



Red Hat OpenStack Platform 17.1

DNS をサービスとして設定

Red Hat OpenStack Platform における DNS サービスを使用した Domain Name System (DNS) 管理方法に関する情報

Red Hat OpenStack Platform 17.1 DNS をサービスとして設定

Red Hat OpenStack Platform における DNS サービスを使用した Domain Name System (DNS) 管理方法に関する情報

OpenStack Team
rhos-docs@redhat.com

法律上の通知

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.

概要

RHOSP DNS サービス (designate) のインストール、設定、操作、トラブルシューティング、およびアップグレードに関するガイドです。

目次

多様性を受け入れるオープンソースの強化	4
RED HAT ドキュメントへのフィードバック (英語のみ)	5
第1章 DNS サービスの概要	6
1.1. DNS (DOMAIN NAME SYSTEM) の基本	6
1.2. RHOSP DNS サービスの導入	9
1.3. DNS サービスコンポーネント	9
1.4. DNS サービスの一般的なデプロイメントシナリオ	10
1.5. DNS サービスを使用するさまざまな方法	11
第2章 DNS サービスデプロイメントのプランニング	13
2.1. DNS サーバー機能のサポートマトリックス	13
2.2. DNS サービスのソフトウェア要件	16
2.3. DNS サービス用の既存の BIND サーバーの設定	16
2.4. 推奨される DNS サービストポロジー	17
2.5. DNS サービスの高可用性について	18
第3章 DNS サービスのインストールと設定	19
3.1. DNS サービスのデプロイ	19
3.2. 既存の BIND 9 サーバーを使用した DNS サービスのデプロイメント	21
3.3. DNS サービスのデフォルト設定の変更	23
第4章 統合 DNS サービスの使用	25
4.1. SETTING UP A PROJECT FOR DNS INTEGRATION	25
4.2. 仮想マシンインスタンスの DNS との統合	29
4.3. DNS とポートの統合	30
4.4. FLOATING IP と DNS の統合	31
第5章 トップレベルドメイン名の管理	33
5.1. トップレベルドメインについて	33
5.2. トップレベルドメインの作成	34
5.3. トップレベルドメインのリスト表示および表示	35
5.4. トップレベルドメインの変更	35
5.5. トップレベルドメインの削除	36
5.6. DNS サービスの拒否リストについて	37
5.7. 拒否リストの DNS サービス正規表現について	38
5.8. DNS サービスの拒否リストの作成	38
5.9. DNS サービスの拒否リストのリスト表示および表示	39
5.10. DNS サービスの拒否リストの変更	40
5.11. DNS サービスの拒否リストの削除	41
第6章 DNS リソースのクォータの表示および管理	42
6.1. DNS リソースのクォータの表示	42
6.2. DNS リソースのクォータの変更	43
6.3. DNS リソースクォータをデフォルト値にリセットする	44
6.4. DNS サービスクォータとそのデフォルト値	45
第7章 ゾーンの管理	47
7.1. DNS サービスのゾーン	47
7.2. ゾーンの作成	47
7.3. ゾーンの更新	48
7.4. ゾーンの削除	49

7.5. ゾーンのエクスポート	50
7.6. ゾーンのインポート	52
7.7. ゾーン所有権の譲渡	54
7.8. ゾーン転送リクエストの変更	57
第8章 レコードセットの管理	59
8.1. DNS サービスのレコードおよびレコードセットについて	59
8.2. レコードセットの作成	60
8.3. レコードセットの更新	62
8.4. レコードセットの削除	63
第9章 ポインターレコード (PTR) の管理	65
9.1. PTR レコードの基本	65
9.2. リバースルックアップゾーンの作成	66
9.3. PTR レコードの作成	68
9.4. 複数の PTR レコードの作成	69
9.5. FLOATING IP アドレスの PTR レコードの設定	71
9.6. FLOATING IP アドレスの PTR レコードの設定解除	73
第10章 DNS サービスのトラブルシューティング	76
10.1. DNS サービスと BIND ログ	76
10.2. DNS サービスプール設定のエクスポート	76
10.3. 利用可能な DNS サービスエンドポイントのリスト表示	78

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

Jira でドキュメントのフィードバックを提供する

ドキュメントに関するフィードバックを提供するには、[Create Issue](#) フォームを使用します。Red Hat OpenStack Platform Jira プロジェクトで Jira Issue が作成され、フィードバックの進行状況を追跡できます。

1. Jira にログインしていることを確認してください。Jira アカウントをお持ちでない場合は、アカウントを作成してフィードバックを送信してください。
2. [Create Issue](#) をクリックして、**Create Issue** ページを開きます。
3. **Summary** フィールドと **Description** フィールドに入力します。**Description** フィールドに、ドキュメントの URL、章またはセクション番号、および問題の詳しい説明を入力します。フォーム内の他のフィールドは変更しないでください。
4. **Create** をクリックします。

第1章 DNS サービスの概要

DNS サービス (designate) は、Red Hat OpenStack Platform (RHOSP) デプロイメント向けの DNS-as-a-Service 実装を提供します。

このセクションでは、Domain Name System (DNS) の基本と DNS サービスコンポーネントについて説明し、簡単な使用例を示し、DNS サービスを実行するさまざまな方法を示します。

このセクションに含まれるトピックは次のとおりです。

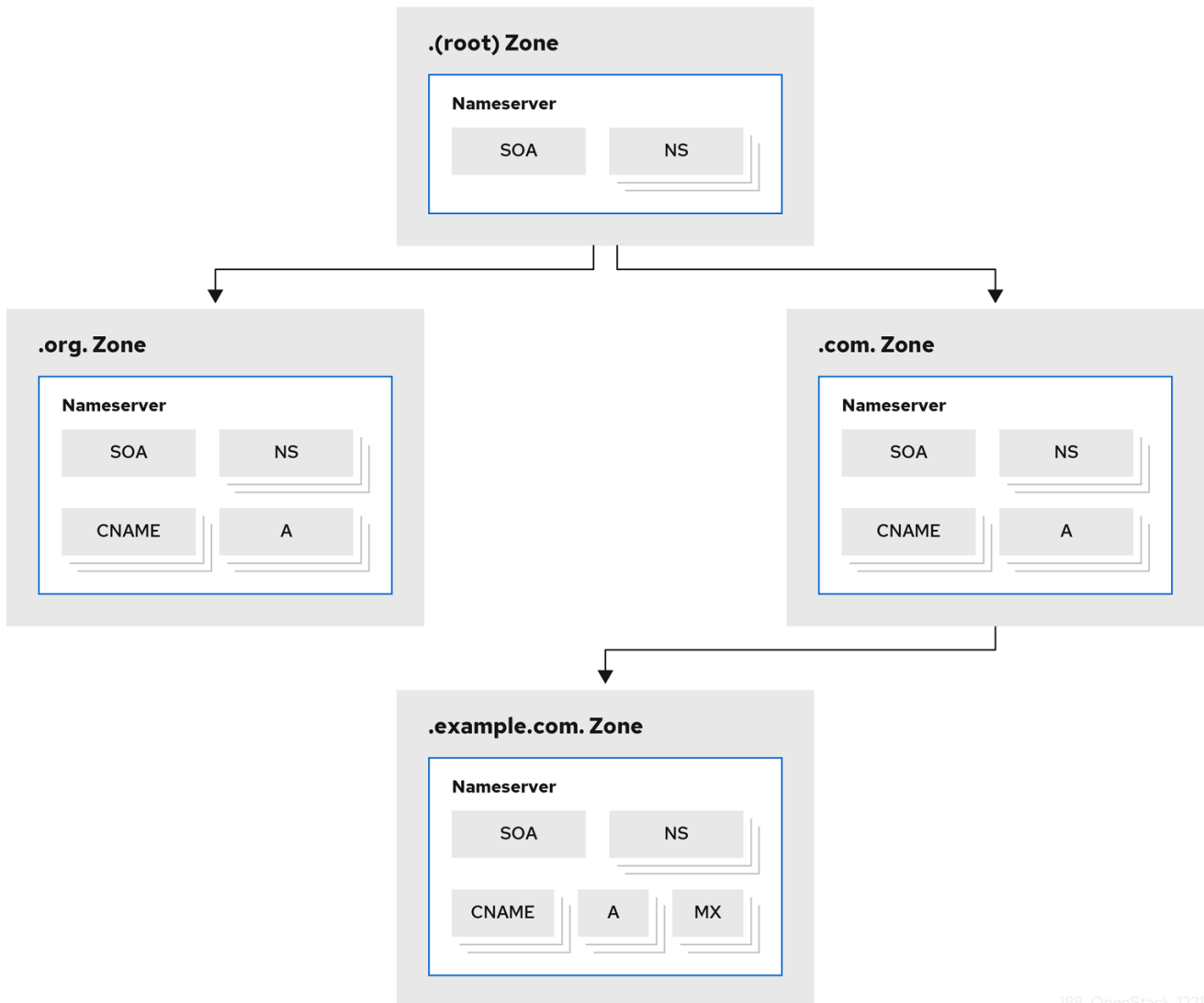
- [「DNS \(Domain Name System\) の基本」](#)
- [「RHOSP DNS サービスの導入」](#)
- [「DNS サービスコンポーネント」](#)
- [「DNS サービスの一般的なデプロイメントシナリオ」](#)
- [「DNS サービスを使用するさまざまな方法」](#)

1.1. DNS (DOMAIN NAME SYSTEM) の基本

DNS (Domain Name System) は、プライベートまたはパブリックネットワークに接続されているリソースの命名システムです。階層型の分散データベースである DNS は、リソースに関する情報を、ゾーンと呼ばれるさまざまなグループに編成されたドメイン名に関連付けます。権威ネームサーバーは、リソースとゾーンの情報をレコードに格納します。リゾルバーはこれらのレコードを照会して、ネットワークデータをルーティングするためのリソースを識別し、その場所を特定できます。

名前はゾーンの階層に分割され、これにより委譲が容易になります。個別のネームサーバーが特定のゾーンを担当します。

図1.1 Domain Name System



188_OpenStack_I221

単純に.(ドット)には、さまざまなトップレベルドメイン (TLD) を他のネームサーバーに委任するレコードが含まれています。これらの種類のレコードはネームサーバー (NS) レコードと呼ばれ、特定のドメインに対して権限のある DNS サーバーを識別します。ドメインのプライマリおよびバックアップネームサーバーを示す NS レコードが複数あることは珍しくありません。

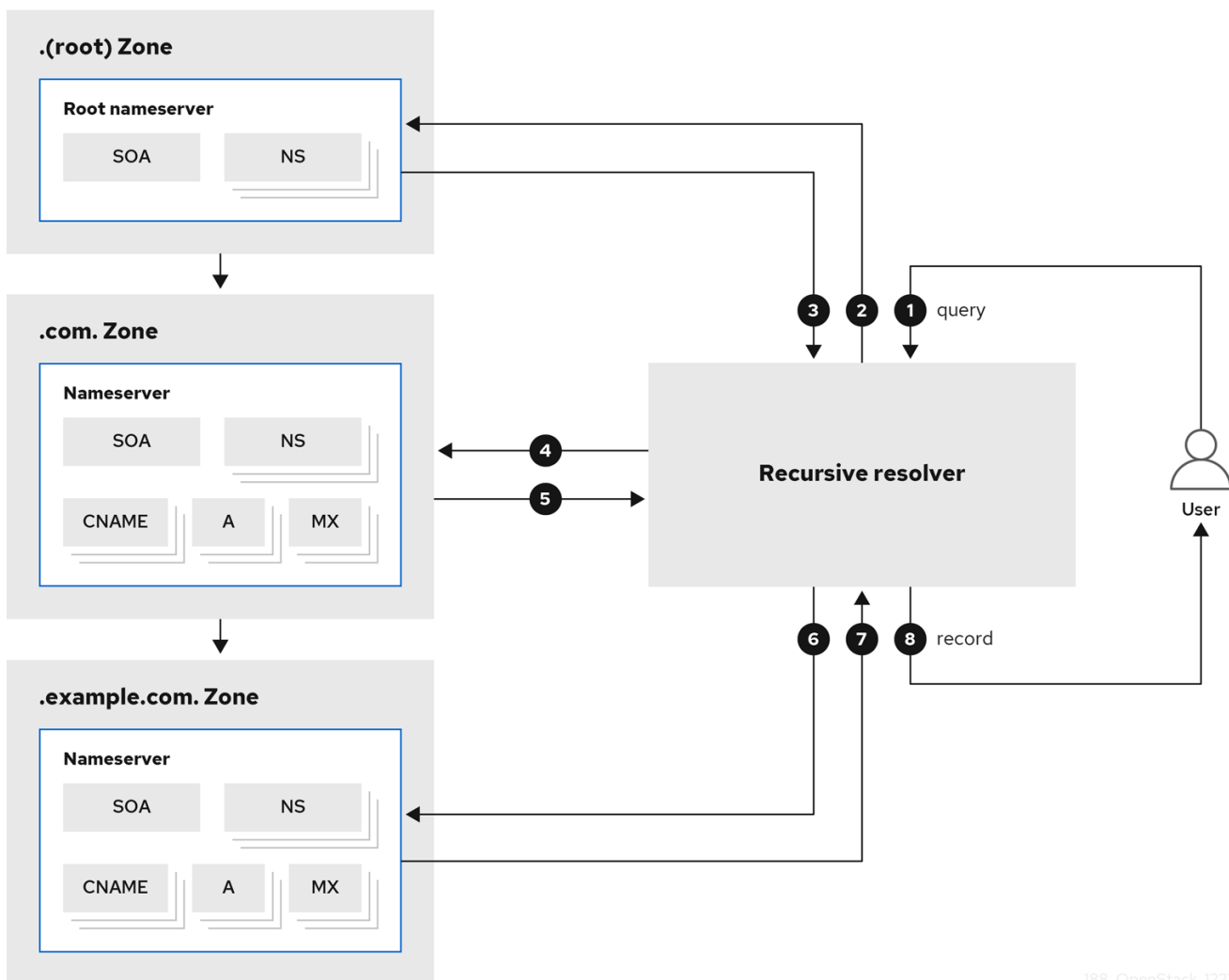
ルートゾーンの下には、TLD 内のドメインのレコードのみを含むさまざまな TLD ネームサーバーがあります。これらはアドレスレコードおよび正規名レコードであり、それぞれ **A** レコードおよび **CNAME** レコードと呼ばれます。

たとえば、**.com** ネームサーバーには、ゾーンを他のネームサーバーに委譲する NS レコードに加えて、**example.com** の **CNAME** レコードが含まれます。ドメイン **example.com** には独自のネームサーバーがあり、それにより **cloud.example.com** などの他のドメインを作成できます。

多くの場合、リゾルバーは2つの部分で設定されています。通常はユーザーのコンピューター上のライブラリーである**スタブリゾルバー**と、結果をユーザーに返す前にネームサーバーに対してクエリーを実行する**再帰リゾルバー**です。ドメインを検索する場合、リゾルバーはドメインの最後から開始し、ドメインの最初に向かって実行します。

たとえば、**cloud.example.com** を検索する場合、リゾルバーはルートネームサーバー **.** から開始します。ルートは、**.com** ネームサーバーの場所で応答します。次に、リゾルバーは **.com** ネームサーバーに接続して、**example.com** ネームサーバーを取得します。最後に、リゾルバーは **cloud.example.com** レコードを特定し、ユーザーに返します。

図1.2 DNS クエリーの解決



188_OpenStack_1221

1	ユーザーは cloud.example.com のアドレスのクエリーを実行します。
2	再帰リゾルバーは、 root ゾーン名前サーバーに cloud.example.com を照会します。
3	レコードが見つからず、 root ゾーンが .com の名前サーバーを提供します。
4	リゾルバーは、 cloud.example.com の .com 名前サーバーにクエリーを実行します。
5	レコードが見つからず、 .com ゾーンが example.com の名前サーバーを提供します。
6	リゾルバーは、 example.com 名前サーバーに cloud.example.com を照会します。
7	example.com 名前サーバーは cloud.example.com を見つけ、 cloud.example.com の A レコードをリゾルバーに提供します。
8	リゾルバーは、 cloud.example.com の A レコードをユーザーに転送します。

この検索をより効率的にするために、結果はリゾルバーにキャッシュされます。そうすることで、最初のユーザーが **cloud.example.com** を要求した後、リゾルバーは後続の要求に対してキャッシュされた結果を迅速に返すことができます。

関連情報

- https://en.wikipedia.org/wiki/Domain_Name_System
- <https://tools.ietf.org/html/rfc1034>
- 「RHOSP DNS サービスの導入」

1.2. RHOSP DNS サービスの導入

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) は、DNS レコード、名前、およびゾーンを管理できるマルチテナントサービスです。RHOSP DNS サービスは REST API を提供し、ユーザー管理のために RHOSP Identity サービス (keystone) と統合されています。

RHOSP director を使用して BIND インスタンスをデプロイして DNS レコードを含めるか、DNS サービスを既存の BIND インフラストラクチャーに統合することができます。また、director は RHOSP Networking サービス (neutron) との DNS サービスの統合を設定して、コンピュートインスタンス、ネットワークポート、および Floating IP のレコードを自動的に作成することができます。

関連情報

- 「DNS (Domain Name System) の基本」
- 「DNS サービスコンポーネント」

1.3. DNS サービスコンポーネント

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) は、1つ以上の RHOSP コントローラーホスト上のコンテナで実行される複数のサービスで設定されています。

