



Red Hat OpenStack Platform 16.2

オーバークラウドの IPv6 ネットワーク

IPv6 ネットワークを使用するようにオーバークラウドを設定する

Red Hat OpenStack Platform 16.2 オーバークラウドの IPv6 ネットワーク

IPv6 ネットワークを使用するようにオーバークラウドを設定する

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

本ガイドでは、Red Hat OpenStack Platform director を使用して、IPv6 をエンドポイントに使用するオーバークラウドを作成する方法を説明します。また、director が IPv6 ベースのオーバークラウドをデプロイする方法や、これを実現する設定オプションについて説明します。

目次

多様性を受け入れるオープンソースの強化	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
第1章 オーバークラウドの IPV6 の概要	5
1.1. IPV6 ネットワークの概要	5
1.2. RED HAT OPENSTACK PLATFORM での IPV6 の使用	7
第2章 IPV6 用オーバークラウドの設定	10
2.1. アンダークラウドの IPV6 アドレス設定	10
2.2. IPV6 デプロイメント用ノードの登録および検査	11
2.3. IPV6 デプロイメント用ノードのタグ付け	13
2.4. IPV6 ネットワークの設定	14
2.5. IPV6 オーバークラウドのデプロイ	17
第3章 デプロイメント後の IPV6 操作	19
3.1. オーバークラウドでの IPV6 プロジェクトネットワークの作成	19
3.2. オーバークラウドでの IPV6 パブリックネットワークの作成	20

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 オーバークラウドの IPV6 の概要

Red Hat OpenStack Platform director は、オーバークラウドと呼ばれるクラウド環境を作成します。デフォルトでは、オーバークラウドは、インターネットプロトコルバージョン 4 (IPv4) を使用してサービスのエンドポイントを設定します。ただし、オーバークラウドはインターネットプロトコルバージョン 6 (IPv6) のエンドポイントもサポートします。これは、IPv6 のインフラストラクチャーをサポートする組織には便利です。

本ガイドは、[director のインストールと使用方法](#) の補足情報を提供します。本ガイドでは [director のインストールと使用方法](#) と同じ要件が適用されます。必要に応じて、これらの要件を実装してください。

1.1. IPV6 ネットワークの概要

IPv6 は、最新バージョンのインターネットプロトコルです。Internet Engineering Task Force (IETF) は、現在の共通 IPv4 規格における IP アドレスの枯渇という問題に対応するために IPv6 を開発しました。IPv6 には、次のような IPv4 との違いがあります。

大規模な IP アドレス範囲

IPv6 範囲は IPv4 範囲よりもはるかに大きくなります。

エンドツーエンドの接続性の改善

IP 範囲が広がるとネットワークアドレス変換への依存度が低くなるため、エンドツーエンドの接続性が向上します。

ブロードキャスト非対応

IPv6 は従来の IP ブロードキャストに対応していません。代わりに、IPv6 ではマルチキャストを使用して、階層的な方法で適切なホストにパケットを送信します。

ステートレスアドレス自動設定 (SLAAC)

IPv6 には、IP アドレスを自動設定し、ネットワーク上の重複するアドレスを検出する機能があります。これにより、アドレスを割り当てる際の DHCP サーバーへの依存が軽減されます。

IPv6 は 128 ビット (16 ビットのグループを使用する 4 進数で表現) を使用してアドレスを定義します。一方、IPv4 は 32 ビット (グループの 8 ビットを使用した 10 進数の数字で表現) を使用します。たとえば、IPv4 アドレス (192.168.0.1) 表現は以下ようになります。

Bits	表現
11000000	192
10101000	168
00000000	0
00000001	1

IPv6 アドレス (2001:db8:88ec:9fb3::1) の場合、表現は以下ようになります。

Bits	表現
0010 0000 0000 0001	2001

Bits	表現
0000 1101 1011 1000	Odb8
1000 1000 1110 1100	88ec
1001 1111 1011 0011	9fb3
0000 0000 0000 0000	0000
0000 0000 0000 0000	0000
0000 0000 0000 0000	0000
0000 0000 0000 0001	0001

