Red Hat Enterprise Linux 9

Utilisation d'utilitaires Red Hat externes avec Identity Management

Intégration des services et des produits Red Hat dans l'IdM

Last Updated: 2024-01-18

Red Hat Enterprise Linux 9 Utilisation d'utilitaires Red Hat externes avec Identity Management

Intégration des services et des produits Red Hat dans l'IdM

Notice légale

Résumé

Les administrateurs peuvent intégrer des services et des produits Red Hat dans un domaine Red Hat Identity Management (IdM). Cela inclut des services, tels que Samba, Ansible et automount, ainsi que des produits, tels que OpenShift Container Platform, OpenStack et Satellite. Les utilisateurs IdM peuvent alors accéder à ces services et produits.

Table des matières

RENDRE L'OPEN SOURCE PLUS INCLUSIF 3

FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT 4

CHAPITRE 1. INTÉGRATION DE L'IDM AVEC D'AUTRES PRODUITS RED HAT 5

CHAPITRE 2. CONFIGURATION DE SAMBA SUR UN MEMBRE DU DOMAINE IDM 6
2.1. PRÉPARATION DU DOMAINE IDM POUR L'INSTALLATION DE SAMBA SUR LES MEMBRES DU DOMAINE 6
2.2. ACTIVATION DU TYPE DE CRYPTAGE AES DANS ACTIVE DIRECTORY À L'AIDE D'UN GPO 8
2.3. INSTALLATION ET CONFIGURATION D'UN SERVEUR SAMBA SUR UN CLIENT IDM 9
2.4. AJOUT MANUEL D'UNE CONFIGURATION DE MAPPAGE D'ID SI IDM FAIT CONFIANCE À UN NOUVEAU DOMAINE 11
2.5. RESSOURCES SUPPLÉMENTAIRES 12

CHAPITRE 3. MIGRATION DE NIS VERS LA GESTION DES IDENTITÉS 13
3.1. ACTIVATION DE NIS DANS IDM 13
3.2. MIGRATION DES ENTRÉES D'UTILISATEURS DE NIS VERS IDM 14
3.3. MIGRATION D'UN GROUPE D'UTILISATEURS DE NIS VERS IDM 15
3.4. MIGRATION DES ENTRÉES D'HÔTES DE NIS VERS IDM 16
3.5. MIGRATION DES ENTRÉES DE GROUPES NETS DE NIS VERS IDM 17
3.6. MIGRATION DES CARTES DE MONTAGE AUTOMATIQUE DE NIS VERS IDM 18

CHAPITRE 4. UTILISATION D'AUTOMOUNT DANS IDM 20
4.1. AUTOFS ET AUTOMOUNT DANS IDM 20
4.2. CONFIGURATION D'UNE BASE DE DONNÉES IDM POUR UN SERVEUR NFS 21
4.3. EXPORTATION DE PARTAGES NFS DANS IDM 22
4.4. CONFIGURATION DES EMPLACEMENTS ET DES CARTES DE MONTAGE AUTOMATIQUE DANS IDM À L'AIDE DE LA CLI D'IDM 23
4.5. CONFIGURATION D'AUTOMOUNT SUR UN CLIENT IDM 25
4.6. VÉRIFIER QU'UN UTILISATEUR IDM PEUT ACCÉDER AUX PARTAGES NFS SUR UN CLIENT IDM 25

CHAPITRE 5. UTILISER ANSIBLE POUR MONTER AUTOMATIQUEMENT DES PARTAGES NFS POUR LES UTILISATEURS IDM 27
5.1. AUTOFS ET AUTOMOUNT DANS IDM 28
5.2. CONFIGURATION D'UNE BASE DE DONNÉES IDM POUR UN SERVEUR NFS 28
5.3. EXPORTATION DE PARTAGES NFS DANS IDM 29
5.4. PRÉPARATION DU NŒUD DE CONTRÔLE ANSIBLE POUR LA GESTION DE L'IDM 31
5.5. CONFIGURER LES EMPLACEMENTS, LES CARTES ET LES CLÉS DE MONTAGE AUTOMATIQUE DANS IDM À L'AIDE D'ANSIBLE 32
5.6. UTILISER ANSIBLE POUR AJOUTER DES UTILISATEURS IDM À UN GROUPE PROPRIÉTAIRE DE PARTAGES NFS 35
5.7. CONFIGURATION D'AUTOMOUNT SUR UN CLIENT IDM 36
5.8. VÉRIFIER QU'UN UTILISATEUR IDM PEUT ACCÉDER AUX PARTAGES NFS SUR UN CLIENT IDM 36

RENDRE L'OPEN SOURCE PLUS INCLUSIF

Red Hat s'engage à remplacer les termes problématiques dans son code, sa documentation et ses propriétés Web. Nous commençons par ces quatre termes : master, slave, blacklist et whitelist. En raison de l'ampleur de cette entreprise, ces changements seront mis en œuvre progressivement au cours de plusieurs versions à venir. Pour plus de détails, voir le message de notre directeur technique Chris Wright.

Dans le domaine de la gestion de l'identité, les remplacements terminologiques prévus sont les suivants :

- ***block list*** remplace *blacklist*
- ***allow list*** remplace *whitelist*
- ***secondary*** remplace *slave*
- Le mot *master* est remplacé par un langage plus précis, en fonction du contexte :
  - ***IdM server*** remplace *IdM master*
  - ***CA renewal server*** remplace *CA renewal master*
  - ***CRL publisher server*** remplace *CRL master*
  - ***multi-supplier*** remplace *multi-master*

FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT

Nous apprécions vos commentaires sur notre documentation. Faites-nous savoir comment nous pouvons l'améliorer.

Soumettre des commentaires sur des passages spécifiques

1. Consultez la documentation au format **Multi-page HTML** et assurez-vous que le bouton **Feedback** apparaît dans le coin supérieur droit après le chargement complet de la page.
2. Utilisez votre curseur pour mettre en évidence la partie du texte que vous souhaitez commenter.
3. Cliquez sur le bouton **Add Feedback** qui apparaît près du texte en surbrillance.
4. Ajoutez vos commentaires et cliquez sur **Submit**.

Soumettre des commentaires via Bugzilla (compte requis)

1. Connectez-vous au site Web de Bugzilla.
2. Sélectionnez la version correcte dans le menu **Version**.
3. Saisissez un titre descriptif dans le champ **Summary**.
4. Saisissez votre suggestion d'amélioration dans le champ **Description**. Incluez des liens vers les parties pertinentes de la documentation.
5. Cliquez sur **Submit Bug**.

CHAPITRE 1. INTÉGRATION DE L'IDM AVEC D'AUTRES PRODUITS RED HAT

Cette section fournit des liens vers la documentation d'autres produits Red Hat qui s'intègrent à IdM. Vous pouvez configurer ces produits pour permettre aux utilisateurs de l'IdM d'accéder à leurs services.

