는 ID 및 인증 서버로 사용됩니다.

기본 디렉터리 서버 기술은 Red Hat Directory Server 및 IdM 모두에서 동일합니다. 그러나 IdM은 회사 내부의 ID를 관리하기 위해 최적화되어 있습니다. 이를 통해 일반적인 확장성을 제한하지만 특정 이점, 간편한 구성, 리소스 관리 향상, 엔터프라이즈 ID 관리 효율성 향상 등의 이점을 제공합니다.

추가 리소스

- Identity Management 또는 Red Hat Directory Server - Red Hat Enterprise Linux 블로그에서 하나의 사용 중단?
- 표준 프로토콜에 대한 지식 베이스 문서 .
- Red Hat Enterprise Linux 8 베타 릴리스 노트

1.2. IDM 서버 및 클라이언트 소개

IdM(Identity Management) 도메인에는 다음과 같은 유형의 시스템이 포함되어 있습니다.

IdM 서버

IdM 서버는 IdM 도메인 내의 ID, 인증 및 권한 부여 요청에 응답하는 Red Hat Enterprise Linux 시스템입니다. 대부분의 배포에서는 통합 CA(인증 기관)도 IdM 서버와 함께 설치됩니다.
IdM 서버는 ID 및 정책 정보를 위한 중앙 리포지토리입니다. IdM 서버는 도메인 멤버가 사용하는 모든 선택적 서비스를 호스팅할 수도 있습니다.

- 인증 기관 (CA)
- 키 복구 기관(KRA)
- DNS
- Active Directory (AD) 신뢰 컨트롤러
- Active Directory (AD) 신뢰 에이전트

IdM 클라이언트

IdM 클라이언트는 서버에 등록된 Red Hat Enterprise Linux 시스템으로, 이러한 서버에서 IdM 서비스를 사용하도록 구성되어 있습니다.
클라이언트는 IdM 서버와 상호 작용하여 해당 서버에서 제공하는 서비스에 액세스합니다. 예를 들어 클라이언트는 Kerberos 프로토콜을 사용하여 인증을 수행하고 엔터프라이즈급 SSO(Single Sign-On)에 대한 티켓을 취득하고, LDAP를 사용하여 ID와 정책의 출처와 서버 및 서비스가 있는 위치 및 연결 방법을 감지합니다.

IdM 서버도 IdM 클라이언트가 포함되어 있습니다. 고객이 직접 등록한 클라이언트로서 서버는 다른 클라이언트와 동일한 기능을 제공합니다.

다수의 클라이언트에 서비스를 제공하고 중복성 및 가용성에 대해 IdM을 통해 단일 도메인에 여러 IdM 서

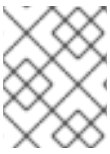
버에 배포할 수 있습니다. 최대 60대의 서버를 배포할 수 있습니다. 이는 현재 IdM 도메인에서 지원되는 복제본이라고도 하는 최대 IdM 서버 수입니다. IdM 서버는 클라이언트에 다양한 서비스를 제공합니다. 모든 서버가 가능한 모든 서비스를 제공해야 하는 것은 아닙니다. Kerberos 및 LDAP와 같은 일부 서버 구성 요소는 모든 서버에서 항상 사용할 수 있습니다. CA, DNS, Trust Controller 또는 Vault와 같은 기타 서비스는 선택 사항입니다. 이는 일반적으로 배포에서 다양한 서버가 다양한 역할을 수행한다는 것을 의미합니다.

IdM 토폴로지에 통합된 CA가 포함된 경우 하나의 서버에 CRL(인증서 해지 목록) 게시자 서버의 역할이 있으며, 한 서버에 CA 갱신 서버의 역할이 있습니다. 기본적으로 설치된 첫 번째 CA 서버는 이러한 두 가지 역할을 충족하지만 이러한 역할을 별도의 서버에 할당할 수 있습니다.

주의

*CA 갱신 서버*는 CA 하위 시스템 인증서 및 키 추적을 담당하는 도메인의 유일한 시스템이므로 IdM 배포에 중요합니다. IdM 배포에 영향을 미치는 재해에서 복구하는 방법에 대한 자세한 내용은 ID 관리를 사용하여 재해 복구 수행을 참조하십시오.

중복 및 로드 밸런싱을 위해 관리자는 기존 서버의 *복제본*을 생성하여 추가 서버를 생성합니다. 복제본을 생성할 때 IdM에서 기존 서버의 구성을 복제합니다. 복제본은 사용자, 시스템, 인증서, 구성된 정책에 대한 내부 정보를 포함하여 초기 서버와 해당 코어 구성을 공유합니다.



참고

생성된 복제본 및 서버는 *CA 갱신* 및 *CRL 게시자* 역할을 제외하고 기능적으로 동일합니다. 따라서 컨텍스트에 따라 ***서버*** 및 ***복제본*** 이라는 용어는 서로 바꿔 사용할 수 있습니다.

1.3. RHEL의 IDM 및 액세스 제어: 중앙과 로컬.

Red Hat Enterprise Linux에서는 전체 시스템 도메인 또는 단일 시스템의 로컬 툴을 사용하여 중앙 집중식 툴을 사용하여 ID 및 액세스 제어 정책을 관리할 수 있습니다.

여러 Red Hat Enterprise Linux 서버에서 ID 및 정책 관리: IdM 사용 및

ID 관리 IdM을 사용하면 IT 관리자가 다음을 수행할 수 있습니다.

- IdM 서버 한 곳에서 ID 및 그룹화 메커니즘을 유지 관리
- 암호, PKI 인증서, OTP 토큰 또는 SSH 키와 같은 다양한 유형의 인증 정보를 중앙에서 관리합니다.
- 여러 대의 컴퓨터에 동시에 정책을 일관되게 적용하십시오.
- 외부 Active Directory 사용자를 위한 POSIX 및 기타 속성 관리
- 호스트 기반 액세스 제어, 위임 및 기타 규칙을 사용하여 사용자에 대해 다른 액세스 수준을 설정합니다.
- 권한 에스컬레이션 규칙(sudo) 및 SELinux 사용자 매핑(SELinux 사용자 매핑)을 중앙에서 관리합니다.
- 중앙 PKI 인프라 및 시크릿 저장소 유지

- 홈 디렉터리 마운트 방법 정의

IdM이 없는 경우:

- 각 서버는 개별적으로 관리됩니다.
- 모든 암호는 로컬 시스템에 저장됩니다.
- IT 관리자는 모든 시스템에서 사용자를 관리하고 인증 및 권한 부여 정책을 별도로 설정하고 로컬 암호를 유지 관리합니다.

1.4. IDM 용어

Active Directory 고려 사항

AD(Active Directory)est는 공통 글로벌 카탈로그, 디렉터리 스키마, 논리 구조 및 디렉터리 구성을 공유하는 하나 이상의 도메인 트리 집합입니다. est는 사용자, 컴퓨터, 그룹 및 기타 개체에 액세스할 수 있는 보안 경계를 나타냅니다. 자세한 내용은 Forests 에 대한 Microsoft 문서를 참조하십시오.

Active Directory 글로벌 카탈로그

전역 카탈로그는 개체가 도메인 컨트롤러 도메인의 멤버인지 여부와 관계없이 도메인 컨트롤러에서 포리스트의 모든 개체에 대한 정보를 제공할 수 있는 AD(Active Directory)의 기능입니다.The global catalog is a feature of Active Directory (AD) that allows a domain controller to provide information on any object in the forest, regardless of whether the object is a member of the domain controller's domain. 글로벌 카탈로그 기능이 활성화된 도메인 컨트롤러를 글로벌 카탈로그 서버라고 합니다. 글로벌 카탈로그는 다중 도메인 AD DS(Active Directory Domain Services)의 모든 도메인에 있는 모든 개체에 대한 검색 가능한 카탈로그를 제공합니다.The global catalog provides a searchable catalog of all objects in every domain in a multi-domain Active Directory Domain Services (AD DS).

Active Directory 보안 식별자

SID(Security identifier)는 사용자, 그룹 또는 호스트와 같은 Active Directory의 개체에 할당된 고유 ID 번호입니다. Linux의 UID 및 GID와 기능적으로 동일합니다.

Ansible 플레이

Ansible 플레이는 Ansible 플레이북 의 구성 요소입니다. 플레이의 목표는 호스트 그룹을 Ansible 작업으로 표시하는 몇 가지 잘 정의된 역할에 매핑하는 것입니다.

Ansible 플레이북

Ansible 플레이북은 하나 이상의 Ansible 플레이가 포함된 파일입니다. 자세한 내용은 플레이북에 대한 공식 Ansible 설명서를 참조하십시오.

Ansible 작업

Ansible 작업은 Ansible의 작업 단위입니다. Ansible 플레이는 여러 작업을 포함할 수 있습니다. 각 작업의 목표는 매우 구체적인 인수를 사용하여 모듈을 실행하는 것입니다. Ansible 작업은 광범위한 용어에서 특정 Ansible 역할 또는 모듈에 의해 정의된 상태를 달성하는 명령 세트이며 해당 역할 또는 모듈의 변수에 의해 미세 조정됩니다. 자세한 내용은 공식 Ansible 작업 설명서를 참조하십시오 .

Apache 웹 서버

Apache를 자주 호출한 Apache HTTP Server는 Apache License 2.0에 따라 제공되는 무료 오픈 소스 크로스 플랫폼 웹 서버 애플리케이션입니다. Apache는 World Wide Web의 초기 성장을 위해 중요한 역할을 했으며 현재 주요 HTTP 서버입니다. 프로세스 이름은 **httpd** 이며, *HTTP 데몬* 의 줄임말입니다. Red Hat IdM(Identity Management)은 Apache 웹 서버를 사용하여 IdM 웹 UI를 표시하고, Directory Server 및 인증 기관과 같은 구성 요소 간 통신을 조정합니다.

certificate

인증서는 개인, 서버, 회사 또는 기타 개체를 식별하고 해당 ID를 공개 키와 연결하는 데 사용되는 전자 문서입니다. 드라이버의 라이선스 또는 여권과 같이 인증서는 일반적으로 사람의 신원을 증명할 수 있습니다. 공개 키 암호화는 인증서를 사용하여 가장 문제를 해결합니다.

IdM의 CA(인증 기관)

디지털 인증서를 발급하는 엔티티입니다. Red Hat Identity Management에서 기본 CA는 IdM CA인 **ipa** 입니다. **ipa** CA 인증서는 다음 유형 중 하나입니다.

- 자체 서명. 이 경우 **ipa** CA는 루트 CA입니다.
- 외부에서 서명합니다. 이 경우 **ipa** CA는 외부 CA로 무효화됩니다.

IdM에서는 여러 **하위 CA** 를 생성할 수도 있습니다. 하위 CA는 인증서가 다음 유형 중 하나인 IdM CA 입니다.

- **ipa** CA에서 서명합니다.
- 자체 및 **ipa** CA 간의 중간 CA에서 서명합니다. 하위 CA의 인증서는 자체 서명할 수 없습니다.

cross-forest trust

신뢰는 두 개의 Kerberos 영역 간의 액세스 관계를 설정하여 한 도메인의 사용자와 서비스가 다른 도메인의 리소스에 액세스할 수 있도록 합니다.
AD(Active Directory) forest 루트 도메인과 IdM 도메인 간의 상호 간 신뢰로 AD forest 도메인의 사용자는 IdM 도메인의 Linux 시스템 및 서비스와 상호 작용할 수 있습니다. AD의 관점에서 Identity Management는 단일 AD 도메인을 사용하는 별도의 AD forest을 나타냅니다. 자세한 내용은 신뢰의 작동 방식을 참조하십시오.

디렉터리 서버

Directory Server는 사용자 ID 및 애플리케이션 정보를 중앙 집중화합니다. 애플리케이션 설정, 사용자 프로필, 그룹 데이터, 정책 및 액세스 제어 정보를 저장하기 위한 운영 체제 독립적 네트워크 기반 레지스트리를 제공합니다. 네트워크의 각 리소스는 디렉터리 서버에서 오브젝트로 간주됩니다. 특정 리소스에 대한 정보는 해당 리소스 또는 오브젝트와 연결된 속성의 컬렉션으로 저장됩니다. Red Hat Directory Server는 LDAP 표준을 준수합니다.

DNS PTR 레코드

PTR(DNS 포인터) 레코드는 호스트의 IP 주소를 도메인 또는 호스트 이름으로 확인합니다. PTR 레코드는 DNS A 및 AAAA 레코드와 반대로 호스트 이름을 IP 주소로 확인합니다. DNS PTR 레코드는 역방향 DNS 조회를 활성화합니다. PTR 레코드는 DNS 서버에 저장됩니다.

DNS SRV 레코드

DNS 서비스(SRV) 레코드는 도메인에서 사용 가능한 서비스의 호스트 이름, 포트 번호, 전송 프로토콜, 우선 순위 및 가중치를 정의합니다. SRV 레코드를 사용하여 IdM 서버 및 복제본을 찾을 수 있습니다.

DC(Domain Controller)

DC(Domain Controller)는 도메인 내의 보안 인증 요청에 응답하고 해당 도메인의 리소스에 대한 액세스를 제어하는 호스트입니다. IdM 서버는 IdM 도메인의 DC로 작동합니다. DC는 사용자를 인증하고, 사용자 계정 정보를 저장하고, 도메인에 대한 보안 정책을 적용합니다. 사용자가 도메인에 로그인하면 DC가 자격 증명을 인증하고 유효성을 검증하고 액세스를 허용하거나 거부합니다.

정규화된 도메인 이름

FQDN(정규화된 도메인 이름)은 DNS(Domain Name System)의 계층 구조 내에서 호스트의 정확한 위치를 지정하는 도메인 이름입니다. 상위 도메인 **example.com** 에 호스트 이름 **myhost** 가 있는 장치에는 FQDN **myhost.example.com** 이 있습니다. FQDN은 다른 도메인에서 **myhost** 라는 다른 호스트와 장치를 고유하게 구분합니다.
DNS 자동 검색을 사용하여 호스트 **machine1** 에 IdM 클라이언트를 설치하고 DNS 레코드가 올바르게 구성된 경우 **machine1** 의 FQDN이 필요한 모든 것입니다. 자세한 내용은 IdM에 대한 호스트 이름 및 DNS 요구 사항을 참조하십시오.

GSSAPI

개발자는 일반 보안 서비스 애플리케이션 인터페이스(GSSAPI 또는 GSS-API)를 통해 개발자는 애플리케이션이 피어 애플리케이션으로 전송되는 데이터를 보호하는 방법을 추상화할 수 있습니다. Security-service 공급업체는 보안 소프트웨어가 포함된 라이브러리로 일반적인 프로시저 호출의 GSSAPI 구현을 제공할 수 있습니다. 이러한 라이브러리는 벤더 독립적인 GSSAPI만 사용하도록 애플리케이션을 작성할 수 있는 애플리케이션 작성자에게 GSSAPI 호환 인터페이스를 제공합니다. 개발자는 이러한 유연성을 통해 특정 플랫폼, 보안 메커니즘, 보호 유형 또는 전송 프로토콜에 보안 구현을 조정할 필요가 없습니다.
Kerberos는 주요 GSSAPI 메커니즘 구현으로, Red Hat Enterprise Linux 및 Microsoft Windows Active Directory Kerberos 구현을 API와 호환할 수 있습니다.

숨겨진 복제본

숨겨진 복제본은 모든 서비스가 실행 중이고 사용 가능한 IdM 복제본이지만 서버 역할은 비활성화되어 있으며 DNS에 SRV 레코드가 없기 때문에 클라이언트는 복제본을 검색할 수 없습니다.
숨겨진 복제본은 주로 백업, 대량 가져오기 및 내보내기 또는 IdM 서비스를 종료해야 하는 작업과 같은 서비스를 위해 설계되었습니다. 숨겨진 복제본을 사용하지 않는 클라이언트가 없으므로 관리자는 클라이언트에 영향을 주지 않고 이 호스트의 서비스를 일시적으로 종료할 수 있습니다. 자세한 내용은 숨겨진 복제본 모드 를 참조하십시오.

HTTP 서버

웹 서버를 참조하십시오.

ID 매핑

SSSD는 AD 사용자의 SID를 사용하여 ID *매핑* 이라는 프로세스에서 POSIX ID를 알고리즘화할 수 있습니다. ID 매핑은 Linux의 AD ID와 AD의 SID 간 맵을 생성합니다.

- SSSD가 새 AD 도메인을 감지하면 사용 가능한 다양한 ID를 새 도메인에 할당합니다. 따라서 각 AD 도메인은 모든 SSSD 클라이언트 시스템에서 동일한 ID 범위를 갖습니다.
- AD 사용자가 SSSD 클라이언트 시스템에 처음으로 로그인하면 SSSD 캐시에 사용자 SID 및 해당 도메인의 ID 범위를 기반으로 UID를 포함하여 사용자에 대한 항목을 만듭니다.
- AD 사용자의 ID는 동일한 SID에서 일관된 방식으로 생성되므로 사용자는 Red Hat Enterprise Linux 시스템에 로그인할 때 동일한 UID 및 GID를 갖습니다.

ID 범위

ID 범위는 IdM 토폴로지 또는 특정 복제본에 할당된 ID 번호 범위입니다. ID 범위를 사용하여 새 사용자, 호스트 및 그룹의 유효한 UID 및 GID 범위를 지정할 수 있습니다. ID 범위는 ID 번호 충돌을 방지하는 데 사용됩니다. IdM에는 두 가지 유형의 ID 범위가 있습니다.

- *IdM ID 범위*
  이 ID 범위를 사용하여 전체 IdM 토폴로지에서 사용자 및 그룹의 UID 및 GID를 정의합니다. 첫 번째 IdM 서버를 설치하면 IdM ID 범위가 생성됩니다. IdM ID 범위를 생성한 후에는 수정할 수 없습니다. 그러나 원래의 고갈에 가까운 경우와 같이 추가 IdM ID 범위를 생성할 수 있습니다.
- *DNA(Distributed Numeric Assignment) ID 범위*
  이 ID 범위를 사용하여 복제본에서 새 사용자를 생성할 때 사용하는 UID 및 GID를 정의합니다. IdM 복제본에 새 사용자 또는 호스트 항목을 처음으로 추가하면 해당 복제본에 DNA ID 범위가 할당됩니다. 관리자는 DNA ID 범위를 수정할 수 있지만 새 정의가 기존 IdM ID 범위 내에 있어야 합니다.

  IdM 범위와 DNA 범위가 일치하지만 상호 연결되지는 않습니다. 하나의 범위를 변경하는 경우 다른 범위가 일치하도록 변경합니다.

자세한 내용은 ID 범위를 참조하십시오.

ID 보기

ID 보기를 사용하면 POSIX 사용자 또는 그룹 속성에 대한 새 값을 지정하고 새 값이 적용되는 클라이언트 호스트 또는 호스트를 정의할 수 있습니다. 예를 들어 ID 보기를 사용하여 다음을 수행할 수 있습니다.

- 다양한 환경에 대해 서로 다른 특성 값을 정의합니다.
- 이전에 생성된 속성 값을 다른 값으로 교체합니다.

IdM-AD 신뢰 설정에서 **Default Trust View** 는 AD 사용자 및 그룹에 적용되는 ID 보기입니다. **기본 신뢰 보기를** 사용하여 AD 사용자 및 그룹에 대해 사용자 지정 POSIX 특성을 정의하여 AD에 정의된 값을 재정의할 수 있습니다.

자세한 내용은 ID 보기를 사용하여 IdM 클라이언트의 사용자 속성 값을 재정의하는 방법을 참조하십시오.

IdM CA 서버

IdM CA(인증 기관) 서비스가 설치되어 실행 중인 IdM 서버입니다.
대체 이름: **CA 서버**

IdM 배포

IdM 설치 전체를 나타내는 용어입니다. 다음 질문에 대답하여 IdM 배포를 설명할 수 있습니다.

