Red Hat Enterprise Linux 9

Identity Management와 함께 외부 Red Hat 유틸리티 사용

Identity Management를 사용하여 Satellite, Open Shift 및 Samba와 같은 기타 Red Hat 유틸리티를 사용하기 위한 가이드입니다.

Last Updated: 2022-08-29

Red Hat Enterprise Linux 9 Identity Management와 함께 외부 Red Hat 유틸리티 사용

Identity Management를 사용하여 Satellite, Open Shift 및 Samba와 같은 기타 Red Hat 유틸리티를 사용하기 위한 가이드입니다.

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

법적 공지

초록

이 문서 컬렉션은 ID 관리 배포를 Ansible Automation Platform, NFS, OpenShift Container Platform, OpenStack Platform, Samba, Satellite, Single Sign-On 및 가상화와 통합하는 방법에 대한 지침을 제공합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체

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

Identity Management에서 용어 교체는 다음과 같습니다.

- *블랙리스트*를 ***블록 목록***으로 대체
- *화이트리스트*를 ***허용 목록***으로 대체
- *슬레이브*를 ***보조***로 대체
- 컨텍스트에 따라 *마스터*라는 단어가 보다 정확한 용어로 교체되고 있습니다.
  - *IdM 마스터*를 ***IdM 서버***로 대체
  - *CA 갱신 마스터*를 ***CA 갱신 서버***로 대체
  - *CRL 마스터*를 ***CRL 게시자 서버***로 대체
  - *multi-master*를 ***multi-supplier***로 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 생성하십시오.
  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.

1장. IDM과 다른 RED HAT 제품과의 통합

이 섹션에서는 IdM과 통합된 다른 Red Hat 제품에 대한 문서에 대한 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

SSO(Single Sign-On)

SSSD 및 FreeIPA Identity Management 통합

가상화

외부 LDAP 공급자 구성

2장. IDM 도메인 멤버에서 SAMBA 설정

이 섹션에서는 Red Hat IdM(Identity Management) 도메인에 연결된 호스트에서 Samba를 설정하는 방법을 설명합니다. IdM의 사용자 및 또한 신뢰할 수 있는 AD(Active Directory) 도메인에서 사용 가능한 경우 Samba에서 제공하는 공유 및 프린터 서비스에 액세스할 수 있습니다.

중요

IdM 도메인 멤버에서 Samba를 사용하는 것은 지원되지 않는 기술 프리뷰 기능이며 특정 제한 사항이 포함되어 있습니다. 예를 들어, IdM 신뢰 컨트롤러는 Active Directory 글로벌 카탈로그 서비스를 지원하지 않으며 DMCE/원격 프로시저 호출(DCE/RPC) 프로토콜을 사용하여 IdM 그룹 해결을 지원하지 않습니다. 결과적으로 AD 사용자는 다른 IdM 클라이언트에 로그인할 때 IdM 클라이언트에서 호스팅되는 Samba 공유 및 프린터에만 액세스할 수 있습니다. Windows 시스템에 로그인한 AD 사용자는 IdM 도메인 멤버에서 호스팅되는 Samba 공유에 액세스할 수 없습니다.

IdM 도메인 구성원에 Samba를 배포하는 고객은 Red Hat에 피드백을 제공하는 것이 좋습니다.

사전 요구 사항

- 호스트는 IdM 도메인에 클라이언트로 결합됩니다.

2.1. 도메인 멤버에 SAMBA를 설치하기 위해 IDM 도메인 준비

IdM 클라이언트에 Samba를 설정하려면 IdM 서버에서 **ipa-adtrust-install** 유틸리티를 사용하여 IdM 도메인을 준비해야 합니다.

참고

ipa-adtrust-install 명령을 실행하는 시스템은 자동으로 AD 신뢰 컨트롤러가 됩니다. 그러나 IdM 서버에서 **ipa-adtrust-install** 을 한 번만 실행해야 합니다.

사전 요구 사항

- IdM 서버가 설치되어 있어야 합니다.
- 패키지를 설치하고 IdM 서비스를 다시 시작하려면 루트 권한이 필요합니다.

절차

1. 필수 패키지를 설치합니다.

```
[root@ipaserver ~]# dnf install ipa-server-trust-ad samba-client
```

2. IdM 관리자로 인증합니다.

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

3. **ipa-adtrust-install** 유틸리티를 실행합니다.

```
[root@ipaserver ~]# ipa-adtrust-install
```

IdM이 통합된 DNS 서버와 함께 설치된 경우 DNS 서비스 레코드가 자동으로 생성됩니다.

통합된 DNS 서버 없이 IdM을 설치한 경우, **ipa-adtrust-install** 은 DNS에 수동으로 추가해야 하는 서비스 레코드 목록을 인쇄합니다.

4. 스크립트에서 **/etc/samba/smb.conf**가 이미 존재하고 다시 작성됨을 묻는 메시지를 표시합니다.

```
WARNING: The smb.conf already exists. Running ipa-adtrust-install will break your existing
Samba configuration.

Do you wish to continue? [no]: yes
```

5. 스크립트에서 이전 Linux 클라이언트가 신뢰할 수 있는 사용자로 작업할 수 있는 호환성 플러그인인 **slapi-nis** 플러그인을 구성하도록 프롬프트를 표시합니다.

```
Do you want to enable support for trusted domains in Schema Compatibility plugin?
This will allow clients older than SSSD 1.9 and non-Linux clients to work with trusted users.

Enable trusted domains support in slapi-nis? [no]: yes
```

6. 메시지가 표시되면 IdM 도메인의 NetBIOS 이름을 입력하거나 **Enter** 를 눌러 제안된 이름을 수락합니다.

```
Trust is configured but no NetBIOS domain name found, setting it now.
Enter the NetBIOS name for the IPA domain.
Only up to 15 uppercase ASCII letters, digits and dashes are allowed.
Example: EXAMPLE.

NetBIOS domain name [IDM]:
```

7. SID 생성 작업을 실행하여 기존 사용자의 SID를 생성하라는 메시지가 표시됩니다.

```
Do you want to run the ipa-sidgen task? [no]: yes
```

이는 리소스 집약적인 작업이므로 사용자가 많은 경우 한 번에 이 작업을 실행할 수 있습니다.

8. **(선택 사항)** 기본적으로 Dynamic RPC 포트 범위는 Windows Server 2008 이상에서는 **49152-65535** 로 정의됩니다. 환경에 대해 다른 Dynamic RPC 포트 범위를 정의해야 하는 경우 다른 포트를 사용하도록 Samba를 구성하고 방화벽 설정에서 해당 포트를 엽니다. 다음 예제에서는 포트 범위를 **55000-65000**으로 설정합니다.

```
[root@ipaserver ~]# net conf setparm global 'rpc server dynamic port range' 55000-65000
[root@ipaserver ~]# firewall-cmd --add-port=55000-65000/tcp
[root@ipaserver ~]# firewall-cmd --runtime-to-permanent
```

9. **ipa** 서비스를 다시 시작하십시오.

```
[root@ipaserver ~]# ipactl restart
```

10. **smbclient** 유틸리티를 사용하여 Samba가 IdM 측에서 Kerberos 인증에 응답하는지 확인합니다.

```
[root@ipaserver ~]# smbclient -L server.idm.example.com -U user_name --use-kerberos=required
```

```
lp_load_ex: changing to config backend registry
   Sharename       Type      Comment
   ---------       ----      -------
   IPC$            IPC       IPC Service (Samba 4.15.2)
...
```

2.2. GPO를 사용하여 ACTIVE DIRECTORY의 AES 암호화 유형 활성화

이 섹션에서는 그룹 정책 개체(GPO)를 사용하여 AD(Active Directory)의 AES 암호화 유형을 활성화하는 방법을 설명합니다. IdM 클라이언트에서 Samba 서버를 실행하는 등의 RHEL의 특정 기능에는 이 암호화 유형이 필요합니다.

RHEL은 더 이상 약한 DES 및 RC4 암호화 유형을 지원하지 않습니다.

사전 요구 사항

- 그룹 정책을 편집할 수 있는 사용자로 AD에 로그인되어 있습니다.
- **그룹 정책 관리 콘솔**이 컴퓨터에 설치되어 있습니다.