Plate-forme d'automatisation Ansible

Configuration de l'authentification LDAP

OpenShift Container Platform

Configuration d'un fournisseur d'identité LDAP

Plate-forme OpenStack

Intégration d'OpenStack Identity (keystone) avec Red Hat Identity Manager (IdM)

Satellite

Utilisation de Red Hat Identity Management

Signature unique

Intégration de SSSD et de FreeIPA Identity Management

Virtualisation

Configuration d'un fournisseur LDAP externe

CHAPITRE 2. CONFIGURATION DE SAMBA SUR UN MEMBRE DU DOMAINE IDM

Cette section décrit comment configurer Samba sur un hôte qui est relié à un domaine Red Hat Identity Management (IdM). Les utilisateurs d'IdM et, le cas échéant, des domaines Active Directory (AD) approuvés, peuvent accéder aux partages et aux services d'impression fournis par Samba.

IMPORTANT

L'utilisation de Samba sur un membre de domaine IdM est une fonctionnalité de l'aperçu technologique qui n'est pas prise en charge et qui comporte certaines limitations. Par exemple, les contrôleurs de confiance IdM ne prennent pas en charge le service Active Directory Global Catalog, ni la résolution des groupes IdM à l'aide des protocoles Distributed Computing Environment / Remote Procedure Calls (DCE/RPC). Par conséquent, les utilisateurs AD ne peuvent accéder aux partages Samba et aux imprimantes hébergées sur des clients IdM que lorsqu'ils sont connectés à d'autres clients IdM ; les utilisateurs AD connectés à une machine Windows ne peuvent pas accéder aux partages Samba hébergés sur un membre du domaine IdM.

Les clients qui déploient Samba sur des membres de domaine IdM sont encouragés à fournir un retour d'information à Red Hat.

Conditions préalables

- L'hôte est joint en tant que client au domaine IdM.
- Les serveurs IdM et le client doivent fonctionner sous RHEL 9.0 ou une version ultérieure.

2.1. PRÉPARATION DU DOMAINE IDM POUR L'INSTALLATION DE SAMBA SUR LES MEMBRES DU DOMAINE

Avant de pouvoir configurer Samba sur un client IdM, vous devez préparer le domaine IdM à l'aide de l'utilitaire **ipa-adtrust-install** sur un serveur IdM.

NOTE

Tout système sur lequel vous exécutez la commande **ipa-adtrust-install** devient automatiquement un contrôleur de confiance AD. Toutefois, vous ne devez exécuter **ipa-adtrust-install** qu'une seule fois sur un serveur IdM.

Conditions préalables

- Le serveur IdM est installé.
- Vous devez disposer des privilèges de root pour installer les paquets et redémarrer les services IdM.

Procédure

1. Installez les paquets nécessaires :

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

2. S'authentifier en tant qu'utilisateur administratif de l'IdM :

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

3. Exécutez l'utilitaire **ipa-adtrust-install**:

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

Les enregistrements de service DNS sont créés automatiquement si IdM a été installé avec un serveur DNS intégré.

Si vous avez installé IdM sans serveur DNS intégré, **ipa-adtrust-install** imprime une liste d'enregistrements de service que vous devez ajouter manuellement au DNS avant de pouvoir continuer.

4. Le script vous indique que le site **/etc/samba/smb.conf** existe déjà et qu'il va être réécrit :

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

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

5. Le script vous invite à configurer le plug-in **slapi-nis**, un plug-in de compatibilité qui permet aux anciens clients Linux de travailler avec des utilisateurs de confiance :

```
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. Lorsque vous y êtes invité, entrez le nom NetBIOS du domaine IdM ou appuyez sur **Enter** pour accepter le nom proposé :

```
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. Vous êtes invité à exécuter la tâche de génération de SID afin de créer un SID pour tous les utilisateurs existants :

```
Voulez-vous exécuter la tâche ipa-sidgen ? [non] : yes
```

Il s'agit d'une tâche gourmande en ressources, donc si vous avez un grand nombre d'utilisateurs, vous pouvez l'exécuter à un autre moment.

8. **(Optional)** Par défaut, la plage de ports Dynamic RPC est définie comme **49152-65535** pour Windows Server 2008 et les versions ultérieures. Si vous devez définir une plage de ports Dynamic RPC différente pour votre environnement, configurez Samba pour qu'il utilise d'autres ports et ouvrez ces ports dans les paramètres de votre pare-feu. L'exemple suivant définit la plage de ports à **55000-65000**.

```
[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. Redémarrez le service **ipa**:

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

10. Utilisez l'utilitaire **smbclient** pour vérifier que Samba répond à l'authentification Kerberos du côté IdM :

```
[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. ACTIVATION DU TYPE DE CRYPTAGE AES DANS ACTIVE DIRECTORY À L'AIDE D'UN GPO

Cette section explique comment activer le type de chiffrement AES dans Active Directory (AD) à l'aide d'un objet de stratégie de groupe (GPO). Certaines fonctionnalités de RHEL, telles que l'exécution d'un serveur Samba sur un client IdM, nécessitent ce type de chiffrement.

Notez que RHEL ne prend plus en charge les types de chiffrement DES et RC4 faibles.

Conditions préalables

- Vous êtes connecté à AD en tant qu'utilisateur pouvant modifier les stratégies de groupe.
- Le site **Group Policy Management Console** est installé sur l'ordinateur.

Procédure

1. Ouvrez le site **Group Policy Management Console**.
2. Cliquez avec le bouton droit de la souris sur **Default Domain Policy**, puis sélectionnez **Edit**. Le site **Group Policy Management Editor** s'ouvre.
3. Naviguez vers **Computer Configuration → Policies → Windows Settings → Security Settings → Local Policies → Security Options**.
4. Double-cliquez sur la politique **Network security: Configure encryption types allowed for Kerberos**.
5. Sélectionnez **AES256_HMAC_SHA1** et, éventuellement, **Future encryption types**.
6. Cliquez sur **OK**.
7. Fermer le site **Group Policy Management Editor**.

8. Répétez les étapes pour le site **Default Domain Controller Policy**.

9. Attendez que les contrôleurs de domaine Windows (DC) appliquent automatiquement la stratégie de groupe. Pour appliquer manuellement le GPO sur un DC, entrez la commande suivante à l'aide d'un compte disposant d'autorisations d'administrateur :

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

2.3. INSTALLATION ET CONFIGURATION D'UN SERVEUR SAMBA SUR UN CLIENT IDM

Cette section décrit comment installer et configurer Samba sur un client inscrit dans un domaine IdM.

Conditions préalables

- Les serveurs IdM et le client doivent fonctionner sous RHEL 9.0 ou une version ultérieure.
- Le domaine IdM est préparé comme décrit dans Préparation du domaine IdM pour l'installation de Samba sur les membres du domaine.
- Si IdM a une confiance configurée avec AD, activez le type de cryptage AES pour Kerberos. Par exemple, utilisez un objet de stratégie de groupe (GPO) pour activer le type de cryptage AES. Pour plus de détails, voir Activation du chiffrement AES dans Active Directory à l'aide d'un GPO .
- Le domaine IdM est préparé comme décrit dans Préparation du domaine IdM pour l'installation de Samba sur les membres du domaine.
- Si IdM a une confiance configurée avec AD, activez le type de cryptage AES pour Kerberos. Par exemple, utilisez un objet de stratégie de groupe (GPO) pour activer le type de cryptage AES. Pour plus de détails, voir Activation du chiffrement AES dans Active Directory à l'aide d'un GPO .