Designate API (designate-api コンテナ)

ユーザーおよび RHOSP Networking サービス (neutron) が designate と対話するための OpenStack 標準 REST API を提供します。API は、リクエストをリモートプロシージャコール (RPC) 経由で Central サービスに送信することで処理します。

Producer (designate-producer コンテナ)

designate が実行する定期的なタスクをオーケストレーションします。これらのタスクは、Ceilometer 用の **dns.zone.exists** の発行、データベースから削除されたゾーンのパーシステンス、セカンダリーゾーンのリフレッシュ間隔でのポーリング、遅延した **NOTIFY** の生成、エラー状態のゾーンの定期的な復旧の起動などの長時間稼働かつ大規模なジョブになり得るものです。

Central (designate-central コンテナ)

ゾーンおよびレコードセットの作成、更新、および削除のオーケストレーションを行います。Central サービスは、Designate API サービスによって送信された RPC 要求を受信し、永続ストレージを調整しながら必要なビジネスロジックをデータに適用します。

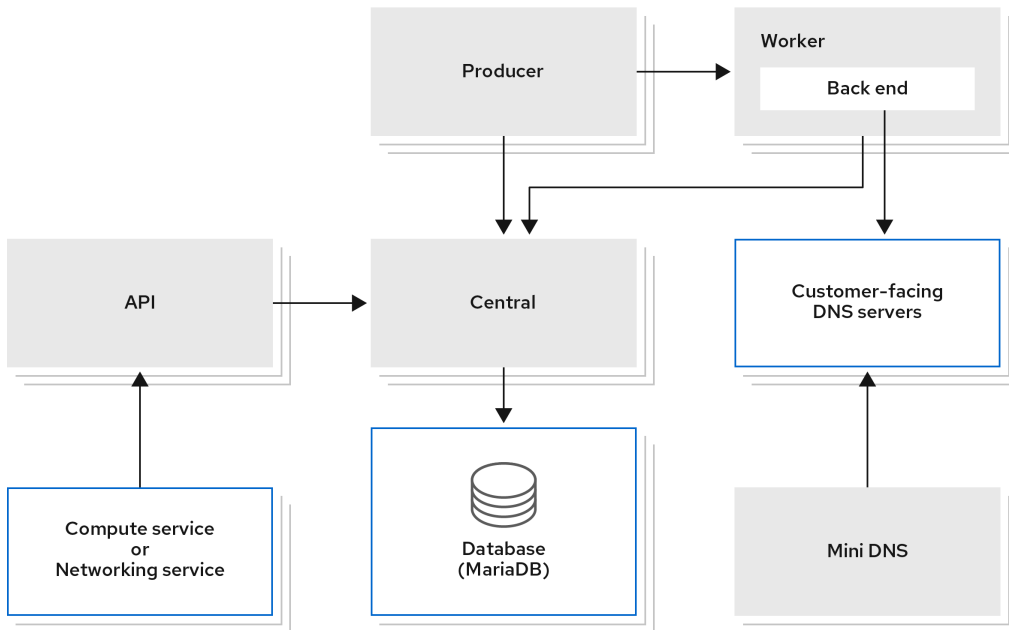
Worker (designate-worker コンテナ)

指定する DNS サーバーのドライバーにインターフェイスを提供します。ワーカーサービスは指定されたデータベースからサーバー設定を読み取り、Producer によって要求される定期タスクも管理します。

Mini DNS (designate-mdns コンテナ)

ネームサーバーからの Zone Authoritative Transfer (AXFR) 要求を管理します。Mini DNS サービスは、指定インフラストラクチャーの外部でホストされている DNS ゾーンに関する DNS 情報も取得します。

図1.3 DNS サービスアーキテクチャー



188_OpenStack_1221

RHOSP では、デフォルトで、DNS コンポーネントは BIND 9 および Unbound です。

BIND 9 (bind コンテナ)

DNS サービスに DNS サーバーを提供します。BIND は、DNS ソフトウェアのオープンソーススイートで、特に権威ネームサーバーとして機能します。

Unbound (unbound コンテナ)

DNS 再帰リゾルバーのロールを果たします。DNS 再帰リゾルバーは、DNS 要求を IP アドレスに変換するために必要なクエリーを開始してシーケンスします。Unbound は、DNS サービスが再帰リゾルバーとして使用するオープンソースプログラムです。

DNS サービスは、oslo 互換データベースを使用してデータを格納し、oslo メッセージングを使用してサービス間の通信を容易にします。DNS サービスの複数のインスタンスは、高可用性のデプロイメントを容易にするために、多くの場合はロードバランサーの後ろにある API プロセスと連携して実行できます。

関連情報

- [「DNS サービスの一般的なデプロイメントシナリオ」](#)
- [「DNS サービスを使用するさまざまな方法」](#)

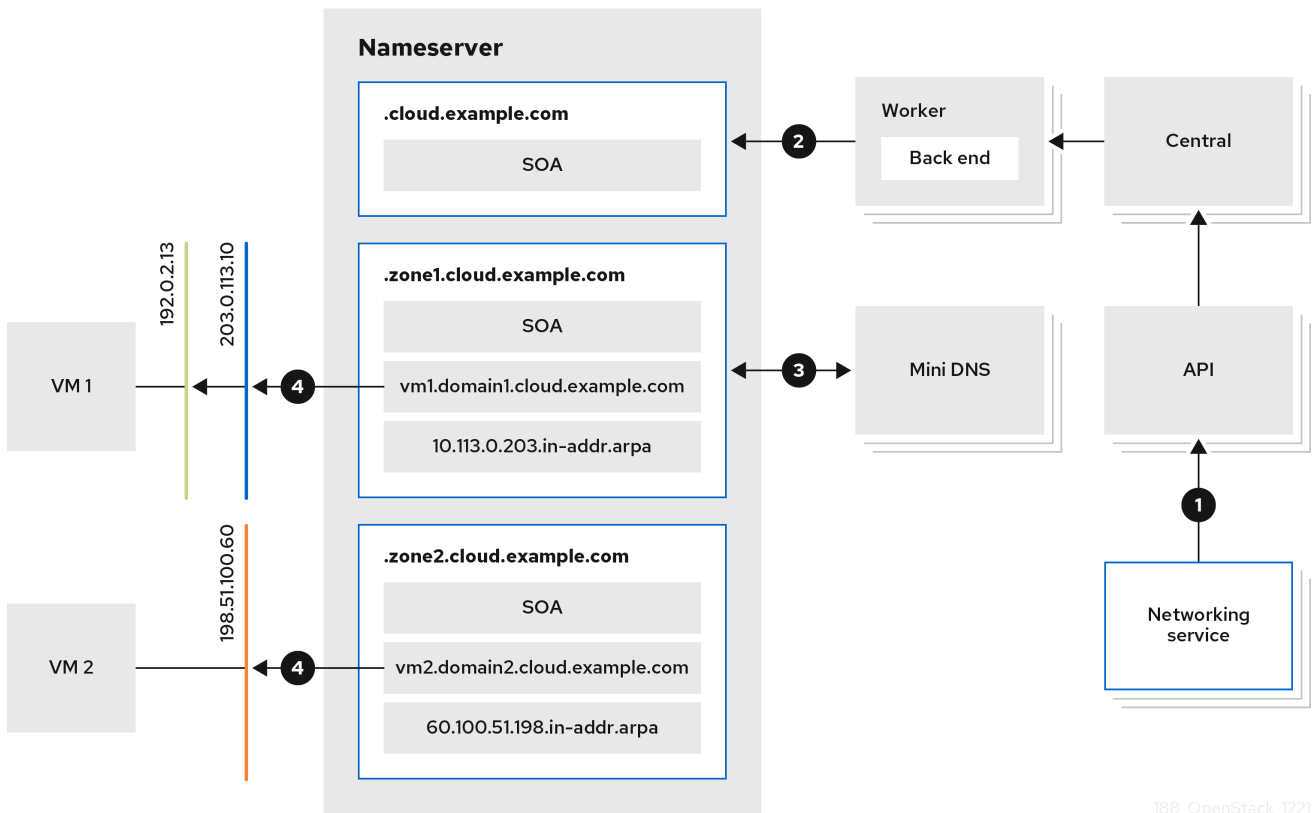
1.4. DNS サービスの一般的なデプロイメントシナリオ

ユーザーが、**zone1.cloud.example.com** と **zone2.cloud.example.com** の 2 つのゾーンを作成し、DNS サービスが DNS ネームサーバーに新しい Start of Authority (SOA) レコードと新しいネームサーバー (NS) レコードをそれぞれ追加します。

ユーザーは、RHOSP Networking サービスを使用し、プライベートネットワークを作成して **zone1** に関連付け、パブリックネットワークを作成して **zone2** に関連付けます。

最後に、ユーザーは仮想マシンインスタンスをプライベートネットワークに接続し、Floating IP をアタッチします。ユーザーは2番目のインスタンスをパブリックネットワークに直接接続します。これらの接続により、ネットワークサービスは、ユーザーに代わってレコードを作成するように DNS サービスに要求します。DNS サービスは、インスタンス名を権威ネームサーバーのドメインにマッピングし、PTR レコードを作成してリバースルックアップを有効にします。

図1.4 一般的な DNS サービスのデプロイメント



188_OpenStack_I221

1	RHOSP Networking サービスでは、ドメインと名前を Floating IP、ポート、およびネットワークに関連付けることができます。RHOSP Networking サービスは、ポートの作成および破棄時に、Designate API を使用してレコードを管理します。
2	Designate の Worker は、ゾーン情報を更新するようにネームサーバーに指示します。
3	ネームサーバーは、Mini DNS から更新されたゾーン情報を要求します。
4	ネームサーバーは、正引きレコードと逆引きレコードの両方を作成します。

関連情報

- [「RHOSP DNS サービスの導入」](#)
- [「DNS サービスコンポーネント」](#)

1.5. DNS サービスを使用するさまざまな方法

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) は REST API を提供し、一般的に 3 つの方法で使用されます。

- 最も一般的なのは、RHOSP サービスと対話するためのコマンドを備えた Python コマンドラインツールである RHOSP OpenStack クライアントを使用する方法です。
- グラフィカルユーザーインターフェイスである RHOSP Dashboard (horizon) を介して DNS サービスを使用することもできます。
- 開発者は、アプリケーションの作成に OpenStack SDK を使用できます。詳細は、[openstacksdk](#) を参照してください。

第2章 DNS サービスデプロイメントのプランニング

このセクションでは、Red Hat OpenStack Platform を使用した DNS サービス (designate) のデプロイメントを計画する際に考慮すべき重要なトピックについて説明します。

このセクションに含まれるトピックは次のとおりです。

- [「DNS サーバー機能のサポートマトリックス」](#)
- [「DNS サービスのソフトウェア要件」](#)
- [「DNS サービス用の既存の BIND サーバーの設定」](#)
- [「推奨される DNS サービストポロジー」](#)
- [「DNS サービスの高可用性について」](#)

2.1. DNS サーバー機能のサポートマトリックス

次の表は、Red Hat OpenStack Platform (RHOSP) 17 がサポートする DNS サービス (designate) の機能を示しています。

表2.1 DNS サービス (designate) 機能のサポートマトリックス

機能	RHOSP 17 でサポートされているか
x86_64 ハードウェアアーキテクチャー	はい
その他すべてのハードウェアアーキテクチャー	いいえ
BIND 9 バックエンド	はい
その他すべてのバックエンド	いいえ
Denylists (ブラックリスト)	はい
Designate v1 API	いいえ
Designate v2 API	はい
Designate admin API	いいえ
Designate Central サービス	はい
Designate Producer サービス	はい
Designate Worker サービス	はい
Designate miniDNS サービス	はい

Designate Agent サービス	いいえ
Designate Zone Manager サービス	いいえ
Designate Pool Manager サービス	いいえ
Designate OpenStack client プラグイン (CLI)	はい
Designate client (CLI)	いいえ
OpenStack Python SDK (designate)	はい
Designate クライアント (SDK)	いいえ
Designate horizon dashboard	はい
Designate tempest プラグイン	はい
Designate データベース MariaDB/Galera	はい
その他すべてのデータベース	いいえ
分散ロックマネージャー (Redis)	はい
その他すべての分散ロックマネージャーのオプション	いいえ
Designate シンク	いいえ
Designate 通知	はい
高可用性デプロイメント	はい
IPv4	はい
IPv6	はい
Monasca の統合	いいえ
デフォルトのプールスケジューラー	はい
その他すべてのプールスケジューラー	いいえ
単一のプール	はい
複数のプール	いいえ

Quotas	はい
ロールベースアクセス制御 (RBAC)	はい
レコードタイプ A	はい
レコードタイプ AAAA	はい
レコードタイプ CNAME	はい
レコードタイプ MX	はい
レコードタイプ SRV	はい
レコードタイプ TXT	はい
レコードタイプ SPF	はい
レコードタイプ NS	はい
レコードタイプ PTR	はい
レコードタイプ SSHFP	はい
レコードタイプ SOA	はい
レコードタイプ NAPTR	はい
レコードタイプ CAA	はい
その他すべてのレコードタイプ	いいえ
トップレベルドメイン (TLD)	はい
TSIG キー	はい
バインドされていない再帰リゾルバー	はい
他のすべての再帰リゾルバー	いいえ
プライマリーゾーン	はい
セカンダリーゾーン	いいえ
ゾーンのインポートとエクスポート	はい
ゾーンの破棄	いいえ

ゾーン所有権の譲渡	はい
-----------	----

2.2. DNS サービスのソフトウェア要件

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) は、次の RHOSP コアコンポーネントに依存します。

- Identity サービス (keystone)
- RabbitMQ
- MariaDB
- Redis

RHOSP のインストールおよび設定ツールセットである director は、これらのコンポーネントを DNS サービス用に自動的に設定します。

DNS サービスで DNS レコードを自動的に作成する VLAN またはオーバーレイネットワークを使用している場合は、これらのネットワーク用にいくつかのネットワークセグメンテーション ID を確保します。DNS サービスは、セグメント ID がネットワークサービス (neutron) `ml2_conf.ini` ファイルで指定された範囲内にあるネットワークの DNS レコードを作成しません。

関連情報

- [「推奨される DNS サービストポロジー」](#)
- [Setting up a project for DNS integration](#)

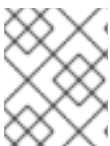
2.3. DNS サービス用の既存の BIND サーバーの設定

Red Hat OpenStack Platform (RHOSP) DNS サービス (指定) を既存の BIND インフラストラクチャーと統合する場合、BIND 9 が正しく設定されていることを確認するために実行する必要があるアクションがいくつかあります。



重要

この機能は、本リリースでは [テクノロジープレビュー](#) として提供しているため、Red Hat では全面的にはサポートしていません。これは、テスト用途にのみご利用いただく機能です。実稼働環境にはデプロイしないでください。テクノロジープレビュー機能の詳細は、[対象範囲の詳細](#) を参照してください。



注記

既存の BIND インフラストラクチャーがない場合は、RHOSP director が BIND を自動的に設定します。

前提条件

- BIND 9 サーバーに変更を加えるには、適切な権限を持つユーザーである必要があります。
- BIND が `/etc/rndc.conf` と `/etc/rndc.key` というファイルにアクセスできることを確認します。

- BIND が RHOSP DNS サービス (designate) から **rndc** ユーティリティーメッセージを受信できることを確認します。

手順

1. BIND9 サーバーにログオンします。
2. **etc/rndc.key** が正しく設定されていることを確認します。
rndc-key は、HMAC(Hash-based Message Authentication Code)、SHA-256 アルゴリズム、Base64 エンコードされたシークレットを持つ必要があります。

```
key "rndc-key" {  
    algorithm hmac-sha256;  
    secret "<base64-encoded string>";  
};
```

3. まだの場合は、BIND が **rndc** ユーティリティーを使用してリモートでゾーンを作成および削除できるようにします。

/etc/named.conf の **options {** の下に、以下の行があることを確認します。ない場合は、新規に行を作成して追加してください。

```
allow-new-zones yes;
```

4. まだの場合は、最小限のレスポンスを送信するように BIND を設定します。
また、**/etc/named.conf** の **options {** に、次の行があることを確認します。ない場合は、新規に行を作成して追加してください。

```
minimal-responses yes;
```

デフォルトでは、BIND 9 はクライアントに送信するレスポンスに、権限アウトオブゾーンレコードと追加セクションを含みます。**minimal-responses** を **yes** に設定すると、out-of-zone の additional が処理されなくなり、DNS キャッシュポイズニング攻撃の影響を受けなくなります。

関連情報

- [既存の BIND 9 サーバーを使用した DNS サービスのデプロイメント](#)

2.4. 推奨される DNS サービストポロジー

推奨されるトポロジーには、Red Hat OpenStack Platform (RHOSP) コントローラーホストへの DNS サービスのデプロイメントが含まれます。RHOSP デプロイメントの可用性が高い場合は、少なくとも 3 つの RHOSP コントローラーがあり、それぞれに DNS サービスが含まれています。

