Red Hat Enterprise Linux 9

Utiliser Ansible pour installer et gérer la gestion des identités

Utiliser Ansible pour maintenir un environnement IdM

Last Updated: 2024-02-22

Red Hat Enterprise Linux 9 Utiliser Ansible pour installer et gérer la gestion des identités

Utiliser Ansible pour maintenir un environnement IdM

Notice légale

Résumé

Red Hat fournit le paquetage ansible-freeipa pour permettre aux administrateurs d'exécuter Red Hat Identity Management (IdM) à l'aide d'Ansible. Vous pouvez utiliser des playbooks pour installer IdM et gérer les utilisateurs, les groupes, les hôtes, le contrôle d'accès et les paramètres de configuration.

Table des matières

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* sera remplacé par des termes 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. TERMINOLOGIE ANSIBLE

Les chapitres de ce titre utilisent la terminologie officielle d'Ansible. Si vous n'êtes pas familier avec cette terminologie, lisez la documentation officielle d'Ansible en amont avant de poursuivre, en particulier les sections suivantes :

- La section Concepts de base d'Ansible donne un aperçu des concepts les plus couramment utilisés dans Ansible.
- Le guide de l'utilisateur décrit les situations et les questions les plus courantes lorsque l'on commence à utiliser Ansible, comme l'utilisation de la ligne de commande, le travail avec un inventaire, l'interaction avec les données, l'écriture de tâches, de jeux et de carnets de jeu, et l'exécution de carnets de jeu.
- Comment construire votre inventaire offre des conseils sur la façon de concevoir votre inventaire. Un inventaire est une liste ou un groupe de listes qu'Ansible utilise pour travailler avec plusieurs nœuds ou hôtes gérés dans votre infrastructure.
- Intro to playbooks présente le concept de playbook Ansible comme un système répétable et réutilisable pour gérer les configurations, déployer des machines et des applications complexes.
- La section sur les rôles Ansible explique comment automatiser le chargement de variables, de tâches et de gestionnaires en fonction d'une structure de fichiers connue.
- Le glossaire explique les termes utilisés ailleurs dans la documentation Ansible.

CHAPITRE 2. INSTALLATION D'UN SERVEUR DE GESTION DES IDENTITÉS À L'AIDE D'UN PLAYBOOK ANSIBLE

Les sections suivantes décrivent comment configurer un système en tant que serveur IdM à l'aide d'Ansible. La configuration d'un système en tant que serveur IdM établit un domaine IdM et permet au système d'offrir des services IdM aux clients IdM. Vous pouvez gérer le déploiement en utilisant le rôle Ansible **ipaserver**.

Conditions préalables

- Vous comprenez les concepts Ansible et IdM :
  - Rôles Ansible
  - Nœuds Ansible
  - Inventaire Ansible
  - Tâches Ansible
  - Modules Ansible
  - Jeux et carnets de jeu Ansible

2.1. ANSIBLE ET SES AVANTAGES POUR L'INSTALLATION D'IDM

Ansible est un outil d'automatisation utilisé pour configurer des systèmes, déployer des logiciels et effectuer des mises à jour continues. Ansible inclut la prise en charge de la gestion des identités (IdM) et vous pouvez utiliser des modules Ansible pour automatiser les tâches d'installation telles que la configuration d'un serveur IdM, d'un réplica, d'un client ou d'une topologie IdM complète.

Avantages de l'utilisation d'Ansible pour l'installation d'IdM
La liste suivante présente les avantages de l'installation de la gestion des identités à l'aide d'Ansible par rapport à une installation manuelle.

- Il n'est pas nécessaire de se connecter au nœud géré.
- Il n'est pas nécessaire de configurer les paramètres de chaque hôte à déployer individuellement. Au lieu de cela, vous pouvez avoir un seul fichier d'inventaire pour déployer un cluster complet.
- Vous pouvez réutiliser un fichier d'inventaire ultérieurement pour des tâches de gestion, par exemple pour ajouter des utilisateurs et des hôtes. Vous pouvez réutiliser un fichier d'inventaire même pour des tâches qui ne sont pas liées à l'IdM.

Ressources supplémentaires

- Automatiser l'installation de Red Hat Identity Management
- Planification de la gestion de l'identité
- Préparation du système pour l'installation du serveur IdM

2.2. INSTALLATION DU PAQUET ANSIBLE-FREEIPA

Cette section décrit comment installer le paquetage **ansible-freeipa** qui fournit des rôles et des modules Ansible pour l'installation et la gestion de la gestion des identités (IdM).

Conditions préalables

- Sur le site **managed node**:
  - Assurez-vous que le nœud géré est un système Red Hat Enterprise Linux 8 avec une adresse IP statique et un gestionnaire de paquets fonctionnel.
- Sur le site **controller**:
  - Assurez-vous que le contrôleur est un système Red Hat Enterprise Linux avec un abonnement valide. Si ce n'est pas le cas, consultez la documentation officielle Ansible Guide d'installation pour obtenir d'autres instructions d'installation.
  - Assurez-vous que vous pouvez atteindre le nœud géré via le protocole **SSH** à partir du contrôleur. Vérifiez que le nœud géré est répertorié dans le fichier **/root/.ssh/known_hosts** du contrôleur.

Procédure

Suivez la procédure suivante sur le contrôleur Ansible.

1. Si votre système fonctionne sous RHEL 8.5 ou une version antérieure, activez le dépôt requis :

```
# subscription-manager repos --enable ansible-2.8-for-rhel-8-x86_64-rpms
```

2. Si votre système fonctionne sous RHEL 8.5 ou une version antérieure, installez le paquetage **ansible**:

```
# dnf install ansible
```

3. Installez le paquetage **ansible-freeipa**:

```
# dnf install ansible-freeipa
```

Les rôles et les modules sont installés dans les répertoires **/usr/share/ansible/roles/** et **/usr/share/ansible/plugins/modules**.

2.3. EMPLACEMENT DES RÔLES ANSIBLE DANS LE SYSTÈME DE FICHIERS

Par défaut, les rôles **ansible-freeipa** sont installés dans le répertoire **/usr/share/ansible/roles/**. La structure du paquetage **ansible-freeipa** est la suivante :

- Le répertoire **/usr/share/ansible/roles/** stocke les rôles **ipaserver**, **ipareplica** et **ipaclient** sur le contrôleur Ansible. Chaque répertoire de rôle contient des exemples, une présentation de base, la licence et la documentation sur le rôle dans un fichier Markdown **README.md**.

```
[root@server]# ls -1 /usr/share/ansible/roles/
ipaclient
ipareplica
ipaserver
```

- Le répertoire **/usr/share/doc/ansible-freeipa/** contient la documentation sur les rôles individuels et la topologie dans des fichiers Markdown **README.md**. Il contient également le sous-répertoire **playbooks/**.

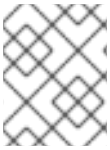
```
[root@server]# ls -1 /usr/share/doc/ansible-freeipa/
playbooks
README-client.md
README.md
README-replica.md
README-server.md
README-topology.md
```

- Le répertoire **/usr/share/doc/ansible-freeipa/playbooks/** contient les playbooks d'exemple :

```
[root@server]# ls -1 /usr/share/doc/ansible-freeipa/playbooks/
install-client.yml
install-cluster.yml
install-replica.yml
install-server.yml
uninstall-client.yml
uninstall-cluster.yml
uninstall-replica.yml
uninstall-server.yml
```

2.4. PARAMÉTRAGE D'UN DÉPLOIEMENT AVEC UN DNS INTÉGRÉ ET UNE AC INTÉGRÉE EN TANT QU'AC RACINE

Suivez cette procédure pour configurer le fichier d'inventaire en vue de l'installation d'un serveur IdM avec une autorité de certification intégrée en tant qu'autorité de certification racine dans un environnement qui utilise la solution DNS intégrée IdM.



NOTE

L'inventaire de cette procédure utilise le format **INI**. Vous pouvez également utiliser les formats **YAML** ou **JSON**.

Procédure

1. Ouvrez le fichier d'inventaire pour le modifier. Spécifiez les noms de domaine pleinement qualifiés (**FQDN**) de l'hôte que vous voulez utiliser comme serveur IdM. Assurez-vous que le site **FQDN** répond aux critères suivants :
   - Seuls les caractères alphanumériques et les tirets (-) sont autorisés. Les caractères de soulignement, par exemple, ne sont pas autorisés et peuvent entraîner des défaillances du système DNS.
   - Le nom d'hôte doit être en minuscules.
2. Spécifiez les informations relatives au domaine et à la sphère IdM.
3. Spécifiez que vous voulez utiliser le DNS intégré en ajoutant l'option suivante :

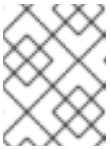
```
ipaserver_setup_dns=yes
```

4. Spécifiez les paramètres de transfert DNS. Choisissez l'une des options suivantes :

   - Utilisez l'option **ipaserver_auto_forwarders=yes** si vous souhaitez que le programme d'installation utilise les redirections du fichier /**etc**/**resolv.conf**. N'utilisez pas cette option si le serveur de noms spécifié dans le fichier /**etc**/**resolv.conf** est l'adresse localhost 127.0.0.1 ou si vous êtes sur un réseau privé virtuel et que les serveurs DNS que vous utilisez sont normalement inaccessibles depuis l'internet public.

   - Utilisez l'option **ipaserver_forwarders** pour spécifier manuellement vos transitaires. Le processus d'installation ajoute les adresses IP des transitaires au fichier /**etc**/**named.conf** du serveur IdM installé.

   - L'option **ipaserver_no_forwarders=yes** permet de configurer les serveurs DNS racine à utiliser à la place.

   NOTE

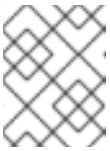
   En l'absence de transitaires DNS, votre environnement est isolé et les noms des autres domaines DNS de votre infrastructure ne sont pas résolus.

5. Spécifiez les paramètres de l'enregistrement inverse et de la zone DNS. Choisissez parmi les options suivantes :

   - Utilisez l'option **ipaserver_allow_zone_overlap=yes** pour autoriser la création d'une zone (inverse) même si la zone est déjà résoluble.

   - Utilisez l'option **ipaserver_reverse_zones** pour spécifier manuellement vos zones inversées.

   - Utilisez l'option **ipaserver_no_reverse=yes** si vous ne souhaitez pas que le programme d'installation crée une zone DNS inversée.

   NOTE

   L'utilisation d'IdM pour gérer les zones inversées est facultative. Vous pouvez utiliser un service DNS externe à cette fin.

6. Spécifiez les mots de passe pour **admin** et pour **Directory Manager**. Utilisez Ansible Vault pour stocker le mot de passe, et faites référence au fichier Vault à partir du fichier playbook. Une autre solution, moins sûre, consiste à spécifier les mots de passe directement dans le fichier d'inventaire.

7. (Facultatif) Spécifiez une zone **firewalld** personnalisée à utiliser par le serveur IdM. Si vous ne définissez pas de zone personnalisée, IdM ajoutera ses services à la zone par défaut **firewalld**. La zone prédéfinie par défaut est **public**.

   IMPORTANT

   La zone **firewalld** spécifiée doit exister et être permanente.

Exemple de fichier d'inventaire contenant les informations requises sur le serveur (à l'exception des mots de passe)

```
[ipaserver]
```

```
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
[...]
```

Exemple de fichier d'inventaire contenant les informations requises sur le serveur (y compris les mots de passe)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Exemple de fichier d'inventaire avec une zone personnalisée `firewalld`

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

Exemple de playbook pour configurer un serveur IdM en utilisant les mots de passe de l'administrateur et du gestionnaire d'annuaire stockés dans un fichier Ansible Vault

```
---
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml

  roles:
  - role: ipaserver
    state: present
```

Exemple de playbook pour configurer un serveur IdM en utilisant les mots de passe de l'administrateur et du gestionnaire d'annuaire à partir d'un fichier d'inventaire

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

  roles:
  - role: ipaserver
    state: present
```

Ressources supplémentaires

- Pour les paramètres par défaut de la politique de transfert, voir la description de **--forward-policy** dans la page de manuel **ipa-dns-install(1)**.
- Pour plus d'informations sur les variables DNS utilisées par le rôle **ipaserver**, voir la section Variables DNS dans le fichier **README-server.md** du répertoire **/usr/share/doc/ansible-freeipa**.

2.5. PARAMÉTRAGE D'UN DÉPLOIEMENT AVEC DNS EXTERNE ET UNE AUTORITÉ DE CERTIFICATION INTÉGRÉE EN TANT QU'AUTORITÉ DE CERTIFICATION RACINE

Suivez cette procédure pour configurer le fichier d'inventaire en vue de l'installation d'un serveur IdM avec une autorité de certification intégrée en tant qu'autorité de certification racine dans un environnement qui utilise une solution DNS externe.

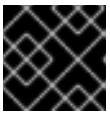
NOTE

Le fichier d'inventaire de cette procédure utilise le format **INI**. Vous pouvez également utiliser les formats **YAML** ou **JSON**.

Procédure

1. Ouvrez le fichier d'inventaire pour le modifier. Spécifiez les noms de domaine pleinement qualifiés (**FQDN**) de l'hôte que vous voulez utiliser comme serveur IdM. Assurez-vous que le site **FQDN** répond aux critères suivants :
   - Seuls les caractères alphanumériques et les tirets (-) sont autorisés. Les caractères de soulignement, par exemple, ne sont pas autorisés et peuvent entraîner des défaillances du système DNS.
   - Le nom d'hôte doit être en minuscules.
2. Spécifiez les informations relatives au domaine et à la sphère IdM.
3. Assurez-vous que l'option **ipaserver_setup_dns** est définie sur **no** ou qu'elle est absente.
4. Spécifiez les mots de passe pour **admin** et pour **Directory Manager**. Utilisez Ansible Vault pour stocker le mot de passe, et faites référence au fichier Vault à partir du fichier playbook. Une autre solution, moins sûre, consiste à spécifier les mots de passe directement dans le fichier d'inventaire.

5. (Facultatif) Spécifiez une zone **firewalld** personnalisée à utiliser par le serveur IdM. Si vous ne définissez pas de zone personnalisée, IdM ajoutera ses services à la zone par défaut **firewalld**. La zone prédéfinie par défaut est **public**.



IMPORTANT

La zone **firewalld** spécifiée doit exister et être permanente.

Exemple de fichier d'inventaire contenant les informations requises sur le serveur (à l'exception des mots de passe)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

Exemple de fichier d'inventaire contenant les informations requises sur le serveur (y compris les mots de passe)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Exemple de fichier d'inventaire avec une zone personnalisée firewalld

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone
```

Exemple de playbook pour configurer un serveur IdM en utilisant les mots de passe de l'administrateur et du gestionnaire d'annuaire stockés dans un fichier Ansible Vault

```
---
```

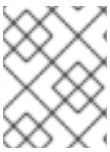
```
- name: Playbook to configure IPA server
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml

  roles:
  - role: ipaserver
    state: present
```

Exemple de playbook pour configurer un serveur IdM en utilisant les mots de passe de l'administrateur et du gestionnaire d'annuaire à partir d'un fichier d'inventaire

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

  roles:
  - role: ipaserver
    state: present
```

2.6. DÉPLOIEMENT D'UN SERVEUR IDM AVEC UNE AUTORITÉ DE CERTIFICATION INTÉGRÉE EN TANT QU'AUTORITÉ DE CERTIFICATION RACINE À L'AIDE D'UN PLAYBOOK ANSIBLE

Suivez cette procédure pour déployer un serveur IdM avec une autorité de certification (AC) intégrée en tant qu'AC racine à l'aide d'un playbook Ansible.



NOTE

L'inventaire de cette procédure utilise le format **INI**. Vous pouvez également utiliser les formats **YAML** ou **JSON**.

Conditions préalables

- Vous avez défini les paramètres correspondant à votre scénario en choisissant l'une des procédures suivantes :
  - Procédure avec DNS intégré
  - Procédure avec DNS externe

Procédure

1. Exécutez la commande **ansible-playbook** avec le nom du fichier playbook, par exemple **install-server.yml**. Spécifiez le fichier d'inventaire avec l'option **-i**:

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/install-
server.yml
```

Spécifiez le niveau de verbosité en utilisant l'option **-v**, **-vv**, ou **-vvv**.

Vous pouvez visualiser la sortie du script Ansible playbook sur l'interface de ligne de commande (CLI). La sortie suivante montre que le script s'est exécuté avec succès puisque 0 tâche a échoué :

```
PLAY RECAP
server.idm.example.com : ok=18   changed=10   unreachable=0   failed=0   skipped=21
rescued=0   ignored=0
```

2. Choisissez l'une des options suivantes :

- Si votre déploiement IdM utilise un DNS externe : ajoutez les enregistrements de ressources DNS contenus dans le fichier **/tmp/ipa.system.records.UFRPto.db** aux serveurs DNS externes existants. Le processus de mise à jour des enregistrements DNS varie en fonction de la solution DNS utilisée.

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```

IMPORTANT

L'installation du serveur n'est pas terminée tant que vous n'avez pas ajouté les enregistrements DNS aux serveurs DNS existants.

- Si votre déploiement IdM utilise le DNS intégré :
  - Ajouter la délégation DNS du domaine parent au domaine DNS IdM. Par exemple, si le domaine DNS IdM est ***idm.example.com***ajoutez un enregistrement de serveur de noms (NS) au domaine parent **example.com**.

IMPORTANT

Répétez cette étape chaque fois qu'un serveur DNS IdM est installé.

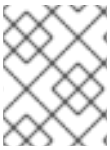
  - Ajoutez un enregistrement de service **_ntp._udp** (SRV) pour votre serveur de temps à votre DNS IdM. La présence de l'enregistrement SRV pour le serveur de temps du serveur IdM nouvellement installé dans le DNS IdM garantit que les futures installations de répliques et de clients sont automatiquement configurées pour se synchroniser avec le serveur de temps utilisé par ce serveur IdM primaire.

Ressources supplémentaires

- Pour savoir comment déployer un serveur IdM avec une autorité de certification **external** en tant qu'autorité de certification racine, voir Déploiement d'un serveur IdM avec une autorité de certification externe en tant qu'autorité de certification racine à l'aide d'un playbook Ansible

2.7. PARAMÉTRAGE D'UN DÉPLOIEMENT AVEC UN DNS INTÉGRÉ ET UNE AUTORITÉ DE CERTIFICATION EXTERNE COMME AUTORITÉ DE CERTIFICATION RACINE

Suivez cette procédure pour configurer le fichier d'inventaire afin d'installer un serveur IdM avec une autorité de certification externe en tant qu'autorité de certification racine dans un environnement qui utilise la solution DNS intégrée IdM.



NOTE

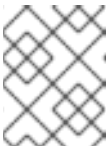
Le fichier d'inventaire de cette procédure utilise le format **INI**. Vous pouvez également utiliser les formats **YAML** ou **JSON**.

Procédure

1. Ouvrez le fichier d'inventaire pour le modifier. Spécifiez les noms de domaine pleinement qualifiés (**FQDN**) de l'hôte que vous voulez utiliser comme serveur IdM. Assurez-vous que le site **FQDN** répond aux critères suivants :
   - Seuls les caractères alphanumériques et les tirets (-) sont autorisés. Les caractères de soulignement, par exemple, ne sont pas autorisés et peuvent entraîner des défaillances du système DNS.
   - Le nom d'hôte doit être en minuscules.
2. Spécifiez les informations relatives au domaine et à la sphère IdM.
3. Spécifiez que vous voulez utiliser le DNS intégré en ajoutant l'option suivante :

```
ipaserver_setup_dns=yes
```

4. Spécifiez les paramètres de transfert DNS. Choisissez l'une des options suivantes :
   - Utilisez l'option **ipaserver_auto_forwarders=yes** si vous souhaitez que le processus d'installation utilise les redirections du fichier **/etc/resolv.conf**. Cette option n'est pas recommandée si le serveur de noms spécifié dans le fichier **/etc/resolv.conf** est l'adresse localhost 127.0.0.1 ou si vous êtes sur un réseau privé virtuel et que les serveurs DNS que vous utilisez sont normalement inaccessibles depuis l'internet public.
   - Utilisez l'option **ipaserver_forwarders** pour spécifier manuellement vos transitaires. Le processus d'installation ajoute les adresses IP des transitaires au fichier **/etc/named.conf** du serveur IdM installé.
   - L'option **ipaserver_no_forwarders=yes** permet de configurer les serveurs DNS racine à utiliser à la place.



NOTE

En l'absence de transitaires DNS, votre environnement est isolé et les noms des autres domaines DNS de votre infrastructure ne sont pas résolus.

5. Spécifiez les paramètres de l'enregistrement inverse et de la zone DNS. Choisissez parmi les options suivantes :
   - Utilisez l'option **ipaserver_allow_zone_overlap=yes** pour autoriser la création d'une zone (inverse) même si la zone est déjà résoluble.
   - Utilisez l'option **ipaserver_reverse_zones** pour spécifier manuellement vos zones inversées.

- Utilisez l'option **ipaserver_no_reverse=yes** si vous ne souhaitez pas que le processus d'installation crée une zone DNS inversée.

NOTE

L'utilisation d'IdM pour gérer les zones inversées est facultative. Vous pouvez utiliser un service DNS externe à cette fin.

6. Spécifiez les mots de passe pour **admin** et pour **Directory Manager**. Utilisez Ansible Vault pour stocker le mot de passe, et faites référence au fichier Vault à partir du fichier playbook. Une autre solution, moins sûre, consiste à spécifier les mots de passe directement dans le fichier d'inventaire.

7. (Facultatif) Spécifiez une zone **firewalld** personnalisée à utiliser par le serveur IdM. Si vous ne définissez pas de zone personnalisée, IdM ajoute ses services à la zone par défaut **firewalld**. La zone prédéfinie par défaut est **public**.

IMPORTANT

La zone **firewalld** spécifiée doit exister et être permanente.

Exemple de fichier d'inventaire contenant les informations requises sur le serveur (à l'exception des mots de passe)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
[...]
```

Exemple de fichier d'inventaire contenant les informations requises sur le serveur (y compris les mots de passe)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Exemple de fichier d'inventaire avec une zone personnalisée firewalld

```
[ipaserver]
```

```
server.idm.example.com

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=yes
ipaserver_auto_forwarders=yes
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone

[...]
```

8. Créez un playbook pour la première étape de l'installation. Saisissez les instructions pour générer la demande de signature de certificat (CSR) et la copier du contrôleur vers le nœud géré.

```
---
- name: Playbook to configure IPA server Step 1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_ca: yes

  roles:
  - role: ipaserver
    state: present

  post_tasks:
  - name: Copy CSR /root/ipa.csr from node to "{{ groups.ipaserver[0] + '-ipa.csr' }}"
    fetch:
      src: /root/ipa.csr
      dest: "{{ groups.ipaserver[0] + '-ipa.csr' }}"
      flat: yes
```

9. Créez un autre playbook pour la dernière étape de l'installation.

```
---
- name: Playbook to configure IPA server Step -1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_cert_files: "/root/chain.crt"

  pre_tasks:
  - name: Copy "{{ groups.ipaserver[0] + '-chain.crt' }}" to /root/chain.crt on node
    copy:
      src: "{{ groups.ipaserver[0] + '-chain.crt' }}"
      dest: "/root/chain.crt"
      force: yes
```

```
roles:
- role: ipaserver
  state: present
```

Ressources supplémentaires

- Pour les paramètres par défaut de la politique de transfert, voir la description de **--forward-policy** dans la page de manuel **ipa-dns-install(1)**.
- Pour plus d'informations sur les variables DNS utilisées par le rôle **ipaserver**, voir la section Variables DNS dans le fichier **README-server.md** du répertoire **/usr/share/doc/ansible-freeipa**.

2.8. PARAMÉTRAGE D'UN DÉPLOIEMENT AVEC DNS EXTERNE ET UNE AUTORITÉ DE CERTIFICATION EXTERNE EN TANT QU'AUTORITÉ DE CERTIFICATION RACINE

Suivez cette procédure pour configurer le fichier d'inventaire en vue de l'installation d'un serveur IdM avec une autorité de certification externe en tant qu'autorité de certification racine dans un environnement qui utilise une solution DNS externe.

NOTE

Le fichier d'inventaire de cette procédure utilise le format **INI**. Vous pouvez également utiliser les formats **YAML** ou **JSON**.

Procédure

1. Ouvrez le fichier d'inventaire pour le modifier. Spécifiez les noms de domaine pleinement qualifiés (**FQDN**) de l'hôte que vous voulez utiliser comme serveur IdM. Assurez-vous que le site **FQDN** répond aux critères suivants :
   - Seuls les caractères alphanumériques et les tirets (-) sont autorisés. Les caractères de soulignement, par exemple, ne sont pas autorisés et peuvent entraîner des défaillances du système DNS.
   - Le nom d'hôte doit être en minuscules.
2. Spécifiez les informations relatives au domaine et à la sphère IdM.
3. Assurez-vous que l'option **ipaserver_setup_dns** est définie sur **no** ou qu'elle est absente.
4. Spécifiez les mots de passe pour **admin** et pour **Directory Manager**. Utilisez Ansible Vault pour stocker le mot de passe, et faites référence au fichier Vault à partir du fichier playbook. Une autre solution, moins sûre, consiste à spécifier les mots de passe directement dans le fichier d'inventaire.
5. (Facultatif) Spécifiez une zone **firewalld** personnalisée à utiliser par le serveur IdM. Si vous ne définissez pas de zone personnalisée, IdM ajoutera ses services à la zone par défaut **firewalld**. La zone prédéfinie par défaut est **public**.

IMPORTANT

La zone **firewalld** spécifiée doit exister et être permanente.

Exemple de fichier d'inventaire contenant les informations requises sur le serveur (à l'exception des mots de passe)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

Exemple de fichier d'inventaire contenant les informations requises sur le serveur (y compris les mots de passe)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234

[...]
```

Exemple de fichier d'inventaire avec une zone personnalisée `firewalld`

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
ipaadmin_password=MySecretPassword123
ipadm_password=MySecretPassword234
ipaserver_firewalld_zone=custom zone

[...]
```

6. Créez un playbook pour la première étape de l'installation. Saisissez les instructions pour générer la demande de signature de certificat (CSR) et la copier du contrôleur vers le nœud géré.

```
---
- name: Playbook to configure IPA server Step 1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_ca: yes
```

```
  roles:
  - role: ipaserver
    state: present

  post_tasks:
  - name: Copy CSR /root/ipa.csr from node to "{{ groups.ipaserver[0] + '-ipa.csr' }}"
    fetch:
      src: /root/ipa.csr
      dest: "{{ groups.ipaserver[0] + '-ipa.csr' }}"
      flat: yes
```

7. Créez un autre playbook pour la dernière étape de l'installation.

```
---
- name: Playbook to configure IPA server Step -1
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_cert_files: "/root/chain.crt"

  pre_tasks:
  - name: Copy "{{ groups.ipaserver[0] + '-chain.crt' }}" to /root/chain.crt on node
    copy:
      src: "{{ groups.ipaserver[0] + '-chain.crt' }}"
      dest: "/root/chain.crt"
      force: yes

  roles:
  - role: ipaserver
    state: present
```

Ressources supplémentaires

- Pour plus de détails sur les options disponibles lors de l'installation d'un serveur IdM avec DNS externe et une autorité de certification signée en externe, voir Installation d'un serveur IdM : Sans DNS intégré, avec une autorité de certification externe comme autorité de certification racine.

2.9. DÉPLOIEMENT D'UN SERVEUR IDM AVEC UNE AUTORITÉ DE CERTIFICATION EXTERNE EN TANT QU'AUTORITÉ DE CERTIFICATION RACINE À L'AIDE D'UN PLAYBOOK ANSIBLE

Suivez cette procédure pour déployer un serveur IdM avec une autorité de certification (AC) externe en tant qu'AC racine à l'aide d'un livre de jeu Ansible.

NOTE

Le fichier d'inventaire de cette procédure utilise le format **INI**. Vous pouvez également utiliser les formats **YAML** ou **JSON**.

Conditions préalables

- Vous avez défini les paramètres correspondant à votre scénario en choisissant l'une des procédures suivantes :
  - Procédure avec DNS intégré
  - Procédure avec DNS externe

Procédure

1. Exécutez la commande **ansible-playbook** avec le nom du fichier playbook qui contient les instructions pour la première étape de l'installation, par exemple **install-server-step1.yml**. Spécifiez le fichier d'inventaire avec l'option **-i**:

   ```
   $ ansible-playbook --vault-password-file=password_file -v -i
   <path_to_inventory_directory>/host.server <path_to_playbooks_directory>/install-
   server-step1.yml
   ```

   Spécifiez le niveau de verbosité en utilisant l'option **-v**, **-vv** ou **-vvv**.

   Vous pouvez visualiser la sortie du script Ansible playbook sur l'interface de ligne de commande (CLI). La sortie suivante montre que le script s'est exécuté avec succès puisque 0 tâche a échoué :

   ```
   PLAY RECAP
   server.idm.example.com : ok=18   changed=10   unreachable=0   failed=0   skipped=21
   rescued=0   ignored=0
   ```

2. Localisez le fichier de demande de signature de certificat **ipa.csr** sur le contrôleur et soumettez-le à l'autorité de certification externe.

3. Placez le certificat de l'autorité de certification IdM signé par l'autorité de certification externe dans le système de fichiers du contrôleur de manière à ce que le manuel de jeu de l'étape suivante puisse le trouver.

4. Exécutez la commande **ansible-playbook** avec le nom du fichier playbook qui contient les instructions pour la dernière étape de l'installation, par exemple **install-server-step2.yml**. Spécifiez le fichier d'inventaire avec l'option **-i**:

   ```
   $ ansible-playbook -v -i <path_to_inventory_directory>/host.server
   <path_to_playbooks_directory>/install-server-step2.yml
   ```

5. Choisissez l'une des options suivantes :

   - Si votre déploiement IdM utilise un DNS externe : ajoutez les enregistrements de ressources DNS contenus dans le fichier **/tmp/ipa.system.records.UFRPto.db** aux serveurs DNS externes existants. Le processus de mise à jour des enregistrements DNS varie en fonction de la solution DNS utilisée.

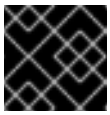
     ```
     ...
     Restarting the KDC
     Please add records in this file to your DNS system:
     ```

```
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```

IMPORTANT

L'installation du serveur n'est pas terminée tant que vous n'avez pas ajouté les enregistrements DNS aux serveurs DNS existants.

- Si votre déploiement IdM utilise le DNS intégré :
  - Ajouter la délégation DNS du domaine parent au domaine DNS IdM. Par exemple, si le domaine DNS IdM est ***idm.example.com***ajoutez un enregistrement de serveur de noms (NS) au domaine parent **example.com**.

    IMPORTANT

    Répétez cette étape chaque fois qu'un serveur DNS IdM est installé.

  - Ajoutez un enregistrement de service **_ntp._udp** (SRV) pour votre serveur de temps à votre DNS IdM. La présence de l'enregistrement SRV pour le serveur de temps du serveur IdM nouvellement installé dans le DNS IdM garantit que les futures installations de répliques et de clients sont automatiquement configurées pour se synchroniser avec le serveur de temps utilisé par ce serveur IdM primaire.

Ressources supplémentaires

Pour savoir comment déployer un serveur IdM avec une autorité de certification **integrated** en tant qu'autorité de certification racine, voir Déploiement d'un serveur IdM avec une autorité de certification intégrée en tant qu'autorité de certification racine à l'aide d'un livre de jeu Ansible

2.10. RESSOURCES SUPPLÉMENTAIRES

- Planification de la topologie du réplica
- Sauvegarde et restauration des serveurs IdM à l'aide de playbooks Ansible
- Notions d'inventaire : formats, hôtes et groupes
- Vous pouvez consulter des exemples de playbooks Ansible pour l'installation d'un serveur IdM et une liste de variables possibles dans la documentation en amont de **ansible-freeipa** .

CHAPITRE 3. INSTALLATION D'UNE RÉPLIQUE DE GESTION DES IDENTITÉS À L'AIDE D'UN PLAYBOOK ANSIBLE

La configuration d'un système en tant que réplique IdM à l'aide d'Ansible l'inscrit dans un domaine IdM et permet au système d'utiliser les services IdM sur les serveurs IdM du domaine.

Le déploiement est géré par le rôle Ansible **ipareplica**. Le rôle peut utiliser le mode de découverte automatique pour identifier les serveurs IdM, le domaine et d'autres paramètres. Cependant, si vous déployez plusieurs réplicas dans un modèle de type tiers, avec différents groupes de réplicas déployés à différents moments, vous devez définir des serveurs ou des réplicas spécifiques pour chaque groupe.

Conditions préalables

- Vous avez installé le paquet ansible-freeipa sur le nœud de contrôle Ansible.
- Vous comprenez les concepts Ansible et IdM :
  - Rôles Ansible
  - Nœuds Ansible
  - Inventaire Ansible
  - Tâches Ansible
  - Modules Ansible
  - Jeux et carnets de jeu Ansible

Ressources supplémentaires

- Planification de la topologie du réplica

3.1. SPÉCIFICATION DES VARIABLES DE BASE, DE SERVEUR ET DE CLIENT POUR L'INSTALLATION DE LA RÉPLIQUE IDM

Suivez cette procédure pour configurer le fichier d'inventaire en vue de l'installation d'une réplique IdM.

Conditions préalables

- 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. Ouvrez le fichier d'inventaire pour le modifier. Spécifiez les noms de domaine pleinement qualifiés (FQDN) des hôtes qui deviendront des répliques IdM. Les FQDN doivent être des noms DNS valides :

   - Seuls les chiffres, les caractères alphabétiques et les traits d'union (**-**) sont autorisés. Par exemple, les caractères de soulignement ne sont pas autorisés et peuvent entraîner des défaillances du système DNS.

   - Le nom d'hôte doit être en minuscules.

   Exemple d'un fichier hosts d'inventaire simple avec seulement le FQDN des répliques défini

   ```
   [ipareplicas]
   replica1.idm.example.com
   replica2.idm.example.com
   replica3.idm.example.com
   [...]
   ```

   Si le serveur IdM est déjà déployé et que les enregistrements SRV sont correctement définis dans la zone DNS IdM, le script découvre automatiquement toutes les autres valeurs requises.

2. [Facultatif] Fournissez des informations supplémentaires dans le fichier d'inventaire en fonction de la façon dont vous avez conçu votre topologie :

   Scénario 1

   Si vous souhaitez éviter l'autodécouverte et que toutes les répliques répertoriées dans la section **[ipareplicas]** utilisent un serveur IdM spécifique, définissez le serveur dans la section **[ipaservers]** du fichier d'inventaire.

   Exemple de fichier hosts d'inventaire avec le FQDN du serveur IdM et les répliques définies

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

   [ipareplicas]
   replica1.idm.example.com
   replica2.idm.example.com
   replica3.idm.example.com
   [...]
   ```

   Scénario 2

   Par ailleurs, si vous souhaitez éviter la découverte automatique mais déployer des répliques spécifiques avec des serveurs spécifiques, définissez les serveurs pour des répliques spécifiques individuellement dans la section **[ipareplicas]** du fichier d'inventaire.

   Exemple de fichier d'inventaire avec un serveur IdM spécifique défini pour une réplique spécifique

   ```
   [ipaservers]
   server.idm.example.com
   replica1.idm.example.com
   ```

```
[ipareplicas]
replica2.idm.example.com
replica3.idm.example.com ipareplica_servers=replica1.idm.example.com
```

Dans l'exemple ci-dessus, **replica3.idm.example.com** utilise le site **replica1.idm.example.com** déjà déployé comme source de réplication.

Scénario 3

Si vous déployez plusieurs répliques en un seul lot et que le temps vous est compté, le déploiement de répliques à plusieurs niveaux peut vous être utile. Définissez des groupes spécifiques de répliques dans le fichier d'inventaire, par exemple **[ipareplicas_tier1]** et **[ipareplicas_tier2]**, et concevez des séquences distinctes pour chaque groupe dans le livre de séquences **install-replica.yml**.

Exemple de fichier d'inventaire avec des niveaux de répliques définis

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

[ipareplicas_tier1]
replica1.idm.example.com

[ipareplicas_tier2]
replica2.idm.example.com \
ipareplica_servers=replica1.idm.example.com,server.idm.example.com
```

La première entrée de **ipareplica_servers** sera utilisée. La deuxième entrée sera utilisée comme option de repli. Lorsque vous utilisez plusieurs niveaux pour déployer les répliques IdM, vous devez avoir des tâches séparées dans le playbook pour déployer d'abord les répliques du niveau 1 et ensuite les répliques du niveau 2 :

Exemple d'un fichier playbook avec des jeux différents pour des groupes de répliques différents

```
---
- name: Playbook to configure IPA replicas (tier1)
  hosts: ipareplicas_tier1
  become: true

  roles:
  - role: ipareplica
    state: present

- name: Playbook to configure IPA replicas (tier2)
  hosts: ipareplicas_tier2
  become: true

  roles:
  - role: ipareplica
    state: present
```

3. [Facultatif] Fournir des informations supplémentaires concernant **firewalld** et DNS :

Scénario 1

Si vous souhaitez que le réplica utilise une zone **firewalld** spécifique au lieu de la zone par défaut, vous pouvez la spécifier dans le fichier d'inventaire. Cela peut être utile, par exemple, lorsque vous souhaitez utiliser une zone **firewalld** interne pour votre installation IdM au lieu d'une zone publique définie par défaut.
Si vous ne définissez pas de zone personnalisée, IdM ajoutera ses services à la zone par défaut **firewalld**. La zone prédéfinie par défaut est **public**.

IMPORTANT

La zone **firewalld** spécifiée doit exister et être permanente.

Exemple d'un fichier hosts d'inventaire simple avec une zone firewalld personnalisée

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
[...]

[ipareplicas:vars]
ipareplica_firewalld_zone=custom zone
```

Scénario 2

Si vous souhaitez que le réplica héberge le service DNS IdM, ajoutez la ligne **ipareplica_setup_dns=yes** à la section **[ipareplicas:vars]**. En outre, indiquez si vous souhaitez utiliser des redirections DNS par serveur :

- Pour configurer les transferts par serveur, ajoutez la variable **ipareplica_forwarders** et une liste de chaînes à la section **[ipareplicas:vars]**, par exemple : **ipareplica_forwarders=192.0.2.1,192.0.2.2**
- Pour ne pas configurer de forwarders par serveur, ajoutez la ligne suivante à la section **[ipareplicas:vars]**: **ipareplica_no_forwarders=yes**.
- Pour configurer les transitaires par serveur en fonction des transitaires répertoriés dans le fichier /**etc/resolv.conf** du réplica, ajoutez la variable **ipareplica_auto_forwarders** à la section **[ipareplicas:vars]**.

Exemple de fichier d'inventaire avec des instructions pour configurer le DNS et les forwarders par serveur sur les répliques

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

[ipareplicas]
replica1.idm.example.com
replica2.idm.example.com
replica3.idm.example.com
```

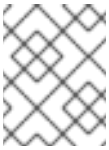
```
[...]

[ipareplicas:vars]
ipareplica_setup_dns=yes
ipareplica_forwarders=192.0.2.1,192.0.2.2
```

Scénario 3

Spécifiez le résolveur DNS à l'aide des options **ipaclient_configure_dns_resolve** et **ipaclient_dns_servers** (le cas échéant) pour simplifier les déploiements de clusters. Ceci est particulièrement utile si votre déploiement IdM utilise un DNS intégré :

Un extrait de fichier d'inventaire spécifiant un résolveur DNS :

```
[...]
[ipaclient:vars]
ipaclient_configure_dns_resolver=true
ipaclient_dns_servers=192.168.100.1
```



NOTE

La liste **ipaclient_dns_servers** ne doit contenir que des adresses IP. Les noms d'hôtes ne sont pas autorisés.

Ressources supplémentaires

- Pour plus d'informations sur les variables **ipareplica**, voir le fichier Markdown **/usr/share/ansible/roles/ipareplica/README.md**.

3.2. SPÉCIFICATION DES INFORMATIONS D'IDENTIFICATION POUR L'INSTALLATION DE LA RÉPLIQUE IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

Suivez cette procédure pour configurer l'autorisation d'installation du réplica IdM.

Conditions préalables

- 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. Spécifiez le **password of a user authorized to deploy replicas**, par exemple l'IdM **admin**.

   - Red Hat recommande d'utiliser Ansible Vault pour stocker le mot de passe et de référencer le fichier Vault à partir du fichier playbook, par exemple **install-replica.yml**:

     Exemple de fichier playbook utilisant le principal d'un fichier d'inventaire et le mot de passe d'un fichier Ansible Vault

     ```
     - name: Playbook to configure IPA replicas
       hosts: ipareplicas
       become: true
       vars_files:
       - playbook_sensitive_data.yml

       roles:
       - role: ipareplica
         state: present
     ```

     Pour plus de détails sur l'utilisation d'Ansible Vault, voir la documentation officielle d'Ansible Vault.

   - De manière moins sûre, fournissez les informations d'identification de **admin** directement dans le fichier d'inventaire. Utilisez l'option **ipaadmin_password** dans la section **[ipareplicas:vars]** du fichier d'inventaire. Le fichier d'inventaire et le fichier playbook **install-replica.yml** peuvent alors se présenter comme suit :

     Exemple de fichier hosts.replica de l'inventaire

     ```
     [...]
     [ipareplicas:vars]
     ipaadmin_password=Secret123
     ```

     Exemple de playbook utilisant le principal et le mot de passe du fichier d'inventaire

     ```
     - name: Playbook to configure IPA replicas
       hosts: ipareplicas
       become: true

       roles:
       - role: ipareplica
         state: present
     ```

   - Une autre solution, moins sûre, consiste à fournir les informations d'identification d'un autre utilisateur autorisé à déployer une réplique directement dans le fichier d'inventaire. Pour spécifier un autre utilisateur autorisé, utilisez l'option **ipaadmin_principal** pour le nom d'utilisateur et l'option **ipaadmin_password** pour le mot de passe. Le fichier d'inventaire et le fichier playbook **install-replica.yml** peuvent alors se présenter comme suit :

     Exemple de fichier hosts.replica de l'inventaire

     ```
     [...]
     [ipareplicas:vars]
     ipaadmin_principal=my_admin
     ```

```
ipaadmin_password=my_admin_secret123
```

Exemple de playbook utilisant le principal et le mot de passe du fichier d'inventaire

```
- name: Playbook to configure IPA replicas
  hosts: ipareplicas
  become: true

  roles:
  - role: ipareplica
    state: present
```

Ressources supplémentaires

- Pour plus de détails sur les options acceptées par le rôle Ansible **ipareplica**, voir le fichier Markdown **/usr/share/ansible/roles/ipareplica/README.md**.

3.3. DÉPLOIEMENT D'UNE RÉPLIQUE IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

Complétez cette procédure pour utiliser un playbook Ansible afin de déployer une réplique IdM.

Conditions préalables

- Vous avez configuré le fichier d'inventaire pour l'installation d'une réplique IdM .
- Vous avez configuré l'autorisation pour l'installation du réplica IdM .

Procédure

- Pour installer une réplique IdM à l'aide d'un playbook Ansible, utilisez la commande **ansible-playbook** avec le nom du fichier du playbook, par exemple **install-replica.yml**. Spécifiez le fichier d'inventaire avec l'option **-i**:

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts.replica <path_to_playbooks_directory>/install-
replica.yml
```

  Spécifiez le niveau de verbosité en utilisant l'option **-v**, **-vv** ou **-vvv**.

  Ansible vous informe de l'exécution du script du playbook Ansible. La sortie suivante montre que le script s'est exécuté avec succès puisque 0 tâche a échoué :

```
PLAY RECAP
replica.idm.example.com : ok=18   changed=10   unreachable=0   failed=0   skipped=21
rescued=0   ignored=0
```

Vous avez maintenant installé une réplique IdM.

CHAPITRE 4. INSTALLATION D'UN CLIENT DE GESTION DES IDENTITÉS À L'AIDE D'UN PLAYBOOK ANSIBLE

Les sections suivantes décrivent comment configurer un système en tant que client de gestion d'identité (IdM) à l'aide d'Ansible. La configuration d'un système en tant que client IdM l'inscrit dans un domaine IdM et permet au système d'utiliser les services IdM sur les serveurs IdM du domaine.

Le déploiement est géré par le rôle Ansible **ipaclient**. Par défaut, le rôle utilise le mode de découverte automatique pour identifier les serveurs IdM, le domaine et d'autres paramètres. Le rôle peut être modifié pour que le playbook Ansible utilise les paramètres spécifiés, par exemple dans le fichier d'inventaire.

Conditions préalables

- Vous avez installé le paquet ansible-freeipa sur le nœud de contrôle Ansible.
- Vous comprenez les concepts Ansible et IdM :
  - Rôles Ansible
  - Nœuds Ansible
  - Inventaire Ansible
  - Tâches Ansible
  - Modules Ansible
  - Jeux et carnets de jeu Ansible

4.1. DÉFINITION DES PARAMÈTRES DU FICHIER D'INVENTAIRE POUR LE MODE D'INSTALLATION DU CLIENT D'AUTODÉCOUVERTE

Pour installer un client de gestion des identités à l'aide d'un playbook Ansible, configurez les paramètres de l'hôte cible dans un fichier d'inventaire, par exemple **inventory/hosts**:

- les informations sur l'hôte
- l'autorisation de la tâche

Le fichier d'inventaire peut être dans l'un des nombreux formats, en fonction des plugins d'inventaire que vous avez. Le format **INI-like** est l'un des formats par défaut d'Ansible et est utilisé dans les exemples ci-dessous.

NOTE

Pour utiliser les cartes à puce avec l'interface utilisateur graphique dans RHEL, assurez-vous d'inclure la variable **ipaclient_mkhomedir** dans votre playbook Ansible.

Conditions préalables

- Vous avez vérifié les instructions de déploiement sur le nœud de contrôle, voir Vérification des paramètres dans le fichier install-client.yml.

Procédure

Procédure

1. Indiquez le nom d'hôte entièrement qualifié (FQDN) de l'hôte qui doit devenir un client IdM. Le nom de domaine entièrement qualifié doit être un nom DNS valide :

   - Seuls les chiffres, les caractères alphabétiques et les traits d'union (**-**) sont autorisés. Par exemple, les caractères de soulignement ne sont pas autorisés et peuvent entraîner des défaillances du système DNS.
   - Le nom d'hôte doit être en minuscules. Aucune majuscule n'est autorisée.

   Si les enregistrements SRV sont correctement définis dans la zone DNS IdM, le script découvre automatiquement toutes les autres valeurs requises.

   Exemple d'un fichier d'inventaire simple avec seulement le FQDN du client défini

   ```
   [ipaclients]
   client.idm.example.com
   [...]
   ```

2. Spécifiez les informations d'identification pour l'inscription du client. Les méthodes d'authentification suivantes sont disponibles :

   - Le site **password of a user authorized to enroll clients**est l'option par défaut.
     - Red Hat recommande d'utiliser Ansible Vault pour stocker le mot de passe et de faire référence au fichier Vault à partir du fichier playbook, par exemple **install-client.yml**, directement :

       Exemple de fichier playbook utilisant le principal d'un fichier d'inventaire et le mot de passe d'un fichier Ansible Vault

       ```
       - name: Playbook to configure IPA clients with username/password
         hosts: ipaclients
         become: true
         vars_files:
         - playbook_sensitive_data.yml

         roles:
         - role: ipaclient
           state: present
       ```

     - De manière moins sûre, fournissez les informations d'identification de **admin** en utilisant l'option **ipaadmin_password** dans la section **[ipaclients:vars]** du fichier **inventory/hosts**. Pour spécifier un autre utilisateur autorisé, utilisez l'option **ipaadmin_principal** pour le nom d'utilisateur et l'option **ipaadmin_password** pour le mot de passe. Le fichier d'inventaire **inventory/hosts** et le fichier playbook **install-client.yml** peuvent alors se présenter comme suit :

       Exemple de fichier d'inventaire des hôtes

       ```
       [...]
       [ipaclients:vars]
       ipaadmin_principal=my_admin
       ipaadmin_password=Secret123
       ```

Exemple de Playbook utilisant le principal et le mot de passe du fichier d'inventaire

```
- name: Playbook to unconfigure IPA clients
  hosts: ipaclients
  become: true

  roles:
  - role: ipaclient
    state: true
```

- Le site **client keytab** de l'inscription précédente, s'il est encore disponible.
  Cette option est disponible si le système a été précédemment enregistré en tant que client de gestion d'identité. Pour utiliser cette méthode d'authentification, décommentez l'option **#ipaclient_keytab**, en spécifiant le chemin d'accès au fichier stockant le keytab, par exemple dans la section **[ipaclient:vars]** de **inventory/hosts**.

- Un **random, one-time password** (OTP) à générer lors de l'inscription. Pour utiliser cette méthode d'authentification, utilisez l'option **ipaclient_use_otp=yes** dans votre fichier d'inventaire. Par exemple, vous pouvez décommenter l'option **ipaclient_use_otp=yes** dans la section **[ipaclients:vars]** du fichier **inventory/hosts**. Notez qu'avec l'option OTP, vous devez également spécifier l'une des options suivantes :

  - L'adresse **password of a user authorized to enroll clients**, par exemple en fournissant une valeur pour **ipaadmin_password** dans la section **[ipaclients:vars]** du fichier **inventory/hosts**.

  - Le site **admin keytab**, par exemple en fournissant une valeur pour **ipaadmin_keytab** dans la section **[ipaclients:vars]** de **inventory/hosts**.

3. [Facultatif] Spécifiez le résolveur DNS à l'aide des options **ipaclient_configure_dns_resolve** et **ipaclient_dns_servers** (le cas échéant) pour simplifier les déploiements de clusters. Ceci est particulièrement utile si votre déploiement IdM utilise le DNS intégré :

   #### Un extrait de fichier d'inventaire spécifiant un résolveur DNS :

```
[...]
[ipaclients:vars]
ipaadmin_password: "{{ ipaadmin_password }}"
ipaclient_domain=idm.example.com
ipaclient_configure_dns_resolver=true
ipaclient_dns_servers=192.168.100.1
```

NOTE

La liste **ipaclient_dns_servers** ne doit contenir que des adresses IP. Les noms d'hôtes ne sont pas autorisés.

Ressources supplémentaires

- **/usr/share/ansible/roles/ipaclient/README.md**

4.2. DÉFINITION DES PARAMÈTRES DU FICHIER D'INVENTAIRE LORSQUE L'AUTODÉCOUVERTE N'EST PAS POSSIBLE LORS DE L'INSTALLATION DU CLIENT

Pour installer un client de gestion des identités à l'aide d'un playbook Ansible, configurez les paramètres de l'hôte cible dans un fichier d'inventaire, par exemple **inventory/hosts**:

- les informations sur l'hôte, le serveur IdM et le domaine IdM ou le realm IdM
- l'autorisation de la tâche

Le fichier d'inventaire peut être dans l'un des nombreux formats, en fonction des plugins d'inventaire que vous avez. Le format **INI-like** est l'un des formats par défaut d'Ansible et est utilisé dans les exemples ci-dessous.

NOTE

Pour utiliser les cartes à puce avec l'interface utilisateur graphique dans RHEL, assurez-vous d'inclure la variable **ipaclient_mkhomedir** dans votre playbook Ansible.

Conditions préalables

- Vous avez vérifié les instructions de déploiement sur le nœud de contrôle, voir Vérification des paramètres dans le fichier install-client.yml.

Procédure

1. Indiquez le nom d'hôte entièrement qualifié (FQDN) de l'hôte qui doit devenir un client IdM. Le nom de domaine entièrement qualifié doit être un nom DNS valide :
   - Seuls les chiffres, les caractères alphabétiques et les traits d'union (**-**) sont autorisés. Par exemple, les caractères de soulignement ne sont pas autorisés et peuvent entraîner des défaillances du système DNS.
   - Le nom d'hôte doit être en minuscules. Aucune majuscule n'est autorisée.
2. Spécifiez d'autres options dans les sections correspondantes du fichier **inventory/hosts**:
   - le FQDN des serveurs dans la section **[ipaservers]** pour indiquer le serveur IdM auprès duquel le client sera enrôlé
   - l'une des deux options suivantes :
     - l'option **ipaclient_domain** dans la section **[ipaclients:vars]** pour indiquer le nom de domaine DNS du serveur IdM auprès duquel le client sera enrôlé
     - l'option **ipaclient_realm** dans la section **[ipaclients:vars]** pour indiquer le nom du domaine Kerberos contrôlé par le serveur IdM

     Exemple de fichier d'inventaire des hôtes avec le FQDN du client, le FQDN du serveur et le domaine défini

     ```
     [ipaclients]
     client.idm.example.com
     ```

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

[ipaclients:vars]
ipaclient_domain=idm.example.com
[...]
```

3. Spécifiez les informations d'identification pour l'inscription du client. Les méthodes d'authentification suivantes sont disponibles :

   - Le site **password of a user authorized to enroll clients**est l'option par défaut.

     - Red Hat recommande d'utiliser Ansible Vault pour stocker le mot de passe et de référencer le fichier Vault à partir du fichier de script, par exemple **install-client.yml**, directement : .exemple de fichier de script utilisant le principal du fichier d'inventaire et le mot de passe d'un fichier Ansible Vault

```
- name: Playbook to configure IPA clients with username/password
  hosts: ipaclients
  become: true
  vars_files:
  - *playbook_sensitive_data.yml*

  roles:
  - role: ipaclient
    state: present
```

   - De manière moins sûre, fournissez les informations d'identification de **admin** en utilisant l'option **ipaadmin_password** dans la section **[ipaclients:vars]** du fichier **inventory/hosts**. Pour spécifier un autre utilisateur autorisé, utilisez l'option **ipaadmin_principal** pour le nom d'utilisateur et l'option **ipaadmin_password** pour le mot de passe. Le fichier du playbook **install-client.yml** peut alors se présenter comme suit :

     Exemple de fichier d'inventaire des hôtes

```
[...]
[ipaclients:vars]
ipaadmin_principal=my_admin
ipaadmin_password=Secret123
```

     Exemple de Playbook utilisant le principal et le mot de passe du fichier d'inventaire

```
- name: Playbook to unconfigure IPA clients
  hosts: ipaclients
  become: true

  roles:
  - role: ipaclient
    state: true
```

   - Le site **client keytab** de l'inscription précédente, s'il est encore disponible :
     Cette option est disponible si le système a été précédemment enregistré en tant que client de gestion d'identité. Pour utiliser cette méthode d'authentification, décommentez l'option **ipaclient_keytab**, en spécifiant le chemin d'accès au fichier stockant le keytab, par exemple dans la section **[ipaclient:vars]** de **inventory/hosts**.

- Un **random, one-time password** (OTP) à générer lors de l'inscription. Pour utiliser cette méthode d'authentification, utilisez l'option **ipaclient_use_otp=yes** dans votre fichier d'inventaire. Par exemple, vous pouvez décommenter l'option **#ipaclient_use_otp=yes** dans la section **[ipaclients:vars]** du fichier **inventory/hosts**. Notez qu'avec l'option OTP, vous devez également spécifier l'une des options suivantes :
  - L'adresse **password of a user authorized to enroll clients** par exemple en fournissant une valeur pour **ipaadmin_password** dans la section **[ipaclients:vars]** du fichier **inventory/hosts**.
  - Le site **admin keytab**, par exemple en fournissant une valeur pour **ipaadmin_keytab** dans la section **[ipaclients:vars]** de **inventory/hosts**.

Ressources supplémentaires

- Pour plus de détails sur les options acceptées par le rôle Ansible **ipaclient**, voir le fichier README de **/usr/share/ansible/roles/ipaclient/README.md**.

4.3. VÉRIFICATION DES PARAMÈTRES DANS LE FICHIER INSTALL-CLIENT.YML

Le fichier **install-client.yml** playbook contient des instructions pour le déploiement du client IdM.

Procédure

- Ouvrez le fichier et vérifiez si les instructions du playbook correspondent à ce que vous prévoyez pour votre déploiement. Le contenu ressemble généralement à ceci :

```
---
- name: Playbook to configure IPA clients with username/password
  hosts: ipaclients
  become: true

  roles:
  - role: ipaclient
    state: present
```

C'est ce que signifient les différentes entrées :

- L'entrée hosts spécifie la section du fichier **inventory/hosts** dans laquelle le script ansible recherche les **FQDNs** des hôtes sur lesquels le script **ipa-client-install** doit être exécuté.
- L'entrée **become: true** spécifie que les informations d'identification de root seront invoquées lors de l'exécution du script **ipa-client-install**.
- L'entrée **role: ipaclient** spécifie le rôle qui sera installé sur l'hôte : dans ce cas, il s'agit du rôle de client IPA.
- L'entrée **state: present** précise que le client doit être installé plutôt que désinstallé (**absent**).

4.4. OPTIONS D'AUTORISATION POUR L'INSCRIPTION D'UN CLIENT IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

Cette section présente les options d'autorisation individuelle pour l'inscription du client IdM avec des exemples d'inventaire et de fichiers de jeu.

Tableau 4.1. Options d'autorisation pour l'enrôlement du client IdM à l'aide d'Ansible

Option d'autorisation	Note	Exemple de fichier d'inventaire	Exemple de fichier playbook **install-client.yml**

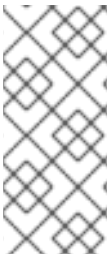
Mot de passe d'un utilisateur autorisé à inscrire un client : Option 1	Mot de passe stocké dans le coffre-fort d'Ansible	`[ipaclients:vars]`
		`[...]`
			`- name: Playbook to configure IPA clients with username/password`
			`  hosts: ipaclients`
			`  become: true`
			`  vars_files:`
			`  - playbook_sensitive_data.yml`
			
			`  roles:`
			`  - role: ipaclient`
			`  state: present`

Mot de passe d'un utilisateur autorisé à inscrire un client : Option 2	Mot de passe stocké dans le fichier d'inventaire	`[ipaclients:vars]`
		`ipaadmin_password=Secret123`
			`- name: Playbook to configure IPA clients`
			`  hosts: ipaclients`
			`  become: true`
			
			`  roles:`
			`  - role: ipaclient`
			`  state: true`

Un mot de passe aléatoire à usage unique (OTP) : Option 1	Mot de passe administrateur OTP	`[ipaclients:vars]`
		`ipaadmin_password=Secret123`
		`ipaclient_use_otp=true`
			`- name: Playbook to configure IPA clients`
			`  hosts: ipaclients`
			`  become: true`
			
			`  roles:`
			`  - role: ipaclient`
			`  state: true`

Un mot de passe aléatoire à usage unique (OTP) : Option 2	OTP un keytab administrateur	`[ipaclients:vars]`
		`ipaadmin_keytab=/root/admin.keytab`
		`ipaclient_use_otp=true`
			`- name: Playbook to configure IPA clients`
			`  hosts: ipaclients`
			`  become: true`
			
			`  roles:`
			`  - role: ipaclient`
			`  state: true`

Option d'autorisation	Note	Exemple de fichier d'inventaire	Exemple de fichier playbook **install-client.yml**
Le fichier clé du client de l'inscription précédente		[ipaclients:vars] ipaclient_keytab=/root/krb5.keytab	- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true



NOTE

Depuis RHEL 9.2, dans les deux scénarios d'autorisation OTP décrits ci-dessus, la demande du TGT de l'administrateur à l'aide de la commande **kinit** se produit sur le premier serveur IdM spécifié ou découvert. Par conséquent, aucune modification supplémentaire du nœud de contrôle Ansible n'est nécessaire. Avant RHEL 9.2, le paquetage **krb5-workstation** était requis sur le nœud de contrôle.

4.5. DÉPLOIEMENT D'UN CLIENT IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

Complétez cette procédure pour utiliser un playbook Ansible afin de déployer un client IdM dans votre environnement IdM.

Conditions préalables

- Vous avez défini les paramètres du déploiement du client IdM en fonction de votre scénario de déploiement :
  - Définition des paramètres du fichier d'inventaire pour le mode d'installation du client d'autodécouverte
  - Définition des paramètres du fichier d'inventaire lorsque l'autodécouverte n'est pas possible lors de l'installation du client
- Vous avez vérifié les paramètres dans install-client.yml.

Procédure

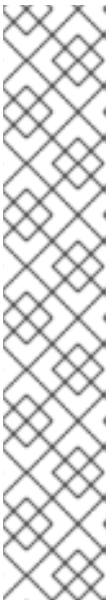
- Pour installer un client IdM à l'aide d'un playbook Ansible, utilisez la commande **ansible-playbook** avec le nom du fichier du playbook, par exemple **install-client.yml**. Spécifiez le fichier d'inventaire avec l'option **-i**:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory/hosts install-client.yml
```

Spécifiez le niveau de verbosité en utilisant l'option **-v**, **-vv** ou **-vvv**.

Ansible vous informe de l'exécution du script du playbook Ansible. La sortie suivante montre que le script s'est exécuté avec succès car aucune tâche n'a échoué :

```
PLAY RECAP
client1.idm.example.com : ok=18 changed=10 unreachable=0 failed=0 skipped=21
rescued=0 ignored=0
```



NOTE

Ansible utilise différentes couleurs pour fournir différents types d'informations sur le processus en cours. Vous pouvez modifier les couleurs par défaut dans la section **[colors]** du fichier **/etc/ansible/ansible.cfg**:

```
[colors]
[...]
#error = red
#debug = dark gray
#deprecate = purple
#skip = cyan
#unreachable = red
#ok = green
#changed = yellow
[...]
```

Vous avez maintenant installé un client IdM sur votre hôte à l'aide d'un playbook Ansible.

4.6. TEST D'UN CLIENT DE GESTION D'IDENTITÉ APRÈS L'INSTALLATION D'ANSIBLE

L'interface de ligne de commande (CLI) vous informe que la commande **ansible-playbook** a réussi, mais vous pouvez également effectuer votre propre test.

Pour vérifier que le client de gestion des identités peut obtenir des informations sur les utilisateurs définis sur le serveur, vérifiez que vous pouvez résoudre un utilisateur défini sur le serveur. Par exemple, pour vérifier l'utilisateur par défaut **admin**:

```
[user@client1 ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

Pour tester que l'authentification fonctionne correctement, **su -** en tant qu'autre utilisateur IdM déjà existant :

```
[user@client1 ~]$ su - idm_user
Last login: Thu Oct 18 18:39:11 CEST 2018 from 192.168.122.1 on pts/0
[idm_user@client1 ~]$
```

4.7. DÉSINSTALLATION D'UN CLIENT IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

Complétez cette procédure pour utiliser un playbook Ansible afin de désinstaller votre hôte en tant que client IdM.

Conditions préalables

- Informations d'identification de l'administrateur de l'IdM.

