Red Hat Enterprise Linux 8

RHEL 8에서 Identity Management로 마이그레이션

RHEL 8에서 실행되도록 IdM 서버 마이그레이션 및 IdM 클라이언트 업그레이드 가이드

Last Updated: 2022-09-04

Red Hat Enterprise Linux 8 RHEL 8에서 Identity Management로 마이그레이션

RHEL 8에서 실행되도록 IdM 서버 마이그레이션 및 IdM 클라이언트 업그레이드 가이드

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 설명서 컬렉션은 Red Hat Enterprise Linux 7 설치, 기타 LDAP 솔루션 또는 기타 Linux 배포판에서 RHEL(Red Hat Enterprise Linux 8) 서버로 마이그레이션하는 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.

ID 관리에서 계획된 용어 교체는 다음과 같습니다.

- ***차단 목록*** *블랙리스트*대체
- ***허용 목록*** 대체 *허용 목록*
- ***슬레이브*** 교체
- *마스터* 라는 단어는 컨텍스트에 따라 보다 정확한 언어로 교체되고 있습니다.
  - ***IdM 서버가*** *IdM 마스터*교체
  - ***CA 갱신 마스터를 대체하는CA 갱신서버***
  - ***CRL 게시자 서버는*** *CRL 마스터*교체
  - ***다중*** *마스터*교체

RED HAT 문서에 관한 피드백 제공

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

- 특정 문구에 대한 간단한 의견 작성 방법은 다음과 같습니다.
  1. 문서가 *Multi-page HTML* 형식으로 표시되는지 확인합니다. 또한 문서 오른쪽 상단에 **피드백** 버튼이 있는지 확인합니다.
  2. 마우스 커서를 사용하여 주석 처리하려는 텍스트 부분을 강조 표시합니다.
  3. 강조 표시된 텍스트 아래에 표시되는 **피드백 추가** 팝업을 클릭합니다.
  4. 표시된 지침을 따릅니다.
- Bugzilla를 통해 피드백을 제출하려면 새 티켓을 생성합니다.
  1. Bugzilla 웹 사이트로 이동하십시오.
  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.

I 부. RHEL 7에서 RHEL 8로 IDM 마이그레이션

1장. RHEL 7 서버에서 RHEL 8 서버로 IDM 환경 마이그레이션

RHEL 7 IdM 환경을 RHEL 8로 업그레이드하려면 먼저 RHEL 7 IdM 환경에 새 RHEL 8 IdM 복제본을 추가한 다음 RHEL 7 서버를 폐기해야 합니다.

주의

RHEL 7 IdM 서버를 RHEL 8으로의 전체 업그레이드는 지원되지 않습니다.

이 섹션에서는 모든 IdM(Identity Management) 데이터 및 구성을 RHEL(Red Hat Enterprise Linux) 7 서버에서 RHEL 8 서버로 **마이그레이션** 하는 방법을 설명합니다. 이 절차를 사용하여 RHEL이 아닌 Linux 배포판의 FreeIPA 서버에서 RHEL 8 서버의 IdM으로 마이그레이션할 수도 있습니다.

마이그레이션 절차에는 다음이 포함됩니다.

1. RHEL 8 IdM 서버를 구성하고 현재 RHEL 7 IdM 환경에 복제본으로 추가합니다. 자세한 내용은 RHEL 8 Replica 설치를 참조하십시오.
2. RHEL 8 서버를 CA(인증 기관) 갱신 서버로 설정합니다. 자세한 내용은 RHEL 8 IdM 서버에 CA 갱신 서버 역할 할당을 참조하십시오.
3. RHEL 7 서버에서 CRL(인증서 취소 목록) 생성을 중지하고 CRL 요청을 RHEL 8로 리디렉션합니다. 자세한 내용은 RHEL 7 IdM CA 서버에서 CRL 생성 중지 를 참조하십시오.
4. RHEL 8 서버에서 CRL 생성을 시작합니다. 자세한 내용은 새로운 RHEL 8 IdM CA 서버에서 CRL 생성 시작을 참조하십시오.
5. 원본 RHEL 7 CA 업데이트 서버 중지 및 해제. 자세한 내용은 RHEL 7 서버 중지 및 해제를 참조하십시오.

다음 절차에서는 다음을 수행합니다.

- **rhel8.example.com** 은 새 CA 갱신 서버가 될 RHEL 8 시스템입니다.
- **rhel7.example.com** 은 원래 RHEL 7 CA 갱신 서버입니다. CA 갱신 서버인 Red Hat Enterprise Linux 7 서버를 식별하려면 IdM 서버에서 다음 명령을 실행합니다.

```
[root@rhel7 ~]# ipa config-show | grep "CA renewal"
IPA CA renewal master: rhel7.example.com
```

IdM 배포에서 CA(인증 기관)를 사용하지 않는 경우 RHEL 7에서 실행 중인 모든 IdM 서버는 **rhel7.example.com** 일 수 있습니다.

참고

IdM 배포에서 포함된 CA를 사용하는 **경우에만** 다음 섹션의 단계를 완료합니다.

- RHEL 8 IdM 서버에 CA 갱신 서버 역할 할당
- RHEL 7 IdM CA 서버에서 CRL 생성 중지
- 새로운 RHEL 8 IdM CA 서버에서 CRL 생성 시작

1.1. RHEL 7에서 8로 IDM을 마이그레이션하기 위한 사전 요구 사항

rhel7.example.com 에서 다음을 수행합니다.

1. 시스템을 최신 RHEL 7 버전으로 업그레이드합니다.
2. 도메인의 도메인 수준이 1로 설정되어 있는지 확인합니다. 자세한 내용은 RHEL 7용 *Linux 도메인 ID, 인증 및 정책 가이드 표시 및 해제* 를 참조하십시오.
3. **ipa-*** 패키지를 최신 버전으로 업데이트합니다.

```
[root@rhel7 ~]# yum update ipa-*
```

주의

IdM(Identity Management) 서버를 여러 개 업그레이드하는 경우 각 업그레이드 사이에 최소 10분 정도 기다립니다.

두 개 이상의 서버를 동시에 업그레이드하거나 업그레이드 간의 짧은 간격만 사용하여 토폴로지 전체에서 업그레이드 후 데이터 변경 사항을 복제하는 데 시간이 충분하지 않으므로 복제 이벤트가 충돌할 수 있습니다.

rhel8.example.com 에서 다음을 수행합니다.

1. 최신 버전의 Red Hat Enterprise Linux가 시스템에 설치되어 있습니다. 자세한 내용은 표준 RHEL 8 설치 수행을 참조하십시오.
2. 시스템이 **rhel7.example.com** IdM 서버에 권한이 있는 도메인에 등록된 IdM 클라이언트인지 확인합니다. 자세한 내용은 IdM 클라이언트 설치: 기본 시나리오를 참조하십시오.
3. 시스템이 IdM 서버 설치에 대한 요구 사항을 충족하는지 확인합니다. IdM 서버 설치 준비를 참조하십시오.
4. 시간 서버 **rhel7.example.com** 이 동기화되었는지 확인합니다.

```
[root@rhel7 ~]# ntpstat
synchronised to NTP server (ntp.example.com) at stratum 3
   time correct to within 42 ms
   polling server every 1024 s
```

중요

RHEL 8에서는 IdM이 자체 시간 서버를 제공하지 않습니다. **rhel8.example.com** 에 IdM을 설치하면 호스트에 NTP 서버가 설치되지 않습니다. 따라서 별도의 NTP 서버(예: **ntp.example.com**)를 사용해야 합니다. 자세한 내용은 IdM 에 대한 chrony 및 시간 서비스 요구 사항으로 마이그레이션 을 참조하십시오.

rhel7.example.com 을 NTP 서버 역할에서 사용할 수 있지만 마이그레이션 프로세스의 일부로 서버를 해제할 수 있습니다. 따라서 **rhel8.example.com** 을 대신 **ntp.example.com** 과 직접 동기화해야 합니다. 설치 프로세스 중에 이를 지정할 수 있습니다.

5. IdM 복제본 설치에 대해 시스템이 승인되었는지 확인합니다. IdM 클라이언트의 복제본 설치 인증 을 참조하십시오.

6. **ipa-*** 패키지를 최신 버전으로 업데이트합니다.

```
[root@rhel7 ~]# yum update ipa-*
```

추가 리소스

- CA 서비스 계획
- DNS 서비스 및 호스트 이름 계획
- 간접 통합
- IdM 서버에 필요한 패키지 설치 .
- RHEL 7에서 RHEL 8로 업그레이드 .

1.2. RHEL 8 복제본 설치

1. RHEL 7 환경에 있는 서버 역할을 나열합니다.

```
[root@rhel7 ~]# ipa server-role-find --status enabled --server rhel7.example.com
----------------------
3 server roles matched
----------------------
  Server name: rhel7.example.com
  Role name: CA server
  Role status: enabled

  Server name: rhel7.example.com
  Role name: DNS server
  Role status: enabled

  Server name: rhel7.example.com
  Role name: NTP server
  Role status: enabled
[... output truncated ...]
```

2. [선택 사항] **rhel7.example.com** 에서 사용 중인 **rhel8.example.com** 에 대해 동일한 서버당 전달자를 사용하려면 **rhel7.example.com** 에 대한 서버별 전달자를 확인합니다.

```
[root@rhel7 ~]# ipa dnsserver-show rhel7.example.com
----------------------------
1 DNS server matched
----------------------------
  Server name: rhel7.example.com
  SOA mname: rhel7.example.com.
  Forwarders: 192.0.2.20
  Forward policy: only
----------------------------------------------------
Number of entries returned 1
----------------------------------------------------
```

3. NTP 서버 역할을 제외한 **rhel7.example.com** 에 있는 모든 서버 역할을 포함하여 IdM RHEL 7 서버의 복제본으로 IdM 서버를 설치합니다. 위 예제에서 역할을 설치하려면 **ipa-replica-install** 명령과 함께 이러한 옵션을 사용하십시오.

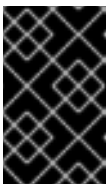
   - **--setup-ca** 에서 Certificate System 구성 요소를 설정
   - **--setup-dns** 및 **--forwarder** 가 통합된 DNS 서버를 구성하고 IdM 도메인 외부에 이동하는 DNS 쿼리를 처리하도록 서버별 전달자를 설정합니다.

   참고

   또한 IdM 배포가 Active Directory(AD)와의 신뢰 관계에 있는 경우 **--setup-adtrust** 옵션을 **ipa-replica-install** 명령에 추가하여 **rhel8.example.com** 에서 AD 신뢰 기능을 구성합니다.