- IdM 배포가 테스트 배포 또는 프로덕션 배포입니까?
  - 몇 개의 IdM 서버가 있습니까?
- IdM 배포에 통합 CA가 포함되어 있습니까?
  - 통합 CA가 자체 서명된 경우 또는 외부 서명입니까?
  - 이 경우 사용 가능한 서버에서 사용 가능한 CA 역할은 무엇입니까? 어떤 서버에서 KRA 역할을 사용할 수 있습니까?
- IdM 배포에 통합된 DNS가 포함되어 있습니까?
  - DNS 역할이 있는 경우 어떤 서버에서 사용 가능한 DNS 역할입니까?
- AD forest 와의 신뢰 계약에 IdM 배포입니까?
  - 이 경우 AD 트러스트 컨트롤러 또는 AD 트러스트 에이전트 역할을 사용할 수 있는 서버가 있습니까?

IdM 서버 및 복제본

IdM 배포에 첫 번째 서버를 설치하려면 **ipa-server-install** 명령을 사용해야 합니다.
그런 다음 관리자는 **ipa-replica-install** 명령을 사용하여 설치된 첫 번째 서버 외에도 **복제본** 을 설치할 수 있습니다. 기본적으로 복제본을 설치하면 생성된 IdM 서버와의 복제 계약이 생성되어 IdM의 나머지 부분에 업데이트를 수신 및 전송할 수 있습니다.

설치된 첫 번째 서버와 복제본 사이에 기능적 차이가 없습니다. 둘 다 완전하게 기능적인 읽기/쓰기 IdM 서버입니다.

더 이상 사용되지 않는 이름: **마스터 서버**

IdM CA 갱신 서버

IdM 토폴로지에 CA(통합 인증 기관)가 포함된 경우 한 서버에 CA 갱신 서버의 고유한 역할이 있습니다. 이 서버는 IdM 시스템 인증서를 유지 관리하고 갱신합니다.
기본적으로 설치하는 첫 번째 CA 서버는 이 역할을 수행하지만 CA 서버는 CA 갱신 서버로 구성할 수 있습니다. 통합 CA가 없는 배포에는 CA 갱신 서버가 없습니다.

더 이상 사용되지 않는 이름: **마스터 CA**

IdM CRL 게시자 서버

IdM 토폴로지에 CA(통합 인증 기관)가 포함된 경우 한 서버에는 CRL(인증서 해지 목록) 게시자 서버의 고유한 역할이 있습니다. 이 서버는 CRL을 유지 관리해야 합니다.
기본적으로 **CA 갱신 서버 역할을 충족하는 서버는** 이 역할을 충족하지만 모든 CA 서버를 CRL 게시자 서버로 구성할 수 있습니다. 통합된 CA가 없는 배포에는 CRL 게시자 서버가 없습니다.

IdM 토폴로지

IdM 솔루션의 구조를 참조하는 용어, 특히 개별 데이터 센터와 클러스터 간의 복제 계약.

Kerberos 인증 표시

인증 지표는 Kerberos 티켓에 연결되어 있으며 티켓을 얻는 데 사용되는 초기 인증 방법을 나타냅니다.

- 이중 인증(암호 + 일회성 암호)의 경우 OTP
- **RD IUS(Remote Authentication Dial-In User Service) 인증 (일반적으로 802.1x 인증의 경우)**
- **PK INIT for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT), 스마트 카드 또는 인증서 인증**
- **brute-force 시도에 대해 강화된 암호의 경우 강화**

자세한 내용은 Kerberos 인증 지표 를 참조하십시오.

Kerberos 키탭

암호는 사용자의 기본 인증 방법이지만 keytabs는 호스트 및 서비스의 기본 인증 방법입니다. Kerberos keytab은 Kerberos 사용자 목록 및 관련 암호화 키를 포함하는 파일이므로, 서비스에서 자체 Kerberos 키를 검색하고 사용자 ID를 확인할 수 있습니다.

예를 들어 모든 IdM 클라이언트에는 Kerberos 영역에 클라이언트 시스템을 나타내는 호스트 주체에 대한 정보를 저장하는 /etc/krb5.keytab 파일이 있습니다.

Kerberos 사용자

고유한 Kerberos 사용자가 Kerberos 영역에서 각 사용자, 서비스 및 호스트를 식별합니다.

엔터티	이름 지정 규칙	예제
사용자	**identifier@REALM**	**admin@EXAMPLE.COM**
서비스	**service/fully-qualified-hostname@REALM**	**http/server.example.com@EXAMPLE.COM**
호스트	**host/fully-qualified-hostname@REALM**	**host/client.example.com@EXAMPLE.COM**

Kerberos 프로토콜

Kerberos는 비밀 키 암호화를 사용하여 클라이언트 및 서버 애플리케이션에 대한 강력한 인증을 제공하는 네트워크 인증 프로토콜입니다. **IdM** 및 **Active Directory**는 사용자, 호스트 및 서비스를 인증하는 데 **Kerberos**를 사용합니다.

Kerberos 영역

Kerberos 영역은 **KerberosDC(Kerberos Key Distribution Center)**에서 관리하는 모든 주체를 포함합니다. **IdM** 배포에서 **Kerberos** 영역에는 모든 **IdM** 사용자, 호스트 및 서비스가 포함됩니다.

Kerberos 티켓 정책

KDC(Kerberos Key Distribution Center)는 연결 정책을 통해 티켓 액세스 제어를 적용하고 티켓 라이프사이클 정책을 통해 **Kerberos** 티켓 기간을 관리합니다. 예를 들어 기본 글로벌 티켓 수명은 **1**일이며 기본 글로벌 최대 갱신 기간은 **1**주일입니다. 자세한 내용은 **IdM Kerberos 티켓 정책 유형을** 참조하십시오.

키 배포 센터 (KDC)

KerberosDC(Kerberos Key Distribution Center)는 **Kerberos** 자격 증명 정보를 관리하는 신뢰할 수 있는 중앙 기관 역할을 하는 서비스입니다. **EgressIP**는 **Kerberos** 티켓을 발행하고 **IdM** 네트워크 내의 엔티티에서 발생하는 데이터의 신뢰성을 보장합니다.

자세한 내용은 **IdM EgressIP의 역할을 참조하십시오.**

LDAP

LDAP(Lightweight Directory Access Protocol)는 네트워크를 통해 분산 디렉터리 정보 서비스에 액세스하고 유지 관리하기 위한 오픈 벤더 중립적인 애플리케이션 프로토콜입니다. 이 사양의 일부는 디렉터리 서비스 항목의 고유 이름(**DN**)으로 구성된 계층형 구조의 데이터를 나타내는 **DIT**(디렉터리 정보 트리)입니다. **LDAP**는 네트워크의 디렉터리 서비스에 대해 **ISO X.500** 표준에서 설명하는 **"lightweight"** 버전의 **DAP(Directory Access Protocol)**입니다.

경량 하위 CA

IdM에서 경량 하위 CA는 인증서를 IdM 루트 CA 또는 하위 CA 중 하나로 서명하는 인증 기관(CA)입니다. 경량 하위 CA는 VPN 또는 HTTP 연결을 보호하는 등 특정 목적으로만 인증서를 발행합니다.

자세한 내용은 인증서의 하위 집합만 신뢰하도록 애플리케이션 제한을 참조하십시오.

암호 정책

암호 정책은 특정 IdM 사용자 그룹의 암호가 충족해야 하는 조건 집합입니다. 조건에는 다음 매개변수가 포함될 수 있습니다.

- 암호의 길이
- 사용되는 문자 클래스 수
- 암호의 최대 수명입니다.

자세한 내용은 암호 정책 을 참조하십시오.

POSIX 속성

POSIX 속성은 운영 체제 간 호환성을 유지하기 위한 사용자 속성입니다.

Red Hat Identity Management 환경에서 사용자를 위한 POSIX 속성은 다음과 같습니다.

- cn, 사용자 이름
- UID, 계정 이름(로그인)
- uidNumber, 사용자 번호 (UID)
- Gid Number , 기본 그룹 번호 (GID)

- **HomeDirectory** 사용자의 홈 디렉터리

Red Hat Identity Management 환경에서 그룹에 대한 **POSIX** 속성은 다음과 같습니다.

- **C n**, 그룹 이름
- **gidNumber**, 그룹 번호 (GID)

이러한 속성은 사용자와 그룹을 별도의 엔터티로 식별합니다.

복제 계약

복제 계약은 동일한 **IdM** 배포에 있는 두 개의 **IdM** 서버 간의 계약입니다. 복제 계약을 통해 데이터와 구성이 두 서버 간에 지속적으로 복제됩니다.

IdM은 두 가지 유형의 복제 계약, 즉 **ID** 정보를 복제하는 *도메인* 복제 계약과 인증서 정보를 복제하는 *인증서 복제* 계약의 두 가지 유형을 사용합니다.

자세한 내용은 다음을 참조하십시오.

- 복제 계약
- 적절한 복제본 수 확인
- 토폴로지의 복제본 연결
- 복제본 토폴로지 예

스마트 카드

스마트 카드는 리소스에 대한 액세스를 제어하는 데 사용되는 이동식 장치 또는 카드입니다. 내장 통합 회로 (IC) 칩, Yubikey와 같은 소형 **USB** 장치 또는 기타 유사한 장치가있는 성가신 카드일 수 있습니다. 스마트 카드는 사용자가 스마트 카드를 호스트 컴퓨터에 연결할 수 있도록 하여 인증을 제공할

수 있으며 호스트 컴퓨터의 소프트웨어는 사용자를 인증하기 위해 스마트 카드에 저장된 주요 자료와 상호 작용합니다.

SSSD

SSSD(System Security Services Daemon)는 RHEL 호스트에서 사용자 인증 및 사용자 권한을 관리하는 시스템 서비스입니다. SSSD는 선택적으로 오프라인 인증을 위해 원격 공급자에서 검색된 사용자 ID 및 인증 정보를 캐시합니다. 자세한 내용은 SSSD 및 해당 이점 이해를 참조하십시오.

SSSD 백엔드

데이터 공급자라고도 하는 SSSD 백엔드는 SSSD 캐시를 관리하고 생성하는 SSSD 하위 프로세스입니다. 이 프로세스는 LDAP 서버와 통신하고 다른 조회 쿼리를 수행하고 결과를 캐시에 저장합니다. 또한 LDAP 또는 Kerberos에 대해 온라인 인증을 수행하고 로그인 중인 사용자에게 액세스 및 암호 정책을 적용합니다.

티켓 통합 티켓 (TGT)

KerberosDC(Kerberos Key Distribution Center)를 인증한 후 사용자는 웹사이트 및 이메일과 같은 다른 서비스에 대한 액세스 티켓을 요청하는 데 사용할 수 있는 임시 인증 정보 세트인 TGT(ticket-granting ticket)를 받습니다.

TGT를 사용하여 추가 액세스를 요청하면 사용자가 여러 서비스에 액세스하기 위해 사용자가 한 번만 인증하면 되기 때문에 Single Sign-On 환경을 사용할 수 있습니다. TGT는 갱신할 수 있으며, Kerberos 티켓 정책은 티켓 갱신 제한 및 액세스 제어를 결정합니다.

자세한 내용은 Kerberos 티켓 정책 관리를 참조하십시오.

웹 서버:: 웹 서버는 컴퓨터 소프트웨어이며 페이지, 이미지 또는 애플리케이션과 같은 웹 콘텐츠에 대한 요청을 수락하는 기본 하드웨어입니다. 웹 브라우저와 같은 사용자 에이전트는 HTTP, 웹 콘텐츠를 배포하는 데 사용되는 네트워크 프로토콜 또는 보안 변형 HTTPS를 사용하여 특정 리소스를 요청합니다. 웹 서버는 해당 리소스의 콘텐츠 또는 오류 메시지로 응답합니다. 웹 서버는 사용자 에이전트에서 전송된 리소스를 수락하고 저장할 수도 있습니다. Red Hat IdM(Identity Management)은 Apache 웹 서버를 사용하여 IdM 웹 UI를 표시하고, Directory Server 및 CA(인증 기관)와 같은 구성 요소 간 통신을 조정합니다. Apache 웹 서버를 참조하십시오.

추가 용어집

이 용어집에서 ID 관리 용어를 찾을 수 없는 경우 Directory Server 및 Certificate System 용어집을 참조하십시오.

- Directory Server 11 용어집

- 인증서 시스템 9 용어집

1.5. 추가 리소스

- Red Hat IdM에 대한 일반적인 정보는 Red Hat 고객 포털의 Red Hat Identity Management 제품 페이지를 참조하십시오.

2장. IDM에서 페일오버, 로드 밸런싱 및 고가용성

IdM(Identity Management)에는 IdM 클라이언트에 대한 기본 제공 장애 조치(failover) 메커니즘과 IdM 서버의 부하 분산 및 고가용성 기능이 있습니다.

2.1. 클라이언트 측 페일오버 기능

- 기본적으로 IdM 클라이언트의 SSSD 서비스는 연결할 최상의 IdM 서버를 자동으로 결정하기 위해 DNS의 서비스(SRV) 리소스 레코드를 사용하도록 구성되어 있습니다. 이 동작은 /etc/sssd/sssd.conf 파일의 ipa_server 매개변수에 있는 _srv_ 옵션으로 제어됩니다.

```
[root@client ~]# cat /etc/sssd/sssd.conf

[domain/example.com]
id_provider = ipa
ipa_server = _srv_, server.example.com
...
```

IdM 서버가 오프라인 상태가 되면 IdM 클라이언트의 SSSD 서비스가 다른 IdM 서버에 자동으로 연결됩니다.

- 성능상의 이유로 DNS 조회를 바이패스하려면 ipa_server 매개변수에서 _srv_ 항목을 제거하고 클라이언트가 연결해야 하는 IdM 서버를 기본 설정으로 지정합니다.

```
[root@client ~]# cat /etc/sssd/sssd.conf

[domain/example.com]
id_provider = ipa
ipa_server = server1.example.com, server2.example.com
...
```

2.2. 서버 측 부하 분산 및 서비스 가용성

여러 IdM 복제본을 설치하여 IdM에서 부하 분산 및 고가용성을 달성할 수 있습니다.

- 지리적으로 분산된 네트워크가 있는 경우 데이터 센터당 여러 IdM 복제본을 구성하여 IdM 클라이언트와 액세스 가능한 서버 간의 경로를 단축할 수 있습니다.

- Red Hat은 최대 60개의 복제본이 있는 환경을 지원합니다.

- **IdM** 복제 메커니즘은 활성/활성 서비스 가용성을 제공합니다. 모든 **IdM** 복제본의 서비스는 동시에 쉽게 사용할 수 있습니다.

참고

Red Hat은 **IdM**과 기타 부하 분산 또는 **HA(**고가용성**)** 소프트웨어를 결합하지 않는 것이 좋습니다.

많은 타사 고가용성 솔루션은 활성/수동 시나리오를 가정하고 **IdM** 가용성에 불필요한 서비스 중단을 초래합니다. 다른 솔루션은 가상 **IP** 또는 클러스터형 서비스당 단일 호스트 이름을 사용합니다. 이러한 모든 방법은 일반적으로 **IdM** 솔루션에서 제공하는 서비스 가용성 유형에서 제대로 작동하지 않습니다. 또한 **Kerberos**와 매우 비효율적으로 통합되어 배포의 전체 보안 및 안정성이 줄어 듭니다.

3장. 복제본 토폴로지 계획

다음 섹션에서는 사용 사례에 적합한 복제본 토폴로지를 결정하는 방법에 대해 설명합니다.

3.1. 고성능 및 재해 복구를 위한 솔루션으로 여러 복제본 서버

리소스에 액세스하는 사용자에게 중요한 기능 및 IdM(Identity Management) 서비스의 고가용성이 중요합니다. 로드 밸런싱을 통해 IdM 인프라의 지속적인 기능 및 고가용성을 수행하는 기본 솔루션 중 하나는 첫 번째 서버의 복제본 서버를 생성하여 중앙 디렉터리의 복제입니다.

IdM을 통해 지리적으로 분산된 데이터 센터에 추가 서버를 배치하여 엔터프라이즈 조직 구조를 반영할 수 있습니다. 이러한 방식으로 IdM 클라이언트와 액세스 가능한 서버 간의 경로가 단축됩니다. 또한 여러 서버를 사용하면 더 많은 클라이언트를 위해 부하를 분산하고 스케일링할 수 있습니다.

여러 개의 중복된 IdM 서버를 유지하고 서로 복제하도록 하는 것도 일반적인 백업 메커니즘으로 서버 손실을 완화하거나 방지할 수 있습니다. 예를 들어 한 서버가 실패하면 다른 서버는 계속 도메인에 서비스를 제공합니다. 나머지 서버 중 하나를 기반으로 새 복제본을 생성하여 손실된 서버를 복구할 수도 있습니다.

3.2. IDM 서버 및 클라이언트 소개

IdM(Identity Management) 도메인에는 다음과 같은 유형의 시스템이 포함되어 있습니다.

IdM 서버

IdM 서버는 IdM 도메인 내의 ID, 인증 및 권한 부여 요청에 응답하는 Red Hat Enterprise Linux 시스템입니다. 대부분의 배포에서는 통합 CA(인증 기관)도 IdM 서버와 함께 설치됩니다.

IdM 서버는 ID 및 정책 정보를 위한 중앙 리포지토리입니다. IdM 서버는 도메인 멤버가 사용하는 모든 선택적 서비스를 호스팅할 수도 있습니다.

- 인증 기관 (CA)
- 키 복구 기관(KRA)
- DNS

- Active Directory (AD) 신뢰 컨트롤러
- Active Directory (AD) 신뢰 에이전트

IdM 클라이언트

IdM 클라이언트는 서버에 등록된 Red Hat Enterprise Linux 시스템으로, 이러한 서버에서 IdM 서비스를 사용하도록 구성되어 있습니다.

클라이언트는 IdM 서버와 상호 작용하여 해당 서버에서 제공하는 서비스에 액세스합니다. 예를 들어 클라이언트는 Kerberos 프로토콜을 사용하여 인증을 수행하고 엔터프라이즈급 SSO(Single Sign-On)에 대한 티켓을 취득하고, LDAP를 사용하여 ID와 정책의 출처와 서버 및 서비스가 있는 위치 및 연결 방법을 감지합니다.

IdM 서버도 IdM 클라이언트가 포함되어 있습니다. 고객이 직접 등록한 클라이언트로서 서버는 다른 클라이언트와 동일한 기능을 제공합니다.

다수의 클라이언트에 서비스를 제공하고 중복성 및 가용성에 대해 IdM을 통해 단일 도메인에 여러 IdM 서버에 배포할 수 있습니다. 최대 60대의 서버를 배포할 수 있습니다. 이는 현재 IdM 도메인에서 지원되는 복제본이라고도 하는 최대 IdM 서버 수입니다. IdM 서버는 클라이언트에 다양한 서비스를 제공합니다. 모든 서버가 가능한 모든 서비스를 제공해야 하는 것은 아닙니다. Kerberos 및 LDAP와 같은 일부 서버 구성 요소는 모든 서버에서 항상 사용할 수 있습니다. CA, DNS, Trust Controller 또는 Vault와 같은 기타 서비스는 선택 사항입니다. 이는 일반적으로 배포에서 다양한 서버가 다양한 역할을 수행한다는 것을 의미합니다.

IdM 토폴로지에 통합된 CA가 포함된 경우 하나의 서버에 CRL(인증서 해지 목록) 게시자 서버의 역할이 있으며, 한 서버에 CA 갱신 서버의 역할이 있습니다. 기본적으로 설치된 첫 번째 CA 서버는 이러한 두 가지 역할을 충족하지만 이러한 역할을 별도의 서버에 할당할 수 있습니다.

주의

*CA 갱신 서버*는 CA 하위 시스템 인증서 및 키 추적을 담당하는 도메인의 유일한 시스템이므로 IdM 배포에 중요합니다. IdM 배포에 영향을 미치는 재해에서 복구하는 방법에 대한 자세한 내용은 ID 관리를 사용하여 재해 복구 수행을 참조하십시오.

중복 및 로드 밸런싱을 위해 관리자는 기존 서버의 *복제본*을 생성하여 추가 서버를 생성합니다. 복제본을 생성할 때 IdM에서 기존 서버의 구성을 복제합니다. 복제본은 사용자, 시스템, 인증서, 구성된 정책에 대한 내부 정보를 포함하여 초기 서버와 해당 코어 구성을 공유합니다.