Procédure

- Pour désinstaller le client IdM, utilisez la commande **ansible-playbook** avec le nom du fichier playbook, par exemple **uninstall-client.yml**. Spécifiez le fichier d'inventaire avec l'option **-i** et, éventuellement, spécifiez le niveau de verbosité en utilisant les options **-v**, **-vv** ou **-vvv**:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory/hosts uninstall-client.yml
```

IMPORTANT

La désinstallation du client supprime uniquement la configuration IdM de base de l'hôte, mais laisse les fichiers de configuration sur l'hôte au cas où vous décideriez de réinstaller le client. En outre, la désinstallation présente les limitations suivantes :

- Elle ne supprime pas l'entrée de l'hôte du client du serveur LDAP IdM. La désinstallation ne fait que désinscrire l'hôte.
- Il ne supprime pas les services résidant sur le client de l'IdM.
- Il ne supprime pas les entrées DNS du serveur IdM pour le client.
- Il ne supprime pas les anciens principes pour les keytabs autres que **/etc/krb5.keytab**.

Notez que la désinstallation supprime tous les certificats émis pour l'hôte par l'autorité de certification IdM.

Ressources supplémentaires

- Voir Désinstallation d'un client IdM.

CHAPITRE 5. PRÉPARATION DE L'ENVIRONNEMENT POUR LA GESTION DE L'IDM À L'AIDE DES PLAYBOOKS ANSIBLE

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**.

Grâce à cette pratique, vous pouvez retrouver tous vos playbooks en un seul endroit.

NOTE

Vous pouvez exécuter vos séquences **ansible-freeipa** sans invoquer les privilèges **root** sur les nœuds gérés. Les exceptions incluent les playbooks qui utilisent les rôles **ipaserver**, **ipareplica**, **ipaclient**, **ipasmartcard_server**, **ipasmartcard_client** et **ipabackup ansible-freeipa** . Ces rôles nécessitent un accès privilégié aux répertoires et au gestionnaire de paquets logiciels **dnf**.

Les playbooks de la documentation de Red Hat Enterprise Linux IdM supposent la configuration de sécurité suivante :

- L'IdM **admin** est votre utilisateur Ansible distant sur les nœuds gérés.
- Vous stockez le mot de passe IdM **admin** crypté dans un coffre-fort Ansible.
- Vous avez placé le mot de passe qui protège le coffre-fort Ansible dans un fichier de mots de passe.
- Vous bloquez l'accès au fichier de mots de passe de l'espace de stockage à tout le monde, sauf à votre utilisateur local ansible.
- Vous devez régulièrement supprimer et recréer le fichier des mots de passe de la chambre forte.

Envisager également d'autres configurations de sécurité .

5.1. PRÉPARATION D'UN NŒUD DE CONTRÔLE ET DE NŒUDS GÉRÉS POUR LA GESTION DE L'IDM À L'AIDE DE PLAYBOOKS ANSIBLE

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
remote_user = admin
```

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

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

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
```

Ces commandes nécessitent la saisie du mot de passe IdM **admin**.

7. Créez un fichier **password_file** qui contient le mot de passe du coffre-fort :

```
redhat
```

8. Modifier les autorisations de modification du fichier :

```
$ chmod 0600 password_file
```

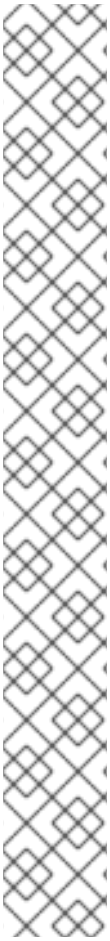
9. Créer un coffre-fort Ansible **secret.yml** pour stocker le mot de passe IdM **admin**:

   a. Configurez **password_file** pour qu'il stocke le mot de passe de l'espace de stockage :

```
$ ansible-vault create --vault-password-file=password_file secret.yml
```

   b. Lorsque vous y êtes invité, saisissez le contenu du fichier **secret.yml**:

```
ipaadmin_password: Secret123
```



NOTE

Pour utiliser la version cryptée de **ipaadmin_password** dans un playbook, vous devez utiliser la directive **vars_file**. Par exemple, un playbook simple pour supprimer un utilisateur IdM peut ressembler à ce qui suit :

```
---
- name: Playbook to handle users
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Delete user robot
    ipauser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: robot
      state: absent
```

Lors de l'exécution d'un playbook, demandez à Ansible d'utiliser le mot de passe de l'espace de stockage pour décrypter **ipaadmin_password** en ajoutant l'option **--vault-password-file=*password_file*** à Ansible. Par exemple :

```
ansible-playbook -i inventory --vault-password-file=password_file del-user.yml
```

AVERTISSEMENT

Pour des raisons de sécurité, supprimez le fichier des mots de passe de la chambre forte à la fin de chaque session et répétez les étapes 7 à 9 au début de chaque nouvelle session.

Ressources supplémentaires

- Différentes méthodes pour fournir les informations d'identification requises pour les playbooks ansible-freeipa
- Installation d'un serveur de gestion des identités à l'aide d'un playbook Ansible
- Comment constituer votre inventaire

5.2. DIFFÉRENTES MÉTHODES POUR FOURNIR LES INFORMATIONS D'IDENTIFICATION REQUISES POUR LES PLAYBOOKS ANSIBLE-FREEIPA

Cette section présente les avantages et les inconvénients des différentes méthodes permettant de fournir les informations d'identification requises pour l'exécution des playbooks qui utilisent les rôles et les modules **ansible-freeipa**.

Stockage des mots de passe en texte clair dans un cahier de jeu

Benefits:

- Ne pas être invité tout le temps à exécuter le manuel de jeu.
- Facile à mettre en œuvre.

Drawbacks:

- Toute personne ayant accès au fichier peut lire le mot de passe. Le fait de définir des autorisations erronées et de partager le fichier, par exemple dans un référentiel interne ou externe, peut compromettre la sécurité.
- Travail de maintenance important : si le mot de passe est modifié, il doit l'être dans tous les playbooks.

Saisie interactive des mots de passe lors de l'exécution d'un playbook

Benefits:

- Personne ne peut voler le mot de passe car il n'est stocké nulle part.
- Vous pouvez facilement mettre à jour le mot de passe.
- Facile à mettre en œuvre.

Drawbacks:

- Si vous utilisez des playbooks Ansible dans des scripts, l'obligation de saisir le mot de passe de manière interactive peut s'avérer gênante.

Stockage des mots de passe dans un coffre-fort Ansible et du mot de passe du coffre-fort dans un fichier :

Benefits:

- Le mot de passe de l'utilisateur est stocké de manière cryptée.
- Vous pouvez facilement mettre à jour le mot de passe de l'utilisateur en créant un nouveau coffre-fort Ansible.

- Vous pouvez mettre à jour le fichier de mots de passe qui protège le coffre-fort ansible facilement, en utilisant la commande **ansible-vault rekey --new-vault-password-file=NEW_VAULT_PASSWORD_FILE secret.yml**.
- Si vous utilisez des playbooks Ansible dans des scripts, il est pratique de ne pas avoir à saisir le mot de passe protégeant le coffre-fort Ansible de manière interactive.

Drawbacks:

- Il est essentiel que le fichier contenant le mot de passe sensible en texte clair soit protégé par des autorisations de fichiers et d'autres mesures de sécurité.

Stocker les mots de passe dans un coffre-fort Ansible et saisir le mot de passe du coffre-fort de manière interactive

Benefits:

- Le mot de passe de l'utilisateur est stocké de manière cryptée.
- Personne ne peut voler le mot de passe du coffre-fort car il n'est stocké nulle part.
- Vous pouvez facilement mettre à jour le mot de passe de l'utilisateur en créant un nouveau coffre-fort Ansible.
- Vous pouvez également mettre à jour le mot de passe de l'espace de stockage facilement, en utilisant la commande **ansible-vault rekey *file_name*** en utilisant la commande

Drawbacks:

- Si vous utilisez des playbooks Ansible dans des scripts, la nécessité de saisir le mot de passe de l'espace de stockage de manière interactive peut s'avérer gênante.

Ressources supplémentaires

- Préparation d'un nœud de contrôle et de nœuds gérés pour la gestion de l'IdM à l'aide de playbooks Ansible
- Qu'est-ce que la confiance zéro ?
- Protéger les données sensibles avec Ansible vault

CHAPITRE 6. CONFIGURATION DES PARAMÈTRES IDM GLOBAUX À L'AIDE DES PLAYBOOKS ANSIBLE

En utilisant le module Ansible **config**, vous pouvez récupérer et définir des paramètres de configuration globale pour la gestion des identités (IdM).

Ce chapitre comprend les sections suivantes :

- Récupération de la configuration IdM à l'aide d'un playbook Ansible
- Configuration du serveur de renouvellement de l'autorité de certification IdM à l'aide d'un carnet de commande Ansible
- Configurer le shell par défaut pour les utilisateurs IdM à l'aide d'un playbook Ansible
- Configurer un nom NETBIOS pour un domaine IdM en utilisant Ansible
- S'assurer que les utilisateurs et les groupes IdM ont des SID en utilisant Ansible

6.1. RÉCUPÉRATION DE LA CONFIGURATION IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

La procédure suivante décrit comment utiliser un playbook Ansible pour récupérer des informations sur la configuration globale actuelle d'IdM.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Ouvrez le fichier **/usr/share/doc/ansible-freeipa/playbooks/config/retrieve-config.yml** Ansible playbook pour l'éditer :

```
---
- name: Playbook to handle global IdM configuration
  hosts: ipaserver
  become: no
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
```

```
tasks:
- name: Query IPA global configuration
  ipaconfig:
    ipaadmin_password: "{{ ipaadmin_password }}"
  register: serverconfig

- debug:
    msg: "{{ serverconfig }}"
```

2. Adaptez le fichier en modifiant les éléments suivants :
   - Le mot de passe de l'administrateur IdM.
   - Autres valeurs, si nécessaire.
3. Enregistrer le fichier.
4. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/config/retrieve-config.yml
[...]
TASK [debug]
ok: [server.idm.example.com] => {
    "msg": {
        "ansible_facts": {
            "discovered_interpreter_
        },
        "changed": false,
        "config": {
            "ca_renewal_master_server": "server.idm.example.com",
            "configstring": [
                "AllowNThash",
                "KDC:Disable Last Success"
            ],
            "defaultgroup": "ipausers",
            "defaultshell": "/bin/bash",
            "emaildomain": "idm.example.com",
            "enable_migration": false,
            "groupsearch": [
                "cn",
                "description"
            ],
            "homedirectory": "/home",
            "maxhostname": "64",
            "maxusername": "64",
            "pac_type": [
                "MS-PAC",
                "nfs:NONE"
            ],
            "pwdexpnotify": "4",
            "searchrecordslimit": "100",
            "searchtimelimit": "2",
            "selinuxusermapdefault": "unconfined_u:s0-s0:c0.c1023",
```

```
            "selinuxusermaporder": [
                "guest_u:s0$xguest_u:s0$user_
            ],
            "usersearch": [
                "uid",
                "givenname",
                "sn",
                "telephonenumber",
                "ou",
                "title"
            ]
        },
        "failed": false
    }
}
```

6.2. CONFIGURATION DU SERVEUR DE RENOUVELLEMENT DE L'AUTORITÉ DE CERTIFICATION IDM À L'AIDE D'UN CARNET DE COMMANDE ANSIBLE

Dans un déploiement de gestion d'identité (IdM) qui utilise une autorité de certification (CA) intégrée, le serveur de renouvellement de la CA maintient et renouvelle les certificats du système IdM. Il garantit la robustesse des déploiements IdM.

Pour plus de détails sur le rôle du serveur de renouvellement de l'autorité de certification IdM, voir Utilisation du serveur de renouvellement de l'autorité de certification IdM .

La procédure suivante décrit comment utiliser un livre de jeu Ansible pour configurer le serveur de renouvellement de l'autorité de certification IdM.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Facultatif : identifiez le serveur de renouvellement de l'autorité de certification IdM :

```
$ ipa config-show | grep 'CA renewal'
  IPA CA renewal master: server.idm.example.com
```

2. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

3. Ouvrez le fichier **/usr/share/doc/ansible-freeipa/playbooks/config/set-ca-renewal-master-server.yml** Ansible playbook pour l'éditer :

```
---
- name: Playbook to handle global DNS configuration
  hosts: ipaserver
  become: no
  gather_facts: no
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: set ca_renewal_master_server
    ipaconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
      ca_renewal_master_server: carenewal.idm.example.com
```

4. Adapter le fichier en le modifiant :

   - Le mot de passe de l'administrateur IdM défini par la variable **ipaadmin_password**.
   - Le nom du serveur de renouvellement de l'autorité de certification défini par la variable **ca_renewal_master_server**.

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/config/set-ca-renewal-master-server.yml
```

Verification steps

Vous pouvez vérifier que le serveur de renouvellement de l'autorité de certification a été modifié :

1. Connectez-vous à **ipaserver** en tant qu'administrateur IdM :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Demander l'identité du serveur de renouvellement de l'autorité de certification IdM :

```
$ ipa config-show | grep 'CA renewal'
IPA CA renewal master:  carenewal.idm.example.com
```

   La sortie montre que le serveur **carenewal.idm.example.com** est le nouveau serveur de renouvellement de l'autorité de certification.

6.3. CONFIGURER LE SHELL PAR DÉFAUT POUR LES UTILISATEURS IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

L'interpréteur de commandes est un programme qui accepte et interprète les commandes. Plusieurs interpréteurs de commandes sont disponibles dans Red Hat Enterprise Linux (RHEL), tels que **bash**, **sh**, **ksh**, **zsh**, **fish**, et d'autres. **Bash** l'interpréteur de commandes, ou **/bin/bash**, est un interpréteur de commandes populaire sur la plupart des systèmes Linux, et c'est normalement l'interpréteur de commandes par défaut pour les comptes d'utilisateurs sur RHEL.

La procédure suivante décrit comment utiliser un playbook Ansible pour configurer **sh**, un shell alternatif, comme shell par défaut pour les utilisateurs d'IdM.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Facultatif : Utilisez le playbook Ansible **retrieve-config.yml** pour identifier le shell actuel des utilisateurs IdM. Voir Récupération de la configuration IdM à l'aide d'un playbook Ansible pour plus de détails.
2. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

3. Ouvrez le fichier **/usr/share/doc/ansible-freeipa/playbooks/config/ensure-config-options-are-set.yml** Ansible playbook pour l'éditer :

```
---
- name: Playbook to ensure some config options are set
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  # Set defaultlogin and maxusername
  - ipaconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
      defaultshell: /bin/bash
      maxusername: 64
```

4. Adaptez le fichier en modifiant les éléments suivants :
   - Le mot de passe de l'administrateur IdM défini par la variable **ipaadmin_password**.
   - Le shell par défaut des utilisateurs de l'IdM est défini par la variable **defaultshell** dans **/bin/sh**.
5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/config/ensure-config-options-are-set.yml
```

Verification steps

Vous pouvez vérifier que le shell de l'utilisateur par défaut a été modifié en démarrant une nouvelle session dans IdM :

1. Connectez-vous à **ipaserver** en tant qu'administrateur IdM :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Affiche l'interpréteur de commandes actuel :

```
[admin@server /]$ echo "$SHELL"
/bin/sh
```

   L'utilisateur connecté utilise l'interpréteur de commandes **sh**.

6.4. CONFIGURATION D'UN NOM NETBIOS POUR UN DOMAINE IDM À L'AIDE D'ANSIBLE

Le nom NetBIOS est utilisé pour le type de partage et de messagerie de Microsoft Windows (SMB). Vous pouvez utiliser les noms NetBIOS pour mapper un lecteur ou vous connecter à une imprimante.

Cette section décrit comment utiliser un playbook Ansible pour configurer un nom NetBIOS pour votre domaine de gestion des identités (IdM).

Conditions préalables

- 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.
  - Le paquet **ansible-freeipa** est installé.

Hypothèses

- 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 Ansible **secret.yml** stocke votre **ipaadmin_password** et que vous connaissez le mot de passe du fichier du coffre-fort.

Procédure

1. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Créer un fichier *netbios-domain-name-present.yml* Ansible playbook.
3. Ajoutez le contenu suivant au fichier :

```
---
- name: Playbook to change IdM domain netbios name
  hosts: ipaserver
  become: no
  gather_facts: no


  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
    - name: Set IdM domain netbios name
      ipaconfig:
        ipaadmin_password: "{{ ipaadmin_password }}"
        netbios_name: IPADOM
```

4. Enregistrer le fichier.
5. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory netbios-domain-name-present.yml
```

Lorsque vous y êtes invité, indiquez le mot de passe du fichier de l'espace de stockage.

Ressources supplémentaires

- Lignes directrices pour la configuration des noms NetBIOS

6.5. S'ASSURER QUE LES UTILISATEURS ET LES GROUPES IDM ONT DES SID EN UTILISANT ANSIBLE

Le serveur de gestion des identités (IdM) peut attribuer des identifiants de sécurité uniques (SID) aux utilisateurs et aux groupes IdM en interne, sur la base des données des plages d'ID du domaine local. Les SID sont stockés dans les objets utilisateurs et groupes.

L'objectif de s'assurer que les utilisateurs et les groupes de l'IdM ont des SID est de permettre la

génération du certificat d'attributs privilégiés (PAC), qui est la première étape vers la confiance entre l'IdM et l'IdM. Si les utilisateurs et les groupes de l'IdM ont des SID, l'IdM est en mesure d'émettre des tickets Kerberos avec des données PAC.

Cette section décrit comment atteindre les objectifs suivants :

- Générer des SID pour les utilisateurs et les groupes d'utilisateurs IdM déjà existants.
- Activer la génération de SID pour les nouveaux utilisateurs et groupes IdM.

Conditions préalables

- 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.
  - Le paquet **ansible-freeipa** est installé.

Hypothèses

- 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 Ansible **secret.yml** stocke votre **ipaadmin_password** et que vous connaissez le mot de passe du fichier du coffre-fort.

Procédure

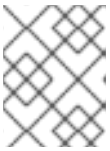
1. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Créer un fichier *sids-for-users-and-groups-present.yml* Ansible playbook.
3. Ajoutez le contenu suivant au fichier :

```
---
- name: Playbook to ensure SIDs are enabled and users and groups have SIDs
  hosts: ipaserver
  become: no
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
    - name: Enable SID and generate users and groups SIDS
      ipaconfig:
        ipaadmin_password: "{{ ipaadmin_password }}"
        enable_sid: true
        add_sids: true
```

La variable **enable_sid** permet de générer des SID pour les futurs utilisateurs et groupes IdM.
La variable **add_sids** génère des SID pour les utilisateurs et groupes IdM existants.



NOTE

Lorsque vous utilisez **add_sids: true**, vous devez également attribuer la valeur **true** à la variable **enable_sid**.

4. Enregistrer le fichier.

5. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory sids-for-users-and-groups-present.yml
```

Lorsque vous y êtes invité, indiquez le mot de passe du fichier de l'espace de stockage.

Ressources supplémentaires

- Le rôle de la sécurité et des identifiants relatifs dans les plages d'IDM.

6.6. RESSOURCES SUPPLÉMENTAIRES

- Voir **README-config.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/config**.

CHAPITRE 7. GÉRER LES COMPTES D'UTILISATEURS À L'AIDE DE PLAYBOOKS ANSIBLE

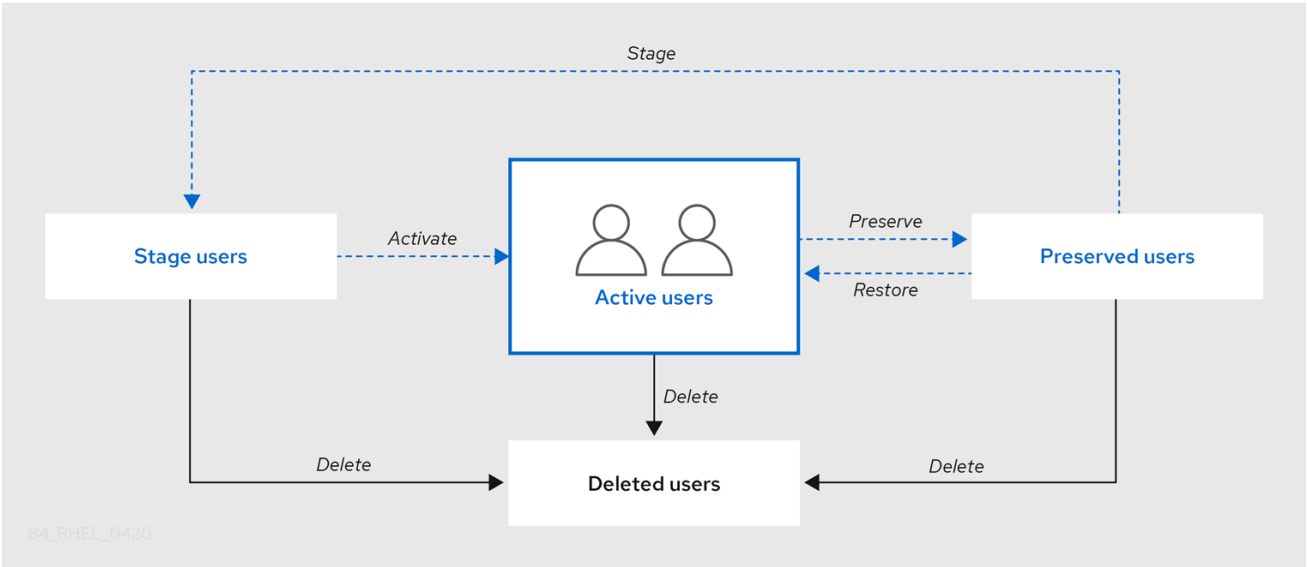
Vous pouvez gérer les utilisateurs dans IdM à l'aide de carnets de commande Ansible. Après avoir présenté le cycle de vie des utilisateurs, ce chapitre décrit comment utiliser les playbooks Ansible pour les opérations suivantes :

- Assurer la présence d'un seul utilisateur répertorié directement dans le fichier **YML**.
- Assurer la présence de plusieurs utilisateurs listés directement dans le fichier **YML**.
- Assurer la présence de plusieurs utilisateurs répertoriés dans un fichier **JSON** référencé à partir du fichier **YML**.
- Garantir l'absence d'utilisateurs listés directement dans le fichier **YML**.

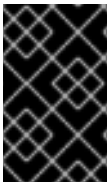
7.1. CYCLE DE VIE DE L'UTILISATEUR

La gestion des identités (IdM) prend en charge trois états de compte utilisateur :

- **Stage** les utilisateurs ne sont pas autorisés à s'authentifier. Il s'agit d'un état initial. Certaines propriétés du compte utilisateur requises pour les utilisateurs actifs ne peuvent pas être définies, par exemple l'appartenance à un groupe.
- **Active** les utilisateurs sont autorisés à s'authentifier. Toutes les propriétés requises du compte utilisateur doivent être définies dans cet état.
- **Preserved** sont d'anciens utilisateurs actifs qui sont considérés comme inactifs et ne peuvent pas s'authentifier auprès de l'IdM. Les utilisateurs préservés conservent la plupart des propriétés du compte qu'ils avaient en tant qu'utilisateurs actifs, mais ils ne font partie d'aucun groupe d'utilisateurs.



Vous pouvez supprimer définitivement les entrées utilisateur de la base de données IdM.



IMPORTANT

Les comptes d'utilisateurs supprimés ne peuvent pas être restaurés. Lorsque vous supprimez un compte d'utilisateur, toutes les informations associées à ce compte sont définitivement perdues.

Un nouvel administrateur ne peut être créé que par un utilisateur disposant de droits d'administrateur, tel que l'utilisateur admin par défaut. Si vous supprimez accidentellement tous les comptes d'administrateur, le gestionnaire de répertoire doit créer manuellement un nouvel administrateur dans le serveur de répertoire.

AVERTISSEMENT

Ne pas supprimer l'utilisateur **admin**. Comme **admin** est un utilisateur prédéfini requis par l'IdM, cette opération pose des problèmes avec certaines commandes. Si vous souhaitez définir et utiliser un autre utilisateur administrateur, désactivez l'utilisateur prédéfini **admin** avec **ipa user-disable admin** après avoir accordé des droits d'administrateur à au moins un autre utilisateur.

AVERTISSEMENT

N'ajoutez pas d'utilisateurs locaux à IdM. Le commutateur de service de noms (NSS) résout toujours les utilisateurs et les groupes IdM avant de résoudre les utilisateurs et les groupes locaux. Cela signifie, par exemple, que l'appartenance à un groupe IdM ne fonctionne pas pour les utilisateurs locaux.

7.2. ASSURER LA PRÉSENCE D'UN UTILISATEUR IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

La procédure suivante décrit comment assurer la présence d'un utilisateur dans IdM à l'aide d'un playbook Ansible.

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. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

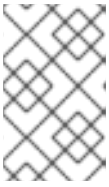
2. Créez un fichier Ansible playbook avec les données de l'utilisateur dont vous voulez assurer la présence dans IdM. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/user/add-user.yml**. Par exemple, pour créer un utilisateur nommé *idm_user* et ajouter *Password123* comme mot de passe :

```
---
- name: Playbook to handle users
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Create user idm_user
    ipauser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: idm_user
      first: Alice
      last: Acme
      uid: 1000111
      gid: 10011
      phone: "+555123457"
      email: idm_user@acme.com
      passwordexpiration: "2023-01-19 23:59:59"
      password: "Password123"
      update_password: on_create
```

Vous devez utiliser les options suivantes pour ajouter un utilisateur :

- **name**le nom d'utilisateur
- **first**: la chaîne de caractères du prénom
- **last**: la chaîne du nom de famille

Pour la liste complète des options disponibles pour l'utilisateur, voir le fichier Markdown de **/usr/share/doc/ansible-freeipa/README-user.md**.



NOTE

Si vous utilisez l'option **update_password: on_create**, Ansible ne crée le mot de passe de l'utilisateur que lorsqu'il crée l'utilisateur. Si l'utilisateur est déjà créé avec un mot de passe, Ansible ne génère pas de nouveau mot de passe.

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/add-IdM-
user.yml
```

Verification steps

- Vous pouvez vérifier si le nouveau compte d'utilisateur existe dans IdM en utilisant la commande **ipa user-show**:

  1. Connectez-vous à **ipaserver** en tant qu'administrateur :

     ```
     $ ssh admin@server.idm.example.com
     Password:
     [admin@server /]$
     ```

  2. Demander un ticket Kerberos pour l'administrateur :

     ```
     $ kinit admin
     Password for admin@IDM.EXAMPLE.COM:
     ```

  3. Demande d'informations sur *idm_user*:

     ```
     $ ipa user-show idm_user
       User login: idm_user
       First name: Alice
       Last name: Acme
       ....
     ```

  L'utilisateur nommé *idm_user* est présent dans IdM.

7.3. ASSURER LA PRÉSENCE DE PLUSIEURS UTILISATEURS IDM À L'AIDE DE PLAYBOOKS ANSIBLE

La procédure suivante décrit comment assurer la présence de plusieurs utilisateurs dans IdM à l'aide d'un playbook Ansible.

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. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créez un fichier Ansible playbook avec les données des utilisateurs dont vous voulez assurer la présence dans IdM. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-users-present.yml**. Par exemple, pour créer les utilisateurs *idm_user_1*, *idm_user_2*, et *idm_user_3*, et ajouter *Password123* comme mot de passe de *idm_user_1*:

```
---
- name: Playbook to handle users
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Create user idm_users
    ipauser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      users:
      - name: idm_user_1
        first: Alice
        last: Acme
        uid: 10001
        gid: 10011
        phone: "+555123457"
        email: idm_user@acme.com
        passwordexpiration: "2023-01-19 23:59:59"
        password: "Password123"
      - name: idm_user_2
        first: Bob
        last: Acme
        uid: 100011
        gid: 10011
      - name: idm_user_3
        first: Eve
        last: Acme
        uid: 1000111
        gid: 10011
```

NOTE

Si vous ne spécifiez pas l'option **update_password: on_create**, Ansible réinitialise le mot de passe de l'utilisateur à chaque fois que le livre de jeu est exécuté : si l'utilisateur a modifié le mot de passe depuis la dernière fois que le livre de jeu a été exécuté, Ansible réinitialise le mot de passe.

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/add-
users.yml
```

Verification steps

- Vous pouvez vérifier si le compte d'utilisateur existe dans IdM en utilisant la commande **ipa user-show**:

  1. Connectez-vous à **ipaserver** en tant qu'administrateur :

     ```
     $ ssh administrator@server.idm.example.com
     Password:
     [admin@server /]$
     ```

  2. Afficher des informations sur *idm_user_1*:

     ```
     $ ipa user-show idm_user_1
       User login: idm_user_1
       First name: Alice
       Last name: Acme
       Password: True
       ....
     ```

  L'utilisateur nommé *idm_user_1* est présent dans IdM.

7.4. ASSURER LA PRÉSENCE DE PLUSIEURS UTILISATEURS IDM À PARTIR D'UN FICHIER JSON EN UTILISANT LES PLAYBOOKS ANSIBLE

La procédure suivante décrit comment assurer la présence de plusieurs utilisateurs dans IdM à l'aide d'un playbook Ansible. Les utilisateurs sont stockés dans un fichier **JSON**.

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. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créez un fichier playbook Ansible avec les tâches nécessaires. Référencez le fichier **JSON** avec les données des utilisateurs dont vous voulez assurer la présence. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/ensure-users-present-ymlfile.yml**:

```
---
- name: Ensure users' presence
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Include users.json
    include_vars:
      file: users.json

  - name: Users present
    ipauser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      users: "{{ users }}"
```

3. Créez le fichier **users.json** et ajoutez-y les utilisateurs IdM. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple du fichier **/usr/share/doc/ansible-freeipa/playbooks/user/users.json**. Par exemple, pour créer les utilisateurs *idm_user_1*, *idm_user_2*, et *idm_user_3*, et ajouter *Password123* comme mot de passe de *idm_user_1*:

```
{
  "users": [
    {
      "name": "idm_user_1",
      "first": "Alice",
      "last": "Acme",
      "password": "Password123"
    },
    {
      "name": "idm_user_2",
      "first": "Bob",
      "last": "Acme"
    },
    {
      "name": "idm_user_3",
      "first": "Eve",
      "last": "Acme"
    }
  ]
}
```

4. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-users-
present-jsonfile.yml
```

Verification steps

- Vous pouvez vérifier si les comptes d'utilisateurs sont présents dans IdM à l'aide de la commande **ipa user-show**:

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh administrator@server.idm.example.com
Password:
[admin@server /]$
```

2. Afficher des informations sur *idm_user_1*:

```
$ ipa user-show idm_user_1
  User login: idm_user_1
  First name: Alice
  Last name: Acme
  Password: True
  ....
```

L'utilisateur nommé *idm_user_1* est présent dans IdM.

7.5. ASSURER L'ABSENCE D'UTILISATEURS UTILISANT DES PLAYBOOKS ANSIBLE

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer que des utilisateurs spécifiques sont absents de l'IdM.

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. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créez un fichier playbook Ansible avec les utilisateurs dont vous voulez garantir l'absence d'IdM. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-users-present.yml**. Par exemple, pour supprimer les utilisateurs *idm_user_1*, *idm_user_2*, et *idm_user_3*:

```
---
- name: Playbook to handle users
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Delete users idm_user_1, idm_user_2, idm_user_3
    ipauser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      users:
      - name: idm_user_1
      - name: idm_user_2
      - name: idm_user_3
      state: absent
```

3. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/delete-
users.yml
```

Verification steps

Vous pouvez vérifier que les comptes d'utilisateurs n'existent pas dans IdM en utilisant la commande **ipa user-show**:

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh administrator@server.idm.example.com
Password:
[admin@server /]$
```

2. Demande d'informations sur *idm_user_1*:

```
$ ipa user-show idm_user_1
ipa: ERROR: idm_user_1: user not found
```

L'utilisateur nommé *idm_user_1* n'existe pas dans IdM.

7.6. RESSOURCES SUPPLÉMENTAIRES

- Voir le fichier Markdown de **README-user.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

- Voir les exemples de playbooks Ansible dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/user**.

CHAPITRE 8. GÉRER LES GROUPES D'UTILISATEURS À L'AIDE DE PLAYBOOKS ANSIBLE

Cette section présente la gestion des groupes d'utilisateurs à l'aide des playbooks Ansible.

Un groupe d'utilisateurs est un ensemble d'utilisateurs ayant des privilèges, des politiques de mot de passe et d'autres caractéristiques communes.

Dans le cadre de la gestion des identités (IdM), un groupe d'utilisateurs peut inclure :

- Utilisateurs de l'IdM
- d'autres groupes d'utilisateurs de l'IdM
- les utilisateurs externes, c'est-à-dire les utilisateurs qui existent en dehors de l'IdM

La section comprend les sujets suivants :

- Les différents types de groupes dans l'IdM
- Membres directs et indirects du groupe
- Assurer la présence de groupes IdM et de membres de groupes à l'aide de playbooks Ansible
- Utiliser Ansible pour permettre aux utilisateurs AD d'administrer IdM
- Assurer la présence de gestionnaires de membres dans les groupes d'utilisateurs IDM à l'aide de playbooks Ansible
- Garantir l'absence de gestionnaires de membres dans les groupes d'utilisateurs IDM à l'aide de playbooks Ansible

8.1. LES DIFFÉRENTS TYPES DE GROUPES DANS L'IDM

IdM prend en charge les types de groupes suivants :

Groupes POSIX (par défaut)

Les groupes POSIX prennent en charge les attributs Linux POSIX pour leurs membres. Notez que les groupes qui interagissent avec Active Directory ne peuvent pas utiliser les attributs POSIX.
Les attributs POSIX identifient les utilisateurs en tant qu'entités distinctes. Parmi les exemples d'attributs POSIX concernant les utilisateurs, on peut citer **uidNumber**, un numéro d'utilisateur (UID), et **gidNumber**, un numéro de groupe (GID).

Groupes non-POSIX

Les groupes non-POSIX ne prennent pas en charge les attributs POSIX. Par exemple, ces groupes n'ont pas de GID défini.
Tous les membres de ce type de groupe doivent appartenir au domaine IdM.

Groupes externes

Utilisez les groupes externes pour ajouter des membres de groupes qui existent dans un magasin d'identité en dehors du domaine IdM, par exemple :

- Un système local

- Un domaine Active Directory
- Un service d'annuaire

Les groupes externes ne prennent pas en charge les attributs POSIX. Par exemple, ces groupes n'ont pas de GID défini.

Tableau 8.1. Groupes d'utilisateurs créés par défaut

Nom du groupe	Membres du groupe par défaut
ipausers	Tous les utilisateurs de l'IdM
admins	Utilisateurs disposant de privilèges administratifs, y compris l'utilisateur par défaut **admin**
editors	Il s'agit d'un groupe ancien qui ne bénéficie plus de privilèges particuliers
trust admins	Utilisateurs ayant des privilèges pour gérer les trusts Active Directory

Lorsque vous ajoutez un utilisateur à un groupe d'utilisateurs, celui-ci bénéficie des privilèges et des règles associés au groupe. Par exemple, pour accorder des privilèges administratifs à un utilisateur, ajoutez-le au groupe **admins**.

AVERTISSEMENT

Ne pas supprimer le groupe **admins**. Comme **admins** est un groupe prédéfini requis par IdM, cette opération pose des problèmes avec certaines commandes.

En outre, IdM crée *user private groups* par défaut chaque fois qu'un nouvel utilisateur est créé dans IdM. Pour plus d'informations sur les groupes privés, voir Ajouter des utilisateurs sans groupe privé.

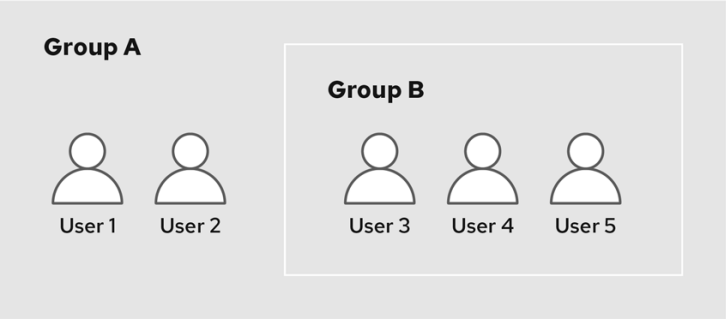
8.2. MEMBRES DIRECTS ET INDIRECTS DU GROUPE

Les attributs des groupes d'utilisateurs dans l'IdM s'appliquent à la fois aux membres directs et indirects : lorsque le groupe B est membre du groupe A, tous les utilisateurs du groupe B sont considérés comme des membres indirects du groupe A.

Par exemple, dans le diagramme suivant :

- L'utilisateur 1 et l'utilisateur 2 sont *direct members* du groupe A.
- L'utilisateur 3, l'utilisateur 4 et l'utilisateur 5 sont *indirect members* du groupe A.

Figure 8.1. Appartenance directe et indirecte à un groupe



Si vous définissez une politique de mot de passe pour le groupe d'utilisateurs A, cette politique s'applique également à tous les utilisateurs du groupe d'utilisateurs B.

8.3. ASSURER LA PRÉSENCE DE GROUPES IDM ET DE MEMBRES DE GROUPES À L'AIDE DE PLAYBOOKS ANSIBLE

La procédure suivante décrit comment assurer la présence de groupes IdM et de membres de groupes - à la fois des utilisateurs et des groupes d'utilisateurs - à l'aide d'un playbook Ansible.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Les utilisateurs que vous souhaitez référencer dans votre manuel de jeu Ansible existent dans IdM. Pour plus de détails sur la façon de garantir la présence des utilisateurs à l'aide d'Ansible, voir Gérer les comptes d'utilisateurs à l'aide de carnets de commande Ansible .

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créez un fichier playbook Ansible avec les informations nécessaires sur l'utilisateur et le groupe :

```
---
- name: Playbook to handle groups
  hosts: ipaserver
```

```
vars_files:
- /home/user_name/MyPlaybooks/secret.yml
tasks:
- name: Create group ops with gid 1234
  ipagroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: ops
    gidnumber: 1234

- name: Create group sysops
  ipagroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: sysops
    user:
    - idm_user

- name: Create group appops
  ipagroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: appops

- name: Add group members sysops and appops to group ops
  ipagroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: ops
    group:
    - sysops
    - appops
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/add-group-
members.yml
```

Verification steps

Vous pouvez vérifier si le groupe **ops** contient **sysops** et **appops** en tant que membres directs et **idm_user** en tant que membre indirect en utilisant la commande **ipa group-show**:

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Afficher des informations sur *ops*:

```
ipaserver]$ ipa group-show ops
  Group name: ops
  GID: 1234
  Member groups: sysops, appops
  Indirect Member users: idm_user
```

Les groupes **appops** et **sysops** - ce dernier comprenant l'utilisateur **idm_user** - existent dans IdM.

Ressources supplémentaires

- Voir le fichier Markdown de **/usr/share/doc/ansible-freeipa/README-group.md**.

8.4. UTILISER ANSIBLE POUR PERMETTRE AUX UTILISATEURS AD D'ADMINISTRER IDM

Cette section décrit comment utiliser un playbook Ansible pour s'assurer qu'un remplacement d'ID d'utilisateur est présent dans un groupe de gestion des identités (IdM). Le remplacement de l'ID utilisateur est le remplacement d'un utilisateur Active Directory (AD) que vous avez créé dans la vue de confiance par défaut après avoir établi une confiance avec AD. Suite à l'exécution du playbook, un utilisateur AD, par exemple un administrateur AD, est en mesure d'administrer entièrement l'IdM sans avoir deux comptes et mots de passe différents.

Conditions préalables

- Vous connaissez le mot de passe de l'IdM **admin**.
- Vous avez installé un trust avec AD.
- Le remplacement de l'ID de l'utilisateur AD existe déjà dans l'IdM. Si ce n'est pas le cas, créez-le avec la commande **ipa idoverrideuser-add 'default trust view'** ***ad_user@ad.example.com*** commande.
- Le groupe auquel vous ajoutez le remplacement de l'ID utilisateur existe déjà dans IdM .
- Vous utilisez la version 4.8.7 d'IdM ou une version ultérieure. Pour connaître la version d'IdM installée sur votre serveur, entrez **ipa --version**.
- 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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Créez un playbook **add-useridoverride-to-group.yml** avec le contenu suivant :

```
---
- name: Playbook to ensure presence of users in a group
```

```
  hosts: ipaserver


  - name: Ensure the ad_user@ad.example.com user ID override is a member of the admins
group:
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: admins
      idoverrideuser:
      - ad_user@ad.example.com
```

Dans l'exemple :

- Secret123 est le mot de passe de l'IdM **admin**.
- **admins** est le nom du groupe POSIX IdM auquel vous ajoutez l'annulation de l'ID **ad_user@ad.example.com**. Les membres de ce groupe disposent de tous les privilèges d'administrateur.
- **ad_user@ad.example.com** est le remplacement de l'ID utilisateur d'un administrateur AD. L'utilisateur est stocké dans le domaine AD avec lequel une confiance a été établie.

3. Enregistrer le fichier.
4. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-
useridoverride-to-group.yml
```

Ressources supplémentaires

- Remplacement des ID pour les utilisateurs AD
- /usr/share/doc/ansible-freeipa/README-group.md
- /usr/share/doc/ansible-freeipa/playbooks/user
- Utilisation des vues d'identification dans les environnements Active Directory
- Permettre aux utilisateurs AD d'administrer l'IdM

8.5. ASSURER LA PRÉSENCE DE GESTIONNAIRES MEMBRES DANS LES GROUPES D'UTILISATEURS IDM À L'AIDE DE PLAYBOOKS ANSIBLE

La procédure suivante décrit comment assurer la présence des gestionnaires membres de l'IdM - à la fois les utilisateurs et les groupes d'utilisateurs - à l'aide d'un playbook Ansible.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.

- Vous devez disposer du nom de l'utilisateur ou du groupe que vous ajoutez en tant que membres gestionnaires et du nom du groupe que vous souhaitez qu'ils gèrent.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créer un fichier playbook Ansible avec les informations nécessaires à la gestion des utilisateurs et des membres du groupe :

```
---
- name: Playbook to handle membership management
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure user test is present for group_a
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: group_a
      membermanager_user: test

  - name: Ensure group_admins is present for group_a
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: group_a
      membermanager_group: group_admins
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/add-member-
managers-user-groups.yml
```

Verification steps

Vous pouvez vérifier si le groupe **group_a** contient **test** en tant que gestionnaire membre et si **group_admins** est un gestionnaire membre de **group_a** en utilisant la commande **ipa group-show**:

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Afficher des informations sur *managergroup1*:

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

Ressources supplémentaires

- Voir **ipa host-add-member-manager --help**.
- Voir la page de manuel **ipa**.

8.6. ASSURER L'ABSENCE DE MEMBRES GESTIONNAIRES DANS LES GROUPES D'UTILISATEURS IDM À L'AIDE DE PLAYBOOKS ANSIBLE

La procédure suivante décrit comment garantir l'absence de gestionnaires membres de l'IdM - à la fois des utilisateurs et des groupes d'utilisateurs - à l'aide d'un playbook Ansible.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Vous devez disposer du nom de l'utilisateur ou du groupe existant que vous supprimez et du nom du groupe qu'il gère.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créer un fichier playbook Ansible avec les informations nécessaires à la gestion des utilisateurs et des membres du groupe :

```
---
- name: Playbook to handle membership management
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure member manager user and group members are absent for group_a
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: group_a
      membermanager_user: test
      membermanager_group: group_admins
      action: member
      state: absent
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-
member-managers-are-absent.yml
```

Verification steps

Vous pouvez vérifier que le groupe **group_a** ne contient pas **test** en tant que gestionnaire membre et **group_admins** en tant que gestionnaire membre de **group_a** en utilisant la commande **ipa group-show**:

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Afficher des informations sur le groupe_a :

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

Ressources supplémentaires

- Voir **ipa host-remove-member-manager --help**.
- Voir la page de manuel **ipa**.

CHAPITRE 9. UTILISER ANSIBLE POUR AUTOMATISER L'APPARTENANCE À UN GROUPE DANS IDM

L'appartenance automatique à un groupe vous permet d'affecter aux utilisateurs et aux hôtes des groupes d'utilisateurs et des groupes d'hôtes automatiquement, en fonction de leurs attributs. Par exemple, vous pouvez

- Répartissez les entrées utilisateur des employés dans des groupes en fonction du responsable, de la localisation, du poste ou de tout autre attribut de l'employé. Vous pouvez dresser la liste de tous les attributs en saisissant **ipa user-add --help** sur la ligne de commande.
- Divisez les hôtes en groupes en fonction de leur classe, de leur emplacement ou de tout autre attribut. Vous pouvez dresser la liste de tous les attributs en entrant **ipa host-add --help** dans la ligne de commande.
- Ajouter tous les utilisateurs ou tous les hôtes à un seul groupe global.

Vous pouvez utiliser Red Hat Ansible Engine pour automatiser la gestion de l'appartenance automatique à un groupe dans Identity Management (IdM).

Cette section couvre les sujets suivants :

- Utiliser Ansible pour s'assurer qu'une règle automember pour un groupe d'utilisateurs IdM est présente
- Utiliser Ansible pour s'assurer qu'une condition est présente dans une règle de membre automatique d'un groupe d'utilisateurs IdM
- Utiliser Ansible pour s'assurer qu'une condition est absente dans une règle de membre automatique d'un groupe d'utilisateurs IdM
- Utiliser Ansible pour s'assurer qu'une règle automember pour un groupe IdM est absente
- Utiliser Ansible pour s'assurer qu'une condition est présente dans une règle de membre automatique d'un groupe d'hôtes IdM

9.1. UTILISER ANSIBLE POUR S'ASSURER QU'UNE RÈGLE AUTOMEMBER POUR UN GROUPE D'UTILISATEURS IDM EST PRÉSENTE

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer de l'existence d'une règle **automember** pour un groupe de gestion des identités (IdM). Dans l'exemple, la présence d'une règle **automember** est assurée pour le groupe d'utilisateurs **testing_group**.

Conditions préalables

- Vous connaissez le mot de passe de l'IdM **admin**.
- Le groupe d'utilisateurs **testing_group** existe dans IdM.
- 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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

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

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

3. Ouvrez le fichier **automember-group-present-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaautomember** task :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Fixer la variable **name** à **testing_group**.
   - Fixer la variable **automember_type** à **group**.
   - Assurez-vous que la variable **state** est définie sur **present**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Automember group present example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure group automember rule admins is present
    ipaautomember:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: testing_group
      automember_type: group
      state: present
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automember-group-present-copy.yml
```

9.2. UTILISER ANSIBLE POUR S'ASSURER QU'UNE CONDITION SPÉCIFIÉE EST PRÉSENTE DANS UNE RÈGLE DE MEMBRE AUTOMATIQUE D'UN GROUPE D'UTILISATEURS IDM

Ressources supplémentaires

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'une condition spécifiée existe dans une règle **automember** pour un groupe de gestion des identités (IdM). Dans l'exemple, la présence d'une condition liée à l'UID dans la règle **automember** est assurée pour le groupe **testing_group**. En spécifiant la condition **.***, vous vous assurez que tous les futurs utilisateurs IdM deviennent automatiquement membres du groupe **testing_group**.

Conditions préalables

- Vous connaissez le mot de passe de l'IdM **admin**.
- Le groupe d'utilisateurs **testing_group** et la règle du groupe d'utilisateurs automember existent dans IdM.
- 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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **automember-hostgroup-rule-present.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/automember/** et nommez-le, par exemple, **automember-usergroup-rule-present.yml**:

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

3. Ouvrez le fichier **automember-usergroup-rule-present.yml** pour le modifier.
4. Adapter le fichier en modifiant les paramètres suivants :
   - Renommez le playbook pour qu'il corresponde à votre cas d'utilisation, par exemple : **Automember user group rule member present**
   - Renommez la tâche pour qu'elle corresponde à votre cas d'utilisation, par exemple : **Ensure an automember condition for a user group is present**.

- Définissez les variables suivantes dans la section **ipaautomember** task :
  - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
  - Fixer la variable **name** à **testing_group**.
  - Fixer la variable **automember_type** à **group**.
  - Assurez-vous que la variable **state** est définie sur **present**.
  - Assurez-vous que la variable **action** est définie sur **member**.
  - Fixez la variable **inclusive key** à **UID**.
  - Fixer la variable **inclusive expression** à **.***

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Automember user group rule member present
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure an automember condition for a user group is present
    ipaautomember:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: testing_group
      automember_type: group
      state: present
      action: member
      inclusive:
        - key: UID
          expression: .*
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automember-usergroup-rule-present.yml
```

Verification steps

1. Se connecter en tant qu'administrateur IdM.

```
$ kinit admin
```

2. Ajouter un utilisateur, par exemple :

```
$ ipa user-add user101 --first user --last 101
----------------------
Added user "user101"
----------------------
```

```
User login: user101
First name: user
Last name: 101
...
Member of groups: ipausers, testing_group
...
```

9.3. UTILISER ANSIBLE POUR S'ASSURER QU'UNE CONDITION EST ABSENTE D'UNE RÈGLE DE MEMBRE AUTOMATIQUE D'UN GROUPE D'UTILISATEURS IDM

Ressources supplémentaires

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'une condition est absente d'une règle **automember** pour un groupe de gestion des identités (IdM). Dans l'exemple, l'absence d'une condition dans la règle **automember** est garantie et spécifie que les utilisateurs dont **initials** est **dp** doivent être inclus. La règle automember est appliquée au groupe **testing_group**. En appliquant la condition, vous vous assurez qu'aucun futur utilisateur IdM dont les initiales sont **dp** ne devienne membre du groupe **testing_group**.

Conditions préalables

- Vous connaissez le mot de passe de l'IdM **admin**.
- Le groupe d'utilisateurs **testing_group** et la règle du groupe d'utilisateurs automember existent dans IdM.
- 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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **automember-hostgroup-rule-absent.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/automember/** et nommez-le, par exemple, **automember-usergroup-rule-absent.yml**:

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

3. Ouvrez le fichier **automember-usergroup-rule-absent.yml** pour le modifier.
4. Adapter le fichier en modifiant les paramètres suivants :
   - Renommez le playbook pour qu'il corresponde à votre cas d'utilisation, par exemple : **Automember user group rule member absent**
   - Renommez la tâche pour qu'elle corresponde à votre cas d'utilisation, par exemple : **Ensure an automember condition for a user group is absent**.
   - Définissez les variables suivantes dans la section **ipaautomember** task :
     - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
     - Fixer la variable **name** à **testing_group**.
     - Fixer la variable **automember_type** à **group**.
     - Assurez-vous que la variable **state** est définie sur **absent**.
     - Assurez-vous que la variable **action** est définie sur **member**.
     - Fixez la variable **inclusive key** à **initials**.
     - Fixez la variable **inclusive expression** à **dp**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Automember user group rule member absent
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure an automember condition for a user group is absent
    ipaautomember:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: testing_group
      automember_type: group
      state: absent
      action: member
      inclusive:
        - key: initials
          expression: dp
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automember-usergroup-rule-absent.yml
```

Verification steps

1. Se connecter en tant qu'administrateur IdM.

```
$ kinit admin
```

2. Affichez le groupe automember :

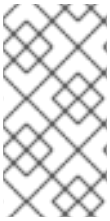
```
$ ipa automember-show --type=group testing_group
 Automember Rule: testing_group
```

L'absence d'une entrée **Inclusive Regex: initials=dp** dans la sortie confirme que la règle de membre automatique **testing_group** ne contient pas la condition spécifiée.

9.4. UTILISER ANSIBLE POUR S'ASSURER QU'UNE RÈGLE AUTOMEMBER POUR UN GROUPE D'UTILISATEURS IDM EST ABSENTE

Ressources supplémentaires

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'une règle **automember** est absente pour un groupe de gestion d'identité (IdM). Dans l'exemple, l'absence d'une règle **automember** est garantie pour le groupe **testing_group**.



NOTE

La suppression d'une règle de membre automatique supprime également toutes les conditions associées à la règle. Pour ne supprimer que des conditions spécifiques d'une règle, voir Utiliser Ansible pour s'assurer qu'une condition est absente d'une règle de membre automatique d'un groupe d'utilisateurs IdM.

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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **automember-group-absent.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/automember/**:

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

3. Ouvrez le fichier **automember-group-absent-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaautomember** task :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Fixer la variable **name** à **testing_group**.
   - Fixer la variable **automember_type** à **group**.
   - Assurez-vous que la variable **state** est définie sur **absent**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Automember group absent example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure group automember rule admins is absent
    ipaautomember:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: testing_group
      automember_type: group
      state: absent
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automember-group-absent.yml
```

Ressources supplémentaires

- Voir le fichier **README-automember.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir le répertoire **/usr/share/doc/ansible-freeipa/playbooks/automember**.

9.5. UTILISER ANSIBLE POUR S'ASSURER QU'UNE CONDITION EST PRÉSENTE DANS UNE RÈGLE DE MEMBRE AUTOMATIQUE D'UN GROUPE D'HÔTES IDM

Cette section décrit comment utiliser Ansible pour s'assurer qu'une condition est présente dans une règle de membre automatique de groupe d'hôtes IdM. L'exemple décrit comment s'assurer que les hôtes dont le **FQDN** est **.*.idm.example.com** sont membres du groupe d'hôtes **primary_dns_domain_hosts** et que les hôtes dont le **FQDN** est **.*.example.org** ne sont pas membres du groupe d'hôtes **primary_dns_domain_hosts**.

Conditions préalables

- Vous connaissez le mot de passe de l'IdM **admin**.
- Le groupe d'hôtes **primary_dns_domain_hosts** et la règle du groupe d'hôtes automember existent dans IdM.
- 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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

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

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

3. Ouvrez le fichier **automember-hostgroup-rule-present-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaautomember** task :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Fixer la variable **name** à **primary_dns_domain_hosts**.
   - Fixer la variable **automember_type** à **hostgroup**.
   - Assurez-vous que la variable **state** est définie sur **present**.
   - Assurez-vous que la variable **action** est définie sur **member**.
   - Assurez-vous que la variable **inclusive key** est fixée à **fqdn**.
   - Définissez la variable **inclusive expression** correspondante à **.*.idm.example.com**.
   - Fixez la variable **exclusive key** à **fqdn**.
   - Définissez la variable **exclusive expression** correspondante à **.*.example.org**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Automember user group rule member present
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure an automember condition for a user group is present
    ipaautomember:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: primary_dns_domain_hosts
      automember_type: hostgroup
      state: present
      action: member
      inclusive:
        - key: fqdn
          expression: .*.idm.example.com
      exclusive:
        - key: fqdn
          expression: .*.example.org
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory automember-hostgroup-rule-present-copy.yml
```

Ressources supplémentaires

- Voir le fichier **README-automember.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir le répertoire **/usr/share/doc/ansible-freeipa/playbooks/automember**.

CHAPITRE 10. UTILISER LES PLAYBOOKS ANSIBLE POUR GÉRER LES RÈGLES DE SELF-SERVICE DANS L'IDM

Cette section présente les règles en libre-service dans la gestion des identités (IdM) et décrit comment créer et modifier des règles d'accès en libre-service à l'aide des playbooks Ansible. Les règles de contrôle d'accès en libre-service permettent à une entité IdM d'effectuer des opérations spécifiques sur son entrée du serveur d'annuaire IdM.

Cette section couvre les sujets suivants :

- Contrôle d'accès en libre-service dans l'IdM
- Utiliser Ansible pour s'assurer qu'une règle de libre-service est présente
- Utiliser Ansible pour s'assurer qu'une règle de libre-service est absente
- Utiliser Ansible pour s'assurer qu'une règle de libre-service possède des attributs spécifiques
- Utiliser Ansible pour s'assurer qu'une règle de libre-service n'a pas d'attributs spécifiques

10.1. CONTRÔLE D'ACCÈS EN LIBRE-SERVICE DANS L'IDM

Les règles de contrôle d'accès en libre-service définissent les opérations qu'une entité de gestion des identités (IdM) peut effectuer sur son entrée du serveur d'annuaire IdM : par exemple, les utilisateurs IdM ont la possibilité de mettre à jour leurs propres mots de passe.

Cette méthode de contrôle permet à une entité IdM authentifiée de modifier des attributs spécifiques dans son entrée LDAP, mais n'autorise pas les opérations **add** ou **delete** sur l'ensemble de l'entrée.

AVERTISSEMENT

Soyez prudent lorsque vous utilisez des règles de contrôle d'accès en libre-service : une mauvaise configuration des règles de contrôle d'accès peut entraîner une élévation involontaire des privilèges d'une entité.

10.2. UTILISER ANSIBLE POUR S'ASSURER QU'UNE RÈGLE DE LIBRE-SERVICE EST PRÉSENTE

La procédure suivante décrit comment utiliser un playbook Ansible pour définir des règles de libre-service et assurer leur présence sur un serveur de gestion des identités (IdM). Dans cet exemple, la nouvelle règle **Users can manage their own name details**permet aux utilisateurs de modifier leurs propres attributs **givenname**, **displayname**, **title** et **initials**. Cela leur permet, par exemple, de modifier leur nom d'affichage ou leurs initiales s'ils le souhaitent.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.

- 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. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **selfservice-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/selfservice/**:

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

3. Ouvrez le fichier **selfservice-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaselfservice** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom de la nouvelle règle de libre-service.
   - Attribuez à la variable **permission** une liste de permissions à accorder, séparées par des virgules : **read** et **write**.
   - Définissez la variable **attribute** avec une liste d'attributs que les utilisateurs peuvent gérer eux-mêmes : **givenname**, **displayname**, **title**, et **initials**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Self-service present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure self-service rule "Users can manage their own name details" is present
    ipaselfservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "Users can manage their own name details"
      permission: read, write
      attribute:
      - givenname
```

```
    - displayname
    - title
    - initials
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory selfservice-present-copy.yml
```

Ressources supplémentaires

- Voir Contrôle d'accès en libre-service dans IdM .
- Voir le fichier **README-selfservice.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir le répertoire **/usr/share/doc/ansible-freeipa/playbooks/selfservice**.

10.3. UTILISER ANSIBLE POUR S'ASSURER QU'UNE RÈGLE DE LIBRE-SERVICE EST ABSENTE

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'une règle de libre-service spécifiée est absente de votre configuration IdM. L'exemple ci-dessous décrit comment s'assurer que la règle de libre-service **Users can manage their own name details** n'existe pas dans IdM. Cela garantit que les utilisateurs ne peuvent pas, par exemple, modifier leur propre nom d'affichage ou leurs initiales.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **selfservice-absent.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/selfservice/**:

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

3. Ouvrez le fichier **selfservice-absent-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaselfservice** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom de la règle de libre-service.
   - Fixer la variable **state** à **absent**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Self-service absent
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure self-service rule "Users can manage their own name details" is absent
    ipaselfservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "Users can manage their own name details"
      state: absent
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory selfservice-
absent-copy.yml
```

Ressources supplémentaires

- Voir Contrôle d'accès en libre-service dans IdM .
- Voir le fichier **README-selfservice.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/selfservice**.

10.4. UTILISER ANSIBLE POUR S'ASSURER QU'UNE RÈGLE DE LIBRE-SERVICE POSSÈDE DES ATTRIBUTS SPÉCIFIQUES

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'une règle de libre-service existante possède des paramètres spécifiques. Dans l'exemple, vous vous assurez que la règle de libre-service **Users can manage their own name details** possède également l'attribut de membre **surname**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- La règle de libre-service **Users can manage their own name details** existe dans IdM.

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **selfservice-member-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/selfservice/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/selfservice/selfservice-member-present.yml selfservice-member-present-copy.yml
```

3. Ouvrez le fichier **selfservice-member-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaselfservice** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom de la règle de libre-service à modifier.
   - Fixer la variable **attribute** à **surname**.
   - Fixer la variable **action** à **member**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Self-service member present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure selfservice "Users can manage their own name details" member attribute surname is present
    ipaselfservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "Users can manage their own name details"
```

```
    attribute:
    - surname
    action: member
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory selfservice-member-present-copy.yml
```

Ressources supplémentaires

- Voir Contrôle d'accès en libre-service dans IdM .
- Voir le fichier **README-selfservice.md** disponible dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/selfservice**.

10.5. UTILISER ANSIBLE POUR S'ASSURER QU'UNE RÈGLE DE LIBRE-SERVICE N'A PAS D'ATTRIBUTS SPÉCIFIQUES

La procédure suivante décrit comment utiliser une séquence Ansible pour s'assurer qu'une règle de libre-service n'a pas de paramètres spécifiques. Vous pouvez utiliser ce livre de lecture pour vous assurer qu'une règle de libre-service n'accorde pas d'accès indésirable. Dans l'exemple, vous vous assurez que la règle de libre-service **Users can manage their own name details** n'a pas les attributs de membre **givenname** et **surname**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- La règle de libre-service **Users can manage their own name details** existe dans IdM.

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **selfservice-member-absent.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/selfservice/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/selfservice/selfservice-member-absent.yml selfservice-member-absent-copy.yml
```

3. Ouvrez le fichier **selfservice-member-absent-copy.yml** Ansible playbook pour l'éditer.

4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaselfservice** task :

   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom de la règle de libre-service que vous souhaitez modifier.
   - Fixez la variable **attribute** à **givenname** et **surname**.
   - Fixer la variable **action** à **member**.
   - Fixer la variable **state** à **absent**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Self-service member absent
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure selfservice "Users can manage their own name details" member attributes givenname and surname are absent
    ipaselfservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "Users can manage their own name details"
      attribute:
      - givenname
      - surname
      action: member
      state: absent
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory selfservice-member-absent-copy.yml
```

Ressources supplémentaires

- Voir Contrôle d'accès en libre-service dans IdM .

- Voir le fichier **README-selfservice.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/selfservice**.

CHAPITRE 11. DÉLÉGUER DES PERMISSIONS À DES GROUPES D'UTILISATEURS POUR GÉRER LES UTILISATEURS À L'AIDE DE PLAYBOOKS ANSIBLE

La délégation est l'une des méthodes de contrôle d'accès dans IdM, avec les règles en libre-service et le contrôle d'accès basé sur les rôles (RBAC). Vous pouvez utiliser la délégation pour attribuer des autorisations à un groupe d'utilisateurs afin de gérer des entrées pour un autre groupe d'utilisateurs.

Cette section couvre les sujets suivants :

- Règles de délégation
- Création du fichier d'inventaire Ansible pour IdM
- Utiliser Ansible pour s'assurer qu'une règle de délégation est présente
- Utiliser Ansible pour s'assurer qu'une règle de délégation est absente
- Utiliser Ansible pour s'assurer qu'une règle de délégation possède des attributs spécifiques
- Utiliser Ansible pour s'assurer qu'une règle de délégation n'a pas d'attributs spécifiques

11.1. RÈGLES DE DÉLÉGATION

Vous pouvez déléguer des autorisations à des groupes d'utilisateurs pour gérer les utilisateurs en créant **delegation rules**.

Les règles de délégation permettent à un groupe d'utilisateurs spécifique d'effectuer des opérations d'écriture (modification) sur des attributs spécifiques pour les utilisateurs d'un autre groupe d'utilisateurs. Cette forme de règle de contrôle d'accès se limite à la modification des valeurs d'un sous-ensemble d'attributs que vous spécifiez dans une règle de délégation ; elle ne permet pas d'ajouter ou de supprimer des entrées entières ni de contrôler des attributs non spécifiés.

Les règles de délégation accordent des autorisations aux groupes d'utilisateurs existants dans IdM. Vous pouvez utiliser la délégation pour, par exemple, permettre au groupe d'utilisateurs **managers** de gérer certains attributs des utilisateurs du groupe d'utilisateurs **employees**.

11.2. CRÉATION D'UN FICHIER D'INVENTAIRE ANSIBLE POUR IDM

Lorsque vous travaillez avec Ansible, il est bon de créer, dans votre répertoire personnel, un sous-répertoire dédié aux playbooks Ansible que vous copiez et adaptez à partir des sous-répertoires **/usr/share/doc/ansible-freeipa/*** et **/usr/share/doc/rhel-system-roles/***. Cette pratique présente les avantages suivants :

- Vous pouvez retrouver tous vos playbooks en un seul endroit.
- Vous pouvez exécuter vos playbooks sans invoquer les privilèges de **root**.

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/<username>/MyPlaybooks/inventory

[privilege_escalation]
become=True
```

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

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

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**.

11.3. UTILISER ANSIBLE POUR S'ASSURER QU'UNE RÈGLE DE DÉLÉGATION EST PRÉSENTE

La procédure suivante décrit comment utiliser un playbook Ansible pour définir les privilèges d'une nouvelle règle de délégation IdM et assurer sa présence. Dans l'exemple, la nouvelle règle de délégation **basic manager attributes** accorde au groupe **managers** la possibilité de lire et d'écrire les attributs suivants pour les membres du groupe **employees**:

- **businesscategory**
- **departmentnumber**
- **employeenumber**
- **employeetype**

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **delegation-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/delegation/**:

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

3. Ouvrez le fichier **delegation-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipadelegation** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom de la nouvelle règle de délégation.
   - Attribuez à la variable **permission** une liste de permissions à accorder, séparées par des virgules : **read** et **write**.
   - Définissez la variable **attribute** avec une liste d'attributs que le groupe d'utilisateurs délégué peut gérer : **businesscategory**, **departmentnumber**, **employeenumber**, et **employeetype**.
   - Définissez la variable **group** avec le nom du groupe auquel on donne accès à la visualisation ou à la modification des attributs.
   - Définissez la variable **membergroup** avec le nom du groupe dont les attributs peuvent être visualisés ou modifiés.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to manage a delegation rule
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure delegation "basic manager attributes" is present
    ipadelegation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "basic manager attributes"
      permission: read, write
      attribute:
      - businesscategory
      - departmentnumber
      - employeenumber
```

```
    - employeetype
    group: managers
    membergroup: employees
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
delegation-present-copy.yml
```

Ressources supplémentaires

- Voir règles de délégation.
- Voir le fichier **README-delegation.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation**.

11.4. UTILISER ANSIBLE POUR S'ASSURER QU'UNE RÈGLE DE DÉLÉGATION EST ABSENTE

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'une règle de délégation spécifiée est absente de votre configuration IdM. L'exemple ci-dessous décrit comment s'assurer que la règle de délégation personnalisée **basic manager attributes** n'existe pas dans IdM.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire **~/MyPlaybooks/** répertoire :

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

2. Faites une copie du fichier **delegation-absent.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/delegation/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-present.yml
delegation-absent-copy.yml
```

3. Ouvrez le fichier **delegation-absent-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipadelegation** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom de la règle de délégation.
   - Fixer la variable **state** à **absent**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Delegation absent
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure delegation "basic manager attributes" is absent
    ipadelegation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "basic manager attributes"
      state: absent
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
delegation-absent-copy.yml
```

Ressources supplémentaires

- Voir règles de délégation.
- Voir le fichier **README-delegation.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation**.

