



Red Hat Enterprise Linux 9

在身份管理中使用 DNS

管理 IdM 集成的 DNS 服务

法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

DNS 是 Red Hat Identity Management (IdM) 域中的重要组件。例如，客户端使用 DNS 查找服务并识别同一站点中的服务器。您可以使用命令行、IdM Web UI 和 Ansible Playbook 管理 IdM 中集成的 DNS 服务器中的记录、区域、位置和转发。

目录

对红帽文档提供反馈	4
第 1 章 使用 ANSIBLE PLAYBOOK 管理 IDM 中的全局 DNS 配置	5
1.1. IDM 如何确保 /ETC/RESOLV.CONF 中的全局转发器不会被 NETWORKMANAGER 删除	5
1.2. 使用 ANSIBLE 在 IDM 中存在 DNS 全局转发器	6
1.3. 使用 ANSIBLE 确保 IDM 中没有 DNS 全局转发器	8
1.4. IPADNSCONFIG ANSIBLE-FREEIPA 模块中的 ACTION: MEMBER 选项	9
1.5. IDM 中的 DNS 转发策略	10
1.6. 使用 ANSIBLE PLAYBOOK 来确保在 IDM DNS 全局配置中设置了转发第一个策略	11
1.7. 使用 ANSIBLE PLAYBOOK 来确保 IDM DNS 中禁用了全局转发器	13
1.8. 使用 ANSIBLE PLAYBOOK 来确保 IDM DNS 中禁用了转发和反向查询区的同步	14
第 2 章 管理 IDM 中的 DNS 区域	16
2.1. 支持的 DNS 区类型	16
2.2. 在 IDM WEB UI 中添加主 DNS 区域	17
2.3. 在 IDM CLI 中添加主 DNS 区	18
2.4. 在 IDM WEB UI 中删除主 DNS 区	19
2.5. 在 IDM CLI 中删除主 DNS 区	19
2.6. DNS 配置优先级	19
2.7. 主 IDM DNS 区的配置属性	20
2.8. 在 IDM WEB UI 中编辑主 DNS 区的配置	21
2.9. 在 IDM CLI 中编辑主 DNS 区的配置	23
2.10. IDM 中的区传输	23
2.11. 在 IDM WEB UI 中启用区传输	24
2.12. 在 IDM CLI 中启用区传输	24
2.13. 其他资源	25
第 3 章 使用 ANSIBLE PLAYBOOK 管理 IDM DNS 区域	26
3.1. 支持的 DNS 区类型	26
3.2. 主 IDM DNS 区的配置属性	27
3.3. 使用 ANSIBLE 在 IDM DNS 中创建主区	28
3.4. 使用 ANSIBLE PLAYBOOK 来确保 IDM 中存在带有多个变量的主 DNS 区域	30
3.5. 使用 ANSIBLE PLAYBOOK 以确保在指定 IP 地址时存在用于反向 DNS 查找的区域	32
第 4 章 管理 IDM 中的 DNS 位置	35
4.1. 基于 DNS 的服务发现	35
4.2. DNS 位置的部署注意事项	36
4.3. DNS 时间到实时(TTL)	36
4.4. 使用 IDM WEB UI 创建 DNS 位置	36
4.5. 使用 IDM CLI 创建 DNS 位置	37
4.6. 使用 IDM WEB UI 将 IDM 服务器分配给 DNS 位置	37
4.7. 使用 IDM CLI 将 IDM 服务器分配给 DNS 位置	39
4.8. 配置 IDM 客户端在同一位置使用 IDM 服务器	40
4.9. 其他资源	40
第 5 章 使用 ANSIBLE 管理 IDM 中的 DNS 位置	41
5.1. 基于 DNS 的服务发现	41
5.2. DNS 位置的部署注意事项	42
5.3. DNS 时间到实时(TTL)	42
5.4. 使用 ANSIBLE 确保存在 IDM 位置	42
5.5. 使用 ANSIBLE 确保不存在 IDM 位置	44
5.6. 其他资源	45

第 6 章 在 IDM 中管理 DNS 转发	46
6.1. IDM DNS 服务器的两个角色	46
6.2. IDM 中的 DNS 转发策略	46
6.3. 在 IDM WEB UI 中添加全局转发器	47
6.4. 在 CLI 中添加全局转发器	50
6.5. 在 IDM WEB UI 中添加 DNS 转发区域	51
6.6. 在 CLI 中添加 DNS 转发区域	54
6.7. 使用 ANSIBLE 在 IDM 中建立 DNS 全局转发器	55
6.8. 使用 ANSIBLE 确保 IDM 中存在 DNS 全局转发器	56
6.9. 使用 ANSIBLE 确保 IDM 中没有 DNS 全局转发器	58
6.10. 使用 ANSIBLE 确保 DNS 全局转发器在 IDM 中被禁用	60
6.11. 使用 ANSIBLE 确保 IDM 中存在 DNS 转发区域	61
6.12. 使用 ANSIBLE 确保 DNS 转发区域在 IDM 中有多个转发器	63
6.13. 使用 ANSIBLE 确保 IDM 中 DNS FORWARD 区域被禁用	64
6.14. 使用 ANSIBLE 确保 IDM 中没有 DNS 转发区域	66
第 7 章 管理 IDM 中的 DNS 记录	69
7.1. IDM 中的 DNS 记录	69
7.2. 在 IDM WEB UI 中添加 DNS 资源记录	70
7.3. 从 IDM CLI 添加 DNS 资源记录	71
7.4. 常见 IPA DNSRECORD-* 选项	72
7.5. 删除 IDM WEB UI 中的 DNS 记录	74
7.6. 删除 IDM WEB UI 中的整个 DNS 记录	75
7.7. 删除 IDM CLI 中的 DNS 记录	76
7.8. 其他资源	76
第 8 章 使用 ANSIBLE 管理 IDM 中的 DNS 记录	78
8.1. IDM 中的 DNS 记录	78
8.2. 常见 IPA DNSRECORD-* 选项	79
8.3. 使用 ANSIBLE 确保 IDM 中存在 A 和 AAAA DNS 记录	81
8.4. 使用 ANSIBLE 确保 IDM 中存在 A 和 PTR DNS 记录	83
8.5. 使用 ANSIBLE 确保 IDM 中存在多个 DNS 记录	84
8.6. 使用 ANSIBLE 确保 IDM 中存在多个 CNAME 记录	86
8.7. 使用 ANSIBLE 确保 IDM 中是否存在 SRV 记录	88
第 9 章 在 IDM 中使用规范的 DNS 主机名	91
9.1. 在主机主体中添加别名	91
9.2. 在客户端中的服务主体中启用主机名的规范	91
9.3. 在启用了 DNS 主机名的情况下使用主机名的选项	92

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

通过 Jira 提交反馈（需要帐户）

1. 登录到 [Jira](#) 网站。
2. 在顶部导航栏中点 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 使用 ANSIBLE PLAYBOOK 管理 IDM 中的全局 DNS 配置

使用 Red Hat Ansible Engine **dnsconfig** 模块，您可以为 Identity Management(IdM)DNS 配置全局配置。全局 DNS 配置中定义的设置适用于所有 IdM DNS 服务器。但是，全局配置的优先级低于特定 IdM DNS 区的配置。

dnsconfig 模块支持以下变量：

- 全局转发器，特别是其 IP 地址以及用于通信的端口。
- 全局转发策略：only, first, 或 none。有关这些 DNS 转发策略类型的详情，请参阅 [IdM 中的 DNS 转发策略](#)。
- 转发查找和反向查找区域的同步。

先决条件

- DNS 服务安装在 IdM 服务器中。有关如何安装带有集成 DNS 的 IdM 服务器的详情，请查看以下链接之一：
 - [安装 IdM 服务器：使用集成的 DNS，集成的 CA 作为 root CA](#)
 - [安装 IdM 服务器：具有集成的 DNS，具有外部 CA 作为根 CA](#)
 - [安装 IdM 服务器：使用集成的 DNS,没有 CA](#)

本章包括以下部分：

- [IdM 如何确保 /etc/resolv.conf 中的全局转发器不会被 NetworkManager 删除](#)
- [使用 Ansible 在 IdM 中存在 DNS 全局转发器](#)
- [使用 Ansible 确保 IdM 中没有 DNS 全局转发器](#)
- [ipadnsconfig ansible-freeipa 模块中的 **action: member** 选项](#)
- [IdM 中 DNS 转发策略介绍](#)
- [使用 Ansible playbook 来确保在 IdM DNS 全局配置中设置了转发第一个策略](#)
- [使用 Ansible playbook 来确保 IdM DNS 中禁用了全局转发器](#)
- [使用 Ansible playbook 来确保 IdM DNS 中禁用了转发和反向查询区的同步](#)

1.1. IDM 如何确保 /ETC/RESOLV.CONF 中的全局转发器不会被 NETWORKMANAGER 删除

安装带有集成 DNS 的身份管理(IdM)，配置 **/etc/resolv.conf** 文件指向 **127.0.0.1** localhost 地址：

```
# Generated by NetworkManager
search idm.example.com
nameserver 127.0.0.1
```

在某些环境中，比如使用 **Dynamic Host Configuration Protocol (DHCP)** 的网络，**NetworkManager** 服务可能会恢复对 `/etc/resolv.conf` 文件的更改。要使 DNS 配置持久，IdM DNS 安装过程还会使用以下方法配置 **NetworkManager** 服务：

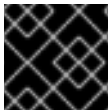
1. DNS 安装脚本会创建一个 `/etc/NetworkManager/conf.d/zzz-ipa.conf` **NetworkManager** 配置文件来控制搜索顺序和 DNS 服务器列表：

```
# auto-generated by IPA installer
[main]
dns=default

[global-dns]
searches=$DOMAIN

[global-dns-domain-*]
servers=127.0.0.1
```

2. **NetworkManager** 服务被重新载入，它总是使用 `/etc/NetworkManager/conf.d/` 目录中的最后一个文件中的设置创建 `/etc/resolv.conf` 文件。在这种情况下，`zzz-ipa.conf` 文件。



重要

不要手动修改 `/etc/resolv.conf` 文件。

1.2. 使用 ANSIBLE 在 IDM 中存在 DNS 全局转发器

按照以下流程，使用 Ansible playbook 确保 DNS 全局转发器在 IdM 中存在。在下例中，IdM 管理员确保在 DNS 服务器中存在一个到 IPv4 地址为 **7.7.9.9**，IP v6 地址为 **2001:db8::1:0**，端口 **53** 的 DNS global forwarder。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 **Ansible 清单文件**。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

- 打开清单文件，并确保 **[ipaserver]** 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 **server.idm.example.com**，请输入：

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

- 复制 **forwarders-absent.yml** Ansible playbook 文件。例如：

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

- 打开 **ensure-presence-of-a-global-forwarder.yml** 文件进行编辑。

- 通过设置以下变量来调整文件：

- 将 playbook 的 **name** 变量更改为 **Playbook**，以确保 IdM DNS 中存在全局转发器。
- 在 **tasks** 部分中，将任务 **name** 更改为 **Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0 on port 53**。
- 在 **ipadnsconfig** 部分的 **forwarders** 部分：
 - 将第一个 **ip_address** 值更改为全局转发器的 IPv4 地址：**7.7.9.9**。
 - 将第二个 **ip_address** 值更改为全局转发器的 IPv6 地址：**2001:db8::1:0**。
 - 验证 **port** 值被设置为 **53**。
- 将 **state** 该为 **present**。
对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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
```

- 保存这个文件。

- 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-
of-a-global-forwarder.yml
```

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsconfig.md` 文件。

1.3. 使用 ANSIBLE 确保 IDM 中没有 DNS 全局转发器

按照以下流程，使用 Ansible playbook 确保 DNS 全局转发器在 IdM 中不存在。在以下示例流程中，IdM 管理员确保在端口 **53** 上没有互联网协议(IP)v4 地址为 **8.8.6.6** 和 IP v6 地址为 **2001:4860:4860::8800** 的 DNS 全局转发器。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 `Ansible 清单文件`。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 `[ipaserver]` 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `forwarders-absent.yml` Ansible playbook 文件。例如：

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. 打开 `ensure-absence-of-a-global-forwarder.yml` 文件进行编辑。

5. 通过设置以下变量来调整文件：

- a. 将 playbook 的 `name` 变量更改为 `Playbook`，以确保 IdM DNS 中没有全局转发器。
- b. 在 `tasks` 部分，将任务的 `name` 改为 `Ensure the absence of a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800 on port 53`。
- c. 在 `ipadnsconfig` 部分的 `forwarders` 部分：
 - i. 将第一个 `ip_address` 值更改为全局转发器的 IPv4 地址：`8.8.6.6`。

- ii. 将第二个 **ip_address** 值更改为全局转发器的 IPv6 地址：**2001:4860:4860::8800**。
- iii. 验证 **port** 值被设置为 **53**。
- d. 将 **action** 变量设置为 **member**。
- e. 验证 **state** 被设置为 **absent**。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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

如果您仅在 playbook 中使用 **state: absent** 选项，而不使用 **action: member**，则 playbook 会失败。

- 6. 保存该文件。
- 7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-of-a-global-forwarder.yml
```

其他资源

- [/usr/share/doc/ansible-freeipa/](#) 目录中的 **README-dnsconfig.md** 文件
- [ipadnsconfig ansible-freeipa](#) 模块中的 **action: member** 选项

1.4. IPADNSCONFIG ANSIBLE-FREEIPA 模块中的 ACTION: MEMBER 选项

使用 **ansible-freeipa ipadnsconfig** 模块在身份管理(IdM)中排除全局转发器，除了使用 **state: absent** 选项外，还需要使用 **action: member** 选项。如果您在 playbook 中只使用 **state: absent**，而不使用 **action: member**，则 playbook 将失败。因此，要删除所有全局转发器，您必须在 playbook 中分别指定它们。相反，**state: present** 选项不需要 **action: member**。

[下表](#) 提供了添加和删除 DNS 全局转发器的配置示例，其演示了 **action: member** 选项的正确使用。表中每一行显示了：

- 执行 playbook 前配置的全局转发器
- playbook 摘录
- 执行 playbook 后配置的全局转发器

表 1.1. 全局转发器的 ipadnsconfig 管理

之前的转发器	Playbook 摘录	之后的转发器
8.8.6.6	<pre>[...] tasks: - name: Ensure the presence of DNS global forwarder 8.8.6.7 ipadnsconfig: forwarders: - ip_address: 8.8.6.7 state: present</pre>	8.8.6.7
8.8.6.6	<pre>[...] 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</pre>	8.8.6.6, 8.8.6.7
8.8.6.6, 8.8.6.7	<pre>[...] tasks: - name: Ensure the absence of DNS global forwarder 8.8.6.7 ipadnsconfig: forwarders: - ip_address: 8.8.6.7 state: absent</pre>	尝试执行 playbook 会 导致错误。 原始配置 - 8.8.6.6、 8.8.6.7 - 保 持不变。
8.8.6.6, 8.8.6.7	<pre>[...] 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</pre>	8.8.6.6

1.5. IDM 中的 DNS 转发策略

IdM 支持 **first** 和 **only** 标准 BIND 转发策略，以及 **none** 特定于 IdM 的转发策略。

Forward first (默认)

IdM BIND 服务将 DNS 查询转发到配置的转发器。如果因为服务器错误或超时导致查询失败，BIND 会使用互联网上的服务器返回到递归解析。**forward first** 策略是默认策略，它适用于优化 DNS 流量。

Forward only

IdM BIND 服务将 DNS 查询转发到配置的转发器。如果因为服务器错误或超时而查询失败，BIND 会将错误返回到客户端。对于带有 split DNS 配置的环境，建议使用 **forward only** 策略。

None (禁用转发)

DNS 查询不会通过 **none** 转发策略转发。禁用转发只作为全局转发配置的特定区覆盖很有用。这个选项等同于在 BIND 配置中指定空转发器列表。



注意

您不能使用转发将 IdM 中的数据与来自其他 DNS 服务器的数据合并。您只能为 IdM DNS 中主区的特定子区转发查询。

默认情况下，如果查询的 DNS 名称属于 IdM 服务器有权威的区域，则 BIND 服务不会将查询转发到另一台服务器。在这种情况下，如果在 IdM 数据库中找不到查询的 DNS 名称，则返回 **NXDOMAIN** 回答。未使用转发功能。

例 1.1. 使用情况示例

IdM 服务器对 **test.example** 具有权威性。DNS 区域。BIND 被配置为把查询转发到带有 **192.0.2.254** IP 地址的 DNS 服务器。

当客户发送对 **nonexistent.test.example.** 的查询 DNS 名称，BIND 检测到 IdM 服务器对 **test.example.** 区域具有权威，且不会将查询转发到 **192.0.2.254.** 服务器。因此，DNS 客户端接收 **NXDomain** 错误消息，告知用户查询域不存在。

1.6. 使用 ANSIBLE PLAYBOOK 来确保在 IDM DNS 全局配置中设置了转发第一个策略

按照以下流程，使用 Ansible playbook 确保 IdM DNS 中的全局转发策略被设置为 **forward first**。

如果您使用 **forward first** DNS 转发策略，DNS 查询会转发到配置的转发器。如果因为服务器错误或超时导致查询失败，BIND 会使用互联网上的服务器返回到递归解析。**forward first** 是默认的策略。它适用于流量优化。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
 - 示例假定在 **~/MyPlaybooks/** 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 **Ansible 清单文件**。
 - 示例假定 **secret.yml** Ansible vault 存储了 **ipadmin_password**。

- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。
- 您的 IdM 环境包含一个集成的 DNS 服务器。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 **[ipaserver]** 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `set-configuration.yml` Ansible playbook 文件。例如：

```
$ cp set-configuration.yml set-forward-policy-to-first.yml
```

4. 打开 `set-forward-policy-to-first.yml` 文件进行编辑。
5. 通过在 `ipadnsconfig` 任务部分中设置以下变量来修改该文件：

- 将 `ipaadmin_password` 变量设置为 IdM 管理员密码。
 - 将 `forward_policy` 变量设置为 `first`。
- 删除原始 playbook 中所有无关的行。这是当前示例修改的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。
7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file set-forward-policy-to-first.yml
```

其他资源

- 请参阅 [IdM 中的 DNS 转发策略](#)。

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsconfig.md` 文件。
- 如需更多 playbook 示例，请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录。

1.7. 使用 ANSIBLE PLAYBOOK 来确保 IDM DNS 中禁用了全局转发器

按照以下流程，使用 Ansible playbook 确保全局转发器在 IdM DNS 中被禁用了。禁用过程可通过将 `forward_policy` 变量设置为 `none` 来完成。

禁用全局转发器会导致无法转发 DNS 查询。禁用转发只作为全局转发配置的特定区覆盖很有用。这个选项等同于在 BIND 配置中指定空转发器列表。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 `Ansible 清单文件`。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。
- 您的 IdM 环境包含一个集成的 DNS 服务器。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 `[ipaserver]` 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `disable-global-forwarders.yml` Ansible playbook 文件。例如：

```
$ cp disable-global-forwarders.yml disable-global-forwarders-copy.yml
```

4. 打开 `disable-global-forwarders-copy.yml` 文件进行编辑。
5. 通过在 `ipadnsconfig` 任务部分中设置以下变量来修改该文件：
 - 将 `ipadmin_password` 变量设置为 IdM 管理员密码。
 - 将 `forward_policy` 变量设置为 `none`。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Playbook to disable global DNS forwarders
  hosts: ipaserver
  become: true

  tasks:
  - name: Disable global forwarders.
    ipadnsconfig:
      ipadmin_password: "{{ ipadmin_password }}"
      forward_policy: none
```

6. 保存这个文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disable-global-forwarders-copy.yml
```

其他资源

- 请参阅 [IdM 中的 DNS 转发策略](#)。
- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsconfig.md` 文件。
- 请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录中的更多 playbook 示例。