各ビットグループの先頭の 0 なしで IPv6 アドレスを表現したり、各 IP アドレスの 0 のビットグループをすべて省略したりすることも可能です。この例では、**0db8** ビットグループを **db8** と表現し、**0000** ビットグループ 3 個を省略して、**2001:0db8:88ec:9fb3:0000:0000:0000:0001** を **2001:db8:88ec:9fb3::1** に短縮できます。詳細は [RFC 5952: A Recommendation for IPv6 Address Text Representation](#) を参照してください。

IPv6 でのサブネット化

IPv4 と同様に、IPv6 アドレスではビットマスクを使用してアドレスの接頭辞をネットワークとして定義します。たとえば、サンプル IP アドレス **2001:db8:88ec:9fb3::1/64** に /64 ビットマスクを追加すると、最初の 64 ビット (**2001:db8:88ec:9fb3**) をネットワークとして定義する接頭辞としてビットマスクが動作します。残りのビット (**0000:0000:0000:0001**) はホストを定義します。

IPv6 は、いくつかの特別なアドレスタイプを使用します。

ループバック

ループバックデバイスは、ホスト内の内部通信に IPv6 を使用します。このデバイスは常に **::1/128** になります。

リンクローカル

リンクローカルアドレスは、特定のネットワークセグメント内で有効な IP アドレスです。IPv6 では、各ネットワークデバイスにリンクローカルアドレスがあり、接頭辞 **fe80::/10** を使用する必要があります。ただし、ほとんどの場合、これらのアドレスには **fe80::/64** の接頭辞が付けられます。

一意のローカル

一意のローカルアドレスは、ローカル通信用です。これらのアドレスは、**fc00::/7** 接頭辞を使用します。

マルチキャスト

ホストはマルチキャストアドレスを使用してマルチキャストグループに参加します。これらのアドレスは、**ff00::/8** 接頭辞を使用します。たとえば、**FF02::1** はネットワーク上の全ノードのマルチキャストグループで、**FF02::2** はすべてのルーターのマルチキャストグループです。

グローバルユニキャスト

通常、これらのアドレスはパブリック IP アドレス用に予約されます。これらのアドレスは **2000::/3** 接頭辞を使用します。

1.2. RED HAT OPENSTACK PLATFORM での IPV6 の使用

Red Hat OpenStack Platform(RHOSP)director は、OpenStack サービスを分離ネットワークにマッピングします。これらのネットワークには、以下が含まれます。

- Internal API
- ストレージ
- Storage Management
- プロジェクト (テナント) ネットワーク (Neutron VLAN モード)
- 外部

ネットワークトラフィックタイプの詳細は、[director のインストールと使用方法](#) を参照してください。

director では、ネットワークに IPv6 通信を使用することもできます。つまり、必要な OpenStack サービス、データベース、その他の関連サービスは IPv6 アドレスを使用して通信できます。これは、複数のコントローラーノードを含む高可用性ソリューションを使用する環境にも適用されます。これにより、組織は RHOSP を IPv6 インフラストラクチャーと統合しやすくなります。

次の表は、RHOSP ネットワークの IPv6 対応に関するガイドとして使用できます。

ネットワークタイプ	サポートされているインターネット プロトコル (IP)	備考
Internal API	<ul style="list-style-type: none"> • IPv6 • IPv4 	
ストレージ	<ul style="list-style-type: none"> • IPv6 • IPv4 	
Storage Management	<ul style="list-style-type: none"> • IPv6 • IPv4 	
プロジェクトネットワーク	<ul style="list-style-type: none"> • デュアルスタック (IPv4/v6) • IPv6 • IPv4 	

