Red Hat Enterprise Linux 9

Ansible を使用した Identity Management のインストールと管理

Ansible を使用した IdM 環境の維持

Last Updated: 2024-06-27

Red Hat Enterprise Linux 9 Ansible を使用した Identity Management のインストールと管理

Ansible を使用した IdM 環境の維持

法律上の通知

概要

Red Hat は、管理者が Ansible を使用して Red Hat Identity Management (IdM) を実行できるようにする ansible-freeipa パッケージを提供します。Playbook を使用して IdM をインストールし、ユーザー、グループ、ホスト、アクセス制御、設定を管理できます。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

Jira からのフィードバック送信 (アカウントが必要)

1. Jira の Web サイトにログインします。
2. 上部のナビゲーションバーで **Create** をクリックします。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
5. ダイアログの下部にある **Create** をクリックします。

第1章 ANSIBLE の用語

本書の章では、公式の Ansible 用語を使用します。Ansible の用語に慣れていない場合は、公式の Ansible アップストリームドキュメント、特に次のセクションを読んでから続行してください。

- Ansible のセクションの基本概念 は、Ansible で最も一般的に使用される概念の概要を説明します。
- ユーザーガイド では、コマンドラインの使用、インベントリーの使用、データの操作、タスク、Play および Playbook の記述、Playbook の実行など、Ansible の使用を開始する際の最も一般的な状況や疑問点について概説しています。
- インベントリーの構築方法 では、インベントリーの設計方法のヒントがあります。インベントリーは、Ansible がインフラストラクチャー内の複数の管理ノードまたはホストに対して機能するために使用するリストのグループです。
- Intro to playbooks では、設定の管理、マシンのデプロイ、および複雑なアプリケーションのデプロイのための反復可能で再利用可能なシステムとして、Ansible Playbook の概念を紹介します。
- Ansible ロール セクションでは、既知のファイル構造に基づいて特定の変数、タスク、およびハンドラーの読み込みを自動化する方法を説明します。
- 用語集 は、他の Ansible ドキュメントで使用される用語です。

第2章 ANSIBLE PLAYBOOK で IDENTITY MANAGEMENT サーバーのインストール

以下のセクションでは、Ansible を使用してシステムを IdM サーバーとして設定する方法を説明します。システムを IdM サーバーとして設定すると、IdM ドメインを確立し、システムが IdM クライアントに IdM サービスを提供できるようになります。デプロイメントは、Ansible ロール **ipaserver** により管理されます。

前提条件

- Ansibleと IdM の概念を理解している:
  - Ansible ロール
  - Ansible ノード
  - Ansible インベントリー
  - Ansible タスク
  - Ansible モジュール
  - Ansible プレイおよび Playbook

2.1. ANSIBLE と、IDM をインストールする利点

Ansible は、システムの設定、ソフトウェアのデプロイ、ローリング更新の実行に使用する自動化ツールです。Ansible には Identity Management (IdM) のサポートが含まれるため、Ansible モジュールを使用して、IdM サーバー、レプリカ、クライアント、または IdM トポロジー全体の設定などのインストールタスクを自動化できます。

IdM のインストールに Ansible を使用する利点
以下のリストは、手動インストールとは対照的に、Ansible を使用して Identity Management をインストールする利点を示しています。

- 管理ノードにログインする必要はありません。
- デプロイする各ホストに個別に設定する必要はありません。代わりに、完全なクラスターをデプロイするためのインベントリーファイルを 1 つ使用できます。
- ユーザーおよびホストを追加するなど、後で管理タスクにインベントリーファイルを再利用できます。IdM には関係のないタスクであっても、インベントリーファイルを再利用できます。

関連情報

- Automating Red Hat Identity Management installation
- Identity Management の計画
- IdM サーバーをインストールするためのシステムの準備

2.2. ANSIBLE-FREEIPA パッケージのインストール

以下の手順では、**ansible-freeipa** ロールをインストールする方法について説明します。

前提条件

- コントローラーが、有効なサブスクリプションを備えた Red Hat Enterprise Linux システムである。そうでない場合は、公式の Ansible ドキュメントの Installation guide で、代替のインストール方法を参照してください。
- コントローラーから、**SSH** プロトコルで管理ノードに到達できる。管理ノードが、コントローラーの **/root/.ssh/known_hosts** ファイルのリストに記載されていることを確認します。

手順

Ansible コントローラーで以下の手順を実行します。

1. 必要なリポジトリーを有効にします。

```
# subscription-manager repos --enable rhel-9-for-x86_64-appstream-rpms
```

2. IdM Ansible ロールをインストールします。

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

ロールが **/usr/share/ansible/roles/** ディレクトリーにインストールされます。

2.3. ファイルシステム内の ANSIBLE ロールの場所

デフォルトでは、**ansible-freeipa** ロールは **/usr/share/ansible/roles/** ディレクトリーにインストールされます。**ansible-freeipa** パッケージの構造は以下のとおりです。

- **/usr/share/ansible/roles/** ディレクトリーには、Ansible コントローラーの **ipaserver** ロール、**ipareplica** ロール、および **ipaclient** ロールが保存されています。各ロールディレクトリーには、サンプル、基本的な概要、ライセンス、および Markdown ファイルの **README.md** のロールに関する情報が保存されています。

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

- **/usr/share/doc/ansible-freeipa/** ディレクトリーには、Markdown ファイルの **README.md** に、各ロールおよびトポロジーに関する情報が保存されています。また、**playbooks/** サブディレクトリーも保存されています。

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

- **/usr/share/doc/ansible-freeipa/playbooks/** ディレクトリーは、Playbook のサンプルを保存します。

```
[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. 統合 DNS と、ROOT CA としての統合 CA を使用したデプロイメントのパラメーターの設定

以下の手順に従って、IdM 統合 DNS ソリューションを使用する環境で、統合 CA を持つ IdM サーバーを root CA としてインストールするようにインベントリーファイルを設定します。

注記

この手順のインベントリーは、**INI** 形式を使用します。または、**YAML** 形式または **JSON** 形式を使用できます。

手順

1. ~/**MyPlaybooks**/ ディレクトリーを作成します。

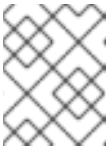
```
$ mkdir MyPlaybooks
```

2. ~/**MyPlaybooks**/**inventory** ファイルを作成します。
3. 編集するインベントリーファイルを開きます。IdM サーバーとして使用するホストの完全修飾ドメイン名 (**FQDN**) を指定します。**FQDN** が以下の基準を満たしていることを確認してください。
   - 英数字およびハイフン (-) のみが使用できる。たとえば、アンダーラインは使用できないため、DNS の障害が発生する原因となる可能性があります。
   - ホスト名がすべて小文字である。
4. IdM ドメインおよびレルムの情報を指定します。
5. 以下のオプションを追加して、統合 DNS を使用することを指定します。

```
ipaserver_setup_dns=true
```

6. DNS 転送設定を指定します。以下のいずれかのオプションを選択します。
   - インストーラーで **/etc/resolv.conf** ファイルのフォワーダーを使用する場合は、**ipaserver_auto_forwarders=true** オプションを使用します。**/etc/resolv.conf** ファイルで指定する nameserver が localhost 127.0.0.1 アドレスである場合、または仮想プライベートネットワークにあり、使用している DNS サーバーが通常パブリックインターネットから到達できない場合は、このオプションは使用しないでください。
   - **ipaserver_forwarders** を使用して、フォワーダーを手動で指定します。インストールプロセスにより、インストールした IdM サーバーの **/etc/named.conf** ファイルに、フォワーダーの IP アドレスが追加されます。

- 代わりにルート DNS サーバーを使用するように設定するには、**ipaserver_no_forwarders=true** オプションを使用します。

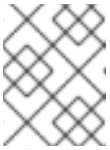


注記

DNS フォワーダーがないと、環境は分離され、インフラストラクチャー内の他の DNS ドメインからの名前は解決されません。

7. DNS の逆引きレコードとゾーンの設定を指定します。次のいずれかのオプションを選択します。

   - ゾーンがすでに解決可能である場合でも (逆引き) ゾーンの作成を許可するには、**ipaserver_allow_zone_overlap=true** オプションを使用します。
   - **ipaserver_reverse_zones** オプションを使用して、逆引きゾーンを手動で指定します。
   - インストーラーで逆引き DNS ゾーンを作成しない場合は、**ipaserver_no_reverse=true** オプションを使用します。



注記

オプションで、逆引きゾーンの管理に IdM を使用できます。代わりに、この目的で外部 DNS サービスを使用することもできます。

8. **admin** と **Directory Manager** のパスワードを指定します。Ansible Vault を使用してパスワードを保存し、Playbook ファイルから Vault ファイルを参照します。あるいは、安全性は低くなりますが、インベントリーファイルにパスワードを直接指定します。

9. (必要に応じて) IdM サーバーで使用する個別の **firewalld** ゾーンを指定します。カスタムゾーンを設定しないと、サービスがデフォルトの **firewalld** ゾーンに追加されます。事前定義されたデフォルトゾーンは **public** です。

重要

指定する **firewalld** ゾーンは存在し、永続的でなければなりません。

必要なサーバー情報を含むインベントリーファイルの例 (パスワードを除く)

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

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

必要なサーバー情報を含むインベントリーファイルの例 (パスワードを含む)

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

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

[...]
```

カスタムの firewalld 損を使用したインベントリーファイルの例

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

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

Ansible Vault ファイルに保存された admin パスワードおよび Directory Manager パスワードを使用して IdM サーバーを設定する Playbook の例

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

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

インベントリーファイルの admin パスワードおよび Directory Manager パスワードを使用して IdM サーバーを設定する Playbook の例

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

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

関連情報

- man **ipa-server-install(1)**

- **/usr/share/doc/ansible-freeipa/README-server.md**

2.5. 外部 DNS と、ROOT CA としての統合 CA を使用したデプロイメントのパラメーターの設定

以下の手順に従って、外部 DNS ソリューションを使用する環境で、統合 CA の IdM サーバーを root CA としてインストールするようにインベントリーファイルを設定します。

注記

この手順のインベントリーファイルは、**INI**形式を使用します。または、**YAML** 形式または **JSON** 形式を使用できます。

手順

1. **~/MyPlaybooks/** ディレクトリーを作成します。

```
$ mkdir MyPlaybooks
```

2. **~/MyPlaybooks/inventory** ファイルを作成します。
3. 編集するインベントリーファイルを開きます。IdM サーバーとして使用するホストの完全修飾ドメイン名 (**FQDN**) を指定します。**FQDN** が以下の基準を満たしていることを確認してください。
   - 英数字およびハイフン (-) のみが使用できる。たとえば、アンダーラインは使用できないため、DNS の障害が発生する原因となる可能性があります。
   - ホスト名がすべて小文字である。
4. IdM ドメインおよびレルムの情報を指定します。
5. **ipaserver_setup_dns** オプションが **no** に設定されているか、存在しないことを確認します。
6. **admin** と **Directory Manager** のパスワードを指定します。Ansible Vault を使用してパスワードを保存し、Playbook ファイルから Vault ファイルを参照します。あるいは、安全性は低くなりますが、インベントリーファイルにパスワードを直接指定します。
7. (必要に応じて) IdM サーバーで使用する個別の **firewalld** ゾーンを指定します。カスタムゾーンを設定しないと、サービスがデフォルトの **firewalld** ゾーンに追加されます。事前定義されたデフォルトゾーンは **public** です。

重要

指定する **firewalld** ゾーンは存在し、永続的でなければなりません。

必要なサーバー情報を含むインベントリーファイルの例 (パスワードを除く)

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

[ipaserver:vars]
ipaserver_domain=idm.example.com
```

```
ipaserver_realm=IDM.EXAMPLE.COM
ipaserver_setup_dns=no
[...]
```

必要なサーバー情報を含むインベントリーファイルの例 (パスワードを含む)

```
[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

[...]
```

カスタムの firewalld 損を使用したインベントリーファイルの例

```
[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
```

Ansible Vault ファイルに保存された admin パスワードおよび Directory Manager パスワードを使用して IdM サーバーを設定する Playbook の例

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

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

インベントリーファイルの admin パスワードおよび Directory Manager パスワードを使用して IdM サーバーを設定する Playbook の例

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

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

関連情報

- man **ipa-server-install(1)**
- **/usr/share/doc/ansible-freeipa/README-server.md**

2.6. ANSIBLE PLAYBOOK を使用して、統合 CA をルート CA とする IDM サーバーをデプロイする

以下の手順に従って、Ansible Playbook を使用して、統合された認証局 (CA) を備えた IdM サーバーをデプロイします。

前提条件

- マネージドノードが、静的 IP アドレスと動作中のパッケージマネージャーを備えた Red Hat Enterprise Linux 9 システムである。
- 以下のいずれかの手順を選択して、シナリオに対応するパラメーターを設定している。
  - 統合 DNS を使用した手順
  - 外部 DNS を使用した手順

手順

1. Ansible Playbook の実行:

```
$ ansible-playbook -i ~/MyPlaybooks/inventory ~/MyPlaybooks/install-server.yml
```

2. 以下のいずれかのオプションを選択します。
   - IdM デプロイメントで外部 DNS を使用する場合: **/tmp/ipa.system.records.UFRPto.db** ファイルに含まれる DNS リソースレコードを、既存の外部 DNS サーバーに追加します。DNS レコードの更新プロセスは、特定の DNS ソリューションによって異なります。

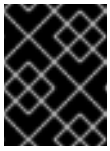
```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```

重要

既存の DNS サーバーに DNS レコードを追加するまで、サーバーのインストールは完了しません。

   - IdM デプロイメントで統合 DNS を使用している場合は、次のコマンドを実行します。

- 親ドメインから IdM DNS ドメインに DNS 委譲を追加します。たとえば、IdM DNS ドメインが **idm.example.com** の場合は、ネームサーバー (NS) レコードを親ドメイン **example.com** に追加します。



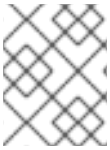
重要

IdM DNS サーバーをインストールするたびに、この手順を繰り返します。

- タイムサーバーの **_ntp._udp** サービス (SRV) レコードを IdM DNS に追加します。IdM DNS に新たにインストールした IdM サーバーのタイムサーバーの SRV レコードが存在すると、今後のレプリカおよびクライアントインストールが、このプライマリー IdM サーバーが使用するタイムサーバーと同期するように自動的に設定されます。

2.7. 統合 DNS と、ルート CA としての外部 CA を使用したデプロイメントのパラメーターの設定

以下の手順に従って、IdM 統合 DNS ソリューションを使用する環境で、外部 CA を持つ IdM サーバーを root CA としてインストールするようにインベントリーファイルを設定します。



注記

この手順のインベントリーファイルは、**INI**形式を使用します。または、**YAML** 形式または **JSON** 形式を使用できます。

手順

1. ~/**MyPlaybooks**/ ディレクトリーを作成します。

```
$ mkdir MyPlaybooks
```

2. ~/**MyPlaybooks**/**inventory** ファイルを作成します。
3. 編集するインベントリーファイルを開きます。IdM サーバーとして使用するホストの完全修飾ドメイン名 (**FQDN**) を指定します。**FQDN** が以下の基準を満たしていることを確認してください。
   - 英数字およびハイフン (-) のみが使用できる。たとえば、アンダーラインは使用できないため、DNS の障害が発生する原因となる可能性があります。
   - ホスト名がすべて小文字である。
4. IdM ドメインおよびレルムの情報を指定します。
5. 以下のオプションを追加して、統合 DNS を使用することを指定します。

```
ipaserver_setup_dns=true
```

6. DNS 転送設定を指定します。以下のいずれかのオプションを選択します。
   - インストールプロセスで **/etc/resolv.conf** ファイルのフォワーダーを使用する場合は、**ipaserver_auto_forwarders=true** オプションを使用します。**/etc/resolv.conf** ファイルで指定する nameserver が localhost 127.0.0.1 アドレスである場合、または仮想プライ

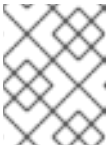
ベートネットワークにあり、使用している DNS サーバーが通常パブリックインターネットから到達できない場合は、このオプションを使用することが推奨されません。

- **ipaserver_forwarders** を使用して、フォワーダーを手動で指定します。インストールプロセスにより、インストールした IdM サーバーの **/etc/named.conf** ファイルに、フォワーダーの IP アドレスが追加されます。
- 代わりにルート DNS サーバーを使用するように設定するには、**ipaserver_no_forwarders=true** オプションを使用します。

注記

DNS フォワーダーがないと、環境は分離され、インフラストラクチャー内の他の DNS ドメインからの名前は解決されません。

7. DNS の逆引きレコードとゾーンの設定を指定します。次のいずれかのオプションを選択します。
   - ゾーンがすでに解決可能である場合でも (逆引き) ゾーンの作成を許可するには、**ipaserver_allow_zone_overlap=true** オプションを使用します。
   - **ipaserver_reverse_zones** オプションを使用して、逆引きゾーンを手動で指定します。
   - インストールプロセスで逆引き DNS ゾーンを作成しない場合は、**ipaserver_no_reverse=true** オプションを使用します。



注記

オプションで、逆引きゾーンの管理に IdM を使用できます。代わりに、この目的で外部 DNS サービスを使用することもできます。

8. **admin** と **Directory Manager** のパスワードを指定します。Ansible Vault を使用してパスワードを保存し、Playbook ファイルから Vault ファイルを参照します。あるいは、安全性は低くなりますが、インベントリーファイルにパスワードを直接指定します。
9. (必要に応じて) IdM サーバーで使用する個別の **firewalld** ゾーンを指定します。カスタムゾーンを設定しないと、サービスがデフォルトの**firewalld** ゾーンに追加されます。事前定義されたデフォルトゾーンは **public** です。

重要

指定する **firewalld** ゾーンは存在し、永続的でなければなりません。

必要なサーバー情報を含むインベントリーファイルの例 (パスワードを除く)

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

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

必要なサーバー情報を含むインベントリーファイルの例 (パスワードを含む)

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

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

[...]
```

カスタムの firewalld 損を使用したインベントリーファイルの例

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

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

[...]
```

10. インストールの最初ステップ用の Playbook を作成します。証明書署名要求 (CSR) を生成し、それをコントローラーからマネージドノードにコピーする指示を入力します。

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

  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: true
```

11. インストールの最終ステップ用に、別の Playbook を作成します。

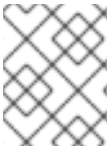
```
---
- name: Playbook to configure IPA server Step 2
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_cert_files:
      - "/root/servercert20240601.pem"
      - "/root/cacert.pem"

  pre_tasks:
  - name: Copy "{{ groups.ipaserver[0] }}-{{ item }}" to "/root/{{ item }}" on node
    ansible.builtin.copy:
      src: "{{ groups.ipaserver[0] }}-{{ item }}"
      dest: "/root/{{ item }}"
      force: true
    with_items:
    - servercert20240601.pem
    - cacert.pem

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

関連情報

- man **ipa-server-install(1)**
- **/usr/share/doc/ansible-freeipa/README-server.md**

2.8. 外部 DNS と、ルート CA としての外部 CA を使用したデプロイメントのパラメーターの設定

以下の手順に従って、外部 DNS ソリューションを使用する環境で、外部 CA を持つ IdM サーバーを root CA としてインストールするようにインベントリーファイルを設定します。



注記

この手順のインベントリーファイルは、**INI**形式を使用します。または、**YAML** 形式または **JSON** 形式を使用できます。

手順

1. **~/MyPlaybooks/** ディレクトリーを作成します。

```
$ mkdir MyPlaybooks
```

2. **~/MyPlaybooks/inventory** ファイルを作成します。

3. 編集するインベントリーファイルを開きます。IdM サーバーとして使用するホストの完全修飾ドメイン名 (**FQDN**) を指定します。**FQDN** が以下の基準を満たしていることを確認してください。

   - 英数字およびハイフン (-) のみが使用できる。たとえば、アンダーラインは使用できないため、DNS の障害が発生する原因となる可能性があります。
   - ホスト名がすべて小文字である。

4. IdM ドメインおよびレルムの情報を指定します。
5. **ipaserver_setup_dns** オプションが **no** に設定されているか、存在しないことを確認します。
6. **admin** と **Directory Manager** のパスワードを指定します。Ansible Vault を使用してパスワードを保存し、Playbook ファイルから Vault ファイルを参照します。あるいは、安全性は低くなりますが、インベントリーファイルにパスワードを直接指定します。
7. (必要に応じて) IdM サーバーで使用する個別の **firewalld** ゾーンを指定します。カスタムゾーンを設定しないと、サービスがデフォルトの **firewalld** ゾーンに追加されます。事前定義されたデフォルトゾーンは **public** です。

重要

指定する **firewalld** ゾーンは存在し、永続的でなければなりません。

必要なサーバー情報を含むインベントリーファイルの例 (パスワードを除く)

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

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

必要なサーバー情報を含むインベントリーファイルの例 (パスワードを含む)

```
[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

[...]
```

カスタムの firewalld 損を使用したインベントリーファイルの例

```
[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

[...]
```

8. インストールの最初ステップ用の Playbook を作成します。証明書署名要求 (CSR) を生成し、それをコントローラーからマネージドノードにコピーする指示を入力します。

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

  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: true
```

9. インストールの最終ステップ用に、別の Playbook を作成します。

```
---
- name: Playbook to configure IPA server Step 2
  hosts: ipaserver
  become: true
  vars_files:
  - playbook_sensitive_data.yml
  vars:
    ipaserver_external_cert_files:
      - "/root/servercert20240601.pem"
      - "/root/cacert.pem"

  pre_tasks:
  - name: Copy "{{ groups.ipaserver[0] }}-{{ item }}" to "/root/{{ item }}" on node
    ansible.builtin.copy:
      src: "{{ groups.ipaserver[0] }}-{{ item }}"
      dest: "/root/{{ item }}"
      force: true
    with_items:
    - servercert20240601.pem
```

```
    - cacert.pem

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

関連情報

- IdM サーバーのインストール: 統合 DNS なしで外部 CA を root CA として使用する場合
- man **ipa-server-install(1)**
- **/usr/share/doc/ansible-freeipa/README-server.md**

2.9. 外部 CA を ROOT CA として備えた IDM サーバーの ANSIBLE PLAYBOOK を使用したデプロイメント

以下の手順に従って、Ansible Playbook を使用して、外部認証局 (CA) を備えた IdM サーバーをデプロイします。

前提条件

- マネージドノードが、静的 IP アドレスと動作中のパッケージマネージャーを備えた Red Hat Enterprise Linux 9 システムである。
- 以下のいずれかの手順を選択して、シナリオに対応するパラメーターを設定している。
  - 統合 DNS を使用した手順
  - 外部 DNS を使用した手順

手順

1. インストールの最初のステップの指示を含む Ansible Playbook を実行します (例: **install-server-step1.yml**)。

```
$ ansible-playbook --vault-password-file=password_file -v -i ~/MyPlaybooks/inventory
~/MyPlaybooks/install-server-step1.yml
```

2. コントローラー上の **ipa.csr** 証明書署名要求ファイルを見つけ、これを外部 CA に送信します。
3. 外部 CA が署名した IdM CA 証明書をコントローラーファイルシステムに配置して、次のステップの Playbook で見つけられるようにします。
4. インストールの最後のステップの指示を含む Ansible Playbook を実行します (例: **install-server-step2.yml**)。

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory ~/MyPlaybooks/install-server-
step2.yml
```

5. 以下のいずれかのオプションを選択します。

- IdM デプロイメントで外部 DNS を使用する場合: **/tmp/ipa.system.records.UFRPto.db** ファイルに含まれる DNS リソースレコードを、既存の外部 DNS サーバーに追加します。DNS レコードの更新プロセスは、特定の DNS ソリューションによって異なります。

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

重要

既存の DNS サーバーに DNS レコードを追加するまで、サーバーのインストールは完了しません。

- IdM デプロイメントで統合 DNS を使用している場合は、次のコマンドを実行します。
  - 親ドメインから IdM DNS ドメインに DNS 委譲を追加します。たとえば、IdM DNS ドメインが **idm.example.com** の場合は、ネームサーバー (NS) レコードを親ドメイン **example.com** に追加します。

重要

IdM DNS サーバーをインストールするたびに、この手順を繰り返します。

  - タイムサーバーの **_ntp._udp** サービス (SRV) レコードを IdM DNS に追加します。IdM DNS に新たにインストールした IdM サーバーのタイムサーバーの SRV レコードが存在すると、今後のレプリカおよびクライアントインストールが、このプライマリー IdM サーバーが使用するタイムサーバーと同期するように自動的に設定されます。

2.10. ANSIBLE PLAYBOOK を使用した IDM サーバーのアンインストール

注記

既存の Identity Management (IdM) デプロイメントでは、**レプリカ** と **サーバー** は置き替え可能な用語です。

以下の手順に従って、Ansible Playbook を使用して IdM レプリカをアンインストールします。この例では、以下が適用されます。

- IdM 設定は、**server123.idm.example.com** からアンインストールされます。
- **server123.idm.example.com** と関連するホストエントリーが IdM トポロジーから削除されます。

前提条件

- コントロールノード:
  - Ansible バージョン 2.14 以降を使用している。

- **ansible-freeipa** パッケージがインストールされている。
- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
- **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。
- **ipaserver_remove_from_topology** オプションを機能させるには、システムが RHEL 9.3 以降で実行されている必要があります。

- マネージドノード:
  - システムが RHEL 9 で実行されている。

手順

1. Ansible Playbook ファイル **uninstall-server.yml** を次の内容で作成します。

```
---
- name: Playbook to uninstall an IdM replica
  hosts: ipaserver
  become: true

  roles:
  - role: ipaserver
    ipaserver_remove_from_domain: true
    state: absent
```

ipaserver_remove_from_domain オプションは、IdM トポロジーからホストを登録解除します。

注記

server123.idm.example.com を削除するとトポロジーが切断される場合は、削除は中止されます。詳細は、Ansible Playbook を使用した IdM サーバーのアンインストール (トポロジーが切断された場合でも) を参照してください。

2. レプリカをアンインストールします。

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

3. **server123.idm.example.com** を指しているネームサーバー (NS) DNS レコードがすべて DNS ゾーンから削除されていることを確認してください。使用する DNS が IdM により管理される統合 DNS であるか、外部 DNS であるかに関わらず、確認を行なってください。IdM から DNS レコードを削除する方法は、 Deleting DNS records in the IdM CLI を参照してください。

2.11. ANSIBLE PLAYBOOK を使用した IDM サーバーのアンインストール (トポロジーが切断された場合でも)

注記

既存の Identity Management (IdM) デプロイメントでは、**レプリカ** と **サーバー** は置き替え可能な用語です。

IdM トポロジーが切断されたとしても、Ansible Playbook を使用して IdM レプリカをアンインストールするには、以下の手順を実行します。この例では、**server456.idm.example.com** を使用して、レプリカと、トポロジーから **server123.idm.example.com** の FQDN を持つ関連付けられたホストエントリーを削除します。これにより、特定のレプリカが **server456.idm.example.com** および残りのトポロジーから切断されます。

注記

remove_server_from_domain のみを使用してトポロジーからレプリカを削除しても、トポロジーは切断されないため、他のオプションは必要ありません。トポロジーが切断される結果となった場合は、ドメインの保持したい部分を指定する必要があります。その場合、以下を実行する必要があります。

- **ipaserver_remove_on_server** 値を指定します。
- **ipaserver_ignore_topology_disconnect** を True に設定します。

前提条件

- コントロールノード:
  - Ansible バージョン 2.14 以降を使用している。
  - システムが RHEL 9.3 以降で実行されている。
  - **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
  - **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。
- マネージドノード:
  - システムが 9 以降で実行されている。

手順

1. Ansible Playbook ファイル **uninstall-server.yml** を次の内容で作成します。

```
---
- name: Playbook to uninstall an IdM replica
  hosts: ipaserver
  become: true

  roles:
  - role: ipaserver
    ipaserver_remove_from_domain: true
    ipaserver_remove_on_server: server456.idm.example.com
    ipaserver_ignore_topology_disconnect: true
    state: absent
```

注記

通常の状況では、server123 を削除してもトポロジーが切断されない場合で、**ipaserver_remove_on_server** の値が設定されていない場合は、server123 が削除されたレプリカは server123 のレプリカ合意を使用して自動的に決定されます。

2. レプリカをアンインストールします。

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

3. **server123.idm.example.com** を指しているネームサーバー (NS) DNS レコードがすべて DNS ゾーンから削除されていることを確認してください。使用する DNS が IdM により管理される統合 DNS であるか、外部 DNS であるかに関わらず、確認を行なってください。IdM から DNS レコードを削除する方法は、 Deleting DNS records in the IdM CLI を参照してください。

関連情報

- インベントリーの基本: 形式、ホスト、およびグループ
- IdM サーバーをインストールするためのサンプルの Ansible Playbook と、**ansible-freeipa** アップストリームのドキュメント で使用できる変数のリストを表示できます。

第3章 ANSIBLE PLAYBOOK で IDENTITY MANAGEMENT レプリカのインストール

Ansible を使用してシステムを IdM レプリカとして設定すると、IdM ドメインに登録され、ドメインの IdM サーバーにある IdM サービスをシステムが使用できるようになります。

デプロイメントは、Ansible ロール **ipareplica** で管理されます。このロールは、自動検出モードを使用して、IdM サーバー、ドメイン、およびその他の設定を識別できます。ただし、階層のような形で複数のレプリカをデプロイし、レプリカの各グループを異なるタイミングでデプロイする場合は、各グループに特定のサーバーまたはレプリカを定義する必要があります。

前提条件

- Ansible コントロールノードに ansible-freeipa パッケージがインストールされている。
- Ansible と IdM の一般的な概念を理解している。
- デプロイメント内のレプリカトポロジーを計画 した。

3.1. IDM レプリカをインストールするためのベース変数、サーバー変数、およびクライアント変数の指定

IdM レプリカをインストールするためのインベントリーファイルを設定するには、以下の手順を完了します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。

手順

1. 編集するインベントリーファイルを開きます。IdM レプリカとなるホストの完全修飾ドメイン名 (FQDN) を指定します。FQDN は有効な DNS 名である必要があります。
   - 数字、アルファベット、およびハイフン (-) のみを使用できる。たとえば、アンダーラインは使用できないため、DNS の障害が発生する原因となる可能性があります。
   - ホスト名がすべて小文字である。

   レプリカの FQDN のみが定義されている単純なインベントリーホストファイルの例

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

   IdM サーバーがデプロイされており、SRV レコードが IdM DNS ゾーンに適切に設定されている場合、スクリプトはその他に必要な値をすべて自動的に検出します。

2. [オプション] トポロジーの設計方法に基づいて、インベントリーファイルに追加情報を入力します。

シナリオ 1

自動検出を回避し、**[ipareplicas]** セクションに記載されているすべてのレプリカが特定の IdM サーバーを使用するようにするには、インベントリーファイルの **[ipaservers]** セクションにそのサーバーを設定します。

IdM サーバーとレプリカの FQDN が定義されているインベントリーホストファイルの例

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

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

シナリオ 2

または、自動検出を回避して、特定のサーバーで特定のレプリカをデプロイする場合は、インベントリーファイルの **[ipareplicas]** セクションに、特定のレプリカのサーバーを個別に設定します。

特定のレプリカ用に特定の IdM サーバーが定義されたインベントリーファイルの例

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

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

上記の例では、**replica3.idm.example.com** が、すでにデプロイされた **replica1.idm.example.com** を複製元として使用します。

シナリオ 3

1 つのバッチに複数のレプリカをデプロイする場合は、多層レプリカのデプロイメントが役に立ちます。インベントリーファイルにレプリカの特定グループ (例: **[ipareplicas_tier1]** および **[ipareplicas_tier2]**) を定義し、Playbook **install-replica.yml** で各グループに個別のプレイを設計します。

レプリカ階層が定義されているインベントリーファイルの例

```
[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
```

ipareplica_servers の最初のエントリーが使用されます。次のエントリーは、フォールバックオプションとして使用されます。IdM レプリカのデプロイに複数の層を使用する場合は、最初に tier1 からレプリカをデプロイし、次に tier2 からレプリカをデプロイするように、Playbook に個別のタスクが必要です。

レプリカグループごとに異なるプレイを定義した Playbook ファイルの例

```
---
- 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. [オプション] **firewalld** と DNS に関する追加情報を入力します。

シナリオ 1

指定の **firewalld** ゾーン (内部ゾーンなど) をレプリカで使用する場合は、インベントリーファイルでゾーンを指定できます。カスタムゾーンを設定しないと、サービスがデフォルトの **firewalld** ゾーンに追加されます。事前定義されたデフォルトゾーンは **public** です。

重要

指定する **firewalld** ゾーンは存在し、永続的でなければなりません。

カスタム firewalld 帯を持つシンプルなインベントリーホストファイルの例

```
[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
```

シナリオ 2

レプリカで IdM DNS サービスをホストする場合は、**[ipareplicas:vars]** セクションに **ipareplica_setup_dns=true** 行を追加します。また、サーバーごとの DNS フォワーダーを使用するかどうかを指定します。

- サーバーごとのフォワーダーを設定するには、**ipareplica_forwarders** 変数と文字列のリストを **[ipareplicas:vars]** セクションに追加します (例: **ipareplica_forwarders=192.0.2.1,192.0.2.2**)。
- サーバーごとのフォワーダーを設定しない場合は、**[ipareplicas:vars]** セクションに **ipareplica_no_forwarders=true** 行を追加します。
- レプリカの **/etc/resolv.conf** ファイルにリスト表示されているフォワーダーに基づいてサーバーごとにフォワーダーを設定するには、**[ipareplicas:vars]** セクションに **ipareplica_auto_forwarders** を追加します。

レプリカに DNS とサーバーごとのフォワーダーを設定する手順を含むインベントリーファイルの例

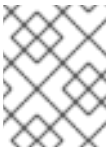
```
[ipaservers]
server.idm.example.com

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

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

シナリオ 3

ipaclient_configure_dns_resolve および **ipaclient_dns_servers** オプション (使用可能な場合) を使用して DNS リゾルバーを指定し、クラスターのデプロイメントを簡素化します。これは、IdM デプロイメントが統合 DNS を使用している場合に特に便利です。

DNS リゾルバーを指定するインベントリーファイルスニペット:

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



注記

ipaclient_dns_servers リストには IP アドレスのみを含める必要があります。ホスト名を含めることはできません。

関連情報

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

3.2. ANSIBLE PLAYBOOK を使用して IDM レプリカをインストールするための認証情報の指定

この手順は、IdM レプリカのインストールに認可を設定します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。

手順

1. **レプリカをデプロイする権限のあるユーザーのパスワード** (IdM の **admin** など) を指定します。
   - Red Hat は、Ansible Vault を使用してパスワードを保存し、Playbook ファイルから Vault ファイルを参照する (**install-replica.yml** など) ことを推奨します。

     Ansible Vault ファイルのインベントリーファイルおよびパスワードのプリンシパルを使用した Playbook ファイルの例

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

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

     Ansible Vault の使用方法は、公式の Ansible Vault ドキュメントを参照してください。
   - あまり安全ではありませんが、インベントリーファイルで **admin** の認証情報を直接提供します。インベントリーファイルの **[ipareplicas:vars]** セクションで **ipaadmin_password** オプションを使用します。インベントリーファイルと、Playbook ファイル **install-replica.yml** は以下のようになります。

     インベントリーの hosts.replica ファイルの例

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

     インベントリーファイルのプリンシパルおよびパスワードを使用した Playbook の例

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

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

- または、安全性は低くなりますが、レプリカをインベントリーファイルに直接デプロイすることを許可されている別のユーザーの認証情報を提供します。別の認証ユーザーを指定するには、ユーザー名に **ipaadmin_principal** オプションを使用し、パスワードに **ipaadmin_password** オプションを使用します。インベントリーファイルと、Playbook ファイル **install-replica.yml** は以下のようになります。

  インベントリーの hosts.replica ファイルの例

```
[...]
[ipareplicas:vars]
ipaadmin_principal=my_admin
ipaadmin_password=my_admin_secret123
```

  インベントリーファイルのプリンシパルおよびパスワードを使用した Playbook の例

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

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

関連情報

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

3.3. ANSIBLE PLAYBOOK で IDM レプリカのデプロイメント

以下の手順に従って、Ansible Playbook を使用して IdM レプリカをデプロイします。

前提条件

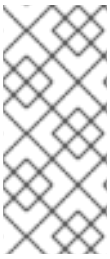
- マネージドノードが、静的 IP アドレスと動作中のパッケージマネージャーを備えた Red Hat Enterprise Linux 9 システムである。
- IdM レプリカをインストールするためのインベントリーファイル を設定しました。
- IdM レプリカをインストールするための認証 を設定しました。

手順

- Ansible Playbook の実行:

```
$ ansible-playbook -i ~/MyPlaybooks/inventory ~/MyPlaybooks/install-replica.yml
```

3.4. ANSIBLE PLAYBOOK を使用した IDM レプリカのアンインストール



注記

既存の Identity Management (IdM) デプロイメントでは、**レプリカ** と **サーバー** は置き替え可能な用語です。IdM サーバーをアンインストールする方法の詳細は、Ansible Playbook を使用した IdM サーバーのアンインストール または トポロジーが切断される場合でも Ansible Playbook を使用して IdM サーバーをアンインストールする を参照してください。

関連情報

- IdM のサーバーおよびクライアントの概要

第4章 ANSIBLE PLAYBOOK で IDENTITY MANAGEMENT クライアントのインストール

Ansible を使用して、システムを Identity Management (IdM) クライアントとして設定する方法を説明します。システムを IdM クライアントとして設定すると、IdM ドメインに登録され、システムがドメインの IdM サーバーで IdM サービスを使用できるようになります。

デプロイメントは、Ansible ロール **ipaclient** により管理されます。デフォルトでは、ロールは自動検出モードを使用して、IdM サーバー、ドメイン、およびその他の設定を特定します。ロールは、Ansible Playbook がインベントリーファイルなどに指定した設定を使用するように変更できます。

前提条件

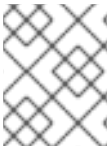
- Ansible コントロールノードに ansible-freeipa パッケージがインストールされている。
- Ansible バージョン 2.14 以降を使用している。
- Ansible と IdM の一般的な概念を理解している。

4.1. 自動検出クライアントインストールモードでインベントリーファイルのパラメーターの設定

Ansible Playbook を使用して Identity Management (IdM) クライアントをインストールするには、インベントリーファイル (例: **inventory**) でターゲットホストのパラメーターを設定します。

- ホストに関する情報
- タスクの承認

インベントリーファイルは、所有するインベントリープラグインに応じて、多数ある形式のいずれかになります。**INI-like** 形式は Ansible のデフォルトで、以下の例で使用されています。



注記

RHEL でグラフィカルユーザーインターフェイスでスマートカードを使用するには、Ansible Playbook に **ipaclient_mkhomedir** 変数を含めるようにします。

手順

1. **inventory** ファイルを開いて編集します。
2. IdM クライアントになるホストの完全修飾ホスト名 (FQDN) を指定します。完全修飾ドメイン名は、有効な DNS 名である必要があります。
   - 数字、アルファベット、およびハイフン (-) のみを使用できる。たとえば、アンダーラインは使用できないため、DNS の障害が発生する原因となる可能性があります。
   - ホスト名がすべて小文字である。大文字は使用できません。

   SRV レコードが IdM DNS ゾーンで正しく設定されている場合は、スクリプトが自動的に必要な値をすべて検出します。

   クライアントの FQDN のみが定義されている単純なインベントリーホストファイルの例

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

3. クライアントを登録するための認証情報を指定します。以下の認証方法を使用できます。

- **クライアントを登録する権限のあるユーザーのパスワード**。以下はデフォルトのオプションになります。
  - Red Hat は、Ansible Vault を使用してパスワードを保存し、Playbook ファイル (**install-client.yml** など) から Vault ファイルを直接参照することを推奨します。

    Ansible Vault ファイルのインベントリーファイルおよびパスワードのプリンシパルを使用した Playbook ファイルの例

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

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

  - あまり安全ではありませんが、**inventory/hosts** ファイルの **[ipaclients:vars]** セクションに **ipaadmin_password** オプションを使用して、**admin** の認証情報を提供します。また、別の認証ユーザーを指定するには、ユーザー名に **ipaadmin_principal** オプション、パスワードに **ipaadmin_password** オプションを使用します。**inventory/hosts** インベントリーファイルと、Playbook ファイル **install-client.yml** は以下のようになります。

    インベントリーホストファイルの例

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

    インベントリーファイルのプリンシパルおよびパスワードを使用した Playbook の例

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

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

- 以前登録した **クライアントキータブ** が利用できる場合は、以下を行います。
  このオプションは、システムが Identity Management クライアントとして登録されたことがある場合に使用できます。この認証方法を使用するには、**#ipaclient_keytab** オプションのコメントを解除して、キータブを保存するファイルへのパスを指定します (例:

inventory/hosts の **[ipaclient:vars]** セクション)。

- 登録時に生成される **ランダムなワンタイムパスワード** (OTP)。この認証方法を使用するには、インベントリーファイルで **ipaclient_use_otp=true** オプションを使用します。たとえば、**inventory/hosts** ファイルの **[ipaclients:vars]** セクションにある **ipaclient_use_otp=true** オプションのコメントを解除できます。OTP では、以下のいずれかのオプションも指定する必要があります。
  - **クライアントを登録する権限のあるユーザーのパスワード** (例: **inventory/hosts** ファイルの **[ipaclients:vars]** セクションに **ipaadmin_password** の値を指定)。
  - **管理者キータブ** (例: **inventory/hosts** の **[ipaclients:vars]** セクションに **ipaadmin_keytab** の値を指定)。

4. (オプション) **ipaclient_configure_dns_resolve** および **ipaclient_dns_servers** オプション (使用可能な場合) を使用して DNS リゾルバーを指定し、クラスターのデプロイメントを簡素化します。これは、IdM デプロイメントが統合 DNS を使用している場合に特に便利です。

   DNS リゾルバーを指定するインベントリーファイルスニペット:

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

   注記

   ipaclient_dns_servers リストには IP アドレスのみを含める必要があります。ホスト名を含めることはできません。

5. RHEL 9.3 以降では、**ipaclient_subid: true** オプションを指定して、IdM ユーザーのサブ ID 範囲を IdM レベルで設定することもできます。

関連情報

- **/usr/share/ansible/roles/ipaclient/README.md**
- subID 範囲の手動管理

4.2. クライアントのインストール時に自動検出ができない場合に備えてインベントリーファイルのパラメーターの設定

Ansible Playbook を使用して Identity Management クライアントをインストールするには、インベントリーファイルでターゲットホストパラメーターを設定します (例: **inventory/hosts**)。

- ホストと、IdM サーバーおよび IdM ドメインまたは IdM レルムに関する情報
- タスクの承認

インベントリーファイルは、所有するインベントリープラグインに応じて、多数ある形式のいずれかになります。**INI-like** 形式は Ansible のデフォルトで、以下の例で使用されています。

注記

RHEL でグラフィカルユーザーインターフェイスでスマートカードを使用するには、Ansible Playbook に **ipaclient_mkhomedir** 変数を含めるようにします。

手順

1. IdM クライアントになるホストの完全修飾ホスト名 (FQDN) を指定します。完全修飾ドメイン名は、有効な DNS 名である必要があります。
   - 数字、アルファベット、およびハイフン (**-**) のみを使用できる。たとえば、アンダーラインは使用できないため、DNS の障害が発生する原因となる可能性があります。
   - ホスト名がすべて小文字である。大文字は使用できません。
2. **inventory/hosts** ファイルの関連セクションに、他のオプションを指定します。
   - **[ipaservers]** セクションのサーバーの FQDN は、クライアントが登録される IdM サーバーを示します。
   - 以下のいずれかのオプションを使用できます。
     - クライアントが登録される IdM サーバーの DNS ドメイン名を指定する **[ipaclients:vars]** セクションの **ipaclient_domain** オプション
     - IdM サーバーが制御する Kerberos レルムの名前を示す **[ipaclients:vars]** セクションの **ipaclient_realm** オプション

       クライアント FQDN、サーバーの FQDN、およびドメインが定義されているインベントリーホストファイルの例

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

       [ipaservers]
       server.idm.example.com

       [ipaclients:vars]
       ipaclient_domain=idm.example.com
       [...]
       ```
3. クライアントを登録するための認証情報を指定します。以下の認証方法を使用できます。
   - **クライアントを登録する権限のあるユーザーのパスワード**。以下はデフォルトのオプションになります。
     - Red Hat は、Ansible Vault を使用してパスワードを保存し、Playbook ファイル (**install-client.yml** など) から Vault ファイルを直接参照することを推奨します。

       Ansible Vault ファイルのインベントリーファイルおよびパスワードのプリンシパルを使用した Playbook ファイルの例

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

```
- playbook_sensitive_data.yml

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

- 安全性は低くなりますが、**inventory/hosts** ファイルの **[ipaclients:vars]** セクションの **ipaadmin_password** オプションを使用して、**admin** の認証情報が提供されます。また、別の認証ユーザーを指定するには、ユーザー名に **ipaadmin_principal** オプション、パスワードに **ipaadmin_password** オプションを使用します。これにより、Playbook ファイル **install-client.yml** は、以下のようになります。

  インベントリーホストファイルの例

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

  インベントリーファイルのプリンシパルおよびパスワードを使用した Playbook の例

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

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

- 以前登録した **クライアントキータブ** が利用できる場合は、以下を行います。
  このオプションは、システムが Identity Management クライアントとして登録されたことがある場合に使用できます。この認証方法を使用するには、**ipaclient_keytab** オプションをコメント解除します。たとえば、**inventory/hosts** の **[ipaclient:vars]** セクションにあるように、キータブを格納しているファイルへのパスを指定します。

- 登録時に生成される **ランダムなワンタイムパスワード** (OTP)。この認証方法を使用するには、インベントリーファイルで **ipaclient_use_otp=true** オプションを使用します。たとえば、**inventory/hosts** ファイルの **[ipaclients:vars]** セクションにある **#ipaclient_use_otp=true** オプションのコメントを解除できます。OTP では、以下のいずれかのオプションも指定する必要があります。

  - **クライアントを登録する権限のあるユーザーのパスワード** (例: **inventory/hosts** ファイルの **[ipaclients:vars]** セクションに **ipaadmin_password** の値を指定)。

  - **管理者キータブ** (例: **inventory/hosts** の **[ipaclients:vars]** セクションに **ipaadmin_keytab** の値を指定)。

4. RHEL 9.3 以降では、**ipaclient_subid: true** オプションを指定して、IdM ユーザーのサブ ID 範囲を IdM レベルで設定することもできます。

関連情報

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

- subID 範囲の手動管理

4.3. ANSIBLE PLAYBOOK で IDM クライアント登録の認可オプション

次のいずれかの方法を使用して、IdM クライアントの登録を許可できます。

- クライアントを登録することを許可されたユーザーのパスワード: Ansible vault に保存されているパスワード
- クライアントを登録することを許可されたユーザーのパスワード: インベントリーファイルに保存されているパスワード
- ランダムなワンタイムパスワード (OTP) + 管理者パスワード
- ランダムなワンタイムパスワード (OTP) + 管理者キータブ
- 前回登録時のクライアントキータブ

これらの方法のサンプルインベントリーファイルと **install-client.yml** Playbook ファイルを以下に示します。

表4.1 クライアントを登録することを許可されたユーザーのパスワード: Ansible vault に保存されているパスワード

インベントリーファイルの例	Playbook ファイル **install-client.yml** の例

[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

表4.2 クライアントを登録することを許可されたユーザーのパスワード: インベントリーファイルに保存されているパスワード

インベントリーファイルの例	Playbook ファイル **install-client.yml** の例

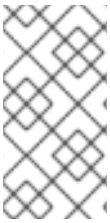
[ipaclients:vars]
ipaadmin_password=Secret123
	- name: Playbook to configure IPA clients
	  hosts: ipaclients
	  become: true
	
	  roles:
	  - role: ipaclient
	  state: true

表4.3 ランダムなワンタイムパスワード (OTP) + 管理者パスワード

インベントリーファイルの例	Playbook ファイル **install-client.yml** の例
[ipaclients:vars] ipaadmin_password=Secret123 ipaclient_use_otp=true Playbook の実行中に OTP を生成する場合 **または** [ipaclients:vars] ipaclient_otp=<W5YpARl=7M.> インストール前に IdM **admin** によって OTP がすでに生成されている場合	- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true

表4.4 ランダムなワンタイムパスワード (OTP) + 管理者キータブ

インベントリーファイルの例	Playbook ファイル **install-client.yml** の例
[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



注記

RHEL 9.2 以降、上記の 2 つの OTP 承認シナリオでは、**kinit** コマンドを使用した管理者の TGT の要求は、最初に指定または検出された IdM サーバーで行われます。したがって、Ansible コントロールノードを追加変更する必要はありません。RHEL 9.2 より前は、制御ノードに **krb5-workstation** パッケージが必要でした。

表4.5 前回登録時のクライアントキータブ

インベントリーファイルの例	Playbook ファイル **install-client.yml** の例

インベントリーファイルの例	Playbook ファイル **install-client.yml** の例
[ipaclients:vars] ipaclient_keytab=/root/krb5.keytab	- name: Playbook to configure IPA clients hosts: ipaclients become: true roles: - role: ipaclient state: true

4.4. ANSIBLE PLAYBOOK を使用した IDM クライアントのデプロイ

Ansible Playbook を使用して IdM 環境に IdM クライアントをデプロイするには、この手順を完了します。

前提条件

- マネージドノードが、静的 IP アドレスと動作中のパッケージマネージャーを備えた Red Hat Enterprise Linux 9 システムである。
- IdM クライアントのデプロイメントのパラメーターを、デプロイメントシナリオに対応するように設定している。
  - 自動検出クライアントインストールモードでインベントリーファイルのパラメーターの設定
  - クライアントのインストール時に自動検出ができない場合に備えてインベントリーファイルのパラメーターの設定

手順

- Ansible Playbook の実行:

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory ~/MyPlaybooks/install-client.yml
```

4.5. ANSIBLE のワンタイムパスワード方式を使用して IDM クライアントをインストールする

Identity Management (IdM) で新しいホストのワンタイムパスワード (OTP) を生成し、それを使用してシステムを IdM ドメインに登録できます。この手順では、別の IdM ホストで IdM クライアントの OTP を生成した後、Ansible を使用して IdM クライアントをインストールする方法について説明します。

この IdM クライアントのインストール方法は、異なる権限を持つ次の 2 人のシステム管理者が組織内に存在する場合に便利です。

- IdM 管理者の認証情報を持つ管理者
- 必要な Ansible 認証情報 (IdM クライアントになるホストへの **root** アクセス権を含む) を持つ別の管理者

IdM 管理者は、手順の前半部分を実行し、OTP パスワードを生成します。Ansible 管理者は、手順の残りの部分を実行し、OTP を使用して IdM クライアントをインストールします。

前提条件

- IdM **admin** 認証情報、または少なくとも **Host Enrollment** 権限と、IdM に DNS レコードを追加する権限を持っている。
- IdM クライアントをインストールできるように、Ansible マネージドノードでユーザーエスカレーション方法を設定した。
- Ansible コントロールノードが RHEL 8.7 以前で実行されている場合、Ansible コントロールノードにパッケージをインストールできる。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイルを作成した。
- マネージドノードが、静的 IP アドレスと動作中のパッケージマネージャーを備えた Red Hat Enterprise Linux 9 システムである。

手順

1. **Host Enrollment** 権限と DNS レコードを追加する権限を持つロールを持つ IdM ユーザーとして IdM ホストに **SSH** 接続します。

```
$ ssh admin@server.idm.example.com
```

2. 新しいクライアントの OTP を生成します。

```
[admin@server ~]$ ipa host-add client.idm.example.com --ip-address=172.25.250.11 --random
----------------------------------------------------
Added host "client.idm.example.com"
----------------------------------------------------
  Host name: client.idm.example.com
  Random password: W5YpARl=7M.n
  Password: True
  Keytab: False
  Managed by: server.idm.example.com
```

--ip-address=<your_host_ip_address> オプションは、指定した IP アドレスを持つホストを IdM DNS に追加します。

3. IdM ホストを終了します。

```
$ exit
logout
Connection to server.idm.example.com closed.
```

4. Ansible コントローラーで、ランダムパスワードを含めるようにインベントリーファイルを更新します。

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

[ipaclients:vars]
ipaclient_domain=idm.example.com
ipaclient_otp=W5YpARl=7M.n
[...]
```

5. Ansible コントローラーが RHEL 9.1 以前を実行している場合は、**krb5-workstation** パッケージによって提供される **kinit** ユーティリティーをインストールします。

```
$ sudo dnf install krb5-workstation
```

6. Playbook を実行してクライアントをインストールします。

```
$ ansible-playbook -i inventory install-client.yml
```

4.6. ANSIBLE インストール後の IDENTITY MANAGEMENT クライアントのテスト

コマンドラインインターフェイス (CLI) により、**ansible-playbook** コマンドが成功したことが表示されますが、独自のテストを行うこともできます。

Identity Management クライアントが、サーバーに定義したユーザーに関する情報を取得できることをテストするには、サーバーに定義したユーザーを解決できることを確認します。たとえば、デフォルトの **admin** ユーザーを確認するには、次のコマンドを実行します。

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

認証が適切に機能していることをテストするには、別の既存 IdM ユーザーで **su -** を実行します。

```
[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. ANSIBLE PLAYBOOK での IDM クライアントのアンインストール

以下の手順に従って、Ansible Playbook を使用して IdM クライアントと機能していたホストをアンインストールします。

前提条件

- IdM 管理者の認証情報
- マネージドノードが、静的 IP アドレスを持つ Red Hat Enterprise Linux 9 システムである。

手順

- クライアントをアンインストールする指示を含む Ansible Playbook を実行します (例: **uninstall-client.yml**)。

```
$ ansible-playbook -v -i ~/MyPlaybooks/inventory ~/MyPlaybooks/uninstall-client.yml
```

重要

クライアントをアンインストールすると、基本的な IdM 設定のみがホストから削除されますが、クライアントの再インストールを行うことになった場合に備え、ホストに設定ファイルが残されます。また、アンインストールには以下の制限があります。

- IdM LDAP サーバーからクライアントホストエントリーは削除されない。アンインストールすると、ホストの登録が解除されるだけである。
- クライアントにあるサービスは、IdM から削除されない。
- クライアントの DNS エントリーは、IdM サーバーから削除されない。
- **/etc/krb5.keytab** を除き、以前の Keytab のプリンシパルは削除されない。

アンインストールを行うと、IdM CA がホスト向けに発行した証明書がすべて削除されることに注意してください。

関連情報

- IdM クライアントのアンインストール

第5章 ANSIBLE PLAYBOOK を使用して IDM を管理する環境の準備

Identity Management (IdM) を管理するシステム管理者は、Red Hat Ansible Engine を使用する際に以下を行うことが推奨されます。

- ホームディレクトリーに Ansible Playbook 専用のサブディレクトリー (例: **~/MyPlaybooks**) を作成します。
- **/usr/share/doc/ansible-freeipa/*** と **/usr/share/doc/rhel-system-roles/*** ディレクトリーおよびサブディレクトリーから **~/MyPlaybooks** ディレクトリーにサンプル Ansible Playbook をコピーして調整します。
- **~/MyPlaybooks** ディレクトリーにインベントリーファイルを追加します。

このプラクティスを使用すると、すべての Playbook を 1 か所で見つけることができます。

注記

マネージドノードで **root** 権限を呼び出さずに **ansible-freeipa** Playbook を実行できます。例外には、**ipaserver**、**ipareplica**、**ipaclient**、**ipasmartcard_server**、**ipasmartcard_client**、および **ipabackup ansible-freeipa** ロールを使用する Playbook が含まれます。これらのロールには、ディレクトリーおよび **dnf** ソフトウェアパッケージマネージャーへの特権アクセスが必要です。

Red Hat Enterprise Linux IdM ドキュメントの Playbook は、次の セキュリティー設定 を前提としています。

- IdM **admin** は、管理ノードのリモート Ansible ユーザーです。
- Ansible vault に暗号化された IdM **admin** パスワードを保存します。
- Ansible vault を保護するパスワードをパスワードファイルに配置しました。
- ローカルの ansible ユーザーを除く全員に対して、vault パスワードファイルへのアクセスをブロックします。
- vault パスワードファイルを定期的に削除して再作成します。

別のセキュリティー設定 も検討してください。

5.1. ANSIBLE PLAYBOOK を使用して IDM を管理するためのコントロールノードと管理ノードの準備

~/MyPlaybooks ディレクトリーを作成し、それを使用して Ansible Playbook を保存および実行できるように設定するには、次の手順に従います。

前提条件

- マネージドノードに IdM サーバー (**server.idm.example.com** および **replica.idm.example.com**) をインストールした。

- DNS およびネットワークを設定し、コントロールノードから直接マネージドノード (**server.idm.example.com** および **replica.idm.example.com**) にログインすることができる。
- IdM **admin** のパスワードを把握している。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

```
$ cd ~/MyPlaybooks
```

2. **~/MyPlaybooks/ansible.cfg** ファイルを以下の内容で作成します。

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

3. **~/MyPlaybooks/inventory** ファイルを以下の内容で作成します。

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

この設定は、これらの場所にあるホストの 2 つのホストグループ (**eu** と **us**) を定義します。さらに、この設定は、**eu** および **us** グループのすべてのホストを含む **ipaserver** ホストグループを定義します。

4. [オプション] SSH 公開鍵および秘密鍵を作成します。テスト環境でのアクセスを簡素化するには、秘密鍵にパスワードを設定しないでください。

```
$ ssh-keygen
```

5. 各マネージドノードの IdM **admin** アカウントに SSH 公開鍵をコピーします。

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

これらのコマンドでは、IdM **管理者** パスワードを入力します。

6. Vault パスワードを含む **password_file** ファイルを作成します。

```
redhat
```

7. ファイルを変更する権限を変更します。

```
$ chmod 0600 password_file
```

8. IdM の **admin** パスワードを保存する **secret.yml** Ansible Vault を作成します。

a. Vault パスワードを保存するように **password_file** を設定します。

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

b. プロンプトが表示されたら、**secret.yml** ファイルの内容を入力します。

```
ipaadmin_password: Secret123
```

注記

Playbook で暗号化された **ipaadmin_password** を使用するには、**vars_file** ディレクティブを使用する必要があります。たとえば、IdM ユーザーを削除する単純な Playbook は次のようになります。