Procédure

1. Installez le paquetage **ipa-client-samba**:

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

2. Utilisez l'utilitaire **ipa-client-samba** pour préparer le client et créer une configuration Samba initiale :

```
[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. Par défaut, **ipa-client-samba** ajoute automatiquement la section **[homes]** au fichier **/etc/samba/smb.conf** qui partage dynamiquement le répertoire personnel d'un utilisateur lorsque celui-ci se connecte. Si les utilisateurs n'ont pas de répertoire personnel sur ce serveur, ou si vous ne voulez pas les partager, supprimez les lignes suivantes de **/etc/samba/smb.conf**:

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

4. Partager des répertoires et des imprimantes. Pour plus de détails, voir les sections suivantes :
   - Configuration d'un partage de fichiers Samba utilisant des listes de contrôle POSIX
   - Configuration d'un partage utilisant les ACL de Windows
   - Configurer Samba en tant que serveur d'impression
5. Ouvrez les ports requis pour un client Samba dans le pare-feu local :

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

6. Activez et démarrez les services **smb** et **winbind**:

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

Verification steps

Exécutez l'étape de vérification suivante sur un autre membre du domaine IdM sur lequel le paquetage **samba-client** est installé :

- Dressez la liste des partages sur le serveur Samba utilisant l'authentification Kerberos :

```
$ 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)
...
```

Ressources supplémentaires

- **ipa-client-samba(1)** page de manuel

2.4. AJOUT MANUEL D'UNE CONFIGURATION DE MAPPAGE D'ID SI IDM FAIT CONFIANCE À UN NOUVEAU DOMAINE

Samba nécessite une configuration de mappage d'ID pour chaque domaine à partir duquel les utilisateurs accèdent aux ressources. Sur un serveur Samba existant fonctionnant sur un client IdM, vous devez ajouter manuellement une configuration de mappage d'identifiants après que l'administrateur a ajouté une nouvelle confiance à un domaine Active Directory (AD).

Conditions préalables

- Vous avez configuré Samba sur un client IdM. Par la suite, une nouvelle confiance a été ajoutée à IdM.
- Les types de chiffrement DES et RC4 pour Kerberos doivent être désactivés dans le domaine AD de confiance. Pour des raisons de sécurité, RHEL 9 ne prend pas en charge ces types de chiffrement faibles.

Procédure

1. S'authentifier à l'aide de la base de données de l'hôte :

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

2. Utilisez la commande **ipa idrange-find** pour afficher l'ID de base et la taille de la plage d'ID du nouveau domaine. Par exemple, la commande suivante affiche les valeurs du domaine **ad.example.com**:

```
[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
----------------------------
```

   Vous aurez besoin des valeurs des attributs **ipabaseid** et **ipaidrangesize** dans les étapes suivantes.

3. Pour calculer l'ID utilisable le plus élevé, utilisez la formule suivante :

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

   Avec les valeurs de l'étape précédente, l'ID utilisable le plus élevé pour le domaine **ad.example.com** est **1918599999** (1918400000 200000 - 1).

4. Modifiez le fichier **/etc/samba/smb.conf** et ajoutez la configuration du mappage d'ID pour le domaine à la section **[global]**:

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

Spécifiez la valeur de l'attribut **ipabaseid** comme étant la plus basse et la valeur calculée à l'étape précédente comme étant la plus haute de la plage.

5. Redémarrez les services **smb** et **winbind**:

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

Verification steps

- Dressez la liste des partages sur le serveur Samba utilisant l'authentification Kerberos :

```
$ 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. RESSOURCES SUPPLÉMENTAIRES

- Installation d'un client de gestion de l'identité

CHAPITRE 3. MIGRATION DE NIS VERS LA GESTION DES IDENTITÉS

Un serveur NIS (Network Information Service) peut contenir des informations sur les utilisateurs, les groupes, les hôtes, les groupes de réseau et les cartes de montage automatique. En tant qu'administrateur système, vous pouvez migrer ces types d'entrées, l'authentification et l'autorisation du serveur NIS vers un serveur de gestion des identités (IdM) afin que toutes les opérations de gestion des utilisateurs soient effectuées sur le serveur IdM. La migration de NIS vers IdM vous permettra également d'accéder à des protocoles plus sûrs tels que Kerberos.

3.1. ACTIVATION DE NIS DANS IDM

Pour permettre la communication entre NIS et le serveur Identity Management (IdM), vous devez activer les options de compatibilité NIS sur le serveur IdM.

Conditions préalables

- Vous avez un accès root sur le serveur IdM.

Procédure

1. Activer l'auditeur NIS et les plug-ins de compatibilité sur le serveur IdM :

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

2. *Optional*: Pour une configuration de pare-feu plus stricte, définissez un port fixe. Par exemple, pour définir le port sur le port inutilisé **514**:

```
[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
```

AVERTISSEMENT

Pour éviter tout conflit avec d'autres services, n'utilisez pas de numéro de port supérieur à 1024.

3. Activer et démarrer le service de cartographie des ports :

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

4. Redémarrer le serveur d'annuaire :

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

3.2. MIGRATION DES ENTRÉES D'UTILISATEURS DE NIS VERS IDM

La carte NIS **passwd** contient des informations sur les utilisateurs, telles que les noms, les UID, le groupe principal, le GECOS, le shell et le répertoire d'origine. Utilisez ces données pour migrer les comptes d'utilisateurs NIS vers la gestion des identités (IdM) :

Conditions préalables

- Vous disposez d'un accès root sur le serveur NIS.
- NIS est activé dans IdM.
- Le serveur NIS est enrôlé dans IdM.

Procédure

1. Installez le paquetage **yp-tools**:

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

2. Sur le serveur NIS, créez le script **/root/nis-users.sh** avec le contenu suivant :

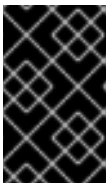
```
#!/bin/sh
# $1 is the NIS domain, $2 is the primary NIS server
ypcat -d $1 -h $2 passwd > /dev/shm/nis-map.passwd 2>&1

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. S'authentifier en tant qu'utilisateur de l'IdM **admin**:

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

4. Exécutez le script. Par exemple :

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



IMPORTANT

Ce script utilise des valeurs codées en dur pour le prénom et le nom, et attribue la valeur **passw0rd1** au mot de passe. L'utilisateur doit modifier le mot de passe temporaire lors de la prochaine connexion.

3.3. MIGRATION D'UN GROUPE D'UTILISATEURS DE NIS VERS IDM

La carte NIS **group** contient des informations sur les groupes, telles que les noms de groupes, les GID ou les membres de groupes. Utilisez ces données pour migrer les groupes NIS vers la gestion des identités (IdM) :

Conditions préalables

- Vous disposez d'un accès root sur le serveur NIS.
- NIS est activé dans IdM.
- Le serveur NIS est enrôlé dans IdM.

Procédure

1. Installez le paquetage **yp-tools**:

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

2. Créez le script **/root/nis-groups.sh** avec le contenu suivant sur le serveur NIS :

```
#!/bin/sh
# $1 is the NIS domain, $2 is the primary NIS 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. S'authentifier en tant qu'utilisateur de l'IdM **admin**:

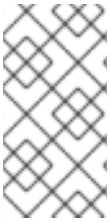
```
[root@nis-server ~]# kinit admin
```

4. Exécutez le script. Par exemple :

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

3.4. MIGRATION DES ENTRÉES D'HÔTES DE NIS VERS IDM

La carte NIS **hosts** contient des informations sur les hôtes, telles que les noms d'hôtes et les adresses IP. Ces données permettent de migrer les entrées d'hôtes NIS vers la gestion des identités (IdM) :



NOTE

Lorsque vous créez un groupe d'hôtes dans IdM, un groupe NIS parallèle correspondant est automatiquement créé. N'utilisez pas les commandes **ipa netgroup-*** sur ces groupes NIS fantômes. Utilisez les commandes **ipa netgroup-*** uniquement pour gérer les groupes nets natifs créés via la commande **netgroup-add**.