   - NTP 서버 풀을 지정할 NTP 서버 또는 **--ntp-pool** 을 지정하는 --NTP- **server**

   중요

   NTP 서버를 수동으로 지정하지 않으면 DNS 레코드에서 자동으로 설정됩니다. 이로 인해 **rhel8.example.com** 이 **rhel7.example.com** 과 동기화될 수 있습니다. 이로 인해 RHEL 7 서버가 해제될 때 문제가 발생합니다.

   IP 주소가 192.0.2.20인 서버별 전달자를 사용하고 **ntp.example.com** NTP 서버와 동기화되는 IP 주소가 192.0.2.1인 IdM 서버를 설정하려면 다음을 실행합니다.

   ```
   [root@rhel8 ~]# ipa-replica-install --setup-ca --ip-address 192.0.2.1 --setup-dns --
   forwarder 192.0.2.20 --ntp-server ntp.example.com
   ```

   DNS가 올바르게 작동하는 경우 **rhel8.example.com** 에서 DNS 자동 검색을 사용하여 이를 찾으므로 RHEL 7 IdM 서버 자체를 지정할 필요가 없습니다.

4. [선택 사항] 새로 설치한 IdM 서버의 DNS에 외부 **NTP** 시간 서버의 **_ntp._udp** 서비스(SRV) 레코드를 **rhel8.example.com**. RHEL 8의 IdM은 자체 시간 서비스를 제공하지 않기 때문에 이 작업을 수행하는 것이 좋습니다. IdM DNS에 시간 서버에 대한 SRV 레코드가 있으면 향후 RHEL 8 복제본 및 클라이언트 설치가 **rhel8.example.com** 에서 사용하는 시간 서버와 동기화되도록 자동으로 구성됩니다. 이는 설치 CLI(명령줄 인터페이스)에 **--ntp-server** 또는 **--ntp-pool** 옵션이 제공되지 않는 한 **ipa-client-install** 이 **_ntp._udp** DNS 항목을 찾기 때문입니다.

검증

1. IdM 서비스가 **rhel8.example.com** 에서 실행 중인지 확인합니다.

```
[root@rhel8 ~]# ipactl status
Directory Service: RUNNING
[... output truncated ...]
ipa: INFO: The ipactl command was successful
```

2. **rhel8.example.com** 의 서버 역할이 NTP 서버 역할을 제외한 **rhel7.example.com** 과 동일한지 확인합니다.

```
[root@rhel8 ~]$ kinit admin
[root@rhel8 ~]$ ipa server-role-find --status enabled --server rhel8.example.com
----------------------
2 server roles matched
----------------------
  Server name: rhel8.example.com
  Role name: CA server
  Role status: enabled

  Server name: rhel8.example.com
  Role name: DNS server
  Role status: enabled
```

3. [선택 사항] **rhel7.example.com** 과 **rhel8.example.com** 간의 복제 계약에 대한 세부 정보를 표시합니다.

```
[root@rhel8 ~]# ipa-csreplica-manage list --verbose rhel8.example.com
Directory Manager password:

rhel7.example.com
last init status: None
last init ended: 1970-01-01 00:00:00+00:00
last update status: Error (0) Replica acquired successfully: Incremental update succeeded
last update ended: 2019-02-13 13:55:13+00:00
```

4. [선택 사항] IdM 배포가 AD와의 신뢰 관계에 있는 경우 작동하는지 확인하십시오.

   a. Kerberos 구성 확인

   b. **rhel8.example.com** 에서 AD 사용자를 확인하려고 합니다.

```
[root@rhel8 ~]# id aduser@ad.domain
```

5. **rhel8.example.com** 이 **NTP** 서버와 동기화되었는지 확인합니다.

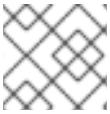
```
[root@rhel8 ~]# chronyc tracking
Reference ID    : CB00710F (ntp.example.com)
Stratum         : 3
Ref time (UTC)  : Tue Nov 16 09:49:17 2021
[... output truncated ...]
```

추가 리소스

- DNS 구성 우선순위
- IdM의 시간 서비스 요구 사항

- chrony로 마이그레이션

1.3. RHEL 8 IDM 서버에 CA 갱신 서버 역할 할당



참고

IdM 배포에서 포함된 인증 기관(CA)을 사용하는 경우에만 이 섹션의 단계를 완료합니다.

rhel8.example.com 에서 **rhel8.example.com** 을 새 CA 갱신 서버로 구성합니다.

1. CA 하위 시스템 인증서 업데이트를 처리하도록 **rhel8.example.com** 을 구성합니다.

```
[root@rhel8 ~]# ipa config-mod --ca-renewal-master-server rhel8.example.com
...
IPA masters: rhel7.example.com, rhel8.example.com
IPA CA servers: rhel7.example.com, rhel8.example.com
IPA NTP servers: rhel7.example.com, rhel8.example.com
IPA CA renewal master: rhel8.example.com
```

   출력에 업데이트가 성공했는지 확인합니다.

2. **rhel8.example.com** 에서 인증서 업데이트r 작업을 활성화합니다.
   a. 편집을 위해 **/etc/pki/pki-tomcat/ca/CS.cfg** 구성 파일을 엽니다.
   b. **ca.certStatusUpdateInterval** 항목을 제거하거나 원하는 간격을 초 단위로 설정합니다. 기본값은 **600** 입니다.
   c. **/etc/pki/pki-tomcat/ca/CS.cfg** 구성 파일을 저장하고 종료합니다.
   d. IdM 서비스를 다시 시작하십시오.

```
[user@rhel8 ~]$ ipactl restart
```

3. **rhel7.example.com** 에서 인증서 업데이트r 작업을 비활성화합니다.
   a. 편집을 위해 **/etc/pki/pki-tomcat/ca/CS.cfg** 구성 파일을 엽니다.
   b. **ca.certStatusUpdateInterval** 을 **0** 으로 변경하거나 존재하지 않는 경우 다음 항목을 추가합니다.

```
ca.certStatusUpdateInterval=0
```

   c. **/etc/pki/pki-tomcat/ca/CS.cfg** 구성 파일을 저장하고 종료합니다.
   d. IdM 서비스를 다시 시작하십시오.

```
[user@rhel7 ~]$ ipactl restart
```

1.4. RHEL 7 IDM CA 서버에서 CRL 생성 중지

참고

IdM 배포에서 포함된 인증 기관(CA)을 사용하는 경우에만 이 섹션의 단계를 완료합니다.

이 섹션에서는 **ipa-crlgen-manage** 명령을 사용하여 **rhel7.example.com** CA 서버에서 CRL(인증서 취소 목록) 생성을 중지하는 방법을 설명합니다.

사전 요구 사항

- root로 로그인해야 합니다.

절차

1. 선택적으로 **rhel7.example.com** 이 CRL을 생성하는지 확인합니다.

```
[root@rhel7 ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

2. **rhel7.example.com** 서버에서 CRL 생성을 중지합니다.

```
[root@rhel7 ~]# ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL generation on another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

3. 선택적으로 **rhel7.example.com** 서버가 CRL 생성을 중지했는지 확인합니다.

```
[root@rhel7 ~]# ipa-crlgen-manage status
```

rhel7.example.com 서버에서 CRL 생성을 중지했습니다. 다음 단계는 **rhel8.example.com** 에서 CRL을 생성할 수 있도록 하는 것입니다.

1.5. 새로운 RHEL 8 IDM CA 서버에서 CRL 생성 시작

참고

IdM 배포에서 포함된 인증 기관(CA)을 사용하는 경우에만 이 섹션의 단계를 완료합니다.

사전 요구 사항

- **rhel8.example.com** 시스템에서 root로 로그인해야 합니다.

절차

1. **rhel8.example.com** 에서 CRL 생성을 시작하려면 **ipa-crlgen-manage enable** 명령을 사용합니다.

```
[root@rhel8 ~]# ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
CRL generation enabled on the local host. Please make sure to have only a single CRL
generation master.
The ipa-crlgen-manage command was successful
```

2. CRL 생성이 활성화되어 있는지 확인하려면 **ipa-crlgen-manage status** 명령을 사용합니다.

```
[root@rhel8 ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:10:00
Last CRL Number: 7
The ipa-crlgen-manage command was successful
```

1.6. RHEL 7 서버 중지 및 해제

1. 최신 변경 사항을 포함한 모든 데이터가 **rhel7.example.com** 에서 **rhel8.example.com** 으로 올바르게 마이그레이션되었는지 확인합니다. 예를 들면 다음과 같습니다.

   a. **rhel7.example.com** 에 새 사용자를 추가합니다.

```
[root@rhel7 ~]# ipa user-add random_user
First name: random
Last name: user
```

   b. 사용자가 **rhel8.example.com** 에 복제되었는지 확인합니다.

```
[root@rhel8 ~]# ipa user-find random_user
--------------
1 user matched
--------------
  User login: random_user
  First name: random
  Last name: user
```

2. **rhel7.example.com** 에서 모든 IdM 서비스를 중지하여 도메인 검색을 새 **rhel8.example.com** 서버로 강제 적용합니다.

```
[root@rhel7 ~]# ipactl stop
Stopping CA Service
Stopping pki-ca:                                        [  OK  ]
Stopping HTTP Service
Stopping httpd:                                         [  OK  ]
Stopping MEMCACHE Service
Stopping ipa_memcached:                                 [  OK  ]
Stopping DNS Service
```

```
Stopping named: .                                  [  OK  ]
Stopping KPASSWD Service
Stopping Kerberos 5 Admin Server:                  [  OK  ]
Stopping KDC Service
Stopping Kerberos 5 KDC:                           [  OK  ]
Stopping Directory Service
Shutting down dirsrv:
    EXAMPLE-COM...                                 [  OK  ]
    PKI-IPA...                                     [  OK  ]
```

이 후 **ipa** 유틸리티는 원격 프로시저 호출(RPC)을 통해 새 서버에 연결합니다.

3. RHEL 8 서버에서 제거 명령을 실행하여 토폴로지에서 RHEL 7 서버를 제거합니다. 자세한 내용은 IdM 서버 설치 제거를 참조하십시오.

2장. IDM 클라이언트를 RHEL 7에서 RHEL 8로 업그레이드

IdM 서버와 달리 RHEL 7에서 RHEL 8로 IdM 클라이언트의 인플레이스 업그레이드를 지원합니다.

RHEL 8에서는 일부 일반 옵션과 사용되지 않는 기능이 IdM 환경에서 인증을 담당하는 SSSD(System Security Services Daemon)에서 제거되었습니다. 이러한 옵션을 제거하는 단계는 다음 섹션을 참조하십시오.

- RHEL 8로 업그레이드한 후 SSSD 설정 업데이트
- RHEL 8에서 제거된 SSSD 기능 목록

2.1. RHEL 8로 업그레이드한 후 SSSD 설정 업데이트

RHEL (Red Hat Enterprise Linux) 7에서 RHEL 8로 IdM(Identity Management) 클라이언트를 업그레이드한 후에는 몇 가지 SSSD 구성 옵션이 더 이상 지원되지 않음에 대한 경고가 표시될 수 있습니다.