참고

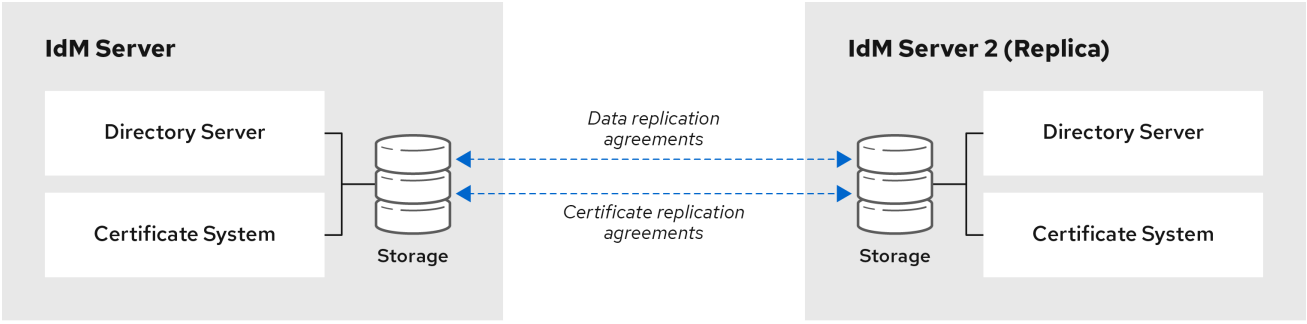
생성된 복제본 및 서버는 ***CA 갱신*** 및 ***CRL 게시자*** 역할을 제외하고 기능적으로 동일합니다. 따라서 컨텍스트에 따라 *서버* 및 *복제본*이라는 용어는 서로 바꿔 사용할 수 있습니다.

3.3. 복제 계약

관리자가 기존 서버를 기반으로 복제본을 생성하는 경우 IdM(Identity Management)은 초기 서버와 복제본 간에 *복제 계약을* 생성합니다. 복제 계약을 통해 데이터와 구성이 두 서버 간에 지속적으로 복제됩니다.

IdM은 *여러 읽기/쓰기 복제 복제*를 사용합니다. 이 구성에서는 복제 계약에 가입된 모든 복제본이 업데이트를 수신 및 제공하므로 공급업체 및 소비자로 간주됩니다. 복제 계약은 항상 바이라터입니다.

그림 3.1. 서버 및 복제본 계약



IdM은 두 가지 유형의 복제 계약을 사용합니다.

도메인 복제 계약

이러한 계약은 ID 정보를 복제합니다.

인증서 복제 계약

이러한 계약은 인증서 정보를 복제합니다.

두 개의 복제 채널은 독립적입니다. 두 서버에는 하나 이상의 복제 계약이 구성될 수 있습니다. 예를 들어 서버 **A** 및 서버 **B**에 도메인 복제 계약만 구성된 경우 인증서 정보가 아닌 **ID** 정보만 복제됩니다.

3.4. 적절한 복제본 수 확인

각 데이터 센터에 최소 두 개의 복제본을 설정(하드라이드 요구 사항이 아님)

데이터 센터는 예를 들어 주요 사무실 또는 지리적 위치일 수 있습니다.

클라이언트를 제공할 수 있도록 충분한 수의 서버 설정

하나의 **IdM(Identity Management)** 서버는 **2000 - 3000** 클라이언트에 서비스를 제공할 수 있습니다. 클라이언트가 서버를 하루에 여러 번 쿼리하는 것으로 가정하지만 **1**분마다는 그렇지 않습니다. 더 자주 쿼리를 예상하는 경우 더 많은 서버를 계획합니다.

충분한 수의 **CA**(인증 기관) 복제본을 설정

CA 역할이 설치된 복제본만 인증서 데이터를 복제할 수 있습니다. **IdM CA**를 사용하는 경우 해당 환경에 인증서 복제 계약과 함께 두 개 이상의 **CA** 복제본이 있는지 확인합니다.

단일 **IdM** 도메인에서 최대 **60**개의 복제본 설정

Red Hat은 최대 **60**개의 복제본이 있는 환경을 지원합니다.

3.5. 토폴로지의 복제본 연결

각 복제본을 두 개 이상의 복제본에 연결

추가 복제 계약을 구성하면 초기 복제본과 설치한 첫 번째 서버뿐만 아니라 다른 복제본 간에도 정보가 복제됩니다.

복제본을 최대 **4**개의 다른 복제본에 연결(하드 요구 사항 아님)

서버당 다수의 복제 계약은 상당한 이점을 추가하지 않습니다. 수신 복제본은 한 번에 하나의 다른 복제본만 업데이트할 수 있으며 다른 복제 계약은 유휴 상태입니다. 복제본당 **4**개 이상의 복제 계약은 일반적으로 리소스 낭비를 의미합니다.

참고

이 권장 사항은 인증서 복제 및 도메인 복제 계약에 모두 적용됩니다.

복제본당 **4**개의 복제 계약 제한에는 다음 두 가지 예외가 있습니다.

- 특정 복제본이 온라인 상태가 아니거나 응답하지 않는 경우 장애 조치(failover) 경로가 필요합니다.

- 대규모 배포에서 특정 노드 간에 추가 직접 링크가 필요합니다.

복제 계약의 구성이 전체 성능에 부정적인 영향을 미칠 수 있습니다. 토폴로지의 여러 복제 계약에서 업데이트를 보내는 경우 특정 복제본에서 들어오는 업데이트와 발신 업데이트 간에 changelog 데이터베이스 파일에 높은 경합이 발생할 수 있습니다.

복제본마다 더 많은 복제 계약을 사용하기로 결정하는 경우 복제 문제 및 대기 시간이 발생하지 않도록 합니다. 그러나 많은 거리와 중간 노드가 많은 경우 대기 시간 문제가 발생할 수 있습니다.

데이터 센터의 복제본을 서로 연결

이렇게 하면 데이터 센터 내의 도메인 복제가 보장됩니다.

각 데이터 센터를 두 개 이상의 다른 데이터 센터에 연결

이렇게 하면 데이터 센터 간 도메인 복제가 보장됩니다.

최소 한 쌍의 복제 계약을 사용하여 데이터 센터 연결

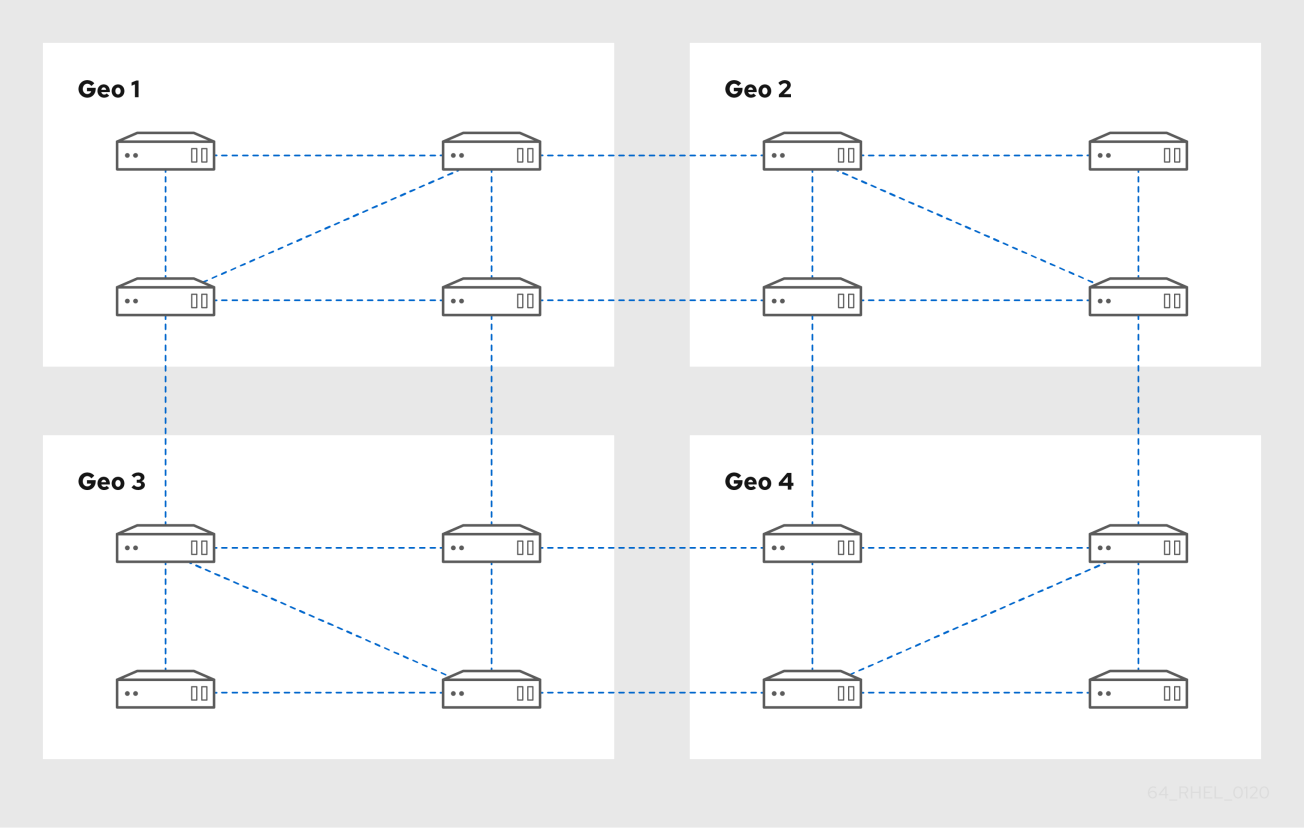
데이터 센터 A 및 B에 A1에서 B1로의 복제 계약이 있는 경우 A2에서 B2로의 복제 계약이 있으면 서버 중 하나가 다운되면 두 데이터 센터 간에 복제가 계속될 수 있습니다.

3.6. 복제본 토폴로지 예

아래 그림은 안정적인 토폴로지 생성 지침을 기반으로 IdM(Identity Management) 토폴로지의 예를 보여줍니다.

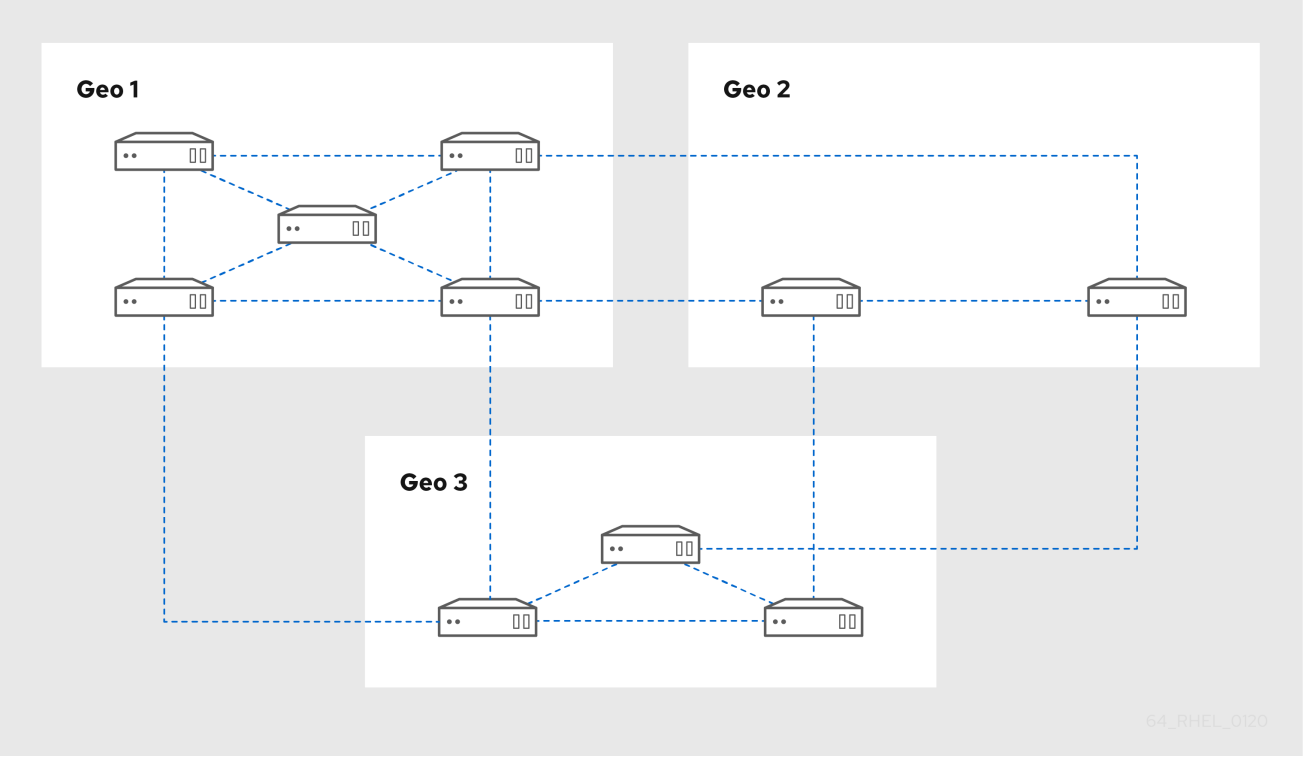
복제본 토폴로지 예 1 은 각각 4개의 서버가 있는 데이터 센터 4개를 보여줍니다. 서버는 복제 계약과 연결되어 있습니다.

그림 3.2. 복제본 토폴로지 예 1



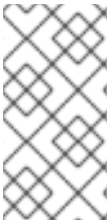
복제본 토폴로지 예 2 는 각각 다른 서버 수가 있는 세 개의 데이터 센터를 보여줍니다. 서버는 복제 계약과 연결되어 있습니다.

그림 3.3. 복제본 토폴로지 예 2



3.7. 숨겨진 복제본 모드

기본적으로 새 복제본을 설정하면 설치 프로그램이 **DNS**에서 서비스(SRV) 리소스 레코드를 자동으로 생성합니다. 이러한 레코드를 사용하면 클라이언트가 복제본 및 해당 서비스를 자동으로 검색할 수 있습니다. 숨겨진 복제본은 모든 서비스가 실행 중이고 사용 가능한 **IdM** 서버입니다. 그러나 **DNS**에는 **SRV** 레코드가 없으며 **LDAP** 서버 역할은 활성화되지 않습니다. 따라서 클라이언트는 서비스 검색을 사용하여 이러한 숨겨진 복제본을 감지할 수 없습니다.



참고

RHEL 8.1에 도입된 숨겨진 복제본 기능은 **RHEL 8.2**부터 기술 프리뷰로 완전히 지원됩니다.

숨겨진 복제본은 주로 클라이언트를 방해할 수 있는 전용 서비스를 위해 설계되었습니다. 예를 들어 전체 **IdM** 백업을 위해서는 서버의 모든 **IdM** 서비스를 종료해야 합니다. 숨겨진 복제본을 사용하지 않는 클라이언트가 없으므로 관리자는 클라이언트에 영향을 주지 않고 이 호스트의 서비스를 일시적으로 종료할 수 있습니다.

참고

- 새 호스트의 숨겨진 복제본에서 백업을 복원하면 항상 **non-hidden (regular)** 복제본이 생성됩니다.
- 클러스터에서 사용되는 모든 서버 역할, 특히 통합 **CA**를 사용하는 경우 해당 서비스를 복원하려면 백업에 대해 숨겨진 복제본에 설치해야 합니다.
- **IdM** 백업 생성 및 작업에 대한 자세한 내용은 **IdM** 백업 및 복원을 참조하십시오.

기타 사용 사례에는 **IdM API**에 대한 고로드 작업 또는 대규모 가져오기 또는 광범위한 쿼리와 같은 **LDAP** 서버가 포함됩니다. 숨겨진 복제본을 설치하려면 **--hidden-replica** 매개 변수를 **ipa-replica-install** 명령에 전달합니다.

복제본 설치에 대한 자세한 내용은 **ID** 관리 복제본 설치를 참조하십시오.

또는 기존 복제본의 상태를 변경할 수 있습니다. 자세한 내용은 **Hidden Replicas**의 데모 및 승격 을 참조하십시오.

4장. DNS 서비스 및 호스트 이름 계획

IdM(Identity Management)은 IdM 서버에 다양한 유형의 DNS 구성을 제공합니다. 다음 섹션에서는 이러한 내용을 설명하고 사용 사례에 가장 적합한지 결정하는 방법에 대한 조언을 제공합니다.

4.1. IDM 서버에서 사용 가능한 DNS 서비스

통합된 DNS를 사용하거나 사용하지 않고 IdM(Identity Management) 서버를 설치할 수 있습니다.

표 4.1. 통합 DNS 없이 IdM과 통합된 DNS 비교

	통합된 DNS가 있는 경우	통합된 DNS가 없는 경우
개요:	IdM은 IdM 도메인에 대한 자체 DNS 서비스를 실행합니다.	IdM은 외부 DNS 서버에서 제공하는 DNS 서비스를 사용합니다.
제한 사항:	IdM에서 제공하는 통합 DNS 서버는 IdM 배포 및 유지 관리와 관련된 기능만 지원합니다. 고급 DNS 기능 중 일부를 지원하지 않습니다. 범용 DNS 서버로 사용하도록 설계되지 않았습니다.	DNS는 기본 IdM 툴과 통합되지 않습니다. 예를 들어 IdM은 토폴로지가 변경된 후 DNS 레코드를 자동으로 업데이트하지 않습니다.
다음을 위해 가장 잘 작동합니다.	IdM 배포 내의 기본 사용량. IdM 서버가 DNS를 관리하면 DNS가 기본 IdM 툴과 긴밀하게 통합되어 일부 DNS 레코드 관리 작업을 자동화할 수 있습니다.	IdM DNS 범위를 벗어난 고급 DNS 기능이 필요한 환경입니다. 외부 DNS 서버를 계속 사용하고자 하는 DNS 인프라가 올바르게 구성된 환경입니다.

ID 관리 서버를 기본 DNS 서버로 사용하더라도 다른 외부 DNS 서버를 보조 서버로 계속 사용할 수 있습니다. 예를 들어 사용자 환경에서 AD(Active Directory)와 통합된 DNS 서버와 같은 다른 DNS 서버를 이미 사용하고 있는 경우 IdM 기본 도메인만 IdM과 통합된 DNS에 위임할 수 있습니다. DNS 영역을 IdM DNS로 마이그레이션할 필요는 없습니다.

참고

SAN(Subject Alternative Name) 확장에 IP 주소로 IdM 클라이언트의 인증서를 발행해야 하는 경우 IdM 통합 DNS 서비스를 사용해야 합니다.

4.2. DNS 도메인 이름 및 KERBEROS 영역 이름 계획에 대한 지침

첫 번째 IdM(Identity Management) 서버를 설치할 때 설치에 IdM 도메인 및 Kerberos 영역 이름의 기본 DNS 이름을 묻는 메시지가 표시됩니다. 이 섹션의 지침은 이름을 올바르게 설정하는 데 도움이 될 수

있습니다.

주의

서버가 이미 설치된 후에는 IdM 기본 도메인 이름과 Kerberos 영역 이름을 변경할 수 없습니다. 예를 들어 이름을 *lab.example.com* 에서 *production.example.com* 으로 변경하여 테스트 환경에서 프로덕션 환경으로 이동할 수 없습니다.

서비스 레코드에 대한 별도의 DNS 도메인

IdM에 사용된 *기본 DNS 도메인이* 다른 시스템과 공유되지 않았는지 확인합니다. 이를 통해 DNS 수준의 충돌을 방지할 수 있습니다.

적절한 DNS 도메인 이름 위임

DNS 도메인의 퍼블릭 DNS 트리에 유효한 위임이 있는지 확인합니다. 사설 네트워크에도 속하지 않고 위임되지 않은 도메인 이름을 사용하지 마십시오.

다중 레이블 DNS 도메인

단일 레이블 도메인 이름(예: .company)을 사용하지 마십시오. IdM 도메인은 하나 이상의 하위 도메인과 최상위 도메인(예: example.com 또는 company. example.com)으로 구성되어야 합니다.

고유한 Kerberos 영역 이름

영역 이름이 AD(Active Directory)에서 사용하는 이름과 같은 다른 기존 Kerberos 영역 이름과 충돌하지 않는지 확인합니다.