11.5. UTILISER ANSIBLE POUR S'ASSURER QU'UNE RÈGLE DE DÉLÉGATION POSSÈDE DES ATTRIBUTS SPÉCIFIQUES

La procédure suivante décrit comment utiliser une séquence Ansible pour s'assurer qu'une règle de délégation dispose de paramètres spécifiques. Vous pouvez utiliser ce livre de jeu pour modifier un rôle de délégation que vous avez précédemment créé. Dans l'exemple, vous vous assurez que la règle de délégation **basic manager attributes** possède uniquement l'attribut membre **departmentnumber**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- La règle de délégation **basic manager attributes** existe dans IdM.

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **delegation-member-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/delegation/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-member-present.yml delegation-member-present-copy.yml
```

3. Ouvrez le fichier **delegation-member-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipadelegation** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom de la règle de délégation à modifier.
   - Fixer la variable **attribute** à **departmentnumber**.
   - Fixer la variable **action** à **member**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Delegation member present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure delegation "basic manager attributes" member attribute departmentnumber is present
    ipadelegation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "basic manager attributes"
```

```
    attribute:
    - departmentnumber
    action: member
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
delegation-member-present-copy.yml
```

Ressources supplémentaires

- Voir règles de délégation.
- Voir le fichier **README-delegation.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation**.

11.6. UTILISER ANSIBLE POUR S'ASSURER QU'UNE RÈGLE DE DÉLÉGATION N'A PAS D'ATTRIBUTS SPÉCIFIQUES

La procédure suivante décrit comment utiliser une séquence Ansible pour s'assurer qu'une règle de délégation n'a pas de paramètres spécifiques. Vous pouvez utiliser ce livre de lecture pour vous assurer qu'un rôle de délégation n'accorde pas d'accès indésirable. Dans l'exemple, vous vous assurez que la règle de délégation **basic manager attributes** n'a pas les attributs de membre **employeenumber** et **employeetype**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- La règle de délégation **basic manager attributes** existe dans IdM.

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **delegation-member-absent.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/delegation/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/delegation/delegation-member-absent.yml delegation-member-absent-copy.yml
```

3. Ouvrez le fichier **delegation-member-absent-copy.yml** Ansible playbook pour l'éditer.

4. Adaptez le fichier en définissant les variables suivantes dans la section **ipadelegation** task :

   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom de la règle de délégation à modifier.
   - Fixez la variable **attribute** à **employeenumber** et **employeetype**.
   - Fixer la variable **action** à **member**.
   - Fixer la variable **state** à **absent**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Delegation member absent
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure delegation "basic manager attributes" member attributes employeenumber and employeetype are absent
    ipadelegation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: "basic manager attributes"
      attribute:
      - employeenumber
      - employeetype
      action: member
      state: absent
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory delegation-member-absent-copy.yml
```

Ressources supplémentaires

- Voir règles de délégation.
- Voir le fichier **README-delegation.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation**.

CHAPITRE 12. UTILISER LES PLAYBOOKS ANSIBLE POUR GÉRER LE CONTRÔLE D'ACCÈS BASÉ SUR LES RÔLES DANS IDM

Le contrôle d'accès basé sur les rôles (RBAC) est un mécanisme de contrôle d'accès neutre défini autour des rôles et des privilèges. Les composants du contrôle d'accès basé sur les rôles dans la gestion de l'identité (IdM) sont les rôles, les privilèges et les permissions :

- **Permissions** accorder le droit d'effectuer une tâche spécifique telle que l'ajout ou la suppression d'utilisateurs, la modification d'un groupe, l'activation de l'accès en lecture, etc.
- **Privileges** combiner les autorisations, par exemple toutes les autorisations nécessaires pour ajouter un nouvel utilisateur.
- **Roles** accorder un ensemble de privilèges à des utilisateurs, des groupes d'utilisateurs, des hôtes ou des groupes d'hôtes.

Dans les grandes entreprises en particulier, l'utilisation de RBAC peut aider à créer un système hiérarchique d'administrateurs avec leurs domaines de responsabilité individuels.

Ce chapitre décrit les opérations suivantes effectuées lors de la gestion de RBAC à l'aide des playbooks Ansible :

- Permissions dans l'IdM
- Permissions gérées par défaut
- Privilèges dans l'IdM
- Rôles dans l'IdM
- Rôles prédéfinis dans l'IdM
- Utiliser Ansible pour s'assurer qu'un rôle IdM RBAC avec des privilèges est présent
- Utiliser Ansible pour s'assurer qu'un rôle IdM RBAC est absent
- Utiliser Ansible pour s'assurer qu'un groupe d'utilisateurs est assigné à un rôle IdM RBAC
- Utiliser Ansible pour s'assurer que des utilisateurs spécifiques ne sont pas affectés à un rôle IdM RBAC
- Utiliser Ansible pour s'assurer qu'un service est membre d'un rôle IdM RBAC
- Utiliser Ansible pour s'assurer qu'un hôte est membre d'un rôle IdM RBAC
- Utiliser Ansible pour s'assurer qu'un groupe d'hôtes est membre d'un rôle IdM RBAC

12.1. PERMISSIONS DANS L'IDM

Les autorisations sont l'unité de niveau le plus bas du contrôle d'accès basé sur les rôles. Elles définissent les opérations ainsi que les entrées LDAP auxquelles ces opérations s'appliquent. Comparables à des blocs de construction, les autorisations peuvent être attribuées à autant de privilèges que nécessaire.
Une ou plusieurs adresses **rights** définissent les opérations autorisées :

- **write**
- **read**
- **search**
- **compare**
- **add**
- **delete**
- **all**

Ces opérations s'appliquent à trois sites de base **targets**:

- **subtree**: un nom de domaine (DN) ; la sous-arborescence sous ce DN
- **target filter**un filtre LDAP
- **target**: DN avec des caractères génériques possibles pour spécifier les entrées

En outre, les options de commodité suivantes définissent le(s) attribut(s) correspondant(s) :

- **type**: un type d'objet (utilisateur, groupe, etc.) ; définit **subtree** et **target filter**
- **memberof**: membres d'un groupe ; fixe un **target filter**
- **targetgroup**: accorde l'accès à la modification d'un groupe spécifique (par exemple en accordant les droits de gérer l'appartenance à un groupe) ; définit une valeur de **target**

Grâce aux autorisations IdM, vous pouvez contrôler quels utilisateurs ont accès à quels objets et même à quels attributs de ces objets. L'IdM vous permet d'autoriser ou de bloquer des attributs individuels ou de modifier la visibilité totale d'une fonction IdM spécifique, telle que les utilisateurs, les groupes ou sudo, pour tous les utilisateurs anonymes, tous les utilisateurs authentifiés ou seulement un certain groupe d'utilisateurs privilégiés.
Par exemple, la flexibilité de cette approche des autorisations est utile pour un administrateur qui souhaite limiter l'accès des utilisateurs ou des groupes uniquement aux sections spécifiques auxquelles ces utilisateurs ou groupes ont besoin d'accéder et rendre les autres sections complètement cachées pour eux.

NOTE

Une autorisation ne peut pas contenir d'autres autorisations.

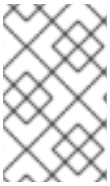
12.2. PERMISSIONS GÉRÉES PAR DÉFAUT

Les autorisations gérées sont des autorisations fournies par défaut avec l'IdM. Elles se comportent comme les autres autorisations créées par l'utilisateur, avec les différences suivantes :

- Vous ne pouvez pas les supprimer ni modifier leur nom, leur emplacement et leurs attributs cibles.
- Ils ont trois séries d'attributs :
  - **Default** l'utilisateur ne peut pas les modifier, car ils sont gérés par IdM

- **Included** les attributs, qui sont des attributs supplémentaires ajoutés par l'utilisateur
- **Excluded** les attributs, qui sont des attributs supprimés par l'utilisateur

Une autorisation gérée s'applique à tous les attributs qui figurent dans les ensembles d'attributs par défaut et inclus, mais pas dans l'ensemble exclu.



NOTE

Bien que vous ne puissiez pas supprimer une autorisation gérée, le fait de définir son type de liaison sur autorisation et de supprimer l'autorisation gérée de tous les privilèges la désactive effectivement.

Les noms de toutes les autorisations gérées commencent par **System:**, par exemple **System: Add Sudo rule** ou **System: Modify Services**. Les versions antérieures de l'IdM utilisaient un schéma différent pour les autorisations par défaut. Par exemple, l'utilisateur ne pouvait pas les supprimer et ne pouvait les attribuer qu'à des privilèges. La plupart de ces autorisations par défaut ont été transformées en autorisations gérées, mais les autorisations suivantes utilisent toujours le schéma précédent :

- Ajouter une tâche de reconstruction de l'adhésion automatique
- Ajouter des sous-enregistrements de configuration
- Ajouter des accords de réplication
- Certificat Supprimer la retenue
- Obtenir l'état des certificats de l'autorité de certification
- Plage de lecture de l'ADN
- Modifier la portée de l'ADN
- Lire la configuration des gestionnaires PassSync
- Modifier la configuration des gestionnaires PassSync
- Lire les accords de réplication
- Modifier les accords de réplication
- Supprimer les accords de réplication
- Lire la configuration de la base de données LDBM
- Demande de certificat
- Demande de certificat ignorant les listes de contrôle de l'autorité de certification
- Demander des certificats à un autre hôte
- Récupérer des certificats auprès de l'autorité de certification
- Révoquer le certificat
- Écriture de la configuration de l'IPA

NOTE

Si vous tentez de modifier une autorisation gérée à partir de la ligne de commande, le système ne vous permet pas de changer les attributs que vous ne pouvez pas modifier, la commande échoue. Si vous tentez de modifier une autorisation gérée à partir de l'interface Web, les attributs que vous ne pouvez pas modifier sont désactivés.

12.3. PRIVILÈGES DANS L'IDM

Un privilège est un groupe de permissions applicables à un rôle.
Alors qu'une autorisation donne le droit d'effectuer une seule opération, certaines tâches de l'IdM nécessitent plusieurs autorisations pour être menées à bien. Par conséquent, un privilège combine les différentes autorisations requises pour effectuer une tâche spécifique.
Par exemple, la création d'un compte pour un nouvel utilisateur IdM nécessite les autorisations suivantes :

- Création d'une nouvelle entrée utilisateur
- Réinitialisation du mot de passe d'un utilisateur
- Ajout du nouvel utilisateur au groupe d'utilisateurs IPA par défaut

La combinaison de ces trois tâches de bas niveau en une tâche de plus haut niveau sous la forme d'un privilège personnalisé nommé, par exemple, **Add User** facilite la gestion des rôles par l'administrateur du système. L'IdM contient déjà plusieurs privilèges par défaut. Outre les utilisateurs et les groupes d'utilisateurs, des privilèges sont également attribués aux hôtes et aux groupes d'hôtes, ainsi qu'aux services de réseau. Cette pratique permet un contrôle fin des opérations effectuées par un ensemble d'utilisateurs sur un ensemble d'hôtes utilisant des services de réseau spécifiques.

NOTE

Un privilège ne peut pas contenir d'autres privilèges.

12.4. RÔLES DANS L'IDM

Un rôle est une liste de privilèges que les utilisateurs spécifiés pour ce rôle possèdent.
En effet, les autorisations permettent d'effectuer certaines tâches de bas niveau (créer une entrée utilisateur, ajouter une entrée à un groupe, etc.), tandis que les privilèges combinent une ou plusieurs de ces autorisations nécessaires à une tâche de plus haut niveau (comme la création d'un nouvel utilisateur dans un groupe donné). Les rôles regroupent les privilèges selon les besoins : par exemple, un administrateur d'utilisateurs peut ajouter, modifier et supprimer des utilisateurs.

IMPORTANT

Les rôles sont utilisés pour classer les actions autorisées. Ils ne sont pas utilisés comme outil pour mettre en œuvre la séparation des privilèges ou pour se protéger contre l'escalade des privilèges.

NOTE

Les rôles ne peuvent pas contenir d'autres rôles.

12.5. RÔLES PRÉDÉFINIS DANS LA GESTION DE L'IDENTITÉ

Red Hat Identity Management fournit la gamme suivante de rôles prédéfinis :

Tableau 12.1. Rôles prédéfinis dans la gestion des identités

Rôle	Privilège	Description
Administrateur des inscriptions	Inscription au programme d'accueil	Responsable de l'inscription du client ou de l'hôte
helpdesk	Modifier les utilisateurs et réinitialiser les mots de passe, Modifier l'appartenance à un groupe	Effectuer des tâches simples d'administration des utilisateurs
Spécialiste de la sécurité informatique	Administrateurs Netgroups, Administrateur HBAC, Administrateur Sudo	Responsable de la gestion de la politique de sécurité telle que les contrôles d'accès basés sur l'hôte, les règles sudo
Spécialiste des technologies de l'information	Administrateurs d'hôtes, administrateurs de groupes d'hôtes, administrateurs de services, administrateurs de montages automatiques	Responsable de la gestion des hôtes
Architecte de sécurité	Administrateur de la délégation, Administrateurs de la réplication, Configuration de l'IPA en écriture, Administrateur de la politique des mots de passe	Responsable de la gestion de l'environnement de gestion des identités, de la création de trusts et d'accords de réplication
Administrateur des utilisateurs	Administrateurs d'utilisateurs, administrateurs de groupes, administrateurs d'utilisateurs de scène	Responsable de la création des utilisateurs et des groupes

12.6. UTILISER ANSIBLE POUR S'ASSURER QU'UN RÔLE IDM RBAC AVEC DES PRIVILÈGES EST PRÉSENT

Pour exercer un contrôle plus granulaire sur l'accès basé sur les rôles (RBAC) aux ressources dans la gestion des identités (IdM) que les rôles par défaut, créez un rôle personnalisé.

La procédure suivante décrit comment utiliser un playbook Ansible pour définir les privilèges d'un nouveau rôle personnalisé IdM et assurer sa présence. Dans l'exemple, le nouveau rôle **user_and_host_administrator** contient une combinaison unique des privilèges suivants qui sont présents dans IdM par défaut :

- **Group Administrators**
- **User Administrators**
- **Stage User Administrators**

- **Group Administrators**

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/<MyPlaybooks>/*** répertoire :

```
$ cd ~/<MyPlaybooks>/
```

2. Faites une copie du fichier **role-member-user-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/role/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-user-present.yml role-member-user-present-copy.yml
```

3. Ouvrez le fichier **role-member-user-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **iparole** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom du nouveau rôle.
   - Définissez la liste **privilege** avec les noms des privilèges IdM que vous souhaitez inclure dans le nouveau rôle.
   - Si vous le souhaitez, définissez la variable **user** avec le nom de l'utilisateur auquel vous souhaitez attribuer le nouveau rôle.
   - Si vous le souhaitez, attribuez à la variable **group** le nom du groupe auquel vous souhaitez attribuer le nouveau rôle.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no
```

```
vars_files:
- /home/user_name/MyPlaybooks/secret.yml
tasks:
- iparole:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: user_and_host_administrator
    user: idm_user01
    group: idm_group01
    privilege:
    - Group Administrators
    - User Administrators
    - Stage User Administrators
    - Group Administrators
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-user-present-copy.yml
```

Ressources supplémentaires

- Voir Chiffrer du contenu avec Ansible Vault.
- Voir Rôles dans IdM.
- Voir le fichier **README-role** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/iparole**.

12.7. UTILISER ANSIBLE POUR S'ASSURER QU'UN RÔLE IDM RBAC EST ABSENT

En tant qu'administrateur système gérant le contrôle d'accès basé sur les rôles (RBAC) dans la gestion des identités (IdM), vous pouvez vouloir garantir l'absence d'un rôle obsolète afin qu'aucun administrateur ne l'attribue accidentellement à un utilisateur.

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer de l'absence d'un rôle. L'exemple ci-dessous décrit comment s'assurer que le rôle personnalisé **user_and_host_administrator** n'existe pas dans IdM.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/<MyPlaybooks>/*** répertoire :

```
$ cd ~/<MyPlaybooks>/
```

2. Faites une copie du fichier **role-is-absent.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/role/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-is-absent.yml role-is-absent-copy.yml
```

3. Ouvrez le fichier **role-is-absent-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **iparole** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom du rôle.
   - Assurez-vous que la variable **state** est définie sur **absent**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: user_and_host_administrator
      state: absent
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/<MyPlaybooks>/inventory role-is-absent-copy.yml
```

Ressources supplémentaires

- Voir Chiffrer du contenu avec Ansible Vault.
- Voir Rôles dans IdM.
- Voir le fichier Markdown de **README-role** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/iparole**.

12.8. UTILISER ANSIBLE POUR S'ASSURER QU'UN GROUPE D'UTILISATEURS EST ASSIGNÉ À UN RÔLE IDM RBAC

En tant qu'administrateur système gérant le contrôle d'accès basé sur les rôles (RBAC) dans la gestion des identités (IdM), vous pouvez vouloir attribuer un rôle à un groupe spécifique d'utilisateurs, par exemple les administrateurs juniors.

L'exemple suivant décrit comment utiliser un playbook Ansible pour s'assurer que le rôle IdM RBAC **helpdesk** est attribué à **junior_sysadmins**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/<MyPlaybooks>/*** répertoire :

```
$ cd ~/<MyPlaybooks>/
```

2. Faites une copie du fichier **role-member-group-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/role/**:

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

3. Ouvrez le fichier **role-member-group-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **iparole** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom du rôle que vous souhaitez attribuer.

- Attribuez à la variable **group** le nom du groupe.
- Fixer la variable **action** à **member**.

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: helpdesk
      group: junior_sysadmins
      action: member
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-group-present-copy.yml
```

Ressources supplémentaires

- Voir Chiffrer du contenu avec Ansible Vault.
- Voir Rôles dans IdM.
- Voir le fichier Markdown de **README-role** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/iparole**.

12.9. UTILISER ANSIBLE POUR S'ASSURER QUE DES UTILISATEURS SPÉCIFIQUES NE SONT PAS AFFECTÉS À UN RÔLE IDM RBAC

En tant qu'administrateur système gérant le contrôle d'accès basé sur les rôles (RBAC) dans la gestion des identités (IdM), vous voudrez peut-être vous assurer qu'un rôle RBAC n'est pas attribué à des utilisateurs spécifiques après qu'ils ont, par exemple, changé de poste au sein de l'entreprise.

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer que les utilisateurs nommés **user_01** et **user_02** ne sont pas affectés au rôle **helpdesk**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.

- 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. Naviguez jusqu'au répertoire ***~/<MyPlaybooks>/*** répertoire :

```
$ cd ~/<MyPlaybooks>/
```

2. Faites une copie du fichier **role-member-user-absent.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/role/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-user-absent.yml role-member-user-absent-copy.yml
```

3. Ouvrez le fichier **role-member-user-absent-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **iparole** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom du rôle que vous souhaitez attribuer.
   - Définissez la liste **user** avec les noms des utilisateurs.
   - Fixer la variable **action** à **member**.
   - Fixer la variable **state** à **absent**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: helpdesk
      user
      - user_01
```

```
    - user_02
    action: member
    state: absent
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-user-absent-copy.yml
```

Ressources supplémentaires

- Voir Chiffrer du contenu avec Ansible Vault.
- Voir Rôles dans IdM.
- Voir le fichier Markdown de **README-role** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/iparole**.

12.10. UTILISER ANSIBLE POUR S'ASSURER QU'UN SERVICE EST MEMBRE D'UN RÔLE IDM RBAC

En tant qu'administrateur système gérant le contrôle d'accès basé sur les rôles (RBAC) dans la gestion des identités (IdM), vous pouvez vouloir vous assurer qu'un service spécifique inscrit dans l'IdM est membre d'un rôle particulier. L'exemple suivant décrit comment s'assurer que le rôle personnalisé **web_administrator** peut gérer le service **HTTP** qui s'exécute sur le serveur **client01.idm.example.com**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Le rôle **web_administrator** existe dans IdM.
- Le service **HTTP/client01.idm.example.com@IDM.EXAMPLE.COM** existe dans IdM.

Procédure

1. Naviguez jusqu'au répertoire ***~/<MyPlaybooks>/*** répertoire :

```
$ cd ~/<MyPlaybooks>/
```

2. Faites une copie du fichier **role-member-service-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/role/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-service-present-absent.yml role-member-service-present-copy.yml
```

3. Ouvrez le fichier **role-member-service-present-copy.yml** Ansible playbook pour l'éditer.

4. Adaptez le fichier en définissant les variables suivantes dans la section **iparole** task :

   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom du rôle que vous souhaitez attribuer.
   - Définissez la liste **service** avec le nom du service.
   - Fixer la variable **action** à **member**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web_administrator
      service:
      - HTTP/client01.idm.example.com
      action: member
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/<MyPlaybooks>/inventory role-member-service-present-copy.yml
```

Ressources supplémentaires

- Voir Chiffrer du contenu avec Ansible Vault.
- Voir Rôles dans IdM.
- Voir le fichier Markdown de **README-role** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/iparole**.

12.11. UTILISER ANSIBLE POUR S'ASSURER QU'UN HÔTE EST MEMBRE D'UN RÔLE IDM RBAC

En tant qu'administrateur système gérant le contrôle d'accès basé sur les rôles dans la gestion des identités (IdM), vous pouvez vouloir vous assurer qu'un hôte ou un groupe d'hôtes spécifique est associé à un rôle spécifique. L'exemple suivant décrit comment s'assurer que le rôle personnalisé **web_administrator** peut gérer l'hôte IdM **client01.idm.example.com** sur lequel le service **HTTP** est exécuté.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Le rôle **web_administrator** existe dans IdM.
- L'hôte **client01.idm.example.com** existe dans IdM.

Procédure

1. Naviguez jusqu'au répertoire ***~/<MyPlaybooks>/*** répertoire :

```
$ cd ~/<MyPlaybooks>/
```

2. Faites une copie du fichier **role-member-host-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/role/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/role/role-member-host-present.yml role-member-host-present-copy.yml
```

3. Ouvrez le fichier **role-member-host-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **iparole** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom du rôle que vous souhaitez attribuer.
   - Définissez la liste **host** avec le nom de l'hôte.

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web_administrator
      host:
      - client01.idm.example.com
      action: member
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-host-present-copy.yml
```

Ressources supplémentaires

- Voir Chiffrer du contenu avec Ansible Vault.
- Voir Rôles dans IdM.
- Voir le fichier Markdown de **README-role** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/iparole**.

12.12. UTILISER ANSIBLE POUR S'ASSURER QU'UN GROUPE D'HÔTES EST MEMBRE D'UN RÔLE IDM RBAC

En tant qu'administrateur système gérant le contrôle d'accès basé sur les rôles dans la gestion des identités (IdM), vous pouvez vouloir vous assurer qu'un hôte ou un groupe d'hôtes spécifique est associé à un rôle spécifique. L'exemple suivant décrit comment s'assurer que le rôle personnalisé **web_administrator** peut gérer le groupe d'hôtes IdM **web_servers** sur lequel le service **HTTP** est exécuté.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.

- Le rôle **web_administrator** existe dans IdM.
- Le groupe d'hôtes **web_servers** existe dans IdM.

Procédure

1. Naviguez jusqu'au répertoire ***~/<MyPlaybooks>/*** répertoire :

```
$ cd ~/<MyPlaybooks>/
```

2. Faites une copie du fichier **role-member-hostgroup-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/role/**:

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

3. Ouvrez le fichier **role-member-hostgroup-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **iparole** task :
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom du rôle que vous souhaitez attribuer.
   - Définissez la liste **hostgroup** avec le nom du groupe d'hôtes.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: yes
  gather_facts: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - iparole:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: web_administrator
      hostgroup:
      - web_servers
      action: member
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-hostgroup-present-copy.yml
```

Ressources supplémentaires

- Voir Chiffrer du contenu avec Ansible Vault.
- Voir Rôles dans IdM.
- Voir le fichier Markdown de **README-role** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/iparole**.

CHAPITRE 13. UTILISER LES PLAYBOOKS ANSIBLE POUR GÉRER LES PRIVILÈGES RBAC

Le contrôle d'accès basé sur les rôles (RBAC) est un mécanisme de contrôle d'accès neutre défini autour de rôles, de privilèges et de permissions. Dans les grandes entreprises en particulier, l'utilisation du RBAC peut aider à créer un système hiérarchique d'administrateurs avec leurs domaines de responsabilité individuels.

Ce chapitre décrit les opérations suivantes pour utiliser les playbooks Ansible afin de gérer les privilèges RBAC dans la gestion des identités (IdM) :

- Utiliser Ansible pour s'assurer qu'un privilège RBAC personnalisé est présent
- Utiliser Ansible pour s'assurer que les permissions des membres sont présentes dans un privilège IdM RBAC personnalisé
- Utiliser Ansible pour s'assurer qu'un privilège IdM RBAC n'inclut pas une permission
- Utiliser Ansible pour renommer un privilège IdM RBAC personnalisé
- Utiliser Ansible pour s'assurer qu'un privilège IdM RBAC est absent

Conditions préalables

- Vous comprenez les concepts et les principes de RBAC .

13.1. UTILISER ANSIBLE POUR S'ASSURER QU'UN PRIVILÈGE IDM RBAC PERSONNALISÉ EST PRÉSENT

Pour disposer d'un privilège personnalisé pleinement fonctionnel dans le contrôle d'accès basé sur les rôles (RBAC) de la gestion des identités (IdM), vous devez procéder par étapes :

1. Créer un privilège sans aucune autorisation.
2. Ajoutez les autorisations de votre choix au privilège.

La procédure suivante décrit comment créer un privilège vide à l'aide d'une séquence Ansible afin de pouvoir y ajouter ultérieurement des autorisations. L'exemple décrit comment créer un privilège nommé **full_host_administration** destiné à combiner toutes les autorisations IdM liées à l'administration des hôtes.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **privilege-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/privilege/**:

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

3. Ouvrez le fichier **privilege-present-copy.yml** Ansible playbook pour l'éditer.

4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaprivilege** task :

   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom du nouveau privilège, **full_host_administration**.
   - En option, décrivez le privilège à l'aide de la variable **description**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Privilege present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure privilege full_host_administration is present
    ipaprivilege:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: full_host_administration
      description: This privilege combines all IdM permissions related to host administration
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory privilege-present-copy.yml
```

13.2. UTILISER ANSIBLE POUR S'ASSURER QUE LES PERMISSIONS DES MEMBRES SONT PRÉSENTES DANS UN PRIVILÈGE IDM RBAC PERSONNALISÉ

Pour disposer d'un privilège personnalisé pleinement fonctionnel dans le contrôle d'accès basé sur les rôles (RBAC) de la gestion des identités (IdM), vous devez procéder par étapes :

1. Créer un privilège sans aucune autorisation.
2. Ajoutez les autorisations de votre choix au privilège.

La procédure suivante explique comment utiliser un cahier de jeu Ansible pour ajouter des autorisations à un privilège créé à l'étape précédente. L'exemple décrit comment ajouter toutes les autorisations IdM liées à l'administration des hôtes à un privilège nommé **full_host_administration**. Par défaut, les autorisations sont réparties entre les privilèges **Host Enrollment**, **Host Administrators** et **Host Group Administrator**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Le privilège ***full_host_administration*** existe. Pour plus d'informations sur la création d'un privilège à l'aide d'Ansible, voir Utilisation d'Ansible pour garantir la présence d'un privilège IdM RBAC personnalisé.

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **privilege-member-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/privilege/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-member-present.yml
privilege-member-present-copy.yml
```

3. Ouvrez le fichier **privilege-member-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaprivilege** task :
   - Adaptez le site **name** de la tâche pour qu'il corresponde à votre cas d'utilisation.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom du privilège.

- Définissez la liste **permission** avec les noms des autorisations que vous souhaitez inclure dans le privilège.
- Assurez-vous que la variable **action** est fixée à **member**.

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Privilege member present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that permissions are present for the "full_host_administration" privilege
    ipaprivilege:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: full_host_administration
      permission:
      - "System: Add krbPrincipalName to a Host"
      - "System: Enroll a Host"
      - "System: Manage Host Certificates"
      - "System: Manage Host Enrollment Password"
      - "System: Manage Host Keytab"
      - "System: Manage Host Principals"
      - "Retrieve Certificates from the CA"
      - "Revoke Certificate"
      - "System: Add Hosts"
      - "System: Add krbPrincipalName to a Host"
      - "System: Enroll a Host"
      - "System: Manage Host Certificates"
      - "System: Manage Host Enrollment Password"
      - "System: Manage Host Keytab"
      - "System: Manage Host Keytab Permissions"
      - "System: Manage Host Principals"
      - "System: Manage Host SSH Public Keys"
      - "System: Manage Service Keytab"
      - "System: Manage Service Keytab Permissions"
      - "System: Modify Hosts"
      - "System: Remove Hosts"
      - "System: Add Hostgroups"
      - "System: Modify Hostgroup Membership"
      - "System: Modify Hostgroups"
      - "System: Remove Hostgroups"
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory privilege-member-present-copy.yml
```

13.3. UTILISER ANSIBLE POUR S'ASSURER QU'UN PRIVILÈGE IDM RBAC N'INCLUT PAS UNE PERMISSION

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez personnaliser le contrôle d'accès basé sur les rôles de l'IdM.

La procédure suivante décrit comment utiliser un livre de jeu Ansible pour supprimer une autorisation d'un privilège. L'exemple décrit comment supprimer l'autorisation **Request Certificates ignoring CA ACLs** du privilège par défaut **Certificate Administrators** parce que, par exemple, l'administrateur considère qu'il s'agit d'un risque de sécurité.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **privilege-member-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/privilege/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-member-absent.yml
privilege-member-absent-copy.yml
```

3. Ouvrez le fichier **privilege-member-absent-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaprivilege** task :
   - Adaptez le site **name** de la tâche pour qu'il corresponde à votre cas d'utilisation.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom du privilège.
   - Définissez la liste **permission** sur les noms des autorisations que vous souhaitez supprimer du privilège.
   - Assurez-vous que la variable **action** est fixée à **member**.
   - Assurez-vous que la variable **state** est fixée à **absent**.

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Privilege absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "Request Certificate ignoring CA ACLs" permission is absent from
the "Certificate Administrators" privilege
    ipaprivilege:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: Certificate Administrators
      permission:
      - "Request Certificate ignoring CA ACLs"
      action: member
      state: absent
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory privilege-
member-absent-copy.yml
```

13.4. UTILISER ANSIBLE POUR RENOMMER UN PRIVILÈGE IDM RBAC PERSONNALISÉ

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez personnaliser le contrôle d'accès basé sur les rôles de l'IdM.

La procédure suivante décrit comment renommer un privilège parce que, par exemple, vous lui avez retiré quelques autorisations. Par conséquent, le nom du privilège n'est plus exact. Dans l'exemple, l'administrateur renomme un privilège **full_host_administration** en **limited_host_administration**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.

- Le privilège ***full_host_administration*** existe. Pour plus d'informations sur l'ajout d'un privilège, voir Utilisation d'Ansible pour s'assurer qu'un privilège IdM RBAC personnalisé est présent .

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **privilege-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/privilege/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-present.yml rename-privilege.yml
```

3. Ouvrez le fichier **rename-privilege.yml** Ansible playbook pour l'éditer.

4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaprivilege** task :

   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définir la variable **name** avec le nom actuel du privilège.
   - Ajoutez la variable **rename** et attribuez-lui le nouveau nom du privilège.
   - Ajoutez la variable **state** et fixez-la à **renamed**.

5. Renommer le playbook lui-même, par exemple :

```
---
- name: Rename a privilege
  hosts: ipaserver
```

6. Renommez la tâche dans le playbook, par exemple :

```
[...]
tasks:
- name: Ensure the full_host_administration privilege is renamed to limited_host_administration
  ipaprivilege:
  [...]
```

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Rename a privilege
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the full_host_administration privilege is renamed to limited_host_administration
    ipaprivilege:
      ipaadmin_password: "{{ ipaadmin_password }}"
```

```
    name: full_host_administration
    rename: limited_host_administration
    state: renamed
```

7. Enregistrer le fichier.

8. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory rename-privilege.yml
```

13.5. UTILISER ANSIBLE POUR S'ASSURER QU'UN PRIVILÈGE IDM RBAC EST ABSENT

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez personnaliser le contrôle d'accès basé sur les rôles de l'IdM. La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'un privilège RBAC est absent. L'exemple décrit comment s'assurer que le privilège **CA administrator** est absent. Grâce à cette procédure, l'administrateur de **admin** devient le seul utilisateur capable de gérer les autorités de certification dans IdM.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **privilege-absent.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/privilege/**:

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

3. Ouvrez le fichier **privilege-absent-copy.yml** Ansible playbook pour l'éditer.

4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaprivilege** task :

- Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
- Attribuez à la variable **name** le nom du privilège que vous souhaitez supprimer.
- Assurez-vous que la variable **state** est fixée à **absent**.

5. Renommez la tâche dans le playbook, par exemple :

```
[...]
tasks:
- name: Ensure privilege "CA administrator" is absent
  ipaprivilege:
  [...]
```

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Privilege absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure privilege "CA administrator" is absent
    ipaprivilege:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: CA administrator
      state: absent
```

6. Enregistrer le fichier.

7. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory privilege-absent-copy.yml
```

13.6. RESSOURCES SUPPLÉMENTAIRES

- Voir les privilèges dans l'IdM.
- Voir les autorisations dans IdM.
- Voir le fichier **README-privilege** disponible dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/ipaprivilege**.

CHAPITRE 14. UTILISER LES PLAYBOOKS ANSIBLE POUR GÉRER LES PERMISSIONS RBAC DANS IDM

Le contrôle d'accès basé sur les rôles (RBAC) est un mécanisme de contrôle d'accès neutre défini autour de rôles, de privilèges et de permissions. Dans les grandes entreprises en particulier, l'utilisation du RBAC peut aider à créer un système hiérarchique d'administrateurs avec leurs domaines de responsabilité individuels.

Ce chapitre décrit les opérations suivantes effectuées lors de la gestion des autorisations RBAC dans la gestion des identités (IdM) à l'aide des playbooks Ansible :

- Utiliser Ansible pour s'assurer qu'une permission RBAC est présente
- Utiliser Ansible pour s'assurer qu'une permission RBAC avec un attribut est présente
- Utiliser Ansible pour s'assurer qu'une permission RBAC est absente
- Utiliser Ansible pour s'assurer qu'un attribut est membre d'une permission IdM RBAC
- Utiliser Ansible pour s'assurer qu'un attribut n'est pas membre d'une permission RBAC IdM
- Utiliser Ansible pour renommer une permission IdM RBAC

Conditions préalables

- Vous comprenez les concepts et les principes de RBAC .

14.1. UTILISER ANSIBLE POUR S'ASSURER QU'UNE PERMISSION RBAC EST PRÉSENTE

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez personnaliser le contrôle d'accès basé sur les rôles (RBAC) de l'IdM.

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'une permission est présente dans IdM afin qu'elle puisse être ajoutée à un privilège. L'exemple décrit comment garantir l'état cible suivant :

- L'autorisation **MyPermission** existe.
- L'autorisation **MyPermission** ne peut être appliquée qu'aux hôtes.
- Un utilisateur bénéficiant d'un privilège contenant l'autorisation peut effectuer toutes les opérations suivantes sur une entrée :
  - Écrire
  - Lire
  - Recherche
  - Comparer
  - Ajouter
  - Supprimer

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **permission-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/permission/**:

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

3. Ouvrez le fichier **permission-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipapermission** task :
   - Adaptez le site **name** de la tâche pour qu'il corresponde à votre cas d'utilisation.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom de l'autorisation.
   - Fixer la variable **object_type** à **host**.
   - Fixer la variable **right** à **all**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Permission present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "MyPermission" permission is present
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
```

```
name: MyPermission
object_type: host
right: all
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-present-copy.yml
```

14.2. UTILISER ANSIBLE POUR S'ASSURER QU'UNE PERMISSION RBAC AVEC UN ATTRIBUT EST PRÉSENTE

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez personnaliser le contrôle d'accès basé sur les rôles (RBAC) de l'IdM.

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'une permission est présente dans IdM afin qu'elle puisse être ajoutée à un privilège. L'exemple décrit comment garantir l'état cible suivant :

- L'autorisation **MyPermission** existe.
- L'autorisation **MyPermission** ne peut être utilisée que pour ajouter des hôtes.
- Un utilisateur bénéficiant d'un privilège contenant l'autorisation peut effectuer toutes les opérations suivantes sur une entrée d'hôte :
  - Écrire
  - Lire
  - Recherche
  - Comparer
  - Ajouter
  - Supprimer
- Les entrées d'hôte créées par un utilisateur bénéficiant d'un privilège contenant l'autorisation **MyPermission** peuvent avoir une valeur **description**.

NOTE

Le type d'attribut que vous pouvez spécifier lors de la création ou de la modification d'une autorisation n'est pas limité par le schéma LDAP de l'IdM. Toutefois, le fait de spécifier, par exemple, **attrs: car_licence** si **object_type** est **host** plus tard entraîne le message d'erreur **ipa: ERROR: attribute "car-license" not allowed** lorsque vous essayez d'exercer l'autorisation et d'ajouter une valeur de permis de conduire spécifique à un hôte.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **permission-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/permission/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-present.yml
permission-present-with-attribute.yml
```

3. Ouvrez le fichier **permission-present-with-attribute.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipapermission** task :
   - Adaptez le site **name** de la tâche pour qu'il corresponde à votre cas d'utilisation.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom de l'autorisation.
   - Fixer la variable **object_type** à **host**.
   - Fixer la variable **right** à **all**.
   - Fixer la variable **attrs** à **description**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Permission present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "MyPermission" permission is present with an attribute
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: MyPermission
```

```
    object_type: host
    right: all
    attrs: description
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-present-with-attribute.yml
```

Ressources supplémentaires

- Voir Schéma des utilisateurs et des groupes dans *Linux Domain Identity, Authentication and Policy Guide* dans RHEL 7.

14.3. UTILISER ANSIBLE POUR S'ASSURER QU'UNE PERMISSION RBAC EST ABSENTE

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez personnaliser le contrôle d'accès basé sur les rôles (RBAC) de l'IdM.

La procédure suivante décrit comment utiliser un manuel de jeu Ansible pour s'assurer qu'une permission est absente dans IdM et qu'elle ne peut pas être ajoutée à un privilège.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **permission-absent.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/permission/**:

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

3. Ouvrez le fichier **permission-absent-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipapermission** task :
   - Adaptez le site **name** de la tâche pour qu'il corresponde à votre cas d'utilisation.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom de l'autorisation.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Permission absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "MyPermission" permission is absent
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: MyPermission
      state: absent
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

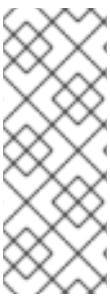
```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-absent-copy.yml
```

14.4. UTILISER ANSIBLE POUR S'ASSURER QU'UN ATTRIBUT EST MEMBRE D'UNE PERMISSION IDM RBAC

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez personnaliser le contrôle d'accès basé sur les rôles (RBAC) de l'IdM.

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'un attribut est membre d'une permission RBAC dans IdM. Par conséquent, un utilisateur disposant de la permission peut créer des entrées dotées de l'attribut.

L'exemple décrit comment garantir que les entrées d'hôte créées par un utilisateur disposant d'un privilège contenant l'autorisation **MyPermission** peuvent avoir les valeurs **gecos** et **description**.



NOTE

Le type d'attribut que vous pouvez spécifier lors de la création ou de la modification d'une autorisation n'est pas limité par le schéma LDAP de l'IdM. Toutefois, le fait de spécifier, par exemple, **attrs: car_licence** si **object_type** est **host** plus tard entraîne le message d'erreur **ipa: ERROR: attribute "car-license" not allowed** lorsque vous essayez d'exercer l'autorisation et d'ajouter une valeur de permis de conduire spécifique à un hôte.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- L'autorisation **MyPermission** existe.

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **permission-member-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/permission/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-member-present.yml permission-member-present-copy.yml
```

3. Ouvrez le fichier **permission-member-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipapermission** task :
   - Adaptez le site **name** de la tâche pour qu'il corresponde à votre cas d'utilisation.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom de l'autorisation.
   - Attribuer la liste **attrs** aux variables **description** et **gecos**.
   - Assurez-vous que la variable **action** est définie sur **member**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Permission member present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "gecos" and "description" attributes are present in "MyPermission"
```

```
  ipapermission:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: MyPermission
    attrs:
    - description
    - gecos
    action: member
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-member-present-copy.yml
```

14.5. UTILISER ANSIBLE POUR S'ASSURER QU'UN ATTRIBUT N'EST PAS MEMBRE D'UNE PERMISSION RBAC IDM

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez personnaliser le contrôle d'accès basé sur les rôles (RBAC) de l'IdM.

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'un attribut n'est pas membre d'une permission RBAC dans IdM. Par conséquent, lorsqu'un utilisateur disposant de cette permission crée une entrée dans IdM LDAP, cette entrée ne peut pas avoir de valeur associée à l'attribut.

L'exemple décrit comment assurer l'état suivant de la cible :

- L'autorisation **MyPermission** existe.
- Les entrées d'hôte créées par un utilisateur disposant d'un privilège contenant l'autorisation **MyPermission** ne peuvent pas avoir l'attribut **description**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- L'autorisation **MyPermission** existe.

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **permission-member-absent.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/permission/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-member-absent.yml permission-member-absent-copy.yml
```

3. Ouvrez le fichier **permission-member-absent-copy.yml** Ansible playbook pour l'éditer.

4. Adaptez le fichier en définissant les variables suivantes dans la section **ipapermission** task :

   - Adaptez le site **name** de la tâche pour qu'il corresponde à votre cas d'utilisation.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom de l'autorisation.
   - Fixer la variable **attrs** à **description**.
   - Fixer la variable **action** à **member**.
   - Assurez-vous que la variable **state** est définie comme suit **absent**

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Permission absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that an attribute is not a member of "MyPermission"
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: MyPermission
      attrs: description
      action: member
      state: absent
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-member-absent-copy.yml
```

14.6. UTILISER ANSIBLE POUR RENOMMER UNE PERMISSION IDM RBAC

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez personnaliser le contrôle d'accès basé sur les rôles de l'IdM.

La procédure suivante décrit comment utiliser un playbook Ansible pour renommer une autorisation. L'exemple décrit comment renommer **MyPermission** en **MyNewPermission**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Le site **MyPermission** existe dans l'IdM.
- Le site **MyNewPermission** n'existe pas dans l'IdM.

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **permission-renamed.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/permission/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-renamed.yml
permission-renamed-copy.yml
```

3. Ouvrez le fichier **permission-renamed-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipapermission** task :
   - Adaptez le site **name** de la tâche pour qu'il corresponde à votre cas d'utilisation.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Attribuez à la variable **name** le nom de l'autorisation.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Permission present example
  hosts: ipaserver

  vars_files:
```

```
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Rename the "MyPermission" permission
    ipapermission:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: MyPermission
      rename: MyNewPermission
      state: renamed
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-renamed-copy.yml
```

14.7. RESSOURCES SUPPLÉMENTAIRES

- Voir les autorisations dans IdM.
- Voir les privilèges dans l'IdM.
- Voir le fichier **README-permission** disponible dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/ipapermission**.

CHAPITRE 15. UTILISER ANSIBLE POUR GÉRER LA TOPOLOGIE DE RÉPLICATION DANS IDM

Vous pouvez maintenir plusieurs serveurs de gestion des identités (IdM) et les laisser se répliquer les uns les autres à des fins de redondance afin d'atténuer ou d'empêcher la perte de serveurs. Par exemple, si un serveur tombe en panne, les autres serveurs continuent à fournir des services au domaine. Vous pouvez également récupérer le serveur perdu en créant une nouvelle réplique basée sur l'un des serveurs restants.

Les données stockées sur un serveur IdM sont répliquées sur la base d'accords de réplication : lorsque deux serveurs ont un accord de réplication configuré, ils partagent leurs données. Les données répliquées sont stockées dans la topologie **suffixes**. Lorsque deux répliques ont un accord de réplication entre leurs suffixes, ces derniers forment une topologie **segment**.

Ce chapitre décrit comment utiliser **Red Hat Ansible Engine** pour gérer les accords de réplication IdM, les segments de topologie et les suffixes de topologie. Ce chapitre contient les sections suivantes :

- Utiliser Ansible pour s'assurer qu'un accord de réplication existe dans IdM
- Utiliser Ansible pour s'assurer que des accords de réplication existent entre plusieurs répliques IdM
- Utiliser Ansible pour vérifier l'existence d'un accord de réplication entre deux répliques
- Utiliser Ansible pour vérifier qu'un suffixe de topologie existe dans IdM
- Utiliser Ansible pour réinitialiser une réplique IdM
- Utiliser Ansible pour s'assurer qu'un accord de réplication est absent dans IdM

15.1. UTILISER ANSIBLE POUR S'ASSURER QU'UN ACCORD DE RÉPLICATION EXISTE DANS IDM

Les données stockées sur un serveur de gestion des identités (IdM) sont répliquées sur la base d'accords de réplication : lorsque deux serveurs ont un accord de réplication configuré, ils partagent leurs données. Les accords de réplication sont toujours bilatéraux : les données sont répliquées de la première réplique vers l'autre et de l'autre réplique vers la première.

Cette section décrit comment utiliser un playbook Ansible pour s'assurer qu'un accord de réplication de type **domain** existe entre **server.idm.example.com** et **replica.idm.example.com**.

Conditions préalables

- Assurez-vous de bien comprendre les recommandations relatives à la conception de votre topologie IdM (voir Connexion des répliques dans une topologie).
- 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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **add-topologysegment.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegment.yml
add-topologysegment-copy.yml
```

3. Ouvrez le fichier **add-topologysegment-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipatopologysegment** task :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Définissez la variable **suffix** à **domain** ou **ca**, en fonction du type de segment que vous souhaitez ajouter.
   - Définissez la variable **left** avec le nom du serveur IdM que vous voulez être le nœud gauche de l'accord de réplication.
   - Définissez la variable **right** avec le nom du serveur IdM que vous voulez être le nœud droit de l'accord de réplication.
   - Assurez-vous que la variable **state** est définie sur **present**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to handle topologysegment
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
- name: Add topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      left: server.idm.example.com
      right: replica.idm.example.com
      state: present
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-topologysegment-copy.yml
```

Ressources supplémentaires

- Voir Explication des accords de réplication, des suffixes de topologie et des segments de topologie.
- Voir le fichier **README-topology.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology**.

15.2. UTILISER ANSIBLE POUR S'ASSURER QUE DES ACCORDS DE RÉPLICATION EXISTENT ENTRE PLUSIEURS RÉPLIQUES IDM

Les données stockées sur un serveur de gestion des identités (IdM) sont répliquées sur la base d'accords de réplication : lorsque deux serveurs ont un accord de réplication configuré, ils partagent leurs données. Les accords de réplication sont toujours bilatéraux : les données sont répliquées de la première réplique vers l'autre et de l'autre réplique vers la première.

Cette section décrit comment garantir l'existence d'accords de réplication entre plusieurs paires de répliques dans IdM.

Conditions préalables

- Assurez-vous de bien comprendre les recommandations relatives à la conception de votre topologie IdM, énumérées dans la section Connexion des répliques dans une topologie
- 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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **add-topologysegments.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegments.yml
add-topologysegments-copy.yml
```

3. Ouvrez le fichier **add-topologysegments-copy.yml** pour le modifier.

4. Adaptez le fichier en définissant les variables suivantes dans la section **vars**:

   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.

   - Pour chaque segment topologique, ajoutez une ligne dans la section **ipatopology_segments** et définissez les variables suivantes :

     - Définissez la variable **suffix** à **domain** ou **ca**, en fonction du type de segment que vous souhaitez ajouter.

     - Définissez la variable **left** avec le nom du serveur IdM que vous voulez être le nœud gauche de l'accord de réplication.

     - Définissez la variable **right** avec le nom du serveur IdM que vous voulez être le nœud droit de l'accord de réplication.

5. Dans la section **tasks** du fichier **add-topologysegments-copy.yml**, assurez-vous que la variable **state** est fixée à **present**.
   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Add topology segments
  hosts: ipaserver
  gather_facts: false

  vars:
    ipaadmin_password: "{{ ipaadmin_password }}"
    ipatopology_segments:
    - {suffix: domain, left: replica1.idm.example.com , right: replica2.idm.example.com }
    - {suffix: domain, left: replica2.idm.example.com , right: replica3.idm.example.com }
    - {suffix: domain, left: replica3.idm.example.com , right: replica4.idm.example.com }
    - {suffix: domain+ca, left: replica4.idm.example.com , right: replica1.idm.example.com }

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Add topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: "{{ item.suffix }}"
      name: "{{ item.name | default(omit) }}"
      left: "{{ item.left }}"
      right: "{{ item.right }}"
      state: present
      #state: absent
      #state: checked
      #state: reinitialized
    loop: "{{ ipatopology_segments | default([]) }}"
```

6. Enregistrer le fichier.

7. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-topologysegments-copy.yml
```

Ressources supplémentaires

- Voir Explication des accords de réplication, des suffixes de topologie et des segments de topologie.
- Voir le fichier **README-topology.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology**.

15.3. UTILISER ANSIBLE POUR VÉRIFIER L'EXISTENCE D'UN ACCORD DE RÉPLICATION ENTRE DEUX RÉPLIQUES

Les données stockées sur un serveur de gestion des identités (IdM) sont répliquées sur la base d'accords de réplication : lorsque deux serveurs ont un accord de réplication configuré, ils partagent leurs données. Les accords de réplication sont toujours bilatéraux : les données sont répliquées de la première réplique vers l'autre et de l'autre réplique vers la première.

Cette section décrit comment vérifier que des accords de réplication existent entre plusieurs paires de répliques dans IdM.

Conditions préalables

- Assurez-vous de bien comprendre les recommandations relatives à la conception de votre topologie de gestion des identités (IdM), énumérées à la section Connexion des répliques dans une topologie.
- 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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **check-topologysegments.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/check-topologysegments.yml
check-topologysegments-copy.yml
```

3. Ouvrez le fichier **check-topologysegments-copy.yml** pour le modifier.

4. Adaptez le fichier en définissant les variables suivantes dans la section **vars**:

   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.

   - Pour chaque segment topologique, ajoutez une ligne dans la section **ipatopology_segments** et définissez les variables suivantes :

     - Définissez la variable **suffix** à **domain** ou **ca**, en fonction du type de segment que vous ajoutez.

     - Définissez la variable **left** avec le nom du serveur IdM que vous voulez être le nœud gauche de l'accord de réplication.

     - Définissez la variable **right** avec le nom du serveur IdM que vous voulez être le nœud droit de l'accord de réplication.

5. Dans la section **tasks** du fichier **check-topologysegments-copy.yml**, assurez-vous que la variable **state** est fixée à **present**.
   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Add topology segments
  hosts: ipaserver
  gather_facts: false

  vars:
    ipaadmin_password: "{{ ipaadmin_password }}"
    ipatopology_segments:
    - {suffix: domain, left: replica1.idm.example.com, right: replica2.idm.example.com }
    - {suffix: domain, left: replica2.idm.example.com , right: replica3.idm.example.com }
    - {suffix: domain, left: replica3.idm.example.com , right: replica4.idm.example.com }
    - {suffix: domain+ca, left: replica4.idm.example.com , right:
replica1.idm.example.com }

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Check topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: "{{ item.suffix }}"
      name: "{{ item.name | default(omit) }}"
      left: "{{ item.left }}"
      right: "{{ item.right }}"
      state: checked
    loop: "{{ ipatopology_segments | default([]) }}"
```

6. Enregistrer le fichier.

7. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory check-topologysegments-copy.yml
```

Ressources supplémentaires

- Pour plus d'informations sur le concept d'accords, de suffixes et de segments de topologie, voir Explication des accords de réplication, des suffixes de topologie et des segments de topologie .
- Voir le fichier **README-topology.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology**.

15.4. UTILISER ANSIBLE POUR VÉRIFIER QU'UN SUFFIXE DE TOPOLOGIE EXISTE DANS IDM

Dans le contexte des accords de réplication de la gestion des identités (IdM), les suffixes de topologie stockent les données qui sont répliquées. L'IdM prend en charge deux types de suffixes de topologie : **domain** et **ca**. Chaque suffixe représente un back-end distinct, une topologie de réplication distincte. Lorsqu'un accord de réplication est configuré, il relie deux suffixes de topologie du même type sur deux serveurs différents.

Le suffixe **domain** contient toutes les données relatives au domaine, telles que les utilisateurs, les groupes et les stratégies. Le suffixe **ca** contient les données relatives au système de certification. Il n'est présent que sur les serveurs sur lesquels une autorité de certification (CA) est installée.

Cette section décrit comment utiliser un playbook Ansible pour s'assurer qu'un suffixe de topologie existe dans IdM. L'exemple décrit comment s'assurer que le suffixe **domain** existe dans IdM.

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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **verify-topologysuffix.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/ verify-topologysuffix.yml
verify-topologysuffix-copy.yml
```

3. Ouvrez le fichier **verify-topologysuffix-copy.yml** Ansible playbook pour l'éditer.

4. Adaptez le fichier en définissant les variables suivantes dans la section **ipatopologysuffix**:

   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Définissez la variable **suffix** à **domain**. Si vous vérifiez la présence du suffixe **ca**, définissez la variable à **ca**.
   - Assurez-vous que la variable **state** est définie sur **verified**. Aucune autre option n'est possible.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to handle topologysuffix
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Verify topology suffix
    ipatopologysuffix:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      state: verified
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory verify-
topologysuffix-copy.yml
```

Ressources supplémentaires

- Voir Explication des accords de réplication, des suffixes de topologie et des segments de topologie.
- Voir le fichier **README-topology.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology**.

15.5. UTILISER ANSIBLE POUR RÉINITIALISER UNE RÉPLIQUE IDM

Si un réplica a été déconnecté pendant une longue période ou si sa base de données a été corrompue, vous pouvez le réinitialiser. La réinitialisation actualise le réplica avec un ensemble de données mises à jour. La réinitialisation peut, par exemple, être utilisée si une restauration autoritaire à partir d'une sauvegarde est nécessaire.

NOTE

Contrairement aux mises à jour de réplication, au cours desquelles les répliques ne s'envoient que les entrées modifiées, la réinitialisation rafraîchit l'ensemble de la base de données.

L'hôte local sur lequel vous exécutez la commande est le réplica réinitialisé. Pour spécifier le réplica à partir duquel les données sont obtenues, utilisez l'option **direction**.

Cette section décrit comment utiliser un playbook Ansible pour réinitialiser les données **domain** sur **replica.idm.example.com** à partir de **server.idm.example.com**.

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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **reinitialize-topologysegment.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/reinitialize-topologysegment.yml reinitialize-topologysegment-copy.yml
```

3. Ouvrez le fichier **reinitialize-topologysegment-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipatopologysegment**:
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Fixez la variable **suffix** à **domain**. Si vous réinitialisez les données **ca**, fixez la variable à **ca**.
   - Définissez la variable **left** sur le nœud gauche de l'accord de réplication.

- Définissez la variable **right** sur le nœud de droite de l'accord de réplication.
- Définissez la variable **direction** en fonction de la direction des données réinitialisées. La direction **left-to-right** signifie que les données circulent du nœud gauche vers le nœud droit.
- Assurez-vous que la variable **state** est définie sur **reinitialized**.
  Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to handle topologysegment
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Reinitialize topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      left: server.idm.example.com
      right: replica.idm.example.com
      direction: left-to-right
      state: reinitialized
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory reinitialize-topologysegment-copy.yml
```

Ressources supplémentaires

- Voir Explication des accords de réplication, des suffixes de topologie et des segments de topologie.
- Voir le fichier **README-topology.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology**.

15.6. UTILISER ANSIBLE POUR S'ASSURER QU'UN ACCORD DE RÉPLICATION EST ABSENT DANS IDM

Les données stockées sur un serveur de gestion des identités (IdM) sont répliquées sur la base d'accords de réplication : lorsque deux serveurs ont un accord de réplication configuré, ils partagent leurs données. Les accords de réplication sont toujours bilatéraux : les données sont répliquées de la première réplique vers l'autre et de l'autre réplique vers la première.

Cette section décrit comment s'assurer qu'un accord de réplication entre deux répliques n'existe pas dans IdM. L'exemple décrit comment s'assurer qu'un accord de réplication de type **domain** n'existe pas entre les serveurs IdM **replica01.idm.example.com** et **replica02.idm.example.com**.

Conditions préalables

Conditions préalables

- Assurez-vous de bien comprendre les recommandations relatives à la conception de votre topologie IdM, énumérées dans la section Connexion des répliques dans une topologie
- 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. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **delete-topologysegment.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/delete-topologysegment.yml
delete-topologysegment-copy.yml
```

3. Ouvrez le fichier **delete-topologysegment-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipatopologysegment** task :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Attribuez la valeur **domain** à la variable **suffix**. Si vous voulez vous assurer que les données de **ca** ne sont pas répliquées entre les nœuds de gauche et de droite, définissez la variable à **ca**.
   - Définissez la variable **left** avec le nom du serveur IdM qui est le nœud gauche de l'accord de réplication.
   - Définissez la variable **right** avec le nom du serveur IdM qui est le nœud droit de l'accord de réplication.
   - Assurez-vous que la variable **state** est définie sur **absent**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to handle topologysegment
  hosts: ipaserver
```

```
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
- name: Delete topology segment
    ipatopologysegment:
      ipaadmin_password: "{{ ipaadmin_password }}"
      suffix: domain
      left: replica01.idm.example.com
      right: replica02.idm.example.com:
      state: absent
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory delete-topologysegment-copy.yml
```

Ressources supplémentaires

- Voir Explication des accords de réplication, des suffixes de topologie et des segments de topologie.
- Voir le fichier **README-topology.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/topology**.

15.7. RESSOURCES SUPPLÉMENTAIRES

- Voir Planification de la topologie du réplica .
- Voir Installation d'un réplica IdM .

CHAPITRE 16. GÉRER LES SERVEURS IDM À L'AIDE D'ANSIBLE

Vous pouvez utiliser **Red Hat Ansible Engine** pour gérer les serveurs dans votre topologie de gestion des identités (IdM). Vous pouvez utiliser le module **server** dans le paquetage **ansible-freeipa** pour vérifier la présence ou l'absence d'un serveur dans la topologie IdM. Vous pouvez également masquer une réplique ou rendre une réplique visible.

Cette section contient les rubriques suivantes :

- Vérification de la présence d'un serveur IdM à l'aide d'Ansible
- S'assurer qu'un serveur IdM est absent d'une topologie IdM en utilisant Ansible
- Assurer l'absence d'un serveur IdM malgré l'hébergement d'un dernier rôle de serveur IdM
- Veiller à ce qu'un serveur IdM soit absent mais pas nécessairement déconnecté des autres serveurs IdM
- S'assurer qu'un serveur IdM existant est caché à l'aide d'un playbook Ansible
- S'assurer qu'un serveur IdM existant est visible à l'aide d'un playbook Ansible
- S'assurer qu'un serveur IdM existant dispose d'un emplacement DNS IdM assigné
- S'assurer qu'aucun emplacement DNS IdM n'est attribué à un serveur IdM existant

16.1. VÉRIFICATION DE LA PRÉSENCE D'UN SERVEUR IDM À L'AIDE D'ANSIBLE

Vous pouvez utiliser le module **ipaserver ansible-freeipa** dans un playbook Ansible pour vérifier l'existence d'un serveur de gestion des identités (IdM).

NOTE

Le module Ansible **ipaserver** n'installe pas le serveur IdM.

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**.
  - La connexion **SSH** entre le nœud de contrôle et le serveur IdM défini dans le fichier d'inventaire fonctionne correctement.

Procédure

1. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

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

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

3. Ouvrez le fichier **server-present-copy.yml** pour le modifier.

4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaserver** task et enregistrez le fichier :

   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Attribuez à la variable **name** la valeur **FQDN** du serveur. L'adresse **FQDN** du serveur de l'exemple est **server123.idm.example.com**.

```
---
- name: Server present example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com is present
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
```

5. Exécutez le playbook Ansible et indiquez le fichier du playbook et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-present-copy.yml
```

Ressources supplémentaires

- Voir Installation d'un serveur de gestion des identités à l'aide d'un playbook Ansible .
- Voir le fichier **README-server.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server**.

16.2. S'ASSURER QU'UN SERVEUR IDM EST ABSENT D'UNE TOPOLOGIE IDM EN UTILISANT ANSIBLE

Utiliser un playbook Ansible pour s'assurer qu'un serveur de gestion des identités (IdM) n'existe pas dans une topologie IdM, même en tant qu'hôte.

Contrairement au rôle **ansible-freeipa ipaserver** , le module **ipaserver** utilisé dans ce playbook ne désinstalle pas les services IdM du serveur.

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**.
  - La connexion **SSH** entre le nœud de contrôle et le serveur IdM défini dans le fichier d'inventaire fonctionne correctement.

