Red Hat Enterprise Linux 8

Identity Management での DNS の操作

IdM 統合 DNS サービスの管理

Last Updated: 2024-06-27

Red Hat Enterprise Linux 8 Identity Management での DNS の操作

IdM 統合 DNS サービスの管理

法律上の通知

概要

DNS は、Red Hat Identity Management (IdM) ドメインの重要なコンポーネントです。たとえば、クライアントは DNS を使用してサービスを見つけ、同じサイト内のサーバーを識別します。コマンドライン、IdM Web UI、および Ansible Playbook を使用して、IdM に統合された DNS サーバーでレコード、ゾーン、ロケーション、および転送を管理できます。

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 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 root CA として使用する場合 を参照してください。
  - IdM サーバーのインストール: 統合 DNS と外部 CA を root CA として使用する場合 を参照してください。
  - IdM サーバーのインストール: 統合 DNS あり、CA なしのサーバー

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

1.1. 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。

1.2. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。

- 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。

1.3. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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**.

      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

1.4. 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1.1 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

1.5. 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)

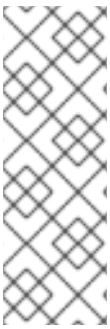
IdM BIND サービスは、DNS クエリーを設定済みのフォワーダーに転送します。サーバーエラーやタイムアウトが原因でクエリーに失敗すると、BIND はインターネット上のサーバーを使用して再帰解決にフォールバックします。**forward first** ポリシーはデフォルトのポリシーで、DNS トラフィックの最適化に適しています。

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1.1 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。

1.6. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

1.7. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

1.8. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。

- 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2章 IDM での DNS ゾーンの管理

Identity Management (IdM) 管理者は、IdM DNS ゾーンの動作を管理できます。本章では、以下のトピックおよび手順を説明します。

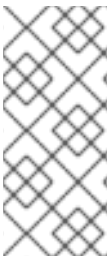
- IdM でサポートされる DNS ゾーンタイプ
  - IdM Web UI を使用してプライマリー IdM DNS ゾーンを追加する方法
  - IdM CLI を使用してプライマリー IdM DNS ゾーンを追加する方法
  - IdM Web UI を使用してプライマリー IdM DNS ゾーンを削除する方法
  - IdM CLI を使用してプライマリー IdM DNS ゾーンを削除する方法
- IdM で設定できる DNS 属性
  - IdM Web UI で DNS 属性を設定する方法
  - IdM CLI で DNS 属性を設定する方法
- IdM でのゾーン転送の仕組み
  - IdM Web UI でゾーン転送を許可する方法
  - IdM CLI でゾーン転送を許可する方法

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 root CA として使用する場合 を参照してください。
  - IdM サーバーのインストール: 統合 DNS と外部 CA を root CA として使用する場合 を参照してください。
  - IdM サーバーのインストール: 統合 DNS あり、CA なしのサーバー

2.1. 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2.1 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。

2.2. IDM WEB UI でのプライマリー DNS ゾーンの追加

Identity Management (IdM) Web UI を使用してプライマリー DNS ゾーンを追加するには、次の手順に従います。

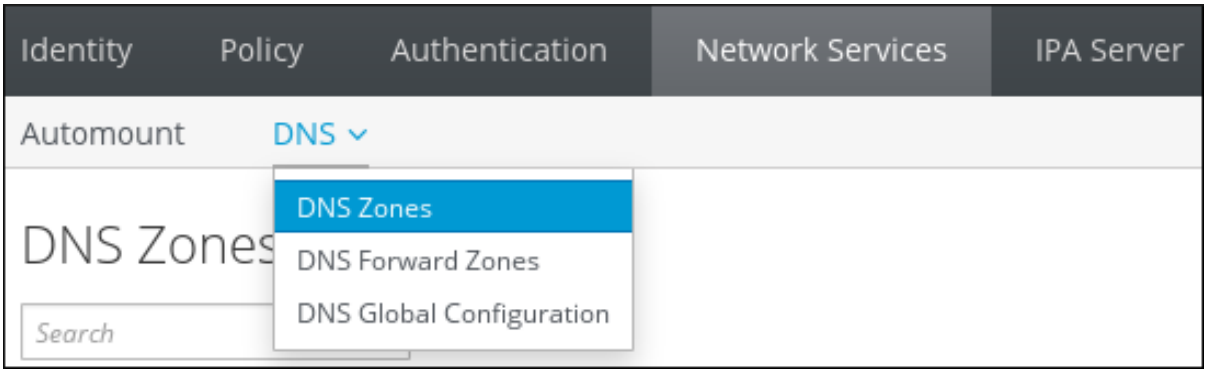
前提条件

- IdM 管理者としてログインしている。

手順

1. IdM Web UI で、**Network Services → DNS → DNS Zones** の順にクリックします。

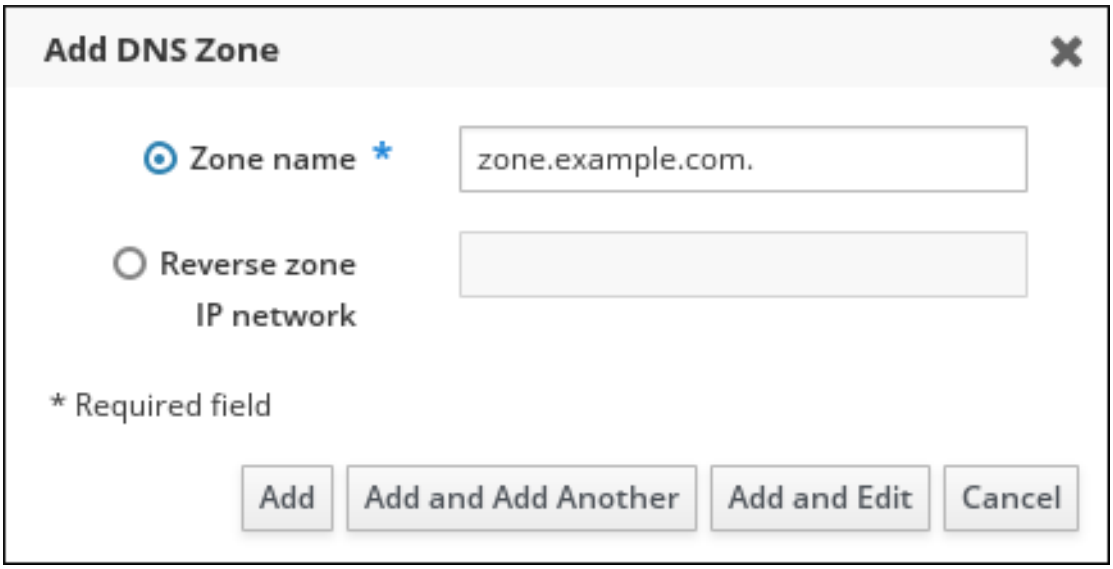
図2.1 IdM DNS プライマリーゾーンの管理



2. すべてのゾーンリストの上部にある **追加** をクリックします。

3. ゾーン名を指定します。

図2.2 新しい IdM プライマリーゾーンの入力



4. **Add** をクリックします。

2.3. IDM CLI でのプライマリー DNS ゾーンの追加

Identity Management (IdM) コマンドラインインターフェイス (CLI) を使用してプライマリー DNS ゾーンを追加するには、次の手順に従います。

前提条件

- IdM 管理者としてログインしている。

手順

- **ipa dnszone-add** コマンドは、新しいゾーンを DNS ドメインに追加します。新しいゾーンを追加するには、新しいサブドメイン名を指定する必要があります。サブドメイン名を直接指定するには、以下のコマンドを実行します。

```
$ ipa dnszone-add newzone.idm.example.com
```

ipa dnszone-add に名前を指定しない場合には、スクリプトにより自動的に名前を求めるプロンプトが表示されます。

関連情報

- **ipa dnszone-add --help** を参照してください。

2.4. IDM WEB UI でのプライマリー DNS ゾーンの削除

IdM Web UI を使用して Identity Management (IdM) からプライマリー DNS ゾーンを削除するには、この手順に従います。

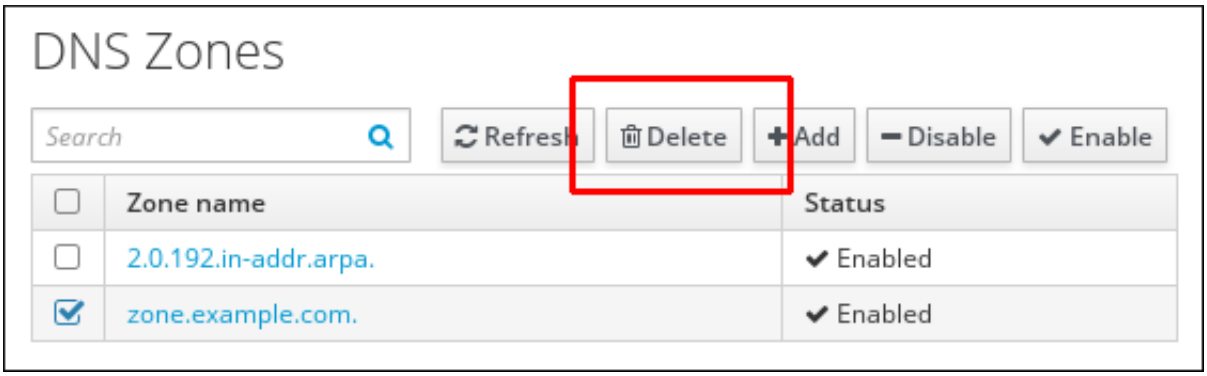
前提条件

- IdM 管理者としてログインしている。

手順

1. IdM Web UI で、**Network Services → DNS → DNS Zones** の順にクリックします。
2. ゾーン名の横にあるチェックボックスを選択し、**削除** をクリックします。

図2.3 プライマリー DNS ゾーンの削除



3. **DNS ゾーンの削除** ダイアログウィンドウで、選択したゾーンの削除を確定します。

2.5. IDM CLI でのプライマリー DNS ゾーンの削除

IdM コマンドラインインターフェイス (CLI) を使用して Identity Management (IdM) からプライマリー DNS ゾーンを削除するには、次の手順に従います。

前提条件

- IdM 管理者としてログインしている。

手順

- プライマリー DNS ゾーンを削除するには、**ipa dnszone-del** コマンドの後に、削除するゾーンの名前を入力します。以下に例を示します。

```
$ ipa dnszone-del idm.example.com
```

2.6. DNS 設定の優先順位

多くの DNS 設定オプションは、次のレベルで設定できます。優先度は、レベルごとに異なります。

ゾーン固有の設定

IdM に定義されている特定のゾーンに固有の設定は、優先度が最も高いレベルです。**ipa dnszone-*** コマンドおよび **ipa dnsforwardzone-*** コマンドを使用して、ゾーン固有の設定を管理できます。