다음 절차에서는 이러한 문제를 해결하기 위해 SSSD 구성을 업데이트하는 방법을 설명합니다.

사전 요구 사항

- **IdM 클라이언트를 RHEL 7에서 RHEL 8로 업그레이드했습니다.**
- **/etc/sssd/sssd.conf 를 편집할 수 있는 루트 권한이 있습니다.**

2.1.1. 로컬 ID 공급자에서 파일 ID 공급자로 전환

다음 오류가 표시되면 로컬 ID 공급자를 파일 ID 공급자로 바꿉니다.

```
SSSD Domain "example.com": local provider is no longer supported and the domain will be ignored.
Local provider is no longer supported.
```

절차

1. **로컬 ID 공급자로 검색한 사용자 및 그룹도 /etc/passwd 및 /etc/group 파일에 있는지 확인합니다. 이렇게 하면 파일 공급자가 해당 사용자 및 그룹에 액세스할 수 있습니다.**

   a. **사용자를 생성해야 하는 경우 useradd 명령을 사용합니다. UID를 지정해야 하는 경우 -u 옵션을 추가합니다.**

```
[root@client ~]# useradd -u 3001 username
```

b. 그룹을 생성해야 하는 경우 **groupadd** 명령을 사용합니다. **GID**를 지정해야 하는 경우 **-g** 옵션을 추가합니다.

```
[root@client ~]# groupadd -g 5001 groupname
```

2. 텍스트 편집기에서 **/etc/sssd/sssd.conf** 구성 파일을 엽니다.

3. **id_provider=local** 을 **id_provider=files** 로 바꿉니다.

```
[domain/example.com]
id_provider = files
...
```

4. **/etc/sssd/sssd.conf** 설정 파일을 저장합니다.

5. **SSSD**를 다시 시작하여 구성 변경 사항을 로드합니다.

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

2.1.2. 더 이상 사용되지 않는 옵션 제거

더 이상 사용되지 않는 옵션과 관련된 다음 오류 중 하나가 표시되면 **/etc/sssd/sssd.conf** 구성 파일에서 해당 옵션을 제거하는 것이 좋습니다.

```
SSSD Domain "example.com": option ldap_groups_use_matching_rule_in_chain has no longer any effect
Option ldap_groups_use_matching_rule_in_chain was removed and it will be ignored.
```

```
SSSD Domain "example.com": option ldap_initgroups_use_matching_rule_in_chain has no longer any effect
Option ldap_initgroups_use_matching_rule_in_chain was removed and it will be ignored.
```

절차

1. 텍스트 편집기에서 **/etc/sssd/sssd.conf** 구성 파일을 엽니다.

2.

ldap_groups_use_matching_rule_in_chain 또는 ldap_initgroups_use_matching_rule_in_chain 옵션을 제거합니다.

3. /etc/sssd/sssd.conf 설정 파일을 저장합니다.

4. SSSD를 다시 시작하여 구성 변경 사항을 로드합니다.

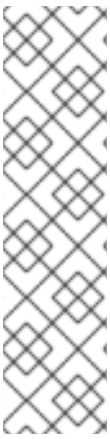
```
[root@client ~]# systemctl restart sssd
```

2.1.3. sudo 규칙에 와일드카드 일치 활성화

다음 경고는 ldap_sudo_include_regexp 옵션이 기본적으로 false 로 설정되므로 와일드카드가 있는 sudo 규칙이 RHEL 8에서 기본적으로 작동하지 않음을 나타냅니다.

```
SSSD Domain "example.com": sudo rules containing wildcards will stop working.
Default value of ldap_sudo_include_regexp changed from true to false for performance
reason.
```

와일드카드와 함께 sudo 규칙을 사용하고 와일드카드 일치를 활성화하려면 ldap_sudo_include_regexp 옵션을 true 로 수동으로 설정합니다.



참고

sudo 규칙과 일치하도록 와일드카드를 사용하지 않는 것이 좋습니다.

ldap_sudo_include_regexp 옵션이 true 로 설정된 경우 SSSD는 sudo Host 특성에 와일드카드가 포함된 모든 sudo 규칙을 다운로드하여 LDAP 검색 성능에 부정적인 영향을 미칩니다.

절차

1. 텍스트 편집기에서 /etc/sssd/sssd.conf 구성 파일을 엽니다.

2. example.com 도메인에서 ldap_sudo_include_regexp=true 를 설정합니다.

```
[domain/example.com]
```

```
...
ldap_sudo_include_regexp = true
...
```

3. /etc/sssd/sssd.conf 설정 파일을 저장합니다.

4. SSSD를 다시 시작하여 구성 변경 사항을 로드합니다.

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

2.2. RHEL 8에서 제거된 SSSD 기능 목록

RHEL 8에서는 다음과 같은 SSSD 기능이 제거되었습니다.

로컬 ID 공급자가 제거됨

로컬 SSSD 캐시의 사용자 정보를 제공하는 데 사용되는 로컬 ID 공급자는 RHEL 7에서 더 이상 사용되지 않으며 RHEL 8에서 더 이상 지원되지 않습니다. /etc/sssd/sssd.conf 설정에 id_provider=local 이 있는 도메인이 있는 경우 SSSD는 이 도메인을 무시하고 정상적으로 시작됩니다.

로컬 도메인에서 사용자 및 그룹을 관리하는 명령줄 도구가 제거되었습니다.

로컬 도메인만 영향을 받는 다음 명령은 제거되었습니다.

- sss_useradd
- sss_userdel
- sss_groupadd
- sss_groupdel

ldap_groups_use_matching_rule_in_chain 옵션에 대한 지원이 제거되었습니다.

이 Active Directory 관련 옵션은 상당한 성능 이점을 제공하지 않으며 RHEL 8 sssd.conf 구성에서 무시됩니다.

ldap_initgroups_use_matching_rule_in_chain 옵션에 대한 지원이 제거되었습니다.

이 **Active Directory** 관련 옵션은 상당한 성능 이점을 제공하지 않으며 **RHEL 8 sssd.conf** 구성에서 무시됩니다.

ldap_sudo_include_regexp 옵션은 이제 기본값은 **false**

RHEL 7에서는 이 옵션이 기본적으로 **true** 로 설정되었습니다. 이 옵션을 **true** 로 설정하면 **SSSD**에서 **sudo Host** 속성에 와일드카드가 포함된 모든 **sudo** 규칙을 다운로드하여 **LDAP** 검색 성능에 부정적인 영향을 미칩니다.

sssd-secrets 응답기가 제거되었습니다.

Kerberos Cache Manager(KCM)는 더 이상 **sssd-secrets** 응답자를 사용하지 않으며 다른 **IdM** 프로세스에서는 사용되지 않았습니다.

2.3. 추가 리소스

- **RHEL 8**로 업그레이드하는 방법에 대한 자세한 내용은 **RHEL 7**에서 **RHEL 8**로 업그레이드를 참조하십시오.

II 부. 외부 소스에서 IDM으로 마이그레이션

3장. RHEL이 아닌 LINUX 배포판의 FREEIPA에서 RHEL 8의 IDM으로 마이그레이션

RHEL 8 서버의 IdM(Identity Management) 배포에 RHEL Linux 배포에서 FreeIPA 배포를 마이그레이션하려면 먼저 새 RHEL 8 IdM CA(인증 기관) 복제본을 기존 FreeIPA 환경에 추가하고 인증서 관련 역할을 이전한 다음, RHEL FreeIPA가 아닌 서버를 폐기해야 합니다.

주의

Convert2RHEL 툴을 사용하여 RHEL 8 IdM 서버로 RHEL FreeIPA 서버의 내부 변환을 수행할 수 없습니다.

마이그레이션을 수행하려면 RHEL 7 서버 역할을 하는 RHEL FreeIPA CA 복제본이 아닌 상태에서 RHEL 7 서버에서 RHEL 8 서버로 IdM 환경을 마이그레이션하는 것과 동일한 절차를 따르십시오.

1. RHEL 8 서버를 구성하고 RHEL이 아닌 Linux 배포판의 현재 FreeIPA 환경에 IdM 복제본으로 추가합니다. 자세한 내용은 RHEL 8 Replica 설치를 참조하십시오.

2. RHEL 8에서 CA(인증 기관) 갱신 서버를 복제하도록 합니다. 자세한 내용은 RHEL 8 IdM 서버에 CA 갱신 서버 역할 할당을 참조하십시오.

3. RHEL이 아닌 서버에서 CRL(인증서 취소 목록)을 생성하고 CRL 요청을 RHEL 8 복제본으로 리디렉션합니다. 자세한 내용은 RHEL 7 IdM CA 서버에서 CRL 생성 중지를 참조하십시오.

4. RHEL 8 서버에서 CRL 생성을 시작합니다. 자세한 내용은 새 RHEL 8 IdM CA 서버에서 CRL 생성 시작을 참조하십시오.

5. RHEL FreeIPA가 아닌 원본 CA 갱신 서버를 중지하고 삭제합니다. 자세한 내용은 RHEL 7 서버 중지 및 비활성화를 참조하십시오.

추가 리소스

- IdM 환경을 RHEL 7 서버에서 RHEL 8 서버로 마이그레이션

4장. LDAP 디렉토리에서 IDM으로 마이그레이션

ID 및 인증 조회를 위해 LDAP 서버를 이전에 배포한 경우 lookup 서비스를 IdM(Identity Management)으로 마이그레이션할 수 있습니다. IdM은 다음 작업을 지원하는 마이그레이션 툴을 제공합니다.

- 데이터 손실 없이 암호 및 그룹 멤버십을 포함한 사용자 계정 전송.
- 클라이언트에서 비용이 많이 드는 구성 업데이트 방지.

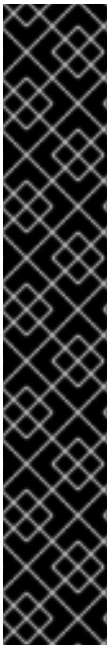
여기에 설명된 마이그레이션 프로세스는 LDAP에서 하나의 네임스페이스와 IdM에 하나의 네임스페이스가 있는 간단한 배포 시나리오를 가정합니다. 여러 네임스페이스 또는 사용자 지정 스키마가 있는 환경과 같이 보다 복잡한 환경에서는 Red Hat 지원 서비스에 문의하십시오.

4.1. LDAP에서 IDM으로 마이그레이션하는 데 대한 고려 사항

LDAP 서버에서 IdM(Identity Management)으로 이동하는 프로세스는 다음과 같습니다.