절차

1. **그룹 정책 관리 콘솔**을 엽니다.
2. **기본 도메인 정책**에서 마우스 오른쪽 버튼으로 클릭하여 **편집**을 선택합니다. **그룹 정책 관리 편집기**가 열립니다.
3. **컴퓨터 구성 → 정책 → Windows 설정 → 보안 설정 → 로컬 정책 → 보안 옵션**으로 이동합니다.
4. **네트워크 보안 을 두 번 클릭합니다. Kerberos 정책에 허용되는 암호화 유형을 구성합니다.**
5. **AES256_HMAC_SHA1**을 선택하고 선택적으로 **Future 암호화 유형**을 선택합니다.
6. **OK**를 클릭합니다.
7. **그룹 정책 관리 편집기** 를 닫습니다.
8. **기본 도메인 컨트롤러 정책**에 대한 단계를 반복합니다.
9. Windows 도메인 컨트롤러(DC)가 그룹 정책을 자동으로 적용할 때까지 기다립니다. 또는 DC에서 GPO를 수동으로 적용하려면 관리자 권한이 있는 계정을 사용하여 다음 명령을 입력합니다.

```
C:\> gpupdate /force /target:computer
```

2.3. IDM 클라이언트에 SAMBA 서버 설치 및 구성

이 섹션에서는 IdM 도메인에 등록된 클라이언트에 Samba를 설치하고 구성하는 방법을 설명합니다.

사전 요구 사항

- IdM 서버와 클라이언트 모두 RHEL 9.0 이상에서 실행해야 합니다.
- IdM 도메인은 도메인 멤버에 Samba를 설치하기 위해 IdM 도메인 준비에 설명된 대로 준비됩니다.

- IdM에 AD로 신뢰가 구성된 경우 Kerberos에 대해 AES 암호화 유형을 활성화합니다. 예를 들어 GPO(그룹 정책 오브젝트)를 사용하여 AES 암호화 유형을 활성화합니다. 자세한 내용은 VDDK를 사용하여 Active Directory에서 AES 암호화 활성화를 참조하십시오 .
- IdM 도메인은 도메인 멤버에 Samba를 설치하기 위해 IdM 도메인 준비에 설명된 대로 준비됩니다.
- IdM에 AD로 신뢰가 구성된 경우 Kerberos에 대해 AES 암호화 유형을 활성화합니다. 예를 들어 GPO(그룹 정책 오브젝트)를 사용하여 AES 암호화 유형을 활성화합니다. 자세한 내용은 VDDK를 사용하여 Active Directory에서 AES 암호화 활성화를 참조하십시오 .

절차

1. **ipa-client-samba** 패키지를 설치합니다.

```
[root@idm_client]# dnf install ipa-client-samba
```

2. **ipa-client-samba** 유틸리티를 사용하여 클라이언트를 준비하고 초기 Samba 구성을 생성합니다.

```
[root@idm_client]# ipa-client-samba
Searching for IPA server...
IPA server: DNS discovery
Chosen IPA master: idm_server.idm.example.com
SMB principal to be created: cifs/idm_client.idm.example.com@IDM.EXAMPLE.COM
NetBIOS name to be used: IDM_CLIENT
Discovered domains to use:

 Domain name: idm.example.com
NetBIOS name: IDM
        SID: S-1-5-21-525930803-952335037-206501584
   ID range: 212000000 - 212199999

 Domain name: ad.example.com
NetBIOS name: AD
        SID: None
   ID range: 1918400000 - 1918599999

Continue to configure the system with these values? [no]: yes
Samba domain member is configured. Please check configuration at /etc/samba/smb.conf
and start smb and winbind services
```

3. 기본적으로 **ipa-client-** 915는 사용자가 연결할 때 사용자의 홈 디렉터리를 동적으로 공유하는 **/etc/ controlPlane/dpdk.conf** 파일에 **[homes]** 섹션을 자동으로 추가합니다. 이 서버에 홈 디렉터리가 없거나 공유하려는 경우 **/etc/samba/smb.conf**에서 다음 행을 제거하십시오.

```
[homes]
   read only = no
```

4. 디렉토리와 프린터를 공유합니다. 자세한 내용은 다음 섹션을 참조하십시오.
   - POSIX ACL을 사용하는 Samba 파일 공유 설정
   - Windows ACL을 사용하는 공유 설정
   - 인쇄 서버로 Samba 설정

5. 로컬 방화벽에서 Samba 클라이언트에 필요한 포트를 엽니다.

```
[root@idm_client]# firewall-cmd --permanent --add-service=samba-client
[root@idm_client]# firewall-cmd --reload
```

6. **smb** 및 **winbind** 서비스를 활성화하고 시작합니다.

```
[root@idm_client]# systemctl enable --now smb winbind
```

검증 단계

samba-client 패키지가 설치된 다른 IdM 도메인 멤버에서 다음 확인 단계를 실행합니다.

- Kerberos 인증을 사용하여 Samba 서버의 공유를 나열합니다.

```
$ smbclient -L idm_client.idm.example.com -U user_name --use-kerberos=required
lp_load_ex: changing to config backend registry

   Sharename       Type      Comment
   ---------       ----      -------
   example         Disk
   IPC$            IPC       IPC Service (Samba 4.15.2)
...
```

추가 리소스

- **ipa-client-cnv(1)** 매뉴얼 페이지.

2.4. IDM이 새 도메인을 신뢰하는 경우 ID 매핑 구성 수동 추가

Samba에는 사용자가 리소스에 액세스하는 각 도메인에 대한 ID 매핑 구성이 필요합니다. IdM 클라이언트에서 실행 중인 기존 Samba 서버에서 관리자가 Active Directory(AD) 도메인에 새 신뢰를 추가한 후 ID 매핑 구성을 수동으로 추가해야 합니다.

사전 요구 사항

- IdM 클라이언트에 Samba가 구성되어 있습니다. 그 후 IdM에 새로운 신뢰가 추가되었습니다.
- Kerberos에 대한 DES 및 RC4 암호화 유형은 신뢰할 수 있는 AD 도메인에서 비활성화해야 합니다. 보안상의 이유로 RHEL 9는 이러한 약한 암호화 유형을 지원하지 않습니다.

절차

1. 호스트의 keytab을 사용하여 인증합니다.

```
[root@idm_client]# kinit -k
```

2. **ipa idrange-find** 명령을 사용하여 새 도메인의 기본 ID와 ID 범위 크기를 모두 표시합니다. 예를 들어 다음 명령은 **ad.example.com** 도메인의 값을 표시합니다.

```
[root@idm_client]# ipa idrange-find --name="AD.EXAMPLE.COM_id_range" --raw
----------------
1 range matched
```

```
---------------
  cn: AD.EXAMPLE.COM_id_range
  ipabaseid: 1918400000
  ipaidrangesize: 200000
  ipabaserid: 0
  ipanttrusteddomainsid: S-1-5-21-968346183-862388825-1738313271
  iparangetype: ipa-ad-trust
----------------------------
Number of entries returned 1
----------------------------
```

다음 단계에서 **ipabaseid** 및 **ipaidrangesize** 속성의 값이 필요합니다.

3. 사용 가능한 최고 ID를 계산하려면 다음 공식을 사용합니다.

```
maximum_range = ipabaseid + ipaidrangesize - 1
```

이전 단계의 값을 사용하면 **ad.example.com** 도메인에서 사용 가능한 가장 높은 ID는 **1918599999** (1918400000 + 200000 - 1)입니다.

4. **/etc/samba/smb.conf** 파일을 편집하고 도메인의 ID 매핑 구성을 **[global]** 섹션에 추가합니다.

```
idmap config AD : range = 1918400000 - 1918599999
idmap config AD : backend = sss
```

ipabaseid 속성의 값을 가장 낮은 값으로 지정하고 이전 단계에서 계산된 값을 범위의 가장 높은 값으로 지정합니다.

5. **smb** 및 **winbind** 서비스를 다시 시작합니다.

```
[root@idm_client]# systemctl restart smb winbind
```

검증 단계

- Kerberos 인증을 사용하여 Samba 서버의 공유를 나열합니다.

```
$ smbclient -L idm_client.idm.example.com -U user_name --use-kerberos=required
lp_load_ex: changing to config backend registry

   Sharename       Type      Comment
   ---------       ----      -------
   example         Disk
   IPC$            IPC       IPC Service (Samba 4.15.2)
...
```

2.5. 추가 리소스

- Identity Management 클라이언트 설치를 참조하십시오.

3장. NIS에서 IDENTITY MANAGEMENT로 마이그레이션

NIS(Network Information Service) 서버에는 사용자, 그룹, 호스트, 넷 그룹 및 자동 마운트 맵에 대한 정보가 포함될 수 있습니다. 시스템 관리자는 이러한 항목 유형, 인증 및 권한 부여를 NIS 서버에서 IdM(Identity Management) 서버로 마이그레이션하여 모든 사용자 관리 작업을 IdM 서버에서 수행할 수 있습니다. NIS에서 IdM로 마이그레이션하면 Kerberos와 같은 더 안전한 프로토콜에 액세스할 수 있습니다.