Procédure

1. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

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

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

3. Ouvrez le fichier **server-absent-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaserver** task et enregistrez le fichier :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Attribuez à la variable **name** la valeur **FQDN** du serveur. L'adresse **FQDN** du serveur de l'exemple est **server123.idm.example.com**.
   - Assurez-vous que la variable **state** est définie sur **absent**.

```
---
- name: Server absent example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com is absent
    ipaserver:
```

```
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: server123.idm.example.com
    state: absent
```

5. Exécutez le playbook Ansible et indiquez le fichier du playbook et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-absent-copy.yml
```

6. Assurez-vous que tous les enregistrements DNS du serveur de noms (NS) pointant vers **server123.idm.example.com** sont supprimés de vos zones DNS. Ceci s'applique indépendamment du fait que vous utilisiez un DNS intégré géré par IdM ou un DNS externe.

Ressources supplémentaires

- Voir Désinstallation d'un serveur IdM .
- Voir le fichier **README-server.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server**.

16.3. ASSURER L'ABSENCE D'UN SERVEUR IDM MALGRÉ L'HÉBERGEMENT D'UN DERNIER RÔLE DE SERVEUR IDM

Vous pouvez utiliser Ansible pour vous assurer qu'un serveur de gestion des identités (IdM) est absent même si la dernière instance de service IdM est en cours d'exécution sur le serveur. Une autorité de certification (CA), une autorité de récupération des clés (KRA) ou un serveur DNS sont des exemples de services IdM.

AVERTISSEMENT

Si vous supprimez le dernier serveur qui sert de CA, de KRA ou de serveur DNS, vous perturbez sérieusement la fonctionnalité IdM. Vous pouvez vérifier manuellement quels services sont exécutés sur quels serveurs IdM à l'aide de la commande **ipa service-find**. Le nom principal d'un serveur CA est **dogtag/*server_name*/*REALM_NAME***.

Contrairement au rôle **ansible-freeipa ipaserver** , le module **ipaserver** utilisé dans ce playbook ne désinstalle pas les services IdM du serveur.

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**.
- La connexion **SSH** entre le nœud de contrôle et le serveur IdM défini dans le fichier d'inventaire fonctionne correctement.

Procédure

1. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **server-absent-ignore-last-of-role.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-absent-ignore-last-of-role.yml server-absent-ignore-last-of-role-copy.yml
```

3. Ouvrez le fichier **server-absent-ignore-last-of-role-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaserver** task et enregistrez le fichier :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Attribuez à la variable **name** la valeur **FQDN** du serveur. L'adresse **FQDN** du serveur de l'exemple est **server123.idm.example.com**.
   - Assurez-vous que la variable **ignore_last_of_role** est définie sur **yes**.
   - Fixer la variable **state** à **absent**.

```
---
- name: Server absent with last of role skip example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server “server123.idm.example.com” is absent with last of role skip
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      ignore_last_of_role: yes
      state: absent
```

5. Exécutez le playbook Ansible et indiquez le fichier du playbook et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-absent-ignore-last-of-role-copy.yml
```

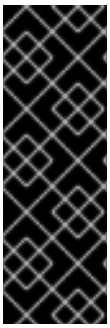
6. Assurez-vous que tous les enregistrements DNS du serveur de noms (NS) qui pointent vers **server123.idm.example.com** sont supprimés de vos zones DNS. Cela s'applique indépendamment du fait que vous utilisiez un DNS intégré géré par IdM ou un DNS externe.

Ressources supplémentaires

- Voir Désinstallation d'un serveur IdM .
- Voir le fichier **README-server.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server**.

16.4. VEILLER À CE QU'UN SERVEUR IDM SOIT ABSENT MAIS PAS NÉCESSAIREMENT DÉCONNECTÉ DES AUTRES SERVEURS IDM

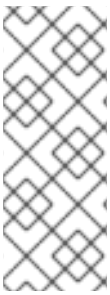
Si vous supprimez un serveur de gestion des identités (IdM) de la topologie, vous pouvez conserver ses accords de réplication intacts à l'aide d'un manuel de jeu Ansible. Le playbook garantit également que le serveur IdM n'existe pas dans IdM, même en tant qu'hôte.



IMPORTANT

Ignorer les accords de réplication d'un serveur lors de sa suppression n'est recommandé que si les autres serveurs sont des serveurs dysfonctionnels que vous prévoyez de supprimer de toute façon. La suppression d'un serveur qui sert de point central dans la topologie peut diviser votre topologie en deux clusters déconnectés.

Vous pouvez supprimer un serveur dysfonctionnel de la topologie à l'aide de la commande **ipa server-del**.



NOTE

Si vous supprimez le dernier serveur qui sert d'autorité de certification (CA), d'autorité de récupération des clés (KRA) ou de serveur DNS, vous perturbez sérieusement la fonctionnalité de gestion des identités (IdM). Pour éviter ce problème, le livre de jeu s'assure que ces services sont exécutés sur un autre serveur du domaine avant de désinstaller un serveur qui sert d'autorité de certification, d'autorité de récupération des clés (KRA) ou de serveur DNS.

Contrairement au rôle **ansible-freeipa ipaserver** , le module **ipaserver** utilisé dans ce playbook ne désinstalle pas les services IdM du serveur.

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**.
- La connexion **SSH** entre le nœud de contrôle et le serveur IdM défini dans le fichier d'inventaire fonctionne correctement.

Procédure

1. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **server-absent-ignore_topology_disconnect.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-absent-ignore_topology_disconnect.yml server-absent-ignore_topology_disconnect-copy.yml
```

3. Ouvrez le fichier **server-absent-ignore_topology_disconnect-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaserver** task et enregistrez le fichier :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Attribuez à la variable **name** la valeur **FQDN** du serveur. L'adresse **FQDN** du serveur de l'exemple est **server123.idm.example.com**.
   - Assurez-vous que la variable **ignore_topology_disconnect** est définie sur **yes**.
   - Assurez-vous que la variable **state** est définie sur **absent**.

```
---
- name: Server absent with ignoring topology disconnects example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server “server123.idm.example.com” with ignoring topology disconnects
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      ignore_topology_disconnect: yes
      state: absent
```

5. Exécutez le playbook Ansible et indiquez le fichier du playbook et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-absent-ignore_topology_disconnect-copy.yml
```

6. [Facultatif] Assurez-vous que tous les enregistrements DNS du serveur de noms (NS) pointant vers **server123.idm.example.com** sont supprimés de vos zones DNS. Ceci s'applique indépendamment du fait que vous utilisiez un DNS intégré géré par IdM ou un DNS externe.

Ressources supplémentaires

- Voir Désinstallation d'un serveur IdM .
- Voir le fichier **README-server.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server**.

16.5. S'ASSURER QU'UN SERVEUR IDM EXISTANT EST CACHÉ À L'AIDE D'UN PLAYBOOK ANSIBLE

Utilisez le module **ipaserver ansible-freeipa** dans un playbook Ansible pour vous assurer qu'un serveur de gestion des identités (IdM) existant est caché. Notez que ce playbook n'installe pas le serveur IdM.

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**.
  - La connexion **SSH** entre le nœud de contrôle et le serveur IdM défini dans le fichier d'inventaire fonctionne correctement.

Procédure

1. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

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

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-hidden.yml server-hidden-copy.yml
```

3. Ouvrez le fichier **server-hidden-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaserver** task et enregistrez le fichier :

- Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
- Attribuez à la variable **name** la valeur **FQDN** du serveur. L'adresse **FQDN** du serveur de l'exemple est **server123.idm.example.com**.
- Assurez-vous que la variable **hidden** est définie sur **True**.

```
---
- name: Server hidden example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com is hidden
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      hidden: True
```

5. Exécutez le playbook Ansible et indiquez le fichier du playbook et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-hidden-copy.yml
```

Ressources supplémentaires

- Voir Installation d'un serveur de gestion des identités à l'aide d'un playbook Ansible .
- Voir Le mode réplique caché.
- Voir le fichier **README-server.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server**.

16.6. ASSURER LA VISIBILITÉ D'UN SERVEUR IDM EXISTANT À L'AIDE D'UN PLAYBOOK ANSIBLE

Utilisez le module **ipaserver ansible-freeipa** dans un playbook Ansible pour vous assurer qu'un serveur de gestion des identités (IdM) existant est visible. Notez que ce playbook n'installe pas le serveur IdM.

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**.
- La connexion **SSH** entre le nœud de contrôle et le serveur IdM défini dans le fichier d'inventaire fonctionne correctement.

Procédure

1. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **server-not-hidden.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-not-hidden.yml server-not-hidden-copy.yml
```

3. Ouvrez le fichier **server-not-hidden-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaserver** task et enregistrez le fichier :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Attribuez à la variable **name** la valeur **FQDN** du serveur. L'adresse **FQDN** du serveur de l'exemple est **server123.idm.example.com**.
   - Assurez-vous que la variable **hidden** est définie sur **no**.

```
---
- name: Server not hidden example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com is not hidden
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      hidden: no
```

5. Exécutez le playbook Ansible et indiquez le fichier du playbook et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-not-hidden-copy.yml
```

Ressources supplémentaires

- Voir Installation d'un serveur de gestion des identités à l'aide d'un playbook Ansible .
- Voir Le mode réplique caché.
- Voir le fichier **README-server.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server**.

16.7. S'ASSURER QU'UN SERVEUR IDM EXISTANT DISPOSE D'UN EMPLACEMENT DNS IDM ASSIGNÉ

Utilisez le module **ipaserver ansible-freeipa** dans un playbook Ansible pour vous assurer qu'un serveur de gestion des identités (IdM) existant se voit attribuer un emplacement DNS IdM spécifique.

Notez que le module Ansible **ipaserver** n'installe pas le serveur IdM.

Conditions préalables

- Vous connaissez le mot de passe de l'IdM **admin**.
- L'emplacement DNS de l'IdM existe. L'exemple d'emplacement est **germany**.
- Vous avez accès au serveur à l'adresse **root**. L'exemple de serveur est **server123.idm.example.com**.
- 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**.
  - La connexion **SSH** entre le nœud de contrôle et le serveur IdM défini dans le fichier d'inventaire fonctionne correctement.

Procédure

1. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

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

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-location.yml server-location-copy.yml
```

3. Ouvrez le fichier **server-location-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaserver** task et enregistrez le fichier :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.

- Fixer la variable **name** à **server123.idm.example.com**.
- Fixer la variable **location** à **germany**.

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Server enabled example
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com with location “germany” is present
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      location: germany
```

5. Exécutez le playbook Ansible et indiquez le fichier du playbook et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-location-copy.yml
```

6. Connectez-vous à **server123.idm.example.com** en tant que **root** en utilisant **SSH**:

```
ssh root@server123.idm.example.com
```

7. Redémarrez le service **named-pkcs11** sur le serveur pour que les mises à jour prennent effet immédiatement :

```
[root@server123.idm.example.com ~]# systemctl restart named-pkcs11
```

Ressources supplémentaires

- Voir Installation d'un serveur de gestion des identités à l'aide d'un playbook Ansible .
- Voir Utiliser Ansible pour s'assurer qu'un emplacement IdM est présent .
- Voir le fichier **README-server.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server**.

16.8. S'ASSURER QU'AUCUN EMPLACEMENT DNS IDM N'EST ATTRIBUÉ À UN SERVEUR IDM EXISTANT

Utilisez le module **ipaserver ansible-freeipa** dans un playbook Ansible pour vous assurer qu'un serveur de gestion des identités (IdM) existant n'a pas d'emplacement DNS IdM qui lui soit assigné. N'attribuez pas d'emplacement DNS aux serveurs qui changent fréquemment d'emplacement géographique. Notez que le playbook n'installe pas le serveur IdM.

Conditions préalables

- Vous connaissez le mot de passe de l'IdM **admin**.
- Vous avez accès au serveur à l'adresse **root**. L'exemple de serveur est **server123.idm.example.com**.
- 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**.
  - La connexion **SSH** entre le nœud de contrôle et le serveur IdM défini dans le fichier d'inventaire fonctionne correctement.

Procédure

1. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Copiez le fichier **server-no-location.yml** Ansible playbook situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server/**:

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-no-location.yml server-no-location-copy.yml
```

3. Ouvrez le fichier **server-no-location-copy.yml** pour le modifier.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipaserver** task et enregistrez le fichier :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'IdM **admin**.
   - Fixer la variable **name** à **server123.idm.example.com**.
   - Assurez-vous que la variable **location** est définie sur **""**.

```
---
- name: Server no location example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure server server123.idm.example.com is present with no location
    ipaserver:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: server123.idm.example.com
      location: “”
```

5. Exécutez le playbook Ansible et indiquez le fichier du playbook et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-no-location-copy.yml
```

6. Connectez-vous à **server123.idm.example.com** en tant que **root** en utilisant **SSH**:

```
ssh root@server123.idm.example.com
```

7. Redémarrez le service **named-pkcs11** sur le serveur pour que les mises à jour prennent effet immédiatement :

```
[root@server123.idm.example.com ~]# systemctl restart named-pkcs11
```

Ressources supplémentaires

- Voir Installation d'un serveur de gestion des identités à l'aide d'un playbook Ansible .
- Voir Utiliser Ansible pour gérer les emplacements DNS dans IdM .
- Voir le fichier **README-server.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/server**.

CHAPITRE 17. GÉRER LES HÔTES À L'AIDE DES PLAYBOOKS ANSIBLE

Ansible est un outil d'automatisation utilisé pour configurer des systèmes, déployer des logiciels et effectuer des mises à jour continues. Ansible prend en charge la gestion des identités (IdM) et vous pouvez utiliser des modules Ansible pour automatiser la gestion des hôtes.

Ce chapitre décrit les concepts suivants et les opérations effectuées lors de la gestion des hôtes et des entrées d'hôtes à l'aide des playbooks Ansible :

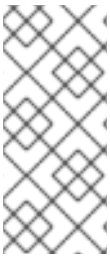
- Assurer la présence d'entrées d'hôtes IdM qui ne sont définies que par leur **FQDNs**
- S'assurer de la présence d'entrées d'hôtes IdM avec des adresses IP
- Assurer la présence de plusieurs entrées d'hôtes IdM avec des mots de passe aléatoires
- Assurer la présence d'une entrée d'hôte IdM avec plusieurs adresses IP
- Garantir l'absence d'entrées d'hôtes IdM

17.1. S'ASSURER DE LA PRÉSENCE D'UNE ENTRÉE D'HÔTE IDM AVEC FQDN À L'AIDE DES PLAYBOOKS ANSIBLE

Cette section décrit comment assurer la présence d'entrées d'hôtes dans la gestion des identités (IdM) à l'aide de playbooks Ansible. Les entrées d'hôte sont uniquement définies par leur **fully-qualified domain names** (FQDN).

Il suffit de spécifier le nom **FQDN** de l'hôte si au moins l'une des conditions suivantes s'applique :

- Le serveur IdM n'est pas configuré pour gérer les DNS.
- L'hôte n'a pas d'adresse IP statique ou l'adresse IP n'est pas connue au moment de la configuration de l'hôte. L'ajout d'un hôte défini uniquement par une adresse **FQDN** crée essentiellement une entrée de remplacement dans le service DNS IdM. Par exemple, des ordinateurs portables peuvent être préconfigurés comme clients IdM, mais ils n'ont pas d'adresse IP au moment de leur configuration. Lorsque le service DNS met à jour dynamiquement ses enregistrements, l'adresse IP actuelle de l'hôte est détectée et son enregistrement DNS est mis à jour.



NOTE

Sans Ansible, les entrées d'hôtes sont créées dans IdM à l'aide de la commande **ipa host-add**. Le résultat de l'ajout d'un hôte à IdM est l'état de l'hôte présent dans IdM. En raison de la dépendance d'Ansible à l'égard de l'idempotence, pour ajouter un hôte à IdM à l'aide d'Ansible, vous devez créer un playbook dans lequel vous définissez l'état de l'hôte comme étant présent : **state: present**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créez un fichier Ansible playbook avec le **FQDN** de l'hôte dont vous voulez assurer la présence dans IdM. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/host/add-host.yml**:

```
---
- name: Host present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Host host01.idm.example.com present
    ipahost:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: host01.idm.example.com
      state: present
      force: yes
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-host-
is-present.yml
```

NOTE

La procédure aboutit à la création d'une entrée d'hôte dans le serveur LDAP IdM, mais pas à l'enrôlement de l'hôte dans le domaine Kerberos IdM. Pour cela, vous devez déployer l'hôte en tant que client IdM. Pour plus de détails, voir Installation d'un client de gestion d'identité à l'aide d'un playbook Ansible.

Verification steps

1. Connectez-vous à votre serveur IdM en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
```

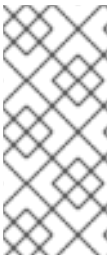
2. Entrez la commande **ipa host-show** et indiquez le nom de l'hôte :

```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Password: False
  Keytab: False
  Managed by: host01.idm.example.com
```

Le résultat confirme que **host01.idm.example.com** existe dans IdM.

17.2. ASSURER LA PRÉSENCE D'UNE ENTRÉE D'HÔTE IDM AVEC DES INFORMATIONS DNS EN UTILISANT LES PLAYBOOKS ANSIBLE

Cette section décrit comment assurer la présence d'entrées d'hôtes dans la gestion des identités (IdM) à l'aide des playbooks Ansible. Les entrées d'hôte sont définies par leur **fully-qualified domain names** (FQDN) et leurs adresses IP.



NOTE

Sans Ansible, les entrées d'hôtes sont créées dans IdM à l'aide de la commande **ipa host-add**. Le résultat de l'ajout d'un hôte à IdM est l'état de l'hôte présent dans IdM. En raison de la dépendance d'Ansible à l'égard de l'idempotence, pour ajouter un hôte à IdM à l'aide d'Ansible, vous devez créer un playbook dans lequel vous définissez l'état de l'hôte comme étant présent : **state: present**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

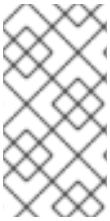
2. Créez un fichier playbook Ansible avec le **fully-qualified domain name** (FQDN) de l'hôte dont vous voulez assurer la présence dans l'IdM. En outre, si le serveur IdM est configuré pour gérer le DNS et que vous connaissez l'adresse IP de l'hôte, spécifiez une valeur pour le paramètre **ip_address**. L'adresse IP est nécessaire pour que l'hôte existe dans les enregistrements de

ressources DNS. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/host/host-present.yml**. Vous pouvez également inclure d'autres informations supplémentaires :

```
---
- name: Host present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure host01.idm.example.com is present
    ipahost:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: host01.idm.example.com
      description: Example host
      ip_address: 192.168.0.123
      locality: Lab
      ns_host_location: Lab
      ns_os_version: CentOS 7
      ns_hardware_platform: Lenovo T61
      mac_address:
      - "08:00:27:E3:B1:2D"
      - "52:54:00:BD:97:1E"
      state: present
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-host-
is-present.yml
```



NOTE

La procédure aboutit à la création d'une entrée d'hôte dans le serveur LDAP IdM, mais pas à l'enrôlement de l'hôte dans le domaine Kerberos IdM. Pour cela, vous devez déployer l'hôte en tant que client IdM. Pour plus de détails, voir Installation d'un client de gestion d'identité à l'aide d'un playbook Ansible.

Verification steps

1. Connectez-vous à votre serveur IdM en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
```

2. Entrez la commande **ipa host-show** et indiquez le nom de l'hôte :

```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Description: Example host
  Locality: Lab
  Location: Lab
  Platform: Lenovo T61
```

```
Operating system: CentOS 7
Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
MAC address: 08:00:27:E3:B1:2D, 52:54:00:BD:97:1E
Password: False
Keytab: False
Managed by: host01.idm.example.com
```

La sortie confirme que **host01.idm.example.com** existe dans IdM.

17.3. ASSURER LA PRÉSENCE DE PLUSIEURS ENTRÉES D'HÔTES IDM AVEC DES MOTS DE PASSE ALÉATOIRES À L'AIDE DES PLAYBOOKS ANSIBLE

Le module **ipahost** permet à l'administrateur système de s'assurer de la présence ou de l'absence d'entrées d'hôtes multiples dans IdM en utilisant une seule tâche Ansible. Cette section décrit comment assurer la présence de plusieurs entrées d'hôtes qui ne sont définies que par leur **fully-qualified domain names** (FQDN). L'exécution du playbook Ansible génère des mots de passe aléatoires pour les hôtes.

NOTE

Sans Ansible, les entrées d'hôtes sont créées dans IdM à l'aide de la commande **ipa host-add**. Le résultat de l'ajout d'un hôte à IdM est l'état de l'hôte présent dans IdM. En raison de la dépendance d'Ansible à l'égard de l'idempotence, pour ajouter un hôte à IdM à l'aide d'Ansible, vous devez créer un playbook dans lequel vous définissez l'état de l'hôte comme étant présent : **state: present**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

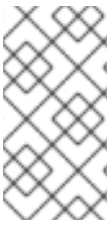
2. Créez un fichier Ansible playbook avec les **fully-qualified domain name** (FQDN) des hôtes dont vous voulez assurer la présence dans IdM. Pour que le playbook Ansible génère un mot de passe aléatoire pour chaque hôte même si l'hôte existe déjà dans IdM et que **update_password**

est limité à **on_create**, ajoutez les options **random: yes** et **force: yes**. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple du fichier Markdown **/usr/share/doc/ansible-freeipa/README-host.md**:

```
---
- name: Ensure hosts with random password
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Hosts host01.idm.example.com and host02.idm.example.com present with random
passwords
    ipahost:
      ipaadmin_password: "{{ ipaadmin_password }}"
      hosts:
      - name: host01.idm.example.com
        random: yes
        force: yes
      - name: host02.idm.example.com
        random: yes
        force: yes
    register: ipahost
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-hosts-
are-present.yml
[...]
TASK [Hosts host01.idm.example.com and host02.idm.example.com present with random
passwords]
changed: [r8server.idm.example.com] => {"changed": true, "host":
{"host01.idm.example.com": {"randompassword": "0HoIRvjUdH0Ycbf6uYdWTxH"},
"host02.idm.example.com": {"randompassword": "5VdLgrf3wvojmACdHC3uA3s"}}}
```



NOTE

Pour déployer les hôtes en tant que clients IdM à l'aide de mots de passe aléatoires à usage unique (OTP), voir Options d'autorisation pour l'inscription d'un client IdM à l'aide d'un playbook Ansible ou Installation d'un client à l'aide d'un mot de passe à usage unique : installation interactive.

Verification steps

1. Connectez-vous à votre serveur IdM en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
```

2. Entrez la commande **ipa host-show** et indiquez le nom de l'un des hôtes :

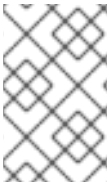
```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
```

```
Password: True
Keytab: False
Managed by: host01.idm.example.com
```

La sortie confirme que **host01.idm.example.com** existe dans IdM avec un mot de passe aléatoire.

17.4. ASSURER LA PRÉSENCE D'UNE ENTRÉE D'HÔTE IDM AVEC PLUSIEURS ADRESSES IP EN UTILISANT LES PLAYBOOKS ANSIBLE

Cette section décrit comment assurer la présence d'une entrée d'hôte dans la gestion des identités (IdM) à l'aide des playbooks Ansible. L'entrée d'hôte est définie par son **fully-qualified domain name** (FQDN) et ses multiples adresses IP.



NOTE

Contrairement à l'utilitaire **ipa host**, le module Ansible **ipahost** permet de s'assurer de la présence ou de l'absence de plusieurs adresses IPv4 et IPv6 pour un hôte. La commande **ipa host-mod** ne peut pas gérer les adresses IP.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

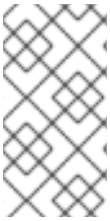
2. Créez un fichier playbook Ansible. Spécifiez, en tant que **name** de la variable **ipahost**, le **fully-qualified domain name** (FQDN) de l'hôte dont vous voulez assurer la présence dans l'IdM. Spécifiez chacune des multiples valeurs IPv4 et IPv6 **ip_address** sur une ligne séparée en utilisant la syntaxe **- *ip_address*** pour spécifier chacune des valeurs IPv4 et IPv6 sur une ligne séparée. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/host/host-member-ipaddresses-present.yml**. Vous pouvez également inclure des informations supplémentaires :

```
---
- name: Host member IP addresses present
  hosts: ipaserver
```

```
vars_files:
- /home/user_name/MyPlaybooks/secret.yml
tasks:
- name: Ensure host101.example.com IP addresses present
  ipahost:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: host01.idm.example.com
    ip_address:
    - 192.168.0.123
    - fe80::20c:29ff:fe02:a1b3
    - 192.168.0.124
    - fe80::20c:29ff:fe02:a1b4
    force: yes
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-host-
with-multiple-IP-addreses-is-present.yml
```



NOTE

La procédure crée une entrée d'hôte dans le serveur LDAP IdM mais n'inscrit pas l'hôte dans le domaine Kerberos IdM. Pour cela, vous devez déployer l'hôte en tant que client IdM. Pour plus de détails, voir Installation d'un client de gestion d'identité à l'aide d'un playbook Ansible.

Verification steps

1. Connectez-vous à votre serveur IdM en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
```

2. Entrez la commande **ipa host-show** et indiquez le nom de l'hôte :

```
$ ipa host-show host01.idm.example.com
  Principal name: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Principal alias: host/host01.idm.example.com@IDM.EXAMPLE.COM
  Password: False
  Keytab: False
  Managed by: host01.idm.example.com
```

Le résultat confirme que **host01.idm.example.com** existe dans IdM.

3. Pour vérifier que les adresses IP multiples de l'hôte existent dans les enregistrements DNS IdM, entrez la commande **ipa dnsrecord-show** et spécifiez les informations suivantes :

- Nom du domaine IdM
- The name of the host

```
$ ipa dnsrecord-show idm.example.com host01
```

```
[...]
  Record name: host01
  A record: 192.168.0.123, 192.168.0.124
  AAAA record: fe80::20c:29ff:fe02:a1b3, fe80::20c:29ff:fe02:a1b4
```

La sortie confirme que toutes les adresses IPv4 et IPv6 spécifiées dans le manuel de jeu sont correctement associées à l'entrée d'hôte **host01.idm.example.com**.

17.5. S'ASSURER DE L'ABSENCE D'UNE ENTRÉE D'HÔTE IDM À L'AIDE DES PLAYBOOKS ANSIBLE

Cette section décrit comment garantir l'absence d'entrées d'hôtes dans la gestion des identités (IdM) à l'aide des playbooks Ansible.

Conditions préalables

- Informations d'identification de l'administrateur IdM

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créez un fichier playbook Ansible avec le **fully-qualified domain name** (FQDN) de l'hôte dont vous voulez garantir l'absence de l'IdM. Si votre domaine IdM a un DNS intégré, utilisez l'option **updatedns: yes** pour supprimer du DNS les enregistrements associés à l'hôte, quels qu'ils soient.
   Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/host/delete-host.yml**:

```
---
- name: Host absent
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Host host01.idm.example.com absent
    ipahost:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: host01.idm.example.com
      updatedns: yes
      state: absent
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-host-
absent.yml
```

NOTE

La procédure aboutit à :

- L'hôte n'est pas présent dans le royaume IdM Kerberos.
- L'entrée de l'hôte n'est pas présente dans le serveur LDAP IdM.

Pour supprimer la configuration IdM spécifique des services système, tels que System Security Services Daemon (SSSD), de l'hôte client lui-même, vous devez exécuter la commande **ipa-client-install --uninstall** sur le client. Pour plus de détails, voir Désinstallation d'un client IdM .

Verification steps

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Afficher des informations sur *host01.idm.example.com*:

```
$ ipa host-show host01.idm.example.com
ipa: ERROR: host01.idm.example.com: host not found
```

Le résultat confirme que l'hôte n'existe pas dans IdM.

17.6. RESSOURCES SUPPLÉMENTAIRES

- Voir le fichier Markdown de **/usr/share/doc/ansible-freeipa/README-host.md**.
- Voir les playbooks supplémentaires dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/host**.

CHAPITRE 18. GÉRER LES GROUPES D'HÔTES À L'AIDE DES PLAYBOOKS ANSIBLE

Ce chapitre présente les groupes d'hôtes dans la gestion des identités (IdM) et décrit l'utilisation d'Ansible pour effectuer les opérations suivantes impliquant des groupes d'hôtes dans la gestion des identités (IdM) :

- Groupes d'accueil dans l'IdM
- Assurer la présence de groupes d'hôtes IdM
- Assurer la présence d'hôtes dans les groupes d'hôtes IdM
- Imbrication des groupes d'hôtes IdM
- Assurer la présence des gestionnaires membres dans les groupes d'accueil de l'IdM
- Garantir l'absence d'hôtes dans les groupes d'hôtes IdM
- Garantir l'absence de groupes d'hôtes imbriqués dans les groupes d'hôtes IdM
- S'assurer de l'absence de membres gestionnaires dans les groupes d'accueil de l'IdM

18.1. GROUPES D'ACCUEIL DANS L'IDM

Les groupes d'hôtes IdM peuvent être utilisés pour centraliser le contrôle des tâches de gestion importantes, en particulier le contrôle d'accès.

Définition des groupes d'accueil

Un groupe d'hôtes est une entité qui contient un ensemble d'hôtes IdM avec des règles de contrôle d'accès communes et d'autres caractéristiques. Par exemple, vous pouvez définir des groupes d'hôtes en fonction des départements de l'entreprise, des emplacements physiques ou des exigences en matière de contrôle d'accès.

Un groupe d'hôtes dans IdM peut comprendre

- Serveurs et clients IdM
- Autres groupes d'accueil IdM

Groupes d'hôtes créés par défaut

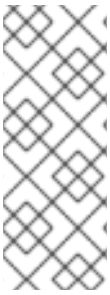
Par défaut, le serveur IdM crée le groupe d'hôtes **ipaservers** pour tous les hôtes du serveur IdM.

Membres directs et indirects du groupe

Les attributs de groupe dans IdM s'appliquent à la fois aux membres directs et indirects : lorsque le groupe d'hôtes B est membre du groupe d'hôtes A, tous les membres du groupe d'hôtes B sont considérés comme des membres indirects du groupe d'hôtes A.

18.2. ASSURER LA PRÉSENCE DES GROUPES D'HÔTES IDM À L'AIDE DES PLAYBOOKS ANSIBLE

Cette section décrit comment assurer la présence de groupes d'hôtes dans la gestion des identités (IdM) à l'aide des playbooks Ansible.



NOTE

Sans Ansible, les entrées de groupes d'hôtes sont créées dans IdM à l'aide de la commande **ipa hostgroup-add**. Le résultat de l'ajout d'un groupe d'hôtes à IdM est l'état du groupe d'hôtes présent dans IdM. En raison de la dépendance d'Ansible à l'égard de l'idempotence, pour ajouter un groupe d'hôtes à IdM à l'aide d'Ansible, vous devez créer un playbook dans lequel vous définissez l'état du groupe d'hôtes comme étant présent : **state: present**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver** avec la liste des serveurs IdM à cibler :

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

2. Créez un fichier playbook Ansible avec les informations nécessaires sur le groupe d'hôtes. Par exemple, pour garantir la présence d'un groupe d'hôtes nommé **databases**, spécifiez **name: databases** dans la tâche **- ipahostgroup**. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-present.yml**.

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure host-group databases is present
  - ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      state: present
```

Dans le playbook, **state: present** signifie une demande d'ajout du groupe d'hôtes à IdM, à moins qu'il n'y existe déjà.

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-
hostgroup-is-present.yml
```

Verification steps

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Demander un ticket Kerberos pour l'administrateur :

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. Affichez les informations sur le groupe d'hôtes dont vous voulez assurer la présence dans l'IdM :

```
$ ipa hostgroup-show databases
  Host-group: databases
```

Le groupe d'hôtes **databases** existe dans IdM.

18.3. ASSURER LA PRÉSENCE D'HÔTES DANS LES GROUPES D'HÔTES IDM À L'AIDE DES PLAYBOOKS ANSIBLE

Cette section décrit comment assurer la présence des hôtes dans les groupes d'hôtes dans la gestion des identités (IdM) à l'aide des playbooks Ansible.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Les hôtes que vous souhaitez référencer dans votre livre de programmation Ansible existent dans IdM. Pour plus de détails, voir Assurer la présence d'une entrée d'hôte IdM à l'aide des carnets de commande Ansible.

- Les groupes d'hôtes que vous avez référencés dans le fichier du livre de jeu Ansible ont été ajoutés à l'IdM. Pour plus de détails, voir Assurer la présence des groupes d'hôtes IdM à l'aide des playbooks Ansible.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver** avec la liste des serveurs IdM à cibler :

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

2. Créez un fichier playbook Ansible avec les informations nécessaires sur l'hôte. Spécifiez le nom du groupe d'hôtes à l'aide du paramètre **name** de la variable **ipahostgroup**. Spécifiez le nom de l'hôte à l'aide du paramètre **host** de la variable **ipahostgroup**. Pour simplifier cette étape, vous pouvez copier et modifier les exemples du fichier **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-present-in-hostgroup.yml**:

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure host-group databases is present
  - ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      host:
      - db.idm.example.com
      action: member
```

Ce livre de jeu ajoute l'hôte **db.idm.example.com** au groupe d'hôtes **databases**. La ligne **action: member** indique que lors de l'exécution de la séquence, aucune tentative n'est faite pour ajouter le groupe **databases** lui-même. Au lieu de cela, seule une tentative d'ajout de **db.idm.example.com** à **databases** est effectuée.

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-hosts-
or-hostgroups-are-present-in-hostgroup.yml
```

Verification steps

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Demander un ticket Kerberos pour l'administrateur :

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. Afficher des informations sur un groupe d'hôtes pour savoir quels sont les hôtes présents dans ce groupe :

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member hosts: db.idm.example.com
```

L'hôte **db.idm.example.com** est présent en tant que membre du groupe d'hôtes **databases**.

18.4. IMBRICATION DES GROUPES D'HÔTES IDM À L'AIDE DES PLAYBOOKS ANSIBLE

Cette section décrit comment assurer la présence de groupes d'hôtes imbriqués dans les groupes d'hôtes de gestion des identités (IdM) à l'aide des playbooks Ansible.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Les groupes d'hôtes auxquels vous faites référence dans le fichier du livre de jeu Ansible existent dans IdM. Pour plus de détails, voir Assurer la présence des groupes d'hôtes IdM à l'aide des playbooks Ansible.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver** avec la liste des serveurs IdM à cibler :

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

2. Créez un fichier playbook Ansible avec les informations nécessaires sur les groupes d'hôtes. Pour s'assurer qu'un groupe d'hôtes imbriqué *A* existe dans un groupe d'hôtes *B*: dans le manuel de jeu Ansible, spécifiez, parmi les variables **- ipahostgroup**, le nom du groupe d'hôtes *B* à l'aide de la variable **name**. Spécifiez le nom du groupe d'hôtes imbriqué *A* avec la variable **hostgroup**. Pour simplifier cette étape, vous pouvez copier et modifier les exemples dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-present-in-hostgroup.yml**:

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure hosts and hostgroups are present in existing databases hostgroup
  - ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      hostgroup:
      - mysql-server
      - oracle-server
      action: member
```

Ce playbook Ansible assure la présence des groupes d'hôtes **myqsl-server** et **oracle-server** dans le groupe d'hôtes **databases**. La ligne **action: member** indique que lorsque le playbook est exécuté, aucune tentative n'est faite pour ajouter le groupe **databases** lui-même à l'IdM.

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-hosts-
or-hostgroups-are-present-in-hostgroup.yml
```

Verification steps

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Demander un ticket Kerberos pour l'administrateur :

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. Affiche des informations sur le groupe d'hôtes dans lequel se trouvent des groupes d'hôtes imbriqués :

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member hosts: db.idm.example.com
  Member host-groups: mysql-server, oracle-server
```

Les groupes d'hôtes **mysql-server** et **oracle-server** existent dans le groupe d'hôtes **databases**.

18.5. ASSURER LA PRÉSENCE DE GESTIONNAIRES MEMBRES DANS LES GROUPES D'HÔTES IDM À L'AIDE DES PLAYBOOKS ANSIBLE

La procédure suivante décrit comment assurer la présence des gestionnaires membres dans les hôtes IdM et les groupes d'hôtes à l'aide d'un livre de jeu Ansible.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Vous devez avoir le nom de l'hôte ou du groupe d'hôtes que vous ajoutez en tant que membres managers et le nom du groupe d'hôtes que vous voulez qu'ils gèrent.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créer un fichier Ansible playbook avec les informations nécessaires à la gestion des hôtes et des membres du groupe d'hôtes :

```
---

- name: Playbook to handle host group membership management
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure member manager user example_member is present for group_name
      ipahostgroup:
        ipaadmin_password: "{{ ipaadmin_password }}"
        name: group_name
        membermanager_user: example_member

  - name: Ensure member manager group project_admins is present for group_name
      ipahostgroup:
        ipaadmin_password: "{{ ipaadmin_password }}"
        name: group_name
        membermanager_group: project_admins
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/add-member-
managers-host-groups.yml
```

Verification steps

Vous pouvez vérifier si le groupe **group_name** contient **example_member** et **project_admins** en tant que gestionnaires membres en utilisant la commande **ipa group-show**:

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Afficher des informations sur *testhostgroup*:

```
ipaserver]$ ipa hostgroup-show group_name
  Host-group: group_name
  Member hosts: server.idm.example.com
  Member host-groups: testhostgroup2
  Membership managed by groups: project_admins
  Membership managed by users: example_member
```

Ressources supplémentaires

- Voir **ipa hostgroup-add-member-manager --help**.
- Voir la page de manuel **ipa**.

18.6. GARANTIR L'ABSENCE D'HÔTES DANS LES GROUPES D'HÔTES IDM À L'AIDE DES PLAYBOOKS ANSIBLE

Cette section décrit comment garantir l'absence d'hôtes dans les groupes d'hôtes de la gestion des identités (IdM) à l'aide des playbooks Ansible.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.

- Les hôtes que vous souhaitez référencer dans votre livre de programmation Ansible existent dans IdM. Pour plus de détails, voir Assurer la présence d'une entrée d'hôte IdM à l'aide des carnets de commande Ansible.

- Les groupes d'hôtes auxquels vous faites référence dans le fichier du livre de jeu Ansible existent dans IdM. Pour plus de détails, voir Assurer la présence des groupes d'hôtes IdM à l'aide des playbooks Ansible.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver** avec la liste des serveurs IdM à cibler :

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

2. Créez un fichier playbook Ansible avec les informations nécessaires sur l'hôte et le groupe d'hôtes. Spécifiez le nom du groupe d'hôtes en utilisant le paramètre **name** de la variable **ipahostgroup**. Spécifiez le nom de l'hôte dont vous voulez garantir l'absence dans le groupe d'hôtes en utilisant le paramètre **host** de la variable **ipahostgroup**. Pour simplifier cette étape, vous pouvez copier et modifier les exemples du fichier **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-absent-in-hostgroup.yml**:

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure host-group databases is absent
  - ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      host:
      - db.idm.example.com
      action: member
      state: absent
```

   Ce livre de jeu garantit l'absence de l'hôte **db.idm.example.com** du groupe d'hôtes **databases**. La ligne **action: member** indique que lors de l'exécution du livre de jeu, aucune tentative n'est faite pour supprimer le groupe **databases** lui-même.

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-hosts-
or-hostgroups-are-absent-in-hostgroup.yml
```

Verification steps

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Demander un ticket Kerberos pour l'administrateur :

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. Affiche des informations sur le groupe d'hôtes et les hôtes qu'il contient :

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member host-groups: mysql-server, oracle-server
```

L'hôte **db.idm.example.com** n'existe pas dans le groupe d'hôtes **databases**.

18.7. GARANTIR L'ABSENCE DE GROUPES D'HÔTES IMBRIQUÉS À PARTIR DES GROUPES D'HÔTES IDM À L'AIDE DES PLAYBOOKS ANSIBLE

Cette section décrit comment garantir l'absence de groupes d'hôtes imbriqués dans les groupes d'hôtes externes dans la gestion des identités (IdM) à l'aide des playbooks Ansible.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Les groupes d'hôtes auxquels vous faites référence dans le fichier du livre de jeu Ansible existent dans IdM. Pour plus de détails, voir Assurer la présence des groupes d'hôtes IdM à l'aide des playbooks Ansible.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver** avec la liste des serveurs IdM à cibler :

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

2. Créez un fichier playbook Ansible avec les informations nécessaires sur le groupe d'hôtes. Spécifiez, parmi les variables **- ipahostgroup**, le nom du groupe d'hôtes extérieur à l'aide de la variable **name**. Spécifiez le nom du groupe d'hôtes imbriqué à l'aide de la variable **hostgroup**. Pour simplifier cette étape, vous pouvez copier et modifier les exemples dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-absent-in-hostgroup.yml**:

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure hosts and hostgroups are absent in existing databases hostgroup
  - ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: databases
      hostgroup:
      - mysql-server
      - oracle-server
      action: member
      state: absent
```

Ce playbook s'assure que les groupes d'hôtes **mysql-server** et **oracle-server** sont absents du groupe d'hôtes **databases**. La ligne **action: member** indique que lorsque le playbook est exécuté, aucune tentative n'est faite pour s'assurer que le groupe **databases** lui-même est supprimé de l'IdM.

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-hosts-
or-hostgroups-are-absent-in-hostgroup.yml
```

Verification steps

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Demander un ticket Kerberos pour l'administrateur :

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

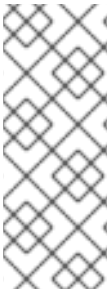
3. Afficher des informations sur le groupe d'hôtes à partir duquel les groupes d'hôtes imbriqués doivent être absents :

```
$ ipa hostgroup-show databases
  Host-group: databases
```

La sortie confirme que les groupes d'hôtes imbriqués **mysql-server** et **oracle-server** sont absents du groupe d'hôtes extérieur **databases**.

18.8. GARANTIR L'ABSENCE DE GROUPES D'HÔTES IDM À L'AIDE DE PLAYBOOKS ANSIBLE

Cette section décrit comment garantir l'absence de groupes d'hôtes dans la gestion des identités (IdM) à l'aide des playbooks Ansible.



NOTE

Sans Ansible, les entrées de groupes d'hôtes sont supprimées de l'IdM à l'aide de la commande **ipa hostgroup-del**. Le résultat de la suppression d'un groupe d'hôtes de l'IdM est l'état du groupe d'hôtes absent de l'IdM. En raison de la dépendance d'Ansible à l'idempotence, pour supprimer un groupe d'hôtes d'IdM à l'aide d'Ansible, vous devez créer un playbook dans lequel vous définissez l'état du groupe d'hôtes comme étant absent : **state: absent**

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver** avec la liste des serveurs IdM à cibler :

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

2. Créez un fichier playbook Ansible avec les informations nécessaires sur le groupe d'hôtes. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-absent.yml**.

```
---
- name: Playbook to handle hostgroups
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
```

```
- Ensure host-group databases is absent
  ipahostgroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: databases
    state: absent
```

Ce playbook garantit l'absence du groupe d'hôtes **databases** dans l'IdM. Le **state: absent** signifie une demande de suppression du groupe d'hôtes de l'IdM, à moins qu'il ne soit déjà supprimé.

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-
hostgroup-is-absent.yml
```

Verification steps

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Demander un ticket Kerberos pour l'administrateur :

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

3. Afficher des informations sur le groupe d'hôtes dont vous avez assuré l'absence :

```
$ ipa hostgroup-show databases
ipa: ERROR: databases: host group not found
```

Le groupe d'hôtes **databases** n'existe pas dans IdM.

18.9. ASSURER L'ABSENCE DES GESTIONNAIRES DE MEMBRES DES GROUPES HÔTES IDM À L'AIDE DES PLAYBOOKS ANSIBLE

La procédure suivante décrit comment garantir l'absence de gestionnaires membres dans les hôtes IdM et les groupes d'hôtes à l'aide d'un playbook Ansible.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.

- Vous devez disposer du nom de l'utilisateur ou du groupe d'utilisateurs que vous supprimez en tant que membres gestionnaires et du nom du groupe d'hôtes qu'ils gèrent.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créer un fichier Ansible playbook avec les informations nécessaires à la gestion des hôtes et des membres du groupe d'hôtes :

```
---

- name: Playbook to handle host group membership management
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure member manager host and host group members are absent for
group_name
    ipahostgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: group_name
      membermanager_user: example_member
      membermanager_group: project_admins
      action: member
      state: absent
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-
member-managers-host-groups-are-absent.yml
```

Verification steps

Vous pouvez vérifier si le groupe **group_name** ne contient pas **example_member** ou **project_admins** en tant que gestionnaires membres en utilisant la commande **ipa group-show**:

1. Connectez-vous à **ipaserver** en tant qu'administrateur :

```
$ ssh admin@server.idm.example.com
Password:
[admin@server /]$
```

2. Afficher des informations sur *testhostgroup*:

```
ipaserver]$ ipa hostgroup-show group_name
  Host-group: group_name
  Member hosts: server.idm.example.com
  Member host-groups: testhostgroup2
```

Ressources supplémentaires

- Voir **ipa hostgroup-add-member-manager --help**.
- Voir la page de manuel **ipa**.

CHAPITRE 19. DÉFINITION DES POLITIQUES DE MOT DE PASSE DE L'IDM

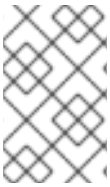
Ce chapitre décrit les politiques de mot de passe de la gestion des identités (IdM) et explique comment ajouter une nouvelle politique de mot de passe dans l'IdM à l'aide d'un playbook Ansible.

19.1. QU'EST-CE QU'UNE POLITIQUE DE MOT DE PASSE ?

Une politique de mot de passe est un ensemble de règles auxquelles les mots de passe doivent répondre. Par exemple, une politique de mot de passe peut définir la longueur minimale et la durée maximale d'un mot de passe. Tous les utilisateurs concernés par cette politique sont tenus de définir un mot de passe suffisamment long et de le modifier assez fréquemment pour satisfaire aux conditions spécifiées. De cette manière, les politiques de mot de passe contribuent à réduire le risque que quelqu'un découvre et utilise à mauvais escient le mot de passe d'un utilisateur.

19.2. POLITIQUES EN MATIÈRE DE MOTS DE PASSE DANS L'IDM

Les mots de passe sont le moyen le plus courant pour les utilisateurs de la gestion de l'identité (IdM) de s'authentifier auprès du domaine Kerberos IdM. Les stratégies de mot de passe définissent les exigences auxquelles doivent répondre les mots de passe des utilisateurs IdM.



NOTE

La politique de mot de passe de l'IdM est définie dans l'annuaire LDAP sous-jacent, mais c'est le centre de distribution de clés Kerberos (KDC) qui applique la politique de mot de passe.

Les attributs de la politique de mot de passe énumèrent les attributs que vous pouvez utiliser pour définir une politique de mot de passe dans IdM.

Tableau 19.1. Attributs de la politique relative aux mots de passe

Attribut	Explication	Exemple :
Durée de vie maximale	Durée maximale en jours pendant laquelle un mot de passe est valide avant que l'utilisateur ne doive le réinitialiser. La valeur par défaut est de 90 jours. Notez que si l'attribut est fixé à 0, le mot de passe n'expire jamais.	Durée de vie maximale = 180 Les mots de passe des utilisateurs ne sont valables que pendant 180 jours. Après cette période, l'IdM invite les utilisateurs à les modifier.
Durée de vie minimale	Le temps minimum en heures qui doit s'écouler entre deux opérations de changement de mot de passe.	Durée de vie minimale = 1 Après avoir modifié leur mot de passe, les utilisateurs doivent attendre au moins une heure avant de le modifier à nouveau.

Attribut	Explication	Exemple :
Taille de l'histoire	Le nombre de mots de passe précédents qui sont stockés. Un utilisateur ne peut pas réutiliser un mot de passe de son historique de mots de passe, mais peut réutiliser d'anciens mots de passe qui ne sont pas stockés.	Taille de l'historique = 0 Dans ce cas, l'historique des mots de passe est vide et les utilisateurs peuvent réutiliser n'importe lequel de leurs mots de passe précédents.
Classes de personnages	Le nombre de classes de caractères différentes que l'utilisateur doit utiliser dans le mot de passe. Les classes de caractères sont les suivantes : * Caractères majuscules * Caractères minuscules * Chiffres * Caractères spéciaux, tels que virgule (,), point (.), astérisque (*) * Autres caractères UTF-8 L'utilisation d'un caractère trois fois ou plus à la suite diminue la classe du caractère d'une unité. Par exemple : * **Secret1** a 3 classes de caractères : majuscules, minuscules, chiffres * **Secret111** a 2 classes de caractères : majuscules, minuscules, chiffres, et une pénalité de -1 pour l'utilisation répétée de **1**	Classes de caractères = 0 Le nombre de classes requis par défaut est de 0. Pour configurer ce nombre, exécutez la commande **ipa pwpolicy-mod** avec l'option **--minclasses**. Voir également la note importante sous ce tableau.
Longueur minimale	Nombre minimum de caractères dans un mot de passe. Si l'une des options de politique de mot de passe supplémentaires est activée, la longueur minimale des mots de passe est de 6 caractères.	Longueur minimale = 8 Les utilisateurs ne peuvent pas utiliser de mots de passe de moins de 8 caractères.
Défaillances maximales	Nombre maximal de tentatives de connexion échouées avant que l'IdM ne verrouille le compte de l'utilisateur.	Nombre maximal d'échecs = 6 L'IdM verrouille le compte de l'utilisateur lorsque celui-ci saisit un mot de passe erroné sept fois de suite.

Attribut	Explication	Exemple :
Intervalle de réinitialisation des défaillances	Délai en secondes après lequel l'IdM réinitialise le nombre actuel de tentatives de connexion infructueuses.	Intervalle de réinitialisation des défaillances = 60 Si l'utilisateur attend plus d'une minute après le nombre d'échecs de connexion défini à l'adresse **Max failures**, il peut tenter de se connecter à nouveau sans risquer de voir son compte bloqué.
Durée du verrouillage	Durée en secondes pendant laquelle le compte de l'utilisateur est verrouillé après le nombre de tentatives de connexion infructueuses défini à l'adresse **Max failures**.	Durée du verrouillage = 600 Les utilisateurs dont le compte est verrouillé ne peuvent pas se connecter pendant 10 minutes.

IMPORTANT

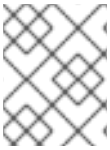
Utilisez l'alphabet anglais et les symboles courants pour les classes de caractères requises si vous disposez d'un ensemble diversifié de matériel qui peut ne pas avoir accès aux caractères et symboles internationaux. Pour plus d'informations sur les politiques de classes de caractères dans les mots de passe, voir Quels sont les caractères valides dans un mot de passe ? dans la base de connaissances de Red Hat.

19.3. ASSURER LA PRÉSENCE D'UNE POLITIQUE DE MOT DE PASSE DANS IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

Cette section décrit comment assurer la présence d'une politique de mot de passe dans la gestion des identités (IdM) à l'aide d'un playbook Ansible.

Dans la politique de mot de passe **global_policy** par défaut dans IdM, le nombre de classes de caractères différents dans le mot de passe est fixé à 0. La taille de l'historique est également fixée à 0.

Effectuez cette procédure pour appliquer une politique de mot de passe plus stricte pour un groupe IdM à l'aide d'un playbook Ansible.



NOTE

Vous ne pouvez définir une politique de mot de passe que pour un groupe IdM. Vous ne pouvez pas définir une politique de mot de passe pour un utilisateur individuel.

Conditions préalables

- 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**.

- Vous connaissez le mot de passe de l'administrateur IdM.
- Le groupe pour lequel vous vous assurez de la présence d'une politique de mot de passe existe dans IdM.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez le **FQDN** de votre serveur IdM dans la section **[ipaserver]**:

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

2. Créez votre fichier playbook Ansible qui définit la politique de mot de passe dont vous voulez assurer la présence. Pour simplifier cette étape, copiez et modifiez l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/pwpolicy/pwpolicy_present.yml**:

```
---
- name: Tests
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure presence of pwpolicy for group ops
    ipapwpolicy:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: ops
      minlife: 7
      maxlife: 49
      history: 5
      priority: 1
      lockouttime: 300
      minlength: 8
      minclasses: 4
      maxfail: 3
      failinterval: 5
```

   Pour plus de détails sur la signification des différentes variables, voir Attributs de la politique de mot de passe.

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file
path_to_playbooks_directory_/new_pwpolicy_present.yml
```

Vous avez utilisé avec succès un playbook Ansible pour vous assurer qu'une politique de mot de passe pour le groupe **ops** est présente dans IdM.

IMPORTANT

La priorité de la politique de mot de passe **ops** est fixée à *1*, alors que la politique de mot de passe **global_policy** n'a pas de priorité définie. Pour cette raison, la politique **ops** remplace automatiquement **global_policy** pour le groupe **ops** et est appliquée immédiatement.

global_policy sert de stratégie de repli lorsqu'aucune stratégie de groupe n'est définie pour un utilisateur, et ne peut jamais avoir la priorité sur une stratégie de groupe.

Ressources supplémentaires

- Voir le fichier **README-pwpolicy.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les priorités politiques du mot de passe .

19.4. OPTIONS SUPPLÉMENTAIRES DE POLITIQUE DE MOT DE PASSE DANS IDM

En tant qu'administrateur Identity Management (IdM), vous pouvez renforcer les exigences en matière de mot de passe par défaut en activant des options de politique de mot de passe supplémentaires basées sur l'ensemble des fonctionnalités de **libpwquality**. Les options supplémentaires de stratégie de mot de passe sont les suivantes :

--maxrepeat

Spécifie le nombre maximal acceptable de caractères consécutifs identiques dans le nouveau mot de passe.

--maxsequence

Spécifie la longueur maximale des séquences de caractères monotones dans le nouveau mot de passe. Des exemples d'une telle séquence sont **12345** ou **fedcb**. La plupart de ces mots de passe ne passeront pas le contrôle de simplicité.

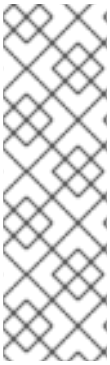
--dictcheck

Si non nul, vérifie si le mot de passe, avec d'éventuelles modifications, correspond à un mot du dictionnaire. Actuellement, **libpwquality** effectue la vérification du dictionnaire à l'aide de la bibliothèque **cracklib**.

--usercheck

S'il est différent de zéro, il vérifie si le mot de passe, avec d'éventuelles modifications, contient le nom de l'utilisateur sous une forme ou une autre. Cette vérification n'est pas effectuée pour les noms d'utilisateur de moins de 3 caractères.

Vous ne pouvez pas appliquer les options supplémentaires de politique de mot de passe aux mots de passe existants. Si vous appliquez l'une des options supplémentaires, l'IdM définit automatiquement l'option **--minlength**, le nombre minimum de caractères dans un mot de passe, à **6** caractères.



NOTE

Dans un environnement mixte avec des serveurs RHEL 7, RHEL 8 et RHEL 9, vous pouvez appliquer les paramètres supplémentaires de la stratégie de mot de passe uniquement sur les serveurs fonctionnant sous RHEL 8.4 ou une version ultérieure. Si un utilisateur est connecté à un client IdM et que ce dernier communique avec un serveur IdM fonctionnant sous RHEL 8.3 ou une version antérieure, les nouvelles exigences en matière de stratégie de mot de passe définies par l'administrateur système ne seront pas appliquées. Pour garantir un comportement cohérent, mettez à niveau ou mettez à jour tous les serveurs vers RHEL 8.4 ou une version ultérieure.

Ressources complémentaires :

- Appliquer des politiques de mot de passe supplémentaires à un groupe IdM
- **pwquality(3)** page de manuel

19.5. APPLIQUER DES OPTIONS SUPPLÉMENTAIRES DE POLITIQUE DE MOT DE PASSE À UN GROUPE IDM

Cette section décrit comment appliquer des options supplémentaires de politique de mot de passe dans la gestion des identités (IdM). L'exemple décrit comment renforcer la politique de mot de passe pour le groupe **managers** en s'assurant que les nouveaux mots de passe ne contiennent pas les noms d'utilisateur respectifs des utilisateurs et que les mots de passe ne contiennent pas plus de deux caractères identiques à la suite.

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Le groupe **managers** existe dans IdM.
- La politique de mot de passe **managers** existe dans IdM.

Procédure

1. Appliquer le contrôle du nom d'utilisateur à tous les nouveaux mots de passe proposés par les utilisateurs du groupe **managers**:

```
$ ipa pwpolicy-mod --usercheck=True managers
```

NOTE

Si vous ne spécifiez pas le nom de la politique de mot de passe, la valeur par défaut **global_policy** est modifiée.

2. Définissez le nombre maximum de caractères identiques consécutifs à 2 dans la politique de mot de passe **managers**:

```
$ ipa pwpolicy-mod --maxrepeat=2 managers
```

Un mot de passe ne sera pas accepté s'il contient plus de 2 caractères identiques consécutifs. Par exemple, la combinaison **eR873mUi111YJQ** est inacceptable parce qu'elle contient trois **1**consécutifs.

Vérification

1. Ajouter un utilisateur test nommé **test_user**:

```
$ ipa user-add test_user
First name: test
Last name: user
----------------------------
Added user "test_user"
----------------------------
```

2. Ajoutez l'utilisateur test au groupe **managers**:
   a. Dans l'interface Web IdM, cliquez sur **Identité → Groupes → Groupes d'utilisateurs**.
   b. Cliquez sur **managers**.
   c. Cliquez sur **Add**.
   d. Dans la page **Add users into user group 'managers'**, vérifiez **test_user**.
   e. Cliquez sur la flèche **>** pour déplacer l'utilisateur dans la colonne **Prospective**.
   f. Cliquez sur **Add**.
3. Réinitialiser le mot de passe de l'utilisateur test :
   a. Aller à **Identité → Utilisateurs**.
   b. Cliquez sur **test_user**.
   c. Dans le menu **Actions**, cliquez sur **Reset Password**.
   d. Entrez un mot de passe temporaire pour l'utilisateur.
4. Sur la ligne de commande, essayez d'obtenir un ticket Kerberos (TGT) pour l'adresse **test_user**:

```
$ kinit test_user
```

   a. Saisissez le mot de passe temporaire.
   b. Le système vous informe que vous devez modifier votre mot de passe. Saisissez un mot de passe qui contient le nom d'utilisateur **test_user**:

```
Password expired. You must change it now.
Enter new password:
Enter it again:
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.
```

NOTE

Kerberos ne dispose pas d'une politique de signalement des erreurs de mot de passe très fine et, dans certains cas, ne fournit pas de raison claire pour laquelle un mot de passe a été rejeté.

c. Le système vous informe que le mot de passe introduit a été rejeté. Introduisez un mot de passe contenant au moins trois caractères identiques successifs :

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

d. Le système vous informe que le mot de passe introduit a été rejeté. Saisissez un mot de passe qui répond aux critères de la politique de mot de passe **managers**:

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

5. Voir le TGT :

```
$ klist
Ticket cache: KCM:0:33945
Default principal: test_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
07/07/2021 12:44:44  07/08/2021 12:44:44
krbtgt@IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
```

La politique de mot de passe **managers** fonctionne désormais correctement pour les utilisateurs du groupe **managers**.

Ressources supplémentaires

- Politiques de mot de passe supplémentaires dans l'IdM

19.6. UTILISATION D'UN PLAYBOOK ANSIBLE POUR APPLIQUER DES OPTIONS DE POLITIQUE DE MOT DE PASSE SUPPLÉMENTAIRES À UN GROUPE IDM

Vous pouvez utiliser un playbook Ansible pour appliquer des options de politique de mot de passe supplémentaires afin de renforcer les exigences de la politique de mot de passe pour un groupe IdM spécifique. Vous pouvez utiliser les options de politique de mot de passe **maxrepeat**, **maxsequence**, **dictcheck** et **usercheck** à cette fin. L'exemple décrit comment définir les exigences suivantes pour le groupe **managers**:

- Les nouveaux mots de passe des utilisateurs ne contiennent pas leurs noms d'utilisateur respectifs.
- Les mots de passe ne contiennent pas plus de deux caractères identiques successifs.
- Les séquences de caractères monotones dans les mots de passe ne doivent pas dépasser 3 caractères. Cela signifie que le système n'accepte pas un mot de passe comportant une séquence telle que **1234** ou **abcd**.

Conditions préalables

- 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.
  - Vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM dans le répertoire ***~/MyPlaybooks/*** dans le répertoire
  - Vous avez stocké votre site **ipaadmin_password** dans le coffre-fort **secret.yml** Ansible.
- Le groupe pour lequel vous vous assurez de la présence d'une politique de mot de passe existe dans IdM.

Procédure

1. Créez votre fichier Ansible playbook **manager_pwpolicy_present.yml** qui définit la politique de mot de passe dont vous voulez assurer la présence. Pour simplifier cette étape, copiez et modifiez l'exemple suivant :

```
---
- name: Tests
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure presence of usercheck and maxrepeat pwpolicy for group managers
    ipapwpolicy:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: managers
      usercheck: True
      maxrepeat: 2
      maxsequence: 3
```

2. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file
path_to_playbooks_directory_/manager_pwpolicy_present.yml
```

Vérification

1. Ajouter un utilisateur test nommé **test_user**:

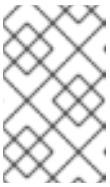
```
$ ipa user-add test_user
First name: test
Last name: user
----------------------------
Added user "test_user"
----------------------------
```

2. Ajoutez l'utilisateur test au groupe **managers**:
   a. Dans l'interface Web IdM, cliquez sur **Identité → Groupes → Groupes d'utilisateurs**.
   b. Cliquez sur **managers**.
   c. Cliquez sur **Add**.
   d. Dans la page **Add users into user group 'managers'**, vérifiez **test_user**.
   e. Cliquez sur la flèche **>** pour déplacer l'utilisateur dans la colonne **Prospective**.
   f. Cliquez sur **Add**.
3. Réinitialiser le mot de passe de l'utilisateur test :
   a. Aller à **Identité → Utilisateurs**.
   b. Cliquez sur **test_user**.
   c. Dans le menu **Actions**, cliquez sur **Reset Password**.
   d. Entrez un mot de passe temporaire pour l'utilisateur.
4. Sur la ligne de commande, essayez d'obtenir un ticket Kerberos (TGT) pour l'adresse **test_user**:

```
$ kinit test_user
```

   a. Saisissez le mot de passe temporaire.
   b. Le système vous informe que vous devez modifier votre mot de passe. Saisissez un mot de passe qui contient le nom d'utilisateur **test_user**:

```
Password expired. You must change it now.
Enter new password:
Enter it again:
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.
```



NOTE

Kerberos ne dispose pas d'une politique de signalement des erreurs de mot de passe très fine et, dans certains cas, ne fournit pas de raison claire pour laquelle un mot de passe a été rejeté.

c. Le système vous informe que le mot de passe introduit a été rejeté. Introduisez un mot de passe contenant au moins trois caractères identiques successifs :

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

d. Le système vous informe que le mot de passe saisi a été rejeté. Introduisez un mot de passe qui contient une séquence de caractères monotone de plus de 3 caractères. Des exemples de telles séquences sont **1234** et **fedc**:

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

e. Le système vous informe que le mot de passe introduit a été rejeté. Saisissez un mot de passe qui répond aux critères de la politique de mot de passe **managers**:

```
Password change rejected: Password not changed.
Unspecified password quality failure while trying to change password.
Please try again.

Enter new password:
Enter it again:
```

5. Vérifiez que vous avez obtenu un TGT, ce qui n'est possible qu'après avoir introduit un mot de passe valide :

```
$ klist
Ticket cache: KCM:0:33945
Default principal: test_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
07/07/2021 12:44:44  07/08/2021 12:44:44
krbtgt@IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
```

Ressources supplémentaires

- Politiques de mot de passe supplémentaires dans l'IdM
- **/usr/share/doc/ansible-freeipa/README-pwpolicy.md**
- **/usr/share/doc/ansible-freeipa/playbooks/pwpolicy**

CHAPITRE 20. ACCORDER UN ACCÈS SUDO À UN UTILISATEUR IDM SUR UN CLIENT IDM

Cette section décrit comment accorder l'accès **sudo** aux utilisateurs dans la gestion des identités.

20.1. ACCÈS SUDO SUR UN CLIENT IDM

Les administrateurs système peuvent accorder l'accès **sudo** pour permettre aux utilisateurs non root d'exécuter des commandes administratives qui sont normalement réservées à l'utilisateur **root**. Par conséquent, lorsque les utilisateurs doivent exécuter une commande administrative normalement réservée à l'utilisateur **root**, ils font précéder cette commande de **sudo**. Après avoir introduit son mot de passe, la commande est exécutée comme s'il s'agissait de l'utilisateur **root**. Pour exécuter une commande **sudo** en tant qu'autre utilisateur ou groupe, tel qu'un compte de service de base de données, vous pouvez configurer une règle *RunAs alias* pour **sudo**.

Si un hôte Red Hat Enterprise Linux (RHEL) 8 est inscrit en tant que client Identity Management (IdM), vous pouvez spécifier les règles **sudo** définissant quels utilisateurs IdM peuvent exécuter quelles commandes sur l'hôte de la manière suivante :

- Localement dans le fichier **/etc/sudoers**
- Au niveau central dans l'IdM

Cette section décrit la création d'un **central sudo rule** pour un client IdM à l'aide de l'interface de ligne de commande (CLI) et de l'interface Web IdM.

Vous pouvez également configurer l'authentification sans mot de passe pour **sudo** à l'aide de l'interface GSSAPI (Generic Security Service Application Programming Interface), le moyen natif pour les systèmes d'exploitation basés sur UNIX d'accéder aux services Kerberos et de les authentifier. Vous pouvez utiliser **pam_sss_gss.so** Pluggable Authentication Module (PAM) pour invoquer l'authentification GSSAPI via le service SSSD, ce qui permet aux utilisateurs de s'authentifier auprès de la commande **sudo** à l'aide d'un ticket Kerberos valide.

Ressources supplémentaires

- Voir Gestion de l'accès sudo.

20.2. ACCORDER UN ACCÈS SUDO À UN UTILISATEUR IDM SUR UN CLIENT IDM À L'AIDE DE LA CLI

Dans la gestion des identités (IdM), vous pouvez accorder l'accès **sudo** pour une commande spécifique à un compte d'utilisateur IdM sur un hôte IdM spécifique. Commencez par ajouter une commande **sudo**, puis créez une règle **sudo** pour une ou plusieurs commandes.

Par exemple, suivez cette procédure pour créer la règle **idm_user_reboot sudo** afin d'autoriser le compte **idm_user** à exécuter la commande **/usr/sbin/reboot** sur la machine **idmclient**.

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Vous avez créé un compte utilisateur pour **idm_user** dans IdM et déverrouillé le compte en créant un mot de passe pour l'utilisateur. Pour plus d'informations sur l'ajout d'un nouvel utilisateur IdM à l'aide de la ligne de commande, voir Ajouter des utilisateurs à l'aide de la ligne

de commande.

- Aucun compte local **idm_user** n'est présent sur l'hôte **idmclient**. L'utilisateur **idm_user** n'est pas répertorié dans le fichier local **/etc/passwd**.