図2.1 推奨される DNS サービストポロジー

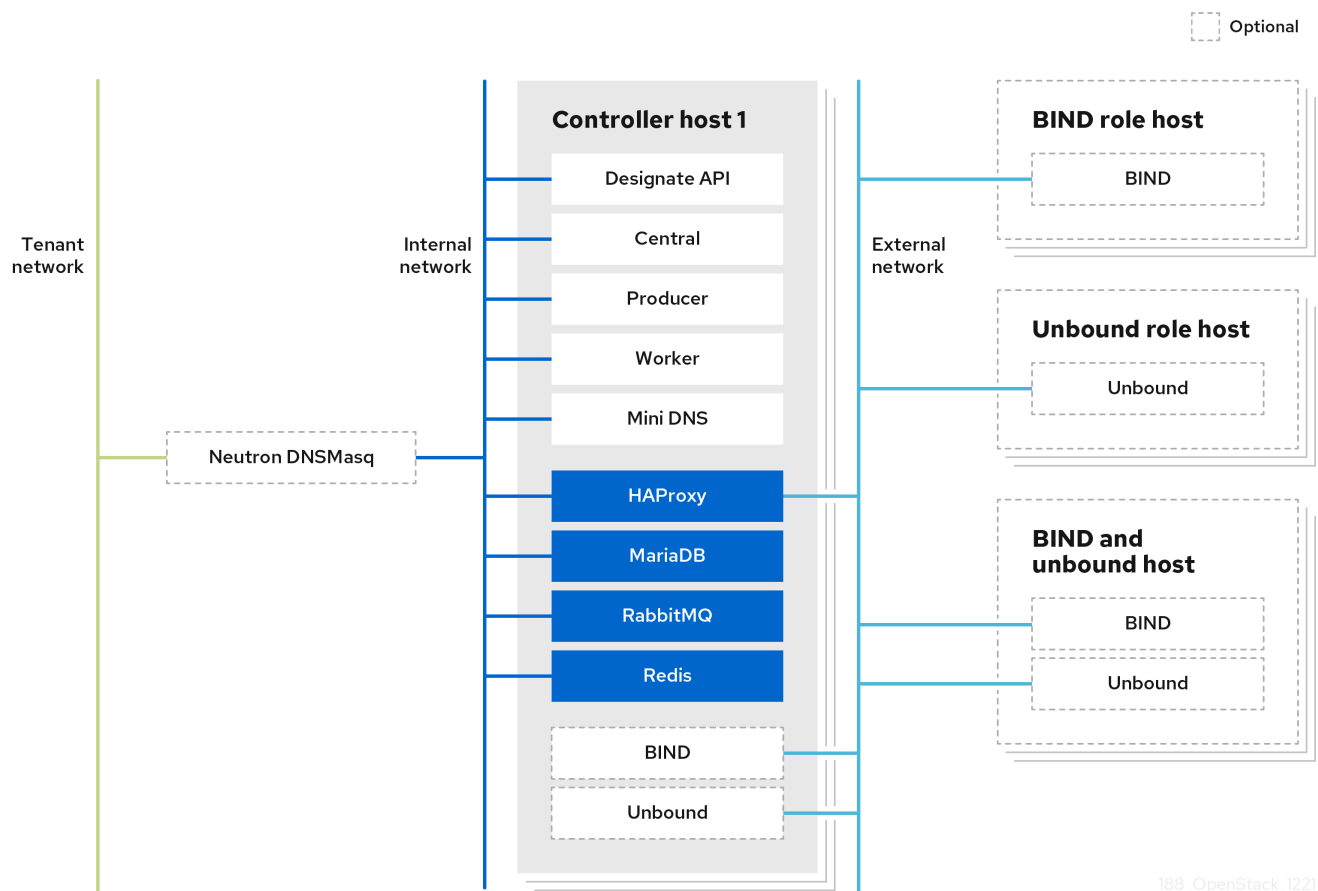


図 2.1では、DNS サービスコンポーネントはそれぞれのコンテナで実行されています。色が濃いコンテナは、DNS サービスが他の RHOSP サービスと共有するリソースです。

点線のコンテナは BIND および Unbound のオプションの配置を表します。サイトのデータトラフィックのフットプリントが大きい場合は、専用のホストを使用して、それぞれ BIND と Unbound を含めることができます。

2.5. DNS サービスの高可用性について

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) は、データトラフィックのロードバランシングとフォールトトレランスを **アクティブ-アクティブ** 高可用性モードと呼ばれる高可用性モードで組み合わせます。アクティブ/アクティブモードでは、DNS サービスは3つ以上のノードでコンポーネントサービスを同時に実行します。ノードの1つに障害が発生した場合、残りのノードは引き続き実行され、中断やパフォーマンスの低下を回避します。DNS サービスは、すべてのサービスインスタンス間で作業の負荷を分散しようとします。

DNS サービスコンポーネントは、RHOSP コントローラーロールでデプロイメントされるサービスとして分類されます。これは、RHOSP のインストールおよび設定ツールセットである director が、すべてのコントローラーホストに DNS サービスを自動的にデプロイすることを意味します。したがって、3つ以上の異なるホストに3つ以上の Controller をデプロイすると、DNS サービスの可用性が高くなります。

関連情報

- [DNS サービスコンポーネント](#)

第3章 DNS サービスのインストールと設定

Red Hat OpenStack Platform (RHOSP) をデプロイまたは再デプロイするときに、designate 環境ファイルを含めることにより、DNS サービス (designate) をインストールして設定します。RHOSP をデプロイメントするためのツールセットである director は、オーケストレーションサービス (heat) 環境テンプレートと環境ファイルを、DNS サービスと RHOSP デプロイメントの残りの部分をインストールおよび設定する方法の一連の計画として使用します。

DNS サービスをデプロイメントする場合、director は、DNS サービスをアクティブ - アクティブ高可用性モードで有効にしたり、ポートおよびフローティング IP アドレスの自動化をアクティブにしたりするなどのアクションを自動的に実行します。また、Director は、Networking サービス (neutron) を設定して、DNS サービスに含まれる Unbound リゾルバーを指すようにします。



注記

カスタム Heat 環境ファイルで **UnboundForwardResolvers** を設定することにより、Unbound リゾルバーの設定を明示的に無効にすることができます。

また、director に必要な DNS サーバー情報を提供することで、DNS サービスを既存の DNS インフラストラクチャーと統合することもできます。



重要

RHOSP 17.1 では、DNS サービスを既存の DNS インフラストラクチャーと統合することは、テクノロジープレビュー機能です。

このセクションに含まれるトピックは次のとおりです。

- [「DNS サービスのデプロイ」](#)
- [「既存の BIND 9 サーバーを使用した DNS サービスのデプロイメント」](#)
- [「DNS サービスのデフォルト設定の変更」](#)

3.1. DNS サービスのデプロイ

Red Hat OpenStack Platform (RHOSP) director を使用して、DNS サービス (designate) をデプロイします。Director は、RHOSP デプロイメントの一連のプランであるオーケストレーションサービス (heat) テンプレートと環境ファイルを使用します。アンダークラウドはこれらの計画をインポートし、その指示に従って DNS サービスと RHOSP デプロイメントをインストールおよび設定します。

前提条件

- RHOSP アンダークラウドにアクセスできる **stack** ユーザーである必要がある。

手順

1. DNS サーバーを既存の DNS インフラストラクチャーと統合する場合は、次のトピックに移動してください。 [「既存の BIND 9 サーバーを使用した DNS サービスのデプロイメント」](#)
2. アンダークラウドホストに stack ユーザーとしてログインします。
3. source コマンドでアンダークラウドの認証情報ファイルを読み込みます。



```
$ source ~/stackrc
```

4. **DesignateBindNSRecords** パラメーターの宣言を含むカスタム環境 YAML ファイルを作成します。このパラメーターの値には、DNS サーバー (designate) プールに存在する子ゾーンのネームサーバーレコード (NS レコード) を指定します。

```
parameter_defaults:
  DesignateBindNSRecords: ['<NS_record_child-zone-1>', '<NS_record_child-zone-2>', '...']
```

例

この例では、DNS プールに親ゾーン **example.org** の子ゾーン **ns1.sales.example.org.**、**ns2.sales.example.org.**、および **ns3.sales.example.org.** が含まれています。

```
parameter_defaults:
  DesignateBindNSRecords: ['ns1.sales.example.org.', 'ns2.sales.example.org.',
  'ns3.sales.example.org.']
```

5. デプロイコマンドを実行します。コア Heat テンプレート、その他の環境ファイル、**designate.yaml** 環境ファイル、およびプール NS レコードを含むファイルをコマンドに追加します。

例

```
$ openstack overcloud deploy --templates \
-e <other_environment_files> \
-e /usr/share/openstack-tripleo-heat-templates/environments/\
services/designate.yaml \
-e /home/stack/my_pool_ns_records.yaml
```



注記

Director は、スタックの更新またはアップグレード中に、さまざまな DNS サービスコンポーネントを最新の指定イメージに更新します。

検証

- DNS サービスがインストールされ、エンドポイントが定義されていることを確認します。

```
$ openstack endpoint list -c "Service Name" -c Enabled -c URL
```

出力例

```
+-----+-----+-----+
| Service Name | Enabled | URL |
+-----+-----+-----+
| swift      | True   | http://198.51.100.61:8080 |
| designate  | True   | http://203.0.113.103:9001 |
| heat-cfn   | True   | http://192.0.2.137:8000/v1 |
| designate  | True   | http://192.0.2.137:9001 |
| placement  | True   | http://203.0.113.103:8778/placement |
| cinderv3   | True   | http://203.0.113.103:8776/v3/%(tenant_id)s |
```



```

| heat      | True | http://203.0.113.103:8004/v1/%(tenant_id)s |
| heat-cfn  | True | http://203.0.113.103:8000/v1 |
| nova      | True | http://203.0.113.103:8774/v2.1 |
| heat      | True | http://192.0.2.137:8004/v1/%(tenant_id)s |
| glance    | True | http://203.0.113.103:9292 |
| heat      | True | http://203.0.113.103:8004/v1/%(tenant_id)s |
| glance    | True | http://203.0.113.103:9292 |
| neutron   | True | http://203.0.113.103:9696 |
| nova      | True | http://192.0.2.137:8774/v2.1 |
| cinderv3  | True | http://192.0.2.137:8776/v3/%(tenant_id)s |
| placement | True | http://203.0.113.103:8778/placement |
| keystone  | True | http://192.168.24.17:35357 |
| neutron   | True | http://192.0.2.137:9696 |
| nova      | True | http://203.0.113.103:8774/v2.1 |
| heat-cfn  | True | http://203.0.113.103:8000/v1 |
| cinderv3  | True | http://203.0.113.103:8776/v3/%(tenant_id)s |
| glance    | True | http://192.0.2.137:9292 |
| placement | True | http://192.0.2.137:8778/placement |
| swift     | True | http://198.51.100.61:8080/v1/AUTH_%(tenant_id)s |
| swift     | True | http://192.0.2.137:8080/v1/AUTH_%(tenant_id)s |
| designate | True | http://203.0.113.103:9001 |
| keystone  | True | http://192.0.2.137:5000 |
| neutron   | True | http://203.0.113.103:9696 |
| keystone  | True | http://203.0.113.103:5000 |
+-----+-----+-----+

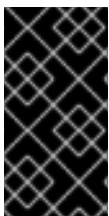
```

関連情報

- [director を使用した Red Hat OpenStack Platform のインストールと管理ガイドの デプロイメントコマンドのオプション](#)

3.2. 既存の BIND 9 サーバーを使用した DNS サービスのデプロイメント

Red Hat OpenStack Platform (RHOSP) director を使用して DNS サービス (designate) をインストールおよび設定し、それを既存の BIND 9 DNS インフラストラクチャーと統合します。Director は、RHOSP デプロイメントの一連のプランであるオーケストレーションサービス (heat) テンプレートと環境ファイルを使用します。DNS サーバーに関する特定の情報を heat 環境ファイルに追加します。アンダークラウドはこれらの計画をインポートし、その指示に従って RHOSP と DNS サービスをインストールして設定し、DNS インフラストラクチャーと統合します。



重要

この機能は、本リリースでは [テクノロジープレビュー](#) として提供しているため、Red Hat では全面的にはサポートしていません。これは、テスト用途にのみご利用いただく機能です。実稼働環境にはデプロイしないでください。テクノロジープレビュー機能の詳細は、[対象範囲の詳細](#) を参照してください。

前提条件

- BIND 9 サーバーに依存する既存の DNS インフラストラクチャーがあります。
- BIND 9 サーバーが [DNS サービス用の既存の BIND サーバーの設定](#) で説明されている設定を満たしていることを確認します。
- RHOSP アンダークラウドにアクセスできる **stack** ユーザーである必要がある。

手順

1. DNS サーバーを既存の DNS インフラストラクチャーと統合しない場合は、次のトピックに進んでください。「[DNS サービスのデプロイ](#)」
2. アンダークラウドホストに stack ユーザーとしてログインします。
3. source コマンドでアンダークラウドの認証情報ファイルを読み込みます。

```
$ source ~/stackrc
```

4. カスタム環境 YAML ファイルを作成します。

例

```
$ vi /home/stack/templates/my-designate-environment.yaml
```

5. 環境ファイルには、キーワード **parameter_defaults** および **DesignateExternalBindServers** が含まれている必要があります。**DesignateExternalBindServers** の下の新しい行に、BIND 9 DNS サーバーごとに IP アドレスと Remote Name Daemon Control (RNDC) キーを追加します。

例

この例では、2つの既存の BIND 9 サーバー **203.0.113.3** と **203.0.113.4** があり、それぞれ RNDC キーがあります。

```
parameter_defaults:
  DesignateExternalBindServers:
    - host: 203.0.113.3
      rndc_key: "FJOdVqZr5gVXbU9klagY0IJVDq7CV/mDVb/M7mILMgY="
    - host: 203.0.113.4
      rndc_key: "QAAACCdIV3KXPJh6U71ImVH0+j4uKRpVV49zVU7A8uvm"
```

6. **DesignateBindNSRecords** パラメーターの宣言を追加します。このパラメーターの値には、DNS サーバー (designate) プールに存在する子ゾーンのネームサーバーレコード (NS レコード) を指定します。

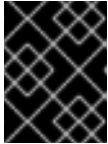
```
parameter_defaults:
  ...
  DesignateBindNSRecords: ['<NS_record_child-zone-1>', '<NS_record_child-zone-2>', '...']
```

例

この例では、DNS プールに親ゾーン **example.org** の子ゾーン **ns1.sales.example.org.**、**ns2.sales.example.org.**、および **ns3.sales.example.org.** が含まれています。

```
parameter_defaults:
  ...
  DesignateBindNSRecords: ['ns1.sales.example.org.', 'ns2.sales.example.org.',
    'ns3.sales.example.org.']
```

7. デプロイコマンドを実行し、コア Heat テンプレート、その他の環境ファイル、**designate.yaml** 環境ファイル、およびこの新しいカスタム環境ファイルを含めます。



重要

後で実行される環境ファイルで定義されているパラメーターとリソースが優先されることになるため、環境ファイルの順序は重要となります。

例

```
$ openstack overcloud deploy --templates \  
-e <other_environment_files> \  
-e /usr/share/openstack-tripleo-heat-templates/environments/  
services/designate.yaml
```



注記

Director は、スタックの更新またはアップグレード中に、さまざまな DNS サービスコンポーネントを最新の指定イメージに更新します。

関連情報

- [director を使用した Red Hat OpenStack Platform のインストールと管理ガイドの デプロイメントコマンドのオプション](#)
- [Red Hat OpenStack Platform デプロイメントのカスタマイズガイドの 環境ファイル](#)
- [Red Hat OpenStack Platform デプロイメントのカスタマイズガイドの オーバークラウド作成に環境ファイルを含める](#)

3.3. DNS サービスのデフォルト設定の変更

YAML 形式の環境ファイルを変更し、RHOSP オーバークラウドを再デプロイすることで、Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) の設定を変更します。RHOSP ディレクターは、オーケストレーションサービス (heat) テンプレートと環境ファイルを DNS サービスを設定するための計画として使用するツールセットです。

前提条件

- RHOSP アンダークラウドにアクセスできる **stack** ユーザーである必要がある。
- 変更する RHOSP DNS サービスパラメーターを決定します。
以下にいくつか例を示します。
 - **DesignateRpcResponseTimeout**
DNS サービスの RPC 応答タイムアウト (秒単位)。デフォルトは 60 秒です。
 - **DesignateWorkers**
Designate サービスのワーカー数。デフォルトはゼロ (0) です。これは、デプロイメントスクリプトがオペレーティングシステムワーカーに RHOSP ディレクターの値を使用することを意味します。

詳細は、[director を使用した Red Hat OpenStack Platform のインストールと管理ガイドの 環境規模の判断](#) を参照してください。

- **DesignateMdnsProxyBasePort**

外部またはパブリックアクセスネットワーク上の MiniDNS プロキシエンドポイントのベースポート。デフォルトのポートは 16000 です。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. `source` コマンドでアンダークラウドの認証情報ファイルを読み込みます。

```
$ source ~/stackrc
```

3. カスタム YAML 環境ファイルを作成します。

例

```
$ vi /home/stack/templates/my-designate-environment.yaml
```

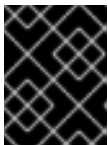
ご自分の環境ファイルには、**parameter_defaults** というキーワードを含める必要があります。**parameter_defaults** のキーワードの後に、ご自分のパラメーター/値ペアを定義します。