サーバーごとの設定

IdM サーバーのインストール時に、サーバーごとのフォワーダーを定義するように求められます。サーバーごとのフォワーダーは、**ipa dnsserver-*** コマンドを使用して管理できます。レプリカのインストール時にサーバーごとのフォワーダーを設定しない場合は、**--no-forwarder** オプションを使用できます。

グローバル DNS 設定

ゾーン固有の設定が定義されていない場合は、IdM は LDAP に保存されているグローバル DNS 設定を使用します。グローバル DNS 設定は、**ipa dnsconfig-*** コマンドを使用して管理できます。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。

/etc/named.confの設定

IdM DNS サーバーごとに **/etc/named.conf** ファイルで定義されている設定の優先度は、最も低くなります。これは各サーバーに固有のものであり、手動で編集する必要があります。
/etc/named.conf ファイルを使用するのは通常、ローカル DNS キャッシュへの DNS 転送を指定する場合のみです。他のオプションは、上記のゾーン固有の設定と、グローバル DNS 設定のコマンドを使用して管理します

DNS オプションは、同時に複数のレベルで設定できます。このような場合に、最も優先度が高い設定は、レベルが低い設定よりも優先されます。

関連情報

- Per Server Config in LDAP の **Priority order of configuration** セクション

2.7. 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2.1 IdM DNS ゾーン属性

属性	コマンドラインオプション	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	コマンドラインオプション	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。

2.8. IDM WEB UI でのプライマリー DNS ゾーン設定の編集

IdM Web UI を使用してプライマリー Identity Management (IdM) DNS の設定属性を編集するには、この手順に従います。

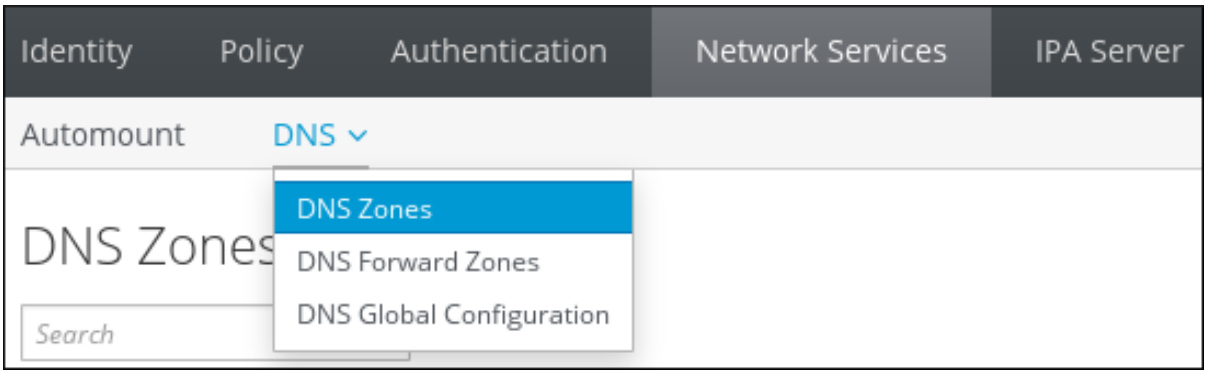
前提条件

- IdM 管理者としてログインしている。

手順

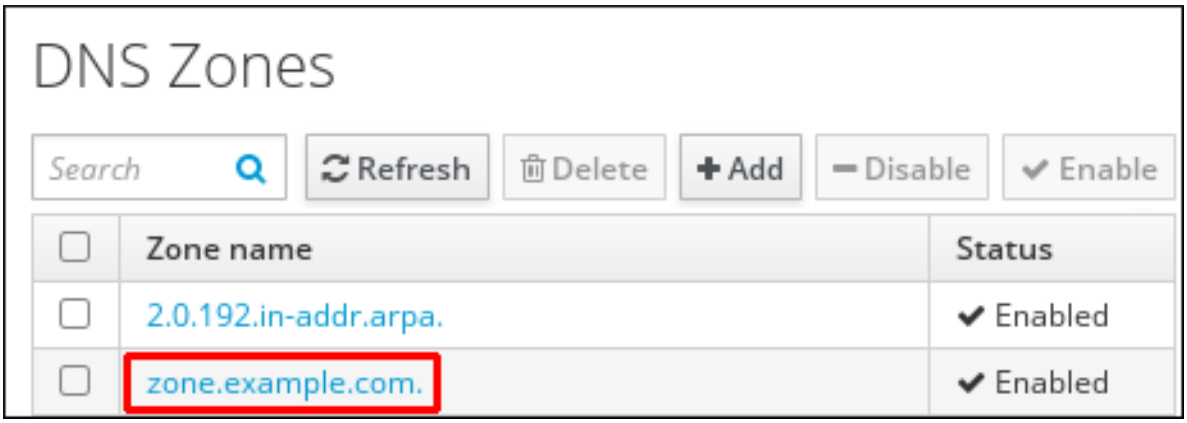
1. IdM Web UI で、**Network Services → DNS → DNS Zones** の順にクリックします。

図2.4 DNS プライマリーゾーンの管理



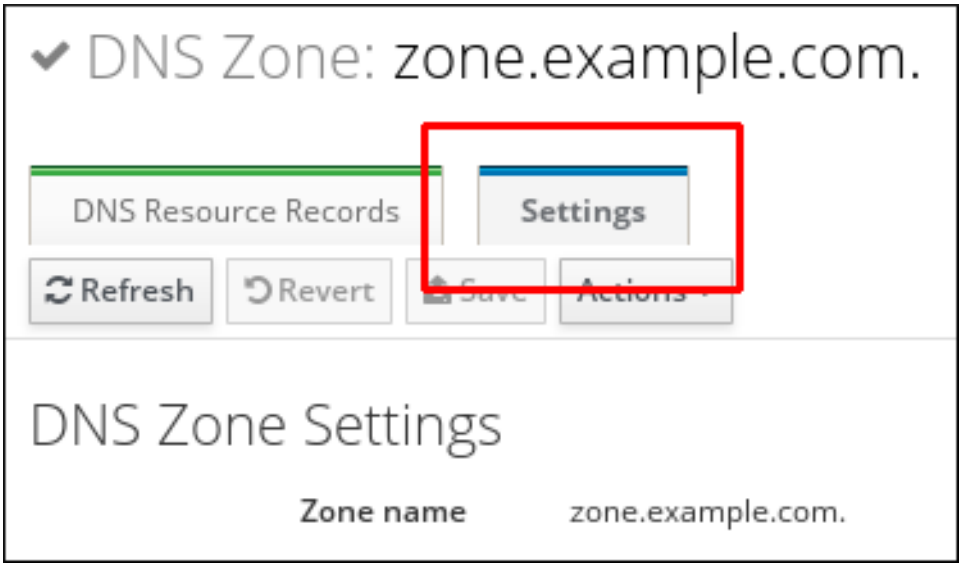
2. **DNS ゾーン** セクションで、全ゾーンのリストにあるゾーン名をクリックし、DNS ゾーンページを開きます。

図2.5 プライマリーゾーンの編集



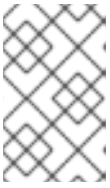
3. **Settings** を クリックします。

図2.6 プライマリーゾーンの編集ページの設定タブ



4. 必要に応じてゾーン設定を変更します。
利用可能な設定の詳細は、IdM DNS ゾーン属性 を参照してください。

5. **Save** をクリックして、新しい設定を確定します。



注記

ゾーンのデフォルトの Time To Live (TTL) を変更する場愛には、全 IdM DNS サーバーで**named-pkcs11** サービスを再起動して、変更を適用します。他の全設定は、すぐに自動的に有効になります。

2.9. IDM CLI でのプライマリー DNS ゾーンの設定の編集

Identity Management (IdM) コマンドラインインターフェイス (CLI) を使用してプライマリー DNS ゾーンの設定を編集するには、次の手順に従います。

前提条件

- IdM 管理者としてログインしている。

手順

- 既存のプライマリー DNS ゾーンを変更するには、**ipa dnszone-mod** コマンドを使用します。たとえば、失敗した更新操作を再試行するまでに待機する時間を 1800 秒に設定します。

```
$ ipa dnszone-mod --retry 1800
```

利用可能な設定と、対応する CLI オプションの詳細は、IdM DNS ゾーン属性 を参照してください。

特定の設定で、変更する DNS ゾーンエントリーに値が指定されていない場合は、**ipa dnszone-mod** コマンドで値を追加します。設定に値がない場合は、このコマンドを実行すると、指定の値に上書きされます。

注記

ゾーンのデフォルトの Time To Live (TTL) を変更する場愛には、全 IdM DNS サーバーで **named-pkcs11** サービスを再起動して、変更を適用します。他の全設定は、すぐに自動的に有効になります。

関連情報

- **ipa dnszone-mod --help** を参照してください。

2.10. IDM でのゾーン転送

DNS が統合された Identity Management (IdM) デプロイメントでは、**ゾーン転送** を使用して、すべてのリソースレコードを 1 つのネームサーバーから別のネームサーバーにコピーできます。ネームサーバーは、ゾーンの権威データを保持します。DNS ゾーン **zone A** に権威のある DNS サーバーでゾーンに変更を加えると、**zone A** 以外にある IdM DNS ドメインの他のネームサーバーに変更を配信する必要があります。

重要

IdM 統合 DNS には、複数のサーバーで同時に記述できます。IdM ゾーンの Start of Authority (SOA) シリアル番号は、個別の IdM DNS サーバーと同期されません。このような理由から、転送予定ゾーン外にある DNS サーバーは、転送予定ゾーン内の特定の DNS サーバー 1 台だけを使用するように設定します。こうすることで、同期されていない SOA シリアル番号が原因でゾーン転送が失敗しないようにします。

IdM は、RFC 5936 (AXFR) および RFC 1995 (IXFR) 標準に準拠するゾーン転送をサポートします。

関連情報

- IdM Web UI でのゾーン転送の有効化 を参照してください。
- IdM CLI でのゾーン転送の有効化 を参照してください。

2.11. IDM WEB UI でのゾーン転送の有効化

IdM Web UI を使用して Identity Management (IdM) でゾーン転送を有効にするには、次の手順に従います。

前提条件

- IdM 管理者としてログインしている。

手順

1. IdM Web UI で、**Network Services → DNS → DNS Zones** の順にクリックします。
2. **Settings** を クリックします。
3. **Allow transfer** で、ゾーンレコードを転送するネームサーバーを指定します。

   図2.7 ゾーン転送の有効化

4. DNS ゾーンページの上部にある **Save** をクリックして、新しい設定を確定します。

2.12. IDM CLI でのゾーン転送の有効化

IdM コマンドラインインターフェイス (CLI) を使用して Identity Management (IdM) でゾーン転送を有効にするには、次の手順に従います。