Kerberos 영역 이름: 기본 DNS 이름의 대문자 버전

영역 이름을 기본 DNS 도메인 이름(*example.com*)의 대문자(*EXAMPLE.COM*) 버전으로 설정하는 것이 좋습니다.

주의

Kerberos 영역 이름을 기본 DNS 이름의 대문자 버전으로 설정하지 않으면 AD trust을 사용할 수 없습니다.

DNS 도메인 이름 및 Kerberos 영역 이름 계획에 대한 추가 참고 사항

- **IdM 배포는 항상 하나의 Kerberos 영역을 나타냅니다.**
- **여러 별도의 DNS 도메인(*example.com*, *example.net*, *example.org*)의 IdM 클라이언트를 단일 Kerberos 영역(*EXAMPLE.COM*)에 결합할 수 있습니다.**
- **IdM 클라이언트는 기본 DNS 도메인에 있을 필요가 없습니다. 예를 들어 IdM 도메인이 *idm.example.com* 인 경우 클라이언트는 *clients.example.com* 도메인에 있을 수 있지만 DNS 도메인과 Kerberos 영역 간에 명확한 매핑을 구성해야 합니다.**

참고

매핑을 생성하는 표준 방법은 _kerberos TXT DNS 레코드를 사용하는 것입니다. IdM 통합 DNS는 이러한 레코드를 자동으로 추가합니다.

계획 DNS 전달

- **전체 IdM 배포에 하나의 전달자만 사용하려면 글로벌 전달자 를 구성하십시오.**
- **회사가 지리적으로 멀리 떨어진 지역에 여러 사이트에 분산되어 있는 경우 글로벌 전달자가 비현실적 일 수 있습니다. 서버별 전달자를 구성합니다.**
- **회사에 퍼블릭 인터넷에서 확인할 수 없는 내부 DNS 네트워크가 있는 경우 IdM 도메인 의 호스트가 다른 내부 DNS 네트워크에서 호스트를 확인할 수 있도록 정방향 영역 및 영역 전달자 를 구성합니다.**

5장. CA 서비스 계획

Red Hat Enterprise Linux의 IdM(Identity Management)은 다양한 유형의 CA(인증 기관) 구성을 제공합니다. 다음 섹션에서는 다양한 시나리오를 설명하고 사용 사례에 가장 적합한 구성을 결정하는 데 도움이 되는 조언을 제공합니다.

CA 주체 DN

CA(인증 기관) 주체 고유 이름(DN)은 CA의 이름입니다. IdM(Identity Management) CA 인프라에서 전역적으로 고유해야 하며 설치 후에는 변경할 수 없습니다. 외부적으로 서명해야 하는 IdM CA가 있는 경우 IdM CA 주체 DN 양식에 대한 외부 CA 관리자를 참조해야 할 수 있습니다.

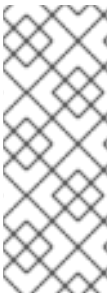
5.1. IDM 서버에서 사용 가능한 CA 서비스

통합된 IdM 인증 기관(CA) 또는 CA 없이 IdM(Identity Management) 서버를 설치할 수 있습니다.

표 5.1. IdM과 CA 없이 통합 CA 비교

	통합 CA	CA 없음
개요:	IdM은 *CA 서명 인증서*와 함께 자체 PKI(공개 키 인프라) 서비스를 사용하여 IdM 도메인에 인증서를 생성하고 서명합니다. ● 루트 CA가 통합 CA인 경우 IdM은 자체 서명된 CA 인증서를 사용합니다. ● 루트 CA가 외부 CA인 경우 통합 IdM CA는 외부 CA에 종속되어 있습니다. IdM에서 사용하는 CA 인증서는 외부 CA에서 서명하지만 IdM 도메인의 모든 인증서는 통합 인증서 시스템 인스턴스에서 발행됩니다. ● 통합된 CA는 사용자, 호스트 또는 서비스의 인증서를 발행할 수도 있습니다. 외부 CA는 회사 CA 또는 타사 CA일 수 있습니다.	IdM은 자체 CA를 설정하지 않고 외부 CA에서 서명된 호스트 인증서를 사용합니다. CA 없이 서버를 설치하려면 타사 기관에서 다음 인증서를 요청해야 합니다. ● LDAP 서버 인증서 ● Apache 서버 인증서 ● PKINIT 인증서 ● LDAP 및 Apache 서버 인증서를 발급한 CA의 전체 CA 인증서 체인

	통합 CA	CA 없음
제한 사항:	통합 CA가 외부 CA에 종속된 경우 IdM 도메인 내에서 발행된 인증서에는 다음과 같은 다양한 인증서 속성에 대해 외부 CA가 설정한 제한이 있을 수 있습니다. ● 유효 기간입니다. ● IDM CA 또는 하위 항목에서 발급한 인증서에 제목 이름이 표시될 수 있는 제약 조건입니다. ● IDM CA가 자체적으로 수행할 수 있는지, CA 인증서를 발급하거나 하위 인증서 체인을 "deep"하는 방법에 대한 제약 조건입니다.	IdM 외부에서 인증서를 관리하면 다음과 같은 많은 추가 활동이 발생합니다. ● 인증서 생성, 업로드 및 갱신은 수동 프로세스입니다. ● **certmonger** 서비스는 IPA 인증서(LDAP 서버, Apache 서버 및 PKINIT 인증서)를 추적하지 않으며 인증서가 만료될 예정이면 알리지 않습니다. 관리자는 외부에서 발급한 인증서에 대한 알림을 수동으로 설정하거나 **certmonger** 를 추적하려면 해당 인증서에 대한 추적 요청을 설정해야 합니다.
다음을 위해 가장 잘 작동합니다.	자체 인증서 인프라를 생성하고 사용할 수 있는 환경입니다.	인프라 내의 제한으로 서버와 통합된 인증서 서비스를 설치할 수 없는 경우 매우 드문 경우입니다.



참고

자체 서명된 CA에서 외부 서명된 CA로 전환하거나 다른 방식으로 전환할 뿐만 아니라 설치 후에도 IdM CA 인증서를 발행하는 외부 CA가 발생할 수 있습니다. CA가 없는 설치 후에도 통합 CA를 구성할 수도 있습니다. 자세한 내용은 IdM 서버 설치를 참조하십시오. CA 없이 통합 DNS를 사용하여 다음을 참조하십시오.

5.2. CA 서비스 배포를 위한 지침

다음 단계에서는 CA(인증 기관) 서비스 배포에 대한 지침을 제공합니다.

절차

1. **토폴로지의 두 개 이상의 서버에 CA 서비스를 설치합니다.**

   CA가 없으면 구성된 복제본은 모든 인증서 작업 요청을 토폴로지의 CA 서버로 요청합니다.

주의

CA가 있는 모든 서버가 손실되면 복구 가능성 없이 모든 CA 구성이 손실됩니다. 이 경우 새 CA를 설정하고 새 인증서를 설치해야 합니다.

2. **배포에서 CA 요청을 처리할 수 있는 충분한 CA 서버를 유지합니다.**

적절한 CA 서버 수에 대한 추가 권장 사항은 다음 표를 참조하십시오.

표 5.2. 적절한 수의 CA 서버 설정 지침

배포에 대한 설명	제안된 CA 서버 수
많은 수의 인증서가 발급된 배포	3개 또는 4개의 CA 서버
여러 지역 간의 대역폭 또는 가용성 문제가 있는 배포	리전당 하나의 CA 서버 1개, 배포용 최소 3개의 서버 총 3개
기타 모든 배포	두 개의 CA 서버

6장. AD와의 통합 계획

다음 섹션에서는 Red Hat Enterprise Linux와 AD(Active Directory)를 통합하는 옵션을 소개합니다.

6.1. 직접 통합

직접 통합에서 Linux 시스템은 AD(Active Directory)에 직접 연결됩니다. 다음과 같은 유형의 통합이 가능합니다.

SSSD(System Security Services Daemon) 통합

SSSD는 AD, Identity Management(IdM) 또는 일반 LDAP 또는 Kerberos 서버 등 다양한 ID 및 인증 저장소와 Linux 시스템을 연결할 수 있습니다.

SSSD와의 통합을 위한 주요 요구 사항:

- AD와 통합할 때 SSSD는 기본적으로 단일 AD forest 내에서만 작동합니다. 다중 포트 설정의 경우 수동 도메인 열거를 구성합니다.
- Remote AD forests는 local forests를 신뢰하여 idmap_ad 플러그인이 원격 forest 사용자를 올바르게 처리할 수 있도록 해야 합니다.

SSSD는 직접 및 간접 통합을 모두 지원합니다. 또한 마이그레이션 비용 없이 하나의 통합 접근법에서 다른 방식으로 전환할 수 있습니다.

Samba Winbind와 통합

Samba 제품군의 Winbind 구성 요소는 Linux 시스템에서 Windows 클라이언트를 에뮬레이션하고 AD 서버와 통신합니다.

Samba Winbind와의 통합을 위한 주요 요구 사항:

- 다중 Forest AD 설정에서 Winbind와 직접 통합하려면 양방향 신뢰가 필요합니다.
- Linux 시스템의 로컬 도메인의 양방향 경로는 원격 AD 도메인에 있는 사용자 도메인에 있어야 idmap_ad 플러그인에서 사용자에 대한 전체 정보를 사용할 수 있도록 해야 합니다.

권장 사항

- SSSD는 AD 통합의 대부분의 사용 사례를 충족하고 클라이언트 시스템과 다른 유형의 ID 및 인증 공급자(AD, IdM, Kerberos, LDAP) 간의 일반 게이트웨이로 강력한 솔루션을 제공합니다.

- winbind는 Samba FS를 배포하려는 AD 도메인 멤버 서버에 배포하는 데 권장됩니다.

6.2. 간접 통합

간접 통합에서 Linux 시스템은 먼저 중앙 서버에 연결되며 AD(Active Directory)에 연결됩니다. 간접 통합을 통해 관리자는 Linux 시스템과 정책을 중앙에서 관리할 수 있지만 AD의 사용자는 Linux 시스템 및 서비스에 투명하게 액세스할 수 있습니다.

AD와의 상호 간 신뢰를 기반으로 통합

IdM(Identity Management) 서버는 Linux 시스템을 제어하는 중앙 서버 역할을 합니다. AD에 대한 교차 영역 Kerberos 신뢰가 설정되어 AD의 사용자가 로그온하여 Linux 시스템 및 리소스에 액세스할 수 있습니다. IdM은 AD에 별도의 포리스트로 표시하고 AD에서 지원하는 Specest 수준 신뢰를 활용합니다.

신뢰를 사용할 때:

- AD 사용자는 IdM 리소스에 액세스할 수 있습니다.

- IdM 서버 및 클라이언트는 AD 사용자 및 그룹의 ID를 확인할 수 있습니다.

- AD 사용자 및 그룹은 호스트 기반 액세스 제어와 같이 IdM에 정의된 조건 하에 IdM에 액세스합니다.

- AD 사용자 및 그룹은 AD 측에서 계속 관리됩니다.

동기화 기반 통합

이 방법은 WinSync 도구를 기반으로합니다. WinSync 복제 계약은 AD의 사용자 계정을 IdM과 동기화합니다.

주의

WinSync는 더 이상 **Red Hat Enterprise Linux 8**에서 적극적으로 개발되지 않습니다. 간접 통합을 위한 기본 솔루션은 상호 신뢰입니다.

동기화 기반 통합의 제한 사항은 다음과 같습니다.

- 그룹은 **IdM**에서 **AD**와 동기화되지 않습니다.
- 사용자는 **AD** 및 **IdM**에서 중복됩니다.
- **WinSync**는 단일 **AD** 도메인만 지원합니다.
- **AD**에서 하나의 도메인 컨트롤러만 사용하여 데이터를 하나의 **IdM** 인스턴스에 동기화할 수 있습니다.
- **AD** 도메인의 모든 도메인 컨트롤러에 **PassSync** 구성 요소를 설치해야 하는 사용자 암호를 동기화해야 합니다.
- 동기화를 구성한 후 모든 **AD** 사용자는 암호를 수동으로 변경해야 **PassSync**를 동기화할 수 있습니다.

6.3. 간접 및 직접 통합 결정

이 섹션의 지침은 사용 사례에 맞는 통합 유형을 결정하는 데 도움이 될 수 있습니다.

Active Directory에 연결할 시스템 수

30~50개 미만의 시스템 연결(하드 제한 아님)

30-50 미만의 시스템을 연결하는 경우 직접 통합을 고려하십시오. 간접 통합으로 인해 불필요한 오버헤드가 발생할 수 있습니다.

30~50개 이상의 시스템 연결(하드 제한 아님)

30~50개 이상의 시스템을 연결하는 경우 ID 관리와 간접 통합을 고려하십시오. 이 접근 방식을 사용하면 Linux 시스템의 중앙 집중식 관리를 활용할 수 있습니다.

소수의 Linux 시스템을 관리하지만, 이 수가 빠르게 증가할 것으로 예상합니다.

이 시나리오에서는 나중에 환경을 마이그레이션할 필요가 없도록 간접 통합을 고려하십시오.

새 시스템 배포 빈도 및 유형

베어 메탈 시스템을 비정상적으로 배포

새 시스템을 거의 배포하지 않고 일반적으로 베어 메탈 시스템인 경우 직접 통합을 고려하십시오. 이러한 경우 직접 통합은 일반적으로 가장 간단하고 쉽습니다.

가상 시스템 자주 배포

새 시스템을 자주 배포하고 일반적으로 필요에 따라 프로비저닝된 가상 시스템인 경우 간접 통합을 고려하십시오. 간접 통합을 통해 중앙 서버를 사용하여 새 시스템을 동적으로 관리하고 Red Hat Satellite와 같은 오케스트레이션 툴과 통합할 수 있습니다.

Active Directory가 필요한 인증 공급자입니다.

내부 정책의 경우 모든 사용자가 Active Directory에 대해 인증해야 함을 확인합니까?

직접 또는 간접 통합을 선택할 수 있습니다. Identity Management와 Active Directory 간의 신뢰와 간접 통합을 사용하는 경우 Linux 시스템에 액세스하는 사용자는 Active Directory에 대해 인증됩니다. Active Directory에 존재하는 정책은 인증 중에 실행되고 적용됩니다.

7장. IDM과 AD 간의 상호 간 신뢰 계획

AD(Active Directory) 및 IdM(Identity Management)은 Kerberos, LDAP, DNS 및 인증서 서비스와 같은 다양한 핵심 서비스를 관리하는 두 가지 대체 환경입니다. 가장 *간 신뢰 관계*는 모든 핵심 서비스가 원활하게 상호 작용할 수 있도록 하여 이러한 두 가지 환경을 투명하게 통합합니다. 다음 섹션에서는 가장 간 신뢰 배포를 계획 및 설계하는 방법에 대한 조언을 제공합니다.

7.1. IDM과 AD 간의 상호 간 신뢰

순수한 AD(Active Directory) 환경에서 간 신뢰는 두 개의 서로 다른 AD forest 루트 도메인을 연결합니다. AD와 IdM 간에 가장 간결한 신뢰를 생성하는 경우 IdM 도메인은 AD에 단일 도메인이 있는 별도의 포리스트로 표시됩니다. 그런 다음 AD forest 루트 도메인과 IdM 도메인 간에 신뢰 관계가 설정됩니다. 결과적으로 AD forest의 사용자는 IdM 도메인의 리소스에 액세스할 수 있습니다.

IdM은 하나의 AD forest 또는 여러 개의 관련이 없는 산모로 신뢰를 구축할 수 있습니다.

참고

두 개의 별도의 Kerberos 영역은 *교차 실제 신뢰* 에서 연결할 수 있습니다. 그러나 Kerberos 영역은 ID 및 권한 부여 작업과 관련된 다른 서비스 및 프로토콜이 아닌 인증에만 적용됩니다. 따라서 Kerberos 교차 실제 신뢰를 구축하는 것만으로는 한 영역의 사용자가 다른 영역의 리소스에 액세스할 수 있도록 하는 것만으로는 충분하지 않습니다.

AD 도메인에 대한 외부 신뢰

외부 신뢰는 IdM과 Active Directory 도메인 간의 신뢰 관계입니다. Forest trust에서는 항상 IdM과 Active Directory forest의 루트 도메인 간의 신뢰가 필요하지만 외부 신뢰는 IdM에서est 내의 모든 도메인으로 설정할 수 있습니다.

7.2. 신뢰 컨트롤러 및 신뢰 에이전트

IdM(Identity Management)은 AD(Active Directory)에 대한 신뢰를 지원하는 다음 유형의 IdM 서버를 제공합니다.

신뢰 컨트롤러

AD 도메인 컨트롤러에 대해 ID 조회를 수행할 수 있는 IdM 서버입니다. 또한 AD와의 신뢰를 구축할 수 있도록 Samba 제품군을 실행합니다. AD 도메인 컨트롤러는 AD에 대한 신뢰를 설정하고 검증할 때 신뢰 컨트롤러에 문의합니다. AD-enrolled 시스템은 Kerberos 인증 요청을 위한 IdM 신뢰 컨트롤러와 통신합니다.

첫 번째 신뢰 컨트롤러는 신뢰를 구성할 때 생성됩니다. 서로 다른 지리적 위치에 여러 도메인 컨트롤러가 있는 경우 **ipa-adtrust-install** 명령을 사용하여 **RHEL IdM** 서버를 이러한 위치에서 신뢰 컨트롤러로 지정합니다.

신뢰 컨트롤러는 신뢰 에이전트보다 더 많은 네트워크 기반 서비스를 실행하므로 잠재적인 침입자를 위해 더 큰 공격 면적을 제공합니다.

신뢰 에이전트

AD 도메인 컨트롤러에 대해 **RHEL IdM** 클라이언트의 **ID** 조회를 확인할 수 있는 **IdM** 서버입니다. 신뢰 컨트롤러와 달리 신뢰 에이전트는 **Kerberos** 인증 요청을 처리할 수 없습니다.

IdM 도메인에는 신뢰 에이전트와 컨트롤러 외에도 표준 **IdM** 서버가 포함될 수 있습니다. 그러나 이러한 서버는 **AD**와 통신하지 않습니다. 따라서 이러한 표준 서버와 통신하는 클라이언트는 **AD** 사용자 및 그룹을 확인하거나 **AD** 사용자를 인증하고 권한을 부여할 수 없습니다.

표 7.1. 신뢰 컨트롤러 및 신뢰 에이전트에서 지원하는 기능 비교

기능	신뢰 에이전트	신뢰 컨트롤러
AD 사용자 및 그룹 해결	있음	있음
신뢰할 수 있는 AD forests에서 사용자가 액세스할 수 있는 서비스를 실행하는 IdM 클라이언트 등록	있음	있음
신뢰 계약 추가, 수정 또는 제거	없음	있음
IdM 서버에 신뢰 에이전트 역할 할당	없음	있음

신뢰 컨트롤러 및 신뢰 에이전트 배포를 계획할 때 다음 지침을 고려하십시오.

- **IdM** 배포당 두 개 이상의 신뢰 컨트롤러를 구성합니다.
- 각 데이터 센터에 두 개 이상의 신뢰 컨트롤러를 구성합니다.

추가 신뢰 컨트롤러를 생성하거나 기존 신뢰 컨트롤러가 실패하는 경우 신뢰 에이전트 또는 표준 서버를 승격하여 새 신뢰 컨트롤러를 생성합니다. 이 작업을 수행하려면 **IdM** 서버에 **ipa-adtrust-install** 유틸리티를 사용하십시오.

중요

기존 신뢰 컨트롤러를 신뢰 에이전트로 다운그레이드할 수 없습니다.

7.3. 단방향 신뢰 및 양방향 신뢰

한 가지 방법으로 IdM(Identity Management)은 AD(Active Directory)를 신뢰하지만 AD는 IdM을 신뢰하지 않습니다. AD 사용자는 IdM 도메인의 리소스에 액세스할 수 있지만, IdM의 사용자는 AD 도메인 내의 리소스에 액세스할 수 없습니다. IdM 서버는 특수 계정을 사용하여 AD에 연결하고 LDAP를 통해 IdM 클라이언트로 전달되는 ID 정보를 읽습니다.