例

この例では、RPC 応答タイムアウトが 120 秒に設定されています。

```
parameter_defaults:
  DesignateRpcResponseTimeout: '120'
```

4. デプロイコマンドを実行し、コア Heat テンプレート、その他の環境ファイル、**designate.yaml** 環境ファイル、およびこの新しいカスタム環境ファイルを含めます。



重要

後で実行される環境ファイルで定義されているパラメーターとリソースが優先されることになるため、環境ファイルの順序は重要となります。

例

```
$ openstack overcloud deploy --templates \
-e <other_environment_files> \
-e /usr/share/openstack-tripleo-heat-templates/environments/\
services/designate.yaml \
-e /home/stack/templates/my-designate-environment.yaml
```

関連情報

- [オーバークラウドパラメーター ガイドの DNS \(designate\) パラメーター](#)
- [Red Hat OpenStack Platform デプロイメントのカスタマイズガイドの 環境ファイル](#)
- [Red Hat OpenStack Platform デプロイメントのカスタマイズガイドの オーバークラウド作成に環境ファイルを含める](#)

第4章 統合 DNS サービスの使用

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) は Networking サービス (neutron) と統合され、ポートの自動レコードセット作成と Compute サービス (nova) を通じて、仮想マシンインスタンスを提供します。

クラウド管理者は、DNS サービスを使用して、ネットワークに関連付けられたゾーンを作成します。クラウド管理者がこのネットワークを使用することで、クラウドユーザーは仮想マシンインスタンス、ポート、または Floating IP を作成し、DNS サービスは必要な DNS レコードを自動的に作成します。

DNS サービスのデプロイメント時に、インストールツールセットの RHOSP director は、Networking サービス (neutron) の拡張機能 **dns_domain_ports** を読み込みます。この拡張機能により、以下の DNS 属性を RHOSP ポート、ネットワーク、および Floating IP に追加できます。

表4.1 RHOSP ネットワークおよび DNS サービスでサポートされる DNS 設定

リソース	DNS 名	DNS ドメイン (ゾーン)
ポート	はい	はい
Networks	いいえ	はい
Floating IP	はい	はい



注記

ネットワークと Floating IP の両方で指定された DNS ドメインの場合、Floating IP のポートのドメインがネットワークで設定されたドメインよりも優先されます。



重要

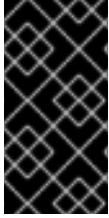
Red Hat OpenStack Platform (RHOSP) 17.1 GA では、RHOSP Networking サービス (neutron) ML2/OVN と RHOSP DNS サービス (designate) を統合するためのテクノロジープレビューを利用できます。その結果、DNS サービスは、新しく作成された VM の DNS エントリーを自動的に追加しません。

このセクションに含まれるトピックは次のとおりです。

- [「Setting up a project for DNS integration」](#)
- [「仮想マシンインスタンスの DNS との統合」](#)
- [「DNS とポートの統合」](#)
- [「Floating IP と DNS の統合」](#)

4.1. SETTING UP A PROJECT FOR DNS INTEGRATION

クラウド管理者は、必須のゾーン、ネットワーク、およびサブネットを作成し、クラウドユーザーは、仮想マシンインスタンス、ポート、または Floating IP の作成時に指定する必要があります。RHOSP Networking サービス (neutron) は DNS サービス (設計) と統合されているため、クラウドユーザーがこれらのオブジェクトを作成すると、自動的に DNS サービスに追加されます。



重要

この機能は、本リリースでは **テクノロジープレビュー** として提供しているため、Red Hat では全面的にはサポートしていません。これは、テスト用途にのみご利用いただく機能です。実稼働環境にはデプロイしないでください。テクノロジープレビュー機能の詳細は、[対象範囲の詳細](#) を参照してください。

前提条件

- **admin** ロールを持つ RHOSP ユーザーである。
- ポートや VM に使用するネットワークは、**router:external** 属性を **True** に設定することはできません。ネットワークの作成時に、**--external** オプションは指定しないでください。
- ネットワークは、FLAT、VLAN、GRE、VXLAN、または GENEVE のタイプのいずれかでなければなりません。
- VLAN、GRE、VXLAN、または GENEVE ネットワークの場合、セグメント ID は Networking サービスの **ml2_conf.ini** ファイルで設定した範囲外である必要があります。**ml2_conf.ini** ファイルは、コントローラーノードホストの **/var/lib/config-data/puppet-generated/neutron/etc/neutron/plugins/ml2** にあります。+ 次の表を使用して、どのセクションとオプションを参照するかを決定してください。ネットワークセグメンテーション ID 範囲:

表4.2 ネットワークのセグメント ID の設定に使用される **ml2_conf.ini** オプション

ネットワークの種類	セクション	オプション
Geneve	[ml2_type_geneve]	vni_ranges
GRE	[ml2_type_gre]	tunnel_id_ranges
VLAN	[ml2_type_vlan]	network_vlan_ranges
VXLAN	[ml2_type_vxlan]	vni_ranges



注記

これらの前提条件が満たされていない場合は、Networking サービスはデフォルトの **dns_domain** 値 **openstacklocal** を使用して内部リゾルバーに DNS 割り当てを作成しません。

手順

1. クラウド管理者として、認証情報ファイルを入手します。

例

```
$ source ~/overcloudrc
```

2. 特定のプロジェクトのユーザーが DNS エントリーを作成できるようにゾーンを作成します。

例

以下の例では、クラウド管理者は **example.com**. という名前のゾーンを作成し、プロジェクト ID (**f75ec24a-d361-ab86-54c0-dfe6093245a3**) に含まれるユーザー ID に、そのゾーンにレコードセットを追加するパーミッションを割り当てます。

```
$ openstack zone create --email example@example.com example.com. --sudo-project-id f75ec24a-d361-ab86-54c0-dfe6093245a3
```



注記

DNS ドメインは、常にピリオドで終わる、完全修飾ドメイン名 (FQDN) である必要があります。

3. 特定のプロジェクトのユーザーが DNS エントリーを作成できるようにネットワークを作成します。

例

以下の例では、クラウド管理者は以前に作成したゾーン **example.com** を使用してネットワーク **example-network** と セグメント ID **2017** (ml2_conf.ini で定義された範囲外) を作成します。

```
$ openstack network create --dns-domain example.com. \
--provider-segment 2017 --provider-network-type geneve \
example-network
```

4. ネットワーク上にサブネットを作成します。

例

以下の例では、クラウド管理者はサブネット **example-subnet** をネットワーク **example-network** に作成します。

```
$ openstack subnet create \
--allocation-pool start=192.0.2.10,end=192.0.2.200 \
--network example-network \
--subnet-range 192.0.2.0/24 \
example-subnet
```

5. インスタンス、ポート、および Floating IP の追加時に作成したゾーンおよびネットワークを使用するように、プロジェクト内のクラウドユーザーに指示します。



警告

インスタンス、ポート、または Floating IP を作成するユーザーにゾーンにレコードセットを作成するパーミッションがない場合や、ゾーンが DNS サービスに存在しない場合には、Networking サービスは次の処理を行います。

- 指定した **dns_domain** を使用して **dns_assignment** フィールドに入力された値でポートを作成します。
- DNS サービスにレコードセットを作成しません。
- Error publishing port data in external DNS service のエラーをログに記録します。

検証

- 作成したネットワークが存在することを確認します。

例

```
$ openstack network show example-network
```

出力例

```
+-----+-----+
| Field          | Value                               |
+-----+-----+
| admin_state_up | UP                                   |
| availability_zone_hints |                                     |
| availability_zones |                                     |
| created_at     | 2022-09-07T19:03:32Z               |
| description    |                                     |
| dns_domain     | example.com.                        |
| id             | 9ae5b3d5-f12c-4a67-b0e5-655d53cd4f7c |
| ipv4_address_scope | None                                 |
| ipv6_address_scope | None                                 |
| is_default     | None                                 |
| is_vlan_transparent | None                                 |
| mtu            | 1450                                 |
| name           | network-example                     |
| port_security_enabled | True                                 |
| project_id     | f75ec24a-d361-ab86-54c0-dfe6093245a3 |
| provider:network_type | vxlan                               |
| provider:physical_network | None                                 |
| provider:segmentation_id | 2017                                |
| qos_policy_id  | None                                 |
| revision_number | 3                                    |
| router:external | Internal                             |
| segments      | None                                 |
| shared        | False                                |
| status        | ACTIVE                               |
```



```
| subnets          | 15546c9d-6faf-43aa-83e7-b1e705eed060 |
| tags              |                                         |
| updated_at       | 2022-09-07T19:03:43Z                 |
+-----+-----+-----+-----+-----+
```

関連情報

- コマンドラインインターフェイスリファレンスの [zone](#)
- コマンドラインインターフェイスリファレンスの [network](#)
- コマンドラインインターフェイスリファレンスの [subnet](#)

4.2. 仮想マシンインスタンスの DNS との統合

Networking サービス (neutron) と DNS サービス (設計) 間の統合により、仮想マシンインスタンスを作成するたびに DNS を自動的に有効にできます。

前提条件

- クラウド管理者が、DNS 対応インスタンスの作成時に、使用するために必要なネットワークが用意する。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. クラウド管理者が提供したネットワークを使用して、インスタンスを作成します。

例

以下の例では、クラウドユーザーは **my_vm** という名前のインスタンスを作成します。

```
$ openstack server create --image cirros-0.5.2-x86_64-disk --flavor m1.micro --nic net-
id=example-network my_vm
```

検証

- 作成したインスタンスの DNS サービスにレコードが存在することを確認します。

例

この例では、DNS サービスは **example.com**. ゾーンに対してクエリーされます。

```
$ openstack recordset list --type A example.com.
```

出力例

```
+-----+-----+-----+-----+-----+-----+
```

id	name	type	records	status	action
7b8d1be6-1b23	my_vm.example.com.	A	192.0.2.44	ACTIVE	NONE
-478a-94d5-60					
b876dca2c8					

関連情報

- コマンドラインインターフェイスリファレンスの [server create](#)

4.3. DNS とポートの統合

Networking サービス (neutron) と DNS サービス (設計) 間の統合により、ポートの作成時には常に DNS レコードセットを自動的に追加できます。

前提条件

- クラウド管理者が、DNS 対応のポートの作成時に使用するために必要なネットワークを用意する。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. クラウド管理者が提供したゾーンおよびネットワークを使用して、ポートを作成します。

例

以下の例では、クラウドユーザーは、**example-network** ネットワークに、DNS 名に **example-port** を指定してポート **my-port** を作成します。

```
$ openstack port create --network example-network \
--dns-name example-port \
my-port
```

検証

- 作成したポートのレコードが DNS サービスに存在することを確認します。

例

この例では、DNS サービスは **example.com.** ゾーンに対してクエリーされます。

```
$ openstack recordset list --type A example.com.
```

出力例

```
+-----+-----+-----+-----+-----+-----+
```

id	name	type	records	status	action
9ebbe94f-2442	example-port.example.com.	A	192.0.2.149	ACTIVE	NONE
-4bb8-9cfa-6d					
ca1daba73f					

関連情報

- コマンドラインインターフェイスリファレンスの [port create](#)

4.4. FLOATING IP と DNS の統合

Networking サービス (neutron) と DNS サービス (設計) 間の統合により、Floating IP の作成時には常に DNS レコードセットを自動的に追加できます。

前提条件

- クラウド管理者が、DNS 対応の Floating IP の作成時に使用するために必要な外部ネットワークを用意する。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. クラウド管理者が提供したゾーンと外部ネットワークを使用して、フローティング IP を作成します。

例

この例では、クラウドユーザーがネットワーク **public** に **example-fip** という DNS 名でフローティング IP を作成します。

```
$ openstack floating ip create --dns-name example-fip \
--dns-domain example.com. \
public
```

検証

- 作成した Floating IP の DNS サービスにレコードが存在することを確認します。

例

この例では、DNS サービスは **example.com.** ゾーンに対してクエリーされます。

```
$ openstack recordset list --type A example.com.
```

出力例

```
+-----+-----+-----+-----+-----+
| id      | name                | type | records | status | action |
+-----+-----+-----+-----+-----+
| e1eca823-169d | example-fip.example.com. | A    | 192.0.2.106 | ACTIVE | NONE |
| -4d0a-975e-91 |                       |      |             |       |      |
| a9907ec0c1   |                       |      |             |       |      |
+-----+-----+-----+-----+-----+
```

関連情報

- コマンドラインインターフェイスリファレンスの [floating ip create](#)

第5章 トップレベルドメイン名の管理

このセクションでは、最上位ドメインを紹介し、Red Hat OpenStack Platform DNS サービス (designate) でそれらを作成および管理する方法について説明します。ユーザーが作成できるドメイン名は、拒否リストを使用して管理します。

このセクションに含まれるトピックは次のとおりです。

- [「トップレベルドメインについて」](#)
- [「トップレベルドメインの作成」](#)
- [「トップレベルドメインのリスト表示および表示」](#)
- [「トップレベルドメインの変更」](#)
- [「トップレベルドメインの削除」](#)
- [「DNS サービスの拒否リストについて」](#)
- [「拒否リストの DNS サービス正規表現について」](#)
- [「DNS サービスの拒否リストの作成」](#)
- [「DNS サービスの拒否リストのリスト表示および表示」](#)
- [「DNS サービスの拒否リストの変更」](#)
- [「DNS サービスの拒否リストの削除」](#)

5.1. トップレベルドメインについて

トップレベルドメイン (TLD) を使用して、ユーザーがゾーンを作成できるドメインを制限できます。Domain Name System (DNS) では、**TLD** という用語は、ルートの下にある **.com** などのドメインのセットを具体的に指します。Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) では、TLD は任意の有効なドメインにすることができます。

TLD は許可されるドメインのセットを定義するため、ユーザーが作成するゾーンはいずれかの TLD 内に存在する必要があります。DNS サービス内に TLD が作成されていない場合は、任意のゾーンを作成できます。TLD には、特権ユーザーに許可された TLD の外部にゾーンを作成することを許可するポリシーはありません。

例

.com TLD の作成後、ユーザーが **.com** TLD の範囲外にゾーンを作成しようとすると、失敗します。

```
$ openstack zone create --email admin@test.net test.net.
```

出力例

```
Invalid TLD
```

OpenStack Client **openstack tld** コマンドを使用して、TLD を作成、リスト表示、表示、変更、削除できます。

関連情報

- コマンドラインインターフェイスリファレンスの [tld](#)
- コマンドラインインターフェイスリファレンスの [zone](#)

5.2. トップレベルドメインの作成

トップレベルドメイン (TLD) を使用すると、ユーザーがゾーンを作成できるドメインを制限できます。Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) では、TLD は任意の有効なドメインにすることができます。TLD を作成するには、OpenStack Client **openstack tld create** コマンドを使用します。

前提条件

- **admin** ロールを持つ RHOSP ユーザーである。

手順

1. クラウド管理者として、認証情報ファイルを入手します。

例

```
$ source ~/overcloudrc
```

2. **openstack tld create** コマンドを実行して TLD を作成します。

例

たとえば、ユーザーに **.org** で終わるゾーンを作成するように要求する場合は、単一の **.org** TLD を作成できます。

```
$ openstack tld create --name org
```

出力例

```
+-----+-----+
| Field  | Value                               |
+-----+-----+
| created_at | 2022-01-10T13:07:33.000000         |
| description | None                               |
| id        | 9fd0a12d-511e-4024-bf76-6ec2e3e71edd |
| name      | org                                 |
| updated_at | None                               |
+-----+-----+
```

ヒント

openstack tld command を使用する場合は、入力した完全修飾ドメイン名 (FQDN) に末尾のドットがないことを確認します (**.net.** など)。

検証

- **openstack tld list** コマンドを実行し、TLD が存在することを確認します。

例

```
$ openstack tld list --name zone1.cloud.example.com
```

関連情報

- [Command Line Interface Reference](#) の [tld create](#)

5.3. トップレベルドメインのリスト表示および表示

Red Hat OpenStack Platform DNS サービス (designate) データベースにクエリーを実行して、すべてのトップレベルドメイン (TLD) をリスト表示したり、特定の TLD のプロパティを表示したりできます。これを実行するための OpenStack Client コマンドは、それぞれ **openstack tld list** と **openstack tld show** です。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. 次のコマンドを使用して、DNS サービスデータベース内のすべての TLD をリスト表示します。

```
$ openstack tld list
```

3. **openstack tld show <TLD_NAME_or_ID>** コマンドを使用して、特定の TLD のプロパティを表示します。

例

```
$ openstack tld show org
```

関連情報

- [コマンドラインインターフェイスリファレンス](#) の [tld list](#)
- [コマンドラインインターフェイスリファレンス](#) の [tld show](#)

5.4. トップレベルドメインの変更

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) を使用すると、名前などのトップレベルドメイン (TLD) のさまざまなプロパティを変更することができます。OpenStack Client **openstack tld set** コマンドを使用して TLD を変更します。

前提条件

- **admin** ロールを持つ RHOSP ユーザーである。

手順

1. クラウド管理者として、認証情報ファイルを入手します。

例

```
$ source ~/overcloudrc
```

2. 次のコマンドオプションを使用して、さまざまな方法で TLD を変更できます。

```
openstack tld set [--name NAME] \  
  [--description DESCRIPTION | --no-description] \  
  [TLD_ID | TLD_NAME]
```