前提条件

- IdM 管理者としてログインしている。
- セカンダリー DNS サーバーへの root アクセス権限がある。

手順

- **BIND** サービスでゾーン転送を有効にするには、**ipa dnszone-mod** コマンドを入力し、ゾーンレコードの転送先となる転送予定ゾーンに含まれないサーバー名のリストを **--allow-transfer** オプションを使用して指定します。以下に例を示します。

```
$ ipa dnszone-mod --allow-transfer=192.0.2.1;198.51.100.1;203.0.113.1
idm.example.com
```

検証手順

1. ゾーン転送が有効な DNS サーバーの 1 つに SSH 接続します。
   -

```
$ ssh 192.0.2.1
```

2. **dig** ユーティリティーなどのツールを使用して、IdM DNS ゾーンを転送します。

```
# dig @ipa-server zone_name AXFR
```

コマンドでエラーが返されない場合は、**zone_name** のゾーン転送が正常に有効化されています。

2.13. 関連情報

- Using Ansible playbooks to manage IdM DNS zones を参照してください。

第3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。

3.1. 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3.1 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。

3.2. 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3.1 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。

3.3. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

3.4. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

3.5. 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ている。

- 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。

- 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4章 IDM での DNS の場所の管理

IdM Web UI および IdM コマンドラインインターフェイス (CLI) を使用して Identity Management (IdM) DNS の場所を管理する方法について詳しくは、次のトピックと手順を参照してください。

- 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)
- IdM Web UI を使用した DNS の場所の作成
- IdM CLI を使用した DNS の場所の作成
- IdM Web UI を使用した DNS の場所への IdM サーバーの割り当て
- IdM Web UI を使用した DNS の場所への IdM サーバーの割り当て
- IdM クライアントが同じ場所にある IdM サーバーを使用するように設定する手順

4.1. 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4.1 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** (優先順位):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4.2 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。

4.2. 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。

4.3. 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。

4.4. IDM WEB UI を使用した DNS の場所の作成

DNS の場所を使用すると、Identity Management (IdM) クライアントとサーバー間の通信速度を増すことができます。IdM Web UI を使用して DNS ロケーションを作成するには、この手順に従ってください。

前提条件

- IdM デプロイメントに DNS が統合されている。
- IdM で DNS の場所を作成するパーミッションがある。(例: IdM 管理者としてログイン)。

手順

1. **IPA Server** タブを開きます。
2. **Topology** サブタブを選択します。
3. ナビゲーションバーの **IPA の場所** をクリックします。
4. ロケーションリストの上部にある **追加** をクリックします。
5. ロケーション名を入力します。
6. **追加** ボタンをクリックして場所を保存します。
7. オプション: 手順を繰り返して、さらに場所を追加します。

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 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。

4.5. IDM CLI を使用した DNS の場所の作成

DNS の場所を使用すると、Identity Management (IdM) クライアントとサーバー間の通信速度を増すことができます。IdM コマンドラインインターフェイス (CLI) で **ipa location-add** コマンドを使用して DNS ロケーションを作成するには、この手順に従います。

前提条件

- IdM デプロイメントに DNS が統合されている。
- IdM で DNS の場所を作成するパーミッションがある。(例: IdM 管理者としてログイン)。

手順

1. たとえば、新しい場所 **germany** を作成するには、以下を入力します。

```
$ ipa location-add germany
----------------------------
```

```
Added IPA location "germany"
----------------------------
  Location name: germany
```

2. オプション: この手順を繰り返して、さらに場所を追加します。

関連情報

- 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。
- 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。

4.6. IDM WEB UI を使用した DNS の場所への IDM サーバーの割り当て

Identity Management (IdM) の DNS の場所を使用すると、IdM クライアントとサーバー間の通信速度を増すことができます。IdM Web UI を使用して IdM サーバーを DNS ロケーションに割り当てるには、この手順に従います。

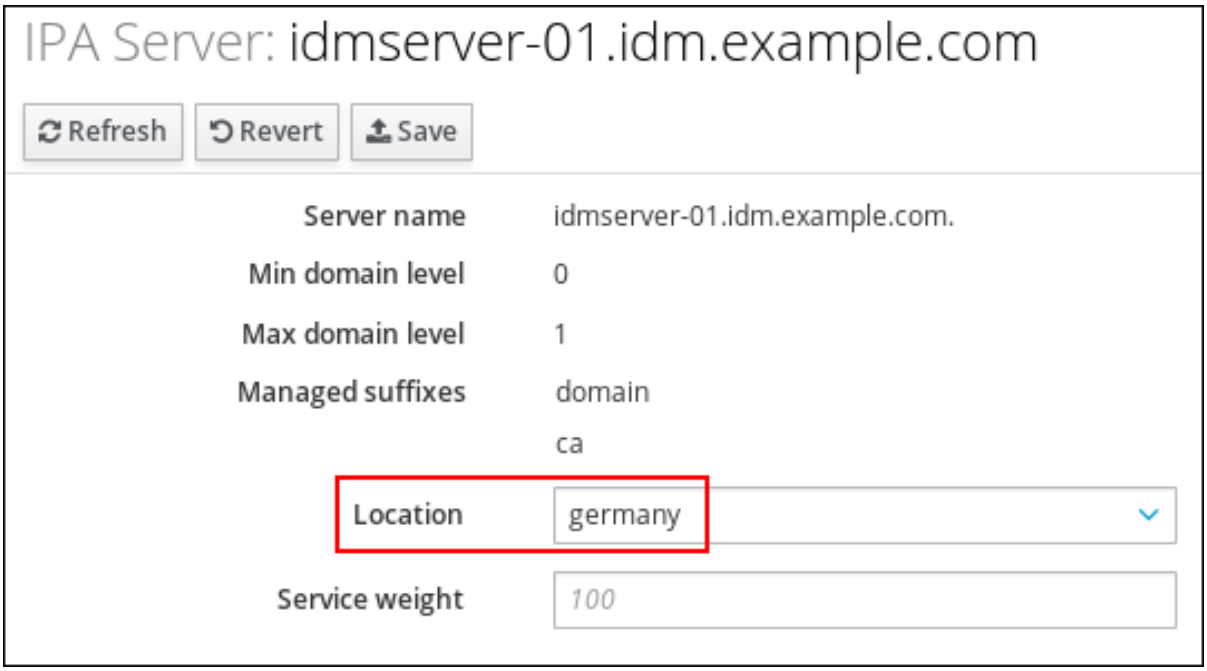
前提条件

- IdM デプロイメントに DNS が統合されている。
- たとえば、IdM admin ユーザーなど、DNS の場所を割り当てるパーミッションがあるユーザーとしてログインしている。
- DNS の場所を割り当てるホストへの **root** アクセス権がある。
- サーバーを割り当てる IdM DNS の場所を作成 している。

手順

1. **IPA Server** タブを開きます。
2. **Topology** サブタブを選択します。
3. ナビゲーションにある **IPA Servers** をクリックします。
4. IdM サーバー名をクリックします。
5. DNS の場所を選択し、必要に応じてサービスの加重を設定します。

図4.1 DNS の場所へのサーバーの割り当て



6. **Save** をクリックします。

7. 前の手順で DNS の場所を割り当てたホストのコマンドラインインターフェイス (CLI) で、**named-pkcs11** サービスを再起動します。

```
[root@idmserver-01 ~]# systemctl restart named-pkcs11
```

8. オプション: この手順を繰り返して、他の IdM サーバーに DNS の場所を割り当てます。

関連情報

- IdM クライアントが同じ場所にある IdM サーバーを使用するように設定する手順 を参照してください。

4.7. IDM CLI を使用した DNS の場所への IDM サーバーの割り当て

Identity Management (IdM) の DNS の場所を使用すると、IdM クライアントとサーバー間の通信速度を増すことができます。IdM コマンドラインインターフェイス (CLI) を使用して IdM サーバーを DNS の場所に割り当てるには、次の手順に従います。

前提条件

- IdM デプロイメントに DNS が統合されている。
- たとえば、IdM admin ユーザーなど、DNS の場所を割り当てるパーミッションがあるユーザーとしてログインしている。
- DNS の場所を割り当てるホストへの **root** アクセス権がある。
- サーバーを割り当てる IdM DNS の場所を作成 している。

手順

1. オプション: 設定済みの DNS の場所をすべて表示します。

```
[root@server ~]# ipa location-find
----------------------
2 IPA locations matched
----------------------
Location name: australia
Location name: germany
----------------------------
Number of entries returned: 2
----------------------------
```

2. サーバーを DNS の場所に割り当てます。たとえば、場所 **germany** を **idmserver-01.idm.example.com** サーバーに割り当てるには、以下を実行します。

```
# ipa server-mod idmserver-01.idm.example.com --location=germany
ipa: WARNING: Service named-pkcs11.service requires restart on IPA server
idmserver-01.idm.example.com to apply configuration changes.
-----------------------------------------------
Modified IPA server "idmserver-01.idm.example.com"
-----------------------------------------------
Servername: idmserver-01.idm.example.com
Min domain level: 0
Max domain level: 1
Location: germany
Enabled server roles: DNS server, NTP server
```

3. 前の手順で DNS の場所を割り当てたホストで **named-pkcs11** サービスを再起動します。

```
# systemctl restart named-pkcs11
```

4. オプション: この手順を繰り返して、他の IdM サーバーに DNS の場所を割り当てます。

関連情報

- IdM クライアントが同じ場所にある IdM サーバーを使用するように設定する手順 を参照してください。

4.8. IDM クライアントが同じ場所にある IDM サーバーを使用するように設定する手順

Identity Management (IdM) サーバーは、IdM Web UI を使用した DNS の場所への IdM サーバーの割り当て で説明されているように、DNS の場所に割り当てます。これで、IdM サーバーと同じ場所にある DNS サーバーを使用するようにクライアントを設定できます。

- **DHCP** サーバーが DNS サーバーの IP アドレスをクライアントに割り当てる場合は、**DHCP** サービスを設定します。**DHCP** サービスで DNS サーバーを割り当てる方法は、**DHCP** サービスのドキュメントを参照してください。

- クライアントに **DHCP** サーバーから DNS サーバーの IP アドレスが割り当てられない場合は、クライアントのネットワーク設定で IP を手動で設定します。Red Hat Enterprise Linux でネットワークを設定する方法は、**Red Hat Enterprise Linux ネットワークガイド**の ネットワー接続の設定 セクションを参照してください。

注記

別のロケーションに割り当てられた DNS サーバーを使用するようにクライアントを設定すると、クライアントは両方の場所にある IdM サーバーに接続します。

例4.3 クライアントの場所により変化するネームサーバーエントリー

以下の例は、場所が異なるクライアントの **/etc/resolv.conf** ファイルにあるさまざまなネームサーバーエントリーを示しています。

プラハのクライアント:

```
nameserver 10.10.0.1
nameserver 10.10.0.2
```

パリのクライアント:

```
nameserver 10.50.0.1
nameserver 10.50.0.3
```

オスロのクライアント:

```
nameserver 10.30.0.1
```

ベルリンのクライアント:

```
nameserver 10.30.0.1
```

各 DNS サーバーが IdM の場所に割り当てられている場合に、クライアントはその場所にある IdM サーバーを使用します。

4.9. 関連情報

- 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。

第5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

5.1. 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5.1 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** (優先順位):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5.2 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。