Conditions préalables

- Vous disposez d'un accès root sur le serveur NIS.
- NIS est activé dans IdM.
- Le serveur NIS est enrôlé dans IdM.

Procédure

1. Installez le paquetage **yp-tools**:

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

2. Créez le script **/root/nis-hosts.sh** avec le contenu suivant sur le serveur NIS :

```
#!/bin/sh
# $1 is the NIS domain, $2 is the primary NIS 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 $2}')
 primary=$(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 $zone 2>/dev/null | wc -l) -eq 0 ] ; then
  ipa dnszone-add --name-server=$primary --admin-email=root.$primary
 fi
 ptrzone=$(echo $ipaddress | awk -F. '{print $3 "." $2 "." $1 ".in-addr.arpa."}')
```

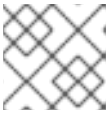
```
 if [ $(ipa dnszone-show $ptrzone 2>/dev/null | wc -l) -eq 0 ] ; then
  ipa dnszone-add  $ptrzone --name-server=$primary --admin-email=root.$primary
 fi
 # Now create this entry
 ipa host-add $hostname --ip-address=$ipaddress
 ipa host-show $hostname
done
```

3. S'authentifier en tant qu'utilisateur de l'IdM **admin**:

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

4. Exécutez le script. Par exemple :

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



NOTE

Ce script ne migre pas les configurations spéciales des hôtes, telles que les alias.

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

3.5. MIGRATION DES ENTRÉES DE GROUPES NETS DE NIS VERS IDM

La carte NIS **netgroup** contient des informations sur les groupes nets. Utilisez ces données pour migrer les groupes nets NIS vers la gestion des identités (IdM) :

Conditions préalables

- Vous disposez d'un accès root sur le serveur NIS.
- NIS est activé dans IdM.
- Le serveur NIS est enrôlé dans IdM.

Procédure

1. Installez le paquetage **yp-tools**:

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

2. Créez le script **/root/nis-netgroups.sh** avec le contenu suivant sur le serveur NIS :

```
#!/bin/sh
# $1 is the NIS domain, $2 is the primary NIS 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. S'authentifier en tant qu'utilisateur de l'IdM **admin**:

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

4. Exécutez le script. Par exemple :

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

3.6. MIGRATION DES CARTES DE MONTAGE AUTOMATIQUE DE NIS VERS IDM

Les cartes Automount sont une série d'entrées imbriquées et interdépendantes qui définissent l'emplacement (l'entrée parent), les clés associées et les cartes. Pour migrer les cartes de montage automatique NIS vers la gestion des identités (IdM) :

Conditions préalables

- Vous disposez d'un accès root sur le serveur NIS.
- NIS est activé dans IdM.
- Le serveur NIS est enrôlé dans IdM.

Procédure

1. Installez le paquetage **yp-tools**:

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

2. Créez le script **/root/nis-automounts.sh** avec le contenu suivant sur le serveur NIS :

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

ipa automountlocation-add $1

# $2 is the NIS domain, $3 is the primary NIS 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
```

NOTE

Le script exporte les informations de montage automatique NIS, génère un format d'échange de données LDAP (LDIF) pour l'emplacement du montage automatique et la carte associée, et importe le fichier LDIF dans le serveur d'annuaire IdM.

3. S'authentifier en tant qu'utilisateur de l'IdM **admin**:

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

4. Exécutez le script. Par exemple :

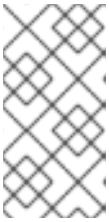
```
[root@nis-server ~]# sh /root/nis-automounts.sh location nisdomain
    nis-server.example.com map_name
```

CHAPITRE 4. UTILISATION D'AUTOMOUNT DANS IDM

Automount est un moyen de gérer, d'organiser et d'accéder à des répertoires sur plusieurs systèmes. Automount monte automatiquement un répertoire lorsque l'accès à celui-ci est demandé. Cela fonctionne bien dans un domaine de gestion des identités (IdM), car cela vous permet de partager facilement des répertoires sur les clients du domaine.

L'exemple utilise le scénario suivant :

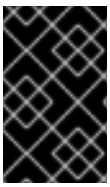
- **nfs-server.idm.example.com** est le nom de domaine complet (FQDN) d'un serveur NFS (Network File System).
- Par souci de simplicité, **nfs-server.idm.example.com** est un client IdM qui fournit les cartes de l'emplacement de montage automatique **raleigh**.



NOTE

Un emplacement automount est un ensemble unique de cartes NFS. Idéalement, ces cartes sont toutes situées dans la même région géographique afin que, par exemple, les clients puissent bénéficier de connexions rapides, mais ce n'est pas obligatoire.

- Le serveur NFS exporte le répertoire **/exports/project** en lecture-écriture.
- Tout utilisateur IdM appartenant au groupe **developers** peut accéder au contenu du répertoire exporté en tant que **/devel/project/** sur tout client IdM qui utilise l'emplacement de montage automatique **raleigh**.
- **idm-client.idm.example.com** est un client IdM qui utilise l'emplacement de montage automatique **raleigh**.



IMPORTANT

Si vous souhaitez utiliser un serveur Samba au lieu d'un serveur NFS pour fournir les partages aux clients IdM, reportez-vous à la section Comment configurer des montages CIFS kerberisés avec Autofs dans un environnement IPA ? Solution KCS.

4.1. AUTOFS ET AUTOMOUNT DANS IDM

Le service **autofs** automatise le montage des répertoires, selon les besoins, en demandant au démon **automount** de monter les répertoires lorsqu'on y accède. En outre, après une période d'inactivité, **autofs** demande à **automount** de démonter les répertoires montés automatiquement. Contrairement au montage statique, le montage à la demande permet d'économiser les ressources du système.

Cartes de montage automatique

Sur un système qui utilise **autofs**, la configuration de **automount** est stockée dans plusieurs fichiers différents. Le fichier de configuration principal de **automount** est **/etc/auto.master**, qui contient le mappage principal des points de montage **automount** et leurs ressources associées sur un système. Ce mappage est connu sous le nom de *automount maps*.
Le fichier de configuration **/etc/auto.master** contient la carte *master map*. Il peut contenir des références à d'autres cartes. Ces cartes peuvent être directes ou indirectes. Les cartes directes utilisent des noms de chemin absolus pour leurs points de montage, tandis que les cartes indirectes utilisent des noms de chemin relatifs.