- *클라이언트* 마이그레이션. 이 단계를 신중하게 계획하십시오. 현재 인프라의 각 클라이언트가 사용하는 서비스를 결정합니다. 여기에는 예를 들어 Kerberos 또는 SSSD(System Security Services Daemon)가 포함될 수 있습니다. 그런 다음 최종 IdM 배포에서 사용할 수 있는 서비스를 확인합니다. 자세한 내용은 LDAP에서 IdM으로 마이그레이션할 때 클라이언트 구성 계획을 참조하십시오.
- *데이터* 마이그레이션.
- *암호* 마이그레이션. 이 단계를 신중하게 계획하십시오. IdM에는 암호 외에도 모든 사용자 계정에 대해 Kerberos 해시가 필요합니다. 암호에 대한 몇 가지 고려 사항 및 마이그레이션 경로에는 LDAP에서 IdM으로 마이그레이션할 때 암호 마이그레이션 플래닝에 포함됩니다.

먼저 서버 부분을 마이그레이션한 다음 클라이언트 또는 클라이언트 우선과 서버를 마이그레이션할 수 있습니다. 두 가지 유형의 마이그레이션에 대한 자세한 내용은 LDAP에서 IdM 마이그레이션 시퀀스를 참조하십시오.



중요

실제 **LDAP** 환경을 마이그레이션하기 전에 테스트 **LDAP** 환경을 설정하고 마이그레이션 프로세스를 테스트하는 것이 좋습니다. 환경을 테스트할 때 다음을 수행합니다.

1. **IdM**에서 테스트 사용자를 생성하고 마이그레이션된 사용자의 출력을 **test** 사용자의 출력과 비교합니다. 마이그레이션된 사용자에게 테스트 사용자에게 최소 한의 속성 및 개체 클래스가 포함되어 있는지 확인합니다.

2. 원래 **LDAP** 서버에 표시된 것처럼 마이그레이션된 사용자의 출력을 소스 사용자와 비교합니다. 가져온 속성이 두 번 복사되지 않고 올바른 값이 있는지 확인합니다.

4.2. LDAP에서 IDM으로 마이그레이션할 때 클라이언트 구성 계획

IdM(Identity Management)은 다양한 수준의 기능, 유연성 및 보안을 통해 다양한 클라이언트 구성을 지원할 수 있습니다. 운영 체제와 **IT** 유지 관리 우선 순위에 따라 각 개별 클라이언트에 가장 적합한 구성을 결정합니다. 클라이언트의 기능 영역도 고려해야 합니다. 일반적으로 개발 시스템에는 프로덕션 서버 또는 사용자 랩탑과 다른 구성이 필요합니다.

중요

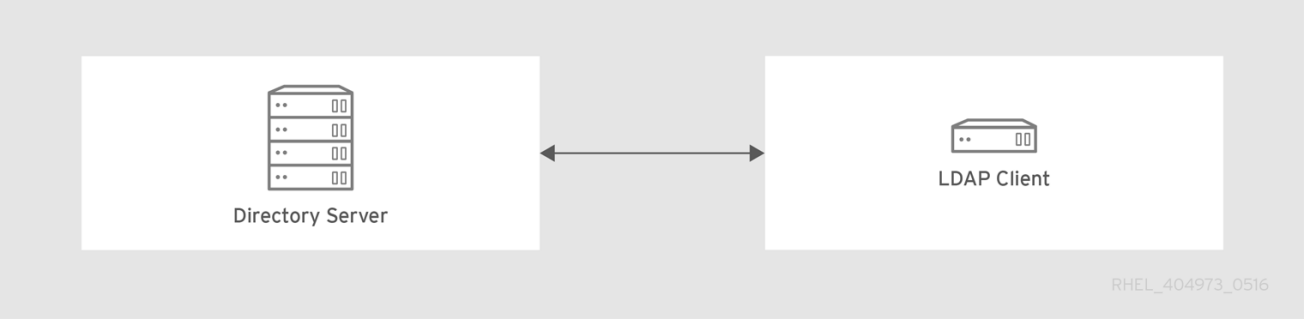
대부분의 환경에서는 클라이언트가 **IdM** 도메인에 연결하는 다양한 방법이 혼합되어 있습니다. 관리자는 각 개별 클라이언트에 가장 적합한 시나리오를 결정해야 합니다.

4.2.1. 초기 마이그레이션 전 클라이언트 구성

IdM(Identity Management)에서 클라이언트 구성의 세부 사항을 결정하기 전에 먼저 현재 마이그레이션 전 구성의 세부 사항을 설정합니다.

마이그레이션할 거의 모든 **LDAP** 배포의 초기 상태는 **ID** 및 인증 서비스를 제공하는 **LDAP** 서비스가 있다는 것입니다.

그림 4.1. 기본 LDAP 디렉터리 및 클라이언트 구성



Linux 및 Unix 클라이언트는 PAM_LDAP 및 NSS_LDAP 라이브러리를 사용하여 LDAP 서비스에 직접 연결합니다. 이러한 라이브러리를 사용하면 /etc/passwd 또는 /etc/shadow 에 데이터가 저장된 것처럼 클라이언트가 LDAP 디렉토리에서 사용자 정보를 검색할 수 있습니다. 실제 환경에서는 클라이언트가 LDAP를 ID 조회에 사용하고 인증 또는 기타 구성에 Kerberos를 사용하는 경우 인프라가 더 복잡할 수 있습니다.

LDAP 디렉터리와 IdM(Identity Management) 서버(특히 스키마 지원 및 디렉터리 트리 구조) 간에는 구조적인 차이점이 있습니다. 이러한 차이점에 대한 자세한 내용은 LDAP에서 IdM으로 마이그레이션할 때 클라이언트 구성 계획에서 표준 LDAP 디렉터리로 IdM 통합 섹션을 참조하십시오. 이러한 차이점은 특히 디렉터리 트리의 경우 입력 이름에 영향을 주는 데이터에 영향을 줄 수 있습니다. 그러나 차이점은 클라이언트 구성 및 클라이언트가 IdM으로 마이그레이션하는 데 거의 영향을 미치지 않습니다.

4.2.2. RHEL 클라이언트에 권장되는 구성

참고

이 섹션에 설명된 클라이언트 구성은 RHEL 6.1 이상 및 RHEL 5.7 이상에서만 지원되며 최신 버전의 SSSD 및 ipa-client 패키지를 지원합니다. 이전 버전의 RHEL은 대체 지원 구성에 설명된 대로 구성할 수 있습니다.

RHEL(Red Hat Enterprise Linux)의 SSSD(System Security Services Daemon)는 특수 PAM 및 NSS 라이브러리인 pam_ss 및 nss_ss 를 사용합니다. SSSD는 이러한 라이브러리를 사용하여 IdM(Identity Management)과 긴밀하게 통합하고 전체 인증 및 ID 기능을 활용할 수 있습니다. SSSD에는 중앙 서버에 대한 연결이 손실된 경우에도 사용자가 로그인할 수 있도록 캐싱 ID 정보와 같은 여러 유용한 기능이 있습니다.

pam_ldap 및 nss_ldap 라이브러리를 사용하는 일반 LDAP 디렉터리 서비스와 달리 SSSD는 *도메인*을 정의하여 ID와 인증 정보 간의 관계를 설정합니다. SSSD의 도메인은 다음 백엔드 기능을 정의합니다.

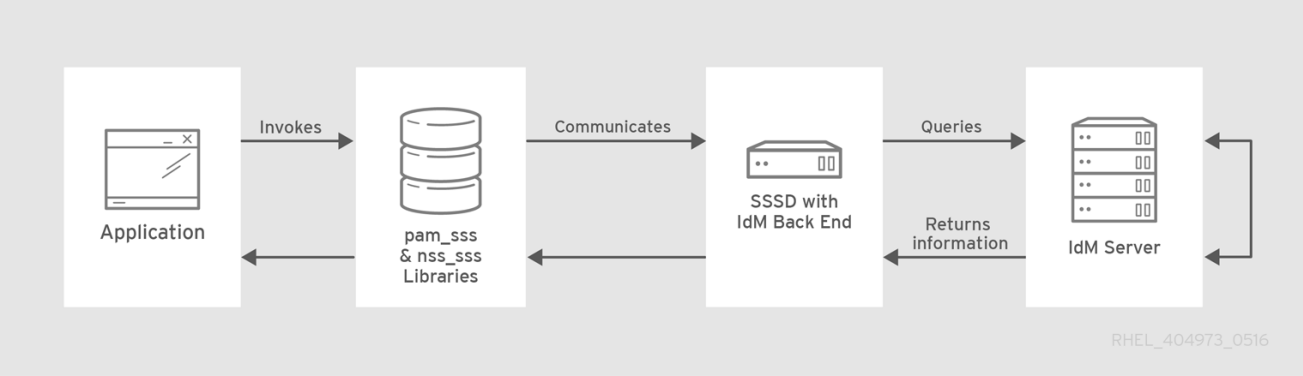
- 인증

- ID 조회
- 액세스
- 암호 변경

그런 다음 **SSSD** 도메인은 *공급자*를 사용하여 이러한 기능의 일부 또는 모두에 대한 정보를 제공하도록 구성됩니다. 도메인 구성에는 항상 ***ID*** 공급자가 필요합니다. 다른 세 개의 공급자는 선택 사항입니다. 인증, 액세스 또는 암호 공급자가 정의되지 않은 경우 **ID** 공급자가 해당 함수에 사용됩니다.

SSSD는 모든 백엔드 기능에 **IdM**을 사용할 수 있습니다. 일반 **LDAP ID** 공급자 또는 **Kerberos** 인증과 달리 모든 범위의 **IdM** 기능을 제공하므로 이상적인 구성입니다. 예를 들어, 메일 작동하는 동안 **SSSD**는 **IdM**에서 호스트 기반 액세스 제어 규칙과 보안 기능을 적용합니다.

그림 4.2. IdM 백엔드를 사용한 클라이언트 및 SSSD



ipa-client-install 스크립트는 모든 백엔드 서비스에 **IdM**을 사용하도록 **SSSD**를 자동으로 구성하여 **RHEL** 클라이언트가 기본적으로 권장 구성으로 설정되도록 합니다.