```
---
- 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
```

Playbook を実行するときに、**--vault-password-file=password_file** オプションを追加して、Ansible に Vault パスワードを使用して **ipaadmin_password** を復号するように指示します。以下に例を示します。

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

警告

セキュリティー上の理由から、各セッションの終了時に Vault パスワードファイルを削除し、新しいセッションの開始時に手順 6 - 8 を繰り返してください。

関連情報

- ansible-freeipa Playbook に必要な認証情報を提供するさまざまな方法
- Ansible Playbook で Identity Management サーバーのインストール
- インベントリーの構築方法

5.2. ANSIBLE-FREEIPA PLAYBOOK に必要な認証情報を提供するさまざまな方法

ansible-freeipa ロールおよびモジュールを使用する Playbook の実行に必要な認証情報を提供するための様々な方法には長所と短所があります。

Playbook にパスワードを平文で保存する

利点:

- Playbook を実行するたびにプロンプトが表示されません。
- 実装が簡単。

短所:

- ファイルにアクセスできるすべてのユーザーがパスワードを読み取ることができます。内部または外部のリポジトリーなどで間違った権限を設定してファイルを共有すると、セキュリティーが損なわれる可能性があります。
- 高いメンテナンス作業: パスワードが変更された場合は、すべての Playbook で変更する必要があります。

Playbook の実行時に対話的にパスワードを入力する

利点:

- パスワードはどこにも保存されないため、誰もパスワードを盗むことはできません。
- パスワードを簡単に更新できます。
- 実装が簡単。

短所:

- スクリプトで Ansible Playbook を使用している場合は、パスワードを対話的に入力する必要があると不便な場合があります。

パスワードを Ansible Vault に保存し、Vault パスワードをファイルに保存します。

利点:

- ユーザーパスワードは暗号化されて保存されます。
- 新しい Ansible Vault を作成することで、ユーザーパスワードを簡単に更新できます。
- **ansible-vault rekey --new-vault-password-file=NEW_VAULT_PASSWORD_FILE secret.yml** コマンドを使用して、ansible Vault を保護するパスワードファイルを簡単に更新できます。
- スクリプトで Ansible Playbook を使用している場合は、Ansible Vault を対話的に保護するパスワードを入力する必要がないのは便利です。

短所:

- 機密性の高いプレーンテキストパスワードを含むファイルは、ファイルのアクセス許可やその他のセキュリティー対策によって保護することが重要です。

パスワードを Ansible Vault に保存し、Vault のパスワードを対話的に入力する

利点:

- ユーザーパスワードは暗号化されて保存されます。
- Vault のパスワードはどこにも保存されていないため、誰も盗むことはできません。
- 新しい Ansible Vault を作成することで、ユーザーパスワードを簡単に更新できます。
- **ansible-vault rekey file_name** コマンドを使用して、Vault パスワードも簡単に更新できます。

短所:

- スクリプトで Ansible Playbook を使用している場合は、Vault のパスワードを対話的に入力する必要があると不便な場合があります。

関連情報

- Ansible Playbook を使用して IdM を管理するためのコントロールノードと管理ノードの準備
- What is Zero trust?
- Ansible Vault による機密データの保護

第6章 ANSIBLE PLAYBOOK でのグローバル IDM 設定

Ansible **設定** モジュールを使用すると、Identity Management (IdM) のグローバル設定パラメーターを取得および設定できます。

- Ansible Playbook での IdM 設定の取得
- Ansible Playbook での IdM CA 更新サーバーの設定
- Ansible Playbook での IdM ユーザーのデフォルトシェルの設定
- Ansible を使用した IdM ドメインの NETBIOS 名の設定
- Ansible を使用して IdM ユーザーとグループに SID があることを確認する

6.1. ANSIBLE PLAYBOOK での IDM 設定の取得

以下の手順では、Ansible Playbook を使用して、現在のグローバル IdM 設定に関する情報を取得する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. Ansible Playbook ファイル **/usr/share/doc/ansible-freeipa/playbooks/config/retrieve-config.yml** を開いて編集します。

```
---
- 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. 以下を変更してファイルを調整します。
   - IdM 管理者のパスワード。
   - その他の値 (必要な場合)。
3. ファイルを保存します。
4. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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. ANSIBLE PLAYBOOK での IDM CA 更新サーバーの設定

組み込みの認証局 (CA) を使用する Identity Management (IdM) デプロイメントでは、CA 更新サーバーが IdM システム証明書を維持および更新します。IdM デプロイメントを確実に堅牢化します。

IdM CA 更新サーバーロールの詳細は、IdM CA 更新サーバーの使用 を参照してください。

以下の手順では、Ansible Playbook を使用して IdM CA 更新サーバーを設定する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. 必要に応じて、現在の IdM CA 更新サーバーを特定します。

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

2. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

3. Ansible Playbook ファイル **/usr/share/doc/ansible-freeipa/playbooks/config/set-ca-renewal-master-server.yml** を開いて編集します。

```
---
- 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. 以下を変更してファイルを調整します。

   - **ipaadmin_password** 変数で設定した IdM 管理者のパスワード
   - **ca_renewal_master_server** 変数で設定した CA 更新サーバーの名前。

5. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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
```

検証手順

CA 更新サーバーが変更されたことを確認します。

1. IdM 管理者として **ipaserver** にログインします。

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

2. IdM CA 更新サーバーの ID を要求します。

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

この出力には、**carenewal.idm.example.com** サーバーが新しい CA 更新サーバーであることが分かります。

6.3. ANSIBLE PLAYBOOK での IDM ユーザーのデフォルトシェルの設定

シェルとは、コマンドを受け入れて変換するプログラムです。Red Hat Enterprise Linux (RHEL) では、**bash**、**sh**、**ksh**、**zsh**、**fish** など、複数のシェルが利用できます。**Bash** (または **/bin/bash**) は、ほとんどの Linux システムで一般的なシェルです。通常は、RHEL のユーザーアカウントのデフォルトシェルです。

以下の手順では、Ansible Playbook を使用して、IdM ユーザーのデフォルトシェルとして別のシェルである **sh** を設定する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. 必要に応じて、Ansible Playbook **retrieve-config.yml** を使用して、IdM ユーザーの現在のシェルを特定します。詳細は、Ansible Playbook での IdM 設定の取得 を参照してください。
2. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

3. Ansible Playbook **/usr/share/doc/ansible-freeipa/playbooks/config/ensure-config-options-are-set.yml** ファイルを開いて編集します。

```
---
- 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. 以下を変更してファイルを調整します。
  - **ipaadmin_password** 変数で設定した IdM 管理者のパスワード
  - **defaultshell** 変数で設定されている IdM ユーザーのデフォルトのシェルが **/bin/sh** に設定されます。

5. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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
```

検証手順

IdM で新しいセッションを開始すると、デフォルトのユーザーシェルが変更されていることを確認できます。

1. IdM 管理者として **ipaserver** にログインします。

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

2. 現在のシェルを表示します。

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

ログインしているユーザーが **sh** シェルを使用している。

6.4. ANSIBLE を使用した IDM ドメインの NETBIOS 名の設定

NetBIOS 名は、Microsoft Windows (SMB) タイプの共有およびメッセージングに使用されます。NetBIOS 名を使用して、ドライブをマップしたり、プリンターに接続したりできます。

以下の手順に従って、Ansible Playbook を使用して Identity Management (IdM) ドメインの NetBIOS 名を設定します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - **ansible-freeipa** パッケージをインストールしている。

想定条件

- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible ボールトに **ipaadmin_password** が保存されており、ボールトファイルのパスワードを知っていることを前提としています。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **netbios-domain-name-present.yml** Ansible Playbook ファイルを作成します。
3. 以下の内容をファイルに追加します。

```
---
- 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. ファイルを保存します。
5. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

プロンプトが表示されたら、ボールトファイルのパスワードを入力します。

関連情報

- NetBIOS 名を設定するためのガイドライン

6.5. ANSIBLE を使用して IDM ユーザーとグループに SID があることを確認する

Identity Management (IdM) サーバーは、ローカルドメインの ID 範囲のデータに基づいて、一意のセキュリティー識別子 (SID) を IdM ユーザーおよびグループに内部的に割り当てることができます。SID は、ユーザーオブジェクトとグループオブジェクトに格納されます。

IdM ユーザーとグループに SID があることを確認する目的は、特権属性証明書 (PAC) の生成を許可することです。これは、IdM-IdM 信頼への最初のステップです。IdM ユーザーおよびグループが SID を持っている場合、IdM は PAC データを使用して Kerberos チケットを発行できます。

次の目標を達成するには、次の手順に従ってください。

- 既存の IdM ユーザーおよびユーザーグループの SID を生成します。
- IdM の新しいユーザーおよびグループの SID の生成を有効にします。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - **ansible-freeipa** パッケージをインストールしている。

想定条件

- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible ボールトに **ipaadmin_password** が保存されており、ボールトファイルのパスワードを知っていることを前提としています。

手順

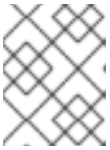
1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **sids-for-users-and-groups-present.yml** Ansible Playbook ファイルを作成します。
3. 以下の内容をファイルに追加します。

```
---
- 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
```

enable_sid 変数は、将来の IdM ユーザーおよびグループの SID 生成を有効にします。**add_sids** 変数は、既存の IdM ユーザーおよびグループの SID を生成します。



注記

add_sids: true を使用する場合は、**enable_sid** 変数を **true** に設定する必要もあります。

4. ファイルを保存します。
5. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

- プロンプトが表示されたら、ボールトファイルのパスワードを入力します。

関連情報

- IdM ID 範囲におけるセキュリティーおよび相対識別子のロール.

6.6. 関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-config.md** を参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/config** ディレクトリーのサンプルの Playbook を参照してください。

第7章 ANSIBLE PLAYBOOK を使用したユーザーアカウントの管理

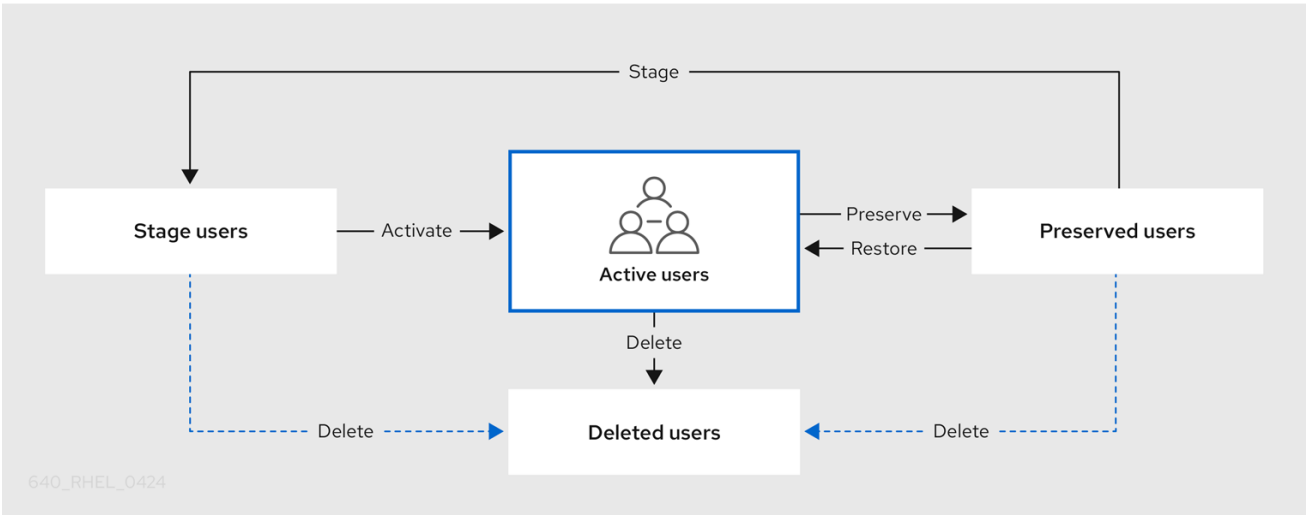
Ansible Playbook を使用して IdM のユーザーを管理できます。ユーザーのライフサイクル を示した後、本章では以下の操作に Ansible Playbook を使用する方法を説明します。

- **YML** ファイルに直接リストされている ユーザーを 1 つ存在させる 手順
- **YML** ファイルに直接リストされている ユーザーを複数存在させる 手順
- **YML** ファイルから参照される **JSON** ファイルにリストされている ユーザーを複数存在させる 手順
- **YML** ファイルに直接リストされている ユーザーがないことを確認 します。

7.1. ユーザーのライフサイクル

Identity Management (IdM) は、次の 3 つのユーザーアカウント状態に対応します

- **ステージ** ユーザーは認証できません。これは初期状態です。アクティブユーザーに必要なユーザーアカウントプロパティーをすべて設定できるわけではありません (例: グループメンバーシップ)。
- **アクティブ** ユーザーは認証が可能です。必要なユーザーアカウントプロパティーはすべて、この状態で設定する必要があります。
- **保存済み** ユーザーは、以前にアクティブであったユーザーで、現在は非アクティブであるとみなされており、IdM への認証ができません。保存済みユーザーには、アクティブユーザーのときに有効になっていたアカウントプロパティーの大部分が保持されていますが、ユーザーグループからは除外されています。



IdM データベースからユーザーエントリーを完全に削除できます。

重要

削除したユーザーアカウントを復元することはできません。ユーザーアカウントを削除すると、そのアカウントに関連する情報がすべて完全に失われます。

新規管理者は、デフォルトの管理ユーザーなど、管理者権限を持つユーザーのみが作成できます。すべての管理者アカウントを誤って削除した場合は、Directory Manager が、Directory Server に新しい管理者を手動で作成する必要があります。

警告

admin ユーザーを削除しないでください。**admin** は IdM で必要な事前定義ユーザーであるため、この操作では特定のコマンドで問題が生じます。別の **admin** ユーザーを定義して使用する場合は、管理者権限を少なくとも 1 つのユーザーに付与してから、**ipa user-disable admin** を使用して、事前定義された admin ユーザーを無効にします。

警告

ローカルユーザーを IdM に追加しないでください。Name Service Switch (NSS) は、ローカルユーザーとグループを解決する前に、IdM ユーザーとグループを常に解決します。つまり、たとえば IdM グループのメンバーシップは、ローカルユーザーでは機能しません。

7.2. ANSIBLE PLAYBOOK を使用して IDM ユーザーを存在させる手順

以下の手順では、Ansible Playbook を使用して IdM にユーザーを 1 つ存在させる方法を説明します。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. IdM に存在させるユーザーのデータを指定して Ansible Playbook ファイルを作成します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/user/add-user.yml** ファイルのサンプルをコピーして変更し、簡素化できます。たとえば、**idm_user** という名前のユーザーを作成し、**Password123** をユーザーパスワードとして追加するには、次のコマンドを実行します。

```
---
- 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
```

ユーザーを追加するには、以下のオプションを使用する必要があります。

- **名前**: ログイン名
- **first**: 名前 (名) の文字列
- **last**: 名前 (姓) の文字列

利用可能なユーザーオプションの完全なリストは、**/usr/share/doc/ansible-freeipa/README-user.md** Markdown ファイルを参照してください。

注記

update_password: on_create オプションを使用する場合には、Ansible はユーザー作成時にのみユーザーパスワードを作成します。パスワードを指定してユーザーが作成されている場合には、Ansible では新しいパスワードは生成されません。

3. Playbook を実行します。

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

検証手順

- **ipa user-show** コマンドを使用して、新しいユーザーアカウントが IdM に存在するかどうかを確認できます。

  1. admin として **ipaserver** にログインします。

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

  2. admin の Kerberos チケットを要求します。

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

  3. **idm_user** に関する情報を要求します。

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

  idm_userという名前のユーザー が IdM に存在しています。

7.3. ANSIBLE PLAYBOOK を使用して IDM ユーザーを複数存在させる手順

以下の手順では、Ansible Playbook を使用して IdM にユーザーを複数存在させる方法を説明します。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. IdM に存在させるユーザーのデータを指定して Ansible Playbook ファイルを作成します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/user/ensure-users-present.yml** ファイルのサンプルをコピーして変更し、簡素化できます。たとえば、ユーザー **idm_user_1**、**idm_user_2**、**idm_user_3** を作成し、**idm_user_1** のパスワードを **Password123** として追加します。

```
---
- 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
```

注記

update_password: on_create オプションを指定しないと、Ansible は Playbook が実行されるたびにユーザーパスワードを再設定します。最後に Playbook が実行されてからユーザーがパスワードを変更した場合には、Ansible はパスワードを再設定します。

3. Playbook を実行します。

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

検証手順

- **ipa user-show** コマンドを使用して、ユーザーアカウントが IdM に存在するかどうかを確認できます。

1. 管理者として **ipaserver** にログインします。

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

2. **idm_user_1** に関する情報を表示します。

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

idm_user_1 という名前のユーザーが IdM に存在しています。

7.4. ANSIBLE PLAYBOOK を使用して JSON ファイルに指定してある複数の IDM ユーザーを存在させる手順

以下の手順では、Ansible Playbook を使用して IdM に複数のユーザーを存在させる方法を説明します。ユーザーは **JSON** ファイルに保存されます。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. 必要なタスクが含まれる Ansible Playbook ファイルを作成します。存在させるユーザーのデータが指定された **JSON** ファイルを参照します。この手順は、**/usr/share/doc/ansible-freeipa/ensure-users-present-ymlfile.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
- 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. **users.json** ファイルを作成し、IdM ユーザーを追加します。この手順を簡素化するには、**/usr/share/doc/ansible-freeipa/playbooks/user/users.json** ファイルのサンプルをコピーして変更できます。たとえば、ユーザー **idm_user_1**、**idm_user_2**、**idm_user_3** を作成し、**idm_user_1** のパスワードを **Password123** として追加します。

```
{
  "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. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

検証手順

- **ipa user-show** コマンドを使用して、ユーザーアカウントが IdM に存在するかどうかを確認できます。

  1. 管理者として **ipaserver** にログインします。

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

2. **idm_user_1** に関する情報を表示します。

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

idm_user_1 という名前のユーザーが IdM に存在しています。

7.5. ANSIBLE PLAYBOOK を使用してユーザーが存在しないことを確認する手順

以下の手順では、Ansible Playbook を使用して、特定のユーザーが IdM に存在しないことを確認する方法を説明します。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. IdM に存在させないユーザーを指定して Ansible Playbook ファイルを作成します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/user/ensure-users-present.yml** ファイルのサンプルをコピーして変更し、簡素化できます。たとえば、ユーザー **idm_user_1**、**idm_user_2**、**idm_user_3** を削除するには、次のコマンドを実行します。

```
---
```

```
- 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. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

検証手順

ipa user-show コマンドを使用して、ユーザーアカウントが IdM に存在しないことを確認できます。

1. 管理者として **ipaserver** にログインします。

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

2. **idm_user_1** に関する要求情報:

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

idm_user_1 という名前のユーザーは IdM に存在しません。

7.6. 関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-user.md** Markdown ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/user** ディレクトリーのサンプルの Ansible Playbook を参照してください。

第8章 ANSIBLE PLAYBOOK を使用したユーザーグループの管理

本セクションでは、Ansible Playbook を使用したユーザーグループの管理について紹介します。

ユーザーグループは、共通の特権、パスワードポリシーなどの特性が指定された一連のユーザーです。

Identity Management (IdM) のユーザーグループには以下が含まれます。

- IdM ユーザー
- 他の IdM ユーザーグループ
- 外部ユーザー (IdM の外部に存在するユーザー)

このセクションには、以下のトピックが含まれます。

- IdM のさまざまなグループタイプ
- 直接および間接のグループメンバー
- Ansible Playbook を使用して IdM グループおよびグループメンバーの存在を確認する
- Ansible を使用して AD ユーザーが IdM を管理できるようにする手順
- Ansible Playbook を使用して IDM ユーザーグループにメンバーマネージャーを存在させる手順
- Ansible Playbook を使用して IDM ユーザーグループにメンバーマネージャーを存在させないようにする手順

8.1. IDM のさまざまなグループタイプ

IdM は、以下のグループタイプをサポートします。

POSIX グループ (デフォルト)

POSIX グループは、メンバーの Linux POSIX 属性に対応します。Active Directory と対話するグループは POSIX 属性を使用できないことに注意してください。
POSIX 属性は、ユーザーを個別のエンティティーとして識別します。ユーザーに関連する POSIX 属性の例には、**uidNumber** (ユーザー番号 (UID))、および **gidNumber** (グループ番号 (GID)) が含まれます。

非 POSIX グループ

非 POSIX グループは、POSIX 属性に対応していません。たとえば、これらのグループには GID が定義されていません。
このタイプのグループのすべてのメンバーは、IdM ドメインに属している必要があります。

外部グループ

外部グループを使用して、以下のような IdM ドメイン外の ID ストアに存在するグループメンバーを追加できます。

- ローカルシステム
- Active Directory ドメイン
- ディレクトリーサービス

外部グループは、POSIX 属性に対応していません。たとえば、これらのグループには GID が定義されていません。

表8.1 デフォルトで作成されたユーザーグループ

グループ名	デフォルトのグループメンバー
ipausers	すべての IdM ユーザー
admins	管理権限を持つユーザー (デフォルトの **admin** ユーザーを含む)
editors	特権のないレガシーグループ
trust admins	Active Directory 信頼を管理する権限を持つユーザー

ユーザーをユーザーグループに追加すると、ユーザーはグループに関連付けられた権限とポリシーを取得します。たとえば、ユーザーに管理権限を付与するには、ユーザーを **admins** グループに追加します。

警告

admins グループを削除しないでください。**admins** は IdM で必要な事前定義グループであるため、この操作では特定のコマンドで問題が生じます。

さらに、IdM で新しいユーザーが作成されるたびに、IdM は、デフォルトで **ユーザーのプライベートグループ** を作成します。プライベートグループの詳細は、プライベートグループのないユーザーの追加 を参照してください。

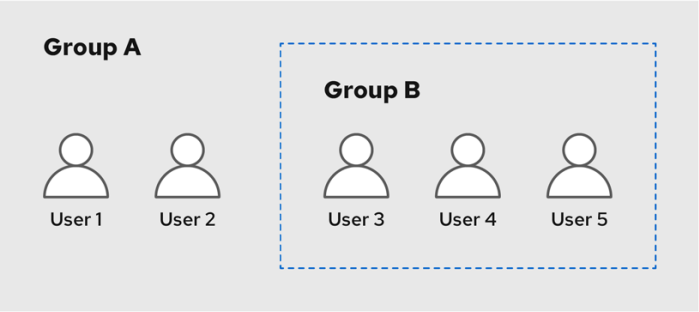
8.2. 直接および間接のグループメンバー

IdM のユーザーグループ属性は、直接メンバーと間接メンバーの両方に適用されます。グループ B がグループ A のメンバーである場合、グループ B のすべてのユーザーはグループ A の間接メンバーと見なされます。

たとえば、以下の図の場合:

- ユーザー 1 とユーザー 2 は、グループ A の **直接メンバー** です。
- ユーザー 3、ユーザー 4、およびユーザー 5 は、グループ A の **間接メンバー** です。

図8.1 直接および間接グループメンバーシップ



ユーザーグループ A にパスワードポリシーを設定すると、そのポリシーはユーザーグループ B のすべてのユーザーにも適用されます。

8.3. ANSIBLE PLAYBOOK を使用して IDM グループおよびグループメンバーの存在を確認する

以下の手順では、Ansible Playbook を使用して、IdM グループとグループメンバー (ユーザーとユーザーグループの両方) を存在させる方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- Ansible Playbook で参照するユーザーが IdM に存在する。Ansible を使用してユーザーの存在を確認する方法は、Ansible Playbook を使用したユーザーアカウントの管理 を参照してください。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. 必要なユーザーおよびグループ情報を使用して Ansible Playbook ファイルを作成します。

```
---
- 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. Playbook を実行します。

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

検証手順

ipa group-show コマンドを使用すると、ops グループに sysops および appops がダイレクトメンバーとして追加され、**idm_user** が間接メンバーとして追加されているかどうかを確認できます。

1. 管理者として **ipaserver** にログインします。

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

2. ops についての情報を表示します。

```
ipaserver]$ ipa group-show ops
  Group name: ops
  GID: 1234
```

```
Member groups: sysops, appops
Indirect Member users: idm_user
```

appops および **sysops** グループ (後者には **idm_user** ユーザーを含む) が IdM に存在します。

関連情報

- **/usr/share/doc/ansible-freeipa/README-group.md** Markdown ファイルを参照してください。

8.4. ANSIBLE を使用して単一のタスクで複数の IDM グループを追加する

ansible-freeipa ipagroup モジュールを使用すると、単一の Ansible タスクで複数の Identity Management (IdM) ユーザーグループを追加、変更、削除できます。そのためには、**ipagroup** モジュールの **groups** オプションを使用します。

groups オプションを使用すると、特定のグループにのみ適用される複数のグループ変数を指定することもできます。このグループは **name** 変数で定義します。これは、**groups** オプションの唯一の必須変数です。

この手順を単一のタスクで完了して、IdM に **sysops** グループと **appops** グループが存在することを確認します。sysops グループは非 posix グループとして定義し、appops グループは外部グループとして定義します。

前提条件

- コントロールノード:
  - Ansible バージョン 2.14 以降を使用している。
  - **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
  - RHEL 9.3 以降を使用している。
  - **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。

手順

1. 次の内容を含む Ansible Playbook ファイル **add-nonposix-and-external-groups.yml** を作成します。

```
---
- name: Playbook to add nonposix and external groups
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Add nonposix group sysops and external group appops
    ipagroup:
      ipaadmin_password: "{{ ipaadmin_password }}"
      groups:
```

```
    - name: sysops
      nonposix: true
    - name: appops
      external: true
```

2. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/add-nonposix-
and-external-groups.yml
```

関連情報

- **ansible-freeipa** アップストリームドキュメントの group モジュール

8.5. ANSIBLE を使用して AD ユーザーが IDM を管理できるようにする手順

Ansible Playbook を使用してユーザー ID オーバーライドが Identity Management (IdM) グループに存在することを確認するには、次の手順に従います。ユーザー ID オーバーライドは、Active Directory (AD) とのトラストを確立した後に、デフォルトの Trust View で作成した AD ユーザーのオーバーライドです。Playbook を実行すると、AD 管理者などの AD ユーザーは、2 つの異なるアカウントとパスワードを持たなくても IdM を完全に管理できるようになります。

前提条件

- IdM **admin** のパスワードを把握している。
- AD とのトラストをインストール している。
- IdM に AD ユーザーのユーザー ID オーバーライドがすでに存在する。存在しない場合は、**ipa idoverrideuser-add 'default trust view' ad_user@ad.example.com** コマンドで作成します。
- ユーザー ID オーバーライドを追加しようとしているグループが、IdM にすでに存在する 。
- IdM 4.8.7 バージョン以降の IdM を使用している。サーバーにインストールされている IdM のバージョンを表示するには、**ipa --version** を実行します。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. 次の内容で **add-useridoverride-to-group.yml** playbook を作成します。

```
---
- 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
```

上記の例では、以下のようになります。

- Secret123 は IdM **admin** パスワードです。
- **admins** は、**ad_user@ad.example.com** ID オーバーライドを追加する IdM POSIX グループの名前です。このグループのメンバーには、完全な管理者権限があります。
- **ad_user@ad.example.com** は、AD 管理者のユーザー ID オーバーライドです。ユーザーは、信頼が確立された AD ドメインに保存されます。

3. ファイルを保存します。

4. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- AD ユーザーの ID のオーバーライド
- /usr/share/doc/ansible-freeipa/README-group.md
- /usr/share/doc/ansible-freeipa/playbooks/user
- Using ID views in Active Directory environments
- IdM を管理する AD ユーザーの有効化

8.6. ANSIBLE PLAYBOOK を使用して IDM ユーザーグループにメンバーマネージャーを存在させる手順

以下の手順では、Ansible Playbook を使用して、ユーザーとユーザーグループ両方の IdM メンバーマネージャーが存在させる方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- メンバーマネージャーとして追加するユーザーまたはグループの名前と、メンバーマネージャーが管理するグループ名を用意する。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. 必要なユーザーおよびグループメンバー管理情報を使用して、Ansible Playbook ファイルを作成します。

```
---
- 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. Playbook を実行します。

```
$ 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
```

検証手順

ipa group-show コマンドを使用すると、**group_a** グループにメンバーマネージャーの **test** が含まれており、**group_admins** が **group_a** のメンバーであることが確認できます。

1. 管理者として **ipaserver** にログインします。

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

2. **managergroup1** についての情報を表示します。

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

関連情報

- **ipa host-add-member-manager --help** を参照してください。
- **ipa** の man ページを参照してください。

8.7. ANSIBLE PLAYBOOK を使用して IDM ユーザーグループにメンバーマネージャーを存在させないようにする手順

以下の手順では、Ansible Playbook を使用して、ユーザーとユーザーグループ両方の IdM メンバーマネージャーが存在させないようにする方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- 削除する既存のメンバーマネージャーのユーザーまたはグループの名前と、そのメンバーマネージャーが管理するグループ名が必要です。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. 必要なユーザーおよびグループメンバー管理情報を使用して、Ansible Playbook ファイルを作成します。

```
---
- 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. Playbook を実行します。

```
$ 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
```

検証手順

ipa group-show コマンドを使用すると、**group_a** グループにメンバーマネージャーの **test** が含まれておらず、**group_admins** が **group_a** のメンバーであることが確認できます。

1. 管理者として **ipaserver** にログインします。

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

2. group_a についての情報を表示します。

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

関連情報

- **ipa host-remove-member-manager --help** を参照してください。
- **ipa** の man ページを参照してください。

第9章 ANSIBLE を使用した IDM のグループメンバーシップの自動化

自動グループメンバーシップを使用すると、ユーザーとホストのユーザーグループとホストグループを、その属性に基づいて自動的に割り当てることができます。たとえば、以下を行うことができます。

- 従業員のユーザーエントリーを、従業員のマネージャー、場所、役職などの属性に基づいてグループに分割します。コマンドラインに **ipa user-add --help** と入力すると、すべての属性をリスト表示できます。
- ホストを、クラス、場所、またはその他の属性に基づいてグループに分割します。コマンドラインに **ipa host-add --help** と入力すると、すべての属性をリスト表示できます。
- 全ユーザーまたは全ホストを 1 つのグローバルグループに追加する。

Red Hat Ansible Engine を使用すると、Identity Management (IdM) で自動グループメンバーシップの管理を自動化できます。

このセクションでは、以下のトピックについて説明します。

- Ansible を使用した IdM ユーザーグループの automember ルールが存在することの確認
- Ansible を使用した IdM ユーザーグループの automember ルールに条件が存在することの確認
- Ansible を使用した IdM ユーザーグループの automember ルールに条件がないことの確認
- Ansible を使用した IdM グループの automember ルールがないことの確認
- Ansible を使用した IdM ホストグループの automember ルールに条件が存在することの確認

9.1. ANSIBLE を使用した IDM ユーザーグループの AUTOMEMBER ルールが存在することの確認

以下の手順では、Ansible Playbook を使用して、Identity Management (IdM) グループの **automember** ルールが存在することを確認する方法について説明しますこの例では、**testing_group** ユーザーグループに対して **automember** ルールの存在が保証されます。

前提条件

- IdM **admin** のパスワードを把握している。
- **testing_group** ユーザーグループが IdM に存在します。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/automember/** ディレクトリーにある **automember-group-present.yml** Ansible Playbook ファイルをコピーします。

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

3. **automember-group-present-copy.yml** ファイルを開いて編集します。

4. **ipaautomember** タスクセクションで次の変数を設定して、ファイルを調整します。

   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **name** 変数を testing_group に設定します。
   - **automember_type** 変数を group に設定します。
   - **state** 変数は **present** に設定されていることを確認します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

9.2. ANSIBLE を使用した、IDM ユーザーグループの AUTOMEMBER ルールに指定した条件が存在することの確認

関連情報

以下の手順では、Ansible Playbook を使用して、Identity Management (IdM) グループの **automember** ルールに、指定した条件が存在することを確認する方法について説明しますこの例では、**testing_group** グループに対して、**automember** ルールに UID 関連の条件が存在することが保証されます。**.*** 条件を指定することで、今後使用する IdM ユーザーがすべて自動的に **testing_group** のメンバーになるようにします。

前提条件

- IdM **admin** のパスワードを把握している。
- **testing_group** ユーザーグループおよび automember ユーザーグループルールが IdM に存在します。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/automember/** ディレクトリーにある **automember-hostgroup-rule-present.yml** Ansible Playbook ファイルをコピーして、名前を付けます (**automember-usergroup-rule-present.yml** など)。

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

3. **automember-usergroup-rule-present.yml** を開いて編集します。
4. 次のパラメーターを変更して、ファイルを調整します。
   - ユースケースに対応するように Playbook の名前を変更します (例: **Automember user group rule member present**)。
   - ユースケースに合わせて、タスクの名前を変更します (例: **Ensure an automember condition for a user group is present**)。
   - **ipaautomember** タスクセクションで、以下の変数を設定します。
     - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。

- **name** 変数を testing_group に設定します。
- **automember_type** 変数を **group** に設定します。
- **state** 変数は **present** に設定されていることを確認します。
- **action** 変数が **member** に設定されていることを確認します。
- **inclusive key** 変数を **UID** に設定します。
- **inclusive expression** 変数を .* に設定します。

以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

検証手順

1. IdM 管理者としてログインします。

```
$ kinit admin
```

2. ユーザーを追加します。以下に例を示します。

```
$ 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. ANSIBLE を使用した IDM ユーザーグループの AUTOMEMBER ルールに条件がないことの確認

関連情報

以下の手順では、Ansible Playbook を使用して、Identity Management (IdM) グループの **automember** ルールに条件がないことを確認する方法を説明します。この例では、**automember** ルールに条件がないことが保証されており、**initials** が**dp** であるユーザーを含める必要があることを指定しています。automember ルールが **testing_group** グループに適用されます。条件を適用することにより、今後は、イニシャルが**dp**である IdM ユーザーが **testing_group**のメンバーにならないようにします。

前提条件

- IdM **admin** のパスワードを把握している。
- **testing_group** ユーザーグループおよび automember ユーザーグループルールが IdM に存在します。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/automember/** ディレクトリーにある **automember-hostgroup-rule-absent.yml** Ansible Playbook ファイルをコピーして、名前を付けます (**automember-usergroup-rule-absent.yml** など)。

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

3. **automember-usergroup-rule-absent.yml** を開いて編集します。
4. 次のパラメーターを変更して、ファイルを調整します。
   - ユースケースに対応するように Playbook の名前を変更します (例: **Automember user**

- ユースケースに対応するように Playbook の名前を変更します (例: **Automember user group rule member absent**)。
- ユースケースに合わせて、タスクの名前を変更します (例: **Ensure an automember condition for a user group is absent**)。
- **ipaautomember** タスクセクションで、以下の変数を設定します。
  - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
  - **name** 変数を **testing_group** に設定します。
  - **automember_type** 変数を **group** に設定します。
  - **state** 変数は、**absent** に設定されていることを確認します。
  - **action** 変数が **member** に設定されていることを確認します。
  - **inclusive key** 変数を **initials** に設定します。
  - **inclusive expression** 変数を **dp** に設定します。

以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

検証手順

1. IdM 管理者としてログインします。

```
$ kinit admin
```

2. automember グループを表示します。

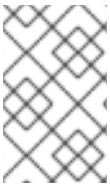
```
$ ipa automember-show --type=group testing_group
 Automember Rule: testing_group
```

出力に **Inclusive Regex: initials=dp** エントリーがない場合は、**testing_group** automember ルールに指定した条件が含まれていないことを確認します。

9.4. ANSIBLE を使用した IDM ユーザーグループの AUTOMEMBER ルールがないことの確認

関連情報

以下の手順では、Ansible Playbook を使用して、Identity Management (IdM) グループに **automember** ルールがないことを確認する方法を説明します。この例では、**testing_group** グループに **automember** ルールがないことが保証されます。



注記

automember ルールを削除すると、そのルールに関連付けられた条件もすべて削除されます。ルールから特定の条件のみを削除するには、Ansible を使用した IdM ユーザーグループの automember ルールに条件がないことの確認 を参照してください。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/automember/** ディレクトリーにある **automember-group-absent.yml** Ansible Playbook ファイルをコピーします。

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

3. **automember-group-absent-copy.yml** を開いて編集します。

4. **ipaautomember** タスクセクションで次の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **name** 変数を **testing_group** に設定します。
   - **automember_type** 変数を **group** に設定します。
   - **state** 変数は、**absent** に設定されていることを確認します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-automember.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/automember** ディレクトリーを参照してください。

9.5. ANSIBLE を使用した IDM ホストグループの AUTOMEMBER ルールに条件が存在することの確認

以下の手順に従って、Ansible を使用して、IdM ホストグループの automember ルールに条件が存在することを確認します。この例では、**FQDN** が **.*.idm.example.com** のホストが、**primary_dns_domain_hosts** ホストグループのメンバーであることと、**FQDN** が **.*.example.org** であるホストが **primary_dns_domain_hosts** ホストグループのメンバーではないことを確認する方法を説明します。

前提条件

- IdM **admin** のパスワードを把握している。

- **primary_dns_domain_hosts** ホストグループおよび automember ホストグループルールが IdM に存在します。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/automember/** ディレクトリーにある **automember-hostgroup-rule-present.yml** Ansible Playbook ファイルをコピーします。

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

3. **automember-hostgroup-rule-present-copy.yml** を開いて編集します。
4. **ipaautomember** タスクセクションで次の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **name** 変数を primary_dns_domain_hosts に設定します。
   - **automember_type** を hostgroup に設定します。
   - **state** 変数は **present** に設定されていることを確認します。
   - **action** 変数が **member** に設定されていることを確認します。
   - **inclusive key** 変数が **fqdn** に設定されていることを確認します。
   - 対応する **inclusive expression** 変数を.*.idm.example.comに設定します。
   - **exclusive key** 変数を **UID** に設定します。
   - 対応する **exclusive expression** 変数を .*.example.org に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-automember.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/automember** ディレクトリーを参照してください。

第10章 ANSIBLE PLAYBOOK を使用した IDM でのセルフサービスルールの管理

本セクションでは、Identity Management (IdM) のセルフサービスルールを紹介し、Ansible Playbook を使用してセルフサービスアクセスルールを作成および編集する方法を説明します。セルフサービスのアクセス制御ルールにより、IdM エンティティーは IdM Directory Server エントリーで指定の操作を実行できます。

- IdM でのセルフサービスアクセス制御
- Ansible を使用してセルフサービスルールを存在させる手順
- Ansible を使用してセルフサービスルールがないことを確認する手順
- Ansible を使用してセルフサービスルールに固有の属性を含める手順
- Ansible を使用してセルフサービスルールに固有の属性を含めいないようにする手順

10.1. IDM でのセルフサービスアクセス制御

セルフサービスのアクセス制御ルールは、Identity Management (IdM) エンティティーが IdM Directory Server エントリーで実行できる操作を定義します (例: IdM ユーザーが独自のパスワードを更新できるなど)。

この制御方法では、認証された IdM エンティティーがその LDAP エントリー内の特定の属性を編集できますが、エントリー全体での **追加** や **削除** の操作は許可されません。

警告

セルフサービスのアクセス制御規則を使用する場合は注意が必要です。アクセス制御ルールを誤って設定すると、エンティティーの特権が誤って昇格する可能性があります。

10.2. ANSIBLE を使用してセルフサービスルールを存在させる手順

以下の手順では、Ansible Playbook を使用してセルフサービスルールを定義し、Identity Management (IdM) サーバーに存在させる方法を説明します。この例では、**ユーザーが自分の名前の情報を管理できる** 新しいルールでは、ユーザーに、自分の **givenname**、**displayname**、**title**、および **initials** 属性を変更できるよ権限を付与します。たとえば、表示名や初期などを変更することができます。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。

- ~/MyPlaybooks/ ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. ~/ MyPlaybooks/ ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/selfservice/** ディレクトリーにある **selfservice-present.yml** のコピーを作成します。

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

3. Ansible Playbook ファイル (**selfservice-present-copy.yml**) を開きます。

4. **ipaselfservice** タスクセクションに以下の変数を設定してファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、新しいセルフサービスルールの名前に設定します。
   - **permission** 変数は、付与するパーミッションをコンマ区切りのリスト (**read** および **write**) で設定します。
   - **attribute** 変数は、ユーザーが管理できる属性 (**givenname**、**displayname**、**title**、および **initials**) をリストとして設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- IdM でのセルフサービスアクセス制御 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-selfservice.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/selfservice** ディレクトリーを参照してください。

10.3. ANSIBLE を使用してセルフサービスルールがないことを確認する手順

以下の手順では、Ansible Playbook を使用して、指定したセルフサービスルールが IdM 設定に存在しないことを確認する方法を説明します。以下の例では、**ユーザーが自分の名前の詳細** のセルフサービスルールが IdM に存在しないことを確認する方法を説明します。これにより、ユーザーは自分の表示名や初期などを変更できないようにします。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/selfservice/** ディレクトリーにある **selfservice-absent.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**selfservice-absent-copy.yml**) を開きます。
4. **ipaselfservice** タスクセクションに以下の変数を設定してファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、セルフサービスルールの名前に設定します。
   - **state** 変数は **absent** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- IdM でのセルフサービスアクセス制御 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-selfservice.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/selfservice** ディレクトリーのサンプルの Playbook を参照してください。

10.4. ANSIBLE を使用してセルフサービスルールに固有の属性を含める手順

以下の手順では、Ansible Playbook を使用して、既存のセルフサービスルールに特定の設定を追加する方法を説明します。この例では、**ユーザーは自分の名前詳細を管理できる** のセルフサービスルールに、**surname** のメンバー属性が含まれるようにします。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。

- Ansible バージョン 2.14 以降を使用している。
- Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **ユーザーが独自の名前の詳細** セルフサービスルールが IdM に存在する。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/selfservice/** ディレクトリーにある **selfservice-member-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**selfservice-member-present-copy.yml**) を開きます。
4. **ipaselfservice** タスクセクションに以下の変数を設定してファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、変更するセルフサービスルールの名前に設定します。
   - **attribte** 変数は **surname** に設定します。
   - **action** 変数は **member** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- IdM でのセルフサービスアクセス制御 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーで利用可能な **README-selfservice.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/selfservice** ディレクトリーのサンプルの Playbook を参照してください。

10.5. ANSIBLE を使用してセルフサービスルールに固有の属性を含めないようにする手順

以下の手順では、Ansible Playbook を使用して、セルフサービスルールに特定の設定が割り当てられないようにする方法を説明します。この Playbook を使用して、セルフサービスルールで不必要なアクセス権限を付与しないようにします。この例では、**ユーザーが独自の名前の詳細を管理できる** セルフサービスルールに **givenname** と **surname** のメンバー属性が含まれないようにします。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **ユーザーが独自の名前の詳細** セルフサービスルールが IdM に存在する。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/selfservice/** ディレクトリーにある **selfservice-member-absent.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**selfservice-member-absent-copy.yml**) を開きます。

4. **ipaselfservice** タスクセクションに以下の変数を設定してファイルを調整します。

   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、変更するセルフサービスルールの名前に設定します。
   - **属性** 変数は **givenname** および **surname** に設定します。
   - **action** 変数は **member** に設定します。
   - **state** 変数は **absent** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- IdM でのセルフサービスアクセス制御 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-selfservice.md** ファイルを参照してください。

- **/usr/share/doc/ansible-freeipa/playbooks/selfservice** ディレクトリーのサンプルの Playbook を参照してください。

第11章 ANSIBLE PLAYBOOK を使用してユーザーグループにパーミッションを委譲してユーザーを管理する手順

委譲は、セルフサービスルールおよびロールベースのアクセス制御 (RBAC) などの IdM のアクセス制御メソッドの 1 つです。委譲を使用して、あるユーザーのグループにパーミッションを割り当てて別のユーザーのグループのエントリーを管理できます。

このセクションでは、以下のトピックについて説明します。

- 委譲ルール
- IdM の Ansible インベントリーファイルの作成
- Ansible を使用して委譲ルールを存在させる手順
- Ansible を使用して委譲ルールがないことを確認する手順
- Ansible を使用して委譲ルールに特定の属性を含める手順
- Ansible を使用して委譲ルールに特定の属性を含めないようにする手順

11.1. 委譲ルール

委譲ルール を作成して、ユーザーグループにパーミッションを委譲してユーザーを管理できます。

委譲ルールを使用すると、特定のユーザーグループが、別のユーザーグループ内のユーザーの特定の属性に対して書き込み (編集) 操作を実行できます。このようなアクセス制御ルールは、委譲ルールで指定された属性のサブセットの編集に限定されており、エントリー全体の追加や削除、未指定の属性の制御はできません。

委譲ルールにより、IdM の既存のユーザーグループにパーミッションが付与されます。委任を使用すると、**managers** ユーザーグループで **employees** ユーザーグループでユーザーの選択された属性を管理できます。

11.2. IDM の ANSIBLE インベントリーファイルの作成

Ansible を使用する場合は、ホームディレクトリーに Ansible Playbook 専用のサブディレクトリーを作成して、**/usr/share/doc/ansible-freeipa/*** と **/usr/share/doc/rhel-system-roles/*** サブディレクトリーからコピーして調整できるようにします。この方法には、以下の利点があります。

- すべての Playbook を 1 カ所で見つけることがでる。
- **root** 権限を呼び出さずに Playbook を実行できる。

手順

1. Ansible 設定および Playbook のディレクトリーをホームディレクトリーに作成します。

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

2. **~/MyPlaybooks/** ディレクトリーに移動します。

```
$ cd ~/MyPlaybooks
```

3. **~/MyPlaybooks/ansible.cfg** ファイルを以下の内容で作成します。

```
[defaults]
inventory = /home/<username>/MyPlaybooks/inventory

[privilege_escalation]
become=True
```

4. **~/MyPlaybooks/inventory** ファイルを以下の内容で作成します。

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

[us]
replica.idm.example.com

[ipaserver:children]
eu
us
```

この設定は、これらの場所にあるホストの 2 つのホストグループ (**eu** と **us**) を定義します。さらに、この設定は、**eu** および **us** グループのすべてのホストを含む **ipaserver** ホストグループを定義します。

11.3. ANSIBLE を使用して委譲ルールを存在させる手順

以下の手順では、Ansible Playbook を使用して、新しい IdM 委譲ルールの特権を定義して、その存在を確認する方法を説明します。この例では、新しい **basic manager attributes** 委譲ルールにより、**managers** グループが **employees** グループのメンバーに対して以下の属性の読み取りと書き込みを行うことができます。

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

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/delegation/** にある **delegation-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル **delegation-present-copy.yml** を開きます。

4. **ipadelegation** タスクセクションに以下の変数を設定して、ファイルを調整します。

   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は新しい委譲ルールの名前に設定します。
   - **permission** 変数は、付与するパーミッションをコンマ区切りのリスト (**read** および **write**) で設定します。
   - **attribute** 変数は、委譲されたユーザーグループが管理できる属性のリスト (**businesscategory**、**departmentnumber**、**employeenumber** および **employeetype**) に変数を設定します。
   - **group** 変数は、属性の表示や変更権限を付与したグループの名前に設定します。
   - **membergroup** 変数は、属性の表示または変更が可能なグループ名に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- 委譲ルール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-delegation.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation** ディレクトリーのサンプルの Playbook を参照してください。

11.4. ANSIBLE を使用して委譲ルールがないことを確認する手順

以下の手順では、Ansible Playbook を使用して、指定した委譲ルールが IdM 設定に存在しないことを確認する方法を説明します。以下の例では、カスタムの **basic manager attributes** 委譲ルールが IdM に存在しないことを確認する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/delegation/** ディレクトリーにある **delegation-absent.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル **delegation-absent-copy.yml** を開きます。
4. **ipadelegation** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は委譲ルールの名前に設定します。
   - **state** 変数は **absent** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- 委譲ルール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-delegation.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation** ディレクトリーのサンプルの Playbook を参照してください。

11.5. ANSIBLE を使用して委譲ルールに特定の属性を含める手順

以下の手順では、Ansible Playbook を使用して、委譲ルールに特定の設定を指定する方法を説明します。この Playbook を使用して、以前に作成した委譲ロールを変更できます。この例では、**basic manager attributes** 委譲ルールに **departmentnumber** メンバー属性のみが含まれるようにします。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。

- Ansible バージョン 2.14 以降を使用している。
- Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **basic manager attributes** 委譲ルールが IdM に存在する。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/delegation/** にある **delegation-member-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル **delegation-member-present-copy.yml** を開きます。
4. **ipadelegation** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、変更する委譲ルールの名前に設定します。
   - **attribute** 変数は **departmentnumber** に設定します。
   - **action** 変数は **member** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- 委譲ルール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-delegation.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation** ディレクトリーのサンプルの Playbook を参照してください。

11.6. ANSIBLE を使用して委譲ルールに特定の属性を含めないようにする手順

以下の手順では、Ansible Playbook を使用して、委譲ルールに特定の設定が割り当てられないようにする方法を説明します。この Playbook を使用して、委譲ロールが不必要なアクセス権限を付与しないようします。この例では、**basic manager attributes** 委譲ルールに **employeenumber** および **employeetype** メンバー属性が含まれないようにします。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **basic manager attributes** 委譲ルールが IdM に存在する。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/delegation/** にある **delegation-member-absent.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル **delegation-member-absent-copy.yml** を開きます。

4. **ipadelegation** タスクセクションに以下の変数を設定して、ファイルを調整します。

   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、変更する委譲ルールの名前に設定します。
   - **attribute** 変数は **employeenumber** および **employeetype** に設定します。
   - **action** 変数は **member** に設定します。
   - **state** 変数は **absent** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- 委譲ルール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-delegation.md** ファイルを参照してください。

- **/usr/share/doc/ansible-freeipa/playbooks/ipadelegation** ディレクトリーのサンプルの Playbook を参照してください。

第12章 ANSIBLE PLAYBOOK を使用した IDM でのロールベースアクセス制御の管理

ロールベースアクセス制御 (RBAC) は、ロールおよび権限関連を定義する、ポリシーに依存しないアクセス制御メカニズムです。Identity Management (IdM) の RBAC のコンポーネントは、ロール、権限、パーミッションです。

- **パーミッション** は、ユーザーの追加または削除、グループの変更、読み取りアクセスの有効化など、特定のタスクを実行する権限を付与します。
- **特権** は、新規ユーザーの追加に必要な全権限など、権限を組み合わせます。
- **ロール** は、ユーザー、ユーザーグループ、ホスト、またはホストグループに特権のセットを付与します。

特に大企業では、RBAC を使用すると、責任の領域を個別に設定する階層管理システムを作成できます。

本章では、Ansible Playbook を使用した RBAC の管理時に行う以下の操作について説明します。

- IdM のパーミッション
- デフォルトの管理パーミッション
- IdM の特権
- IdM のロール
- IdM の事前定義されたロール
- Ansible を使用して特権のある IdM RBAC ロールを存在させる手順
- Ansible を使用して IdM RBAC ロールを設定しないようにする手順
- Ansible を使用して、ユーザーグループに IdM RBAC ロールを割り当てる手順
- Ansible を使用して特定のユーザーに IdM RBAC ロールが割り当てられないようにする手順
- Ansible を使用してサービスを IdM RBAC ロールに所属させるように設定する手順
- Ansible を使用してホストを IdM RBAC ロールに所属させるように設定する手順
- Ansible を使用してホストグループを IdM RBAC ロールに所属させるように設定する手順

12.1. IDM のパーミッション

パーミッションは、ロールベースのアクセス制御の中で最も低いレベルの単位で、操作を適用する LDAP エントリーと合わせて操作を定義します。ブロックの構築と同様に、パーミッションは必要に応じて多くの権限に割り当てることができます。
1 つ以上の **権限** を使用して、許容される操作を定義します。

- **write**
- **read**
- **search**

- **compare**
- **add**
- **delete**
- **all**

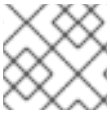
上記の操作は、3 つの基本的な **ターゲット** に適用されます。

- **subtree**: ドメイン名 (DN) (この DN のサブツリー)
- **target filter**: LDAP フィルター
- **target**: DN。ワイルドカードでエントリーを指定可能。

また、以下の便利なオプションは、対応する属性を設定します。

- **type**: オブジェクトのタイプ (ユーザー、グループなど) (**subtree** および **target filter** を設定します)。
- **memberof**: グループのメンバー。**target filter** を設定します。
- **targetgroup**: 特定のグループを変更する権限 (グループメンバーシップの管理権限の付与など) を付与します (**target** を設定します)。

IdM パーミッションを使用すると、どのユーザーがどのオブジェクトにアクセスできるか、さらにこのようなオブジェクトの属性にアクセスできるかどうかを制御できます。IdM を使用すると、個別の属性を許可または拒否したり、ユーザー、グループ、sudo などの特定の IdM 機能を、全匿名ユーザー、全認証済みユーザー、または特定の特権ユーザーグループ限定などと、全体的な表示設定を変更したりできます。
たとえば、このアプローチではパーミッション指定に柔軟性があるので、アクセスが必要な特定のセクションのみにユーザーまたはグループのアクセスを制限し、他のセクションをこれらのユーザーまたはグループには完全に表示されないように設定する場合に、管理者にとって便利です。



注記

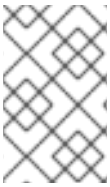
パーミッションには他のパーミッションを含めることはできません。

12.2. デフォルトの管理パーミッション

管理パーミッションは、IdM にデフォルトで含まれているパーミッションです。このパーミッションはユーザーが作成した他のパーミッションと同様に機能しますが、以下の相違点があります。

- この管理パーミッションは削除できず、名前、場所、ターゲットの属性を変更できません。
- このパーミッションには 3 つの属性セットがあります。
  - **デフォルト** の属性。IdM で管理されているため、ユーザーは変更できません。
  - **包含** 属性。ユーザーが別途追加する属性。
  - **除外** 属性。ユーザーが削除する属性。

管理パーミッションは、デフォルトおよび包含属性セットに表示されている属性すべてに適用されますが、除外セットに表示されている属性には適用されません。

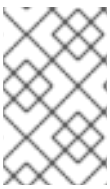


注記

管理パーミッションを削除できませんが、パーミッションにバインドタイプを設定し、すべての特権から管理パーミッションを削除して管理パーミッションを効果的に無効にできます。

管理パーミッションの名前はすべて **System:** から始まります (例: **System: Add Sudo rule** または **System: Modify Services**)。以前のバージョンの IdM では、デフォルトのパーミッションに異なるスキームを使用していました。たとえば、ユーザーはパーミッションの削除はできず、特権に割り当てるしかできませんでした。これらのデフォルトパーミッションのほとんどは、管理パーミッションに切り替わっていますが、以下のパーミッションは引き続き以前のスキームを使用します。

- Automember Rebuild メンバーシップタスクの追加
- 設定サブエントリーの追加
- レプリカ合意の追加
- 証明書削除保留
- CA から証明書のステータス取得
- DNA 範囲の読み取り
- DNA 範囲の変更
- PassSync Manager の設定の読み取り
- PassSync Manager 設定の変更
- レプリカ合意の読み込み
- レプリカ合意の修正
- レプリカ合意の削除
- LDBM データベース設定の読み取り
- 証明書の要求
- CA ACL を無視する証明書の要求
- 別のホストからの証明書の要求
- CA からの証明書の取得
- 証明書の取り消し
- IPA 設定の書き込み



注記

コマンドラインから管理パーミッションを変更しようとし、変更不可な属性の変更をシステム側が許可しない場合には、コマンドに失敗します。Web UI から管理パーミッションを変更しようとした場合には、変更できない属性が無効になります。

12.3. IDM の特権

特権は、ロールに適用されるパーミッションのグループです。
パーミッションは単一の操作を実行する権限を提供しますが、IdM タスクを成功させるには、複数のパーミッションが必要なものがあります。したがって、特権は、特定のタスクを実行するために必要な異なるパーミッションを組み合わせたものです。
たとえば、新しい IdM ユーザーにアカウントを設定するには、以下の権限が必要です。

- 新規ユーザーエントリーの作成
- ユーザーパスワードのリセット
- 新規ユーザーのデフォルト IPA ユーザーグループへの追加

これらの 3 つの低レベルのタスクを、**ユーザーの追加** という名前のカスタム特権の形式で、権限がより高いレベルのタスクに組み合わせることで、システム管理者はロールを管理しやすくなります。IdM には、すでにいくつかのデフォルト特権が含まれています。ユーザーとユーザーグループとは別に、権限はホストおよびホストグループ、およびネットワークサービスにも割り当てられます。これにより、特定のネットワークサービスを使用するホストセットのユーザーセットによって、操作をきめ細かく制御できます。

注記

特権には、他の特権を含めることはできません。

12.4. IDM のロール

ロールは、ロールに指定したユーザーが所有する権限のリストです。
実際には、パーミッションでは、指定の低階層のタスク (ユーザーエントリーの作成、グループへのエントリーの追加など) を実行する権限を付与し、特権では、高階層のタスク (指定のグループへの新規ユーザーの作成など) に必要なこれらのパーミッションの 1 つ以上を組み合わせます。ロールは必要に応じて、管理者ロールでユーザーの追加、変更、削除ができるなど、特権をまとめます。

重要

ロールは、許可されたアクションを分類するために使用されます。ロールは、特権昇格されないようにしたり、特権の分離を実装するツールとしては使用しません。

注記

ロールに他のロールを含めることはできません。

12.5. IDENTITY MANAGEMENT で事前定義されたロール

Red Hat Identity Management には、以下の事前定義済みのロールが含まれています。

表12.1 Identity Management の定義済みロール

ロール	特権	説明
登録管理者	ホストの登録	クライアントまたはホストの登録を行います。

ロール	特権	説明
helpdesk	Modify Users、Reset passwords、Modify Group membership	簡単なユーザー管理タスクを実行します。
IT Security Specialist	Netgroups Administrators、HBAC Administrator、Sudo Administrator	ホストベースのアクセス制御、sudo ルールなどのセキュリティーポリシーを管理します。
IT Specialist	Host Administrators、Host Group Administrators、Service Administrators、Automount Administrators	ホストの管理を行います
Security Architect	Delegation Administrator、Replication Administrators、Write IPA Configuration、Password Policy Administrator	Identity Management 環境の管理、信頼の作成、レプリカ合意を作成します。
User Administrator	User Administrators、Group Administrators、Stage User Administrators	ユーザーおよびグループの作成を行います

12.6. ANSIBLE を使用して特権のある IDM RBAC ロールを存在させる手順

デフォルトのロール以外で、ロールベースのアクセス制御 (RBAC) を使用して Identity Management (IdM) のリソースを詳細にわたり制御するには、カスタムロールを作成します。

以下の手順では、Ansible Playbook を使用して、新しい IdM カスタムロールの特権を定義し、その存在を確認する方法を説明します。この例では、新しい **user_and_host_administrator** ロールには、デフォルトで IdM で存在する以下の特権を一意に組み合わせたものが含まれます。

- **Group Administrators**
- **User Administrators**
- **Stage User Administrators**
- **Group Administrators**

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。

- ~/MyPlaybooks/ ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/<MyPlaybooks>/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/role/** にある **role-member-user-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**role-member-user-present-copy.yml**) を開きます。
4. **iparole** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は新規ロールの名前に設定します。
   - **特権** リストは、新しいロールに追加する IdM 権限の名前に設定します。
   - 必要に応じて、**user** 変数は、新規ロールを付与するユーザーの名前に設定します。
   - 必要に応じて、**group** 変数は、新規ロールを付与するグループの名前に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- Ansible Vault を使用したコンテンツの暗号化 を参照してください。
- IdM のロール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-role** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/iparole** ディレクトリーのサンプルの Playbook を参照してください。

12.7. ANSIBLE を使用して IDM RBAC ロールを設定しないようにする手順

Identity Management (IdM) のロールベースアクセス制御 (RBAC) を管理するシステム管理者は、誤って管理者がユーザーに割り当てることがないように、使用しなくなったロールが削除されていることを確認する必要があります。

以下の手順では、Ansible Playbook を使用してロールが削除されていることを確認する方法を説明します。以下の例では、カスタムの **user_and_host_administrator** ロールが IdM に存在しないことを確認する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/<MyPlaybooks>/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/role/** にある **role-is-absent.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**role-is-absent-copy.yml**) を開きます。

4. **iparole** タスクセクションに以下の変数を設定して、ファイルを調整します。

   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、ロールの名前に設定します。
   - **state** 変数は、**absent** に設定されていることを確認します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- Ansible Vault を使用したコンテンツの暗号化 を参照してください。
- IdM のロール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-role** Markdown ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/iparole** ディレクトリーのサンプルの Playbook を参照してください。

12.8. ANSIBLE を使用して、ユーザーグループに IDM RBAC ロールを割り当てる手順

Identity Management (IdM) のロールベースアクセス制御 (RBAC) を管理するシステム管理者は、junior administrators など、特定のユーザーグループにロールを割り当てます。

以下の例では、Ansible Playbook を使用して、同梱の IdM RBAC **helpdesk** ロールを **junior_sysadmins** に割り当てる方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/<MyPlaybooks>/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/role/** にある **role-member-group-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**role-member-group-present-copy.yml**) を開きます。
4. **iparole** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、割り当てるロールの名前に設定します。
   - **group** 変数はグループ名に設定します。
   - **action** 変数は **member** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- Ansible Vault を使用したコンテンツの暗号化 を参照してください。
- IdM のロール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-role** Markdown ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/iparole** ディレクトリーのサンプルの Playbook を参照してください。

12.9. ANSIBLE を使用して特定のユーザーに IDM RBAC ロールが割り当てられないようにする手順

Identity Management (IdM) のロールベースアクセス制御 (RBAC) を管理するシステム管理者は、会社内で別の役職に異動した後など、特定のユーザーに RBAC ロールが割り当てられないようにすることもできます。

以下の手順では、Ansible Playbook を使用して、 **user_01** および **user_02** という名前のユーザーが **helpdesk** ロールに割り当てられないようにする方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/<MyPlaybooks>/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/role/** にある **role-member-user-absent.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**role-member-user-absent-copy.yml**) を開きます。

4. **iparole** タスクセクションに以下の変数を設定して、ファイルを調整します。

   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、割り当てるロールの名前に設定します。
   - **user** リストをユーザーの名前に設定します。
   - **action** 変数は **member** に設定します。
   - **state** 変数は **absent** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ ansible-playbook --vault-password-file=password_file -v -i
~/<MyPlaybooks>/inventory role-member-user-absent-copy.yml
```