5.2. 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。

5.3. 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。

5.4. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

5.5. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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
```

5.6. 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6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

6.1. 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。

6.2. 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)

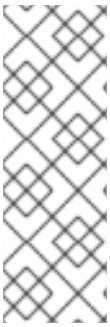
IdM BIND サービスは、DNS クエリーを設定済みのフォワーダーに転送します。サーバーエラーやタイムアウトが原因でクエリーに失敗すると、BIND はインターネット上のサーバーを使用して再帰解決にフォールバックします。**forward first** ポリシーはデフォルトのポリシーで、DNS トラフィックの最適化に適しています。

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6.1 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。

6.3. IDM WEB UI でのグローバルフォワーダーの追加

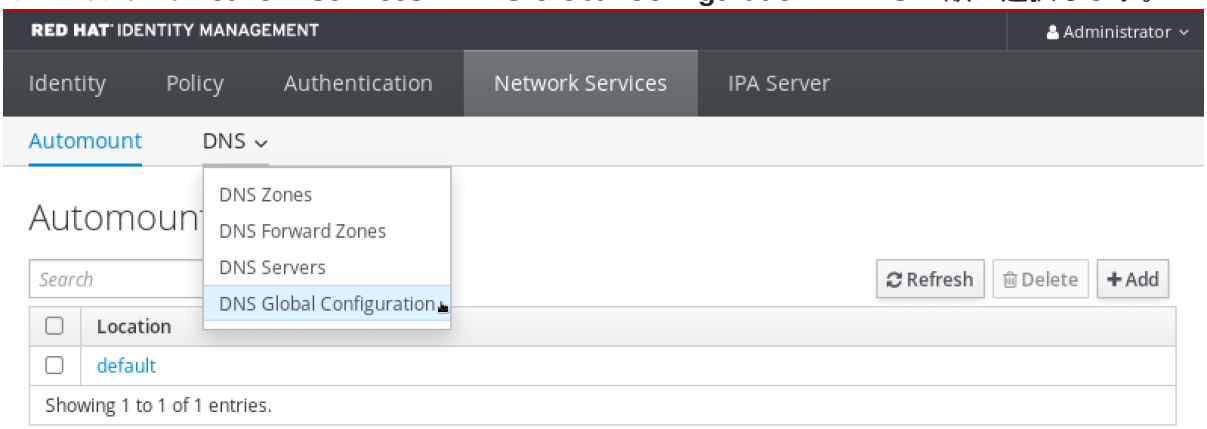
以下の手順に従って、Identity Management (IdM) Web UI でグローバル DNS フォワーダーを追加します。

前提条件

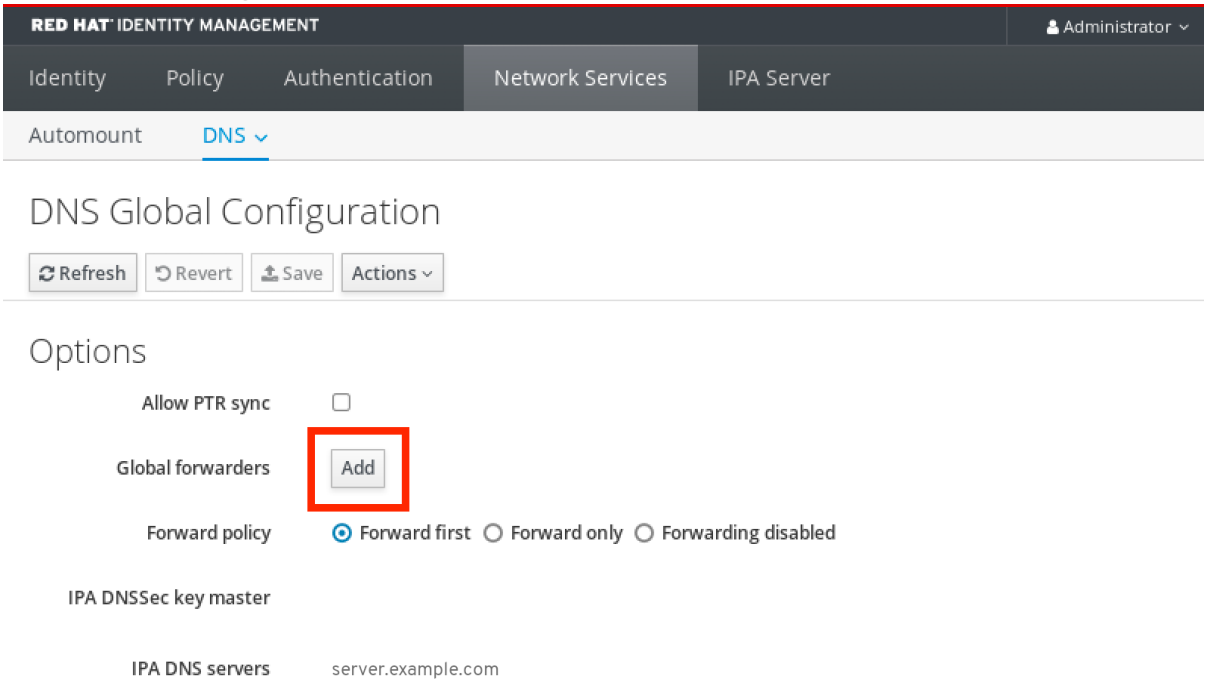
- IdM 管理者として IdM WebUI にログインしている。
- クエリーを転送する DNS サーバーのインターネットプロトコル (IP) アドレスを知っている。

手順

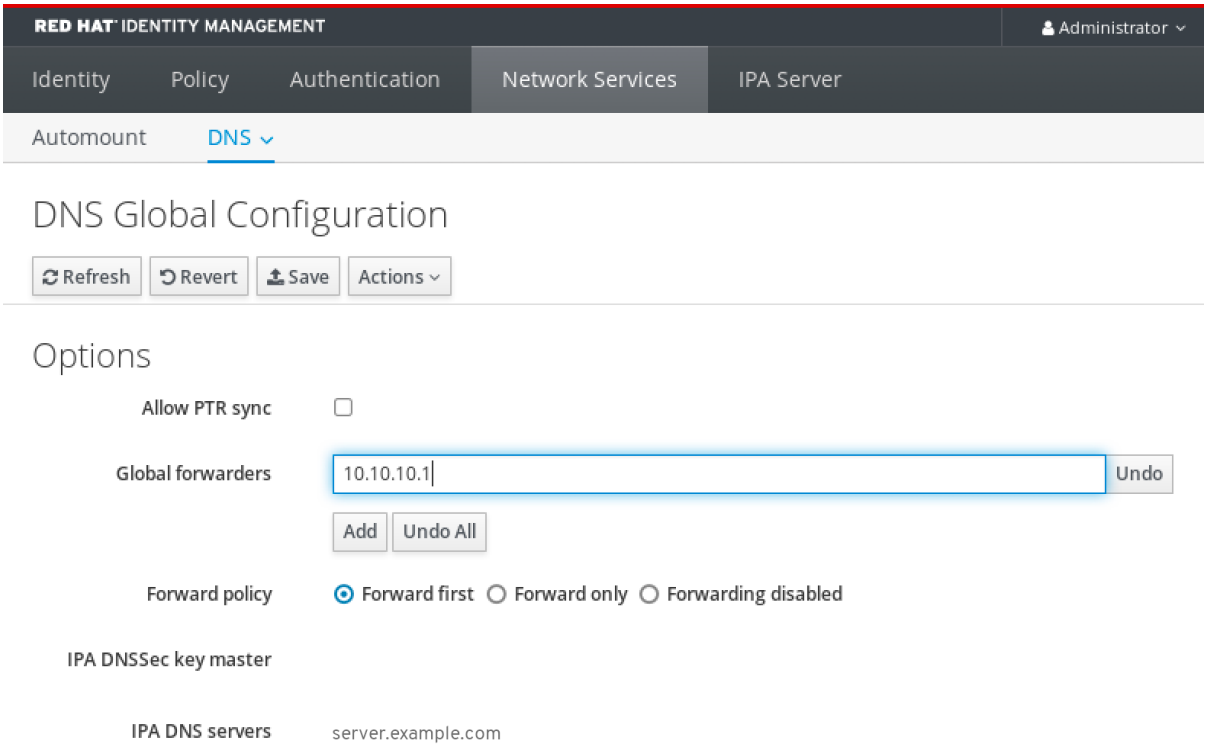
1. IdM Web UI で **Network Services → DNS Global Configuration → DNS** の順に選択します。



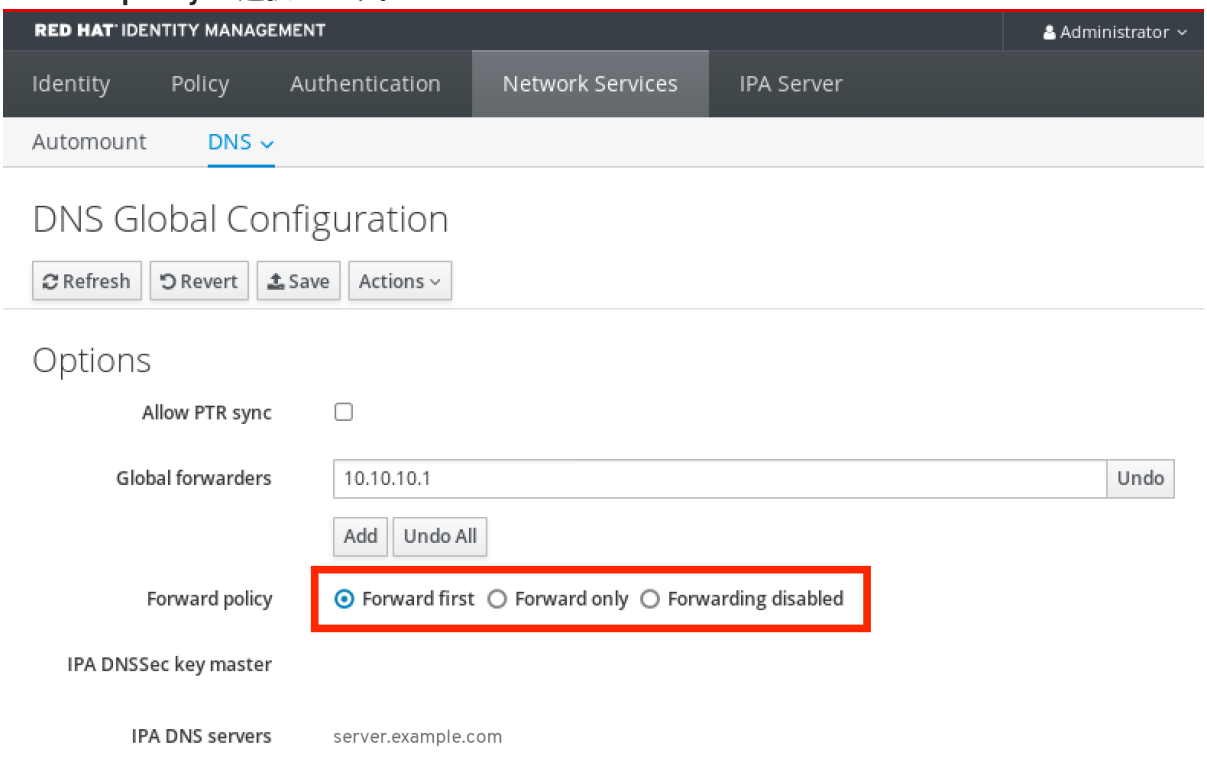
2. **DNS Global Configuration** セクションで、**Add** をクリックします。



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。



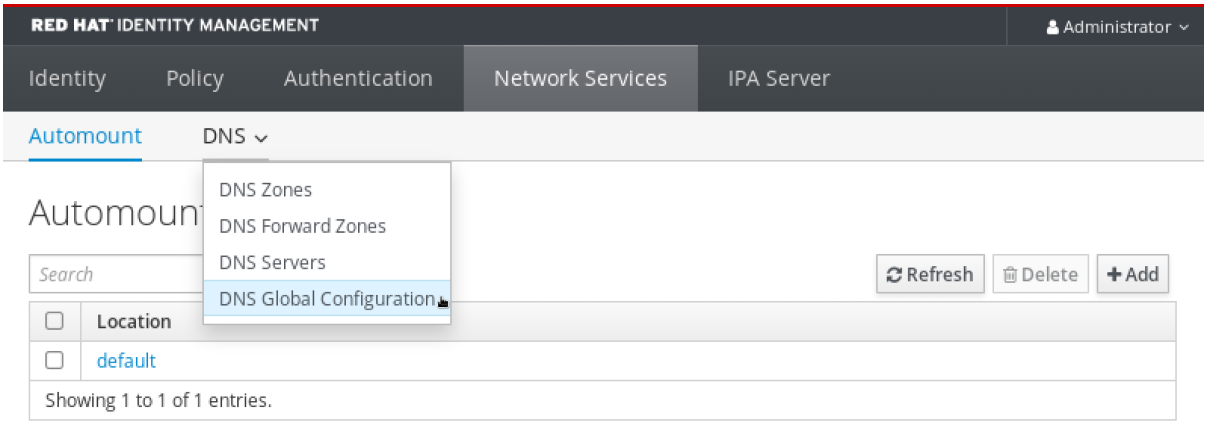
4. **Forward policy** を選択します。



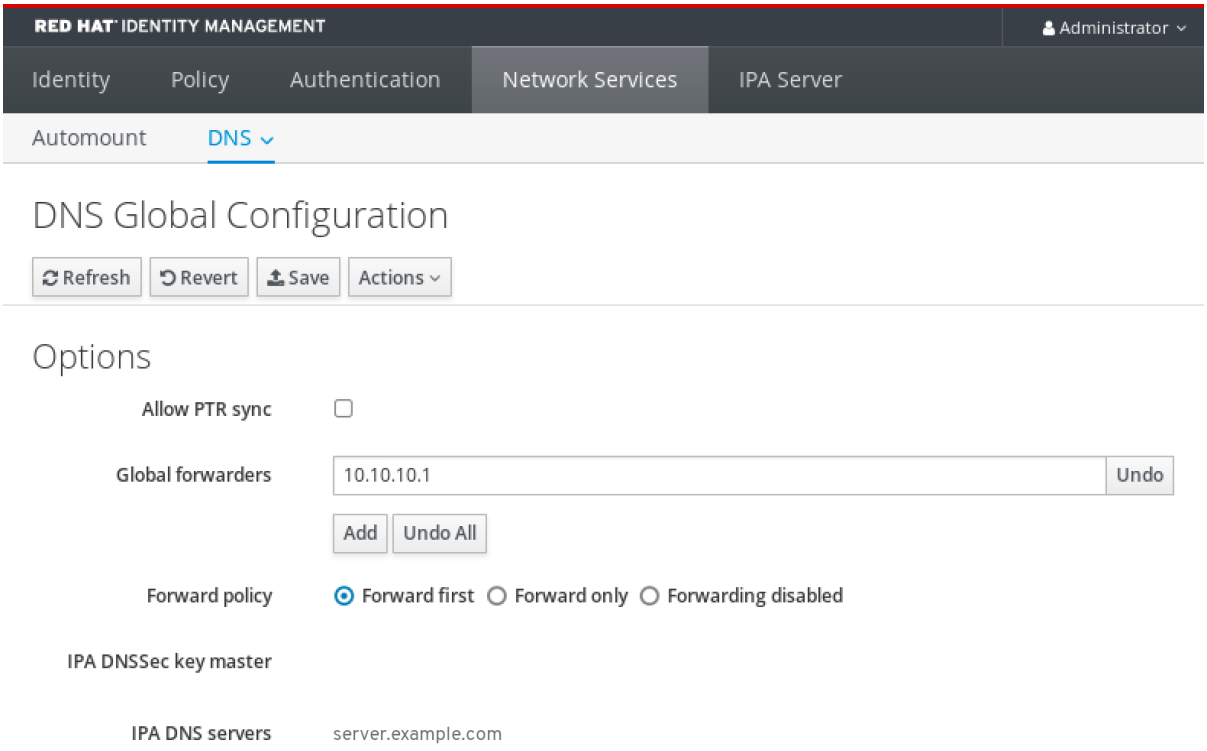
5. ウィンドウの上部にある **Save** をクリックします。

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。



6.4. 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
```