Procédure

1. Récupérer un ticket Kerberos en tant qu'IdM **admin**.

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

2. Ajouter la commande **/usr/sbin/reboot** à la base de données IdM des commandes **sudo**:

```
[root@idmclient ~]# ipa sudocmd-add /usr/sbin/reboot
----------------------------------
Added Sudo Command "/usr/sbin/reboot"
----------------------------------
  Sudo Command: /usr/sbin/reboot
```

3. Créez une règle **sudo** nommée **idm_user_reboot**:

```
[root@idmclient ~]# ipa sudorule-add idm_user_reboot
--------------------------------
Added Sudo Rule "idm_user_reboot"
--------------------------------
  Rule name: idm_user_reboot
  Enabled: TRUE
```

4. Ajoutez la commande **/usr/sbin/reboot** à la règle **idm_user_reboot**:

```
[root@idmclient ~]# ipa sudorule-add-allow-command idm_user_reboot --sudocmds
'/usr/sbin/reboot'
  Rule name: idm_user_reboot
  Enabled: TRUE
  Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```

5. Appliquer la règle **idm_user_reboot** à l'hôte IdM **idmclient**:

```
[root@idmclient ~]# ipa sudorule-add-host idm_user_reboot --hosts
idmclient.idm.example.com
Rule name: idm_user_reboot
Enabled: TRUE
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```

6. Ajoutez le compte **idm_user** à la règle **idm_user_reboot**:

```
[root@idmclient ~]# ipa sudorule-add-user idm_user_reboot --users idm_user
```

```
Rule name: idm_user_reboot
Enabled: TRUE
Users: idm_user
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```

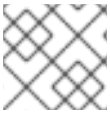
7. Il est possible de définir la validité de la règle **idm_user_reboot**:

   a. Pour définir l'heure à laquelle une règle **sudo** commence à être valide, utilisez la commande **ipa sudorule-mod *sudo_rule_name*** avec l'option **--setattr sudonotbefore=*DATE*** option. La valeur de *DATE* doit suivre le format de **yyyymmddHHMMSSZ**, en spécifiant explicitement les secondes. Par exemple, pour définir le début de la validité de la règle **idm_user_reboot** au 31 décembre 2025 12:34:00, entrez :

   ```
   [root@idmclient ~]# ipa sudorule-mod idm_user_reboot --setattr
   sudonotbefore=20251231123400Z
   ```

   b. Pour définir l'heure à laquelle une règle sudo cesse d'être valide, utilisez l'option **--setattr sudonotafter=DATE**. Par exemple, pour fixer la fin de la validité de la règle **idm_user_reboot** au 31 décembre 2026 12:34:00, entrez :

   ```
   [root@idmclient ~]# ipa sudorule-mod idm_user_reboot --setattr
   sudonotafter=20261231123400Z
   ```



NOTE

La propagation des changements du serveur au client peut prendre quelques minutes.

Verification steps

1. Connectez-vous à l'hôte **idmclient** en tant que compte **idm_user**.

2. Affiche les règles **sudo** que le compte **idm_user** est autorisé à appliquer.

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

User idm_user may run the following commands on idmclient:
    (root) /usr/sbin/reboot
```

3. Redémarrez la machine en utilisant **sudo**. Saisissez le mot de passe de **idm_user** lorsque vous y êtes invité :

```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

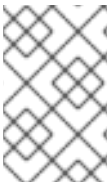
20.3. ACCORDER UN ACCÈS SUDO À UN UTILISATEUR AD SUR UN CLIENT IDM À L'AIDE DE LA CLI

Les administrateurs du système de gestion des identités (IdM) peuvent utiliser les groupes d'utilisateurs IdM pour définir les autorisations d'accès, le contrôle d'accès basé sur l'hôte, les règles **sudo** et d'autres contrôles sur les utilisateurs IdM. Les groupes d'utilisateurs IdM accordent et restreignent l'accès aux ressources du domaine IdM.

Vous pouvez ajouter à la fois Active Directory (AD) *users* et AD *groups* aux groupes d'utilisateurs IdM. Pour ce faire, procédez comme suit

1. Ajouter les utilisateurs ou groupes AD à un groupe IdM externe *non-POSIX*.
2. Ajouter le groupe IdM externe non-POSIX à un groupe IdM *POSIX*.

Vous pouvez ensuite gérer les privilèges des utilisateurs AD en gérant les privilèges du groupe POSIX. Par exemple, vous pouvez accorder l'accès **sudo** pour une commande spécifique à un groupe d'utilisateurs POSIX IdM sur un hôte IdM spécifique.



NOTE

Il est également possible d'ajouter des groupes d'utilisateurs AD aux groupes externes IdM. Cela peut faciliter la définition de politiques pour les utilisateurs Windows, en maintenant la gestion des utilisateurs et des groupes dans le domaine unique d'AD.

IMPORTANT

Ne **not** utilisez pas les ID overrides des utilisateurs AD pour les règles SUDO dans IdM. Les substitutions d'ID des utilisateurs AD ne représentent que les attributs POSIX des utilisateurs AD, et non les utilisateurs AD eux-mêmes.

Vous pouvez ajouter des dérogations d'ID en tant que membres d'un groupe. Toutefois, vous ne pouvez utiliser cette fonctionnalité que pour gérer les ressources IdM dans l'API IdM. La possibilité d'ajouter des dérogations d'ID en tant que membres d'un groupe n'est pas étendue aux environnements POSIX et vous ne pouvez donc pas l'utiliser pour l'appartenance à **sudo** ou à des règles de contrôle d'accès basées sur l'hôte (HBAC).

Cette procédure décrit comment créer la règle **ad_users_reboot sudo** pour accorder à l'utilisateur **administrator@ad-domain.com** AD la permission d'exécuter la commande **/usr/sbin/reboot** sur l'hôte **idmclient** IdM, qui est normalement réservée à l'utilisateur **root**. **administrator@ad-domain.com** est membre du groupe **ad_users_external** non-POSIX, qui est à son tour membre du groupe **ad_users** POSIX.

Conditions préalables

- Vous avez obtenu l'IdM **admin** Kerberos ticket-granting ticket (TGT).
- Une confiance inter-forêts existe entre le domaine IdM et le domaine AD **ad-domain.com**.

- Aucun compte local **administrator** n'est présent sur l'hôte **idmclient**: l'utilisateur **administrator** n'est pas répertorié dans le fichier local **/etc/passwd**.

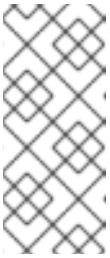
Procédure

1. Créer le groupe ***ad_users*** qui contient le groupe ***ad_users_external*** avec le membre **administrator@ad-domain**:

   a. *Optional*: Créez ou sélectionnez un groupe correspondant dans le domaine AD à utiliser pour gérer les utilisateurs AD dans le domaine IdM. Vous pouvez utiliser plusieurs groupes AD et les ajouter à différents groupes du côté IdM.

   b. Créez le groupe ***ad_users_external*** et indiquer qu'il contient des membres extérieurs au domaine IdM en ajoutant l'option **--external**:

```
[root@ipaserver ~]# ipa group-add --desc='AD users external map'
ad_users_external --external
-------------------------------
Added group "ad_users_external"
-------------------------------
  Group name: ad_users_external
  Description: AD users external map
```



NOTE

Assurez-vous que le groupe externe que vous spécifiez ici est un groupe de sécurité AD avec une portée de groupe **global** ou **universal** comme défini dans le document sur les groupes de sécurité Active Directory . Par exemple, les groupes de sécurité AD **Domain users** ou **Domain admins** ne peuvent pas être utilisés car leur périmètre de groupe est **domain local**.

   c. Créez le groupe **ad_users**:

```
[root@ipaserver ~]# ipa group-add --desc='AD users' ad_users
----------------------
Added group "ad_users"
----------------------
  Group name: ad_users
  Description: AD users
  GID: 129600004
```

   d. Ajoutez l'utilisateur AD **administrator@ad-domain.com** à **ad_users_external** en tant que membre externe :

```
[root@ipaserver ~]# ipa group-add-member ad_users_external --external
"administrator@ad-domain.com"
 [member user]:
 [member group]:
  Group name: ad_users_external
  Description: AD users external map
  External member: S-1-5-21-3655990580-1375374850-1633065477-513
------------------------
Number of members added 1
------------------------
```

L'utilisateur AD doit être identifié par un nom complet, tel que **DOMAIN\user_name** ou **user_name@DOMAIN**. L'identité AD est ensuite mappée au SID AD de l'utilisateur. Il en va de même pour l'ajout de groupes AD.

e. Ajoutez **ad_users_external** à **ad_users** en tant que membre :

```
[root@ipaserver ~]# ipa group-add-member ad_users --groups ad_users_external
  Group name: ad_users
  Description: AD users
  GID: 129600004
  Member groups: ad_users_external
-------------------------
Number of members added 1
-------------------------
```

2. Accordez aux membres de **ad_users** la permission d'exécuter **/usr/sbin/reboot** sur l'hôte **idmclient**:

a. Ajouter la commande **/usr/sbin/reboot** à la base de données IdM des commandes **sudo**:

```
[root@idmclient ~]# ipa sudocmd-add /usr/sbin/reboot
-------------------------------------
Added Sudo Command "/usr/sbin/reboot"
-------------------------------------
  Sudo Command: /usr/sbin/reboot
```

b. Créez une règle **sudo** nommée **ad_users_reboot**:

```
[root@idmclient ~]# ipa sudorule-add ad_users_reboot
---------------------------------
Added Sudo Rule "ad_users_reboot"
---------------------------------
  Rule name: ad_users_reboot
  Enabled: True
```

c. Ajoutez la commande **/usr/sbin/reboot** à la règle **ad_users_reboot**:

```
[root@idmclient ~]# ipa sudorule-add-allow-command ad_users_reboot --sudocmds
'/usr/sbin/reboot'
  Rule name: ad_users_reboot
  Enabled: True
  Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```

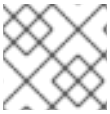
d. Appliquer la règle **ad_users_reboot** à l'hôte IdM **idmclient**:

```
[root@idmclient ~]# ipa sudorule-add-host ad_users_reboot --hosts
idmclient.idm.example.com
Rule name: ad_users_reboot
Enabled: True
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
```

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

e. Ajoutez le groupe **ad_users** à la règle **ad_users_reboot**:

```
[root@idmclient ~]# ipa sudorule-add-user ad_users_reboot --groups ad_users
Rule name: ad_users_reboot
Enabled: TRUE
User Groups: ad_users
Hosts: idmclient.idm.example.com
Sudo Allow Commands: /usr/sbin/reboot
-------------------------
Number of members added 1
-------------------------
```



NOTE

La propagation des changements du serveur au client peut prendre quelques minutes.

Verification steps

1. Connectez-vous à l'hôte **idmclient** en tant que **administrator@ad-domain.com**, un membre indirect du groupe **ad_users**:

```
$ ssh administrator@ad-domain.com@ipaclient
Password:
```

2. Optionnellement, afficher les commandes **sudo** que **administrator@ad-domain.com** est autorisé à exécuter :

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

User administrator@ad-domain.com may run the following commands on idmclient:
    (root) /usr/sbin/reboot
```

3. Redémarrez la machine en utilisant **sudo**. Saisissez le mot de passe de **administrator@ad-domain.com** lorsque vous y êtes invité :

```
[administrator@ad-domain.com@idmclient ~]$ sudo /usr/sbin/reboot
[sudo] password for administrator@ad-domain.com:
```

Ressources supplémentaires

Ressources supplémentaires

- Utilisateurs d'Active Directory et groupes de gestion des identités
- Inclure les utilisateurs et les groupes d'un domaine Active Directory de confiance dans les règles SUDO

20.4. ACCORDER UN ACCÈS SUDO À UN UTILISATEUR IDM SUR UN CLIENT IDM À L'AIDE DE L'INTERFACE WEB IDM

Dans la gestion des identités (IdM), vous pouvez accorder l'accès **sudo** pour une commande spécifique à un compte d'utilisateur IdM sur un hôte IdM spécifique. Commencez par ajouter une commande **sudo**, puis créez une règle **sudo** pour une ou plusieurs commandes.

Suivez cette procédure pour créer la règle sudo **idm_user_reboot** afin d'autoriser le compte **idm_user** à exécuter la commande **/usr/sbin/reboot** sur la machine **idmclient**.

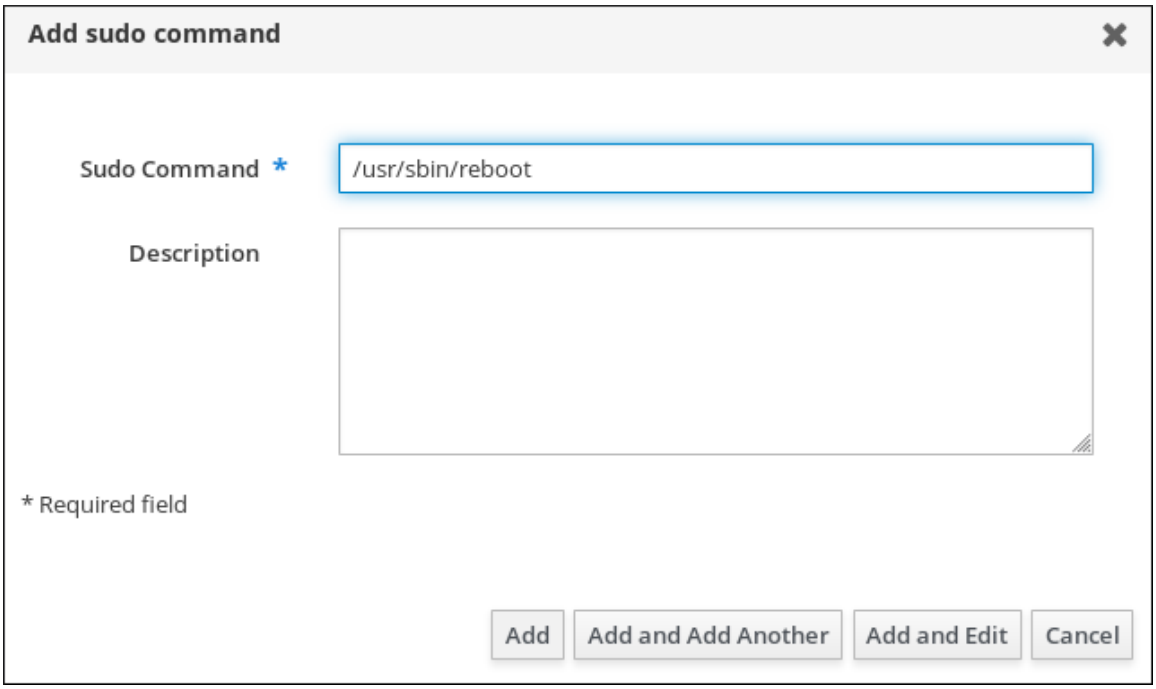
Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Vous avez créé un compte utilisateur pour **idm_user** dans IdM et déverrouillé le compte en créant un mot de passe pour l'utilisateur. Pour plus de détails sur l'ajout d'un nouvel utilisateur IdM à l'aide de l'interface de ligne de commande, voir Ajouter des utilisateurs à l'aide de la ligne de commande.
- Aucun compte local **idm_user** n'est présent sur l'hôte **idmclient**. L'utilisateur **idm_user** n'est pas répertorié dans le fichier local **/etc/passwd**.

Procédure

1. Ajouter la commande **/usr/sbin/reboot** à la base de données IdM des commandes **sudo**:
   a. Naviguez vers **Policy → Sudo → Sudo Commands**.
   b. Cliquez sur **Add** dans le coin supérieur droit pour ouvrir la boîte de dialogue **Add sudo command**.
   c. Saisissez la commande que vous souhaitez que l'utilisateur puisse exécuter en utilisant **sudo**: **/usr/sbin/reboot**.

Figure 20.1. Ajout de la commande sudo de l'IdM



d. Cliquez sur **Add**.

2. Utilisez la nouvelle entrée de commande **sudo** pour créer une règle sudo permettant à **idm_user** de redémarrer la machine **idmclient**:

   a. Naviguez vers **Policy → Sudo → Sudo rules**.

   b. Cliquez sur **Add** dans le coin supérieur droit pour ouvrir la boîte de dialogue **Add sudo rule**.

   c. Saisissez le nom de la règle **sudo**: **idm_user_reboot**.

   d. Cliquez sur **Add and Edit**

   e. Spécifiez l'utilisateur :

      i. Dans la section **Who**, cochez le bouton radio **Specified Users and Groups**.

      ii. Dans la sous-section **User category the rule applies to**, cliquez sur **Add** pour ouvrir la boîte de dialogue **Add users into sudo rule "idm_user_reboot"**.

      iii. Dans la boîte de dialogue **Add users into sudo rule "idm_user_reboot"**, dans la colonne **Available**, cochez la case **idm_user** et déplacez-la dans la colonne **Prospective**.

      iv. Cliquez sur **Add**.

   f. Spécifiez l'hôte :

      i. Dans la section **Access this host**, cochez le bouton radio **Specified Hosts and Groups**.

      ii. Dans la sous-section **Host category this rule applies to**, cliquez sur **Add** pour ouvrir la boîte de dialogue **Add hosts into sudo rule "idm_user_reboot"**.

      iii. Dans la boîte de dialogue **Add hosts into sudo rule "idm_user_reboot"**, dans la colonne **Available**, cochez la case **idmclient.idm.example.com** et déplacez-la dans la colonne **Prospective**.

iv. Cliquez sur **Add**.

g. Spécifiez les commandes :

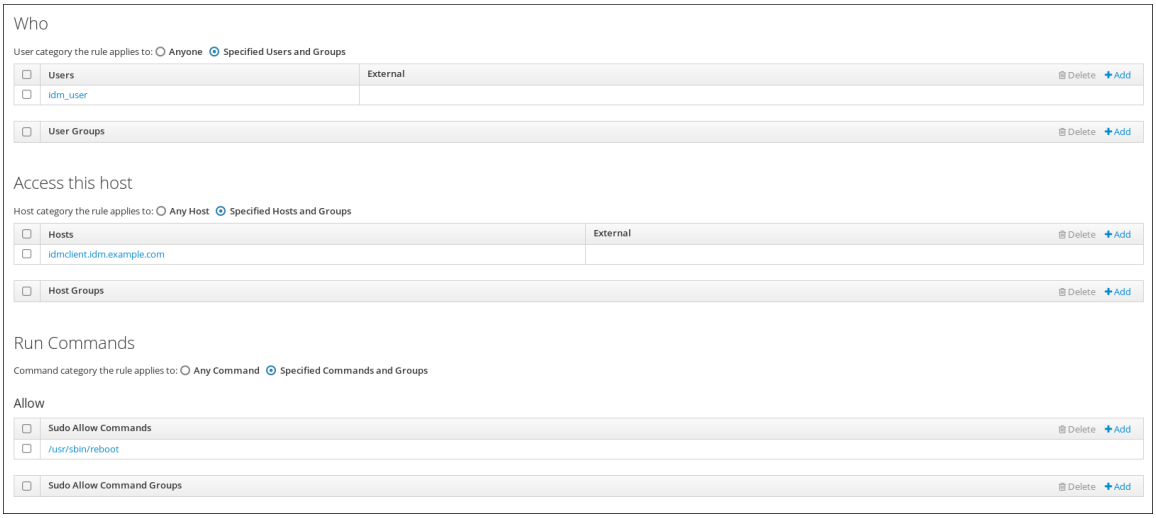
i. Dans la sous-section **Command category the rule applies to** de la section **Run Commands**, cochez le bouton radio **Specified Commands and Groups**.

ii. Dans la sous-section **Sudo Allow Commands**, cliquez sur **Add** pour ouvrir la boîte de dialogue **Add allow sudo commands into sudo rule "idm_user_reboot"**.

iii. Dans la boîte de dialogue **Add allow sudo commands into sudo rule "idm_user_reboot"**, dans la colonne **Available**, cochez la case **/usr/sbin/reboot** et déplacez-la dans la colonne **Prospective**.

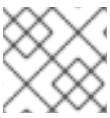
iv. Cliquez sur **Add** pour revenir à la page **idm_sudo_reboot**.

Figure 20.2. Ajout d'une règle sudo à l'IdM



h. Cliquez sur **Save** dans le coin supérieur gauche.

La nouvelle règle est activée par défaut.



NOTE

La propagation des changements du serveur au client peut prendre quelques minutes.

Verification steps

1. Connectez-vous à **idmclient** en tant que **idm_user**.

2. Redémarrez la machine en utilisant **sudo**. Saisissez le mot de passe de **idm_user** lorsque vous y êtes invité :

```
$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

Si la règle **sudo** est configurée correctement, la machine redémarre.

20.5. CRÉATION D'UNE RÈGLE SUDO SUR LA CLI QUI EXÉCUTE UNE COMMANDE EN TANT QUE COMPTE DE SERVICE SUR UN CLIENT IDM

Dans IdM, vous pouvez configurer une règle **sudo** avec une règle *RunAs alias* pour exécuter une commande **sudo** en tant qu'autre utilisateur ou groupe. Par exemple, vous pouvez avoir un client IdM qui héberge une application de base de données et vous devez exécuter des commandes en tant que compte de service local correspondant à cette application.

Utilisez cet exemple pour créer une règle **sudo** sur la ligne de commande appelée **run_third-party-app_report** afin d'autoriser le compte **idm_user** à exécuter la commande **/opt/third-party-app/bin/report** en tant que compte de service **thirdpartyapp** sur l'hôte **idmclient**.

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Vous avez créé un compte utilisateur pour **idm_user** dans IdM et déverrouillé le compte en créant un mot de passe pour l'utilisateur. Pour plus d'informations sur l'ajout d'un nouvel utilisateur IdM à l'aide de la ligne de commande, voir Ajouter des utilisateurs à l'aide de la ligne de commande.
- Aucun compte local **idm_user** n'est présent sur l'hôte **idmclient**. L'utilisateur **idm_user** n'est pas répertorié dans le fichier local **/etc/passwd**.
- Vous avez une application personnalisée nommée **third-party-app** installée sur l'hôte **idmclient**.
- La commande **report** pour l'application **third-party-app** est installée dans le répertoire **/opt/third-party-app/bin/report**.
- Vous avez créé un compte de service local nommé **thirdpartyapp** pour exécuter les commandes de l'application **third-party-app**.

Procédure

1. Récupérer un ticket Kerberos en tant qu'IdM **admin**.

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

2. Ajouter la commande **/opt/third-party-app/bin/report** à la base de données IdM des commandes **sudo**:

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

3. Créez une règle **sudo** nommée **run_third-party-app_report**:

```
[root@idmclient ~]# ipa sudorule-add run_third-party-app_report
-------------------------------------------
Added Sudo Rule "run_third-party-app_report"
-------------------------------------------
  Rule name: run_third-party-app_report
  Enabled: TRUE
```

4. Utilisez l'option **--users=*<user>*** pour spécifier l'utilisateur RunAs pour la commande **sudorule-add-runasuser**:

```
[root@idmclient ~]# ipa sudorule-add-runasuser run_third-party-app_report --users=thirdpartyapp
  Rule name: run_third-party-app_report
  Enabled: TRUE
  RunAs External User: thirdpartyapp
-------------------------
Number of members added 1
-------------------------
```

L'utilisateur (ou le groupe spécifié avec l'option **--groups=***) peut être externe à IdM, comme un compte de service local ou un utilisateur Active Directory. N'ajoutez pas de préfixe **%** pour les noms de groupes.

5. Ajoutez la commande **/opt/third-party-app/bin/report** à la règle **run_third-party-app_report**:

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

6. Appliquer la règle **run_third-party-app_report** à l'hôte IdM **idmclient**:

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

7. Ajoutez le compte **idm_user** à la règle **run_third-party-app_report**:

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

NOTE

La propagation des changements du serveur au client peut prendre quelques minutes.

Verification steps

1. Connectez-vous à l'hôte **idmclient** en tant que compte **idm_user**.
2. Testez la nouvelle règle sudo :
   a. Affiche les règles **sudo** que le compte **idm_user** est autorisé à appliquer.

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

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

   b. Exécutez la commande **report** en tant que compte de service **thirdpartyapp**.

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

20.6. CRÉATION D'UNE RÈGLE SUDO DANS L'INTERFACE WEB IDM QUI EXÉCUTE UNE COMMANDE EN TANT QUE COMPTE DE SERVICE SUR UN CLIENT IDM

Dans IdM, vous pouvez configurer une règle **sudo** avec une règle *RunAs alias* pour exécuter une commande **sudo** en tant qu'autre utilisateur ou groupe. Par exemple, vous pouvez avoir un client IdM qui héberge une application de base de données et vous devez exécuter des commandes en tant que compte de service local correspondant à cette application.

Utilisez cet exemple pour créer une règle **sudo** dans l'IdM WebUI appelée **run_third-party-app_report** pour permettre au compte **idm_user** d'exécuter la commande **/opt/third-party-app/bin/report** en tant que compte de service **thirdpartyapp** sur l'hôte **idmclient**.

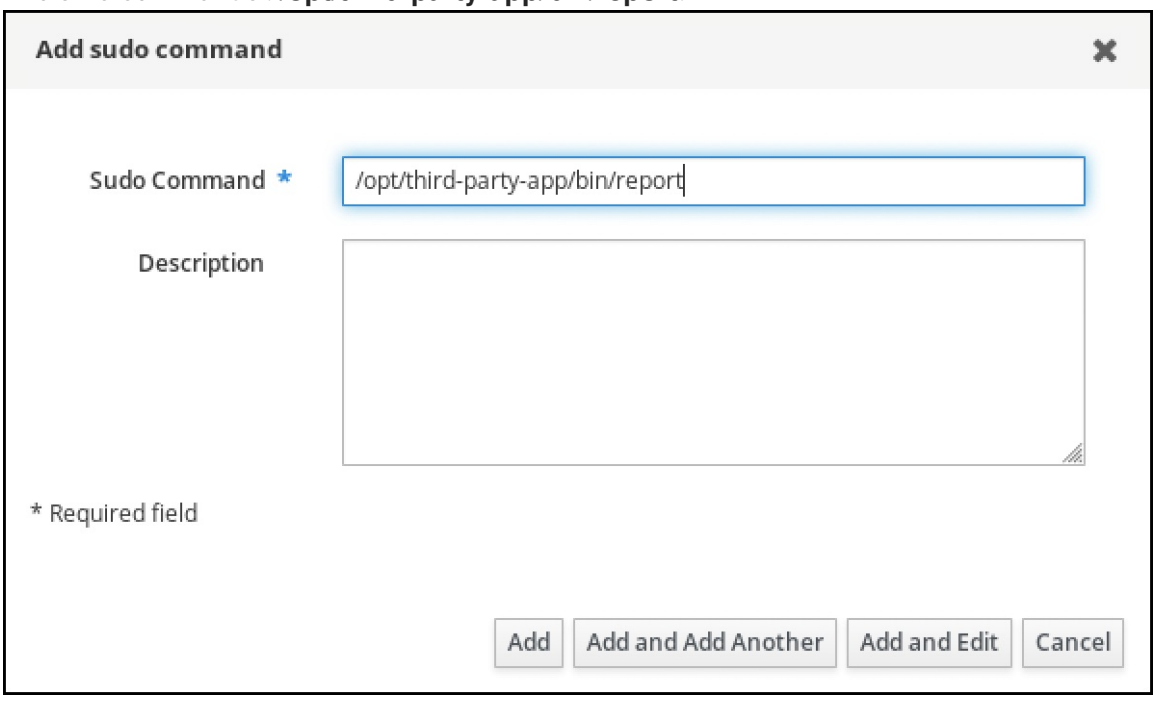
Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Vous avez créé un compte utilisateur pour **idm_user** dans IdM et déverrouillé le compte en créant un mot de passe pour l'utilisateur. Pour plus d'informations sur l'ajout d'un nouvel utilisateur IdM à l'aide de la ligne de commande, voir Ajouter des utilisateurs à l'aide de la ligne de commande.

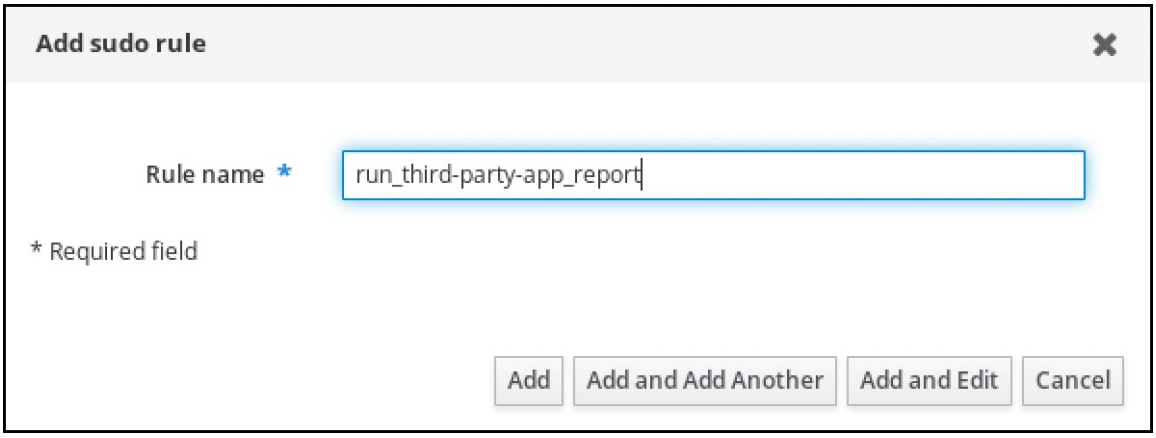
- Aucun compte local **idm_user** n'est présent sur l'hôte **idmclient**. L'utilisateur **idm_user** n'est pas répertorié dans le fichier local **/etc/passwd**.
- Vous avez une application personnalisée nommée **third-party-app** installée sur l'hôte **idmclient**.
- La commande **report** pour l'application **third-party-app** est installée dans le répertoire **/opt/third-party-app/bin/report**.
- Vous avez créé un compte de service local nommé **thirdpartyapp** pour exécuter les commandes de l'application **third-party-app**.

Procédure

1. Ajouter la commande **/opt/third-party-app/bin/report** à la base de données IdM des commandes **sudo**:
   a. Naviguez vers **Policy → Sudo → Sudo Commands**.
   b. Cliquez sur **Add** dans le coin supérieur droit pour ouvrir la boîte de dialogue **Add sudo command**.
   c. Entrez la commande : **/opt/third-party-app/bin/report**.



   d. Cliquez sur **Add**.
2. Utilisez l'entrée de commande new **sudo** pour créer la nouvelle règle **sudo**:
   a. Naviguez vers **Policy → Sudo → Sudo rules**.
   b. Cliquez sur **Add** dans le coin supérieur droit pour ouvrir la boîte de dialogue **Add sudo rule**.
   c. Saisissez le nom de la règle **sudo**: **run_third-party-app_report**.



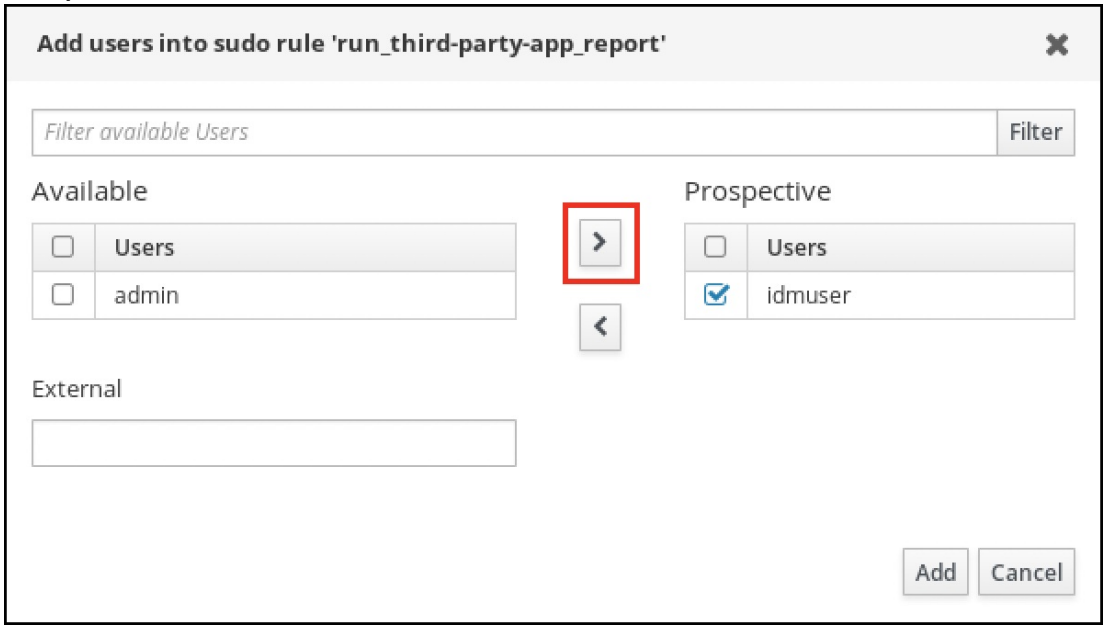
d. Cliquez sur **Add and Edit**

e. Spécifiez l'utilisateur :

i. Dans la section **Who**, cochez le bouton radio **Specified Users and Groups**.

ii. Dans la sous-section **User category the rule applies to**, cliquez sur **Add** pour ouvrir la boîte de dialogue **Add users into sudo rule "run_third-party-app_report"**.

iii. Dans la boîte de dialogue **Add users into sudo rule "run_third-party-app_report"**, dans la colonne **Available**, cochez la case **idm_user** et déplacez-la dans la colonne **Prospective**.



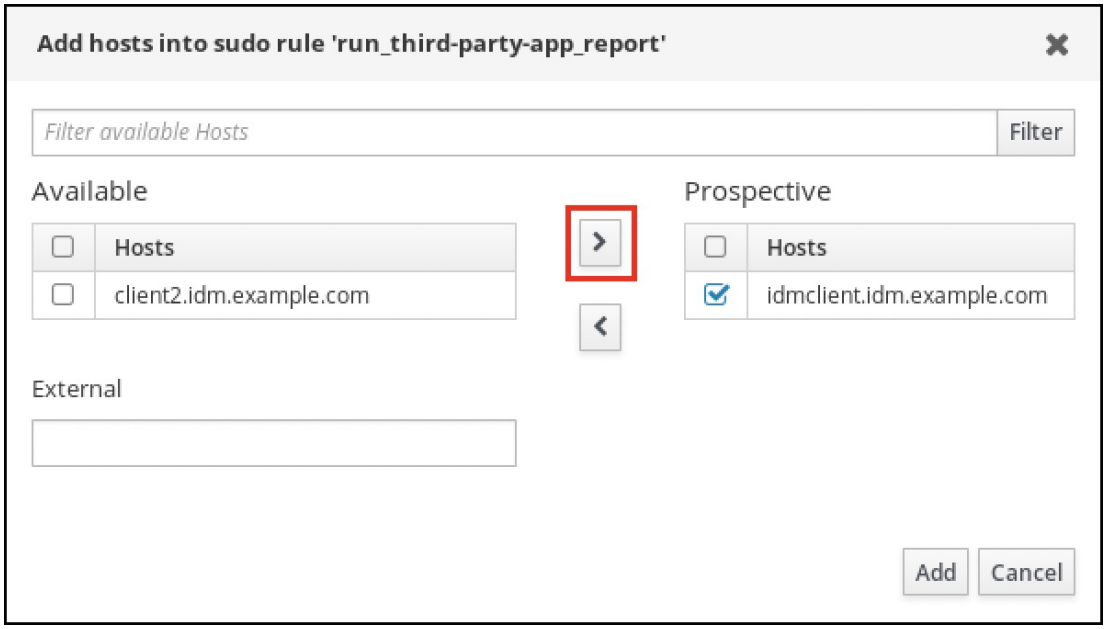
iv. Cliquez sur **Add**.

f. Spécifiez l'hôte :

i. Dans la section **Access this host**, cochez le bouton radio **Specified Hosts and Groups**.

ii. Dans la sous-section **Host category this rule applies to**, cliquez sur **Add** pour ouvrir la boîte de dialogue **Add hosts into sudo rule "run_third-party-app_report"**.

iii. Dans la boîte de dialogue **Add hosts into sudo rule "run_third-party-app_report"**, dans la colonne **Available**, cochez la case **idmclient.idm.example.com** et déplacez-la dans la colonne **Prospective**.



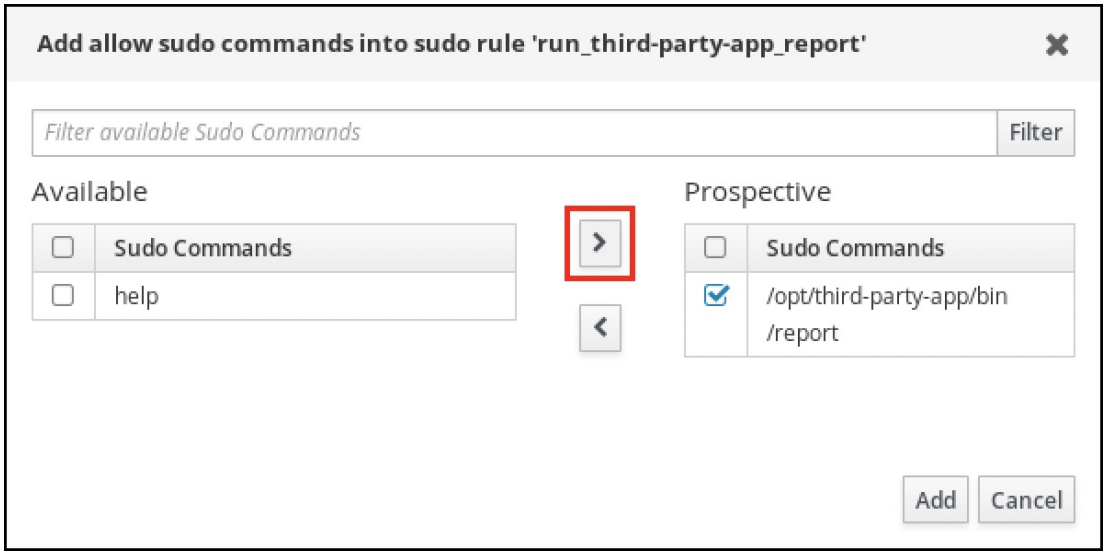
iv. Cliquez sur **Add**.

g. Spécifiez les commandes :

i. Dans la sous-section **Command category the rule applies to** de la section **Run Commands**, cochez le bouton radio **Specified Commands and Groups**.

ii. Dans la sous-section **Sudo Allow Commands**, cliquez sur **Add** pour ouvrir la boîte de dialogue **Add allow sudo commands into sudo rule "run_third-party-app_report"**.

iii. Dans la boîte de dialogue **Add allow sudo commands into sudo rule "run_third-party-app_report"**, dans la colonne **Available**, cochez la case **/opt/third-party-app/bin/report** et déplacez-la dans la colonne **Prospective**.



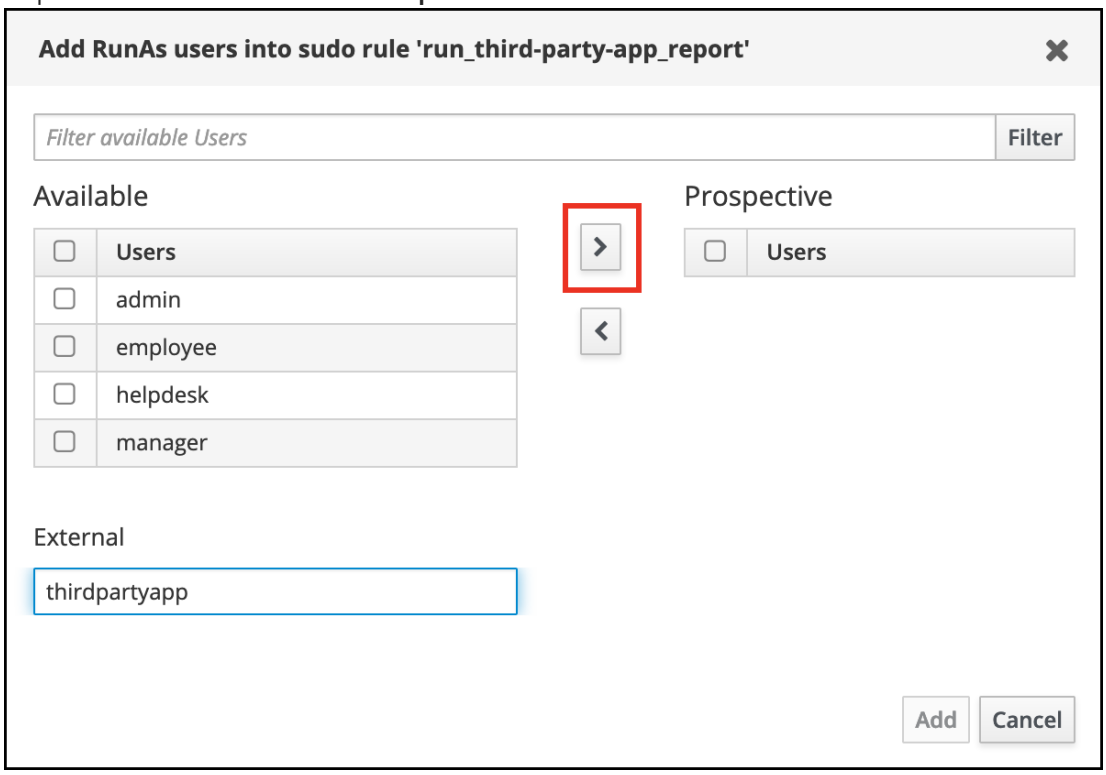
iv. Cliquez sur **Add** pour revenir à la page **run_third-party-app_report**.

h. Spécifiez l'utilisateur RunAs :

i. Dans la section **As Whom**, cochez le bouton radio **Specified Users and Groups**.

ii. Dans la sous-section **RunAs Users**, cliquez sur **Add** pour ouvrir la boîte de dialogue **Add RunAs users into sudo rule "run_third-party-app_report"**.

iii. Dans la boîte de dialogue **Add RunAs users into sudo rule "run_third-party-app_report"**, saisissez le compte de service **thirdpartyapp** dans la case **External** et déplacez-le dans la colonne **Prospective**.

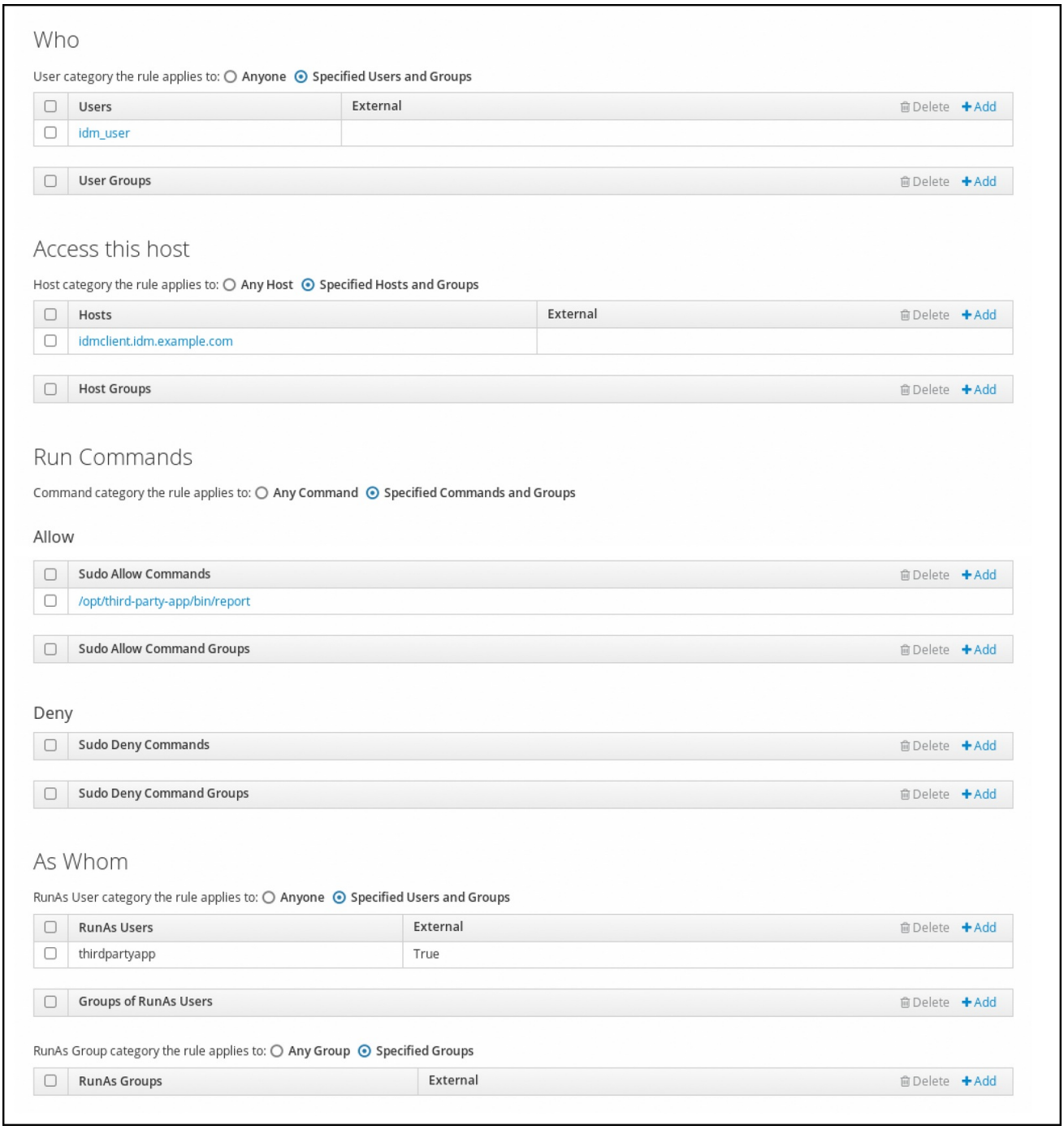


iv. Cliquez sur **Add** pour revenir à la page **run_third-party-app_report**.

i. Cliquez sur **Save** dans le coin supérieur gauche.

La nouvelle règle est activée par défaut.

Figure 20.3. Détails de la règle sudo



NOTE

La propagation des changements du serveur au client peut prendre quelques minutes.

Verification steps

1. Connectez-vous à l'hôte **idmclient** en tant que compte **idm_user**.
2. Testez la nouvelle règle sudo :
   a. Affiche les règles **sudo** que le compte **idm_user** est autorisé à appliquer.

```
[idm_user@idmclient ~]$ sudo -l
Matching Defaults entries for idm_user@idm.example.com on idmclient:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
```

```
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER
LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
XAUTHORITY KRB5CCNAME",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

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

b. Exécutez la commande **report** en tant que compte de service **thirdpartyapp**.

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

20.7. ACTIVATION DE L'AUTHENTIFICATION GSSAPI POUR SUDO SUR UN CLIENT IDM

La procédure suivante décrit l'activation de l'authentification GSSAPI (Generic Security Service Application Program Interface) sur un client IdM pour les commandes **sudo** et **sudo -i** via le module PAM **pam_sss_gss.so**. Avec cette configuration, les utilisateurs IdM peuvent s'authentifier à la commande **sudo** avec leur ticket Kerberos.

Conditions préalables

- Vous avez créé une règle **sudo** pour un utilisateur IdM qui s'applique à un hôte IdM. Pour cet exemple, vous avez créé la règle **idm_user_reboot sudo** pour accorder au compte **idm_user** la permission d'exécuter la commande **/usr/sbin/reboot** sur l'hôte **idmclient**.
- Vous devez disposer des privilèges **root** pour modifier le fichier **/etc/sssd/sssd.conf** et les fichiers PAM dans le répertoire **/etc/pam.d/**.

Procédure

1. Ouvrez le fichier de configuration **/etc/sssd/sssd.conf**.
2. Ajoutez l'entrée suivante à la section **[domain/*<domain_name>*]** l'entrée suivante.

```
[domain/<domain_name>]
pam_gssapi_services = sudo, sudo-i
```

3. Enregistrez et fermez le fichier **/etc/sssd/sssd.conf**.
4. Redémarrez le service SSSD pour charger les modifications de configuration.

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

5. Si vous utilisez RHEL 9.2 ou une version ultérieure :

   a. [Facultatif] Déterminez si vous avez sélectionné le profil **sssd authselect** :

   ```
   # authselect current
   Profile ID: sssd
   ```

   Le résultat indique que le profil **sssd authselect** est sélectionné.

   b. Si le profil **sssd authselect** est sélectionné, activez l'authentification GSSAPI :

   ```
   # authselect enable-feature with-gssapi
   ```

   c. Si le profil **sssd authselect** n'est pas sélectionné, sélectionnez-le et activez l'authentification GSSAPI :

   ```
   # authselect select sssd with-gssapi
   ```

6. Si vous utilisez RHEL 9.1 ou une version antérieure :

   a. Ouvrez le fichier de configuration PAM de **/etc/pam.d/sudo**.

   b. Ajoutez l'entrée suivante comme première ligne de la section **auth** dans le fichier **/etc/pam.d/sudo**.

   ```
   #%PAM-1.0
   auth sufficient pam_sss_gss.so
   auth       include      system-auth
   account    include      system-auth
   password   include      system-auth
   session    include      system-auth
   ```

   c. Enregistrez et fermez le fichier **/etc/pam.d/sudo**.

Verification steps

1. Connectez-vous à l'hôte en tant que compte **idm_user**.

```
[root@idm-client ~]# ssh -l idm_user@idm.example.com localhost
idm_user@idm.example.com's password:
```

2. Vérifiez que vous disposez d'un ticket d'attribution de tickets en tant que compte **idm_user**.

```
[idmuser@idmclient ~]$ klist
Ticket cache: KCM:1366201107
Default principal: idm_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
01/08/2021 09:11:48  01/08/2021 19:11:48
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
 renew until 01/15/2021 09:11:44
```

3. *(Optional)* Si vous ne disposez pas d'informations d'identification Kerberos pour le compte **idm_user**, détruisez vos informations d'identification Kerberos actuelles et demandez les informations correctes.

```
[idm_user@idmclient ~]$ kdestroy -A

[idm_user@idmclient ~]$ kinit idm_user@IDM.EXAMPLE.COM
Password for idm_user@idm.example.com:
```

4. Redémarrez la machine à l'aide de **sudo**, sans spécifier de mot de passe.

```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
```

Ressources supplémentaires

- L'entrée GSSAPI dans la liste terminologique IdM
- Accorder un accès sudo à un utilisateur IdM sur un client IdM à l'aide de l'interface Web IdM
- Accorder un accès sudo à un utilisateur IdM sur un client IdM à l'aide de la CLI
- **pam_sss_gss (8)** page de manuel
- **sssd.conf (5)** page de manuel

20.8. ACTIVATION DE L'AUTHENTIFICATION GSSAPI ET APPLICATION DES INDICATEURS D'AUTHENTIFICATION KERBEROS POUR SUDO SUR UN CLIENT IDM

La procédure suivante décrit l'activation de l'authentification GSSAPI (Generic Security Service Application Program Interface) sur un client IdM pour les commandes **sudo** et **sudo -i** via le module PAM **pam_sss_gss.so**. En outre, seuls les utilisateurs qui se sont connectés avec une carte à puce s'authentifieront à ces commandes avec leur ticket Kerberos.

NOTE

Vous pouvez utiliser cette procédure comme modèle pour configurer l'authentification GSSAPI avec SSSD pour d'autres services compatibles avec PAM, et restreindre davantage l'accès aux seuls utilisateurs dont le ticket Kerberos est associé à un indicateur d'authentification spécifique.

Conditions préalables

- Vous avez créé une règle **sudo** pour un utilisateur IdM qui s'applique à un hôte IdM. Pour cet exemple, vous avez créé la règle **idm_user_reboot sudo** pour accorder au compte **idm_user** la permission d'exécuter la commande **/usr/sbin/reboot** sur l'hôte **idmclient**.
- Vous avez configuré l'authentification par carte à puce pour l'hôte **idmclient**.
- Vous devez disposer des privilèges **root** pour modifier le fichier **/etc/sssd/sssd.conf** et les fichiers PAM dans le répertoire **/etc/pam.d/**.

Procédure

1. Ouvrez le fichier de configuration **/etc/sssd/sssd.conf**.
2. Ajoutez les entrées suivantes à la section **[domain/*<domain_name>*]** les entrées suivantes.

```
[domain/<domain_name>]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:pkinit, sudo-i:pkinit
```

3. Enregistrez et fermez le fichier **/etc/sssd/sssd.conf**.
4. Redémarrez le service SSSD pour charger les modifications de configuration.

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

5. Ouvrez le fichier de configuration PAM de **/etc/pam.d/sudo**.
6. Ajoutez l'entrée suivante comme première ligne de la section **auth** dans le fichier **/etc/pam.d/sudo**.

```
#%PAM-1.0
auth sufficient pam_sss_gss.so
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    include      system-auth
```

7. Enregistrez et fermez le fichier **/etc/pam.d/sudo**.
8. Ouvrez le fichier de configuration PAM de **/etc/pam.d/sudo-i**.
9. Ajoutez l'entrée suivante comme première ligne de la section **auth** dans le fichier **/etc/pam.d/sudo-i**.

```
#%PAM-1.0
auth sufficient pam_sss_gss.so
auth       include      sudo
account    include      sudo
password   include      sudo
session    optional     pam_keyinit.so force revoke
session    include      sudo
```

10. Enregistrez et fermez le fichier **/etc/pam.d/sudo-i**.

Verification steps

1. Connectez-vous à l'hôte en tant que compte **idm_user** et authentifiez-vous à l'aide d'une carte à puce.

```
[root@idmclient ~]# ssh -l idm_user@idm.example.com localhost
PIN for smart_card
```

2. Vérifiez que vous disposez d'un ticket d'attribution de billets en tant qu'utilisateur de la carte à puce.

```
[idm_user@idmclient ~]$ klist
Ticket cache: KEYRING:persistent:1358900015:krb_cache_TObtNMd
Default principal: idm_user@IDM.EXAMPLE.COM

Valid starting       Expires              Service principal
02/15/2021 16:29:48  02/16/2021 02:29:48
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
 renew until 02/22/2021 16:29:44
```

3. Affiche les règles **sudo** que le compte **idm_user** est autorisé à appliquer.

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

User idm_user may run the following commands on idmclient:
    (root) /usr/sbin/reboot
```

4. Redémarrez la machine à l'aide de **sudo**, sans spécifier de mot de passe.

```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
```

Ressources supplémentaires

- Options SSSD contrôlant l'authentification GSSAPI pour les services PAM
- L'entrée GSSAPI dans la liste terminologique IdM
- Configuration de la gestion des identités pour l'authentification par carte à puce
- Indicateurs d'authentification Kerberos
- Accorder un accès sudo à un utilisateur IdM sur un client IdM à l'aide de l'interface Web IdM
- Accorder un accès sudo à un utilisateur IdM sur un client IdM en utilisant le CLI .
- **pam_sss_gss (8)** page de manuel
- **sssd.conf (5)** page de manuel

20.9. OPTIONS SSSD CONTRÔLANT L'AUTHENTIFICATION GSSAPI POUR LES SERVICES PAM

Vous pouvez utiliser les options suivantes pour le fichier de configuration **/etc/sssd/sssd.conf** afin d'ajuster la configuration GSSAPI au sein du service SSSD.

pam_gssapi_services

L'authentification GSSAPI avec SSSD est désactivée par défaut. Vous pouvez utiliser cette option pour spécifier une liste de services PAM, séparés par des virgules, qui sont autorisés à essayer l'authentification GSSAPI à l'aide du module PAM **pam_sss_gss.so**. Pour désactiver explicitement l'authentification GSSAPI, définissez cette option sur **-**.

pam_gssapi_indicators_map

Cette option ne s'applique qu'aux domaines de gestion des identités (IdM). Cette option permet de répertorier les indicateurs d'authentification Kerberos requis pour accorder à PAM l'accès à un service. Les paires doivent être au format ***<PAM_service>*:_<required_authentication_indicator>_**.
Les indicateurs d'authentification valides sont les suivants :

- **otp** pour l'authentification à deux facteurs
- **radius** pour l'authentification RADIUS
- **pkinit** pour l'authentification par PKINIT, carte à puce ou certificat
- **hardened** pour des mots de passe renforcés

pam_gssapi_check_upn

Cette option est activée et définie par défaut sur **true**. Si cette option est activée, le service SSSD exige que le nom d'utilisateur corresponde aux informations d'identification Kerberos. Si l'option **false** est activée, le module PAM **pam_sss_gss.so** authentifie chaque utilisateur capable d'obtenir le ticket de service requis.

Examples

Les options suivantes activent l'authentification Kerberos pour les services **sudo** et **sudo-i**, exigent que les utilisateurs de **sudo** s'authentifient avec un mot de passe à usage unique et que les noms d'utilisateur correspondent au principal Kerberos. Ces paramètres se trouvant dans la section **[pam]**, ils s'appliquent à tous les domaines :

```
[pam]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:otp
pam_gssapi_check_upn = true
```

Vous pouvez également définir ces options dans des sections **[domain]** individuelles afin d'écraser toutes les valeurs globales dans la section **[pam]**. Les options suivantes appliquent des paramètres GSSAPI différents à chaque domaine :

Pour le domaine idm.example.com

- Activez l'authentification GSSAPI pour les services **sudo** et **sudo -i**.
- Exiger des authentificateurs de type certificat ou carte à puce pour la commande **sudo**.
- Exiger des authentificateurs à mot de passe unique pour la commande **sudo -i**.
- Assurer la correspondance entre les noms d'utilisateurs et les principes Kerberos.

Pour le domaine ad.example.com

- Activez l'authentification GSSAPI uniquement pour le service **sudo**.

- Ne pas imposer la correspondance entre les noms d'utilisateurs et les mandants.

```
[domain/idm.example.com]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:pkinit, sudo-i:otp
pam_gssapi_check_upn = true
...

[domain/ad.example.com]
pam_gssapi_services = sudo
pam_gssapi_check_upn = false
...
```

Ressources supplémentaires

- Indicateurs d'authentification Kerberos

20.10. DÉPANNAGE DE L'AUTHENTIFICATION GSSAPI POUR SUDO

Si vous ne parvenez pas à vous authentifier auprès du service **sudo** à l'aide d'un ticket Kerberos provenant de l'IdM, utilisez les scénarios suivants pour résoudre votre problème.

Conditions préalables

- Vous avez activé l'authentification GSSAPI pour le service **sudo**. Voir Activation de l'authentification GSSAPI pour sudo sur un client IdM.
- Vous devez disposer des privilèges **root** pour modifier le fichier **/etc/sssd/sssd.conf** et les fichiers PAM dans le répertoire **/etc/pam.d/**.

Procédure

- Si l'erreur suivante s'affiche, il se peut que le service Kerberos ne soit pas en mesure de résoudre le domaine correct pour le ticket de service en se basant sur le nom d'hôte :

```
Serveur introuvable dans la base de données Kerberos
```

  Dans ce cas, ajoutez le nom d'hôte directement à la section **[domain_realm]** dans le fichier de configuration Kerberos **/etc/krb5.conf**:

```
[idm-user@idm-client ~]$ cat /etc/krb5.conf
...

[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
 server.example.com = EXAMPLE.COM
```

- Si l'erreur suivante s'affiche, vous ne disposez pas d'informations d'identification Kerberos :

```
Pas d'identifiants Kerberos disponibles
```

Dans ce cas, récupérez les informations d'identification Kerberos à l'aide de l'utilitaire **kinit** ou authentifiez-vous à l'aide de SSSD :

```
[idm-user@idm-client ~]$ kinit idm-user@IDM.EXAMPLE.COM
Password for idm-user@idm.example.com:
```

- Si l'une des erreurs suivantes apparaît dans le fichier journal **/var/log/sssd/sssd_pam.log**, les informations d'identification Kerberos ne correspondent pas au nom d'utilisateur de l'utilisateur actuellement connecté :

```
User with UPN [<UPN>] was not found.

UPN [<UPN>] does not match target user [<username>].
```

Dans ce cas, vérifiez que vous vous êtes authentifié avec SSSD, ou envisagez de désactiver l'option **pam_gssapi_check_upn** dans le fichier **/etc/sssd/sssd.conf**:

```
[idm-user@idm-client ~]$ cat /etc/sssd/sssd.conf
...

pam_gssapi_check_upn = false
```

- Pour un dépannage supplémentaire, vous pouvez activer la sortie de débogage pour le module PAM de **pam_sss_gss.so**.
  - Ajoutez l'option **debug** à la fin de toutes les entrées **pam_sss_gss.so** dans les fichiers PAM, telles que **/etc/pam.d/sudo** et **/etc/pam.d/sudo-i**:

```
[root@idm-client ~]# cat /etc/pam.d/sudo
#%PAM-1.0
auth       sufficient   pam_sss_gss.so   debug
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    include      system-auth
```

```
[root@idm-client ~]# cat /etc/pam.d/sudo-i
#%PAM-1.0
auth       sufficient   pam_sss_gss.so   debug
auth       include      sudo
account    include      sudo
password   include      sudo
session    optional     pam_keyinit.so force revoke
session    include      sudo
```

  - Essayez de vous authentifier avec le module **pam_sss_gss.so** et examinez la sortie de la console. Dans cet exemple, l'utilisateur n'avait pas d'identifiants Kerberos.

```
[idm-user@idm-client ~]$ sudo ls -l /etc/sssd/sssd.conf
pam_sss_gss: Initializing GSSAPI authentication with SSSD
pam_sss_gss: Switching euid from 0 to 1366201107
pam_sss_gss: Trying to establish security context
pam_sss_gss: SSSD User name: idm-user@idm.example.com
pam_sss_gss: User domain: idm.example.com
```

```
pam_sss_gss: User principal:
pam_sss_gss: Target name: host@idm.example.com
pam_sss_gss: Using ccache: KCM:
pam_sss_gss: Acquiring credentials, principal name will be derived
pam_sss_gss: Unable to read credentials from [KCM:] [maj:0xd0000, min:0x96c73ac3]
pam_sss_gss: GSSAPI: Unspecified GSS failure.  Minor code may provide more
information
pam_sss_gss: GSSAPI: No credentials cache found
pam_sss_gss: Switching euid from 1366200907 to 0
pam_sss_gss: System error [5]: Input/output error
```

20.11. UTILISATION D'UN PLAYBOOK ANSIBLE POUR GARANTIR L'ACCÈS SUDO À UN UTILISATEUR IDM SUR UN CLIENT IDM

Dans la gestion des identités (IdM), vous pouvez vous assurer que l'accès **sudo** à une commande spécifique est accordé à un compte d'utilisateur IdM sur un hôte IdM spécifique.

Effectuez cette procédure pour vous assurer qu'une règle **sudo** nommée **idm_user_reboot** existe. La règle accorde à **idm_user** la permission d'exécuter la commande **/usr/sbin/reboot** sur la machine **idmclient**.

Conditions préalables

- 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**.
- Vous vous êtes assuré de la présence d'un compte utilisateur pour **idm_user** dans IdM et vous avez déverrouillé le compte en créant un mot de passe pour l'utilisateur. Pour plus de détails sur l'ajout d'un nouvel utilisateur IdM à l'aide de l'interface de ligne de commande, voir le lien : Ajouter des utilisateurs à l'aide de la ligne de commande .
- Aucun compte local **idm_user** n'existe sur **idmclient**. L'utilisateur **idm_user** n'est pas listé dans le fichier **/etc/passwd** sur **idmclient**.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaservers**:

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

2. Ajouter une ou plusieurs commandes **sudo**:
   a. Créer un playbook Ansible **ensure-reboot-sudocmd-is-present.yml** qui assure la présence de la commande **/usr/sbin/reboot** dans la base de données IdM des commandes **sudo**.

Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/sudocmd/ensure-sudocmd-is-present.yml**:

```
---
- name: Playbook to manage sudo command
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure sudo command is present
  - ipasudocmd:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: /usr/sbin/reboot
      state: present
```

b. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-
reboot-sudocmd-is-present.yml
```

3. Créez une règle **sudo** qui fait référence aux commandes :

a. Créez un playbook Ansible **ensure-sudorule-for-idmuser-on-idmclient-is-present.yml** qui utilise l'entrée de commande **sudo** pour s'assurer de la présence d'une règle sudo. La règle sudo permet à **idm_user** de redémarrer la machine **idmclient**. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/sudorule/ensure-sudorule-is-present.yml**:

```
---
- name: Tests
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure a sudorule is present granting idm_user the permission to run /usr/sbin/reboot
on idmclient
  - ipasudorule:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: idm_user_reboot
      description: A test sudo rule.
      allow_sudocmd: /usr/sbin/reboot
      host: idmclient.idm.example.com
      user: idm_user
      state: present
```

b. Exécutez le manuel de jeu :

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-sudorule-for-idmuser-on-idmclient-is-
present.yml
```

Verification steps

Testez que la règle **sudo** dont vous avez assuré la présence sur le serveur IdM fonctionne sur **idmclient** en vérifiant que **idm_user** peut redémarrer **idmclient** à l'aide de **sudo**. Notez qu'il peut s'écouler quelques minutes avant que les changements effectués sur le serveur ne prennent effet sur le client.