関連情報

- Ansible Vault を使用したコンテンツの暗号化 を参照してください。
- IdM のロール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-role** Markdown ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/iparole** ディレクトリーのサンプルの Playbook を参照してください。

12.10. ANSIBLE を使用してサービスを IDM RBAC ロールに所属させるように設定する手順

Identity Management (IdM) のロールベースアクセス制御 (RBAC) を管理するシステム管理者は、IdM に登録されている特定のサービスが、特定のロールのメンバーになっていることを確認する必要がある場合があります。以下の例では、カスタムの **web_administrator** ロールを使用して **client01.idm.example.com** サーバー上で実行中の **HTTP** サービスを管理できるようにする方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **web_administrator** ロールが IdM に存在する。
- **HTTP/client01.idm.example.com@IDM.EXAMPLE.COM** サービスが IdM に存在する。

手順

1. **~/<MyPlaybooks>/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/role/** にある **role-member-service-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**role-member-service-present-copy.yml**) を開きます。

4. **iparole** タスクセクションに以下の変数を設定して、ファイルを調整します。

   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、割り当てるロールの名前に設定します。
   - **service** リストはサービス名に設定します。
   - **action** 変数は **member** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- Ansible Vault を使用したコンテンツの暗号化 を参照してください。
- IdM のロール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-role** Markdown ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/iparole** ディレクトリーのサンプルの Playbook を参照してください。

12.11. ANSIBLE を使用してホストを IDM RBAC ロールに所属させるように設定する手順

Identity Management (IdM) でロールベースアクセス制御を管理するシステム管理者は、特定のホストまたはホストグループが特定のロールに関連付けられていることを確認する必要がある場合があります。以下の例では、カスタムの **web_administrator** ロールが **HTTP** サービスを実行している **client01.idm.example.com** の IdM ホストを管理できるようにする方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **web_administrator** ロールが IdM に存在する。
- **client01.idm.example.com** ホストが IdM に存在する。

手順

1. **~/<MyPlaybooks>/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/role/** にある **role-member-host-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**role-member-host-present-copy.yml**) を開きます。
4. **iparole** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、割り当てるロールの名前に設定します。
   - **host** リストをホストの名前に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- Ansible Vault を使用したコンテンツの暗号化 を参照してください。
- IdM のロール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-role** Markdown ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/iparole** ディレクトリーのサンプルの Playbook を参照してください。

12.12. ANSIBLE を使用してホストグループを IDM RBAC ロールに所属させるように設定する手順

Identity Management (IdM) でロールベースアクセス制御を管理するシステム管理者は、特定のホストまたはホストグループが特定のロールに関連付けられていることを確認する必要がある場合があります。以下の例では、カスタムの **web_administrator** ロールが **HTTP** サービスを実行している IdM ホストの **web_servers** グループを管理できるようにする方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。

- ~/MyPlaybooks/ ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **web_administrator** ロールが IdM に存在する。
- **web_servers** ホストグループが IdM に存在する。

手順

1. **~/<MyPlaybooks>/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/role/** にある **role-member-hostgroup-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**role-member-hostgroup-present-copy.yml**) を開きます。
4. **iparole** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、割り当てるロールの名前に設定します。
   - **hostgroup** リストはホストグループ名に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- name: Playbook to manage IPA role with members.
  hosts: ipaserver
  become: true
  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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- Ansible Vault を使用したコンテンツの暗号化 を参照してください。
- IdM のロール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-role** Markdown ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/iparole** ディレクトリーのサンプルの Playbook を参照してください。

第13章 ANSIBLE PLAYBOOK を使用した RBAC 権限の管理

ロールベースアクセス制御 (RBAC) は、ロール、権限およびパーミッションで定義する、ポリシーに依存しないアクセス制御メカニズムです。特に大企業では、RBAC を使用すると、責任の領域を個別に設定する階層管理システムを作成できます。

本章では、Ansible Playbook を使用して Identity Management (IdM) で RBAC 権限を管理する以下の操作について説明します。

- Ansible を使用してカスタムの RBAC 特権を存在させる手順
- Ansible を使用してカスタムの IdM RBAC 特権にメンバーパーミッションを存在させる手順
- Ansible を使用して IdM RBAC 特権にパーミッションが含まれないようにする手順
- Ansible を使用してカスタムの IdM RBAC 権限の名前を変更する手順
- Ansible を使用して IdM RBAC 特権が含まれないようにする手順

前提条件

- RBAC の概念と原則 を理解している。

13.1. ANSIBLE を使用してカスタムの IDM RBAC 特権を存在させる手順

Identity Management (IdM) のロールベースアクセス制御 (RBAC) でカスタム権限を完全に機能させるには、ステージごとに進めていく必要があります。

1. パーミッションが割り当てられていない特権を作成します。
2. 選択したパーミッションを特権に追加します。

以下の手順では、後でパーミッションを追加できるように、Ansible Playbook を使用して空の特権を作成する方法を説明します。この例では、ホスト管理に関連するすべての IdM パーミッションを組み合わせられるように **full_host_administration** という名前の特権を作成する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/privilege/** にある **privilege-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**privilege-present-copy.yml**) を開きます。

4. **ipaprivilege** タスクセクションに以下の変数を設定してファイルを調整します。

   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、新しい特権 **full_host_administration** の名前に設定します。
   - 必要に応じて、**description** 変数を使用して特権を記述します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

13.2. ANSIBLE を使用してカスタムの IDM RBAC 特権にメンバーパーミッションを存在させる手順

Identity Management (IdM) のロールベースアクセス制御 (RBAC) でカスタム権限を完全に機能させるには、ステージごとに進めていく必要があります。

1. パーミッションが割り当てられていない特権を作成します。

2. 選択したパーミッションを特権に追加します。

以下の手順では、Ansible Playbook を使用して、前の手順で作成した特権にパーミッションを追加する方法を説明します。この例では、ホスト管理に関連する IdM パーミッションをすべて、**full_host_administration** という名前の特権に追加する方法を説明します。デフォルトでは、パーミッションは **Host Enrollment**、**Host Administrators** および **Host Group Administrator** 特権の間で分散されます。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **full_host_administration** 権限が存在する。Ansible を使用して特権を作成する方法の詳細は、Ansible を使用してカスタムの IdM RBAC 特権を存在させる手順 を参照してください。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/privilege/** にある **privilege-member-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**privilege-member-present-copy.yml**) を開きます。
4. **ipaprivilege** タスクセクションに以下の変数を設定してファイルを調整します。
   - 使用しているユースケースに合わせて、タスクの **名前** を調節します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、特権の名前に設定します。
   - **permission** は、特権に追加するパーミッションの名前を設定します。
   - **action** 変数が **member** に設定されていることを確認します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

13.3. ANSIBLE を使用して IDM RBAC 特権にパーミッションが含まれないようにする手順

Identity Management (IdM) のシステム管理者は、IdM のロールベースアクセス制御をカスタマイズできます。

以下の手順では、Ansible Playbook を使用して、特権からパーミッションを削除する方法を説明します。この例では、管理者がセキュリティー上のリスクを考慮するため、デフォルトの **Certificate Administrators** 特権から **Request Certificates ignoring CA ACLs** を削除する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/privilege/** にある **privilege-member-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**privilege-member-absent-copy.yml**) を開きます。
4. **ipaprivilege** タスクセクションに以下の変数を設定してファイルを調整します。
   - 使用しているユースケースに合わせて、タスクの **名前** を調節します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、特権の名前に設定します。
   - **permission** のリストは、特権から削除するパーミッションの名前に設定します。
   - **action** 変数が **member** に設定されていることを確認します。
   - **state** 変数は **absent** に設定されていることを確認します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

13.4. ANSIBLE を使用してカスタムの IDM RBAC 権限の名前を変更する手順

Identity Management (IdM) のシステム管理者は、IdM のロールベースアクセス制御をカスタマイズできます。

以下の手順では、たとえば、パーミッションの一部を削除したなどの理由から、特権の名前を変更する方法を説明します。パーミッションを削除した結果、特権の名前は正確ではなくなりました。この例では、管理者は **full_host_administration** 特権の名前を **limited_host_administration** に変更します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **full_host_administration** 権限が存在する。特権の追加方法は、Ansible を使用してカスタムの IdM RBAC 特権を存在させる手順 を参照してください。

手順

1. ~/ MyPlaybooks/ ディレクトリーに移動します。

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

2. /usr/share/doc/ansible-freeipa/playbooks/privilege/ にある **privilege-present.yml** ファイルのコピーを作成します。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/privilege/privilege-present.yml rename-privilege.yml
```

3. Ansible Playbook ファイル (**rename-privilege.yml**) を開いて編集します。

4. **ipaprivilege** タスクセクションに以下の変数を設定してファイルを調整します。

- **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
- **name** 変数は、現在の特権名に設定します。
- **rename** 変数を追加して、特権の新しい名前に設定します。
- **state** 変数を追加し、 **renamed** に設定します。

5. 以下のように、Playbook 自体の名前を変更します。

```
---
- name: Rename a privilege
  hosts: ipaserver
```

6. 以下のように、Playbook のタスクの名前を変更します。

```
[...]
tasks:
- name: Ensure the full_host_administration privilege is renamed to limited_host_administration
  ipaprivilege:
  [...]
```

以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

8. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory rename-privilege.yml
```

13.5. ANSIBLE を使用して IDM RBAC 特権が含まれないようにする手順

Identity Management (IdM) のシステム管理者は、IdM のロールベースアクセス制御をカスタマイズできます。以下の手順では、Ansible Playbook を使用して RBAC 特権が削除されていることを確認する方法を説明します。この例では、**CA administrator** 特権が存在しないことを確認する方法を説明します。この手順が終わると、IdM で認証局を管理できるユーザーは **admin** だけになります。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/privilege/** ディレクトリーにある **privilege-absent.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**privilege-absent-copy.yml**) を開きます。
4. **ipaprivilege** タスクセクションに以下の変数を設定してファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、削除する権限の名前に設定します。
   - **state** 変数が **absent** に設定されていることを確認します。
5. 以下のように、Playbook のタスクの名前を変更します。

```
[...]
tasks:
- name: Ensure privilege "CA administrator" is absent
  ipaprivilege:
  [...]
```

以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

7. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

13.6. 関連情報

- IdM の特権 を参照してください。
- IdM のパーミッション を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーで利用可能な **README-privilege** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/ipaprivilege** ディレクトリーのサンプルの Playbook を参照してください。

第14章 ANSIBLE PLAYBOOK を使用した IDM での RBAC パーミッションの管理

ロールベースアクセス制御 (RBAC) は、ロール、権限およびパーミッションで定義する、ポリシーに依存しないアクセス制御メカニズムです。特に大企業では、RBAC を使用すると、責任の領域を個別に設定する階層管理システムを作成できます。

本章では、Ansible Playbook を使用して Identity Management (IdM) で RBAC パーミッションの管理時に行う、以下の操作について説明します。

- Ansible を使用して RBAC パーミッションを存在させる手順
- Ansible を使用して属性を含めて RBAC パーミッションを追加する手順
- Ansible を使用して RBAC パーミッションをバインドする手順
- Ansible を使用して属性を IdM RBAC パーミッションをメンバーにする手順
- Ansible を使用して属性が IdM RBAC パーミッションのメンバーに含まれないようにする手順
- Ansible を使用して IdM RBAC パーミッションの名前を変更する手順

前提条件

- RBAC の概念と原則 を理解している。

14.1. ANSIBLE を使用して RBAC パーミッションを存在させる手順

Identity Management (IdM) のシステム管理者は、IdM のロールベースアクセス制御 (RBAC) をカスタマイズできます。

以下の手順では、Ansible Playbook を使用して、パーミッションを特権に追加できるように IdM にパーミッションを追加する方法を説明します。この例では、目的とする以下の状態を達成する方法を説明します。

- **MyPermission** パーミッションが存在する。
- **MyPermission** パーミッションだけがホストに適用できる。
- パーミッションを含む特権を付与されたユーザーは、エントリーに対して以下の操作すべてを実行できます。
  - Write
  - Read
  - Search
  - Compare
  - Add
  - Delete

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/permission/** ディレクトリーにある **permission-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**permission-present-copy.yml**) を開きます。
4. **ipapermission** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - 使用しているユースケースに合わせて、タスクの **名前** を調節します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数はパーミッションの名前に設定します。
   - **object_type** 変数は **host** に設定します。
   - **right** 変数は **all** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

14.2. ANSIBLE を使用して属性を含めて RBAC パーミッションを追加する手順

Identity Management (IdM) のシステム管理者は、IdM のロールベースアクセス制御 (RBAC) をカスタマイズできます。

以下の手順では、Ansible Playbook を使用して、パーミッションを特権に追加できるように IdM にパーミッションを追加する方法を説明します。この例では、目的とする以下の状態を達成する方法を説明します。

- **MyPermission** パーミッションが存在する。
- **MyPermission** パーミッションだけがホストの追加に使用できる。
- パーミッションを含む特権を付与されたユーザーは、ホストのエントリーに対して以下の操作すべてを実行できる。
  - Write
  - Read
  - Search
  - Compare
  - Add
  - Delete
- **MyPermission** パーミッションを含む特権のあるユーザーが作成したホストエントリーに **description** の値を追加できる。

注記

IdM LDAP スキーマでは、パーミッションの作成または変更時に指定できる属性のタイプは制約されません。ただし、**object_type** が **host** の場合に **attrs: car_licence** を指定すると、後でパーミッションを実行して、車のライセンス値をホストに追加使用とすると **ipa: ERROR: attribute "car-license" not allowed** というエラーメッセージが表示されます。

前提条件

- IdM 管理者パスワードを把握している。

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/permission/** ディレクトリーにある **permission-present.yml** ファイルのコピーを作成します。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-present.yml
permission-present-with-attribute.yml
```

3. Ansible Playbook ファイル (**permission-present-with-attribute.yml**) を開きます。
4. **ipapermission** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - 使用しているユースケースに合わせて、タスクの **名前** を調節します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数はパーミッションの名前に設定します。
   - **object_type** 変数は **host** に設定します。
   - **right** 変数は **all** に設定します。
   - **attrs** 変数は **description** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory permission-present-with-attribute.yml
```

関連情報

- RHEL 7 の **Linux Domain Identity, Authentication and Policy Guide**の User and group schema を参照してください。

14.3. ANSIBLE を使用して RBAC パーミッションをバインドする手順

Identity Management (IdM) のシステム管理者は、IdM のロールベースアクセス制御 (RBAC) をカスタマイズできます。

以下の手順では、Ansible Playbook を使用して、パーミッションを特権に追加できないように IdM からパーミッションを削除する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/permission/** ディレクトリーにある **permission-absent.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**permission-absent-copy.yml**) を開きます。
4. **ipapermission** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - 使用しているユースケースに合わせて、タスクの **名前** を調節します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数はパーミッションの名前に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

14.4. ANSIBLE を使用して属性を IDM RBAC パーミッションをメンバーにする手順

Identity Management (IdM) のシステム管理者は、IdM のロールベースアクセス制御 (RBAC) をカスタマイズできます。

以下の手順では、Ansible Playbook を使用して、属性が IdM の RBAC パーミッションのメンバーであることを確認する方法を説明します。この手順を行うと、パーミッションのあるユーザーは、属性のあるエントリーを作成できます。

この例では、**MyPermission** パーミッションを含む特権を持つユーザーにより作成されたホストエントリーに、**gecos** および **description** の値を設定する方法を説明します。

注記

IdM LDAP スキーマでは、パーミッションの作成または変更時に指定できる属性のタイプは制約されません。ただし、**object_type** が **host** の場合に **attrs: car_licence** を指定すると、後でパーミッションを実行して、車のライセンス値をホストに追加使用とすると **ipa: ERROR: attribute "car-license" not allowed** というエラーメッセージが表示されます。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **MyPermission** パーミッションが存在する。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/permission/** ディレクトリーにある **permission-member-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**permission-member-present-copy.yml**) を開きます。
4. **ipapermission** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - 使用しているユースケースに合わせて、タスクの **名前** を調節します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数はパーミッションの名前に設定します。
   - **attrs** リストを **description** および **gecos** 変数に設定します。
   - **action** 変数が **member** に設定されていることを確認します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

14.5. ANSIBLE を使用して属性が IDM RBAC パーミッションのメンバーに含まれないようにする手順

Identity Management (IdM) のシステム管理者は、IdM のロールベースアクセス制御 (RBAC) をカスタマイズできます。

以下の手順では、Ansible Playbook を使用して、属性が IdM の RBAC パーミッションに含まれないようにする方法を説明します。この手順を実行すると、パーミッションのあるユーザーが IdM LDAP にエントリーを作成すると、そのエントリーで、値に属性を関連付けて設定できません。

この例では、目的とする以下の状態を達成する方法を説明します。

- **MyPermission** パーミッションが存在する。
- **MyPermission** パーミッションを含む特権のあるユーザーが作成したホストエントリーに **description** 属性を使用できない。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。

- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **MyPermission** パーミッションが存在する。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/permission/** ディレクトリーにある **permission-member-absent.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**permission-member-absent-copy.yml**) を開きます。
4. **ipapermission** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - 使用しているユースケースに合わせて、タスクの **名前** を調節します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数はパーミッションの名前に設定します。
   - **attrs** 変数は **description** に設定します。
   - **action** 変数は **member** に設定します。
   - **state** 変数が **absent** に設定されていることを確認します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

14.6. ANSIBLE を使用して IDM RBAC パーミッションの名前を変更する手順

Identity Management (IdM) のシステム管理者は、IdM のロールベースアクセス制御をカスタマイズできます。

以下の手順では、Ansible Playbook を使用してパーミッションの名前を変更する方法を説明します。この例では、**MyPermission** の名前を **MyNewPermission** に変更する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **MyPermission** が IdM に存在する。
- **MyNewPermission** が IdM に存在しない。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/permission/** ディレクトリーにある **permission-renamed.yml** ファイルのコピーを作成します。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/permission/permission-renamed.yml permission-renamed-copy.yml
```

3. Ansible Playbook ファイル (**permission-renamed-copy.yml**) を開きます。
4. **ipapermission** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - 使用しているユースケースに合わせて、タスクの **名前** を調節します。

- **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
- **name** 変数はパーミッションの名前に設定します。

以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

14.7. 関連情報

- IdM のパーミッション を参照してください。
- IdM の特権 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーで利用可能な **README-permission** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/ipapermission** ディレクトリーのサンプルの Playbook を参照してください。

第15章 ANSIBLE を使用した IDM でのレプリケーショントポロジーの管理

複数の Identity Management (IdM) サーバーを維持し、冗長性の目的で相互に複製して、サーバーの損失を軽減または防止することができます。たとえば、1 台のサーバーに障害が発生しても、その他のサーバーがドメインにサービスを提供し続けます。障害が発生していないサーバーの 1 台から新しいレプリカを作成し、失われたサーバーを回復することもできます。

IdM サーバーに保存されているデータは、レプリカ合意に基づいて複製されます。2 台のサーバーでレプリカ合意が設定されている場合は、データを共有します。レプリケートされるデータはトポロジーの **suffix** に保存されます。2 つのレプリカに接尾辞間でレプリカ合意があると、接尾辞はトポロジー **segment** を形成します。

本章では、**Red Hat Ansible Engine** を使用して IdM レプリカ合意、トポロジーセグメント、およびトポロジー接尾辞を管理する方法を説明します。この章には次のセクションが含まれています。

- Ansible を使用して、レプリカ合意が IdM に存在することを確認
- Ansible を使用して複数の IdM レプリカ間でレプリカ合意を存在させる手順
- Ansible を使用して 2 つのレプリカ間でレプリカ合意が存在するかどうかの確認
- Ansible を使用してトポロジーの接尾辞が IdM に存在することを確認
- Ansible を使用した IdM レプリカの再初期化
- Ansible を使用して IdM にレプリカ合意がないことを確認する手順

15.1. ANSIBLE を使用して、レプリカ合意が IDM に存在することを確認

Identity Management (IdM) サーバーに保存されているデータは、レプリカ合意に基づいて複製されます。2 台のサーバーでレプリカ合意が設定されている場合は、データを共有します。レプリカ合意は常に双方向のものです。最初のレプリカからサーバーから別のレプリカにデータが複製されるだけでなく、別ののレプリカから最初のレプリカにもデータが複製されます。

この手順に従い、Ansible Playbook を使用して、**server.idm.example.com** と **replica.idm.example.com** との間で **domain** タイプのレプリカ合意が存在することを確認説明します。

前提条件

- トポロジーで IdM レプリカを接続するためのガイドライン に記載されている IdM トポロジーを設計するための推奨事項を確実に理解している。
- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible -freeipa/playbooks/topology/** ディレクトリーにある **add-topologysegment.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegment.yml
add-topologysegment-copy.yml
```

3. **add-topologysegment-copy.yml** ファイルを開いて編集します。

4. **ipatopologysegment** タスクセクションに以下の変数を設定して、ファイルを調整します。

   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - 追加するセグメントのタイプに応じて、**suffix** 変数を **domain** または **ca** のいずれかに設定します。
   - **left** の変数をレプリカ合意の左ノードに設定する IdM サーバーの名前に設定します。
   - レプリカ合意の適切なノードとなる IdM サーバーの名前に **right** 変数を設定します。
   - **state** 変数は **present** に設定されていることを確認します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- レプリカ合意、トポロジー接尾辞、およびトポロジーセグメントの説明 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-topology.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/topology** ディレクトリーのサンプルの Playbook を参照してください。

15.2. ANSIBLE を使用して複数の IDM レプリカ間でレプリカ合意を存在させる手順

Identity Management (IdM) サーバーに保存されているデータは、レプリカ合意に基づいて複製されます。2 台のサーバーでレプリカ合意が設定されている場合は、データを共有します。レプリカ合意は常に双方向のものです。最初のレプリカからサーバーから別のレプリカにデータが複製されるだけでなく、別ののレプリカから最初のレプリカにもデータが複製されます。

以下の手順に従って、IdM の複数のレプリカのペア間でレプリカ合意が存在することを確認します。

前提条件

- トポロジーで IdM レプリカを接続するためのガイドライン に記載されている IdM トポロジーを設計するための推奨事項を確実に理解している。
- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible -freeipa/playbooks/topology/** ディレクトリーにある **add-topologysegments.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/add-topologysegments.yml
add-topologysegments-copy.yml
```

3. **add-topologysegments-copy.yml** ファイルを開いて編集します。

4. **vars** セクションに以下の変数を設定して、ファイルを調整します。

   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - すべてのトポロジーセグメントについて、**ipatopology_segments** セクションに行を追加し、以下の変数を設定します。
     - 追加するセグメントのタイプに応じて、**suffix** 変数を **domain** または **ca** のいずれかに設定します。
     - **left** の変数をレプリカ合意の左ノードに設定する IdM サーバーの名前に設定します。
     - レプリカ合意の適切なノードとなる IdM サーバーの名前に **right** 変数を設定します。

5. **add-topologysegments-copy.yml** ファイルの **tasks** セクションで、**state** 変数が **present** に設定されていることを確認します。
   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

7. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- レプリカ合意、トポロジー接尾辞、およびトポロジーセグメントの説明 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-topology.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/topology** ディレクトリーのサンプルの Playbook を参照してください。

15.3. ANSIBLE を使用して 2 つのレプリカ間でレプリカ合意が存在するかどうかの確認

Identity Management (IdM) サーバーに保存されているデータは、レプリカ合意に基づいて複製されます。2 台のサーバーでレプリカ合意が設定されている場合は、データを共有します。レプリカ合意は常に双方向のものです。最初のレプリカからサーバーから別のレプリカにデータが複製されるだけでなく、別ののレプリカから最初のレプリカにもデータが複製されます。

以下の手順に従って、IdM のレプリカのペア間でレプリカ合意が存在することを確認します。

前提条件

- トポロジーで IdM レプリカを接続するためのガイドライン に記載されている Identity Management (IdM) トポロジーを設計するための推奨事項を確実に理解している。
- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible -freeipa/playbooks/topology/** ディレクトリーにある **check-topologysegments.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/check-topologysegments.yml
check-topologysegments-copy.yml
```

3. **check-topologysegments-copy.yml** ファイルを開いて編集します。

4. **vars** セクションに以下の変数を設定して、ファイルを調整します。

   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - すべてのトポロジーセグメントについて、**ipatopology_segments** セクションに行を追加し、以下の変数を設定します。
     - 追加するセグメントのタイプに応じて、**suffix** 変数を **domain** または **ca** のいずれかに設定します。
     - **left** の変数をレプリカ合意の左ノードに設定する IdM サーバーの名前に設定します。
     - レプリカ合意の適切なノードとなる IdM サーバーの名前に **right** 変数を設定します。

5. **check-topologysegments-copy.yml** ファイルの **tasks** セクションで、**state** 変数が **present** に設定されていることを確認します。
   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

7. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- トポロジー合意、接尾辞、およびセグメントの概念の詳細は、レプリカ合意、トポロジー接尾辞、およびトポロジーセグメント を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-topology.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/topology** ディレクトリーのサンプルの Playbook を参照してください。

15.4. ANSIBLE を使用してトポロジーの接尾辞が IDM に存在することを確認

Identity Management (IdM) のレプリカ合意のコンテキストでは、トポロジー接尾辞はレプリケートされるデータを保存します。IdM は、**domain** と **ca** の 2 種類のトポロジー接尾辞に対応します。それぞれの接尾辞は、個別のバックエンドである個別のレプリケーショントポロジーを表します。レプリカ合意が設定されると、同じタイプのトポロジー接尾辞を 2 つの異なるサーバーに結合します。

domain 接尾辞には、ユーザー、グループ、ポリシーなどのドメイン関連のデータがすべて含まれます。**ca** 接尾辞には、Certificate System コンポーネントのデータが含まれます。これは認証局 (CA) がインストールされているサーバーにのみ存在します。

以下の手順に従って、Ansible Playbook を使用して、トポロジー接尾辞が IdM に存在することを確認します。この例では、**domain** 接尾辞が IdM に存在することを確認する方法を説明します。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/topology/** ディレクトリーにある **verify-topologysuffix.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/ verify-topologysuffix.yml
verify-topologysuffix-copy.yml
```

3. Ansible Playbook ファイル **verify-topologysuffix-copy.yml** を開きます。
4. **ipatopologysuffix** セクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **suffix** 変数は **domain** に設定し ます。**ca** 接尾辞が存在することを確認する場合は、変数を **ca** に設定します。
   - **state** 変数が **verified** に設定されていることを確認します。他のオプションは使用できません。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

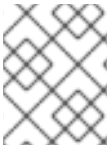
```
$ ansible-playbook --vault-password-file=password_file -v -i inventory verify-topologysuffix-copy.yml
```

関連情報

- レプリカ合意、トポロジー接尾辞、およびトポロジーセグメントの説明 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-topology.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/topology** ディレクトリーのサンプルの Playbook を参照してください。

15.5. ANSIBLE を使用した IDM レプリカの再初期化

レプリカが長期間オフラインである場合や、そのデータベースが破損している場合は、初期化できます。初期化により、更新リストのデータでレプリカが更新されます。たとえば、バックアップからの権威復元が必要な場合に使用できます。



注記

レプリケーションの更新とは対照的に、レプリカが変更エントリーのみを送信する間、データベース全体を再初期化します。

コマンドを実行するローカルホストは、再初期化されたレプリカです。データの取得元となるレプリカを指定するには、**direction** オプションを使用します。

以下の手順に従って、Ansible Playbook を使用して **server.idm.example.com** から **replica.idm.example.com** の **domain** データを再初期化します。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible -freeipa/playbooks/topology/** ディレクトリーにある **reinitialize-topologysegment.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/reinitialize-topologysegment.yml reinitialize-topologysegment-copy.yml
```

3. **reinitialize-topologysegment-copy.yml** ファイルを開いて編集します。
4. **ipatopologysegment** セクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **suffix** 変数は **domain** に設定し ます。**ca** データを再初期化する場合は、変数を **ca** に設定します。
   - **left** の変数をレプリカ合意の左ノードに設定します。
   - レプリカ合意の **right** なノードに正しい変数を設定します。
   - **direction** 変数は再初期化されるデータの方向に設定します。**left-to-right** は、左のノードから適切なノードにデータフローがあることを意味します。
   - **state** 変数が **reinitialized** に設定されていることを確認します。
     以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- レプリカ合意、トポロジー接尾辞、およびトポロジーセグメントの説明 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-topology.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/topology** ディレクトリーのサンプルの Playbook を参照してください。

15.6. ANSIBLE を使用して IDM にレプリカ合意がないことを確認する手順

Identity Management (IdM) サーバーに保存されているデータは、レプリカ合意に基づいて複製されます。2 台のサーバーでレプリカ合意が設定されている場合は、データを共有します。レプリカ合意は常に双方向のものです。最初のレプリカからサーバーから別のレプリカにデータが複製されるだけでなく、別ののレプリカから最初のレプリカにもデータが複製されます。

以下の手順に従って、2 つのレプリカ間のレプリカ合意が IdM に存在しないことを確認します。この例では、**domain** タイプのレプリカ合意が、**replica01.idm.example.com** と **replica02.idm.example.com** 間で存在させないようにする方法を説明します。

前提条件

- トポロジーで IdM レプリカを接続するためのガイドライン に記載されている IdM トポロジーを設計するための推奨事項を確実に理解している。
- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。

- Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible -freeipa/playbooks/topology/** ディレクトリーにある **delete-topologysegment.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/topology/delete-topologysegment.yml
delete-topologysegment-copy.yml
```

3. **delete-topologysegment-copy.yml** ファイルを開いて編集します。
4. **ipatopologysegment** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **suffix** 変数は **domain** に設定し ます。また、**ca** データが左右ノードと右のノード間で複製されないようにするには、変数を **ca** に設定します。
   - **left** の変数を、レプリカ合意の左ノードである IdM サーバーの名前に設定します。
   - **右側** の変数を、レプリカ合意の右のノードである IdM サーバーの名前に設定します。
   - **state** 変数は、**absent** に設定されていることを確認します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- レプリカ合意、トポロジー接尾辞、およびトポロジーセグメントの説明 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-topology.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/topology** ディレクトリーのサンプルの Playbook を参照してください。

15.7. 関連情報

- Planning the replica topology を参照してください。
- Installing an IdM replica を参照してください。

第16章 ANSIBLE を使用した IDM サーバーの管理

Red Hat Ansible Engine を使用すると、Identity Management (IdM) トポロジーのサーバーを管理できます。**ansible-freeipa** パッケージの **server** モジュールを使用して、IdM トポロジーにサーバーの有無を確認できます。任意のレプリカを非表示にしたり、レプリカを表示したりすることもできます。

このセクションには、以下のトピックが含まれます。

- Ansible を使用した IdM サーバーの存在の確認
- Ansible を使用した IdM トポロジーに IdM サーバーが存在しないことの確認
- 最後の IdM サーバーロールをホストしているにもかかわらず IdM サーバーがないことの確認
- IdM サーバーが存在しないが、必ずしも他の IdM サーバーから切断されていないことの確認
- Ansible Playbook を使用した既存の IdM サーバーが非表示であることの確認
- Ansible Playbook を使用した既存の IdM サーバーが表示されていることの確認
- 既存の IdM サーバーに IdM DNS の場所が割り当てられていることの確認
- 既存の IdM サーバーに IdM DNS の場所が割り当てられていないことの確認

16.1. ANSIBLE を使用した IDM サーバーの存在の確認

Ansible Playbook で **ipaserver ansible-freeipa** モジュールを使用して、Identity Management (IdM) サーバーが存在することを確認できます。

注記

ipaserver Ansible モジュールは、IdM サーバーをインストールしません。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
  - 制御ノードからインベントリーファイルに定義した IdM サーバーへの **SSH** 接続が正常に動作している。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/server/** ディレクトリーにある **server-present.yml** Ansible Playbook ファイルをコピーします。

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

3. **server-present-copy.yml** を開いて編集します。

4. **ipaserver** タスクセクションで次の変数を設定してファイルを調整し、ファイルを保存します。

- **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
- **name** 変数をサーバーの **FQDN** に設定します。サンプルサーバーの **FQDN** は server123.idm.example.com です。

```
---
- 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. Ansible Playbook を実行し、Playbook ファイルとインベントリーファイルを指定します。

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

関連情報

- Ansible Playbook を使用した Identity Management サーバーのインストール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-server.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/server** ディレクトリーのサンプルの Playbook を参照してください。

16.2. ANSIBLE を使用した IDM トポロジーに IDM サーバーが存在しないことの確認

Ansible Playbook を使用して、Identity Management (IdM) サーバーが、ホストとしても IdM トポロジーに存在しないようにします。

ansible-freeipa ipaserver ロールとは対照的に、この Playbook で使用する **ipaserver** モジュールは、サーバーから IdM サービスをアンインストールしません。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
  - 制御ノードからインベントリーファイルに定義した IdM サーバーへの **SSH** 接続が正常に動作している。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/server/** ディレクトリーにある **server-absent.yml** Ansible Playbook ファイルをコピーします。

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

3. **server-absent-copy.yml** を開いて編集します。
4. **ipaserver** タスクセクションで次の変数を設定してファイルを調整し、ファイルを保存します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **name** 変数をサーバーの **FQDN** に設定します。サンプルサーバーの **FQDN** は **server123.idm.example.com** です。
   - **state** 変数は、**absent** に設定されていることを確認します。

```
---
- 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. Ansible Playbook を実行し、Playbook ファイルとインベントリーファイルを指定します。

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

6. **server123.idm.example.com** を指定しているネームサーバー (NS) の DNS レコードがすべて DNS ゾーンから削除されていることを確認してください。使用する DNS が IdM により管理される統合 DNS であるか、外部 DNS であるかに関わらず、確認を行なってください。

関連情報

- IdM server のアンインストール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-server.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/server** ディレクトリーのサンプルの Playbook を参照してください。

16.3. 最後の IDM サーバーロールをホストしているにもかかわらず IDM サーバーがないことの確認

Ansible を使用すると、最後の IdM サービスインスタンスがサーバーで実行している場合でも、Identity Management (IdM) サーバーがないことを確認できます。認証局 (CA)、キーリカバリー認証局 (KRA)、または DNS サーバーはすべて IdM サービスの例です。

警告

CA サーバー、KRA サーバー、または DNS サーバーとして機能する最後のサーバーを削除すると、IdM 機能に深刻な不具合が生じます。**ipa service-find** を使用すると、どのサービスがどの IdM サーバーで実行されているかを手動で確認できます。認証局サーバーのプリンシパル名は **dogtag/server_name/REALM_NAME** です。

ansible-freeipa ipaserver ロールとは対照的に、この Playbook で使用する **ipaserver** モジュールは、サーバーから IdM サービスをアンインストールしません。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。

- Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
  - 制御ノードからインベントリーファイルに定義した IdM サーバーへの **SSH** 接続が正常に動作している。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/server/** ディレクトリーにある **server-absent-ignore-last-of-role.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-absent-ignore-last-of-role.yml server-absent-ignore-last-of-role-copy.yml
```

3. **server-absent-ignore-last-of-role-copy.yml** ファイルを開いて編集します。
4. **ipaserver** タスクセクションで次の変数を設定してファイルを調整し、ファイルを保存します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **name** 変数をサーバーの **FQDN** に設定します。サンプルサーバーの **FQDN** は **server123.idm.example.com** です。
   - **ignore_last_of_role** 変数が **true** に設定されていることを確認します。
   - **state** 変数は **absent** に設定します。

```
---
- 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: true
      state: absent
```

5. Ansible Playbook を実行し、Playbook ファイルとインベントリーファイルを指定します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-absent-ignore-last-of-role-copy.yml
```

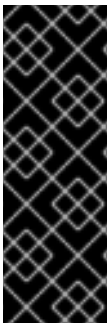
6. server123.idm.example.com を指定するネームサーバー (NS) の DNS レコードが、すべて DNS ゾーンから削除されていることを確認してください。使用する DNS が IdM により管理される統合 DNS であるか、外部 DNS であるかに関わらず、確認を行なってください。

関連情報

- IdM server のアンインストール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-server.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/server** ディレクトリーのサンプルの Playbook を参照してください。

16.4. IDM サーバーが存在しないが、必ずしも他の IDM サーバーから切断されていないことの確認

トポロジーから Identity Management (IdM) サーバーを削除する場合は、Ansible Playbook でレプリケーションアグリーメントをそのまま保持できます。Playbook では、IdM サーバーがホストとしても IdM に存在しないことも確認します。



重要

削除する際にサーバーのレプリカ合意を無視することが推奨されるのは、削除を予定している他のサーバーが機能不全のサーバーである場合のみです。トポロジーの中心点として機能するサーバーを削除すると、トポロジーが 2 つの切断されたクラスターに分割される可能性があります。

機能不全のサーバーは、**ipa server-del** コマンドを使用してトポロジーから削除できます。

注記

認証局 (CA)、キーリカバリー認証局 (KRA)、または DNS サーバーとして機能する最後のサーバーを削除すると、Identity Management (IdM) 機能に深刻な不具合が生じます。この問題を防ぐため、Playbook は、CA サーバー、KRA サーバー、または DNS サーバーとして機能するサーバーをアンインストールする前に、これらのサービスがドメインの別のサーバーで実行していることを確認します。

ansible-freeipa ipaserver ロールとは対照的に、この Playbook で使用する **ipaserver** モジュールは、サーバーから IdM サービスをアンインストールしません。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。

- ~/MyPlaybooks/ ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
  - 制御ノードからインベントリーファイルに定義した IdM サーバーへの **SSH** 接続が正常に動作している。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/server/** ディレクトリーにある **server-absent-ignore_topology_disconnect.yml** Ansible Playbook ファイルをコピーします。

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

3. **server-absent-ignore_topology_disconnect-copy.yml** を開いて編集します。
4. **ipaserver** タスクセクションで次の変数を設定してファイルを調整し、ファイルを保存します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **name** 変数をサーバーの **FQDN** に設定します。サンプルサーバーの **FQDN** は **server123.idm.example.com** です。
   - **ignore_topology_disconnect** 変数が **true** に設定されていることを確認します。
   - **state** 変数は、**absent** に設定されていることを確認します。

```
---
- 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: true
      state: absent
```

5. Ansible Playbook を実行し、Playbook ファイルとインベントリーファイルを指定します。

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

6. [オプション] **server123.idm.example.com** を指すすべてのネームサーバー (NS) DNS レコードが DNS ゾーンから削除されていることを確認します。使用する DNS が IdM により管理される統合 DNS であるか、外部 DNS であるかに関わらず、確認を行なってください。

関連情報

- IdM server のアンインストール を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-server.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/server** ディレクトリーのサンプルの Playbook を参照してください。

16.5. ANSIBLE PLAYBOOK を使用した既存の IDM サーバーが非表示であることの確認

Ansible Playbook の **ipaserver ansible-freeipa** モジュールを使用して、既存の Identity Management (IdM) サーバーが非表示になっていることを確認します。この Playbook では、IdM サーバーがインストールされないことに注意してください。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
  - 制御ノードからインベントリーファイルに定義した IdM サーバーへの **SSH** 接続が正常に動作している。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/server/** ディレクトリーにある **server-hidden.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-hidden.yml server-hidden-copy.yml
```

3. **server-hidden-copy.yml** ファイルを開いて編集します。
4. **ipaserver** タスクセクションで次の変数を設定してファイルを調整し、ファイルを保存します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **name** 変数をサーバーの **FQDN** に設定します。サンプルサーバーの **FQDN** は server123.idm.example.com です。
   - **hidden** 変数が **True** に設定されていることを確認します。

```
---
- 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. Ansible Playbook を実行し、Playbook ファイルとインベントリーファイルを指定します。

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

関連情報

- Ansible Playbook を使用した Identity Management サーバーのインストール を参照してください。
- 非表示のレプリカモード を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-server.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/server** ディレクトリーのサンプルの Playbook を参照してください。

16.6. ANSIBLE PLAYBOOK を使用した既存の IDM サーバーが表示されていることの確認

Ansible Playbook で **ipaserver ansible-freeipa** モジュールを使用して、既存の Identity Management (IdM) サーバーが表示されていることを確認します。この Playbook では、IdM サーバーがインストールされないことに注意してください。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。

- Ansible バージョン 2.14 以降を使用している。
- Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
  - 制御ノードからインベントリーファイルに定義した IdM サーバーへの **SSH** 接続が正常に動作している。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/server/** ディレクトリーにある **server-not-hidden.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-not-hidden.yml server-not-hidden-copy.yml
```

3. **server-not-hidden-copy.yml** ファイルを開いて編集します。
4. **ipaserver** タスクセクションで次の変数を設定してファイルを調整し、ファイルを保存します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **name** 変数をサーバーの **FQDN** に設定します。サンプルサーバーの **FQDN** は **server123.idm.example.com** です。
   - **hidden** 変数が **no** に設定されていることを確認します。

```
---
- 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. Ansible Playbook を実行し、Playbook ファイルとインベントリーファイルを指定します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-not-hidden-copy.yml
```

関連情報

- Ansible Playbook を使用した Identity Management サーバーのインストール を参照してください。
- 非表示のレプリカモード を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-server.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/server** ディレクトリーのサンプルの Playbook を参照してください。

16.7. 既存の IDM サーバーに IDM DNS の場所が割り当てられていることの確認

Ansible Playbook の **ipaserver ansible-freeipa** モジュールを使用して、既存の Identity Management (IdM) サーバーに特定の IdM DNS の場所が割り当てられていることを確認します。

ipaserver Ansible モジュールは、IdM サーバーをインストールしないことに注意してください。

前提条件

- IdM **admin** のパスワードを把握している。
- IdM DNS の場所が存在します。サンプルの場所は **germany** です。
- サーバーへの **root** アクセス権限がある。サンプルサーバーは **server123.idm.example.com** である。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
  - 制御ノードからインベントリーファイルに定義した IdM サーバーへの **SSH** 接続が正常に動作している。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/server/** ディレクトリーにある **server-location.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-location.yml server-location-copy.yml
```

3. **server-location-copy.yml** ファイルを開いて編集します。

4. **ipaserver** タスクセクションで次の変数を設定してファイルを調整し、ファイルを保存します。

   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **name** 変数を **server123.idm.example.com** に設定します。
   - **location** 変数を **germany** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. Ansible Playbook を実行し、Playbook ファイルとインベントリーファイルを指定します。

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

6. **SSH** を使用して、**root** として **server123.idm.example.com** に接続します。

```
ssh root@server123.idm.example.com
```

7. 更新をすぐに有効にするには、サーバーで **named-pkcs11** サービスを再起動します。

```
[root@server123.idm.example.com ~]# systemctl restart named-pkcs11
```

関連情報

- Ansible Playbook を使用した Identity Management サーバーのインストール を参照してください。
- Ansible を使用して IdM の場所が存在することを確認する を参照してください。

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-server.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/server** ディレクトリーのサンプルの Playbook を参照してください。

16.8. 既存の IDM サーバーに IDM DNS の場所が割り当てられていないことの確認

Ansible Playbook の **ipaserver ansible-freeipa** モジュールを使用して、既存の Identity Management (IdM) サーバーに IdM DNS の場所が割り当てられていないことを確認します。地理的な場所を頻繁に変更するサーバーに DNS の場所を割り当てないでください。Playbook では IdM サーバーがインストールされないことに注意してください。

前提条件

- IdM **admin** のパスワードを把握している。
- サーバーへの **root** アクセス権限がある。サンプルサーバーは **server123.idm.example.com** である。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
  - 制御ノードからインベントリーファイルに定義した IdM サーバーへの **SSH** 接続が正常に動作している。

手順

1. **~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/server/** ディレクトリーにある **server-no-location.yml** Ansible Playbook ファイルをコピーします。

```
$ cp /usr/share/doc/ansible-freeipa/playbooks/server/server-no-location.yml server-no-location-copy.yml
```

3. **server-no-location-copy.yml** を開いて編集します。
4. **ipaserver** タスクセクションで次の変数を設定してファイルを調整し、ファイルを保存します。

- **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
- **name** 変数を server123.idm.example.com に設定します。
- **location** 変数が "" に設定されていることを確認してください。

```
---
- 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. Ansible Playbook を実行し、Playbook ファイルとインベントリーファイルを指定します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory server-no-location-copy.yml
```

6. **SSH** を使用して、**root** として server123.idm.example.com に接続します。

```
ssh root@server123.idm.example.com
```

7. 更新をすぐに有効にするには、サーバーで **named-pkcs11** サービスを再起動します。

```
[root@server123.idm.example.com ~]# systemctl restart named-pkcs11
```

関連情報

- Ansible Playbook を使用した Identity Management サーバーのインストール を参照してください。
- Ansible を使用した IdM での DNS の場所の管理 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-server.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/server** ディレクトリーのサンプルの Playbook を参照してください。

第17章 ANSIBLE PLAYBOOK を使用したホストの管理

Ansible は、システムの設定、ソフトウェアのデプロイ、ローリング更新の実行に使用する自動化ツールです。Ansible には Identity Management (IdM) のサポートが含まれ、Ansible モジュールを使用してホスト管理を自動化できます。

Ansible Playbook を使用してホストおよびホストエントリーを管理する際に、以下のコンセプトに基づき、操作が実行されます。

- **FQDN** でのみ定義されている IdM ホストエントリーを存在させる手順
- IP アドレスを使用して IdM ホストエントリーを存在させる手順
- 無作為のパスワードが指定された IdM ホストエントリーを複数存在させる手順
- 複数の IP アドレスが指定された IdM ホストエントリーを存在させる手順
- IdM ホストエントリーがないことを確認する手順

17.1. ANSIBLE PLAYBOOK を使用して FQDN が指定された IDM ホストエントリーを存在させる手順

以下の手順に従って、Ansible Playbook を使用して、Identity Management (IdM) にホストエントリーが存在することを確認します。ホストエントリーは、**完全修飾ドメイン名** (FQDN) によってのみ定義されます。

以下の条件のいずれかが当てはまる場合は、ホストの **FQDN** 名を指定するだけで十分です。

- IdM サーバーが DNS を管理するよう設定されていない。
- ホストに静的 IP アドレスがないか、ホストの設定時に IP アドレスが不明である。**FQDN** だけで定義されたホストを追加すると、基本的に IdM DNS サービスにプレースホルダーエントリーが作成されます。たとえば、ラップトップは IdM クライアントとして事前設定されている場合がありますが、設定時には IP アドレスがありません。DNS サービスがレコードを動的に更新すると、ホストの現在の IP アドレスが検出され、DNS レコードが更新されます。

注記

Ansible ない場合に、**ipa host-add** コマンドを使用すると、ホストエントリーが IdM に作成されます。ホストを IdM に追加すると、IdM でのホストの状態が present になります。Ansible は冪等性に依存しているので、Ansible を使用して IdM にホストを追加するには、ホストの状態を Present (**state: present**) として定義した Playbook を作成する必要があります。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。

- ~/MyPlaybooks/ ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

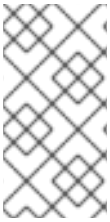
```
[ipaserver]
server.idm.example.com
```

2. IdM に存在させるホストの **FQDN** を使用して Ansible Playbook ファイルを作成します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/host/add-host.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
- 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: true
```

3. Playbook を実行します。

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



注記

以下の手順では、IdM LDAP サーバーにホストエントリーが作成されますが、ホストは IdM Kerberos レルムには登録されません。登録されるようにするには、ホストを IdM クライアントとしてデプロイする必要があります。詳細は、Ansible Playbook を使用した Identity Management クライアントのインストール を参照してください。

検証手順

1. admin として IdM サーバーにログインします。

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

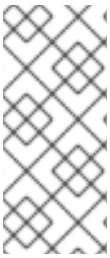
2. **ipa host-show** コマンドを入力し、ホストの名前を指定します。

```
$ 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
```

この出力で、**host01.idm.example.com** が IdM に存在することを確認します。

17.2. ANSIBLE PLAYBOOK を使用して DNS 情報など IDM ホストエントリーを存在させる手順

以下の手順に従って、Ansible Playbook を使用して、Identity Management (IdM) にホストエントリーが存在することを確認します。ホストエントリーは、ホストの **完全修飾ドメイン名** (FQDN) および IP アドレスで定義されます。



注記

Ansible ない場合に、**ipa host-add** コマンドを使用すると、ホストエントリーが IdM に作成されます。ホストを IdM に追加すると、IdM でのホストの状態が present になります。Ansible は冪等性に依存しているので、Ansible を使用して IdM にホストを追加するには、ホストの状態を Present (**state: present**) として定義した Playbook を作成する必要があります。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

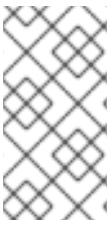
```
[ipaserver]
server.idm.example.com
```

2. IdM に存在させるホストの **完全修飾ドメイン名** (FQDN) で Ansible Playbook ファイルを作成します。また、IdM サーバーが DNS を管理するように設定され、ホストの IP アドレスが分かっている場合は、**ip_address** パラメーターの値を指定します。ホストを DNS リソースレコードに存在させるには、IP アドレスが必要です。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/host/host-present.yml** ファイルのサンプルをコピーして変更し、簡素化できます。また、その他の追加情報を含めることもできます。

```
---
- 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. Playbook を実行します。

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



注記

以下の手順では、IdM LDAP サーバーにホストエントリーが作成されますが、ホストは IdM Kerberos レルムには登録されません。登録されるようにするには、ホストを IdM クライアントとしてデプロイする必要があります。詳細は、Ansible Playbook を使用した Identity Management クライアントのインストール を参照してください。

検証手順

1. admin として IdM サーバーにログインします。

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

2. **ipa host-show** コマンドを入力し、ホストの名前を指定します。

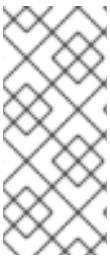
```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Description: Example host
```

```
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
```

この出力で、**host01.idm.example.com** が IdM に存在することを確認します。

17.3. ANSIBLE PLAYBOOK を使用して無作為のパスワードが指定された IDM ホストエントリーを複数存在させる手順

ipahost モジュールでは、システム管理者は、Ansible タスク 1 つだけを使用して、IdM に複数のホストエントリーが存在するか、存在しないかを確認できます。以下の手順に従って、**fully-qualified domain names** (FQDN) でのみ定義されるホストエントリーを複数存在することを確認します。Ansible Playbook を実行すると、ホストのパスワードが無作為に生成されます。



注記

Ansible ない場合に、**ipa host-add** コマンドを使用すると、ホストエントリーが IdM に作成されます。ホストを IdM に追加すると、IdM でのホストの状態が present になります。Ansible は冪等性に依存しているので、Ansible を使用して IdM にホストを追加するには、ホストの状態を Present (**state: present**) として定義した Playbook を作成する必要があります。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. IdM に存在させるホストの **完全修飾ドメイン名** (FQDN) で Ansible Playbook ファイルを作成します。ホストがすでに IdM に存在し、**update_password** が **on_create** に制限されている場合でも、Ansible Playbook で各ホストに対してランダムなパスワードを生成するには、**random: true** および **force: true** オプションを追加します。この手順を簡素化するには、**/usr/share/doc/ansible-freeipa/README-host.md** Markdown ファイルからサンプルをコピーして変更できます。

```
---
- 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: true
        force: true
      - name: host02.idm.example.com
        random: true
        force: true
    register: ipahost
```

3. Playbook を実行します。

```
$ 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"}}}
```

注記

ランダムなワンタイムパスワード (OTP) を使用して、ホストを IdM クライアントとしてデプロイする場合は、Authorization options for IdM client enrollment using an Ansible playbook または Installing a client by using a one-time password: Interactive installation を参照してください。

検証手順

1. admin として IdM サーバーにログインします。

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

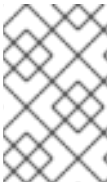
2. **ipa host-show** コマンドを入力し、ホストのいずれかの名前を指定します。

```
$ ipa host-show host01.idm.example.com
  Host name: host01.idm.example.com
  Password: True
  Keytab: False
  Managed by: host01.idm.example.com
```

この出力で、**host01.idm.example.com** が無作為に作成されたパスワードが指定された IdM に存在することを確認します。

17.4. ANSIBLE PLAYBOOK を使用して複数の IP アドレスが指定された IDM ホストエントリーを存在させる手順

以下の手順に従って、Ansible Playbook を使用して Identity Management (IdM) にホストエントリーが存在することを確認します。ホストエントリーは、**完全修飾ドメイン名** (FQDN) と複数の IP アドレスで定義されます。



注記

Ansible **ipahost** モジュールでは、**ipa host** ユーティリティーとは対照的に、ホストの IPv4 および IPv6 アドレスが複数存在させたり、または存在させなかったりできます。**ipa host-mod** コマンドは IP アドレスを処理できません。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

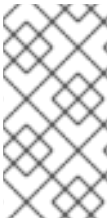
2. Ansible Playbook ファイルを作成します。**ipahost** 変数の **名前** として、IdM に存在させるホストの **完全修飾ドメイン名** (FQDN) を指定します。**ip_address** 構文を使用して、複数の IPv4 および IPv6 **ip_address** 値をそれぞれ別の行に指定します。この手順

は、**/usr/share/doc/ansible-freeipa/playbooks/host/host-member-ipaddresses-present.yml** ファイルのサンプルをコピーして変更し、簡素化できます。追加情報を含めることもできます。

```
---
- 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: true
```

3. Playbook を実行します。

```
$ 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
```



注記

この手順では、IdM LDAP サーバーにホストエントリーは作成されますが、ホストは IdM Kerberos レルムに登録されません。登録されるようにするには、ホストを IdM クライアントとしてデプロイする必要があります。詳細は、Ansible Playbook を使用した Identity Management クライアントのインストール を参照してください。

検証手順

1. admin として IdM サーバーにログインします。

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

2. **ipa host-show** コマンドを入力し、ホストの名前を指定します。

```
$ 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
```

この出力で、**host01.idm.example.com** が IdM に存在することを確認します。

3. IdM DNS レコードにホストの複数の IP アドレスが存在することを確認するには、**ipa dnsrecord-show** コマンドを入力し、以下の情報を指定します。

   - IdM ドメインの名前
   - ホストの名前

   ```
   $ 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
   ```

   この出力では、Playbook で指定された IPv4 アドレスおよび IPv6 アドレスがすべて **host01.idm.example.com** ホストエントリーに正しく関連付けられていることを確認します。

17.5. ANSIBLE PLAYBOOK を使用して IDM ホストエントリーがないことを確認する手順

以下の手順に従って、Ansible Playbook を使用して Identity Management (IdM) にホストエントリーがないことを確認します。

前提条件

- IdM 管理者の認証情報

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. IdM に存在させないホストの **完全修飾ドメイン名** (FQDN) を指定して Ansible Playbook ファイルを作成します。IdM ドメインに DNS が統合されている場合は、**updatedns: true** オプションを使用して、ホストに関連するあらゆる種類のレコードを DNS から削除します。
   この手順は、**/usr/share/doc/ansible-freeipa/playbooks/host/delete-host.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

   ```
   ---
   - 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: true
         state: absent
   ```

3. Playbook を実行します。

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

注記

この手順の結果は以下のようになります。

- IdM Kerberos レルムにホストが存在していない。
- IdM LDAP サーバーにホストエントリーが存在しない。

SSSD (System Security Services Daemon) などのシステムサービスの特定の IdM 設定をクライアントホスト自体から削除するには、クライアントで **ipa-client-install --uninstall** コマンドを実行する必要があります。詳細は、IdM クライアントのアンインストール を参照してください。

検証手順

1. admin として **ipaserver** にログインします。

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

2. **host01.idm.example.com** に関する情報を表示します。

```
$ ipa host-show host01.idm.example.com
ipa: ERROR: host01.idm.example.com: host not found
```

この出力では、ホストが IdM に存在しないことを確認します。

17.6. 関連情報

- **/usr/share/doc/ansible-freeipa/README-host.md** Markdown ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/host** ディレクトリーにある追加の Playbook を表示します。

第18章 ANSIBLE PLAYBOOK を使用したホストグループの管理

Identity Management (IdM) のホストグループ と、Ansible を使用して Identity Management (IdM) のホストグループに関係する操作を実行する方法について詳しく説明します。

- IdM のホストグループ
- IdM ホストグループを存在させる手順
- IdM ホストグループにホストを存在させる手順
- IdM ホストグループのネスト化
- IdM ホストグループにメンバーマネージャーを存在させる手順
- IdM ホストグループにホストを存在させないようにする方法
- ネスト化されたホストグループを IdM ホストグループに存在させないようにする方法
- IdM ホストグループにメンバーマネージャーを存在させないようにする方法

18.1. IDM のホストグループ

IdM ホストグループを使用すると、重要な管理タスク (特にアクセス制御) を一元管理できます。

ホストグループの定義

ホストグループは、一般的なアクセス制御ルールやその他の特性を持つ IdM ホストセットが含まれるエンティティーです。たとえば、企業の部門、物理的な場所、またはアクセス制御要件に基づいてホストグループを定義できます。

IdM のホストグループには以下が含まれます。

- IdM サーバーおよびクライアント
- その他の IdM ホストグループ

デフォルトで作成されたホストグループ

デフォルトでは、IdM サーバーは、全 IdM サーバーホストのホストグループ **ipaservers** を作成します。

直接および間接のグループメンバー

IdM のグループ属性は、直接メンバーと間接メンバーの両方に適用されます。ホストグループ B がホストグループ A のメンバーである場合、ホストグループ B のすべてのユーザーはホストグループ A の間接メンバーと見なされます。

18.2. ANSIBLE PLAYBOOK を使用して IDM ホストグループを存在させる手順

以下の手順に従って、Ansible Playbook を使用して、Identity Management (IdM) にホストグループが存在することを確認します。

注記

Ansible を使用しない場合には、**ipa hostgroup-add** コマンドでホストグループエントリーを IdM に作成します。ホストグループを IdM に追加すると、IdM でのホストグループの状態が present になります。Ansible は冪等性に依存しているので、Ansible を使用して IdM にホストグループを追加するには、ホストの状態を Present (**state: present**) として定義した Playbook を作成する必要があります。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **inventory.file** などのインベントリーファイル を作成し、そのファイルで、ターゲットに設定する IdM サーバーのリストと合わせて **ipaserver** を定義します。

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

2. 必要なホストグループ情報を使用して Ansible Playbook ファイルを作成します。たとえば、**databases** という名前のホストグループを存在させるには、**- ipahostgroup** タスクで **name: databases** を指定します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-present.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
- 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
```