추가 정보

- SSSD 및 이점 이해

4.2.3. 대체 지원 구성

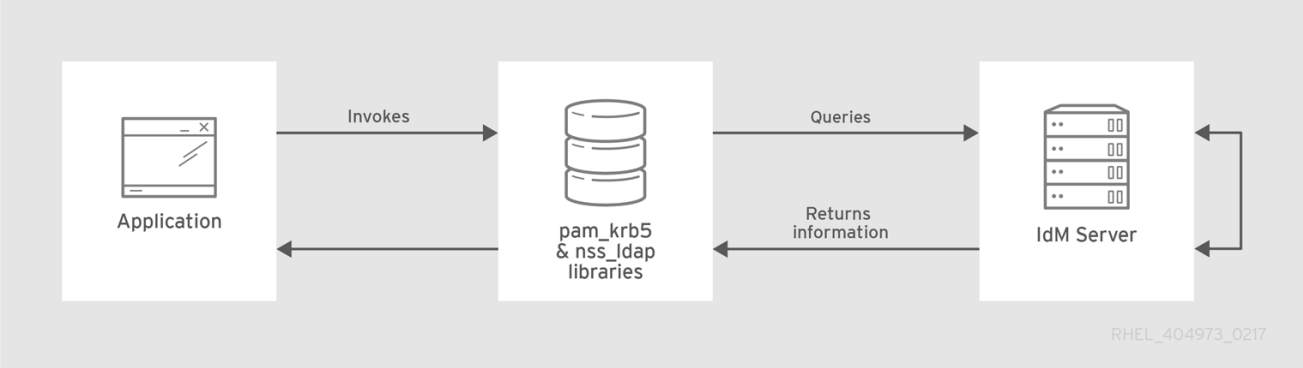
Mac, 솔라리스, **HP-UX**, **Vertical**, **similar Linux**와 같은 **UNIX** 및 **Linux** 시스템은 **IdM(Identity Management)**이 관리하지만 **SSSD**를 사용하지 않는 모든 서비스를 지원합니다. 마찬가지로 이전 **RHEL(Red Hat Enterprise Linux)** 버전, 특히 **6.1** 및 **5.6**은 **SSSD**를 지원하지만 **IdM**을 **ID** 공급자로 지원하지 않는 이전 버전이 있습니다.

시스템에서 최신 버전의 SSSD를 사용할 수 없는 경우 다음과 같은 방식으로 클라이언트를 구성할 수 있습니다.

- 클라이언트는 nss_ldap 를 사용하여 ID 조회를 위한 LDAP 디렉터리 서버인 것처럼 IdM 서버에 연결합니다.
- 클라이언트는 pam_krb5.를 사용하여 일반 Kerberosmtls인 것처럼 IdM 서버에 연결합니다.

IdM 서버를 ID 공급자 및 Kerberos 인증 도메인으로 사용하도록 *이전 버전의 SSSD를 사용하여 RHEL 클라이언트를 구성하는 방법에 대한 자세한 내용은 RHEL 7 시스템 수준 인증 가이드 의 SSSD에 대한 ID 및 인증 공급자 구성* 섹션을 참조하십시오.

그림 4.3. LDAP 및 Kerberos를 사용한 클라이언트 및 IdM



일반적으로 클라이언트에 가능한 가장 안전한 구성을 사용하는 것이 좋습니다. 즉, ID에 사용되는 SSSD 또는 LDAP와 인증을 위한 Kerberos입니다. 그러나 일부 유지 관리 상황 및 IT 구조의 경우 가장 간단한 시나리오에 의존해야 할 수 있습니다. 클라이언트의 nss_ldap 및 pam_ldap 라이브러리를 사용하여 ID와 인증을 모두 제공하도록 LDAP를 구성해야 할 수 있습니다.

4.3. LDAP에서 IDM으로 마이그레이션할 때 암호 마이그레이션

LDAP에서 IdM(Identity Management)으로 사용자를 마이그레이션하기 전에 응답하는 중요한 질문은 사용자 암호를 마이그레이션할지 여부입니다. 다음 옵션을 사용할 수 있습니다.

암호 없이 사용자 마이그레이션

더 빠르게 수행할 수 있지만 관리자와 사용자에 의해 더 많은 수동 작업이 필요합니다. 예를 들어, 원래 LDAP 환경에서 일반 텍스트 사용자 암호를 저장하는 경우 또는 암호가 IdM에 정의된 암호 정책 요구 사항을 충족하지 않는 경우와 같이 사용 가능한 유일한 옵션입니다.

암호 없이 사용자 계정을 마이그레이션할 때 모든 사용자 암호를 재설정합니다. 마이그레이션된 사용자에게는 처음 로그인할 때 변경되는 임시 암호가 할당됩니다. 암호를 재설정하는 방법에 대한 자세한 내용은 **HREL 7 IdM** 설명서에서 사용자 암호 변경 및 재설정 을 참조하십시오.

암호를 사용하여 사용자 마이그레이션

보다 원활한 전환을 제공하지만 마이그레이션 및 전환 프로세스 중에 **LDAP** 디렉터리 및 **IdM**을 병렬 관리해야 합니다. 기본적으로 **IdM**은 인증에 **Kerberos**를 사용하므로 각 사용자에게 표준 사용자 암호 외에도 **IdM Directory Server**에 저장된 **Kerberos** 해시가 있어야 합니다. 해시를 생성하려면 일반 텍스트로 **IdM** 서버에서 사용자 암호를 사용할 수 있어야 합니다. 새 사용자 암호를 생성하면 암호를 해시하고 **IdM**에 저장하기 전에 일반 텍스트로 사용할 수 있습니다. 그러나 사용자가 **LDAP** 디렉터리에서 마이그레이션되면 연결된 사용자 암호가 이미 해시되므로 해당 **Kerberos** 키를 생성할 수 없습니다.

중요

기본적으로 사용자는 사용자 계정이 이미 있는 경우에도 **IdM** 도메인에 인증하거나 **Kerberos** 해시가 있을 때까지 **IdM** 리소스에 액세스할 수 없습니다. **Kerberos** 인증 대신 **IdM**에서 **LDAP** 인증을 사용하는 한 가지 해결 방법을 사용할 수 있습니다. 이 해결 방법으로 **Kerberos** 해시는 사용자에게 필요하지 않습니다. 그러나 이 해결 방법은 **IdM**의 기능을 제한하므로 권장되지 않습니다.

다음 섹션에서는 사용자 및 암호를 마이그레이션하는 방법에 대해 설명합니다.

- **LDAP**를 **IdM**로 마이그레이션할 때 암호를 마이그레이션하는 방법
  - 웹 페이지 사용
  - **SSSD** 사용
- 일반 텍스트 **LDAP** 암호 마이그레이션 계획
- **IdM** 요구 사항을 충족하지 않는 **LDAP** 암호 마이그레이션 계획

4.3.1. LDAP를 IdM로 마이그레이션할 때 암호를 마이그레이션하는 방법

사용자가 암호를 변경하지 않고 **LDAP**에서 **IdM(Identity Management)**으로 사용자 계정을 마이그레

이션하려면 다음 방법을 사용합니다.

방법 1: 마이그레이션 웹 페이지 사용

사용자에게 IdM 웹 UI의 특수 페이지인 *https://ipaserver.example.com/ipa/migration* 의 특수 페이지에 LDAP 자격 증명을 한 번 입력하도록 지시합니다. 백그라운드에서 실행되는 스크립트는 일반 텍스트 암호를 캡처하고 사용자 계정을 암호 및 적절한 Kerberos 해시로 올바르게 업데이트합니다.

방법 2(권장): SSSD를사용합니다.

SSSD(System Security Services Daemon)를 사용하여 필요한 사용자 키를 생성하여 마이그레이션의 사용자 영향을 완화합니다. 많은 사용자가 있거나 암호 변경에 부담을 주지 않아야 하는 경우 최상의 시나리오입니다.

워크플로

1. 사용자는 SSSD를 사용하여 시스템에 로그인을 시도합니다.
2. SSSD는 IdM 서버에 대해 Kerberos 인증을 수행합니다.
3. 사용자가 시스템에 있지만 Kerberos 해시가 아직 존재하지 않기 때문에 오류 *키 유형으로 인증이 지원되지* 않습니다.
4. SSSD는 보안 연결을 통해 일반 텍스트 LDAP 바인딩을 수행합니다.
5. IdM은 이 바인딩 요청을 가로채고 있습니다. Kerberos 주체가 있지만 Kerberos 해시가 없는 경우 IdM ID 공급자는 해시를 생성하고 사용자 항목에 저장합니다.
6. 인증에 성공하면 SSSD가 IdM에서 연결을 끊고 Kerberos 인증을 다시 시도합니다. 이번에는 항목에 해시가 있기 때문에 요청이 성공합니다.

메서드 2에서는 전체 프로세스가 사용자에게 표시되지 않습니다. 암호를 LDAP에서 IdM으로 이동했음을 인식하지 않고 클라이언트 서비스에 로그인합니다.

4.3.2. 일반 텍스트 LDAP 암호 마이그레이션 계획

대부분의 배포 LDAP 암호는 암호화되어 있지만 일부 사용자 또는 사용자 항목에 일반 텍스트 암호를 사용하는 일부 환경이 있을 수 있습니다.

사용자가 LDAP 서버에서 IdM 서버로 마이그레이션되면 IdM에서 일반 텍스트 암호를 허용하지 않기 때문에 일반 텍스트 암호가 마이그레이션되지 않습니다. 대신 각 사용자에 대해 Kerberos 사용자가 생성되면 keytab이 true로 설정되고 암호가 expired로 설정됩니다. 즉, IdM에서는 사용자가 다음 로그인 시 암호를 재설정해야 합니다. 자세한 내용은 IdM 요구 사항을 충족하지 않는 LDAP 암호 마이그레이션 플래닝을 참조하십시오.

4.3.3. IdM 요구 사항을 충족하지 않는 LDAP 암호 마이그레이션 계획

원래 디렉터리의 사용자 암호가 IdM(Identity Management)에 정의된 암호 정책을 충족하지 않는 경우 마이그레이션 후 암호가 유효하지 않습니다.

kinit 를 입력하여 사용자가 IdM 도메인에서 TGT(KGT)를 취득하려고 할 때 암호 재설정이 자동으로 수행됩니다. 사용자가 암호를 변경해야 합니다.

```
[migrated_idm_user@idmclient ~]$ kinit
Password for migrated_idm_user@IDM.EXAMPLE.COM:
Password expired.  You must change it now.
Enter new password:
Enter it again:
```

4.4. 추가 마이그레이션 고려 사항 및 요구 사항

LDAP 서버에서 IdM(Identity Management)으로 마이그레이션을 계획하므로 LDAP 환경에서 IdM 마이그레이션 스크립트를 사용할 수 있는지 확인합니다.

4.4.1. 마이그레이션에 지원되는 LDAP 서버

LDAP 서버에서 IdM으로 마이그레이션 프로세스는 특수 스크립트 ipa migrate-ds 를 사용하여 마이그레이션을 수행합니다. 이 스크립트에는 LDAP 디렉터리 및 LDAP 항목의 구조와 관련된 특정 요구 사항이 있습니다. 몇 가지 공통 디렉터리가 포함된 LDAPv3 호환 디렉터리 서비스에만 마이그레이션이 지원됩니다.