1.8. 使用 ANSIBLE PLAYBOOK 来确保 IDM DNS 中禁用了转发和反向查询区的同步

按照以下流程，使用 Ansible playbook 确保正向和反向查找区域在 IdM DNS 中未同步。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。
- 您的 IdM 环境包含一个集成的 DNS 服务器。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 `[ipaserver]` 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `disallow-reverse-sync.yml` Ansible playbook 文件。例如：

```
$ cp disallow-reverse-sync.yml disallow-reverse-sync-copy.yml
```

4. 打开 `disallow-reverse-sync-copy.yml` 文件进行编辑。
5. 通过在 `ipadnsconfig` 任务部分中设置以下变量来修改该文件：

- 将 `ipaadmin_password` 变量设置为 IdM 管理员密码。
 - 将 `allow_sync_ptr` 变量设置为 `no`。
- 这是当前示例修改的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。
7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disallow-reverse-sync-copy.yml
```

其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsconfig.md` 文件。
- 如需更多 playbook 示例，请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录。

第 2 章 管理 IDM 中的 DNS 区域

作为 Identity Management(IdM)管理员，您可以管理 IdM DNS 区域如何工作。本章论述了以下主题和步骤：

- [IdM 支持哪些 DNS 区类型](#)
 - [如何使用 IdM Web UI 添加主 IdM DNS 区域](#)
 - [如何使用 IdM CLI 添加主 IdM DNS 区域](#)
 - [如何使用 IdM Web UI 删除主 IdM DNS 区域](#)
 - [如何使用 IdM CLI 删除主 IdM DNS 区域](#)
- [您可以在 IdM 中配置哪些 DNS 属性](#)
 - [如何在 IdM Web UI 中配置这些属性](#)
 - [如何在 IdM CLI 中配置这些属性](#)
- [IdM 中的区传输如何工作](#)
 - [如何在 IdM Web UI 中允许区传输](#)
 - [如何在 IdM CLI 中允许区传输](#)

先决条件

- DNS 服务安装在 IdM 服务器中。有关如何安装带有集成 DNS 的 IdM 服务器的详情，请查看以下链接之一：
 - [安装 IdM 服务器：使用集成的 DNS，集成的 CA 作为 root CA](#)
 - [安装 IdM 服务器：具有集成的 DNS，具有外部 CA 作为根 CA](#)
 - [安装 IdM 服务器：使用集成的 DNS,没有 CA](#)

2.1. 支持的 DNS 区类型

身份管理 (IdM) 支持两种类型的 DNS 区域：*primary* 和 *forward* 区域。此处描述了这两种类型的区，包括 DNS 转发的示例场景。



注意

本指南对区域类型使用 BIND 术语，它与用于 Microsoft Windows DNS 的术语不同。BIND 服务器中的 Primary zones 与 Microsoft Windows DNS 中的 *forward lookup zones* 和 *reverse lookup zones* 作用相同。BIND 中的转发区与 Microsoft Windows DNS 中的 *条件转发转发器* 相同。

主 DNS 区域

主 DNS 区域包含权威 DNS 数据，并可以接受动态 DNS 更新。这个行为等同于标准 BIND 配置中的 **type master** 设置。您可以使用 **ipa dnszone-*** 命令管理主区。

在符合标准 DNS 规则的情况下，每个主区域必须包含 **start of authority (SOA)** and **nameserver (NS)** 记录。IdM 在创建 DNS 区域时自动生成这些记录，但您必须手动将 NS 记录复制到父区以创建正确的委托。

根据标准 BIND 行为，查询该服务器不是权威服务器将转发到其他 DNS 服务器的名称。这些 DNS 服务器（如转发器）可能或对查询没有权威。

例 2.1. DNS 转发示例

IdM 服务器包含 **test.example.** primary zone。此区域包含 **sub.test.example.** name 的 NS 委派记录。另外，**test.example.** 区域被配置为 **sub.test.example** 子区的 **192.0.2.254** 转发器 IP 地址。

查询名称 **nonexistent.test.example.** 的客户端会接收到 **NXDomain** 回答，且不会发生转发，因为 IdM 服务器对该名称具有权威。

另一方面，查询 **host1.sub.test.example.** 名称将转发到配置的 forwarder **192.0.2.254**，因为 IdM 服务器对这个名称没有权威。

转发 DNS 区域

从 IdM 的角度来看，转发 DNS 区域不包含任何权威数据。实际上，转发的"zone"通常仅包含两部分信息：

- 一个域名
- 与域关联的 DNS 服务器的 IP 地址

属于定义域的名称的所有查询都转发到指定的 IP 地址。这个行为等同于标准 BIND 配置中的 **type forward** 设置。您可以使用 **ipa dnsforwardzone-*** 命令管理转发区。

在 IdM-Active Directory(AD)信任上下文中转发 DNS 区域特别有用。如果 IdM DNS 服务器对 **idm.example.com** 区域有权威，并且 AD DNS 服务器对 **ad.example.com** 区域有权威，则 **ad.example.com** 是 **idm.example.com** 主区域的 DNS 转发区。这意味着，当来自一个 IdM 客户端查询 **somehost.ad.example.com** 的 IP 地址，查询将转发到 **ad.example.com** IdM DNS 转发区中指定的 AD 域控制器。

2.2. 在 IDM WEB UI 中添加主 DNS 区域

按照以下流程，使用身份管理(IdM) Web UI 添加主 DNS 区域。

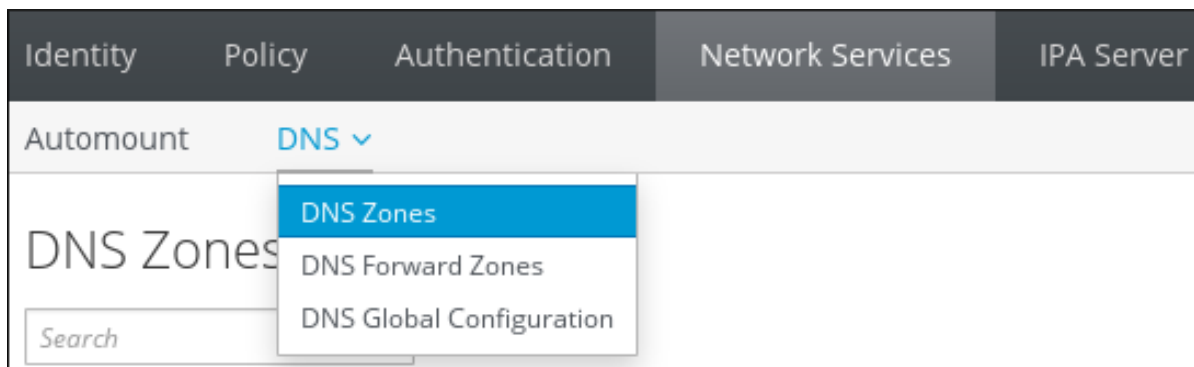
先决条件

- 以 IdM 管理员身份登录。

步骤

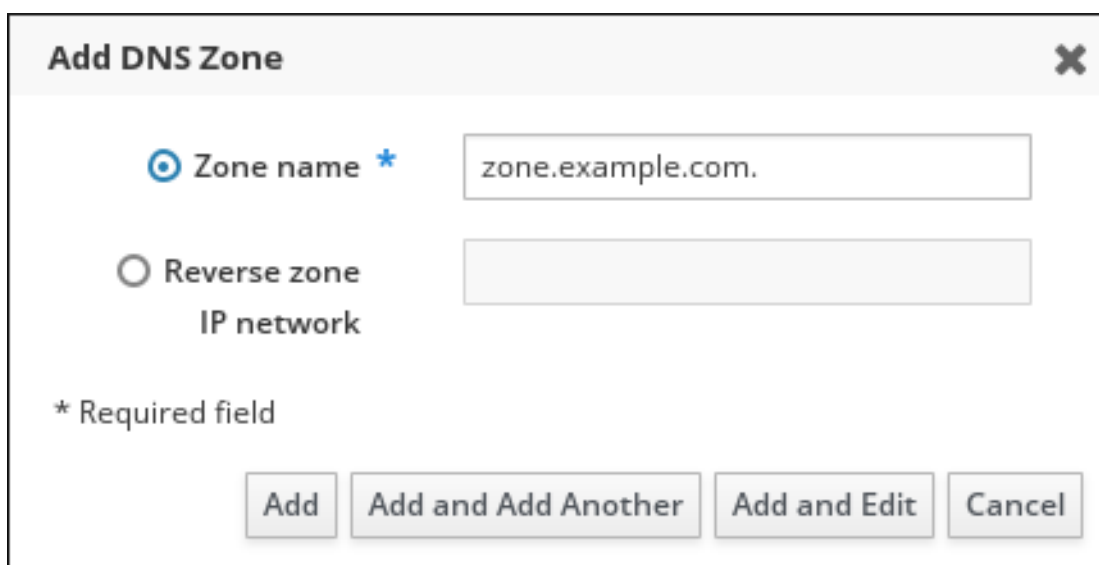
1. 在 IdM Web UI 中，点 **Network Services** → **DNS** → **DNS Zones**。

图 2.1. 管理 IdM DNS 主区



2. 点所有区域列表顶部的 **Add**。
3. 提供区域名称。

图 2.2. 输入一个新的 IdM 主区



4. 点击 **Add**。

2.3. 在 IDM CLI 中添加主 DNS 区

按照以下流程，使用身份管理(IdM)命令行界面(CLI)添加主 DNS 区域。

先决条件

- 以 IdM 管理员身份登录。

步骤

- **ipa dnszone-add** 命令在 DNS 域中添加新区。添加新区要求您指定新子域的名称。您可以使用命令直接传递子域名称：

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

如果您没有将名称传递给 **ipa dnszone-add**，该脚本会自动提示它。

其他资源

- 请参阅 `ipa dnszone-add --help`。

2.4. 在 IDM WEB UI 中删除主 DNS 区

按照以下流程，使用 IdM Web UI 从身份管理(IdM)中删除主 DNS 区域。

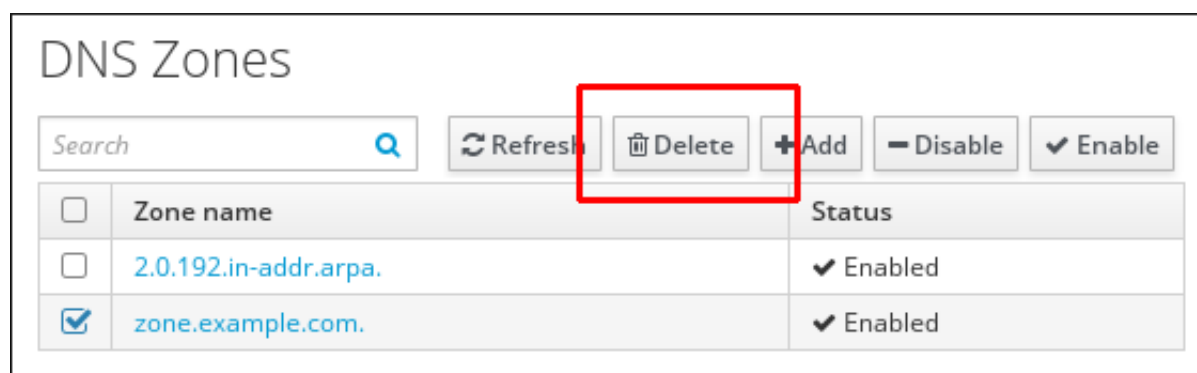
先决条件

- 以 IdM 管理员身份登录。

步骤

1. 在 IdM Web UI 中，点 **Network Services** → **DNS** → **DNS Zones**。
2. 按区域名称选择复选框，再点删除。

图 2.3. 删除主 DNS 区域



3. 在 **Remove DNS zone** 对话框窗口中，确认您要删除所选区域。

2.5. 在 IDM CLI 中删除主 DNS 区

按照以下流程，使用 IdM 命令行界面(CLI)从身份管理(IdM)中删除主 DNS 区域。

先决条件

- 以 IdM 管理员身份登录。

步骤

- 要删除主 DNS 区域，请输入 `ipa dnszone-del` 命令，后跟您要删除的区域的名称。例如：

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

2.6. DNS 配置优先级

您可以在以下层面上配置多个 DNS 配置选项：每个级别具有不同的优先级。

特定于区的配置

IdM 中定义的特定区的配置级别最高的优先级。您可以使用 `ipa dnszone-*` 和 `ipa dnsforwardzone-*` 命令来管理特定于区的配置。

每服务器配置

安装 IdM 服务器过程中，需要您定义每服务器转发器。您可以使用 `ipa dnsserver-*` 命令管理每服务器转发器。如果您不想在安装副本时设置每服务器转发器，您可以使用 `--no-forwarder` 选项。

全局 DNS 配置

如果没有定义特定于区的配置，IdM 将使用 LDAP 中存储的全局 DNS 配置。您可以使用 `ipa dnsconfig-*` 命令管理全局 DNS 配置。全局 DNS 配置中定义的设置适用于所有 IdM DNS 服务器。

在 `/etc/named.conf` 中配置

每个 IdM DNS 服务器中 `/etc/named.conf` 文件中定义的配置具有最低优先级。它特定于每台服务器，必须手动编辑。

`/etc/named.conf` 文件通常仅用于指定 DNS 转发到本地 DNS 缓存。其他选项可使用对上述区域和全局 DNS 配置的命令进行管理。

您可以同时在多个级别上配置 DNS 选项。在这种情况下，具有最高优先级的配置优先于在较低级别定义的配置。

其他资源

- [LDAP 中每服务配置](#) 中的 [配置的优先级顺序](#) 部分

2.7. 主 IDM DNS 区的配置属性

Identity Management(IdM)会创建一个带有特定默认配置的新区，如刷新周期、传输设置或缓存设置。在 [IdM DNS 区域属性](#) 中，您可以找到使用以下选项之一修改的默认区域配置的属性：

- 命令行界面(CLI)中的 `dnszone-mod` 命令。如需更多信息，请参阅 [在 IdM CLI 中编辑主 DNS 区的配置](#)。
- IdM Web UI。如需更多信息，请参阅 [在 IdM Web UI 中编辑主 DNS 区的配置](#)。
- 使用 `ipadnszone` 模块的 Ansible playbook。如需更多信息，请参阅 [在 IdM 中管理 DNS 区域](#)。

除了设置区的实际信息外，该设置还会定义 DNS 服务器如何处理 *start of authority* (SOA)记录条目，以及如何从 DNS 名称服务器更新其记录。

表 2.1. IdM DNS 区属性

属性	命令行选项	描述
权威名称服务器	<code>--name-server</code>	设置主 DNS 名称服务器的域名，也称为 SOA MNAME。 默认情况下，每个 IdM 服务器在 SOA MNAME 字段中公告自己。因此，使用 <code>--name-server</code> 的 LDAP 中存储的值将被忽略。
管理员电子邮件地址	<code>--admin-email</code>	设置用于区域管理员的电子邮件地址。默认为主机上的 root 帐户。
SOA 串行	<code>--serial</code>	在 SOA 记录中设置序列号。请注意，IdM 会自动设置版本号，用户不应该修改它。
SOA 刷新	<code>--refresh</code>	在从主 DNS 服务器请求更新前，设置二级 DNS 服务器要等待的时间间隔（以秒为单位）。

属性	命令行选项	描述
SOA 重试	--retry	设定重试失败的刷新操作前等待的时间（以秒为单位）。
SOA 过期	--expire	设定二级 DNS 服务器在操作尝试前尝试执行刷新更新的时间（以秒为单位）。
最少 SOA	--minimum	根据 RFC 2308 ，将负缓存的时间设置为 live(TTL)值（以秒为单位）。
SOA 时间到实时	--ttl	在 zone apex 的记录设置 TTL（以秒为单位）。例如，在区域 example.com 中，名称 example.com 下的所有记录(A、NS 或 SOA)都配置了，但其他域名（如 test.example.com ）不会受到影响。
默认时间变为实时	--default-ttl	将一个区中所有值的默认时间(TTL)设置为 live(TTL)值（以秒为单位）设置之前设置的独立 TTL 值。更改后，需要在所有 IdM DNS 服务器上重启 named-pkcs11 服务。
BIND 更新策略	--update-policy	设置 DNS 区域中客户端允许的权限。
动态更新	--dynamic-update=TRUE FALSE	为客户端启用对 DNS 记录的动态更新。 请注意，如果将其设置为 false，IdM 客户端机器将无法添加或更新其 IP 地址。
允许传输	--allow-transfer=string	指定允许传输给定区的 IP 地址或网络名称列表，用分号(;)分隔。 默认情况下禁用区域传送。默认的 --allow-transfer 值是 none 。
允许查询	--allow-query	指定允许发出 DNS 查询的 IP 地址或网络名称列表，用分号(;)分隔。
允许 PTR 同步	--allow-sync-ptr=1 0	设定区域的 A 或 AAAA 记录（转发记录）是否与 PTR（逆转）记录自动同步。
域转发	--forwarder=IP_address	指定一个只为 DNS 区域配置的转发器。这与 IdM 域中使用的任何全局转发分开。 要指定多个转发器，请多次使用选项。
转发策略	--forward-policy=none only first	指定转发策略。有关支持的策略的详情，请参考 IdM 中的 DNS 转发策略 。

2.8. 在 IDM WEB UI 中编辑主 DNS 区的配置

按照以下流程，使用 IdM Web UI 编辑主身份管理(IdM) DNS 的配置属性。

先决条件

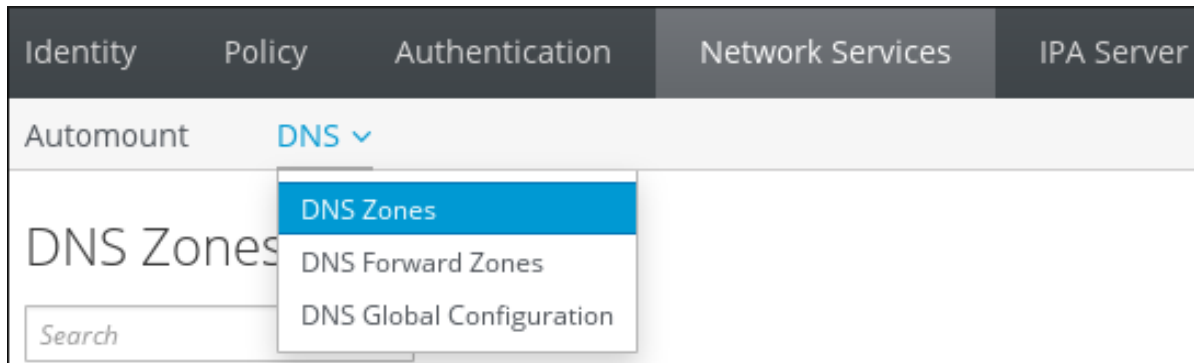
先决条件

- 以 IdM 管理员身份登录。

步骤

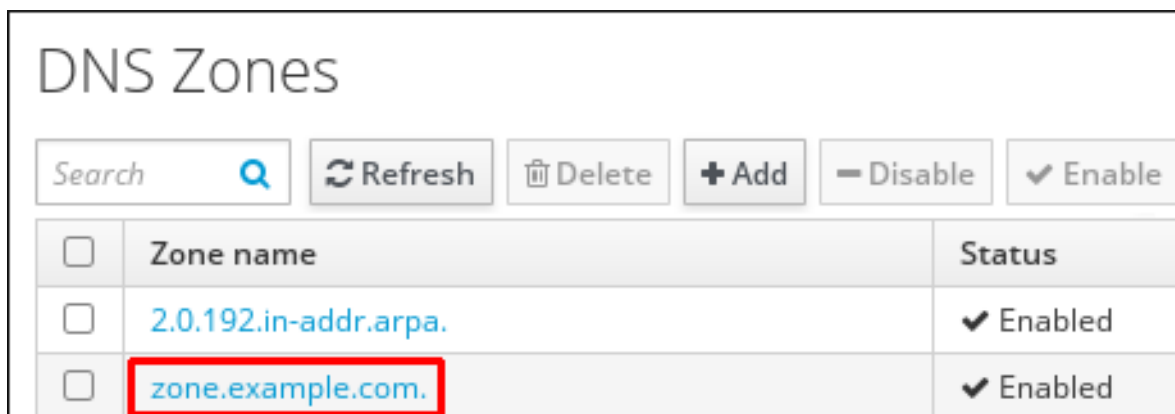
1. 在 IdM Web UI 中，点 **Network Services** → **DNS** → **DNS Zones**。

图 2.4. DNS 主区管理



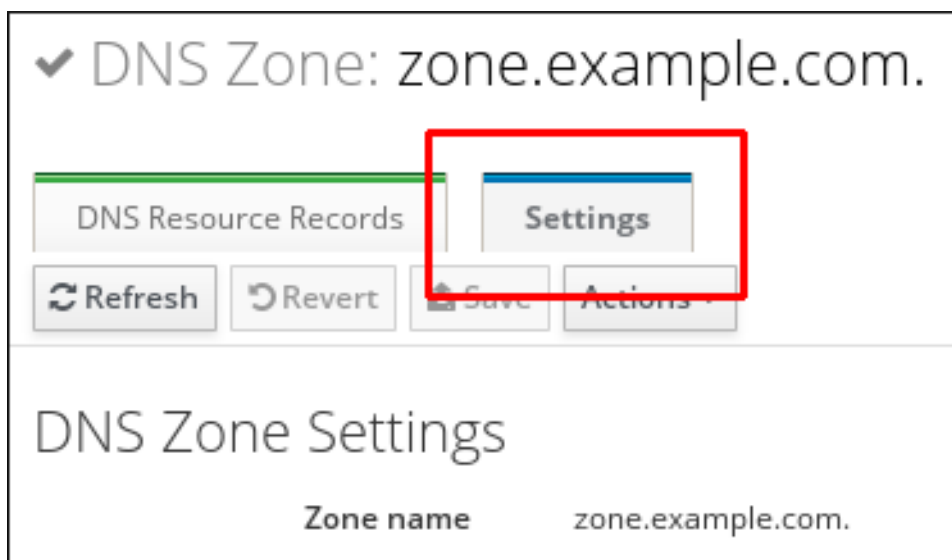
2. 在 **DNS Zones** 部分，点所有区列表中的区名称打开 DNS 区页面。

图 2.5. 编辑一个主区

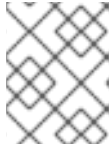


3. 点 **Settings**。

图 2.6. 主区编辑页面中的 Settings 标签页



4. 根据需要更改区配置。
有关可用设置的详情，请参考 [IdM DNS 区域属性](#)。
5. 点 **Save** 以确认新配置。



注意

如果您要将区的默认时间改为 live(TTL)，请在所有 IdM DNS 服务器中重启 **named-pkcs11** 服务以使更改生效。所有其他设置都会被立即自动激活。

2.9. 在 IDM CLI 中编辑主 DNS 区的配置

按照以下流程，使用身份管理(IdM)命令行界面(CLI)编辑主 DNS 区域的配置。

先决条件

- 以 IdM 管理员身份登录。

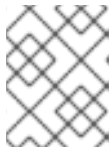
步骤

- 要修改现有主 DNS 区域，请使用 **ipa dnszone-mod** 命令。例如，要在重试失败的刷新操作前将时间设置为 1800 秒：

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

有关可用设置及其相应 CLI 选项的更多信息，请参阅 [IdM DNS 区域属性](#)。

如果特定设置没有您修改的 DNS 区条目中的值，**ipa dnszone-mod** 命令添加了值。如果设置没有值，该命令会使用指定的值覆盖当前的值。



注意

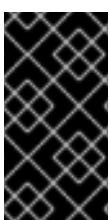
如果您要将区的默认时间改为 live(TTL)，请在所有 IdM DNS 服务器中重启 **named-pkcs11** 服务以使更改生效。所有其他设置都会被立即自动激活。

其他资源

- 请参阅 **ipa dnszone-mod --help**。

2.10. IDM 中的区传输

在有集成 DNS 的身份管理(IdM)部署中，您可以使用 *zone transfers* 将所有资源记录从一个名称服务器复制到另一个名称服务器。名称服务器维护其区域的权威数据。如果您更改了对 *zone A* DNS 区域具有权威的 DNS 服务器上的区域，您必须在位于 *zone A* 外的 IdM DNS 域中的其他名称服务器间分发更改。



重要

IdM 集成的 DNS 可同时由不同的服务器写入。IdM 区中的授权(SOA)序列号在单独的 IdM DNS 服务器中没有同步。因此，将位于 to-be-transferred 区域以外的 DNS 服务器配置为只使用 to-be-transferred 区域中的一个特定的 DNS 服务器。这可防止非同步 SOA 序列号导致的区域传送失败。

IdM 支持根据 [RFC 5936](#) (AXFR)和 [RFC 1995](#)(IXFR)标准进行区域传输。

其他资源

- 请参阅 [在 IdM Web UI 中启用区域传送](#)。
- 请参阅 [在 IdM CLI 中启用区域传送](#)。

2.11. 在 IDM WEB UI 中启用区传输

按照以下流程，使用 IdM Web UI 在身份管理(IdM)中启用区域传送。

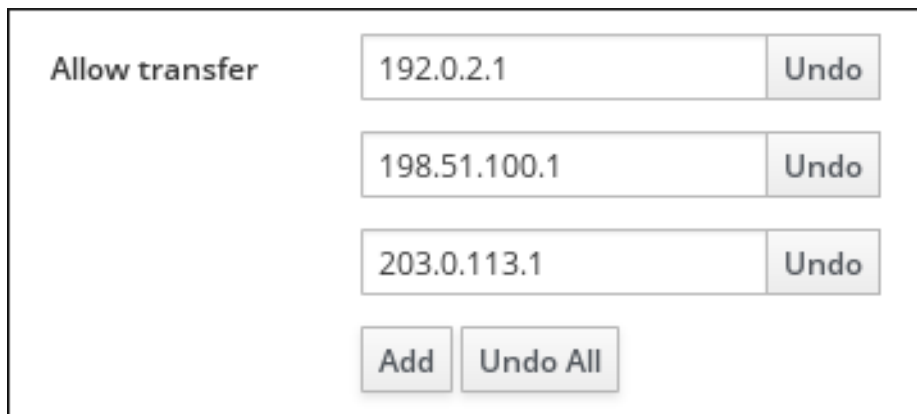
先决条件

- 以 IdM 管理员身份登录。

步骤

1. 在 IdM Web UI 中，点 **Network Services** → **DNS** → **DNS Zones**。
2. 点 **Settings**。
3. 在 **Allow transfer** 下，指定要传输区域记录的名称服务器。

图 2.7. 启用区传输



Allow transfer	IP Address	Action
	192.0.2.1	Undo
	198.51.100.1	Undo
	203.0.113.1	Undo
	Add Undo All	

4. 单击 DNS 区域页面顶部的 **Save** 以确认新配置。

2.12. 在 IDM CLI 中启用区传输

按照以下流程，使用 IdM 命令行界面(CLI)在身份管理(IdM)中启用区域传送。

先决条件

- 以 IdM 管理员身份登录。
- 有到二级 DNS 服务器的 root 访问权限。

步骤

- 要在 **BIND** 服务中启用区传输，输入 **ipa dnszone-mod** 命令，并指定使用 **--allow-transfer** 选项向该区域记录的待传输的名称服务器列表。例如：

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

验证步骤

1. SSH 到启用了区传输的其中一个 DNS 服务器：