注記

上記の構文ダイアグラムは、**openstack tld set** コマンドのさまざまなフォーマットオプションを示していません。すべてのコマンドオプションのリストについては、後で示す関連情報を参照してください。

この例では、**openstack tld set** コマンドで **org** TLD の名前を **example.net** に変更します。

例

```
$ openstack tld set org --name example.net
```

出力例

```
+-----+-----+  
| Field  | Value                               |  
+-----+-----+  
| created_at | 2022-01-10T13:07:33.000000          |  
| description |                                     |  
| id        | 9fd0a12d-511e-4024-bf76-6ec2e3e71edd |  
| name      | example.net                         |  
| updated_at | 2022-01-10T22:35:20.000000          |  
+-----+-----+
```

検証

- **openstack tld show <TLD_NAME_or_ID>** コマンドを実行し、変更が存在することを確認します。

関連情報

- コマンドラインインターフェイスリファレンスの [tld set](#)

5.5. トップレベルドメインの削除

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) では、OpenStack Client **openstack tld delete** コマンドを使用してトップレベルドメイン (TLD) を削除できます。

#####

前提条件

- **admin** ロールを持つ RHOSP ユーザーである。

手順

1. クラウド管理者として、認証情報ファイルを入手します。

例

```
$ source ~/overcloudrc
```

2. 以下のコマンドを実行して、削除する TLD の ID または名前を取得します。

```
$ openstack tld list
```

3. 前の手順で取得した名前または ID を使用して、次のコマンドを入力します。

```
$ openstack tld delete <TLD_NAME_or_ID>
```

このコマンドが正常に実行されると、出力はありません。

検証

- **openstack tld show <TLD_NAME_or_ID>** を実行し、TLD が削除されたことを確認します。

関連情報

- コマンドラインインターフェイスリファレンスの [tld delete](#)

5.6. DNS サービスの拒否リストについて

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) には拒否リスト機能があり、ユーザーが特定の正規表現に一致する名前でゾーンを作成できないようにします。たとえば、拒否リストを使用して、ユーザーが次のことを実行できないようにすることができます。

- 特定のゾーンの作成。
- 特定の文字列を含むゾーンの作成
- 特定ゾーンのサブゾーンの作成。

example.com. が拒否リストのメンバーで、ドメインまたはプロジェクトユーザーが **foo.example.com.** や **example.com.** などのゾーンを作成しようとすると、エラーが発生します。

```
$ openstack zone create --email admin@example.com example.com.
Blacklisted zone name
$ openstack zone create --email admin@example.com foo.example.com.
Blacklisted zone name
```



注記

`use_blacklisted_zone` のロールベースのアクセス制御を満たすユーザーは、拒否リストにある名前で作成できます。デフォルトでは、このオーバーライドを持つユーザーは RHOSP システム管理者のみです。

OpenStack Client `openstack zone blacklist` コマンドを使用して拒否リストを作成、リスト表示、表示、変更、および削除できます。

関連情報

- コマンドラインインターフェイスリファレンスの [zone blacklist create](#)

5.7. 拒否リストの DNS サービス正規表現について

Red Hat OpenStack Platform DNS サービス (designate) での拒否リストの操作の大部分は、使いにくい正規表現 (regexes) を使用しています。正規表現に関する Python ドキュメントは、入門書として役に立つかもしれませんが。オンライン正規表現ツールは、拒否リスト API で使用する正規表現の構築およびテストに役立ちます。

関連情報

- Python 3 ドキュメントの [Regular Expression HOWTO](#)
- 「DNS サービスの拒否リストについて」

5.8. DNS サービスの拒否リストの作成

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) の拒否リストを使用すると、ユーザーが特定の正規表現に一致する名前で作成できないようにすることができます。OpenStack Client `openstack zone blacklist create` コマンドを使用して拒否リストを作成します。

前提条件

- `admin` ロールを持つ RHOSP ユーザーである。

手順

1. クラウド管理者として、認証情報ファイルを入手します。

例

```
$ source ~/overcloudrc
```

2. `openstack zone blacklist create` コマンドを使用して拒否リストを作成します。この例では、ドメイン `example.com`、およびそのサブドメインのすべてが拒否リストに追加されます。

例

```
$ openstack zone blacklist create --pattern ".*example.com."
```

出力例

```

+-----+-----+
| Field   | Value                               |
+-----+-----+
| created_at | 2021-10-20T16:15:18.000000         |
| description | None                               |
| id        | 7622e241-8c3d-4c03-a692-8747e3cf2658 |
| pattern   | .*example.com.                    |
| updated_at | None                               |
+-----+-----+

```

検証

- **openstack zone blacklist list** コマンドを実行し、拒否リストが存在することを確認します。

関連情報

- コマンドラインインターフェイスリファレンスの [zone blacklist create](#)
- 「[拒否リストのDNS サービス正規表現について](#)」

5.9. DNS サービスの拒否リストのリスト表示および表示

Red Hat OpenStack Platform DNS サービス (designate) データベースにクエリーを実行して、すべての拒否リストを表示したり、特定の拒否リストのプロパティを表示したりできます。これを実行するための OpenStack Client コマンドは、それぞれ **openstack zone blacklist list** および **openstack zone blacklist show** です。

他の拒否リストコマンドを使用するためには拒否リスト ID を知っている必要があるため、拒否リストを表示すると便利です。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. 以下のコマンドを使用して、DNS サービスデータベース内の拒否リストをリスト表示します。

```
$ openstack zone blacklist list
```

- 前の手順で取得した拒否リスト ID を使用して、**openstack zone blacklist show <denylist_ID>** コマンドを使用して、特定の拒否リストのプロパティを表示します。

例

```
$ openstack zone blacklist show 7622e241-8c3d-4c03-a692-8747e3cf2658
```

関連情報

- コマンドラインインターフェイスリファレンスの [zone blacklist list](#)

- コマンドラインインターフェイスリファレンスの [zone blacklist show](#)

5.10. DNS サービスの拒否リストの変更

Red Hat OpenStack Platform DNS サービス (designate) を使用して、拒否リストを変更することができます。たとえば、拒否リストを変更して、ユーザーが以前は制限されていた特定のドメイン名でゾーンを作成できるようにする場合があります。拒否リストを変更するには、OpenStack Client **openstack zone blacklist set** コマンドを使用します。

前提条件

- **admin** ロールを持つ RHOSP ユーザーである。

手順

1. クラウド管理者として、認証情報ファイルを入手します。

例

```
$ source ~/overcloudrc
```

2. 以下のコマンドを実行して、変更する拒否リストの ID を取得します。

```
$ openstack zone blacklist list
```

3. 次のコマンドオプションを使用して、さまざまな方法で拒否リストを変更できます。

```
$ openstack zone blacklist set \
  [--description DESCRIPTION | --no-description] denylist_ID
```



注記

上記の構文ダイアグラムは、**openstack zone blacklist set** コマンドのさまざまなフォーマットオプションを示していません。すべてのコマンドオプションのリストについては、後で示す関連情報を参照してください。

この例では、**web.example.com** ドメインを許可するように正規表現 (regex) が変更されています。

例

```
$ openstack zone blacklist set 81fbfe02-6bf9-4812-a40e-1522ab6862ca --pattern
".*web.example.com"
```

出力例

```
+-----+-----+
| Field  | Value                               |
+-----+-----+
| created_at | 2022-01-08T09:11:43.000000          |
| description | None                               |
| id       | 81fbfe02-6bf9-4812-a40e-1522ab6862ca |
```

```
| pattern | .*web.example.com |
| updated_at | 2022-01-15T14:26:18.000000 |
+-----+-----+
```

検証

- **openstack zone blacklist show <denylist_ID>** コマンドを実行し、変更が存在することを確認します。

関連情報

- コマンドラインインターフェイスリファレンス で [設定されたゾーンブラックリスト](#)
- [「拒否リストの DNS サービス正規表現について」](#)

5.11. DNS サービスの拒否リストの削除

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) の拒否リストを使用すると、ユーザーが特定の正規表現に一致する名前で作成できないようにすることができます。OpenStack Client **openstack zone blacklist delete** コマンドを使用して拒否リストを削除します。

前提条件

- **admin** ロールを持つ RHOSP ユーザーである。

手順

1. クラウド管理者として、認証情報ファイルを入手します。

例

```
$ source ~/overcloudrc
```

2. 以下のコマンドを実行して、削除する拒否リストの ID を取得します。

```
$ openstack zone blacklist list
```

3. 前の手順で取得した ID を使用して、次のコマンドを入力します。

```
$ openstack zone blacklist delete <denylist_ID>
```

このコマンドが正常に実行されると、出力はありません。

検証

- **openstack zone blacklist show <denylist_ID>** コマンドを実行し、拒否リストが削除されたことを確認します。

関連情報

- コマンドラインインターフェイスリファレンスの [zone blacklist delete](#)

第6章 DNS リソースのクォータの表示および管理

Red Hat OpenStack Platform (RHOSP) は、クラウド管理者が DNS サービス (designate) を使用して変更できる一連の DNS リソースクォータを提供します。DNS クォータを使用すると、プロジェクトの DNS リソースに制限を設定することで、サービス拒否攻撃などのイベントから RHOSP サイトを保護するのに役立ちます。DNS クォータを使用すると、ユーザーの DNS リソース消費を追跡することもできます。クラウド管理者は、すべてのプロジェクトに適用される DNS クォータ値を設定するか、プロジェクトごとに1つ以上のクォータを設定できます。

このセクションに含まれるトピックは次のとおりです。

- [「DNS リソースのクォータの表示」](#)
- [「DNS リソースのクォータの変更」](#)
- [「DNS リソースクォータをデフォルト値にリセットする」](#)
- [「DNS サービスクォータとそのデフォルト値」](#)

6.1. DNS リソースのクォータの表示

DNS サービス (designate) を使用して、Red Hat OpenStack Platform (RHOSP) プロジェクトのリソースクォータを表示できます。

前提条件

- クォータを表示するプロジェクトのメンバーである必要があります。
- **admin** ロールを持つ RHOSP ユーザーは、任意のプロジェクトのクォータを表示できます。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. プロジェクトに設定されている DNS リソースクォータを表示します。

```
$ openstack dns quota list
```

出力例

```
+-----+
| Field      | Value |
+-----+
| api_export_size | 1000 |
| recordset_records | 20 |
| zone_records    | 500 |
| zone_recordsets | 500 |
| zones          | 10 |
+-----+
```

3. **admin** ロールを持つ RHOSP ユーザーは、他のプロジェクトのクォータを照会できます。

- a. クォータを変更するプロジェクトの ID を取得します。
ID は後のステップに必要なので、覚えておいてください。

```
$ openstack project list
```

- b. プロジェクト ID を使用して、プロジェクトに設定された DNS リソースクォータを表示します。

例

この例では、プロジェクト ID **ecd4341280d645e5959d32a4b7659da1** の DNS クォータが表示されます。

```
$ openstack dns quota list --project-id ecd4341280d645e5959d32a4b7659da1
```

出力例

```
+-----+-----+
| Field      | Value |
+-----+-----+
| api_export_size | 2500 |
| recordset_records | 25 |
| zone_records    | 750 |
| zone_recordsets | 750 |
| zones          | 25 |
+-----+-----+
```

関連情報

- コマンドラインインターフェイスリファレンスの [dns quota リスト](#)

6.2. DNS リソースのクォータの変更

DNS サービス (designate) を使用して、Red Hat OpenStack Platform (RHOSP) プロジェクトの DNS リソースクォータを変更できます。

前提条件

- **admin** ロールを持つ RHOSP ユーザーである。

手順

1. クラウド管理者として、認証情報ファイルを入手します。

例

```
$ source ~/overcloudrc
```

2. クォータを変更するプロジェクトの ID を取得します。
ID は後のステップに必要なので、覚えておいてください。

```
$ openstack project list
```

- プロジェクト ID を使用して、プロジェクトの DNS リソースクォータを変更します。使用可能なクォータのリストについては、次を参照してください。[「DNS サービスクォータとそのデフォルト値」](#)

例

この例では、**zones** クォータが変更されています。プロジェクト ID **ecd4341280d645e5959d32a4b7659da1** に含めることができるゾーンの総数は 30 です。

```
$ openstack dns quota set --project-id ecd4341280d645e5959d32a4b7659da1 --zones 30
```

出力例

```
+-----+-----+
| Field      | Value |
+-----+-----+
| api_export_size | 1000 |
| recordset_records | 20 |
| zone_records   | 500 |
| zone_recordsets | 500 |
| zones         | 30 |
+-----+-----+
```

関連情報

- コマンドラインインターフェイスリファレンス で [設定された dns クォータ](#)
- [「DNS サービスクォータとそのデフォルト値」](#)

6.3. DNS リソースクォータをデフォルト値にリセットする

DNS サービス (designate) を使用して、Red Hat OpenStack Platform (RHOSP) プロジェクトの DNS リソースクォータをデフォルト値にリセットできます。

前提条件

- **admin** ロールを持つ RHOSP ユーザーである。

手順

- クラウド管理者として、認証情報ファイルを入手します。

例

```
$ source ~/overcloudrc
```

- クォータをリセットするプロジェクトの ID を取得します。ID は後のステップに必要なので、覚えておいてください。

```
$ openstack project list
```


- プロジェクト ID を使用して、プロジェクトの DNS リソースクォータをリセットします。

例

この例では、ID が **ecd4341280d645e5959d32a4b7659da1** であるプロジェクトのクォータがデフォルト値にリセットされています。

```
$ openstack dns quota reset --project-id ecd4341280d645e5959d32a4b7659da1
```

成功した **openstack dns quota reset** コマンドからの出力はありません。

検証

- プロジェクトの DNS リソースクォータがリセットされたことを確認します。

例

```
$ openstack dns quota list --project-id ecd4341280d645e5959d32a4b7659da1
```

出力例

```
+-----+-----+
| Field      | Value |
+-----+-----+
| api_export_size | 1000 |
| recordset_records | 20 |
| zone_records    | 500 |
| zone_recordsets | 500 |
| zones          | 10 |
+-----+-----+
```

関連情報

- コマンドラインインターフェイスリファレンスの [dns quota reset](#)
- [「DNS サービスクォータとそのデフォルト値」](#)

6.4. DNS サービスクォータとそのデフォルト値

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) には、1つまたはすべての RHOSP プロジェクトでクラウドユーザーによる DNS リソースの消費を制限するために管理者が設定できるクォータがあります。

表6.1 ゾーンクォータ

クォータ	デフォルト	説明
zones	10	プロジェクトごとに許可されるゾーンの数。

表6.2 レコードおよびレコードセットクォータ

クォータ	デフォルト	説明
zone_recordsets	500	ゾーンごとに許可されるレコードセットの数。
zone_records	500	ゾーンごとに許可されるレコード数。
recordset_records	20	レコードセットごとに許可されるレコードの数。

表6.3 ゾーンエクスポートクォータ

クォータ	デフォルト	説明
api_export_size	1000	ゾーンエクスポートで許可されるレコードセットの数。

第7章 ゾーンの管理

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) は、ゾーンを使用して namespace を簡単に管理できるように分割します。RHOSP プロジェクトがゾーンを所有している場合、ユーザーはそのゾーンを作成、変更、削除、エクスポート、およびインポートできます。

このセクションに含まれるトピックは次のとおりです。

- 「DNS サービスのゾーン」
- 「ゾーンの作成」
- 「ゾーンの更新」
- 「ゾーンの削除」
- 「ゾーンのエクスポート」
- 「ゾーンのインポート」
- 「ゾーン所有権の譲渡」
- 「ゾーン転送リクエストの変更」

7.1. DNS サービスのゾーン

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) は、DNS と同様のゾーン所有権モデルを使用しますが、2つの大きな違いがあります。

たとえば DNS では、ルートゾーン (.) 内には **.org.** や **.com.** などのトップレベルドメイン (TLD) ごとにゾーンがあります。TLD ゾーン内では、**example.org.** や **example.com** など、他の組織 (または他のネームサーバーセット) による所有および管理が可能な他のゾーンへの委譲が存在する可能性があります。この例は責任の階層を示しており、ほとんどの上位ゾーンは下位ゾーンへの委譲で設定されています。

DNS と同様に、RHOSP DNS サービスの場合、ゾーンは1つのテナントのみが所有できます。ただし、DNS とは異なり、DNS サービスはテナント間のゾーン委譲をサポートしません。つまり、テナントは、親ゾーンが別のテナントが所有する子ゾーンを作成できません。

DNS と RHOSP DNS サービスの2つ目の違いは、DNS サービスがゾーンとは異なる方法で TLD を管理する点です。DNS サービスは、テナントがマネージド TLD の外にゾーンを作成することを制限します。DNS サービスが TLD を管理しない場合、テナントはルートゾーン以外の任意の TLD と任意のゾーンを作成できます。

7.2. ゾーンの実成

ゾーンを使用すると、namespace をより簡単に管理できます。デフォルトでは、どのユーザーでも Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) ゾーンを作成できます。

前提条件

- RHOSP プロジェクトは、サブゾーンを作成するゾーンを所有する必要があります。そうでない場合、ゾーンは許可された TLD である必要があります。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. ゾーンの名前と、そのゾーンの責任者の電子メールアドレスを指定して、ゾーンを作成します。

例

```
$ openstack zone create --email dnsprimary@example.com example.com.
```

ゾーンを作成すると、DNS サービスは2つのレコード (SOA レコードと NS レコード) のセットを自動的に作成します。

検証

- **openstack zone list** コマンドを実行して、ゾーンが存在することを確認します。

出力例

```
+-----+-----+-----+-----+-----+
| id           | name           | type  | serial | status | action |
+-----+-----+-----+-----+-----+
| 14093115-0f0f-497a-ac69-42235e46c26f | example.com. | PRIMARY | 1468421656 | ACTIVE | NONE |
+-----+-----+-----+-----+-----+
```

関連情報

- コマンドラインインターフェイスリファレンスの [zone create](#)
- コマンドラインインターフェイスリファレンスの [zone list](#)

7.3. ゾーンの更新

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) が管理するゾーンを更新する必要がある場合があります。たとえば、ゾーンに関連付けられた電子メールアドレスを変更する場合や、ゾーン TTL (time to live) 値を変更する場合などです。デフォルトでは、どのユーザーでもゾーンを変更できます。

前提条件

- RHOSP プロジェクトは、変更するゾーンを所有する必要があります。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

-
- 2. ゾーンの名前と、変更するゾーン属性を指定して、ゾーンを変更します。

--email <email_address>

ゾーンの責任者 (所有者) の有効な電子メールアドレス。

--ttl <seconds>

(Time To Live) たとえばリゾルバー、Web ブラウザー、オペレーティングシステムなどの DNS クライアントが更新されたかどうかを確認する前にレコードをキャッシュできる期間 (秒単位)。

--description <string> | --no-description

ゾーンの目的を記述する文字列。

--masters <dns-server> [<dns-server> ...]