- Sun ONE Directory Server

- **Apache Directory Server**
- **OpenLDAP**

LDAP 서버에서 **IdM**으로의 마이그레이션은 **Red Hat Directory Server** 및 **OpenLDAP**에서 테스트되었습니다.

참고

LDAPv3 호환 디렉터리가 *아니므로* 마이그레이션 스크립트를 사용한 마이그레이션이 **Microsoft Active Directory**에서 지원되지 않습니다. **Active Directory**에서 마이그레이션하는 데 도움이 필요한 경우 **Red Hat Professional Services**에 문의하십시오.

4.4.2. 마이그레이션을 위한 LDAP 환경 요구 사항

LDAP 서버 및 **IdM(Identity Management)**에 대해 가능한 다양한 구성 시나리오가 있으며 이는 마이그레이션 프로세스의 원활한 영향을 미칩니다. 이 장의 마이그레이션 절차 예제에서는 환경에 대한 가정입니다.

- 단일 **LDAP** 디렉터리 도메인이 하나의 **IdM** 영역으로 마이그레이션되고 있습니다. 통합과 관련이 없습니다.
- 사용자 암호는 **LDAP** 디렉터리에 해시로 저장됩니다. 지원되는 해시 목록은 **Red Hat Directory Server** 문서 의 **Red Hat Directory Server 10** 섹션에서 제공되는 *구성, 명령 및 파일 참조* 제목의 암호 스토리지 체계 섹션을 참조하십시오.
- **LDAP** 디렉터리 인스턴스는 **ID** 저장소 및 인증 방법입니다. 클라이언트 시스템은 **pam_ldap** 또는 **nss_ldap** 라이브러리를 사용하여 **LDAP** 서버에 연결하도록 구성됩니다.
- 항목은 표준 **LDAP** 스키마만 사용합니다. 사용자 지정 오브젝트 클래스 또는 속성이 포함된 항목은 **IdM**으로 마이그레이션되지 않습니다.
- **migrate-ds** 명령은 다음 계정만 마이그레이션합니다.
  - **gid Number** 특성이 포함된 경우 속성은 **posixAccount** 객체 클래스에 필요합니다.

- sn 속성이 포함된 경우. 속성은 person 오브젝트 클래스에 필요합니다.

4.4.3. 마이그레이션을 위한 IdM 시스템 요구 사항

약 10,000명의 사용자 및 10개의 그룹이 중간 규모화된 디렉토리를 사용하면 마이그레이션을 진행할 수 있도록 충분한 대상 IdM 시스템이 필요합니다. 마이그레이션을 위한 최소 요구 사항은 다음과 같습니다.

- 4개 코어
- 4GB RAM
- 30GB의 디스크 공간
- IdM 서버의 기본값인 2MB의 SASL 버퍼 크기

  마이그레이션 오류의 경우 버퍼 크기를 늘립니다.

```
[root@ipaserver ~]# ldapmodify -x -D 'cn=directory manager' -w password -h
ipaserver.example.com -p 389

dn: cn=config
changetype: modify
replace: nsslapd-sasl-max-buffer-size
nsslapd-sasl-max-buffer-size: 4194304

modifying entry "cn=config"
```

  nsslapd-sasl-max-buffer-size 값을 바이트 단위로 설정합니다.

추가 리소스

- IdM 서버 하드웨어 권장 사항

4.4.4. sudo 규칙에 대한 고려 사항

LDAP와 함께 sudo 를 사용하는 경우 LDAP에 저장된 sudo 규칙을 IdM(Identity Management)로 수동으로 마이그레이션해야 합니다. IdM에서 호스트 그룹으로 netgroups를 다시 생성하는 것이 좋습니다. IdM은 SSSD sudo 공급자를 사용하지 않는 sudo 구성의 기존 네트워크 그룹으로 hostgroups를 자동으로 제공합니다.

4.4.5. LDAP에서 IdM 마이그레이션 툴

IdM(Identity Management)은 특정 명령 ipa migrate-ds 를 사용하여 마이그레이션 프로세스를 실행하여 LDAP 디렉터리 데이터가 올바르게 포맷되고 IdM 서버로 정리되도록 합니다. ipa migrate-ds 를 사용하는 경우 --bind-dn 옵션으로 지정된 원격 시스템 사용자는 userPassword 속성에 대한 읽기 액세스 권한이 있어야 합니다. 그렇지 않으면 암호는 마이그레이션되지 않습니다.

마이그레이션 모드에서 실행되도록 IdM 서버를 구성해야 하며 마이그레이션 스크립트를 사용할 수 있습니다. 자세한 내용은 IdM으로 LDAP 서버 마이그레이션을 참조하십시오.

4.4.6. IdM으로의 마이그레이션 성능 개선

LDAP 마이그레이션은 기본적으로 IdM 서버 내의 389 Directory Server(DS) 인스턴스에 대한 특수 가져오기 작업입니다. 가져오기 작업 성능을 개선하기 위해 389 DS 인스턴스를 튜닝하면 전체 마이그레이션 성능을 개선하는 데 도움이 될 수 있습니다.

가져오기 성능에 직접적인 영향을 주는 매개변수는 다음 두 가지가 있습니다.

- 항목 캐시에 허용되는 크기를 정의하는 nsslapd-cachememsize 속성입니다. 이는 전체 캐시 메모리 크기 중 80%로 자동 설정된 버퍼입니다. 대규모 가져오기 작업의 경우 이 매개변수와 메모리 캐시 자체를 늘릴 수 있습니다. 이렇게 증가하면 많은 수의 항목 또는 특성이 많은 항목을 처리하는 디렉터리 서비스의 효율성이 향상됩니다.

  dsconf 명령을 사용하여 속성을 수정하는 방법에 대한 자세한 내용은 항목 캐시 크기 조정을 참조하십시오.

- 시스템 ulimit 구성 옵션은 시스템 사용자에게 허용되는 최대 프로세스 수를 설정합니다. 큰 데이터베이스를 처리하는 것은 제한을 초과할 수 있습니다. 이 경우 값을 늘립니다.

```
[root@server ~]# ulimit -u 4096
```

추가 리소스

- IdM Directory Server 성능 조정

4.4.7. LDAP에서 IdM 마이그레이션 시퀀스

IdM으로 마이그레이션할 때는 네 가지 주요 단계가 있지만, 먼저 *서버* 또는 *클라이언트를* 마이그레이션하려는지 여부에 따라 순서가 다릅니다.

중요

클라이언트 우선 및 서버 우선 마이그레이션 모두 일반적인 마이그레이션 절차를 제공하지만 모든 환경에서 작동하지 않을 수 있습니다. 실제 LDAP 환경을 마이그레이션하기 전에 테스트 LDAP 환경을 설정하고 마이그레이션 프로세스를 테스트합니다.

클라이언트 우선 마이그레이션

SSSD는 IdM(Identity Management) 서버가 구성된 동안 클라이언트 구성을 변경하는 데 사용됩니다.

1. SSSD를 배포합니다.
2. 클라이언트를 다시 구성하여 현재 LDAP 서버에 연결한 다음 IdM에 장애 조치합니다.
3. IdM 서버를 설치합니다.
4. IdM ipa migrate-ds 스크립트를 사용하여 사용자 데이터를 마이그레이션합니다. 이를 통해 LDAP 디렉터리, IdM 스키마 형식의 데이터를 내보낸 다음 IdM으로 가져옵니다.
5. LDAP 서버를 오프라인으로 전환하고 클라이언트가 IdM에 투명하게 장애 조치를 취할 수 있도록 합니다.

서버 우선 마이그레이션

IdM으로의 LDAP 마이그레이션 우선 순위:

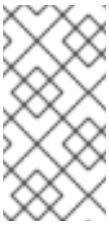
1. IdM 서버를 설치합니다.

2. **IdM ipa migrate-ds** 스크립트를 사용하여 사용자 데이터를 마이그레이션합니다. 이렇게 하면 **LDAP** 디렉터리에서 데이터를 내보내고, **IdM** 스키마용으로 포맷한 다음, **IdM**으로 가져옵니다.

3. *선택 사항:* **SSSD**를 배포합니다.

4. **IdM**에 연결하도록 클라이언트를 재구성합니다. **LDAP** 서버를 교체하기만 하면 안 됩니다. **IdM** 디렉터리 트리 및 사용자 항목 **DN**은 이전 디렉터리 트리와 다릅니다.

   클라이언트를 재구성해야 하지만 클라이언트를 즉시 재구성할 필요는 없습니다. 업데이트된 클라이언트는 **IdM** 서버를 가리킬 수 있지만 다른 클라이언트는 기존 **LDAP** 디렉터리를 가리키므로 데이터를 마이그레이션한 후 합리적인 테스트 및 전환 단계를 수행할 수 있습니다.



참고

LDAP 디렉터리 서비스와 **IdM** 서버를 병렬로 실행하지 마십시오. 이로 인해 두 서비스 간에 사용자 데이터가 일관되지 않을 위험이 있습니다.

4.5. LDAP에서 IDM으로 마이그레이션 사용자 정의

ipa migrate-ds 명령을 사용하여 **LDAP** 서버에서 **IdM(Identity Management)**으로 인증 및 권한 부여 서비스를 마이그레이션할 수 있습니다. 추가 옵션 없이 명령은 일반적인 기본 설정에 따라 데이터를 마이그레이션 및 내보내는 디렉터리의 **LDAP URL**을 사용합니다.

다른 **ipa migrate-ds** 명령 옵션을 사용하여 마이그레이션 프로세스 및 데이터를 식별하고 내보내는 방법을 사용자 지정할 수 있습니다. **LDAP** 디렉터리 트리에 고유한 구조가 있거나 항목 내의 특정 항목 또는 속성을 제외해야 하는 경우 마이그레이션을 사용자 지정합니다.