Playbook で **state: present** が指定されていると、IdM にホストグループがない場合のホストグループの追加要求という意味です。

3. Playbook を実行します。

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

検証手順

1. admin として **ipaserver** にログインします。

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

2. admin の Kerberos チケットを要求します。

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

3. IdM に存在させるホストグループに関する情報を表示します。

```
$ ipa hostgroup-show databases
  Host-group: databases
```

データベース ホストグループが IdM に存在します。

18.3. ANSIBLE PLAYBOOK を使用して IDM ホストグループにホストを存在させる手順

以下の手順に従って、Ansible Playbook を使用して、Identity Management (IdM) のホストグループにホストが存在することを確認します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- Ansible Playbook で参照するホストが IdM に存在する。詳細は Ansible Playbook を使用した IdM ホストエントリーの存在の確認 を参照してください。

- Ansible Playbook ファイルから参照するホストグループが IdM に追加されている。詳細は、Ansible Playbook を使用して IdM ホストグループを存在させる手順 を参照してください。

手順

1. **inventory.file** などのインベントリーファイル を作成し、そのファイルで、ターゲットに設定する IdM サーバーのリストと合わせて **ipaserver** を定義します。

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

2. 必要なホスト情報を使用して Ansible Playbook ファイルを作成します。**ipahostgroup** 変数の **name** パラメーターを使用してホストグループの名前を指定します。**ipahostgroup** 変数の **host** パラメーターを使用してホストの名前を指定します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-present-in-hostgroup.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
- 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
```

この Playbook は、**db.idm.example.com** ホストを **データベース** ホストグループに追加します。**action: member** の行は、Playbook の実行時に **databases** グループ自体の追加を試行しないことを示します。代わりに、**db.idm.example.com** の **databases** への追加を試行するだけです。

3. Playbook を実行します。

```
$ 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
```

検証手順

1. admin として **ipaserver** にログインします。

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

2. admin の Kerberos チケットを要求します。

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

3. ホストグループに関する情報を表示して、どのホストが存在するかを確認します。

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member hosts: db.idm.example.com
```

db.idm.example.com ホストは、**database** ホストグループのメンバーとして存在します。

18.4. ANSIBLE PLAYBOOK を使用した IDM ホストグループのネスト化

以下の手順に従って、Ansible Playbook を使用して、Identity Management (IdM) ホストグループにネスト化されたホストグループが存在することを確認します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- Ansible Playbook ファイルから参照するホストグループが IdM に存在する。詳細は、Ansible Playbook を使用して IdM ホストグループを存在させる手順 を参照してください。

手順

1. **inventory.file** などのインベントリーファイル を作成し、そのファイルで、ターゲットに設定する IdM サーバーのリストと合わせて **ipaserver** を定義します。

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

2. 必要なホストグループ情報を使用して Ansible Playbook ファイルを作成します。ネストされたホストグループ **A** が、Ansible Playbook のホストグループ **B** に存在することを確認するには、**- ipahostgroup** 変数で **name** 変数を使用して、ホストグループ **B** の名前を指定します。**hostgroup** 変数でネスト化されたホストグループ **A** の名前を指定します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-present-in-hostgroup.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
```

```
- 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
```

この Ansible Playbook は、**database** ホストグループに **myqsl-server** および **oracle-server** ホストグループが存在することを確認します。**action: member** 行は、Playbook が実行されると、**databases** グループ自体を IdM に追加しようとはしません。

3. Playbook を実行します。

```
$ 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
```

検証手順

1. admin として **ipaserver** にログインします。

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

2. admin の Kerberos チケットを要求します。

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

3. ネスト化されたホストグループが含まれるホストグループに関する情報を表示します。

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member hosts: db.idm.example.com
  Member host-groups: mysql-server, oracle-server
```

mysql-server および **oracle-server** ホストグループは、**databases** ホストグループに存在します。

18.5. ANSIBLE PLAYBOOK を使用して IDM ホストグループにメンバーマネージャーを存在させる手順

以下の手順では、Ansible Playbook を使用して、IdM ホストおよびホストグループにメンバーマネージャーを存在させる方法を説明します。

前提条件

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- メンバーマネージャーとして追加するホストまたはホストグループの名前と、管理するホストグループ名が必要です。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. 必要なホストおよびホストグループメンバー管理情報を使用して Ansible Playbook ファイルを作成します。

```
---

- 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. Playbook を実行します。

```
$ 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
```

検証手順

ipa group-show コマンドを使用して group_name グループのメンバーマネージャーとして example_member と project_admins が含まれていることを確認できます。

1. 管理者として **ipaserver** にログインします。

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

2. **testhostgroup** に関する情報を表示します。

```
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
```

関連情報

- **ipa hostgroup-add-member-manager --help** を参照してください。
- **ipa** の man ページを参照してください。

18.6. ANSIBLE PLAYBOOK を使用して IDM ホストグループにホストを存在させないようにする方法

以下の手順に従って、Ansible Playbook を使用して、Identity Management (IdM) のホストグループにホストがないことを確認します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

- Ansible Playbook で参照するホストが IdM に存在する。詳細は Ansible Playbook を使用した IdM ホストエントリーの存在の確認 を参照してください。
- Ansible Playbook ファイルから参照するホストグループが IdM に存在する。詳細は、Ansible Playbook を使用して IdM ホストグループを存在させる手順 を参照してください。

手順

1. **inventory.file** などのインベントリーファイル を作成し、そのファイルで、ターゲットに設定する IdM サーバーのリストと合わせて **ipaserver** を定義します。

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

2. 必要なホストおよびホストグループ情報を使用して Ansible Playbook ファイルを作成します。**ipahostgroup** 変数の **name** パラメーターを使用してホストグループの名前を指定します。**ipahostgroup** 変数の **host** パラメーターを使用して、ホストグループに、存在させないようにするホストの名前を指定します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-absent-in-hostgroup.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
- 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
```

この Playbook では、**databases** ホストグループに **db.idm.example.com** ホストを存在させないようにできます。**action: member** の行で、Playbook の実行時に **databases** グループ自体の削除を試行しないように指定します。

3. Playbook を実行します。

```
$ 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
```

検証手順

1. admin として **ipaserver** にログインします。

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

2. admin の Kerberos チケットを要求します。

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

3. ホストグループと、そのホストグループに含まれるホストに関する情報を表示します。

```
$ ipa hostgroup-show databases
  Host-group: databases
  Member host-groups: mysql-server, oracle-server
```

db.idm.example.com ホストは **データベース** ホストグループに存在していません。

18.7. ANSIBLE PLAYBOOK を使用して IDM ホストグループに、ネスト化されたホストグループを存在させないようにする方法

以下の手順に従って、Ansible Playbook を使用して、Identity Management (IdM) の外部ホストグループからネスト化されたホストグループがないことを確認します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- Ansible Playbook ファイルから参照するホストグループが IdM に存在する。詳細は、Ansible Playbook を使用して IdM ホストグループを存在させる手順 を参照してください。

手順

1. **inventory.file** などのインベントリーファイル を作成し、そのファイルで、ターゲットに設定する IdM サーバーのリストと合わせて **ipaserver** を定義します。

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

2. 必要なホストグループ情報を使用して Ansible Playbook ファイルを作成します。**-ipahostgroup** 変数の **name** 変数を使用して、外部ホストグループの名前を指定します。**hostgroup** 変数でネスト化されたホストグループの名前を指定します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/hostgroup/ensure-hosts-and-hostgroups-are-absent-in-hostgroup.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
- 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
```

この Playbook は、**mysql-server** および **oracle-server** ホストグループが **databases** ホストグループにないことを確認します。**action: member** 行は、Playbook の実行時に、**databases** グループ自体の IdM からの削除は試行されないようにします。

3. Playbook を実行します。

```
$ 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
```

検証手順

1. admin として **ipaserver** にログインします。

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

2. admin の Kerberos チケットを要求します。

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

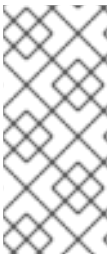
3. ネスト化されたホストグループを存在させないホストグループに関する情報を表示します。

```
$ ipa hostgroup-show databases
  Host-group: databases
```

この出力では、**mysql-server** および **oracle-server** のネスト化されたホストグループが、外部 **databases** のホストグループにないことを確認します。

18.8. ANSIBLE PLAYBOOK を使用して IDM ホストグループを存在させないようにする方法

以下の手順に従って、Ansible Playbook を使用して Identity Management (IdM) にホストグループがないことを確認します。



注記

Ansible を使用しない場合には、**ipa hostgroup-del** コマンドでホストグループエントリーを IdM から削除します。IdM からホストグループを削除すると、IdM にホストグループが存在しない状態になります。Ansible は冪等性に依存しているので、Ansible を使用して IdM からホストグループを削除するには、ホストの状態を Absent (**state: absent**) として定義した Playbook を作成する必要があります。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. **inventory.file** などのインベントリーファイル を作成し、そのファイルで、ターゲットに設定する IdM サーバーのリストと合わせて **ipaserver** を定義します。

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

2. 必要なホストグループ情報を使用して Ansible Playbook ファイルを作成します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/user/ensure-hostgroup-is-absent.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
- 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
```

この Playbook では、IdM から **databases** ホストグループを存在させないようにします。**state: absent** は、IdM からホストグループが削除されていない限り、ホストグループの削除要求を意味します。

3. Playbook を実行します。

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

検証手順

1. admin として **ipaserver** にログインします。

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

2. admin の Kerberos チケットを要求します。

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

3. 存在させないようにするホストグループの情報を表示します。

```
$ ipa hostgroup-show databases
ipa: ERROR: databases: host group not found
```

databases ホストグループが IdM に存在しません。

18.9. ANSIBLE PLAYBOOK を使用して IDM ホストグループからホストを存在させないようにする方法

以下の手順では、Ansible Playbook を使用して、IdM ホストおよびホストグループにメンバーマネージャーを存在させないようにする方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- メンバーマネージャーから削除するユーザーまたはユーザーグループの名前と、管理するホストグループの名前が必要です。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. 必要なホストおよびホストグループメンバー管理情報を使用して Ansible Playbook ファイルを作成します。

```
---

- 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. Playbook を実行します。

```
$ 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
```

検証手順

ipa group-show コマンドを使用して、**group_name** グループに **example_member** または **project_admins** がメンバーマネージャーとして含まれているかどうかを確認できます。

1. 管理者として **ipaserver** にログインします。

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

2. **testhostgroup** に関する情報を表示します。

```
ipaserver]$ ipa hostgroup-show group_name
  Host-group: group_name
  Member hosts: server.idm.example.com
  Member host-groups: testhostgroup2
```

関連情報

- **ipa hostgroup-add-member-manager --help** を参照してください。
- **ipa** の man ページを参照してください。

第19章 IDM パスワードポリシーの定義

本章では、Identity Management (IdM) パスワードポリシーについて、また、Ansible Playbook を使用して IdM に新規パスワードポリシーを追加する方法を説明します。

19.1. パスワードポリシーとは

パスワードポリシーは、パスワードが満たす必要のある一連のルールです。たとえば、パスワードポリシーでは、パスワードの最小長と最大有効期間を定義できます。このポリシーの対象となる全ユーザーには、十分に長いパスワードを設定して、指定の条件を満たす頻度でパスワードを変更する必要があります。このようにパスワードポリシーを使用することで、ユーザーのパスワードが検出されて悪用されるリスクが軽減されます。

19.2. IDM のパスワードポリシー

パスワードは、Identity Management (IdM) ユーザーが IdM Kerberos ドメインに対して認証する最も一般的な方法です。パスワードポリシーでは、このような IdM ユーザーのパスワードが満たす必要条件を定義します。

注記

IdM パスワードポリシーは基礎となる LDAP ディレクトリーで設定されますが、Kerberos Key Distribution Center (KDC) はパスワードポリシーを強制的に使用します。

パスワードポリシー属性 は、IdM でのパスワードポリシーの定義に使用できる属性をリスト表示します。

表19.1 パスワードポリシーの属性

属性	説明	例
Max lifetime	パスワードのリセットが必要になるまでの、パスワードの有効期間 (日) の上限です。デフォルト値は 90 日です。 この属性が 0 に設定されている場合、パスワードの有効期限は切れないことに注意してください。	Max lifetime = 180 ユーザーパスワードは 180 日間のみ有効です。有効期限が経過すると、IdM は変更を求めるプロンプトを表示します。
Min lifetime	パスワードを変更してから次に変更操作を行うまでに最小限開ける必要のある時間。	Min lifetime = 1 ユーザーがパスワードの変更後に、次に変更するまでに最低でも 1 時間待機する必要があります。
History size	保存される以前のパスワード数。パスワードの履歴にあるパスワードを再利用できませんが、保存されていない以前のものは使用できます。	History size = 0 この場合、パスワード履歴は空になり、ユーザーは以前のパスワードをどれでも再利用できます。

属性	説明	例
Character classes	パスワードで使用する必要のある文字クラスの数。文字クラスは次のとおりです。 * 大文字 * 小文字 * 数字 * コンマ (,)、ピリオド (.)、アスタリスク (*) などの特殊文字 * 他の UTF-8 文字 1 つの文字を複数回連続で使用すると、文字クラスが 1 つ減少します。以下に例を示します。 * **Secret1** には、大文字、小文字、数字の 3 つの文字クラスがあります。 * **Secret111** には、大文字、小文字、数字が含まれていますが、**1** を 繰り返し使用使用したため、ペナルティが -1 で文字クラスが 2 つになりります。	Character classes = 0 必要なクラスのデフォルト数は 0 です。番号を設定するには、**--minclasses** オプションを指定して **ipa pwpolicy-mod** コマンドを実行します。 この表の下に記載されている 重要 の注意事項も併せて参照してください。
Min length	パスワードの最小長。 追加のパスワードポリシーオプション のいずれかが設定されていると、パスワードの最小長は 6 文字です。	Min length = 8 8 文字未満のパスワードは使用できません。
Max failures	IdM がユーザーアカウントをロックするまでのログイン試行の最大失敗数。	Max failures = 6 ユーザーがパスワードを誤って 7 回入力すると、IdM はユーザーアカウントをロックします。
Failure reset interval	失敗したログイン試行回数を IdM がリセットするまでの時間 (秒単位)。	Failure reset interval = 60 **Max failures** で定義されたログイン試行回数が 1 分以上経過すると、ユーザーはユーザーアカウントがロックされる心配なく再ログインを試みることができます。

属性	説明	例
Lockout duration	**Max failures** で定義された回数のログイン試行に失敗した後にユーザーアカウントがロックされる時間 (秒単位)。	Lockout duration = 600 アカウントがロックされると、10 分間ログインできません。

重要

国際文字や記号を使用できないハードウェアセットが各種ある場合には、文字クラス要件に英語と共通記号を使用してください。パスワードの文字クラスポリシーの詳細は、Red Hat ナレッジベースの What characters are valid in a password? を参照してください。

19.3. ANSIBLE PLAYBOOK を使用して IDM にパスワードポリシーを存在させる手順

Ansible Playbook を使用して Identity Management (IdM) にパスワードポリシーを存在させるには、次の手順に従います。

IdM におけるデフォルトの **global_policy** パスワードポリシーでは、パスワード内の異なる文字クラスの数は 0 に設定されています。履歴サイズも 0 に設定されています。

以下の手順に従って、Ansible Playbook を使用して、IdM グループにより強力なパスワードポリシーを適用します。

注記

IdM グループのパスワードポリシーのみを定義できます。個別ユーザーにパスワードポリシーを定義できません。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。
- IdM にパスワードポリシーが存在することを確認するグループ。

手順

1. **inventory.file** などのインベントリーファイルを作成し、**[ipaserver]** セクションに IdM サーバーの **FQDN** を定義します。

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

2. Ansible Playbook を作成して、存在させるパスワードポリシーを定義します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/pwpolicy/pwpolicy_present.yml** ファイルの例をコピーして変更し、簡素化できます。

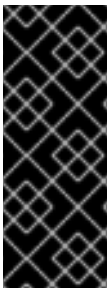
```
---
- name: Tests
  hosts: ipaserver

  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
```

   各変数の意味については、パスワードポリシーの属性 を参照してください。

3. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file
path_to_playbooks_directory_/new_pwpolicy_present.yml
```

Ansible Playbook を使用して、**ops** グループのパスワードポリシーを IdM に存在させることができました。



重要

ops パスワードポリシーの優先度は 1に設定されますが、**global_policy** パスワードポリシーには優先度が設定されません。上記の理由から、**ops** ポリシーは **ops** グループの**global_policy** より自動的に優先され、すぐに有効になります。

global_policy は、ユーザーにポリシーが設定されていない場合のフォールバックポリシーとして機能し、グループポリシーよりも優先されることはありません。

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-pwpolicy.md** ファイルを参照してください。

- Password policy priorities を参照してください。

19.4. IDM での追加のパスワードポリシーオプション

Identity Management (IdM) 管理者は、**libpwquality** 機能セットに基づく追加のパスワードポリシーオプションを有効にすることで、デフォルトのパスワード要件を強化できます。追加のパスワードポリシーオプションには、以下が含まれます。

--maxrepeat

新しいパスワードに使用できる、連続する同一文字数の上限を指定します。

--maxsequence

新しいパスワードにおける単調な文字シーケンスの最大長を指定します。このような配列の例は、12345 または **fedcb** です。このようなパスワードのほとんどは、簡素化チェックに合格しません。

--dictcheck

ゼロ以外の場合は、パスワード (修正可能) が辞書の単語と一致するかどうかを確認します。現在、**libpwquality** は、**cracklib** ライブラリーを使用してディクショナリーの確認を実行しています。

--usercheck

ゼロ以外の場合は、パスワード (修正可能) に、何らかの形式でユーザー名が含まれているかどうかを確認します。ユーザー名が 3 文字より短い場合は実行されません。

既存のパスワードには、追加のパスワードポリシーオプションを適用できません。追加オプションのいずれかを適用すると、IdM は、パスワードの最小文字数である **--minlength** オプションを自動的に 6 文字に設定します。

注記

RHEL 7、RHEL 8、RHEL 9 サーバーが混在する環境では、RHEL 8.4 以降で実行されているサーバーにのみ追加のパスワードポリシー設定を適用できます。ユーザーが IdM クライアントにログインし、IdM クライアントが RHEL 8.3 以前で実行している IdM サーバーと通信している場合は、システム管理者が設定した新しいパスワードポリシーの要件は適用されません。一貫した動作を確認するには、すべてのサーバーを RHEL 8.4 以降にアップグレードまたは更新します。

関連情報:

- IdM グループへの追加のパスワードポリシーの適用
- **pwquality(3)** man ページ

19.5. IDM グループへの追加のパスワードポリシーオプションの適用

Identity Management (IdM) で追加のパスワードポリシーオプションを適用するには、次の手順に従います。ここでは、新しいパスワードにユーザー名が含まれていないことと、パスワードに同じ文字が連続して 2 文字以内になるようにすることで、**マネージャー** グループのパスワードポリシーを強化する方法を説明します。

前提条件

- IdM 管理者としてログインしている。

- **マネージャー** グループが IdM に存在している。
- **マネージャー** パスワードポリシーが IdM に存在している。

手順

1. **マネージャー** グループのユーザーが提案するすべての新しいパスワードに、ユーザー名の確認を適用します。

```
$ ipa pwpolicy-mod --usercheck=True managers
```

注記

パスワードポリシーの名前を指定しないと、デフォルトの **global_policy** が変更されます。

2. **マネージャー** パスワードポリシーで、同一の連続した文字の上限を 2 に設定します。

```
$ ipa pwpolicy-mod --maxrepeat=2 managers
```

パスワードに、同一の連続した文字が 2 文字を超える場合は、パスワードが使用できなくなります。たとえば、**eR873mUi111YJQ** の組み合わせは、連続して 3 つの **1** を含むため、使用できません。

検証

1. **test_user** という名前のテストユーザーを追加します。

```
$ ipa user-add test_user
First name: test
Last name: user
----------------------------
Added user "test_user"
----------------------------
```

2. テストユーザーを **マネージャー** グループに追加します。
   a. IdM Web UI で、**Identity → Groups → User Groups** をクリックします。
   b. **managers** をクリックします。
   c. **Add** をクリックします。
   d. **Add users into user group 'managers'**ページで、**test_user** をチェックします。
   e. **>** 矢印をクリックして、ユーザーを **Prospective** 列に移動します。
   f. **Add** をクリックします。
3. テストユーザーのパスワードをリセットします。
   a. **Identity → Users** に移動します。
   b. **test_user** をクリックします。

c. **Actions** メニューで、**Reset Password** をクリックします。

d. ユーザーの一時パスワードを入力します。

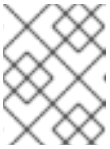
4. コマンドラインで、**test_user** の Kerberos Ticket-Granting Ticket (TGT) を取得してみてください。

```
$ kinit test_user
```

a. 一時パスワードを入力します。

b. パスワードを変更する必要があることがシステムから通知されます。**test_user** のユーザー名を含むパスワードを入力します。

```
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.
```



注記

Kerberos には、詳細なエラーパスワードポリシーの報告はなく、特定のケースでは、パスワードが拒否された理由を明確に示していません。

c. 入力したパスワードが拒否されたことがシステムから通知されます。連続して 3 文字以上の同一文字を含むパスワードを入力します。

```
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. 入力したパスワードが拒否されたことがシステムから通知されます。**マネージャー** パスワードポリシーの基準を満たすパスワードを入力します。

```
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. 取得した TGT を表示します。

```
$ 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
```

マネージャー のパスワードポリシーが、**マネージャー** グループのユーザーに対して正しく機能するようになりました。

関連情報

- IdM での追加のパスワードポリシー

19.6. ANSIBLE PLAYBOOK を使用して追加のパスワードポリシーオプションを IDM グループに適用する

Ansible Playbook を使用して追加のパスワードポリシーオプションを適用し、特定の IdM グループのパスワードポリシー要件を強化できます。この目的には、**maxrepeat**、**maxsequence**、**dictcheck**、および **usercheck** パスワードポリシーオプションを使用できます。この例では、**managers** グループに次の要件を設定する方法を説明します。

- ユーザーの新しいパスワードに、ユーザーのそれぞれのユーザー名が含まれていない。
- パスワードに含まれる連続する同一の文字が 2 文字以下である。
- パスワードに含まれる単調な文字列が 3 文字以内である。これは、システムが **1234** や **abcd** などの文字列を含むパスワードを受け入れないことを意味します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
  - **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。
- IdM にパスワードポリシーが存在することを確認するグループ。

手順

1. Ansible Playbook ファイル **manager_pwpolicy_present.yml** を作成して、存在させるパスワードポリシーを定義します。この手順を簡素化するには、次の例をコピーして変更します。

```
---
- 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. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file
path_to_playbooks_directory_/manager_pwpolicy_present.yml
```

検証

1. **test_user** という名前のテストユーザーを追加します。

```
$ ipa user-add test_user
First name: test
Last name: user
----------------------------
Added user "test_user"
----------------------------
```

2. テストユーザーを **マネージャー** グループに追加します。
   a. IdM Web UI で、**Identity → Groups → User Groups** をクリックします。
   b. **managers** をクリックします。
   c. **Add** をクリックします。
   d. **Add users into user group 'managers'**ページで、**test_user** をチェックします。
   e. **>** 矢印をクリックして、ユーザーを **Prospective** 列に移動します。
   f. **Add** をクリックします。
3. テストユーザーのパスワードをリセットします。
   a. **Identity → Users** に移動します。
   b. **test_user** をクリックします。
   c. **Actions** メニューで、**Reset Password** をクリックします。
   d. ユーザーの一時パスワードを入力します。
4. コマンドラインで、**test_user** の Kerberos Ticket-Granting Ticket (TGT) を取得してみてください。

```
$ kinit test_user
```

   a. 一時パスワードを入力します。
   b. パスワードを変更する必要があることがシステムから通知されます。**test_user** のユーザー名を含むパスワードを入力します。

```
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.
```

注記

Kerberos には、詳細なエラーパスワードポリシーの報告はなく、特定のケースでは、パスワードが拒否された理由を明確に示していません。

c. 入力したパスワードが拒否されたことがシステムから通知されます。連続して 3 文字以上の同一文字を含むパスワードを入力します。

```
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. 入力したパスワードが拒否されたことがシステムから通知されます。3 文字を超える単調な文字列を含むパスワードを入力します。たとえば、**1234** や **fedc** などの文字列です。

```
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. 入力したパスワードが拒否されたことがシステムから通知されます。**マネージャー** パスワードポリシーの基準を満たすパスワードを入力します。

```
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. TGT を取得したことを確認します。これは、有効なパスワードを入力した後にのみ可能です。

```
$ 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
```

関連情報

- IdM での追加のパスワードポリシー
- **/usr/share/doc/ansible-freeipa/README-pwpolicy.md**
- **/usr/share/doc/ansible-freeipa/playbooks/pwpolicy**

第20章 IDM クライアントの IDM ユーザーへの SUDO アクセスの許可

Identity Management でユーザーに **sudo** アクセス権を付与する方法を詳しく説明します。

20.1. IDM クライアントの SUDO アクセス

システム管理者は、root 以外のユーザーに、通常 **root** ユーザー用に予約されている管理コマンドを実行できるようにする **sudo** アクセスを付与できます。その結果、ユーザーが、通常、**root** ユーザー用に予約される管理コマンドを実行する場合は、コマンドの前に **sudo** を付けることができます。パスワードを入力すると、そのコマンドは **root** ユーザーとして実行されます。データベースサービスアカウントなどの別のユーザーまたはグループとして **sudo** コマンドを実行するには、**sudo** ルールの **RunAs エイリアス** を設定できます。

Red Hat Enterprise Linux (RHEL) 8 ホストが Identity Management (IdM) クライアントとして登録されている場合は、以下の方法で、どの IdM ユーザーがホストでどのコマンドを実行できるかを定義する **sudo** ルールを指定できます。

- ローカルの **/etc/sudoers** ファイル
- IdM での一元設定

コマンドラインインターフェイス (CLI) と IdM Web UI を使用して、IdM クライアントの **sudo 集約ルール** を作成できます。

Generic Security Service Application Programming Interface (GSSAPI) を使用して **sudo** のパスワードレス認証を設定することもできます。これは、UNIX ベースのオペレーティングシステムがネイティブで Kerberos サービスにアクセスして認証する方法です。**pam_sss_gss.so** Pluggable Authentication Module (PAM) を使用して SSSD サービスを介して GSSAPI 認証を呼び出し、有効な Kerberos チケットを使用して **sudo** コマンドに対して認証を行うことができます。

関連情報

- Managing sudo access を参照してください。

20.2. CLI での IDM クライアントの IDM ユーザーへの SUDO アクセス許可

Identity Management (IdM) では、特定の IdM ホストで IdM ユーザーアカウントの特定コマンドに **sudo** アクセスを付与できます。最初に **sudo** コマンドを追加してから、1 つまたは複数のコマンドに対して **sudo** ルールを作成します。

たとえば、**idmclient** マシンで **/usr/sbin/reboot** コマンドを実行する権限を **idm_user** に付与する **idm_user_reboot** の **sudo** ルールを作成するには、以下の手順を実行します。

前提条件

- IdM 管理者としてログインしている。
- IdM で **idm_user** のユーザーアカウントを作成し、ユーザーのパスワードを作成してそのアカウントのロックを解除している。CLI を使用して新しい IdM ユーザーを追加する方法の詳細は、コマンドラインを使用したユーザーの追加 を参照してください。
- **idmclient** ホストにローカル **idm_user** アカウントが存在しない。**idm_user** ユーザーは、ローカルの **/etc/passwd** ファイルには表示されません。

手順

1. IdM の **管理者** として Kerberos チケットを取得します。

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

2. **sudo** コマンドの IdM データベースに **/usr/sbin/reboot** コマンドを追加します。

```
[root@idmclient ~]# ipa sudocmd-add /usr/sbin/reboot
----------------------------------------
Added Sudo Command "/usr/sbin/reboot"
----------------------------------------
  Sudo Command: /usr/sbin/reboot
```

3. **idm_user_reboot** という名前の **sudo** ルールを作成します。

```
[root@idmclient ~]# ipa sudorule-add idm_user_reboot
--------------------------------
Added Sudo Rule "idm_user_reboot"
--------------------------------
  Rule name: idm_user_reboot
  Enabled: TRUE
```

4. **/usr/sbin/reboot** コマンドを **idm_user_reboot** ルールに追加します。

```
[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. **idm_user_reboot** ルールを IdM **idmclient** ホストに適用します。

```
[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. **idm_user** アカウントを **idm_ user_reboot** ルールに追加します。

```
[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. 必要に応じて、**idm_user_reboot** ルールの有効性を定義します。

   a. **sudo** ルールが有効である時間を定義するには、**--setattr sudonotbefore=DATE** オプションを指定して **ipa sudorule-mod sudo_rule_name** コマンドを使用します。**DATE** 値は、**yyyymmddHHMMSSZ** 形式に準拠し、明示的に指定される秒数である必要があります。たとえば、**idm_user_reboot** ルールの有効性の開始を 2025 12:34:00 年 12 月 31 に設定するには、次のコマンドを実行します。

   ```
   [root@idmclient ~]# ipa sudorule-mod idm_user_reboot --setattr
   sudonotbefore=20251231123400Z
   ```

   b. sudo ルールが有効な停止時間を定義するには、**--setattr sudonotafter=DATE** オプションを使用します。たとえば、**idm_user_reboot** ルールの有効期間の最後を 2026 12:34:00 年 12 月 31 に設定するには、次のコマンドを実行します。

   ```
   [root@idmclient ~]# ipa sudorule-mod idm_user_reboot --setattr
   sudonotafter=20261231123400Z
   ```

注記

サーバーからクライアントへの変更の伝播には数分かかる場合があります。

検証手順

1. **idmclient** ホストに **idm_user** アカウントとしてログインします。

2. **idm_user** アカウントが実行可能な **sudo** ルールを表示します。

```
[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. **sudo** を使用してマシンを再起動します。プロンプトが表示されたら、**idm_user** のパスワードを入力します。

```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

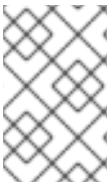
20.3. AD での IDM クライアントの IDM ユーザーへの SUDO アクセス許可

Identity Management (IdM) システム管理者は、IdM ユーザーグループを使用して、アクセス許可、ホストベースのアクセス制御、**sudo** ルール、および IdM ユーザーに対するその他の制御を設定できます。IdM ユーザーグループは、IdM ドメインリソースへのアクセスを許可および制限します。

Active Directory (AD) **ユーザー** と AD **グループ** の両方を IdM ユーザーグループに追加できます。これを実行するには、以下を行います。

1. AD ユーザーまたはグループを **非 POSIX** 外部 IdM グループに追加します。
2. 非 POSIX 外部 IdM グループを IdM **POSIX** グループに追加します。

その後、POSIX グループの権限を管理することで、AD ユーザーの権限を管理できます。例えば、特定のコマンドの **sudo** アクセスを、特定の IdM ホストの IdM POSIX ユーザーグループに付与できます。



注記

AD ユーザーグループを、IdM 外部グループにメンバーとして追加することもできます。これにより、1 つの AD レルムにユーザーおよびグループの管理を維持することで、Windows ユーザーのポリシーの定義が容易になります。

重要

IdM の SUDO ルールに AD ユーザーの ID オーバーライドを使用 **しない** でください。AD ユーザーの ID オーバーライドは、AD ユーザー自体ではなく、AD ユーザーの POSIX 属性のみを表します。

ID オーバーライドをグループメンバーとして追加できます。ただし、この機能は IdM API で IdM リソースを管理するためにのみ使用できます。グループメンバーとして ID オーバーライドを追加する可能性は POSIX 環境に拡張されていないため、**sudo** またはホストベースのアクセス制御 (HBAC) ルールのメンバーシップには使用できません。

この手順では、**ad_users_reboot sudo** ルールを作成して、**administrator@ad-domain.com** AD ユーザーに、**idmclient** IdM ホストで **/usr/sbin/reboot** コマンドを実行するパーミッションを付与します。これは通常、**root** ユーザー用に予約されています。**administrator@ad-domain.com** は **ad_users_external** 非 POSIX グループのメンバーであり、これは **ad_users** POSIX グループのメンバーでもあります。

前提条件

- IdM **admin** Kerberos の チケット許可チケット (TGT) を取得しました。
- IdM ドメインと **ad-domain.com** AD ドメインの間にフォレスト間の信頼が存在します。
- **idmclient** ホストにローカル **管理者** アカウントが存在しません。**管理者** ユーザーがローカルの **/etc/passwd** ファイルにリストされていません。

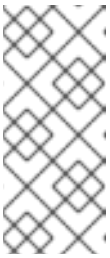
手順

1. **administrator@ad-domain** メンバーを持つ **ad_users_external** グループを含む **ad_users** グループを作成します。

a. **オプション**: AD ドメインで対応するグループを作成または選択して、IdM レルムで AD ユーザーを管理するために使用します。複数の AD グループを使用して、それらを IdM 側の異なるグループに追加できます。

b. **ad_users_external** グループを作成し、**--external** オプションを追加して、IdM ドメイン外のメンバーが含まれていることを示します。

```
[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
```



注記

ここで指定する外部グループが、Active Directory セキュリティーグループドキュメントで定義されているように、**global** または **universal** グループスコープを持つ AD セキュリティーグループであることを確認してください。たとえば、グループスコープが **domain local** であるため、**Domain users** または **Domain admins** AD セキュリティーグループは使用できません。

c. **ad_users** グループを作成します。

```
[root@ipaserver ~]# ipa group-add --desc='AD users' ad_users
----------------------
Added group "ad_users"
----------------------
  Group name: ad_users
  Description: AD users
  GID: 129600004
```

d. **administrator@ad-domain.com** AD ユーザーを外部メンバーとして **ad_users_external** に追加します。

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

AD ユーザーは、**DOMAIN\user_name** または **user_name@DOMAIN** などの完全修飾名で識別される必要があります。次に、AD ID がユーザーの AD SID にマップされます。同じことが AD グループの追加にも当てはまります。

e. **ad_users_external** を **ad_users** にメンバーとして追加します。

```
[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. **ad_users** のメンバーに、**idmclient** ホストで **/usr/sbin/reboot** を実行する権限を付与します。

   a. **sudo** コマンドの IdM データベースに **/usr/sbin/reboot** コマンドを追加します。

```
[root@idmclient ~]# ipa sudocmd-add /usr/sbin/reboot
-----------------------------------
Added Sudo Command "/usr/sbin/reboot"
-----------------------------------
  Sudo Command: /usr/sbin/reboot
```

   b. **ad_users_reboot** という名前の **sudo** ルールを作成します。

```
[root@idmclient ~]# ipa sudorule-add ad_users_reboot
--------------------------------
Added Sudo Rule "ad_users_reboot"
--------------------------------
  Rule name: ad_users_reboot
  Enabled: True
```

   c. **/usr/sbin/reboot** コマンドを **ad_users_reboot** ルールに追加します。

```
[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. **ad_users_reboot** ルールを IdM **idmclient** ホストに適用します。

```
[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. **ad_users** グループを **ad_users_reboot** ルールに追加します。

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

注記

サーバーからクライアントへの変更の伝播には数分かかる場合があります。

検証手順

1. **ad_users** グループの間接メンバーである **administrator@ad-domain.com** で **idmclient** ホストにログインします。

```
$ ssh administrator@ad-domain.com@ipaclient
Password:
```

2. オプションで、**administrator@ad-domain.com** が実行できる **sudo** コマンドを表示します。

```
[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. **sudo** を使用してマシンを再起動します。プロンプトが表示されたら、**administrator@ad-domain.com** のパスワードを入力します。

```
[administrator@ad-domain.com@idmclient ~]$ sudo /usr/sbin/reboot
[sudo] password for administrator@ad-domain.com:
```

関連情報

- Active Directory ユーザーおよび Identity Management グループ
- Include users and groups from a trusted Active Directory domain into SUDO rules

20.4. IDM WEB UI を使用した IDM クライアントでの IDM ユーザーへの SUDO アクセス権の付与

Identity Management (IdM) では、特定の IdM ホストで IdM ユーザーアカウントの特定コマンドに **sudo** アクセスを付与できます。最初に **sudo** コマンドを追加してから、1 つまたは複数のコマンドに対して **sudo** ルールを作成します。

idmclient マシンで **/usr/sbin/reboot** コマンドを実行する権限を **idm_user** に付与する **idm_user_reboot** の sudo ルールを作成するには、以下の手順を実行します。

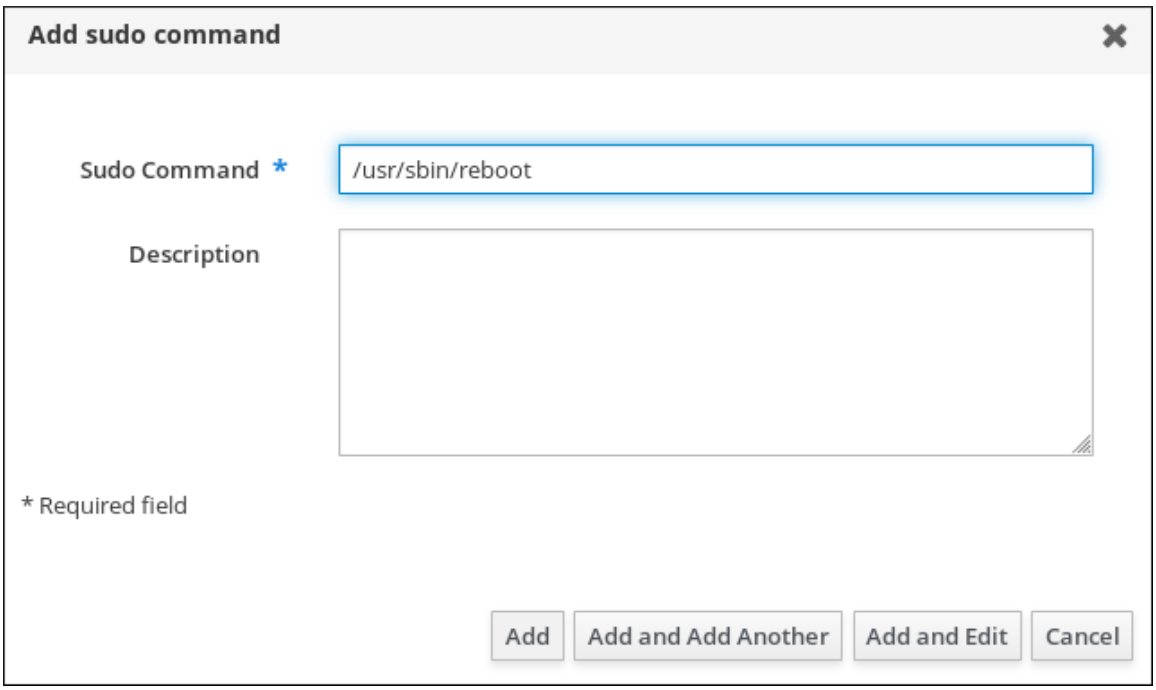
前提条件

- IdM 管理者としてログインしている。
- IdM で **idm_user** のユーザーアカウントを作成し、ユーザーのパスワードを作成してそのアカウントのロックを解除している。コマンドラインインターフェイスを使用して新しい IdM ユーザーを追加する方法の詳細は、コマンドラインを使用したユーザーの追加 を参照してください。
- **idmclient** ホストにローカル **idm_user** アカウントが存在しない。**idm_user** ユーザーは、ローカルの **/etc/passwd** ファイルには表示されません。

手順

1. **sudo** コマンドの IdM データベースに **/usr/sbin/reboot** コマンドを追加します。
   a. **Policy → Sudo → Sudo Commands** の順に移動します。
   b. 右上にある **Add** をクリックして、**Add sudo command** ダイアログボックスを開きます。
   c. **sudo**: **/usr/sbin/reboot** を使用してユーザーが実行できるコマンドを入力します。

   図20.1 IdM sudo コマンドの追加



   d. **Add** をクリックします。
2. 新しい **sudo** コマンドエントリーを使用して sudo ルールを作成し、**idm_user** が **idmclient** マシンを再起動できるようにします。
   a. **Policy → Sudo → Sudo** ルールに移動します。

b. 右上にある Add をクリックして、Add sudo rule ダイアログボックスを開きます。

c. **sudo** ルールの名前を入力します (idm_user_reboot)。

d. Add and Edit をクリックします。

e. ユーザーを指定します。

   i. Who セクションで、Specified Users and Groups のラジオボタンを選択します。

   ii. User category the rule applies to のサブセクションで Add をクリックして、Add users into sudo rule "idm_user_reboot" ダイアログボックスを開きます。

   iii. Add users into sudo rule "idm_user_reboot" ダイアログボックスにある Available 列で、idm_user チェックボックスを選択し、これを Prospective 列に移動します。

   iv. Add をクリックします。

f. ホストを指定します。

   i. Access this host セクションで、Specified Hosts and Groups ラジオボタンを確認します。

   ii. Host category this rule applies to サブセクションで Add をクリックして、Add hosts into sudo rule "idm_user_reboot" ダイアログボックスを開きます。

   iii. Add hosts into sudo rule "idm_user_reboot" ダイアログボックスにある Available 列で、idmclient.idm.example.com チェックボックスを選択し、これを Prospective 列に移動します。

   iv. Add をクリックします。

g. コマンドを指定します。

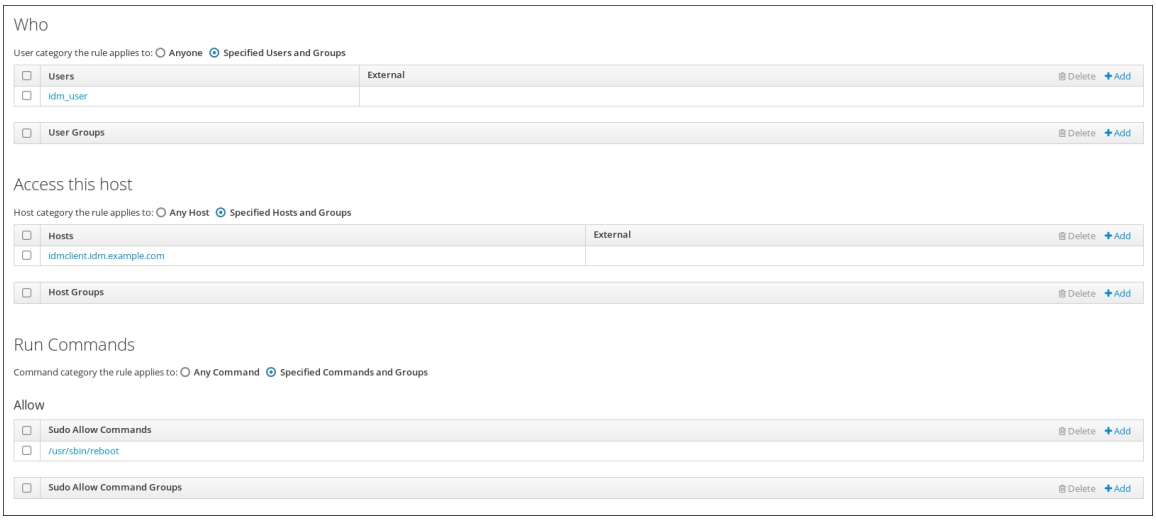
   i. Run Commands セクションの Command category the rule applies to サブセクションで、Specified Commands and Groups ラジオボタンにチェックを入れます。

   ii. Sudo Allow Commands サブセクションで Add をクリックして、Add allow sudo commands into sudo rule "idm_user_reboot" ダイアログボックスを開きます。

   iii. Add allow sudo commands into sudo rule "idm_user_reboot" ダイアログボックスにある Available 列で、**/usr/sbin/reboot** チェックボックスを選択し、これを Prospective 列に移動します。

   iv. Add をクリックして、idm_sudo_reboot ページに戻ります。

図20.2 IdM sudo ルールの追加



h. 左上隅にある **Save** をクリックします。

新しいルールはデフォルトで有効になります。

注記

サーバーからクライアントへの変更の伝播には数分かかる場合があります。

検証手順

1. **idmclient** に **idm_user** としてログインします。
2. **sudo** を使用してマシンを再起動します。プロンプトが表示されたら、**idm_user** のパスワードを入力します。

```
$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

sudo ルールが正しく設定されている場合には、マシンが再起動します。

20.5. IDM クライアントでサービスアカウントとしてコマンドを実行する CLI での SUDO ルールの作成

IdM では、**RunAs エイリアス** を使用して、**sudo** ルールを設定し、別のユーザーまたはグループとして **sudo** コマンドを実行できます。たとえば、データベースアプリケーションをホストする IdM クライアントが存在し、そのアプリケーションに対応するローカルサービスアカウントとしてコマンドを実行する必要があるとします。

この例を使用して、**run_third-party-app_report** と呼ばれるコマンドラインに **sudo** ルールを作成し、**idm_user** アカウントが **idmclient** ホストの **thirdpartyapp** サービスアカウントとして **/opt/third-party-app/bin/report** コマンドを実行できるようにします。

前提条件

- IdM 管理者としてログインしている。

- IdM で **idm_user** のユーザーアカウントを作成し、ユーザーのパスワードを作成してそのアカウントのロックを解除している。CLI を使用して新しい IdM ユーザーを追加する方法の詳細は、コマンドラインを使用したユーザーの追加 を参照してください。
- **idmclient** ホストにローカル **idm_user** アカウントが存在しない。**idm_user** ユーザーは、ローカルの **/etc/passwd** ファイルには表示されません。
- **idmclient** ホストに、**third-party-app** という名前のカスタムアプリケーションがインストールされている。
- **third-party-app** アプリケーションの **report** コマンドが、**/opt/third-party-app/bin/report** ディレクトリーにインストールされている。
- **third-party-app** アプリケーションにコマンドを実行するために、**thirdpartyapp** という名前のローカルサービスアカウントを作成している。

手順

1. IdM の **管理者** として Kerberos チケットを取得します。

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

2. **/opt/third-party-app/bin/report** コマンドを、**sudo** コマンドの IdM データベースに追加します。

```
[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. **run_third-party-app_report** という名前の **sudo** ルールを作成します。

```
[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. **--users=<user>** オプションを使用して、**sudorule-add-runasuser** コマンドに RunAs ユーザーを指定します。

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

ユーザー (または **--groups=*** オプションで指定したグループ) は、ローカルサービスアカウントや Active Directory ユーザーなどの IdM の外部に配置できます。グループ名には **%** 接頭辞を追加しないでください。

5. **/opt/third-party-app/bin/report** コマンドを **run_third-party-app_report** ルールに追加します。

```
[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. **run_third-party-app_report** ルールを IdM **idmclient** ホストに適用します。

```
[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. **idm_user** アカウントーを **run_third-party-app_report** ルールに追加します。

```
[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
```

注記

サーバーからクライアントへの変更の伝播には数分かかる場合があります。

検証手順

1. **idmclient** ホストに **idm_user** アカウントとしてログインします。
2. 新しい sudo ルールをテストします。
   a. **idm_user** アカウントが実行可能な **sudo** ルールを表示します。

```
[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. **report** コマンドを **thirdpartyapp** サービスアカウントとして実行します。

```
[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. IDM クライアントでサービスアカウントとしてコマンドを実行する IDM WEBUI での SUDO ルールの作成

IdM では、**RunAs エイリアス** を使用して、**sudo** ルールを設定し、別のユーザーまたはグループとして **sudo** コマンドを実行できます。たとえば、データベースアプリケーションをホストする IdM クライアントが存在し、そのアプリケーションに対応するローカルサービスアカウントとしてコマンドを実行する必要があるとします。

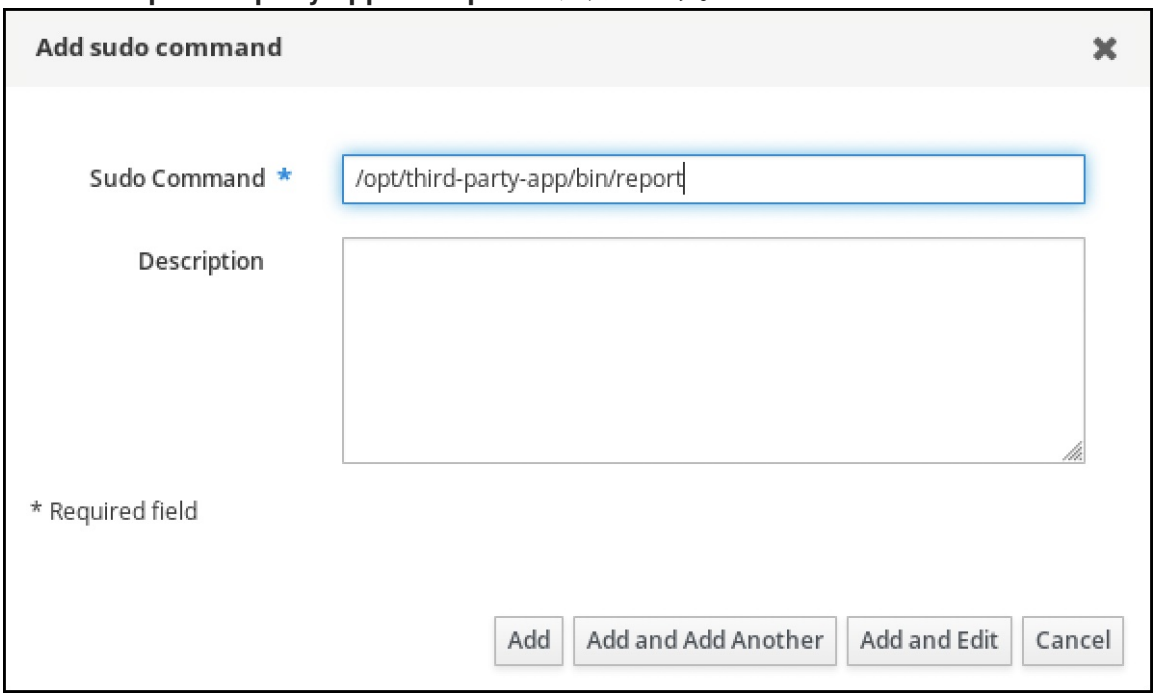
この例を使用して、**run_third-party-app_report** という IdM WebUI に **sudo** ルールを作成し、**idm_user** アカウントが **idmclient** ホストで **thirdpartyapp** サービスアカウントとして **/opt/third-party-app/bin/report** コマンドを実行できるようにします。

前提条件

- IdM 管理者としてログインしている。
- IdM で **idm_user** のユーザーアカウントを作成し、ユーザーのパスワードを作成してそのアカウントのロックを解除している。CLI を使用して新しい IdM ユーザーを追加する方法の詳細は、コマンドラインを使用したユーザーの追加 を参照してください。
- **idmclient** ホストにローカル **idm_user** アカウントが存在しない。**idm_user** ユーザーは、ローカルの **/etc/passwd** ファイルには表示されません。
- **idmclient** ホストに、**third-party-app** という名前のカスタムアプリケーションがインストールされている。
- **third-party-app** アプリケーションの **report** コマンドが、**/opt/third-party-app/bin/report** ディレクトリーにインストールされている。
- **third-party-app** アプリケーションにコマンドを実行するために、**thirdpartyapp** という名前のローカルサービスアカウントを作成している。

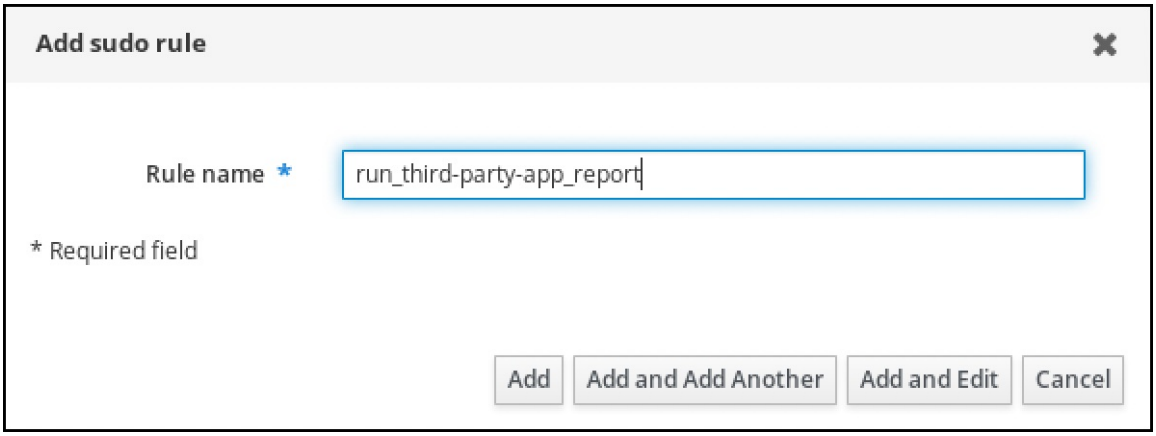
手順

1. **/opt/third-party-app/bin/report** コマンドを、**sudo** コマンドの IdM データベースに追加します。
   a. **Policy → Sudo → Sudo Commands** の順に移動します。
   b. 右上にある **Add** をクリックして、**Add sudo command** ダイアログボックスを開きます。
   c. コマンド **/opt/third-party-app/bin/report** を入力します。



   d. **Add** をクリックします。

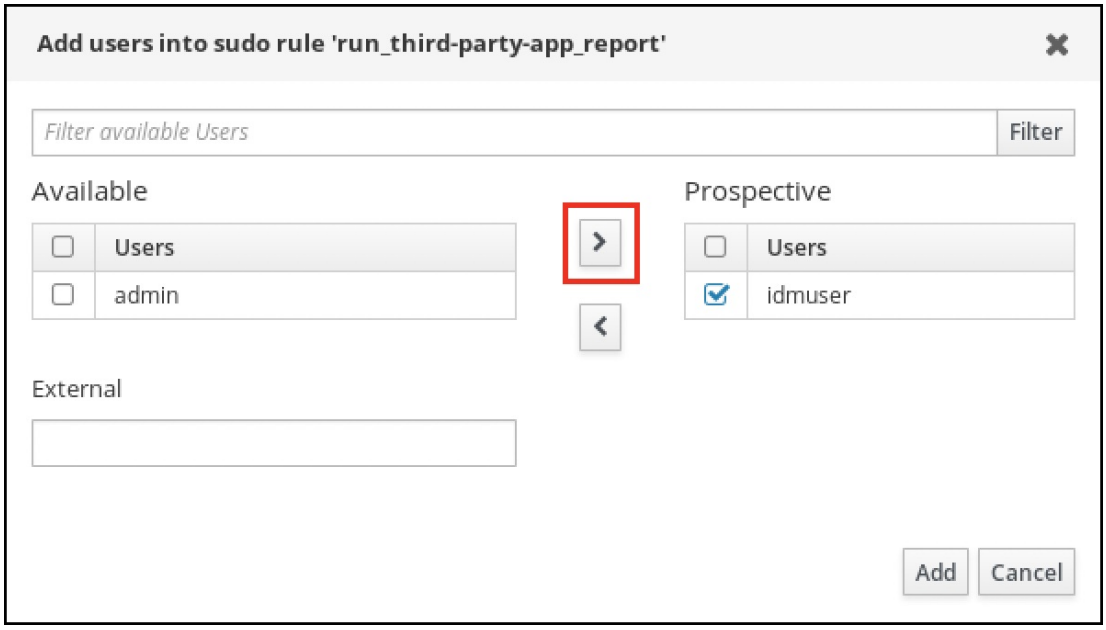
2. 新しい **sudo** コマンドエントリーを使用して、新しい **sudo** ルールを作成します。
   a. **Policy → Sudo → Sudo ルール**に移動します。
   b. 右上にある **Add** をクリックして、**Add sudo rule** ダイアログボックスを開きます。
   c. **sudo** ルールの名前 **run_third-party-app_report** を入力します。



   d. **Add and Edit** をクリックします。
   e. ユーザーを指定します。
      i. **Who** セクションで、**Specified Users and Groups** のラジオボタンを選択します。

ii. User category the rule applies toのサブセクションで Add をクリックして、Add users into sudo rule "run_third-party-app_report" ダイアログボックスを開きます。

iii. Available 列の Add users into sudo rule "run_third-party-app_report" ダイアログボックスで、idm_user チェックボックスをオンにして、これを Prospective 列に移動します。



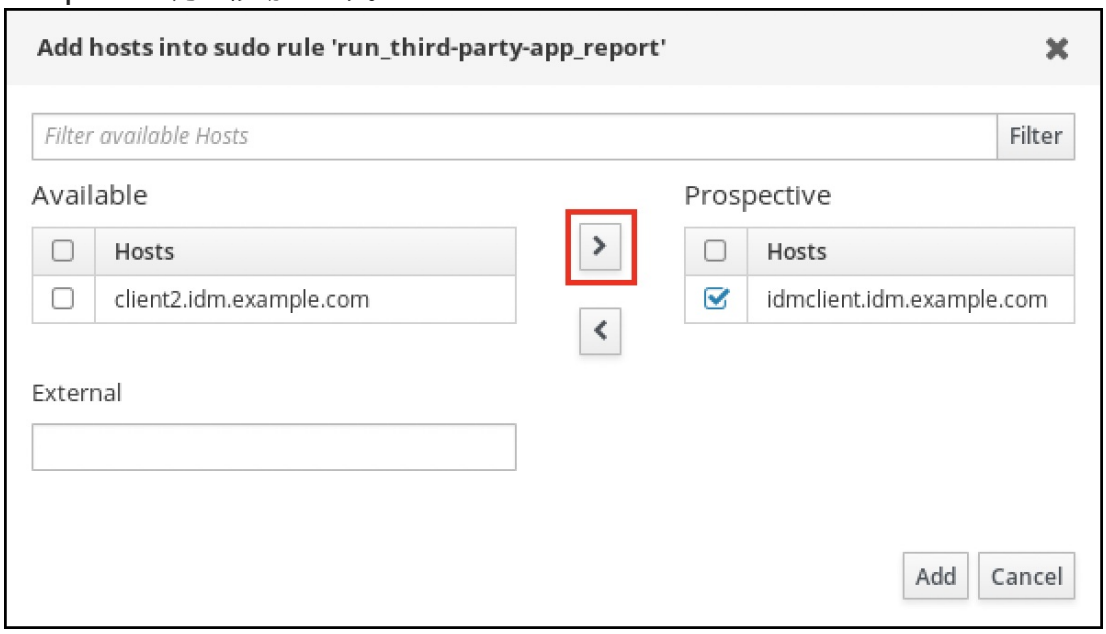
iv. Add をクリックします。

f. ホストを指定します。

i. Access this host セクションで、Specified Hosts and Groups ラジオボタンを確認します。

ii. Host category this rule applies to サブセクションで Add をクリックして、 Add hosts into sudo rule "run_third-party-app_report" ダイアログボックスを開きます。

iii. Available 列の Add hosts into sudo rule "run_third-party-app_report" ダイアログボックスで、idmclient.idm.example.com チェックボックスをオンにして、これを Prospective 列に移動します。



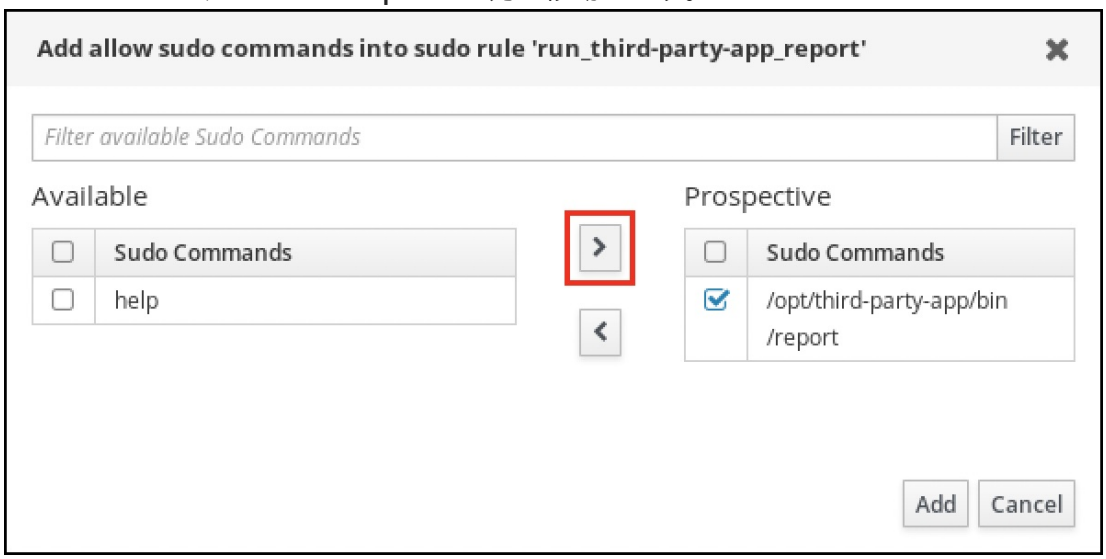
iv. Add をクリックします。

g. コマンドを指定します。

i. Run Commands セクションの Command category the rule applies to サブセクションで、Specified Commands and Groups ラジオボタンにチェックを入れます。

ii. Sudo Allow Commands サブセクションで Add をクリックして、Add allow sudo commands into sudo rule "run_third-party-app_report" ダイアログボックスを開きます。

iii. Available 列の Add allow sudo commands into sudo rule "run_third-party-app_report" ダイアログボックスで、**/opt/third-party-app/bin/report** チェックボックスをオンにして、これを Prospective 列に移動します。



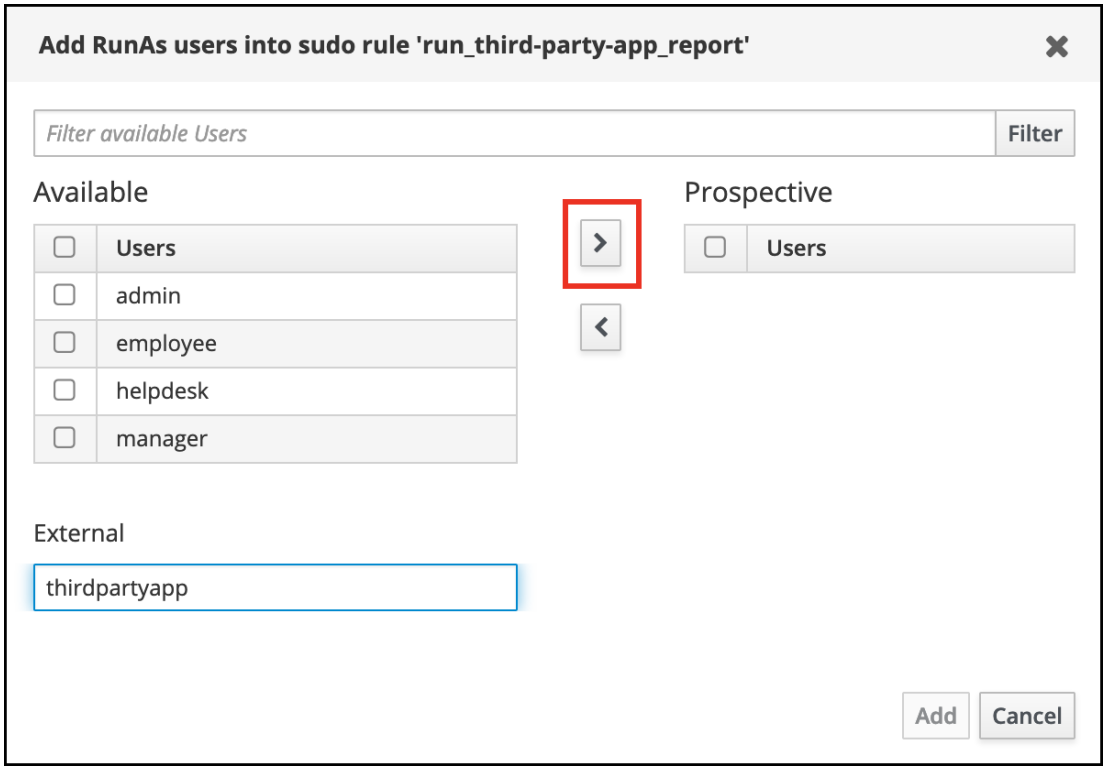
iv. Add をクリックして、run_third-party-app_report のページに戻ります。

h. RunAs ユーザーを指定します。

i. As Whom で、**指定したユーザーとグループ** のラジオボタンを確認します。

ii. RunAs **ユーザー** サブセクションで Add をクリックして、Add RunAs users into sudo rule "run_third-party-app_report" ダイアログボックスを開きます。

iii. Add RunAs users into sudo rule "run_third-party-app_report" ダイアログボックスで、External ボックスに **thirdpartyapp** サービスアカウントを入力し、これを Prospective 列に移動します。

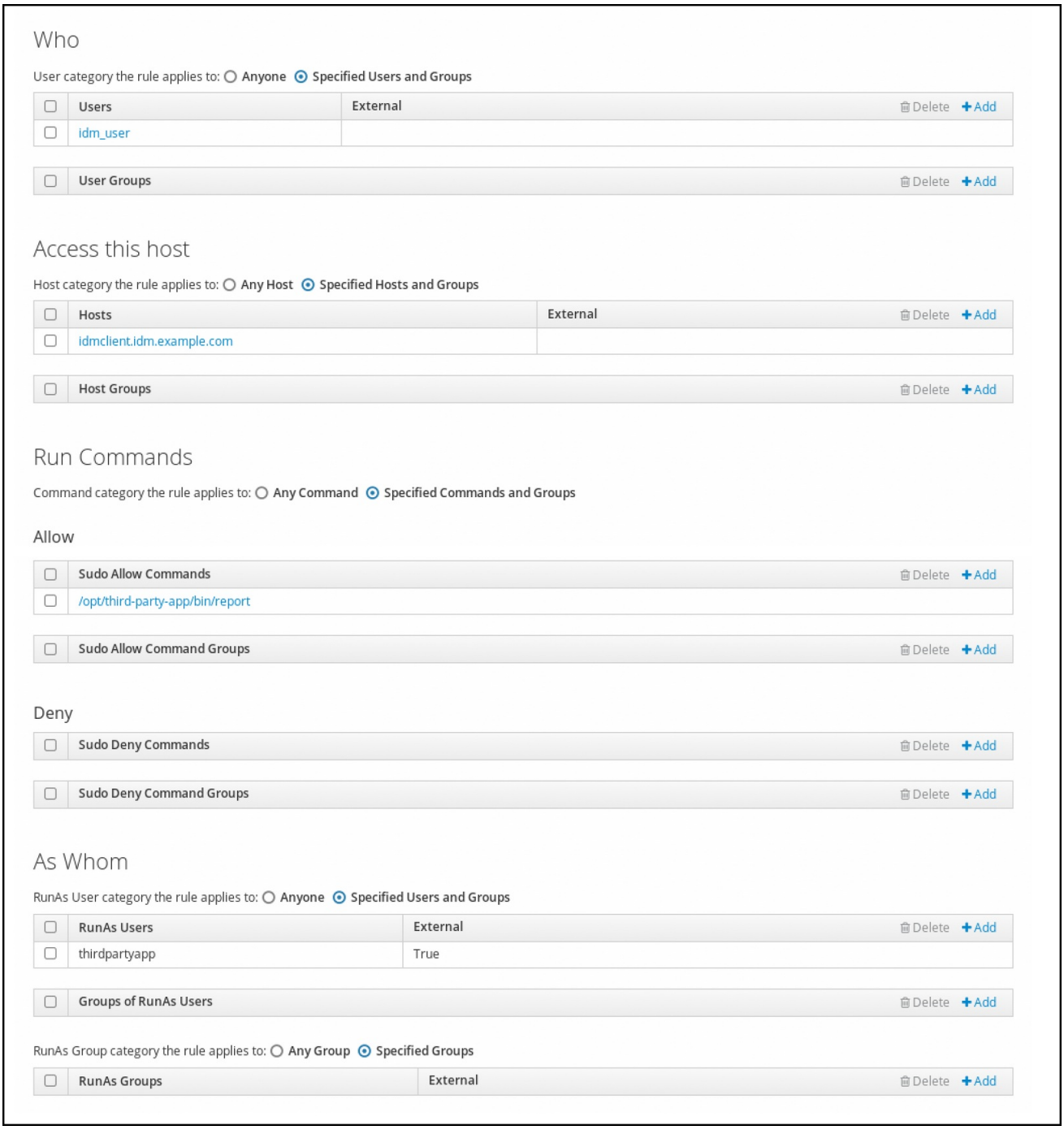


iv. **Add** をクリックして、**run_third-party-app_report** のページに戻ります。

i. 左上隅にある **Save** をクリックします。

新しいルールはデフォルトで有効になります。

図20.3 sudo ルールの詳細



注記

サーバーからクライアントへの変更の伝播には数分かかる場合があります。

検証手順

1. **idmclient** ホストに **idm_user** アカウントとしてログインします。
2. 新しい sudo ルールをテストします。
   a. **idm_user** アカウントが実行可能な **sudo** ルールを表示します。