Configuration du montage automatique dans l'IdM

Bien que **automount** récupère généralement ses données cartographiques à partir du site local **/etc/auto.master** et des fichiers associés, il peut également récupérer des données cartographiques à partir d'autres sources. Une source courante est un serveur LDAP. Dans le contexte de la gestion des identités (IdM), il s'agit d'un serveur d'annuaire.
Si un système qui utilise **autofs** est un client dans un domaine IdM, la configuration **automount** n'est pas stockée dans des fichiers de configuration locaux. Au lieu de cela, la configuration de **autofs**, telle que les cartes, les emplacements et les clés, est stockée sous forme d'entrées LDAP dans l'annuaire IdM. Par exemple, pour le domaine IdM **idm.example.com**, la valeur par défaut de *master map* est stockée comme suit :

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

Ressources supplémentaires

- Montage de systèmes de fichiers à la demande

4.2. CONFIGURATION D'UNE BASE DE DONNÉES IDM POUR UN SERVEUR NFS

Configurer un serveur NFS compatible avec Kerberos afin que les utilisateurs connectés à d'autres clients Identity Management (IdM) puissent accéder aux répertoires et aux fichiers de ce serveur NFS.

L'exemple décrit comment configurer le service NFS fonctionnant sur **nfs-server.idm.example.com**.

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Vous avez accès au serveur NFS à l'adresse **root**.
- Vous avez installé les paquets nécessaires pour exporter des partages NFS .
- [Facultatif] Vous avez configuré le serveur NFS pour qu'il fonctionne derrière un pare-feu .

Procédure

1. Sur n'importe quel hôte inscrit à l'IdM, ajoutez le service NFS à l'IdM :

```
$ 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. Sur le serveur NFS, obtenez le keytab du service NFS :

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

3. Sur le serveur NFS, redémarrez le service NFS :

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

4. Sur le serveur NFS, activez le service NFS :

```
# 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. EXPORTATION DE PARTAGES NFS DANS IDM

En tant qu'administrateur du système de gestion des identités (IdM), vous pouvez utiliser un serveur NFS pour partager un répertoire avec les utilisateurs IdM sur le réseau.

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Vous avez accès au serveur NFS à l'adresse **root**.
- Vous avez installé les paquets nécessaires pour exporter des partages NFS .
- [Facultatif] Vous avez configuré le serveur NFS pour qu'il fonctionne derrière un pare-feu .

Procédure

1. Créez le répertoire que vous souhaitez exporter :

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

2. Donner au propriétaire et au groupe les droits de lecture, d'écriture et d'exécution du répertoire :

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

3. Ajoutez le bit collant **GSID** pour que tous les fichiers créés dans le répertoire aient leur propriété de groupe définie sur celle du propriétaire du répertoire :

```
# chmod g s /exports/project
```

4. Créez un groupe IdM dont les membres pourront accéder à l'annuaire. L'exemple de groupe IdM est **developers**:

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

5. Changez la propriété du groupe du répertoire **/exports/project** en **developers** afin que chaque utilisateur IdM du groupe puisse y accéder :

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

6. Ajouter un utilisateur IdM au groupe. L'exemple d'utilisateur est **idm_user**:

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

7. Créer un fichier dans le répertoire avec un certain contenu :

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

8. Dans un fichier du répertoire **/etc/exports.d/**, ajoutez les informations suivantes :

   - Le répertoire que vous souhaitez exporter
   - Comment vous voulez que les utilisateurs s'authentifient pour accéder aux fichiers du répertoire
   - Les permissions que vous voulez que les utilisateurs aient sur les fichiers du répertoire

```
# echo "/exports/project *(sec=krb5,rw)" > /etc/exports.d/project.exports
```

   sec=krb5 utilise le protocole Kerberos V5 au lieu des UID et GID UNIX locaux pour authentifier les utilisateurs.

   Vous pouvez également utiliser **sec=krb5i** ou **sec=krb5p**:

   sec=krb5i

   utilise Kerberos V5 pour l'authentification des utilisateurs et effectue un contrôle d'intégrité des opérations NFS à l'aide de sommes de contrôle sécurisées afin d'empêcher la falsification des données.

   sec=krb5p

   utilise Kerberos V5 pour l'authentification des utilisateurs, le contrôle d'intégrité et le chiffrement du trafic NFS afin d'empêcher le reniflage du trafic. Il s'agit du paramètre le plus sûr, mais c'est aussi celui qui entraîne le plus de surcoût en termes de performances.

9. Réexporter tous les répertoires, en synchronisant la table d'exportation principale conservée dans **/var/lib/nfs/etab** avec **/etc/exports** et les fichiers sous **/etc/exports.d**:

```
# exportfs -r
```

10. Affichez la liste d'exportation actuelle adaptée à **/etc/exports**:

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

Ressources supplémentaires

- Pour plus d'informations sur les méthodes de **krb5**, voir la page de manuel **nfs**.

4.4. CONFIGURATION DES EMPLACEMENTS ET DES CARTES DE MONTAGE AUTOMATIQUE DANS IDM À L'AIDE DE LA CLI D'IDM

Un emplacement est un ensemble de cartes, qui sont toutes stockées sur **auto.master**. Un emplacement peut contenir plusieurs cartes. L'entrée de l'emplacement fonctionne uniquement comme un conteneur pour les entrées de cartes ; il ne s'agit pas d'une configuration de montage automatique en soi.

En tant qu'administrateur système dans Identity Management (IdM), vous pouvez configurer des emplacements de montage automatique et des cartes dans IdM afin que les utilisateurs IdM des emplacements spécifiés puissent accéder aux partages exportés par un serveur NFS en naviguant vers des points de montage spécifiques sur leurs hôtes. Le répertoire du serveur NFS exporté et les points de montage sont spécifiés dans les cartes. L'exemple décrit comment configurer l'emplacement **raleigh** et une carte qui monte le partage **nfs-server.idm.example.com:/exports/project** sur le point de montage **/devel/** sur le client IdM en tant que répertoire en lecture-écriture.

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM sur n'importe quel hôte inscrit à IdM.