1. Connectez-vous à **idmclient** en tant que **idm_user**.

2. Redémarrez la machine en utilisant **sudo**. Saisissez le mot de passe de **idm_user** lorsque vous y êtes invité :

```
$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

Si **sudo** est configuré correctement, la machine redémarre.

Ressources supplémentaires

- Voir les fichiers **README-sudocmd.md**, **README-sudocmdgroup.md**, et **README-sudorule.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

CHAPITRE 21. ASSURER LA PRÉSENCE DE RÈGLES DE CONTRÔLE D'ACCÈS BASÉES SUR L'HÔTE DANS IDM EN UTILISANT LES PLAYBOOKS ANSIBLE

Ce chapitre décrit les stratégies d'accès basées sur l'hôte de la gestion des identités (IdM) et la manière de les définir à l'aide d'Ansible.

Ansible est un outil d'automatisation utilisé pour configurer des systèmes, déployer des logiciels et effectuer des mises à jour continues. Il inclut la prise en charge de la gestion des identités (IdM).

21.1. RÈGLES DE CONTRÔLE D'ACCÈS BASÉES SUR L'HÔTE DANS L'IDM

Les règles de contrôle d'accès basé sur l'hôte (HBAC) définissent quels utilisateurs ou groupes d'utilisateurs peuvent accéder à quels hôtes ou groupes d'hôtes en utilisant quels services ou services d'un groupe de services. En tant qu'administrateur système, vous pouvez utiliser les règles HBAC pour atteindre les objectifs suivants :

- Limiter l'accès à un système spécifique de votre domaine aux membres d'un groupe d'utilisateurs spécifique.
- Autoriser uniquement l'utilisation d'un service spécifique pour accéder aux systèmes de votre domaine.

Par défaut, IdM est configuré avec une règle HBAC par défaut nommée **allow_all**, ce qui signifie un accès universel à chaque hôte pour chaque utilisateur via chaque service pertinent dans l'ensemble du domaine IdM.

Vous pouvez affiner l'accès à différents hôtes en remplaçant la règle par défaut **allow_all** par votre propre ensemble de règles HBAC. Pour une gestion centralisée et simplifiée du contrôle d'accès, vous pouvez appliquer les règles HBAC à des groupes d'utilisateurs, des groupes d'hôtes ou des groupes de services plutôt qu'à des utilisateurs, des hôtes ou des services individuels.

21.2. ASSURER LA PRÉSENCE D'UNE RÈGLE HBAC DANS IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

Cette section décrit comment assurer la présence d'une règle de contrôle d'accès basé sur l'hôte (HBAC) dans la gestion des identités (IdM) à l'aide d'un playbook Ansible.

Conditions préalables

- 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**.

- Les utilisateurs et les groupes d'utilisateurs que vous voulez utiliser pour votre règle HBAC existent dans IdM. Pour plus de détails, voir Gérer les comptes utilisateurs à l'aide des playbooks An sible et Assurer la présence des groupes IdM et des membres des groupes à l'aide des playbooks Ansible.

- Les hôtes et les groupes d'hôtes auxquels vous voulez appliquer votre règle HBAC existent dans IdM. Pour plus de détails, voir Gérer les hôtes à l'aide des playbooks Ansible et Gérer les groupes d'hôtes à l'aide des playbooks Ansible.

Procédure

1. Créez un fichier d'inventaire, par exemple **inventory.file**, et définissez-y **ipaserver**:

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

2. Créez votre fichier playbook Ansible qui définit la politique HBAC dont vous voulez assurer la présence. Pour simplifier cette étape, vous pouvez copier et modifier l'exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/hbacrule/ensure-hbacrule-allhosts-present.yml**:

```
---
- name: Playbook to handle hbacrules
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure idm_user can access client.idm.example.com via the sshd service
  - ipahbacrule:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: login
      user: idm_user
      host: client.idm.example.com
      hbacsvc:
      - sshd
      state: present
```

3. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory/ensure-new-
hbacrule-present.yml
```

Verification steps

1. Connectez-vous à l'interface Web IdM en tant qu'administrateur.
2. Naviguez vers **Policy → Host-Based-Access-Control → HBAC Test**
3. Dans l'onglet **Who**, sélectionnez idm_user.
4. Dans l'onglet **Accessing**, sélectionnez **client.idm.example.com**.
5. Dans l'onglet **Via service**, sélectionnez **sshd**.

6. Dans l'onglet **Rules**, sélectionnez **login**.

7. Dans l'onglet **Run test**, cliquez sur le bouton **Run test**. Si vous voyez ACCÈS ACCORDÉ, la règle HBAC a été mise en œuvre avec succès.

Ressources supplémentaires

- Voir les fichiers **README-hbacsvc.md**, **README-hbacsvcgroup.md**, et **README-hbacrule.md** dans le répertoire **/usr/share/doc/ansible-freeipa**.

- Voir les playbooks dans les sous-répertoires du répertoire **/usr/share/doc/ansible-freeipa/playbooks**.

CHAPITRE 22. COFFRES-FORTS DANS L'IDM

Ce chapitre décrit les chambres fortes dans la gestion des identités (IdM). Il présente les sujets suivants :

- Le concept de la chambre forte .
- Les différents rôles associés à un coffre-fort .
- Les différents types de chambres fortes disponibles dans l'IdM en fonction du niveau de sécurité et de contrôle d'accès.
- Les différents types de coffres disponibles dans l'IdM en fonction de la propriété .
- Le concept des conteneurs de la chambre forte .
- Commandes de base pour la gestion des coffres-forts dans IdM .
- Installation de l'autorité de récupération des clés (KRA), qui est une condition préalable à l'utilisation des chambres fortes dans IdM.

22.1. LES CHAMBRES FORTES ET LEURS AVANTAGES

Un coffre-fort est une fonction utile pour les utilisateurs de la gestion des identités (IdM) qui souhaitent conserver toutes leurs données sensibles en un seul endroit, de manière sûre et pratique. Cette section explique les différents types d'espaces de stockage et leurs utilisations, ainsi que le choix de l'espace de stockage en fonction de vos besoins.

Un coffre-fort est un emplacement sécurisé dans (IdM) pour le stockage, l'extraction, le partage et la récupération d'un secret. Un secret est une donnée sensible sur le plan de la sécurité, généralement des identifiants d'authentification, à laquelle seul un groupe limité de personnes ou d'entités peut avoir accès. Par exemple, les secrets comprennent

- mots de passe
- NIP
- clés SSH privées

Un coffre-fort est comparable à un gestionnaire de mots de passe. Tout comme un gestionnaire de mots de passe, un coffre-fort exige généralement que l'utilisateur génère et mémorise un mot de passe principal pour déverrouiller et accéder à toutes les informations stockées dans le coffre-fort. Toutefois, un utilisateur peut également décider d'opter pour un coffre-fort standard. Dans ce cas, l'utilisateur n'a pas besoin de saisir de mot de passe pour accéder aux secrets stockés dans le coffre-fort.

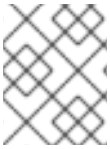
NOTE

L'objectif des chambres fortes dans l'IdM est de stocker les informations d'authentification qui vous permettent de vous authentifier auprès de services externes non liés à l'IdM.

Les autres caractéristiques importantes des chambres fortes IdM sont les suivantes :

- Les chambres fortes ne sont accessibles qu'au propriétaire de la chambre forte et aux utilisateurs de l'IdM que le propriétaire de la chambre forte sélectionne comme membres de la chambre forte. En outre, l'administrateur IdM a accès à l'espace de stockage.

- Si un utilisateur ne dispose pas de privilèges suffisants pour créer un coffre-fort, un administrateur IdM peut créer le coffre-fort et désigner l'utilisateur comme son propriétaire.
- Les utilisateurs et les services peuvent accéder aux secrets stockés dans un coffre-fort à partir de n'importe quelle machine inscrite dans le domaine IdM.
- Un coffre-fort ne peut contenir qu'un seul secret, par exemple un fichier. Toutefois, le fichier lui-même peut contenir plusieurs secrets tels que des mots de passe, des tableaux de clés ou des certificats.



NOTE

Vault n'est disponible qu'à partir de la ligne de commande IdM (CLI), et non à partir de l'interface Web IdM.

22.2. PROPRIÉTAIRES, MEMBRES ET ADMINISTRATEURS DE CHAMBRES FORTES

La gestion de l'identité (IdM) distingue les types d'utilisateurs de la chambre forte suivants :

Propriétaire de la chambre forte

Le propriétaire d'un coffre-fort est un utilisateur ou un service qui dispose de privilèges de gestion de base sur le coffre-fort. Par exemple, un propriétaire de coffre-fort peut modifier les propriétés du coffre-fort ou ajouter de nouveaux membres au coffre-fort.
Chaque chambre forte doit avoir au moins un propriétaire. Un coffre-fort peut également avoir plusieurs propriétaires.

Membre de la chambre forte

Un membre d'un coffre-fort est un utilisateur ou un service qui peut accéder à un coffre-fort créé par un autre utilisateur ou service.

Administrateur de la chambre forte

Les administrateurs de coffre-fort ont un accès illimité à tous les coffres-forts et sont autorisés à effectuer toutes les opérations sur les coffres-forts.

NOTE

Les chambres fortes symétriques et asymétriques sont protégées par un mot de passe ou une clé et appliquent des règles spéciales de contrôle d'accès (voir Types de chambres fortes). L'administrateur doit respecter ces règles pour :

- Secrets d'accès dans les chambres fortes symétriques et asymétriques.
- Modifier ou réinitialiser le mot de passe ou la clé du coffre-fort.

Un administrateur de coffre-fort est un utilisateur disposant du privilège **Vault Administrators**. Dans le contexte du contrôle d'accès basé sur les rôles (RBAC) dans IdM, un privilège est un groupe de permissions que vous pouvez appliquer à un rôle.

Utilisateur de la chambre forte

L'utilisateur de l'espace de stockage représente l'utilisateur dans le conteneur duquel se trouve l'espace de stockage. L'information **Vault user** est affichée dans la sortie de commandes spécifiques, telles que **ipa vault-show**:

```
$ ipa vault-show my_vault
  Vault name: my_vault
  Type: standard
  Owner users: user
  Vault user: user
```

Pour plus d'informations sur les conteneurs de coffre-fort et les coffres-forts utilisateur, voir Conteneurs de coffre-fort.

Ressources supplémentaires

- Voir voûtes standard, symétriques et asymétriques pour plus de détails sur les types de voûtes.

22.3. VOÛTES STANDARD, SYMÉTRIQUES ET ASYMÉTRIQUES

En fonction du niveau de sécurité et de contrôle d'accès, l'IdM classe les chambres fortes dans les types suivants :

Voûtes standard

Les propriétaires et les membres des chambres fortes peuvent archiver et récupérer les secrets sans avoir à utiliser de mot de passe ou de clé.

Voûtes symétriques

Les secrets contenus dans le coffre-fort sont protégés par une clé symétrique. Les propriétaires et les membres du coffre-fort peuvent archiver et récupérer les secrets, mais ils doivent fournir le mot de passe du coffre-fort.

Voûtes asymétriques

Les secrets de la chambre forte sont protégés par une clé asymétrique. Les utilisateurs archivent le secret à l'aide d'une clé publique et le récupèrent à l'aide d'une clé privée. Les membres du coffre-fort ne peuvent qu'archiver les secrets, tandis que les propriétaires du coffre-fort peuvent faire les deux, archiver et récupérer les secrets.

22.4. COFFRES-FORTS D'UTILISATEURS, DE SERVICES ET PARTAGÉS

En fonction de la propriété, l'IdM classe les chambres fortes en plusieurs types. Le tableau ci-dessous contient des informations sur chaque type, son propriétaire et son utilisation.

Tableau 22.1. Coffres-forts de l'IdM basés sur la propriété

Type	Description	Propriétaire	Note
User vault	Un coffre-fort privé pour un utilisateur	Un seul utilisateur	Tout utilisateur peut posséder un ou plusieurs coffres-forts d'utilisateur si l'administrateur IdM l'autorise
Service vault	Un caveau privé pour un service	Un seul service	Tout service peut posséder un ou plusieurs coffres-forts d'utilisateur si l'administrateur IdM l'autorise

Type	Description	Propriétaire	Note
Shared vault	Un coffre-fort partagé par plusieurs utilisateurs et services	L'administrateur du coffre-fort qui a créé le coffre-fort	Les utilisateurs et les services peuvent posséder un ou plusieurs coffres-forts d'utilisateur si l'administrateur IdM l'autorise. Les administrateurs de coffre-fort autres que celui qui a créé le coffre-fort ont également un accès complet au coffre-fort.

22.5. CONTENEURS À CLAIRE-VOIE

Un conteneur d'espace de stockage est un ensemble d'espaces de stockage. Le tableau ci-dessous répertorie les conteneurs de coffre-fort par défaut fournis par Identity Management (IdM).

Tableau 22.2. Conteneurs de coffre-fort par défaut dans l'IdM

Type	Description	Objectif
Conteneur utilisateur	Un conteneur privé pour un utilisateur	Stocke les coffres-forts d'un utilisateur particulier
Conteneur de services	Un conteneur privé pour un service	Stocke les coffres-forts d'un service particulier
Conteneur partagé	Un conteneur pour plusieurs utilisateurs et services	Stocke des coffres-forts qui peuvent être partagés par plusieurs utilisateurs ou services

L'IdM crée automatiquement des conteneurs d'utilisateurs et de services pour chaque utilisateur ou service lorsque le premier coffre-fort privé de l'utilisateur ou du service est créé. Après la suppression de l'utilisateur ou du service, l'IdM supprime le conteneur et son contenu.

22.6. COMMANDES DE BASE DU COFFRE-FORT IDM

Cette section décrit les commandes de base que vous pouvez utiliser pour gérer les coffres-forts de la gestion des identités (IdM). Le tableau ci-dessous contient une liste des commandes **ipa vault-*** avec l'explication de leur fonction.

NOTE

Avant d'exécuter une commande **ipa vault-***, installez le composant du système de certificats Key Recovery Authority (KRA) sur un ou plusieurs serveurs de votre domaine IdM. Pour plus de détails, voir Installation de l'autorité de recouvrement des clés dans IdM .

Tableau 22.3. Commandes de base du coffre-fort de l'IdM avec explications

Commandement	Objectif
ipa help vault	Affiche des informations conceptuelles sur les espaces de stockage IdM et des exemples de commandes d'espaces de stockage.
ipa vault-add --help, **ipa vault-find --help**	L'ajout de l'option **--help** à une commande **ipa vault-*** spécifique permet d'afficher les options et l'aide détaillée disponibles pour cette commande.

ipa vault-show user_vault --user idm_user	Lorsque vous accédez à un coffre-fort en tant que membre du coffre-fort, vous devez spécifier le propriétaire du coffre-fort. Si vous n'indiquez pas le propriétaire de l'espace de stockage, l'IdM vous informe qu'il n'a pas trouvé l'espace de stockage :
	`[admin@server ~]$ ipa vault-show user_vault`
	`ipa: ERROR: user_vault: vault not found`

ipa vault-show shared_vault --shared	Lorsque vous accédez à un coffre-fort partagé, vous devez spécifier que le coffre-fort auquel vous voulez accéder est un coffre-fort partagé. Sinon, l'IdM vous informe qu'il n'a pas trouvé l'espace de stockage :
	`[admin@server ~]$ ipa vault-show shared_vault`
	`ipa: ERROR: shared_vault: vault not found`

22.7. INSTALLATION DE L'AUTORITÉ DE RECOUVREMENT DES CLÉS DANS IDM

Cette section explique comment activer les chambres fortes dans la gestion des identités (IdM) en installant le composant Système de certificats (CS) de l'Autorité de recouvrement des clés (KRA) sur un serveur IdM spécifique.

Conditions préalables

- Vous êtes connecté en tant que **root** sur le serveur IdM.
- Une autorité de certification IdM est installée sur le serveur IdM.
- Vous avez les références **Directory Manager**.

Procédure

- Installer l'ARK :

```
# ipa-kra-install
```

IMPORTANT

Vous pouvez installer le premier ARK d'un cluster IdM sur un réplica caché. Toutefois, l'installation d'ARK supplémentaires nécessite l'activation temporaire du réplica caché avant d'installer le clone de l'ARK sur un réplica non caché. Vous pouvez ensuite masquer à nouveau le réplica initialement masqué.

NOTE

Pour que le service de coffre-fort soit hautement disponible et résilient, installez l'ARK sur deux serveurs IdM ou plus. La maintenance de plusieurs serveurs KRA permet d'éviter les pertes de données.

Ressources supplémentaires

- Voir Rétrograder ou promouvoir des répliques cachées.
- Voir Le mode réplique caché.

CHAPITRE 23. UTILISER ANSIBLE POUR GÉRER LES COFFRES-FORTS DES UTILISATEURS IDM : STOCKER ET RÉCUPÉRER LES SECRETS

Ce chapitre décrit comment gérer les coffres-forts des utilisateurs dans Identity Management à l'aide du module Ansible **vault**. Plus précisément, il décrit comment un utilisateur peut utiliser les playbooks Ansible pour effectuer les trois actions consécutives suivantes :

- Créer un coffre-fort d'utilisateur dans IdM .
- Conserver un secret dans la chambre forte .
- Récupérer un secret dans le coffre .

L'utilisateur peut effectuer le stockage et l'extraction à partir de deux clients IdM différents.

Conditions préalables

- Le composant du système de certificats de l'autorité de récupération des clés (KRA) a été installé sur un ou plusieurs serveurs de votre domaine IdM. Pour plus de détails, voir Installation de l'autorité de recouvrement des clés dans IdM.

23.1. ASSURER LA PRÉSENCE D'UN COFFRE-FORT UTILISATEUR STANDARD DANS IDM À L'AIDE D'ANSIBLE

Cette section montre comment un utilisateur de la gestion des identités (IdM) peut utiliser un playbook Ansible pour créer un conteneur d'espace de stockage avec un ou plusieurs espaces de stockage privés pour stocker en toute sécurité des informations sensibles. Dans l'exemple utilisé dans la procédure ci-dessous, l'utilisateur **idm_user** crée un espace de stockage de type standard nommé **my_vault**. Le type de coffre-fort standard garantit que **idm_user** n'aura pas à s'authentifier pour accéder au fichier. **idm_user** pourra récupérer le fichier à partir de n'importe quel client IdM auquel l'utilisateur est connecté.

Conditions préalables

- Vous avez installé le paquet ansible-freeipa sur le contrôleur Ansible, c'est-à-dire l'hôte sur lequel vous exécutez les étapes de la procédure.
- Vous connaissez le mot de passe de **idm_user**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ touch inventory.file
```

3. Ouvrez **inventory.file** et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

4. Faites une copie du fichier **ensure-standard-vault-is-present.yml** Ansible playbook. Par exemple :

```
$ cp ensure-standard-vault-is-present.yml ensure-standard-vault-is-present-copy.yml
```

5. Ouvrez le fichier **ensure-standard-vault-is-present-copy.yml** pour le modifier.

6. Adaptez le fichier en définissant les variables suivantes dans la section **ipavault** task :

   - Fixer la variable **ipaadmin_principal** à **idm_user**.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de **idm_user**.
   - Fixer la variable **user** à **idm_user**.
   - Fixer la variable **name** à **my_vault**.
   - Fixer la variable **vault_type** à **standard**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      vault_type: standard
```

7. Enregistrer le fichier.

8. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-standard-vault-is-present-copy.yml
```

23.2. ARCHIVAGE D'UN SECRET DANS UN COFFRE-FORT UTILISATEUR STANDARD DANS IDM À L'AIDE D'ANSIBLE

Cette section montre comment un utilisateur de la gestion des identités (IdM) peut utiliser un playbook Ansible pour stocker des informations sensibles dans un coffre-fort personnel. Dans l'exemple utilisé, l'utilisateur **idm_user** archive un fichier contenant des informations sensibles nommé **password.txt** dans un coffre-fort nommé **my_vault**.

Conditions préalables

- Vous avez installé le paquet ansible-freeipa sur le contrôleur Ansible, c'est-à-dire l'hôte sur lequel vous exécutez les étapes de la procédure.
- Vous connaissez le mot de passe de **idm_user**.
- **idm_user** est le propriétaire, ou au moins un membre utilisateur de **my_vault**.
- Vous avez accès à **password.txt**, le secret que vous voulez archiver dans **my_vault**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **data-archive-in-symmetric-vault.yml** Ansible playbook, mais remplacez "symmetric" par "standard". Par exemple :

```
$ cp data-archive-in-symmetric-vault.yml data-archive-in-standard-vault-copy.yml
```

4. Ouvrez le fichier **data-archive-in-standard-vault-copy.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes dans la section **ipavault** task :
   - Fixer la variable **ipaadmin_principal** à **idm_user**.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de **idm_user**.
   - Fixer la variable **user** à **idm_user**.
   - Fixer la variable **name** à **my_vault**.
   - Définissez la variable **in** avec le chemin d'accès complet au fichier contenant des informations sensibles.
   - Fixer la variable **action** à **member**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
```

```
- ipavault:
  ipaadmin_principal: idm_user
  ipaadmin_password: idm_user_password
  user: idm_user
  name: my_vault
  in: /usr/share/doc/ansible-freeipa/playbooks/vault/password.txt
  action: member
```

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-archive-in-standard-vault-copy.yml
```

23.3. RÉCUPÉRER UN SECRET À PARTIR D'UN COFFRE-FORT D'UTILISATEUR STANDARD DANS IDM EN UTILISANT ANSIBLE

Cette section montre comment un utilisateur de la gestion des identités (IdM) peut utiliser un playbook Ansible pour récupérer un secret dans le coffre-fort personnel de l'utilisateur. Dans l'exemple utilisé dans la procédure ci-dessous, l'utilisateur **idm_user** récupère un fichier contenant des données sensibles à partir d'un coffre-fort de type standard nommé **my_vault** sur un client IdM nommé **host01**. **idm_user** n'a pas besoin de s'authentifier pour accéder au fichier. **idm_user** peut utiliser Ansible pour récupérer le fichier à partir de n'importe quel client IdM sur lequel Ansible est installé.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de **idm_user**.
- **idm_user** est le propriétaire de **my_vault**.
- **idm_user** a stocké un secret dans **my_vault**.
- Ansible peut écrire dans le répertoire de l'hôte IdM dans lequel vous souhaitez récupérer le secret.
- **idm_user** peut lire le répertoire de l'hôte IdM dans lequel vous souhaitez récupérer le secret.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. Ouvrez votre fichier d'inventaire et mentionnez, dans une section clairement définie, le client IdM sur lequel vous souhaitez récupérer le secret. Par exemple, pour demander à Ansible de récupérer le secret sur **host01.idm.example.com**, entrez :

```
[ipahost]
host01.idm.example.com
```

3. Effectuez une copie du fichier **retrive-data-symmetric-vault.yml** Ansible playbook. Remplacez "symétrique" par "standard". Par exemple :

```
$ cp retrive-data-symmetric-vault.yml retrieve-data-standard-vault.yml-copy.yml
```

4. Ouvrez le fichier **retrieve-data-standard-vault.yml-copy.yml** pour le modifier.

5. Adapter le fichier en fixant la variable **hosts** à **ipahost**.

6. Adaptez le fichier en définissant les variables suivantes dans la section **ipavault** task :

   - Fixer la variable **ipaadmin_principal** à **idm_user**.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de **idm_user**.
   - Fixer la variable **user** à **idm_user**.
   - Fixer la variable **name** à **my_vault**.
   - Définissez la variable **out** avec le chemin complet du fichier dans lequel vous souhaitez exporter le secret.
   - Fixer la variable **state** à **retrieved**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Tests
  hosts: ipahost
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      out: /tmp/password_exported.txt
      state: retrieved
```

7. Enregistrer le fichier.

8. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-data-standard-vault.yml-copy.yml
```

Verification steps

1. **SSH** à **host01** comme **user01**:

```
$ ssh user01@host01.idm.example.com
```

2. Afficher le fichier spécifié par la variable **out** dans le fichier playbook Ansible :

```
$ vim /tmp/password_exported.txt
```

Vous pouvez maintenant voir le secret exporté.

- Pour plus d'informations sur l'utilisation d'Ansible pour gérer les coffres-forts IdM et les secrets d'utilisateur, ainsi que sur les variables des playbooks, consultez le fichier README-vault.md Markdown disponible dans le répertoire **/usr/share/doc/ansible-freeipa/** et les exemples de playbooks disponibles dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault/**.

CHAPITRE 24. UTILISER ANSIBLE POUR GÉRER LES COFFRES-FORTS DES SERVICES IDM : STOCKER ET RÉCUPÉRER LES SECRETS

Cette section montre comment un administrateur peut utiliser le module **ansible-freeipa vault** pour stocker en toute sécurité un secret de service dans un emplacement centralisé. Le coffre-fort utilisé dans l'exemple est asymétrique, ce qui signifie que pour l'utiliser, l'administrateur doit effectuer les étapes suivantes :

1. Générez une clé privée en utilisant, par exemple, l'utilitaire **openssl**.
2. Générer une clé publique à partir de la clé privée.

Le secret de service est crypté avec la clé publique lorsqu'un administrateur l'archive dans le coffre-fort. Ensuite, une instance de service hébergée sur une machine spécifique du domaine récupère le secret à l'aide de la clé privée. Seuls le service et l'administrateur sont autorisés à accéder au secret.

Si le secret est compromis, l'administrateur peut le remplacer dans le coffre-fort du service, puis le redistribuer aux instances de service individuelles qui n'ont pas été compromises.

Conditions préalables

- Le composant du système de certificats de l'autorité de récupération des clés (KRA) a été installé sur un ou plusieurs serveurs de votre domaine IdM. Pour plus de détails, voir Installation de l'autorité de recouvrement des clés dans IdM.

Cette section comprend ces procédures :

- Assurer la présence d'un coffre-fort de service asymétrique dans IdM à l'aide d'Ansible
- Stocker un secret de service IdM dans un coffre-fort asymétrique à l'aide d'Ansible
- Récupérer un secret de service pour un service IdM en utilisant Ansible
- Changer le secret du coffre d'un service IdM en cas de compromission en utilisant Ansible

Dans les procédures :

- **admin** est l'administrateur qui gère le mot de passe du service.
- **private-key-to-an-externally-signed-certificate.pem** est le fichier contenant le secret du service, dans ce cas une clé privée d'un certificat signé en externe. Ne confondez pas cette clé privée avec la clé privée utilisée pour récupérer le secret dans le coffre-fort.
- **secret_vault** est le coffre-fort créé pour stocker le secret de service.
- **HTTP/webserver1.idm.example.com** est le service propriétaire de la chambre forte.
- **HTTP/webserver2.idm.example.com** et **HTTP/webserver3.idm.example.com** sont les services aux membres de la voûte.
- **service-public.pem** est la clé publique du service utilisée pour crypter le mot de passe stocké dans **password_vault**.
- **service-private.pem** est la clé privée du service utilisée pour décrypter le mot de passe stocké dans **secret_vault**.

24.1. ASSURER LA PRÉSENCE D'UN COFFRE-FORT DE SERVICE ASYMÉTRIQUE DANS IDM À L'AIDE D'ANSIBLE

Cette section montre comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour créer un conteneur de coffre-fort de service avec un ou plusieurs coffres-forts privés pour stocker en toute sécurité des informations sensibles. Dans l'exemple utilisé dans la procédure ci-dessous, l'administrateur crée un coffre-fort asymétrique nommé **secret_vault**. Cela garantit que les membres de l'espace de stockage doivent s'authentifier à l'aide d'une clé privée pour récupérer le secret dans l'espace de stockage. Les membres du coffre-fort pourront récupérer le fichier à partir de n'importe quel client IdM.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de **IdM administrator**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. Obtenir la clé publique de l'instance de service. Par exemple, en utilisant l'utilitaire **openssl**:
   a. Générer la clé privée **service-private.pem**.

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....................................+++
e is 65537 (0x10001)
```

   b. Générer la clé publique **service-public.pem** à partir de la clé privée.

```
$ openssl rsa -in service-private.pem -out service-public.pem -pubout
writing RSA key
```

3. Facultatif : Créez un fichier d'inventaire s'il n'existe pas, par exemple **inventory.file**:

```
$ touch inventory.file
```

4. Ouvrez votre fichier d'inventaire et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

5. Faites une copie du fichier **ensure-asymmetric-vault-is-present.yml** Ansible playbook. Par exemple :

```
$ cp ensure-asymmetric-vault-is-present.yml ensure-asymmetric-service-vault-is-present-copy.yml
```

6. Ouvrez le fichier **ensure-asymmetric-vault-is-present-copy.yml** pour le modifier.

7. Ajoutez une tâche qui copie la clé publique **service-public.pem** du contrôleur Ansible vers le serveur **server.idm.example.com**.

8. Modifiez le reste du fichier en définissant les variables suivantes dans la section **ipavault** task :

   - Fixer la variable **ipaadmin_password** au mot de passe de l'administrateur de l'IdM.
   - Définissez le nom de la chambre forte à l'aide de la variable **name**, par exemple **secret_vault**.
   - Fixer la variable **vault_type** à **asymmetric**.
   - Définissez la variable **service** comme étant le principal du service propriétaire de la chambre forte, par exemple **HTTP/webserver1.idm.example.com**.
   - Réglez l'adresse **public_key_file** sur l'emplacement de votre clé publique.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Copy public key to ipaserver.
    copy:
      src: /path/to/service-public.pem
      dest: /usr/share/doc/ansible-freeipa/playbooks/vault/service-public.pem
      mode: 0600
  - name: Add data to vault, from a LOCAL file.
    ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      vault_type: asymmetric
      service: HTTP/webserver1.idm.example.com
      public_key_file: /usr/share/doc/ansible-freeipa/playbooks/vault/service-public.pem
```

9. Enregistrer le fichier.

10. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-asymmetric-service-vault-is-present-copy.yml
```

24.2. AJOUTER DES SERVICES MEMBRES À UN COFFRE-FORT ASYMÉTRIQUE EN UTILISANT ANSIBLE

Cette section montre comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour ajouter des services membres à un coffre-fort de service afin qu'ils puissent tous récupérer le secret stocké dans le coffre-fort. Dans l'exemple utilisé dans la procédure ci-dessous, l'administrateur IdM ajoute les principaux services **HTTP/webserver2.idm.example.com** et **HTTP/webserver3.idm.example.com** au coffre-fort **secret_vault** qui appartient à **HTTP/webserver1.idm.example.com**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de **IdM administrator**.
- Vous avez créé un coffre-fort asymétrique pour stocker le secret de service.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. Facultatif : Créez un fichier d'inventaire s'il n'existe pas, par exemple **inventory.file**:

```
$ touch inventory.file
```

3. Ouvrez votre fichier d'inventaire et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

4. Faites une copie du fichier **data-archive-in-asymmetric-vault.yml** Ansible playbook. Par exemple :

```
$ cp data-archive-in-asymmetric-vault.yml add-services-to-an-asymmetric-vault.yml
```

5. Ouvrez le fichier **data-archive-in-asymmetric-vault-copy.yml** pour le modifier.
6. Modifiez le fichier en définissant les variables suivantes dans la section **ipavault** task :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'administrateur de l'IdM.
   - Attribuez à la variable **name** le nom de la chambre forte, par exemple **secret_vault**.
   - Définissez la variable **service** comme étant le propriétaire du service de la chambre forte, par exemple **HTTP/webserver1.idm.example.com**.
   - Définissez les services qui doivent avoir accès au secret de l'espace de stockage à l'aide de la variable **services**.
   - Fixer la variable **action** à **member**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      services:
      - HTTP/webserver2.idm.example.com
      - HTTP/webserver3.idm.example.com
      action: member
```

7. Enregistrer le fichier.
8. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file add-services-to-an-asymmetric-vault.yml
```

24.3. STOCKER UN SECRET DE SERVICE IDM DANS UN COFFRE-FORT ASYMÉTRIQUE À L'AIDE D'ANSIBLE

Cette section montre comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour stocker un secret dans un coffre-fort de service afin qu'il puisse être récupéré ultérieurement par le service. Dans l'exemple utilisé dans la procédure ci-dessous, l'administrateur stocke un fichier **PEM** avec le secret dans un coffre-fort asymétrique nommé **secret_vault**. Cela garantit que le service devra s'authentifier à l'aide d'une clé privée pour récupérer le secret dans le coffre-fort. Les membres du coffre-fort pourront récupérer le fichier à partir de n'importe quel client IdM.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de **IdM administrator**.
- Vous avez créé un coffre-fort asymétrique pour stocker le secret de service.
- Le secret est stocké localement sur le contrôleur Ansible, par exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. Facultatif : Créez un fichier d'inventaire s'il n'existe pas, par exemple **inventory.file**:

```
$ touch inventory.file
```

3. Ouvrez votre fichier d'inventaire et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

4. Faites une copie du fichier **data-archive-in-asymmetric-vault.yml** Ansible playbook. Par exemple :

```
$ cp data-archive-in-asymmetric-vault.yml data-archive-in-asymmetric-vault-copy.yml
```

5. Ouvrez le fichier **data-archive-in-asymmetric-vault-copy.yml** pour le modifier.
6. Modifiez le fichier en définissant les variables suivantes dans la section **ipavault** task :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'administrateur de l'IdM.
   - Attribuez à la variable **name** le nom de la chambre forte, par exemple **secret_vault**.
   - Définissez la variable **service** comme étant le propriétaire du service de la chambre forte, par exemple **HTTP/webserver1.idm.example.com**.

- Définissez la variable **in** à **"{{ lookup('file', 'private-key-to-an-externally-signed-certificate.pem') | b64encode }}"**, ce qui garantit qu'Ansible récupère le fichier contenant la clé privée dans le répertoire de travail du contrôleur Ansible plutôt que sur le serveur IdM.

- Fixer la variable **action** à **member**.
  Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      in: "{{ lookup('file', 'private-key-to-an-externally-signed-certificate.pem') | b64encode }}"
      action: member
```

7. Enregistrer le fichier.

8. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-archive-in-asymmetric-vault-copy.yml
```

24.4. RÉCUPÉRER UN SECRET DE SERVICE POUR UN SERVICE IDM EN UTILISANT ANSIBLE

Cette section montre comment un utilisateur de la gestion des identités (IdM) peut utiliser un livre de jeu Ansible pour récupérer un secret dans un coffre-fort de service au nom du service. Dans l'exemple utilisé dans la procédure ci-dessous, l'exécution du livre de jeu récupère un fichier **PEM** avec le secret d'un coffre-fort asymétrique nommé **secret_vault**, et le stocke à l'emplacement spécifié sur tous les hôtes répertoriés dans le fichier d'inventaire Ansible sous le nom **ipaservers**.

Les services s'authentifient auprès de l'IdM à l'aide de keytabs, et ils s'authentifient auprès du coffre-fort à l'aide d'une clé privée. Vous pouvez récupérer le fichier au nom du service à partir de n'importe quel client IdM sur lequel **ansible-freeipa** est installé.

Conditions préalables

- 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**.

- Vous connaissez le mot de passe de l'administrateur IdM.
- Vous avez créé un coffre-fort asymétrique pour stocker le secret de service.
- Vous avez archivé le secret dans la chambre forte .
- Vous avez stocké la clé privée utilisée pour récupérer le secret du coffre-fort du service dans l'emplacement spécifié par la variable **private_key_file** sur le contrôleur Ansible.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. Facultatif : Créez un fichier d'inventaire s'il n'existe pas, par exemple **inventory.file**:

```
$ touch inventory.file
```

3. Ouvrez votre fichier d'inventaire et définissez les hôtes suivants :
    - Définissez votre serveur IdM dans la section **[ipaserver]**.
    - Définissez les hôtes sur lesquels vous souhaitez récupérer le secret dans la section **[webservers]**. Par exemple, pour demander à Ansible de récupérer le secret sur **webserver1.idm.example.com**, **webserver2.idm.example.com**, et **webserver3.idm.example.com**, entrez :

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

[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
webserver3.idm.example.com
```

4. Faites une copie du fichier **retrieve-data-asymmetric-vault.yml** Ansible playbook. Par exemple :

```
$ cp retrieve-data-asymmetric-vault.yml retrieve-data-asymmetric-vault-copy.yml
```

5. Ouvrez le fichier **retrieve-data-asymmetric-vault-copy.yml** pour le modifier.
6. Modifiez le fichier en définissant les variables suivantes dans la section **ipavault** task :
    - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
    - Attribuez à la variable **name** le nom de la chambre forte, par exemple **secret_vault**.
    - Définissez la variable **service** comme étant le propriétaire du service de la chambre forte, par exemple **HTTP/webserver1.idm.example.com**.

- Définissez la variable **private_key_file** à l'emplacement de la clé privée utilisée pour récupérer le secret du coffre-fort de service.

- Définissez la variable **out** à l'emplacement du serveur IdM où vous souhaitez récupérer le secret **private-key-to-an-externally-signed-certificate.pem**, par exemple le répertoire de travail actuel.

- Fixer la variable **action** à **member**.
  Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Retrieve data from the service vault
    ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      vault_type: asymmetric
      private_key: "{{ lookup('file', 'service-private.pem') | b64encode }}"
      out: private-key-to-an-externally-signed-certificate.pem
      state: retrieved
```

7. Ajouter une section au playbook qui récupère le fichier de données du serveur IdM vers le contrôleur Ansible :

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false
  tasks:
[...]
  - name: Retrieve data file
    fetch:
      src: private-key-to-an-externally-signed-certificate.pem
      dest: ./
      flat: yes
      mode: 0600
```

8. Ajoutez une section au playbook qui transfère le fichier **private-key-to-an-externally-signed-certificate.pem** récupéré depuis le contrôleur Ansible vers les serveurs web répertoriés dans la section **webservers** du fichier d'inventaire :

```
---
- name: Send data file to webservers
  become: no
  gather_facts: no
  hosts: webservers
```

```
tasks:
- name: Send data to webservers
  copy:
    src: private-key-to-an-externally-signed-certificate.pem
    dest: /etc/pki/tls/private/httpd.key
    mode: 0444
```

9. Enregistrer le fichier.
10. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-data-asymmetric-vault-copy.yml
```

24.5. CHANGER LE SECRET DU COFFRE D'UN SERVICE IDM EN CAS DE COMPROMISSION EN UTILISANT ANSIBLE

Cette section montre comment un administrateur de gestion des identités (IdM) peut réutiliser un playbook Ansible pour modifier le secret stocké dans un coffre-fort de service lorsqu'une instance de service a été compromise. Le scénario de l'exemple suivant suppose que sur **webserver3.idm.example.com**, le secret récupéré a été compromis, mais pas la clé du coffre-fort asymétrique stockant le secret. Dans cet exemple, l'administrateur réutilise les playbooks Ansible utilisés pour stocker un secret dans un coffre-fort as ymétrique et pour récupérer un secret du coffre-fort asymétrique sur les hôtes IdM. Au début de la procédure, l'administrateur IdM stocke un nouveau fichier **PEM** avec un nouveau secret dans le coffre-fort asymétrique, adapte le fichier d'inventaire de manière à ne pas récupérer le nouveau secret sur le serveur web compromis, **webserver3.idm.example.com**, puis réexécute les deux procédures.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de **IdM administrator**.
- Vous avez créé un coffre-fort asymétrique pour stocker le secret de service.
- Vous avez généré une nouvelle clé **httpd** pour les services web fonctionnant sur les hôtes IdM afin de remplacer l'ancienne clé compromise.
- La nouvelle clé **httpd** est stockée localement sur le contrôleur Ansible, par exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que les hôtes suivants sont définis correctement :

   - Le serveur IdM dans la section **[ipaserver]**.
   - Les hôtes sur lesquels vous souhaitez récupérer le secret dans la section **[webservers]**. Par exemple, pour demander à Ansible de récupérer le secret sur **webserver1.idm.example.com** et **webserver2.idm.example.com**, entrez :

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

[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
```

IMPORTANT

Assurez-vous que la liste ne contient pas le serveur web compromis, dans l'exemple actuel **webserver3.idm.example.com**.

3. Ouvrez le fichier **data-archive-in-asymmetric-vault-copy.yml** pour le modifier.
4. Modifiez le fichier en définissant les variables suivantes dans la section **ipavault** task :

   - Fixer la variable **ipaadmin_password** au mot de passe de l'administrateur de l'IdM.
   - Attribuez à la variable **name** le nom de la chambre forte, par exemple **secret_vault**.
   - Définissez la variable **service** comme étant le propriétaire du service de la chambre forte, par exemple **HTTP/webserver.idm.example.com**.
   - Définissez la variable **in** à **"{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem') | b64encode }}"**, ce qui garantit qu'Ansible récupère le fichier contenant la clé privée dans le répertoire de travail du contrôleur Ansible plutôt que sur le serveur IdM.
   - Fixer la variable **action** à **member**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver.idm.example.com
```

```
      in: "{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem') | b64encode
}}"
      action: member
```

5. Enregistrer le fichier.
6. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-
archive-in-asymmetric-vault-copy.yml
```

7. Ouvrez le fichier **retrieve-data-asymmetric-vault-copy.yml** pour le modifier.
8. Modifiez le fichier en définissant les variables suivantes dans la section **ipavault** task :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Attribuez à la variable **name** le nom de la chambre forte, par exemple **secret_vault**.
   - Définissez la variable **service** comme étant le propriétaire du service de la chambre forte, par exemple **HTTP/webserver1.idm.example.com**.
   - Définissez la variable **private_key_file** à l'emplacement de la clé privée utilisée pour récupérer le secret du coffre-fort de service.
   - Définissez la variable **out** à l'emplacement du serveur IdM où vous souhaitez récupérer le secret **new-private-key-to-an-externally-signed-certificate.pem**, par exemple le répertoire de travail actuel.
   - Fixer la variable **action** à **member**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Retrieve data from the service vault
    ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      vault_type: asymmetric
      private_key: "{{ lookup('file', 'service-private.pem') | b64encode }}"
      out: new-private-key-to-an-externally-signed-certificate.pem
      state: retrieved
```

9. Ajouter une section au playbook qui récupère le fichier de données du serveur IdM vers le contrôleur Ansible :

```
---
```

```
- name: Retrieve data from vault
  hosts: ipaserver
  become: yes
  gather_facts: false
  tasks:
[...]
  - name: Retrieve data file
    fetch:
      src: new-private-key-to-an-externally-signed-certificate.pem
      dest: ./
      flat: yes
      mode: 0600
```

10. Ajoutez une section au playbook qui transfère le fichier **new-private-key-to-an-externally-signed-certificate.pem** récupéré depuis le contrôleur Ansible vers les serveurs web répertoriés dans la section **webservers** du fichier d'inventaire :

```
---
- name: Send data file to webservers
  become: yes
  gather_facts: no
  hosts: webservers
  tasks:
  - name: Send data to webservers
    copy:
      src: new-private-key-to-an-externally-signed-certificate.pem
      dest: /etc/pki/tls/private/httpd.key
      mode: 0444
```

11. Enregistrer le fichier.

12. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-data-asymmetric-vault-copy.yml
```

24.6. RESSOURCES SUPPLÉMENTAIRES

- Voir le fichier README-vault.md Markdown dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault/**.

CHAPITRE 25. ASSURER LA PRÉSENCE ET L'ABSENCE DE SERVICES DANS IDM À L'AIDE D'ANSIBLE

Avec le module Ansible **service**, l'administrateur de la gestion des identités (IdM) peut s'assurer que des services spécifiques qui ne sont pas natifs de l'IdM sont présents ou absents de l'IdM. Par exemple, vous pouvez utiliser le module **service** pour :

- Vérifier qu'un service installé manuellement est présent sur un client IdM et installer automatiquement ce service s'il est absent. Pour plus de détails, voir :
  - Assurer la présence d'un service HTTP dans IdM sur un client IdM.
  - Assurer la présence d'un service HTTP dans IdM sur un client non-IdM.
  - Assurer la présence d'un service HTTP sur un client IdM sans DNS.
- Vérifier qu'un service enrôlé dans IdM a un certificat attaché et installer automatiquement ce certificat s'il est absent. Pour plus de détails, voir :
- Assurer la présence d'un certificat signé en externe dans une entrée de service IdM.
- Autoriser les utilisateurs et les hôtes IdM à récupérer et à créer la table de clés du service. Pour plus de détails, voir :
  - Permettre aux utilisateurs, groupes, hôtes ou groupes d'hôtes de l'IdM de créer un keytab d'un service.
  - Permettre aux utilisateurs, groupes, hôtes ou groupes d'hôtes de l'IdM de récupérer un keytab d'un service.
- Permet aux utilisateurs et hôtes IdM d'ajouter un alias Kerberos à un service. Pour plus de détails, voir :
  - Assurer la présence d'un alias principal Kerberos pour un service.
- Vérifier qu'un service n'est pas présent sur un client IdM et supprimer automatiquement ce service s'il est présent. Pour plus de détails, voir :
  - Garantir l'absence d'un service HTTP dans IdM sur un client IdM.

25.1. ASSURER LA PRÉSENCE D'UN SERVICE HTTP DANS IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

Cette section décrit comment assurer la présence d'un serveur HTTP dans IdM à l'aide d'un playbook Ansible.

Conditions préalables

- Le système qui héberge le service HTTP est un client IdM.
- Vous avez le mot de passe de l'administrateur IdM.

Procédure

1. Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ touch inventory.file
```

2. Ouvrez le site **inventory.file** et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present.yml** Ansible playbook. Par exemple :

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present.yml
/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-copy.yml
```

4. Ouvrez le fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-copy.yml** Ansible playbook pour l'éditer :

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure service is present
  - ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/client.idm.example.com
```

5. Adapter le dossier :

   - Modifier le mot de passe de l'administrateur IdM défini par la variable **ipaadmin_password**.
   - Changez le nom de votre client IdM sur lequel le service HTTP est exécuté, comme défini par la variable **name** de la tâche **ipaservice**.

6. Save and exit the file.

7. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-is-present-copy.yml
```

Verification steps

1. Se connecter à l'interface Web IdM en tant qu'administrateur IdM.
2. Naviguez jusqu'à **Identity → Services**.

Si **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** figure dans la liste **Services**, le playbook Ansible a été ajouté avec succès à IdM.

Ressources supplémentaires

- Pour sécuriser la communication entre le serveur HTTP et les clients du navigateur, voir l'ajout du cryptage TLS à un serveur HTTP Apache.
- Pour demander un certificat pour le service HTTP, voir la procédure décrite dans la section Obtention d'un certificat IdM pour un service à l'aide de certmonger .

25.2. ASSURER LA PRÉSENCE D'UN SERVICE HTTP DANS IDM SUR UN CLIENT NON-IDM EN UTILISANT UN PLAYBOOK ANSIBLE

Cette section décrit comment assurer la présence d'un serveur HTTP dans IdM sur un hôte qui n'est pas un client IdM à l'aide d'un playbook Ansible. En ajoutant le serveur HTTP à IdM, vous ajoutez également l'hôte à IdM.

Conditions préalables

- Vous avez installé un service HTTP sur votre hôte.
- L'hôte sur lequel vous avez configuré HTTP n'est pas un client IdM. Sinon, suivez les étapes de l'enrôlement du service HTTP dans IdM .
- Vous avez le mot de passe de l'administrateur IdM.
- L'enregistrement DNS A - ou l'enregistrement AAAA si IPv6 est utilisé - de l'hôte est disponible.

Procédure

1. Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ touch inventory.file
```

2. Ouvrez le site **inventory.file** et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check.yml** Ansible playbook. Par exemple :

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check-copy.yml
```

4. Ouvrez le fichier copié, **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-without-host-check-copy.yml**, pour l'éditer. Localisez les variables **ipaadmin_password** et **name** dans la tâche **ipaservice**:

```
---
```

```
- name: Playbook to manage IPA service.
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure service is present
  - ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/www2.example.com
      skip_host_check: yes
```

5. Adapter le dossier :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Définissez la variable **name** avec le nom de l'hôte sur lequel le service HTTP est exécuté.
6. Save and exit the file.
7. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-is-present-without-host-check-copy.yml
```

Verification steps

1. Se connecter à l'interface Web IdM en tant qu'administrateur IdM.
2. Naviguez jusqu'à **Identity → Services**.

Vous pouvez maintenant voir **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** dans la liste **Services**.

Ressources supplémentaires

- Pour sécuriser la communication, voir l'ajout du cryptage TLS à un serveur HTTP Apache .

25.3. ASSURER LA PRÉSENCE D'UN SERVICE HTTP SUR UN CLIENT IDM SANS DNS À L'AIDE D'UN PLAYBOOK ANSIBLE

Cette section décrit comment assurer la présence d'un serveur HTTP fonctionnant sur un client IdM qui n'a pas d'entrée DNS à l'aide d'un playbook Ansible. Le scénario sous-entendu est que l'hôte IdM n'a pas d'entrée DNS A disponible - ou pas d'entrée DNS AAAA si IPv6 est utilisé à la place d'IPv4.

Conditions préalables

- Le système qui héberge le service HTTP est inscrit dans l'IdM.

- L'enregistrement DNS A ou DNS AAAA pour l'hôte peut ne pas exister. Sinon, si l'enregistrement DNS de l'hôte existe, suivez la procédure décrite dans la section Assurer la présence d'un service HTTP dans IdM à l'aide d'un livre de jeu Ansible.
- Vous avez le mot de passe de l'administrateur IdM.

Procédure

1. Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ touch inventory.file
```

2. Ouvrez le site **inventory.file** et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force.yml** Ansible playbook. Par exemple :

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml
```

4. Ouvrez le fichier copié, **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml**, pour l'éditer. Localisez les variables **ipaadmin_password** et **name** dans la tâche **ipaservice**:

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure service is present
  - ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/ihavenodns.info
      force: yes
```

5. Adapter le dossier :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Définissez la variable **name** avec le nom de l'hôte sur lequel le service HTTP est exécuté.
6. Save and exit the file.
7. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-is-present-with-host-force-copy.yml
```

Verification steps

1. Se connecter à l'interface Web IdM en tant qu'administrateur IdM.
2. Naviguez jusqu'à **Identity → Services**.

Vous pouvez maintenant voir **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** dans la liste **Services**.

Ressources supplémentaires

- Pour sécuriser la communication, voir l'ajout du cryptage TLS à un serveur HTTP Apache .

25.4. ASSURER LA PRÉSENCE D'UN CERTIFICAT SIGNÉ EN EXTERNE DANS UNE ENTRÉE DE SERVICE IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

Cette section décrit comment utiliser le module **ansible-freeipa service** pour s'assurer qu'un certificat émis par une autorité de certification (AC) externe est attaché à l'entrée IdM du service HTTP. Le fait que le certificat d'un service HTTP soit signé par une autorité de certification externe plutôt que par l'autorité de certification IdM est particulièrement utile si votre autorité de certification IdM utilise un certificat auto-signé.

Conditions préalables

- Vous avez installé un service HTTP sur votre hôte.
- Vous avez inscrit le service HTTP dans IdM .
- Vous avez le mot de passe de l'administrateur IdM.
- Vous disposez d'un certificat signé en externe dont le Subject correspond au principal du service HTTP.

Procédure

1. Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ touch inventory.file
```

2. Ouvrez le site **inventory.file** et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present.yml**, par exemple :

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present-copy.yml
```

4. Facultatif : si le certificat est au format PEM (Privacy Enhanced Mail), convertissez-le au format DER (Distinguished Encoding Rules) pour faciliter sa manipulation par l'interface de ligne de commande (CLI) :

```
$ openssl x509 -outform der -in cert1.pem -out cert1.der
```

5. Décodez le fichier **DER** sur la sortie standard à l'aide de la commande **base64**. Utilisez l'option **-w0** pour désactiver le wrapping :

```
$ base64 cert1.der -w0
MIIC/zCCAeegAwIBAgIUV74O+4kXeg21o4vxfRRtyJm...
```

6. Copier le certificat de la sortie standard dans le presse-papiers.

7. Ouvrez le fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present-copy.yml** pour l'éditer et visualiser son contenu :

```
---
- name: Service certificate present.
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure service certificate is present
  - ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/www.example.com
      certificate: |
        - MIICBjCCAW8CFHnm32VcXaUDGfEGdDL/...
      [...]
      action: member
      state: present
```

8. Adapter le dossier :

   - Remplacez le certificat, défini à l'aide de la variable **certificate**, par le certificat que vous avez copié depuis l'interface de programmation. Notez que si vous utilisez la variable **certificate:** avec le caractère de pipe "|" comme indiqué, vous pouvez entrer le certificat DE CETTE MANIERE plutôt que de devoir l'entrer sur une seule ligne. Cela facilite la lecture du certificat.

   - Modifier le mot de passe de l'administrateur IdM, défini par la variable **ipaadmin_password**.

   - Modifiez le nom du client IdM sur lequel le service HTTP est exécuté, défini par la variable **name**.

   - Modifier toute autre variable pertinente.

9. Save and exit the file.

10. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-member-certificate-present-copy.yml
```

Verification steps

1. Se connecter à l'interface Web IdM en tant qu'administrateur IdM.

2. Naviguez jusqu'à **Identity → Services**.

3. Cliquez sur le nom du service avec le nouveau certificat ajouté, par exemple **HTTP/client.idm.example.com**.

Dans la section **Service Certificate** à droite, vous pouvez maintenant voir le nouveau certificat ajouté.

25.5. UTILISATION D'UN PLAYBOOK ANSIBLE POUR PERMETTRE AUX UTILISATEURS, GROUPES, HÔTES OU GROUPES D'HÔTES IDM DE CRÉER UN KEYTAB D'UN SERVICE

Un fichier keytab est un fichier contenant des paires de mandants Kerberos et de clés cryptées. Les fichiers keytab sont généralement utilisés pour permettre aux scripts de s'authentifier automatiquement à l'aide de Kerberos, sans nécessiter d'interaction humaine ni d'accès au mot de passe stocké dans un fichier en texte clair. Le script est alors en mesure d'utiliser les informations d'identification acquises pour accéder aux fichiers stockés sur un système distant.

En tant qu'administrateur de la gestion des identités (IdM), vous pouvez permettre à d'autres utilisateurs de récupérer ou même de créer un keytab pour un service fonctionnant dans IdM. En autorisant des utilisateurs et des groupes d'utilisateurs spécifiques à créer des keytabs, vous pouvez leur déléguer l'administration du service sans partager le mot de passe de l'administrateur IdM. Cette délégation permet une administration plus fine du système.

Cette section explique comment autoriser des utilisateurs, des groupes d'utilisateurs, des hôtes et des groupes d'hôtes spécifiques de l'IdM à créer un keytab pour le service HTTP exécuté sur un client IdM. En particulier, elle décrit comment vous pouvez autoriser l'utilisateur IdM **user01** à créer un keytab pour le service HTTP s'exécutant sur un client IdM nommé **client.idm.example.com**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.

- Vous avez inscrit le service HTTP dans IdM .
- Le système qui héberge le service HTTP est un client IdM.
- Les utilisateurs et groupes d'utilisateurs de l'IdM que vous souhaitez autoriser à créer le fichier clé existent dans l'IdM.
- Les hôtes et groupes d'hôtes de l'IdM que vous souhaitez autoriser à créer le fichier clé existent dans l'IdM.

Procédure

1. Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ touch inventory.file
```

2. Ouvrez le site **inventory.file** et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present.yml** Ansible playbook. Par exemple :

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present-copy.yml
```

4. Ouvrez le fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present-copy.yml** Ansible playbook pour l'éditer.

5. Adaptez le fichier en modifiant les éléments suivants :

   - Le mot de passe de l'administrateur IdM spécifié par la variable **ipaadmin_password**.
   - Le nom du client IdM sur lequel le service HTTP est exécuté. Dans l'exemple actuel, il s'agit de **HTTP/client.idm.example.com**
   - Les noms des utilisateurs IdM répertoriés dans la section **allow_create_keytab_user:**. Dans l'exemple actuel, il s'agit de **user01**.
   - Les noms des groupes d'utilisateurs IdM qui sont répertoriés dans la section **allow_create_keytab_group:**.
   - Les noms des hôtes IdM répertoriés dans la section **allow_create_keytab_host:**.
   - Les noms des groupes d'hôtes IdM répertoriés dans la section **allow_create_keytab_hostgroup:**.
   - Le nom de la tâche spécifiée par la variable **name** dans la section **tasks**.
     Après avoir été adapté à l'exemple présent, le fichier copié se présente comme suit :

```
---
- name: Service member allow_create_keytab present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Service HTTP/client.idm.example.com members allow_create_keytab present for
user01
    ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/client.idm.example.com
      allow_create_keytab_user:
      - user01
      action: member
```

6. Enregistrer le fichier.

7. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-member-allow_create_keytab-present-copy.yml
```

Verification steps

1. SSH vers un serveur IdM en tant qu'utilisateur IdM ayant le privilège de créer un keytab pour le service HTTP en question :

```
$ ssh user01@server.idm.example.com
Password:
```

2. Utilisez la commande **ipa-getkeytab** pour générer le nouveau keytab pour le service HTTP :

```
$ ipa-getkeytab -s server.idm.example.com -p HTTP/client.idm.example.com -k
/etc/httpd/conf/krb5.keytab
```

   L'option **-s** spécifie un serveur de centre de distribution de clés (KDC) pour générer le fichier de clés.

   L'option **-p** spécifie le principal dont vous voulez créer le keytab.

   L'option **-k** indique le fichier keytab auquel ajouter la nouvelle clé. Le fichier sera créé s'il n'existe pas.

Si la commande ne génère pas d'erreur, vous avez créé avec succès un keytab de **HTTP/client.idm.example.com** sous le nom de **user01**.

25.6. UTILISATION D'UN PLAYBOOK ANSIBLE POUR PERMETTRE AUX UTILISATEURS, GROUPES, HÔTES OU GROUPES D'HÔTES IDM DE RÉCUPÉRER UN KEYTAB D'UN SERVICE

Un fichier keytab est un fichier contenant des paires de mandants Kerberos et de clés cryptées. Les fichiers keytab sont généralement utilisés pour permettre aux scripts de s'authentifier automatiquement à l'aide de Kerberos, sans nécessiter d'interaction humaine ni d'accès à un mot de passe stocké dans un fichier en texte clair. Le script est alors en mesure d'utiliser les informations d'identification acquises pour accéder aux fichiers stockés sur un système distant.

En tant qu'administrateur IdM, vous pouvez autoriser d'autres utilisateurs à récupérer ou même à créer un keytab pour un service fonctionnant dans IdM.

Cette section explique comment autoriser des utilisateurs, des groupes d'utilisateurs, des hôtes et des groupes d'hôtes spécifiques de l'IdM à récupérer un keytab pour le service HTTP s'exécutant sur un client IdM. En particulier, elle décrit comment autoriser l'utilisateur IdM **user01** à récupérer le keytab du service HTTP fonctionnant sur **client.idm.example.com**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Vous avez inscrit le service HTTP dans IdM .
- Les utilisateurs et groupes d'utilisateurs de l'IdM que vous souhaitez autoriser à récupérer la base de données de clés existent dans l'IdM.
- Les hôtes et groupes d'hôtes IdM que vous souhaitez autoriser à récupérer le fichier clé existent dans IdM.

Procédure

1. Créer un fichier d'inventaire, par exemple **inventory.file**:

   ```
   touche inventory.file
   ```

2. Ouvrez le site **inventory.file** et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present.yml** Ansible playbook. Par exemple :

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present-copy.yml
```

4. Ouvrez le fichier copié, **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present-copy.yml**, pour le modifier :

5. Adapter le dossier :

   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Définissez la variable **name** de la tâche **ipaservice** comme étant le principal du service HTTP. Dans l'exemple actuel, il s'agit de **HTTP/client.idm.example.com**
   - Spécifiez les noms des utilisateurs IdM dans la section **allow_retrieve_keytab_group:**. Dans l'exemple actuel, il s'agit de **user01**.
   - Spécifiez les noms des groupes d'utilisateurs IdM dans la section **allow_retrieve_keytab_group:**.
   - Spécifiez les noms des hôtes IdM dans la section **allow_retrieve_keytab_group:**.
   - Spécifiez les noms des groupes d'hôtes IdM dans la section **allow_retrieve_keytab_group:**.
   - Spécifiez le nom de la tâche en utilisant la variable **name** dans la section **tasks**.
     Après avoir été adapté à l'exemple présent, le fichier copié se présente comme suit :

```
---
- name: Service member allow_retrieve_keytab present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Service HTTP/client.idm.example.com members allow_retrieve_keytab present for user01
    ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/client.idm.example.com
      allow_retrieve_keytab_user:
      - user01
      action: member
```

6. Enregistrer le fichier.

7. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i path_to_inventory_directory/inventory.file /usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present-copy.yml
```

Verification steps

1. SSH à un serveur IdM en tant qu'utilisateur IdM avec le privilège de récupérer un keytab pour le service HTTP :

```
$ ssh user01@server.idm.example.com
Password:
```

2. Utilisez la commande **ipa-getkeytab** avec l'option **-r** pour récupérer le fichier de clés :

```
$ ipa-getkeytab -r -s server.idm.example.com -p HTTP/client.idm.example.com -k
/etc/httpd/conf/krb5.keytab
```

L'option **-s** spécifie un serveur Key Distribution Center (KDC) à partir duquel vous souhaitez récupérer le keytab.

L'option **-p** spécifie le principal dont vous souhaitez récupérer le keytab.

L'option **-k** indique le fichier keytab auquel vous souhaitez ajouter la clé récupérée. Le fichier sera créé s'il n'existe pas.

Si la commande n'entraîne pas d'erreur, vous avez réussi à récupérer un fichier clé de **HTTP/client.idm.example.com** sous la forme **user01**.

25.7. ASSURER LA PRÉSENCE D'UN ALIAS PRINCIPAL KERBEROS D'UN SERVICE À L'AIDE D'UN PLAYBOOK ANSIBLE

Dans certains cas, l'administrateur IdM a intérêt à permettre aux utilisateurs, hôtes ou services IdM de s'authentifier auprès d'applications Kerberos à l'aide d'un alias de principal Kerberos. Il s'agit notamment des scénarios suivants

- Le nom d'utilisateur a changé, mais l'utilisateur doit pouvoir se connecter au système en utilisant à la fois l'ancien et le nouveau nom d'utilisateur.
- L'utilisateur doit se connecter à l'aide de l'adresse électronique, même si le domaine Kerberos de l'IdM diffère de celui de l'adresse électronique.

Cette section décrit comment créer l'alias principal de **HTTP/mycompany.idm.example.com** pour le service HTTP fonctionnant sur **client.idm.example.com**.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Vous avez mis en place un service HTTP sur votre hôte.

- Vous avez inscrit le service HTTP dans IdM .
- L'hôte sur lequel vous avez configuré HTTP est un client IdM.

Procédure

1. Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ touch inventory.file
```

2. Ouvrez le site **inventory.file** et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present.yml** Ansible playbook. Par exemple :

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present.yml /usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present-copy.yml
```

4. Ouvrez le fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present-copy.yml** Ansible playbook pour l'éditer.

5. Adaptez le fichier en modifiant les éléments suivants :

   - Le mot de passe de l'administrateur IdM spécifié par la variable **ipaadmin_password**.
   - Le nom du service spécifié par la variable **name**. Il s'agit du nom principal canonique du service. Dans l'exemple actuel, il s'agit de **HTTP/client.idm.example.com**.
   - L'alias du principal Kerberos spécifié par la variable **principal**. Il s'agit de l'alias que vous souhaitez ajouter au service défini par la variable **name**. Dans l'exemple actuel, il s'agit de **host/mycompany.idm.example.com**.
   - Le nom de la tâche spécifiée par la variable **name** dans la section **tasks**.
     Après avoir été adapté à l'exemple présent, le fichier copié se présente comme suit :

```
---
- name: Service member principal present
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Service HTTP/client.idm.example.com member principals
host/mycompany.idm.exmaple.com present
    ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/client.idm.example.com
```

```
principal:
  - host/mycompany.idm.example.com
action: member
```

6. Enregistrer le fichier.

7. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-member-principal-present-copy.yml
```

Si l'exécution du playbook se solde par 0 tâche inaccessible et 0 tâche échouée, vous avez créé avec succès le principal Kerberos **host/mycompany.idm.example.com** pour le service **HTTP/client.idm.example.com**.

Ressources supplémentaires

- Voir Gestion des alias principaux Kerberos pour les utilisateurs, les hôtes et les services .

25.8. GARANTIR L'ABSENCE D'UN SERVICE HTTP DANS L'IDM À L'AIDE D'UN PLAYBOOK ANSIBLE

Cette section décrit comment désenrôler un service de l'IdM. Plus précisément, elle décrit comment utiliser un playbook Ansible pour garantir l'absence d'un serveur HTTP nommé **HTTP/client.idm.example.com** dans IdM.

Conditions préalables

- Vous avez le mot de passe de l'administrateur IdM.

Procédure

1. Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ touch inventory.file
```

2. Ouvrez le site **inventory.file** et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent.yml** Ansible playbook. Par exemple :

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent.yml
/usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent-copy.yml
```

4. Ouvrez le fichier **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent-copy.yml** Ansible playbook pour l'éditer.

5. Adaptez le fichier en modifiant les éléments suivants :

   - Le mot de passe de l'administrateur IdM défini par la variable **ipaadmin_password**.
   - Le principal Kerberos du service HTTP, tel que défini par la variable **name** de la tâche **ipaservice**.
     Après avoir été adapté à l'exemple présent, le fichier copié se présente comme suit :

```
---
- name: Playbook to manage IPA service.
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  # Ensure service is absent
  - ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: HTTP/client.idm.example.com
      state: absent
```

6. Save and exit the file.

7. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file /usr/share/doc/ansible-
freeipa/playbooks/service/service-is-absent-copy.yml
```

Verification steps

1. Se connecter à l'interface Web IdM en tant qu'administrateur IdM.

2. Naviguez jusqu'à **Identity → Services**.

Si vous ne voyez pas le service **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** dans la liste **Services**, vous avez réussi à garantir son absence dans l'IdM.