3.1. IDM에서 NIS 활성화

NIS(Identity Management) 서버 간 통신을 허용하려면 IdM 서버에서 NIS 호환성 옵션을 활성화해야 합니다.

사전 요구 사항

- IdM 서버에서 루트 액세스 권한이 있습니다.

절차

1. IdM 서버에서 NIS 리스너 및 호환성 플러그인을 활성화합니다.

```
[root@ipaserver ~]# ipa-nis-manage enable
[root@ipaserver ~]# ipa-compat-manage enable
```

2. *선택 사항*: 더 엄격한 방화벽 구성을 위해 고정 포트를 설정합니다.
예를 들어 포트를 사용하지 않는 포트 **514** 로 설정하려면 다음을 수행합니다.

```
[root@ipaserver ~]# ldapmodify -x -D 'cn=directory manager' -W
dn: cn=NIS Server,cn=plugins,cn=config
changetype: modify
add: nsslapd-pluginarg0
nsslapd-pluginarg0: 514
```

주의

다른 서비스와 충돌하지 않도록 하려면 1024 이상의 포트 번호를 사용하지 않습니다.

3. 포트 매퍼 서비스를 활성화하고 시작합니다.

```
[root@ipaserver ~]# systemctl enable rpcbind.service
[root@ipaserver ~]# systemctl start rpcbind.service
```

4. 디렉터리 서버 재시작:

```
[root@ipaserver ~]# systemctl restart dirsrv.target
```

3.2. NIS에서 IDM로 사용자 항목 마이그레이션

NIS **passwd** 맵에는 이름, UID, 기본 그룹, GECOS, 쉘 및 홈 디렉터리와 같은 사용자 정보가 포함되어 있습니다. 이 데이터를 사용하여 NIS 사용자 계정을 IdM(Identity Management)으로 마이그레이션합니다.

사전 요구 사항

- NIS 서버에 root 액세스 권한이 있습니다.
- NIS는 IdM에서 사용할 수 있습니다.
- NIS 서버가 IdM에 등록되어 있습니다.

절차

1. **yp-tools** 패키지를 설치합니다.

```
[root@nis-server ~]# dnf install yp-tools -y
```

2. NIS 서버에서 다음 콘텐츠를 사용하여 **/root/nis-users.sh** 스크립트를 만듭니다.

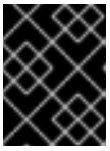
```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -d $1 -h $2 passwd > /dev/shm/nis-map.passwd 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.passwd) ; do
 IFS=' '
 username=$(echo $line | cut -f1 -d:)
 # Not collecting encrypted password because we need cleartext password
 # to create kerberos key
 uid=$(echo $line | cut -f3 -d:)
 gid=$(echo $line | cut -f4 -d:)
 gecos=$(echo $line | cut -f5 -d:)
 homedir=$(echo $line | cut -f6 -d:)
 shell=$(echo $line | cut -f7 -d:)

 # Now create this entry
 echo passw0rd1 | ipa user-add $username --first=NIS --last=USER \
     --password --gidnumber=$gid --uid=$uid --gecos='$gecos' --homedir=$homedir \
     --shell=$shell
 ipa user-show $username
done
```

3. IdM **관리자로** 인증합니다.

```
[root@nis-server ~]# kinit admin
```

4. 스크립트를 실행합니다. 예를 들어 다음과 같습니다.

```
[root@nis-server ~]# sh /root/nis-users.sh nisdomain nis-server.example.com
```



중요

이 스크립트는 첫 번째 이름, 성에 하드 코딩된 값을 사용하고 암호를 **passw0rd1** 로 설정합니다. 사용자는 다음 로그인 시 임시 암호를 변경해야 합니다.

3.3. NIS에서 IDM로 사용자 그룹 마이그레이션

NIS **그룹 맵에** 는 그룹 이름, GID 또는 그룹 멤버와 같은 그룹에 대한 정보가 포함되어 있습니다. 이 데이터를 사용하여 NIS 그룹을 IdM(Identity Management)으로 마이그레이션합니다.

사전 요구 사항

- NIS 서버에 root 액세스 권한이 있습니다.
- NIS는 IdM에서 사용할 수 있습니다.
- NIS 서버가 IdM에 등록되어 있습니다.

절차

1. **yp-tools** 패키지를 설치합니다.

```
[root@nis-server ~]# dnf install yp-tools -y
```

2. NIS 서버에서 다음 콘텐츠를 사용하여 **/root/nis-groups.sh** 스크립트를 생성합니다.

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -d $1 -h $2 group > /dev/shm/nis-map.group 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.group); do
 IFS=' '
 groupname=$(echo $line | cut -f1 -d:)
 # Not collecting encrypted password because we need cleartext password
 # to create kerberos key
 gid=$(echo $line | cut -f3 -d:)
 members=$(echo $line | cut -f4 -d:)

 # Now create this entry
 ipa group-add $groupname --desc=NIS_GROUP_$groupname --gid=$gid
 if [ -n "$members" ]; then
  ipa group-add-member $groupname --users={$members}
 fi
 ipa group-show $groupname
done
```

3. IdM **관리자로** 인증합니다.

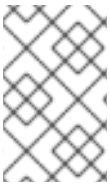
```
[root@nis-server ~]# kinit admin
```

4. 스크립트를 실행합니다. 예를 들어 다음과 같습니다.

```
[root@nis-server ~]# sh /root/nis-groups.sh nisdomain nis-server.example.com
```

3.4. NIS에서 IDM로 호스트 항목 마이그레이션

NIS **호스트 맵에** 는 호스트 이름 및 IP 주소와 같은 호스트에 대한 정보가 포함되어 있습니다. 이 데이터를 사용하여 NIS 호스트 항목을 IdM(Identity Management)으로 마이그레이션합니다.



참고

IdM에서 호스트 그룹을 생성하면 해당 shadow NIS 그룹이 자동으로 생성됩니다. 이러한 shadow NIS 그룹에서 **ipa netgroup-*** 명령을 사용하지 마십시오. **ipa netgroup-*** 명령만 사용하여 **netgroup-add** 명령을 통해 생성된 기본 netgroups를 관리합니다.

사전 요구 사항

- NIS 서버에 root 액세스 권한이 있습니다.
- NIS는 IdM에서 사용할 수 있습니다.
- NIS 서버가 IdM에 등록되어 있습니다.

절차

1. **yp-tools** 패키지를 설치합니다.

```
[root@nis-server ~]# dnf install yp-tools -y
```

2. NIS 서버에서 다음 콘텐츠를 사용하여 **/root/nis-hosts.sh** 스크립트를 만듭니다.

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -d $1 -h $2 hosts | egrep -v "localhost|127.0.0.1" > /dev/shm/nis-map.hosts 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.hosts); do
 IFS=' '
 ipaddress=$(echo $line | awk '{print $1}')
 hostname=$(echo $line | awk '{print $2}')
 master=$(ipa env xmlrpc_uri | tr -d '[:space:]' | cut -f3 -d: | cut -f3 -d/)
 domain=$(ipa env domain | tr -d '[:space:]' | cut -f2 -d:)
 if [ $(echo $hostname | grep "\." |wc -l) -eq 0 ] ; then
  hostname=$(echo $hostname.$domain)
 fi
 zone=$(echo $hostname | cut -f2- -d.)
 if [ $(ipa dnszone-show $zone 2>/dev/null | wc -l) -eq 0 ] ; then
  ipa dnszone-add --name-server=$master --admin-email=root.$master
 fi
 ptrzone=$(echo $ipaddress | awk -F. '{print $3 "." $2 "." $1 ".in-addr.arpa."}')
 if [ $(ipa dnszone-show $ptrzone 2>/dev/null | wc -l) -eq 0 ] ; then
  ipa dnszone-add  $ptrzone --name-server=$master --admin-email=root.$master
 fi
 # Now create this entry
 ipa host-add $hostname --ip-address=$ipaddress
 ipa host-show $hostname
done
```

3. IdM **관리자로** 인증합니다.

```
[root@nis-server ~]# kinit admin
```

4. 스크립트를 실행합니다. 예를 들어 다음과 같습니다.

```
[root@nis-server ~]# sh /root/nis-hosts.sh nisdomain nis-server.example.com
```

참고

이 스크립트는 별칭과 같은 특수 호스트 구성을 마이그레이션하지 않습니다.

```
:_content-type: PROCEDURE
// Module included in the following assemblies:
//
// assembly_migrating-from-nis-to-identity-management.adoc
```

3.5. NIS에서 IDM로 NETGROUP 항목 마이그레이션

NIS **netgroup** map에는 netgroups에 대한 정보가 포함되어 있습니다. 이 데이터를 사용하여 NIS netgroups를 IdM(Identity Management)으로 마이그레이션합니다.