Procédure

1. Créez l'emplacement de montage automatique **raleigh**:

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

2. Créez une carte de montage automatique **auto.devel** à l'emplacement **raleigh**:

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

3. Ajoutez les clés et les informations de montage pour le partage **exports/**:

   a. Ajoutez les informations relatives à la clé et à la monture de la carte **auto.devel**:

```
$ 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. Ajoutez les informations relatives à la clé et à la monture de la carte **auto.master**:

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

4.5. CONFIGURATION D'AUTOMOUNT SUR UN CLIENT IDM

En tant qu'administrateur du système de gestion des identités (IdM), vous pouvez configurer les services de montage automatique sur un client IdM afin que les partages NFS configurés pour un emplacement auquel le client a été ajouté soient automatiquement accessibles à un utilisateur IdM lorsque celui-ci se connecte au client. L'exemple décrit comment configurer un client IdM pour qu'il utilise les services de montage automatique disponibles dans l'emplacement **raleigh**.

Conditions préalables

- Vous avez un accès **root** au client IdM.
- Vous êtes connecté en tant qu'administrateur IdM.
- L'emplacement du montage automatique existe. L'exemple d'emplacement est **raleigh**.

Procédure

1. Sur le client IdM, entrez la commande **ipa-client-automount** et indiquez l'emplacement. Utilisez l'option **-U** pour exécuter le script sans surveillance :

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

2. Arrêtez le service autofs, effacez le cache SSSD et démarrez le service autofs pour charger les nouveaux paramètres de configuration :

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

4.6. VÉRIFIER QU'UN UTILISATEUR IDM PEUT ACCÉDER AUX PARTAGES NFS SUR UN CLIENT IDM

En tant qu'administrateur du système de gestion des identités (IdM), vous pouvez vérifier si un utilisateur IdM membre d'un groupe spécifique peut accéder aux partages NFS lorsqu'il est connecté à un client IdM spécifique.

Dans l'exemple, le scénario suivant est testé :

- Un utilisateur IdM nommé **idm_user** et appartenant au groupe **developers** peut lire et écrire le contenu des fichiers du répertoire **/devel/project** monté automatiquement sur **idm-client.idm.example.com**, un client IdM situé dans l'emplacement de montage automatique **raleigh**.

Conditions préalables

- Vous avez configuré un keytab IdM pour un serveur NFS et exporté un partage NFS.
- Vous avez configuré des emplacements, des cartes et des points de montage automatiques dans IdM, dans lesquels vous avez configuré la manière dont les utilisateurs d'IdM peuvent accéder au partage NFS.
- Vous avez configuré automount sur le client IdM .

Procédure

1. Vérifiez que l'utilisateur IdM peut accéder au répertoire **read-write**:

   a. Se connecter au client IdM en tant qu'utilisateur IdM :

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

   b. Obtenir le ticket d'attribution de ticket (TGT) pour l'utilisateur IdM :

   ```
   $ kinit idm_user
   ```

   c. [Facultatif] Afficher l'appartenance au groupe de l'utilisateur IdM :

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

   d. Naviguez jusqu'au répertoire **/devel/project**:

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

   e. Liste le contenu du répertoire :

   ```
   $ ls
   rw_file
   ```

   f. Ajoutez une ligne au fichier dans le répertoire pour tester l'autorisation **write**:

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

   g. [Facultatif] Affichez le contenu mis à jour du fichier :

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

La sortie confirme que **idm_user** peut écrire dans le fichier.

CHAPITRE 5. UTILISER ANSIBLE POUR MONTER AUTOMATIQUEMENT DES PARTAGES NFS POUR LES UTILISATEURS IDM

Automount est un moyen de gérer, d'organiser et d'accéder à des répertoires sur plusieurs systèmes. Automount monte automatiquement un répertoire lorsque l'accès à celui-ci est demandé. Cela fonctionne bien dans un domaine de gestion des identités (IdM), car cela vous permet de partager facilement des répertoires sur les clients du domaine.

Vous pouvez utiliser Ansible pour configurer les partages NFS afin qu'ils soient montés automatiquement pour les utilisateurs IdM connectés aux clients IdM dans un emplacement IdM.

L'exemple de ce chapitre utilise le scénario suivant :

- **nfs-server.idm.example.com** est le nom de domaine complet (FQDN) d'un serveur NFS (Network File System).
- **nfs-server.idm.example.com** est un client IdM situé dans l'emplacement de montage automatique **raleigh**.
- Le serveur NFS exporte le répertoire **/exports/project** en lecture-écriture.
- Tout utilisateur IdM appartenant au groupe **developers** peut accéder au contenu du répertoire exporté en tant que **/devel/project/** sur n'importe quel client IdM situé dans le même emplacement de montage automatique **raleigh** que le serveur NFS.
- **idm-client.idm.example.com** est un client IdM situé dans l'emplacement de montage automatique **raleigh**.

IMPORTANT

Si vous souhaitez utiliser un serveur Samba au lieu d'un serveur NFS pour fournir les partages aux clients IdM, reportez-vous à la section Comment configurer des montages CIFS kerberisés avec Autofs dans un environnement IPA ? Solution KCS.

Le chapitre contient les sections suivantes :

1. Autofs et automount dans IdM
2. Configuration d'une base de données IdM pour un serveur NFS
3. Exportation de partages NFS dans IdM
4. Préparation du nœud de contrôle Ansible pour la gestion de l'IdM
5. Configurer les emplacements, les cartes et les clés de montage automatique dans IdM à l'aide d'Ansible
6. Utiliser Ansible pour ajouter des utilisateurs IdM à un groupe propriétaire de partages NFS
7. Configuration d'automount sur un client IdM
8. Vérifier qu'un utilisateur IdM peut accéder aux partages NFS sur un client IdM

5.1. AUTOFS ET AUTOMOUNT DANS IDM

Le service **autofs** automatise le montage des répertoires, selon les besoins, en demandant au démon **automount** de monter les répertoires lorsqu'on y accède. En outre, après une période d'inactivité, **autofs** demande à **automount** de démonter les répertoires montés automatiquement. Contrairement au montage statique, le montage à la demande permet d'économiser les ressources du système.

Cartes de montage automatique

Sur un système qui utilise **autofs**, la configuration de **automount** est stockée dans plusieurs fichiers différents. Le fichier de configuration principal de **automount** est **/etc/auto.master**, qui contient le mappage principal des points de montage **automount** et leurs ressources associées sur un système. Ce mappage est connu sous le nom de *automount maps*.

Le fichier de configuration **/etc/auto.master** contient la carte *master map*. Il peut contenir des références à d'autres cartes. Ces cartes peuvent être directes ou indirectes. Les cartes directes utilisent des noms de chemin absolus pour leurs points de montage, tandis que les cartes indirectes utilisent des noms de chemin relatifs.

Configuration du montage automatique dans l'IdM

Bien que **automount** récupère généralement ses données cartographiques à partir du site local **/etc/auto.master** et des fichiers associés, il peut également récupérer des données cartographiques à partir d'autres sources. Une source courante est un serveur LDAP. Dans le contexte de la gestion des identités (IdM), il s'agit d'un serveur d'annuaire.

Si un système qui utilise **autofs** est un client dans un domaine IdM, la configuration **automount** n'est pas stockée dans des fichiers de configuration locaux. Au lieu de cela, la configuration de **autofs**, telle que les cartes, les emplacements et les clés, est stockée sous forme d'entrées LDAP dans l'annuaire IdM. Par exemple, pour le domaine IdM **idm.example.com**, la valeur par défaut de *master map* est stockée comme suit :

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

Ressources supplémentaires

- Montage de systèmes de fichiers à la demande

5.2. CONFIGURATION D'UNE BASE DE DONNÉES IDM POUR UN SERVEUR NFS

Configurer un serveur NFS compatible avec Kerberos afin que les utilisateurs connectés à d'autres clients Identity Management (IdM) puissent accéder aux répertoires et aux fichiers de ce serveur NFS.

L'exemple décrit comment configurer le service NFS fonctionnant sur **nfs-server.idm.example.com**.

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Vous avez accès au serveur NFS à l'adresse **root**.

- Vous avez installé les paquets nécessaires pour exporter des partages NFS .
- [Facultatif] Vous avez configuré le serveur NFS pour qu'il fonctionne derrière un pare-feu .

Procédure

1. Sur n'importe quel hôte inscrit à l'IdM, ajoutez le service NFS à l'IdM :

```
$ 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. Sur le serveur NFS, obtenez le keytab du service NFS :