6.5. 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。

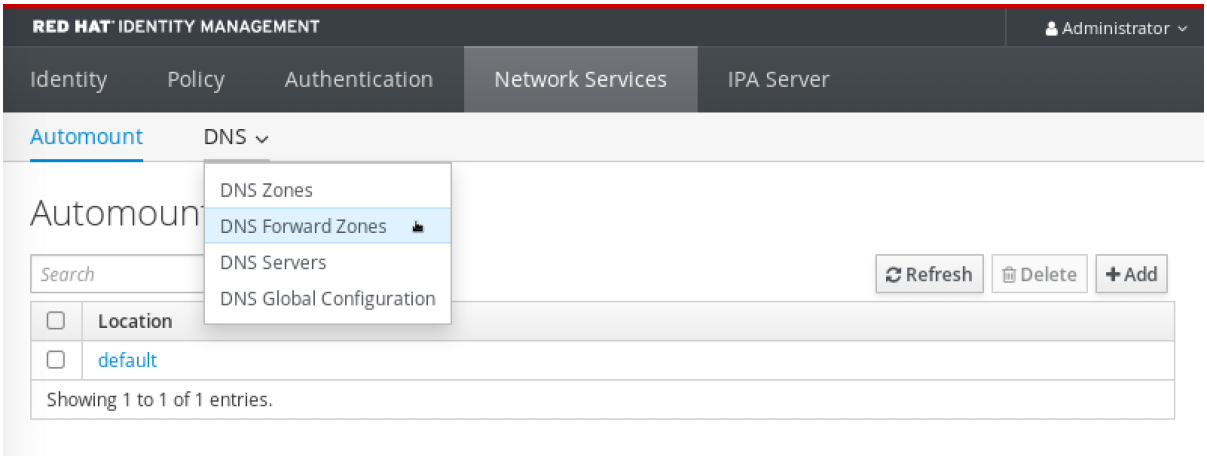
新しい DNS ゾーンを作成する場合には、Red Hat は、ネームサーバー (NS) レコードで標準の DNS 委譲を常に使用し、正引きゾーンを回避することを推奨します。多くの場合、グローバルフォワーダーを使用するだけで十分なため、正引きゾーンは必要ありません。

前提条件

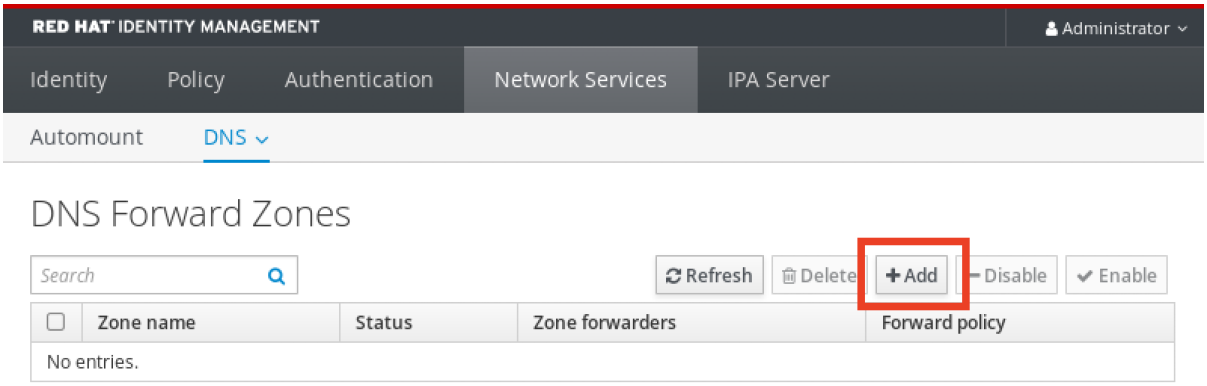
- IdM 管理者として IdM WebUI にログインしている。
- クエリーを転送する DNS サーバーのインターネットプロトコル (IP) アドレスを知っている。