사전 요구 사항

- NIS 서버에 root 액세스 권한이 있습니다.
- NIS는 IdM에서 사용할 수 있습니다.
- NIS 서버가 IdM에 등록되어 있습니다.

절차

1. **yp-tools** 패키지를 설치합니다.

```
[root@nis-server ~]# dnf install yp-tools -y
```

2. NIS 서버에서 다음 콘텐츠를 사용하여 **/root/nis-netgroups.sh** 스크립트를 만듭니다.

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -k -d $1 -h $2 netgroup > /dev/shm/nis-map.netgroup 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.netgroup); do
 IFS=' '
 netgroupname=$(echo $line | awk '{print $1}')
 triples=$(echo $line | sed "s/^$netgroupname //")
 echo "ipa netgroup-add $netgroupname --desc=NIS_NG_$netgroupname"
 if [ $(echo $line | grep "(," | wc -l) -gt 0 ]; then
  echo "ipa netgroup-mod $netgroupname --hostcat=all"
 fi
 if [ $(echo $line | grep ",," | wc -l) -gt 0 ]; then
```

```
 echo "ipa netgroup-mod $netgroupname --usercat=all"
fi

for triple in $triples; do
 triple=$(echo $triple | sed -e 's/-//g' -e 's/(//' -e 's/)//')
 if [ $(echo $triple | grep ",.*," | wc -l) -gt 0 ]; then
  hostname=$(echo $triple | cut -f1 -d,)
  username=$(echo $triple | cut -f2 -d,)
  domain=$(echo $triple | cut -f3 -d,)
  hosts=""; users=""; doms="";
  [ -n "$hostname" ] && hosts="--hosts=$hostname"
  [ -n "$username" ] && users="--users=$username"
  [ -n "$domain"   ] && doms="--nisdomain=$domain"
  echo "ipa netgroup-add-member $netgroup $hosts $users $doms"
 else
  netgroup=$triple
  echo "ipa netgroup-add $netgroup --desc=<NIS_NG>_$netgroup"
 fi
done
done
```

3. IdM **관리자로** 인증합니다.

```
[root@nis-server ~]# kinit admin
```

4. 스크립트를 실행합니다. 예를 들어 다음과 같습니다.

```
[root@nis-server ~]# sh /root/nis-netgroups.sh nisdomain nis-server.example.com
```

3.6. NIS에서 IDM로 자동 마운트 맵 마이그레이션

자동 마운트 맵은 위치(상위 항목), 관련 키 및 맵을 정의하는 일련의 중첩 및 상호 관련 항목입니다. NIS 자동 마운트를 마이그레이션하려면 IdM(Identity Management)에 매핑합니다.

사전 요구 사항

- NIS 서버에 root 액세스 권한이 있습니다.
- NIS는 IdM에서 사용할 수 있습니다.
- NIS 서버가 IdM에 등록되어 있습니다.

절차

1. **yp-tools** 패키지를 설치합니다.

```
[root@nis-server ~]# dnf install yp-tools -y
```

2. NIS 서버에서 다음 콘텐츠를 사용하여 **/root/nis-automounts.sh** 스크립트를 만듭니다.

```
#!/bin/sh
# $1 is for the automount entry in ipa

ipa automountlocation-add $1
```

```
# $2 is the NIS domain, $3 is the NIS master server, $4 is the map name

ypcat -k -d $2 -h $3 $4 > /dev/shm/nis-map.$4 2>&1

ipa automountmap-add $1 $4

basedn=$(ipa env basedn | tr -d '[:space:]' | cut -f2 -d:)
cat > /tmp/amap.ldif <<EOF
dn: nis-domain=$2+nis-map=$4,cn=NIS Server,cn=plugins,cn=config
objectClass: extensibleObject
nis-domain: $2
nis-map: $4
nis-base: automountmapname=$4,cn=$1,cn=automount,$basedn
nis-filter: (objectclass=*)
nis-key-format: %{automountKey}
nis-value-format: %{automountInformation}
EOF
ldapadd -x -h $3 -D "cn=Directory Manager" -W -f /tmp/amap.ldif

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.$4); do
 IFS=" "
 key=$(echo "$line" | awk '{print $1}')
 info=$(echo "$line" | sed -e "s^$key[ \t]*")
 ipa automountkey-add nis $4 --key="$key" --info="$info"
done
```

참고

이 스크립트는 NIS 자동 마운트 정보를 내보내고 자동 마운트 위치 및 관련 맵을 위해 LDAP 데이터 상호 변경 형식(LDIF)을 생성하고, LDIF 파일을 IdM 디렉터리 서버로 가져옵니다.

3. IdM **관리자로** 인증합니다.

```
[root@nis-server ~]# kinit admin
```

4. 스크립트를 실행합니다. 예를 들어 다음과 같습니다.

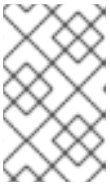
```
[root@nis-server ~]# sh /root/nis-automounts.sh location nisdomain
    nis-server.example.com map_name
```

4장. IDM의 자동 마운트 사용

자동 마운트는 여러 시스템에서 디렉토리를 관리, 구성 및 액세스하는 방법입니다. 자동 마운트는 요청이 있을 때마다 디렉토리를 자동으로 마운트합니다. IdM(Identity Management) 도메인 내에서는 도메인 내 클라이언트의 디렉터리를 쉽게 공유할 수 있으므로 이 도메인은 잘 작동합니다.

예에서는 다음 시나리오를 사용합니다.

- **NFS-server.idm.example.com** 은 NFS(네트워크 파일 시스템) 서버의 정규화된 도메인 이름(FQDN)입니다.
- 간단히 하기 위해 **nfs-server.idm.example.com** 은 **raleigh** 자동 마운트 위치에 대한 맵을 제공하는 IdM 클라이언트입니다.



참고

자동 마운트 위치는 고유한 NFS 맵 세트입니다. 이상적으로 이러한 맵은 모두 동일한 지역에 있으므로 예를 들어 클라이언트가 빠른 연결로 이점을 얻을 수 있지만 필수는 아닙니다.

- NFS 서버는 **/exports/project** 디렉터리를 읽기-쓰기로 내보냅니다.
- **developers** 그룹에 속하는 IdM 사용자는 **raleigh** mount 위치를 사용하는 IdM 클라이언트의 **/devel/project/** 로 내보낸 디렉터리의 콘텐츠에 액세스할 수 있습니다.
- IdM **-client.idm.example.com** 은 **raleigh** 자동 마운트 위치를 사용하는 IdM 클라이언트입니다.

중요

NFS 서버 대신 Samba 서버를 사용하여 IdM 클라이언트의 공유를 제공하려면 IPA 환경의 Autofs로 kerberized CIFS 마운트를 구성하려면 How do the How do I configure kerberized CIFS mount with Autofs in an IPA environment? KCS 솔루션.

4.1. IDM의 NETNAMESPACE 및 자동 마운트

Net **Namespace** 서비스는 **자동 마운트** 데몬을 통해 액세스 시 디렉토리를 마운트하도록 하여 필요에 따라 디렉터리 마운트를 자동화합니다. 또한, disabled에서는 period of inactivity을 통해 **자동 마운트** 된 디렉토리를 마운트 해제하도록 mount에 지시합니다. 정적 마운트와 달리 온디맨드 마운트는 시스템 리소스를 절약합니다.

자동 마운트 맵

NetNamespace 를 사용하는 **시스템에서는 자동 마운트** 설정이 여러 다른 파일에 저장됩니다. 기본 **자동 마운트** 설정 파일은 자동 마운트 마운트 지점의 마스터 매핑과 시스템의 관련 리소스가 포함된 **/etc/auto.master** 입니다. **이 매핑을 *자동 마운트 맵* 이라고 합니다.**

/etc/auto.master 구성 파일에는 *마스터 맵이* 포함되어 있습니다. 다른 맵에 대한 참조를 포함할 수 있습니다. 이러한 맵은 직접 또는 간접일 수 있습니다. 직접 맵은 마운트 지점에 절대 경로 이름을 사용하지만 간접 맵은 상대 경로 이름을 사용합니다.

IdM의 자동 마운트 설정

자동 마운트 는 일반적으로 로컬 **/etc/auto.master** 및 관련 파일에서 맵 데이터를 검색하지만 다른 소스에서 맵 데이터를 검색할 수도 있습니다. 일반적인 소스 중 하나는 **LDAP** 서버입니다. **IdM(Identity Management)**의 컨텍스트에서는 **389 Directory Server**입니다.

NetNamespace를 사용하는 시스템이 **IdM** 도메인의 클라이언트이면 자동 마운트 구성이 로컬 구성 파일에 저장되지 않습니다. 대신 맵, 위치 및 키와 같은 **NetNamespace** 구성이 **IdM** 디렉터리에 **LDAP** 항목으로 저장됩니다. 예를 들어 **idm.example.com IdM** 도메인의 경우 기본 *마스터 맵* 은 다음과 같이 저장됩니다.