두 가지 방식으로 IdM 사용자는 AD에 인증할 수 있으며 AD 사용자는 IdM에 인증할 수 있습니다. AD 사용자는 하나의 신뢰 사례처럼 IdM 도메인의 리소스에 인증하고 액세스할 수 있습니다. IdM 사용자는 인증할 수 있지만 AD의 대부분의 리소스에 액세스할 수 없습니다. 액세스 제어 검사가 필요하지 않은 AD forests의 Kerberized 서비스에만 액세스할 수 있습니다.

AD 리소스에 대한 액세스 권한을 부여하려면 IdM에서 Global Catalog 서비스를 구현해야 합니다. 이 서비스는 IdM 서버의 현재 버전에 아직 존재하지 않습니다. 따라서 IdM과 AD 간의 양방향 신뢰는 거의 IdM과 AD 간의 단방향 신뢰와 거의 동일합니다.

7.4. 신뢰할 수 있는 도메인의 KERBEROS FAST

Kerberos flexible Authentication Secure Tunneling(FAST)은 AD(Active Directory) 환경에서도 Kerberos 무장이라고 합니다. Kerberos FAST는 클라이언트와 KDC(Key Distribution Center) 간의 Kerberos 통신을 위한 추가 보안 계층을 제공합니다. IdM에서 requirements는 IdM 서버에서 실행되고 FAST는 기본적으로 활성화되어 있습니다. IdM의 이중 인증 (2FA)도 FAST를 활성화해야 합니다.

AD AD에서 Kerberos 무장 기능은 기본적으로 AD Domain Controller(DC)에서 비활성화되어 있습니다. 아래 그룹 정책 설정을 사용하여 도메인 컨트롤러에서 활성화할 수 있습니다.

- 컴퓨터 설정,
- 정책,
- 관리 템플릿

- 시스템,
- KDC,
- fsGroup은 클레임, 복합 인증 및 Kerberos 대체를 지원합니다.

정책 설정은 다음 옵션을 허용합니다.

- "지원되지 않음",
- "지원",
- "항상 요구 사항 제공",
- "수증되지 않은 인증 요청 실패".

Kerberos FAST는 IdM 클라이언트의 Kerberos 클라이언트 라이브러리에 구현됩니다. IdM 클라이언트는 FAST를 광고하는 신뢰할 수 있는 모든 도메인에 대해 FAST를 사용하도록 또는 Kerberos FAST를 전혀 사용하지 않도록 IdM 클라이언트를 구성할 수 있습니다. 신뢰할 수 있는 AD forest에서 Kerberos 무장 기능을 활성화하면 IdM 클라이언트는 기본적으로 Kerberos FAST를 사용합니다. FAST는 암호화 키의 도움으로 보안 터널링을 설정합니다. 신뢰할 수 있는 도메인의 도메인 컨트롤러에 대한 연결을 보호하려면 Kerberos FAST가 Kerberos 영역 내에서만 유효하므로 Kerberos FAST는 신뢰할 수 있는 도메인에서 TGT(cross-realm ticket Granting Tickets)를 가져와야 합니다. Kerberos FAST는 IdM 클라이언트의 Kerberos 호스트 키를 사용하여 IdM 서버의 도움으로 교차 실제 TGT를 요청합니다. 이는 AD forest에서 양방향 신뢰가 필요한 IdM 도메인을 신뢰할 때만 작동합니다.

AD 정책에 강제 Kerberos FAST를 사용해야 하는 경우 IdM 도메인과 AD forest 간의 양방향 신뢰를 구축해야 합니다. IdM과 AD 측 모두 방향과 신뢰 유형에 대한 레코드가 있어야 하므로 연결이 설정되기 전에 이를 계획해야 합니다.

이미 단방향 신뢰를 설정한 경우 ipa trust-add ... --two-way=true 명령을 실행하면 기존 신뢰 계약이 제거되고 다시 생성됩니다. 이를 위해서는 관리자 자격 증명을 사용해야 합니다. IdM에서 AD 측에서 기존 신뢰 계약을 제거하려고 하므로 AD 액세스에 대한 관리자 권한이 필요합니다. AD 관리 계정이 아닌 공유 보안을 사용하여 원래의 신뢰를 설정하는 경우 신뢰를 양방향으로 다시 만들고 IdM 측에서 신뢰할 수 있

는 도메인 오브젝트만 변경합니다. Windows 관리자는 양방향 신뢰를 선택하고 동일한 공유 시크릿을 사용하여 신뢰를 재생성하여 Windows UI를 사용하여 동일한 프로세스를 반복해야 합니다.

양방향 트러스트를 사용할 수 없는 경우 모든 IdM 클라이언트에서 Kerberos FAST를 비활성화해야 합니다. 신뢰할 수 있는 AD forest의 사용자는 암호 또는 직접 스마트 카드로 인증할 수 있습니다. 이렇게 하려면 [domain] 섹션의 sssd.conf 파일에 다음 설정을 추가합니다.

```
krb5_use_fast = never
```

인증이 원격 Windows 클라이언트의 스마트 카드가 있는 ssh-keys, GSSAPI 인증 또는 ssh를 기반으로 하는 경우에는 이 옵션을 사용할 필요가 없습니다. IdM 클라이언트가 DC와 통신할 필요가 없기 때문에 이러한 방법은 Kerberos FAST를 사용하지 않습니다. 또한 IdM 클라이언트에서 FAST를 비활성화한 후 이중 인증 IdM 기능도 사용할 수 없습니다.

7.5. AD 사용자의 POSIX 및 ID 매핑 ID 범위 유형

IdM(Identity Management)은 사용자의 POSIX 사용자 ID(UID) 및 그룹 ID(GID)를 기반으로 액세스 제어 규칙을 적용합니다. 그러나 AD(Active Directory) 사용자는 SID(Security Identifiers)로 식별됩니다. AD 관리자는 uidNumber,gidNumber,unixHomeDirectory, loginShell 과 같은 AD 사용자 및 그룹에 대한 POSIX 속성을 저장하도록 AD를 구성할 수 있습니다.

ipa-ad-trust-posix ID 범위와 트러스트를 설정하여 이 정보를 참조하도록 상호 신뢰를 구성할 수 있습니다.

```
[server ~]# ipa trust-add --type=ad ad.example.com --admin administrator --password  --
range-type=ipa-ad-trust-posix
```

AD에 POSIX 속성을 저장하지 않는 경우 SSSD(System Security Services Daemon)는 ID 매핑이라는 프로세스의 사용자 SID에 따라 고유한 UID를 일관되게 매핑 할 수 있습니다. ipa-ad-trust ID 범위로 신뢰를 생성하여 이 동작을 명시적으로 선택할 수 있습니다.

```
[server ~]# ipa trust-add --type=ad ad.example.com --admin administrator --password  --
range-type=ipa-ad-trust
```

주의

신뢰를 생성할 때 ID 범위 유형을 지정하지 않으면 IdM에서est 루트 도메인의 AD 도메인 컨트롤러에서 세부 정보를 요청하여 적절한 범위 유형을 자동으로 선택합니다. IdM에서 POSIX 속성을 감지하지 않는 경우 신뢰 설치 스크립트는 Active Directory 도메인 ID 범위를 선택합니다.

IdM이 forest 루트 도메인의 POSIX 속성을 감지하면 신뢰 설치 스크립트에서 POSIX 속성 ID 범위가 있는 Active Directory 도메인 을 선택하고 해당 UID 및 GID 가 AD에 올바르게 정의되어 있다고 가정합니다. AD AD에서 POSIX 속성이 올바르게 설정되지 않은 경우 AD 사용자를 확인할 수 없습니다.

예를 들어 IdM 시스템에 대한 액세스 권한이 필요한 사용자 및 그룹이 forest root 도메인의 하위 도메인에 포함되지 않은 경우 설치 스크립트에서 하위 AD 도메인에 정의된 POSIX 특성을 탐지하지 못할 수 있습니다. 이 경우 신뢰를 설정할 때 POSIX ID 범위 유형을 명시적으로 선택하는 것이 좋습니다.

추가 리소스

- AD 사용자의 개인 그룹을 자동으로 매핑하는 옵션

7.6. AD 사용자의 개인 그룹을 자동으로 매핑하는 옵션: POSIX 신뢰

Linux 환경의 각 사용자에게는 기본 사용자 그룹이 있습니다. RHEL(Red Hat Enterprise Linux)은 사용자 개인 그룹(UPG) 스키마를 사용합니다. UPG는 해당 그룹이 생성된 사용자와 같고 해당 사용자는 UPG의 유일한 멤버입니다.

AD 사용자에 대해 UID를 할당했지만 GID가 추가되지 않은 경우 해당 ID 범위에 대한 auto_private_groups 설정을 조정하여 UID를 기반으로 사용자의 개인 그룹을 자동으로 매핑하도록 SSSD를 구성할 수 있습니다.

기본적으로 POSIX 신뢰에 사용되는 ipa-ad-trust-posix ID 범위에 대해 auto_private_groups 옵션이 false로 설정됩니다. 이 설정을 사용하면 SSSD는 각 AD 사용자 항목에서 uidNumber 및 gidNumber 를 검색합니다.

```
auto_private_groups = false
```

SSSD는 사용자의 UID인 gidNumber 값을 사용자의 GID에 uidNumber 값을 할당합니다. 해당 GID가 있는 그룹은 AD에 있어야 합니다. 그렇지 않으면 해당 사용자를 확인할 수 없습니다. 다음 표에서는 다른 AD 구성에 따라 AD 사용자를 확인할 수 있는지 여부를 보여 줍니다.The following table demonstrates whether you will be able to resolve AD users, depending on different AD configurations.

표 7.2. POSIX ID 범위에 대해 auto_private_groups 변수가 false 로 설정된 경우 SSSD 동작

AD에서 사용자 구성	ID 사용자 이름출력
AD 사용자 항목에는 다음이 포함됩니다. • **uidNumber** = 4000 • **gidNumber** 가 정의되지 않았습니다. • AD에 **gidNumber** = 4000이 있는 그룹이 없습니다.	SSSD는 사용자를 확인할 수 없습니다.
AD 사용자 항목에는 다음이 포함됩니다. • **uidNumber** = 4000 • **gidNumber** = 4000 • AD에 **gidNumber** = 4000이 있는 그룹이 없습니다.	SSSD는 사용자를 확인할 수 없습니다.
AD 사용자 항목에는 다음이 포함됩니다. • **uidNumber** = 4000 • **gidNumber** = 4000 • AD에는 **gidNumber** = 4000인 그룹이 있습니다.	**# ID aduser@AD-DOMAIN.COM** **uid=4000(aduser@ad-domain.com)** **gid=4000(adgroup@ad-domain.com)** **groups=4000(adgroup@ad-domain.com), …**

AD 사용자에게 기본 그룹이 구성되어 있지 않거나 해당 gidNumber 가 기존 그룹에 해당하지 않는 경우, 사용자가 속하는 모든 그룹을 찾을 수 없기 때문에 IdM 서버가 해당 사용자를 올바르게 확인할 수 없습니다. 이 문제를 해결하려면 auto_private_groups 옵션을 true 또는 hybrid 로 설정하여 SSSD에서 자동 개인 그룹 매핑을 활성화할 수 있습니다.

auto_private_groups = true

SSSD는 항상 AD 사용자 항목의 uidNumber 와 일치하도록 gidNumber 가 설정된 개인 그룹을 매핑합니다.

표 7.3. POSIX ID 범위에 대해 auto_private_groups 변수가 true로 설정된 경우 SSSD 동작

AD에서 사용자 구성	ID 사용자 이름출력
AD 사용자 항목에는 다음이 포함됩니다. • **uidNumber** = 4000 • **gidNumber** 가 정의되지 않았습니다. • AD에는 GID=4000인 그룹이 없습니다.	**# ID aduser@AD-DOMAIN.COM** **uid=4000(aduser@ad-domain.com)** **gid=4000(aduser@ad-domain.com)** **groups=4000(aduser@ad-domain.com), ...**
AD 사용자 항목에는 다음이 포함됩니다. • **uidNumber** = 4000 • **gidNumber** = 5000 • AD에는 **gidNumber** = 5000이 있는 그룹이 없습니다.	**# ID aduser@AD-DOMAIN.COM** **uid=4000(aduser@ad-domain.com)** **gid=4000(aduser@ad-domain.com)** **groups=4000(aduser@ad-domain.com), ...**
AD 사용자 항목에는 다음이 포함됩니다. • **uidNumber** = 4000 • **gidNumber** = 4000 • AD에는 **gidNumber** = 4000인 그룹이 없습니다.	**# ID aduser@AD-DOMAIN.COM** **uid=4000(aduser@ad-domain.com)** **gid=4000(aduser@ad-domain.com)** **groups=4000(aduser@ad-domain.com), ...**
AD 사용자 항목에는 다음이 포함됩니다. • **uidNumber** = 4000 • **gidNumber** = 5000 • AD에는 **gidNumber** = 5000이 있는 그룹이 있습니다.	**# ID aduser@AD-DOMAIN.COM** **uid=4000(aduser@ad-domain.com)** **gid=4000(aduser@ad-domain.com)** **groups=4000(aduser@ad-domain.com), ...**

auto_private_groups = hybrid

uidNumber 값이 gidNumber 와 일치하지만 이 gidNumber 가 있는 그룹이 없는 경우 SSSD는 개인 그룹을 uidNumber와 일치하는 gidNumber 와 함께 사용자의 기본 사용자 그룹으로 매핑합니다. uidNumber 및 gidNumber 값이 다르고 이 gidNumber가 있는 그룹이 있는 경우 SSSD는 gidNumber 의 값을 사용합니다.

표 7.4. POSIX ID 범위에 대해 auto_private_groups 변수가 hybrid 로 설정된 경우 SSSD 동작

AD에서 사용자 구성	ID 사용자 이름출력
AD 사용자 항목: • **uidNumber** = 4000 • **gidNumber** 가 정의되지 않았습니다. • AD에는 **gidNumber** = 4000인 그룹이 없습니다.	SSSD는 사용자를 확인할 수 없습니다.
AD 사용자 항목: • **uidNumber** = 4000 • **gidNumber** = 5000 • AD에는 **gidNumber** = 5000이 있는 그룹이 없습니다.	SSSD는 사용자를 확인할 수 없습니다.
AD 사용자 항목: • **uidNumber** = 4000 • **gidNumber** = 4000 • AD에는 **gidNumber** = 4000인 그룹이 없습니다.	**# ID aduser@AD-DOMAIN.COM** **uid=4000(aduser@ad-domain.com)** **gid=4000(aduser@ad-domain.com)** **groups=4000(aduser@ad-domain.com), …**
AD 사용자 항목: • **uidNumber** = 4000 • **gidNumber** = 5000 • AD에는 **gidNumber** = 5000이 있는 그룹이 있습니다.	**# ID aduser@AD-DOMAIN.COM** **uid=4000(aduser@ad-domain.com)** **gid=5000(aduser@ad-domain.com)** **groups=5000(adgroup@ad-domain.com), …**

추가 리소스

- AD 사용자의 POSIX 및 ID 매핑 ID 범위 유형
- CLI에서 POSIX ID 범위에 대한 자동 개인 그룹 매핑 활성화
- IdM WebUI에서 POSIX ID 범위에 대한 자동 개인 그룹 매핑 활성화

7.7. AD 사용자의 개인 그룹을 자동으로 매핑하는 옵션: ID 매핑 신뢰

Linux 환경의 각 사용자에게는 기본 사용자 그룹이 있습니다. RHEL(Red Hat Enterprise Linux)은 사용자 개인 그룹(UPG) 스키마를 사용합니다. UPG는 해당 그룹이 생성된 사용자와 같고 해당 사용자는 UPG의 유일한 멤버입니다.

AD 사용자에 대해 UID를 할당했지만 GID가 추가되지 않은 경우 해당 ID 범위에 대한 auto_private_groups 설정을 조정하여 UID를 기반으로 사용자의 개인 그룹을 자동으로 매핑하도록 SSSD를 구성할 수 있습니다.

기본적으로 ID 매핑 신뢰에 사용되는 ipa-ad-trust ID 범위에 대해 auto_private_groups 옵션이 true 로 설정됩니다. 이 구성을 사용하면 SSSD는 SID(Security Identifier)를 기반으로 AD 사용자의 UID 및 GID를 계산합니다. SSSD는 uidNumber,gidNumber 와 같은 AD의 POSIX 속성을 무시하고 primaryGroupID 를 무시합니다.

auto_private_groups = true

SSSD는 항상 UID와 일치하도록 GID가 설정된 개인 그룹을 AD 사용자의 SID에 따라 매핑합니다.

표 7.5. ID 매핑 ID 범위에 대해 auto_private_groups 변수가 true 로 설정된 경우 SSSD 동작

AD에서 사용자 구성	ID 사용자 이름출력

AD 사용자 항목:
• SID는 7000에 매핑됩니다.
• primaryGroupID 가 8000에 매핑됩니다.
	# ID aduser@AD-DOMAIN.COM
	uid=7000(aduser@ad-domain.com)
	gid=7000(aduser@ad-domain.com)
	groups=7000(aduser@ad-domain.com),
	8000(adgroup@ad-domain.com), …

auto_private_groups = false

auto_private_groups 옵션을 false 로 설정하면 SSSD에서 AD 항목에 설정된 primaryGroupID 를 GID 번호로 사용합니다. primaryGroupID 의 기본값은 AD의 Domain Users 그룹에 해당합니다.

표 7.6. ID 매핑 ID 범위에 대해 auto_private_groups 변수가 false 로 설정된 경우 SSSD 동작

AD에서 사용자 구성	ID 사용자 이름출력

AD 사용자 항목:
• SID는 7000에 매핑됩니다.
• primaryGroupID 가 8000에 매핑됩니다.
	# ID aduser@AD-DOMAIN.COM
	uid=7000(aduser@ad-domain.com)
	gid=8000(adgroup@ad-domain.com)
	groups=8000(adgroup@ad-domain.com), …

추가 리소스

- AD 사용자의 POSIX 및 ID 매핑 ID 범위 유형

7.8. CLI에서 POSIX ID 범위에 대한 자동 개인 그룹 매핑 활성화

기본적으로, AD에 저장된 POSIX 데이터에 의존하는 POSIX 신뢰를 설정한 경우 SSSD는 AD(Active Directory) 사용자에게 개인 그룹을 매핑하지 않습니다. AD 사용자에게 기본 그룹이 구성되어 있지 않으면 IdM에서 해당 그룹을 확인할 수 없습니다.

이 절차에서는 명령줄에서 auto_private_groups SSSD 매개변수에 대해 hybrid 옵션을 설정하여 ID 범위에 대한 자동 개인 그룹 매핑을 활성화하는 방법을 설명합니다. 결과적으로 IdM은 AD에 기본 그룹이 구성되지 않은 AD 사용자를 확인할 수 있습니다.

사전 요구 사항

- IdM과 AD 환경 간에 POSIX 간 신뢰를 성공적으로 설정했습니다.

절차

1. 모든 ID 범위를 표시하고 수정할 AD ID 범위를 기록합니다.

```
[root@server ~]# 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: AD.EXAMPLE.COM_id_range
  First Posix ID of the range: 1337000000
  Number of IDs in the range: 200000
  Domain SID of the trusted domain: S-1-5-21-4123312420-990666102-3578675309
  Range type: Active Directory trust range with POSIX attributes
----------------------------
Number of entries returned 2
----------------------------
```

2. ipa idrange-mod 명령을 사용하여 AD ID 범위의 자동 개인 그룹 동작을 조정합니다.

```
[root@server ~]# ipa idrange-mod --auto-private-groups=hybrid
AD.EXAMPLE.COM_id_range
```

3. 새 설정을 활성화하려면 SSSD 캐시를 재설정합니다.

```
[root@server ~]# sss_cache -E
```

추가 리소스

- AD 사용자의 개인 그룹을 자동으로 매핑하는 옵션

7.9. IDM WEBUI에서 POSIX ID 범위에 대한 자동 개인 그룹 매핑 활성화

기본적으로, AD에 저장된 POSIX 데이터에 의존하는 POSIX 신뢰를 설정한 경우 SSSD는 AD(Active Directory) 사용자에게 개인 그룹을 매핑하지 않습니다. AD 사용자에게 기본 그룹이 구성되어 있지 않으면 IdM에서 해당 그룹을 확인할 수 없습니다.