4.5.1. LDAP에서 IdM으로 마이그레이션하는 동안 바인딩 DN 및 기본 DN 사용자 정의 예

ipa migrate-ds 명령을 사용하여 **LDAP**에서 **IdM(Identity Management)**으로 마이그레이션합니다. 추가 옵션 없이 명령은 일반적인 기본 설정에 따라 데이터를 마이그레이션 및 내보내는 디렉터리의 **LDAP URL**을 사용합니다. 이 섹션에서는 기본 설정을 수정하는 예제에 대해 설명합니다.

```
# ipa migrate-ds ldap://ldap.example.com:389
```

바인딩 **DN** 사용자 지정

기본적으로 **DN "cn=Directory Manager"**는 원격 **LDAP** 디렉터리에 바인딩하는 데 사용됩니다. **--bind-dn** 옵션을 사용하여 사용자 정의 바인딩 **DN**을 지정합니다.

```
# ipa migrate-ds ldap://ldap.example.com:389 --bind-dn=cn=Manager,dc=example,dc=com
```

이름 지정 컨텍스트 사용자 지정

LDAP 서버 이름 지정 컨텍스트가 **IdM**에서 사용되는 컨텍스트와 다른 경우 오브젝트의 기본 **DN**이 변환됩니다. 예를 들어 **uid =*사용자*,ou=people,dc=ldap,dc=example,dc=com** 은 **uid =*사용자*,ou=people,dc=idm,dc=example,dc=com** 으로 마이그레이션됩니다. **base -dn** 옵션을 사용하여 컨테이너 하위 트리의 대상을 변경하여 마이그레이션을 위해 원격 **LDAP** 서버에서 사용되는 기본 **DN**을 설정할 수 있습니다.

```
# ipa migrate-ds --base-dn="ou=people,dc=example,dc=com" ldap://ldap.example.com:389
```

추가 리소스

- **ipa migrate-ds --help**

4.5.2. 특정 하위 트리 마이그레이션

기본 디렉터리 구조에서는 **ou=People** 하위 트리 및 그룹 항목에 **ou=Groups** 하위 트리에 **person** 항목을 배치합니다. 이러한 하위 트리는 다양한 유형의 디렉터리 데이터에 대한 컨테이너 항목입니다. **migrate-ds** 명령과 함께 옵션을 사용하지 않는 경우 유틸리티는 지정된 **LDAP** 디렉터리가 **ou=People** 및 **ou=Groups** 구조를 사용한다고 가정합니다.

대부분의 배포에는 완전히 다른 디렉터리 구조가 있거나 원래 디렉터리 트리의 특정 부분만 내보낼 수 있습니다. 관리자는 다음 옵션을 사용하여 소스 **LDAP** 서버에서 다른 사용자 또는 그룹 하위 트리의 **RDN**을 지정할 수 있습니다.

- **--user-container**
- **--group-container**

참고

두 경우 모두 하위 트리는 상대 고유 이름(RDN)이어야 하며 기본 DN을 기준으로 해야 합니다. 예를 들어 **--user-container =ou=ou=hieras**를 사용하여 **>ou=mtlss,dc=example,dc= com** 디렉터리 트리를 마이그레이션할 수 있습니다.

예를 들면 다음과 같습니다.

```
[ipaserver ~]# ipa migrate-ds --user-container=ou=employees \
--group-container="ou=employee groups" ldap://ldap.example.com:389
```

선택적으로 **ipa migrate-ds** 명령에 **--scope** 옵션을 추가하여 범위를 설정합니다.

- **onelevel: Default** 지정된 컨테이너의 항목만 마이그레이션됩니다.
- 지정된 컨테이너의 하위 트리: 항목 및 모든 하위 컨테이너가 마이그레이션됩니다.
- 지정된 개체 자체만 마이그레이션됩니다.**Only the specified object itself is migrated.**

4.5.3. 항목 포함 및 제외

기본적으로 **ipa migrate-ds** 스크립트는 **person** 오브젝트 클래스와 **groupOfUniqueNames** 또는 **groupOfNames** 오브젝트 클래스를 사용하여 모든 사용자 항목을 가져옵니다.

일부 마이그레이션 경로에서 특정 유형의 사용자 및 그룹만 내보내야 할 수도 있고, 대안으로 특정 사용자 및 그룹만 제외해야 할 수도 있습니다. 사용자 또는 그룹 항목을 찾을 때 검색할 오브젝트 클래스를 설정하여 포함할 사용자 및 그룹 *유형을* 선택할 수 있습니다.

이 옵션은 다양한 사용자 유형에 대해 사용자 지정 개체 클래스를 사용할 때 특히 유용합니다. *예를 들어 다음 명령은 사용자 지정* ***fullTime mtls*** *오브젝트 클래스를 사용하는 사용자만 마이그레이션합니다.*

```
[root@ipaserver ~]# ipa migrate-ds --user-objectclass=fullTimeEmployee
ldap://ldap.example.com:389
```

다양한 유형의 그룹으로 인해 이는 사용자 그룹과 같은 특정 유형의 그룹 만 마이그레이션하는 동시에 인증서 그룹과 같은 다른 유형의 그룹을 제외하는 데도 유용합니다. 예를 들면 다음과 같습니다.

```
[root@ipaserver ~]# ipa migrate-ds --group-objectclass=groupOfNames --group-
objectclass=groupOfUniqueNames ldap://ldap.example.com:389
```

오브젝트 클래스를 기반으로 마이그레이션할 사용자 및 그룹 항목을 지정하면 다른 모든 사용자 및 그룹이 마이그레이션에서 암시적으로 제외됩니다.

또는 소수의 항목만 제외하고 모든 사용자 및 그룹 항목을 마이그레이션하는 것이 유용할 수 있습니다. 해당 유형의 다른 모든 사용자를 마이그레이션하는 동안 특정 사용자 또는 그룹 계정을 제외할 수 있습니다. 예를 들어, 이 예제에서는 **hobbies** 그룹과 두 명의 사용자만 제외합니다.

```
[root@ipaserver ~]# ipa migrate-ds --exclude-groups="Golfers Group" --exclude-
users=idmuser101 --exclude-users=idmuser102 ldap://ldap.example.com:389
```

exclude 문은 **uid**의 패턴과 일치하는 사용자 및 **cn** 특성에 일치하는 그룹에 적용됩니다.

일반 개체 클래스를 마이그레이션하지만 해당 클래스의 특정 항목을 제외할 수 있습니다. 예를 들어, 여기에는 **fullTime mtls** 오브젝트 클래스가 있는 사용자가 구체적으로 포함되어 있지만 세 명의 관리자가 제외됩니다.

```
[root@ipaserver ~]# ipa migrate-ds --user-objectclass=fullTimeEmployee --exclude-
users=jsmith --exclude-users=bjensen --exclude-users=mreynolds
ldap://ldap.example.com:389
```

4.5.4. 엔트리 속성의 제외

기본적으로 사용자 또는 그룹 항목의 모든 속성 및 개체 클래스가 마이그레이션됩니다. 특정 시나리오에서는 대역폭 및 네트워크 제약 조건으로 인해 또는 속성 데이터가 더 이상 관련이 없기 때문에 현실적이지 않을 수 있습니다. 예를 들어 **IdM(Identity Management)** 도메인에 가입하면 **userCertificate** 특성을 마이그레이션할 때 사용자에게 새 사용자 인증서가 할당되지 않습니다.

migrate-ds 명령과 함께 다음 옵션을 사용하여 특정 오브젝트 클래스와 속성을 무시할 수 있습니다.

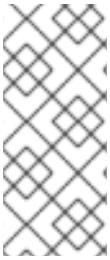
- **--user-ignore-objectclass**
- **--user-ignore-attribute**

- *--group-ignore-objectclass*

- *--group-ignore-attribute*

예를 들어 사용자에 대한 userCertificate 속성 및 strongAuthenticationUser 개체 클래스를 제외하려면 그룹에 대한 groupOfCertificates 오브젝트 클래스를 제외합니다.

```
[root@ipaserver ~]# ipa migrate-ds --user-ignore-attribute=userCertificate --user-ignore-objectclass=strongAuthenticationUser --group-ignore-objectclass=groupOfCertificates ldap://ldap.example.com:389
```



참고

필요한 속성은 무시하지 않도록 합니다. 또한 개체 클래스를 제외할 때는 해당 개체 클래스만 지원하는 속성을 제외해야 합니다.Also, when excluding object classes, make sure to exclude any attributes that only that object class supports.

추가 리소스

- *마이그레이션을 위한 LDAP 환경 요구 사항*

4.5.5. LDAP에서 IdM으로 마이그레이션할 때 사용할 스키마 및 스키마 비교 기능

IdM(Identity Management)은 RFC2307bis 스키마를 사용하여 사용자, 호스트, 호스트 그룹 및 기타 네트워크 ID를 정의합니다. 그러나 마이그레이션의 소스로 사용된 LDAP 서버에서 RFC2307 스키마를 사용하는 경우 ipa migrate-ds 명령을 사용하여 --schema 옵션을 지정합니다.

```
[root@ipaserver ~]# ipa migrate-ds --schema=RFC2307 ldap://ldap.example.com:389
```

또는 IdM에는 RFC2307bis를 지원하지 않는 시스템의 데이터를 다시 포맷할 수 있는 기본 제공 schema compat 기능이 있습니다. compat 플러그인은 기본적으로 활성화되어 있습니다. 즉, 디렉터리 서버에서 사용자 및 그룹의 대체 보기를 계산하고 cn=users,cn=compat,dc=example,dc=com 컨테이너 항목에 이 보기를 제공합니다. 이 작업은 시작 시 해당 항목의 내용을 미리 보고 필요에 따라 항목을 새로 고침하여 수행합니다.

시스템 오버헤드를 줄이기 위해 마이그레이션 중에 이 기능을 사용하지 않도록 설정하는 것이 좋습니다.

4.6. LDAP 서버를 IDM으로 마이그레이션

ipa migrate-ds 명령을 사용하여 **LDAP** 서버에서 **IdM(Identity Management)**으로 인증 및 권한 부여 서비스를 마이그레이션할 수 있습니다.

주의

이는 모든 환경에서 작동하지 않을 수 있는 일반적인 마이그레이션 절차입니다.

실제 **LDAP** 환경을 마이그레이션하기 전에 테스트 **LDAP** 환경을 설정하고 마이그레이션 프로세스를 테스트하는 것이 좋습니다. 환경을 테스트할 때 다음을 수행합니다.

1. **IdM**에서 테스트 사용자를 생성하고 마이그레이션된 사용자의 출력을 **test** 사용자의 출력과 비교합니다.
2. 원래 **LDAP** 서버에 표시된 것처럼 마이그레이션된 사용자의 출력을 소스 사용자와 비교합니다.

자세한 내용은 아래의 확인 섹션을 참조하십시오.

사전 요구 사항

- **LDAP** 디렉터리에 대한 관리자 권한이 있습니다.
- **IdM**이 이미 설치되어 있는 경우 **IdM**에 대한 관리자 권한이 있습니다.
- 아래 절차를 실행하는 **RHEL** 시스템에서 **root** 로 로그인했습니다.
- 다음 부분을 읽고 이해했습니다.

- LDAP에서 IdM으로의 마이그레이션에 대한 고려 사항.
- LDAP에서 IdM으로 마이그레이션할 때 클라이언트 구성 계획.
- LDAP에서 IdM으로 마이그레이션할 때 암호 마이그레이션 계획.
- 추가 마이그레이션 고려 사항 및 요구 사항
- LDAP에서 IdM으로 마이그레이션 사용자 정의

절차

1. IdM이 아직 설치되지 않은 경우 사용자 정의 LDAP 디렉터리 스키마를 포함하여 IdM 서버를 설치합니다. 기존 LDAP 디렉터리가 설치된 것과 다른 시스템에 사용자 지정 LDAP 디렉터리 스키마를 포함합니다. 자세한 내용은 Identity Management 설치를 참조하십시오.