プライマリーインスタンス (他の DNS サーバーが同期してセカンダリーサーバーになることができるインスタンス) である DNS サーバーの完全修飾ドメイン名。

例

```
$ openstack zone set example.com. --ttl 3000
```

検証

- ゾーンへの変更が成功したことを確認します。

例

```
$ openstack zone show example.com.
```

関連情報

- コマンドラインインターフェイスリファレンスの [zone set](#)
- コマンドラインインターフェイスリファレンスの [zone show](#)

7.4. ゾーンの削除

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) が管理するゾーンを削除できます。たとえば、ゾーン名が変更されたときにゾーンを削除します。

前提条件

- RHOSP プロジェクトは、削除するゾーンを所有する必要があります。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

- ゾーンを削除します。

例

```
$ openstack zone delete example.com.
```

検証

- **openstack zone list** コマンドを実行して、ゾーンが存在しないことを確認します。

関連情報

- コマンドラインインターフェイスリファレンスの [zone delete](#)
- コマンドラインインターフェイスリファレンスの [zone list](#)

7.5. ゾーンのエクスポート

Red Hat OpenStack Platform (RHOSP) DNS サービスからゾーンデータをエクスポートする場合、DNS サービスがデフォルトで内部に保存するゾーンエクスポートリソースを作成します。

例:**designate://v2/zones/tasks/exports/e75aef2c-b562-4cd9-a426-4a73f6cb82be/export**。ゾーンエクスポートデータリソースを作成したら、そのコンテンツにアクセスできます。ゾーンデータのエクスポートは、RHOSP デプロイメントの DNS 情報を保護するための全体的なバックアップストラテジーの一部です。

前提条件

- RHOSP プロジェクトは、データのエクスポート元となるゾーンを所有する必要があります。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. ゾーンをエクスポートします。

例

```
$ openstack zone export create example.com.
```

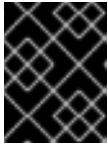
出力例

```
+-----+-----+
| Field | Value |
+-----+-----+
| created_at | 2022-02-11T02:01:30.000000 |
| id | e75aef2c-b562-4cd9-a426-4a73f6cb82be |
| location | None |
| message | None |
| project_id | cf5a8f5cc5834d2dacd1d54cd0a354b7 |
```

```

| status   | PENDING           |
| updated_at | None              |
| version  | 1                 |
| zone_id  | d8f81db6-937b-4388-bfb3-ba620e6c09fb |
+-----+-----+

```



重要

ゾーンエクスポートリソースを作成した後、DNS サービスは、ゾーンに加えられたその後の変更でリソースを更新し続けます。

3. ゾーンのエクスポート ID (**e75aef2c-b562-4cd9-a426-4a73f6cb82be**) を記録します。これは、ゾーンエクスポートの確認と、ゾーンエクスポートデータへのアクセスに使用します。

検証

1. DNS サービスがゾーンエクスポートリソースを正常に作成したことを確認します。

例

```
$ openstack zone export show e75aef2c-b562-4cd9-a426-4a73f6cb82be
```

出力例

```

+-----+-----+
| Field   | Value                                     |
+-----+-----+
| created_at | 2022-02-11T02:01:30.000000           |
| id        | e75aef2c-b562-4cd9-a426-4a73f6cb82be |
| location  | designate://v2/zones/tasks/exports/e75aef2c-b562-4cd9-a426-4a73f6cb82be/export |
| message   | None                                     |
| project_id | cf5a8f5cc5834d2dacd1d54cd0a354b7    |
| status    | COMPLETE                                |
| updated_at | 2022-02-11T02:01:30.000000           |
| version   | 2                                       |
| zone_id   | d8f81db6-937b-4388-bfb3-ba620e6c09fb |
+-----+-----+

```

zone export create コマンドは、DNS サービスがデフォルトで内部に保存するリソースを作成します。

2. 先ほど取得したゾーンエクスポート ID を使用して、ゾーンエクスポートファイルのコンテンツにアクセスします。

ヒント

-f value オプションを使用すると、ゾーンファイルのコンテンツがタブなしで出力されます。コンテンツをローカルテキストファイルにリダイレクトすることもできます。これは、エクスポートされたゾーンファイルをローカルで変更してから、DNS サービスにインポートしてゾーンを更新する場合に便利です。

例

```
$ openstack zone export showfile e75aef2c-b562-4cd9-a426-4a73f6cb82be -f value
```

出力例

```
$ORIGIN example.com.
$TTL 3600

example.com. IN NS ns1.example.com.
example.com. IN SOA ns1.example.com. admin.example.com. 1624414033 3583 600 86400
3600

www.example.com. IN A 192.0.2.2
www.example.com. IN A 192.0.2.1
```

関連情報

- ゾーンファイル形式:
 - [RFC1034, section 3.6](#)
 - [RFC1035, section 5.1](#)
- コマンドラインインターフェイスリファレンスの [zone export create](#)
- コマンドラインインターフェイスリファレンスの [zone export show](#)
- コマンドラインインターフェイスリファレンスの [zone export showfile](#)

7.6. ゾーンのインポート

ゾーンデータを Red Hat OpenStack Platform (RHOSP) DNS サービスにインポートするには、たとえば **openstack zone export showfile** コマンドで生成されるデータから作成されるファイルなど、DNS ゾーンのデータファイル形式に準拠したファイルで **openstack zone import** コマンドを実行します。データをインポートする理由の1つは、ユーザーが誤ってゾーンを削除した場合です。

前提条件

- RHOSP プロジェクトは、サブゾーンを作成するゾーンを所有している必要があります。そうでない場合、ゾーンは許可された TLD である必要があります。
- すでに存在しているゾーンをインポートしてはなりません。
- インポートするゾーンデータには、ゾーン TTL (time to live) 値が含まれている必要があります。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. システムのゾーンをリスト表示します。


```
$ openstack zone list
```

- インポートするゾーンがすでに存在する場合は、最初に **openstack zone delete** コマンドを実行してそのゾーンを削除する必要があります。

例

```
$ openstack zone delete example.com.
```

- システムのゾーンをリスト表示して、ゾーンが存在しないことを確認します。

```
$ openstack zone list
```

- インポート予定のゾーンデータにゾーン TTL 値が含まれていることを確認します。

例

```
$ cat /home/stack/zone_file
```

出力例

```
$ORIGIN example.com.
$TTL 3000

example.com. IN NS test.example.com.
example.com. IN SOA test.example.com. admin.example.com. 1624415706 9000 500 86000
5000
www.example.com. IN A 192.0.2.2
test.example.com. IN NS test.example.com.
```

- 有効なゾーンデータファイルをインポートします。

例

```
$ openstack zone import create /home/stack/zone_file
```

検証

- DNS サービスがゾーンを正常にインポートしたことを確認します。

例

```
$ openstack recordset list -c name -c type -c records -c status example.com.
```

出力例

```
+-----+-----+-----+-----+-----+
| name           | type | records                                     | status |
+-----+-----+-----+-----+-----+
| example.com.   | SOA  | ns1.example.com. admin.example.com. 1624415706 3582 500
86000 3600 | ACTIVE |
| test.example.com. | NS   | test.example.com.                                     | ACTIVE |
```

```

| example.com.   | NS | ns1.example.com.   | ACTIVE |
| www.example.com. | A  | 192.0.2.2         | ACTIVE |
+-----+-----+-----+-----+

```

関連情報

- ゾーンファイル形式:
 - [RFC1034, section 3.6](#)
 - [RFC1035, section 5.1](#)
- コマンドラインインターフェイスリファレンスの [zone import create](#)
- コマンドラインインターフェイスリファレンスの [zone list](#)

7.7. ゾーン所有権の譲渡

ゾーンの所有権をあるプロジェクトから別のプロジェクトに譲渡できます。たとえば、財務チームは、**wow.example.com.** ゾーンの所有権を、同チームのプロジェクトから、営業チームのプロジェクトに委譲するとします。

クラウド管理者の関与なしに、ゾーンの所有権を譲渡できます。ただし、現在のプロジェクトゾーンの所有者と譲渡先プロジェクトのメンバーの両方が譲渡に同意する必要があります。

前提条件

- プロジェクトは、転送するゾーンを所有している必要があります。
- 転送要求を作成した後、転送先のプロジェクトのメンバーは、転送するゾーンを承認する必要があります。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. ゾーンの所有権を譲渡するプロジェクトの ID を取得します。

例

```
$ openstack project list
```

出力例

```

+-----+-----+-----+-----+
| ID                | Name          |
+-----+-----+-----+-----+
| 7af0acba0486472da2447ff55df4a26d | Finance      |
| 1d12e87fad0d437286c2873b36a12316 | Sales       |
+-----+-----+-----+-----+

```

3. 前の手順で取得したプロジェクト ID を使用して、転送するゾーンのゾーン転送要求を作成します。



注記

ターゲットプロジェクト ID を使用する場合、他のプロジェクトはゾーン転送を受け入れることができません。ターゲットプロジェクト ID を指定しない場合、転送リクエスト ID とそのキーを持つすべてのプロジェクトがゾーン転送を受け取ることができます。

例

ゾーン **wow.example.com** を転送します。 **1d12e87fad0d437286c2873b36a12316** をプロジェクトするには、次を実行します。

```
$ openstack zone transfer request create --target-project-id
1d12e87fad0d437286c2873b36a12316 wow.example.com.
```

出力例

```
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| created_at  | 2022-05-26T22:06:39.000000              |
| description | None                                     |
| id          | 63cab5e5-65fa-4480-b26c-c16c267c44b2    |
| key        | BIFJIQWH                                 |
| links       | {'self': 'http://127.0.0.1:60053/v2/zones/tasks/tra |
|             | nsfer_requests/63cab5e5-65fa-4480-b26c-c16c267c44b2 |
|             | '}                                       |
| project_id  | 6265985fc493465db6a978b318a01996        |
| status      | ACTIVE                                   |
| target_project_id | 1d12e87fad0d437286c2873b36a12316    |
| updated_at  | None                                     |
| zone_id     | 962f08b4-b671-4096-bf24-8908c9d4af0c    |
| zone_name   | wow.example.com.                         |
+-----+-----+
```

4. ゾーン転送リクエスト ID とそのキーを取得します。

例

```
$ openstack zone transfer request list -c id -c zone_name -c key
```

出力例

```
+-----+-----+-----+
| id          | zone_name   | key      |
+-----+-----+-----+
| 63cab5e5-65fa-4480-b26c-c16c267c44b2 | wow.example.com. | BIFJIQWH |
+-----+-----+-----+
```

- ゾーン転送リクエスト ID とそのキーを、受信側プロジェクトのメンバーに提供します。
- 受信側プロジェクトのメンバーが受信側プロジェクトにログインし、認証情報ファイルを入手します。

例

```
$ source ~/overcloudrc
```

- ゾーン転送要求 ID とそのキーを使用して、ゾーン転送を受け入れます。

例

```
$ openstack zone transfer accept request --transfer-id 63cab5e5-65fa-4480-b26c-c16c267c44b2 --key BIFJIQWH
```

出力例

```
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| created_at     | 2022-05-27T21:37:43.000000             |
| id             | a4c4f872-c98c-411b-a787-58ed0e2dce11   |
| key           | BIFJIQWH                               |
| links         | {'self': 'http://127.0.0.1:60053/v2/zones/ta |
|               | sks/transfer_accepts/a4c4f872-c98c-411b-a787 |
|               | -58ed0e2dce11', 'zone': 'http://127.0.0.1:60 |
|               | 053/v2/zones/962f08b4-b671-4096-bf24-8908c9d |
|               | 4af0c'}                                 |
| project_id    | 1d12e87fad0d437286c2873b36a12316     |
| status        | COMPLETE                               |
| updated_at    | 2022-05-27T21:37:43.000000             |
| zone_id       | 962f08b4-b671-4096-bf24-8908c9d4af0c   |
| zone_transfer_request_id | 63cab5e5-65fa-4480-b26c-c16c267c44b2 |
+-----+-----+
```

検証

- 前のステップのゾーン転送受け入れ ID を使用して、ゾーン転送のステータスを確認します。

例

この例では、ゾーンステータス受け入れ ID は **a4c4f872-c98c-411b-a787-58ed0e2dce11** です。

```
$ openstack zone transfer accept show a4c4f872-c98c-411b-a787-58ed0e2dce11
```

出力例

```
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| created_at     | 2022-05-27T21:37:43.000000             |
| id             | a4c4f872-c98c-411b-a787-58ed0e2dce11   |
```

```

| key          | None          |
| links        | {'self': 'http://127.0.0.1:60053/v2/zones/ta |
|              | sks/transfer_accepts/a4c4f872-c98c-411b-a787 |
|              | -58ed0e2dce11', 'zone': 'http://127.0.0.1:60 |
|              | 053/v2/zones/962f08b4-b671-4096-bf24-8908c9d |
|              | 4af0c'}      |
| project_id   | 1d12e87fad0d437286c2873b36a12316          |
| status       | COMPLETE                                       |
| updated_at   | 2022-05-27T21:37:43.000000                 |
| zone_id      | 962f08b4-b671-4096-bf24-8908c9d4af0c      |
| zone_transfer_request_id | 63cab5e5-65fa-4480-b26c-c16c267c44b2 |
+-----+-----+

```

関連情報

- コマンドラインインターフェイスリファレンスの [zone transfer request create](#) コマンド
- コマンドラインインターフェイスリファレンスの [zone transfer accept request](#) コマンド

7.8. ゾーン転送リクエストの変更

あるプロジェクトから別のプロジェクトにゾーンの所有権を譲渡する最初のステップは、ゾーン譲渡要求を作成することです。ゾーン転送要求を変更または削除する必要がある場合は、それを行うことができます。

前提条件

- プロジェクトは、変更する転送リクエストのゾーンを所有している必要があります。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. 変更するゾーン転送要求の ID を取得します。

例

```
$ openstack zone transfer request list -c id -c zone_name
```

出力例

```

+-----+-----+
| id          | zone_name    |
+-----+-----+
| 63cab5e5-65fa-4480-b26c-c16c267c44b2 | wow.example.com. |
+-----+-----+

```

3. 前の手順で取得したゾーン転送リクエスト ID を使用して、説明やターゲットプロジェクト ID など、ゾーン転送リクエストの限定されたフィールドセットを更新できます。

例

```
$ openstack zone transfer request set --description "wow zone transfer" 63cab5e5-65fa-4480-b26c-c16c267c44b2
```

出力例

```
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| created_at  | 2022-05-26T22:06:39.000000              |
| description | wow zone transfer                        |
| id          | 63cab5e5-65fa-4480-b26c-c16c267c44b2    |
| key        | BIFJIQWH                                |
| links       | {'self': 'http://127.0.0.1:60053/v2/zones/tasks/tra |
|             | nsfer_requests/63cab5e5-65fa-4480-b26c-c16c267c44b2 |
|             | '}                                        |
| project_id  | 6265985fc493465db6a978b318a01996        |
| status      | ACTIVE                                   |
| target_project_id | 1d12e87fad0d437286c2873b36a12316    |
| updated_at  | 2022-05-27T20:52:08.000000              |
| zone_id     | 962f08b4-b671-4096-bf24-8908c9d4af0c    |
| zone_name   | wow.example.com.                         |
+-----+-----+
```

- 手順 2 で取得したゾーン転送要求 ID を使用して、ゾーン転送要求を削除することにより、保留中のゾーン転送をキャンセルできます。

例

```
$ openstack zone transfer request delete 63cab5e5-65fa-4480-b26c-c16c267c44b2
```

zone transfer request delete コマンドからの出力はありません。**zone transfer request list** コマンドを実行して、ゾーン転送要求が削除されたことを確認します。