25.9. RESSOURCES SUPPLÉMENTAIRES

- Voir le fichier Markdown de **README-service.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/config**.

CHAPITRE 26. GÉRER LA CONFIGURATION DNS GLOBALE DANS IDM À L'AIDE DE PLAYBOOKS ANSIBLE

En utilisant le module Red Hat Ansible Engine **dnsconfig**, vous pouvez configurer la configuration globale pour le DNS de la gestion des identités (IdM). Les paramètres définis dans la configuration DNS globale sont appliqués à tous les serveurs DNS IdM. Cependant, la configuration globale a une priorité inférieure à la configuration d'une zone DNS IdM spécifique.

Le module **dnsconfig** prend en charge les variables suivantes :

- Les transitaires globaux, en particulier leurs adresses IP et le port utilisé pour la communication.
- La politique de transfert globale : seulement, d'abord ou aucune. Pour plus de détails sur ces types de politiques de transfert DNS, voir les politiques de transfert DNS dans IdM .
- La synchronisation des zones de recherche avancée et de recherche inversée.

Conditions préalables

- Le service DNS est installé sur le serveur IdM. Pour plus d'informations sur l'installation d'un serveur IdM avec DNS intégré, voir l'un des liens suivants :
  - Installation d'un serveur IdM : Avec DNS intégré, avec une autorité de certification intégrée comme autorité de certification racine
  - Installation d'un serveur IdM : Avec DNS intégré, avec une autorité de certification externe comme autorité de certification racine
  - Installation d'un serveur IdM : Avec DNS intégré, sans CA

Ce chapitre comprend les sections suivantes :

- Comment IdM s'assure que les forwarders globaux du fichier /etc/resolv.conf ne sont pas supprimés par NetworkManager
- Assurer la présence d'un DNS global forwarder dans IdM en utilisant Ansible
- S'assurer de l'absence d'un DNS global forwarder dans l'IdM en utilisant Ansible
- L'option **action: member** dans les modules ipadnsconfig ansible-freeipa
- Introduction aux politiques de transfert DNS dans l'IdM
- Utilisation d'un playbook Ansible pour s'assurer que la politique "forward first" est définie dans la configuration globale du DNS IdM
- Utilisation d'un playbook Ansible pour s'assurer que les redirections globales sont désactivées dans le DNS IdM
- Utilisation d'un playbook Ansible pour s'assurer que la synchronisation des zones de recherche directe et inversée est désactivée dans IdM DNS

26.1. COMMENT IDM S'ASSURE QUE LES FORWARDERS GLOBAUX DU FICHIER /ETC/RESOLV.CONF NE SONT PAS SUPPRIMÉS PAR NETWORKMANAGER

L'installation de la gestion des identités (IdM) avec DNS intégré configure le fichier **/etc/resolv.conf** pour qu'il pointe vers l'adresse **127.0.0.1** localhost :

```
# Generated by NetworkManager
search idm.example.com
nameserver 127.0.0.1
```

Dans certains environnements, tels que les réseaux qui utilisent **Dynamic Host Configuration Protocol** (DHCP), le service **NetworkManager** peut annuler les modifications apportées au fichier **/etc/resolv.conf**. Pour rendre la configuration DNS persistante, le processus d'installation de IdM DNS configure également le service **NetworkManager** de la manière suivante :

1. Le script d'installation DNS crée un fichier de configuration **/etc/NetworkManager/conf.d/zzz-ipa.conf NetworkManager** pour contrôler l'ordre de recherche et la liste des serveurs DNS :

```
# auto-generated by IPA installer
[main]
dns=default

[global-dns]
searches=$DOMAIN

[global-dns-domain-*]
servers=127.0.0.1
```

2. Le service **NetworkManager** est rechargé, ce qui crée toujours le fichier **/etc/resolv.conf** avec les paramètres du dernier fichier du répertoire **/etc/NetworkManager/conf.d/**. Dans le cas présent, il s'agit du fichier **zzz-ipa.conf**.

IMPORTANT

Ne modifiez pas manuellement le fichier **/etc/resolv.conf**.

26.2. ASSURER LA PRÉSENCE D'UN DNS GLOBAL FORWARDER DANS IDM EN UTILISANT ANSIBLE

Cette section décrit comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour assurer la présence d'un transitaire global DNS dans IdM. Dans l'exemple de procédure ci-dessous, l'administrateur IdM assure la présence d'un transitaire global DNS vers un serveur DNS avec une adresse Internet Protocol (IP) v4 de **7.7.9.9** et une adresse IP v6 de **2001:db8::1:0** sur le port **53**.

Conditions préalables

- 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**.

- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **forwarders-absent.yml** Ansible playbook. Par exemple :

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. Ouvrez le fichier **ensure-presence-of-a-global-forwarder.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes :
   a. Modifiez la variable **name** du playbook en **Playbook to ensure the presence of a global forwarder in IdM DNS**.
   b. Dans la section **tasks**, changez le **name** de la tâche en **Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0 on port 53**.
   c. Dans la section **forwarders** de la partie **ipadnsconfig**:
      i. Remplacez la première valeur de **ip_address** par l'adresse IPv4 du transitaire global : **7.7.9.9**.
      ii. Remplacer la deuxième valeur **ip_address** par l'adresse IPv6 du transitaire global : **2001:db8::1:0**.
      iii. Vérifiez que la valeur **port** est définie sur **53**.
   d. Modifier le **state** en **present**.
   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to ensure the presence of a global forwarder in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
```

```
  - name: Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0 on port
53
    ipadnsconfig:
      forwarders:
        - ip_address: 7.7.9.9
        - ip_address: 2001:db8::1:0
          port: 53
      state: present
```

6. Enregistrer le fichier.

7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-
of-a-global-forwarder.yml
```

Ressources supplémentaires

- Voir le fichier **README-dnsconfig.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

26.3. S'ASSURER DE L'ABSENCE D'UN DNS GLOBAL FORWARDER DANS L'IDM EN UTILISANT ANSIBLE

Cette section décrit comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour garantir l'absence d'un transitaire global DNS dans IdM. Dans l'exemple de procédure ci-dessous, l'administrateur IdM s'assure de l'absence d'un transitaire global DNS avec une adresse Internet Protocol (IP) v4 de **8.8.6.6** et une adresse IP v6 de **2001:4860:4860::8800** sur le port **53**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **forwarders-absent.yml** Ansible playbook. Par exemple :

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. Ouvrez le fichier **ensure-absence-of-a-global-forwarder.yml** pour le modifier.

5. Adaptez le fichier en définissant les variables suivantes :

   a. Modifiez la variable **name** du playbook en **Playbook to ensure the absence of a global forwarder in IdM DNS**.

   b. Dans la section **tasks**, changez le **name** de la tâche en **Ensure the absence of a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800 on port 53**.

   c. Dans la section **forwarders** de la partie **ipadnsconfig**:

      i. Remplacez la première valeur de **ip_address** par l'adresse IPv4 du transitaire global : **8.8.6.6**.

      ii. Remplacer la deuxième valeur **ip_address** par l'adresse IPv6 du transitaire global : **2001:4860:4860::8800**.

      iii. Vérifiez que la valeur **port** est définie sur **53**.

   d. Fixer la variable **action** à **member**.

   e. Vérifiez que **state** est défini sur **absent**.

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to ensure the absence of a global forwarder in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the absence of a DNS global forwarder to 8.8.6.6 and
2001:4860:4860::8800 on port 53
    ipadnsconfig:
      forwarders:
        - ip_address: 8.8.6.6
        - ip_address: 2001:4860:4860::8800
          port: 53
      action: member
      state: absent
```

IMPORTANT

Si vous n'utilisez que l'option **state: absent** dans votre séquence sans utiliser également **action: member**, la séquence échoue.

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-of-a-global-forwarder.yml
```

Ressources supplémentaires

- Le fichier **README-dnsconfig.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**
- L'option **action: member** dans les modules ipadnsconfig ansible-freeipa

26.4. L'OPTION ACTION: MEMBER DANS LES MODULES IPADNSCONFIG ANSIBLE-FREEIPA

L'exclusion des expéditeurs globaux dans la gestion de l'identité (IdM) à l'aide du module **ansible-freeipa ipadnsconfig** nécessite l'utilisation de l'option **action: member** en plus de l'option **state: absent**. Si vous utilisez uniquement **state: absent** dans votre manuel sans utiliser également **action: member**, le manuel échoue. Par conséquent, pour supprimer tous les transitaires globaux, vous devez tous les spécifier individuellement dans le cahier d'exécution. En revanche, l'option **state: present** ne nécessite pas **action: member**.

Le tableau suivant fournit des exemples de configuration pour l'ajout et la suppression de transitaires globaux DNS qui démontrent l'utilisation correcte de l'option action : member. Le tableau indique, dans chaque ligne :

- Les transitaires globaux configurés avant l'exécution d'un playbook
- Un extrait du manuel de jeu
- Les transitaires globaux configurés après l'exécution du playbook

Tableau 26.1. ipadnsconfig gestion des transitaires globaux

Transporteurs avant	Extrait du Playbook	Transporteurs après
8.8.6.6	[...] tasks: - name: Ensure the presence of DNS global forwarder 8.8.6.7 ipadnsconfig: forwarders: - ip_address: 8.8.6.7 state: present	8.8.6.7

Transporteurs avant	Extrait du Playbook	Transporteurs après
8.8.6.6	[...] tasks: - name: Ensure the presence of DNS global forwarder 8.8.6.7 ipadnsconfig: forwarders: - ip_address: 8.8.6.7 action: member state: present	8.8.6.6, 8.8.6.7
8.8.6.6, 8.8.6.7	[...] tasks: - name: Ensure the absence of DNS global forwarder 8.8.6.7 ipadnsconfig: forwarders: - ip_address: 8.8.6.7 state: absent	La tentative d'exécution du playbook aboutit à une erreur. La configuration originale - 8.8.6.6, 8.8.6.7 - reste inchangée.
8.8.6.6, 8.8.6.7	[...] tasks: - name: Ensure the absence of DNS global forwarder 8.8.6.7 ipadnsconfig: forwarders: - ip_address: 8.8.6.7 action: member state: absent	8.8.6.6

26.5. POLITIQUES DE TRANSFERT DNS DANS L'IDM

IdM prend en charge les politiques d'acheminement standard de BIND **first** et **only**, ainsi que la politique d'acheminement spécifique à IdM **none**.

En avant, d'abord *(default)*

Le service BIND de l'IdM transmet les requêtes DNS au transitaire configuré. Si une requête échoue en raison d'une erreur de serveur ou d'un dépassement de délai, BIND se rabat sur la résolution récursive en utilisant des serveurs sur l'internet. La politique **forward first** est la politique par défaut et convient pour optimiser le trafic DNS.

En avant seulement

Le service IdM BIND transmet les requêtes DNS au transitaire configuré. Si une requête échoue en raison d'une erreur du serveur ou d'un dépassement de délai, BIND renvoie une erreur au client. La stratégie **forward only** est recommandée pour les environnements avec une configuration DNS

divisée.

Aucun *(forwarding disabled)*

Les requêtes DNS ne sont pas transférées avec la politique de transfert **none**. La désactivation de la redirection n'est utile que pour remplacer la configuration globale de la redirection dans une zone spécifique. Cette option est l'équivalent pour IdM de la spécification d'une liste vide de transitaires dans la configuration de BIND.

NOTE

Vous ne pouvez pas utiliser le transfert pour combiner des données dans IdM avec des données provenant d'autres serveurs DNS. Vous ne pouvez transférer des requêtes que pour des sous-zones spécifiques de la zone primaire dans le DNS IdM.

Par défaut, le service BIND ne transmet pas les requêtes à un autre serveur si le nom DNS demandé appartient à une zone pour laquelle le serveur IdM fait autorité. Dans une telle situation, si le nom DNS demandé ne peut être trouvé dans la base de données IdM, la réponse **NXDOMAIN** est renvoyée. Le transfert n'est pas utilisé.

Exemple 26.1. Exemple de scénario

Le serveur IdM fait autorité pour la zone DNS **test.example.**. BIND est configuré pour transmettre les requêtes au serveur DNS avec l'adresse IP **192.0.2.254**.

Lorsqu'un client envoie une requête pour le nom DNS **nonexistent.test.example.**, BIND détecte que le serveur IdM fait autorité pour la zone **test.example.** et ne transmet pas la requête au serveur **192.0.2.254.**. En conséquence, le client DNS reçoit le message d'erreur **NXDomain**, informant l'utilisateur que le domaine interrogé n'existe pas.

26.6. UTILISATION D'UN PLAYBOOK ANSIBLE POUR S'ASSURER QUE LA POLITIQUE "FORWARD FIRST" EST DÉFINIE DANS LA CONFIGURATION GLOBALE DU DNS IDM

Cette section décrit comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour s'assurer que la politique de transfert global dans IdM DNS est définie sur **forward first**.

Si vous utilisez la stratégie **forward first** DNS forwarding, les requêtes DNS sont transmises au forwarder configuré. Si une requête échoue en raison d'une erreur de serveur ou d'un dépassement de délai, BIND revient à la résolution récursive en utilisant des serveurs sur Internet. La stratégie "forward first" est la stratégie par défaut. Elle convient à l'optimisation du trafic.

Conditions préalables

- 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**.

- Vous connaissez le mot de passe de l'administrateur IdM.
- Votre environnement IdM contient un serveur DNS intégré.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **set-configuration.yml** Ansible playbook. Par exemple :

```
$ cp set-configuration.yml set-forward-policy-to-first.yml
```

4. Ouvrez le fichier **set-forward-policy-to-first.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnsconfig** task :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Fixer la variable **forward_policy** à **first**.
     Supprimez toutes les autres lignes du playbook original qui ne sont pas pertinentes. Voici le fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to set global forwarding policy to first
  hosts: ipaserver
  become: true

  tasks:
  - name: Set global forwarding policy to first.
    ipadnsconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
      forward_policy: first
```

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file set-forward-policy-to-first.yml
```

Ressources supplémentaires

- Voir les politiques de transfert de DNS dans IdM .
- Voir le fichier **README-dnsconfig.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Pour obtenir d'autres exemples de playbooks, consultez le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**.

26.7. UTILISATION D'UN PLAYBOOK ANSIBLE POUR S'ASSURER QUE LES REDIRECTIONS GLOBALES SONT DÉSACTIVÉES DANS LE DNS IDM

Cette section décrit comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour s'assurer que les redirections globales sont désactivées dans le DNS IdM. La désactivation s'effectue en définissant la variable **forward_policy** sur **none**.

La désactivation des transferts globaux a pour effet de ne pas transférer les requêtes DNS. La désactivation de la redirection n'est utile que pour remplacer la configuration de la redirection globale dans une zone spécifique. Cette option est l'équivalent pour IdM de la spécification d'une liste vide de transitaires dans la configuration de BIND.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.
- Votre environnement IdM contient un serveur DNS intégré.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **disable-global-forwarders.yml** Ansible playbook. Par exemple :

```
$ cp disable-global-forwarders.yml disable-global-forwarders-copy.yml
```

4. Ouvrez le fichier **disable-global-forwarders-copy.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnsconfig** task :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Fixer la variable **forward_policy** à **none**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to disable global DNS forwarders
  hosts: ipaserver
  become: true

  tasks:
  - name: Disable global forwarders.
    ipadnsconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
      forward_policy: none
```

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disable-global-forwarders-copy.yml
```

Ressources supplémentaires

- Voir les politiques de transfert de DNS dans IdM .
- Voir le fichier **README-dnsconfig.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir d'autres exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**.

26.8. UTILISATION D'UN PLAYBOOK ANSIBLE POUR S'ASSURER QUE LA SYNCHRONISATION DES ZONES DE RECHERCHE DIRECTE ET INVERSÉE EST DÉSACTIVÉE DANS IDM DNS

Cette section décrit comment un administrateur de gestion des identités (IdM) peut utiliser un manuel de jeu Ansible pour s'assurer que les zones de recherche directe et inversée ne sont pas synchronisées dans le DNS IdM.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.
- Votre environnement IdM contient un serveur DNS intégré.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **disallow-reverse-sync.yml** Ansible playbook. Par exemple :

```
$ cp disallow-reverse-sync.yml disallow-reverse-sync-copy.yml
```

4. Ouvrez le fichier **disallow-reverse-sync-copy.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnsconfig** task :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Fixer la variable **allow_sync_ptr** à **no**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to disallow reverse record synchronization
  hosts: ipaserver
  become: true

  tasks:
  - name: Disallow reverse record synchronization.
    ipadnsconfig:
      ipaadmin_password: "{{ ipaadmin_password }}"
      allow_sync_ptr: no
```

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disallow-reverse-sync-copy.yml
```

Ressources supplémentaires

- Voir le fichier **README-dnsconfig.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Pour obtenir d'autres exemples de playbooks, consultez le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**.

CHAPITRE 27. UTILISER LES PLAYBOOKS ANSIBLE POUR GÉRER LES ZONES DNS DE L'IDM

En tant qu'administrateur Identity Management (IdM), vous pouvez gérer le fonctionnement des zones DNS IdM à l'aide du module **dnszone** disponible dans le package **ansible-freeipa**. Ce chapitre décrit les sujets et procédures suivants :

- Quels sont les types de zones DNS pris en charge par IdM ?
- Quels sont les attributs DNS que vous pouvez configurer dans IdM ?
- Comment utiliser un playbook Ansible pour créer une zone primaire dans IdM DNS
- Comment utiliser un playbook Ansible pour s'assurer de la présence d'une zone DNS IdM primaire avec plusieurs variables
- Comment utiliser un playbook Ansible pour s'assurer de la présence d'une zone pour la recherche DNS inversée lorsqu'une adresse IP est donnée ?

Conditions préalables

- Le service DNS est installé sur le serveur IdM. Pour plus d'informations sur l'utilisation de Red Hat Ansible Engine pour installer un serveur IdM avec DNS intégré, voir Installation d'un serveur de gestion d'identité à l'aide d'un playbook Ansible.

27.1. TYPES DE ZONES DNS PRISES EN CHARGE

Identity Management (IdM) prend en charge deux types de zones DNS : les zones *primary* et *forward*. Cette section décrit ces deux types de zones et inclut un exemple de scénario de transfert DNS.

NOTE

Ce guide utilise la terminologie BIND pour les types de zones, qui est différente de la terminologie utilisée pour le DNS de Microsoft Windows. Les zones primaires dans BIND ont la même fonction que *forward lookup zones* et *reverse lookup zones* dans le DNS de Microsoft Windows. Les zones de transfert dans BIND ont la même fonction que *conditional forwarders* dans le DNS de Microsoft Windows.

Zones DNS primaires

Les zones DNS primaires contiennent des données DNS faisant autorité et peuvent accepter des mises à jour DNS dynamiques. Ce comportement est équivalent au paramètre **type master** dans la configuration standard de BIND. Vous pouvez gérer les zones primaires à l'aide des commandes **ipa dnszone-***.
Conformément aux règles DNS standard, chaque zone primaire doit contenir des enregistrements **start of authority** (SOA) et **nameserver** (NS). L'IdM génère automatiquement ces enregistrements lors de la création de la zone DNS, mais vous devez copier manuellement les enregistrements NS dans la zone mère pour créer une délégation correcte.

Conformément au comportement standard de BIND, les requêtes portant sur des noms pour lesquels le serveur ne fait pas autorité sont transmises à d'autres serveurs DNS. Ces serveurs DNS, appelés "forwarders", peuvent ou non faire autorité pour la requête.

Exemple 27.1. Exemple de scénario pour le transfert DNS

> Le serveur IdM contient la zone primaire **test.example.**. Cette zone contient un enregistrement de délégation NS pour le nom **sub.test.example.**. En outre, la zone **test.example.** est configurée avec l'adresse IP du transitaire **192.0.2.254** pour la sous-zone **sub.text.example**.
>
> Un client interrogeant le nom **nonexistent.test.example.** reçoit la réponse **NXDomain** et aucun transfert n'a lieu car le serveur IdM fait autorité pour ce nom.
>
> D'autre part, les requêtes portant sur le nom **host1.sub.test.example.** sont transmises au transitaire configuré **192.0.2.254**, car le serveur IdM ne fait pas autorité pour ce nom.

Transférer des zones DNS

Du point de vue de l'IdM, les zones DNS avancées ne contiennent aucune donnée faisant autorité. En fait, une "zone" avancée ne contient généralement que deux éléments d'information :

- Un nom de domaine
- L'adresse IP d'un serveur DNS associé au domaine

Toutes les requêtes portant sur des noms appartenant au domaine défini sont transmises à l'adresse IP spécifiée. Ce comportement est équivalent au paramètre **type forward** dans la configuration standard de BIND. Vous pouvez gérer les zones de transfert à l'aide des commandes **ipa dnsforwardzone-***.

Les zones DNS à suivre sont particulièrement utiles dans le contexte des trusts IdM-Active Directory (AD). Si le serveur DNS IdM fait autorité pour la zone **idm.example.com** et que le serveur DNS AD fait autorité pour la zone **ad.example.com**, alors **ad.example.com** est une zone DNS de renvoi pour la zone primaire **idm.example.com**. Cela signifie que lorsqu'un client IdM demande l'adresse IP de **somehost.ad.example.com**, la requête est transmise à un contrôleur de domaine AD spécifié dans la zone de transfert DNS IdM **ad.example.com**.

27.2. ATTRIBUTS DE CONFIGURATION DES ZONES DNS PRIMAIRES DE L'IDM

Identity Management (IdM) crée une nouvelle zone avec certaines configurations par défaut, telles que les périodes de rafraîchissement, les paramètres de transfert ou les paramètres de cache. Dans les attributs de la zone DNS IdM , vous trouverez les attributs de la configuration de la zone par défaut que vous pouvez modifier à l'aide de l'une des options suivantes :

- La commande **dnszone-mod** dans l'interface de ligne de commande (CLI). Pour plus d'informations, voir Modifier la configuration d'une zone DNS primaire dans l'interface CLI de l'IdM.
- L'interface Web IdM. Pour plus d'informations, voir Modifier la configuration d'une zone DNS primaire dans l'interface Web IdM.
- Un playbook Ansible qui utilise le module **ipadnszone**. Pour plus d'informations, voir Gestion des zones DNS dans IdM.

Outre les informations relatives à la zone, les paramètres définissent la manière dont le serveur DNS traite les entrées de l'enregistrement *start of authority* (SOA) et la manière dont il met à jour ses enregistrements à partir du serveur de noms DNS.

Tableau 27.1. Attributs de la zone DNS de l'IdM

Attribut	variable ansible-freeipa	Description
Serveur de noms faisant autorité	**name_server**	Définit le nom de domaine du serveur de noms DNS primaire, également connu sous le nom de SOA MNAME. Par défaut, chaque serveur IdM s'annonce lui-même dans le champ SOA MNAME. Par conséquent, la valeur stockée dans LDAP à l'aide de **--name-server** est ignorée.
Adresse électronique de l'administrateur	**admin_email**	Définit l'adresse électronique à utiliser pour l'administrateur de zone. Par défaut, il s'agit du compte root de l'hôte.
Série SOA	**serial**	Définit un numéro de série dans l'enregistrement SOA. Notez que l'IdM définit automatiquement le numéro de version et que les utilisateurs ne sont pas censés le modifier.
Actualisation de l'AOS	**refresh**	Définit l'intervalle, en secondes, pendant lequel un serveur DNS secondaire doit attendre avant de demander des mises à jour au serveur DNS primaire.
Réessai SOA	**retry**	Définit le délai, en secondes, à attendre avant de réessayer une opération de rafraîchissement qui a échoué.
SOA expirer	**expire**	Définit la durée, en secondes, pendant laquelle un serveur DNS secondaire tentera d'effectuer une mise à jour avant de mettre fin à la tentative d'opération.
Minimum SOA	**minimum**	Définit la valeur TTL (time to live) en secondes pour la mise en cache négative conformément à la RFC 2308.
Délai de mise en œuvre de la SOA	**ttl**	Définit le TTL en secondes pour les enregistrements à l'apex de la zone. Dans la zone **example.com**, par exemple, tous les enregistrements (A, NS ou SOA) sous le nom **example.com** sont configurés, mais aucun autre nom de domaine, comme **test.example.com**, n'est affecté.
Durée de vie par défaut	**default_ttl**	Définit la valeur par défaut du TTL (Time to Live) en secondes pour la mise en cache négative de toutes les valeurs d'une zone qui n'ont jamais eu de valeur TTL individuelle définie auparavant. Nécessite un redémarrage du service **named-pkcs11** sur tous les serveurs DNS IdM pour que les modifications soient prises en compte.
Politique de mise à jour de BIND	**update_policy**	Définit les autorisations accordées aux clients dans la zone DNS.

Attribut	variable ansible-freeipa	Description
Mise à jour dynamique	**dynamic_update**=VRAI|FAUX	Active les mises à jour dynamiques des enregistrements DNS pour les clients. Notez que si cette valeur est fixée à false, les machines clientes IdM ne pourront pas ajouter ou mettre à jour leur adresse IP.
Autoriser le transfert	**allow_transfer**=*string*	Donne une liste d'adresses IP ou de noms de réseau autorisés à transférer la zone donnée, séparés par des points-virgules (;). Les transferts de zone sont désactivés par défaut. La valeur par défaut de **allow_transfer** est **none**.
Autoriser l'interrogation	**allow_query**	Donne une liste d'adresses IP ou de noms de réseau autorisés à émettre des requêtes DNS, séparés par des points-virgules (;).
Autoriser la synchronisation PTR	**allow_sync_ptr**=1|0	Définit si les enregistrements A ou AAAA (enregistrements directs) de la zone seront automatiquement synchronisés avec les enregistrements PTR (enregistrements inversés).
Transitaires de zone	**forwarder**=*IP_address*	Spécifie un transitaire spécifiquement configuré pour la zone DNS. Ce transitaire est distinct des transitaires globaux utilisés dans le domaine IdM. Pour spécifier plusieurs transitaires, utilisez l'option plusieurs fois.
Politique prévisionnelle	**forward_policy**=unique|uniquement|premier	Spécifie la politique de transfert. Pour plus d'informations sur les politiques prises en charge, voir Politiques de transfert DNS dans IdM.

Ressources supplémentaires

- Voir le fichier **README-dnszone.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

27.3. UTILISER ANSIBLE POUR CRÉER UNE ZONE PRIMAIRE DANS IDM DNS

Cette section montre comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour s'assurer de l'existence d'une zone DNS primaire. Dans l'exemple utilisé dans la procédure ci-dessous, un administrateur IdM s'assure de la présence de la zone DNS **zone.idm.example.com**.

Conditions préalables

- 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**.

- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnszone**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **dnszone-present.yml** Ansible playbook. Par exemple :

```
$ cp dnszone-present.yml dnszone-present-copy.yml
```

4. Ouvrez le fichier **dnszone-present-copy.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnszone** task :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Fixer la variable **zone_name** à **zone.idm.example.com**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure zone is present.
    ipadnszone:
      ipaadmin_password: "{{ ipaadmin_password }}"
      zone_name: zone.idm.example.com
      state: present
```

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file dnszone-present-copy.yml
```

Ressources supplémentaires

- Voir Types de zones DNS pris en charge .
- Voir le fichier **README-dnszone.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks Ansible dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnszone**.

27.4. UTILISATION D'UN PLAYBOOK ANSIBLE POUR ASSURER LA PRÉSENCE D'UNE ZONE DNS PRIMAIRE DANS L'IDM AVEC PLUSIEURS VARIABLES

Cette section montre comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour s'assurer de l'existence d'une zone DNS primaire. Dans l'exemple utilisé dans la procédure ci-dessous, un administrateur IdM s'assure de la présence de la zone DNS **zone.idm.example.com**. Le playbook Ansible configure plusieurs paramètres de la zone.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnszone**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **dnszone-all-params.yml** Ansible playbook. Par exemple :

```
$ cp dnszone-all-params.yml dnszone-all-params-copy.yml
```

4. Ouvrez le fichier **dnszone-all-params-copy.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnszone** task :

- Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
- Fixer la variable **zone_name** à **zone.idm.example.com**.
- Attribuez la valeur true à la variable **allow_sync_ptr** si vous souhaitez autoriser la synchronisation des enregistrements en aval et en amont, c'est-à-dire la synchronisation des enregistrements A et AAAA avec les enregistrements PTR.
- Définissez la variable **dynamic_update** sur true pour permettre aux machines clientes IdM d'ajouter ou de mettre à jour leurs adresses IP.
- Attribuez la valeur true à la variable **dnssec** pour permettre la signature DNSSEC en ligne des enregistrements dans la zone.
- Définissez la variable **allow_transfer** avec les adresses IP des serveurs de noms secondaires de la zone.
- Définissez la variable **allow_query** en fonction des adresses IP ou des réseaux autorisés à émettre des requêtes.
- Définissez la variable **forwarders** avec les adresses IP des transitaires globaux.
- Attribuer à la variable **serial** le numéro de série de l'enregistrement SOA.
- Définissez les valeurs **refresh**, **retry**, **expire**, **minimum**, **ttl**, et **default_ttl** pour les enregistrements DNS de la zone.
- Définir l'enregistrement NSEC3PARAM pour la zone en utilisant la variable **nsec3param_rec**.
- Définissez la variable **skip_overlap_check** sur true pour forcer la création d'un DNS même s'il chevauche une zone existante.
- Attribuez la valeur true à **skip_nameserver_check** pour forcer la création d'une zone DNS même si le serveur de noms n'est pas résolvable.
  Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Ensure dnszone present
 hosts: ipaserver
 become: true

 tasks:
 - name: Ensure zone is present.
  ipadnszone:
    ipaadmin_password: "{{ ipaadmin_password }}"
    zone_name: zone.idm.example.com
    allow_sync_ptr: true
    dynamic_update: true
    dnssec: true
    allow_transfer:
      - 1.1.1.1
      - 2.2.2.2
    allow_query:
      - 1.1.1.1
      - 2.2.2.2
    forwarders:
```

```
    - ip_address: 8.8.8.8
    - ip_address: 8.8.4.4
      port: 52
  serial: 1234
  refresh: 3600
  retry: 900
  expire: 1209600
  minimum: 3600
  ttl: 60
  default_ttl: 90
  name_server: server.idm.example.com.
  admin_email: admin.admin@idm.example.com
  nsec3param_rec: "1 7 100 0123456789abcdef"
  skip_overlap_check: true
  skip_nameserver_check: true
  state: present
```

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file dnszone-all-params-copy.yml
```

Ressources supplémentaires

- Voir Types de zones DNS pris en charge .
- Voir Attributs de configuration des zones DNS primaires de l'IdM .
- Voir le fichier **README-dnszone.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks Ansible dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnszone**.

27.5. UTILISATION D'UN PLAYBOOK ANSIBLE POUR S'ASSURER DE LA PRÉSENCE D'UNE ZONE POUR LA RECHERCHE DNS INVERSÉE LORSQU'UNE ADRESSE IP EST DONNÉE

Cette section montre comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour s'assurer de l'existence d'une zone DNS inversée. Dans l'exemple utilisé dans la procédure ci-dessous, un administrateur IdM s'assure de la présence d'une zone de recherche DNS inverse en utilisant l'adresse IP et la longueur du préfixe d'un hôte IdM.

En indiquant la longueur du préfixe de l'adresse IP de votre serveur DNS à l'aide de la variable **name_from_ip**, vous pouvez contrôler le nom de la zone. Si vous n'indiquez pas la longueur du préfixe, le système interroge les serveurs DNS sur les zones et, en fonction de la valeur **name_from_ip** de **192.168.1.2**, la requête peut renvoyer n'importe laquelle des zones DNS suivantes :

- **1.168.192.in-addr.arpa.**
- **168.192.in-addr.arpa.**
- **192.in-addr.arpa.**

Étant donné que la zone renvoyée par la requête peut ne pas correspondre à ce que vous attendez, **name_from_ip** ne peut être utilisé qu'avec l'option **state** réglée sur **present** afin d'éviter les suppressions accidentelles de zones.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnszone**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **dnszone-reverse-from-ip.yml** Ansible playbook. Par exemple :

```
$ cp dnszone-reverse-from-ip.yml dnszone-reverse-from-ip-copy.yml
```

4. Ouvrez le fichier **dnszone-reverse-from-ip-copy.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnszone** task :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Définissez la variable **name_from_ip** avec l'IP de votre serveur de noms IdM et indiquez la longueur de son préfixe.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure zone for reverse DNS lookup is present.
```

```
    ipadnszone:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name_from_ip: 192.168.1.2/24
      state: present
    register: result
  - name: Display inferred zone name.
    debug:
      msg: "Zone name: {{ result.dnszone.name }}"
```

Le playbook crée une zone pour la recherche DNS inversée à partir de l'adresse IP **192.168.1.2** et de sa longueur de préfixe de 24. Ensuite, le playbook affiche le nom de la zone résultante.

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file dnszone-reverse-from-ip-copy.yml
```

Ressources supplémentaires

- Voir Types de zones DNS pris en charge .
- Voir le fichier **README-dnszone.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks Ansible dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnszone**.

CHAPITRE 28. UTILISER ANSIBLE POUR GÉRER LES EMPLACEMENTS DNS DANS IDM

En tant qu'administrateur Identity Management (IdM), vous pouvez gérer les emplacements DNS IdM à l'aide du module **location** disponible dans le package **ansible-freeipa**. Ce chapitre décrit les sujets et procédures suivants :

- Découverte de services basée sur le DNS
- Considérations relatives au déploiement des sites DNS
- Durée de vie du DNS (TTL)
- Utiliser Ansible pour s'assurer qu'un emplacement IdM est présent
- Utiliser Ansible pour s'assurer qu'un emplacement IdM est absent

28.1. DÉCOUVERTE DE SERVICES BASÉE SUR LE DNS

La découverte de services basée sur le DNS est un processus dans lequel un client utilise le protocole DNS pour localiser les serveurs d'un réseau qui offrent un service spécifique, tel que **LDAP** ou **Kerberos**. Un type d'opération typique consiste à permettre aux clients de localiser les serveurs d'authentification dans l'infrastructure réseau la plus proche, parce qu'ils offrent un débit plus élevé et une latence de réseau plus faible, ce qui réduit les coûts globaux.

Les principaux avantages de la découverte de services sont les suivants

- Il n'est pas nécessaire de configurer explicitement les clients avec les noms des serveurs proches.
- Les serveurs DNS sont utilisés comme fournisseurs centraux de politiques. Les clients qui utilisent le même serveur DNS ont accès à la même politique concernant les fournisseurs de services et leur ordre préférentiel.

Dans un domaine de gestion d'identité (IdM), il existe des enregistrements de service DNS (enregistrements SRV) pour **LDAP**, **Kerberos** et d'autres services. Par exemple, la commande suivante interroge le serveur DNS sur les hôtes fournissant un service **Kerberos** basé sur TCP dans un domaine DNS IdM :

Exemple 28.1. Résultats indépendants de l'emplacement du DNS

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
0 100 88 idmserver-01.idm.example.com.
0 100 88 idmserver-02.idm.example.com.
```

La sortie contient les informations suivantes :

- **0** (priorité) : Priorité de l'hôte cible. Une valeur inférieure est préférable.
- **100** (poids). Spécifie un poids relatif pour les entrées ayant la même priorité. Pour plus d'informations, voir RFC 2782, section 3.
- **88** (numéro de port) : Numéro de port du service.
- Nom canonique de l'hôte fournissant le service.

Dans l'exemple, les deux noms d'hôte renvoyés ont la même priorité et le même poids. Dans ce cas, le client utilise une entrée aléatoire de la liste des résultats.

Lorsque le client est configuré pour interroger un serveur DNS configuré dans un emplacement DNS, le résultat est différent. Pour les serveurs IdM qui sont affectés à un emplacement, des valeurs adaptées sont renvoyées. Dans l'exemple ci-dessous, le client est configuré pour interroger un serveur DNS dans l'emplacement **germany**:

Exemple 28.2. Résultats basés sur la localisation DNS

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
_kerberos._tcp.germany._locations.idm.example.com.
0 100 88 idmserver-01.idm.example.com.
50 100 88 idmserver-02.idm.example.com.
```

Le serveur DNS IdM renvoie automatiquement un alias DNS (CNAME) pointant vers un enregistrement SRV spécifique à l'emplacement DNS qui privilégie les serveurs locaux. Cet enregistrement CNAME est indiqué sur la première ligne de la sortie. Dans l'exemple, l'hôte **idmserver-01.idm.example.com** a la valeur de priorité la plus basse et est donc préféré. L'hôte **idmserver-02.idm.example.com** a une priorité plus élevée et n'est donc utilisé qu'en cas de sauvegarde, lorsque l'hôte préféré n'est pas disponible.

28.2. CONSIDÉRATIONS RELATIVES AU DÉPLOIEMENT DES SITES DNS

Identity Management (IdM) peut générer des enregistrements de service spécifiques à un emplacement (SRV) lors de l'utilisation du DNS intégré. Comme chaque serveur DNS IdM génère des enregistrements SRV spécifiques à l'emplacement, vous devez installer au moins un serveur DNS IdM dans chaque emplacement DNS.

L'affinité du client avec un emplacement DNS n'est définie que par les enregistrements DNS reçus par le client. C'est pourquoi vous pouvez combiner des serveurs DNS IdM avec des serveurs consommateurs et des récursoires DNS non IdM si les clients qui découvrent le service DNS résolvent les enregistrements spécifiques à l'emplacement à partir des serveurs DNS IdM.

Dans la majorité des déploiements avec des services DNS IdM et non IdM mixtes, les récursoires DNS sélectionnent automatiquement le serveur DNS IdM le plus proche en utilisant des mesures de temps d'aller-retour. En règle générale, cela garantit que les clients utilisant des serveurs DNS non IdM obtiennent des enregistrements pour l'emplacement DNS le plus proche et utilisent donc l'ensemble optimal de serveurs IdM.

28.3. DURÉE DE VIE DU DNS (TTL)

Les clients peuvent mettre en cache les enregistrements de ressources DNS pendant une durée définie dans la configuration de la zone. En raison de cette mise en cache, un client peut ne pas être en mesure de recevoir les modifications avant l'expiration de la valeur TTL (time to live). La valeur TTL par défaut dans Identity Management (IdM) est **1 day**.

Si les ordinateurs de vos clients se déplacent d'un site à l'autre, vous devez adapter la valeur TTL de votre zone DNS IdM. Définissez une valeur inférieure au temps nécessaire aux clients pour se déplacer d'un site à l'autre. Cela garantit que les entrées DNS mises en cache sur le client expirent avant qu'il ne

se reconnecte à un autre site et n'interroge le serveur DNS pour actualiser les enregistrements SRV spécifiques à l'emplacement.

Ressources supplémentaires

- Voir Attributs de configuration des zones DNS primaires de l'IdM .

28.4. UTILISER ANSIBLE POUR S'ASSURER QU'UN EMPLACEMENT IDM EST PRÉSENT

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez configurer les emplacements DNS IdM pour permettre aux clients de localiser les serveurs d'authentification dans l'infrastructure réseau la plus proche.

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'un emplacement DNS est présent dans IdM. L'exemple décrit comment s'assurer que l'emplacement DNS **germany** est présent dans IdM. En conséquence, vous pouvez assigner des serveurs IdM particuliers à cet emplacement afin que les clients IdM locaux puissent les utiliser pour réduire le temps de réponse du serveur.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- 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**.
- Vous comprenez les considérations relatives au déploiement des sites DNS .

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **location-present.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/location/**:

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

3. Ouvrez le fichier **location-present-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipalocation** task :

- Adaptez le site **name** de la tâche pour qu'il corresponde à votre cas d'utilisation.
- Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
- Définissez la variable **name** avec le nom de l'emplacement.

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: location present example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "germany" location is present
    ipalocation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: germany
```

5. Enregistrer le fichier.
6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory location-present-copy.yml
```

Ressources supplémentaires

- Voir Affectation d'un serveur IdM à un emplacement DNS à l'aide de l'interface Web Id M ou Affectation d'un serveur IdM à un emplacement DNS à l'aide de l'interface CLI IdM .

28.5. UTILISER ANSIBLE POUR S'ASSURER QU'UN EMPLACEMENT IDM EST ABSENT

En tant qu'administrateur système de la gestion des identités (IdM), vous pouvez configurer les emplacements DNS IdM pour permettre aux clients de localiser les serveurs d'authentification dans l'infrastructure réseau la plus proche.

La procédure suivante décrit comment utiliser un playbook Ansible pour s'assurer qu'un emplacement DNS est absent de l'IdM. L'exemple décrit comment s'assurer que l'emplacement DNS **germany** est absent de l'IdM. Par conséquent, vous ne pouvez pas attribuer de serveurs IdM particuliers à cet emplacement et les clients IdM locaux ne peuvent pas les utiliser.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.
- Aucun serveur IdM n'est assigné à l'emplacement DNS **germany**.
- 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**.

- L'exemple suppose que vous avez créé et configuré le répertoire ***~/MyPlaybooks/*** en tant qu'emplacement central pour stocker les copies des exemples de playbooks.

Procédure

1. Naviguez jusqu'au répertoire ***~/MyPlaybooks/*** répertoire :

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

2. Faites une copie du fichier **location-absent.yml** situé dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/location/**:

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

3. Ouvrez le fichier **location-absent-copy.yml** Ansible playbook pour l'éditer.
4. Adaptez le fichier en définissant les variables suivantes dans la section **ipalocation** task :
   - Adaptez le site **name** de la tâche pour qu'il corresponde à votre cas d'utilisation.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de l'administrateur IdM.
   - Définissez la variable **name** avec le nom de l'emplacement DNS.
   - Assurez-vous que la variable **state** est fixée à **absent**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: location absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "germany" location is absent
    ipalocation:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: germany
      state: absent
```

5. Enregistrer le fichier.

6. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory location-absent-copy.yml
```

28.6. RESSOURCES SUPPLÉMENTAIRES

- Voir le fichier **README-location.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks Ansible dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/location**.

CHAPITRE 29. GESTION DE LA REDIRECTION DNS DANS L'IDM

Les procédures suivantes décrivent comment configurer les forwarders globaux DNS et les zones de forward DNS dans l'interface Web de gestion des identités (IdM), dans la CLI IdM et à l'aide d'Ansible :

- Les deux rôles d'un serveur DNS IdM
- Politiques de transfert DNS dans l'IdM
- Ajout d'un transitaire global dans l'interface Web IdM
- Ajout d'un transitaire global dans l'interface de gestion
- Ajout d'une zone de transfert DNS dans l'interface Web IdM
- Ajout d'une zone de transfert DNS dans l'interface de programmation
- Mise en place d'un DNS Global Forwarder dans IdM à l'aide d'Ansible
- Assurer la présence d'un DNS global forwarder dans IdM en utilisant Ansible
- S'assurer de l'absence d'un DNS global forwarder dans l'IdM en utilisant Ansible
- S'assurer que les DNS Global Forwarders sont désactivés dans IdM à l'aide d'Ansible
- Assurer la présence d'une zone de transfert DNS dans IdM en utilisant Ansible
- S'assurer qu'une zone de transfert DNS a plusieurs transitaires dans IdM à l'aide d'Ansible
- S'assurer qu'une zone de transfert DNS est désactivée dans l'IdM à l'aide d'Ansible
- Garantir l'absence d'une zone de transfert DNS dans l'IdM à l'aide d'Ansible

29.1. LES DEUX RÔLES D'UN SERVEUR DNS IDM

La redirection DNS affecte la manière dont un service DNS répond aux requêtes DNS. Par défaut, le service Berkeley Internet Name Domain (BIND) intégré à l'IdM fait office de serveur DNS *authoritative* et *recursive*:

Serveur DNS autoritaire

Lorsqu'un client DNS interroge un nom appartenant à une zone DNS pour laquelle le serveur IdM fait autorité, BIND répond avec les données contenues dans la zone configurée. Les données faisant autorité ont toujours la priorité sur les autres données.

Serveur DNS récursif

Lorsqu'un client DNS interroge un nom pour lequel le serveur IdM ne fait pas autorité, BIND tente de résoudre la requête en utilisant d'autres serveurs DNS. Si les forwarders ne sont pas définis, BIND interroge les serveurs racine sur Internet et utilise un algorithme de résolution récursif pour répondre à la requête DNS.

Dans certains cas, il n'est pas souhaitable de laisser BIND contacter directement d'autres serveurs DNS et d'effectuer la récursivité sur la base des données disponibles sur Internet. Vous pouvez configurer BIND pour qu'il utilise un autre serveur DNS, *forwarder*, pour résoudre la requête.

Lorsque vous configurez BIND pour utiliser un transitaire, les requêtes et les réponses sont transmises entre le serveur IdM et le transitaire, et le serveur IdM fait office de cache DNS pour les données ne faisant pas autorité.

29.2. POLITIQUES DE TRANSFERT DNS DANS L'IDM

IdM prend en charge les politiques d'acheminement standard de BIND **first** et **only**, ainsi que la politique d'acheminement spécifique à IdM **none**.

En avant, d'abord *(default)*

Le service BIND de l'IdM transmet les requêtes DNS au transitaire configuré. Si une requête échoue en raison d'une erreur de serveur ou d'un dépassement de délai, BIND se rabat sur la résolution récursive en utilisant des serveurs sur l'internet. La politique **forward first** est la politique par défaut et convient pour optimiser le trafic DNS.

En avant seulement

Le service IdM BIND transmet les requêtes DNS au transitaire configuré. Si une requête échoue en raison d'une erreur du serveur ou d'un dépassement de délai, BIND renvoie une erreur au client. La stratégie **forward only** est recommandée pour les environnements avec une configuration DNS divisée.

Aucun *(forwarding disabled)*

Les requêtes DNS ne sont pas transférées avec la politique de transfert **none**. La désactivation de la redirection n'est utile que pour remplacer la configuration globale de la redirection dans une zone spécifique. Cette option est l'équivalent pour IdM de la spécification d'une liste vide de transitaires dans la configuration de BIND.

NOTE

Vous ne pouvez pas utiliser le transfert pour combiner des données dans IdM avec des données provenant d'autres serveurs DNS. Vous ne pouvez transférer des requêtes que pour des sous-zones spécifiques de la zone primaire dans le DNS IdM.

Par défaut, le service BIND ne transmet pas les requêtes à un autre serveur si le nom DNS demandé appartient à une zone pour laquelle le serveur IdM fait autorité. Dans une telle situation, si le nom DNS demandé ne peut être trouvé dans la base de données IdM, la réponse **NXDOMAIN** est renvoyée. Le transfert n'est pas utilisé.

Exemple 29.1. Exemple de scénario

Le serveur IdM fait autorité pour la zone DNS **test.example.**. BIND est configuré pour transmettre les requêtes au serveur DNS avec l'adresse IP **192.0.2.254**.

Lorsqu'un client envoie une requête pour le nom DNS **nonexistent.test.example.**, BIND détecte que le serveur IdM fait autorité pour la zone **test.example.** et ne transmet pas la requête au serveur **192.0.2.254.**. En conséquence, le client DNS reçoit le message d'erreur **NXDomain**, informant l'utilisateur que le domaine interrogé n'existe pas.

29.3. AJOUT D'UN TRANSITAIRE GLOBAL DANS L'INTERFACE WEB IDM

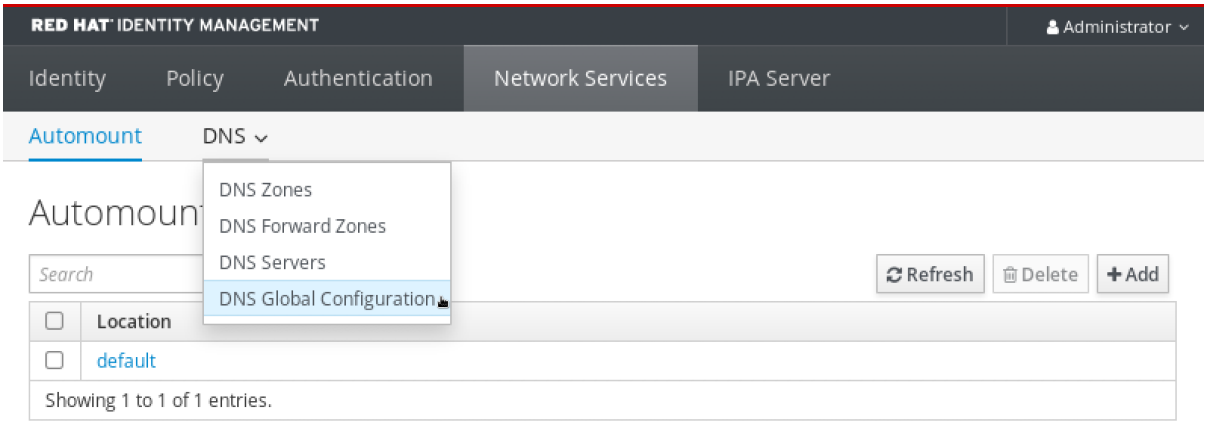
Cette section décrit comment ajouter un transitaire DNS global dans l'interface utilisateur Web de la gestion des identités (IdM).

Conditions préalables

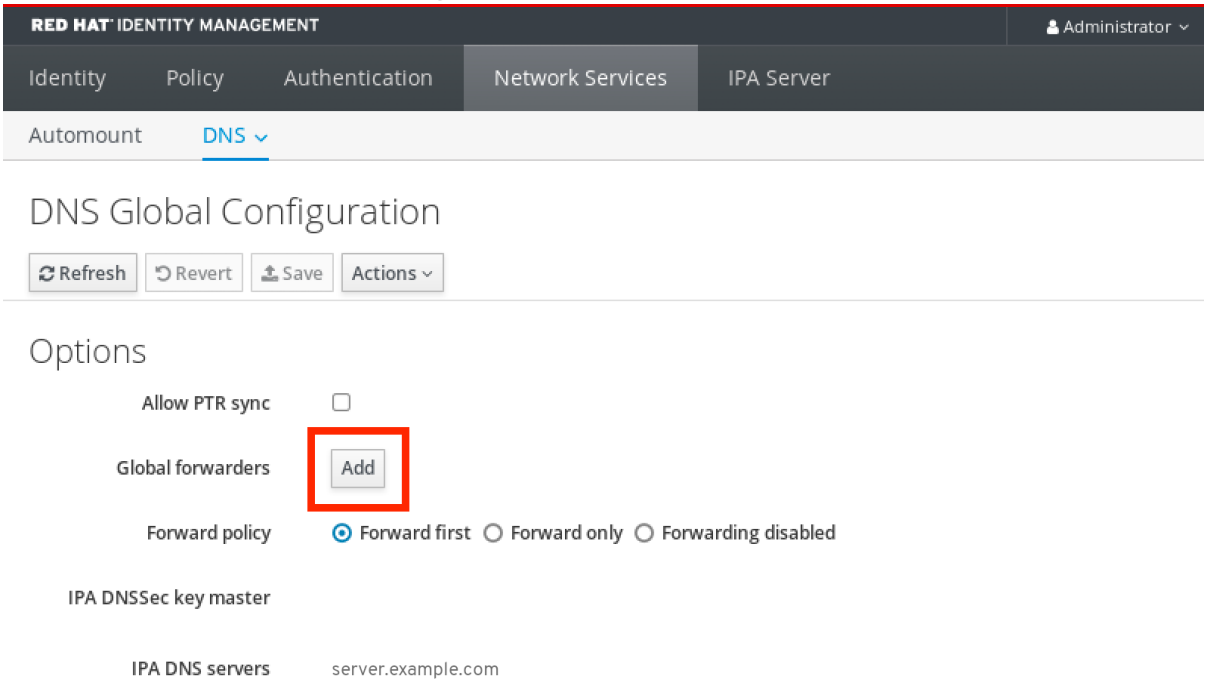
- Vous êtes connecté à l'interface Web de l'IdM en tant qu'administrateur de l'IdM.
- Vous connaissez l'adresse IP (Internet Protocol) du serveur DNS vers lequel transférer les requêtes.

Procédure

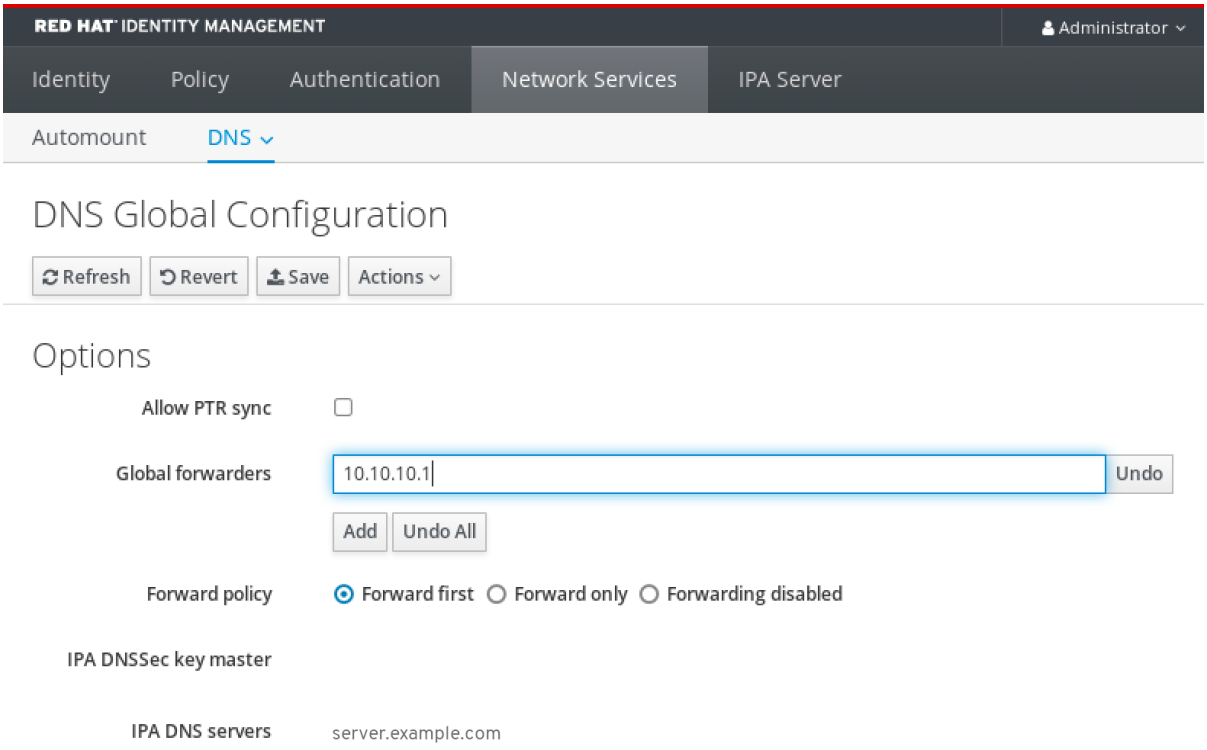
1. Dans l'interface Web IdM, sélectionnez **Network Services → DNS Global Configuration → DNS**.



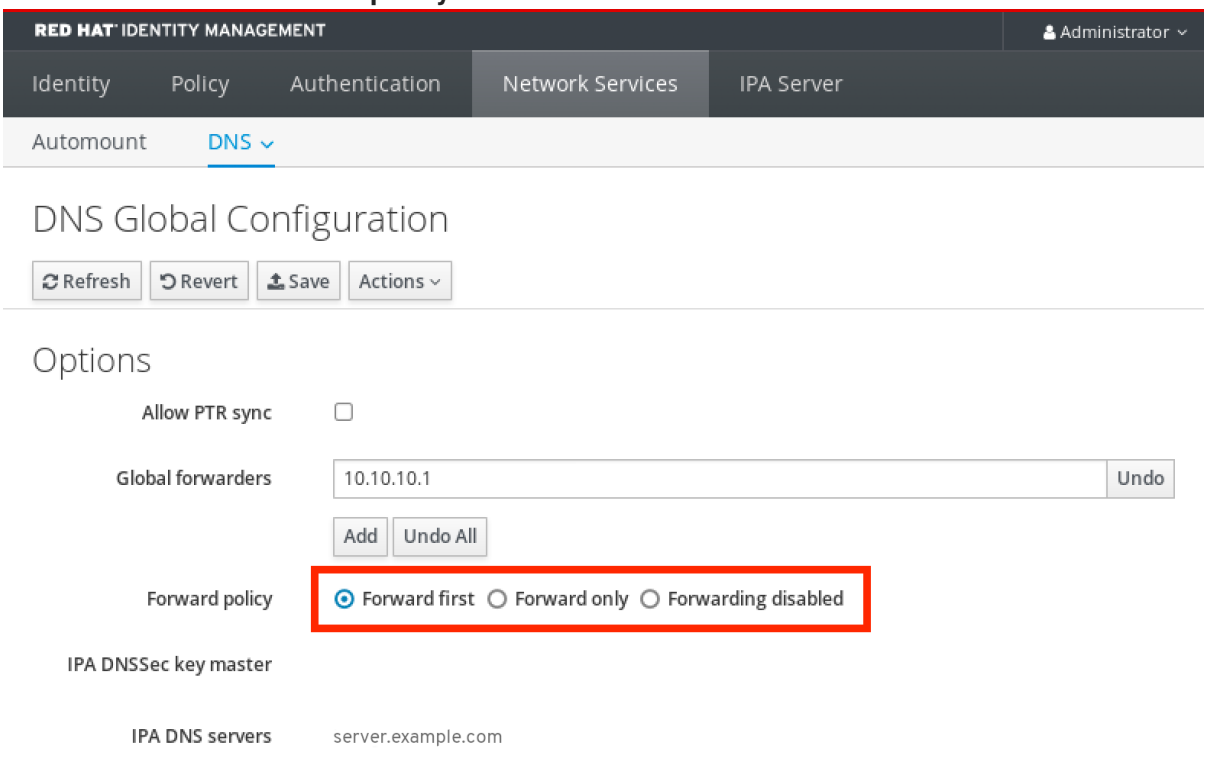
2. Dans la section **DNS Global Configuration**, cliquez sur **Add**.



3. Spécifiez l'adresse IP du serveur DNS qui recevra les requêtes DNS transférées.



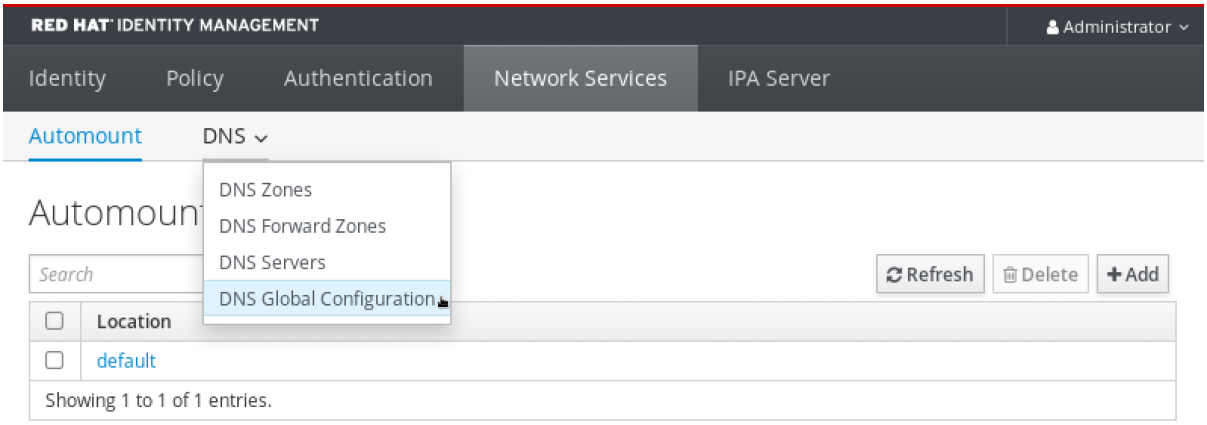
4. Sélectionnez le site **Forward policy**.



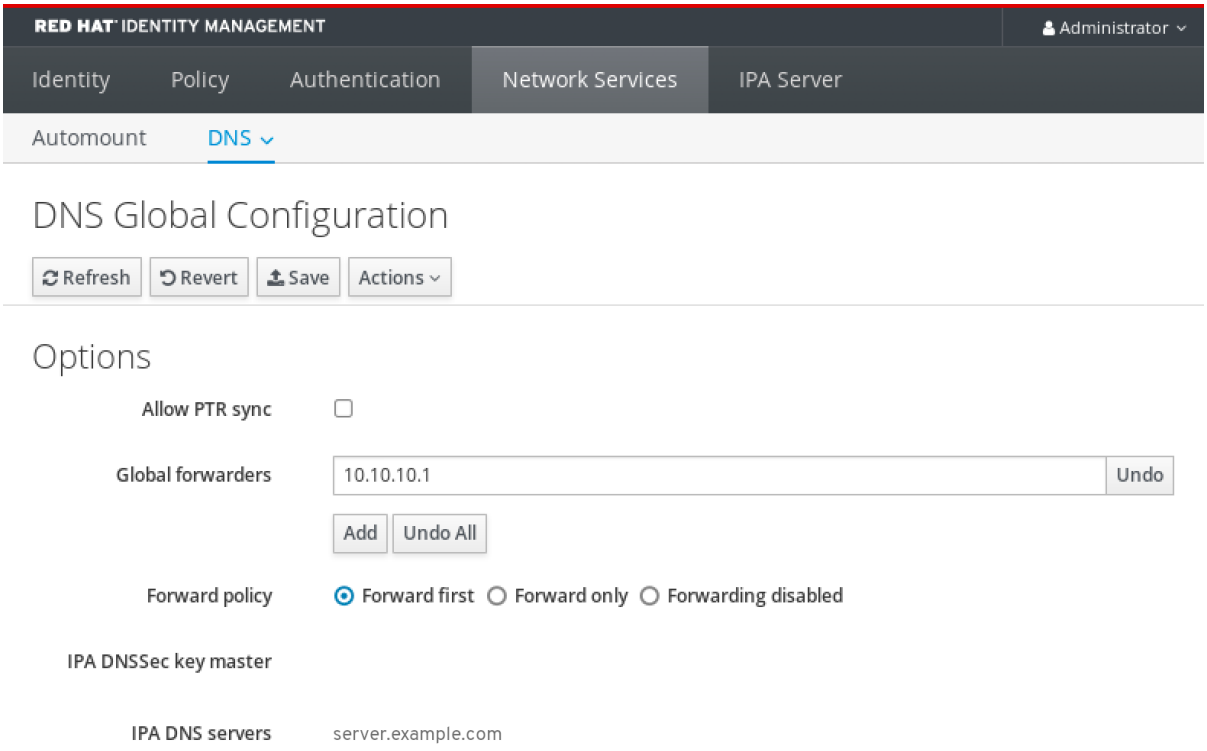
5. Cliquez sur **Save** en haut de la fenêtre.

Verification steps

1. Sélectionnez **Network Services → DNS Global Configuration → DNS**.



2. Vérifiez que le transitaire global, avec la politique de transfert que vous avez spécifiée, est présent et activé dans l'interface utilisateur Web IdM.



29.4. AJOUT D'UN TRANSITAIRE GLOBAL DANS L'INTERFACE DE GESTION

Cette section décrit comment ajouter un transitaire DNS global à partir de l'interface de ligne de commande (CLI).

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Vous connaissez l'adresse IP (Internet Protocol) du serveur DNS vers lequel transférer les requêtes.