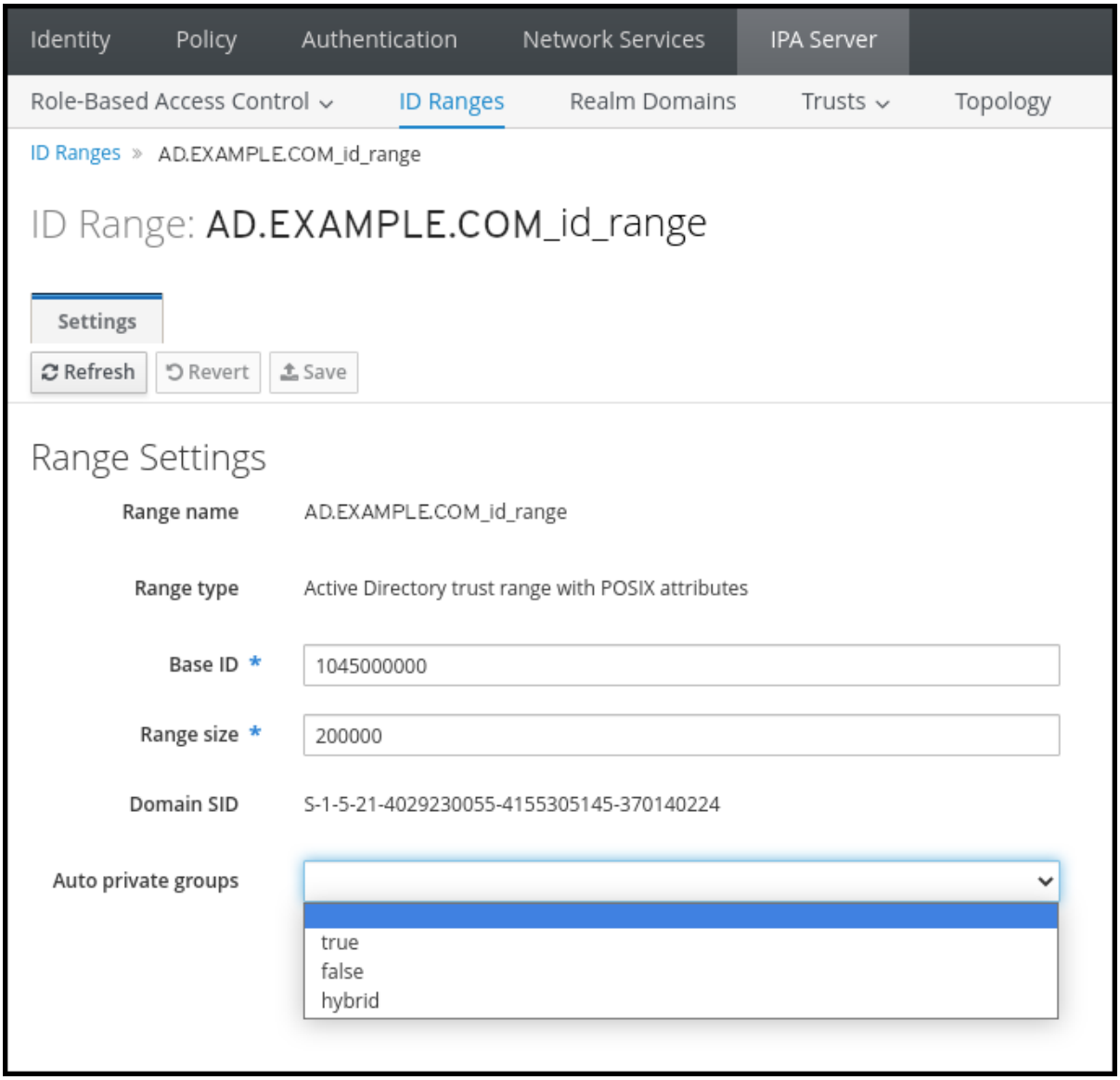
이 절차에서는 IdM(Identity Management) WebUI에서 auto_private_groups SSSD 매개변수에 대한 하이브리드 옵션을 설정하여 ID 범위에 대한 자동 개인 그룹 매핑을 활성화하는 방법을 설명합니다. 결과적으로 IdM은 AD에 기본 그룹이 구성되지 않은 AD 사용자를 확인할 수 있습니다.

사전 요구 사항

- IdM과 AD 환경 간에 POSIX 간 신뢰를 성공적으로 설정했습니다.

절차

1. 사용자 이름과 암호를 사용하여 IdM 웹 UI에 로그인합니다.
2. IPA 서버 → ID 범위 탭을 엽니다.
3. AD.EXAMPLE.COM_id_range 와 같이 수정할 ID 범위를 선택합니다.
4. Auto private groups (자동 개인 그룹) 드롭다운 메뉴에서 하이브리드 옵션을 선택합니다.



5. 저장 버튼을 클릭하여 변경 사항을 저장합니다.

추가 리소스

- AD 사용자의 개인 그룹을 자동으로 매핑하는 옵션

7.10. 비POSIX 외부 그룹 및 SID 매핑

IdM(Identity Management)은 그룹 관리에 LDAP를 사용합니다. Active Directory(AD) 항목이 동기화되거나 IdM으로 복사되지 않습니다. 즉, AD 사용자 및 그룹에는 LDAP 서버에 LDAP 오브젝트가 없으므로 IdM LDAP에서 그룹 멤버십을 표현하는데 직접 사용할 수 없습니다. 이러한 이유로 IdM의 관리자는 IdM에서 AD 사용자 및 그룹에 대해 그룹 멤버십을 서명하기 위해 일반 IdM LDAP 오브젝트로 참조되는 비POSIX 외부 그룹을 생성해야 합니다.

POSIX 이외의 외부 그룹의 SID(보안 ID)는 Active Directory의 그룹 SID를 IdM의 POSIX 그룹에 매핑하는 SSSD에서 처리합니다. Active Directory에서 SID는 사용자 이름과 연결됩니다. AD 사용자 이름이 IdM 리소스에 액세스하는 데 사용되는 경우 SSSD는 사용자의 SID를 사용하여 IdM 도메인의 사용자에 대한 전체 그룹 멤버십 정보를 빌드합니다.

7.11. DNS 설정

이러한 지침은 IdM(Identity Management)과 AD(Active Directory) 간 상호 신뢰를 구축하기 위한 올바른 DNS 구성을 달성하는 데 도움이 될 수 있습니다.

고유한 기본 DNS 도메인

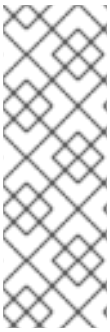
AD와 IdM 모두 고유한 기본 DNS 도메인이 구성되어 있는지 확인합니다. 예를 들어 다음과 같습니다.

- AD의 경우 *ad.example.com*, IdM의 경우 *idm.example.com*
- AD의 경우 *example.com*, IdM의 경우 *idm.example.com*

가장 편리한 관리 솔루션은 각 DNS 도메인이 통합된 DNS 서버에서 관리되는 환경이지만 다른 표준 규격 DNS 서버를 사용할 수도 있습니다.

IdM 및 AD DNS 도메인

IdM에 가입된 시스템은 여러 DNS 도메인을 통해 배포할 수 있습니다. Red Hat은 Active Directory가 소유한 것과 다른 DNS 영역에 IdM 클라이언트를 배포하는 것이 좋습니다. AD 트러스트를 지원하려면 기본 IdM DNS 도메인에 적절한 SRV 레코드가 있어야 합니다.



참고

IdM과 Active Directory 간의 신뢰가 있는 일부 환경에서는 Active Directory DNS 도메인에 포함된 호스트에 IdM 클라이언트를 설치할 수 있습니다. 그런 다음 호스트는 Linux에 중점을 둔 IdM 기능을 활용할 수 있습니다. 권장 구성이 아니며 몇 가지 제한 사항이 있습니다. 자세한 내용은 Active Directory DNS 도메인에서 IdM 클라이언트 구성을 참조하십시오.

적절한 SRV 레코드

AD 트러스트를 지원하기 위한 기본 IdM DNS 도메인에 적절한 SRV 레코드가 있는지 확인합니다.

동일한 IdM 영역의 일부인 다른 DNS 도메인의 경우 AD에 대한 신뢰가 설정될 때 SRV 레코드를 구성할 필요가 없습니다. 그 이유는 AD 도메인 컨트롤러에서 SRV 레코드를 사용하여 Kerberos 키 배포 센터(KDC)를 검색하지 않고 이름 접미사 라우팅 정보를 기반으로 하기 때문입니다.

신뢰의 모든 DNS 도메인에서 확인할 수 있는 DNS 레코드

모든 시스템이 신뢰 관계에 관련된 모든 DNS 도메인에서 DNS 레코드를 확인할 수 있는지 확인합니다.

- IdM DNS를 구성할 때 외부 CA를 사용하여 IdM 서버 설치에 설명된 지침을 따르십시오.
- 통합 DNS 없이 IdM을 사용하는 경우 통합 DNS 없이 IdM 서버 설치에 설명된 지침을 따르십시오.

Kerberos 영역 이름은 기본 DNS 도메인 이름의 대문자 버전으로

Kerberos 영역 이름이 모두 대문자로 기본 DNS 도메인 이름과 같은지 확인합니다. 예를 들어, 도메인 이름이 AD의 경우 *ad.example.com* 이고 IdM의 경우 *idm.example.com* 이 IdM의 경우 Kerberos 영역 이름은 *AD. EXAMPLE.COM* 이어야 합니다.

7.12. NETBIOS 이름

NetBIOS 이름은 일반적으로 도메인 이름의 왼쪽 구성 요소입니다. 예를 들어 다음과 같습니다.

- 도메인 이름 *linux.example.com* 에서 NetBIOS 이름은 linux 입니다.
- 도메인 이름 *example*.com 에서 NetBIOS 이름은 입니다.

IdM(Identity Management) 및 AD(Active Directory) 도메인의 서로 다른 NetBIOS 이름

IdM 및 AD 도메인의 NetBIOS 이름이 다른지 확인하십시오.

NetBIOS 이름은 AD 도메인을 식별하는 데 매우 중요합니다. IdM 도메인이 AD DNS의 하위 도메인에 있는 경우, IdM 도메인 및 서비스를 식별하는 데 NetBIOS 이름도 중요합니다.

NetBIOS 이름에 대한 문자 제한

NetBIOS 이름의 최대 길이는 15자입니다.

7.13. 지원되는 WINDOWS SERVER 버전

IdM(Identity Management)은 Windows Server 2008 R2 또는 이전 버전을 실행하는 Active Directory에 대한 신뢰 설정을 지원하지 않습니다. RHEL IdM은 Windows Server 2012 이상에서만 지원되는 신뢰 관계를 설정할 때 SMB 암호화가 필요합니다.

다음 포리스트 및 도메인 기능 수준을 사용하는 AD(Active Directory) 버스트와의 신뢰 관계를 설정할 수 있습니다.You can establish a trust relationship with Active Directory (AD) forests that use the following forest and domain functional levels:

- Forest 기능 수준 범위: Windows Server 2012:Windows Server 2016
- 도메인 기능 수준 범위: Windows Server 2012:Windows Server 2016

IdM(Identity Management)은 다음 운영 체제를 실행하는 Active Directory 도메인 컨트롤러에서 신뢰 설정을 지원합니다.

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019

7.14. AD 서버 검색 및 선호도 구성

서버 검색 및 선호도 구성은 IdM(Identity Management) 클라이언트가 통신하는 AD(Active Directory) 서버에 영향을 미칩니다. 이 섹션에서는 IdM과 AD 간의 상호 신뢰가 있는 환경에서 검색 및 선호도가 작동하는 방식에 대해 설명합니다.

동일한 지리적 위치에 있는 서버를 선호하도록 클라이언트를 구성하면 클라이언트가 다른 원격 데이터 센터의 서버에 연결할 때 발생하는 시간 지연 및 기타 문제를 방지할 수 있습니다. 클라이언트가 로컬 서버와 통신하도록 하려면 다음 조건을 충족해야 합니다.

- 클라이언트는 **LDAP** 및 **Kerberos**를 통해 로컬 **IdM** 서버와 통신
- 클라이언트가 **Kerberos**를 통해 로컬 **AD** 서버와 통신
- **IdM** 서버의 내장 클라이언트는 **LDAP** 및 **Kerberos**를 통해 로컬 **AD** 서버와 통신합니다.

로컬 **IdM** 서버와의 통신을 위해 **IdM** 클라이언트에서 **LDAP** 및 **Kerberos** 구성 옵션

통합 **DNS**에서 **IdM**을 사용하는 경우

기본적으로 클라이언트는 **DNS** 레코드를 기반으로 자동 서비스 조회를 사용합니다. 이 설정에서 *DNS 위치 기능을 사용하여 DNS* 기반 서비스 검색을 구성할 수도 있습니다.

자동 조회를 덮어쓰려면 다음 방법 중 하나에서 **DNS** 검색을 비활성화할 수 있습니다.

- 명령줄에서 장애 조치 매개 변수를 제공하여 **IdM** 클라이언트 설치 중
- **SSSD(System Security Services Daemon)** 구성을 수정하여 클라이언트 설치 후

통합된 **DNS** 없이 **IdM**을 사용하는 경우

다음 방법 중 하나로 명시적으로 클라이언트를 구성해야 합니다.

- 명령줄에서 장애 조치 매개 변수를 제공하여 **IdM** 클라이언트 설치 중
- **SSSD** 구성을 수정하여 클라이언트 설치 후

로컬 **AD** 서버와의 통신을 위해 **IdM** 클라이언트에 **Kerberos**를 구성하는 옵션

IdM 클라이언트는 통신할 **AD** 서버를 자동으로 검색할 수 없습니다. **AD** 서버를 수동으로 지정하려면 **krb5.conf** 파일을 수정합니다.

- AD 영역 정보 추가
- 통신할 AD 서버를 명시적으로 나열합니다.

예를 들어 다음과 같습니다.

```
[realms]
AD.EXAMPLE.COM = {
kdc = server1.ad.example.com
kdc = server2.ad.example.com
}
```

Kerberos 및 LDAP를 통한 로컬 AD 서버와의 통신을 위해 IdM 서버에 포함된 클라이언트를 구성하는 옵션

IdM 서버에 포함된 클라이언트는 AD 서버의 클라이언트로도 작동합니다. 적절한 AD 사이트를 자동으로 검색하고 사용할 수 있습니다.

포함된 클라이언트가 검색을 수행할 때 먼저 원격 위치에서 AD 서버를 검색할 수 있습니다. 원격 서버에 연결하려고 하면 클라이언트가 연결을 설정하지 않고 작업을 중지할 수 있습니다. 클라이언트의 sssd.conf 파일에서 dns_resolver_timeout 옵션을 사용하여 클라이언트가 DNS 확인자에서 응답을 기다리는 시간을 늘립니다. 자세한 내용은 *sssd.conf(5)* 매뉴얼 페이지를 참조하십시오.

포함된 클라이언트가 로컬 AD 서버와 통신하도록 구성되면 SSSD는 포함된 클라이언트가 속한 AD 사이트를 기억합니다. SSSD는 일반적으로 로컬 도메인 컨트롤러에 LDAP ping을 직접 전송하여 사이트 정보를 새로 고침합니다. 사이트가 더 이상 존재하지 않거나 클라이언트가 다른 사이트에 할당된 경우 SSSD는 overflow에서 SRV 레코드를 조회하기 시작하고 자동 검색 전체 프로세스를 거칩니다.

sssd.conf 에서 *신뢰할 수 있는 도메인 섹션을* 사용하면 기본적으로 자동으로 검색되는 일부 정보를 명시적으로 덮어쓸 수도 있습니다.

7.15. IDM을 AD에 간접 통합 중 수행되는 작업

이 섹션에서는 IdM을 AD에 간접 통합 중에 수행되는 작업 및 요청을 자세히 설명합니다.

표에서 IdM(Identity Management)을 생성하는 동안 AD(Identity Management)에서 AD(Active Directory)에 대한 신뢰로 수행한 작업 및 요청에 대해 알아보십시오.

표 7.7. AD 도메인 컨트롤러에 대한 IdM 신뢰 컨트롤러에서 수행되는 작업

작업	사용된 프로토콜	목적
IdM 신뢰 컨트롤러에 구성된 AD DNS 확인기에 대한 DNS 확인	DNS	AD 도메인 컨트롤러의 IP 주소를 검색하려면 다음을 수행합니다.
AD DC의 UDP/UDP6 포트 389에 대한 요청	연결 없는 LDAP(CLDAP)	AD DC 검색을 수행하려면
AD DC의 TCP/TCP6 포트 389 및 3268에 대한 요청	LDAP	AD 사용자 및 그룹 정보를 쿼리하려면
AD DC의 TCP/TCP6 포트 389 및 3268에 대한 요청	DCE RPC 및 SMB	AD에 대한 상호 신뢰를 설정하고 지원하는 방법
AD DC의 TCP/TCP6 포트 135, 139, 445에 대한 요청	DCE RPC 및 SMB	AD에 대한 상호 신뢰를 설정하고 지원하는 방법
49152-65535(TCP/TCP6) 범위의 Active Directory 도메인 컨트롤러에서 지시한 대로 AD DC에서 동적으로 열린 포트로 요청	DCE RPC 및 SMB	DCE RPC 엔드 포인트 매퍼(포트 135 TCP/TCP6)의 요청에 응답하기
AD DC의 포트 88(TCP/TCP6 및 UDP/UDP6), 464(TCP/TCP6 및 UDP/UDP6), 749(TCP/TCP6)에 대한 요청	Kerberos	Kerberos 티켓을 얻으려면 Kerberos 암호를 변경하고 Kerberos를 원격으로 관리합니다.

표를 읽고 IdM 신뢰 컨트롤러에 이르기까지 AD 도메인 컨트롤러에서 AD 신뢰로 IdM에 IdM을 생성하는 동안 수행된 작업 및 요청에 대해 알아보십시오.

표 7.8. IdM 신뢰 컨트롤러에 대해 AD 도메인 컨트롤러에서 수행되는 작업

작업	사용된 프로토콜	목적
AD 도메인 컨트롤러에 구성된 IdM DNS 확인에 대한 DNS 확인	DNS	IdM 신뢰 컨트롤러의 IP 주소 검색
IdM 신뢰 컨트롤러의 UDP/UDP6 포트 389에 대한 요청	CLDAP	IdM 신뢰 컨트롤러 검색을 수행하려면
IdM 신뢰 컨트롤러의 TCP/TCP6 포트 135, 139, 445에 대한 요청	DCE RPC 및 SMB	AD에 대한 가장 큰 신뢰를 확인하려면 To verify the cross-forest trust to AD

작업	사용된 프로토콜	목적
IdM 신뢰 컨트롤러에서 안내하는 대로 IdM 신뢰 컨트롤러에서 동적으로 열린 포트에 대한 요청은 49152-65535(TCP/TCP6) 범위의 경우	DCE RPC 및 SMB	DCE RPC 엔드 포인트 매퍼(포트 135 TCP/TCP6)의 요청에 응답하기
IdM 신뢰 컨트롤러의 포트 88(TCP/TCP6 및 UDP/UDP6), 464(TCP/TCP6 및 UDP/UDP6), 749(TCP/TCP6)에 대한 요청	Kerberos	Kerberos 티켓을 얻으려면 Kerberos 암호를 변경하고 Kerberos를 원격으로 관리합니다.

8장. 기타 RED HAT 제품과 IDM 통합

이 섹션에서는 IdM과 통합되는 다른 Red Hat 제품의 문서 링크를 제공합니다. IdM 사용자가 서비스에 액세스할 수 있도록 이러한 제품을 구성할 수 있습니다.