関連情報

- [「ゾーン所有権の譲渡」](#)
- コマンドラインインターフェイスリファレンスの [zone transfer request set](#) コマンド
- コマンドラインインターフェイスリファレンスの [zone transfer request delete](#) コマンド

第8章 レコードセットの管理

Red Hat OpenStack (RHOSP) DNS サービス (designate) は、ゾーンについてのデータをレコードセットに保存します。レコードセットは、1つ以上の DNS リソースレコードで設定されます。ゾーンに対してクエリーを実行し、レコードセットの追加、変更、および削除だけでなくリスト表示もできます。

このセクションに含まれるトピックは次のとおりです。

- 「DNS サービスのレコードおよびレコードセットについて」
- 「レコードセットの作成」
- 「レコードセットの更新」
- 「レコードセットの削除」

8.1. DNS サービスのレコードおよびレコードセットについて

Domain Name System (DNS) はリソースレコードを使用して namespace 内にゾーンデータを保存します。Red Hat OpenStack (RHOSP) DNS サービス (designate) の DNS レコードは、レコードセットを使用して管理されます。

各 DNS レコードには以下の属性が含まれます。

- **Name:** DNS namespace 内の場所を示す文字列。
- **Type:** レコードの使用方法を識別する文字コードのセット。たとえば、**A** はアドレスレコードを識別し、**CNAME** は正規名レコードを識別します。
- **Class:** レコードの namespace を指定する文字コードのセット。通常、これはインターネットの場合は **IN** になりますが、他の namespace も存在します。
- **TTL:** (time to live) レコードが有効なままである期間 (秒単位)。
- **Rdata:** A レコードの IP アドレスや CNAME レコードの別のレコード名などのレコードデータ。

各ゾーン namespace には、SOA (start of authority) レコードが含まれている必要があり、権威ネームサーバー (NS) レコードとその他のさまざまなタイプのレコードを持つこともできます。SOA レコードは、このネームサーバーがゾーンに関する最適な情報源であることを示しています。NS レコードは、ゾーンで権限のあるネームサーバーを識別します。ゾーンの SOA レコードおよび NS レコードは読み取り可能ですが、変更することはできません。

必要な SOA レコードと NS レコード以外で最も一般的な3つのレコードタイプは、アドレス (A) レコード、正規名 (CNAME) レコード、およびポインター (PTR) レコードです。A レコードはホスト名を IP アドレスにマッピングします。PTR レコードは、IP アドレスをホスト名にマッピングします。CNAME レコードはエイリアスの完全なホスト名を特定します。

レコードセットは、名前とタイプが同じ1つ以上の DNS レコードを表しますが、データは異なる場合があります。たとえば、データ **192.0.2.1** と **192.0.2.2** を含むタイプ **A** の **web.example.com** という名前のレコードセットは、これら2つの IP アドレスにある **web.example.com** をホストする2つの Web サーバーを反映している可能性があります。

レコードセットはゾーン内に作成する必要があります。レコードセットを含むゾーンを削除すると、そのゾーン内のレコードセットも削除されます。

`openstack recordset list -c name -c type -c records example.com` コマンドで `example.com` ゾーンにクエリーを実行して得られたこの出力について考えてください。

```
+-----+-----+-----+
| name      | type | records                               |
+-----+-----+-----+
| example.com. | SOA | ns1.example.net. admin.example.com. 16200126 |
|             |     | 16 3599 600 8640 0 3600                |
|             |     |                                         |
| example.com. | NS  | ns1.example.net.                       |
|             |     |                                         |
| web.example.com. | A  | 192.0.2.1                             |
|             |     | 192.0.2.2                             |
|             |     |                                         |
| www.example.com. | A  | 192.0.2.1                             |
+-----+-----+-----+
```

この例では、`example.com.` ゾーンの権威ネームサーバーは NS レコードの `ns1.example.net` です。これを確認するには、BIND `dig` ツールを使用して、ネームサーバーに NS レコードのクエリーを実行します。

```
$ dig @ns1.example.net example.com. -t NS +short
ns1.example.net.
```

A レコードセットを確認することもできます。

```
$ dig @ns1.example.net web.example.com. +short
192.0.2.2
192.0.2.1
$ dig @ns1.example.net www.example.com. +short
192.0.2.1
```

8.2. レコードセットの作成

デフォルトでは、どのユーザーでも Red Hat OpenStack Platform DNS サービス (`designate`) のレコードセットを作成できます。

前提条件

- プロジェクトは、レコードセットを作成するゾーンを所有している必要があります。

手順

- Source コマンドで認証情報ファイルを読み込みます。

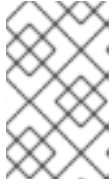
例

```
$ source ~/overcloudrc
```

- `openstack recordset create` コマンドを使用して、レコードセットを作成します。レコードセットには、ゾーン、名前、タイプ、およびデータが必要です。

例


```
$ openstack recordset create --type A --record 192.0.2.1 example.com. www
```



注記

完全修飾ドメイン名 (FQDN) を使用する場合は、末尾のドット (.) が必要です。末尾のドットを省略すると、ゾーン名が結果のレコード名に複製されます (例: **www.example.com.example.com.**)。

上記の例では、ユーザーは **example.com.** という名前のゾーンを作成しています。レコードセット名 **www** は FQDN ではないため、DNS サービスはそれをゾーン名の前に追加します。レコードセット名引数に FQDN を使用しても、同じ結果を得ることができます。

```
$ openstack recordset create --type A --record 192.0.2.1 example.com. www.example.com.
```

3. 文字列の最大長 (255 文字) を超える TXT レコードセットを作成する場合は、レコードセットを作成するときに、文字列を複数の小さい文字列に分割する必要があります。この例では、ユーザーは 2 つの文字列 (それぞれ最大 255 文字未満) を指定して、410 文字の 1 つの文字列を含む TXT レコードセット (**_domainkey.example.com**) を作成します。

```
$ openstack recordset create --type TXT --record "'210 characters string" "200 characters string" example.com. _domainkey
```

4. **--record** 引数を複数回指定して、1 つのレコードセット内に複数のレコードを作成できます。複数の **--record** 引数の一般的な使用方法はラウンドロビン DNS です。

例

```
$ openstack recordset create --type A --record 192.0.2.1 --record 192.0.2.2 example.com. web
```

検証

- list コマンドを実行して、作成したレコードセットが存在することを確認します。

例

```
$ openstack recordset list -c name -c type -c records example.com.
```

出力例

```
+-----+-----+-----+
| name      | type | records                                     |
+-----+-----+-----+
| example.com. | SOA | ns1.example.net. admin.example.com 162001261 |
|             |     | 6 3599 600 86400 3600                       |
|             |     |                                             |
| example.com. | NS  | ns1.example.net.                             |
|             |     |                                             |
| web.example.com. | A  | 192.0.2.1 192.0.2.2                         |
|             |     |                                             |
| www.example.com. | A  | 192.0.2.1                                     |
+-----+-----+-----+
```

-

関連情報

- コマンドラインインターフェイスリファレンスの `recordset create` コマンド
- コマンドラインインターフェイスリファレンスの `recordset list` コマンド
- `dig` の man ページ

8.3. レコードセットの更新

デフォルトでは、どのユーザーでも Red Hat OpenStack Platform DNS サービス (designate) のレコードセットを更新できます。

前提条件

- プロジェクトは、レコードセットを更新するゾーンを所有している必要があります。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. `openstack recordset set` コマンドを使用して、レコードセットを変更します。

例

この例では、ユーザーはレコードセット `web.example.com.` を、2つのレコードを含めるように更新しています。

```
$ openstack recordset set example.com. web.example.com. --record 192.0.2.5 --record 192.0.2.6
```



注記

レコードセットを更新する場合は、その ID または名前で識別できます。名前を使用する場合は、完全修飾ドメイン名 (FQDN) を使用する必要があります。

検証

- `list` コマンドを実行して変更を確認します。

例

```
$ openstack recordset list -c name -c type -c records example.com.
```

出力例

```
+-----+-----+-----+
| name          | type | records          |
```

```

+-----+-----+-----+-----+
| example.com. | SOA | ns1.example.net. admin.example.com 162001261 | | | |
|             |     | 6 3599 600 86400 3600 |
|             |     | | | | |
| example.com. | NS  | ns1.example.net. |
|             |     | | | | |
| web.example.com. | A  | 192.0.2.5 192.0.2.6 |
|             |     | | | | |
| www.example.com. | A  | 192.0.2.1 |
+-----+-----+-----+-----+

```

関連情報

- コマンドラインインターフェイスリファレンスの [recordset create](#) コマンド
- コマンドラインインターフェイスリファレンスの [recordset list](#) コマンド

8.4. レコードセットの削除

デフォルトでは、どのユーザーでも Red Hat OpenStack Platform DNS サービス (designate) のレコードセットを削除できます。

前提条件

- プロジェクトは、レコードセットを削除するゾーンを所有している必要があります。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. **openstack recordset delete** コマンドを使用して、レコードセットを削除します。

例

この例では、ユーザーはレコードセット **web.example.com.** を、example.com. ゾーンから削除しています。

```
$ openstack recordset delete example.com. web.example.com.
```

検証

- list コマンドを実行して削除を確認します。

例

```
$ openstack recordset list -c name -c type -c records example.com.
```

出力例

```
+-----+-----+
| name      | type | records                                     |
+-----+-----+
| example.com. | SOA | ns1.example.net. admin.example.com 162001261 |
|            |     | 6 3599 600 86400 3600                     |
|            |     |                                             |
| example.com. | NS  | ns1.example.net.                           |
|            |     |                                             |
| www.example.com. | A  | 192.0.2.1                                   |
+-----+-----+
```

関連情報

- コマンドラインインターフェイスリファレンスの [recordset delete](#) コマンド
- コマンドラインインターフェイスリファレンスの [recordset list](#) コマンド

第9章 ポインターレコード (PTR) の管理

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) を設定する手順には、リバースルックアップとも呼ばれる IP address-to-domain-name-lookups の設定があります。DNS リソースであるポインター (PTR) レコードには、アドレスから名前へのマッピングデータが含まれ、リバースルックアップゾーンに格納されます。DNS サービスを使用すると、Floating IP アドレスのリバースルックアップを管理することもできます。

このセクションに含まれるトピックは次のとおりです。

- [「PTR レコードの基本」](#)
- [「リバースルックアップゾーンの作成」](#)
- [「PTR レコードの作成」](#)
- [「複数の PTR レコードの作成」](#)
- [「Floating IP アドレスの PTR レコードの設定」](#)
- [「Floating IP アドレスの PTR レコードの設定解除」](#)

9.1. PTR レコードの基本

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) では、ポインター (PTR) レコードを使用して、単一 IP アドレスまたは複数の IP アドレスのセット完全就職ドメイン名 (FQDN) に名前をマッピング (リバースマッピング) するための番号を作成します。Domain Name System (DNS) はアドレスを名前として検索するため、IP アドレスの名前が含まれる PTR レコードを作成します。この名前は、特定の規則に従って作成します。IP アドレスを逆引きし、特別な文字列 (IPv4 アドレスの場合は **in-addr.arpa**、IPv6 アドレスの場合は **ip6.arpa**) を追加します。

たとえば、**my-server.example.com** の IP アドレスが **198.51.100.42** の場合、リバースルックアップゾーンの対応するノードに **42.100.51.198.in-addr.arpa** という名前を付けます。IP アドレスの名前を逆方向にリストすると、検索が容易になります。これは、標準の完全修飾ドメイン名 (FQDN) と同様に、逆方向の IP アドレスは、左側から右側に移動するにつれて具体性が低くなるためです。

DNS サービスは、PTR レコードの内容を、アドレスから名前へのルックアップを提供することを唯一の目的とする、リバースルックアップゾーンと呼ばれる特別なゾーンに書き込みます。PTR レコードには標準 FQDN と同様に構造化されているデータが含まれているため、他のゾーンを委譲するのと同じ方法で、リバースルックアップゾーンの子ゾーンを委譲できます。上記の例では、ホスト (**198.51.100.42**) は **198.in-addr.arpa** ゾーンの子ノードであり、このゾーンはネットワーク (**198.51.100.0/8**) の管理者に委譲できます。

ユーザーの RHOSP プロジェクトが IP アドレスを含むゾーンを所有している必要があるため、DNS サービスは Floating IP アドレスの PTR レコードを標準の IP アドレスとは異なる方法で管理します。名前のリバースルックアップに関連するほとんどのユースケースでは、この要件は簡単に満たされます。標準 IP アドレスのリバースルックアップを管理する場合は、他の DNS リソースレコードタイプを管理する場合と同じように **openstack recordset** コマンドを使用します。

ただし、Floating IP アドレスを使用する場合は、複数のプロジェクトで Floating IP アドレスのプールを共有するのが一般的です。アドレスの共有プールのプロジェクト所有権に関する問題を解決するには、Floating IP のリバースルックアップを管理する際に、**openstack ptr record** コマンドという別のコマンドを使用する必要があります。

関連情報

- 「PTR レコードの作成」
- 「Floating IP アドレスの PTR レコードの設定」

9.2. リバースルックアップゾーンの作成

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) を適切に設定するには、リバースルックアップゾーンが必要です。リバースルックアップゾーンには、アドレスから名前の検索を実行するために必要な PTR レコードが含まれます。次の規則に従ってリバースルックアップゾーンに名前を付ける必要があります。IPv4 アドレスの場合は **<backward_IP_address>.in-addr.arpa**、IPv6 アドレスの場合は **<backward_IP_address>.ip6.arpa**。

通常、RHOSP デプロイメントのゾーンをお使いのサブネットプランに合わせます。たとえば、外部ネットワーク用に /24 サブネットがある場合、PTR レコードを格納する /24 サブネットリバースルックアップゾーンを作成します。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. **openstack zone create** コマンドを使用し、次の必須の引数を指定して、リバースルックアップゾーンを作成します。

--email <email_address>

ゾーンの責任者 (所有者) の有効な電子メールアドレス。

<name>

次の規則に準拠するリバースルックアップゾーンの名前: IPv4 アドレスの場合は **<backward_IP_address>.in-addr.arpa**、IPv6 アドレスの場合は **<backward_IP_address>.ip6.arpa**。

例

この例では、リバースルックアップゾーンは、198.51.100.42 アドレスの1つの PTR レコード用に設計されています。

```
$ openstack zone create --email admin@example.com \
  42.100.51.198.in-addr.arpa.
```

出力例

```
+-----+
| Field      | Value                                     |
+-----+
| action     | CREATE                                  |
| attributes |                                          |
| created_at | 2022-02-02T17:32:47.000000             |
| description | None                                    |
| email      | admin@example.com                       |
| id         | f5546034-b27e-4326-bf9d-c53ed879f7fa  |
| masters    |                                          |
+-----+
```

```

| name      | 42.100.51.198.in-addr.arpa. |
| pool_id   | 794ccc2c-d751-44fe-b57f-8894c9f5c842 |
| project_id | 123d51544df443e790b8e95cce52c285 |
| serial    | 1591119166 |
| status    | PENDING |
| transferred_at | None |
| ttl       | 3600 |
| type      | PRIMARY |
| updated_at | None |
| version   | 1 |
+-----+

```

例

別の例は 198.51.100.0/24 サブネット用のリバースゾーンで、ゾーンを作成します。

```

$ openstack zone create --email admin@example.com \
  100.51.198.in-addr.arpa.