```
dn:
automountmapname=auto.master,cn=default,cn=automount,dc=idm,dc=example,dc=com
objectClass: automountMap
objectClass: top
automountMapName: auto.master
```

추가 리소스

- 필요에 따라 파일 시스템 마운트

4.2. NFS 서버의 IDM 키탭 구성

다른 **IdM(Identity Management)** 클라이언트에 로그인한 사용자가 이 **NFS** 서버의 디렉터리 및 파일에 액세스할 수 있도록 **Kerberos** 인식 **NFS** 서버를 구성합니다.

이 예제에서는 **nfs-server.idm.example.com** 에서 실행되는 **NFS** 서비스를 구성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자로 로그인되어 있습니다.
- **NFS** 서버에 대한 루트 액세스 권한이 있습니다.
- **NFS** 공유를 내보내는 데 필요한 패키지가 설치되어 있습니다.
- [선택 사항] 방화벽 뒤에서 실행되도록 **NFS** 서버를 구성했습니다.

절차

1. IdM 지원 호스트에서 IdM에 NFS 서비스를 추가합니다.

```
$ ipa service-add nfs/nfs-server.idm.example.com
---------------------------------------------------------
Added service "nfs/nfs-server.idm.example.com@IDM.EXAMPLE.COM"
---------------------------------------------------------
  Principal name: nfs/nfs-server.idm.example.com@IDM.EXAMPLE.COM
  Principal alias: nfs/nfs-server.idm.example.com@IDM.EXAMPLE.COM
  Managed by: nfs-server.idm.example.com
```

2. NFS 서버에서 NFS 서비스의 keytab을 가져옵니다.

```
# ipa-getkeytab -p nfs/nfs-server.idm.example.com -k /etc/krb5.keytab
Keytab successfully retrieved and stored in: /etc/krb5.keytab
```

3. NFS 서버에서 NFS 서비스를 다시 시작합니다.

```
# systemctl restart nfs-server
```

4. NFS 서버에서 NFS 서비스를 활성화합니다.

```
# systemctl enable nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service →
/usr/lib/systemd/system/nfs-server.service.
```

4.3. IDM에서 NFS 공유 내보내기

IdM(Identity Management) 시스템 관리자는 NFS 서버를 사용하여 네트워크를 통해 IdM 사용자와 디렉터리를 공유할 수 있습니다.

사전 요구 사항

- IdM 관리자로 로그인되어 있습니다.
- NFS 서버에 대한 루트 액세스 권한이 있습니다.

- NFS 공유를 내보내는 데 필요한 패키지가 설치되어 있습니다.
- [선택 사항] 방화벽 뒤에서 실행되도록 NFS 서버를 구성했습니다.

절차

1. 내보낼 디렉터리를 생성합니다.

```
# mkdir -p /exports/project
```

2. 소유자와 그룹에 디렉토리를 읽고 쓰고 실행할 권한을 부여합니다.

```
# chmod 770 /exports/project
```

3. 디렉토리에 생성된 모든 파일에 디렉터리 소유자의 그룹 소유권이 설정되도록 GSID Sticky bit를 추가합니다.

```
# chmod g+s /exports_ro/documentation /exports/project
```

4. 멤버가 디렉터리에 액세스할 수 있는 IdM 그룹을 생성합니다. IdM 그룹의 예는 developers 입니다.

```
# ipa group-add developers
```

5. 그룹의 모든 IdM 사용자가 액세스할 수 있도록 /exports/project 디렉터리의 그룹 소유권을 developers 로 변경합니다.

```
# chgrp developers /exports/project
```

6. IdM 사용자를 그룹에 추가합니다. 예제 사용자는 idm_user 입니다.

```
# ipa group-add-member developers --users=idm_user
```

7. 디렉터리에 몇 가지 내용이 포함된 파일을 생성합니다.

```
# echo "this is a read-write file" > /exports/project/rw_file
```

8. /etc/exports.d/ 디렉토리의 파일에 다음 정보를 추가합니다.

- 내보낼 디렉터리
- 사용자가 디렉터리의 파일에 액세스하기 위해 인증하는 방법
- 디렉토리의 파일에 대해 사용자에게 부여하려는 권한은 무엇입니까?

```
# echo "/exports/project *(sec=krb5p,rw)" > /etc/exports.d/project.exports
```

glusterfs=krb5 는 로컬 UNIX UID 및 GID 대신 Kerberos V5 프로토콜을 사용하여 사용자를 인증합니다.

또는rb 5i 또는 sa =krb5p 를 사용합니다.

sec=krb5i

사용자 인증에 Kerberos V5를 사용하고 데이터 변조를 방지하기 위해 보안 체크섬을 사용하여 NFS 작업의 무결성 검사를 수행합니다.

sec=krb5p

Kerberos V5를 사용하여 사용자 인증, 무결성 검사 및 NFS 트래픽을 암호화하여 트래픽 스니핑을 방지합니다. 이는 가장 안전한 설정이지만 성능 오버헤드가 가장 많습니다.

9. 모든 디렉터리를 다시 내보내 마스터 내보내기 테이블을 /var/lib/nfs/etab 과 /etc/exports 및 /etc/exports.d:

```
# exportfs -r
```

10. /etc/exports 에 적합한 현재 내보내기 목록을 표시합니다.

```
# exportfs -s
/exports/project  *
(sync,wdelay,hide,no_subtree_check,sec=krb5p,rw,secure,root_squash,no_all_squas
```

```
h)
```

추가 리소스

- NetNamespace 5 방법에 대한 자세한 내용은 nfs man 페이지를 참조하십시오.

4.4. IDM CLI를 사용하여 IDM에서 자동 마운트 위치 및 맵 구성

위치는 맵 세트이며, 모두 auto.master 에 저장됩니다. 하나의 위치는 여러 맵을 저장할 수 있습니다. 위치 항목은 맵 항목용 컨테이너로만 작동합니다. 이 항목은 및 의 자동 마운트 설정이 아닙니다.

IdM(Identity Management)의 시스템 관리자는 IdM에서 자동 마운트 위치와 맵을 구성하여 지정된 위치의 IdM 사용자가 호스트의 특정 마운트 지점으로 이동하여 NFS 서버에서 내보낸 공유에 액세스할 수 있습니다. 내보낸 NFS 서버 디렉터리와 마운트 지점은 모두 맵에 지정됩니다. 이 예제에서는 raleigh 위치와 IdM 클라이언트의 /devel/ 마운트 지점에 nfs-server.idm.example.com:/exports/project 공유를 읽기-쓰기 디렉터리로 마운트하는 맵을 구성하는 방법을 설명합니다.

사전 요구 사항

- IdM 지원 호스트에 IdM 관리자로 로그인되어 있습니다.

절차

1. raleigh Autoscaler 위치를 만듭니다.

```
$ ipa automountlocation-add raleigh
--------------------------------
Added automount location "raleigh"
--------------------------------
  Location: raleigh
```

2. raleigh 위치에 auto.devel 자동 자동 마운트 맵을 만듭니다.

```
$ ipa automountmap-add raleigh auto.devel
-------------------------------
Added automount map "auto.devel"
-------------------------------
  Map: auto.devel
```

3. exports/ share에 대한 키 및 마운트 정보를 추가합니다.

a. auto.devel 맵의 키 및 마운트 정보를 추가합니다.

```
$ ipa automountkey-add raleigh auto.devel --key='*' --info='-sec=krb5p,vers=4 nfs-server.idm.example.com:/exports/&'
----------------------
Added automount key "*"
----------------------
  Key: *
  Mount information: -sec=krb5p,vers=4 nfs-server.idm.example.com:/exports/&
```

b. auto.master 맵에 대한 키 및 마운트 정보를 추가합니다.

```
$ ipa automountkey-add raleigh auto.master --key=/devel --info=auto.devel
---------------------------
Added automount key "/devel"
---------------------------
  Key: /devel
  Mount information: auto.devel
```

4.5. IDM 클라이언트에서 자동 마운트 구성

IdM(Identity Management) 시스템 관리자는 사용자가 클라이언트에 로그인할 때 클라이언트가 추가된 위치에 NFS 공유에 액세스할 수 있도록 IdM 클라이언트의 자동 자동 마운트 서비스를 구성할 수 있습니다. 이 예제에서는 raleigh 위치에서 사용할 수 있는 자동 마운트 서비스를 사용하도록 IdM 클라이언트를 구성하는 방법을 설명합니다.

사전 요구 사항

- IdM 클라이언트에 대한 루트 액세스 권한이 있습니다.
- IdM 관리자로 로그인되어 있습니다.
- 자동 마운트 위치가 있습니다. 예를 들어 location은 raleigh 입니다.

절차

1. IdM 클라이언트에서 ipa-client-automount 명령을 입력하고 위치를 지정합니다. -U 옵션을 사용하여 축소 스크립트를 실행합니다.