手順

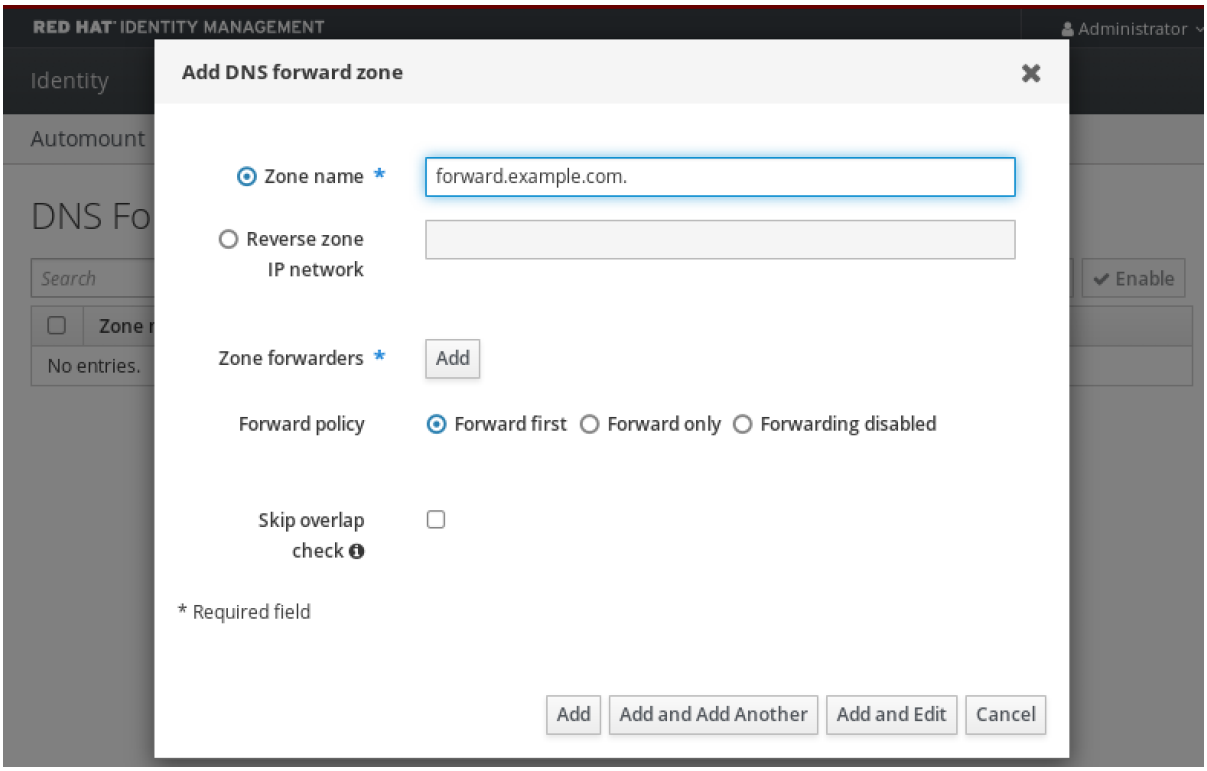
1. IdM Web UI で **Network Services → DNS Forward Zones → DNS** の順に選択します。



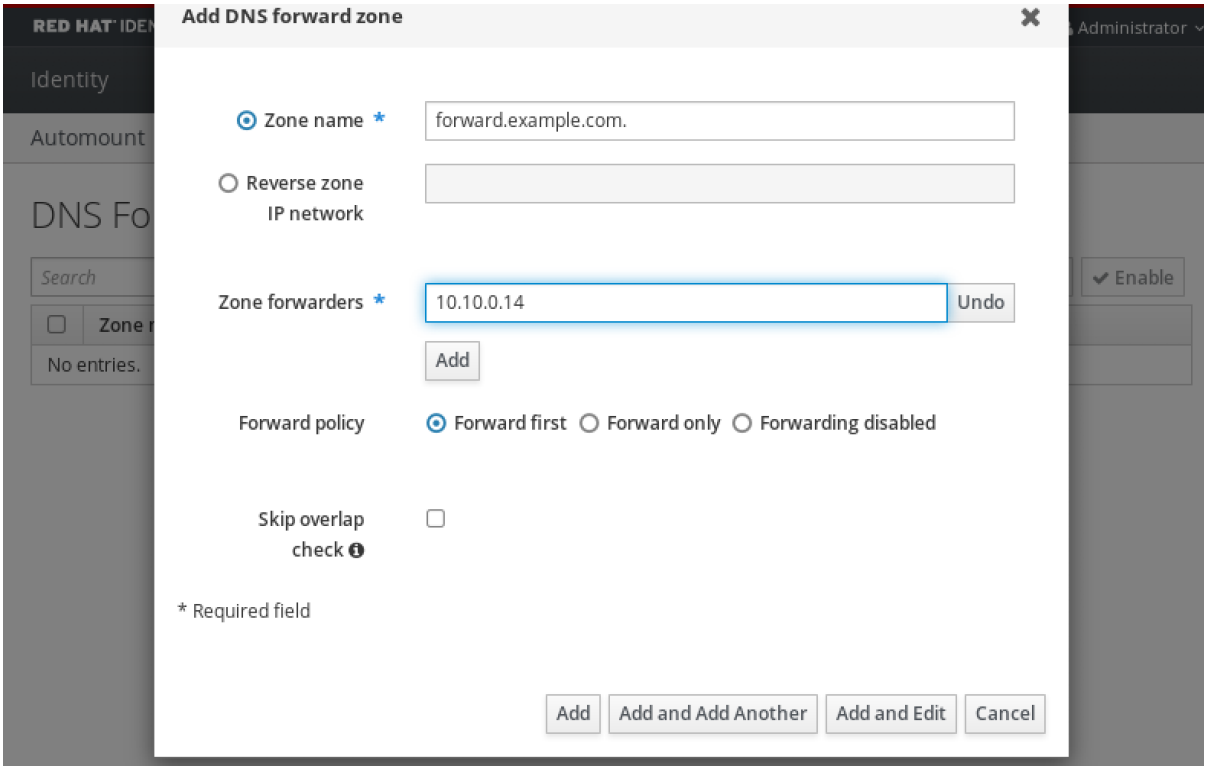
2. **DNS Forward Zones** セクションで、**Add** をクリックします。



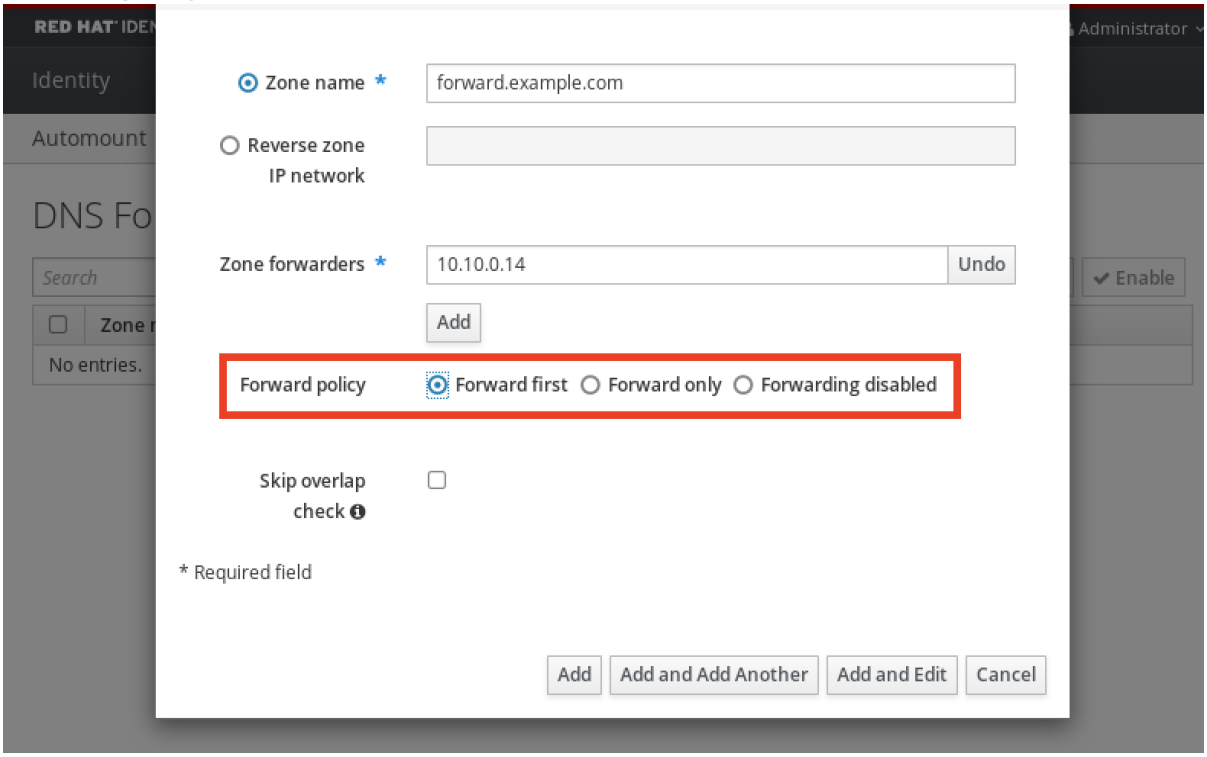
3. **Add DNS forward zone** ウィンドウで、正引きゾーン名を指定します。



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。



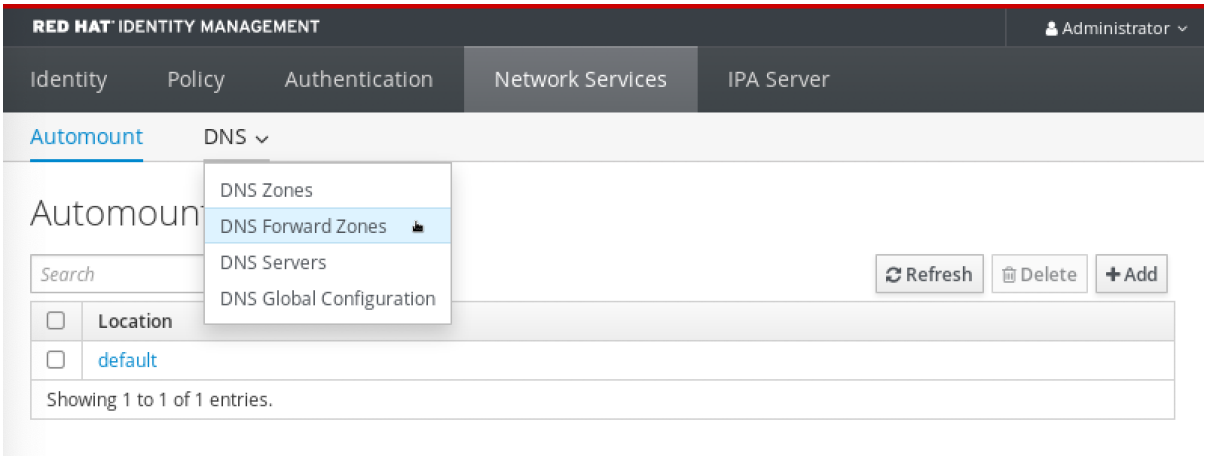
5. **Forward policy** を選択します。



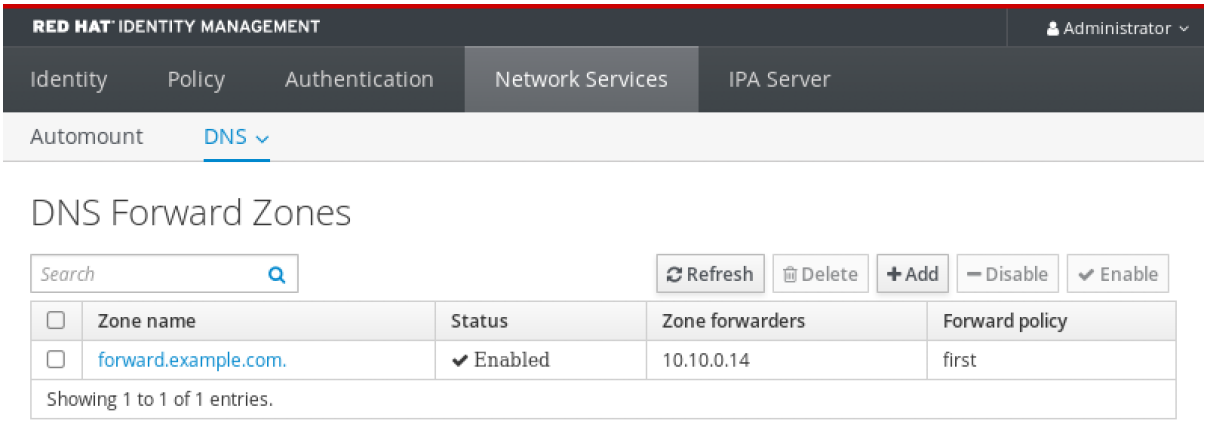
6. ウィンドウの下部にある **Add** をクリックして、新しい正引きゾーンを追加します。