```

出力例

```

+-----+
| Field   | Value |
+-----+
| action  | CREATE |
| attributes | |
| created_at | 2022-02-02T17:40:23.000000 |
| description | None |
| email    | admin@example.com |
| id       | 5669caad86a04256994cdf755df4d3c1 |
| masters  | |
| name     | 100.51.198.in-addr.arpa. |
| pool_id  | 794ccc2c-d751-44fe-b57f-8894c9f5c842 |
| project_id | 123d51544df443e790b8e95cce52c285 |
| serial   | 1739276248 |
| status   | PENDING |
| transferred_at | None |
| ttl      | 3600 |
| type     | PRIMARY |
| updated_at | None |
| version  | 1 |
+-----+

```

検証

1. 作成したリバースルックアップゾーンが存在することを確認します。

```

$ openstack zone list -c id -c name -c status

```

出力例

```

+-----+
| id           | name | status |
+-----+

```

```
+-----+-----+
| f5546034-b27e-4326-bf9d-c53ed879f7fa | 42.100.51.198.in-addr.arpa. | ACTIVE |
+-----+-----+
```

2. アドレスから名前へのマッピングを完了するには、フォワードゾーン (IP アドレスを含むゾーン) が存在している必要があります。フォワードゾーンが存在しない場合は、これを作成します。

関連情報

- [ゾーンの作成](#)
- コマンドラインインターフェイスリファレンスの [zone create](#)

9.3. PTR レコードの作成

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) では、PTR レコードを作成して、リバースルックアップ (アドレスから名前へのマッピング) を有効にします。リバースルックアップを有効にすることは、RHOSP デプロイメントで DNS サービスを適切に設定するために必要です。

前提条件

- RHOSP プロジェクトは、PTR レコードを作成するゾーンを所有している必要があります。
- PTR レコードを保存するリバースルックアップゾーン。詳細は、「[リバースルックアップゾーンの作成](#)」を参照してください。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. **openstack recordset create** コマンドを使用し、次の必須の引数を指定して、PTR レコードを作成します。

--record <domain_name>

リバースルックアップを実行する際に返されるターゲット (ドメイン名)。

--type PTR

作成しているレコードの種類 (PTR)。

<zone_name>

レコードが存在するゾーンの名前 (リバースルックアップゾーン)。

<record_name>

PTR レコードの名前。

レコード名は <zone_name> と一致するか、ゾーンのメンバーである必要があります。たとえば、リバースルックアップゾーン **100.51.198.in-addr.arpa.** の場合、これらは有効な PTR レコード名です: **1.100.51.198.in-addr.arpa.**、**2.100.51.198.in-addr.arpa.**、および **198.51.100.0/24** サブネット内のその他の任意のリバース IP アドレス。

例


```
openstack recordset create --record www.example.com. --type PTR \
42.100.51.198.in-addr.arpa. 42.100.51.198.in-addr.arpa.
```

出力例

```
+-----+-----+
| Field   | Value                               |
+-----+-----+
| action  | CREATE                              |
| created_at | 2022-02-02T19:55:50.000000         |
| description | None                                |
| id      | ca604f72-83e6-421f-bf1c-bb4dc1df994a |
| name    | 42.100.51.198.in-addr.arpa.        |
| project_id | 123d51544df443e790b8e95cce52c285  |
| records  | www.example.com.                   |
| status   | PENDING                             |
| ttl     | 3600                                |
| type    | PTR                                  |
| updated_at | None                                |
| version  | 1                                    |
| zone_id  | f5546034-b27e-4326-bf9d-c53ed879f7fa |
| zone_name | 42.100.51.198.in-addr.arpa.        |
+-----+-----+
```

検証

- リバーズルックアップを実行して、IP アドレス (**198.51.100.42**) がドメイン名 (**www.example.com**) にマッピングされていることを確認します。

例

この例では、**203.0.113.5** はデプロイメント内の DNS サーバーの1つです。

```
$ dig @203.0.113.5 -x 198.51.100.42 +short
```

出力例

```
www.example.com.
```

関連情報

- コマンドラインインターフェイスリファレンスの [recordset create](#)
- dig** コマンドの man ページ

9.4. 複数の PTR レコードの作成

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) では、より広範囲に定義されたりバースルックアップゾーンを使用して、多数の PTR レコードを大規模なサブネットに追加できます。

前提条件

- RHOSP プロジェクトは、PTR レコードを作成するゾーンを所有している必要があります。
- より広範囲に定義されている、PTR レコードを保存するリバースルックアップゾーン。たとえば、**198.51.100.0/24** リバースルックアップゾーン、**100.51.198.in-addr.arpa**。詳細は、「[リバースルックアップゾーンの作成](#)」を参照してください。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. **openstack recordset create** コマンドを使用し、次の必須の引数を指定して、PTR レコードを作成します。

--record <domain_name>

ルックアップのドメイン名。

--type PTR

作成しているレコードの種類 (**PTR**)。

<zone_name>

レコードが存在するリバースルックアップゾーンの名前。

<record_name>

PTR レコードの名前。

レコード名は <zone_name> と一致するか、ゾーンのメンバーである必要があります。たとえば、リバースルックアップゾーン **100.51.198.in-addr.arpa** の場合、これらは有効な PTR レコード名です: **1.100.51.198.in-addr.arpa.**、**2.100.51.198.in-addr.arpa.**、および **198.51.100.0/24** サブネット内のその他の任意のリバース IP アドレス。

例

この例では、リバースルックアップゾーンはより広範囲に定義されています。たとえば、**100.51.198.0/24** リバースルックアップゾーン、**100.51.198.in-addr.arpa**。

```
$ openstack recordset create --record cats.example.com. --type PTR \
--ttl 3600 100.51.198.in-addr.arpa. 3.100.51.198.in-addr.arpa.
```

出力例

```
+-----+-----+
| Field  | Value                                |
+-----+-----+
| action | CREATE                               |
| created_at | 2022-02-02T20:10:54.000000          |
| description | None                                |
| id      | c843729b-7aaf-4f99-a40a-d9bf70edf271 |
| name    | 3.100.51.198.in-addr.arpa.          |
| project_id | 123d51544df443e790b8e95cce52c285   |
| records  | cats.example.com.                  |
| status   | PENDING                             |
| ttl      | 3600                                |
| type     | PTR                                  |
```

```
| updated_at | None |
| version | 1 |
| zone_id | e9fd0ced-1d3e-43fa-b9aa-6d4b7a73988d |
| zone_name | 100.51.198.in-addr.arpa. |
+-----+-----+
```

検証

- リバーสลックアップを実行して、IP アドレス (**198.51.100.3**) がドメイン名 (**cats.example.com**) にマッピングされていることを確認します。

例

この例では、**203.0.113.5** はデプロイメント内の DNS サーバーの1つです。

```
$ dig @203.0.113.5 -x 198.51.100.3 +short
```

出力例

```
cats.example.com.
```

- リバーสลックアップを実行して、他の IP アドレス (**198.51.100.0/24**) がドメイン名 (**example.com**) にマッピングされていることを確認します。

例

この例では、**203.0.113.5** はデプロイメント内の DNS サーバーの1つです。

```
$ dig @203.0.113.5 -x 198.51.100.10 +short
```

出力例

```
example.com.
```

関連情報

- コマンドラインインターフェイスリファレンスの [recordset create](#)
- **dig** コマンドの man ページ

9.5. FLOATING IP アドレスの PTR レコードの設定

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) では、Floating IP アドレスの PTR レコードを作成して、リバーสลックアップを有効にします。

前提条件

- 1つ以上の Floating IP が定義されている。
- PTR レコードを作成する Floating IP のリバーสลックアップゾーン。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. PTR レコードを削除する Floating IP アドレスの ID を決定します。この情報は後の手順で必要になります。

```
$ openstack floating ip list -c ID -c "Floating IP Address"
```

出力例

```
+-----+-----+
| ID                | Floating IP Address |
+-----+-----+
| 5c02c519-4928-4a38-bd10-c748c200912f | 192.0.2.11          |
| 89532684-13e1-4af3-bd79-f434c9920cc3 | 192.0.2.12          |
| ea3ebc6d-a146-47cd-aaa8-35f06e1e8c3d | 192.0.2.13          |
+-----+-----+
```

3. Floating IP をホストする neutron インスタンスの RHOSP リージョン名を決定します。この情報は後の手順で必要になります。

```
$ openstack endpoint list -c ID -c Region -c "Service Name"
```

出力例

```
+-----+-----+-----+
| ID                | Region  | Service Name |
+-----+-----+-----+
| 16526452effd467a915155ceccf79dae | RegionOne | placement    |
| 21bf826a62a14456a61bd8f39648e849 | RegionOne | keystone     |
| 9cb1956999c54001a39d11ea14e037a1 | RegionOne | nova         |
| bdeec4e2665d4605bb89e16a8b1bc50d | RegionOne | glance       |
| ced05a1c03ab44caa1a351ace95429e6 | RegionOne | neutron      |
| e79e3113ea544d039b3a6378e60bdf3f | RegionOne | nova         |
| f91ee44123954b6c82162dcd2d4fc965 | RegionOne | designate    |
+-----+-----+-----+
```

4. **openstack ptr record set** コマンドを使用して PTR レコードを作成し、以下の必要な引数を指定します。

<floating_IP_ID>

<region_name>:<floating_IP_ID> の形式の Floating IP ID。

<ptrd_name>

リバースルックアップを実行する際に返されるターゲット (ドメイン名)。

例

```
$ openstack ptr record set RegionOne:5c02c519-4928-4a38-bd10-c748c200912f
ftp.example.com.
```

出力例

```
+-----+
| Field | Value |
+-----+
| action | CREATE |
| address | 192.0.2.11 |
| description | None |
| id | RegionOne:5c02c519-4928-4a38-bd10-c748c200912f |
| ptrdname | ftp.example.com. |
| status | PENDING |
| ttl | 3600 |
+-----+
```

検証

- リバーサルルックアップを実行して、Floating IP アドレス (**192.0.2.11**) がドメイン名 (**ftp.example.com**) にマッピングされていることを確認します。

例

この例では、**203.0.113.5** はデプロイメント内の DNS サーバーの1つです。

```
$ dig @203.0.113.5 -x 192.0.2.11 +short
```

出力例

```
ftp.example.com.
```

関連情報

- コマンドラインインターフェイスリファレンスの [ptr record set](#)
- `dig` コマンドの man ページ

9.6. FLOATING IP アドレスの PTR レコードの設定解除

Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) では、Floating IP アドレスに関連付けられた PTR レコードを削除できます。

前提条件

- Floating IP の PTR レコード

手順

- Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. PTR レコードを削除する Floating IP アドレスの ID を決定します。この情報は後の手順で必要になります。

```
$ openstack floating ip list -c ID -c "Floating IP Address"
```

出力例

```
+-----+-----+
| ID                | Floating IP Address |
+-----+-----+
| 5c02c519-4928-4a38-bd10-c748c200912f | 192.0.2.11          |
| 89532684-13e1-4af3-bd79-f434c9920cc3 | 192.0.2.12          |
| ea3ebc6d-a146-47cd-aaa8-35f06e1e8c3d | 192.0.2.13          |
+-----+-----+
```

3. RHOSP リージョンの名前を決定します。この情報は後の手順で必要になります。

```
$ openstack endpoint list -c ID -c Region -c "Service Name"
```

出力例

```
+-----+-----+-----+
| ID                | Region | Service Name |
+-----+-----+-----+
| 16526452effd467a915155ceccf79dae | RegionOne | placement    |
| 21bf826a62a14456a61bd8f39648e849 | RegionOne | keystone     |
| 9cb1956999c54001a39d11ea14e037a1 | RegionOne | nova         |
| bdeec4e2665d4605bb89e16a8b1bc50d | RegionOne | glance       |
| ced05a1c03ab44caa1a351ace95429e6 | RegionOne | neutron      |
| e79e3113ea544d039b3a6378e60bdf3f | RegionOne | nova         |
| f91ee44123954b6c82162dcd2d4fc965 | RegionOne | designate    |
+-----+-----+-----+
```

4. **openstack ptr record unset** コマンドを使用して PTR レコードを削除し、以下の必要な引数を指定します。

<floating_IP_ID>

<region>:<floating_IP_ID> の形式の Floating IP ID。

例

```
$ openstack ptr record unset RegionOne:5c02c519-4928-4a38-bd10-c748c200912f
```

検証

- PTR レコードが削除されたことを確認します。

```
$ openstack ptr record list
```

関連情報

- コマンドラインインターフェイスリファレンスの [ptr record unset](#)

第10章 DNS サービスのトラブルシューティング

Red Hat OpenStack Platform DNS サービス (designate) のログを確認し、いくつかの簡単なコマンドを使用することで、サービスが適切に実行されていることを確認できます。これらのアクションは、DNS サービスのトラブルシューティングの最初の手順です。

このセクションに含まれるトピックは次のとおりです。

- [「DNS サービスと BIND ログ」](#)
- [「DNS サービスプール設定のエクスポート」](#)
- [「利用可能な DNS サービスエンドポイントのリスト表示」](#)

10.1. DNS サービスと BIND ログ

Red Hat OpenStack Platform DNS サービス (designate) のログを確認すると、問題のトラブルシューティングに役立つ場合があります。

DNS サービスログは `/var/log/containers/designate` にあります。コンポーネントサービスごとに1つのログがあります。

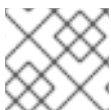
- `central.log`
- `designate-api.log`
- `mdns.log`
- `producer.log`
- `worker.log`

Red Hat が RHOSP DNS サービスで提供する BIND サーバーのログは、`/var/log/containers/designate/designate-bind` にあります。

- `central.log`
- `designate-api.log`

10.2. DNS サービスプール設定のエクスポート

DNS プール設定のコピーを使用して、Red Hat OpenStack Platform (RHOSP) DNS サービス (designate) のトラブルシューティングを実行できます。



注記

RHOSP 17.1 では、複数のプールはサポートされていません。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```


-
2. **designate-central** コンテナから **designate-manage pool** コマンドを実行します。まず、コントロールプレーン仮想マシンインスタンスの IP アドレスを取得します。

例

この例のコントローラーの名前は **overcloud-controller-0** です。

```
$ CTRL_IP=$(openstack server list -f value -c Networks \
--name overcloud-controller-0 | sed 's/ctlplane=//')
```

3. いずれかのコントローラーノードにログインし、現在実行中の DNS サービスプール設定をコンソールに表示します。

```
$ ssh tripleo-admin@${CTRL_IP} sudo podman exec -i -u root \
designate-central designate-manage pool show_config
```

出力例

```
Pool Configuration:
-----
also_notifies: []
attributes: {}
description: Default Pool
id: 794ccc2c-d751-44fe-b57f-8894c9f5c842
name: default
nameservers:
- host: 192.0.2.111
  port: 53
- host: 192.0.2.109
  port: 53
- host: 192.0.2.131
  port: 53
ns_records:
- hostname: ns2.example.com.
  priority: 2
- hostname: ns1.example.com.
  priority: 1
- hostname: ns3.example.com.
  priority: 3
targets:
- description: BIND9 Server 3
  masters:
  - host: 192.0.2.137
    port: 16002
  - host: 192.0.2.137
    port: 16001
  - host: 192.0.2.137
    port: 16000
options:
  host: 192.0.2.111
  port: '53'
  rndc_config_file: /etc/designate/private/bind3.conf
  rndc_host: 192.0.2.111
  rndc_port: '953'
```

```

type: bind9
- description: BIND9 Server 2
masters:
- host: 192.0.2.137
  port: 16001
- host: 192.0.2.137
  port: 16002
- host: 192.0.2.137
  port: 16000
options:
  host: 192.0.2.131
  port: '53'
  rndc_config_file: /etc/designate/private/bind2.conf
  rndc_host: 192.0.2.131
  rndc_port: '953'
type: bind9
- description: BIND9 Server 1
masters:
- host: 192.0.2.137
  port: 16002
- host: 192.0.2.137
  port: 16001
- host: 192.0.2.137
  port: 16000
options:
  host: 192.0.2.109
  port: '53'
  rndc_config_file: /etc/designate/private/bind1.conf
  rndc_host: 192.0.2.109
  rndc_port: '953'
type: bind9

```

4. 現在のプール設定をファイルにエクスポートするには、**designate-manage pool generated_file** コマンドを使用します。

例

```

$ sudo podman exec -i designate-manage pool generate_file \
--file ~/my_dns_service_config.yaml

```

ヒント

podman cp コマンドを使用して、コンテナからローカルシステムにファイルをコピーします。

10.3. 利用可能な DNS サービスエンドポイントのリスト表示

Red Hat OpenStack Platform DNS サービス (designate) エンドポイントとそのステータスを特定すると、問題のトラブルシューティングに役立ちます。

手順

1. Source コマンドで認証情報ファイルを読み込みます。

例

```
$ source ~/overcloudrc
```

2. RHOSP サービスエンドポイントをリスト表示します。

```
$ openstack endpoint list -c "Service Name" -c Enabled -c URL
```

出力例

```
+-----+-----+-----+
| Service Name | Enabled | URL                                     |
+-----+-----+-----+
| swift       | True   | http://198.51.100.61:8080             |
| designate   | True   | http://203.0.113.103:9001            |
| heat-cfn    | True   | http://192.0.2.137:8000/v1           |
| designate   | True   | http://192.0.2.137:9001              |
| placement   | True   | http://203.0.113.103:8778/placement   |
| cinderv3    | True   | http://203.0.113.103:8776/v3/%(tenant_id)s |
| heat        | True   | http://203.0.113.103:8004/v1/%(tenant_id)s |
| heat-cfn    | True   | http://203.0.113.103:8000/v1         |
| nova        | True   | http://203.0.113.103:8774/v2.1       |
| heat        | True   | http://192.0.2.137:8004/v1/%(tenant_id)s |
| glance      | True   | http://203.0.113.103:9292            |
| heat        | True   | http://203.0.113.103:8004/v1/%(tenant_id)s |
| glance      | True   | http://203.0.113.103:9292            |
| neutron     | True   | http://203.0.113.103:9696            |
| nova        | True   | http://192.0.2.137:8774/v2.1         |
| cinderv3    | True   | http://192.0.2.137:8776/v3/%(tenant_id)s |
| placement   | True   | http://203.0.113.103:8778/placement   |
| keystone    | True   | http://192.168.24.17:35357           |
| neutron     | True   | http://192.0.2.137:9696              |
| nova        | True   | http://203.0.113.103:8774/v2.1       |
| heat-cfn    | True   | http://203.0.113.103:8000/v1         |
| cinderv3    | True   | http://203.0.113.103:8776/v3/%(tenant_id)s |
| glance      | True   | http://192.0.2.137:9292              |
| placement   | True   | http://192.0.2.137:8778/placement     |
| swift       | True   | http://198.51.100.61:8080/v1/AUTH_%(tenant_id)s |
| swift       | True   | http://192.0.2.137:8080/v1/AUTH_%(tenant_id)s |
| designate   | True   | http://203.0.113.103:9001            |
| keystone    | True   | http://192.0.2.137:5000              |
| neutron     | True   | http://203.0.113.103:9696            |
| keystone    | True   | http://203.0.113.103:5000            |
+-----+-----+-----+
```