```
[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. **report** コマンドを **thirdpartyapp** サービスアカウントとして実行します。

```
[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. IDM クライアントでの SUDO の GSSAPI 認証の有効化

以下の手順では、**pam_sss_gss.so** PAM モジュールを介して、**sudo** コマンドおよび **sudo -i** コマンドの IdM クライアントで、Generic Security Service Application Program Interface (GSSAPI) 認証を有効にする方法を説明します。この設定により、IdM ユーザーは Kerberos チケットを使用して **sudo** コマンドに対する認証が可能になります。

前提条件

- IdM ホストに適用する IdM ユーザーの **sudo** ルールを作成している。この例では、**idmclient** ホストで **/usr/sbin/reboot** コマンドを実行するパーミッションを **idm_user** アカウントに付与する **idm_user_reboot sudo** ルールが作成済みです。
- **/etc/sssd/sssd.conf** ファイルと、**/etc/pam.d/** ディレクトリーの PAM ファイルを変更するための **root** 特権がある。

手順

1. **/etc/sssd/sssd.conf** 設定ファイルを開きます。
2. **[domain/<domain_name>]** セクションに以下のエントリーを追加します。

```
[domain/<domain_name>]
pam_gssapi_services = sudo, sudo-i
```

3. **/etc/sssd/sssd.conf** ファイルを保存して閉じます。
4. SSSD サービスを再起動して、設定の変更を読み込みます。

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

5. RHEL 9.2 以降を実行している場合:

a. (オプション) **sssd authselect** プロファイルを選択したかどうかを確認します。

```
# authselect current
Profile ID: sssd
```

出力に、**sssd authselect** プロファイルが選択されていることが示されます。

b. **sssd authselect** プロファイルが選択されている場合は、GSSAPI 認証を有効にします。

```
# authselect enable-feature with-gssapi
```

c. **sssd authselect** プロファイルが選択されていない場合は、それを選択して GSSAPI 認証を有効にします。

```
# authselect select sssd with-gssapi
```

6. RHEL 9.1 以前を実行している場合:

a. **/etc/pam.d/sudo** の PAM 設定ファイルを開きます。

b. 以下のエントリーを、**/etc/pam.d/sudo** ファイルの **auth** セクションの最初の行に追加します。

```
#%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. **/etc/pam.d/sudo** ファイルを保存して閉じます。

検証手順

1. **idm_user** アカウントとしてホストにログインします。

```
[root@idm-client ~]# ssh -l idm_user@idm.example.com localhost
idm_user@idm.example.com's password:
```

2. **idm_user** アカウントで Ticket-Granting Ticket があることを確認します。

```
[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. **(オプション) idm_user** アカウントの Kerberos 認証情報がない場合は、現在の Kerberos 認証情報を削除し、正しい認証情報を要求します。

```
[idm_user@idmclient ~]$ kdestroy -A
```

```
[idm_user@idmclient ~]$ kinit idm_user@IDM.EXAMPLE.COM
Password for idm_user@idm.example.com:
```

4. パスワードを指定せずに **sudo** を使用してマシンを再起動します。

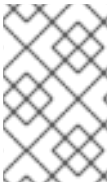
```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
```

関連情報

- IdM 用語 リストの GSSAPI エントリー
- IdM Web UI で IdM クライアントの IdM ユーザーへの sudo アクセスの許可
- CLI での IdM クライアントの IdM ユーザーへの sudo アクセス許可
- **pam_sss_gss(8)** の man ページ
- **sssd.conf (5)** の man ページ

20.8. IDM クライアントでの GSSAPI 認証の有効化および SUDO の KERBEROS 認証インジケーターの有効化

以下の手順では、**pam_sss_gss.so** PAM モジュールを介して、**sudo** コマンドおよび **sudo -i** コマンドの IdM クライアントで、Generic Security Service Application Program Interface (GSSAPI) 認証を有効にする方法を説明します。また、スマートカードを使用してログインしたユーザーのみが Kerberos チケットでこれらのコマンドに対して認証されます。



注記

この手順をテンプレートとして使用し、他の PAM 対応サービスに対して SSSD で GSSAPI 認証を設定して、さらに特定の認証インジケーターが Kerberos チケットにアタッチされているユーザーだけにアクセスを限定することができます。

前提条件

- IdM ホストに適用する IdM ユーザーの **sudo** ルールを作成している。この例では、**idmclient** ホストで **/usr/sbin/reboot** コマンドを実行するパーミッションを **idm_user** アカウントに付与する **idm_user_reboot sudo** ルールが作成済みです。
- **idmclient** ホストにスマートカード認証を設定している。
- **/etc/sssd/sssd.conf** ファイルと、**/etc/pam.d/** ディレクトリーの PAM ファイルを変更するための **root** 特権がある。

手順

1. **/etc/sssd/sssd.conf** 設定ファイルを開きます。
2. **[domain/<domain_name>]** セクションに以下のエントリーを追加します。

```
[domain/<domain_name>]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:pkinit, sudo-i:pkinit
```

3. **/etc/sssd/sssd.conf** ファイルを保存して閉じます。
4. SSSD サービスを再起動して、設定の変更を読み込みます。

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

5. **/etc/pam.d/sudo** の PAM 設定ファイルを開きます。
6. 以下のエントリーを、**/etc/pam.d/sudo** ファイルの **auth** セクションの最初の行に追加します。

```
#%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. **/etc/pam.d/sudo** ファイルを保存して閉じます。
8. **/etc/pam.d/sudo-i** の PAM 設定ファイルを開きます。
9. 以下のエントリーを、**/etc/pam.d/sudo-i** ファイルの **auth** セクションの最初の行に追加します。

```
#%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. **/etc/pam.d/sudo-i** ファイルを保存して閉じます。

検証手順

1. **idm_user** アカウントとしてホストにログインし、スマートカードで認証します。

```
[root@idmclient ~]# ssh -l idm_user@idm.example.com localhost
PIN for smart_card
```

2. スマートカードユーザーを使用して Ticket-Granting Ticket があることを確認します。

```
[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. **idm_user** アカウントが実行可能な **sudo** ルールを表示します。

```
[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. パスワードを指定せずに **sudo** を使用してマシンを再起動します。

```
[idm_user@idmclient ~]$ sudo /usr/sbin/reboot
```

関連情報

- PAM サービスの GSSAPI 認証を制御する SSSD オプション
- IdM 用語 リストの GSSAPI エントリー
- スマートカード認証用の Identity Management の設定
- Kerberos 認証インジケーター
- IdM Web UI で IdM クライアントの IdM ユーザーへの sudo アクセスの許可
- CLI での IdM クライアントの IdM ユーザーへの sudo アクセス許可
- **pam_sss_gss(8)** の man ページ
- **sssd.conf (5)** の man ページ

20.9. PAM サービスの GSSAPI 認証を制御する SSSD オプション

/etc/sssd/sssd.conf 設定ファイルに以下のオプションを使用すると、SSSD サービス内の GSSAPI 設定を調整できます。

pam_gssapi_services

SSSD を使用した GSSAPI 認証はデフォルトで無効になっています。このオプションを使用すると、PAM モジュール **pam_sss_gss.so** を使用して GSSAPI 認証を試すことができる PAM サービスをコンマ区切りのリストで指定できます。GSSAPI 認証を明示的に無効にするには、このオプションを **-** に設定します。

pam_gssapi_indicators_map

このオプションは、Identity Management (IdM) ドメインにのみ適用されます。このオプションを使用して、サービスへの PAM のアクセスを付与するのに必要な Kerberos 認証インジケーターをリスト表示します。ペアの形式は **<PAM_service>:_<required_authentication_indicator>_** でなければなりません。
有効な認証インジケーターは以下のとおりです。

- **OTP** - 2 要素認証
- **radius** - RADIUS 認証
- **pkinit** - PKINIT、スマートカード、または証明書での認証
- **hardened** - 強化パスワード

pam_gssapi_check_upn

このオプションはデフォルトで有効となっており、**true** に設定されています。このオプションを有効にすると、SSSD サービスでは Kerberos 認証情報と同じユーザー名が必要になります。**false** の場合には、**pam_sss_gss.so** の PAM モジュールは、必要なサービスチケットを取得できるすべてのユーザーを認証します。

例

次のオプションでは、**sudo** と **sudo-i** サービスの Kerberos 認証を有効にします。この認証では、**sudo** ユーザーはワンタイムパスワードで認証する必要があり、ユーザー名と Kerberos プリンシパルが同じでなければなりません。この設定は **[pam]** セクションにあるため、すべてのドメインに適用されます。

```
[pam]
pam_gssapi_services = sudo, sudo-i
pam_gssapi_indicators_map = sudo:otp
pam_gssapi_check_upn = true
```

これらのオプションを個別の **[domain]** セクションで設定して、**[pam]** セクションのグローバル値を上書きすることもできます。次のオプションは、異なる GSSAPI 設定を各ドメインに適用します。

idm.example.com ドメインの場合

- **sudo** と **sudo -i** サービスの GSSAPI 認証を有効にする。
- **sudo** コマンドには、証明書またはスマートカード認証オーセンティケーターが必要である。
- **sudo -i** コマンドにはワンタイムパスワード認証が必要である。
- ユーザー名と Kerberos プリンシパルを常に一致させる必要がある。

ad.example.com ドメインの場合

- **sudo** サービスに対してのみ GSSAPI 認証を有効にする。
- ユーザー名とプリンシパルを強制的に一致させない。

```
[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
...
```

関連情報

- Kerberos 認証インジケーター

20.10. SUDO の GSSAPI 認証のトラブルシューティング

IdM から Kerberos チケットを使用して **sudo** サービスに対する認証できない場合は、以下のシナリオを使用して設定のトラブルシューティングを行います。

前提条件

- **sudo** サービスの GSSAPI 認証が有効化されている。IdM クライアントでの sudo の GSSAPI 認証の有効化 を参照してください。
- **/etc/sssd/sssd.conf** ファイルと、**/etc/pam.d/** ディレクトリーの PAM ファイルを変更するための **root** 特権がある。

手順

- 以下のエラーが表示された場合、Kerberos サービスはホスト名をもとに、サービスチケットに合わせて正しいレルムを解決できない可能性があります。

```
Server not found in Kerberos database
```

このような場合は、**/etc/krb5.conf** の Kerberos 設定ファイルの **[domain_realm]** セクションにホスト名を直接追加します。

```
[idm-user@idm-client ~]$ cat /etc/krb5.conf
...

[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
 server.example.com = EXAMPLE.COM
```

- 以下のエラーが表示される場合には、Kerberos 認証情報がありません。

```
No Kerberos credentials available
```

このような場合は、**kinit** ユーティリティーを使用して Kerberos 認証情報を取得するか、SSSD で認証します。

```
[idm-user@idm-client ~]$ kinit idm-user@IDM.EXAMPLE.COM
Password for idm-user@idm.example.com:
```

- **/var/log/sssd/sssd_pam.log** ログファイルに以下のエラーのいずれかが表示される場合には、Kerberos 認証情報と、現在ログインしたユーザーのユーザー名とが一致しません。

```
User with UPN [<UPN>] was not found.

UPN [<UPN>] does not match target user [<username>].
```

このような場合は、SSSD で認証されたことを確認するか、**/etc/sssd/sssd.conf** ファイルで **pam_gssapi_check_upn** オプションを無効にすることを検討してください。

```
[idm-user@idm-client ~]$ cat /etc/sssd/sssd.conf
...

pam_gssapi_check_upn = false
```

- 他のトラブルシューティングを行う場合は、PAM モジュール **pam_sss_gss.so** のデバッグ出力を有効してください。
  - **/etc/pam.d/sudo** や **/etc/pam.d/sudo-i** など、PAM ファイルに **pam_sss_gss.so** の全エントリーの最後に **debug** オプションを追加します。

```
[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
```

  - **pam_sss_gss.so** モジュールで認証を試み、コンソールの出力を確認します。この例では、ユーザーには Kerberos 認証情報がありません。

```
[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. ANSIBLE PLAYBOOK を使用して IDM クライアントでの IDM ユーザーの SUDO アクセスを確認する

Identity Management (IdM) では、特定の IdM ホストの IdM ユーザーアカウントに **sudo** アクセスが付与されるようにできます。

この手順では、**idm_user_reboot** という名前の **sudo** ルールが存在するように設定します。このルールは、**idmclient** マシンで **/usr/sbin/reboot** コマンドを実行するパーミッションを **idm_user** に付与します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM に **idm_user** のユーザーアカウントが存在することを確認し、そのユーザーのパスワードを作成してアカウントのロックを解除している。コマンドラインインターフェイスを使用して新しい IdM ユーザーを追加する方法の詳細は、コマンドラインを使用したユーザーの追加 を参照してください。
- **idmclient** にローカルの **idm_user** アカウントがない。(**idm_user** ユーザーは **idmclient** の **/etc/passwd** ファイルに表示されていない)。

手順

1. **inventory.file** などのインベントリーファイルを作成し、そこに **ipaservers** を定義します。

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

2. **sudo** コマンドを 1 つまたは複数追加します。
   a. **ensure-reboot-sudocmd-is-present.yml** Ansible Playbook を作成し、**sudo** コマンドの IdM データベースに **/usr/sbin/reboot** コマンドが存在するようにします。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/sudocmd/ensure-sudocmd-is-present.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
- 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. Playbook を実行します。

```
$ 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. コマンドを参照する **sudo** ルールを作成します。

a. **sudo** コマンドエントリーを使用して sudo ルールが存在することを確認する **ensure-sudorule-for-idmuser-on-idmclient-is-present.yml** Ansible Playbook を作成します。sudo ルールは、 **idm_user** が **idmclient** マシンを再起動することを許可します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/sudorule/ensure-sudorule-is-present.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
- 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. Playbook を実行します。

```
$ ansible-playbook -v -i path_to_inventory_directory/inventory.file
path_to_playbooks_directory/ensure-sudorule-for-idmuser-on-idmclient-is-
present.yml
```

検証手順

idm_user が **sudo** を使用して **idmclient** を再起動できることを確認し、IdM サーバーに存在するように設定した **sudo** ルールが **idmclient** で機能することをテストします。サーバーに加えられた変更がクライアントで反映されるまで数分かかる場合があります。

1. **idmclient** に **idm_user** としてログインします。

2. **sudo** を使用してマシンを再起動します。プロンプトが表示されたら、**idm_user** のパスワードを入力します。

```
$ sudo /usr/sbin/reboot
[sudo] password for idm_user:
```

sudo が正しく設定されている場合には、マシンが再起動します。

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-sudocmd.md** ファイル、**README-sudocmdgroup.md** ファイル、および **README-sudorule.md** ファイルを参照してください。

第21章 ANSIBLE PLAYBOOK を使用して IDM にホストベースのアクセス制御ルールを存在させる手順

Ansible は、システムの設定、ソフトウェアのデプロイ、ローリング更新の実行に使用する自動化ツールです。これには、Identity Management (IdM) のサポートが含まれます。

Identity Management (IdM) ホストベースのアクセスポリシーと、Ansible を使用してそれを定義する方法を説明します。

21.1. IDM のホストベースのアクセス制御ルール

ホストベースのアクセス制御 (HBAC) ルールは、サービスグループ内のサービスを使用して、どのユーザーまたはグループがどのホストまたはホストグループにアクセスできるかを定義します。システム管理者は、HBAC ルールを使用して以下の目的を達成できます。

- 指定のユーザーグループのメンバーだけがドメイン内の特定のシステムにアクセスできるように制限する。
- ドメイン内のシステムにアクセスする時に特定のサービスだけを使用できるようにする。

デフォルトでは、IdM は **allow_all** という名前のデフォルトの HBAC ルールで設定されます。この設定では、どのユーザーでも関連のあるサービスをどれでも使用して IdM ドメイン全体にあるすべてのホストに普遍的にアクセスできます。

デフォルトの **allow_all** ルールを独自の HBAC ルールセットに置き換えることで、さまざまなホストへのアクセスを微調整できます。個別のユーザー、ホスト、またはサービスではなく、ユーザーグループ、ホストグループ、またはサービスグループに HBAC ルールを適用して、アクセス制御管理を集中化および簡素化できます。

21.2. ANSIBLE PLAYBOOK を使用して IDM に HBAC ルールを存在させる手順

以下の手順に従って、Ansible Playbook を使用して Identity Management (IdM) にホストベースのアクセス制御 (HBAC) ルールが存在することを確認します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

- HBAC ルールに使用するユーザーとユーザーグループが IdM に存在する。詳細は、Ansible Playbook を使用したユーザーアカウントの管理 および Ansible Playbook を使用して IdM グループおよびグループメンバーの存在を確認する を参照してください。
- HBAC ルールを適用するホストおよびホストグループが IdM に存在する。詳細は、Ansible Playbook を使用したホストの管理 および Ansible Playbook を使用したホストグループの管理 を参照してください。

手順

1. **inventory.file** などのインベントリーファイルを作成して、そのファイルに **ipaserver** を定義します。

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

2. Ansible Playbook ファイルを作成して、存在させる HBAC ポリシーを定義します。この手順は、**/usr/share/doc/ansible-freeipa/playbooks/hbacrule/ensure-hbacrule-allhosts-present.yml** ファイルのサンプルをコピーして変更し、簡素化できます。

```
---
- 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. Playbook を実行します。

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

検証手順

1. 管理者として IdM Web UI にログインします。
2. **Policy → Host-Based-Access-Control → HBAC Test** の順に選択します。
3. **Who** タブで idm_user を選択します。
4. **Accessing** タブで **client.idm.example.com** を選択します。
5. **Via service** タブで **sshd** を選択します。

6. **Rules** タブで **login** を選択します。

7. **Run test** タブで **Run test** ボタンをクリックします。ACCESS GRANTED が表示されると、HBAC ルールが正常に実装されています。

関連情報

- **/usr/share/doc/ansible-freeipa** ディレクトリーの **README-hbacsvc.md** ファイル、**README-hbacsvcgroup.md** ファイル、および **README-hbacrule.md** ファイルを参照してください。

- **/usr/share/doc/ansible-freeipa/playbooks** ディレクトリーのサブディレクトリーにある Playbook を参照してください。

第22章 ANSIBLE を使用した IDM 証明書の管理

ansible-freeipa ipacert モジュールを使用して、Identity Management (IdM) ユーザー、ホスト、およびサービスの SSL 証明書を要求、取り消し、および取得できます。保留された証明書を復元することもできます。

22.1. ANSIBLE を使用した IDM ホスト、サービス、ユーザーの SSL 証明書の要求

ansible-freeipa ipacert モジュールを使用して、Identity Management (IdM) ユーザー、ホスト、およびサービスの SSL 証明書を要求できます。その後、これらの証明書を使用して IdM に対する認証を行うことができます。

Ansible Playbook を使用して IdM 認証局 (CA) から HTTP サーバーの証明書を要求するには、この手順を完了します。

前提条件

- コントロールノード:
  - Ansible バージョン 2.14 以降を使用している。
  - **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
  - **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。
- IdM デプロイメントに統合 CA がある。

手順

1. ユーザー、ホスト、またはサービスの証明書署名要求 (CSR) を生成します。たとえば、**openssl** ユーティリティーを使用して client.idm.example.com で実行されている **HTTP** サービスの CSR を生成するには、次のように入力します。

   ```
   # openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout new.key -out new.csr -subj '/CN=client.idm.example.com,O=IDM.EXAMPLE.COM'
   ```

   その結果、CSR は **new.csr** に保存されます。

2. 次の内容を含む Ansible Playbook ファイル **request-certificate.yml** を作成します。

   ```
   ---
   - name: Playbook to request a certificate
     hosts: ipaserver

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

     tasks:
     - name: Request a certificate for a web server
       ipacert:
         ipaadmin_password: "{{ ipaadmin_password }}"
   ```

```
    state: requested
    csr: |
      -----BEGIN CERTIFICATE REQUEST-----

MIGYMEwCAQAwGTEXMBUGA1UEAwwOZnJlZWlwYSBydWxlcyEwKjAFBgMrZXADIQBs
      HlqIr4b/XNK+K8QLJKIzfvuNK0buBhLz3LAzY7QDEqAAMAUGAytlcANBAF4oSCbA
      5aIPukCidnZJdr491G4LBE+URecYXsPknwYb+V+ONnf5ycZHyaFv+jkUBFGFeDgU
      SYaXm/gF8cDYjQI=
      -----END CERTIFICATE REQUEST-----
    principal: HTTP/client.idm.example.com
  register: cert
```

証明書要求を **new.csr** の CSR に置き換えます。

3. 証明書を要求します。

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

関連情報

- **ansible-freeipa** アップストリームドキュメントの cert モジュール

22.2. ANSIBLE を使用して IDM ホスト、サービス、ユーザーの SSL 証明書を取り消す

ansible-freeipa ipacert モジュールを使用して、Identity Management (IdM) ユーザー、ホスト、およびサービスが IdM への認証に使用する SSL 証明書を取り消すことができます。

Ansible Playbook を使用して HTTP サーバーの証明書を取り消すには、この手順を完了します。証明書を取り消す理由は “keyCompromise” です。

前提条件

- コントロールノード:
  - Ansible バージョン 2.14 以降を使用している。
  - **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
  - **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。
  - **openssl x509 -noout -text -in <path_to_certificate>** コマンドを入力するなどして、証明書のシリアル番号を取得している。この例では、証明書のシリアル番号は 123456789 です。
- IdM デプロイメントに統合 CA がある。

手順

1. 次の内容を含む Ansible Playbook ファイル **revoke-certificate.yml** を作成します。

```
---
- name: Playbook to revoke a certificate
  hosts: ipaserver

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

  tasks:
  - name: Revoke a certificate for a web server
    ipacert:
      ipaadmin_password: "{{ ipaadmin_password }}"
      serial_number: 123456789
      revocation_reason: "keyCompromise"
      state: revoked
```

2. 証明書を取り消します。

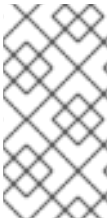
```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/revoke-
certificate.yml
```

関連情報

- **ansible-freeipa** アップストリームドキュメントの cert モジュール
- RFC 5280 の Reason Code

22.3. ANSIBLE を使用して IDM ユーザー、ホスト、およびサービスの SSL 証明書を復元する

ansible-freeipa ipacert モジュールを使用すると、Identity Management (IdM) ユーザー、ホスト、またはサービスが IdM への認証に以前に使用した、取り消された SSL 証明書を復元できます。



注記

復元できるのは、保留された証明書のみです。たとえば、秘密キーを紛失したかどうかがわからないなどの理由で、秘密キーを保留した可能性があります。ただし、キーを回復し、回復までの間に誰もそのキーにアクセスしていないと確信しているため、証明書を復元したいと考えています。

Ansible Playbook を使用して、IdM に登録されたサービスの証明書を保留から解放するには、この手順を完了します。この例では、HTTP サービスの証明書の保留を解除する方法について説明します。

前提条件

- コントロールノード:
  - Ansible バージョン 2.14 以降を使用している。
  - **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。

- **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。

- IdM デプロイメントに統合 CA がある。
- たとえば、**openssl x509 -noout -text -in path/to/certificate** コマンドを入力するなどして、証明書のシリアル番号を取得している。この例では、証明書のシリアル番号は **123456789** です。

手順

1. 次の内容を含む Ansible Playbook ファイル **restore-certificate.yml** を作成します。

```
---
- name: Playbook to restore a certificate
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Restore a certificate for a web service
    ipacert:
      ipaadmin_password: "{{ ipaadmin_password }}"
      serial_number: 123456789
      state: released
```

2. Playbook を実行します。

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

関連情報

- **ansible-freeipa** アップストリームドキュメントの cert モジュール

22.4. ANSIBLE を使用して IDM ユーザー、ホスト、およびサービスの SSL 証明書を取得する

ansible-freeipa ipacert モジュールを使用すると、Identity Management (IdM) ユーザー、ホスト、またはサービスに対して発行された SSL 証明書を取得し、マネージドノード上のファイルに保存できます。

前提条件

- コントロールノード:
  - Ansible バージョン 2.14 以降を使用している。
  - **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
  - **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。

- **openssl x509 -noout -text -in <path_to_certificate>** コマンドを入力するなどして、証明書のシリアル番号を取得している。この例では、証明書のシリアル番号は 123456789 で、取得した証明書を保存するファイルは **cert.pem** です。

手順

1. 次の内容を含む Ansible Playbook ファイル **retrieve-certificate.yml** を作成します。

```
---
- name: Playbook to retrieve a certificate and store it locally on the managed node
  hosts: ipaserver

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

  tasks:
  - name: Retrieve a certificate and save it to file 'cert.pem'
    ipacert:
      ipaadmin_password: "{{ ipaadmin_password }}"
      serial_number: 123456789
      certificate_out: cert.pem
      state: retrieved
```

2. 証明書を取得します。

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

関連情報

- **ansible-freeipa** アップストリームドキュメントの cert モジュール

第23章 IDM の VAULT

本章では、Identity Management(IdM) の vault について説明します。本章では、以下のトピックを紹介します。

- vault の概念。
- vault に関連付けられる各種ロール。
- IdM で利用可能な各種 vault - セキュリティーおよびアクセス制御のレベル別 。
- IdM で利用可能な各種 vault - 所有権別 。
- vault コンテナーの概念。
- IdM での vault 管理向けの基本的なコマンド 。
- IdM で vault を使用するのに必要な KPA (Key Recovery Authority) のインストール 。

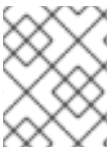
23.1. VAULT およびその利点

vault は、機密データをすべてセキュアに保存しつつも、1 箇所で都合よく Identity Management (IdM) を使用するのに便利な機能です。 vault にはさまざまなタイプがあり、要件に応じて使用する vault を選択する必要があります。

vault とは、シークレットの保存、取得、共有、および復旧を行うための (IdM の) セキュアな場所を指し、シークレットは、通常は一部のユーザーまたはエンティティーグループのみがアクセスできる、認証情報などの機密データを指します。たとえば、シークレットには以下が含まれます。

- パスワード
- 暗証番号
- SSH 秘密鍵

vault はパスワードマネージャーと類似しています。valut を使用する場合、通常、パスワードマネージャーと同様に、ロックを解除するためのプライマリーのパスワードを 1 つ生成し、記憶して、vault に保存されている情報にアクセスする必要があります。ただし、標準の vault を指定することも可能です。標準の vault では、vault に保存されているシークレットにアクセスするためにパスワードを入力する必要はありません。



注記

IdM の vault は、認証情報を保存して、IdM 関連以外の外部サービスに対して認証を可能にすることを目的としています。

IdM vault には他にも、次のような重要な特徴があります。

- vault にアクセスできるのは、vault の所有者と、vault メンバーとして vault の所有者が選択した IdM ユーザーだけです。また、IdM 管理者も vault にアクセスできます。
- ユーザーに vault を作成する権限がない場合には、IdM 管理者が vault を作成し、そのユーザーを所有者として設定できます。

- ユーザーおよびサービスは、IdM ドメインに登録されているマシンからであれば、vault に保存されているシークレットにアクセスできます。
- vault 1 つに追加できるシークレットは 1 つのみです (例: ファイル 1 つ)。ただし、ファイル自体には、パスワード、キータブ、証明書など複数のシークレットを含めることができます。

注記

Vault は、IdM Web UI ではなく、IdM コマンドライン (CLI) からしか利用できません。

23.2. VAULT の所有者、メンバー、および管理者

Identity Management (IdM) で識別される vault ユーザータイプは以下のとおりです。

Vault 所有者

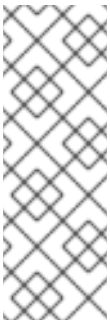
vault 所有者は、vault の基本的な管理権限のあるユーザーまたはサービスです。たとえば、vault の所有者は vault のプロパティーを変更したり、新しい vault メンバーを追加したりできます。
各 vault には最低でも所有者が 1 人必要です。vault には複数の所有者を指定することもできます。

Vault メンバー

vault メンバーは、別のユーザーまたはサービスが作成した vault にアクセスできるユーザーまたはサービスです。

Vault 管理者

vault 管理者は全 vault に制限なくアクセスでき、vault の操作をすべて実行できます。



注記

対称と非対称 vault は、パスワードまたは鍵で保護されており、特別なアクセス制御ルールが適用されます (Vault タイプ を参照)。管理者は、以下を行うためにこの特別なルールを満たす必要があります。

- 対称および非対称 vault のシークレットにアクセスする。
- vault パスワードまたはキーを変更またはリセットする。

vault 管理者は、**vault administrators** 特権を持つユーザーです。IdM のロールベースアクセス制御 (RBAC) のコンテキストでの特権とは、ロールに適用できるパーミッションのグループのことです。

Vault ユーザー

vault ユーザーは、vault のあるコンテナー内のユーザーです。**Vault ユーザー** 情報は、**ipa vault-show** などの特定のコマンドの出力に表示されます。

```
$ ipa vault-show my_vault
  Vault name: my_vault
  Type: standard
  Owner users: user
  Vault user: user
```

vault コンテナーおよびユーザー vault の詳細は、Vault コンテナー を参照してください。

関連情報

- vault のタイプの詳細は、標準、対称および非対称 vault を参照してください。

23.3. 標準、対称および非対称 VAULT

IdM では、セキュリティーおよびアクセス制御のレベルをもとに vault を以下のタイプに分類します。

標準 vault

Vault の所有者と vault メンバーは、パスワードやキーを使用せずにシークレットをアーカイブして取得できます。

対称 vault

vault のシークレットは対称キーを使用して保護されます。Vault の所有者とメンバーは、シークレットをアーカイブして取得できますが、vault パスワードを指定する必要があります。

非対称 vault

vault のシークレットは非対称キーを使用して保護されます。ユーザーは公開鍵でシークレットをアーカイブし、秘密鍵でシークレットを取得します。vault メンバーはシークレットのアーカイブのみが可能ですが、vault 所有者はシークレットのアーカイブと取得の両方が可能です。

23.4. ユーザー、サービスおよび共有 VAULT

IdM では、所有権をもとに vault を複数のタイプに分類します。以下の表 には、各タイプ、所有者、および使用方法に関する情報が含まれます。

表23.1 所有権に基づく IdM vault

タイプ	説明	所有者	備考
ユーザー vault	ユーザーのプライベート vault	ユーザー x 1	IdM 管理者が許可すれば、ユーザーは 1 つまたは複数のユーザー vault を所有できます。
サービス vault	サービスのプライベート vault	サービス x 1	IdM 管理者が許可すれば、ユーザーは 1 つまたは複数のサービス vault を所有できます。
共有 vault	複数のユーザーおよびグループで共有される vault	vault を作成した vault の管理者	IdM 管理者が許可すれば、ユーザーおよびサービスは 1 つまたは複数のユーザー vault を所有できます。vault の作成者以外に、vault 管理者が vault に対して完全なアクセス権があります。

23.5. VAULT コンテナー

vault コンテナーは vault のコレクションです。以下の表 は、Identity Management (IdM) が提供するデフォルトの vault コンテナーのリストです。

表23.2 IdM のデフォルトの vault コンテナー

タイプ	説明	目的

タイプ	説明	目的
ユーザーコンテナー	ユーザーのプライベートコンテナー	特定ユーザーのユーザー vault を格納します。
サービスコンテナー	サービスのプライベートコンテナー	特定のサービスのサービス vault を格納します。
共有コンテナー	複数のユーザーおよびサービスのコンテナー	複数のユーザーまたはサービスで共有可能な vault を格納します。

IdM では、ユーザーまたはサービスのプライベート vault が初めて作成されると、ユーザーまたはサービスごとにユーザーコンテナーおよびサービスコンテナーを自動的に作成します。ユーザーまたはサービスが削除されると、IdM はコンテナーとそのコンテンツを削除します。

23.6. 基本的な IDM VAULT コマンド

以下に概説する基本的なコマンドを使用して、Identity Management (IdM) vault を管理できます。以下の表 には、**ipa vault-*** コマンドとその目的が記載されています。

注記

ipa vault-* コマンドを実行する前に、IdM ドメインのサーバー 1 台以上に Key Recovery Authority (KRA) 証明書システムコンポーネントをインストールします。詳細は IdM での Key Recovery Authority (KRA) のインストール を参照してください。

表23.3 基本的な IdM vault コマンドおよび説明

コマンド	目的
ipa help vault	IdM vault コマンドおよびサンプル vault コマンドの概念などの情報を表示します。
ipa vault-add --help、ipa vault-find --help	特定の **ipa vault-*** コマンドに **--help** オプションを追加すると、このコマンドで利用可能なオプションと詳細なヘルプが表示されます。

ipa vault-show user_vault --user idm_user	vault メンバーとして vault にアクセスする場合は、vault 所有者を指定する必要があります。vault 所有者を指定しない場合には、IdM により vault が見つからない旨が通知されます。
	`[admin@server ~]$ ipa vault-show user_vault`
	`ipa: ERROR: user_vault: vault not found`

コマンド	目的
ipa vault-show shared_vault --shared	共有 vault にアクセスする場合には、アクセスする vault が共有 vault であることを指定する必要があります。それ以外の場合は、IdM により vault が見つからない旨が通知されます。 [admin@server ~]$ ipa vault-show shared_vault ipa: ERROR: shared_vault: vault not found

23.7. IDM での KEY RECOVERY AUTHORITY (KRA) のインストール

以下の手順に従って、特定の IdM サーバーに Key Recovery Authority (KRA) Certificate System (CS) コンポーネントをインストールして、Identity Management (IdM) の vault を有効にします。

前提条件

- IdM サーバーに **root** としてログインしている。
- IdM 認証局が IdM サーバーにインストールされている。
- **Directory Manager** 認証情報がある。

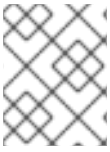
手順

- KRA をインストールします。

```
# ipa-kra-install
```

重要

非表示のレプリカに、IdM クラスターの最初の KRA をインストールできます。ただし、追加の KRA をインストールするには、非表示レプリカを一時的にアクティベートしてから、表示されているレプリカに KRA のクローンをインストールする必要があります。その後に、最初に非表示レプリカを再度非表示にできます。



注記

vault サービスの可用性と耐障害性を高めるには、2 台以上の IdM サーバーに KRA をインストールします。複数の KRA サーバーを保持することで、データの損失を防ぎます。

関連情報

- Demoting or promoting hidden replicas を参照してください。
- 非表示のレプリカモード を参照してください。

第24章 ANSIBLE を使用した IDM ユーザー VAULT の管理: シークレットの保存および取得

本章では、Ansible **vault** モジュールを使用して Identity Management でユーザー vault を管理する方法を説明します。具体的には、ユーザーが Ansible Playbook を使用して以下の 3 つのアクションを実行する方法を説明しています。

- IdM でユーザーコンテナーを作成する。
- シークレットを vault に保存する。
- vault からシークレットを取得する。

異なる IdM クライアント 2 台から保存と取得が可能です。

前提条件

- Key Recovery Authority (KRA) Certificate System コンポーネントが IdM ドメインの 1 つ以上のサーバーにインストールされている。詳細は IdM での Key Recovery Authority (KRA) のインストール を参照してください。

24.1. ANSIBLE を使用して IDM に標準ユーザー VAULT を存在させる手順

以下の手順に従って、Ansible Playbook を使用して 1 つ以上のプライベート vault を持つ vault コンテナーを作成し、機密情報を安全に保存します。以下の手順で使用する例では、**idm_user ユーザー** は **my_vault** という名前の標準タイプの vault を作成します。標準タイプの vault では、ファイルへのアクセス時に **idm_user** を認証する必要がありません。**IdM_user** は、ユーザーがログインしている IdM クライアントからファイルを取得できます。

前提条件

- Ansible コントローラー (手順の内容を実行するホスト) に ansible-freeipa パッケージがインストールされている。
- **idm_user** のパスワードを知っている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. **inventory.file** などのインベントリーファイルを作成します。

```
$ touch inventory.file
```

3. **inventory.file** を開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

4. Ansible Playbook の **ensure-standard-vault-is-present.yml** ファイルのコピーを作成します。以下に例を示します。

```
$ cp ensure-standard-vault-is-present.yml ensure-standard-vault-is-present-copy.yml
```

5. **ensure-standard-vault-is-present-copy.yml** ファイルを開いて編集します。

6. **ipavault** タスクセクションに以下の変数を設定して、ファイルを調整します。

- **ipaadmin_principal** 変数は idm_user に設定します。
- **ipaadmin_password** 変数は idm_user のパスワードに設定します。
- **user** 変数は idm_user に設定します。
- **name** 変数は my_vault に設定します。
- **vault_type** 変数は standard に設定します。
  今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。

8. Playbook を実行します。

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

24.2. ANSIBLE を使用して IDM の標準ユーザー VAULT でシークレットをアーカイブする手順

以下の手順に従って、Ansible Playbook を使用してパーソナル vault に機密情報を保存します。この例では、**idm_user** ユーザーは **my_vault** という名前の vault に **password.txt** という名前で機密情報が含まれるファイルをアーカイブします。

前提条件

- Ansible コントローラー (手順の内容を実行するホスト) に ansible-freeipa パッケージがインストールされている。

- idm_user のパスワードを知っている。
- IdM_user が所有者であるか、my_vault のメンバーユーザーである。
- password.txt (my_vault にアーカイブするシークレット) にアクセスできる。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook の **data-archive-in-symmetric-vault.yml** ファイルのコピーを作成して symmetric を standard に置き換えます。以下に例を示します。

```
$ cp data-archive-in-symmetric-vault.yml data-archive-in-standard-vault-copy.yml
```

4. **data-archive-in-standard-vault-copy.yml** ファイルを開いて編集します。
5. **ipavault** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_principal** 変数は idm_user に設定します。
   - **ipaadmin_password** 変数は idm_user のパスワードに設定します。
   - **user** 変数は idm_user に設定します。
   - **name** 変数は my_vault に設定します。
   - **in** 変数は機密情報が含まれるファイルへのパスに設定します。
   - **action** 変数は member に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-archive-in-standard-vault-copy.yml
```

24.3. ANSIBLE を使用して IDM の標準ユーザー VAULT からシークレットを取得する手順

以下の手順に従って、Ansible Playbook を使用してユーザーのパーソナル vault からシークレットを取得します。以下の手順で使用する例では、**idm_user** ユーザーは、**my_vault** という名前の標準タイプの vault から機密データを含むファイルを取得して、**host01** という名前の IdM クライアントに配置します。ファイルへのアクセス時に **IdM_user** を認証する必要はありません。**IdM_user** は Ansible を使用して、Ansible がインストールされている IdM クライアントからファイルを取得できます。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **idm_user** のパスワードを知っている。
- **IdM_user** が **my_vault** の所有者である。
- **IdM_user** が **my_vault** にシークレットを保存している。
- Ansible が、シークレットを取得して配置する先の IdM ホストのディレクトリーに書き込みができる。
- **IdM_user** はシークレットを取得して配置する先の IdM ホストのディレクトリーから読み取りができる。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. インベントリーファイルを開き、明確に定義されたセクションで、シークレットを取得する IdM クライアントを記載します。たとえば、Ansible に対して **host01.idm.example.com** にシークレットを取得して配置するよう指示するには、次のコマンドを実行します。

```
[ipahost]
host01.idm.example.com
```

3. Ansible Playbook ファイル (**retrive-data-symmetric-vault.yml**) のコピーを作成します。symbolicmetric を Standard に置き換えます。以下に例を示します。

```
$ cp retrive-data-symmetric-vault.yml retrieve-data-standard-vault.yml-copy.yml
```

4. **retrieve-data-standard-vault.yml-copy.yml** ファイルを開いて編集します。
5. **hosts** 変数は **ipahost** に設定して、ファイルを調整します。
6. **ipavault** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_principal** 変数は **idm_user** に設定します。
   - **ipaadmin_password** 変数は **idm_user** のパスワードに設定します。
   - **user** 変数は **idm_user** に設定します。
   - **name** 変数は **my_vault** に設定します。
   - **out** 変数は、シークレットをエクスポートするファイルの完全パスに設定します。
   - **state** 変数は **retrieved** に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。
8. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-data-standard-vault.yml-copy.yml
```

検証手順

1. **host01** に **user01** として **SSH** 接続します。

```
$ ssh user01@host01.idm.example.com
```

2. Ansible Playbook ファイルに **out** 変数で指定したファイルを表示します。

```
$ vim /tmp/password_exported.txt
```

これで、エクスポートされたシークレットが表示されます。

- Ansible を使用して IdM vault およびユーザーシークレットを管理する方法および、Playbook 変数の情報は、**/usr/share/doc/ansible-freeipa/** ディレクトリーで利用可能な README-vault.md Markdown ファイルおよび **/usr/share/doc/ansible-freeipa/playbooks/vault/** で利用可能なサンプルの Playbook を参照してください。

第25章 ANSIBLE を使用した IDM サービス VAULT の管理: シークレットの保存および取得

本セクションでは、管理者が **ansible-freeipa vault** モジュールを使用してサービスシークレットを一元的にセキュアに保存する方法を説明します。この例で使用される vault は非対称であるため、これを使用する場合は、管理者は以下の手順を実行する必要があります。

1. **openssl** ユーティリティーなどを使用して秘密鍵を生成する。
2. 秘密鍵をもとに公開鍵を生成する。

サービスシークレットは、管理者が vault にアーカイブする時に公開鍵を使用して暗号化されます。その後、ドメイン内の特定のマシンでホストされるサービスインスタンスが、秘密鍵を使用してシークレットを取得します。シークレットにアクセスできるのは、サービスと管理者のみです。

シークレットが漏洩した場合には、管理者はサービス Vault でシークレットを置き換えて、漏洩されていないサービスインスタンスに配布しなおすことができます。

前提条件

- Key Recovery Authority (KRA) Certificate System コンポーネントが IdM ドメインの 1 つ以上のサーバーにインストールされている。詳細は IdM での Key Recovery Authority (KRA) のインストール を参照してください。

このセクションでは、以下の手順について説明します。

- Ansible を使用して IdM に非対称サービス vault を存在させる手順
- Ansible を使用した非対称 vault への IdM サービスシークレットの保存
- Ansible を使用した IdM サービスのサービスシークレットの取得
- シークレットが漏洩した場合の Ansible での IdM サービス vault シークレットの変更

本手順での以下の用語について説明します。

- **admin** は、サービスパスワードを管理する管理者です。
- **private-key-to-an-externally-signed-certificate.pem** は、サービスシークレットを含むファイルです (ここでは外部署名証明書への秘密鍵)。この秘密鍵と、vault からのシークレットの取得に使用する秘密鍵と混同しないようにしてください。
- **secret_vault** は、サービスシークレット保存向けに作成された vault です。
- **HTTP/webserver1.idm.example.com** は vault の所有者となるサービスです。
- **HTTP/webserver2.idm.example.com** および **HTTP/webserver3.idm.example.com** は vault メンバーサービスです。
- **service-public.pem** は、**password_vault** に保存されているパスワードの暗号化に使用するサービスの公開鍵です。
- **service-private.pem** は、**secret_vault** に保存されているパスワードの復号化に使用するサービスの秘密鍵です。

25.1. ANSIBLE を使用して IDM に非対称サービス VAULT を存在させる手順

以下の手順に従って、Ansible Playbook を使用して 1 つ以上のプライベート Vault を持つサービス vault コンテナーを作成し、機密情報を安全に保存します。以下の手順で使用する例では、管理者は **secret_vault** という名前の非対称 vault を作成します。こうすることで、vault メンバーは、vault のシークレットを取得する際に、秘密鍵を使用して必ず認証することになります。vault メンバーは、IdM クライアントからファイルを取得できます。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **IdM 管理者** パスワードが分かっている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. サービスインスタンスの公開鍵を取得します。たとえば、**openssl** ユーティリティーを使用する場合は以下を行います。
   a. **service-private.pem** 秘密鍵を生成します。

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....................................+++
e is 65537 (0x10001)
```

   b. 秘密鍵をもとに **service-public.pem** 公開鍵を生成します。

```
$ openssl rsa -in service-private.pem -out service-public.pem -pubout
writing RSA key
```

3. オプション: **inventory.file** など、存在しない場合はインベントリーファイルを作成します。

```
$ touch inventory.file
```

4. インベントリーファイルを開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

5. Ansible Playbook ファイル **ensure-asymmetric-vault-is-present.yml** のコピーを作成します。以下に例を示します。

```
$ cp ensure-asymmetric-vault-is-present.yml ensure-asymmetric-service-vault-is-present-copy.yml
```

6. **ensure-asymmetric-vault-is-present-copy.yml** ファイルを開いて編集します。

7. Ansible コントローラーから **server.idm.example.com** サーバーに **service-public.pem** の公開鍵をコピーするタスクを追加します。

8. **ipavault** タスクセクションに以下の変数を設定して、残りのファイルを変更します。

   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **secret_vault** などの **name** 変数を使用して vault の名前を定義します。
   - **vault_type** 変数は **asymmetric** に設定します。
   - **service** 変数は、vault を所有するサービスのプリンシパル (例: **HTTP/webserver1.idm.example.com**) に設定します。
   - **public_key_file** は、公開鍵の場所に設定します。
     以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

10. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-asymmetric-service-vault-is-present-copy.yml
```

25.2. ANSIBLE を使用した非対称 VAULT へのメンバーサービスの追加

以下の手順に従って、Ansible Playbook を使用してメンバーサービスをサービス vault に追加し、これらすべてが vault に保存されているシークレットを取得できるようにします。以下の手順で使用する例では、IdM の管理者は **HTTP/webserver2.idm.example.com** と **HTTP/webserver3.idm.example.com** のサービスプリンシパルを **HTTP/webserver1.idm.example.com** が所有する **secret_vault** vault に追加します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **IdM 管理者** パスワードが分かっている。
- サービスシークレットの保存先の 非対称 vault を作成している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. オプション: **inventory.file** など、存在しない場合はインベントリーファイルを作成します。

```
$ touch inventory.file
```

3. インベントリーファイルを開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

4. Ansible Playbook ファイル (**data-archive-in-asymmetric-vault.yml**) を作成します。以下に例を示します。

```
$ cp data-archive-in-asymmetric-vault.yml add-services-to-an-asymmetric-vault.yml
```

5. **data-archive-in-asymmetric-vault-copy.yml** ファイルを開いて編集します。
6. ファイルを変更するには、**ipavault** タスクセクションに以下の変数を設定します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は vault の名前 (例: **secret_vault**) に設定します。
   - **service** 変数は vault のサービス所有者に設定します (例: **HTTP/webserver1.idm.example.com**)。
   - services 変数を使用して、vault シークレットにアクセスできる **サービス** を定義します。
   - **action** 変数は **member** に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。
8. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file add-services-to-an-asymmetric-vault.yml
```

25.3. ANSIBLE を使用した非対称 VAULT への IDM サービスシークレットの保存

以下の手順に従って、Ansible Playbook を使用して、サービス vault にシークレットを保存し、サービスが後で取得できるようにします。以下の手順で使用する例では、管理者は **secret_vault** という名前の非対称 vault にシークレットが含まれる **PEM** ファイルを保存します。こうすることで、サービスは、vault からシークレットを取得する際に、秘密鍵を使用して必ず認証することになります。vault メンバーは、IdM クライアントからファイルを取得できます。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。

- Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **IdM 管理者** パスワードが分かっている。
- サービスシークレットの保存先の 非対称 vault を作成している。
- シークレットが **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem** ファイルなど、Ansible コントローラーのローカルに保存されている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. オプション: **inventory.file** など、存在しない場合はインベントリーファイルを作成します。

```
$ touch inventory.file
```

3. インベントリーファイルを開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

4. Ansible Playbook ファイル (**data-archive-in-asymmetric-vault.yml**) を作成します。以下に例を示します。

```
$ cp data-archive-in-asymmetric-vault.yml data-archive-in-asymmetric-vault-copy.yml
```

5. **data-archive-in-asymmetric-vault-copy.yml** ファイルを開いて編集します。
6. ファイルを変更するには、**ipavault** タスクセクションに以下の変数を設定します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は vault の名前 (例: **secret_vault**) に設定します。
   - **service** 変数は vault のサービス所有者に設定します (例: **HTTP/webserver1.idm.example.com**)。
   - **in** 変数は "{{ lookup('file', 'private-key-to-an-externally-signed-certificate.pem')| b64encode }}" に設定します。この設定では、Ansible が IdM サーバーではなく、Ansible コントローラーの作業ディレクトリーから秘密鍵のあるファイルを取得するようになりま

す。

- **action** 変数は **member** に設定します。
  今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。

8. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-archive-in-asymmetric-vault-copy.yml
```

25.4. ANSIBLE を使用した IDM サービスのサービスシークレットの取得

以下の手順に従って、Ansible Playbook を使用してサービスの代わりにサービス vault からシークレットを取得します。以下の手順で使用する例では、Playbook を実行して **secret_vault** という名前の **PEM** ファイルを取得し、Ansible インベントリーファイルに **ipaservers** として記載されている全ホストの指定の場所に保存します。