ネットワークタイプ	サポートされているインターネット プロトコル (IP)	備考
プロジェクトネットワークのエンドポイント	<ul style="list-style-type: none"> ● デュアルスタック (IPv4/v6) ● IPv6 ● IPv4 	<p>プロジェクトネットワーク自体ではなく、プロジェクトネットワークトンネルをホストするネットワークの IP アドレスを参照します。</p> <p>ネットワークエンドポイントの IPv6 は、VXLAN および Geneve のみサポートしています。GRE (Generic Routing Encapsulation) はまだサポートされていません。</p>
外部 - パブリック API (および Horizon)	<ul style="list-style-type: none"> ● IPv6 ● IPv4 	
外部 - Floating IP	<ul style="list-style-type: none"> ● デュアルスタック (IPv4/v6) ● IPv4 	<p>IPv6 は、NAT および Floating IP アドレスの代わりにグローバルユニキャストアドレス (GUA) を使用します。ネットワーキング (neutron) サービスは、プロジェクトネットワーク間の IPv6 アドレスが GUA を使用することを想定しており、プロジェクトネットワーク間で GUA が重複することはないため、NAT なしでルーティングできます。</p> <p>デュアルスタック (IPv4/v6) では、Floating IP アドレスを使用して、IPv4 サブネット上の IP アドレスにのみ到達できます。</p>
プロバイダーネットワーク	<ul style="list-style-type: none"> ● デュアルスタック (IPv4/v6) ● IPv6 ● IPv4 	<p>IPv6 サポートはプロジェクトのオペレーティングシステムによって異なります。</p>
プロビジョニング (PXE/DHCP)	<ul style="list-style-type: none"> ● IPv6 ● IPv4 	

ネットワークタイプ	サポートされているインターネット プロトコル (IP)	備考
IPMI またはその他の BMC	<ul style="list-style-type: none"> IPv6 IPv4 	<p>RHOSP は、プロビジョニングネットワークを通じてベースボード管理コントローラー (BMC) インターフェイスと通信します。</p> <p>BMC インターフェイスがデュアルスタック IPv4 または IPv6 に対応している場合、RHOSP に含まれていないツールは IPv6 を使用して BMC と通信できます。</p>
オーバークラウドプロビジョニングネットワーク	IPv6	オーバークラウドのベアメタルプロビジョニングサービスに使用されるプロビジョニングネットワーク。
オーバークラウドクリーニングネットワーク	None	再利用の準備が整うまでマシンをクリーンアップするために使用される分離ネットワーク。

シナリオの定義

このガイドのシナリオでは、IPv6 を使用する分離ネットワークでオーバークラウドを作成します。heat テンプレートと環境ファイルを使用して、ネットワーク分離を設定します。このシナリオでは、設定の違いを示すために、異なる heat テンプレートと環境ファイルも指定します。

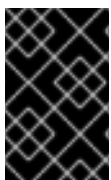


注記

このシナリオでは、アンダークラウドは引き続き PXE ブート、イントロスペクション、デプロイメント、およびその他のサービスに IPv4 の接続を使用します。

このガイドでは、高度なオーバークラウドシナリオと同様のシナリオを使用します。主な違いは、Ceph Storage ノードが省略される点です。

このシナリオの詳細は、[director のインストールと使用方法](#) を参照してください。



重要

このガイドでは、[RFC 3849](#) で定義されているとおり、このガイドではドキュメント用に 2001:DB8::/32 IPv6 接頭辞を使用します。これらのサンプルアドレスは、独自のネットワークの IPv6 アドレスに置き換えてください。

第2章 IPV6 用オーバークラウドの設定

この章では、**openstack overcloud deploy** コマンドを実行する前に必要となる設定について説明します。これには、プロビジョニング用のノードの準備、アンダークラウドでの IPv6 アドレスの設定、およびオーバークラウドの IPv6 パラメーターを定義するためのネットワーク環境ファイルの作成が含まれます。

前提条件

- アンダークラウドの正常なインストール。詳細は、[アンダークラウドへの director のインストール](#) を参照してください。
- 使用しているネットワークが IPv6 ネイティブ VLAN と IPv4 ネイティブ VLAN をサポートしている。

2.1. アンダークラウドの IPV6 アドレス設定

アンダークラウドには、オーバークラウドのパブリック API (外部ネットワーク) へのアクセスが必要です。そのためには、アンダークラウドのホストに、外部ネットワークに接続するインターフェイスで IPv6 アドレスが必要です。

前提条件

- アンダークラウドの正常なインストール。詳細は、[アンダークラウドへの director のインストール](#) を参照してください。
- 使用しているネットワークが IPv6 ネイティブ VLAN と IPv4 ネイティブ VLAN をサポートしている。
- アンダークラウドで利用可能な IPv6 アドレス

ネイティブ VLAN または専用インターフェイス

アンダークラウドがネイティブ VLAN または外部ネットワークに接続されている専用インターフェイスを使用する場合は、**ip** コマンドを使用して、インターフェイスに IPv6 アドレスを追加します。以下の例の専用インターフェイスは **eth0** です。

```
$ sudo ip link set dev eth0 up; sudo ip addr add 2001:db8::1/64 dev eth0
```