```
$ ssh 192.0.2.1
```

2. 使用 **dig** 实用程序等工具传输 IdM DNS 区：

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

如果命令返回错误，您已成功为 *zone_name* 启用区传输。

2.13. 其他资源

- 请参阅 [使用 Ansible playbook 来管理 IdM DNS 区域](#)。

第 3 章 使用 ANSIBLE PLAYBOOK 管理 IDM DNS 区域

作为 Identity Management (IdM) 管理员，您可以使用 **ansible-freeipa** 软件包中的 **dnszone** 模块来管理 IdM DNS 区域的工作方式。

- [IdM 支持哪些 DNS 区类型](#)
- [您可以在 IdM 中配置哪些 DNS 属性](#)
- [如何使用 Ansible playbook 在 IdM DNS 中创建主区](#)
- [如何使用 Ansible playbook 确保使用多个变量的主 IdM DNS 区域](#)
- [在提供了 IP 地址时，如何使用 Ansible playbook 确保存在用于反向 DNS 查找的区域](#)

先决条件

- DNS 服务安装在 IdM 服务器中。有关如何使用 Red Hat Ansible Engine 安装带有集成 DNS 的 IdM 服务器的更多信息，请参阅 [使用 Ansible playbook 安装身份管理服务](#)。

3.1. 支持的 DNS 区类型

身份管理 (IdM) 支持两种类型的 DNS 区域：*primary* 和 *forward* 区域。此处描述了这两种类型的区，包括 DNS 转发的示例场景。



注意

本指南对区域类型使用 BIND 术语，它与用于 Microsoft Windows DNS 的术语不同。BIND 服务器中的 Primary zones 与 Microsoft Windows DNS 中的 *forward lookup zones* 和 *reverse lookup zones* 作用相同。BIND 中的转发区与 Microsoft Windows DNS 中的 *条件转发转发器* 相同。

主 DNS 区域

主 DNS 区域包含权威 DNS 数据，并可以接受动态 DNS 更新。这个行为等同于标准 BIND 配置中的 **type master** 设置。您可以使用 **ipa dnszone-*** 命令管理主区。

在符合标准 DNS 规则的情况下，每个主区域必须包含 **start of authority** (SOA) 和 **nameserver** (NS) 记录。IdM 在创建 DNS 区域时自动生成这些记录，但您必须手动将 NS 记录复制到父区以创建正确的委托。

根据标准 BIND 行为，查询该服务器不是权威服务器将转发到其他 DNS 服务器的名称。这些 DNS 服务器（如转发器）可能或对查询没有权威。

例 3.1. DNS 转发示例

IdM 服务器包含 **test.example.** primary zone。此区域包含 **sub.test.example.** name 的 NS 委派记录。另外，**test.example.** 区域被配置为 **sub.test.example** 子区的 **192.0.2.254** 转发器 IP 地址。

查询名称 **nonexistent.test.example.** 的客户端会接收到 **NXDomain** 回答，且不会发生转发，因为 IdM 服务器对该名称具有权威。

另一方面，查询 **host1.sub.test.example.** 名称将转发到配置的 forwarder **192.0.2.254**，因为 IdM 服务器对这个名称没有权威。

转发 DNS 区域

从 IdM 的角度来看，转发 DNS 区域不包含任何权威数据。实际上，转发的"zone"通常仅包含两部分信息：

- 一个域名
- 与域关联的 DNS 服务器的 IP 地址

属于定义域的名称的所有查询都转发到指定的 IP 地址。这个行为等同于标准 BIND 配置中的 **type forward** 设置。您可以使用 **ipa dnsforwardzone-*** 命令管理转发区。

在 IdM-Active Directory(AD)信任上下文中转发 DNS 区域特别有用。如果 IdM DNS 服务器对 **idm.example.com** 区域有权威，并且 AD DNS 服务器对 **ad.example.com** 区域有权威，则 **ad.example.com** 是 **idm.example.com** 主区域的 DNS 转发区。这意味着，当来自一个 IdM 客户端查询 **somehost.ad.example.com** 的 IP 地址，查询将转发到 **ad.example.com** IdM DNS 转发区中指定的 AD 域控制器。

3.2. 主 IDM DNS 区的配置属性

Identity Management(IdM)会创建一个带有特定默认配置的新区，如刷新周期、传输设置或缓存设置。在 [IdM DNS 区域属性](#) 中，您可以找到使用以下选项之一修改的默认区域配置的属性：

- 命令行界面(CLI)中的 **dnszone-mod** 命令。如需更多信息，请参阅 [在 IdM CLI 中编辑主 DNS 区的配置](#)。
- IdM Web UI。如需更多信息，请参阅 [在 IdM Web UI 中编辑主 DNS 区的配置](#)。
- 使用 **ipadnszone** 模块的 Ansible playbook。如需更多信息，请参阅 [在 IdM 中管理 DNS 区域](#)。

除了设置区的实际信息外，该设置还会定义 DNS 服务器如何处理 *start of authority* (SOA)记录条目，以及如何从 DNS 名称服务器更新其记录。

表 3.1. IdM DNS 区属性

属性	ansible-freeipa 变量	描述
权威名称服务器	name_server	设置主 DNS 名称服务器的域名，也称为 SOA MNAME。 默认情况下，每个 IdM 服务器在 SOA MNAME 字段中公告自己。因此，使用 --name-server 的 LDAP 中存储的值将被忽略。
管理员电子邮件地址	admin_email	设置用于区域管理员的电子邮件地址。默认为主机上的 root 帐户。
SOA 串行	serial	在 SOA 记录中设置序列号。请注意，IdM 会自动设置版本号，用户不应该修改它。
SOA 刷新	刷新	在从主 DNS 服务器请求更新前，设置二级 DNS 服务器要等待的时间间隔（以秒为单位）。
SOA 重试	retry	设定重试失败的刷新操作前等待的时间（以秒为单位）。

属性	ansible-freeipa 变量	描述
SOA 过期	expire	设定二级 DNS 服务器在操作尝试前尝试执行刷新更新的时间（以秒为单位）。
最少 SOA	minimum	根据 RFC 2308 ，将负缓存的时间设置为 live(TTL)值（以秒为单位）。
SOA 时间到实时	ttl	在 zone apex 的记录设置 TTL（以秒为单位）。例如，在区域 example.com 中，名称 example.com 下的所有记录(A、NS 或 SOA)都配置了，但其他域名（如 test.example.com ）不会受到影响。
默认时间变为实时	default_ttl	将一个区中所有值的默认时间(TTL)设置为 live(TTL)值（以秒为单位）设置之前设置的独立 TTL 值。更改后，需要在所有 IdM DNS 服务器上重启 named-pkcs11 服务。
BIND 更新策略	update_policy	设置 DNS 区域中客户端允许的权限。
动态更新	dynamic_update=TRUE FALSE	为客户端启用对 DNS 记录的动态更新。 请注意，如果将其设置为 false，IdM 客户端机器将无法添加或更新其 IP 地址。
允许传输	allow_transfer=string	指定允许传输给定区的 IP 地址或网络名称列表，用分号(;)分隔。 默认情况下禁用区域传送。默认的 allow_transfer 值是 none 。
允许查询	allow_query	指定允许发出 DNS 查询的 IP 地址或网络名称列表，用分号(;)分隔。
允许 PTR 同步	allow_sync_ptr=1 0	设定区域的 A 或 AAAA 记录（转发记录）是否与 PTR（逆转）记录自动同步。
域转发	forwarder=IP_addresses	指定一个只为 DNS 区域配置的转发器。这与 IdM 域中使用的任何全局转发分开。 要指定多个转发器，请多次使用选项。
转发策略	forward_policy=none only first	指定转发策略。有关支持的策略的详情，请参考 IdM 中的 DNS 转发策略 。

其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 **README-dnszone.md** 文件。

3.3. 使用 ANSIBLE 在 IDM DNS 中创建主区

按照以下流程，使用 Ansible playbook 确保主 DNS 区域存在。在以下流程使用的示例中，您确保 `zone.idm.example.com` DNS 区域存在。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnszone` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 `[ipaserver]` 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `dnszone-present.yml` Ansible playbook 文件。例如：

```
$ cp dnszone-present.yml dnszone-present-copy.yml
```

4. 打开 `dnszone-present-copy.yml` 文件进行编辑。
5. 通过在 `ipadnszone` task 部分中设置以下变量来修改该文件：

- 将 `ipadmin_password` 变量设置为 IdM 管理员密码。
- 将 `zone_name` 变量设置为 `zone.idm.example.com`。
这是当前示例修改的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。
7. 运行 playbook:

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

其他资源

- 请参阅[支持的 DNS 区域类型](#)。
- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnszone.md` 文件。
- 请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnszone` 目录中的 Ansible playbook 示例。

3.4. 使用 ANSIBLE PLAYBOOK 来确保 IDM 中存在带有多个变量的主 DNS 区域

按照以下流程，使用 Ansible playbook 确保主 DNS 区域存在。在下面的流程中使用的示例中，IdM 管理员可确保存在 `zone.idm.example.com` DNS 区域。Ansible playbook 配置区的多个参数。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 `Ansible 清单文件`。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipaadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnszone` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 `[ipaserver]` 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `dnszone-all-params.yml` Ansible playbook 文件。例如：

```
$ cp dnszone-all-params.yml dnszone-all-params-copy.yml
```

4. 打开 `dnszone-all-params-copy.yml` 文件进行编辑。

5. 通过在 `ipadnszone` task 部分中设置以下变量来修改该文件：

- 将 `ipaadmin_password` 变量设置为 IdM 管理员密码。
- 将 `zone_name` 变量设置为 `zone.idm.example.com`。
- 如果要允许同步转发和反向记录，请将 `allow_sync_ptr` 变量设置为 `true`，即 A 和 AAAA 记录与 PTR 记录同步。
- 将 `dynamic_update` 变量设置为 `true`，以启用 IdM 客户端机器添加或更新其 IP 地址。
- 将 `dnssec` 变量设置为 `true`，以允许在区域中进行内联 DNSSEC 签名。
- 将 `allow_transfer` 变量设置为区域中次要名称服务器的 IP 地址。
- 将 `allow_query` 变量设置为允许发出查询的 IP 地址或网络。
- 将 `forwarders` 变量设置为全局转发器的 IP 地址。
- 将 `serial` 变量设置为 SOA 记录序列号。
- 为区中的 DNS 定义 `refresh`, `retry`, `expire`, `minimum`, `ttl`, 和 `default_ttl` 值。
- 使用 `nsec3param_rec` 变量，为区域定义 NSEC3PARAM 记录。
- 将 `skip_overlap_check` 变量设置为 `true`，以便在它与现有区重叠时也强制进行 DNS 创建。
- 将 `skip_nameserver_check` 设置为 `true`，以便在名称服务器无法解析时也强制进行 DNS 区创建。

这是当前示例修改的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file dnszone-all-params-copy.yml
```

其他资源

- 请参阅 [支持的 DNS 区域类型](#)。
- 请参阅 [主 IdM DNS 区域的配置属性](#)。
- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnszone.md` 文件。
- 请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnszone` 目录中的 Ansible playbook 示例。

3.5. 使用 ANSIBLE PLAYBOOK 以确保在指定 IP 地址时存在用于反向 DNS 查找的区域

按照以下流程，使用 Ansible playbook 确保反向 DNS 区域存在。在下面的流程中使用的示例中，IdM 管理员确保使用 IdM 主机的 IP 地址和前缀长度，是否存在反向 DNS 查找区。

使用 `name_from_ip` 变量提供 DNS 服务器的 IP 地址的前缀长度可让您控制区名称。如果没有声明前缀长度，系统会查询区的 DNS 服务器，并根据 `name_from_ip` 值 `192.168.1.2`，查询会返回以下 DNS 区域：

- `1.168.192.in-addr.arpa.`
- `168.192.in-addr.arpa.`
- `192.in-addr.arpa.`

由于查询返回的区域可能不是您所期望的，`name_from_ip` 只能与 `state` 选项设置为 `present` 一起使用，以防止意外移除区域。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN) 的 **Ansible 清单文件**。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnszone` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnszone
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 `[ipaserver]` 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `dnszone-reverse-from-ip.yml` Ansible playbook 文件。例如：

```
$ cp dnszone-reverse-from-ip.yml dnszone-reverse-from-ip-copy.yml
```