サービスは、キータブを使用して IdM に対して、さらに秘密鍵を使用して vault に対して認証を実行します。**ansible-freeipa** がインストールされている IdM クライアントから、サービスの代わりにファイルを取得できます。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

- IdM 管理者パスワードを把握している。
- サービスシークレットの保存先の 非対称 vault を作成している。
- vault にシークレットをアーカイブ している。
- Ansible コントローラーで **private_key_file** 変数が指定する場所にサービス vault のシークレットの取得に使用する秘密鍵が保存されている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. オプション: **inventory.file** など、存在しない場合はインベントリーファイルを作成します。

```
$ touch inventory.file
```

3. インベントリーファイルを開き、以下のホストを定義します。
   - **[ipaserver]** セクションで、使用する IdM サーバーを定義します。
   - **[webservers]** セクションで、シークレットを取得するホストを定義します。たとえば Ansible に対して **webserver1.idm.example.com**、**webserver2.idm.example.com** および **webserver3.idm.example.com** にシークレットを取得するように指示するには以下を実行します。

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

[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
webserver3.idm.example.com
```

4. Ansible Playbook ファイル (**retrieve-data-asymmetric-vault.yml**) のコピーを作成します。以下に例を示します。

```
$ cp retrieve-data-asymmetric-vault.yml retrieve-data-asymmetric-vault-copy.yml
```

5. **retrieve-data-asymmetric-vault-copy.yml** ファイルを開いて編集します。
6. ファイルを変更するには、**ipavault** タスクセクションに以下の変数を設定します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は vault の名前 (例: **secret_vault**) に設定します。
   - **service** 変数は vault のサービス所有者に設定します (例: **HTTP/webserver1.idm.example.com**)。
   - **private_key_file** 変数は、サービス vault のシークレットの取得に使用する秘密鍵の場所に設定します。

- **out** 変数は、現在の作業ディレクトリーなど、**private-key-to-an-externally-signed-certificate.pem** シークレットの取得先となる IdM サーバーの場所に設定します。
- **action** 変数は **member** に設定します。
  今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. Playbook に、IdM サーバーから Ansible コントローラーにデータファイルを取得するセクションを追加します。

```
---
- 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: true
      mode: 0600
```

8. インベントリーファイルの **webservers** セクションに記載されている Web サーバーに、Ansible コントローラーから取得した **private-key-to-an-externally-signed-certificate.pem** ファイルを転送するセクションを Playbook に追加します。

```
---
- 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. ファイルを保存します。
10. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-data-asymmetric-vault-copy.yml
```

25.5. シークレットが漏洩した場合の ANSIBLE での IDM サービス VAULT シークレットの変更

以下の手順に従って、サービスインスタンスが危険にさらされたときに Ansible Playbook を再利用し、サービス vault に保存されているシークレットを変更します。以下の例のシナリオでは、シークレットを保存する非対称 vault への鍵は漏洩されておらず、**webserver3.idm.example.com** で取得したシークレットの情報が漏洩した場合を前提としています。この例では、管理者は 非対称 vault でシークレットを保存する時 および IdM ホストに非対称 vault からシークレットを取得する時 に Ansible Playbook を再利用します。この手順のはじめに、IdM 管理者は新規シークレットを含む新しい **PEM** ファイルを非対称 vault に保存し、不正アクセスのあった Web サーバー (**webserver3.idm.example.com**) に新しいシークレットを配置しないようにインベントリーファイルを調節し、もう一度この 2 つの手順を実行します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- **IdM 管理者** パスワードが分かっている。
- サービスシークレットの保存先の 非対称 vault を作成している。
- IdM ホストで実行中の Web サービスの **httpd** 鍵を新たに生成し、不正アクセスのあった以前の鍵を置き換えている。
- 新しい **httpd** キーが **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem** ファイルなど、Ansible コントローラーのローカルに保存されている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. インベントリーファイルを開き、以下のホストが正しく定義されていることを確認します。

   - **[ipaserver]** セクションの IdM サーバー
   - **[webservers]** セクションのシークレット取得先のホスト。たとえば Ansible に対して **webserver1.idm.example.com** と **webserver2.idm.example.com** にシークレットを取得するように指示するには、以下を入力します。

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

[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
```

重要

このリストに不正アクセスのあった Web サーバーが含まれないようにしてください (現在の例では **webserver3.idm.example.com**)。

3. **data-archive-in-asymmetric-vault-copy.yml** ファイルを開いて編集します。
4. ファイルを変更するには、**ipavault** タスクセクションに以下の変数を設定します。

   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は vault の名前 (例: **secret_vault**) に設定します。
   - **service** 変数は vault のサービス所有者に設定します (例: **HTTP/webserver.idm.example.com**)。
   - **in** 変数は **"{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem')| b64encode }}"** に設定します。この設定では、Ansible が IdM サーバーではなく、Ansible コントローラーの作業ディレクトリーから秘密鍵のあるファイルを取得するようになります。
   - **action** 変数は **member** に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。
6. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-
archive-in-asymmetric-vault-copy.yml
```

7. **retrieve-data-asymmetric-vault-copy.yml** ファイルを開いて編集します。
8. ファイルを変更するには、**ipavault** タスクセクションに以下の変数を設定します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は vault の名前 (例: **secret_vault**) に設定します。
   - **service** 変数は vault のサービス所有者に設定します (例: **HTTP/webserver1.idm.example.com**)。
   - **private_key_file** 変数は、サービス vault のシークレットの取得に使用する秘密鍵の場所に設定します。
   - **out** 変数は、現在の作業ディレクトリーなど、**new-private-key-to-an-externally-signed-certificate.pem** シークレットの取得先となる IdM サーバーの場所に設定します。
   - **action** 変数は **member** に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. Playbook に、IdM サーバーから Ansible コントローラーにデータファイルを取得するセクションを追加します。

```
---
- name: Retrieve data from vault
```

```
  hosts: ipaserver
  become: true
  gather_facts: false
  tasks:
[...]
  - name: Retrieve data file
    fetch:
      src: new-private-key-to-an-externally-signed-certificate.pem
      dest: ./
      flat: true
      mode: 0600
```

10. インベントリーファイルの **webservers** セクションに記載されている Web サーバーに、Ansible コントローラーから取得した **new-private-key-to-an-externally-signed-certificate.pem** ファイルを転送するセクションを Playbook に追加します。

```
---
- name: Send data file to webservers
  become: true
  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. ファイルを保存します。

12. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-data-asymmetric-vault-copy.yml
```

25.6. 関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの README-vault.md Markdown ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/vault/** ディレクトリーのサンプルの Playbook を参照してください。

第26章 ANSIBLE を使用して IDM にサービスを配置させる手順およびさせない手順

Ansible **サービス** モジュールを使用すると、Identity Management (IdM) 管理者は、IdM ネイティブではない特定のサービスが IdM に存在するか、存在しないかを確認できます。たとえば、**サービス** モジュールを使用すると以下を行うことができます。

- 手動でインストールしたサービスが IdM クライアントに存在していることを確認します。サービスが存在しない場合には自動的にインストールされます。詳細は、次を参照してください。
  - IdM クライアントの IdM で HTTP サービスが存在することを確認する。
  - IdM 以外のクライアントの IdM で HTTP サービスが存在することを確認する。
  - DNS のない IdM クライアントに HTTP サービスが存在することを確認する。
- IdM に登録したサービスに、証明書がアタッチされていることを確認して、証明書がない場合には自動的にその証明書がインストールされます。詳細は、次を参照してください。
- IdM サービスエントリーで外部署名の証明書が存在することを確認する。
- IdM ユーザーとホストが、サービスキータブを取得して作成できるようにします。詳細は、次を参照してください。
  - IdM ユーザー、グループ、ホスト、またはホストグループがサービスのキータブを作成できるようにする。
  - IdM ユーザー、グループ、ホスト、またはホストグループがサービスのキータブを取得できるようにする。
- IdM ユーザーとホストが Kerberos エイリアスをサービスに追加できるようにします。詳細は、次を参照してください。
  - サービスの Kerberos プリンシパルエイリアスを存在させる手順
- サービスが IdM クライアントにないかを確認して、そのサービスが存在する場合は自動的にそのサービスを削除します。詳細は、次を参照してください。
  - IdM クライアントの IdM で HTTP サービスがないことを確認する。

26.1. ANSIBLE PLAYBOOK を使用して IDM に HTTP サービスを存在させる手順

以下の手順に従って、Ansible Playbook を使用して IdM に HTTP サーバーが存在することを確認します。

前提条件

- HTTP サービスをホストするシステムが IdM クライアントである。
- IdM 管理者パスワードがある。

手順

1. **inventory.file** などのインベントリーファイルを作成します。

```
$ touch inventory.file
```

2. **inventory.file** を開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present.yml**) のコピーを作成します。以下に例を示します。

```
$ 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. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-copy.yml**) を開きます。

```
---
- 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. ファイルを調整します。

   - **ipaadmin_password** 変数で定義されている IdM 管理者パスワードを変更します。
   - **ipaservice** タスクの **name** 変数で定義されているように、HTTP サービスが実行されている IdM クライアントの名前を変更します。

6. ファイルを保存し、終了します。

7. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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
```

検証手順

1. 管理者として IdM Web UI にログインします。
2. **Identity → Services** に移動します。

Services リストに **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** がある場合は、Ansible Playbook は IdM に正常に追加されています。

関連情報

- HTTP サーバーとブラウザークライアント間の通信を保護する方法は、Apache HTTP Server への TLS 暗号化の追加 を参照してください。
- HTTP サービスの証明書を要求する場合は、certmonger を使用したサービスの IdM 証明書の取得 に記載されている手順を参照してください。

26.2. 単一の ANSIBLE タスクを使用して、IDM クライアント上の IDM に複数のサービスが存在することを確認する

ansible-freeipa ipaservice モジュールを使用すると、単一の Ansible タスクで複数の Identity Management (IdM) サービスを追加、変更、削除できます。そのためには、**ipaservice** モジュールの **services** オプションを使用します。

services オプションを使用すると、特定のサービスにのみ適用される複数のサービス変数を指定することもできます。このサービスを **name** 変数で定義します。これは、**services** オプションの唯一の必須変数です。

この手順を完了して、単一タスクで IdM に **HTTP/client01.idm.example.com@IDM.EXAMPLE.COM** サービスと **ftp/client02.idm.example.com@IDM.EXAMPLE.COM** サービスが存在することを確認します。

前提条件

- コントロールノード:
  - Ansible バージョン 2.14 以降を使用している。
  - **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
  - RHEL 9.3 以降を使用している。
  - **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。

手順

1. 次の内容を含む Ansible Playbook ファイル **add-http-and-ftp-services.yml** を作成します。

```
---
- name: Playbook to add multiple services in a single task
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Add HTTP and ftp services
    ipaservice:
      ipaadmin_password: "{{ ipaadmin_password }}"
```

```
    services:
    - name:  HTTP/client01.idm.example.com@IDM.EXAMPLE.COM
    - name: ftp/client02.idm.example.com@IDM.EXAMPLE.COM
```

2. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory add-http-and-ftp-services.yml
```

関連情報

- **ansible-freeipa** アップストリームドキュメントのサービスモジュール

26.3. ANSIBLE PLAYBOOK を使用して、IDM 以外のクライアントの IDM で HTTP サービスを存在させる手順

以下の手順に従って、Ansible Playbook を使用して IdM クライアントではないホストの IdM に HTTP サーバーが存在することを確認します。HTTP サーバーを IdM に追加して、ホストを IdM に追加することもできます。

前提条件

- ホストに HTTP サービスをインストール している。
- HTTP を設定したホストが IdM クライアントではない。それ以外の場合は HTTP サービスを IdM に登録する 手順に従います。
- IdM 管理者パスワードがある。
- ホスト向けの DNS A レコード (または IPv6 を使用している場合は AAAA レコード) がある。
- サーバーが RHEL 9.2 以降を実行し、FIPS モードが有効になっている場合、クライアントが Extended Master Secret (EMS) 拡張機能をサポートしているか、TLS 1.3 を使用している必要があります。EMS を使用しない TLS 1.2 接続は失敗します。詳細は、ナレッジベースの記事 TLS extension "Extended Master Secret" enforced を参照してください。

手順

1. **inventory.file** などのインベントリーファイルを作成します。

```
$ touch inventory.file
```

2. **inventory.file** を開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-using-host-check.yml**) のコピーを作成します。以下に例を示します。

```
$ 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. コピーしたファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-using-host-check-copy.yml**) を開きます。**ipaservice** タスクで **ipaadmin_password** と **name** 変数の場所を特定します。

```
---
- 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: true
```

5. ファイルを調整します。

   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は、HTTP サービスが実行されているホストの名前に設定します。

6. ファイルを保存し、終了します。

7. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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
```

検証手順

1. 管理者として IdM Web UI にログインします。
2. **Identity → Services** に移動します。

HTTP/client.idm.example.com@IDM.EXAMPLE.COM が Services リストに表示されています。

関連情報

- 通信の安全を確保する方法は、Apache HTTP Server への TLS 暗号化の追加 を参照してください。

26.4. ANSIBLE PLAYBOOK を使用して、DNS を使用せずに IDM クライアントで HTTP サービスを存在させる手順

以下の手順に従って、Ansible Playbook を使用して DNS エントリーがない IdM クライアントで HTTP サーバーが存在することを確認します。このシナリオでは、IdM ホストに DNS A エントリーがないことを前提としています (IPv4 の代わりに IPv6 を使用する場合には DNS AAAA エントリーがない)。

前提条件

- HTTP サービスをホストするシステムが IdM に登録されている。
- ホストの DNS A または DNS AAAA レコードを存在させない。ホストの DNS レコードが存在する場合には、ホストの DNS レコードが存在する場合は、Ansible Playbook を使用して IdM に HTTP サービスを存在させる手順 に従うようにしてください。
- IdM 管理者パスワードがある。

手順

1. **inventory.file** などのインベントリーファイルを作成します。

```
$ touch inventory.file
```

2. **inventory.file** を開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force.yml**) のコピーを作成します。以下に例を示します。

```
$ 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. コピーしたファイル **/usr/share/doc/ansible-freeipa/playbooks/service/service-is-present-with-host-force-copy.yml** を開きます。**ipaservice** タスクで **ipaadmin_password** と **name** 変数の場所を特定します。

```
---
- 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: true
```

5. ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。

- **name** 変数は、HTTP サービスが実行されているホストの名前に設定します。

6. ファイルを保存し、終了します。
7. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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
```

検証手順

1. 管理者として IdM Web UI にログインします。
2. **Identity → Services** に移動します。

HTTP/client.idm.example.com@IDM.EXAMPLE.COM が Services リストに表示されています。

関連情報

- 通信の安全を確保する方法は、Apache HTTP Server への TLS 暗号化の追加 を参照してください。

26.5. ANSIBLE PLAYBOOK を使用して IDM サービスエントリーに外部署名証明書を存在させる手順

この手順に従い、**ansible-freeipa service** モジュールを使用して、外部の認証局 (CA) が HTTP サービスの IdM エントリーにアタッチされていることを確認します。IdM CA が自己署名の証明書を使用する場合には、IdM CA ではなく外部 CA が署名した HTTP サービスの証明書を使用すると便利です。

前提条件

- ホストに HTTP サービスをインストール している。
- HTTP サービスを IdM に登録 している。
- IdM 管理者パスワードがある。
- 発行元が HTTP サービスのプリンシパルに対応する外部署名証明書がある。

手順

1. **inventory.file** などのインベントリーファイルを作成します。

```
$ touch inventory.file
```

2. **inventory.file** を開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. 以下のように **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present.yml** ファイルのコピーを作成します。

```
$ 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. オプション: 証明書の形式が Privacy Enhanced Mail (PEM) の場合には、証明書を識別名エンコーディングルール (DER) 形式に変換し、コマンドラインインターフェイス (CLI) でより簡単に処理できるようにします。

```
$ openssl x509 -outform der -in cert1.pem -out cert1.der
```

5. **base64** を使用して **DER** ファイルを復号化します。ラッピングを無効にするには、**-w0** オプションを使用します。

```
$ base64 cert1.der -w0
MIIC/zCCAeegAwIBAgIUV74O+4kXeg21o4vxfRRtyJm...
```

6. 標準出力からクリップボードに証明書をコピーします。

7. **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-certificate-present-copy.yml** ファイルを開き、このファイルの内容を編集および表示します。

```
---
- 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/client.idm.example.com
      certificate: |
        - MIICBjCCAW8CFHnm32VcXaUDGfEGdDL/...
      [...]
      action: member
      state: present
```

8. ファイルを調整します。

   - **certificate** 変数を使用して定義した証明書は、CLI からコピーした証明書に置き換えます。上記のように | パイプ文字と **certificate:** 変数を併用する場合は、1 行に入力せずに、このように証明書を入力してください。これで、証明書の読み取りが容易になります。

   - **ipaadmin_password** 変数で定義されている IdM 管理者パスワードを変更します。

   - HTTP サービスを実行する IdM クライアントの名前 (**name** 変数で定義) を変更します。

   - その他の関連する変数を変更します。

9. ファイルを保存し、終了します。

10. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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
```

検証手順

1. 管理者として IdM Web UI にログインします。
2. **Identity → Services** に移動します。
3. **HTTP/client.idm.example.com** など、新しく追加した証明書が指定されたサービス名をクリックします。

右側の **サービス証明書** セクションで、新たに追加した証明書を確認できるようになりました。

26.6. ANSIBLE PLAYBOOK を使用して IDM ユーザー、グループ、ホスト、またはホストグループでサービスのキータブを作成できるようにする手順

キータブは、Kerberos プリンシパルと暗号化鍵のペアを含むファイルです。キータブファイルは通常、人の介入なしで、またはプレーンテキストファイルに保存されたパスワードを使用せずに、スクリプトで自動的に Kerberos で認証を行えるようにするために使用します。その後、スクリプトは取得した認証情報を使用してリモートシステムに保存されているファイルにアクセスできます。

Identity Management (IdM) 管理者は、他のユーザーが、IdM で実行しているサービスのキータブを取得したり、作成したりできるようにします。特定のユーザーおよびユーザーグループがキータブを作成できるようにすると、IdM 管理者パスワードを共有せずにサービスの管理を委譲できます。このように委譲することで、システムを詳細にわたり管理できます。

以下の手順に従って、特定の IdM ユーザー、ユーザーグループ、ホスト、およびホストグループが、IdM クライアントで実行している HTTP サービスのキータブを作成できるようにします。具体的には、IdM ユーザー **user01** が、**client.idm.example.com** という名前の IdM クライアントで実行中の HTTP サービスのキータブを作成できるように設定する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

- HTTP サービスを IdM に登録 している。
- HTTP サービスをホストするシステムが IdM クライアントである。
- キータブの作成を許可する IdM ユーザーおよびユーザーグループが IdM に存在する。
- キータブの作成を許可する IdM ホストおよびホストグループが IdM に存在する。

手順

1. **inventory.file** などのインベントリーファイルを作成します。

```
$ touch inventory.file
```

2. **inventory.file** を開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present.yml**) のコピーを作成します。以下に例を示します。

```
$ 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. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_create_keytab-present-copy.yml**) を開きます。
5. 以下を変更してファイルを調整します。
   - **ipaadmin_password** 変数で指定した IdM 管理者パスワード。
   - HTTP サービスが実行されている IdM クライアントの名前。現在の例では、**HTTP/client.idm.example.com** です。
   - **allow_create_keytab_user:** セクションに記載されている IdM ユーザー名。現在の例では **user01** です。
   - **allow_create_keytab_group:** セクションに記載されている IdM ユーザーグループ名。
   - **allow_create_keytab_host:** セクションに記載されている IdM ホスト名。
   - **allow_create_keytab_hostgroup:** セクションに記載されている IdM ホストグループ名。
   - **tasks** セクションの **name** 変数で指定されているタスク名。
     現在の例に合わせて調節すると、コピーされたファイルは以下のようになります。

```
---
- 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. ファイルを保存します。

7. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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
```

検証手順

1. 特定の HTTP サービスのキータブを作成する権限のある IdM ユーザーで、IdM サーバーに SSH 接続します。

```
$ ssh user01@server.idm.example.com
Password:
```

2. **ipa-getkeytab** コマンドを使用して、HTTP サービスの新規キータブを生成します。

```
$ ipa-getkeytab -s server.idm.example.com -p HTTP/client.idm.example.com -k
/etc/httpd/conf/krb5.keytab
```

-s オプションは、キータブを生成するキー配布センター (KDC) サーバーを指定します。

-p オプションは、作成するキータブのプリンシパルを指定します。

-k オプションは、新規キーを追加するキータブファイルを指定します。ファイルが存在しない場合には、作成されます。

このコマンドでエラーが表示されない場合は、**user01** で、**HTTP/client.idm.example.com** のキータブが正常に作成されています。

26.7. ANSIBLE PLAYBOOK を使用して IDM ユーザー、グループ、ホスト、またはホストグループでサービスのキータブを取得できるようにする手順

キータブは、Kerberos プリンシパルと暗号化鍵のペアを含むファイルです。キータブファイルは通常、人の介入なしで、またはプレーンテキストファイルに保存されたパスワードを使用せずに、スクリプトで自動的に Kerberos で認証を行えるようにするために使用します。その後、スクリプトは取得した認証情報を使用してリモートシステムに保存されているファイルにアクセスできます。

IdM 管理者は、他のユーザーが、IdM で実行しているサービスのキータブを取得したり、作成したりできるようにします。

以下の手順に従って、特定の IdM ユーザー、ユーザーグループ、ホスト、およびホストグループが、IdM クライアントで実行している HTTP サービスのキータブを取得できるようにします。具体的には IdM ユーザー **user01** が、**client.idm.example.com** で実行する HTTP サービスのキータブを取得できるように設定する方法を説明します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- HTTP サービスを IdM に登録 している。
- キータブの取得を許可する IdM ユーザーおよびグループが IdM に存在する。
- キータブの取得を許可する IdM ホストおよびホストグループが IdM に存在する。

手順

1. **inventory.file** などのインベントリーファイルを作成します。

```
$ touch inventory.file
```

2. **inventory.file** を開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present.yml**) のコピーを作成します。以下に例を示します。

```
$ 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. コピーしたファイル **/usr/share/doc/ansible-freeipa/playbooks/service/service-member-allow_retrieve_keytab-present-copy.yml** を開きます。
5. ファイルを調整します。

- **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
- **ipaservice** タスクの **name** 変数は、HTTP サービスのプリンシパルに設定します。現在の例では、**HTTP/client.idm.example.com** です。
- **allow_retrieve_keytab_group:** セクションで IdM ユーザーの名前を指定します。現在の例では **user01** です。
- **allow_retrieve_keytab_group:** セクションで、IdM ユーザーグループの名前を指定します。
- **allow_retrieve_keytab_group:** セクションで IdM ホストの名前を指定します。
- **allow_retrieve_keytab_group:** セクションで IdM ホストグループの名前を指定します。
- **tasks** セクションの **name** 変数を使用して、タスク名を指定します。
  現在の例に合わせて調節すると、コピーされたファイルは以下のようになります。

```
---
- 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. ファイルを保存します。
7. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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
```

検証手順

1. HTTP サービスのキータブ取得権限がある IdM ユーザーとして、IdM サーバーに SSH 接続します。

```
$ ssh user01@server.idm.example.com
Password:
```

2. **-r** オプションを指定して **ipa-getkeytab** コマンドを使用してキータブを取得します。

```
$ ipa-getkeytab -r -s server.idm.example.com -p HTTP/client.idm.example.com -k
/etc/httpd/conf/krb5.keytab
```

-

-s オプションは、キータブの取得元となるキー配布センター (KDC) サーバーを指定します。

-p オプションは、キータブを取得するプリンシパルを指定します。

-k オプションは、取得した鍵を追加するキータブファイルを指定します。ファイルが存在しない場合には、作成されます。

このコマンドでエラーが表示されない場合は、**user01** で **HTTP/client.idm.example.com** のキータブが正常に取得されています。

26.8. ANSIBLE PLAYBOOK を使用してサービスの KERBEROS プリンシパルのエイリアスを存在させる手順

シナリオによっては、IdM ユーザー、ホストまたはサービスが Kerberos アプリケーションに対して Kerberos プリンシパルエイリアスを使用して認証できるように IdM 管理者が設定すると便利です。次のようなシナリオになります。

- ユーザー名の変更後に、ユーザーが以前のユーザー名と新しいユーザー名の両方でシステムにログインできるようにする。
- IdM Kerberos レルムがメールドメインと異なる場合でも、ユーザーはメールアドレスを使用してログインする必要がある。

以下の手順に従って、**client.idm.example.com** で実行される HTTP サービスの **HTTP/mycompany.idm.example.com** のプリンシパルエイリアスを作成します。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- ホストに HTTP サービスを設定 している。
- HTTP サービスを IdM に登録 している。
- HTTP を設定しているホストが IdM クライアントである。

手順

1. **inventory.file** などのインベントリーファイルを作成します。

```
$ touch inventory.file
```

2. **inventory.file** を開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present.yml**) のコピーを作成します。以下に例を示します。

```
$ 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. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-member-principal-present-copy.yml**) を開きます。

5. 以下を変更してファイルを調整します。

   - **ipaadmin_password** 変数で指定した IdM 管理者パスワード。
   - **name** 変数で指定したサービス名。これは、サービスの標準プリンシパル名です。現在の例では **HTTP/client.idm.example.com** です。
   - **principal** 変数で指定した Keroberos プリンシパルエイリアス。これは、 **name** 変数で定義したサービスに追加するエイリアスです。現在の例では、**host/mycompany.idm.example.com** です。
   - **tasks** セクションの **name** 変数で指定されているタスク名。
   現在の例に合わせて調節すると、コピーされたファイルは以下のようになります。

```
---
- 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. ファイルを保存します。

7. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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
```

Playbook を実行すると、到達できないタスクが 0 件、失敗したタスクが 0 件になる場合には、**HTTP/client.idm.example.com** サービスの **host/mycompany.idm.example.com** Kerberos プリンシパルが正常に作成されています。

関連情報

- Managing Kerberos principal aliases for users, hosts, and services を参照してください。

26.9. ANSIBLE PLAYBOOK を使用して IDM に HTTP サービスを存在させないようにする手順

以下の手順に従って、IdM からサービスの登録を解除します。より具体的には、Ansible Playbook を使用して、IdM にある **HTTP/client.idm.example.com** という名前の HTTP サーバーを削除する方法を説明します。

前提条件

- IdM 管理者パスワードがある。

手順

1. **inventory.file** などのインベントリーファイルを作成します。

```
$ touch inventory.file
```

2. **inventory.file** を開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent.yml**) のコピーを作成します。以下に例を示します。

```
$ 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. Ansible Playbook ファイル (**/usr/share/doc/ansible-freeipa/playbooks/service/service-is-absent-copy.yml**) を開きます。

5. 以下を変更してファイルを調整します。

   - **ipaadmin_password** 変数で定義されている IdM 管理者パスワード。
   - **ipaservice** タスクの **name** 変数で定義されている、HTTP サービスの Kerberos プリンシパル。
     現在の例に合わせて調節すると、コピーされたファイルは以下のようになります。

```
---
- 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. ファイルを保存し、終了します。

7. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ 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
```

検証手順

1. 管理者として IdM Web UI にログインします。

2. **Identity → Services** に移動します。

Services リストに **HTTP/client.idm.example.com@IDM.EXAMPLE.COM** サービスが表示されていない場合には、IdM から正常に削除されています。

26.10. 関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-service.md** Markdown ファイルを参照してください。

- **/usr/share/doc/ansible-freeipa/playbooks/config** ディレクトリーのサンプルの Playbook を参照してください。

第27章 ANSIBLE PLAYBOOK を使用した IDM でのグローバル DNS 設定の管理

Red Hat Ansible Engine の **dnsconfig** モジュールを使用して、Identity Management (IdM) DNS のグローバル設定を設定できます。グローバル DNS 設定で定義したオプションは、すべての IdM DNS サーバーに適用されます。ただし、グローバル設定は、特定の IdM DNS ゾーンの設定よりも優先度が低くなります。

dnsconfig モジュールは以下の変数をサポートします。

- グローバルフォワーダー (特に通信に使用する IP アドレスとポート)
- グローバル転送ポリシー: only、first、または noneDNS 転送ポリシーの上記のタイプの詳細は、IdM の DNS 転送ポリシー を参照してください。
- 正引きルックアップおよび逆引きルックアップゾーンの同期。

前提条件

- DNS サービスが IdM サーバーにインストールされている。統合 DNS のある IdM サーバーをインストールする方法は、以下のリンクのいずれかを参照してください。
  - IdM サーバーのインストール: 統合 DNS と統合 CA を root CA として使用する場合
  - IdM サーバーのインストール: 統合 DNS と外部 CA を root CA として使用する場合
  - IdM サーバーのインストール: 統合 DNS があり外部 CA がない場合

本章では、以下のセクションを説明します。

- IdM を使用して /etc/resolv.conf のグローバルフォワーダーが NetworkManager に削除されないようにする方法
- Ansible を使用して IdM に DNS グローバルフォワーダーを存在させる手順
- Ansible を使用して IdM に DNS グローバルフォワーダーを存在させないようにする手順
- ipadnsconfig ansible-freeipa モジュールの **action: member** オプション
- IdM の DNS 転送ポリシーの 概要
- Ansible Playbook を使用して forward first ポリシーを IdM DNS グローバル設定で指定する手順
- Ansible Playbook を使用して IdM DNS でグローバルフォワーダーを無効にする手順
- Ansible Playbook を使用して IdM DNS で正引きおよび逆引きルックアップゾーンの同期を無効にする手順

27.1. IDM を使用して /ETC/RESOLV.CONF のグローバルフォワーダーが NETWORKMANAGER に削除されないようにする方法

統合 DNS で Identity Management (IdM) をインストールすると、**/etc/resolv.conf** ファイルが localhost アドレス (**127.0.0.1**) を参照するように設定されます。

```
# Generated by NetworkManager
search idm.example.com
nameserver 127.0.0.1
```

DHCP (**Dynamic Host Configuration Protocol**) を使用するネットワークなど、環境によっては、**/etc/resolv.conf** ファイルへの変更が **NetworkManager** サービスにより元に戻されてしまう場合があります。IdM DNS のインストールプロセスでは、以下のように **NetworkManager** サービスも設定し、DNS 設定を永続化します。

1. DNS インストールスクリプトを使用して、**/etc/NetworkManager/conf.d/zzzz-ipa.conf NetworkManager** 設定ファイルを作成し、検索の順序と DNS サーバーリストを制御します。

```
# auto-generated by IPA installer
[main]
dns=default

[global-dns]
searches=$DOMAIN

[global-dns-domain-*]
servers=127.0.0.1
```

2. **NetworkManager** サービスが再読み込みされ、**/etc/NetworkManager/conf.d/** ディレクトリーにある以前のファイルの設定を使用して **/etc/resolv.conf** ファイルを作成します。今回の場合は、**zzz-ipa.conf** ファイルです。

重要

/etc/resolv.conf ファイルは手動で変更しないでください。

27.2. ANSIBLE を使用して IDM に DNS グローバルフォワーダーを存在させる手順

以下の手順に従って、Ansible Playbook を使用して、IdM に DNS グローバルフォワーダーを追加します。以下の例では、IdM 管理者は、ポート **53** にインターネットプロトコル (IP) v4 アドレスが **7.7.9.9**、IPv6 アドレスが **2001:db8::1:0** で指定されている DNS サーバーに、DNS グローバルフォワーダーが配置されるようにします。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**forwarders-absent.yml**) のコピーを作成します。以下に例を示します。

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. **ensure-presence-of-a-global-forwarder.yml** ファイルを開いて編集します。

5. 以下の変数を設定してファイルを調整します。

   a. Playbook の **name** 変数は、**IdM DNS にグローバルフォワーダーを追加する Playbook** の設定に変更します。

   b. **tasks** セクションで、タスクの **name** を **Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0 on port 53** に変更します。

   c. **ipadnsconfig** の **forwarders** セクションで以下を行います。

      i. 最初の **ip_address** の値は、グローバルフォワーダーの IPv4 アドレス (**7.7.9.9**) に変更します。

      ii. 2 番目の **ip_address** の値は、グローバルフォワーダーの IPv6 アドレス (**2001:db8::1:0**) に変更します。

      iii. **port** の値が **53** に設定されていることを確認します。

   d. **state** を **present** に変更します。
   今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-of-a-global-forwarder.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsconfig.md** ファイルを参照してください。

27.3. ANSIBLE を使用して IDM に DNS グローバルフォワーダーを存在させないようにする手順

以下の手順に従って、Ansible Playbook を使用して IdM で DNS グローバルフォワーダーを削除します。以下の手順では、IdM 管理者が、ポート **53** で、IP (Internet Protocol) v4 アドレス **8.8.6.6** および IP v6 アドレス **2001:4860:4860::8800** を持つ DNS グローバルフォワーダーが存在しないことを確認します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**forwarders-absent.yml**) のコピーを作成します。以下に例を示します。

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. **ensure-absence-of-a-global-forwarder.yml** ファイルを開いて編集します。

5. 以下の変数を設定してファイルを調整します。

   a. Playbook の **name** 変数は、**IdM DNS でグローバルフォワーダーを配置しない Playbook** の設定に変更します。

   b. **tasks** セクションで、タスクの **name** を **Ensure the absence of a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800 on port 53** に変更します。

   c. **ipadnsconfig** の **forwarders** セクションで以下を行います。

      i. 最初の **ip_address** の値は、グローバルフォワーダーの IPv4 アドレス (**8.8.6.6**) に変更します。

      ii. 2 番目の **ip_address** の値は、グローバルフォワーダーの IPv6 アドレス (**2001:4860:4860::8800**) に変更します。

      iii. **port** の値が **53** に設定されていることを確認します。

   d. **action** 変数は **member** に設定します。

   e. **state** が **absent** に設定されていることを確認します。

   今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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
```

重要

Playbook で **action: member** を使用せずに **state: absent** オプションだけを使用すると、その Playbook は失敗します。

6. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-of-a-global-forwarder.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsconfig.md** ファイル
- ipadnsconfig ansible-freeipa モジュールの **action: member** オプション

27.4. IPADNSCONFIG ANSIBLE-FREEIPA モジュールのACTION: MEMBER オプション

ansible-freeipa ipadnsconfig モジュールを使用して Identity Management (IdM) のグローバルフォワーダーを除外するには、**state: absent** オプションの他に **action: member** オプションを使用する必要があります。Playbook で **action: member** を使用せずに **state: absent** だけを使用すると、その Playbook は失敗します。そのため、すべてのグローバルフォワーダーを削除するには、Playbook でこれらをすべて個別に指定する必要があります。一方、**state: present** オプションに **action: member** は必要ありません。

次の表 に、action: member オプションの正しい使用法を示す DNS グローバルフォワーダーの追加と削除の両方の設定例を示します。この表の各行には、以下が含まれます。

- Playbook を実行する前に設定されたグローバルフォワーダー
- Playbook からの抜粋
- Playbook の実行後に設定されたグローバルフォワーダー

表27.1 グローバルフォワーダーの ipadnsconfig 管理

以前のフォワーダー	Playbook の抜粋	後のフォワーダー

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

以前のフォワーダー	Playbook の抜粋	後のフォワーダー

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
		Playbook を実行しようとすると、エラーが発生します。元の設定 (8.8.6.6、8.8.6.7) は変更されません。

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

27.5. IDM での DNS 転送ポリシー

IdM は、**first** および **only** の BIND 転送ポリシーと、IdM 固有の転送ポリシー **none** をサポートします。

forward first (デフォルト)

IdM BIND サービスは、DNS クエリーを設定済みのフォワーダーに転送します。サーバーエラーやタイムアウトが原因でクエリーに失敗すると、BIND はインターネット上のサーバーを使用して再帰解決にフォールバックします。**forward first** ポリシーはデフォルトのポリシーで、DNS トラフィックの最適化に適しています。

Forward only

IdM BIND サービスは、DNS クエリーを設定済みのフォワーダーに転送します。サーバーエラーやタイムアウトが原因でクエリーに失敗すると、BIND はエラーをクライアントに返します。分割された DNS 設定の環境では、**forward only** ポリシーが推奨されます。

None (転送の無効化)

DNS クエリーは、**none** 転送ポリシーで転送されません。グローバル転送設定をゾーン別にオーバーライドする場合にのみ、転送の無効化は有用です。このオプションは、IdM の BIND 設定で空のフォワーダーリストを指定するのと同じです。

注記

転送を使用して、IdM のデータと、他の DNS サーバーのデータと統合できません。IdM DNS のプライマリーゾーン内にある特定のサブゾーンのクエリーのみを転送できます。

デフォルトでは、IdM サーバーが権威サーバーとなっているゾーンに、クエリーされた DNS 名が所属する場合には、BIND サービスは、クエリーを別のサーバーに転送しません。このような場合は、クエリーされた DNS 名が IdM データベースに見つからない場合は、**NXDOMAIN** との応答が返されます。転送は使用されません。

例27.1 サンプルシナリオ

IdM サーバーは、**test.example** の権威サーバーです。DNS ゾーン。BIND は、IP アドレス **192.0.2.254** でクエリーを DNS サーバーに転送するように設定されています。

クライアントが **nonexistent.test.example** のクエリーを送信する場合DNS 名である BIND は、IdM サーバーが **test.example.** ゾーンの権威サーバーであることを検出して、クエリーを **192.0.2.254.** サーバーには転送しません。その結果、DNS クライアントは **NXDomain** エラーメッセージを受け取り、クエリーされたドメインが存在しないことをユーザーに通知します。

27.6. ANSIBLE PLAYBOOK を使用して FORWARD FIRST ポリシーを IDM DNS グローバル設定で指定する手順

以下の手順に従って、Ansible Playbook を使用して、IdM DNS のグローバル転送ポリシーが **forward first** に設定されていることを確認します。

forward first DNS 転送ポリシーを使用する場合には、DNS クエリーは設定済みのフォワーダーに転送されます。サーバーエラーやタイムアウトが原因でクエリーに失敗すると、BIND はインターネット上のサーバーを使用して再帰解決にフォールバックします。Forward first ポリシーはデフォルトのポリシーです。トラフィックの最適化に適しています。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

- IdM 管理者パスワードを把握している。
- IdM 環境に統合 DNS サーバーが含まれている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**set-configuration.yml**) のコピーを作成します。以下に例を示します。

```
$ cp set-configuration.yml set-forward-policy-to-first.yml
```

4. **set-forward-policy-to-first.yml** ファイルを開いて編集します。
5. **ipadnsconfig** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **forward_policy** 変数は **first** に設定します。
     元の Playbook で関連性の他の行はすべて削除します。以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file set-forward-policy-to-first.yml
```

関連情報

- IdM での DNS 転送ポリシー を参照してください。

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsconfig.md** ファイルを参照してください。
- サンプルの Playbook は、**/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーを参照してください。

27.7. ANSIBLE PLAYBOOK を使用して IDM DNS でグローバルフォワーダーを無効にする手順

以下の手順に従って、Ansible Playbook を使用して、IdM DNS でグローバルフォワーダーが無効になっていることを確認します。グローバルフォワーダーの無効化は、**forward_policy** 変数を **none** に設定します。

グローバルフォワーダーを無効にすると、DNS クエリーは転送されません。グローバル転送設定をゾーン別にオーバーライドする場合にのみ、転送の無効化は有用です。このオプションは、IdM の BIND 設定で空のフォワーダーリストを指定するのと同じです。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。
- IdM 環境に統合 DNS サーバーが含まれている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**disable-global-forwarders.yml**) のコピーを作成します。以下に例を示します。

```
$ cp disable-global-forwarders.yml disable-global-forwarders-copy.yml
```

4. **disable-global-forwarders-copy.yml** ファイルを開いて編集します。
5. **ipadnsconfig** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **forward_policy** 変数を **none** に設定します。
     以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disable-global-forwarders-copy.yml
```

関連情報

- IdM での DNS 転送ポリシー を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsconfig.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーにあるその他のサンプル Playbook を参照してください。

27.8. ANSIBLE PLAYBOOK を使用して IDM DNS で正引きおよび逆引きルックアップゾーンの同期を無効にする手順

以下の手順に従って、Ansible Playbook を使用して、正引きおよび逆引きルックアップゾーンが IdM DNS で同期されないようにします。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。

    - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。
- IdM 環境に統合 DNS サーバーが含まれている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**disallow-reverse-sync.yml**) のコピーを作成します。以下に例を示します。

```
$ cp disallow-reverse-sync.yml disallow-reverse-sync-copy.yml
```

4. **disallow-reverse-sync-copy.yml** ファイルを開きます。
5. **ipadnsconfig** タスクセクションに以下の変数を設定して、ファイルを調整します。
    - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
    - **allow_sync_ptr** 変数を **no** に設定します。
      以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。
7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disallow-reverse-sync-copy.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsconfig.md** ファイルを参照してください。
- サンプルの Playbook は、**/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーを参照してください。

第28章 ANSIBLE PLAYBOOK を使用した IDM DNS ゾーンの管理

Identity Management (IdM) 管理者は、**ansible-freeipa** パッケージに含まれる **dnszone** モジュールを使用して IdM DNS ゾーンの動作を管理できます。

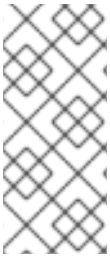
- IdM でサポートされる DNS ゾーンタイプ
- IdM で設定できる DNS 属性
- Ansible Playbook を使用して IdM DNS にプライマリーゾーンを作成する方法
- Ansible Playbook を使用して複数の変数が含まれるプライマリー IdM DNS ゾーンを存在させる手順
- IP アドレスが指定されている場合に Ansible Playbook を使用して逆引き DNS ルックアップのゾーンを存在させる手順

前提条件

- DNS サービスが IdM サーバーにインストールされている。Red Hat Ansible Engine を使用して、統合 DNS のある IdM サーバーをインストールする方法は、Ansible Playbook を使用した Identity Management サーバーのインストール を参照してください。

28.1. サポート対象の DNS ゾーンタイプ

Identity Management (IdM) は、2 種類の DNS ゾーン (**primary** および **forward**) をサポートします。ここでは、DNS 転送のシナリオ例を含め、2 種類のゾーンについて説明します。



注記

本ガイドでは、ゾーンタイプには BIND の用語を使用し、Microsoft Windows DNS で使用する用語とは異なります。BIND のプライマリーゾーンは、Microsoft Windows DNS の **正引きルックアップゾーン** と **逆引きルックアップゾーン** と同じ目的で使用されます。BIND の正引きゾーンは、Microsoft Windows DNS の **条件付きフォワーダー** と同じ目的で使用されます。

プライマリー DNS ゾーン

プライマリー DNS ゾーンには、権威 DNS データが含まれ、DNS を動的に更新できます。この動作は、標準 BIND 設定の **type master** 設定と同じです。プライマリーゾーンは、**ipa dnszone-*** コマンドを使用して管理できます。
標準 DNS ルールに準拠するには、プライマリーゾーンすべてに **start of authority** (SOA) と **nameserver** (NS) レコードを含める必要があります。IdM では、DNS ゾーンの作成時にこれらのレコードが自動的に生成されますが、NS レコードを親ゾーンに手動でコピーして適切な委譲を作成する必要があります。

標準の BIND の動作に合わせて、権威サーバーではない名前のクエリーは、他の DNS サーバーに転送されます。DNS サーバー (別称: フォワーダー) は、クエリーに対して権威がある場合と、ない場合があります。

> **例28.1 DNS 転送のシナリオ例**
>
> IdM サーバーには **test.example.** プライマリーゾーンが含まれています。このゾーンには、**sub.test.example.** 名前の NS 委譲レコードが含まれます。さらに、**test.example.** ゾーンは、**sub.text.example** サブゾーンのフォワーダー IP アドレス **192.0.2.254** で設定されます。

> クライアントが **nonexistent.test.example.** の名前をクエリーすると、**NXDomain** の応答を受け取りますが、IdM サーバーはこの名前に対して権威があるため、転送は発生しません。
>
> 反対に、**host1.sub.test.example.** の名前をクエリーすると、IdM サーバーはこの名前に対して権威がないので、設定済みのフォワーダー (**192.0.2.254**) に転送されます。

正引き DNS ゾーン

IdM の観点からは、正引き DNS ゾーンには権威データは含まれません。実際、正引きのゾーンには、通常以下情報 2 つのみが含まれます。

- ドメイン名
- ドメインに関連付けられた DNS サーバーの IP アドレス

定義済みのドメインに所属する名前のクエリーはすべて、指定の IP アドレスに転送されます。この動作は、標準 BIND 設定の **type forward** 設定と同じです。正引きゾーンは、**ipa dnsforwardzone-*** コマンドを使用して管理できます。

正引き DNS ゾーンは、IdM-Active Directory (AD) 信頼のコンテキストで特に便利です。IdM DNS サーバーが **idm.example.com** ゾーンに対して、AD DNS サーバーが **ad.example.com** ゾーンに対して権威がある場合には、**ad.example.com** が **idm.example.com** プライマリーゾーンの DNS 正引きゾーンになります。つまり、IP アドレスが **somehost.ad.example.com** の IdM クライアントからクエリーが送信されると、**ad.example.com** IdM DNS 正引きゾーンに指定の AD ドメインコントローラーに転送されます。

28.2. プライマリー IDM DNS ゾーンの設定属性

Identity Management (IdM) は、更新期間、転送設定、キャッシュ設定など、特定のデフォルト設定を指定して新しいゾーンを作成します。IdM DNS ゾーン属性 には、デフォルトのゾーン設定属性があります。これは、以下のオプションのいずれかを使用して変更できます。

- コマンドラインインターフェイス (CLI) の **dnszone-mod** コマンド詳細は IdM CLI でのプライマリー DNS ゾーンの設定の編集 を参照してください。
- IdM Web UI詳細は IdM Web UI でのプライマリー DNS ゾーンの設定の編集 を参照してください。
- **ipadnszone** モジュールを使用する Ansible Playbook詳細は、IdM での DNS ゾーンの管理 を参照してください。

ここではゾーンの実際の情報を設定するほか、DNS サーバーが **start of authority** (SOA) レコードエントリーを処理する方法と、DNS ネームサーバーからの記録を更新する方法を定義します。

表28.1 IdM DNS ゾーン属性

属性	ansible-freeipa 変数	説明

属性	ansible-freeipa 変数	説明
権威ネームサーバー	**name_server**	プライマリー DNS ネームサーバーのドメイン名 (別称: SOA MNAME) を設定します。 デフォルトでは、各 IdM サーバーは SOA MNAME フィールドで自己アドバタイズします。そのため、**--name-server** を使用して LDAP に保存されている値は無視されます。
管理者の電子メールアドレス	**admin_email**	ゾーン管理者が使用する電子メールアドレスを設定します。デフォルトでは、ホストの root アカウントになります。
SOA serial	**serial**	SOA レコードにシリアル番号を設定します。IdM ではバージョン番号が自動的に設定され、この番号のユーザーによる変更は想定されていません。
SOA refresh	**refresh**	セカンダリー DNS サーバーがプライマリー DNS サーバーから更新を要求するまでの待機時間を秒単位で設定します。
SOA retry	**retry**	失敗した更新操作を再試行するまでに待機する時間を秒単位で設定します。
SOA expire	**expire**	セカンダリー DNS サーバーが操作の試行を終了するまでに、更新操作を実行する時間を秒単位で設定します。
SOA minimum	**minimum**	RFC 2308 に準拠し、ネガティブキャッシュの TTL (TTL) 値を秒単位で設定します。
SOA time to live	**ttl**	ゾーン apex のレコードの TTL を秒単位で設定します。たとえば、**example.com** ゾーンでは、名前が **example.com** のすべてのレコード (A、NS または SOA) が設定されますが、**test.example.com** などの他のドメイン名には影響はありません。
デフォルトの TTL	**default_ttl**	これまでに個別の Time To Live (TTL) 値が設定されたことのないゾーンで、すべての値のネガティブキャッシュのデフォルト TTL を秒単位で設定します。変更を有効にするには、すべての IdM DNS サーバーで **named-pkcs11** サービスを再起動する必要があります。
BIND 更新ポリシー	**update_policy**	DNS ゾーンでクライアントに許可されるパーミッションを設定します。
Dynamic update	**dynamic_update**=TRUE|FALSE	クライアントの DNS レコードへの動的更新を有効にします。 false に設定すると、IdM クライアントマシンは IP アドレスを追加または更新できなくなる点に注意してください。

属性	ansible-freeipa 変数	説明
Allow transfer	**allow_transfer**=string	指定のゾーンを転送できる IP アドレスまたはネットワーク名のセミコロン区切りのリストを指定します。 デフォルトでは、ゾーン転送は無効です。**allow_transfer** のデフォルト値は **none** です。
Allow query	**allow_query**	DNS クエリーを発行できる IP アドレスまたはネットワーク名のセミコロン区切りのリストを指定します。
Allow PTR sync	**allow_sync_ptr**=1|0	ゾーンの A または AAAA レコード (正引きレコード) が自動的に PTR (逆引き) レコードと同期されるかどうかを設定します。
Zone forwarder	**forwarder**=IP_address	DNS ゾーン向けに特別に設定されたフォワーダーを指定します。これは、IdM ドメインで使用されるグローバルフォワーダーとは別のものです。 複数のフォワーダーを指定する場愛には、オプションを複数回使用します。
転送ポリシー	**forward_policy**=none|only|first	転送ポリシーを指定します。サポート対象のポリシーに関する情報は、IdM での DNS 転送ポリシー を参照してください。

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnszone.md** ファイルを参照してください。

28.3. ANSIBLE を使用した IDM DNS でのプライマリーゾーンの作成

以下の手順に従って、Ansible Playbook を使用して、プライマリー DNS ゾーンが存在することを確認します。以下の手順で使用する例では、**zone.idm.example.com** DNS ゾーンが存在するようにします。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnszone** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイルのコピー (**dnszone-present.yml**) を作成します。以下に例を示します。

```
$ cp dnszone-present.yml dnszone-present-copy.yml
```

4. **dnszone-present-copy.yml** ファイルを開いて編集します。

5. **ipadnszone** タスクセクションに以下の変数を設定してファイルを調整します。

   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **zone_name** 変数は **zone.idm.example.com** に設定します。
     以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

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

関連情報

- サポート対象の DNS ゾーンタイプ を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnszone.md** ファイルを参照してください。

- **/usr/share/doc/ansible-freeipa/playbooks/dnszone** ディレクトリーのサンプルの Ansible Playbook を参照してください。

28.4. ANSIBLE PLAYBOOK を使用して、変数が複数ある IDM にプライマリー DNS ゾーンを存在させる手順

以下の手順に従って、Ansible Playbook を使用して、プライマリー DNS ゾーンが存在することを確認します。以下の手順で使用する例では、IdM 管理者は **zone.idm.example.com** の DNS ゾーンを追加します。Ansible Playbook は、ゾーンのパラメーターを複数設定します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnszone** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイルのコピー (**dnszone-all-params.yml**) を作成します。以下に例を示します。

```
$ cp dnszone-all-params.yml dnszone-all-params-copy.yml
```

4. **dnszone-all-params-copy.yml** ファイルを開いて編集します。
5. **ipadnszone** タスクセクションに以下の変数を設定してファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **zone_name** 変数は zone.idm.example.com に設定します。

- 正引きレコードと逆引きレコードを同期できるように場合は (A および AAAA レコードを PTR レコードと同期)、**allow_sync_ptr** 変数を true に設定します。
- **dynamic_update** 変数は、true に設定して、IdM クライアントマシンが IP アドレスを追加または更新できるようにします。
- **dnssec** 変数は、true に設定して、ゾーン内のレコードのインラインの DNSSEC 署名を許可します。
- **allow_transfer** 変数は、ゾーン内のセカンダリーネームサーバーの IP アドレスに設定します。
- **allow_query** 変数は、クエリーを発行できる IP アドレスまたはネットワークに設定します。
- **forwarders** 変数は、グローバルフォワーダー の IP アドレスに設定します。
- **serial** 変数は SOA レコードのシリアル番号に設定します。
- ゾーン内の DNS レコードの **refresh**、**retry**、**expire**、**minimum**、**ttl** および **default_ttl** の値を定義します。
- **nsec3param_rec** 変数を使用して、ゾーンの NSEC3PARAM レコードを定義します。
- **skip_overlap_check** 変数は、true に設定して、既存のゾーンと重複していても DNS を強制的に作成します。
- **skip_nameserver_check** は、true に設定して、ネームサーバーが解決できない場合でも DNS ゾーンを強制的に作成します。
  以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file dnszone-all-params-copy.yml
```

関連情報

- サポート対象の DNS ゾーンタイプ を参照してください。
- プライマリー IdM DNS ゾーンの設定属性 を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnszone.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/dnszone** ディレクトリーのサンプルの Ansible Playbook を参照してください。

28.5. IP アドレスが指定されている場合に ANSIBLE PLAYBOOK を使用して逆引き DNS ルックアップのゾーンを存在させる手順

以下の手順に従って、Ansible Playbook を使用して、逆引き DNS ゾーンが存在することを確認します。以下の手順で使用する例では、IdM 管理者は、IdM ホストの IP アドレスおよび接頭辞長を使用して、逆引き DNS ルックアップゾーンを追加します。

name_from_ip 変数を使用して DNS サーバーの IP アドレスの接頭辞の長さを指定すると、ゾーン名を制御できます。接頭辞の長さを指定しない場合には、システムが DNS サーバーにゾーンに関するクエリーを出し、192.168.1.2 の **name_from_ip** の値をもとに、このクエリーで、以下の DNS ゾーンのいずれかを返します。

- 1.168.192.in-addr.arpa.
- 168.192.in-addr.arpa.
- 192.in-addr.arpa.

クエリーが返すゾーンは想定しているゾーンとは異なる可能性があるため、ゾーンが誤って削除されないように **state** オプションが **present** に設定されている場合のみ、**name_from_ip** を使用できます。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。

- Ansible バージョン 2.14 以降を使用している。
- Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnszone** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイルのコピー (**dnszone-reverse-from-ip.yml**) を作成します。以下に例を示します。

```
$ cp dnszone-reverse-from-ip.yml dnszone-reverse-from-ip-copy.yml
```

4. **dnszone-reverse-from-ip-copy.yml** ファイルを開いて編集します。
5. **ipadnszone** タスクセクションに以下の変数を設定してファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name_from_ip** 変数は IdM ネームサーバーの IP に設定し、接頭辞の長さを指定します。以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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 }}"
```

この Playbook は、IP アドレス **192.168.1.2** と接頭辞長 24 をもとに、逆引き DNS ルックアップのゾーンを作成します。次に、Playbook は生成されたゾーン名を表示します。

6. ファイルを保存します。
7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file dnszone-reverse-from-ip-copy.yml
```

関連情報

- サポート対象の DNS ゾーンタイプ を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnszone.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/dnszone** ディレクトリーのサンプルの Ansible Playbook を参照してください。

第29章 ANSIBLE を使用した IDM での DNS の場所の管理

Identity Management (IdM) 管理者は、**ansible-freeipa** パッケージで利用可能な **location** モジュールを使用して IdM DNS の場所を管理できます。

- DNS ベースのサービス検出
- DNS の場所のデプロイに関する考慮事項
- DNS の Time to live (TTL)
- Ansible を使用して IdM の場所が存在することを確認する
- Ansible を使用して IdM の場所を削除する手順

29.1. DNS ベースのサービス検出

DNS ベースのサービス検出は、クライアントが DNS プロトコルを使用するプロセスで、**LDAP** や **Kerberos** など、特定のサービスを提供するネットワークでサーバーを見つけ出します。一般的な操作の 1 つとして、クライアントが最寄りのネットワークインフラストラクチャー内にある認証サーバーを特定できるようにすることが挙げられます。理由は、スループットが向上してネットワークレイテンシーが短縮されるので全体的なコスト削減を図ることができるためです。

サービス検出の主な利点は以下のとおりです。

- 近くにあるサーバーの名前を明示的に設定する必要がない。
- DNS サーバーをポリシーの中央プロバイダーとして使用する。同じ DNS サーバーを使用するクライアントは、サービスプロバイダーと優先順序に関する同じポリシーにアクセスできます。

Identity Management (IdM) ドメインには、**LDAP**、**Kerberos**、およびその他のサービスに DNS サービスレコード (SRV レコード) があります。たとえば、次のコマンドは、IdM DNS ドメインで TCP ベースの **Kerberos** サービスを提供するホストの DNS サーバーをクエリーします。

例29.1 DNS の場所に関する独立した結果

```
$ 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.
```

出力には、以下の情報が含まれます。

- **0** (優先度): ターゲットホストの優先度。値が小さいほど優先度が高くなります。
- **100** (加重)。優先順位が同じエントリーの相対的な重みを指定します。詳細は RFC 2782, section 3 を参照してください。
- **88** (ポート番号): サービスのポート番号
- サービスを提供するホストの正規名。

この例では、2 つのホスト名が返され、どちらも同じ優先順位と重みでした。この場合には、クライアントは結果リストから無作為にエントリーを使用します。

代わりに、クライアントを設定して、DNS の場所に設定されている DNS サーバーをクエリーすると、出力が異なります。場所が割り当てられた IdM サーバーの場合は、カスタマイズした値が返されます。以下の例では、クライアントは、場所 **germany** にある DNS サーバーをクエリーするように設定されています。

例29.2 DNS の場所ベースの結果