```
# ipa-client-automount --location raleigh -U
```

2. NetNamespace 서비스를 중지하고 SSSD 캐시를 지우고 NetNamespace 서비스를 시작하여 새 구성 설정을 로드합니다.

```
# systemctl stop autofs ; sss_cache -E ; systemctl start autofs
```

4.6. IDM 사용자가 IDM 클라이언트의 NFS 공유에 액세스할 수 있는지 확인

IdM(Identity Management) 시스템 관리자는 특정 그룹의 멤버인 IdM 사용자가 특정 IdM 클라이언트에 로그인할 때 NFS 공유에 액세스할 수 있는지 테스트할 수 있습니다.

이 예제에서는 다음 시나리오를 테스트합니다.

- developers 그룹에 속하는 idm_user 라는 IdM 사용자는 /devel/project 디렉터리에 있는 idm-client.idm.example.com 에 자동 마운트된 파일의 내용을 읽고 쓸 수 있습니다.

사전 요구 사항

- NFS 서버의 IdM 키탭을 구성하고 NFS 공유를 내보냈습니다.
- IdM 사용자가 NFS 공유에 액세스할 수 있는 방법을 구성한 IdM에 자동 마운트 위치, 맵, 마운트 지점을 구성했습니다.
- IdM 클라이언트에 자동 마운트를 구성했습니다.

절차

1. IdM 사용자가 읽기-쓰기 디렉터리에 액세스할 수 있는지 확인합니다.

   a. IdM 사용자로 IdM 클라이언트에 연결합니다.

   ```
   $ ssh idm_user@idm-client.idm.example.com
   Password:
   ```

   b. IdM 사용자의 티켓 통합 티켓 (TGT)을 받으십시오.

```
$ kinit idm_user
```

c. [선택 사항] IdM 사용자의 그룹 멤버십을 확인합니다.

```
$ ipa user-show idm_user
  User login: idm_user
  [...]
  Member of groups: developers, ipausers
```

d. /devel/project 디렉터리로 이동합니다.

```
$ cd /devel/project
```

e. 디렉터리 콘텐츠를 나열합니다.

```
$ ls
rw_file
```

f. 디렉터리의 파일에 행을 추가하여 쓰기 권한을 테스트합니다.

```
$ echo "idm_user can write into the file" > rw_file
```

g. [선택 사항] 파일의 업데이트된 콘텐츠를 봅니다.

```
$ cat rw_file
this is a read-write file
idm_user can write into the file
```

출력에서 idm_user 가 파일에 쓸 수 있는지 확인합니다.

5장. ANSIBLE을 사용하여 IDM 사용자의 NFS 공유 자동 마운트

자동 마운트는 여러 시스템에서 디렉토리를 관리, 구성 및 액세스하는 방법입니다. 자동 마운트는 요청이 있을 때마다 디렉토리를 자동으로 마운트합니다. IdM(Identity Management) 도메인 내에서는 도메인 내 클라이언트의 디렉터리를 쉽게 공유할 수 있으므로 이 도메인은 잘 작동합니다.

IdM 사용자가 IdM 위치에 IdM 클라이언트에 로그인하기 위해 Ansible을 사용하여 자동으로 마운트되도록 NFS 공유를 구성할 수 있습니다.

이 장의 예제에서는 다음 시나리오를 사용합니다.

- NFS-server.idm.example.com 은 NFS(네트워크 파일 시스템) 서버의 정규화된 도메인 이름(FQDN)입니다.
- nfs-server.idm.example.com 은 raleigh 자동 마운트 위치에 있는 IdM 클라이언트입니다.
- NFS 서버는 /exports/project 디렉터리를 읽기-쓰기로 내보냅니다.
- developers 그룹에 속하는 IdM 사용자는 NFS 서버와 동일한 raleigh 자동 마운트 위치에 있는 IdM 클라이언트의 /devel/project/ 로 내보낸 디렉터리의 콘텐츠에 액세스할 수 있습니다.
- IdM -client.idm.example.com 은 raleigh 자동 마운트 위치에 있는 IdM 클라이언트입니다.

중요

NFS 서버 대신 Samba 서버를 사용하여 IdM 클라이언트의 공유를 제공하려면 IPA 환경의 Autofs로 kerberized CIFS 마운트를 구성하려면 How do the How do I configure kerberized CIFS mount with Autofs in an IPA environment? KCS 솔루션.

이 장에는 다음 섹션이 포함되어 있습니다.

1. IdM의 NetNamespace 및 자동 마운트
2. NFS 서버의 IdM 키탭 구성
3. IdM에서 NFS 공유 내보내기
4. IdM 관리를 위해 Ansible 제어 노드 준비
5. Ansible을 사용하여 IdM에서 자동 마운트 위치, 맵 및 키 구성
6. Ansible을 사용하여 NFS 공유를 소유하는 그룹에 IdM 사용자 추가
7. Ansible을 사용하여 자동 마운트 해제 위치에 IdM 클라이언트 추가
8. IdM 클라이언트에서 자동 마운트 구성
9. IdM 사용자가 IdM 클라이언트의 NFS 공유에 액세스할 수 있는지 확인

5.1. IDM의 NETNAMESPACE 및 자동 마운트

Net Namespace 서비스는 자동 마운트 데몬을 통해 액세스 시 디렉토리를 마운트하도록 하여 필요에 따라 디렉터리 마운트를 자동화합니다. 또한, **disabled**에서는 **period of inactivity**을 통해 자동 마운트 된 디렉토리를 마운트 해제하도록 **mount**에 지시합니다. 정적 마운트와 달리 온디맨드 마운트는 시스템 리소스를 절약합니다.

자동 마운트 맵

NetNamespace 를 사용하는 시스템에서는 자동 마운트 설정이 여러 다른 파일에 저장됩니다. 기본 자동 마운트 설정 파일은 자동 마운트 마운트 지점의 마스터 매핑과 시스템의 관련 리소스가 포함된 **/etc/auto.master** 입니다. 이 매핑을 *자동 마운트 맵* 이라고 합니다.

/etc/auto.master 구성 파일에는 *마스터 맵이* 포함되어 있습니다. 다른 맵에 대한 참조를 포함할 수 있습니다. 이러한 맵은 직접 또는 간접일 수 있습니다. 직접 맵은 마운트 지점에 절대 경로 이름을 사용하지만 간접 맵은 상대 경로 이름을 사용합니다.

IdM의 자동 마운트 설정

자동 마운트 는 일반적으로 로컬 **/etc/auto.master** 및 관련 파일에서 맵 데이터를 검색하지만 다른 소스에서 맵 데이터를 검색할 수도 있습니다. 일반적인 소스 중 하나는 **LDAP** 서버입니다. **IdM(Identity Management)**의 컨텍스트에서는 **389 Directory Server**입니다.

NetNamespace를 사용하는 시스템이 **IdM** 도메인의 클라이언트이면 자동 마운트 구성이 로컬 구성 파일에 저장되지 않습니다. 대신 맵, 위치 및 키와 같은 **NetNamespace** 구성이 **IdM** 디렉터리에 **LDAP** 항목으로 저장됩니다. 예를 들어 **idm.example.com IdM** 도메인의 경우 기본 *마스터 맵* 은 다음과 같이 저장됩니다.

```
dn:
automountmapname=auto.master,cn=default,cn=automount,dc=idm,dc=example,dc=com
objectClass: automountMap
objectClass: top
automountMapName: auto.master
```

추가 리소스

- 필요에 따라 파일 시스템 마운트

5.2. NFS 서버의 IDM 키탭 구성

다른 **IdM(Identity Management)** 클라이언트에 로그인한 사용자가 이 **NFS** 서버의 디렉터리 및 파일에 액세스할 수 있도록 **Kerberos** 인식 **NFS** 서버를 구성합니다.

이 예제에서는 **nfs-server.idm.example.com** 에서 실행되는 **NFS** 서비스를 구성하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자로 로그인되어 있습니다.
- **NFS** 서버에 대한 루트 액세스 권한이 있습니다.
- NFS 공유를 내보내는 데 필요한 패키지가 설치되어 있습니다.

- [선택 사항] 방화벽 뒤에서 실행되도록 NFS 서버를 구성했습니다.

절차

1. IdM 지원 호스트에서 IdM에 NFS 서비스를 추가합니다.