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

3. Sur le serveur NFS, redémarrez le service NFS :

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

4. Sur le serveur NFS, activez le service NFS :

```
# 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. EXPORTATION DE PARTAGES NFS DANS IDM

En tant qu'administrateur du système de gestion des identités (IdM), vous pouvez utiliser un serveur NFS pour partager un répertoire avec les utilisateurs IdM sur le réseau.

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Vous avez accès au serveur NFS à l'adresse **root**.
- Vous avez installé les paquets nécessaires pour exporter des partages NFS .
- [Facultatif] Vous avez configuré le serveur NFS pour qu'il fonctionne derrière un pare-feu .

Procédure

1. Créez le répertoire que vous souhaitez exporter :

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

2. Donner au propriétaire et au groupe les droits de lecture, d'écriture et d'exécution du répertoire :

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

3. Ajoutez le bit collant **GSID** pour que tous les fichiers créés dans le répertoire aient leur propriété de groupe définie sur celle du propriétaire du répertoire :

```
# chmod g s /exports/project
```

4. Créer un fichier dans le répertoire avec un certain contenu :

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

5. Dans un fichier du répertoire **/etc/exports.d/**, ajoutez les informations suivantes :

   - Le répertoire que vous souhaitez exporter
   - Comment vous voulez que les utilisateurs s'authentifient pour accéder aux fichiers du répertoire
   - Les permissions que vous voulez que les utilisateurs aient sur les fichiers du répertoire

   ```
   # echo "/exports/project *(sec=krb5,rw)" > /etc/exports.d/project.exports
   ```

   sec=krb5 utilise le protocole Kerberos V5 au lieu des UID et GID UNIX locaux pour authentifier les utilisateurs.

   Vous pouvez également utiliser **sec=krb5i** ou **sec=krb5p**:

   sec=krb5i

   utilise Kerberos V5 pour l'authentification des utilisateurs et effectue un contrôle d'intégrité des opérations NFS à l'aide de sommes de contrôle sécurisées afin d'empêcher la falsification des données.

   sec=krb5p

   utilise Kerberos V5 pour l'authentification des utilisateurs, le contrôle d'intégrité et le chiffrement du trafic NFS afin d'empêcher le reniflage du trafic. Il s'agit du paramètre le plus sûr, mais c'est aussi celui qui entraîne le plus de surcoût en termes de performances.

6. Réexporter tous les répertoires, en synchronisant la table d'exportation principale conservée dans **/var/lib/nfs/etab** avec **/etc/exports** et les fichiers sous **/etc/exports.d**:

```
# exportfs -r
```

7. Affichez la liste d'exportation actuelle adaptée à **/etc/exports**:

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

Ressources supplémentaires

- Pour plus d'informations sur les méthodes de **krb5**, voir la page de manuel **nfs**.

5.4. PRÉPARATION DU NŒUD DE CONTRÔLE ANSIBLE POUR LA GESTION DE L'IDM

En tant qu'administrateur système gérant la gestion des identités (IdM), lorsque vous travaillez avec Red Hat Ansible Engine, il est recommandé de procéder comme suit :

- Créez un sous-répertoire dédié aux playbooks Ansible dans votre répertoire personnel, par exemple **~/MyPlaybooks**.
- Copiez et adaptez les exemples de playbooks Ansible des répertoires et sous-répertoires **/usr/share/doc/ansible-freeipa/*** et **/usr/share/doc/rhel-system-roles/*** dans votre répertoire **~/MyPlaybooks**.
- Incluez votre fichier d'inventaire dans votre répertoire **~/MyPlaybooks**.

En suivant cette pratique, vous pouvez trouver tous vos playbooks en un seul endroit et vous pouvez exécuter vos playbooks sans invoquer les privilèges root.

NOTE

Vous n'avez besoin que des privilèges **root** sur les nœuds gérés pour exécuter les rôles **ipaserver**, **ipareplica**, **ipaclient**, **ipabackup**, **ipasmartcard_server** et **ipasmartcard_client ansible-freeipa** . Ces rôles nécessitent un accès privilégié aux répertoires et au gestionnaire de paquets logiciels **dnf**.

Cette section décrit comment créer le répertoire **~/MyPlaybooks** et le configurer de manière à ce que vous puissiez l'utiliser pour stocker et exécuter des playbooks Ansible.

Conditions préalables

- Vous avez installé un serveur IdM sur vos nœuds gérés, ***server.idm.example.com*** et ***replica.idm.example.com***.
- Vous avez configuré le DNS et le réseau pour pouvoir vous connecter aux nœuds gérés, ***server.idm.example.com*** et ***replica.idm.example.com***directement à partir du nœud de contrôle.
- Vous connaissez le mot de passe de l'IdM **admin**.

Procédure

1. Créez un répertoire pour votre configuration Ansible et vos playbooks dans votre répertoire personnel :

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

2. Allez dans le répertoire **~/MyPlaybooks/**:

```
$ cd ~/MyPlaybooks
```

3. Créez le fichier **~/MyPlaybooks/ansible.cfg** avec le contenu suivant :

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

```
[privilege_escalation]
become=True
```

4. Créez le fichier **~/MyPlaybooks/inventory** avec le contenu suivant :

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

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

[ipacluster:children]
ipaserver
ipareplicas

[ipacluster:vars]
ipaadmin_password=SomeADMINpassword

[ipaclients]
ipaclient1.example.com
ipaclient2.example.com

[ipaclients:vars]
ipaadmin_password=SomeADMINpassword
```

Cette configuration définit deux groupes d'hôtes, **eu** et **us**, pour les hôtes de ces sites. En outre, cette configuration définit le groupe d'hôtes **ipaserver**, qui contient tous les hôtes des groupes **eu** et **us**.

5. [Facultatif] Créez une clé publique et une clé privée SSH. Pour simplifier l'accès dans votre environnement de test, ne définissez pas de mot de passe pour la clé privée :

```
$ ssh-keygen
```

6. Copiez la clé publique SSH dans le compte IdM **admin** sur chaque nœud géré :

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

Vous devez saisir le mot de passe IdM **admin** lorsque vous entrez dans ces commandes.

Ressources supplémentaires

- Installation d'un serveur de gestion des identités à l'aide d'un playbook Ansible .
- Comment constituer votre inventaire .

5.5. CONFIGURER LES EMPLACEMENTS, LES CARTES ET LES CLÉS DE MONTAGE AUTOMATIQUE DANS IDM À L'AIDE D'ANSIBLE

En tant qu'administrateur du système de gestion des identités (IdM), vous pouvez configurer des emplacements de montage automatique et des cartes dans IdM afin que les utilisateurs IdM des emplacements spécifiés puissent accéder aux partages exportés par un serveur NFS en naviguant vers

des points de montage spécifiques sur leurs hôtes. Le répertoire du serveur NFS exporté et les points de montage sont spécifiés dans les cartes. En termes de LDAP, un emplacement est un conteneur pour ces entrées de carte.

L'exemple décrit comment utiliser Ansible pour configurer l'emplacement **raleigh** et une carte qui monte le partage **nfs-server.idm.example.com:/exports/project** sur le point de montage **/devel/project** sur le client IdM en tant que répertoire en lecture-écriture.

Conditions préalables

- Vous connaissez le mot de passe de l'IdM **admin**.
- Vous avez configuré votre nœud de contrôle Ansible pour qu'il réponde aux exigences suivantes :
  - Vous utilisez la version 2.8 ou ultérieure d'Ansible.
  - Vous avez installé le paquetage **ansible-freeipa** sur le contrôleur Ansible.
  - L'exemple suppose que dans le répertoire ***~/MyPlaybooks/*** vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM.
  - L'exemple suppose que le coffre-fort **secret.yml** Ansible stocke votre **ipaadmin_password**.