4. 打开 `dnszone-reverse-from-ip-copy.yml` 文件进行编辑。

5. 通过在 `ipadnszone` task 部分中设置以下变量来修改该文件：

- 将 `ipadmin_password` 变量设置为 IdM 管理员密码。
- 将 `name_from_ip` 变量设置为 IdM 名称服务器的 IP，并提供其前缀长度。这是当前示例修改的 Ansible playbook 文件：

```
---
- name: Ensure dnszone present
  hosts: ipaserver
  become: true

  tasks:
  - name: Ensure zone for reverse DNS lookup is present.
    ipadnszone:
      ipadmin_password: "{{ ipadmin_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 }}"
```

playbook 创建一个区，用于从 192.168.1.2 IP 地址及其前缀长度 24 中反向 DNS 查找。接下来，playbook 显示生成的区域名称。

6. 保存这个文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file dnszone-  
reverse-from-ip-copy.yml
```

其他资源

- 请参阅[支持的 DNS 区域类型](#)。
- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnszone.md` 文件。
- 请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnszone` 目录中的 Ansible playbook 示例。

第 4 章 管理 IDM 中的 DNS 位置

要了解更多有关使用 IdM Web UI 和 IdM 命令行界面(CLI)管理身份管理(IdM) DNS 位置的信息，请参阅以下主题和流程：

- [基于 DNS 的服务发现](#)
- [DNS 位置的部署注意事项](#)
- [DNS 时间到实时\(TTL\)](#)
- [使用 IdM Web UI 创建 DNS 位置](#)
- [使用 IdM CLI 创建 DNS 位置](#)
- [使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置](#)
- [使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置](#)
- [配置 IdM 客户端在同一位置使用 IdM 服务器](#)

4.1. 基于 DNS 的服务发现

基于 DNS 的服务发现过程是一个进程，客户端使用 DNS 协议在提供特定服务（如 **LDAP** 或 **Kerberos**）的网络中查找服务器。一种典型的操作是允许客户端在最接近的网络基础架构内找到身份验证服务器，因为它们提供更高的吞吐量并降低网络延迟，从而降低整体成本。

服务发现的主要优点是：

- 不需要为客户端使用近似服务器名称显式配置。
- DNS 服务器用作策略的中央提供程序。使用相同 DNS 服务器的客户端可以访问与服务提供商及其首选顺序相同的策略。

在 Identity Management(IdM)域中，**LDAP**、**Kerberos** 和其他服务都存在 DNS 服务记录（SRV 记录）。例如，以下命令查询 DNS 服务器以获取在 IdM DNS 域中提供基于 TCP 的 **Kerberos** 服务的主机：

例 4.1. DNS 位置独立结果

```
$ 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.
```

输出包含以下信息：

- **0**（优先级）：目标主机的优先级。一个较低的值是首选的。
- **100**（权重）.为具有相同优先级的条目指定一个相对权重。如需更多信息，请参阅 [RFC 2782, 第 3 节](#)。
- **88**（端口号）：服务的端口号。
- 提供该服务的主机的规范名称。

在示例中，返回的两个主机名具有相同的优先级和权重。在这种情况下，客户端使用结果列表中的随机条目。

当客户端改为时，配置为查询在 DNS 位置中配置的 DNS 服务器，输出会有所不同。对于分配给位置的 IdM 服务器，返回定制的值。在以下示例中，客户端配置为查询位置 **germany** 中的 DNS 服务器：

例 4.2. 基于 DNS 位置的结果

```
$ 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 服务器自动返回一个 DNS 别名(CNAME)，指向首选本地服务器的 DNS 位置特定 SRV 记录。此 CNAME 记录显示在输出的第一行中。在示例中，主机 **idmserver-01.idm.example.com** 具有最低优先级值，因此首选。**idmserver-02.idm.example.com** 具有更高的优先级，因此仅在首选主机不可用时用作备份。

4.2. DNS 位置的部署注意事项

在使用集成的 DNS 时，身份管理(IdM)可以生成特定于位置的服务(SRV)记录。因为每个 IdM DNS 服务器都会生成特定于位置的 SRV 记录，所以您必须在每个 DNS 位置至少安装一个 IdM DNS 服务器。

客户端与 DNS 位置的关联性仅由客户端收到的 DNS 记录定义。因此，您可以将 IdM DNS 服务器与非 IdM DNS 消费者服务器合并，并在客户端从 IdM DNS 服务器中解析特定于位置的记录时进行 recursors。

在带有混合 IdM 和非 IdM DNS 服务的大部分部署中，DNS 递归器会使用往返时间指标自动选择最接近的 IdM DNS 服务器。通常，这样可确保使用非 IdM DNS 服务器的客户端为最接近的 DNS 位置获取记录，然后使用 IdM 服务器的最佳组。

4.3. DNS 时间到实时(TTL)

客户端可以将 DNS 资源记录缓存在区域配置中设定的时间。由于此缓存，客户端可能无法在生存时间(TTL)值过期前收到更改。Identity Management(IdM)中的默认 TTL 值是 **1 天**。

如果您的客户端计算机在站点间所需，您应该调整 IdM DNS 区的 TTL 值。将值设置为比客户端在站点间的 roam 需要的时间值低。这样可确保在客户端上缓存的 DNS 条目在重新连接到另一个站点前过期，因此查询 DNS 服务器以刷新特定于位置的 SRV 记录。

其他资源

- 请参阅[主 IdM DNS 区域的配置属性](#)。

4.4. 使用 IDM WEB UI 创建 DNS 位置

您可以使用 DNS 位置增加 Identity Management(IdM)客户端和服务器间的通信速度。按照以下流程，使用 IdM Web UI 创建 DNS 位置。

先决条件

- 您的 IdM 部署集成了 DNS。

- 您有在 IdM 中创建 DNS 位置的权限。例如，您以 IdM admin 身份登录。

步骤

1. 打开 **IPA Server** 选项卡。
2. 选择 **Topology** 子选项卡。
3. 点导航栏中的 **IPA Locations**。
4. 点位置列表顶部的 **Add**。
5. 填写位置名称。
6. 点 **Add** 按钮保存位置。
7. 可选：重复这些步骤来添加其他位置。

其他资源

- 请参阅 [使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置](#)。
- 请参阅 [使用 Ansible 来确保 IdM 位置存在](#)。

4.5. 使用 IDM CLI 创建 DNS 位置

您可以使用 DNS 位置增加 Identity Management(IdM)客户端和服务端间的通信速度。按照以下流程，使用 IdM 命令行界面(CLI)中的 **ipa location-add** 命令创建 DNS 位置。

先决条件

- 您的 IdM 部署集成了 DNS。
- 您有在 IdM 中创建 DNS 位置的权限。例如，您以 IdM admin 身份登录。

流程

1. 例如，要创建新位置 **germany**，请输入：

```
$ ipa location-add germany
-----
Added IPA location "germany"
-----
Location name: germany
```

2. 可选：重复步骤以添加其他位置。

其他资源

- 请参阅 [使用 IdM CLI 将 IdM 服务器分配给 DNS 位置](#)。
- 请参阅 [使用 Ansible 来确保 IdM 位置存在](#)。

4.6. 使用 IDM WEB UI 将 IDM 服务器分配给 DNS 位置

您可以使用 Identity Management(IdM)DNS 位置来增加 IdM 客户端和服务端间的通信速度。按照以下流程，使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置。

先决条件

- 您的 IdM 部署集成了 DNS。
- 以有权将服务器分配给 DNS 位置的用户身份登录，如 IdM admin 用户。
- 具有您要为其分配 DNS 位置的主机的 **root** 访问权限。
- 您已[创建了您要为其分配服务器的 IdM DNS 位置](#)。

流程

1. 打开 **IPA Server** 选项卡。
2. 选择 **Topology** 子选项卡。
3. 在导航中点 **IPA Servers**。
4. 点 IdM 服务器名称。
5. 选择 DNS 位置，并选择性地设置服务权重：

图 4.1. 将服务器分配给 DNS 位置



IPA Server: idmserver-01.idm.example.com

Refresh Revert Save

Server name	idmserver-01.idm.example.com.
Min domain level	0
Max domain level	1
Managed suffixes	domain ca
Location	germany
Service weight	100

6. 点 **Save**。
7. 在您在前面的步骤中分配的主机的命令行界面(CLI)中，重启 **named-pkcs11** 服务：

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

8. 可选：重复为其他 IdM 服务器分配 DNS 位置的步骤。

其他资源

- 请参阅 [配置 IdM 客户端以在同一位置上使用 IdM 服务器](#)。

4.7. 使用 IDM CLI 将 IDM 服务器分配给 DNS 位置

您可以使用 Identity Management(IdM)DNS 位置来增加 IdM 客户端和服务器间的通信速度。按照以下流程，使用 IdM 命令行界面(CLI)将 IdM 服务器分配给 DNS 位置。

先决条件

- 您的 IdM 部署集成了 DNS。
- 以有权将服务器分配给 DNS 位置的用户身份登录，如 IdM admin 用户。
- 具有您要为其分配 DNS 位置的主机的 **root** 访问权限。
- 您已创建了您要为其分配服务器的 IdM DNS 位置。

流程

1. 可选：列出所有配置的 DNS 位置：

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

2. 将服务器分配给 DNS 位置。例如，要将位置 **germany** 分配给服务器 **idmserver-01.idm.example.com**，请运行：

```
# 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. 在您之前步骤中分配的主机上的 **named-pkcs11** 服务重启 DNS 位置要：

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

4. 可选：重复为其他 IdM 服务器分配 DNS 位置的步骤。

其他资源

- 请参阅 [配置 IdM 客户端以在同一位置上使用 IdM 服务器](#)。

4.8. 配置 IDM 客户端在同一位置使用 IDM 服务器

身份管理(IdM)服务器被分配给 DNS 位置，如[使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置](#) 所述。现在，您可以将客户端配置为使用与 IdM 服务器位于同一个位置的 DNS 服务器：

- 如果 **DHCP** 服务器为客户端分配 DNS 服务器 IP 地址，请配置 **DHCP** 服务。有关在 **DHCP** 服务中分配 DNS 服务器的详情，请查看 **DHCP** 服务文档。
- 如果您的客户端没有从 **DHCP** 服务器接收 DNS 服务器 IP 地址，请在客户端的网络配置中手动设置 IP。有关在 Red Hat Enterprise Linux 中配置网络的详情，请参考 *Red Hat Enterprise Linux 网络指南* 中的[置网络连接设置](#)部分。



注意

如果您将客户端配置为使用分配给不同位置的 DNS 服务器，客户端会联系两个位置中的 IdM 服务器。

例 4.3. 根据客户端的位置的不同名称服务器条目

以下示例显示了 `/etc/resolv.conf` 文件中的不同位置客户端的不同名称服务器条目：

Prague 的客户端：

```
nameserver 10.10.0.1
nameserver 10.10.0.2
```

Paris 的客户端：

```
nameserver 10.50.0.1
nameserver 10.50.0.3
```

Oslo 的客户端：

```
nameserver 10.30.0.1
```

Berlin 的客户端：

```
nameserver 10.30.0.1
```

如果每个 DNS 服务器被分配给 IdM 中的位置，客户端将使用其位置中的 IdM 服务器。

4.9. 其他资源

- 请参阅在 [IdM 中使用 Ansible 来管理 DNS 位置](#)。

第 5 章 使用 ANSIBLE 管理 IDM 中的 DNS 位置

作为 Identity Management(IdM)管理员，您可以使用 **ansible-freeipa** 软件包中的 **location** 模块来管理 IdM DNS 位置。

- [基于 DNS 的服务发现](#)
- [DNS 位置的部署注意事项](#)
- [DNS 时间到实时\(TTL\)](#)
- [使用 Ansible 确保存在 IdM 位置](#)
- [使用 Ansible 确保不存在 IdM 位置](#)

5.1. 基于 DNS 的服务发现

基于 DNS 的服务发现过程是一个进程，客户端使用 DNS 协议在提供特定服务（如 **LDAP** 或 **Kerberos**）的网络中查找服务器。一种典型的操作是允许客户端在最接近的网络基础架构内找到身份验证服务器，因为它们提供更高的吞吐量并降低网络延迟，从而降低整体成本。

服务发现的主要优点是：

- 不需要为客户端使用近似服务器名称显式配置。
- DNS 服务器用作策略的中央提供程序。使用相同 DNS 服务器的客户端可以访问与服务提供商及其首选顺序相同的策略。

在 Identity Management(IdM)域中，**LDAP**、**Kerberos** 和其他服务都存在 DNS 服务记录（SRV 记录）。例如，以下命令查询 DNS 服务器以获取在 IdM DNS 域中提供基于 TCP 的 **Kerberos** 服务的主机：

例 5.1. DNS 位置独立结果

```
$ 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.
```

输出包含以下信息：

- **0**（优先级）：目标主机的优先级。一个较低的值是首选的。
- **100**（权重）：为具有相同优先级的条目指定一个相对权重。如需更多信息，请参阅 [RFC 2782, 第 3 节](#)。
- **88**（端口号）：服务的端口号。
- 提供该服务的主机的规范名称。

在示例中，返回的两个主机名具有相同的优先级和权重。在这种情况下，客户端使用结果列表中的随机条目。

当客户端改为时，配置为查询在 DNS 位置中配置的 DNS 服务器，输出会有所不同。对于分配给位置的 IdM 服务器，返回定制的值。在以下示例中，客户端配置为查询位置 **germany** 中的 DNS 服务器：

例 5.2. 基于 DNS 位置的结果

```
$ 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 服务器自动返回一个 DNS 别名(CNAME)，指向首选本地服务器的 DNS 位置特定 SRV 记录。此 CNAME 记录显示在输出的第一行中。在示例中，主机 `idmserver-01.idm.example.com` 具有最低优先级值，因此首选。`idmserver-02.idm.example.com` 具有更高的优先级，因此仅在首选主机不可用时用作备份。

5.2. DNS 位置的部署注意事项

在使用集成的 DNS 时，身份管理(IdM)可以生成特定于位置的服务(SRV)记录。因为每个 IdM DNS 服务器都会生成特定于位置的 SRV 记录，所以您必须在每个 DNS 位置至少安装一个 IdM DNS 服务器。

客户端与 DNS 位置的关联性仅由客户端收到的 DNS 记录定义。因此，您可以将 IdM DNS 服务器与非 IdM DNS 消费者服务器合并，并在客户端从 IdM DNS 服务器中解析特定于位置的记录时进行 recursors。

在带有混合 IdM 和非 IdM DNS 服务的大部分部署中，DNS 递归器会使用往返时间指标自动选择最接近的 IdM DNS 服务器。通常，这样可确保使用非 IdM DNS 服务器的客户端为最接近的 DNS 位置获取记录，然后使用 IdM 服务器的最佳组。

5.3. DNS 时间到实时(TTL)

客户端可以将 DNS 资源记录缓存在区域配置中设定的时间。由于此缓存，客户端可能无法在生存时间(TTL)值过期前收到更改。Identity Management(IdM)中的默认 TTL 值是 **1 天**。

如果您的客户端计算机在站点间所需，您应该调整 IdM DNS 区的 TTL 值。将值设置为比客户端在站点间的 roam 需要的时间值低。这样可确保在客户端上缓存的 DNS 条目在重新连接到另一个站点前过期，因此查询 DNS 服务器以刷新特定于位置的 SRV 记录。

其他资源

- 请参阅[主 IdM DNS 区域的配置属性](#)。

5.4. 使用 ANSIBLE 确保存在 IDM 位置

作为 Identity Management(IdM)的系统管理员，您可以将 IdM DNS 位置配置为允许客户端查找最接近的网络基础架构中的身份验证服务器。

以下流程描述了如何使用 Ansible playbook 来确保 IdM 中存在 DNS 位置。这个示例描述了如何确保 IdM 中存在 `germany` DNS 位置。因此，您可以将特定的 IdM 服务器分配给这个位置，以便本地 IdM 客户端使用它来减少服务器响应时间。

先决条件

- 您知道 IdM 管理员密码。
- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。

- 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
- 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
- 示例假定 `secret.yml` Ansible vault 存储了 `ipaadmin_password`。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您了解 [DNS 位置的部署注意事项](#)。

步骤

1. 进入 `~/MyPlaybooks/` 目录：

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

2. 生成位于 `/usr/share/doc/ansible-freeipa/playbooks/location/` 目录中的 `location-present.yml` 文件的副本：

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

3. 打开 `location-present-copy.yml` Ansible playbook 文件进行编辑。
4. 通过在 `ipalocation` task 部分中设置以下变量来修改该文件：
 - 调整任务的 `name`，使其与您的用例对应。
 - 将 `ipaadmin_password` 变量设置为 IdM 管理员的密码。
 - 将 `name` 变量设置为位置的名称。

这是当前示例修改的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。
6. 运行 Ansible playbook。指定 playbook 文件、存储保护 `secret.yml` 文件的密码，以及清单文件：

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

其他资源

- 请参阅 [使用 IdM Web UI 将 IdM 服务器分配给 DNS 位置](#)，或 [使用 IdM CLI 将 IdM 服务器分配给 DNS 位置](#)。

5.5. 使用 ANSIBLE 确保不存在 IDM 位置

作为 Identity Management(IdM)的系统管理员，您可以将 IdM DNS 位置配置为允许客户端查找最接近的网络基础架构中的身份验证服务器。

以下流程描述了如何使用 Ansible playbook 来确保 IdM 中没有 DNS 位置。这个示例描述了如何确保 IdM 中没有 **germany** DNS 位置。因此，您无法为这个位置分配特定的 IdM 服务器，本地 IdM 客户端无法使用它们。

先决条件

- 您知道 IdM 管理员密码。
- 没有 IdM 服务器被分配给 **germany** DNS 位置。
- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 示例假定您已 [创建并配置了](#) `~/MyPlaybooks/` 目录，来作为存储示例 playbook 副本的中心位置。

步骤

1. 进入 `~/MyPlaybooks/` 目录：

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

2. 在 `/usr/share/doc/ansible-freeipa/playbooks/location/` 目录中复制 `location-absent.yml` 文件：

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

3. 打开 `location-absent-copy.yml` Ansible playbook 文件进行编辑。
4. 通过在 `ipalocation` task 部分中设置以下变量来修改该文件：
 - 调整任务的 `name`，使其与您的用例对应。
 - 将 `ipadmin_password` 变量设置为 IdM 管理员的密码。

- 将 **name** 变量设置为 DNS 位置的名称。
- 确保 **state** 变量设置为 **absent**。

这是当前示例修改的 Ansible playbook 文件：

```
---
- name: location absent example
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Ensure that the "germany" location is absent
    ipalocation:
      ipadmin_password: "{{ ipadmin_password }}"
      name: germany
      state: absent
```

5. 保存这个文件。
6. 运行 Ansible playbook。指定 playbook 文件、存储保护 `secret.yml` 文件的密码，以及清单文件：

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

5.6. 其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-location.md` 文件。
- 请参阅 `/usr/share/doc/ansible-freeipa/playbooks/location` 目录中的 Ansible playbook 示例。

第 6 章 在 IDM 中管理 DNS 转发

按照以下流程，在身份管理(IdM) Web UI、IdM CLI 中以及使用 Ansible 来配置 DNS 全局转发器和 DNS 转发区域：

- IdM DNS 服务器的两个角色
- IdM 中的 DNS 转发策略
- 在 IdM Web UI 中添加全局转发器
- 在 CLI 中添加全局转发器
- 在 IdM Web UI 中添加 DNS 转发区域
- 在 CLI 中添加 DNS 转发区域
- 使用 Ansible 在 IdM 中建立 DNS 全局转发器
- 使用 Ansible 确保 IdM 中存在 DNS 全局转发器
- 使用 Ansible 确保 IdM 中没有 DNS 全局转发器
- 使用 Ansible 确保 DNS 全局转发器在 IdM 中被禁用
- 使用 Ansible 确保 IdM 中存在 DNS 转发区域
- 使用 Ansible 确保 DNS 转发区域 在 IdM 中有多个转发器
- 使用 Ansible 确保 IdM 中 DNS Forward 区域被禁用
- 使用 Ansible 确保 IdM 中没有 DNS 转发区域