```
$ ipa service-add nfs/nfs-server.idm.example.com
---------------------------------------------------------
Added service "nfs/nfs-server.idm.example.com@IDM.EXAMPLE.COM"
---------------------------------------------------------
  Principal name: nfs/nfs-server.idm.example.com@IDM.EXAMPLE.COM
  Principal alias: nfs/nfs-server.idm.example.com@IDM.EXAMPLE.COM
  Managed by: nfs-server.idm.example.com
```

2. NFS 서버에서 NFS 서비스의 keytab을 가져옵니다.

```
# ipa-getkeytab -p nfs/nfs-server.idm.example.com -k /etc/krb5.keytab
Keytab successfully retrieved and stored in: /etc/krb5.keytab
```

3. NFS 서버에서 NFS 서비스를 다시 시작합니다.

```
# systemctl restart nfs-server
```

4. NFS 서버에서 NFS 서비스를 활성화합니다.

```
# systemctl enable nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service →
/usr/lib/systemd/system/nfs-server.service.
```

5.3. IDM에서 NFS 공유 내보내기

IdM(Identity Management) 시스템 관리자는 NFS 서버를 사용하여 네트워크를 통해 IdM 사용자와 디렉터리를 공유할 수 있습니다.

사전 요구 사항

- IdM 관리자로 로그인되어 있습니다.

- NFS 서버에 대한 루트 액세스 권한이 있습니다.
- NFS 공유를 내보내는 데 필요한 패키지가 설치되어 있습니다.
- [선택 사항] 방화벽 뒤에서 실행되도록 NFS 서버를 구성했습니다.

절차

1. 내보낼 디렉터리를 생성합니다.

```
# mkdir -p /exports/project
```

2. 소유자와 그룹에 디렉토리를 읽고 쓰고 실행할 권한을 부여합니다.

```
# chmod 770 /exports/project
```

3. 디렉토리에 생성된 모든 파일에 디렉터리 소유자의 그룹 소유권이 설정되도록 GSID Sticky bit를 추가합니다.

```
# chmod g+s /exports_ro/documentation /exports/project
```

4. 디렉터리에 몇 가지 내용이 포함된 파일을 생성합니다.

```
# echo "this is a read-write file" > /exports/project/rw_file
```

5. /etc/exports.d/ 디렉토리의 파일에 다음 정보를 추가합니다.
   - 내보낼 디렉터리
   - 사용자가 디렉터리의 파일에 액세스하기 위해 인증하는 방법
   - 디렉토리의 파일에 대해 사용자에게 부여하려는 권한은 무엇입니까?

```
# echo "/exports/project *(sec=krb5p,rw)" > /etc/exports.d/project.exports
```

glusterfs=krb5 는 로컬 UNIX UID 및 GID 대신 Kerberos V5 프로토콜을 사용하여 사용자를 인증합니다.

또는rb 5i 또는 sa =krb5p 를 사용합니다.

sec=krb5i

사용자 인증에 Kerberos V5를 사용하고 데이터 변조를 방지하기 위해 보안 체크섬을 사용하여 NFS 작업의 무결성 검사를 수행합니다.

sec=krb5p

Kerberos V5를 사용하여 사용자 인증, 무결성 검사 및 NFS 트래픽을 암호화하여 트래픽 스니핑을 방지합니다. 이는 가장 안전한 설정이지만 성능 오버헤드가 가장 많습니다.

6. 모든 디렉터리를 다시 내보내 마스터 내보내기 테이블을 /var/lib/nfs/etab 과 /etc/exports 및 /etc/exports.d:

```
# exportfs -r
```

7. /etc/exports 에 적합한 현재 내보내기 목록을 표시합니다.

```
# exportfs -s
/exports/project  *
(sync,wdelay,hide,no_subtree_check,sec=krb5p,rw,secure,root_squash,no_all_squash)
```

추가 리소스

- NetNamespace 5 방법에 대한 자세한 내용은 nfs man 페이지를 참조하십시오.

5.4. IDM 관리를 위해 ANSIBLE 제어 노드 준비

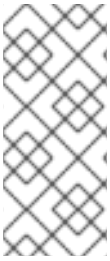
Red Hat Ansible Engine으로 작업할 때 IdM(Identity Management)을 관리하는 시스템 관리자는 다음을 수행하는 것이 좋습니다.

- 홈 디렉터리에서 **Ansible** 플레이북 전용 하위 디렉터리를 생성합니다(예: **~/MyPlaybooks**).

- **/usr/share/doc/ansible-freeipa/*** 및 **/usr/share/doc/rhel-system-roles/*** 디렉터리 및 하위 디렉터리에서 **~/MyPlaybooks** 디렉터리에 복사 및 조정.

- 인벤토리 파일을 **~/MyPlaybook** 디렉터리에 포함합니다.

이 방법을 따라 모든 플레이북을 한 곳에서 찾을 수 있으며 루트 권한을 호출하지 않고 플레이북을 실행할 수 있습니다.



참고

ipaserver,ipareplica,ipaclient 및 **ipabackup ansible-freeipa** 역할을 실행하려면 관리형 노드의 루트 권한만 있으면 됩니다. 이러한 역할에는 디렉터리 및 **dnf** 소프트웨어 패키지 관리자에 대한 권한이 있어야 합니다.

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

사전 요구 사항

- 관리형 노드 ***server.idm.example.com*** 및 ***replica.idm.example.com*** 에 **IdM** 서버를 설치했습니다.

- 제어 노드에서 직접 관리형 노드, ***server.idm.example.com*** 및 ***replica.idm.example.com*** 에 로그인할 수 있도록 **DNS** 및 네트워킹을 구성했습니다.

- **IdM** 관리자 암호를 알고 있습니다.

절차

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

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

2. ~/MyPlaybooks/ 디렉터리로 변경합니다.

```
$ cd ~/MyPlaybooks
```

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

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

[privilege_escalation]
become=True
```

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

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

이 구성은 이러한 위치에 있는 호스트에 대해 eu 와 us 이라는 두 개의 호스트 그룹을 정의합니다. 또한 이 구성은 eu 및 us 그룹의 모든 호스트를 포함하는 ipaserver 호스트 그룹을 정의합니다.

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

```
$ ssh-keygen
```

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

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

이러한 명령을 입력할 때 IdM 관리자 암호를 입력해야 합니다.

추가 리소스

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

5.5. ANSIBLE을 사용하여 IDM에서 자동 마운트 위치, 맵 및 키 구성

IdM(Identity Management) 시스템 관리자는 IdM에서 자동 마운트 위치 및 맵을 구성하여 지정된 위치의 IdM 사용자가 호스트의 특정 마운트 지점으로 이동하여 NFS 서버에서 내보낸 공유에 액세스할 수 있습니다. 내보낸 NFS 서버 디렉터리와 마운트 지점은 모두 맵에 지정됩니다. LDAP 용어에서 위치는 이러한 맵 항목의 컨테이너입니다.

이 예제에서는 Ansible을 사용하여 raleigh 위치를 구성하는 방법과 IdM 클라이언트의 /devel/project 마운트 지점에 nfs-server.idm.example.com:/exports/project 공유를 읽기-쓰기 디렉터리로 마운트하는 맵을 구성하는 방법을 설명합니다.

사전 요구 사항

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

절차

1. Ansible 제어 노드에서 ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. /usr/share/doc/ansible -freeipa/playbooks/automount/ 디렉터리에 있는mount-location-present.yml Ansible 플레이북 파일을 복사합니다.

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/automount/automount-location-present.yml automount-location-map-and-key-present.yml
```

3. 편집을 위해 autoscale-location-map-and-key-present.yml 파일을 엽니다.

4. ipaautomountlocation 작업 섹션에서 다음 변수를 설정하여 파일을 조정합니다.

- ipaadmin_password 변수를 IdM 관리자 의 암호로 설정합니다.
- name 변수를 raleigh 로 설정합니다.
- state 변수가 present 로 설정되어 있는지 확인합니다.

이는 현재 예제에서 수정된 Ansible 플레이북 파일입니다.

```
---
- name: Automount location present example
  hosts: ipaserver
  become: true
  tasks:
  - name: Ensure automount location is present
    ipaautomountlocation:
      ipaadmin_password: Secret123
      name: raleigh
      state: present
```

5. autoscaling -location-map-and-key-present.yml 파일을 계속 편집합니다.

a. tasks 섹션에서 자동 마운트 맵이 있는지 확인하는 작업을 추가합니다.

```
[...]
  tasks:
[...]
  - name: ensure map named auto.devel in location raleigh is created
    ipaautomountmap:
      ipaadmin_password: Secret123
```

```
    name: auto.devel
    location: raleigh
    state: present
```

b. **tasks** 섹션에서 마운트 지점 및 **NFS** 서버 정보를 맵에 추가하는 작업을 추가합니다.