Ansible Automation Platform

- LDAP 인증 설정

OpenShift Container Platform

- LDAP ID 공급자 구성

OpenStack Platform

- OpenStack Identity(keystone)와 Red Hat IdM(Identity Manager) 통합

Satellite

- Red Hat Identity Management 사용

로그인

- SSSD 및 FreeIPA ID 관리 통합

가상화

- 외부 LDAP 공급자 구성

9장. IDM 백업 및 복원

Red Hat Enterprise Linux Identity Management는 IdM 시스템을 수동으로 백업하고 복원할 수 있는 솔루션을 제공합니다. 이는 데이터 손실 이벤트 이후에 필요할 수 있습니다.

백업 중에 시스템은 IdM 설정에 대한 정보가 포함된 디렉터리를 생성하여 저장합니다. 복원하는 동안 이 백업 디렉터리를 사용하여 원래 IdM 설정을 다시 가져올 수 있습니다.

참고

IdM 백업 및 복원 기능은 데이터 손실을 방지하도록 설계되었습니다. 서버 손실의 영향을 완화하고 클라이언트에 대체 서버를 제공하여 지속적인 작업을 보장하기 위해 복제를 통해 Mitigating server loss 에 따라 복제본 토폴로지가 있어야 합니다.

9.1. IDM 백업 유형

ipa-backup 유틸리티를 사용하면 다음 두 가지 유형의 백업을 생성할 수 있습니다.

전체 서버 백업

- IdM과 관련된 모든 서버 구성 파일과 LDAP 데이터를 LDIF(LDAP 데이터 교환 형식) 파일에 포함
- IdM 서비스는 오프라인 상태여야 합니다.
- IdM 배포를 처음부터 다시 빌드하는 데 적합합니다.

데이터 전용 백업

- LDIF 파일에 LDAP 데이터 및 복제 변경 로그가 포함되어 있습니다.
- IdM 서비스는 온라인 또는 오프라인 상태일 수 있습니다.
- IdM 데이터를 과거 상태로 복원하는 데 적합합니다.

9.2. IDM 백업 파일에 대한 명명 규칙

기본적으로 IdM은 /var/lib/ipa/backup/ 디렉터리의 하위 디렉토리에 백업을 .tar 아카이브로 저장합니다.

아카이브 및 하위 디렉터리는 다음과 같은 명명 규칙을 따릅니다.

전체 서버 백업

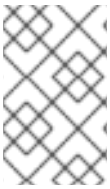
ipa-full- < *YEAR-MM-DD-HH-MM-SS*> *디렉터리의 ipa- full*.tar 이라는 아카이브로 scc 시간에 지정된 시간입니다.

```
[root@server ~]# ll /var/lib/ipa/backup/ipa-full-2021-01-29-12-11-46
total 3056
-rw-r--r--. 1 root root     158 Jan 29 12:11 header
-rw-r--r--. 1 root root 3121511 Jan 29 12:11 ipa-full.tar
```

데이터 전용 백업

ipa-data- < *YEAR-MM-DD-HH-MM-SS*> *디렉터리의 ipa- data*.tar 이라는 아카이브로 kafka 시간에 지정됩니다.

```
[root@server ~]# ll /var/lib/ipa/backup/ipa-data-2021-01-29-12-14-23
total 1072
-rw-r--r--. 1 root root     158 Jan 29 12:14 header
-rw-r--r--. 1 root root 1090388 Jan 29 12:14 ipa-data.tar
```



참고

IdM 서버를 설치 제거해도 백업 파일이 자동으로 제거되지는 않습니다.

9.3. 백업 생성 시 고려 사항

이 섹션에서는 ipa-backup 명령의 중요한 동작 및 제한 사항에 대해 설명합니다.

- 기본적으로 ipa-backup 유틸리티는 오프라인 모드로 실행되므로 모든 IdM 서비스가 중지됩니다. 백업이 완료된 후 유틸리티에서 IdM 서비스를 자동으로 다시 시작합니다.
- 전체 서버 백업은 항상 IdM 서비스를 오프라인 상태로 실행해야 하지만 데이터 전용 백업은

온라인 서비스를 통해 수행할 수 있습니다.

- 기본적으로 **ipa-backup** 유틸리티는 **/var/lib/ipa/backup/** 디렉토리를 포함하는 파일 시스템에 백업을 생성합니다. IdM에서 사용하는 프로덕션 파일 시스템과는 별도의 파일 시스템에 백업을 정기적으로 생성하고, 테이프 또는 광 스토리지와 같은 고정된 매체로 백업을 보관할 것을 권장합니다.

- 숨겨진 복제본에서 백업을 수행하는 것이 좋습니다. IdM 서비스는 IdM 클라이언트에 영향을 미치지 않고 숨겨진 복제본에서 종료할 수 있습니다.

- **ipa-backup** 유틸리티는 IdM 클러스터에 사용된 모든 서비스(인증 기관), DNS(Domain Name System) 및 KRA(키 복구 에이전트)가 백업을 실행하는 서버에 설치되어 있는지 확인합니다. 서버에 이러한 서비스가 모두 설치되어 있지 않으면 해당 호스트에서 가져온 백업이 전체 클러스터 복원에 충분하지 않기 때문에 **ipa-backup** 유틸리티는 경고로 종료됩니다.

  예를 들어 IdM 배포에서 통합 CA(인증 기관)를 사용하는 경우 CA가 아닌 복제본에서 실행된 백업이 CA 데이터를 캡처하지 않습니다. **ipa-backup** 을 수행하는 복제본에 설치된 클러스터에 사용된 모든 IdM 서비스가 있는지 확인하는 것이 좋습니다.

  ipa-backup --disable-role-check 명령을 사용하여 IdM 서버 역할 검사를 우회할 수 있지만 결과 백업에는 IdM을 완전히 복원하는 데 필요한 모든 데이터가 포함되어 있지 않습니다.

9.4. IDM 백업 생성

이 섹션에서는 **ipa-backup** 명령을 사용하여 오프라인 및 온라인 모드에서 전체 서버 및 데이터 전용 백업을 생성하는 방법을 설명합니다.

사전 요구 사항

- **ipa-backup** 유틸리티를 실행하려면 **root** 권한이 있어야 합니다.

절차

- 오프라인 모드에서 전체 서버 백업을 생성하려면 추가 옵션 없이 **ipa-backup** 유틸리티를 사용합니다.

```
[root@server ~]# ipa-backup
Preparing backup on server.example.com
Stopping IPA services
```

```
Backing up ipaca in EXAMPLE-COM to LDIF
Backing up userRoot in EXAMPLE-COM to LDIF
Backing up EXAMPLE-COM
Backing up files
Starting IPA service
Backed up to /var/lib/ipa/backup/ipa-full-2020-01-14-11-26-06
The ipa-backup command was successful
```

- 오프라인 데이터 전용 백업을 생성하려면 **--data** 옵션을 지정합니다.

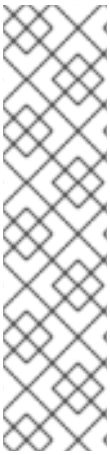
```
[root@server ~]# ipa-backup --data
```

- IdM 로그 파일이 포함된 전체 서버 백업을 생성하려면 **--logs** 옵션을 사용합니다.

```
[root@server ~]# ipa-backup --logs
```

- IdM 서비스가 실행되는 동안 데이터 전용 백업을 생성하려면 **--data** 및 **--online** 옵션을 모두 지정합니다.

```
[root@server ~]# ipa-backup --data --online
```



참고

/tmp 디렉터리의 공간이 부족하여 백업이 실패하는 경우 **TMPDIR** 환경 변수를 사용하여 백업 프로세스에서 생성한 임시 파일의 대상을 변경합니다.

```
[root@server ~]# TMPDIR=/new/location ipa-backup
```

자세한 내용은 ipa-backup Command Fails to Finish 를 참조하십시오.

검증 단계

- 백업 디렉터리에는 백업이 있는 아카이브가 포함되어 있습니다.

```
[root@server ~]# ls /var/lib/ipa/backup/ipa-full-2020-01-14-11-26-06
header  ipa-full.tar
```

9.5. GPG2 암호화 IDM 백업 생성

GPG(GNU Privacy Guard) 암호화를 사용하여 암호화된 백업을 만들 수 있습니다. 다음 절차에서는 IdM 백업을 생성하고 GPG2 키를 사용하여 암호화합니다.

사전 요구 사항

- GPG2 키를 생성했습니다. GPG2 키 생성을 참조하십시오.

절차

- --gpg 옵션을 지정하여 GPG 암호화 백업을 만듭니다.

```
[root@server ~]# ipa-backup --gpg
Preparing backup on server.example.com
Stopping IPA services
Backing up ipaca in EXAMPLE-COM to LDIF
Backing up userRoot in EXAMPLE-COM to LDIF
Backing up EXAMPLE-COM
Backing up files
Starting IPA service
Encrypting /var/lib/ipa/backup/ipa-full-2020-01-13-14-38-00/ipa-full.tar
Backed up to /var/lib/ipa/backup/ipa-full-2020-01-13-14-38-00
The ipa-backup command was successful
```

검증 단계

- 백업 디렉터리에 .gpg 파일 확장자가 있는 암호화된 아카이브가 포함되어 있는지 확인합니다.

```
[root@server ~]# ls /var/lib/ipa/backup/ipa-full-2020-01-13-14-38-00
header  ipa-full.tar.gpg
```

추가 리소스

- 백업 생성.

9.6. GPG2 키 생성

다음 절차에서는 암호화 유틸리티에 사용할 GPG2 키를 생성하는 방법을 설명합니다.

사전 요구 사항

- 루트 권한이 필요합니다.

절차

1. pinentry 유틸리티를 설치하고 구성합니다.

```
[root@server ~]# dnf install pinentry
[root@server ~]# mkdir ~/.gnupg -m 700
[root@server ~]# echo "pinentry-program /usr/bin/pinentry-curses" >> ~/.gnupg/gpg-agent.conf
```

2. 기본 세부 정보를 사용하여 GPG 키 쌍을 생성하는 데 사용되는 키 입력 파일을 만듭니다. 예를 들어 다음과 같습니다.

```
[root@server ~]# cat >key-input <<EOF
%echo Generating a standard key
Key-Type: RSA
Key-Length: 2048
Name-Real: GPG User
Name-Comment: first key
Name-Email: root@example.com
Expire-Date: 0
%commit
%echo Finished creating standard key
EOF
```

3. *(선택 사항)* 기본적으로 GPG2는 인증 키를 ~/.gnupg 파일에 저장합니다. 사용자 정의 인증 키 위치를 사용하려면 GNUPGHOME 환경 변수를 root만 액세스할 수 있는 디렉터리로 설정합니다.

```
[root@server ~]# export GNUPGHOME=/root/backup

[root@server ~]# mkdir -p $GNUPGHOME -m 700
```

4. 키 입력 파일의 콘텐츠를 기반으로 새 GPG2 키를 생성합니다.

```
[root@server ~]# gpg2 --batch --gen-key key-input
```

5. GPG2 키를 보호하는 암호를 입력합니다. 이 암호를 사용하여 암호 해독을 위해 개인 키에 액세스합니다.

```
┌──────────────────────────────────────────────────────┐
│ Please enter the passphrase to                       │
│ protect your new key                                 │
│                                                      │
│ Passphrase: <passphrase>                             │
│                                                      │
│  <OK>                    <Cancel>                    │
└──────────────────────────────────────────────────────┘
```

6. 암호를 다시 입력하여 올바른 암호를 확인합니다.

```
┌──────────────────────────────────────────────────────┐
│ Please re-enter this passphrase                      │
│                                                      │
│ Passphrase: <passphrase>                             │
│                                                      │
│  <OK>                    <Cancel>                    │
└──────────────────────────────────────────────────────┘
```

7. 새 GPG2 키가 성공적으로 생성되었는지 확인합니다.

```
gpg: keybox '/root/backup/pubring.kbx' created
gpg: Generating a standard key
gpg: /root/backup/trustdb.gpg: trustdb created
gpg: key BF28FFA302EF4557 marked as ultimately trusted
gpg: directory '/root/backup/openpgp-revocs.d' created
gpg: revocation certificate stored as '/root/backup/openpgp-
revocs.d/8F6FCF10C80359D5A05AED67BF28FFA302EF4557.rev'
gpg: Finished creating standard key
```

검증 단계

- 서버의 GPG 키를 나열합니다.

```
[root@server ~]# gpg2 --list-secret-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
/root/backup/pubring.kbx
------------------------
sec   rsa2048 2020-01-13 [SCEA]
      8F6FCF10C80359D5A05AED67BF28FFA302EF4557
uid           [ultimate] GPG User (first key) <root@example.com>
```

추가 리소스

- GNU 개인 정보 보호 워크워크

9.7. IDM 백업에서 복원해야 하는 경우

IdM 백업에서 복원하여 여러 재해 시나리오에 대응할 수 있습니다.

- 바람직하지 않은 변경 사항은 LDAP 콘텐츠를 수정 또는 삭제하고, 복제는 배포 전체에서 수행된 후 해당 변경 사항을 되돌리려는 것입니다. 데이터 전용 백업을 복원하면 IdM 구성 자체에 영향을 주지 않고 LDAP 항목이 이전 상태로 반환됩니다.

- 총 인프라 손실 또는 모든 CA 인스턴스의 손실: 재해로 인해 모든 인증 기관 복제본이 손상되면 추가 서버를 배포하여 다시 빌드할 수 있는 기능이 손실되었습니다. 이 경우 CA Replica의 백업을 복원하고 새 복제본을 작성합니다.

- 격리된 서버의 업그레이드 실패: 운영 체제가 계속 작동하지만 IdM 데이터가 손상되므로 IdM 시스템을 알려진 양호한 상태로 복원해야 합니다. Red Hat은 문제를 진단하고 해결하기 위해 기술 지원과 협력하는 것이 좋습니다. 이러한 작업이 실패하면 전체 서버 백업에서 복원하십시오.

중요

하드웨어 또는 업그레이드 장애에 대한 기본 솔루션은 복제본에서 손실된 서버를 다시 빌드하는 것입니다. 자세한 내용은 복제를 사용하여 단일 서버 복구를 참조하십시오.

9.8. IDM 백업에서 복원할 때 고려 사항

ipa-backup 유틸리티를 사용하여 백업을 생성한 경우 IdM 서버 또는 LDAP 콘텐츠를 백업이 수행된 상태로 복원할 수 있습니다.

IdM 백업에서 복원하는 동안 고려해야 할 주요 사항은 다음과 같습니다.

- 백업이 원래 생성된 서버의 구성과 일치하는 서버에서만 백업을 복원할 수 있습니다. 서버에는 다음이 있어야 합니다.
  - 동일한 호스트 이름
  - 동일한 IP 주소

- 동일한 버전의 IdM 소프트웨어

- 다수의 IdM 서버 중 하나가 복원되면 복원된 서버는 IdM에 대한 정보의 유일한 소스가 됩니다. 다른 모든 서버는 복원된 서버에서 다시 초기화 해야 합니다.

- 마지막 백업 후에 생성된 모든 데이터는 손실되므로 정상적인 시스템 유지 관리를 위해 백업 및 복원 솔루션을 사용하지 마십시오.

- 서버가 손실되면 백업에서 복원하는 대신 서버를 복제본으로 다시 설치하여 서버를 다시 빌드하는 것이 좋습니다. 새 복제본을 만들면 현재 작업 환경의 데이터가 유지됩니다. 자세한 내용은 복제를 사용하여 서버 손실 준비를 참조하십시오.

- 백업 및 복원 기능은 명령줄에서만 관리할 수 있으며 IdM 웹 UI에서는 사용할 수 없습니다.

- /tmp 또는 /var/tmp 디렉터리에 있는 백업 파일에서 복원할 수 없습니다. IdM 디렉터리 서버는 PrivateTmp 디렉토리를 사용하며 운영 체제에서 일반적으로 사용할 수 있는 /tmp 또는 /var/tmp 디렉터리에 액세스할 수 없습니다.

작은 정보

백업에서 복원하려면 백업을 수행할 때 설치된 것과 동일한 소프트웨어(RPM) 버전이 대상 호스트에 있어야 합니다. 이로 인해 백업 대신 가상 머신 스냅샷에서 복원하는 것이 좋습니다. 자세한 내용은 VM 스냅샷으로 데이터 손실에서 복구를 참조하십시오.

9.9. 백업에서 IDM 서버 복원

다음 절차에서는 IdM 백업에서 IdM 서버 또는 LDAP 데이터 복원을 설명합니다.

그림 9.1. 이 예에서는 사용되는 복제 토폴로지

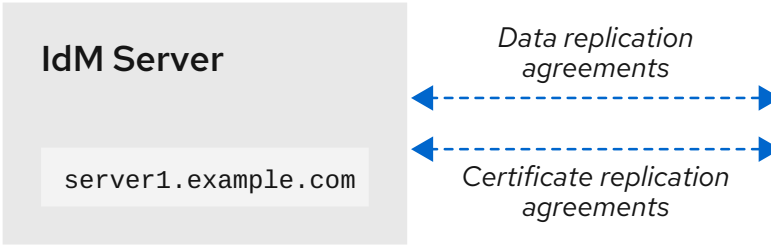


표 9.1. 이 예제에서 사용되는 서버 이름 지정 규칙

서버 호스트 이름	함수
server1.example.com	백업에서 복원해야 하는 서버입니다.
caReplica2.example.com	**server1.example.com** 호스트에 연결된 CA(인증 기관) 복제본.
replica3.example.com	**caReplica2.example.com** 호스트에 연결된 복제본입니다.

사전 요구 사항

- **ipa-backup** 유틸리티를 사용하여 **IdM** 서버의 전체 서버 또는 데이터 전용 백업이 생성되어 있습니다. 백업 생성을 참조하십시오.
- 백업 파일은 **/tmp** 또는 **/var/tmp** 디렉터리에 없습니다.
- 전체 서버 백업에서 전체 서버 복원을 수행하기 전에 서버에서 **IdM**을 제거하고 이전과 동일한 서버 구성을 사용하여 **IdM**을 다시 설치합니다.

절차

1. **ipa-restore** 유틸리티를 사용하여 전체 서버 또는 데이터 전용 백업을 복원합니다.
   - 백업 디렉터리가 기본 **/var/lib/ipa/backup/** 위치에 있는 경우 디렉터리 이름만 입력합니다.

```
[root@server1 ~]# ipa-restore ipa-full-2020-01-14-12-02-32
```

- 백업 디렉터리가 기본 위치에 없는 경우 전체 경로를 입력합니다.**If the backup directory is not in the default location, enter its full path:**

```
[root@server1 ~]# ipa-restore /mybackups/ipa-data-2020-02-01-05-30-00
```

참고

ipa-restore 유틸리티는 디렉터리에 포함된 백업 유형을 자동으로 감지하고 기본적으로 동일한 유형의 복원을 수행합니다. 전체 서버 백업에서 데이터 전용 복원을 수행하려면 **--data** 옵션을 **ipa-restore** 명령에 추가합니다.

```
[root@server1 ~]# ipa-restore --data ipa-full-2020-01-14-12-02-32
```

2. **Directory Manager** 암호를 입력합니다.

```
Directory Manager (existing master) password:
```

3. **yes** 를 입력하여 백업과 함께 현재 데이터를 덮어쓰는지 확인합니다.

```
Preparing restore from /var/lib/ipa/backup/ipa-full-2020-01-14-12-02-32 on
server1.example.com
Performing FULL restore from FULL backup
Temporary setting umask to 022
Restoring data will overwrite existing live data. Continue to restore? [no]: yes
```

4. **ipa-restore** 유틸리티는 사용 가능한 모든 서버에서 복제를 비활성화합니다.

```
Each master will individually need to be re-initialized or
re-created from this one. The replication agreements on
masters running IPA 3.1 or earlier will need to be manually
re-enabled. See the man page for details.
Disabling all replication.
Disabling replication agreement on server1.example.com to caReplica2.example.com
Disabling CA replication agreement on server1.example.com to
caReplica2.example.com
Disabling replication agreement on caReplica2.example.com to server1.example.com
Disabling replication agreement on caReplica2.example.com to replica3.example.com
Disabling CA replication agreement on caReplica2.example.com to
server1.example.com
Disabling replication agreement on replica3.example.com to caReplica2.example.com
```

그런 다음 유틸리티는 **IdM** 서비스를 중지하고 백업을 복원한 후 서비스를 다시 시작합니다.

```
Stopping IPA services
Systemwide CA database updated.
Restoring files
Systemwide CA database updated.
Restoring from userRoot in EXAMPLE-COM
Restoring from ipaca in EXAMPLE-COM
Restarting GSS-proxy
Starting IPA services
Restarting SSSD
Restarting oddjobd
Restoring umask to 18
The ipa-restore command was successful
```

5. 복원된 서버에 연결된 모든 복제본을 다시 초기화합니다.

a. 복원된 서버와 관련된 토폴로지 세그먼트를 기록하여 도메인 접미사의 모든 복제 토폴로지 세그먼트를 나열합니다.

```
[root@server1 ~]# ipa topologysegment-find domain
------------------
2 segments matched
------------------
  Segment name: server1.example.com-to-caReplica2.example.com
  Left node: server1.example.com
  Right node: caReplica2.example.com
  Connectivity: both

  Segment name: caReplica2.example.com-to-replica3.example.com
  Left node: caReplica2.example.com
  Right node: replica3.example.com
  Connectivity: both
----------------------------
Number of entries returned 2
----------------------------
```

b. 복원된 서버가 있는 모든 토폴로지 세그먼트의 도메인 접미사를 다시 초기화합니다.