6.1. IDM DNS 服务器的两个角色

DNS 转发会影响 DNS 服务如何响应 DNS 查询。默认情况下，集成了 IdM 的 Berkeley Internet Name Domain (BIND) 作为一个 *authoritative* 和一个 *recursive* DNS 服务器：

权威 DNS 服务器

当 DNS 客户端查询属于 IdM 服务器权威的 DNS 区域的名称时，BIND 会回复配置区中包含的数据。权威数据始终优先于任何其他数据。

递归 DNS 服务器

当 DNS 客户端查询 IdM 服务器不是权威的名称时，BIND 会尝试使用其他 DNS 服务器解析查询。如果没有定义转发器，BIND 会询问互联网上的 root 服务器，并使用递归解析算法来响应 DNS 查询。

在某些情况下，不需要让 BIND 直接联系其他 DNS 服务器，并根据 Internet 上可用的数据执行递归。您可以将 BIND 配置为使用另外一个 DNS 服务器（转发器）来解析查询。

当您将 BIND 配置为使用转发器时，在 IdM 服务器和转发器之间转发查询和答案，IdM 服务器充当非授权数据的 DNS 缓存。

6.2. IDM 中的 DNS 转发策略

IdM 支持 **first** 和 **only** 标准 BIND 转发策略，以及 **none** 特定于 IdM 的转发策略。

Forward first(默认)

IdM BIND 服务将 DNS 查询转发到配置的转发器。如果因为服务器错误或超时导致查询失败，BIND 会使用互联网上的服务器返回到递归解析。**forward first** 策略是默认策略，它适用于优化 DNS 流量。

Forward only

IdM BIND 服务将 DNS 查询转发到配置的转发器。如果因为服务器错误或超时而查询失败，BIND 会将错误返回到客户端。对于带有 split DNS 配置的环境，建议使用 **forward only** 策略。

None (禁用转发)

DNS 查询不会通过 **none** 转发策略转发。禁用转发只作为全局转发配置的特定区覆盖很有用。这个选项等同于在 BIND 配置中指定空转发器列表。



注意

您不能使用转发将 IdM 中的数据与来自其他 DNS 服务器的数据合并。您只能为 IdM DNS 中主区的特定子区转发查询。

默认情况下，如果查询的 DNS 名称属于 IdM 服务器有权威的区域，则 BIND 服务不会将查询转发到另一台服务器。在这种情况下，如果在 IdM 数据库中找到查询的 DNS 名称，则返回 **NXDOMAIN** 回答。未使用转发功能。

例 6.1. 使用情况示例

IdM 服务器对 **test.example** 具有权威性。DNS 区域。BIND 被配置为把查询转发到带有 **192.0.2.254** IP 地址的 DNS 服务器。

当客户发送对 **nonexistent.test.example.** 的查询 DNS 名称，BIND 检测到 IdM 服务器对 **test.example.** 区域具有权威，且不会将查询转发到 **192.0.2.254.** 服务器。因此，DNS 客户端接收 **NXDomain** 错误消息，告知用户查询域不存在。

6.3. 在 IDM WEB UI 中添加全局转发器

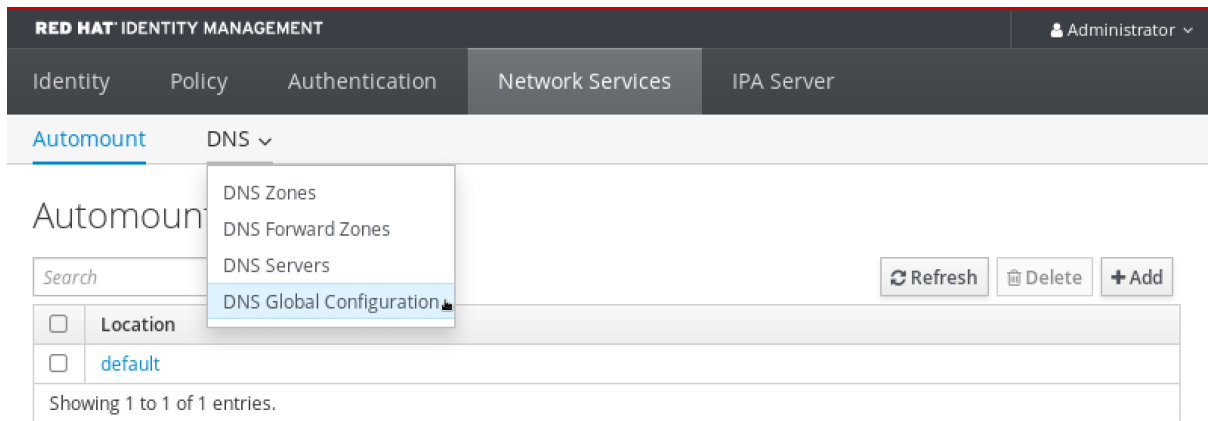
按照以下流程在身份管理(IdM) Web UI 中添加全局 DNS 转发器。

先决条件

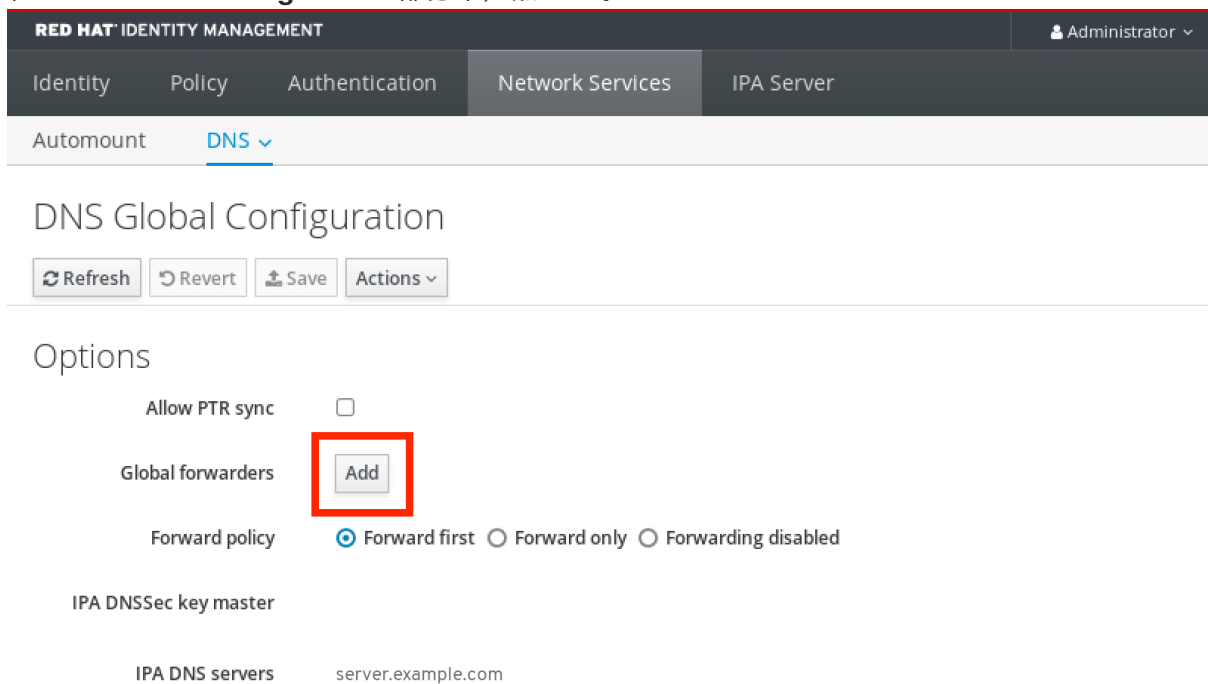
- 以 IdM 管理员身份登录 IdM WebUI。
- 您知道 DNS 服务器的 Internet 协议(IP)地址，以将查询转发到。

步骤

1. 在 IdM Web UI 中，选择 **Network Services → DNS Global Configuration → DNS**。



2. 在 **DNS Global Configuration** 部分中，点 **Add**。



3. 指定接收转发 DNS 查询的 DNS 服务器的 IP 地址。

RED HAT IDENTITY MANAGEMENT Administrator

Identity Policy Authentication Network Services IPA Server

Automount DNS

DNS Global Configuration

Refresh Revert Save Actions

Options

Allow PTR sync

Global forwarders Undo

Add Undo All

Forward policy Forward first Forward only Forwarding disabled

IPA DNSSec key master

IPA DNS servers server.example.com

4. 选择转发策略。

RED HAT IDENTITY MANAGEMENT Administrator

Identity Policy Authentication Network Services IPA Server

Automount DNS

DNS Global Configuration

Refresh Revert Save Actions

Options

Allow PTR sync

Global forwarders Undo

Add Undo All

Forward policy Forward first Forward only Forwarding disabled

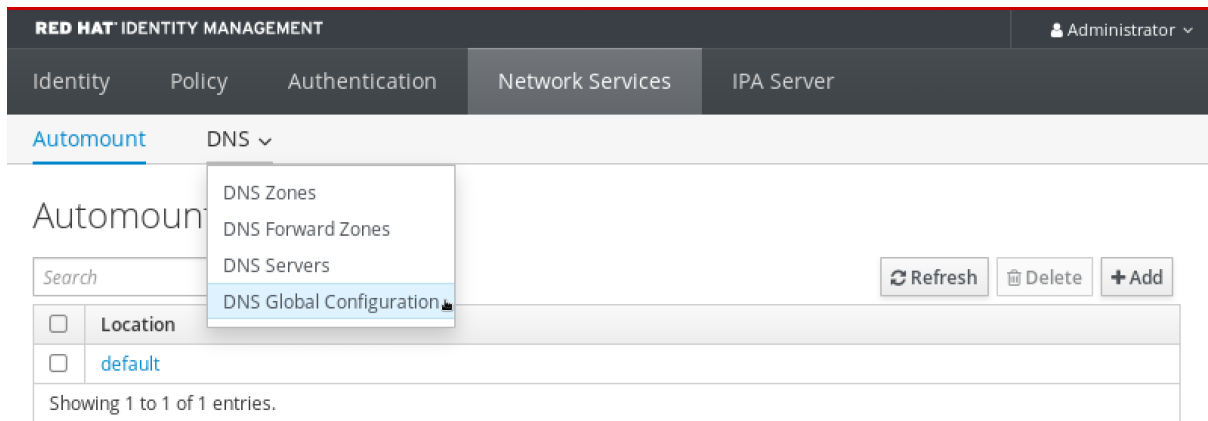
IPA DNSSec key master

IPA DNS servers server.example.com

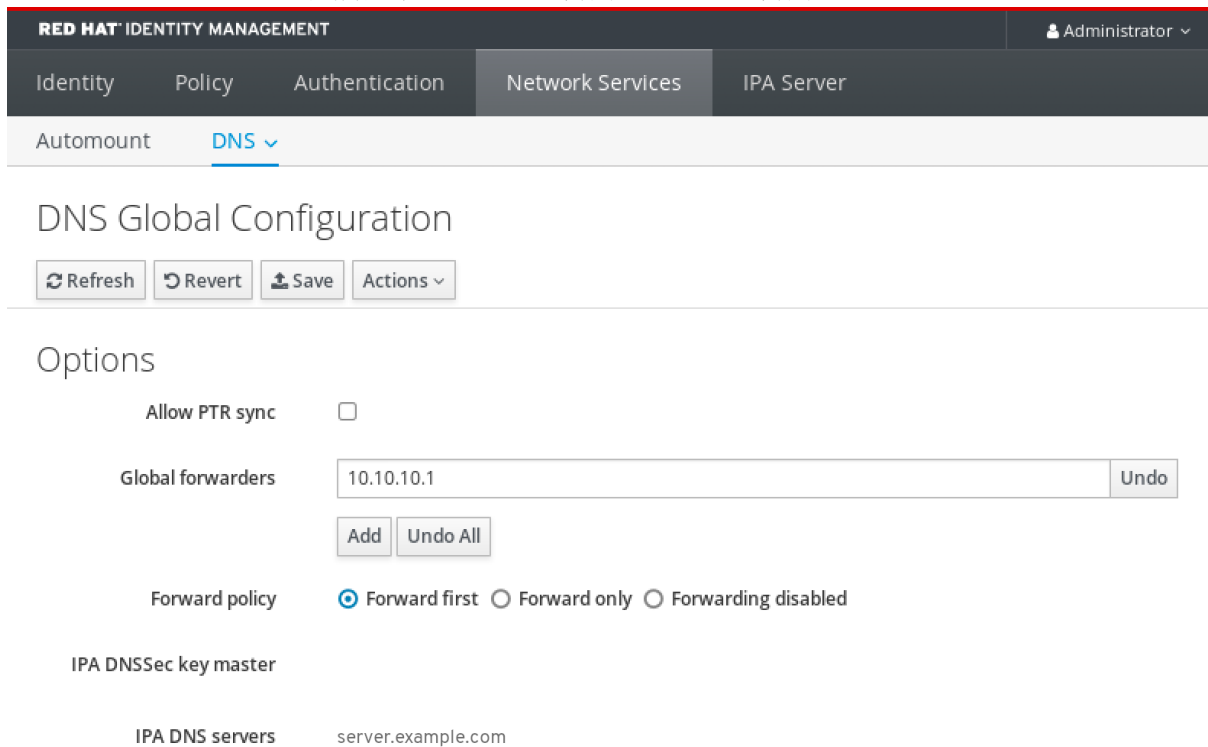
5. 点窗口顶部的 **Save**。

验证步骤

1. 选择 **Network Services** → **DNS Global Configuration** → **DNS**。



2. 验证 IdM Web UI 中存在并启用带有您指定的转发策略的全局转发器。



6.4. 在 CLI 中添加全局转发器

按照以下流程，使用命令行界面(CLI)添加全局 DNS 转发器。

先决条件

- 以 IdM 管理员身份登录。
- 您知道 DNS 服务器的 Internet 协议(IP)地址，以将查询转发到。

步骤

- 使用 `ipa dnsconfig-mod` 命令添加新的全局转发器。使用 `--forwarder` 选项指定 DNS 转发器的 IP 地址。

```
[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
```

验证步骤

- 使用 **dnsconfig-show** 命令显示全局转发器。

```
[user@server ~]$ ipa dnsconfig-show
Global forwarders: 10.10.0.1
IPA DNS servers: server.example.com
```

6.5. 在 IDM WEB UI 中添加 DNS 转发区域

按照以下流程在身份管理(IdM) Web UI 中添加 DNS 转发区域。



重要

除非绝对需要，否则不要使用转发区域。转发区不是标准解决方案，使用它们可能会导致意外和有问题的行为。如果必须使用 forward 区域，请限制其使用来覆盖全局转发配置。

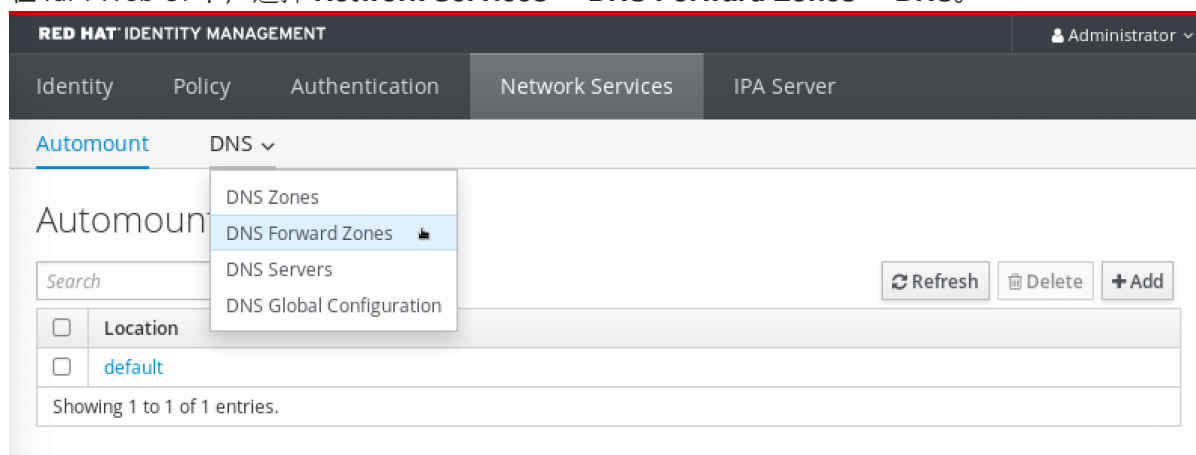
在创建新的 DNS 区域时，红帽建议使用名称服务器(NS)记录和避免转发区域，始终使用标准 DNS 委托。在大多数情况下，使用全局转发器就足够了，不需要转发区域。

先决条件

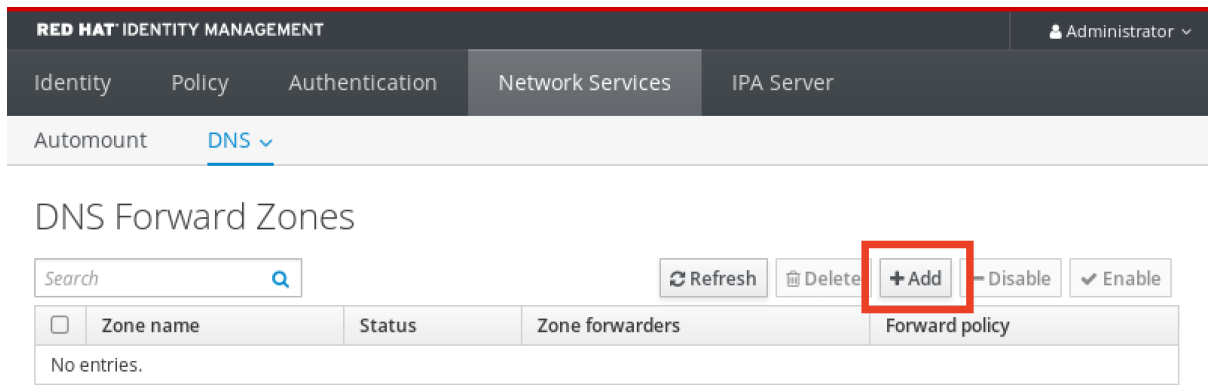
- 以 IdM 管理员身份登录 IdM WebUI。
- 您知道 DNS 服务器的 Internet 协议(IP)地址，以将查询转发到。

步骤

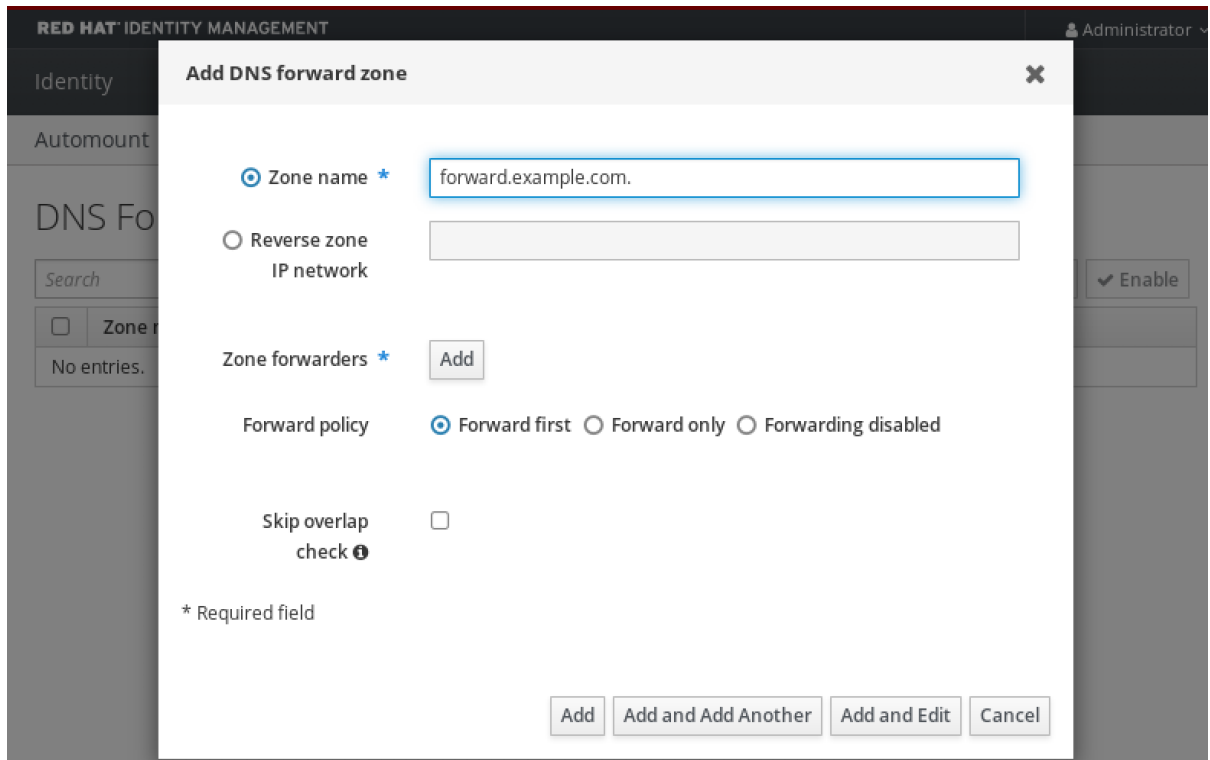
1. 在 IdM Web UI 中，选择 **Network Services** → **DNS Forward Zones** → **DNS**。



2. 在 **DNS Forward Zones** 部分，点 **Add**。



3. 在 **Add DNS forward zone** 窗口中指定转发区名称。



4. 点 **Add** 按钮并指定 DNS 服务器的 IP 地址来接收转发请求。您可以为每个转发区指定多个转发器。

Add DNS forward zone

Zone name * forward.example.com.