```
[...]
  tasks:
[...]
  - name: ensure automount key /devel/project is present
    ipaautomountkey:
      ipaadmin_password: Secret123
      location: raleigh
      mapname: auto.devel
      key: /devel/project
      info: nfs-server.idm.example.com:/exports/project
      state: present
```

6. 파일을 저장합니다.

7. **Ansible** 플레이북을 실행하고 플레이북 및 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i inventory automount-location-map-and-key-present.yml
```

5.6. ANSIBLE을 사용하여 NFS 공유를 소유하는 그룹에 IDM 사용자 추가

IdM(Identity Management) 시스템 관리자는 **Ansible**을 사용하여 **NFS** 공유에 액세스할 수 있는 사용자 그룹을 생성하고 이 그룹에 **IdM** 사용자를 추가할 수 있습니다.

이 예제에서는 **Ansible** 플레이북을 사용하여 **idm_user** 계정이 **developers** 그룹에 속하도록 하는 방법을 설명하므로 **idm_user** 는 **/exports/project NFS** 공유에 액세스할 수 있습니다.

사전 요구 사항

- **nfs-server.idm.example.com NFS** 서버에 대한 루트 액세스 권한이 있습니다. 이 서버는 **raleigh** 자동 마운트 위치에 있는 **IdM** 클라이언트입니다.
- **IdM** 관리자 암호를 알고 있습니다.

- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - ansible-freeipa 패키지가 설치되어 있습니다.
  - ~/*MyPlaybooks*/ 디렉터리에 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.
  - ~/*MyPlaybooks*/ .에서 Ansible을 사용하여 자동 마운트 위치 구성, 맵 및 키 구성의 작업이 이미 포함된mount-location-map-and -key-present.yml 파일을 생성했습니다.

절차

1. Ansible 제어 노드에서 ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. 편집을 위해 autoscale-location-map-and-key-present.yml 파일을 엽니다.

3. tasks 섹션에서 IdM 개발자 그룹이 존재하고 idm_user 가 이 그룹에 추가되었는지 확인하는 작업을 추가합니다.

```
[...]
  tasks:
[...]
  - ipagroup:
     ipaadmin_password: Secret123
      name: developers
      user:
      - idm_user
      state: present
```

4. 파일을 저장합니다.

5. Ansible 플레이북을 실행하고 플레이북 및 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i inventory automount-location-map-and-key-present.yml
```

6. NFS 서버에서 /exports/project 디렉터리의 그룹 소유권을 developers 로 변경하여 그룹의 모든 IdM 사용자가 디렉터리에 액세스할 수 있도록 합니다.

```
# chgrp developers /exports/project
```

5.7. ANSIBLE을 사용하여 자동 마운트 해제 위치에 IDM 클라이언트 추가

IdM(Identity Management) 시스템 관리자는 Ansible을 사용하여 자동 마운트 위치에 IdM 클라이언트를 추가하여 IdM 사용자를 IdM 사용자에게 제공한 IdM 사용자가 이 IdM 클라이언트에 로그인할 수 있도록 자동 마운트 위치에 추가할 수 있습니다.

사전 요구 사항

- 호스트는 IdM에 있습니다. 호스트 이름은 idm-client.idm.example.com 입니다.
- 자동 마운트 위치가 있습니다. 예를 들어 location은 raleigh 입니다.
- IdM 관리자 암호를 알고 있습니다.
- 다음 요구 사항을 충족하는 Ansible 제어 노드를 구성했습니다.
  - Ansible 버전 2.8 이상을 사용하고 있습니다.
  - ansible-freeipa 패키지가 설치되어 있습니다.
  - ~/*MyPlaybooks*/ 디렉터리에 IdM 서버의 FQDN(정규화된 도메인 이름)을 사용하여 Ansible 인벤토리 파일을 생성했습니다.
  - ~/*MyPlaybooks*/ .에서 Ansible을 사용하여 자동 마운트 위치 구성, 맵 및 키 구성의 작업이 이미 포함된mount-location-map-and -key-present.yml 파일을 생성했습니다.

절차

1. Ansible 제어 노드에서 ~/*MyPlaybooks*/ 디렉터리로 이동합니다.

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

2. 편집을 위해 autoscale-location-map-and-key-present.yml 파일을 엽니다.

3. tasks 섹션에서 idm-client.idm.example.com IdM 호스트가 raleigh Autoscaler 위치에 추가되었는지 확인하는 작업을 추가합니다.

```
[...]
  tasks:
[...]
  - ipahost:
      ipaadmin_password: Secret123
      name: idm-client.idm.example.com
      location: raleigh
      state: present
```

4. 파일을 저장합니다.

5. Ansible 플레이북을 실행하고 플레이북 및 인벤토리 파일을 지정합니다.

```
$ ansible-playbook -v -i inventory automount-location-map-and-key-present.yml
```

5.8. IDM 클라이언트에서 자동 마운트 구성

IdM(Identity Management) 시스템 관리자는 사용자가 클라이언트에 로그인할 때 클라이언트가 추가된 위치에 NFS 공유에 액세스할 수 있도록 IdM 클라이언트의 자동 자동 마운트 서비스를 구성할 수 있습니다. 이 예제에서는 raleigh 위치에서 사용할 수 있는 자동 마운트 서비스를 사용하도록 IdM 클라이언트를 구성하는 방법을 설명합니다.

사전 요구 사항

- IdM 클라이언트에 대한 루트 액세스 권한이 있습니다.
- IdM 관리자로 로그인되어 있습니다.

- 자동 마운트 위치가 있습니다. 예를 들어 **location**은 **raleigh** 입니다.

절차

1. IdM 클라이언트에서 **ipa-client-automount** 명령을 입력하고 위치를 지정합니다. **-U** 옵션을 사용하여 축소 스크립트를 실행합니다.

```
# ipa-client-automount --location raleigh -U
```

2. **NetNamespace** 서비스를 중지하고 **SSSD** 캐시를 지우고 **NetNamespace** 서비스를 시작하여 새 구성 설정을 로드합니다.

```
# systemctl stop autofs ; sss_cache -E ; systemctl start autofs
```

5.9. IDM 사용자가 IDM 클라이언트의 NFS 공유에 액세스할 수 있는지 확인

IdM(Identity Management) 시스템 관리자는 특정 그룹의 멤버인 **IdM** 사용자가 특정 **IdM** 클라이언트에 로그인할 때 **NFS** 공유에 액세스할 수 있는지 테스트할 수 있습니다.

이 예제에서는 다음 시나리오를 테스트합니다.

- **developers** 그룹에 속하는 **idm_user** 라는 **IdM** 사용자는 **/devel/project** 디렉터리에 있는 **idm-client.idm.example.com** 에 자동 마운트된 파일의 내용을 읽고 쓸 수 있습니다.

사전 요구 사항

- NFS 서버의 IdM 키탭을 구성하고 NFS 공유를 내보냈습니다.
- IdM 사용자가 NFS 공유에 액세스할 수 있는 방법을 구성한 IdM에 자동 마운트 위치, 맵, 마운트 지점을 구성했습니다.
- Ansible을 사용하여 NFS 공유를 보유한 developers 그룹에 IdM 사용자를 추가했습니다.
- Ansible을 사용하여 IdM 클라이언트를 관련 자동 마운트 위치에 추가했습니다.

- IdM 클라이언트에 자동 마운트를 구성했습니다.

절차

1. IdM 사용자가 읽기-쓰기 디렉터리에 액세스할 수 있는지 확인합니다.

a. IdM 사용자로 IdM 클라이언트에 연결합니다.

```
$ ssh idm_user@idm-client.idm.example.com
Password:
```

b. IdM 사용자의 티켓 통합 티켓 (TGT)을 받으십시오.

```
$ kinit idm_user
```

c. [선택 사항] IdM 사용자의 그룹 멤버십을 확인합니다.

```
$ ipa user-show idm_user
  User login: idm_user
  [...]
  Member of groups: developers, ipausers
```

d. /devel/project 디렉터리로 이동합니다.

```
$ cd /devel/project
```

e. 디렉터리 콘텐츠를 나열합니다.

```
$ ls
rw_file
```

f. 디렉터리의 파일에 행을 추가하여 쓰기 권한을 테스트합니다.

```
$ echo "idm_user can write into the file" > rw_file
```

g. [선택 사항] 파일의 업데이트된 콘텐츠를 봅니다.

```
$ cat rw_file
this is a read-write file
idm_user can write into the file
```

출력에서 **idm_user** 가 파일에 쓸 수 있는지 확인합니다.