참고

사용자 지정 사용자 또는 그룹 스키마는 IdM에서 지원이 제한됩니다. 호환되지 않는 오브젝트 정의로 인해 마이그레이션 중에 문제가 발생할 수 있습니다.

2. 성능상의 이유로 compat 플러그인을 비활성화합니다.

```
# ipa-compat-manage disable
```

스키마 compat 기능에 대한 자세한 내용과 마이그레이션을 위해 이를 비활성화하는 이점에 대한 자세한 내용은 LDAP에서 IdM으로 마이그레이션할 때 사용할 스키마 및 스키마 비교 기능을 참조하십시오.

3. IdM Directory Server 인스턴스를 다시 시작합니다.

```
# systemctl restart dirsrv.target
```

4. 마이그레이션을 허용하도록 IdM 서버를 구성합니다.

```
# ipa config-mod --enable-migration=TRUE
```

--enable-migration 을 TRUE로 설정하면 다음을 수행합니다.

- LDAP 추가 작업 중에 사전 해시된 암호를 허용합니다.
- 초기 Kerberos 인증에 실패하는 경우 암호 마이그레이션 시퀀스를 시도하도록 SSSD 를 구성합니다. 자세한 내용은 LDAP에서 IdM으로 암호를 마이그레이션할 때 SSSD 사용 의 워크플로 섹션을 참조하십시오.

5. 사용 사례에 맞는 옵션을 사용하여 IdM 마이그레이션 스크립트 ipa migrate-ds. 자세한 내용은 LDAP에서 IdM으로 마이그레이션 사용자 지정을 참조하십시오.

```
# ipa migrate-ds --your-options ldap://ldap.example.com:389
```

참고

이전 단계 중 하나에서 compat 플러그인을 비활성화하지 않은 경우 ipa migrate-ds 에 --with-compat 옵션을 추가합니다.

```
# ipa migrate-ds --your-options --with-compat
ldap://ldap.example.com:389
```

6. compat 플러그인을 다시 활성화합니다.

```
# ipa-compat-manage enable
```

7. IdM 디렉터리 서버를 다시 시작하십시오.

```
# systemctl restart dirsrv.target
```

8. 모든 사용자가 암호를 마이그레이션한 경우 마이그레이션 모드를 비활성화합니다.

```
# ipa config-mod --enable-migration=FALSE
```

9. *[선택 사항] 모든 사용자를 마이그레이션하고 비SSSD 클라이언트를 재구성하여 Kerberos 인증을 사용하도록 재구성합니다. 즉, LDAP 인증 대신 pam_ krb5 입니다. 자세한 내용은 RHEL 7 시스템 수준 인증 가이드 의 Kerberos 클라이언트 구성을 참조하십시오.*

10. *사용자가 해시된 Kerberos 암호를 생성하도록 합니다. LDAP에서 IdM으로 마이그레이션할 때 계획 암호 마이그레이션에 설명된 방법 중 하나를 선택합니다.*

- *SSSD 방법을 결정하는 경우 :*
  - *SSSD가 LDAP 디렉터리에서 IdM 디렉터리로 설치된 클라이언트를 이동하고 IdM에 클라이언트로 등록합니다. 이렇게 하면 필요한 키와 인증서가 다운로드됩니다.*

    Red Hat Enterprise Linux 클라이언트에서 ipa-client-install 명령을 사용하여 이 작업을 수행할 수 있습니다. 예를 들면 다음과 같습니다.

    ```
    # ipa-client-install --enable-dns-update
    ```

- *IdM 마이그레이션 웹 페이지 방법을 결정하는 경우:*
  - *마이그레이션 웹 페이지를 사용하여 IdM에 로그인하도록 사용자에게 지시합니다.*

    ```
    https://ipaserver.example.com/ipa/migration
    ```

11. *사용자 마이그레이션 프로세스를 모니터링하려면 기존 LDAP 디렉터리를 쿼리하여 암호가 있지만 Kerberos 보안 키가 없는 사용자 계정을 확인합니다.*

```
$ ldapsearch -LL -x -D 'cn=Directory Manager' -w secret -b
'cn=users,cn=accounts,dc=example,dc=com' '(&(!(krbprincipalkey=))(userpassword=))'
uid
```

참고

쉘에서 해석하지 않도록 필터 주변의 작은따옴표를 포함합니다.

12. 모든 클라이언트 및 사용자의 마이그레이션이 완료되면 **LDAP** 디렉토리를 제거하십시오.

검증

1. **ipa user-add** 명령을 사용하여 **IdM**에 테스트 사용자를 생성합니다. 마이그레이션된 사용자의 출력을 **test** 사용자의 출력과 비교합니다. 마이그레이션된 사용자에게 테스트 사용자에게 최소한의 속성 및 개체 클래스가 포함되어 있는지 확인합니다. 예를 들면 다음과 같습니다.

```
$ ipa user-show --all testing_user
dn: uid=testing_user,cn=users,cn=accounts,dc=idm,dc=example,dc=com
User login: testing_user
First name: testing
Last name: user
Full name: testing user
Display name: testing user
Initials: tu
Home directory: /home/testing_user
GECOS: testing user
Login shell: /bin/sh
Principal name: testing_user@IDM.EXAMPLE.COM
Principal alias: testing_user@IDM.EXAMPLE.COM
Email address: testing_user@idm.example.com
UID: 1689700012
GID: 1689700012
Account disabled: False
Preserved user: False
Password: False
Member of groups: ipausers
Kerberos keys available: False
ipauniqueid: 843b1ac8-6e38-11ec-8dfe-5254005aad3e
mepmanagedentry:
cn=testing_user,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
objectclass: top, person, organizationalperson, inetorgperson, inetuser, posixaccount,
krbprincipalaux, krbticketpolicyaux, ipaobject,
        ipasshuser, ipaSshGroupOfPubKeys, mepOriginEntry
```

2. 원래 **LDAP** 서버에 표시된 것처럼 마이그레이션된 사용자의 출력을 소스 사용자와 비교합니다. 가져온 속성이 두 번 복사되지 않고 올바른 값이 있는지 확인합니다.

추가 리소스

- **LDAP에서 SSL을 통해 IdM으로 마이그레이션**

4.7. LDAP에서 SSL을 통해 IDM으로 마이그레이션

ipa migrate-ds 명령을 사용하여 **LDAP** 서버에서 **IdM(Identity Management)**으로 인증 및 권한 부여

서비스를 마이그레이션할 수 있습니다. 이 섹션에서는 마이그레이션 중에 전송된 데이터를 암호화하는 방법에 대해 설명합니다.

주의

이는 모든 환경에서 작동하지 않을 수 있는 일반적인 마이그레이션 절차입니다.

실제 **LDAP** 환경을 마이그레이션하기 전에 테스트 **LDAP** 환경을 설정하고 마이그레이션 프로세스를 테스트하는 것이 좋습니다. 환경을 테스트할 때 다음을 수행합니다.

1. **IdM**에서 테스트 사용자를 생성하고 마이그레이션된 사용자의 출력을 **test** 사용자의 출력과 비교합니다.
2. 원래 **LDAP** 서버에 표시된 것처럼 마이그레이션된 사용자의 출력을 소스 사용자와 비교합니다.

자세한 내용은 아래의 확인 섹션을 참조하십시오.

사전 요구 사항

- **LDAP** 디렉터리에 대한 관리자 권한이 있습니다.
- **IdM**이 이미 설치되어 있는 경우 **IdM**에 대한 관리자 권한이 있습니다.
- 아래 절차를 실행하는 **RHEL** 시스템에서 **root** 로 로그인했습니다.
- 다음 부분을 읽고 이해했습니다.
  - *LDAP에서 IdM으로의 마이그레이션에 대한 고려 사항*.

- *LDAP에서 IdM으로 마이그레이션할 때 클라이언트 구성 계획.*
- *LDAP에서 IdM으로 마이그레이션할 때 암호 마이그레이션 계획.*
- *추가 마이그레이션 고려 사항 및 요구 사항*
- *LDAP에서 IdM으로 마이그레이션 사용자 정의*

절차

1. *향후 **IdM** 서버의 파일에 원격 **LDAP** 서버 인증서를 발급한 **CA** 인증서를 저장합니다. 예: **/tmp/remote.crt.***

2. *LDAP 서버를 IdM으로 마이그레이션하는 방법에 설명된 단계를 따르십시오. 그러나 마이그레이션 중에 암호화된 **LDAP** 연결의 경우 **URL**에서 **ldaps** 프로토콜을 사용하고 **--ca-cert-file** 옵션을 **ipa migrate-ds** 명령에 전달합니다. 예를 들면 다음과 같습니다.*

```
# ipa migrate-ds --ca-cert-file=/tmp/remote.crt --your-other-options
ldaps://ldap.example.com:636
```

검증

1. ***ipa user-add** 명령을 사용하여 **IdM**에 테스트 사용자를 생성합니다. 마이그레이션된 사용자의 출력을 **test** 사용자의 출력과 비교합니다. 마이그레이션된 사용자에게 테스트 사용자에게 최소한의 속성 및 개체 클래스가 포함되어 있는지 확인합니다. 예를 들면 다음과 같습니다.*

```
$ ipa user-show --all testing_user
dn: uid=testing_user,cn=users,cn=accounts,dc=idm,dc=example,dc=com
User login: testing_user
First name: testing
Last name: user
Full name: testing user
Display name: testing user
Initials: tu
Home directory: /home/testing_user
GECOS: testing user
Login shell: /bin/sh
Principal name: testing_user@IDM.EXAMPLE.COM
Principal alias: testing_user@IDM.EXAMPLE.COM
Email address: testing_user@idm.example.com
UID: 1689700012
```

```
GID: 1689700012
Account disabled: False
Preserved user: False
Password: False
Member of groups: ipausers
Kerberos keys available: False
ipauniqueid: 843b1ac8-6e38-11ec-8dfe-5254005aad3e
mepmanagedentry:
cn=testing_user,cn=groups,cn=accounts,dc=idm,dc=example,dc=com
objectclass: top, person, organizationalperson, inetorgperson, inetuser, posixaccount,
krbprincipalaux, krbticketpolicyaux, ipaobject,
        ipasshuser, ipaSshGroupOfPubKeys, mepOriginEntry
```

2. *원래 **LDAP** 서버에 표시된 것처럼 마이그레이션된 사용자의 출력을 소스 사용자와 비교합니다. 가져온 속성이 두 번 복사되지 않고 올바른 값이 있는지 확인합니다.*