Procédure

- Utilisez la commande **ipa dnsconfig-mod** pour ajouter un nouveau transitaire global. Spécifiez l'adresse IP du transitaire DNS avec l'option **--forwarder**.

```
[user@server ~]$ ipa dnsconfig-mod --forwarder=10.10.0.1
Server will check DNS forwarder(s).
This may take some time, please wait ...
  Global forwarders: 10.10.0.1
  IPA DNS servers: server.example.com
```

Verification steps

- Utilisez la commande **dnsconfig-show** pour afficher les transitaires globaux.

```
[user@server ~]$ ipa dnsconfig-show
  Global forwarders: 10.10.0.1
  IPA DNS servers: server.example.com
```

29.5. AJOUT D'UNE ZONE DE TRANSFERT DNS DANS L'INTERFACE WEB IDM

Cette section décrit comment ajouter une zone de transfert DNS dans l'interface utilisateur Web de la gestion des identités (IdM).

IMPORTANT

N'utilisez pas de zones de transmission, sauf en cas d'absolue nécessité. Les zones de transfert ne constituent pas une solution standard et leur utilisation peut entraîner un comportement inattendu et problématique. Si vous devez utiliser des zones de transfert, limitez leur utilisation à l'annulation d'une configuration de transfert globale.

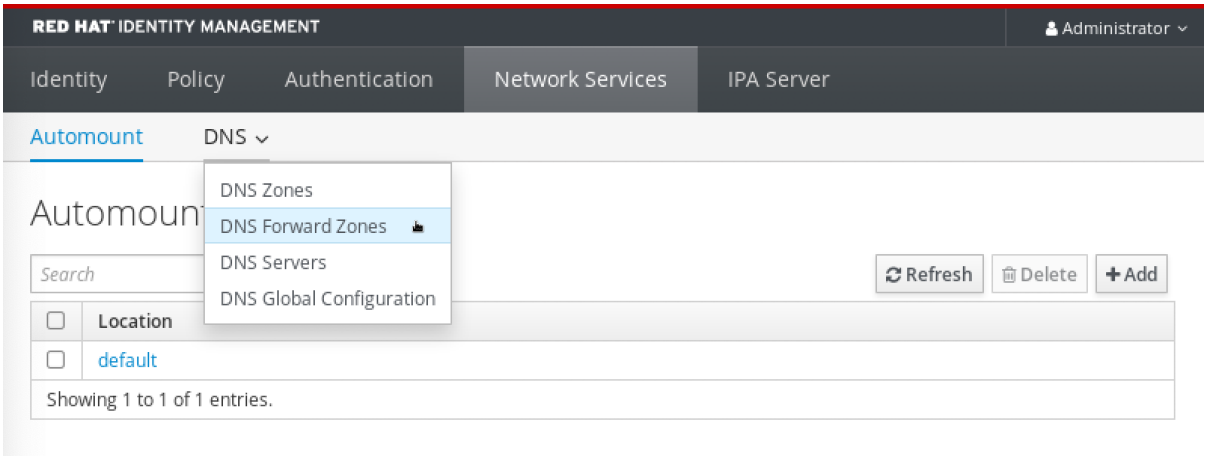
Lors de la création d'une nouvelle zone DNS, Red Hat recommande de toujours utiliser la délégation DNS standard à l'aide d'enregistrements de serveurs de noms (NS) et d'éviter les zones de transfert. Dans la plupart des cas, l'utilisation d'un transitaire global est suffisante et les zones de transfert ne sont pas nécessaires.

Conditions préalables

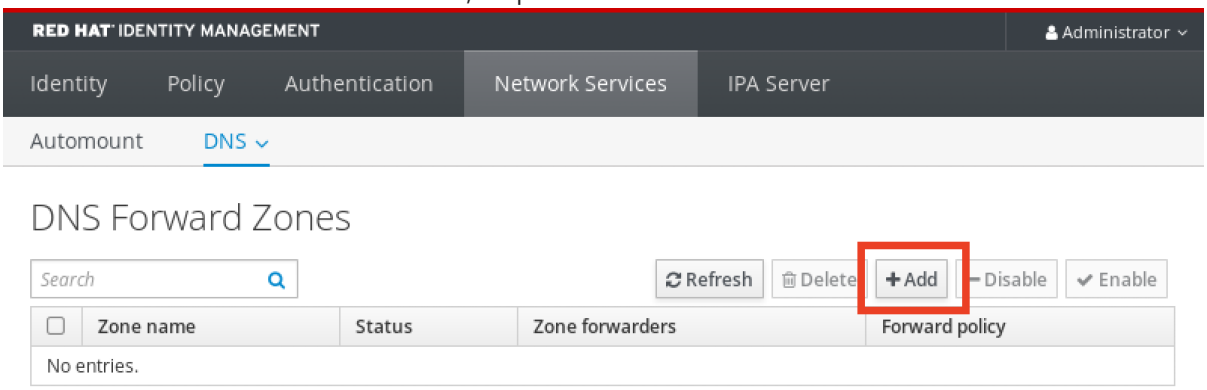
- Vous êtes connecté à l'interface Web de l'IdM en tant qu'administrateur de l'IdM.
- Vous connaissez l'adresse IP (Internet Protocol) du serveur DNS vers lequel transférer les requêtes.

Procédure

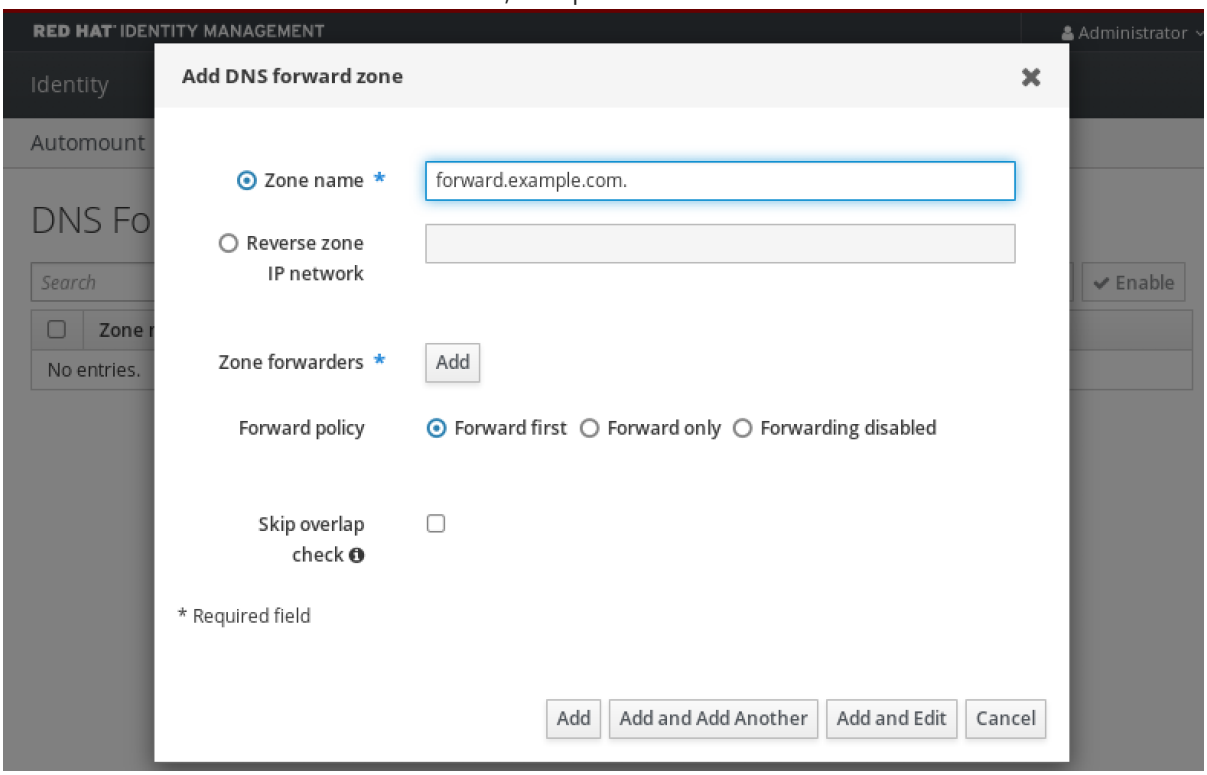
1. Dans l'interface Web IdM, sélectionnez **Network Services → DNS Forward Zones → DNS**.



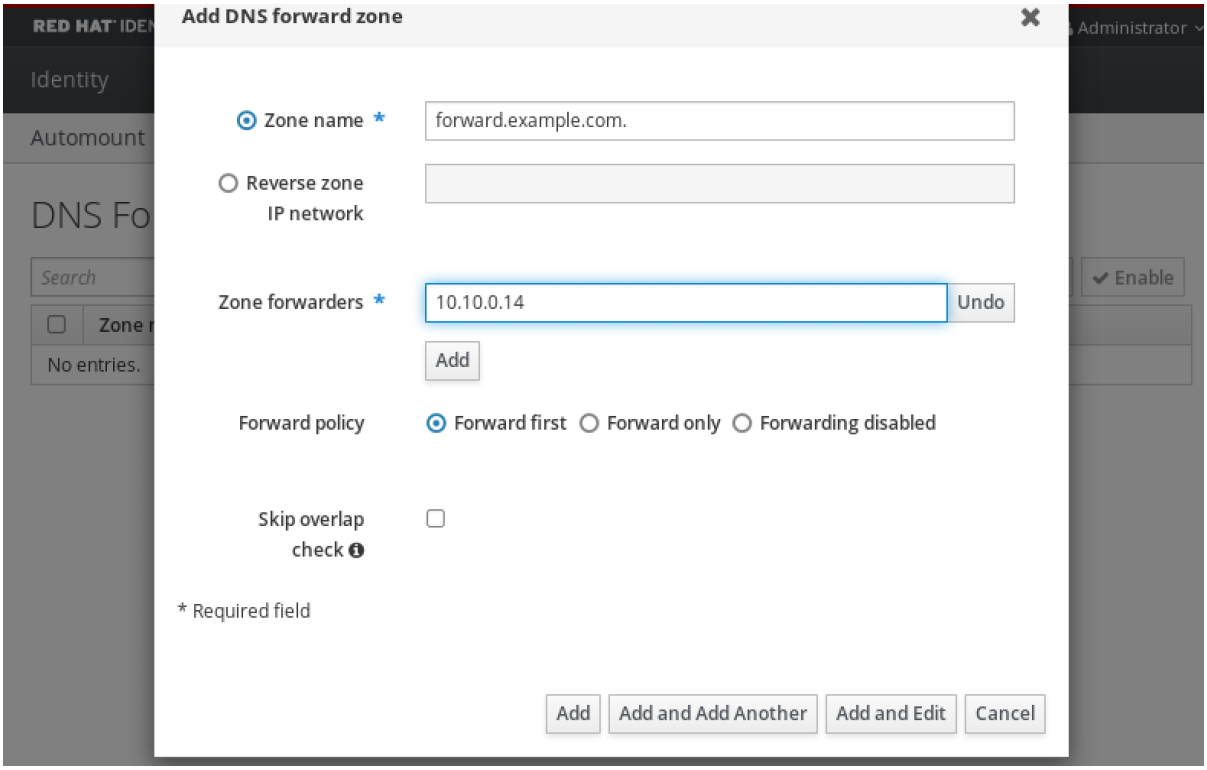
2. Dans la section **DNS Forward Zones**, cliquez sur **Add**.



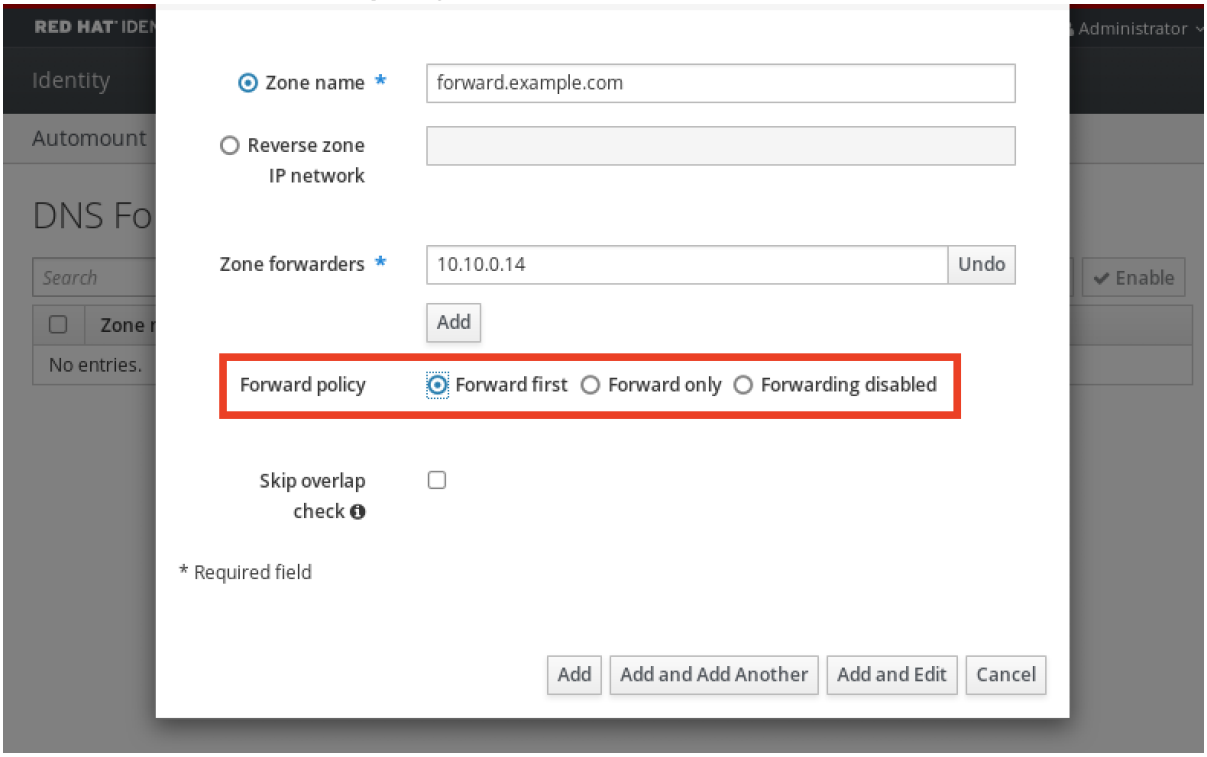
3. Dans la fenêtre **Add DNS forward zone**, indiquez le nom de la zone de transfert.



4. Cliquez sur le bouton **Add** et indiquez l'adresse IP d'un serveur DNS qui recevra la demande de transfert. Vous pouvez spécifier plusieurs serveurs de transfert par zone de transfert.



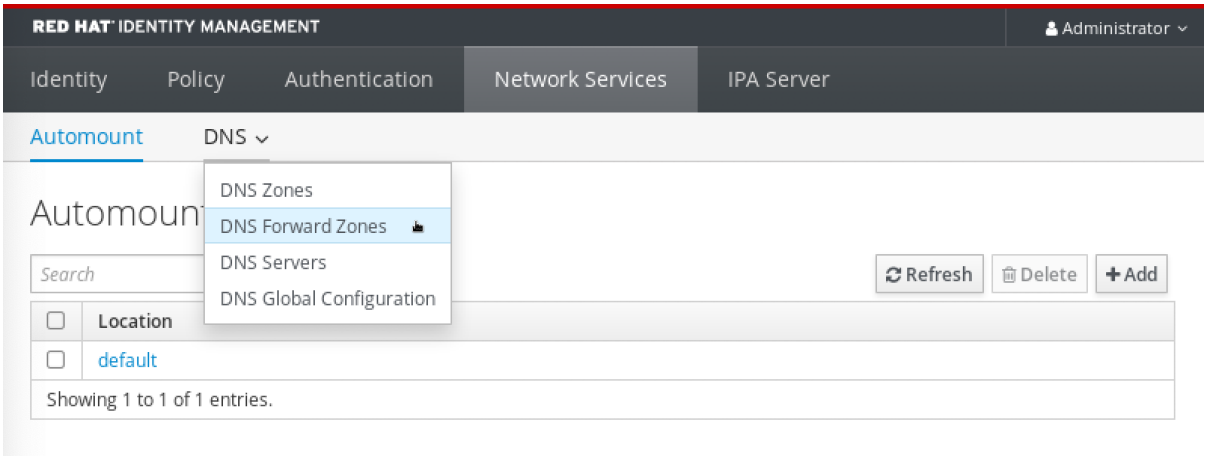
5. Sélectionnez le site **Forward policy**.



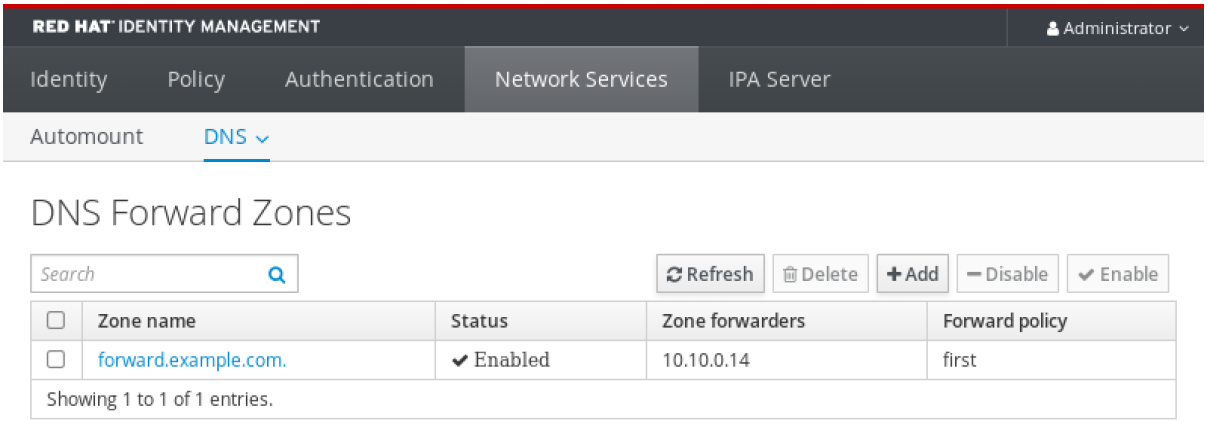
6. Cliquez sur **Add** en bas de la fenêtre pour ajouter la nouvelle zone de transmission.

Verification steps

1. Dans l'interface Web IdM, sélectionnez **Network Services → DNS Forward Zones → DNS**.

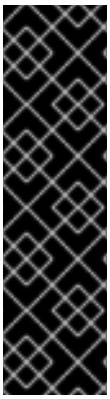


2. Vérifiez que la zone de transfert que vous avez créée, avec les transitaires et la politique de transfert que vous avez spécifiés, est présente et activée dans l'interface utilisateur Web d'IdM.



29.6. AJOUT D'UNE ZONE DE TRANSFERT DNS DANS L'INTERFACE DE PROGRAMMATION

Cette section décrit comment ajouter une zone de transfert DNS à partir de l'interface de ligne de commande (CLI).



IMPORTANT

N'utilisez pas de zones de transmission, sauf en cas d'absolue nécessité. Les zones de transfert ne constituent pas une solution standard et leur utilisation peut entraîner un comportement inattendu et problématique. Si vous devez utiliser des zones de transfert, limitez leur utilisation à l'annulation d'une configuration de transfert globale.

Lors de la création d'une nouvelle zone DNS, Red Hat recommande de toujours utiliser la délégation DNS standard à l'aide d'enregistrements de serveurs de noms (NS) et d'éviter les zones de transfert. Dans la plupart des cas, l'utilisation d'un transitaire global est suffisante et les zones de transfert ne sont pas nécessaires.

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.
- Vous connaissez l'adresse IP (Internet Protocol) du serveur DNS vers lequel transférer les requêtes.

Procédure

- Utilisez la commande **dnsforwardzone-add** pour ajouter une nouvelle zone de transfert. Spécifiez au moins un transitaire avec l'option **--forwarder** si la politique de transfert n'est pas **none**, et spécifiez la politique de transfert avec l'option **--forward-policy**.

```
[user@server ~]$ ipa dnsforwardzone-add forward.example.com. --
forwarder=10.10.0.14 --forwarder=10.10.1.15 --forward-policy=first

Zone name: forward.example.com.
Zone forwarders: 10.10.0.14, 10.10.1.15
Forward policy: first
```

Verification steps

- Utilisez la commande **dnsforwardzone-show** pour afficher la zone de transfert DNS que vous venez de créer.

```
[user@server ~]$ ipa dnsforwardzone-show forward.example.com.

Zone name: forward.example.com.
Zone forwarders: 10.10.0.14, 10.10.1.15
Forward policy: first
```

29.7. MISE EN PLACE D'UN DNS GLOBAL FORWARDER DANS IDM À L'AIDE D'ANSIBLE

Cette section décrit comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour établir un DNS Global Forwarder dans IdM.

Dans l'exemple de procédure ci-dessous, l'administrateur IdM crée une redirection globale DNS vers un serveur DNS avec une adresse Internet Protocol (IP) v4 de **8.8.6.6** et une adresse IPv6 de **2001:4860:4860::8800** sur le port **53**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **set-configuration.yml** Ansible playbook. Par exemple :

```
cp set-configuration.yml establish-global-forwarder.yml
```

4. Ouvrez le fichier **establish-global-forwarder.yml** pour le modifier.

5. Adaptez le fichier en définissant les variables suivantes :

   a. Modifiez la variable **name** du playbook en **Playbook to establish a global forwarder in IdM DNS**.

   b. Dans la section **tasks**, changez le **name** de la tâche en **Create a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800**.

   c. Dans la section **forwarders** de la partie **ipadnsconfig**:

      i. Remplacez la première valeur de **ip_address** par l'adresse IPv4 du transitaire global : **8.8.6.6**.

      ii. Remplacer la deuxième valeur **ip_address** par l'adresse IPv6 du transitaire global : **2001:4860:4860::8800**.

      iii. Vérifiez que la valeur **port** est définie sur **53**.

   d. Modifier le **forward_policy** en **first**.
   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to establish a global forwarder in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Create a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800
    ipadnsconfig:
      forwarders:
        - ip_address: 8.8.6.6
        - ip_address: 2001:4860:4860::8800
          port: 53
      forward_policy: first
      allow_sync_ptr: yes
```

6. Enregistrer le fichier.

7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file establish-global-forwarder.yml
```

Ressources supplémentaires

- Voir le fichier **README-dnsconfig.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

29.8. ASSURER LA PRÉSENCE D'UN DNS GLOBAL FORWARDER DANS IDM EN UTILISANT ANSIBLE

Cette section décrit comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour assurer la présence d'un transitaire global DNS dans IdM. Dans l'exemple de procédure ci-dessous, l'administrateur IdM assure la présence d'un transitaire global DNS vers un serveur DNS avec une adresse Internet Protocol (IP) v4 de **7.7.9.9** et une adresse IP v6 de **2001:db8::1:0** sur le port **53**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **forwarders-absent.yml** Ansible playbook. Par exemple :

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. Ouvrez le fichier **ensure-presence-of-a-global-forwarder.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes :
   a. Modifiez la variable **name** du playbook en **Playbook to ensure the presence of a global**

forwarder in IdM DNS.

b. Dans la section **tasks**, changez le **name** de la tâche en **Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0 on port 53**.

c. Dans la section **forwarders** de la partie **ipadnsconfig**:

i. Remplacez la première valeur de **ip_address** par l'adresse IPv4 du transitaire global : **7.7.9.9**.

ii. Remplacer la deuxième valeur **ip_address** par l'adresse IPv6 du transitaire global : **2001:db8::1:0**.

iii. Vérifiez que la valeur **port** est définie sur **53**.

d. Modifier le **state** en **present**.
Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to ensure the presence of a global forwarder in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0 on port
53
    ipadnsconfig:
      forwarders:
        - ip_address: 7.7.9.9
        - ip_address: 2001:db8::1:0
          port: 53
      state: present
```

6. Enregistrer le fichier.

7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-
of-a-global-forwarder.yml
```

Ressources supplémentaires

- Voir le fichier **README-dnsconfig.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

29.9. S'ASSURER DE L'ABSENCE D'UN DNS GLOBAL FORWARDER DANS L'IDM EN UTILISANT ANSIBLE

Cette section décrit comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour garantir l'absence d'un transitaire global DNS dans IdM. Dans l'exemple de procédure ci-dessous, l'administrateur IdM s'assure de l'absence d'un transitaire global DNS avec une adresse Internet Protocol (IP) v4 de **8.8.6.6** et une adresse IP v6 de **2001:4860:4860::8800** sur le port **53**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **forwarders-absent.yml** Ansible playbook. Par exemple :

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. Ouvrez le fichier **ensure-absence-of-a-global-forwarder.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes :
   a. Modifiez la variable **name** du playbook en **Playbook to ensure the absence of a global forwarder in IdM DNS**.
   b. Dans la section **tasks**, changez le **name** de la tâche en **Ensure the absence of a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800 on port 53**.
   c. Dans la section **forwarders** de la partie **ipadnsconfig**:
      i. Remplacez la première valeur de **ip_address** par l'adresse IPv4 du transitaire global : **8.8.6.6**.
      ii. Remplacer la deuxième valeur **ip_address** par l'adresse IPv6 du transitaire global : **2001:4860:4860::8800**.
      iii. Vérifiez que la valeur **port** est définie sur **53**.
   d. Fixer la variable **action** à **member**.
   e. Vérifiez que **state** est défini sur **absent**.

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to ensure the absence of a global forwarder in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the absence of a DNS global forwarder to 8.8.6.6 and
2001:4860:4860::8800 on port 53
    ipadnsconfig:
      forwarders:
        - ip_address: 8.8.6.6
        - ip_address: 2001:4860:4860::8800
          port: 53
      action: member
      state: absent
```

IMPORTANT

Si vous n'utilisez que l'option **state: absent** dans votre séquence sans utiliser également **action: member**, la séquence échoue.

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-of-
a-global-forwarder.yml
```

Ressources supplémentaires

- Le fichier **README-dnsconfig.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**
- L'option **action: member** dans les modules ipadnsconfig ansible-freeipa

29.10. S'ASSURER QUE LES DNS GLOBAL FORWARDERS SONT DÉSACTIVÉS DANS IDM À L'AIDE D'ANSIBLE

Cette section décrit comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour s'assurer que les transitaires globaux DNS sont désactivés dans IdM. Dans l'exemple de procédure ci-dessous, l'administrateur IdM s'assure que la politique de transfert du transitaire global est définie sur **none**, ce qui désactive effectivement le transitaire global.

Conditions préalables

- 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**.

- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Vérifiez le contenu du fichier **disable-global-forwarders.yml** Ansible playbook qui est déjà configuré pour désactiver toutes les redirections globales DNS. Par exemple :

```
$ cat disable-global-forwarders.yml
---
- name: Playbook to disable global DNS forwarders
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Disable global forwarders.
    ipadnsconfig:
      forward_policy: none
```

4. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disable-global-forwarders.yml
```

Ressources supplémentaires

- Voir le fichier **README-dnsconfig.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

29.11. ASSURER LA PRÉSENCE D'UNE ZONE DE TRANSFERT DNS DANS IDM EN UTILISANT ANSIBLE

Cette section décrit comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour assurer la présence d'une zone de transfert DNS dans IdM. Dans l'exemple de procédure ci-dessous, l'administrateur IdM assure la présence d'une zone de transfert DNS pour **example.com** vers un serveur DNS avec une adresse IP (Internet Protocol) de **8.8.8.8**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **forwarders-absent.yml** Ansible playbook. Par exemple :

```
cp forwarders-absent.yml ensure-presence-forwardzone.yml
```

4. Ouvrez le fichier **ensure-presence-forwardzone.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes :
   a. Modifiez la variable **name** du playbook en **Playbook to ensure the presence of a dnsforwardzone in IdM DNS**.
   b. Dans la section **tasks**, changez le **name** de la tâche en **Ensure presence of a dnsforwardzone for example.com to 8.8.8.8**.
   c. Dans la section **tasks**, remplacez le titre **ipadnsconfig** par **ipadnsforwardzone**.
   d. Dans la section **ipadnsforwardzone**:
      i. Ajoutez la variable **ipaadmin_password** et définissez-la comme votre mot de passe d'administrateur IdM.
      ii. Ajoutez la variable **name** et fixez-la à **example.com**.
      iii. Dans la section **forwarders**:
         A. Supprimer les lignes **ip_address** et **port**.

B. Ajoutez l'adresse IP du serveur DNS qui doit recevoir les requêtes transférées en la spécifiant après un tiret :

```
- 8.8.8.8
```

iv. Ajoutez la variable **forwardpolicy** et fixez-la à **first**.

v. Ajoutez la variable **skip_overlap_check** et fixez-la à **true**.

vi. Remplacez la variable **state** par **present**.

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to ensure the presence of a dnsforwardzone in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the presence of a dnsforwardzone for example.com to 8.8.8.8
  ipadnsforwardzone:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: example.com
      forwarders:
          - 8.8.8.8
      forwardpolicy: first
      skip_overlap_check: true
      state: present
```

6. Enregistrer le fichier.

7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-forwardzone.yml
```

Ressources supplémentaires

- Voir le fichier **README-dnsforwardzone.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

29.12. S'ASSURER QU'UNE ZONE DE TRANSFERT DNS A PLUSIEURS TRANSITAIRES DANS IDM À L'AIDE D'ANSIBLE

Cette section décrit comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour s'assurer qu'une zone de transfert DNS dans IdM a plusieurs transitaires. Dans l'exemple de procédure ci-dessous, l'administrateur IdM s'assure que la zone de transfert DNS pour **example.com** est transférée vers **8.8.8.8** et **4.4.4.4**.

Conditions préalables

- 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**.

- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **forwarders-absent.yml** Ansible playbook. Par exemple :

```
cp forwarders-absent.yml ensure-presence-multiple-forwarders.yml
```

4. Ouvrez le fichier **ensure-presence-multiple-forwarders.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes :
   a. Modifiez la variable **name** du playbook en **Playbook to ensure the presence of multiple forwarders in a dnsforwardzone in IdM DNS**.
   b. Dans la section **tasks**, changez le **name** de la tâche en **Ensure presence of 8.8.8.8 and 4.4.4.4 forwarders in dnsforwardzone for example.com**.
   c. Dans la section **tasks**, remplacez le titre **ipadnsconfig** par **ipadnsforwardzone**.
   d. Dans la section **ipadnsforwardzone**:
      i. Ajoutez la variable **ipaadmin_password** et définissez-la comme votre mot de passe d'administrateur IdM.
      ii. Ajoutez la variable **name** et fixez-la à **example.com**.
      iii. Dans la section **forwarders**:
         A. Supprimer les lignes **ip_address** et **port**.
         B. Ajoutez l'adresse IP des serveurs DNS dont vous voulez vous assurer de la présence, précédée d'un tiret :

```
    - 8.8.8.8
    - 4.4.4.4
```

iv. Modifier la variable d'état pour qu'elle devienne présente.

Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: name: Playbook to ensure the presence of multiple forwarders in a dnsforwardzone
in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure presence of 8.8.8.8 and 4.4.4.4 forwarders in dnsforwardzone for
example.com
  ipadnsforwardzone:
      ipaadmin_password: "{{ ipaadmin_password }}"
     name: example.com
      forwarders:
          - 8.8.8.8
          - 4.4.4.4
      state: present
```

6. Enregistrer le fichier.

7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-
multiple-forwarders.yml
```

Ressources supplémentaires

- Voir le fichier **README-dnsforwardzone.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

29.13. S'ASSURER QU'UNE ZONE DE TRANSFERT DNS EST DÉSACTIVÉE DANS L'IDM À L'AIDE D'ANSIBLE

Cette section décrit comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour s'assurer qu'une zone de transfert DNS est désactivée dans IdM. Dans l'exemple de procédure ci-dessous, l'administrateur IdM s'assure que la zone de transfert DNS pour **example.com** est désactivée.

Conditions préalables

- 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**.

- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **forwarders-absent.yml** Ansible playbook. Par exemple :

```
$ cp forwarders-absent.yml ensure-disabled-forwardzone.yml
```

4. Ouvrez le fichier **ensure-disabled-forwardzone.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes :
   a. Modifiez la variable **name** du playbook en **Playbook to ensure a dnsforwardzone is disabled in IdM DNS**.
   b. Dans la section **tasks**, changez le **name** de la tâche en **Ensure a dnsforwardzone for example.com is disabled**.
   c. Dans la section **tasks**, remplacez le titre **ipadnsconfig** par **ipadnsforwardzone**.
   d. Dans la section **ipadnsforwardzone**:
      i. Ajoutez la variable **ipaadmin_password** et définissez-la comme votre mot de passe d'administrateur IdM.
      ii. Ajoutez la variable **name** et fixez-la à **example.com**.
      iii. Retirer toute la section **forwarders**.
      iv. Remplacez la variable **state** par **disabled**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to ensure a dnsforwardzone is disabled in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
```

```
tasks:
- name: Ensure a dnsforwardzone for example.com is disabled
ipadnsforwardzone:
    ipaadmin_password: "{{ ipaadmin_password }}"
    name: example.com
    state: disabled
```

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-disabled-forwardzone.yml
```

Ressources supplémentaires

- Voir le fichier **README-dnsforwardzone.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

29.14. GARANTIR L'ABSENCE D'UNE ZONE DE TRANSFERT DNS DANS L'IDM À L'AIDE D'ANSIBLE

Cette section décrit comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour garantir l'absence d'une zone de transfert DNS dans IdM. Dans l'exemple de procédure ci-dessous, l'administrateur IdM s'assure de l'absence d'une zone de transfert DNS pour **example.com**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **forwarders-absent.yml** Ansible playbook. Par exemple :

```
$ cp forwarders-absent.yml ensure-absence-forwardzone.yml
```

4. Ouvrez le fichier **ensure-absence-forwardzone.yml** pour le modifier.

5. Adaptez le fichier en définissant les variables suivantes :

   a. Modifiez la variable **name** du playbook en **Playbook to ensure the absence of a dnsforwardzone in IdM DNS**.

   b. Dans la section **tasks**, changez le **name** de la tâche en **Ensure the absence of a dnsforwardzone for example.com**.

   c. Dans la section **tasks**, remplacez le titre **ipadnsconfig** par **ipadnsforwardzone**.

   d. Dans la section **ipadnsforwardzone**:

      i. Ajoutez la variable **ipaadmin_password** et définissez-la comme votre mot de passe d'administrateur IdM.

      ii. Ajoutez la variable **name** et fixez-la à **example.com**.

      iii. Retirer toute la section **forwarders**.

      iv. Laissez la variable **state** comme **absent**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Playbook to ensure the absence of a dnsforwardzone in IdM DNS
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure the absence of a dnsforwardzone for example.com
    ipadnsforwardzone:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: example.com
      state: absent
```

6. Enregistrer le fichier.

7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-
forwardzone.yml
```

Ressources supplémentaires

- Voir le fichier **README-dnsforwardzone.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.

29.15. RESSOURCES SUPPLÉMENTAIRES

- Transfert DNS

CHAPITRE 30. UTILISER ANSIBLE POUR GÉRER LES ENREGISTREMENTS DNS DANS IDM

Ce chapitre décrit comment gérer les enregistrements DNS dans Identity Management (IdM) à l'aide d'un playbook Ansible. En tant qu'administrateur IdM, vous pouvez ajouter, modifier et supprimer des enregistrements DNS dans IdM. Ce chapitre contient les sections suivantes :

- Assurer la présence des enregistrements DNS A et AAAA dans l'IdM en utilisant Ansible
- Assurer la présence des enregistrements DNS A et PTR dans IdM en utilisant Ansible
- Assurer la présence de plusieurs enregistrements DNS dans IdM en utilisant Ansible
- Assurer la présence de plusieurs enregistrements CNAME dans IdM en utilisant Ansible
- Assurer la présence d'un enregistrement SRV dans IdM en utilisant Ansible

30.1. ENREGISTREMENTS DNS DANS L'IDM

La gestion des identités (IdM) prend en charge de nombreux types d'enregistrements DNS. Les quatre types suivants sont les plus fréquemment utilisés :

A

Il s'agit d'une correspondance de base entre un nom d'hôte et une adresse IPv4. Le nom d'un enregistrement A est un nom d'hôte, tel que **www**. La valeur **IP Address** d'un enregistrement A est une adresse IPv4, telle que **192.0.2.1**.
Pour plus d'informations sur les enregistrements A, voir RFC 1035.

AAAA

Il s'agit d'une correspondance de base entre un nom d'hôte et une adresse IPv6. Le nom d'un enregistrement AAAA est un nom d'hôte, tel que **www**. La valeur **IP Address** est une adresse IPv6, telle que **2001:DB8::1111**.
Pour plus d'informations sur les enregistrements AAAA, voir RFC 3596.

SRV

Service (SRV) resource records font correspondre les noms de service au nom DNS du serveur qui fournit ce service particulier. Par exemple, ce type d'enregistrement peut associer un service tel qu'un annuaire LDAP au serveur qui le gère.
Le nom d'un enregistrement SRV a le format suivant ***_service._protocol***, tel que **_ldap._tcp**. Les options de configuration des enregistrements SRV comprennent la priorité, le poids, le numéro de port et le nom d'hôte du service cible.

Pour plus d'informations sur les enregistrements SRV, voir RFC 2782.

PTR

Un enregistrement de pointeur (PTR) ajoute un enregistrement DNS inverse, qui fait correspondre une adresse IP à un nom de domaine.

NOTE

Toutes les recherches DNS inversées pour les adresses IPv4 utilisent des entrées inversées définies dans le domaine **in-addr.arpa.**. L'adresse inversée, sous une forme lisible par l'homme, est l'inverse exact de l'adresse IP normale, à laquelle est ajouté le domaine **in-addr.arpa.**. Par exemple, pour l'adresse réseau **192.0.2.0/24**, la zone inversée est **2.0.192.in-addr.arpa**.

Le nom d'enregistrement d'un PTR doit être au format standard spécifié dans le RFC 1035, étendu dans le RFC 2317 et le RFC 3596. La valeur du nom d'hôte doit être un nom d'hôte canonique de l'hôte pour lequel vous souhaitez créer l'enregistrement.

NOTE

Les zones inversées peuvent également être configurées pour les adresses IPv6, avec des zones dans le domaine **.ip6.arpa.**. Pour plus d'informations sur les zones inversées IPv6, voir la RFC 3596.

Lors de l'ajout d'enregistrements de ressources DNS, il convient de noter que de nombreux enregistrements nécessitent des données différentes. Par exemple, un enregistrement CNAME nécessite un nom d'hôte, tandis qu'un enregistrement A nécessite une adresse IP. Dans l'interface Web de l'IdM, les champs du formulaire d'ajout d'un nouvel enregistrement sont mis à jour automatiquement pour refléter les données requises pour le type d'enregistrement sélectionné.

30.2. OPTIONS COURANTES D'IPA DNSRECORD-*

Cette section décrit les options que vous pouvez utiliser pour ajouter, modifier et supprimer les types d'enregistrements de ressources DNS les plus courants dans la gestion des identités (IdM) :

- A (IPv4)
- AAAA (IPv6)
- SRV
- PTR

Dans **Bash**, vous pouvez définir plusieurs entrées en énumérant les valeurs dans une liste séparée par des virgules à l'intérieur d'accolades, comme **--option={val1,val2,val3}**.

Tableau 30.1. Options générales d'enregistrement

Option	*Description*
--ttl=*number*	Définit la durée de vie de l'enregistrement.
--structured	Analyse les enregistrements DNS bruts et les renvoie dans un format structuré.

Tableau 30.2. \Options d'enregistrement

Option	*Description*	*Examples*
--a-rec=*ARECORD*	Transmet un seul enregistrement A ou une liste d'enregistrements A.	**ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123**
	Peut créer un enregistrement A de type "wildcard" avec une adresse IP donnée.	**ipa dnsrecord-add idm.example.com "*" --a-rec=192.168.122.123** [a]
--a-ip-address=*string*	Indique l'adresse IP de l'enregistrement. Lors de la création d'un enregistrement, l'option permettant de spécifier la valeur de l'enregistrement **A** est **--a-rec**. Toutefois, lors de la modification d'un enregistrement **A**, l'option **--a-rec** est utilisée pour spécifier la valeur actuelle de l'enregistrement **A**. La nouvelle valeur est définie à l'aide de l'option **--a-ip-address**.	**ipa dnsrecord-mod idm.example.com --a-rec 192.168.122.123 --a-ip-address 192.168.122.124**

[a] L'exemple crée un enregistrement **A** avec l'adresse IP 192.0.2.123.

Tableau 30.3. \Options d'enregistrement "AAAA"

Option	*Description*	*Example*
--aaaa-rec=*AAAARECORD*	Transmet un seul enregistrement AAAA (IPv6) ou une liste d'enregistrements AAAA.	**ipa dnsrecord-add idm.example.com www --aaaa-rec 2001:db8::1231:5675**
--aaaa-ip-address=*string*	Indique l'adresse IPv6 de l'enregistrement. Lors de la création d'un enregistrement, l'option permettant de spécifier la valeur de l'enregistrement **A** est **--aaaa-rec**. Toutefois, lors de la modification d'un enregistrement **A**, l'option **--aaaa-rec** est utilisée pour spécifier la valeur actuelle de l'enregistrement **A**. La nouvelle valeur est définie à l'aide de l'option **--a-ip-address**.	**ipa dnsrecord-mod idm.example.com --aaaa-rec 2001:db8::1231:5675 --aaaa-ip-address 2001:db8::1231:5676**

Tableau 30.4. \Options d'enregistrement "PTR

Option	*Description*	*Example*

Option	*Description*	*Example*
--ptr-rec=*PTRRECORD*	Transmet un seul enregistrement PTR ou une liste d'enregistrements PTR. Lors de l'ajout d'un enregistrement DNS inversé, le nom de la zone utilisé avec la commande **ipa dnsrecord-add** est inversé par rapport à l'utilisation pour l'ajout d'autres enregistrements DNS. Généralement, l'adresse IP de l'hôte est le dernier octet de l'adresse IP dans un réseau donné. Le premier exemple à droite ajoute un enregistrement PTR pour **server4.idm.example.com** avec l'adresse IPv4 **192.168.122.4.**. Le deuxième exemple ajoute une entrée DNS inverse à la zone inverse **0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.** IPv6 pour l'hôte **server2.example.com** avec l'adresse IP **2001:DB8::1111**.	**ipa dnsrecord-add 122.168.192.in-addr.arpa 4 --ptr-rec server4.idm.example.com.** **$ ipa dnsrecord-add 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. 1.1.1.0.0.0.0.0.0.0.0.0.0.0.0.0 --ptr-rec server2.idm.example.com.**
--ptr-hostname=*string*	Indique le nom d'hôte de l'enregistrement.	

Tableau 30.5. \SRV Options d'enregistrement

Option	*Description*	*Example*
--srv-rec=*SRVRECORD*	Transmet un seul enregistrement SRV ou une liste d'enregistrements SRV. Dans les exemples ci-contre, **_ldap._tcp** définit le type de service et le protocole de connexion pour l'enregistrement SRV. L'option **--srv-rec** définit les valeurs de priorité, de poids, de port et de cible. Les valeurs de poids de 51 et 49 dans les exemples totalisent 100 et représentent la probabilité, en pourcentage, qu'un enregistrement particulier soit utilisé.	**# ipa dnsrecord-add idm.example.com _ldap._tcp --srv-rec="0 51 389 server1.idm.example.com."** **# ipa dnsrecord-add server.idm.example.com _ldap._tcp --srv-rec="1 49 389 server2.idm.example.com."**
--srv-priority=*number*	Définit la priorité de l'enregistrement. Il peut y avoir plusieurs enregistrements SRV pour un type de service. La priorité (0 - 65535) définit le rang de l'enregistrement ; plus le chiffre est bas, plus la priorité est élevée. Un service doit utiliser en premier l'enregistrement ayant la priorité la plus élevée.	**# ipa dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="1 49 389 server2.idm.example.com." --srv-priority=0**

Option	*Description*	*Example*
--srv-weight=*number*	Définit le poids de l'enregistrement. Cela permet de déterminer l'ordre des enregistrements SRV ayant la même priorité. La somme des poids définis doit être égale à 100, ce qui représente la probabilité (en pourcentage) qu'un enregistrement particulier soit utilisé.	**# ipa dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 49 389 server2.idm.example.com." --srv-weight=60**
--srv-port=*number*	Indique le port du service sur l'hôte cible.	**# ipa dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 60 389 server2.idm.example.com." --srv-port=636**
--srv-target=*string*	Indique le nom de domaine de l'hôte cible. Il peut s'agir d'un seul point (.) si le service n'est pas disponible dans le domaine.	

Ressources supplémentaires

- Exécuter **ipa dnsrecord-add --help**.

30.3. ASSURER LA PRÉSENCE DES ENREGISTREMENTS DNS A ET AAAA DANS L'IDM EN UTILISANT ANSIBLE

Cette section montre comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour s'assurer que les enregistrements A et AAAA d'un hôte IdM particulier sont présents. Dans l'exemple utilisé dans la procédure ci-dessous, un administrateur IdM s'assure de la présence d'enregistrements A et AAAA pour **host1** dans la zone DNS **idm.example.com**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.

- La zone **idm.example.com** existe et est gérée par IdM DNS. Pour plus d'informations sur l'ajout d'une zone DNS primaire dans IdM DNS, voir Utilisation des playbooks Ansible pour gérer les zones IdM DNS.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **ensure-A-and-AAAA-records-are-present.yml** Ansible playbook. Par exemple :

```
$ cp ensure-A-and-AAAA-records-are-present.yml ensure-A-and-AAAA-records-are-present-copy.yml
```

4. Ouvrez le fichier **ensure-A-and-AAAA-records-are-present-copy.yml** pour le modifier.

5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnsrecord** task :

   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Fixer la variable **zone_name** à **idm.example.com**.
   - Dans la variable **records**, fixez la variable **name** à **host1**, et la variable **a_ip_address** à **192.168.122.123**.
   - Dans la variable **records**, fixez la variable **name** à **host1**, et la variable **aaaa_ip_address** à **::1**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Ensure A and AAAA records are present
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure A and AAAA records are present
  - name: Ensure that 'host1' has A and AAAA records.
    ipadnsrecord:
      ipaadmin_password: "{{ ipaadmin_password }}"
      zone_name: idm.example.com
      records:
      - name: host1
        a_ip_address: 192.168.122.123
      - name: host1
        aaaa_ip_address: ::1
```

6. Enregistrer le fichier.

7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-A-and-AAAA-records-are-present-copy.yml
```

Ressources supplémentaires

- Voir les enregistrements DNS dans IdM.
- Voir le fichier **README-dnsrecord.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks Ansible dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord**.

30.4. ASSURER LA PRÉSENCE DES ENREGISTREMENTS DNS A ET PTR DANS IDM EN UTILISANT ANSIBLE

Cette section montre comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour s'assurer qu'un enregistrement A pour un hôte IdM particulier est présent, avec un enregistrement PTR correspondant. Dans l'exemple utilisé dans la procédure ci-dessous, un administrateur IdM s'assure de la présence d'enregistrements A et PTR pour **host1** avec une adresse IP de **192.168.122.45** dans la zone **idm.example.com**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.
- La zone **idm.example.com** DNS existe et est gérée par IdM DNS. Pour plus d'informations sur l'ajout d'une zone DNS primaire dans IdM DNS, voir Utilisation des playbooks Ansible pour gérer les zones IdM DNS.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **ensure-dnsrecord-with-reverse-is-present.yml** Ansible playbook. Par exemple :

```
$ cp ensure-dnsrecord-with-reverse-is-present.yml ensure-dnsrecord-with-reverse-is-present-copy.yml
```

4. Ouvrez le fichier **ensure-dnsrecord-with-reverse-is-present-copy.yml** pour le modifier.

5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnsrecord** task :

   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Fixer la variable **name** à **host1**.
   - Fixer la variable **zone_name** à **idm.example.com**.
   - Fixer la variable **ip_address** à 192.168.122.45.
   - Fixer la variable **create_reverse** à **yes**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Ensure DNS Record is present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure that dns record is present
  - ipadnsrecord:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: host1
      zone_name: idm.example.com
      ip_address: 192.168.122.45
      create_reverse: yes
      state: present
```

6. Enregistrer le fichier.

7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-dnsrecord-with-reverse-is-present-copy.yml
```

Ressources supplémentaires

- Voir les enregistrements DNS dans IdM.

- Voir le fichier **README-dnsrecord.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks Ansible dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord**.

30.5. ASSURER LA PRÉSENCE DE PLUSIEURS ENREGISTREMENTS DNS DANS IDM EN UTILISANT ANSIBLE

Cette section montre comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour s'assurer que plusieurs valeurs sont associées à un enregistrement DNS IdM particulier. Dans l'exemple utilisé dans la procédure ci-dessous, un administrateur IdM s'assure de la présence de plusieurs enregistrements A pour **host1** dans la zone DNS **idm.example.com**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.
- La zone **idm.example.com** existe et est gérée par IdM DNS. Pour plus d'informations sur l'ajout d'une zone DNS primaire dans IdM DNS, voir Utilisation des playbooks Ansible pour gérer les zones IdM DNS.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **ensure-presence-multiple-records.yml** Ansible playbook. Par exemple :

```
$ cp ensure-presence-multiple-records.yml ensure-presence-multiple-records-copy.yml
```

4. Ouvrez le fichier **ensure-presence-multiple-records-copy.yml** pour le modifier.

5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnsrecord** task :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Dans la section **records**, fixez la variable **name** à **host1**.
   - Dans la section **records**, la variable **zone_name** est remplacée par **idm.example.com**.
   - Dans la section **records**, la variable **a_rec** doit être remplacée par **192.168.122.112** et **192.168.122.122**.
   - Définir un deuxième enregistrement dans la section **records**:
     - Fixer la variable **name** à **host1**.
     - Fixer la variable **zone_name** à **idm.example.com**.
     - Fixer la variable **aaaa_rec** à **::1**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Test multiple DNS Records are present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure that multiple dns records are present
  - ipadnsrecord:
      ipaadmin_password: "{{ ipaadmin_password }}"
      records:
        - name: host1
          zone_name: idm.example.com
          a_rec: 192.168.122.112
          a_rec: 192.168.122.122
        - name: host1
          zone_name: idm.example.com
          aaaa_rec: ::1
```

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-multiple-records-copy.yml
```

Ressources supplémentaires

- Voir les enregistrements DNS dans IdM.
- Voir le fichier **README-dnsrecord.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks Ansible dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord**.

30.6. ASSURER LA PRÉSENCE DE PLUSIEURS ENREGISTREMENTS CNAME DANS IDM EN UTILISANT ANSIBLE

Un enregistrement de nom canonique (enregistrement CNAME) est un type d'enregistrement de ressources dans le système de noms de domaine (DNS) qui fait correspondre un nom de domaine, un alias, à un autre nom, le nom canonique.

Les enregistrements CNAME peuvent s'avérer utiles lorsque plusieurs services sont exécutés à partir d'une même adresse IP : par exemple, un service FTP et un service web, chacun fonctionnant sur un port différent.

Cette section montre comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour s'assurer que plusieurs enregistrements CNAME sont présents dans le DNS IdM. Dans l'exemple utilisé dans la procédure ci-dessous, **host03** est à la fois un serveur HTTP et un serveur FTP. L'administrateur IdM s'assure de la présence des enregistrements CNAME **www** et **ftp** pour l'enregistrement A **host03** dans la zone **idm.example.com**.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.
- La zone **idm.example.com** existe et est gérée par IdM DNS. Pour plus d'informations sur l'ajout d'une zone DNS primaire dans IdM DNS, voir Utilisation des playbooks Ansible pour gérer les zones IdM DNS.
- L'enregistrement A de **host03** existe dans la zone **idm.example.com**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **ensure-CNAME-record-is-present.yml** Ansible playbook. Par exemple :

```
$ cp ensure-CNAME-record-is-present.yml ensure-CNAME-record-is-present-copy.yml
```

4. Ouvrez le fichier **ensure-CNAME-record-is-present-copy.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnsrecord** task :
   - (Facultatif) Adaptez la description de la pièce fournie par le site **name**.
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Fixer la variable **zone_name** à **idm.example.com**.
   - Dans la section des variables de **records**, définissez les variables et valeurs suivantes :
     - Fixer la variable **name** à **www**.
     - Fixer la variable **cname_hostname** à **host03**.
     - Fixer la variable **name** à **ftp**.
     - Fixer la variable **cname_hostname** à **host03**.

   Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Ensure that 'www.idm.example.com' and 'ftp.idm.example.com' CNAME records
point to 'host03.idm.example.com'.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  - ipadnsrecord:
      ipaadmin_password: "{{ ipaadmin_password }}"
      zone_name: idm.example.com
      records:
      - name: www
        cname_hostname: host03
      - name: ftp
        cname_hostname: host03
```

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-
CNAME-record-is-present.yml
```

Ressources supplémentaires

- Voir le fichier **README-dnsrecord.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks Ansible dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord**.

30.7. ASSURER LA PRÉSENCE D'UN ENREGISTREMENT SRV DANS IDM EN UTILISANT ANSIBLE

Un enregistrement de service DNS (SRV) définit le nom d'hôte, le numéro de port, le protocole de transport, la priorité et le poids d'un service disponible dans un domaine. Dans la gestion des identités (IdM), vous pouvez utiliser les enregistrements SRV pour localiser les serveurs IdM et les répliques.

Cette section montre comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour s'assurer qu'un enregistrement SRV est présent dans le DNS IdM. Dans l'exemple utilisé dans la procédure ci-dessous, un administrateur IdM s'assure de la présence de l'enregistrement SRV **_kerberos._udp.idm.example.com** avec la valeur **10 50 88 idm.example.com**. Cela définit les valeurs suivantes :

- Il fixe la priorité du service à 10.
- Il fixe le poids du service à 50.
- Il fixe le port à utiliser par le service à 88.

Conditions préalables

- 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**.
- Vous connaissez le mot de passe de l'administrateur IdM.
- La zone **idm.example.com** existe et est gérée par IdM DNS. Pour plus d'informations sur l'ajout d'une zone DNS primaire dans IdM DNS, voir Utilisation des playbooks Ansible pour gérer les zones IdM DNS.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord**:

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **ensure-SRV-record-is-present.yml** Ansible playbook. Par exemple :

```
$ cp ensure-SRV-record-is-present.yml ensure-SRV-record-is-present-copy.yml
```

4. Ouvrez le fichier **ensure-SRV-record-is-present-copy.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes dans la section **ipadnsrecord** task :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Fixer la variable **name** à **_kerberos._udp.idm.example.com**.
   - Fixer la variable **srv_rec** à **'10 50 88 idm.example.com'**.
   - Fixer la variable **zone_name** à **idm.example.com**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- name: Test multiple DNS Records are present.
  hosts: ipaserver
  become: true
  gather_facts: false

  tasks:
  # Ensure a SRV record is present
  - ipadnsrecord:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: _kerberos._udp.idm.example.com
      srv_rec: '10 50 88 idm.example.com'
      zone_name: idm.example.com
      state: present
```

6. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-SRV-record-is-present.yml
```

Ressources supplémentaires

- Voir les enregistrements DNS dans IdM.
- Voir le fichier **README-dnsrecord.md** dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks Ansible dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord**.

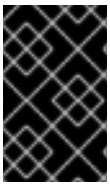
CHAPITRE 31. UTILISER ANSIBLE POUR MONTER AUTOMATIQUEMENT DES PARTAGES NFS POUR LES UTILISATEURS IDM

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

31.1. 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

31.2. 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.
```

31.3. 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**.

31.4. 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 .

31.5. 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
```

31.6. 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
```

31.7. 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
```

31.8. 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.

CHAPITRE 32. UTILISER ANSIBLE POUR INTÉGRER IDM AVEC LES DOMAINES NIS ET LES GROUPES NETS

32.1. LE SNI ET SES AVANTAGES

Dans les environnements UNIX, le service d'information sur le réseau (NIS) est un moyen courant de gérer les identités et l'authentification de manière centralisée. NIS, qui s'appelait à l'origine **Yellow Pages** (YP), gère de manière centralisée les informations d'authentification et d'identité telles que :

- Utilisateurs et mots de passe
- Noms d'hôtes et adresses IP
- Groupes POSIX

Pour les infrastructures de réseau modernes, NIS est considéré comme trop peu sûr car, par exemple, il ne fournit pas d'authentification de l'hôte et les données ne sont pas envoyées de manière cryptée sur le réseau. Pour contourner ces problèmes, le NIS est souvent intégré à d'autres protocoles afin de renforcer la sécurité.

Si vous utilisez la gestion des identités (IdM), vous pouvez utiliser le plug-in du serveur NIS pour connecter les clients qui ne peuvent pas être entièrement migrés vers IdM. IdM intègre les groupes de réseau et autres données NIS dans le domaine IdM. En outre, vous pouvez facilement migrer les identités des utilisateurs et des hôtes d'un domaine NIS vers IdM.

Les groupes nets peuvent être utilisés partout où les groupes NIS sont attendus.

Ressources supplémentaires

- NIS dans IdM
- Groupes nets NIS dans IdM
- Migration de NIS vers la gestion des identités

32.2. NIS DANS IDM

Objets NIS dans IdM

Les objets NIS sont intégrés et stockés dans le serveur d'annuaire conformément à la RFC 2307. IdM crée des objets NIS dans l'annuaire LDAP et les clients les récupèrent, par exemple, via System Security Services Daemon (SSSD) ou **nss_ldap** en utilisant une connexion LDAP cryptée.

IdM gère les groupes de réseau, les comptes, les groupes, les hôtes et d'autres données. IdM utilise un auditeur NIS pour faire correspondre les mots de passe, les groupes et les groupes de réseau aux entrées IdM.

Plug-ins NIS dans IdM

Pour la prise en charge de NIS, IdM utilise les modules d'extension suivants, fournis dans le paquetage **slapi-nis** :

Plug-in pour le serveur NIS

Le plug-in NIS Server permet au serveur LDAP intégré à IdM de jouer le rôle de serveur NIS pour les clients. Dans ce rôle, Directory Server génère et met à jour dynamiquement les cartes NIS en fonction

de la configuration. En utilisant le plug-in, IdM sert les clients utilisant le protocole NIS en tant que serveur NIS.

Plug-in de compatibilité des schémas

Le plug-in Schema Compatibility permet au back-end du serveur d'annuaire de fournir une vue alternative des entrées stockées dans une partie de l'arborescence des informations d'annuaire (DIT). Il permet notamment d'ajouter, de supprimer ou de renommer des valeurs d'attributs et, éventuellement, de récupérer des valeurs d'attributs à partir de plusieurs entrées de l'arborescence. Pour plus de détails, voir le fichier **/usr/share/doc/slapi-nis-*version*/sch-getting-started.txt** fichier.

32.3. GROUPES NETS NIS DANS IDM

Les entités NIS peuvent être stockées dans des groupes nets. Par rapport aux groupes UNIX, les groupes nets prennent en charge les éléments suivants

- Groupes imbriqués (groupes en tant que membres d'autres groupes).
- Regroupement des hôtes.

Un groupe de réseau définit un ensemble d'informations : hôte, utilisateur et domaine. Cet ensemble est appelé **triple**. Ces trois champs peuvent contenir :

- Une valeur.
- Un tiret (**-**), qui indique "aucune valeur valide"
- Pas de valeur. Un champ vide indique un caractère générique.

> (*host.example.com*,,*nisdomain.example.com*)
> (-,*user*,*nisdomain.example.com*)

Lorsqu'un client demande un groupe net NIS, IdM traduit l'entrée LDAP :

- à une carte NIS traditionnelle et l'envoie au client via le protocole NIS à l'aide du module d'extension NIS.
- dans un format LDAP conforme à la norme RFC 2307 ou RFC 2307bis.

32.4. UTILISER ANSIBLE POUR S'ASSURER QU'UN NETGROUP EST PRÉSENT

Vous pouvez utiliser un playbook Ansible pour vous assurer qu'un groupe net IdM est présent. L'exemple décrit comment s'assurer que le groupe **TestNetgroup1** est présent.

Conditions préalables

- 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.

- Vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM dans le répertoire ***~/MyPlaybooks/*** dans le répertoire
- Vous avez stocké votre site **ipaadmin_password** dans le coffre-fort **secret.yml** Ansible.

Procédure

1. Créez votre fichier Ansible playbook **netgroup-present.yml** avec le contenu suivant :

```
---
- name: Playbook to manage IPA netgroup.
  hosts: ipaserver
  become: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure netgroup members are present
    ipanetgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: TestNetgroup1
```

2. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory_/netgroup-
present.yml
```

Ressources supplémentaires

- NIS dans IdM
- **/usr/share/doc/ansible-freeipa/README-netgroup.md**
- **/usr/share/doc/ansible-freeipa/playbooks/netgroup**

32.5. UTILISER ANSIBLE POUR S'ASSURER QUE LES MEMBRES SONT PRÉSENTS DANS UN NETGROUP

Vous pouvez utiliser un script Ansible pour vous assurer que les utilisateurs, les groupes et les groupes nets IdM sont membres d'un groupe net. L'exemple décrit comment s'assurer que le groupe **TestNetgroup1** a les membres suivants :

- Les utilisateurs de **user1** et **user2** IdM
- Le groupe IdM **group1**
- Le groupe net **admins**
- Un hôte **idmclient1** qui est un client IdM

Conditions préalables

- 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.
  - Vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM dans le répertoire ***~/MyPlaybooks/*** dans le répertoire
  - Vous avez stocké votre site **ipaadmin_password** dans le coffre-fort **secret.yml** Ansible.
- Le groupe de réseau **TestNetgroup1** IdM existe.
- Les utilisateurs **user1** et **user2** IdM existent.
- Le groupe IdM **group1** existe.
- Le groupe de réseau **admins** IdM existe.

Procédure

1. Créez votre fichier Ansible playbook **IdM-members-present-in-a-netgroup.yml** avec le contenu suivant :

```
---
- name: Playbook to manage IPA netgroup.
  hosts: ipaserver
  become: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure netgroup members are present
    ipanetgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: TestNetgroup1
      user: user1,user2
      group: group1
      host: idmclient1
      netgroup: admins
      action: member
```

2. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory_/IdM-
members-present-in-a-netgroup.yml
```

Ressources supplémentaires

- NIS dans IdM
- **/usr/share/doc/ansible-freeipa/README-netgroup.md**
- **/usr/share/doc/ansible-freeipa/playbooks/netgroup**

32.6. UTILISER ANSIBLE POUR S'ASSURER QU'UN MEMBRE EST ABSENT D'UN NETGROUP

Vous pouvez utiliser un playbook Ansible pour vous assurer que les utilisateurs IdM sont membres d'un netgroup. L'exemple décrit comment s'assurer que le groupe **TestNetgroup1** ne compte pas l'utilisateur IdM **user1** parmi ses membres. netgroup

Conditions préalables

- 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.
  - Vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM dans le répertoire ***~/MyPlaybooks/*** dans le répertoire
  - Vous avez stocké votre site **ipaadmin_password** dans le coffre-fort **secret.yml** Ansible.
- Le groupe de réseau **TestNetgroup1** existe.

Procédure

1. Créez votre fichier Ansible playbook **IdM-member-absent-from-a-netgroup.yml** avec le contenu suivant :

```
---
- name: Playbook to manage IPA netgroup.
  hosts: ipaserver
  become: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure netgroup user, "user1", is absent
    ipanetgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: TestNetgroup1
      user: "user1"
      action: member
      state: absent
```

2. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory_/IdM-
member-absent-from-a-netgroup.yml
```

Ressources supplémentaires

- NIS dans IdM
- **/usr/share/doc/ansible-freeipa/README-netgroup.md**

- **/usr/share/doc/ansible-freeipa/playbooks/netgroup**

32.7. UTILISER ANSIBLE POUR S'ASSURER QU'UN NETGROUP EST ABSENT

Vous pouvez utiliser un playbook Ansible pour vous assurer qu'un netgroup n'existe pas dans la gestion des identités (IdM). L'exemple décrit comment s'assurer que le groupe **TestNetgroup1** n'existe pas dans votre domaine IdM.

Conditions préalables

- 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.
  - Vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM dans le répertoire ***~/MyPlaybooks/*** dans le répertoire
  - Vous avez stocké votre site **ipaadmin_password** dans le coffre-fort **secret.yml** Ansible.

Procédure

1. Créez votre fichier Ansible playbook **netgroup-absent.yml** avec le contenu suivant :

```
---
- name: Playbook to manage IPA netgroup.
  hosts: ipaserver
  become: no

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure netgroup my_netgroup1 is absent
    ipanetgroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: my_netgroup1
      state: absent
```

2. Exécutez le manuel de jeu :

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory_/netgroup-
absent.yml
```

Ressources supplémentaires

- NIS dans IdM
- **/usr/share/doc/ansible-freeipa/README-netgroup.md**
- **/usr/share/doc/ansible-freeipa/playbooks/netgroup**