Procédure

1. Sur votre nœud de contrôle Ansible, naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **automount-location-present.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/automount/**:

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

3. Ouvrez le fichier **automount-location-map-and-key-present.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaautomountlocation** task :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Fixer la variable **name** à **raleigh**.
   - Assurez-vous que la variable **state** est définie sur **present**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Automount location present example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
```

```
tasks:
- name: Ensure automount location is present
  ipaautomountlocation:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: raleigh
    state: present
```

5. Poursuivre l'édition du fichier **automount-location-map-and-key-present.yml**:

   a. Dans la section **tasks**, ajoutez une tâche pour assurer la présence d'une carte automount :

```
[...]
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
[...]
  - name: ensure map named auto.devel in location raleigh is created
    ipaautomountmap:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: auto.devel
      location: raleigh
      state: present
```

   b. Ajoutez une autre tâche pour ajouter le point de montage et les informations sur le serveur NFS à la carte :

```
[...]
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
[...]
  - name: ensure automount key /devel/project is present
    ipaautomountkey:
      ipaadmin_password: "{{ ipaadmin_password }}"
      location: raleigh
      mapname: auto.devel
      key: /devel/project
      info: nfs-server.idm.example.com:/exports/project
      state: present
```

   c. Ajouter une autre tâche pour s'assurer que **auto.devel** est connecté à **auto.master**:

```
[...]
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
[...]
- name: Ensure auto.devel is connected in auto.master:
  ipaautomountkey:
    ipaadmin_password: "{{ ipaadmin_password }}"
    location: raleigh
    mapname: auto.map
    key: /devel
    info: auto.devel
    state: present
```

6. Enregistrer le fichier.

7. Exécutez le playbook Ansible et spécifiez les fichiers de playbook et d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automount-location-map-and-key-present.yml
```

5.6. UTILISER ANSIBLE POUR AJOUTER DES UTILISATEURS IDM À UN GROUPE PROPRIÉTAIRE DE PARTAGES NFS

En tant qu'administrateur du système de gestion des identités (IdM), vous pouvez utiliser Ansible pour créer un groupe d'utilisateurs capable d'accéder aux partages NFS et ajouter des utilisateurs IdM à ce groupe.

Cet exemple décrit comment utiliser un script Ansible pour s'assurer que le compte **idm_user** appartient au groupe **developers**, afin que **idm_user** puisse accéder au partage NFS **/exports/project**.

Conditions préalables

- Vous avez un accès **root** au serveur NFS **nfs-server.idm.example.com**, qui est un client IdM situé dans l'emplacement automount **raleigh**.
- Vous connaissez le mot de passe de l'IdM **admin**.
- Vous avez configuré votre nœud de contrôle Ansible pour qu'il réponde aux exigences suivantes :
  - Vous utilisez la version 2.8 ou ultérieure d'Ansible.
  - Vous avez installé le paquetage **ansible-freeipa** sur le contrôleur Ansible.
  - L'exemple suppose que dans le répertoire ***~/MyPlaybooks/*** vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM.
  - L'exemple suppose que le coffre-fort **secret.yml** Ansible stocke votre **ipaadmin_password**.
  - Dans ***~/MyPlaybooks/***vous avez créé le fichier **automount-location-map-and-key-present.yml** qui contient déjà les tâches de Configuration des emplacements de montage automatique, des cartes et des clés dans IdM à l'aide d'Ansible.

Procédure

1. Sur votre nœud de contrôle Ansible, naviguez jusqu'au répertoire ***~/MyPlaybooks/*** (répertoire) :

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

2. Ouvrez le fichier **automount-location-map-and-key-present.yml** pour le modifier.

3. Dans la section **tasks**, ajoutez une tâche pour vous assurer que le groupe IdM **developers** existe et que **idm_user** est ajouté à ce groupe :

```
[...]
  vars_files:
```

```
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
[...]
  - ipagroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
     name: developers
     user:
     - idm_user
     state: present
```

4. Enregistrer le fichier.

5. Exécutez le playbook Ansible et spécifiez les fichiers de playbook et d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automount-location-map-and-key-present.yml
```

6. Sur le serveur NFS, modifiez la propriété du groupe du répertoire **/exports/project** en **developers** afin que chaque utilisateur IdM du groupe puisse accéder au répertoire :

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

5.7. CONFIGURATION D'AUTOMOUNT SUR UN CLIENT IDM

En tant qu'administrateur du système de gestion des identités (IdM), vous pouvez configurer les services de montage automatique sur un client IdM afin que les partages NFS configurés pour un emplacement auquel le client a été ajouté soient automatiquement accessibles à un utilisateur IdM lorsque celui-ci se connecte au client. L'exemple décrit comment configurer un client IdM pour qu'il utilise les services de montage automatique disponibles dans l'emplacement **raleigh**.

Conditions préalables

- Vous avez un accès **root** au client IdM.
- Vous êtes connecté en tant qu'administrateur IdM.
- L'emplacement du montage automatique existe. L'exemple d'emplacement est **raleigh**.

Procédure

1. Sur le client IdM, entrez la commande **ipa-client-automount** et indiquez l'emplacement. Utilisez l'option **-U** pour exécuter le script sans surveillance :

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

2. Arrêtez le service autofs, effacez le cache SSSD et démarrez le service autofs pour charger les nouveaux paramètres de configuration :

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

5.8. VÉRIFIER QU'UN UTILISATEUR IDM PEUT ACCÉDER AUX PARTAGES NFS SUR UN CLIENT IDM

En tant qu'administrateur du système de gestion des identités (IdM), vous pouvez vérifier si un utilisateur IdM membre d'un groupe spécifique peut accéder aux partages NFS lorsqu'il est connecté à un client IdM spécifique.

Dans l'exemple, le scénario suivant est testé :

- Un utilisateur IdM nommé **idm_user** et appartenant au groupe **developers** peut lire et écrire le contenu des fichiers du répertoire **/devel/project** monté automatiquement sur **idm-client.idm.example.com**, un client IdM situé dans l'emplacement de montage automatique **raleigh**.

Conditions préalables

- Vous avez configuré un keytab IdM pour un serveur NFS et exporté un partage NFS.
- Vous avez configuré des emplacements, des cartes et des points de montage automatiques dans IdM, dans lesquels vous avez configuré la manière dont les utilisateurs d'IdM peuvent accéder au partage NFS.
- Vous avez utilisé Ansible pour ajouter des utilisateurs IdM au groupe de développeurs qui possède les partages NFS.
- Vous avez configuré automount sur le client IdM .

Procédure

1. Vérifiez que l'utilisateur IdM peut accéder au répertoire **read-write**:
   a. Se connecter au client IdM en tant qu'utilisateur IdM :

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

   b. Obtenir le ticket d'attribution de ticket (TGT) pour l'utilisateur IdM :

```
$ kinit idm_user
```

   c. [Facultatif] Afficher l'appartenance au groupe de l'utilisateur IdM :

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

   d. Naviguez jusqu'au répertoire **/devel/project**:

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

   e. Liste le contenu du répertoire :

```
$ ls
rw_file
```

   f. Ajoutez une ligne au fichier dans le répertoire pour tester l'autorisation **write**:

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

g. [Facultatif] Affichez le contenu mis à jour du fichier :

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

La sortie confirme que **idm_user** peut écrire dans le fichier.