Reverse zone
IP network

Zone forwarders * 10.10.0.14 Undo

Add

Forward policy Forward first Forward only Forwarding disabled

Skip overlap check

* Required field

Add Add and Add Another Add and Edit Cancel

5. 选择 转发策略。

Add DNS forward zone

Zone name * forward.example.com

Reverse zone
IP network

Zone forwarders * 10.10.0.14 Undo

Add

Forward policy Forward first Forward only Forwarding disabled

Skip overlap check

* Required field

Add Add and Add Another Add and Edit Cancel

6. 单窗口底部的 **Add** 以添加新转发区域。

验证步骤

1. 在 IdM Web UI 中，选择 **Network Services** → **DNS Forward Zones** → **DNS**。

The screenshot shows the Red Hat Identity Management web interface. The top navigation bar includes 'Identity', 'Policy', 'Authentication', 'Network Services', and 'IPA Server'. The 'Automount' section is active, and a dropdown menu is open, highlighting 'DNS Forward Zones'. Below the menu, there is a search bar, a table with one entry 'default' under the 'Location' column, and buttons for 'Refresh', 'Delete', and '+Add'. The text 'Showing 1 to 1 of 1 entries.' is visible at the bottom of the table.

2. 验证您创建的转发区（带有您指定的转发器和转发策略）是否在 IdM Web UI 中存在并启用。

The screenshot shows the 'DNS Forward Zones' configuration page in the Red Hat Identity Management web interface. The top navigation bar is the same as in the previous screenshot. The 'DNS' section is active. Below the navigation, there is a search bar, buttons for 'Refresh', 'Delete', '+Add', '- Disable', and '✓ Enable'. A table displays the configuration for a single zone:

<input type="checkbox"/>	Zone name	Status	Zone forwarders	Forward policy
<input type="checkbox"/>	forward.example.com.	✓ Enabled	10.10.0.14	first

Showing 1 to 1 of 1 entries.

6.6. 在 CLI 中添加 DNS 转发区域

按照以下流程使用命令行界面(CLI)添加 DNS 转发区。



重要

除非绝对需要，否则不要使用转发区域。转发区不是标准解决方案，使用它们可能会导致意外和有问题的行为。如果必须使用 forward 区域，请限制其使用来覆盖全局转发配置。

在创建新的 DNS 区域时，红帽建议使用名称服务器(NS)记录 and 避免转发区域，始终使用标准 DNS 委托。在大多数情况下，使用全局转发器就足够了，不需要转发区域。

先决条件

- 以 IdM 管理员身份登录。
- 您知道 DNS 服务器的 Internet 协议(IP)地址，以将查询转发到。

步骤

- 使用 **dnsforwardzone-add** 命令添加新转发区。如果转发策略不是 **none**，使用 **--forwarder** 选项指定至少一个转发器，并使用 **--forward-policy** 选项指定转发策略。

```
[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
```

验证步骤

- 使用 **dnsforwardzone-show** 命令显示您刚才创建的 DNS 转发区。

```
[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. 使用 ANSIBLE 在 IDM 中建立 DNS 全局转发器

按照以下流程，使用 Ansible playbook 在 IdM 中建立 DNS Global Forwarder。

在以下示例中，IdM 管理员会创建一个 DNS 全局转发程序到带有 IPv4 地址为 **8.8.6.6**，IPv6 地址为 **2001:4860:4860::8800** 的端口 **53** DNS 服务器。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 `[ipaserver]` 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `set-configuration.yml` Ansible playbook 文件。例如：

```
$ cp set-configuration.yml establish-global-forwarder.yml
```

4. 打开 **establish-global-forwarder.yml** 文件进行编辑。
5. 通过设置以下变量来调整文件：
 - a. 将 **playbook** 的 **name** 变量更改为 **Playbook**，以在 **IdM DNS** 中建立全局转发器。
 - b. 在 **tasks** 部分中，将任务的 **name** 更改为 **Create a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800**。
 - c. 在 **ipadnsconfig** 部分的 **forwarders** 部分：
 - i. 将第一个 **ip_address** 值更改为全局转发器的 IPv4 地址：**8.8.6.6**。
 - ii. 将第二个 **ip_address** 值更改为全局转发器的 IPv6 地址：**2001:4860:4860::8800**。
 - iii. 验证 **port** 值被设置为 **53**。
 - d. 将 **forward_policy** 更改为 **first**。
对于当前示例为修改过的 Ansible playbook 文件：

```

---
- 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. 保存该文件。
7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file establish-global-forwarder.yml
```

其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsconfig.md` 文件。

6.8. 使用 ANSIBLE 确保 IDM 中存在 DNS 全局转发器

按照以下流程，使用 Ansible playbook 确保 DNS 全局转发器在 IdM 中存在。在下例中，IdM 管理员确保在 DNS 服务器中存在一个到 IPv4 地址为 **7.7.9.9**，IP v6 地址为 **2001:db8::1:0**，端口 **53** 的 DNS global forwarder。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 **Ansible 清单文件**。
 - 示例假定 **secret.yml** Ansible vault 存储了 **ipadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 **[ipaserver]** 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 **server.idm.example.com**，请输入：

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

3. 复制 **forwarders-absent.yml** Ansible playbook 文件。例如：

```
$ cp forwarders-absent.yml ensure-presence-of-a-global-forwarder.yml
```

4. 打开 **ensure-presence-of-a-global-forwarder.yml** 文件进行编辑。

5. 通过设置以下变量来调整文件：

- a. 将 playbook 的 **name** 变量更改为 **Playbook**，以确保 IdM DNS 中存在全局转发器。
 - b. 在 **tasks** 部分中，将任务 **name** 更改为 **Ensure the presence of a DNS global forwarder to 7.7.9.9 and 2001:db8::1:0 on port 53**。
 - c. 在 **ipadnsconfig** 部分的 **forwarders** 部分：
 - i. 将第一个 **ip_address** 值更改为全局转发器的 IPv4 地址：**7.7.9.9**。
 - ii. 将第二个 **ip_address** 值更改为全局转发器的 IPv6 地址：**2001:db8::1:0**。
 - iii. 验证 **port** 值被设置为 **53**。
 - d. 将 **state** 该为 **present**。
- 对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 保存该文件。

7. 运行 playbook :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-of-a-global-forwarder.yml
```

其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsconfig.md` 文件。

6.9. 使用 ANSIBLE 确保 IDM 中没有 DNS 全局转发器

按照以下流程，使用 Ansible playbook 确保 DNS 全局转发器在 IdM 中不存在。在以下示例流程中，IdM 管理员确保在端口 **53** 上没有互联网协议(IP)v4 地址为 **8.8.6.6** 和 IP v6 地址为 **2001:4860:4860::8800** 的 DNS 全局转发器。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipaadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 **[ipaserver]** 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 **server.idm.example.com**，请输入：

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

3. 复制 **forwarders-absent.yml** Ansible playbook 文件。例如：

```
$ cp forwarders-absent.yml ensure-absence-of-a-global-forwarder.yml
```

4. 打开 **ensure-absence-of-a-global-forwarder.yml** 文件进行编辑。

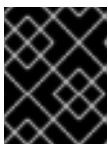
5. 通过设置以下变量来调整文件：

- a. 将 playbook 的 **name** 变量更改为 **Playbook**，以确保 IdM DNS 中没有全局转发器。
- b. 在 **tasks** 部分，将任务的 **name** 改为 **Ensure the absence of a DNS global forwarder to 8.8.6.6 and 2001:4860:4860::8800 on port 53**。
- c. 在 **ipadnsconfig** 部分的 **forwarders** 部分：
 - i. 将第一个 **ip_address** 值更改为全局转发器的 IPv4 地址：**8.8.6.6**。
 - ii. 将第二个 **ip_address** 值更改为全局转发器的 IPv6 地址：**2001:4860:4860::8800**。
 - iii. 验证 **port** 值被设置为 **53**。
- d. 将 **action** 变量设置为 **member**。
- e. 验证 **state** 被设置为 **absent**。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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

如果您仅在 playbook 中使用 **state: absent** 选项，而不使用 **action: member**，则 playbook 会失败。

6. 保存这个文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-of-a-global-forwarder.yml
```

其他资源

- `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsconfig.md` 文件
- `ipadnsconfig ansible-freeipa` 模块中的 `action: member` 选项

6.10. 使用 ANSIBLE 确保 DNS 全局转发器在 IDM 中被禁用

按照以下流程，使用 Ansible playbook 确保 DNS Global Forwarders 在 IdM 中被禁用。在以下示例中，IdM 管理员确保将全局转发器的转发策略设置为 `none`，这将有效禁用全局转发器。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 `[ipaserver]` 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 验证 `disable-global-forwarders.yml` Ansible playbook 文件的内容，该文件已被配置为禁用所有 DNS 全局转发器。例如：

```
$ 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.
  ipadsnconfig:
    forward_policy: none
```

4. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file disable-global-forwarders.yml
```

其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsconfig.md` 文件。

6.11. 使用 ANSIBLE 确保 IDM 中存在 DNS 转发区域

按照以下流程，使用 Ansible playbook 确保 DNS 转发区域在 IdM 中存在。在以下示例中，IdM 管理员确保将 `example.com` 的 DNS 转发区存在到带有 Internet 协议(IP)地址的 DNS 服务器，地址为 `8.8.8.8`。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 `Ansible 清单文件`。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 `[ipaserver]` 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `forwarders-absent.yml` Ansible playbook 文件。例如：

```
$ cp forwarders-absent.yml ensure-presence-forwardzone.yml
```

4. 打开 **ensure-presence-forwardzone.yml** 文件进行编辑。
5. 通过设置以下变量来调整文件：
 - a. 将 **playbook** 的 **name** 变量更改为 **Playbook**，以确保 IdM DNS 中存在 **dnsforwardzone**。
 - b. 在 **tasks** 部分中，将任务的 **name** 更改为 **Ensure presence of a dnsforwardzone for example.com to 8.8.8.8**。
 - c. 在 **tasks** 部分中，将 **ipadnsconfig** 标题更改为 **ipadnsforwardzone**。
 - d. 在 **ipadnsforwardzone** 部分：
 - i. 添加 **ipaadmin_password** 变量，并将其设置为您的 IdM 管理员密码。
 - ii. 添加 **name** 变量，并将它设置为 **example.com**。
 - iii. 在 **forwarders** 部分：
 - A. 删除 **ip_address** 和 **port** 行。
 - B. 通过在横线后指定该 DNS 服务器的 IP 地址来接收转发的请求：

```
- 8.8.8.8
```

- iv. 添加 **forwardpolicy** 变量，并将它设为 **first**。
- v. 添加 **skip_overlap_check** 变量，并将其设置为 **true**。
- vi. 将 **state** 变量更改为 **present**。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。
7. 运行 **playbook**:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-forwardzone.yml
```

其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsforwardzone.md` 文件。

6.12. 使用 ANSIBLE 确保 DNS 转发区域 在 IDM 中有多个转发器

按照以下流程，使用 Ansible playbook 确保 IdM 中的 DNS Forward Zone 有多个转发器。在以下示例中，IdM 管理员确保 `example.com` 的 DNS 转发区转发到 `8.8.8.8` 和 `4.4.4.4`。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 `[ipaserver]` 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `forwarders-absent.yml` Ansible playbook 文件。例如：

```
$ cp forwarders-absent.yml ensure-presence-multiple-forwarders.yml
```

4. 打开 `ensure-presence-multiple-forwarders.yml` 文件进行编辑。

5. 通过设置以下变量来调整文件：

- a. 将 playbook 的 `name` 变量更改为 `Playbook`，以确保 IdM DNS 中的 `dnsforwardzone` 中存在多个转发器。

- b. 在 **tasks** 项中，把任务的 **name** 改为 **Ensure presence of 8.8.8.8 and 4.4.4.4 forwarders in dnsforwardzone for example.com**。
- c. 在 **tasks** 部分中，将 **ipadnsconfig** 标题更改为 **ipadnsforwardzone**。
- d. 在 **ipadnsforwardzone** 部分：
 - i. 添加 **ipaadmin_password** 变量，并将其设置为您的 IdM 管理员密码。
 - ii. 添加 **name** 变量，并将它设置为 **example.com**。
 - iii. 在 **forwarders** 部分：
 - A. 删除 **ip_address** 和 **port** 行。
 - B. 添加您要保证的 DNS 服务器的 IP 地址存在，在前面添加一个短划线：

```
- 8.8.8.8
- 4.4.4.4
```

- iv. 将 **state** 变量更改为 **present**。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。
7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-
multiple-forwarders.yml
```

其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsforwardzone.md` 文件。

6.13. 使用 ANSIBLE 确保 IDM 中 DNS FORWARD 区域被禁用

按照以下流程，使用 Ansible playbook 确保 DNS Forward Zone 在 IdM 中被禁用。在以下示例中，IdM 管理员确保 **example.com** 的 DNS 转发区已被禁用。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 **Ansible 清单文件**。
 - 示例假定 **secret.yml** Ansible vault 存储了 **ipadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 **[ipaserver]** 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 **server.idm.example.com**，请输入：

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

3. 复制 **forwarders-absent.yml** Ansible playbook 文件。例如：

```
$ cp forwarders-absent.yml ensure-disabled-forwardzone.yml
```

4. 打开 **ensure-disabled-forwardzone.yml** 文件进行编辑。

5. 通过设置以下变量来调整文件：

- a. 将 playbook 的 **name** 变量更改为 **Playbook**，以确保 IdM DNS 中禁用了 **dnsforwardzone**。
- b. 在 **tasks** 项中，将任务的 **name** 改为 **Ensure a dnsforwardzone for example.com is disabled**。
- c. 在 **tasks** 部分中，将 **ipadnsconfig** 标题更改为 **ipadnsforwardzone**。
- d. 在 **ipadnsforwardzone** 部分：
 - i. 添加 **ipadmin_password** 变量，并将其设置为您的 IdM 管理员密码。
 - ii. 添加 **name** 变量，并将它设置为 **example.com**。
 - iii. 删除整个 **forwarders** 部分。

iv. 将 **state** 变量更改为 **disabled**。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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:
      ipadmin_password: "{{ ipadmin_password }}"
      name: example.com
      state: disabled
```

6. 保存这个文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-disabled-forwardzone.yml
```

其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsforwardzone.md` 文件。

6.14. 使用 ANSIBLE 确保 IDM 中没有 DNS 转发区域

按照以下流程，使用 Ansible playbook 确保 DNS 转发区域在 IdM 中不存在。在以下示例中，IdM 管理员确保 **example.com** 没有 DNS 转发区。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 [ansible-freeipa](#) 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsconfig` 目录：

■

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsconfig
```

2. 打开清单文件，并确保 **[ipaserver]** 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 **server.idm.example.com**，请输入：

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

3. 复制 **forwarders-absent.yml** Ansible playbook 文件。例如：

```
$ cp forwarders-absent.yml ensure-absence-forwardzone.yml
```

4. 打开 **ensure-absence-forwardzone.yml** 文件进行编辑。

5. 通过设置以下变量来调整文件：

- a. 将 playbook 的 **name** 变量更改为 **Playbook**，以确保 IdM DNS 中不存在 **dnsforwardzone**。
- b. 在 **tasks** 项中，把任务的 **name** 改为 **Ensure the absence of a dnsforwardzone for example.com**。
- c. 在 **tasks** 部分中，将 **ipadnsconfig** 标题更改为 **ipadnsforwardzone**。
- d. 在 **ipadnsforwardzone** 部分：
 - i. 添加 **ipadmin_password** 变量，并将其设置为您的 IdM 管理员密码。
 - ii. 添加 **name** 变量，并将它设置为 **example.com**。
 - iii. 删除整个 **forwarders** 部分。
 - iv. 将 **state** 变量保留为 **absent**。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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:
      ipadmin_password: "{{ ipadmin_password }}"
      name: example.com
      state: absent
```