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。



6.6. 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
```

6.7. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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**.

      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。

6.8. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

6.9. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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**.

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。

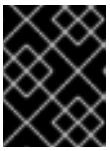
d. **action** 変数は **member** に設定します。

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

6.10. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

6.11. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

6.12. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

6.13. 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ている。

- 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。

- 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。

6.14. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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7章 IDM での DNS レコードの管理

本章では、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 レコードを追加、変更、および削除できます。本章は以下のセクションで設定されます。

- IdM の DNS レコード
- IdM Web UI からの DNS リソースレコードの追加
- IdM CLI からの DNS リソースレコードの追加
- 一般的な ipa dnsrecord-add オプション
- IdM Web UI での DNS レコードの削除
- IdM Web UI での DNS レコード全体の削除
- IdM CLI での DNS レコードの削除

前提条件

- IdM デプロイメントに統合 DNS サーバーが含まれている。統合 DNS のある IdM のインストール方法は、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 のある統合 CA をルート CA とするサーバー
  - IdM サーバーのインストール: 統合 DNS と外部 CA をルート CA とするサーバー

7.1. 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。

7.2. IDM WEB UI での DNS リソースレコードの追加

Identity Management (IdM) Web UI に DNS リソースレコードを追加するには、次の手順に従います。

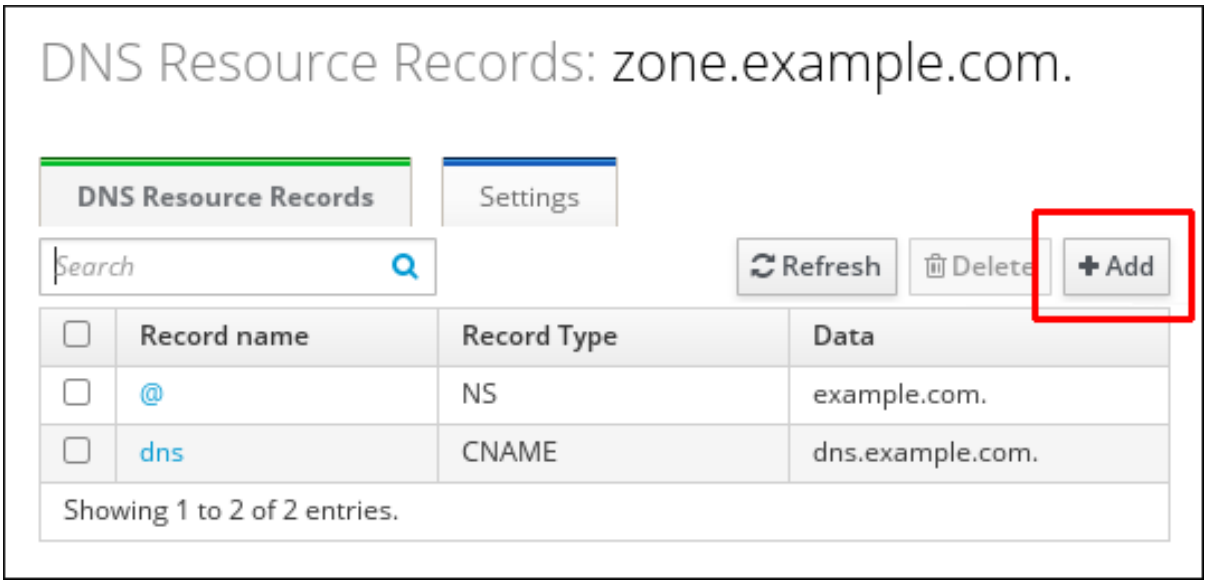
前提条件

- DNS レコードの追加先の DNS ゾーンが存在し、IdM が管理している。IdM DNS での DNS ゾーンの作成に関する詳細は、IdM の DNS ゾーンの管理 を参照してください。
- IdM 管理者としてログインしている。

手順

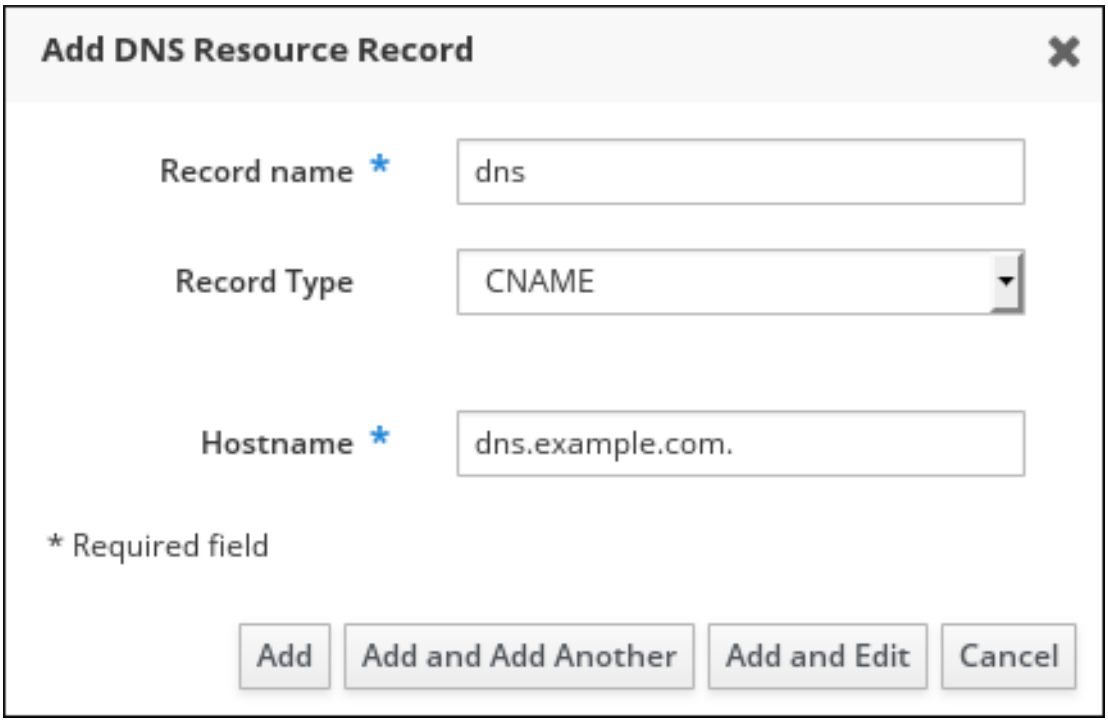
1. IdM Web UI で、**Network Services → DNS → DNS Zones** の順にクリックします。
2. DNS レコードを追加する DNS ゾーンをクリックします。
3. **DNS Resource Record** セクションで、**Add** をクリックして新規レコードを追加します。

図7.1 新しい DNS リソースレコードの追加



4. 作成するレコードのタイプを選択し、必要に応じて他のフィールドにも入力します。

図7.2 新しい DNS リソースレコードの定義



5. **Add** をクリックして、新規レコードを確定します。

7.3. IDM CLI からの DNS リソースレコードの追加

コマンドラインインターフェイス (CLI) から任意のタイプの DNS リソースレコードを追加するには、次の手順に従います。

前提条件

- DNS レコードを追加する DNS ゾーンが存在する。IdM DNS での DNS ゾーンの作成に関する詳細は、IdM の DNS ゾーンの管理 を参照してください。

- IdM 管理者としてログインしている。

手順

1. DNS リソースレコードを追加するには、**ipa dnsrecord-add** コマンドを使用します。このコマンドは、以下の構文に従います。

```
$ ipa dnsrecord-add zone_name record_name --record_type_option=data
```

上記のコマンドでは、以下のようになります。

- **zone_name** は、レコードを追加する DNS ゾーンの名前です。
- **record_name** は、新しい DNS リソースレコードの識別子です。

たとえば、**host1** の A タイプ DNS レコードを **idm.example.com** ゾーンに追加するには、次のコマンドを実行します。