```
$ 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.
```

IdM DNS サーバーは、ローカルサーバーを優先する DNS の場所固有の SRV レコードを参照する DNS エイリアス (CNAME) を自動的に返します。この CNAME レコードは、出力の最初の行に表示されます。この例では、ホスト**idmserver-01.idm.example.com**の優先度の値が最も低いため、優先されます。**idmserver-02.idm.example.com** の優先度の値が高く、推奨されるホストが使用できない場合にバックアップとしてのみ使用されます。

29.2. DNS の場所のデプロイに関する考慮事項

Identity Management (IdM) は、統合 DNS を使用する際に、場所固有のサービス (SRV) レコードを生成できます。各 IdM DNS サーバーはロケーション固有の SRV レコードを生成するため、DNS の場所ごとに 1 つ以上の IdM DNS サーバーをインストールする必要があります。

クライアントの DNS の場所に対するアフィニティーは、クライアントが受け取った DNS レコードでのみ定義されます。そのため、DNS のサービス検出を行うクライアントが、IdM DNS サーバーからの場所固有のレコードを解決した場合には、IdM DNS サーバーと IdM 以外の DNS コンシューマーサーバーと recursor を組み合わせることができます。

IdM サービスおよび IdM DNS サービス以外のほとんどのデプロイメントでは、DNS recursor はラウンドトリップタイム (RTT) メトリックを使用して、最寄りの IdM DNS サーバーを自動的に選択します。通常、IdM DNS サーバーを使用するクライアントが、最寄りの DNS の場所のレコードを取得し、最寄りの DNS サーバーの最適なセットを使用するようになります。

29.3. DNS の TIME TO LIVE (TTL)

クライアントは、ゾーンの設定に指定された期間の DNS リソースレコードをキャッシュできます。このキャッシュにより、クライアントは Time to Live (TTL) 値の有効期限が切れるまで変更を受け取れない場合があります。Identity Management (IdM) のデフォルトの TTL 値は **1 日** です。

クライアントコンピューターがサイト間でローミングする場合には、IdM DNS ゾーンの TTL 値を調整する必要があります。この値は、クライアントがサイト間のローミングに必要とする時間よりも低い値に設定します。これにより、別のサイトに再接続する前にクライアントでキャッシュされた DNS エントリーが期限切れになり、DNS サーバーに対してクエリーを実行し、場所固有の SRV レコードを更新します。

関連情報

- プライマリー IdM DNS ゾーンの設定属性 を参照してください。

29.4. ANSIBLE を使用して IDM の場所が存在することを確認する

Identity Management (IdM) のシステム管理者は、クライアントが最寄りのネットワークインフラストラクチャーで認証サーバーを特定できるように IdM DNS の場所を設定できます。

以下の手順では、Ansible Playbook を使用して IdM に DNS の場所を追加する方法を説明します。この例では、DNS の場所 **germany** が IdM に存在することを確認する方法を説明します。IdM に DNS の場所を追加して、ローカルの IdM クライアントがサーバーの応答時間を短縮できるように、特定の IdM サーバーをこの場所に割り当てることができます。

前提条件

- IdM 管理者パスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- DNS の場所のデプロイメントに関する考慮事項 を理解している。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/location/** ディレクトリーにある **location-present.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook の **location-present-copy.yml** ファイルを開いて編集します。
4. **ipalocation** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - 使用しているユースケースに合わせて、タスクの **名前** を調節します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は、場所の名前に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

関連情報

- IdM Web UI を使用した DNS の場所への IdM サーバーの割り当て または IdM CLI を使用した DNS の場所への IdM サーバーの割り当て を参照してください。

29.5. ANSIBLE を使用して IDM の場所を削除する手順

Identity Management (IdM) のシステム管理者は、クライアントが最寄りのネットワークインフラストラクチャーで認証サーバーを特定できるように IdM DNS の場所を設定できます。

以下の手順では、Ansible Playbook を使用して、IdM から DNS の場所を削除する方法を説明します。この例では、DNS の場所 (**germany**) が IdM から削除されていることを確認する方法を説明します。DNS の場所を削除すると、その場所に、特定の IdM サーバーを割り当てられず、ローカルの IdM クライアントでその場所を使用できなくなります。

前提条件

- IdM 管理者パスワードを把握している。
- **germany** DNS の場所に IdM サーバーが割り当てられていません。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- この例では、サンプルの Playbook のコピーを保存する一元管理場所として **~/MyPlaybooks/** ディレクトリーを 作成して設定 していることを前提とします。

手順

1. **~/ MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/location/** ディレクトリーにある **location-absent.yml** ファイルのコピーを作成します。

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

3. Ansible Playbook ファイル (**location-absent-copy.yml**) を開きます。

4. **ipalocation** タスクセクションに以下の変数を設定して、ファイルを調整します。

   - 使用しているユースケースに合わせて、タスクの **名前** を調節します。
   - **ipaadmin_password** 変数は IdM 管理者のパスワードに設定します。
   - **name** 変数は DNS の場所の名前に設定します。
   - **state** 変数は **absent** に設定されていることを確認します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

6. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

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

29.6. 関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-location.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/location** ディレクトリーのサンプルの Ansible Playbook を参照してください。

第30章 IDM での DNS 転送の管理

以下の手順に従い、Identity Management (IdM) Web UI、IdM CLI、および Ansible を使用して DNS グローバルフォワーダーおよび DNS 正引きゾーンを設定します。

- IdM DNS サーバーの 2 つのロール
- IdM での DNS 転送ポリシー
- IdM Web UI でのグローバルフォワーダーの追加
- CLI でのグローバルフォワーダーの追加
- IdM Web UI での DNS 正引きゾーンの追加
- CLI での DNS 正引きゾーンの追加
- Ansible を使用した IdM での DNS グローバルフォワーダーの確立
- Ansible を使用して IdM に DNS グローバルフォワーダーを存在させる手順
- Ansible を使用して IdM に DNS グローバルフォワーダーを存在させないようにする手順
- Ansible を使用した IdM での DNS グローバルフォワーダーの無効化
- Ansible を使用して IdM に DNS 正引きゾーンを存在させる手順
- Ansible を使用して IdM で DNS 正引きゾーンを複数配置する手順
- Ansible を使用して IdM で DNS 正引きゾーンを無効にする手順
- Ansible を使用して IdM から DNS 正引きゾーンを削除する手順

30.1. IDM DNS サーバーの 2 つのロール

DNS 転送は、DNS サービスが DNS クエリーに応答する方法を左右します。デフォルトでは、IdM と統合されている Berkeley Internet Name Domain (BIND) サービスは、**権威** および **再帰** DNS サーバーの両方として機能します。

権威 DNS サーバー

IdM サーバーが権威のある DNS ゾーンに所属する名前のクエリーを DNS クライアントが出した場合に、BIND は設定済みのゾーンに含まれるデータで応答します。権威データは常に他のデータよりも優先されます。

再帰 DNS サーバー

IdM サーバーが権威のない名前のクエリーを DNS クライアントが出した場合に、BIND は他の DNS サーバーを使用してこのクエリーを解決しようとします。フォワーダーが定義されていない場合は、BIND がインターネット上のルートサーバーにクエリーを出し、再帰解決アルゴリズムを使用して DNS クエリーに応答します。

BIND を使用して他の DNS サーバーに直接問い合わせて、インターネットで利用可能なデータをもとに再帰を実行することは推奨されません。別の DNS サーバーである **フォワーダー** を使用してクエリーを解決するように BIND を設定できます。

フォワーダーを使用するように BIND を設定すると、クエリーと応答が IdM サーバーとフォワーダーの間で送受信され、IdM サーバーが権威データ以外の DNS キャッシュとして機能します。

30.2. IDM での DNS 転送ポリシー

IdM は、**first** および **only** の BIND 転送ポリシーと、IdM 固有の転送ポリシー **none** をサポートします。

forward first (デフォルト)

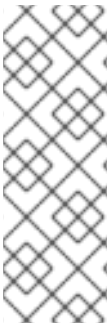
IdM BIND サービスは、DNS クエリーを設定済みのフォワーダーに転送します。サーバーエラーやタイムアウトが原因でクエリーに失敗すると、BIND はインターネット上のサーバーを使用して再帰解決にフォールバックします。**forward first** ポリシーはデフォルトのポリシーで、DNS トラフィックの最適化に適しています。

Forward only

IdM BIND サービスは、DNS クエリーを設定済みのフォワーダーに転送します。サーバーエラーやタイムアウトが原因でクエリーに失敗すると、BIND はエラーをクライアントに返します。分割された DNS 設定の環境では、**forward only** ポリシーが推奨されます。

None (転送の無効化)

DNS クエリーは、**none** 転送ポリシーで転送されません。グローバル転送設定をゾーン別にオーバーライドする場合にのみ、転送の無効化は有用です。このオプションは、IdM の BIND 設定で空のフォワーダーリストを指定するのと同じです。



注記

転送を使用して、IdM のデータと、他の DNS サーバーのデータと統合できません。IdM DNS のプライマリーゾーン内にある特定のサブゾーンのクエリーのみを転送できます。

デフォルトでは、IdM サーバーが権威サーバーとなっているゾーンに、クエリーされた DNS 名が所属する場合には、BIND サービスは、クエリーを別のサーバーに転送しません。このような場合は、クエリーされた DNS 名が IdM データベースに見つからない場合は、**NXDOMAIN** との応答が返されます。転送は使用されません。

例30.1 サンプルシナリオ

IdM サーバーは、**test.example** の権威サーバーです。DNS ゾーン。BIND は、IP アドレス **192.0.2.254** でクエリーを DNS サーバーに転送するように設定されています。

クライアントが **nonexistent.test.example** のクエリーを送信する場合DNS 名である BIND は、IdM サーバーが **test.example.** ゾーンの権威サーバーであることを検出して、クエリーを **192.0.2.254.** サーバーには転送しません。その結果、DNS クライアントは **NXDomain** エラーメッセージを受け取り、クエリーされたドメインが存在しないことをユーザーに通知します。

30.3. IDM WEB UI でのグローバルフォワーダーの追加

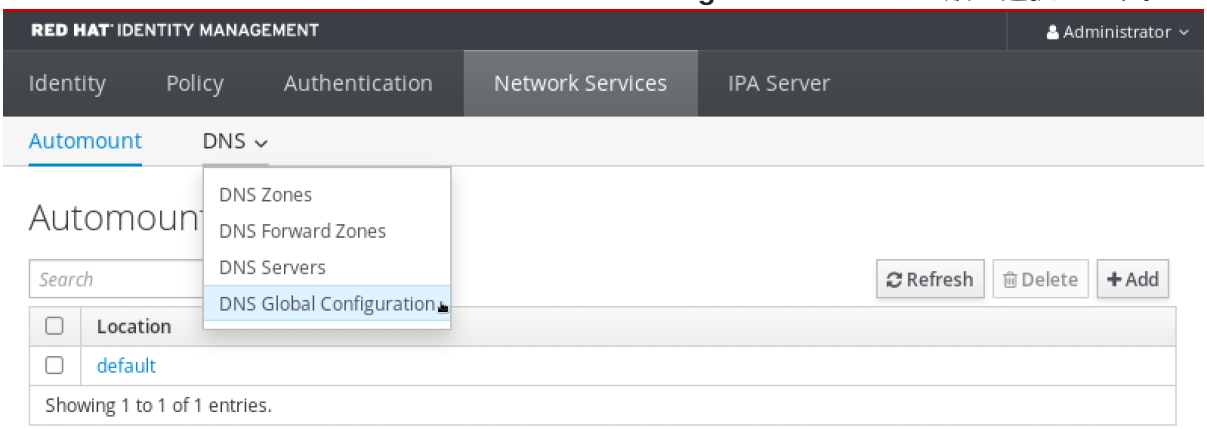
以下の手順に従って、Identity Management (IdM) Web UI でグローバル DNS フォワーダーを追加します。

前提条件

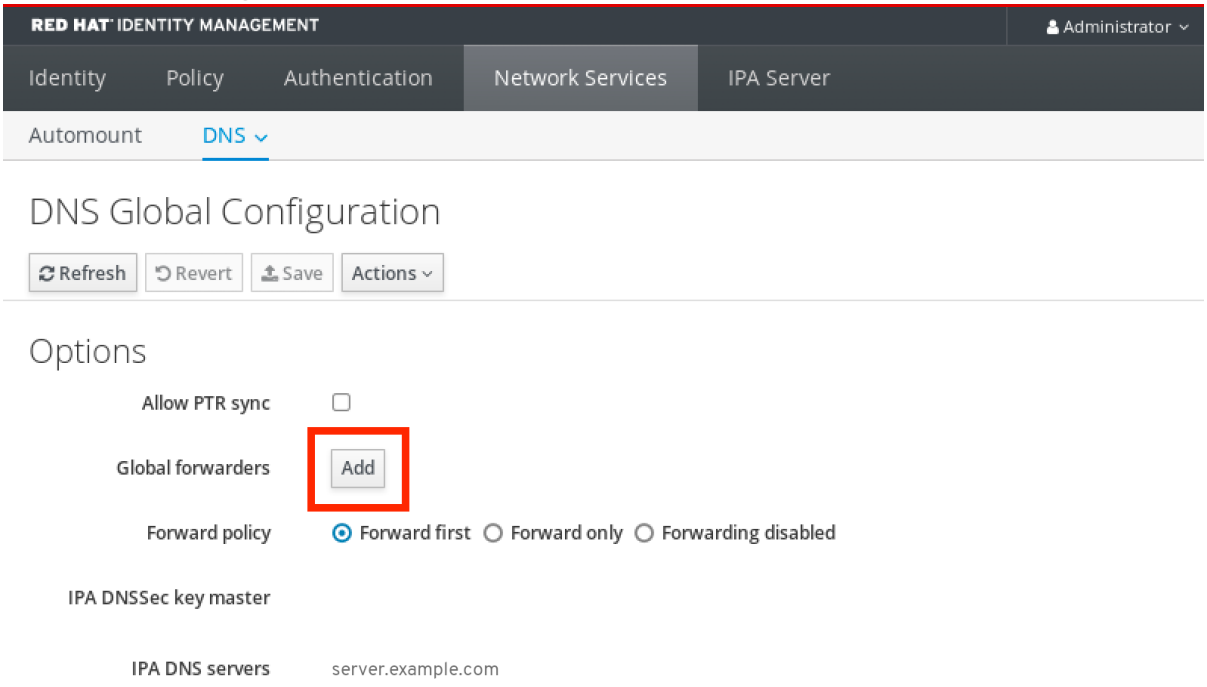
- IdM 管理者として IdM WebUI にログインしている。
- クエリーを転送する DNS サーバーのインターネットプロトコル (IP) アドレスを知っている。

手順

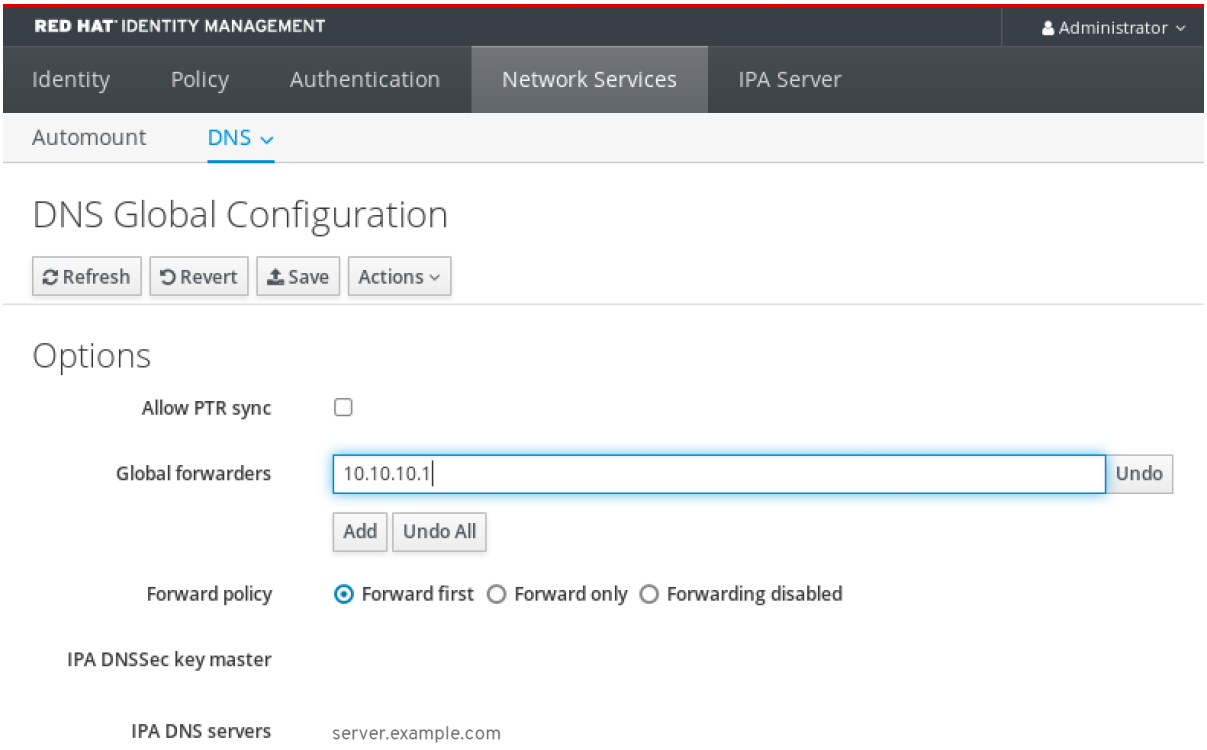
1. IdM Web UI で **Network Services → DNS Global Configuration → DNS** の順に選択します。



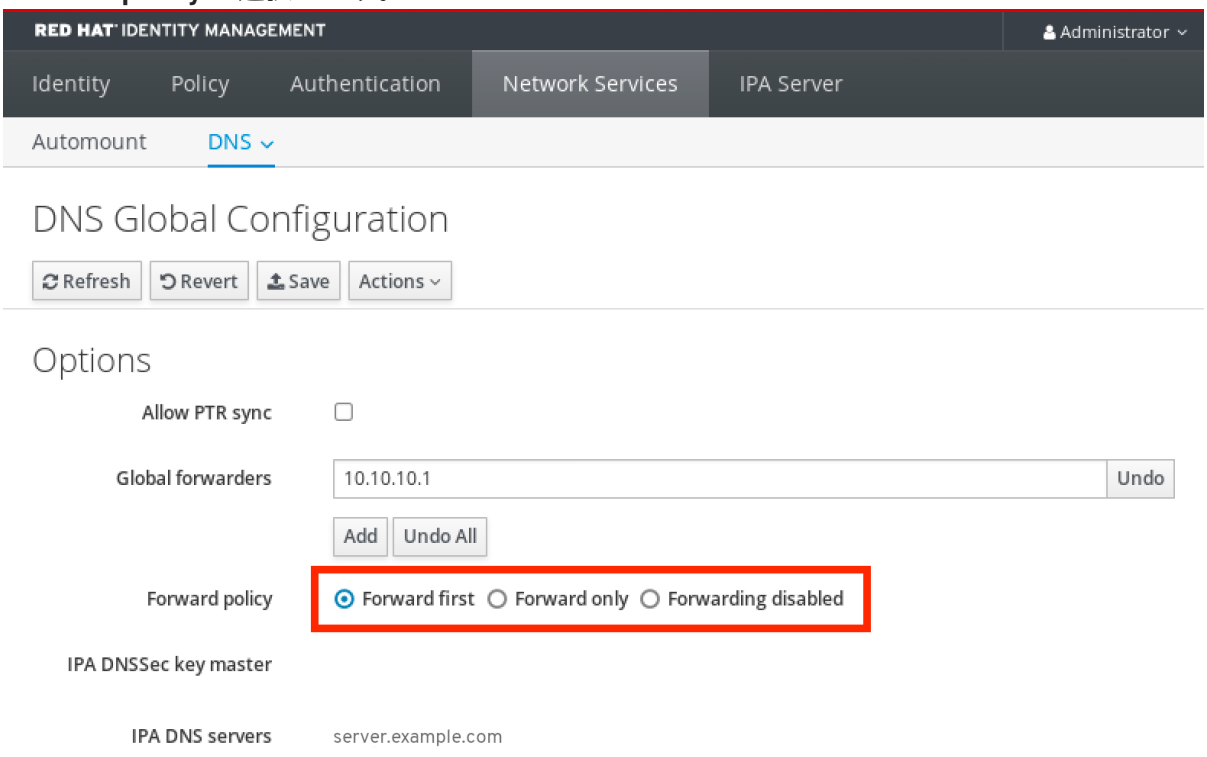
2. **DNS Global Configuration** セクションで、**Add** をクリックします。



3. 転送された DNS クエリーを受信する DNS サーバーの IP アドレスを指定します。



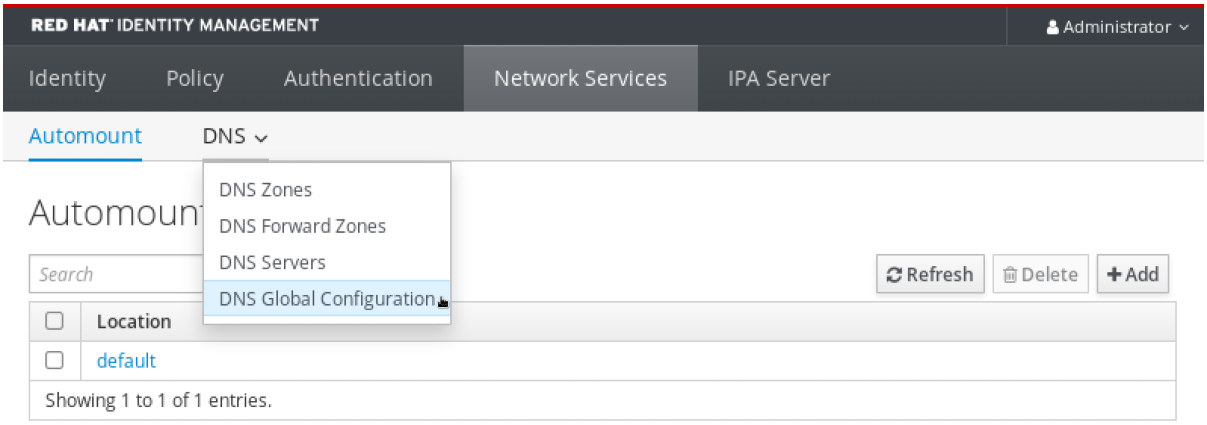
4. **Forward policy** を選択します。



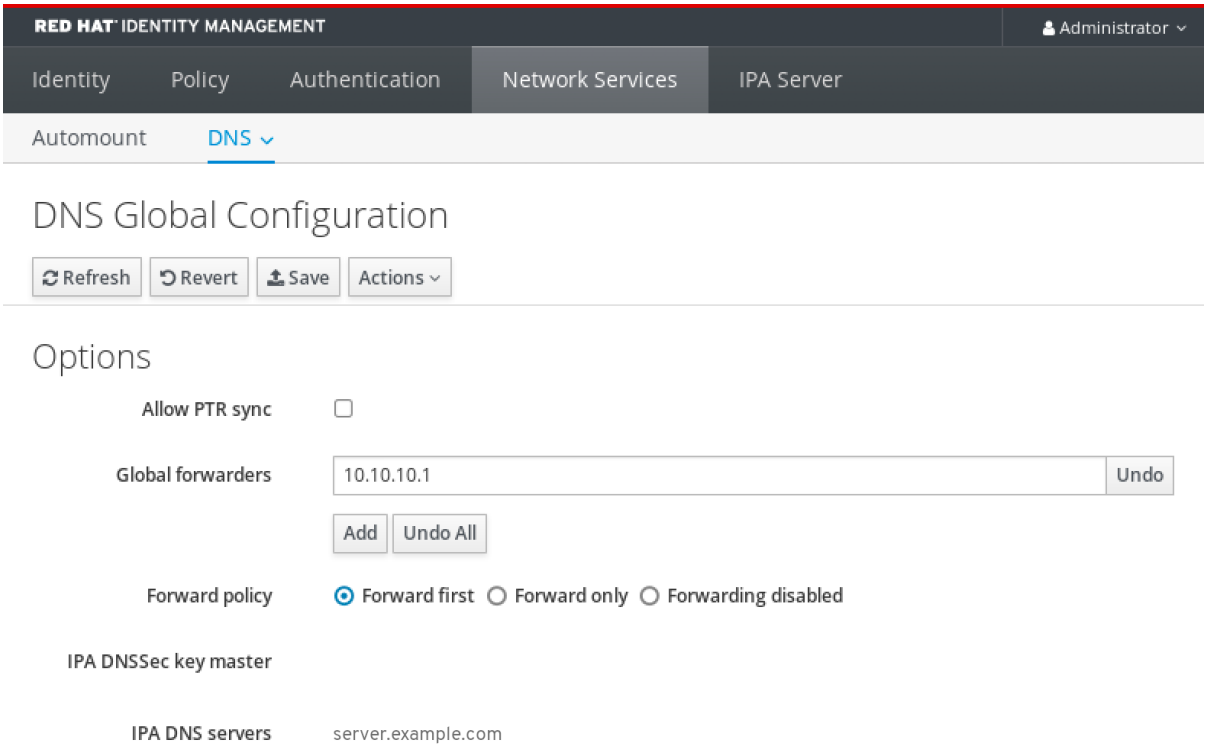
5. ウィンドウの上部にある **Save** をクリックします。

検証手順

1. **Network Services → DNS Global Configuration → DNS** の順に選択します。



2. 指定した転送ポリシーで、グローバルフォワーダーが IdM Web UI で存在し、有効化されていることを確認します。



30.4. CLI でのグローバルフォワーダーの追加

コマンドラインインターフェイス(CLI)を使用してグローバル DNS フォワーダーを追加するには、以下の手順に従います。

前提条件

- IdM 管理者としてログインしている。
- クエリーを転送する DNS サーバーのインターネットプロトコル (IP) アドレスを知っている。

手順

- **ipa dnsconfig-mod** コマンドを使用して、新しいグローバルフォワーダーを追加します。**--forwarder** オプションで DNS フォワーダーの IP アドレスを指定します。

```
[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
```

検証手順

- **dnsconfig-show** コマンドを使用して、グローバルフォワーダーを表示します。

```
[user@server ~]$ ipa dnsconfig-show
  Global forwarders: 10.10.0.1
  IPA DNS servers: server.example.com
```

30.5. IDM WEB UI での DNS 正引きゾーンの追加

以下の手順に従って、Identity Management (IdM) Web UI に DNS 正引きゾーンを追加します。

重要

絶対に必要な場合を除き、正引きゾーンは使用しないでください。正引きゾーンは、標準的な解決策ではないので、正引きゾーンを使用すると予期しない動作が発生する可能性があります。正引きゾーンを使用する必要がある場合は、グローバル転送設定が優先されるように、正引きゾーンの使用を制限します。

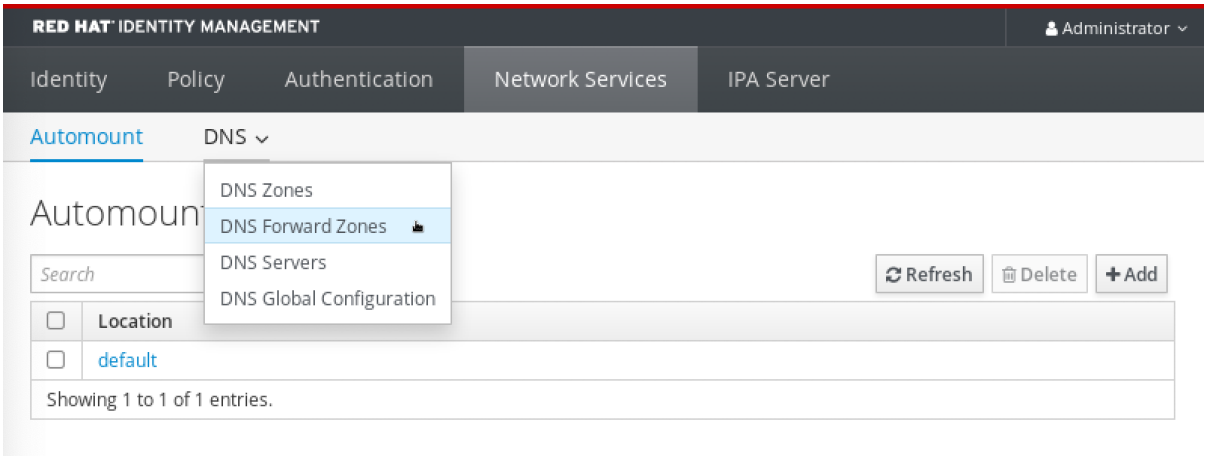
新しい DNS ゾーンを作成する場合には、Red Hat は、ネームサーバー (NS) レコードで標準の DNS 委譲を常に使用し、正引きゾーンを回避することを推奨します。多くの場合、グローバルフォワーダーを使用するだけで十分なため、正引きゾーンは必要ありません。

前提条件

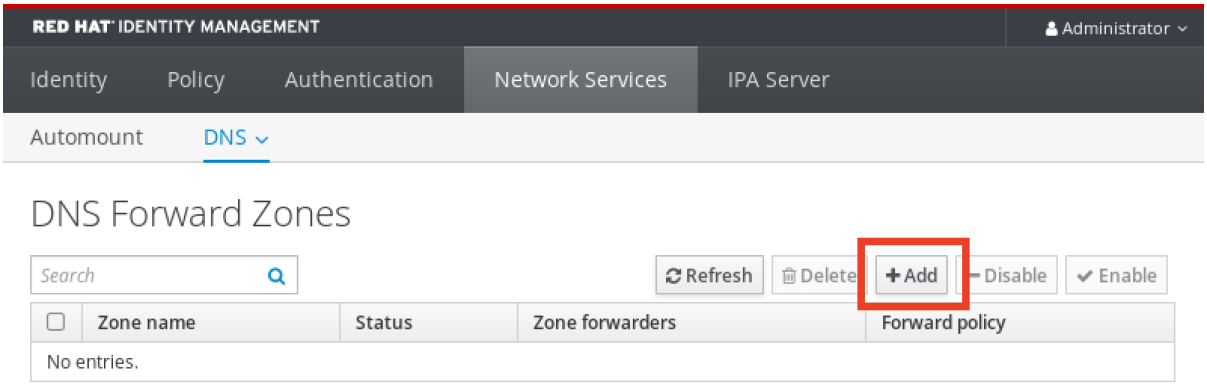
- IdM 管理者として IdM WebUI にログインしている。
- クエリーを転送する DNS サーバーのインターネットプロトコル (IP) アドレスを知っている。

手順

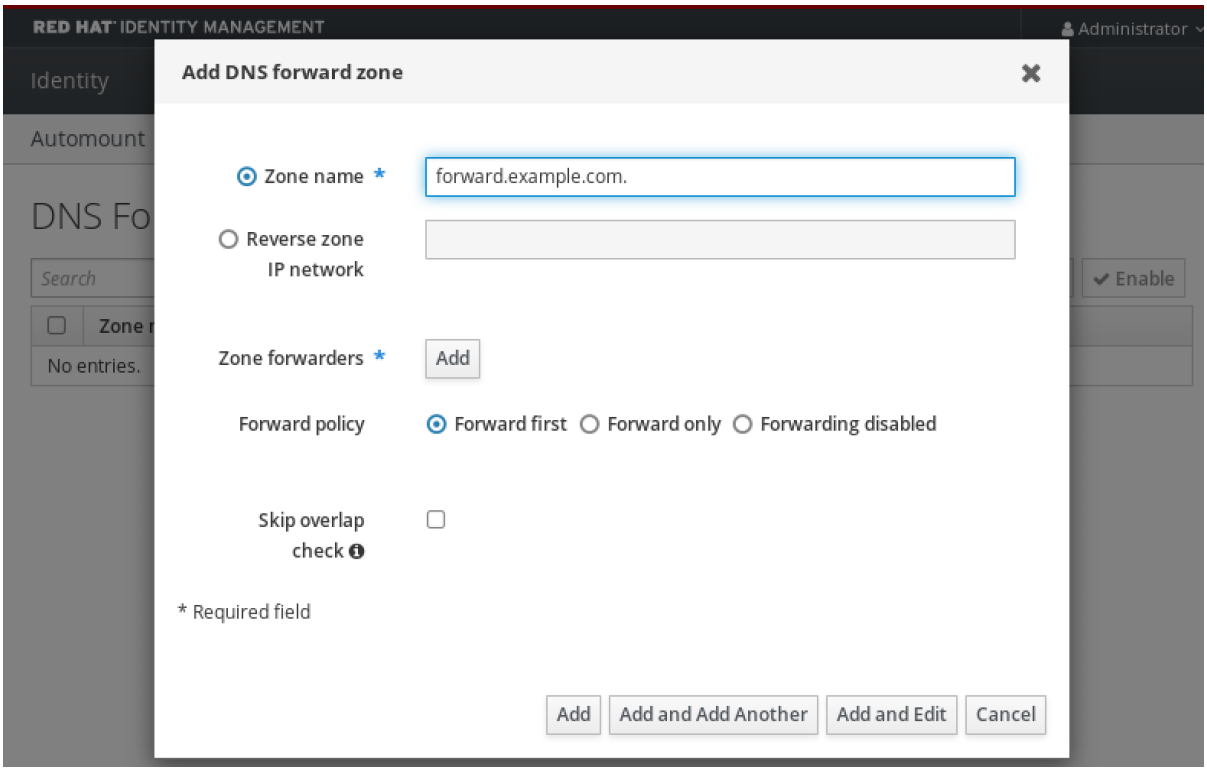
1. IdM Web UI で **Network Services → DNS Forward Zones → DNS** の順に選択します。



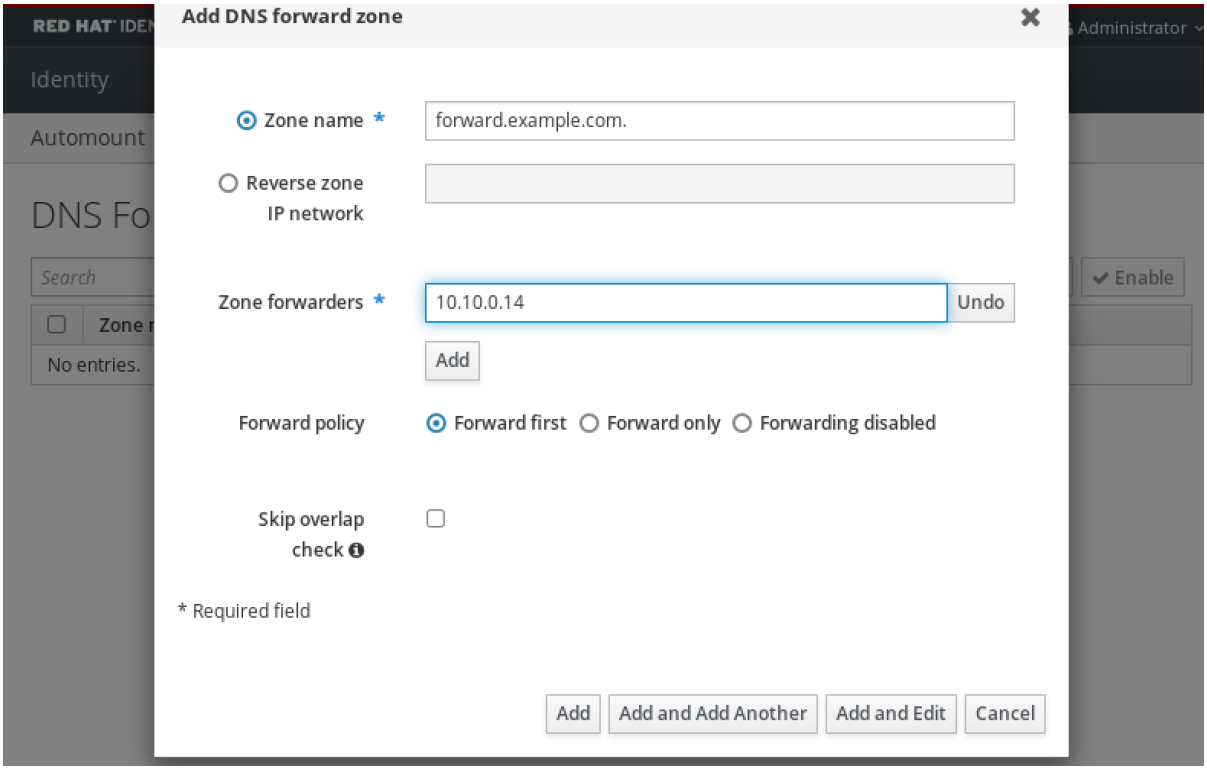
2. **DNS Forward Zones** セクションで、**Add** をクリックします。



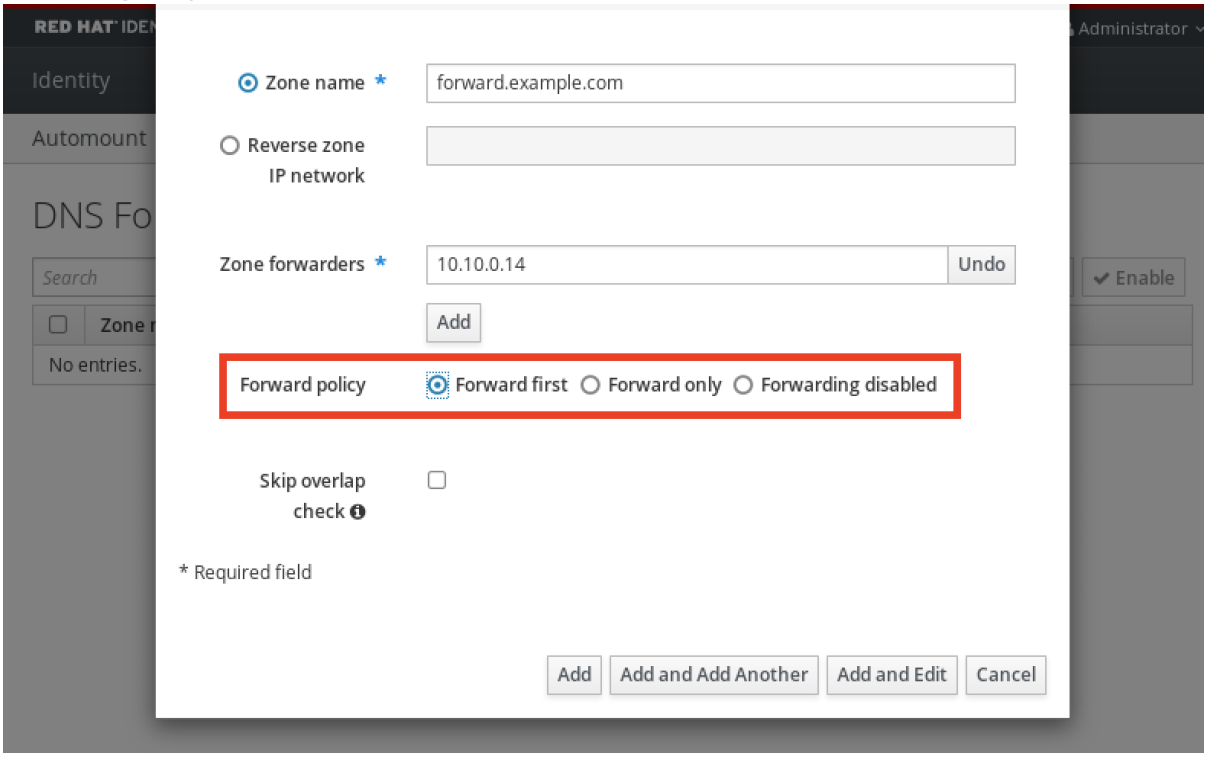
3. **Add DNS forward zone** ウィンドウで、正引きゾーン名を指定します。



4. **Add** ボタンをクリックして、転送要求を受信する DNS サーバーの IP アドレスを指定します。正引きゾーンごとに複数のフォワーダーを指定できます。



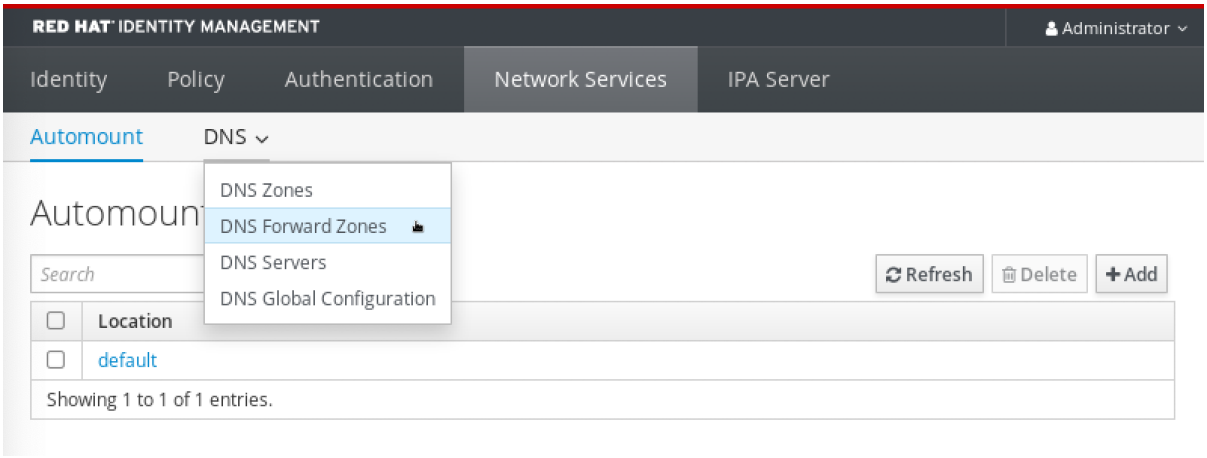
5. **Forward policy** を選択します。



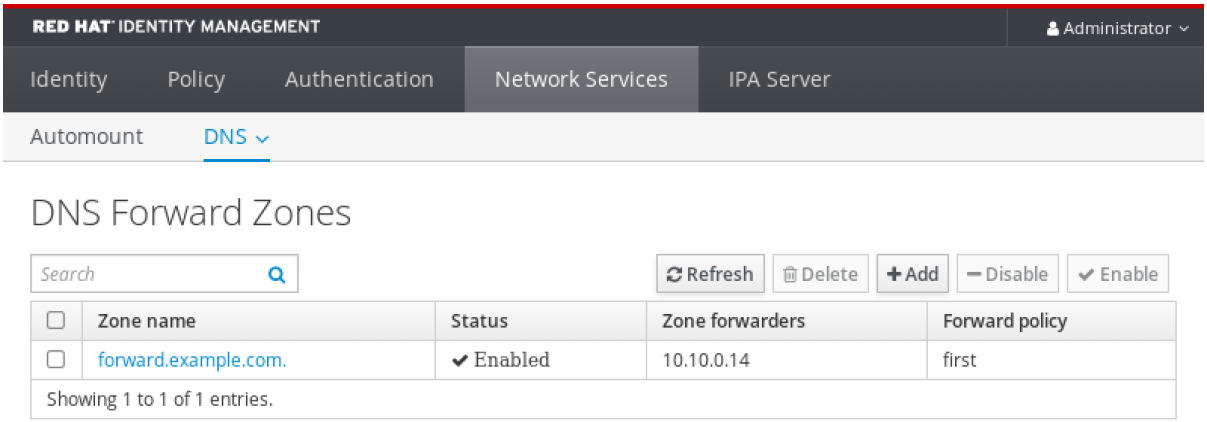
6. ウィンドウの下部にある **Add** をクリックして、新しい正引きゾーンを追加します。

検証手順

1. IdM Web UI で **Network Services** → **DNS Forward Zones** → **DNS** の順に選択します。



2. 指定したフォワーダーおよび転送ポリシーで、正引きゾーンが IdM Web UI で存在し、有効化されていることを確認します。



30.6. CLI での DNS 正引きゾーンの追加

コマンドラインインターフェイス(CLI)を使用して DNS 正引きゾーンを追加するには、以下の手順に従います。

重要

絶対に必要な場合を除き、正引きゾーンは使用しないでください。正引きゾーンは、標準的な解決策ではないので、正引きゾーンを使用すると予期しない動作が発生する可能性があります。正引きゾーンを使用する必要がある場合は、グローバル転送設定が優先されるように、正引きゾーンの使用を制限します。

新しい DNS ゾーンを作成する場合には、Red Hat は、ネームサーバー (NS) レコードで標準の DNS 委譲を常に使用し、正引きゾーンを回避することを推奨します。多くの場合、グローバルフォワーダーを使用するだけで十分なため、正引きゾーンは必要ありません。

前提条件

- IdM 管理者としてログインしている。
- クエリーを転送する DNS サーバーのインターネットプロトコル (IP) アドレスを知っている。

手順

- **dnsforwardzone-add** コマンドを使用して、新しい正引きゾーンを追加します。転送ポリシーが **none** ではない場合には、**--forwarder** オプションを使用して最低でもフォワーダーを 1 つ指定し、**--forward-policy** オプションで転送ポリシーを指定します。

```
[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
```

検証手順

- **dnsforwardzone-show** コマンドを使用して、作成した DNS 正引きゾーンを表示します。

```
[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
```

30.7. ANSIBLE を使用した IDM での DNS グローバルフォワーダーの確立

以下の手順に従って、Ansible Playbook を使用して IdM で DNS グローバルフォワーダーを確立します。

以下の手順の例では、IdM 管理者はポート **53** にインターネットプロトコル (IP) v4 アドレスが **8.8.6.6**、IPv6 アドレスが **2001:4860:4860::8800** で指定されている DNS サーバーに DNS グローバルフォワーダーを作成します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**set-configuration.yml**) のコピーを作成します。以下に例を示します。

```
$ cp set-configuration.yml establish-global-forwarder.yml
```

4. **establish-global-forwarder.yml** ファイルを開いて編集します。

5. 以下の変数を設定してファイルを調整します。

   a. Playbook の **name** 変数は、**IdM DNS でグローバルフォワーダーを確立する Playbook** の設定に変更します。

   b. **tasks** セクションで、タスクの **name** を **Create a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800** に変更します。

   c. **ipadnsconfig** の **forwarders** セクションで以下を行います。

      i. 最初の **ip_address** の値は、グローバルフォワーダーの IPv4 アドレス (**8.8.6.6**) に変更します。

      ii. 2 番目の **ip_address** の値は、グローバルフォワーダーの IPv6 アドレス (**2001:4860:4860::8800**) に変更します。

      iii. **port** の値が **53** に設定されていることを確認します。

   d. **forward_policy** を **first** に変更します。
   今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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: true
```

6. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file establish-global-forwarder.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsconfig.md** ファイルを参照してください。

30.8. ANSIBLE を使用して IDM に DNS グローバルフォワーダーを存在させる手順

以下の手順に従って、Ansible Playbook を使用して、IdM に DNS グローバルフォワーダーを追加します。以下の例では、IdM 管理者は、ポート **53** にインターネットプロトコル (IP) v4 アドレスが **7.7.9.9**、IPv6 アドレスが **2001:db8::1:0** で指定されている DNS サーバーに、DNS グローバルフォワーダーが配置されるようにします。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**forwarders-absent.yml**) のコピーを作成します。以下に例を示します。

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. **ensure-presence-of-a-global-forwarder.yml** ファイルを開いて編集します。

5. 以下の変数を設定してファイルを調整します。

   a. Playbook の **name** 変数は、**IdM DNS にグローバルフォワーダーを追加する Playbook** の設定に変更します。

   b. **tasks** セクションで、タスクの **name** を **Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0 on port 53** に変更します。

   c. **ipadnsconfig** の **forwarders** セクションで以下を行います。

      i. 最初の **ip_address** の値は、グローバルフォワーダーの IPv4 アドレス (**7.7.9.9**) に変更します。

      ii. 2 番目の **ip_address** の値は、グローバルフォワーダーの IPv6 アドレス (**2001:db8::1:0**) に変更します。

      iii. **port** の値が **53** に設定されていることを確認します。

   d. **state** を **present** に変更します。
   今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-
of-a-global-forwarder.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsconfig.md** ファイルを参照してください。

30.9. ANSIBLE を使用して IDM に DNS グローバルフォワーダーを存在させないようにする手順

以下の手順に従って、Ansible Playbook を使用して IdM で DNS グローバルフォワーダーを削除しま

す。以下の手順では、IdM 管理者が、ポート **53** で、IP (Internet Protocol) v4 アドレス **8.8.6.6** および IP v6 アドレス **2001:4860:4860::8800** を持つ DNS グローバルフォワーダーが存在しないことを確認します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**forwarders-absent.yml**) のコピーを作成します。以下に例を示します。

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. **ensure-absence-of-a-global-forwarder.yml** ファイルを開いて編集します。
5. 以下の変数を設定してファイルを調整します。
   a. Playbook の **name** 変数は、**IdM DNS でグローバルフォワーダーを配置しない Playbook** の設定に変更します。
   b. **tasks** セクションで、タスクの **name** を **Ensure the absence of a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800 on port 53** に変更します。
   c. **ipadnsconfig** の **forwarders** セクションで以下を行います。
      i. 最初の **ip_address** の値は、グローバルフォワーダーの IPv4 アドレス (**8.8.6.6**) に変更します。

ii. 2 番目の **ip_address** の値は、グローバルフォワーダーの IPv6 アドレス (**2001:4860:4860::8800**) に変更します。

iii. **port** の値が **53** に設定されていることを確認します。

d. **action** 変数は **member** に設定します。

e. **state** が **absent** に設定されていることを確認します。

今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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
```

重要

Playbook で **action: member** を使用せずに **state: absent** オプションだけを使用すると、その Playbook は失敗します。

6. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-of-a-global-forwarder.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsconfig.md** ファイル
- ipadnsconfig ansible-freeipa モジュールの **action: member** オプション

30.10. ANSIBLE を使用した IDM での DNS グローバルフォワーダーの無効化

Ansible Playbook を使用して、IdM で DNS グローバルフォワーダーを無効にするには、以下の手順に従います。以下の手順の例では、IdM の管理者がグローバルフォワーダーの転送ポリシーが **none** に設定されていることを確認し、グローバルフォワーダーを実質的に無効にします。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. 全 DNS グローバルフォワーダーを無効にするように設定済みの Ansible Playbook ファイル (**disable-global-forwarders.yml**) の内容を確認します。以下に例を示します。

```
$ 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. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disable-global-forwarders.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsconfig.md** ファイルを参照してください。

30.11. ANSIBLE を使用して IDM に DNS 正引きゾーンを存在させる手順

以下の手順に従って、Ansible Playbook を使用して IdM に DNS 正引きゾーンを追加します。以下の手順の例では、IdM 管理者は、インターネットプロトコル (IP) プロトコルが **8.8.8.8** の DNS サーバーに **example.com** の DNS 正引きゾーンが配置されるようにします。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**forwarders-absent.yml**) のコピーを作成します。以下に例を示します。

```
$ cp forwarders-absent.yml ensure-presence-forwardzone.yml
```

4. **ensure-presence-forwardzone.yml** ファイルを開いて編集します。
5. 以下の変数を設定してファイルを調整します。
   a. Playbook の **name** 変数は、**IdM DNS に DNS 正引きゾーンを追加する Playbook** の設定に変更します。
   b. **tasks** セクションで、タスクの **name** を **Ensure presence of a dnsforwardzone for example.com to 8.8.8.8** に変更します。

c. **tasks** セクションで、**ipadnsconfig** のヘディングを **ipadnsforwardzone** に変更します。

d. **ipadnsforwardzone** セクションで以下を実行します。

i. **ipaadmin_password** 変数を追加して、IdM 管理者パスワードに設定します。

ii. **name** 変数を追加して **example.com** に設定します。

iii. **forwarders** セクションで、以下を実行します。

A. **ip_address** と **port** の行を削除します。

B. 転送要求を受信できるように DNS サーバーの IP アドレスをダッシュの後に指定して追加します。

```
- 8.8.8.8
```

iv. **forwardpolicy** 変数を追加して **first** に設定します。

v. **skip_overlap_check** 変数を追加し、**true** に設定します。

vi. **state** 変数は **present** に変更します。

今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-forwardzone.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsforwardzone.md** ファイルを参照してください。

30.12. ANSIBLE を使用して IDM で DNS 正引きゾーンを複数配置する手順

以下の手順に従って、Ansible Playbook を使用して、IdM の DNS 正引きゾーンに複数のフォワーダーがあることを確認します。以下の手順の例では、IdM 管理者が **example.com** の DNS 正引きゾーンが **8.8.8.8** と **4.4.4.4** に転送されるようにします。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**forwarders-absent.yml**) のコピーを作成します。以下に例を示します。

```
$ cp forwarders-absent.yml ensure-presence-multiple-forwarders.yml
```

4. **ensure-presence-multiple-forwarders.yml** ファイルを開いて編集します。
5. 以下の変数を設定してファイルを調整します。
   a. Playbook の **name** 変数は、**IdM DNS の DNS 正引きゾーンに複数のフォワーダーを配置する Playbook** の設定に変更します。
   b. **tasks** セクションで、タスクの **name** を **Ensure presence of 8.8.8.8 and 4.4.4.4 forwarders in dnsforwardzone for example.com** に変更します。
   c. **tasks** セクションで、**ipadnsconfig** のヘディングを **ipadnsforwardzone** に変更します。
   d. **ipadnsforwardzone** セクションで以下を実行します。
      i. **ipaadmin_password** 変数を追加して、IdM 管理者パスワードに設定します。

ii. **name** 変数を追加して **example.com** に設定します。

iii. **forwarders** セクションで、以下を実行します。

A. **ip_address** と **port** の行を削除します。

B. 配置する DNS サーバーの IP アドレスを、前にダッシュをつけて追加します。

```
- 8.8.8.8
- 4.4.4.4
```

iv. state 変数を present に変更します。

今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-
multiple-forwarders.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsforwardzone.md** ファイルを参照してください。

30.13. ANSIBLE を使用して IDM で DNS 正引きゾーンを無効にする手順

Ansible Playbook を使用して IdM で DNS 正引きゾーンを無効にするには、以下の手順に従います。以下の手順の例では、IdM 管理者は **example.com** の DNS 正引きゾーンが無効になっていることを確認します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。

- Ansible バージョン 2.14 以降を使用している。
- Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
- この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**forwarders-absent.yml**) のコピーを作成します。以下に例を示します。

```
$ cp forwarders-absent.yml ensure-disabled-forwardzone.yml
```

4. **ensure-disabled-forwardzone.yml** ファイルを開いて編集します。
5. 以下の変数を設定してファイルを調整します。
   a. Playbook の **name** 変数は、**IdM DNS に DNS 正引きゾーンを無効にする Playbook** の設定に変更します。
   b. **tasks** セクションで、タスクの **name** を **Ensure a dnsforwardzone for example.com is disabled** に変更します。
   c. **tasks** セクションで、**ipadnsconfig** のヘディングを **ipadnsforwardzone** に変更します。
   d. **ipadnsforwardzone** セクションで以下を実行します。
      i. **ipaadmin_password** 変数を追加して、IdM 管理者パスワードに設定します。
      ii. **name** 変数を追加して **example.com** に設定します。
      iii. **forwarders** セクション全体を削除します。
      iv. **state** 変数を **disabled** に変更します。

   今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-disabled-forwardzone.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsforwardzone.md** ファイルを参照してください。

30.14. ANSIBLE を使用して IDM から DNS 正引きゾーンを削除する手順

Ansible Playbook を使用して IdM に DNS 正引きゾーンを削除するには、以下の手順に従います。以下の例では、IdM 管理者は **example.com** の DNS 正引きゾーンを削除します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsconfig** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**forwarders-absent.yml**) のコピーを作成します。以下に例を示します。

```
$ cp forwarders-absent.yml ensure-absence-forwardzone.yml
```

4. **ensure-absence-forwardzone.yml** ファイルを開いて編集します。

5. 以下の変数を設定してファイルを調整します。

   a. Playbook の **name** 変数は、**IdM DNS に DNS 正引きゾーンを削除する Playbook** の設定に変更します。

   b. **tasks** セクションで、タスクの **name** を **Ensure the absence of a dnsforwardzone for example.com** に変更します。

   c. **tasks** セクションで、**ipadnsconfig** のヘディングを **ipadnsforwardzone** に変更します。

   d. **ipadnsforwardzone** セクションで以下を実行します。

      i. **ipaadmin_password** 変数を追加して、IdM 管理者パスワードに設定します。

      ii. **name** 変数を追加して **example.com** に設定します。

      iii. **forwarders** セクション全体を削除します。

      iv. **state** 変数を **absent** のままにします。

   今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-
forwardzone.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsforwardzone.md** ファイルを参照してください。

第31章 ANSIBLE を使用した IDM での DNS レコードの管理

本章では、Ansible Playbook を使用して Identity Management (IdM) で DNS レコードを管理する方法を説明します。IdM 管理者は、IdM で DNS レコードの追加、変更、および削除が可能です。この章には次のセクションが含まれています。

- Ansible を使用して IdM に A および AAAA DNS レコードが存在させる手順
- Ansible を使用して IdM に A および PTR DNS レコードを存在させる手順
- Ansible を使用して IdM に複数の DNS レコードを存在させる手順
- Ansible を使用して IdM に複数の CNAME レコードを存在させる手順
- Ansible を使用して IdM に SRV レコードを存在させる手順

31.1. IDM の DNS レコード

Identity Management (IdM) は、多種の DNS レコードに対応します。以下の 4 つが最も頻繁に使用されます。

A

これは、ホスト名および IPv4 アドレスの基本マップです。A レコードのレコード名は、**www** などのホスト名です。A レコードの **IP アドレス** 値は、**192.0.2.1** などの IPv4 アドレスです。
A レコードの詳細は、RFC 1035 を参照してください。

AAAA

これは、ホスト名および IPv6 アドレスの基本マップです。AAAA レコードのレコード名は **www** などのホスト名です。**IP アドレス** の値は、**2001:DB8::1111** などの IPv6 アドレスです。
AAAA レコードの詳細は RFC 3596 を参照してください。

SRV

サービス (SRV) リソースレコード は、特定のサービスを提供するサーバーの DNS 名にサービス名をマッピングします。たとえば、このタイプのレコードは LDAP ディレクトリーのようなサービスを管理するサーバーに、このサービスをマッピングします。
SRV レコードのレコード名は、**_ldap._tcp** など、**_service._protocol** の形式を取ります。SRV レコードの設定オプションには、ターゲットサービスの優先順位、加重、ポート番号、およびホスト名が含まれます。

SRV レコードの詳細は、RFC 2782 を参照してください。

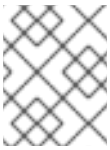
PTR

ポインターレコード (PTR) は、IP アドレスをドメイン名にマッピングする逆引き DNS レコードを追加します。

注記

IPv4 アドレスの逆引き DNS ルックアップはすべて、**in-addr.arpa.** ドメインで定義される逆引きエントリーを使用します。人間が判別可能な形式の逆アドレスは、通常の IP とまったく逆で、**in-addr.arpa.** ドメインが最後に付いています。たとえば、ネットワークアドレス **192.0.2.0/24** の逆引きゾーンは、**2.0.192.in-addr.arpa** になります。

PTR レコード名は、RFC 1035 (RFC 2317 および RFC 3596 で拡張) で指定の標準形式を仕様する必要があります。ホスト名の値は、レコードを作成するホストの正規のホスト名である必要があります。



注記

また、IPv6 アドレスの逆引きゾーンは、**.ip6.arpa.** ドメインのゾーンを使用して設定できます。IPv6 逆引きゾーンの詳細は、RFC 3596 を参照してください。

DNS リソースレコードの追加時には、レコードの多くで異なるデータが必要になることに注意してください。たとえば、CNAME レコードにはホスト名が必要ですが、A レコードには IP アドレスが必要です。IdM Web UI では、新しいレコードを追加するフォームのフィールドが自動的に更新され、現在選択されているレコードタイプに必要なデータが反映されます。

31.2. 一般的な IPA DNSRECORD-* オプション

Identity Management (IdM) で最も一般的な DNS リソースレコードタイプを追加、変更、および削除する場合は、以下のオプションを使用できます。

- A (IPv4)
- AAAA (IPv6)
- SRV
- PTR

Bash では、**--option={val1,val2,val3}** など、波括弧の中にコンマ区切りで値を指定して、複数のエントリーを定義できます。

表31.1 全般的なレコードのオプション

オプション	説明
--ttl=number	レコードの有効期間を設定します。
--structured	raw DNS レコードを解析し、それらを構造化された形式で返します。

表31.2 "A" レコードのオプション

オプション	説明	例
--a-rec=ARECORD	A レコードを 1 つまたはリストで指定します。	**ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123**
	指定の IP アドレスでワイルドカード A レコードを作成できます。	**ipa dnsrecord-add idm.example.com "*" --a-rec=192.168.122.123** [a]

オプション	説明	例
--a-ip-address=string	レコードの IP アドレスを渡します。レコードの作成時に、**A** レコードの値を指定するオプションは **--a-rec** です。ただし **A** レコードを変更する時に、**--a-rec** オプションを使用して **A** レコードの現在の値を指定します。新しい値は、**--a-ip-address** オプションで設定します。	**ipa dnsrecord-mod idm.example.com --a-rec 192.168.122.123 --a-ip-address 192.168.122.124**
[a] この例では、IP アドレスが 192.0.2.123 のワイルドカード **A** レコードを作成します。		