6. 保存这个文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-absence-forwardzone.yml
```

其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsforwardzone.md` 文件。

第 7 章 管理 IDM 中的 DNS 记录

本章论述了如何在 Identity Management(IdM)中管理 DNS 记录。作为 IdM 管理员，您可以在 IdM 中添加、修改和删除 DNS 记录。本章包含以下部分：

- [IdM 中的 DNS 记录](#)
- [从 IdM Web UI 添加 DNS 资源记录](#)
- [从 IdM CLI 添加 DNS 资源记录](#)
- [通用 ipa dnsrecord-add 选项](#)
- [删除 IdM Web UI 中的 DNS 记录](#)
- [删除 IdM Web UI 中的整个 DNS 记录](#)
- [删除 IdM CLI 中的 DNS 记录](#)

先决条件

- 您的 IdM 部署包含一个集成的 DNS 服务器。有关如何使用集成 DNS 安装 IdM 的详情，请查看以下链接之一：
 - [安装 IdM 服务器：使用集成的 DNS，集成的 CA 作为 root CA。](#)
 - [安装 IdM 服务器：使用集成的 DNS，外部 CA 作为 root CA。](#)

7.1. IDM 中的 DNS 记录

身份管理(IdM)支持许多不同的 DNS 记录类型。以下是最频繁使用的四个项：

一个

这是主机名和 IPv4 地址的基本映射。A 记录的记录名称是一个主机名，如 **www**。A 记录的 **IP 地址** 值是一个 IPv4 地址，如 **192.0.2.1**。

有关 A 记录的更多信息，请参阅 [RFC 1035](#)。

AAAA

这是主机名和 IPv6 地址的基本映射。AAAA 记录的记录名称是一个主机名，如 **www**。IP 地址 值是一个 IPv6 地址，如 **2001:DB8::1111**。

有关 AAAA 记录的更多信息，请参阅 [RFC 3596](#)。

SRV

服务 (SRV) 资源记录 将服务名称映射到提供该特定服务的服务器的 DNS 名称。例如，此记录类型可以将类似 LDAP 目录的服务映射到管理它的服务器。

SRV 记录的记录名称格式为 **_service._protocol**，如 **_ldap._tcp**。SRV 记录的配置选项包括优先级、权重、端口号和目标服务的主机名。

有关 SRV 记录的更多信息，请参阅 [RFC 2782](#)。

PTR

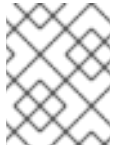
指针记录(PTR)添加反向 DNS 记录，它将 IP 地址映射到域名。



注意

IPv4 地址的所有反向 DNS 查找都使用在 **in-addr.arpa** 域中定义的反向条目。反向地址采用人类可读形式，与常规 IP 地址完全相反，它附加了 **in-addr.arpa** 域。例如，对于本网络地址 **192.0.2.0/24**，反向区域为 **2.0.192.in-addr.arpa**。

PTR 的记录名称必须是 [RFC 1035](#) 中指定的标准格式，它在 [RFC 2317](#) 和 [RFC 3596](#) 中扩展。主机名值必须是您要为其创建记录的主机的规范主机名。



注意

还可以为 IPv6 地址配置反向区域，即 **.ip6.arpa** 域中的区域。有关 IPv6 反向区的更多信息，请参阅 [RFC 3596](#)。

添加 DNS 资源记录时，请注意很多记录需要不同的数据。例如，CNAME 记录需要一个主机名，而 A 记录则需要一个 IP 地址。在 IdM Web UI 中，用于添加新记录的表单字段会自动更新，以反映当前所选记录类型所需的数据。

7.2. 在 IDM WEB UI 中添加 DNS 资源记录

按照以下流程在身份管理(IdM) Web UI 中添加 DNS 资源记录。

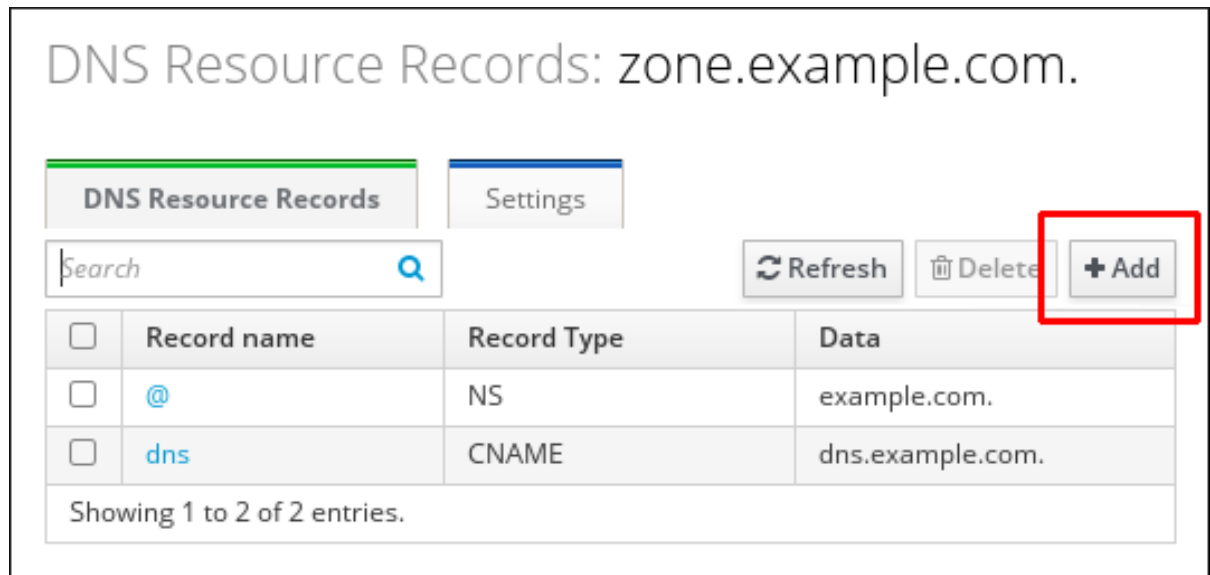
先决条件

- 您要添加 DNS 记录的 DNS 区域存在并由 IdM 管理。有关在 IdM DNS 中创建 DNS 区域的详情，请参考 [IdM 中的管理 DNS 区域](#)。
- 以 IdM 管理员身份登录。

步骤

1. 在 IdM Web UI 中，点 **Network Services** → **DNS** → **DNS Zones**。
2. 点您要添加 DNS 记录的 DNS 区域。
3. 在 **DNS Resource Records** 部分，点 **Add** 添加新记录。

图 7.1. 添加新 DNS 资源记录



4. 根据需要选择要创建的记录类型，并填写其他字段。

图 7.2. 定义新的 DNS 资源记录

Add DNS Resource Record X

Record name * dns

Record Type CNAME

Hostname * dns.example.com.

* Required field

Add Add and Add Another Add and Edit Cancel

5. 点 **Add** 以确认新记录。

7.3. 从 IDM CLI 添加 DNS 资源记录

按照以下流程，通过命令行界面(CLI)添加任何类型的 DNS 资源记录。

先决条件

- 已存在您要添加 DNS 记录的 DNS 区域。有关在 IdM DNS 中创建 DNS 区域的详情，请参考 [IdM 中的管理 DNS 区域](#)。
- 以 IdM 管理员身份登录。

步骤

1. 要添加 DNS 资源记录，请使用 `ipa dnsrecord-add` 命令。该命令遵循这个语法：

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

在以上命令中：

- `zone_name` 是将记录添加到的 DNS 区域的名称。
- `record_name` 是新的 DNS 资源记录的标识符。

例如，要将 `host1` 的 A 类型 DNS 记录添加到 `idm.example.com` 区域，请输入：

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

7.4. 常见 IPA DNSRECORD-* 选项

在身份管理(IdM)中添加、修改和删除最常见的 DNS 资源记录类型时，您可以使用以下选项：

- A (IPv4)
- AAAA (IPv6)
- SRV
- PTR

在 **Bash** 中，您可以通过列出大括号内的逗号分隔列表中的值来定义多个条目，如 `--option={val1,val2,val3}`。

表 7.1. 常规记录选项

选项	描述
<code>--ttl=number</code>	将记录的时间设置为实时。
<code>--structured</code>	解析原始 DNS 记录，并以结构化格式返回它们。

表 7.2. "A" 记录选项

选项	描述	示例
<code>--a-rec=ARECORD</code>	传递单个 A 记录或 A 记录列表。	<code>ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123</code>
	可以创建具有给定 IP 地址的通配符 A 记录。	<code>ipa dnsrecord-add idm.example.com "*" --a-rec=192.168.122.123^[a]</code>

选项	描述	示例
--a-ip-address=string	为记录指定 IP 地址。在创建记录时，指定 A 记录值的选项为 --a-rec 。但是，在修改 A 记录时，使用 --a-rec 选项指定 A 记录的当前值。使用 --a-ip-address 选项设置新值。	ipa dnsrecord-mod idm.example.com --a-rec 192.168.122.123 --a-ip-address 192.168.122.124
[a] 这个示例创建通配符 A 记录，其 IP 地址为 192.0.2.123。		

表 7.3. "AAAA"记录选项

选项	描述	示例
--aaaa-rec=AAAARECORD	通过单个 AAAA(IPv6)记录或 AAAA 记录列表。	ipa dnsrecord-add idm.example.com www --aaaa-rec 2001:db8::1231:5675
--aaaa-ip-address=string	为记录指定 IPv6 地址。在创建记录时，指定 A 记录值的选项为 --aaaa-rec 。但是，在修改 A 记录时， --aaaa-rec 选项用于指定 A 记录的当前值。使用 --a-ip-address 选项设置新值。	ipa dnsrecord-mod idm.example.com --aaaa-rec 2001:db8::1231:5675 --aaaa-ip-address 2001:db8::1231:5676

表 7.4. "PTR"记录选项

选项	描述	示例
--ptr-rec=PTRRECORD	传递单个 PTR 记录或 PTR 记录列表。当添加反向 DNS 记录时，与 ipa dnsrecord-add 命令使用的区名称会被相反，与添加其他 DNS 记录的用法不同。通常，主机 IP 地址是给定网络中的 IP 地址的最后一个八进制数。右侧的第一个示例为 server4.idm.example.com 添加 IPv4 地址为 192.168.122.4 的 PTR 记录。第二个示例为 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa 。添加了一个反向 DNS 条目。主机 server2.example.com 的 IPv6 反向区域，IP 地址为 2001:DB8::1111 。	ipa dnsrecord-add 122.168.192.in-addr.arpa 4 --ptr-rec server4.idm.example.com. \$ ipa dnsrecord-add 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.1.1.1.0.0.0.0.0.0.0.0.0 --ptr-rec server2.idm.example.com.
--ptr-hostname=string	为记录提供主机名。	

表 7.5. "SRV"记录选项

选项	描述	示例
--srv-rec=SRVRECORD	通过单个 SRV 记录或 SRV 记录列表。在右侧的示例中， <code>_ldap_tcp</code> 定义 SRV 记录的服务类型和连接协议。 --srv-rec 选项定义优先级、权重、端口和目标值。示例中的权重值为 51 和 49（总和为 100），它们代表使用特定记录的可能性（以百分比表示）。	<pre># 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."</pre>
--srv-priority=number	设置记录的优先级。服务类型可以有多个 SRV 记录。优先级(0 - 65535)设置记录排名；编号越低，优先级越高。服务必须首先使用优先级最高的记录。	<pre># ipa dnsrecord-mod server.idm.example.com _ldap_tcp --srv-rec="1 49 389 server2.idm.example.com." --srv-priority=0</pre>
--srv-weight=number	设置记录的权重。这有助于决定具有相同优先级的 SRV 记录顺序。设定的权重应添加最多 100，代表使用特定记录的概率（以百分比表示）。	<pre># ipa dnsrecord-mod server.idm.example.com _ldap_tcp --srv-rec="0 49 389 server2.idm.example.com." --srv-weight=60</pre>
--srv-port=number	为目标主机上的服务指定端口。	<pre># ipa dnsrecord-mod server.idm.example.com _ldap_tcp --srv-rec="0 60 389 server2.idm.example.com." --srv-port=636</pre>
--srv-target=string	指定目标主机的域名。如果域中没有服务，则这可以是一个句点(.)。	

其他资源

- 运行 `ipa dnsrecord-add --help`。

7.5. 删除 IDM WEB UI 中的 DNS 记录

按照以下流程，使用 IdM Web UI 删除身份管理(IdM)中的 DNS 记录。

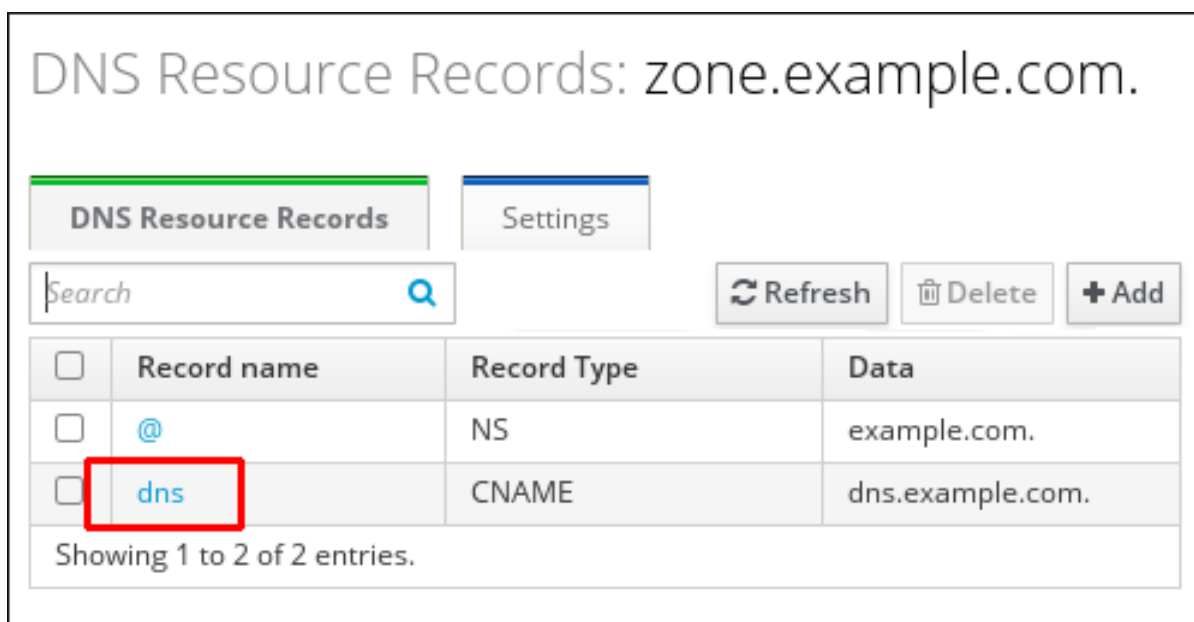
先决条件

- 以 IdM 管理员身份登录。

步骤

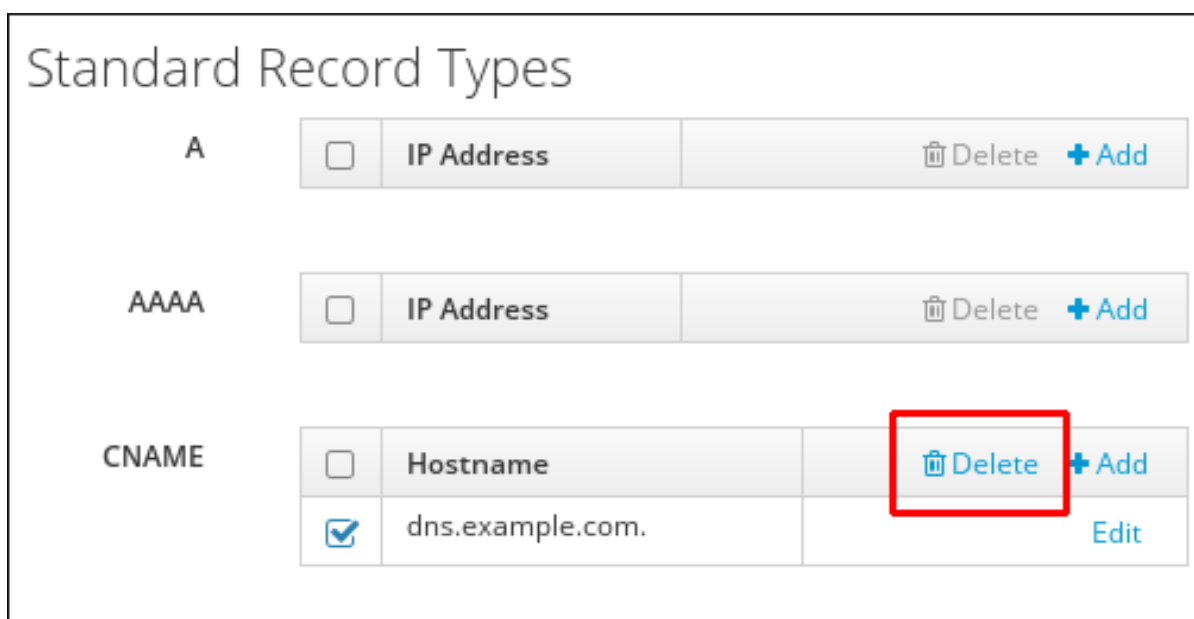
1. 在 IdM Web UI 中，点 **Network Services** → **DNS** → **DNS Zones**。
2. 点您要从中删除 DNS 记录的区域，如 `example.com`。
3. 在 **DNS Resource Records** 部分，点资源记录的名称。

图 7.3. 选择 DNS 资源记录



4. 根据要删除的记录类型的名称选择复选框。
5. 单击 **Delete**。

图 7.4. 删除 DNS 资源记录



所选记录类型现已删除。资源记录的其他配置保持不变。

其他资源

- 请参阅 [在 IdM Web UI 中删除整个 DNS 记录](#)。

7.6. 删除 IDM WEB UI 中的整个 DNS 记录

按照以下流程，使用身份管理(IdM) Web UI 删除区域中特定资源的所有记录。

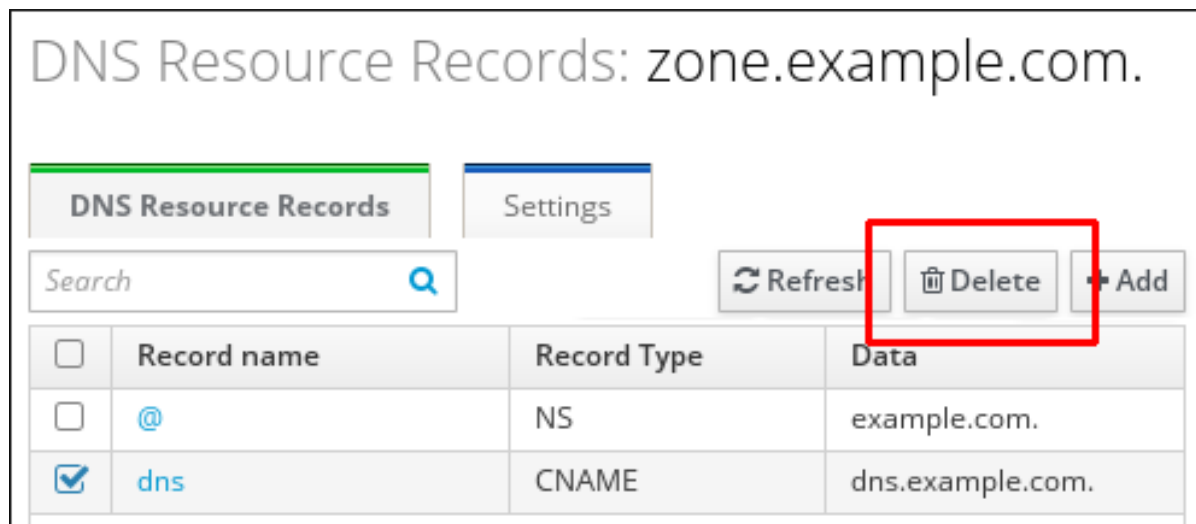
先决条件

- 以 IdM 管理员身份登录。

步骤

1. 在 IdM Web UI 中，点 **Network Services** → **DNS** → **DNS Zones**。
2. 点您要从中删除 DNS 记录的区域，如 `zone.example.com`。
3. 在 **DNS Resource Records** 部分，选择要删除的资源记录的复选框。
4. 单击 **Delete**。

图 7.5. 删除一个 Entire 资源记录



整个资源记录现已被删除。

7.7. 删除 IDM CLI 中的 DNS 记录

按照以下流程，从身份管理(IdM) DNS 管理的区中删除 DNS 记录。

先决条件

- 以 IdM 管理员身份登录。

步骤

- 要从区中删除记录，请使用 `ipa dnsrecord-del` 命令，将 `--recordType-rec` 选项和记录值一起添加。例如，要删除 A 类型记录：

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

如果您在没有任何选项的情况下运行 `ipa dnsrecord-del`，该命令会提示输入有关要删除的记录的信息。请注意，为命令传递 `--del-all` 选项会删除区域的所有相关记录。

其他资源

- 运行 `ipa dnsrecord-del --help` 命令。

7.8. 其他资源

- 请参阅[使用 Ansible 管理 IdM 中的 DNS 记录](#)。

第 8 章 使用 ANSIBLE 管理 IDM 中的 DNS 记录

本章论述了如何使用 Ansible playbook 管理 Identity Management(IdM)中的 DNS 记录。作为 IdM 管理员，您可以在 IdM 中添加、修改和删除 DNS 记录。本章包含以下部分：

- 使用 Ansible 确保 IdM 中存在 A 和 AAAA DNS 记录
- 使用 Ansible 确保 IdM 中存在 A 和 PTR DNS 记录
- 使用 Ansible 确保 IdM 中存在多个 DNS 记录
- 使用 Ansible 确保 IdM 中存在多个 CNAME 记录
- 使用 Ansible 确保 IdM 中是否存在 SRV 记录

8.1. IDM 中的 DNS 记录

身份管理(IdM)支持许多不同的 DNS 记录类型。以下是最频繁使用的四个项：

一个

这是主机名和 IPv4 地址的基本映射。A 记录的记录名称是一个主机名，如 **www**。A 记录的 **IP 地址** 值是一个 IPv4 地址，如 **192.0.2.1**。

有关 A 记录的更多信息，请参阅 [RFC 1035](#)。

AAAA

这是主机名和 IPv6 地址的基本映射。AAAA 记录的记录名称是一个主机名，如 **www**。**IP 地址** 值是一个 IPv6 地址，如 **2001:DB8::1111**。

有关 AAAA 记录的更多信息，请参阅 [RFC 3596](#)。

SRV

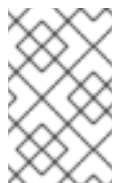
服务 (SRV) 资源记录 将服务名称映射到提供该特定服务的服务器的 DNS 名称。例如，此记录类型可以将类似 LDAP 目录的服务映射到管理它的服务器。

SRV 记录的记录名称格式为 **_service._protocol**，如 **_ldap._tcp**。SRV 记录的配置选项包括优先级、权重、端口号和目标服务的主机名。

有关 SRV 记录的更多信息，请参阅 [RFC 2782](#)。

PTR

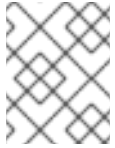
指针记录(PTR)添加反向 DNS 记录，它将 IP 地址映射到域名。



注意

IPv4 地址的所有反向 DNS 查找都使用在 **in-addr.arpa** 域中定义的反向条目。反向地址采用人类可读形式，与常规 IP 地址完全相反，它附加了 **in-addr.arpa** 域。例如，对于本网络地址 **192.0.2.0/24**，反向区域为 **2.0.192.in-addr.arpa**。

PTR 的记录名称必须是 [RFC 1035](#) 中指定的标准格式，它在 [RFC 2317](#) 和 [RFC 3596](#) 中扩展。主机名值必须是您要为其创建记录的主机的规范主机名。



注意

还可以为 IPv6 地址配置反向区域，即 `.ip6.arpa.` 域中的区域。有关 IPv6 反向区的更多信息，请参阅 [RFC 3596](#)。

添加 DNS 资源记录时，请注意很多记录需要不同的数据。例如，CNAME 记录需要一个主机名，而 A 记录则需要一个 IP 地址。在 IdM Web UI 中，用于添加新记录的表单字段会自动更新，以反映当前所选记录类型所需的数据。

8.2. 常见 IPA DNSRECORD-* 选项

在身份管理(IdM)中添加、修改和删除最常见的 DNS 资源记录类型时，您可以使用以下选项：

- A (IPv4)
- AAAA (IPv6)
- SRV
- PTR

在 **Bash** 中，您可以通过列出大括号内的逗号分隔列表中的值来定义多个条目，如 `--option={val1,val2,val3}`。

表 8.1. 常规记录选项

选项	描述
<code>--ttl=number</code>	将记录的时间设置为实时。
<code>--structured</code>	解析原始 DNS 记录，并以结构化格式返回它们。

表 8.2. "A" 记录选项

选项	描述	示例
<code>--a-rec=ARECORD</code>	传递单个 A 记录或 A 记录列表。	<code>ipa dnsrecord-add idm.example.com host1 --a-rec=192.168.122.123</code>
	可以创建具有给定 IP 地址的通配符 A 记录。	<code>ipa dnsrecord-add idm.example.com "*" --a-rec=192.168.122.123^[a]</code>
<code>--a-ip-address=string</code>	为记录指定 IP 地址。在创建记录时，指定 A 记录值的选项为 <code>--a-rec</code> 。但是，在修改 A 记录时，使用 <code>--a-rec</code> 选项指定 A 记录的当前值。使用 <code>--a-ip-address</code> 选项设置新值。	<code>ipa dnsrecord-mod idm.example.com --a-rec 192.168.122.123 --a-ip-address 192.168.122.124</code>

^[a] 这个示例创建通配符 A 记录，其 IP 地址为 192.0.2.123。

表 8.3. "AAAA"记录选项

选项	描述	示例
<code>--aaaa-rec=AAAARECORD</code>	通过单个 AAAA(IPv6)记录或 AAAA 记录列表。	<code>ipa dnsrecord-add idm.example.com www --aaaa-rec 2001:db8::1231:5675</code>
<code>--aaaa-ip-address=string</code>	为记录指定 IPv6 地址。在创建记录时，指定 A 记录值的选项为 <code>--aaaa-rec</code> 。但是，在修改 A 记录时， <code>--aaaa-rec</code> 选项用于指定 A 记录的当前值。使用 <code>--ip-address</code> 选项设置新值。	<code>ipa dnsrecord-mod idm.example.com --aaaa-rec 2001:db8::1231:5675 --aaaa-ip-address 2001:db8::1231:5676</code>

表 8.4. "PTR"记录选项

选项	描述	示例
<code>--ptr-rec=PTRRECORD</code>	传递单个 PTR 记录或 PTR 记录列表。当添加反向 DNS 记录时，与 <code>ipa dnsrecord-add</code> 命令使用的区名称会被相反，与添加其他 DNS 记录的用法不同。通常，主机 IP 地址是给定网络中的 IP 地址的最后一个八进制数。右侧的第一个示例为 <code>server4.idm.example.com</code> 添加 IPv4 地址为 <code>192.168.122.4</code> 的 PTR 记录。第二个示例为 <code>0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa</code> 。添加了一个反向 DNS 条目。主机 <code>server2.example.com</code> 的 IPv6 反向区域，IP 地址为 <code>2001:DB8::1111</code> 。	<pre>ipa dnsrecord-add 122.168.192.in-addr.arpa 4 --ptr-rec server4.idm.example.com.</pre> <pre>\$ 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 --ptr-rec server2.idm.example.com.</pre>
<code>--ptr-hostname=string</code>	为记录提供主机名。	

表 8.5. "SRV"记录选项

选项	描述	示例
<code>--srv-rec=SRVRECORD</code>	通过单个 SRV 记录或 SRV 记录列表。在右侧的示例中， <code>_ldap._tcp</code> 定义 SRV 记录的服务类型和连接协议。 <code>--srv-rec</code> 选项定义优先级、权重、端口和目标值。示例中的权重值为 51 和 49（总和为 100），它们代表使用特定记录的可能性（以百分比表示）。	<pre># ipa dnsrecord-add idm.example.com _ldap._tcp --srv-rec="0 51 389 server1.idm.example.com."</pre> <pre># ipa dnsrecord-add server.idm.example.com _ldap._tcp --srv-rec="1 49 389 server2.idm.example.com."</pre>

选项	描述	示例
<code>--srv-priority=number</code>	设置记录的优先级。服务类型可以有多个 SRV 记录。优先级(0 - 65535)设置记录排名；编号越低，优先级越高。服务必须首先使用优先级最高的记录。	<pre># ipa dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="1 49 389 server2.idm.example.com." --srv- priority=0</pre>
<code>--srv-weight=number</code>	设置记录的权重。这有助于决定具有相同优先级的 SRV 记录顺序。设定的权重应添加最多 100，代表使用特定记录的概率（以百分比表示）。	<pre># ipa dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 49 389 server2.idm.example.com." --srv- weight=60</pre>
<code>--srv-port=number</code>	为目标主机上的服务指定端口。	<pre># ipa dnsrecord-mod server.idm.example.com _ldap._tcp --srv-rec="0 60 389 server2.idm.example.com." --srv- port=636</pre>
<code>--srv-target=string</code>	指定目标主机的域名。如果域中没有服务，则这可以是一个句点(.)。	

其他资源

- 运行 `ipa dnsrecord-add --help`。

8.3. 使用 ANSIBLE 确保 IDM 中存在 A 和 AAAA DNS 记录

按照以下流程，使用 Ansible playbook 确保特定 IdM 主机的 A 和 AAAA 记录存在。在以下流程中使用的示例中，IdM 管理员可确保在 `idm.example.com` DNS 区域中存在 `host1` 的 A 和 AAAA 记录。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。
- `idm.example.com` 区域存在，由 IdM DNS 管理。有关在 IdM DNS 中添加主 DNS 区域的更多信息，请参阅[使用 Ansible playbook 管理 IdM DNS 区域](#)。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 `[ipaserver]` 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `ensure-A-and-AAAA-records-are-present.yml` Ansible playbook 文件。例如：