```
$ ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123
```

7.4. 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7.1 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7.2 "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**

オプション	説明	例
	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]
--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7.3 "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7.4 "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.**

オプション	説明	例
--ptr-hostname=string	レコードのホスト名を指定します。	

表7.5 "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。

7.5. IDM WEB UI での DNS レコードの削除

IdM Web UI を使用して Identity Management (IdM) の DNS レコードを削除するには、この手順に従います。

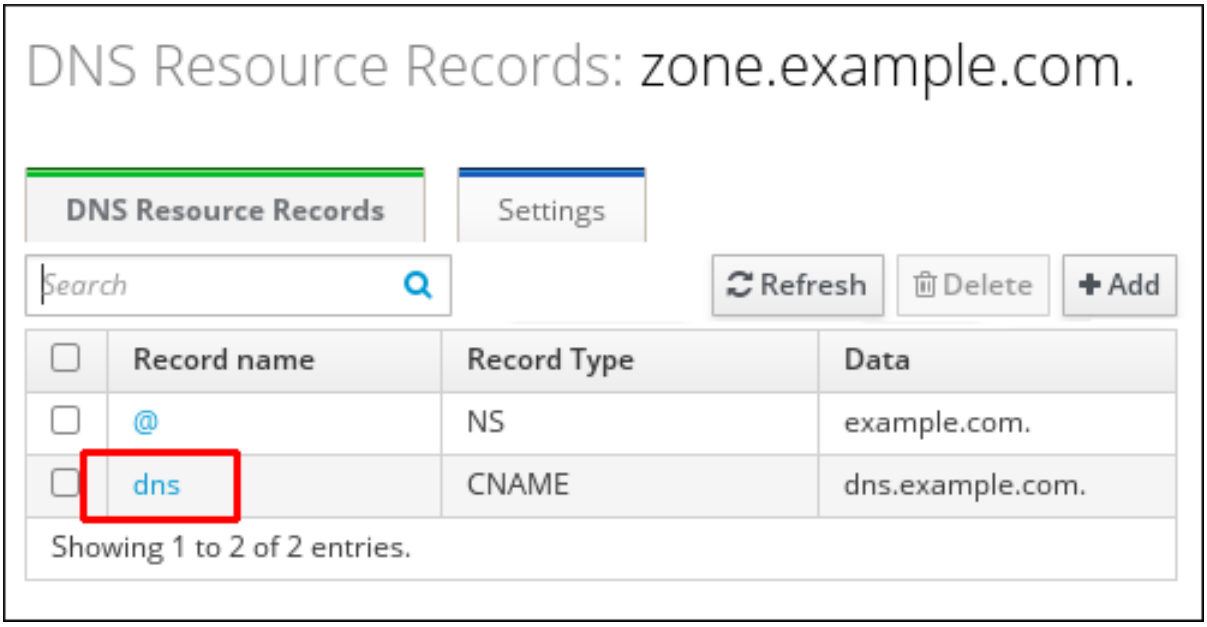
前提条件

- IdM 管理者としてログインしている。

手順

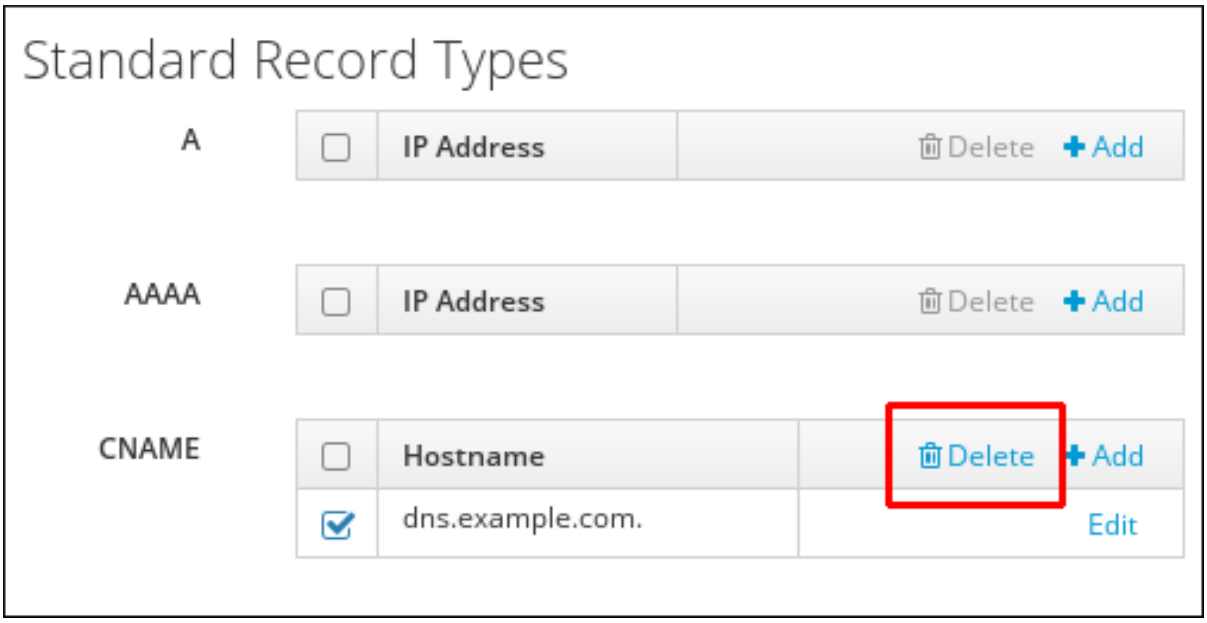
1. IdM Web UI で、**Network Services → DNS → DNS Zones** の順にクリックします。
2. DNS レコードを削除するゾーン (**example.com** など) をクリックします。
3. **DNS Resource Record** のセクションで、リソースレコードの名前をクリックします。

   図7.3 DNS リソースレコードの選択



4. 削除するレコードタイプの名前の横にあるチェックボックスを選択します。
5. **Delete** をクリックします。

図7.4 DNS リソースレコードの削除



選択したレコードタイプが削除されました。リソースレコードの他の設定はそのままになります。

関連情報

- IdM Web UI での DNS レコード全体の削除 を参照してください。

7.6. IDM WEB UI での DNS レコード全体の削除

Identity Management (IdM) Web UI を使用してゾーン内の特定のリソースのすべてのレコードを削除するには、次の手順に従います。

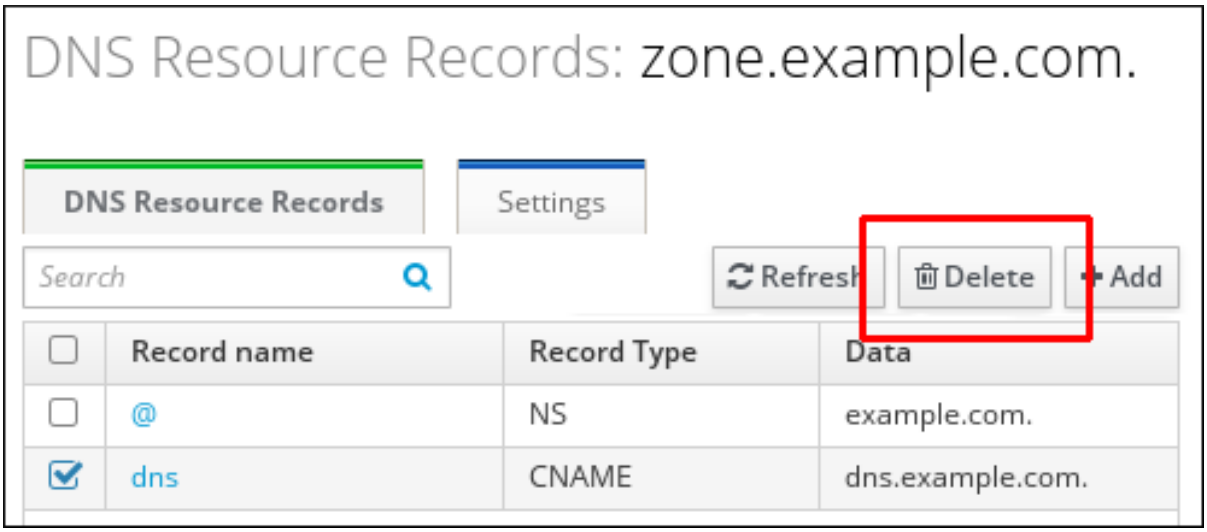
前提条件

- IdM 管理者としてログインしている。

手順

1. IdM Web UI で、**Network Services → DNS → DNS Zones** の順にクリックします。
2. DNS レコードを削除するゾーン (例: **zone.example.com**) をクリックします。
3. **DNS Resource Record** セクションで、削除するリソースレコードのチェックボックスを選択します。
4. **Delete** をクリックします。

図7.5 全リソースレコードの削除



リソースレコードがすべて削除されました。

7.7. IDM CLI での DNS レコードの削除

Identity Management (IdM) DNS によって管理されるゾーンから DNS レコードを削除するには、次の手順に従います。

前提条件

- IdM 管理者としてログインしている。

手順

- ゾーンからレコードを削除するには **ipa dnsrecord-del** コマンドを使用して、**--recordType-rec** オプションでレコードの値を指定して追加します。たとえば、A タイプのレコードを削除するには以下を実行します。

```
$ ipa dnsrecord-del example.com www --a-rec 192.0.2.1
```

オプションなしで **ipa dnsrecord-del** コマンドを実行すると、削除するレコードについての情報の入力が求められます。**--del-all** オプションを指定してコマンドを実行すると、ゾーンに関連するレコードがすべて削除されることに注意してください。

関連情報

- **ipa dnsrecord-del --help** コマンドを実行します。

7.8. 関連情報

- Ansible を使用した IdM での DNS レコードの管理 を参照してください。

第8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。本章は以下のセクションで設定されます。

- 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

8.1. 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。

8.2. 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8.1 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8.2 "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8.3 "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8.4 "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8.5 "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。

8.3. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

8.4. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。

- 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。

8.5. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

8.6. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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。

8.7. 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ている。
  - 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 ボールトに **ipaadmin_password** が保存されていることを前提としている。
- 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9章 IDM で標準 DNS ホスト名の使用

DNS 正規化は、潜在的なセキュリティーリスクを回避するために、Identity Management (IdM) クライアントでデフォルトで無効になっています。たとえば、攻撃者がドメインの DNS サーバーとホストを制御している場合、攻撃者は短いホスト名 (**demo** など) を、侵害されたホスト (**malicious.example.com** など) に解決させることができます。この場合、ユーザーは想定とは異なるサーバーに接続します。

この手順では、IdM クライアントで正規化されたホスト名を使用する方法について説明します。

9.1. ホストプリンシパルへのエイリアスの追加

デフォルトでは、**ipa-client-install** コマンドを使用して登録した Identity Management (IdM) クライアントでは、サービスプリンシパルで短縮ホスト名を使用することができません。たとえば、ユーザーがサービスにアクセスするときに、**host/demo@EXAMPLE.COM** ではなく、**host/demo.example.com@EXAMPLE.COM** のみを使用できます。

Kerberos プリンシパルにエイリアスを追加するには、次の手順に従います。または、**/etc/krb5.conf** ファイルでホスト名の正規化を有効にできます。詳細は、 クライアントのサービスプリンシパルでのホスト名の正規化の有効化 を参照してください。

前提条件

- IdM クライアントがインストールされている。
- ホスト名が、ネットワーク内で一意の名前である。

手順

1. **admin** ユーザーとして、IdM に対して認証します。

```
$ kinit admin
```

2. エイリアスをホストプリンシパルに追加します。たとえば、**demo** エイリアスを、**demo.examle.com** ホストプリンシパルに追加するには、次のコマンドを実行します。

```
$ ipa host-add-principal demo.example.com --principal=demo
```

9.2. クライアントのサービスプリンシパルでのホスト名の正規化の有効化

クライアント上のサービスプリンシパルのホスト名の正規化を有効にするには、次の手順に従います。

ホストプリンシパルへのエイリアスの追加 の説明に従って、ホストプリンシパルのエイリアスを使用する場合は、正規化を有効にする必要がないことに注意してください。

前提条件

- Identity Management (IdM) クライアントがインストールされている。
- **root** ユーザーとして IdM クライアントにログインしている。
- ホスト名が、ネットワーク内で一意の名前である。

手順

1. **/etc/krb5.conf** ファイルの **[libdefaults]** セクションで、**dns_canonicalize_hostname** パラメーターを **false** に設定します。

```
[libdefaults]
...
dns_canonicalize_hostname = true
```

9.3. DNS ホスト名の正規化を有効にしてホスト名を使用するためのオプション

クライアントのサービスプリンシパルでのホスト名の正規化の有効化 の説明に従って、**/etc/krb5.conf** ファイルに **dns_canonicalize_hostname = true** を設定すると、サービスプリンシパルでホスト名を使用する際に、以下のオプションがあります。

- Identity Management (IdM) 環境では、**host/demo.example.com@EXAMPLE.COM** などのサービスプリンシパルで完全なホスト名を使用できます。
- IdM がない環境では、RHEL ホストを Active Directory (AD) ドメインのメンバーとする場合に、AD ドメインコントローラー (DC) が、AD に登録されているマシンの NetBIOS 名のサービスプリンシパルを自動的に作成するため、これ以上考慮が必要な事項はありません。