表31.3 "AAAA" レコードのオプション

オプション	説明	例
--aaaa-rec=AAAARECORD	AAAA (IPv6) レコードを 1 つまたはリストで指定します。	**ipa dnsrecord-add idm.example.com www --aaaa-rec 2001:db8::1231:5675**
--aaaa-ip-address=string	レコードの IPv6 アドレスを渡します。レコードの作成時に、**A** レコードの値を指定するオプションは **--aaaa-rec** です。ただし、**A** レコードを変更する時に、**--aaaa-rec** オプションを使用して **A** レコードの現在の値を指定します。新しい値は、**--a-ip-address** オプションで設定します。	**ipa dnsrecord-mod idm.example.com --aaaa-rec 2001:db8::1231:5675 --aaaa-ip-address 2001:db8::1231:5676**

表31.4 "PTR" レコードのオプション

オプション	説明	例
--ptr-rec=PTRRECORD	PTR レコードを 1 つまたはリストで指定します。逆引き DNS レコードを追加する時には、他の DNS レコードの追加の方法と比べ、**ipa dnsrecord-add** コマンドで使用するゾーン名は、逆になります。通常、ホストの IP アドレスは、指定のネットワークにおける IP アドレスの最後のオクテットを使用します。右側の最初の例では、IPv4 アドレスが 192.168.122.4. の server4.idm.example.com の PTR レコードを追加します。2 番目の例では、0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. に逆引き DNS エントリーを追加します。IP アドレスが 2001:DB8::1111 の server2.example.com ホストの IPv6 逆引きゾーン。	**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 --ptr-rec server2.idm.example.com.**
--ptr-hostname=string	レコードのホスト名を指定します。	

表31.5 "SRV" レコードのオプション

オプション	説明	例
--srv-rec=SRVRECORD	SRV レコードを 1 つまたはリストで指定します。右側の例では、_ldap._tcp は、SRV レコードのサービスタイプと接続プロトコルを定義します。**--srv-rec** オプションは、優先順位、加重、ポート、およびターゲットの値を定義します。この例では、加重値 51 と 49 が最大 100 まで加算され、特定のレコードが使用される確率を % で表します。	**# 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	レコードの優先順位を設定します。あるサービスタイプに複数の SRV レコードがある場合もあります。優先順位 (0 - 65535) はレコードの階級を設定し、数字が小さいほど優先順位が高くなります。サービスは、優先順位の最も高いレコードを最初に使用する必要があります。	**# ipa dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="1 49 389 server2.idm.example.com." --srv-priority=0**
--srv-weight=number	レコードの加重を設定します。これは、SRV レコードの優先順位が同じ場合に順序を判断する際に役立ちます。設定された加重は最大 100 とし、これは特定のレコードが使用される可能性をパーセンテージで示しています。	**# ipa dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 49 389 server2.idm.example.com." --srv-weight=60**
--srv-port=number	ターゲットホスト上のサービスのポートを渡します。	**# ipa dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 60 389 server2.idm.example.com." --srv-port=636**
--srv-target=string	ターゲットホストのドメイン名を提供します。該当サービスがドメイン内で利用可能でない場合は、単一のピリオド (.) として指定される場合があります。	

関連情報

- **ipa dnsrecord-add --help** を実行します。

31.3. ANSIBLE を使用して IDM に A および AAAA DNS レコードが存在させる手順

Ansible Playbook を使用して、特定の IdM ホストの A および AAAA レコードが存在することを確認するには、以下の手順に従います。以下の手順で使用する例では、IdM 管理者は **idm.example.com** DNS ゾーンに **host1** の A レコードおよび AAAA レコードを追加します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。
- **idm.example.com** ゾーンが存在しており、IdM DNS が管理する。IdM DNS にプライマリー DNS ゾーンを追加する方法は、Ansible Playbook を使用した IdM DNS ゾーンの管理 を参照してください。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**ensure-A-and-AAAA-records-are-present.yml**) のコピーを作成します。以下に例を示します。

```
$ cp ensure-A-and-AAAA-records-are-present.yml ensure-A-and-AAAA-records-are-present-copy.yml
```

4. **ensure-A-and-AAAA-records-are-present-copy.yml** ファイルを開いて編集します。
5. **ipadnsrecord** タスクセクションで以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **zone_name** 変数は **idm.example.com** に設定します。
   - **record** 変数で、**name** 変数は **host1** に、**a_ip_address** 変数は 192.168.122.123 に設定します。
   - **record** 変数で、**name** 変数は **host1** に、**aaaa_ip_address** 変数は **::1** に設定します。
     以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-A-and-AAAA-records-are-present-copy.yml
```

関連情報

- IdM の DNS レコード を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsrecord.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** ディレクトリーのサンプルの Ansible Playbook を参照してください。

31.4. ANSIBLE を使用して IDM に A および PTR DNS レコードを存在させる手順

以下の手順に従って、Ansible Playbook を使用して、特定の IdM ホストの A レコードと、対応する PTR レコードが存在することを確認します。以下の手順で使用する例では、IdM 管理者は、**idm.example.com** ゾーンで IP アドレスが **192.168.122.45** の **host1** の A レコードと PTR レコードを追加します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。

    - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。
- **idm.example.com** DNS ゾーンが存在しており、IdM DNS が管理する。IdM DNS にプライマリー DNS ゾーンを追加する方法は、Ansible Playbook を使用した IdM DNS ゾーンの管理 を参照してください。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**ensure-dnsrecord-with-reverse-is-present.yml**) のコピーを作成します。以下に例を示します。

```
$ cp ensure-dnsrecord-with-reverse-is-present.yml ensure-dnsrecord-with-reverse-is-present-copy.yml
```

4. **ensure-dnsrecord-with-reverse-is-present-copy.yml** ファイルを開いて編集します。
5. **ipadnsrecord** タスクセクションで以下の変数を設定して、ファイルを調整します。
    - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
    - **name** 変数は **host1** に設定します。
    - **zone_name** 変数は **idm.example.com** に設定します。
    - **ip_address** 変数は、**192.168.122.45** に設定します。
    - **create_reverse** 変数を **true** に設定します。
      以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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: true
state: present
```

6. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-dnsrecord-with-reverse-is-present-copy.yml
```

関連情報

- IdM の DNS レコード を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsrecord.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** ディレクトリーのサンプルの Ansible Playbook を参照してください。

31.5. ANSIBLE を使用して IDM に複数の DNS レコードを存在させる手順

Ansible Playbook を使用して、複数の値が特定の IdM DNS レコードに関連付けられるようにするには、以下の手順に従います。以下の手順で使用する例では、IdM 管理者は **idm.example.com** DNS ゾーンに **host1** の A レコードを複数追加します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。
- **idm.example.com** ゾーンが存在しており、IdM DNS が管理する。IdM DNS にプライマリー DNS ゾーンを追加する方法は、Ansible Playbook を使用した IdM DNS ゾーンの管理 を参照してください。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**ensure-presence-multiple-records.yml**) のコピーを作成します。以下に例を示します。

```
$ cp ensure-presence-multiple-records.yml ensure-presence-multiple-records-copy.yml
```

4. **ensure-presence-multiple-records-copy.yml** ファイルを開いて編集します。

5. **ipadnsrecord** タスクセクションで以下の変数を設定して、ファイルを調整します。

   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **records** セクションで、**name** 変数を **host1** に設定します。
   - **record** セクションで、**zone_name** 変数を **idm.example.com** に設定します。
   - **record** セクションで、**a_rec** 変数を 192.168.122.112 に、192.168.122.122 **に** 設定し **ます**。
   - **records** セクションの 2 番目のレコードを定義します。
     - **name** 変数は **host1** に設定します。
     - **zone_name** 変数は **idm.example.com** に設定します。
     - **aaaa_rec** 変数は **::1** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-multiple-records-copy.yml
```

関連情報

- IdM の DNS レコード を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsrecord.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** ディレクトリーのサンプルの Ansible Playbook を参照してください。

31.6. ANSIBLE を使用して IDM に複数の CNAME レコードを存在させる手順

Canonical Name レコード (CNAME レコード) は、DNS (Domain Name System) のリソースレコードの一種で、別の名前 (CNAME) にドメイン名、エイリアスをマッピングします。

CNAME レコードは、FTP サービスと Web サービスがそれぞれ別のポートで実行されている場合など、1 つの IP アドレスから複数のサービスを実行する場合に、役立つ可能性があります。

Ansible Playbook を使用して、複数の CNAME レコードが IdM DNS に存在することを確認するには、以下の手順に従います。以下の手順で使用する例では、**host03** は HTTP サーバーと FTP サーバーの両方として機能します。IdM 管理者は、**idm.example.com** ゾーンに **host03** A レコードの **www** および **ftp** CNAME レコードを追加します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
- IdM 管理者パスワードを把握している。

- **idm.example.com** ゾーンが存在しており、IdM DNS が管理する。IdM DNS にプライマリー DNS ゾーンを追加する方法は、Ansible Playbook を使用した IdM DNS ゾーンの管理 を参照してください。
- **host03** A レコードが **idm.example.com** ゾーンに存在している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**ensure-CNAME-record-is-present.yml**) のコピーを作成します。以下に例を示します。

```
$ cp ensure-CNAME-record-is-present.yml ensure-CNAME-record-is-present-copy.yml
```

4. **ensure-CNAME-record-is-present-copy.yml** ファイルを開いて編集します。
5. **ipadnsrecord** タスクセクションで以下の変数を設定して、ファイルを調整します。
   - (任意) Play の **name** で提示された説明を調整します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **zone_name** 変数は **idm.example.com** に設定します。
   - **record** 変数セクションで、以下の変数および値を設定します。
     - **name** 変数は **www** に設定します。
     - **cname_hostname** 変数は **host03** に設定します。
     - **name** 変数は **ftp** に設定します。
     - **cname_hostname** 変数は **host03** に設定します。

   以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. ファイルを保存します。

7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-CNAME-record-is-present.yml
```

関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsrecord.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** ディレクトリーのサンプルの Ansible Playbook を参照してください。

31.7. ANSIBLE を使用して IDM に SRV レコードを存在させる手順

DNS サービス (SRV) レコードは、ドメインで利用可能なサービスのホスト名、ポート番号、トランスポートプロトコル、優先度、および加重を定義します。Identity Management (IdM) では、SRV レコードを使用して、IdM サーバーとレプリカを特定できます。

以下の手順に従って、Ansible Playbook を使用して、SRV レコードが IdM DNS に存在することを確認します。以下の手順で使用する例では、IdM の管理者が **10 50 88 idm.example.com** の値を指定して **_kerberos._udp.idm.example.com** SRV レコードを追加します。この例では、以下の値を指定します。

- サービスの優先度を 10 に設定します。
- サービスの加重を 50 に設定します。
- サービスが使用するポートを 88 に設定します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

- IdM 管理者パスワードを把握している。
- **idm.example.com** ゾーンが存在しており、IdM DNS が管理する。IdM DNS にプライマリー DNS ゾーンを追加する方法は、Ansible Playbook を使用した IdM DNS ゾーンの管理 を参照してください。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

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

3. Ansible Playbook ファイル (**ensure-SRV-record-is-present.yml**) のコピーを作成します。以下に例を示します。

```
$ cp ensure-SRV-record-is-present.yml ensure-SRV-record-is-present-copy.yml
```

4. **ensure-SRV-record-is-present-copy.yml** ファイルを開いて編集します。
5. **ipadnsrecord** タスクセクションで以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は **_kerberos._udp.idm.example.com** に設定します。
   - **srv_rec** 変数は **'10 50 88 idm.example.com'** に設定します。
   - **zone_name** 変数は **idm.example.com** に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- 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. ファイルを保存します。
7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-SRV-record-is-present.yml
```

関連情報

- IdM の DNS レコード を参照してください。
- **/usr/share/doc/ansible-freeipa/** ディレクトリーの **README-dnsrecord.md** ファイルを参照してください。
- **/usr/share/doc/ansible-freeipa/playbooks/dnsrecord** ディレクトリーのサンプルの Ansible Playbook を参照してください。

第32章 ANSIBLE を使用して IDM ユーザーの NFS 共有を自動マウントする

自動マウントは、複数のシステムにわたってディレクトリーを管理、整理、およびアクセスする方法です。Automount は、ディレクトリーへのアクセスが要求されるたびに、そのディレクトリーを自動的にマウントします。これは、ドメイン内のクライアント上のディレクトリーを簡単に共有できるため、Identity Management (IdM) ドメイン内でうまく機能します。

Ansible を使用して、IdM ロケーションの IdM クライアントにログインしている IdM ユーザーに対して NFS 共有が自動的にマウントされるように設定できます。

この章の例では、次のシナリオを使用します。

- **nfs-server.idm.example.com** は、ネットワークファイルシステム (NFS) サーバーの完全修飾ドメイン名 (FQDN) です。
- **nfs-server.idm.example.com** は、**raleigh** の自動マウントの場所にある IdM クライアントです。
- NFS サーバーは、**/exports/project** ディレクトリーを読み取り/書き込みとしてエクスポートします。
- **開発者** グループに属する IdM ユーザーは、NFS サーバーと同じ **raleigh**の自動マウントの場所にある IdM クライアントであれば、**/devel/project/** としてエクスポートしたディレクトリーのコンテンツにアクセスできます。
- **idm-client.idm.example.com** は、**raleigh** の自動マウントの場所にある IdM クライアントです。

重要

NFS サーバーの代わりに Samba サーバーを使用して IdM クライアントに共有を提供する場合は、以下の How do I configure kerberized CIFS mounts with Autofs in an IPA environment? を参照してください。を参照してください。

この章には次のセクションが含まれています。

1. IdM の Autofs と automount
2. IdM で Kerberos を使用する NFS サーバーをセットアップする
3. Ansible を使用した IdM での自動マウントの場所、マップ、およびキーの設定
4. Ansible を使用して IdM ユーザーを NFS 共有を所有するグループに追加する
5. IdM クライアントでの自動マウントの設定
6. IdM クライアントで、IdM ユーザーが NFS 共有にアクセスできることの確認

32.1. IDM の AUTOFS と AUTOMOUNT

autofs サービスは、アクセス時にディレクトリーをマウントするように **automount** デーモンに指示することにより、必要に応じてディレクトリーのマウントを自動化します。また、しばらく操作を行わな

いと、**autofs** は、**automount** に自動マウントされたディレクトリーのマウントを解除するように指示します。静的マウントとは異なり、オンデマンドマウントはシステムリソースを節約します。

マップの自動マウント

autofs を使用するシステムでは、**automount** 設定は複数のファイルに保存されます。プライマリー **automount** 設定ファイルは **/etc/auto.master** です。これには、システムの **automount** マウントポイントのマスターマッピングと、その関連リソースが含まれます。このマッピングは **自動マウントマップ** として知られています。
/etc/auto.master 設定ファイルには、**マスターマップ** が含まれます。他のマップへの参照を含めることができます。このマップは、直接または間接のいずれかになります。ダイレクトマップではマウントポイントに絶対パス名を使用し、間接マップでは相対パス名を使用します。

IdM の自動マウント設定

automount は通常、ローカルの **/etc/auto.master** と関連ファイルからマップデータを取得しますが、他のソースからマップデータを取得することもできます。一般的なソースの 1 つが LDAP サーバーです。Identity Management (Id M) のコンテキストでは、これは 389 Directory Server です。
autofs を使用するシステムが IdM ドメインのクライアントである場合、**automount** 設定はローカル設定ファイルに保存されません。代わりに、マップ、場所、キーなどの **autofs** 設定は、LDAP エントリーとして IdM ディレクトリーに保存されます。たとえば、**idm.example.com** IdM ドメインの場合、デフォルトの **マスターマップ** は以下のように保存されます。

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

関連情報

- オンデマンドでのファイルシステムのマウント

32.2. RED HAT IDENTITY MANAGEMENT ドメインで KERBEROS を使用する NFS サーバーをセットアップする

Red Hat Identity Management (IdM) を使用すると、NFS サーバーを IdM ドメインに参加させることができます。これにより、ユーザーとグループを一元管理し、認証、整合性保護、トラフィック暗号化に Kerberos を使用できるようになります。

前提条件

- NFS サーバーが Red Hat Identity Management (IdM) ドメインに 登録 されている。
- NFS サーバーが実行および設定されている。

手順

1. IdM 管理者として Kerberos チケットを取得します。

```
# kinit admin
```

2. **nfs/<FQDN>** サービスプリンシパルを作成します。

```
# ipa service-add nfs/nfs_server.idm.example.com
```

3. IdM から **nfs** サービスプリンシパルを取得し、**/etc/krb5.keytab** ファイルに保存します。

```
# ipa-getkeytab -s idm_server.idm.example.com -p nfs/nfs_server.idm.example.com -k
/etc/krb5.keytab
```

4. オプション: **/etc/krb5.keytab** ファイル内のプリンシパルを表示します。

```
# klist -k /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   1 nfs/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
   7 host/nfs_server.idm.example.com@IDM.EXAMPLE.COM
```

デフォルトでは、ホストを IdM ドメインに参加させると、IdM クライアントがホストプリンシパルを **/etc/krb5.keytab** ファイルに追加します。ホストプリンシパルがない場合は、**ipa-getkeytab -s idm_server.idm.example.com -p host/nfs_server.idm.example.com -k /etc/krb5.keytab** コマンドを使用して追加します。

5. **ipa-client-automount** ユーティリティーを使用して、IdM ID のマッピングを設定します。

```
#  ipa-client-automount
Searching for IPA server...
IPA server: DNS discovery
Location: default
Continue to configure the system with these values? [no]: yes
Configured /etc/idmapd.conf
Restarting sssd, waiting for it to become available.
Started autofs
```

6. **/etc/exports** ファイルを更新し、クライアントオプションに Kerberos セキュリティー方式を追加します。以下に例を示します。

```
/nfs/projects/       192.0.2.0/24(rw,sec=krb5i)
```

クライアントが複数のセキュリティー方式を選択できるようにするには、それらをコロンで区切って指定します。

```
/nfs/projects/       192.0.2.0/24(rw,sec=krb5:krb5i:krb5p)
```

7. エクスポートされたファイルシステムを再ロードします。

```
# exportfs -r
```

32.3. ANSIBLE を使用した IDM での自動マウントの場所、マップ、およびキーの設定

Identity Management (IdM) のシステム管理者は、IdM で自動マウントの場所とマップを設定できます。これにより、指定した場所の IdM ユーザーが、ホストの特定のマウントポイントに移動して、NFS サーバーがエクスポートした共有にアクセスできるようになります。エクスポートされた NFS サーバーディレクトリーとマウントポイントの両方が、マップで指定されます。LDAP 用語では、ロケーションはそのようなマップエントリーのコンテナーです。

この例では、Ansible を使用して、**raleigh** の場所と、IdM クライアント上の **/devel/project** マウントポイントに **nfs-server.idm.example.com:/exports/project** 共有を読み取り/書き込みディレクトリーとしてマウントするマップを設定する方法を説明します。

前提条件

- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。

手順

1. Ansible コントロールノードで、**~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **/usr/share/doc/ansible-freeipa/playbooks/automount/** ディレクトリーにある **automount-location-present.yml** Ansible Playbook ファイルをコピーします。

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

3. **automount-location-map-and-key-present.yml** ファイルを編集用に開きます。
4. **ipaautomountlocation** タスクセクションで次の変数を設定して、ファイルを調整します。
   - **ipaadmin_password** 変数は IdM **admin** のパスワードに設定します。
   - **name** 変数を **raleigh** に設定します。
   - **state** 変数は **present** に設定されていることを確認します。
     以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- 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. **automount-location-map-and-key-present.yml** ファイルの編集を続けます。

   a. **tasks** セクションで、自動マウントマップの存在を確認するタスクを追加します。

```
[...]
  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. 別のタスクを追加して、マウントポイントと NFS サーバー情報をマップに追加します。

```
[...]
  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. auto.devel が auto.master に接続されていることを確認する別のタスクを追加します。

```
[...]
  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. ファイルを保存します。

7. Ansible Playbook を実行し、Playbook とインベントリーファイルを指定します。

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

32.4. ANSIBLE を使用した NFS 共有を所有するグループへの IDM ユーザーの追加

Identity Management (IdM) システム管理者は、Ansible を使用して、NFS 共有にアクセスできるユーザーのグループを作成し、IdM ユーザーをこのグループに追加できます。

この例では、Ansible Playbook を使用して **idm_user** アカウントが **developers** グループに属していることを確認し、**idm_user** が **/exports/project** NFS 共有にアクセスできるようにする方法について説明します。

前提条件

- **nfs-server.idm.example.com** NFS サーバーへの **root** アクセス権があり、これは **raleigh** automount の場所にある IdM クライアントです。
- IdM **admin** のパスワードを把握している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。
- ターゲットノード (**ansible-freeipa** モジュールが実行されるノード) が、IdM クライアント、サーバー、またはレプリカとして IdM ドメインに含まれている。
  - **~/MyPlaybooks/** で、Ansible を使用して IdM で automount の場所、マップ、およびキーを設定 するからのタスクがすでに含まれている **automount-location-map-and-key-present.yml** ファイルを作成しました。

手順

1. Ansible コントロールノードで、**~/MyPlaybooks/** ディレクトリーに移動します。

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

2. **automount-location-map-and-key-present.yml** ファイルを編集用に開きます。

3. **tasks** セクションで、IdM **developers** グループが存在し、**idm_user** がこのグループに追加されていることを確認するタスクを追加します。

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

4. ファイルを保存します。

5. Ansible Playbook を実行し、Playbook とインベントリーファイルを指定します。

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

6. NFS サーバーで、**/exports/project** ディレクトリーのグループ所有権を **developers** に変更して、グループ内のすべての IdM ユーザーがディレクトリーにアクセスできるようにします。

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

32.5. IDM クライアントでの自動マウントの設定

Identity Management (IdM) システム管理者は、IdM クライアントに自動マウントサービスを設定して、クライアントが追加された場所に設定した NFS 共有に、ユーザーがクライアントにログインしたときに IdM ユーザーが自動的にアクセスできるようにすることができます。この例では、**raleigh** の場所で利用可能な自動マウントサービスを使用するように IdM クライアントを設定する方法を説明します。

前提条件

- IdM クライアントへの **root** アクセス権限がある。
- IdM 管理者としてログインしている。
- 自動マウントの場所が存在します。サンプルの場所は **raleigh** です。

手順

1. IdM クライアントで、**ipa-client-automount** コマンドを入力して場所を指定します。**-U** オプションを使用して、スクリプトを無人で実行します。

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

2. autofs サービスを停止し、SSSD キャッシュをクリアし、autofs サービスを開始して新しい設定をロードします。

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

32.6. IDM クライアントで、IDM ユーザーが NFS 共有にアクセスできることの確認

Identity Management (IdM) システム管理者は、特定のグループのメンバーである IdM ユーザーが、特定の IdM クライアントにログインしたときに NFS 共有にアクセスできるかどうかをテストできます。

この例では、以下のシナリオがテストされています。

- **developers** グループに属する **idm_user** という名前の IdM ユーザーは、 **raleigh** automount ロケーションにある IdM クライアントである **idm-client.idm.example.com** に自動マウントされた **/devel/project** ディレクトリー内のファイルの内容を読み書きできます。

前提条件

- IdM ホスト上に Kerberos を使用して NFS サーバーをセットアップした 。
- IdM で自動マウントのロケーション、マップ、マウントポイントを設定 し、そこで IdM ユーザーが NFS共有にアクセスできるように設定している。
- Ansible を使用して、NFS 共有を所有する開発者グループに IdM ユーザーを追加しました 。
- IdM クライアントに自動マウントを設定 している。

手順

1. IdM ユーザーが **read-write** ディレクトリーにアクセスできることを確認します。
   a. IdM ユーザーとして IdM クライアントに接続します。

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

   b. IdM ユーザーの Ticket Granting Ticket (TGT) を取得します。

   ```
   $ kinit idm_user
   ```

   c. [オプション] Id M ユーザーのグループメンバーシップを表示します。

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

   d. **/devel/project** ディレクトリーに移動します。

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

   e. ディレクトリーの内容をリスト表示します。

   ```
   $ ls
   rw_file
   ```

f. ディレクトリーのファイルに行を追加し、**write** パーミッションをテストします。

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

g. [オプション] ファイルの更新された内容を表示します。

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

出力は、**idm_user** がファイルに書き込めることを確認します。

第33章 ANSIBLE を使用して IDM を NIS ドメインおよびネットグループと統合する

33.1. NIS とその利点

UNIX 環境では、ネットワーク情報サービス (NIS) は ID と認証を一元管理する一般的な方法です。当初は **Yellow Pages** (YP) という名前が付けられていた NIS は、以下のような認証および ID 情報を一元管理します。

- ユーザーおよびパスワード
- ホスト名および IP アドレス
- POSIX グループ

今日のネットワークインフラストラクチャーでは、NIS は、ホスト認証を提供しておらず、データが暗号化せずにネットワークに送信されるため、セキュリティーが非常に低いと見なされます。この問題を回避するため、NIS はセキュリティーを強化するために他のプロトコルと統合されることが多くあります。

Identity Management(IdM) を使用する場合は、NIS サーバープラグインを使用して、IdM に完全に移行することができないクライアントに接続できます。IdM は、ネットグループおよびその他の NIS データを IdM ドメインに統合します。また、NIS ドメインから IdM にユーザーおよびホストの ID を簡単に移行することもできます。

ネットグループは、NIS グループが想定されるあらゆる場所で使用できます。

関連情報

- IdM の NIS
- IdM の NIS ネットグループ
- NIS から Identity Management への移行

33.2. IDM の NIS

IdM の NIS オブジェクト

NIS オブジェクトは、RFC 2307 に準拠し、Directory Server バックエンドに統合され、保存されます。IdM は、LDAP ディレクトリーに NIS オブジェクトを作成し、クライアントは、たとえば System Security Services Daemon (SSSD) または暗号化された LDAP 接続を使用する **nss_ldap** を通じてそのオブジェクトを取得します。

IdM は、ネットグループ、アカウント、グループ、ホスト、およびその他のデータを管理します。IdM は NIS リスナーを使用してパスワード、グループ、およびネットグループを IdM エントリーにマッピングします。

IdM の NIS プラグイン

NIS サポートの場合、IdM は **slapi-nis** パッケージで提供される以下のプラグインを使用します。

NIS サーバープラグイン

NIS サーバープラグインにより、IdM 統合 LDAP サーバーがクライアントの NIS サーバーとして機能できるようになります。このロールでは、Directory Server は設定に応じて NIS マップを動的に生

成し、更新します。プラグインを使用すると、IdM は NIS プロトコルを使用するクライアントに対して NIS サーバーとして機能します。

スキーマ互換性プラグイン

スキーマ互換性プラグインを使用すると、Directory Server バックエンドは、ディレクトリー情報ツリー (DIT) の一部に保存されたエントリーの代替ビューを提供できるようになります。これには、属性値の追加、ドロップ、名前変更、およびオプションでツリー内の複数のエントリーからの属性値の取得が含まれます。
詳細は、**/usr/share/doc/slapi-nis-version/sch-getting-started.txt** ファイルを参照してください。

33.3. IDM の NIS ネットグループ

NIS エンティティーはネットグループに保存できます。UNIX グループと比較すると、ネットグループは以下のサポートを提供します。

- ネスト化されたグループ (他のグループのメンバーとしてのグループ)。
- ホストのグループ化

ネットグループは、ホスト、ユーザー、およびドメインなどの一連の情報を定義します。このセットは **トリプル** と呼ばれています。以下の 3 つのフィールドを含めることができます。

- 値。
- 「有効な値なし」を指定するダッシュ (**-**)
- 値なし。空のフィールドはワイルドカードを指定します。

> (**host.example.com**,,**nisdomain.example.com**)
> (-,**user**,**nisdomain.example.com**)

クライアントが NIS ネットグループを要求すると、IdM は以下の項目に LDAP エントリーを変換します。

- 従来の NIS マップへと変換し、これを NIS プラグインを使用して NIS プロトコル経由でクライアントに送信します。
- RFC 2307 または RFC 2307bis に準拠する LDAP 形式に変換します。

33.4. ANSIBLE を使用してネットグループが存在することを確認する

Ansible Playbook を使用して、IdM ネットグループが存在することを確認できます。この例では、**TestNetgroup1** グループが存在することを確認する方法を説明します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。

- **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。

手順

1. 次の内容を含む Ansible Playbook ファイル **netgroup-present.yml** を作成します。

```
---
- 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. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory_/netgroup-
present.yml
```

関連情報

- IdM の NIS
- **/usr/share/doc/ansible-freeipa/README-netgroup.md**
- **/usr/share/doc/ansible-freeipa/playbooks/netgroup**

33.5. ANSIBLE を使用してメンバーがネットグループに存在していることを確認する

Ansible Playbook を使用すると、IdM ユーザー、グループ、およびネットグループがネットグループのメンバーであることを確認できます。この例では、**TestNetgroup1** グループに次のメンバーが含まれていることを確認する方法を説明します。

- **user1** および **user2** IdM ユーザー
- **group1** IdM グループ
- **admins** ネットグループ
- IdM クライアントである **idmclient1** ホスト

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。

- Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
- **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
- **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。

- **TestNetgroup1** IdM ネットグループが存在します。
- **user1** と **user2** の IdM ユーザーが存在します。
- **group1** IdM グループが存在します。
- **admins** IdM ネットグループが存在します。

手順

1. 次の内容を含む Ansible Playbook ファイル **IdM-members-present-in-a-netgroup.yml** を作成します。

```
---
- 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. Playbook を実行します。

```
$ 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
```

関連情報

- IdM の NIS
- **/usr/share/doc/ansible-freeipa/README-netgroup.md**
- **/usr/share/doc/ansible-freeipa/playbooks/netgroup**

33.6. ANSIBLE を使用してメンバーがネットグループに存在しないことを確認する

Ansible Playbook を使用して、IdM ユーザーがネットグループのメンバーであることを確認できます。この例では、**TestNetgroup1** グループのメンバーに IdM ユーザー **user1** が含まれていないことを確認する方法を説明します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
  - **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。
- **TestNetgroup1** ネットグループが存在します。

手順

1. 次の内容を含む Ansible Playbook ファイル **IdM-member-absent-from-a-netgroup.yml** を作成します。

```
---
- 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. Playbook を実行します。

```
$ 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
```

関連情報

- IdM の NIS
- **/usr/share/doc/ansible-freeipa/README-netgroup.md**
- **/usr/share/doc/ansible-freeipa/playbooks/netgroup**

33.7. ANSIBLE を使用してネットグループが存在しないことを確認する

Ansible Playbook を使用して、Identity Management (IdM) にネットグループが存在しないことを確認できます。この例では、**TestNetgroup1** グループが IdM ドメインに存在しないことを確認する方法を説明します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
  - **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。

手順

1. 次の内容を含む Ansible Playbook ファイル **netgroup-absent.yml** を作成します。

```
---
- 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. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i
path_to_inventory_directory/inventory.file path_to_playbooks_directory_/netgroup-
absent.yml
```

関連情報

- IdM の NIS
- **/usr/share/doc/ansible-freeipa/README-netgroup.md**
- **/usr/share/doc/ansible-freeipa/playbooks/netgroup**

第34章 ANSIBLE を使用した IDM での HBAC ルールおよび SUDO ルールの設定

Identity Management (IdM) でホストベースのアクセス制御 (HBAC) を使用すると、以下に基づいてホストまたはサービスへのアクセスを制限するポリシーを定義できます。

- ログインを試行しているユーザーおよびこのユーザーのグループ
- ユーザーがアクセスしようとしているホストおよびそのホストが属するホストグループ
- ホストへのアクセスに使用されるサービス

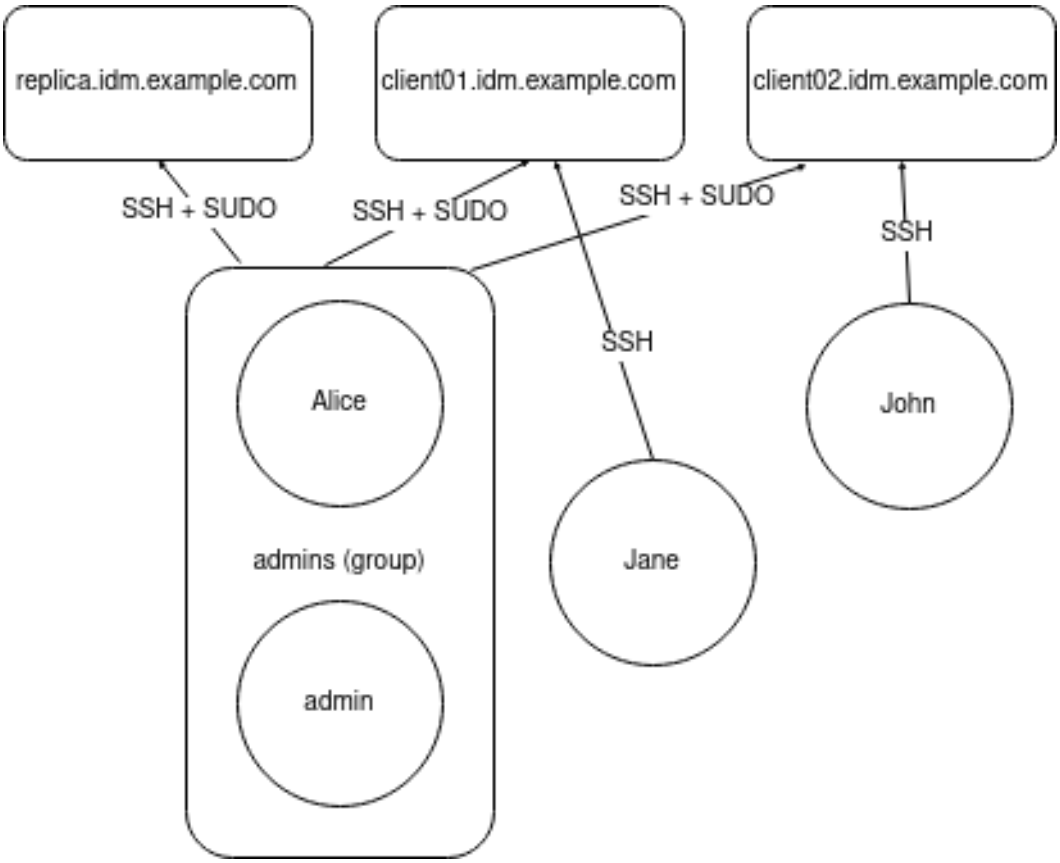
sudo を使用すると、ユーザーは **root** 権限などの別の権限で、別のユーザーとしてプログラムを実行できます。IdM では、sudo ルールを一元管理できます。**sudo** ルールは、ユーザーグループ、ホストグループ、コマンドグループのほか、個々のユーザー、個々のホスト、個々のコマンドに基づいて定義できます。

以下の手順を実行し、IdM ユーザーに対して以下の HBAC ルールおよび **sudo** ルールが設定されていることを確認します。

- **jane** はホスト **client01.idm.example.com** のみにアクセスできます。
- **john** は、ホスト **client02.idm.example.com** のみにアクセスできます。
- デフォルトの **admin** ユーザーと通常の **alice** ユーザーを含む **admins** グループのメンバーは、任意の IdM ホストにアクセスできます。
- **admins** グループのメンバーは、任意の IdM ホストで以下のコマンドを使用して **sudo** を実行できます。
  - **/usr/sbin/reboot**
  - **/usr/bin/less**
  - **/usr/sbin/setenforce**

以下の図は、上述の必要な設定を示しています。

図34.1 IdM HBAC および SUDO ルールの図



前提条件

- コントロールノード:
  - Ansible バージョン 2.14 以降を使用している。
  - **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している。
  - **secret.yml** Ansible vault に **ipaadmin_password** が保存されている。
- IdM に、ユーザー **jane**、**john**、**alice** が存在する。これらのアカウントにはパスワードが設定されます。

手順

1. Ansible Playbook ファイル **add-hbac-and-sudo-rules-to-idm.yml** を以下の内容で作成します。

```
---
- name: Playbook to manage IPA HBAC and SUDO rules
  hosts: ipaserver
  become: false
  gather_facts: false

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

```
module_defaults:
  ipahbacrule:
    ipaadmin_password: "{{ ipaadmin_password }}"
  ipagroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
  ipasudocmd:
    ipaadmin_password: "{{ ipaadmin_password }}"
  ipasudocmdgroup:
    ipaadmin_password: "{{ ipaadmin_password }}"
  ipasudorule:
    ipaadmin_password: "{{ ipaadmin_password }}"

tasks:
- name: HBAC Rule for Jane - can log in to client01
  ipahbacrule: # Creates the rule
    name: Jane_rule
    hbacsvc:
    - sshd
    - login
    host: # Host name
    - client01.idm.example.com
    user:
    - jane

- name: HBAC Rule for John - can log in to client02
  ipahbacrule: # Creates the rule
    name: john_rule
    hbacsvc:
    - sshd
    - login
    host: # Host name
    - client02.idm.example.com
    user:
    - john

- name: Add user member alice to group admins
  ipagroup:
    name: admins
    action: member
    user:
    - alice

- name: HBAC Rule for IdM administrators
  ipahbacrule: # Rule to allow admins full access
    name: admin_access # Rule name
    servicecat: all # All services
    hostcat: all # All hosts
    group: # User group
    - admins

  - name: Add reboot command to SUDO
    ipasudocmd:
      name: /usr/sbin/reboot
      state: present
  - name: Add less command to SUDO
    ipasudocmd:
```

```
      name: /usr/bin/less
      state: present
  - name: Add setenforce command to SUDO
    ipasudocmd:
      name: /usr/sbin/setenforce
      state: present

- name: Create a SUDO command group
  ipasudocmdgroup:
    name: cmd_grp_1
    description: "Group of important commands"
    sudocmd:
    - /usr/sbin/setenforce
    - /usr/bin/less
    - /usr/sbin/reboot
    action: sudocmdgroup
    state: present

- name: Create a SUDO rule with a SUDO command group
  ipasudorule:
    name: sudo_rule_1
    allow_sudocmdgroup:
    - cmd_grp_1
    group: admins
    state: present

- name: Disable allow_all HBAC Rule
  ipahbacrule: # Rule to allow admins full access
    name: allow_all # Rule name
    state: disabled # Disables rule to allow everyone the ability to login
```

2. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -i inventory add-hbac-and-sudo-rules-to-idm.yml
```

検証

1. jane ユーザーとして client01 に接続します。

```
~]$ ssh jane@client01
Password:

Last login: Fri Aug 11 15:32:18 2023 from 192.168.122.1
[jane@client01 ~]$
```

この出力では、jane が client01 にログインしていることが確認できます。

2. jane ユーザーとして client02 への接続を試みます。

```
~]$ ssh jane@client02
Password:
Connection closed by 192.168.122.47 port 22
```

この出力では、jane が client02 にログインできないことが確認できます。

3. alice ユーザーとして client02 に接続します。

```
~]$ ssh alice@client02
Password:

Last login: Fri Aug 10 16:13:43 2023 from 192.168.122.1
```

この出力では、alice が client02 にログインしていることが確認できます。

4. スーパーユーザー権限を呼び出さずに **less** を使用して、**/etc/sssd/sssd.conf** ファイルの内容の表示を試みます。

```
[alice@client02 ~]$ less /etc/sssd/sssd.conf
/etc/sssd/sssd.conf: Permission denied
```

ファイルは、ファイルの所有者(**root**)以外は読み取れないため、試行に失敗します。

5. **root** 権限を呼び出し、**less** を使用して **/etc/sssd/sssd.conf** ファイルの内容を表示します。

```
[alice@client02 ~]$ sudo less /etc/sssd/sssd.conf
[sudo] password for alice:

[domain/idm.example.com]

id_provider = ipa
ipa_server_mode = True
[...]
```

この出力では、alice が **/etc/sssd/sssd.conf** ファイルで **less** コマンドを実行できることが確認できます。

関連情報

- IdM のホストベースのアクセス制御ルール
- IdM クライアントの sudo アクセス

第35章 ANSIBLE を使用して IDM ユーザーの認証を外部アイデンティティープロバイダーに委譲する

idp ansible-freeipa モジュールを使用して、OAuth 2 デバイス認証フローをサポートする外部アイデンティティープロバイダー (IdP) にユーザーを関連付けることができます。IdP 参照と関連付けられた IdP ユーザーの ID が存在する場合は、それらを使用して、**useransible-freeipa** モジュールにより IdM ユーザーの IdP 認証を有効にすることができます。

その後、これらのユーザーが RHEL 9.1 以降で利用可能な SSSD バージョンで認証すると、外部 IdP で認証と認可が実行された後、Kerberos チケットを使用した RHEL Identity Management (IdM) シングルサインオン機能がユーザーに提供されます。

35.1. IDM を外部 IDP に接続する利点

管理者は、クラウドサービスプロバイダーなどの外部 ID ソースに保存されているユーザーが、Identity Management (IdM) 環境に追加された RHEL システムにアクセスできるようにすることができます。そのために、これらのユーザーの Kerberos チケットを発行する認証および認可プロセスをその外部エンティティーに委任できます。

この機能を使用して IdM の機能を拡張し、外部 ID プロバイダー (IdP) に保存されているユーザーが IdM によって管理される Linux システムにアクセスできるようにすることができます。

35.2. IDM が外部 IDP を介してログインを組み込む方法

SSSD 2.7.0 には、**idp** Kerberos 事前認証方法を実装する **sssd-idp** パッケージが含まれています。この認証方法は、OAuth 2.0 Device Authorization Grant フローに従って、認可の判断を外部 IdP に委任します。

1. IdM クライアントユーザーは、コマンドラインで **kinit** ユーティリティーを使用して Kerberos TGT の取得を試行するなどして、OAuth 2.0 デバイス認可付与フローを開始します。
2. 特別なコードと Web サイトのリンクが認可サーバーから IdM KDC バックエンドに送信されます。
3. IdM クライアントは、リンクとコードをユーザーに表示します。この例では、IdM クライアントはコマンドラインにリンクとコードを出力します。
4. ユーザーは、別のホストや携帯電話などのブラウザーで Web サイトのリンクを開きます。
   a. ユーザーは特別なコードを入力します。
   b. 必要に応じて、ユーザーは OAuth 2.0 ベースの IdP にログインします。
   c. ユーザーは、クライアントによる情報へのアクセスを許可するよう求められます。
5. ユーザーは、元のデバイスのプロンプトでアクセスを確認します。この例では、ユーザーはコマンドラインで **Enter** キーを押します。
6. IdM KDC バックエンドは、ユーザー情報にアクセスするために OAuth 2.0 認可サーバーをポーリングします。

サポート対象:

- Pluggable Authentication Module (PAM) ライブラリーの呼び出しを可能にする **keyboard-interactive** 認証方法を有効にして、SSH 経由でリモートからログインする場合。

- **logind** サービスを介してコンソールでローカルにログインする場合。
- **kinit** ユーティリティーを使用して Kerberos TGT (Ticket-granting ticket) を取得する場合。

現在のサポート対象外:

- IdM WebUI に直接ログインする場合。IdM WebUI にログインするには、最初に Kerberos チケットを取得する必要があります。
- Cockpit WebUI に直接ログインする場合。Cockpit WebUI にログインするには、最初に Kerberos チケットを取得する必要があります。

関連情報

- Authentication against external Identity Providers
- RFC 8628: OAuth 2.0 Device Authorization Grant

35.3. ANSIBLE を使用して外部アイデンティティープロバイダーへの参照を作成する

外部アイデンティティープロバイダー (IdP) を Identity Management (IdM) 環境に接続するには、IdM で IdP 参照を作成します。この手順では、**idp ansible-freeipa** モジュールを使用して **github** 外部 IdP への参照を設定します。

前提条件

- IdM を OAuth アプリケーションとして外部 IdP に登録し、IdM ユーザーが IdM への認証に使用するデバイス上でクライアント ID とクライアントシークレットを生成した。この例では、以下を前提としています。
  - **my_github_account_name** が github ユーザーであり、そのアカウントを IdM ユーザーが IdM への認証に使用する。
  - **client ID** が **2efe1acffe9e8ab869f4** である。
  - **client secret** が **656a5228abc5f9545c85fa626aecbf69312d398c** である。
- IdM サーバーで RHEL 9.1 以降を使用している。
- IdM サーバーで SSSD 2.7.0 以降を使用している。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - RHEL 9.4 以降を使用している。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

手順

1. Ansible コントロールノードで、**configure-external-idp-reference.yml** Playbook を作成します。

```
---
- name: Configure external IdP
  hosts: ipaserver
  become: false
  gather_facts: false

  tasks:
  - name: Ensure a reference to github external provider is available
    ipaidp:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: github_idp
      provider: github
      client_ID: 2efe1acffe9e8ab869f4
      secret: 656a5228abc5f9545c85fa626aecbf69312d398c
      idp_user_id: my_github_account_name
```

2. ファイルを保存します。

3. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory configure-external-idp-reference.yml
```

検証

- IdM クライアントで、**ipa idp-show** コマンドの出力に、作成した IdP 参照が表示されることを確認します。

```
[idmuser@idmclient ~]$ ipa idp-show github_idp
```

次のステップ

- Ansible を使用して IdM ユーザーが外部 IdP 経由で認証できるようにする

関連情報

- idp **ansible-freeipa** アップストリームドキュメント

35.4. ANSIBLE を使用して IDM ユーザーが外部 IDP 経由で認証できるようにする

user ansible-freeipa モジュールを使用すると、Identity Management (IdM) ユーザーが外部アイデンティティープロバイダー (IdP) 経由で認証できるようになります。これを行うには、以前に作成した外部 IdP 参照を IdM ユーザーアカウントに関連付けます。この手順では、Ansible を使用して、**github_idp** という名前の外部 IdP 参照を **idm-user-with-external-idp** という名前の IdM ユーザーに関連付けます。この手順の結果、ユーザーが **my_github_account_name** github アイデンティティーを使用して、**idm-user-with-external-idp** として IdM に認証できるようになります。

前提条件

- IdM クライアントと IdM サーバーで RHEL 9.1 以降を使用している。
- IdM クライアントと IdM サーバーで SSSD 2.7.0 以降を使用している。
- IdM で外部 IdP への参照を作成した。Ansible を使用して外部アイデンティティープロバイダーへの参照を作成する を参照してください。
- 次の要件を満たすように Ansible コントロールノードを設定した。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - RHEL 9.4 以降を使用している。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible vault に **ipaadmin_password** が保存されていることを前提としています。

手順

1. Ansible コントロールノードで、**enable-user-to-authenticate-via-external-idp.yml** Playbook を作成します。

```
---
- name: Ensure an IdM user uses an external IdP to authenticate to IdM
  hosts: ipaserver
  become: false
  gather_facts: false

  tasks:
  - name: Retrieve Github user ID
    ansible.builtin.uri:
      url: “https://api.github.com/users/my_github_account_name”
      method: GET
      headers:
        Accept: “application/vnd.github.v3+json”
    register: user_data

  - name: Ensure IdM user exists with an external IdP authentication
    ipauser:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: idm-user-with-external-idp
      first: Example
      last: User
      userauthtype: idp
      idp: github_idp
      idp_user_id: my_github_account_name
```

2. ファイルを保存します。

3. Ansible Playbook を実行します。Playbook ファイル、**secret.yml** ファイルを保護するパスワードを格納するファイル、およびインベントリーファイルを指定します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory enable-user-to-authenticate-via-external-idp.yml
```

検証

- IdM クライアントにログインし、**idm-user-with-external-idp** ユーザーの **ipa user-show** コマンドの出力に IdP への参照が表示されることを確認します。

```
$ ipa user-show idm-user-with-external-idp
User login: idm-user-with-external-idp
First name: Example
Last name: User
Home directory: /home/idm-user-with-external-idp
Login shell: /bin/sh
Principal name: idm-user-with-external-idp@idm.example.com
Principal alias: idm-user-with-external-idp@idm.example.com
Email address: idm-user-with-external-idp@idm.example.com
ID: 35000003
GID: 35000003
User authentication types: idp
External IdP configuration: github
External IdP user identifier: idm-user-with-external-idp@idm.example.com
Account disabled: False
Password: False
Member of groups: ipausers
Kerberos keys available: False
```

関連情報

- idp **ansible-freeipa** アップストリームドキュメント

35.5. 外部 IDP ユーザーとして IDM TICKET-GRANTING TICKET を取得する

Identity Management (IdM) ユーザーの認証を外部アイデンティティープロバイダー (IdP) に委譲している場合、IdM ユーザーは外部 IdP に対して認証することで Kerberos Ticket-Granting Ticket (TGT) を要求できます。

この手順では、以下を実行します。

1. 匿名の Kerberos チケットを取得してローカルに保存します。
2. **-T** オプションを指定した **kinit** を使用して **idm-user-with-external-idp** ユーザーの TGT を要求し、Flexible Authentication via Secure Tunneling (FAST) チャネルを有効にして、Kerberos クライアントと Kerberos Distribution Center (KDC) 間のセキュアな接続を提供します。

前提条件

- IdM クライアントと IdM サーバーが RHEL 9.1 以降を使用している。
- IdM クライアントと IdM サーバーが SSSD 2.7.0 以降を使用している。

- IdM で外部 IdP への参照を作成した。Ansible を使用して外部アイデンティティープロバイダーへの参照を作成する を参照してください。
- 外部 IdP 参照をユーザーアカウントに関連付けている。Ansible を使用して IdM ユーザーが外部 IdP 経由で認証できるようにする を参照してください。
- 最初にログインするユーザーに、ローカルファイルシステム内のディレクトリーに対する書き込み権限がある。

手順

1. 匿名 PKINIT を使用して Kerberos チケットを取得し、それを **./fast.ccache** という名前のファイルに保存します。

```
$ kinit -n -c ./fast.ccache
```

2. [オプション] 取得したチケットを表示します。

```
$ *klist -c fast.ccache *
Ticket cache: FILE:fast.ccache
Default principal: WELLKNOWN/ANONYMOUS@WELLKNOWN:ANONYMOUS

Valid starting       Expires              Service principal
03/03/2024 13:36:37  03/04/2024 13:14:28
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
```

3. **-T** オプションを使用して FAST 通信チャネルを有効にし、IdM ユーザーとして認証を開始します。

```
[root@client ~]# kinit -T ./fast.ccache idm-user-with-external-idp
Authenticate at https://oauth2.idp.com:8443/auth/realms/master/device?user_code=YHMQ-
XKTL and press ENTER.:
```

4. ブラウザーで、コマンド出力に提供される Web サイトでユーザーとして認証します。
5. コマンドラインで **Enter** キーを押して、認証プロセスを終了します。

検証

- Kerberos チケット情報を表示し、**config: pa_type** の行が外部 IdP による事前認証の **152** を示していることを確認します。

```
[root@client ~]# klist -C
Ticket cache: KCM:0:58420
Default principal: idm-user-with-external-idp@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
05/09/22 07:48:23  05/10/22 07:03:07  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: fast_avail(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = yes
08/17/2022 20:22:45  08/18/2022 20:22:43
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: pa_type(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = 152
```

pa_type = 152 は、外部 IdP 認証を示します。

35.6. 外部 IDP ユーザーとして SSH 経由で IDM クライアントにログインする

外部 ID プロバイダー (IdP) ユーザーとして SSH 経由で IdM クライアントにログインするには、コマンドラインでログインプロセスを開始します。プロンプトが表示されたら、IdP に関連付けられた Web サイトで認証プロセスを実行し、Identity Management (IdM) クライアントでプロセスを終了します。

前提条件

- IdM クライアントと IdM サーバーで RHEL 9.1 以降を使用している。
- IdM クライアントと IdM サーバーで SSSD 2.7.0 以降を使用している。
- IdM で外部 IdP への参照を作成した。Ansible を使用して外部アイデンティティープロバイダーへの参照を作成する を参照してください。
- 外部 IdP 参照をユーザーアカウントに関連付けている。Ansible を使用して IdM ユーザーが外部 IdP 経由で認証できるようにする を参照してください。

手順

1. SSH 経由で IdM クライアントへのログインを試みます。

```
[user@client ~]$ ssh idm-user-with-external-idp@client.idm.example.com
(idm-user-with-external-idp@client.idm.example.com) Authenticate at
https://oauth2.idp.com:8443/auth/realms/main/device?user_code=XYFL-ROYR and press
ENTER.
```

2. ブラウザーで、コマンド出力に提供される Web サイトでユーザーとして認証します。
3. コマンドラインで **Enter** キーを押して、認証プロセスを終了します。

検証

- Kerberos チケット情報を表示し、**config: pa_type** の行が外部 IdP による事前認証の **152** を示していることを確認します。

```
[idm-user-with-external-idp@client ~]$ klist -C
Ticket cache: KCM:0:58420
Default principal: idm-user-with-external-idp@IDM.EXAMPLE.COM

Valid starting     Expires            Service principal
05/09/22 07:48:23  05/10/22 07:03:07  krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: fast_avail(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = yes
08/17/2022 20:22:45  08/18/2022 20:22:43
krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM
config: pa_type(krbtgt/IDM.EXAMPLE.COM@IDM.EXAMPLE.COM) = 152
```

35.7. IPAIDP ANSIBLE モジュールの PROVIDER オプション

次の ID プロバイダー (IdP) は、OAuth 2.0 デバイス認可グラントフローをサポートしています。

- Azure AD を含む Microsoft Identity Platform

- Google
- GitHub
- Red Hat Single Sign-On (SSO) を含む Keycloak
- Okta

idp ansible-freeipa モジュールを使用してこれらの外部 IdP の 1 つへの参照を作成する場合、**ipaidp ansible-freeipa** Playbook タスクの **provider** オプションで IdP のタイプを指定できます。指定すると、以下で説明する追加オプションをさらに指定できます。

provider: microsoft

Microsoft Azure IdP を使用すると、Azure テナント ID に基づくパラメーター設定を行うことができます。Azure テナント ID は、**organization** オプションで指定できます。live.com IdP のサポートが必要な場合は、オプション **organization common** を指定します。
provider: microsoft を選択すると、次のオプションを使用するように拡張されます。表内の文字列 **${ipaidporg}** は、**organization** オプションの値に置き換えます。

オプション	値
auth_uri: URI	**https://login.microsoftonline.com/${ipaidporg}/oauth2/v2.0/authorize**
dev_auth_uri: URI	**https://login.microsoftonline.com/${ipaidporg}/oauth2/v2.0/devicecode**
token_uri: URI	**https://login.microsoftonline.com/${ipaidporg}/oauth2/v2.0/token**
userinfo_uri: URI	**https://graph.microsoft.com/oidc/userinfo**
keys_uri: URI	**https://login.microsoftonline.com/common/discovery/v2.0/keys**
scope: STR	**openid email**
idp_user_id: STR	**email**

provider: google

provider: google を選択すると、次のオプションを使用するように拡張されます。

オプション	値
auth_uri: URI	**https://accounts.google.com/o/oauth2/auth**
dev_auth_uri: URI	**https://oauth2.googleapis.com/device/code**
token_uri: URI	**https://oauth2.googleapis.com/token**

オプション	値
userinfo_uri: URI	**https://openidconnect.googleapis.com/v1/userinfo**
keys_uri: URI	**https://www.googleapis.com/oauth2/v3/certs**
scope: STR	**openid email**
idp_user_id: STR	**email**

provider: github

provider: github を選択すると、次のオプションを使用するように拡張されます。

オプション	値
auth_uri: URI	**https://github.com/login/oauth/authorize**
dev_auth_uri: URI	**https://github.com/login/device/code**
token_uri: URI	**https://github.com/login/oauth/access_token**
userinfo_uri: URI	**https://openidconnect.googleapis.com/v1/userinfo**
keys_uri: URI	**https://api.github.com/user**
scope: STR	**user**
idp_user_id: STR	**login**

provider: keycloak

Keycloak を使用すると、複数のレルムまたは組織を定義できます。多くの場合、Keycloak はカスタムデプロイメントの一部であるため、ベース URL とレルム ID の両方が必要です。これらは、**ipaidp** Playbook タスクの **base_url** および **organization** オプションで指定できます。

```
---
- name: Playbook to manage IPA idp
  hosts: ipaserver
  become: false

  tasks:
  - name: Ensure keycloak idp my-keycloak-idp is present using provider
    ipaidp:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: my-keycloak-idp
      provider: keycloak
      organization: main
      base_url: keycloak.domain.com:8443/auth
      client_id: my-keycloak-client-id
```

provider: keycloak を選択すると、次のオプションを使用するように拡張されます。**base_url** オプションで指定した値により表内の文字列 **${ipaidpbaseurl}** が置き換えられ、**organization** オプションで指定した値により文字列 ${ipaidporg} が置き換えられます。

オプション	値
auth_uri: URI	**https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/auth**
dev_auth_uri: URI	**https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/auth/device**
token_uri: URI	**https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/token**
userinfo_uri: URI	**https://${ipaidpbaseurl}/realms/${ipaidporg}/protocol/openid-connect/userinfo**
scope: STR	**openid email**
idp_user_id: STR	**email**

provider: okta

Okta に新しい組織を登録すると、新しいベース URL が関連付けられます。このベース URL は、**ipaidp** Playbook タスクの **base_url** オプションで指定できます。

```
---
- name: Playbook to manage IPA idp
  hosts: ipaserver
  become: false

  tasks:
  - name: Ensure okta idp my-okta-idp is present using provider
    ipaidp:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: my-okta-idp
      provider: okta
      base_url: dev-12345.okta.com
      client_id: my-okta-client-id
```

provider: okta を選択すると、次のオプションを使用するように拡張されます。**base_url** オプションに指定した値により、テーブル内の文字列 **${ipaidpbaseurl}** が置き換えられます。

オプション	値
auth_uri: URI	**https://${ipaidpbaseurl}/oauth2/v1/authorize**
dev_auth_uri: URI	**https://${ipaidpbaseurl}/oauth2/v1/device/authorize**

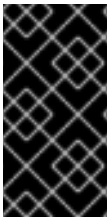
オプション	値
token_uri: URI	**https://${ipaidpbaseurl}/oauth2/v1/token**
userinfo_uri: URI	**https://${ipaidpbaseurl}/oauth2/v1/userinfo**
scope: STR	**openid email**
idp_user_id: STR	**email**

関連情報

- 事前入力された IdP テンプレート

第36章 RHEL システムロールを使用した RHEL システムと AD の直接統合

ad_integration システムロールを使用すると、Red Hat Ansible Automation Platform を使用して RHEL システムと Active Directory (AD) の直接統合を自動化できます。



重要

ad_integration システムロールは **ansible-freeipa** パッケージには含まれていません。このシステムロールは **rhel-system-roles** パッケージの一部です。**rhel-system-roles** は、**Red Hat Enterprise Linux Server** サブスクリプションが割り当てられているシステムにインストールできます。

36.1. AD_INTEGRATION RHEL システムロール

ad_integration システムロールを使用すると、RHEL システムを Active Directory (AD) に直接接続できます。

ロールは次のコンポーネントを使用します。

- 中央の ID および認証ソースと対話するための SSSD
- 使用可能な AD ドメインを検出し、基盤となる RHEL システムサービス (この場合は SSSD) を設定して、選択した AD ドメインに接続する **realmd**

注記

ad_integration ロールは、Identity Management (IdM) 環境を使用せずに直接 AD 統合を使用するデプロイメント用です。IdM 環境の場合は、**ansible-freeipa** ロールを使用します。

関連情報

- **/usr/share/ansible/roles/rhel-system-roles.ad_integration/README.md** ファイル
- **/usr/share/doc/rhel-system-roles/ad_integration/** ディレクトリー
- SSSD を使用して RHEL システムを AD に直接接続