```
$ cp ensure-A-and-AAAA-records-are-present.yml ensure-A-and-AAAA-records-are-present-copy.yml
```

4. 打开 `ensure-A-and-AAAA-records-are-present-copy.yml` 文件进行编辑。

5. 通过在 `ipadnsrecord` task 部分中设置以下变量来修改该文件：

- 将 `ipaadmin_password` 变量设置为 IdM 管理员密码。
- 将 `zone_name` 变量设置为 `idm.example.com`。
- 在 `records` 变量中，将 `name` 变量设置为 `host1`，将 `a_ip_address` 变量设置为 `192.168.122.123`。
- 在 `records` 变量中，将 `name` 变量设置为 `host1`，并将 `aaaa_ip_address` 变量设置为 `::1`。这是当前示例修改的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-A-and-AAAA-records-are-present-copy.yml
```

其他资源

- 请参阅 [IdM 中的 DNS 记录](#)。
- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsrecord.md` 文件。
- 请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 目录中的 Ansible playbook 示例。

8.4. 使用 ANSIBLE 确保 IDM 中存在 A 和 PTR DNS 记录

按照以下流程，使用 Ansible playbook 确保特定 IdM 主机的 A 记录存在，且包含对应的 PTR 记录。在以下流程中使用的示例中，IdM 管理员可确保在 `idm.example.com` 区域中有 IP 地址为 `192.168.122.45` 的 `host1` 的 A 和 PTR 记录。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。
- `idm.example.com` DNS 区域存在，并由 IdM DNS 管理。有关在 IdM DNS 中添加主 DNS 区域的更多信息，请参阅[使用 Ansible playbook 管理 IdM DNS 区域](#)。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 `[ipaserver]` 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `ensure-dnsrecord-with-reverse-is-present.yml` Ansible playbook 文件。例如：

```
$ cp ensure-dnsrecord-with-reverse-is-present.yml ensure-dnsrecord-with-reverse-is-present-copy.yml
```

4. 打开 `ensure-dnsrecord-with-reverse-is-present-copy.yml` 文件进行编辑。

5. 通过在 `ipadnsrecord` task 部分中设置以下变量来修改该文件：

- 将 `ipaadmin_password` 变量设置为 IdM 管理员密码。
- 将 `name` 变量设置为 `host1`。
- 将 `zone_name` 变量设置为 `idm.example.com`。
- 将 `ip_address` 变量设置为 `192.168.122.45`。
- 将 `create_reverse` 变量设置为 `true`。
这是当前示例修改的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-
dnsrecord-with-reverse-is-present-copy.yml
```

其他资源

- 请参阅 [IdM 中的 DNS 记录](#)。
- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsrecord.md` 文件。
- 请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 目录中的 Ansible playbook 示例。

8.5. 使用 ANSIBLE 确保 IDM 中存在多个 DNS 记录

按照以下流程，使用 Ansible playbook 确保多个值与特定 IdM DNS 记录相关联。在以下示例中，IdM 管理员确保在 `idm.example.com` DNS 区域中存在 `host1` 的多个 A 记录。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。

- 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
- 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
- 示例假定 `secret.yml` Ansible vault 存储了 `ipaadmin_password`。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。
- `idm.example.com` 区域存在，由 IdM DNS 管理。有关在 IdM DNS 中添加主 DNS 区域的更多信息，请参阅[使用 Ansible playbook 管理 IdM DNS 区域](#)。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 `[ipaserver]` 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `ensure-presence-multiple-records.yml` Ansible playbook 文件。例如：

```
$ cp ensure-presence-multiple-records.yml ensure-presence-multiple-records-copy.yml
```

4. 打开 `ensure-presence-multiple-records-copy.yml` 文件进行编辑。

5. 通过在 `ipadnsrecord` task 部分中设置以下变量来修改该文件：

- 将 `ipaadmin_password` 变量设置为 IdM 管理员密码。
- 在 `records` 部分中，将 `name` 变量设置为 `host1`。
- 在 `records` 部分中，将 `zone_name` 变量设置为 `idm.example.com`。
- 在 `records` 项中，将 `a_rec` 变量设置为 `192.168.122.112`，以及 `192.168.122.122`。
- 在 `records` 部分中定义第二条记录：
 - 将 `name` 变量设置为 `host1`。
 - 将 `zone_name` 变量设置为 `idm.example.com`。
 - 将 `aaaa_rec` 变量设置为 `::1`。

这是当前示例修改的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-presence-multiple-records-copy.yml
```

其他资源

- 请参阅 [IdM 中的 DNS 记录](#)。
- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsrecord.md` 文件。
- 请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 目录中的 Ansible playbook 示例。

8.6. 使用 ANSIBLE 确保 IDM 中存在多个 CNAME 记录

Canonical Name 记录（CNAME 记录）是域名系统(DNS)中的一种资源记录，可将一个域名（别名）映射到另一个名称（规范名称）。

从一个 IP 地址运行多个服务时，您可能会发现 CNAME 记录很有用：例如，FTP 服务和 Web 服务，各自在不同端口中运行。

按照以下流程，使用 Ansible playbook 确保多个 CNAME 记录在 IdM DNS 中存在。在以下示例中，`host03` 既是 HTTP 服务器和 FTP 服务器。IdM 管理员确保 `idm.example.com` 区域中存在 `host03` A 记录的 `www` 和 `ftp` CNAME 记录。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 [ansible-freeipa](#) 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。

- 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。
- `idm.example.com` 区域存在，由 IdM DNS 管理。有关在 IdM DNS 中添加主 DNS 区域的更多信息，请参阅[使用 Ansible playbook 管理 IdM DNS 区域](#)。
- `host03 A` 记录存在于 `idm.example.com` 区域中。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 `[ipaserver]` 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `ensure-CNAME-record-is-present.yml` Ansible playbook 文件。例如：

```
$ cp ensure-CNAME-record-is-present.yml ensure-CNAME-record-is-present-copy.yml
```

4. 打开 `ensure-CNAME-record-is-present-copy.yml` 文件进行编辑。
5. 通过在 `ipadnsrecord` task 部分中设置以下变量来修改该文件：

- （可选）使用 play 的 `name` 提供的描述。
- 将 `ipadmin_password` 变量设置为 IdM 管理员密码。
- 将 `zone_name` 变量设置为 `idm.example.com`。
- 在 `records` 变量部分，设置以下变量和值：
 - 将 `name` 变量设置为 `www`。
 - 将 `cname_hostname` 变量设置为 `host03`。
 - 将 `name` 变量设置为 `ftp`。
 - 将 `cname_hostname` 变量设置为 `host03`。

这是当前示例修改的 Ansible playbook 文件：

```
---
- 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:
- ipadsrecord:
  ipadmin_password: "{{ ipadmin_password }}"
  zone_name: idm.example.com
  records:
  - name: www
    cname_hostname: host03
  - name: ftp
    cname_hostname: host03

```

6. 保存这个文件。

7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-CNAME-record-is-present.yml
```

其他资源

- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsrecord.md` 文件。
- 请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 目录中的 Ansible playbook 示例。

8.7. 使用 ANSIBLE 确保 IDM 中是否存在 SRV 记录

DNS 服务 (SRV) 记录定义域中可用服务的主机名、端口号、传输协议、优先级和权重。在 Identity Management(IdM)中，您可以使用 SRV 记录来定位 IdM 服务器和副本。

按照以下流程，使用 Ansible playbook 确保 SRV 记录在 IdM DNS 中存在。在以下示例中，IdM 管理员可确保存在 `_kerberos_udp.idm.example.com` SRV 记录，其值为 `10 50 88 idm.example.com`。这将设置以下值：

- 它将服务的优先级设置为 10。
- 它将服务的权重设置为 50。
- 它将服务要使用的端口设置为 88。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 `ansible-freeipa` 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 `Ansible 清单文件`。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点（这是执行 `ansible-freeipa` 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。

- 您知道 IdM 管理员密码。
- `idm.example.com` 区域存在，由 IdM DNS 管理。有关在 IdM DNS 中添加主 DNS 区域的更多信息，请参阅[使用 Ansible playbook 管理 IdM DNS 区域](#)。

步骤

1. 进入 `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 目录：

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/dnsrecord
```

2. 打开您的清单文件，并确保您要配置的 IdM 服务器列在 `[ipaserver]` 部分。例如，要指示 Ansible 配置 `server.idm.example.com`，请输入：

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

3. 复制 `ensure-SRV-record-is-present.yml` Ansible playbook 文件。例如：

```
$ cp ensure-SRV-record-is-present.yml ensure-SRV-record-is-present-copy.yml
```

4. 打开 `ensure-SRV-record-is-present-copy.yml` 文件进行编辑。
5. 通过在 `ipadnsrecord` task 部分中设置以下变量来修改该文件：

- 将 `ipaadmin_password` 变量设置为 IdM 管理员密码。
 - 将 `name` 变量设置为 `_kerberos._udp.idm.example.com`。
 - 将 `srv_rec` 变量设置为 `'10 50 88 idm.example.com'`。
 - 将 `zone_name` 变量设置为 `idm.example.com`。
- 对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。
7. 运行 playbook:

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-SRV-record-is-present.yml
```

其他资源

- 请参阅 [IdM 中的 DNS 记录](#)。
- 请参阅 `/usr/share/doc/ansible-freeipa/` 目录中的 `README-dnsrecord.md` 文件。
- 请参阅 `/usr/share/doc/ansible-freeipa/playbooks/dnsrecord` 目录中的 Ansible playbook 示例。

第 9 章 在 IDM 中使用规范的 DNS 主机名

在 Identity Management(IdM)客户端默认禁用 DNS 规范，以避免潜在的安全风险。例如，如果攻击者控制了域中的 DNS 服务器和一个主机，攻击者可能会导致在用户使用短主机名（如 **demo**）时将其解析到被入侵的主机，如 **malicious.example.com**。在这种情况下，用户可以连接到与预期不同的服务器。

这个流程描述了如何在 IdM 客户端中使用规范化主机名。

9.1. 在主机主体中添加别名

默认情况下，使用 **ipa-client-install** 命令注册的 Identity Management(IdM)客户端不允许在服务主体中使用短主机名。例如，在访问服务时，用户只能使用 **host/demo.example.com@EXAMPLE.COM** 而不是 **host/demo@EXAMPLE.COM**。

按照以下流程在 Kerberos 主体中添加别名。请注意，您还可以在 **/etc/krb5.conf** 文件中启用主机名的规范。详情请参阅 [在客户端上的服务主体中启用主机名规范](#)。

先决条件

- 已安装 IdM 客户端。
- 主机名在网络中是唯一的。

步骤

1. 以 **admin** 用户身份向 IdM 进行身份验证：

```
$ kinit admin
```

2. 在主机主体中添加别名。例如，将 **demo** 别名添加到 **demo.example.com** 主机主体：

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

9.2. 在客户端中的服务主体中启用主机名的规范

按照以下流程，在客户端上的服务主体中启用主机名规范化。

请注意，如果您使用主机主体别名，如 [将别名添加到主机主体](#) 中所述，则不需要启用规范。

先决条件

- 安装了 Identity Management(IdM)客户端。
- 以 **root** 用户身份登录到 IdM 客户端。
- 主机名在网络中是唯一的。

步骤

1. 将 **/etc/krb5.conf** 文件中的 **[libdefaults]** 部分中的 **dns_canonicalize_hostname** 参数设置为 **false**：

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

9.3. 在启用了 DNS 主机名的情况下使用主机名的选项

如果您在 `/etc/krb5.conf` 文件中设置了 `dns_canonicalize_hostname = true`，如 [在客户端上的服务主体中启用主机名规范](#) 中所述，在服务主体中使用主机名时，您有如下选择：

- 在 Identity Management(IdM)环境中，您可以在服务主体中使用完整主机名，如 **host/demo.example.com@EXAMPLE.COM**。
- 在没有 IdM 的环境中，但 RHEL 主机是 Active Directory(AD)域的成员，则不需要进一步的考虑，因为 AD 域控制器(DC)会自动为注册到 AD 的虚拟机的 NetBIOS 名称创建服务主体。