이 예에서는 **server1** 의 데이터가 있는 **caReplica2** 의 다시 초기화를 수행합니다.

```
[root@caReplica2 ~]# ipa-replica-manage re-initialize --from=server1.example.com
Update in progress, 2 seconds elapsed
Update succeeded
```

c.

인증 기관 데이터로 이동하여 **ca** 접미사의 모든 복제 토폴로지 세그먼트를 나열합니다.

```
[root@server1 ~]# ipa topologysegment-find ca
----------------
1 segment matched
----------------
  Segment name: server1.example.com-to-caReplica2.example.com
  Left node: server1.example.com
  Right node: caReplica2.example.com
  Connectivity: both
----------------------------
Number of entries returned 1
----------------------------
```

d.

복원된 서버에 연결된 모든 **CA** 복제본을 다시 초기화합니다.

이 예에서는 **server1** 의 데이터를 사용하여 **caReplica2** 의 **csreplica** 다시 초기화를 수행합니다.

```
[root@caReplica2 ~]# ipa-csreplica-manage re-initialize --
from=server1.example.com
Directory Manager password:

Update in progress, 3 seconds elapsed
Update succeeded
```

6.

모든 서버가 **server1.example.com** 의 데이터로 업데이트될 때까지 복제 토폴로지를 통해 계속 이동하면서 연속 복제본을 다시 시작합니다.

이 예제에서는 **caReplica2** 의 데이터로 **replica3** 의 도메인 접미사만 다시 초기화해야 합니다.

```
[root@replica3 ~]# ipa-replica-manage re-initialize --from=caReplica2.example.com
Directory Manager password:

Update in progress, 3 seconds elapsed
Update succeeded
```

7.

잘못된 데이터로 인해 인증 문제가 발생하지 않도록 모든 서버에서 **SSSD**의 캐시를 지웁니다.

a.

SSSD 서비스를 중지합니다.

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

-

b. SSSD에서 캐시된 모든 내용을 삭제합니다.

```
[root@server ~]# sss_cache -E
```

c. SSSD 서비스를 시작합니다.

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

d. 서버를 재부팅합니다.

추가 리소스

- ipa-restore (1) 도움말 페이지에서는 복원 중에 복잡한 복제 시나리오를 처리하는 방법에 대해서도 설명합니다.

9.10. 암호화된 백업에서 복원

이 절차에서는 암호화된 IdM 백업에서 IdM 서버를 복원합니다. ipa-restore 유틸리티는 IdM 백업이 암호화되었는지 자동으로 감지하여 GPG2 루트 인증 키를 사용하여 복원합니다.

사전 요구 사항

- GPG 암호화 IdM 백업. 암호화된 IdM 백업 생성을 참조하십시오.
- LDAP Directory Manager 암호
- GPG 키를 만들 때 사용되는 암호

절차

1. GPG2 키를 생성할 때 사용자 정의 인증 키 위치를 사용한 경우 $GNUPGHOME 환경 변수가 해당 디렉터리로 설정되어 있는지 확인합니다. GPG2 키 생성을 참조하십시오.

```
[root@server ~]# echo $GNUPGHOME
/root/backup
```

2. 백업 디렉터리 위치와 함께 **ipa-restore** 유틸리티를 제공합니다.

```
[root@server ~]# ipa-restore ipa-full-2020-01-13-18-30-54
```

a. **Directory Manager** 암호를 입력합니다.

```
Directory Manager (existing master) password:
```

b. **GPG** 키를 만들 때 사용한 암호를 입력합니다.

```
┌──────────────────────────────────────────────────────────────────────┐
│ Please enter the passphrase to unlock the OpenPGP secret key:        │
│ "GPG User (first key) <root@example.com>"                            │
│ 2048-bit RSA key, ID BF28FFA302EF4557,                               │
│ created 2020-01-13.                                                  │
│                                                                      │
│                                                                      │
│ Passphrase: <passphrase>                                             │
│                                                                      │
│       <OK>                          <Cancel>                         │
└──────────────────────────────────────────────────────────────────────┘
```

3. 복원된 서버에 연결된 모든 복제본을 다시 초기화합니다. 백업에서 IdM 서버 복원을 참조하십시오.

10장. ANSIBLE 플레이북을 사용하여 IDM 서버 백업 및 복원

ipabackup Ansible 역할을 사용하여 IdM 서버 백업, 서버와 Ansible 컨트롤러 간에 백업 파일을 전송하고 백업에서 IdM 서버를 복원할 수 있습니다.

10.1. ANSIBLE을 사용하여 IDM 서버 백업 생성

다음 절차에서는 Ansible 플레이북에서 ipabackup 역할을 사용하여 IdM 서버 백업을 생성하고 IdM 서버에 저장하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. ~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible-freeipa/playbooks 디렉터리에 있는 backup-server.yml 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/backup-server.yml backup-my-server.yml
```

3. 편집할 **backup-my-server.yml Ansible** 플레이북 파일을 엽니다.

4. 인벤토리 파일에서 **hosts** 변수를 호스트 그룹으로 설정하여 파일을 조정합니다. 이 예제에서는 **ipaserver** 호스트 그룹으로 설정합니다.

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

  roles:
  - role: ipabackup
    state: present
```

5. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory backup-my-server.yml
```

검증 단계

1. 백업한 **IdM** 서버에 로그인합니다.

2. 백업이 **/var/lib/ipa/backup** 디렉토리에 있는지 확인합니다.

```
[root@server ~]# ls /var/lib/ipa/backup/
ipa-full-2021-04-30-13-12-00
```

추가 리소스

- **ipabackup** 역할을 사용하는 더 샘플 **Ansible** 플레이북은 다음을 참조하십시오.
  - **/usr/share/doc/ansible-freeipa/roles/ipabackup** 디렉터리의 **README.md** 파일입니다.
  - **/usr/share/doc/ansible-freeipa/playbooks/** 디렉토리.

10.2. ANSIBLE을 사용하여 ANSIBLE 컨트롤러에 IDM 서버 백업 생성

다음 절차에서는 **Ansible** 플레이북에서 **ipabackup** 역할을 사용하여 **IdM** 서버 백업을 생성하고 **Ansible** 컨트롤러에 자동으로 전송하는 방법을 설명합니다. 백업 파일 이름은 **IdM** 서버의 호스트 이름으로 시작됩니다.

사전 요구 사항

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

절차

1. 백업을 저장하려면 **Ansible** 컨트롤러의 홈 디렉터리에 하위 디렉터리를 생성합니다.

```
$ mkdir ~/ipabackups
```

2. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

3. **/usr/share/doc/ansible-freeipa/playbooks** 디렉터리에 있는 **backup-server-to-controller.yml** 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/backup-server-to-controller.yml backup-my-server-to-my-controller.yml
```

4. 편집할 **backup-my-server-to-my-controller.yml** 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

   a. 인벤토리 파일에서 **hosts** 변수를 호스트 그룹으로 설정합니다. 이 예에서는 **ipaserver** 호스트 그룹으로 설정합니다.

   b. *(선택 사항)* **IdM** 서버에 백업 사본을 유지하려면 다음 줄의 주석 처리를 해제합니다.

```
# ipabackup_keep_on_server: yes
```

6. 기본적으로 백업은 **Ansible** 컨트롤러의 현재 작업 디렉터리에 저장됩니다. **1**단계에서 만든 백업 디렉터리를 지정하려면 **ipabackup_controller_path** 변수를 추가하고 **/home/user/ipabackups** 디렉터리로 설정합니다.

```
---
- name: Playbook to backup IPA server to controller
  hosts: ipaserver
  become: true
  vars:
    ipabackup_to_controller: yes
    # ipabackup_keep_on_server: yes
    ipabackup_controller_path: /home/user/ipabackups

  roles:
  - role: ipabackup
    state: present
```

7. 파일을 저장합니다.

8. 인벤토리 파일과 플레이북 파일을 지정하여 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory backup-my-server-to-my-
controller.yml
```

검증 단계

- **Ansible** 컨트롤러의 **/home/user/ipabackups** 디렉터리에 백업이 있는지 확인합니다.

```
[user@controller ~]$ ls /home/user/ipabackups
server.idm.example.com_ipa-full-2021-04-30-13-12-00
```

추가 리소스

- ipabackup 역할을 사용하는 더 샘플 Ansible 플레이북은 다음을 참조하십시오.
  - /usr/share/doc/ansible-freeipa/roles/ipabackup 디렉터리의 README.md 파일입니다.
  - /usr/share/doc/ansible-freeipa/playbooks/ 디렉토리.

10.3. ANSIBLE을 사용하여 ANSIBLE 컨트롤러에 IDM 서버 백업 복사

다음 절차에서는 Ansible 플레이북을 사용하여 IdM 서버에서 Ansible 컨트롤러로 IdM 서버 백업을 복사하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. 백업을 저장하려면 Ansible 컨트롤러의 홈 디렉터리에 하위 디렉터리를 생성합니다.

```
$ mkdir ~/ipabackups
```

-

2. ~/MyPlaybooks/ 디렉터리로 이동합니다.

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

3. /usr/share/doc/ansible-freeipa/playbooks 디렉터리에 있는 copy-backup-from-server.yml 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/copy-backup-from-server.yml copy-backup-from-my-server-to-my-controller.yml
```

4. 편집을 위해 copy-my-backup-from-my-server-to-my-controller.yml 파일을 엽니다.

5. 다음 변수를 설정하여 파일을 조정합니다.

a. 인벤토리 파일에서 hosts 변수를 호스트 그룹으로 설정합니다. 이 예에서는 ipaserver 호스트 그룹으로 설정합니다.

b. ipabackup_name 변수를 IdM 서버의 ipabackup 이름으로 설정하여 Ansible 컨트롤러에 복사합니다.

c. 기본적으로 백업은 Ansible 컨트롤러의 현재 작업 디렉터리에 저장됩니다. 1단계에서 만든 디렉터리를 지정하려면 ipabackup_controller_path 변수를 추가하고 /home/user/ipabackups 디렉터리로 설정합니다.

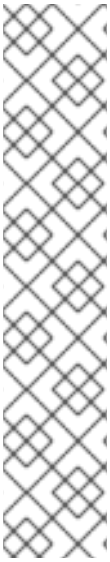
```
---
- name: Playbook to copy backup from IPA server
  hosts: ipaserver
  become: true
  vars:
    ipabackup_name: ipa-full-2021-04-30-13-12-00
    ipabackup_to_controller: yes
    ipabackup_controller_path: /home/user/ipabackups

  roles:
  - role: ipabackup
    state: present
```

6. 파일을 저장합니다.

7. 인벤토리 파일과 플레이북 파일을 지정하여 **Ansible** 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory copy-backup-from-my-server-to-my-controller.yml
```



참고

모든 **IdM** 백업을 컨트롤러에 복사하려면 **Ansible** 플레이북의 **ipabackup_name** 변수를 **all** 로 설정합니다.

```
vars:
  ipabackup_name: all
  ipabackup_to_controller: yes
```

예를 들어 **/usr/share/doc/ansible-freeipa/playbooks** 디렉터리의 **copy-all-backups-from-server.yml Ansible** 플레이북을 참조하십시오.

검증 단계

- **Ansible** 컨트롤러의 **/home/user/ipabackups** 디렉터리에 백업이 있는지 확인합니다.

```
[user@controller ~]$ ls /home/user/ipabackups
server.idm.example.com_ipa-full-2021-04-30-13-12-00
```

추가 리소스

- **/usr/share/doc/ansible-freeipa/roles/ipabackup** 디렉터리의 **README.md** 파일입니다.
- **/usr/share/doc/ansible-freeipa/playbooks/** 디렉토리.

10.4. ANSIBLE을 사용하여 ANSIBLE 컨트롤러에서 IDM 서버로의 IDM 서버 백업 복사

다음 절차에서는 **Ansible** 플레이북을 사용하여 **Ansible** 컨트롤러에서 **IdM** 서버로의 **IdM** 서버 백업을 복사하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. **~/MyPlaybooks/** 디렉터리로 이동합니다.

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

2. **/usr/share/doc/ansible-freeipa/playbooks** 디렉터리에 있는 **copy-backup-from-controller.yml** 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/copy-backup-from-controller.yml copy-backup-from-my-controller-to-my-server.yml
```

3. 편집을 위해 **copy-my-backup-from-my-controller-to-my-server.yml** 파일을 엽니다.

4. 다음 변수를 설정하여 파일을 조정합니다.

   a. 인벤토리 파일에서 **hosts** 변수를 호스트 그룹으로 설정합니다. 이 예에서는 **ipaserver** 호스트 그룹으로 설정합니다.

   b. **ipabackup_name** 변수를 **Ansible** 컨트롤러의 **ipabackup** 이름으로 설정하여 **IdM** 서버에 복사합니다.

```
---
```

```
- name: Playbook to copy a backup from controller to the IPA server
  hosts: ipaserver
  become: true

  vars:
    ipabackup_name: server.idm.example.com_ipa-full-2021-04-30-13-12-00
    ipabackup_from_controller: yes

  roles:
  - role: ipabackup
    state: copied
```

5. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory copy-backup-from-my-controller-to-my-server.yml
```

추가 리소스

- **/usr/share/doc/ansible-freeipa/roles/ipabackup** 디렉터리의 **README.md** 파일입니다.
- **/usr/share/doc/ansible-freeipa/playbooks/** 디렉토리.

10.5. ANSIBLE을 사용하여 IDM 서버에서 백업 제거

다음 절차에서는 **Ansible** 플레이북을 사용하여 **IdM** 서버에서 백업을 제거하는 방법을 설명합니다.

사전 요구 사항

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

인벤토리 파일을 생성했습니다.

- Ansible 인벤토리 파일은 ~/MyPlaybooks/ 디렉터리에 있습니다.

절차

1. ~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible-freeipa/playbooks 디렉터리에 있는 remove-backup-from-server.yml 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/remove-backup-from-server.yml
remove-backup-from-my-server.yml
```

3. 편집할 remove-backup-from-my-server.yml 파일을 엽니다.

4. 다음 변수를 설정하여 파일을 조정합니다.

a. 인벤토리 파일에서 hosts 변수를 호스트 그룹으로 설정합니다. 이 예에서는 ipaserver 호스트 그룹으로 설정합니다.

b. ipabackup_name 변수를 IdM 서버에서 제거하려면 ipabackup 의 이름으로 설정합니다.

```
---
- name: Playbook to remove backup from IPA server
  hosts: ipaserver
  become: true

  vars:
    ipabackup_name: ipa-full-2021-04-30-13-12-00

  roles:
  - role: ipabackup
    state: absent
```

5. 파일을 저장합니다.

6. 인벤토리 파일과 플레이북 파일을 지정하여 Ansible 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory remove-backup-from-my-server.yml
```

참고

IdM 서버에서 모든 IdM 백업을 제거하려면 Ansible 플레이북의 ipabackup_name 변수를 all 로 설정합니다.

```
vars:
  ipabackup_name: all
```

예를 들어 /usr/share/doc/ansible-freeipa/playbooks 디렉터리의 remove-all-backups-from-server.yml Ansible 플레이북을 참조하십시오.

추가 리소스

- /usr/share/doc/ansible-freeipa/roles/ipabackup 디렉터리의 README.md 파일입니다.
- /usr/share/doc/ansible-freeipa/playbooks/ 디렉토리.

10.6. ANSIBLE을 사용하여 서버에 저장된 백업에서 IDM 서버 복원

다음 절차에서는 Ansible 플레이북을 사용하여 해당 호스트에 저장된 백업에서 IdM 서버를 복원하는 방법을 설명합니다.

사전 요구 사항

- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.

- ansible-freeipa 패키지를 설치했습니다.
- 이러한 옵션을 구성하는 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.
- Ansible 인벤토리 파일은 ~/MyPlaybooks/ 디렉터리에 있습니다.

- LDAP Directory Manager 암호를 알고 있습니다.

절차

1. ~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible-freeipa/playbooks 디렉터리에 있는 restore-server.yml 파일의 사본을 만듭니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/restore-server.yml restore-my-server.yml
```

3. 편집할 restore-my-server.yml Ansible 플레이북 파일을 엽니다.

4. 다음 변수를 설정하여 파일을 조정합니다.

a. 인벤토리 파일에서 hosts 변수를 호스트 그룹으로 설정합니다. 이 예에서는 ipaserver 호스트 그룹으로 설정합니다.

b. ipabackup_name 변수를 restore로 ipabackup 의 이름으로 설정합니다.

c. ipabackup_password 변수를 LDAP Directory Manager 암호로 설정합니다.

```
---
- name: Playbook to restore an IPA server
```

```
hosts: ipaserver
become: true

vars:
 ipabackup_name: ipa-full-2021-04-30-13-12-00
 ipabackup_password: <your_LDAP_DM_password>

roles:
- role: ipabackup
 state: restored
```

5. 파일을 저장합니다.

6. 인벤토리 파일과 플레이북 파일을 지정하는 Ansible 플레이북을 실행합니다.

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory restore-my-server.yml
```

추가 리소스

- /usr/share/doc/ansible-freeipa/roles/ipabackup 디렉터리의 README.md 파일입니다.
- /usr/share/doc/ansible-freeipa/playbooks/ 디렉토리.

10.7. ANSIBLE을 사용하여 ANSIBLE 컨트롤러에 저장된 백업에서 IDM 서버 복원

다음 절차에서는 Ansible 플레이북을 사용하여 Ansible 컨트롤러에 저장된 백업에서 IdM 서버를 복원하는 방법을 설명합니다.

사전 요구 사항

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

- Ansible 인벤토리 파일은 ~/MyPlaybooks/ 디렉터리에 있습니다.

- LDAP Directory Manager 암호를 알고 있습니다.

절차

1. ~/MyPlaybooks/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible-freeipa/playbooks 디렉터리에 있는 restore-server-from-controller.yml 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/restore-server-from-controller.yml
restore-my-server-from-my-controller.yml
```

3. 편집할 restore-my-server-from-my-controller.yml 파일을 엽니다.

4. 다음 변수를 설정하여 파일을 조정합니다.

   a. 인벤토리 파일에서 hosts 변수를 호스트 그룹으로 설정합니다. 이 예에서는 ipaserver 호스트 그룹으로 설정합니다.

   b. ipabackup_name 변수를 restore로 ipabackup 의 이름으로 설정합니다.

   c. ipabackup_password 변수를 LDAP Directory Manager 암호로 설정합니다.

```
---
- name: Playbook to restore IPA server from controller
  hosts: ipaserver
  become: true

  vars:
    ipabackup_name: server.idm.example.com_ipa-full-2021-04-30-13-12-00
    ipabackup_password: <your_LDAP_DM_password>
```

```
  ipabackup_from_controller: yes

roles:
- role: ipabackup
  state: restored
```

5. 파일을 저장합니다.

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

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory restore-my-server-from-my-controller.yml
```

추가 리소스

- **/usr/share/doc/ansible-freeipa/roles/ipabackup** 디렉터리의 **README.md** 파일입니다.
- **/usr/share/doc/ansible-freeipa/playbooks/** 디렉토리.