とランキングされた VLAN インターフェイス

外部ネットワークにアクセスするためのコントロールプレーンブリッジ (**br-ctlplane**) と同じインターフェイスで、とランキングされた VLAN がアンダークラウドにより使用される場合は、新しい VLAN インターフェイスを作成して、コントロールプレーンにアタッチし、IPv6 アドレスを VLAN に追加します。以下の例の場合、外部ネットワークの VLAN ID は **100** です。

```
$ sudo ovs-vsctl add-port br-ctlplane vlan100 tag=100 -- set interface vlan100 type=internal
$ sudo ip l set dev vlan100 up; sudo ip addr add 2001:db8::1/64 dev vlan100
```

IPv6 アドレスの確認

ip コマンドを使用して、IPv6 アドレスが追加されたことを確認します。

```
$ ip addr
```

IPv6 アドレスは、選択したインターフェイスに表示されます。

永続的な IPv6 アドレスの設定

IPv6 アドレスを永続化するには、`/etc/sysconfig/network-scripts/` で適切なインターフェイスファイルを修正または作成します。この例では、`ifcfg-eth0` または `ifcfg-vlan100` のいずれかに以下の行を追加します。

```
IPV6INIT=yes
IPV6ADDR=2001:db8::1/64
```

詳細は、Red Hat カスタマーポータル [の IPv6 用ネットワークインターフェイスの設定方法](#) を参照してください。

2.2. IPV6 デプロイメント用ノードの登録および検査

ノード定義のテンプレート (`instackenv.json`) は JSON ファイル形式で、ノード登録用のハードウェアおよび電源管理の情報が含まれています。以下に例を示します。

```
{
  "nodes":[
    {
      "mac":[
        "bb:bb:bb:bb:bb:bb"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.205"
    },
    {
      "mac":[
        "cc:cc:cc:cc:cc:cc"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.206"
    },
    {
      "mac":[
        "dd:dd:dd:dd:dd:dd"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
```

```

    "pm_type":"ipmi",
    "pm_user":"admin",
    "pm_password":"p@55w0rd!",
    "pm_addr":"192.0.2.207"
  },
  {
    "mac":[
      "ee:ee:ee:ee:ee:ee"
    ],
    "cpu":"4",
    "memory":"6144",
    "disk":"40",
    "arch":"x86_64",
    "pm_type":"ipmi",
    "pm_user":"admin",
    "pm_password":"p@55w0rd!",
    "pm_addr":"192.0.2.208"
  }
  {
    "mac":[
      "ff:ff:ff:ff:ff:ff"
    ],
    "cpu":"4",
    "memory":"6144",
    "disk":"40",
    "arch":"x86_64",
    "pm_type":"ipmi",
    "pm_user":"admin",
    "pm_password":"p@55w0rd!",
    "pm_addr":"192.0.2.209"
  }
  {
    "mac":[
      "gg:gg:gg:gg:gg:gg"
    ],
    "cpu":"4",
    "memory":"6144",
    "disk":"40",
    "arch":"x86_64",
    "pm_type":"ipmi",
    "pm_user":"admin",
    "pm_password":"p@55w0rd!",
    "pm_addr":"192.0.2.210"
  }
]
}

```

前提条件

- アンダークラウドの正常なインストール。詳細は、[アンダークラウドへの director のインストール](#) を参照してください。
- オーバークラウドデプロイメントで利用可能なノード。

手順

1. ノード定義のテンプレートの作成後に、stack ユーザーのホームディレクトリー (`/home/stack/instackenv.json`) にそのファイルを保存し、director にインポートします。

```
$ openstack overcloud node import ~/instackenv.json
```

このコマンドはテンプレートをインポートし、テンプレートから director に各ノードを登録します。

2. カーネルと ramdisk イメージを全ノードに割り当てます。

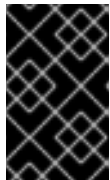
```
$ openstack overcloud node configure
```

これで director でのノードの登録、設定が完了しました。

検証手順

- ノードの登録後に、各ノードのハードウェア属性を確認します。

```
$ openstack overcloud node introspect --all-manageable
```



重要

ノードは **manageable** の状態である必要があります。このプロセスが最後まで実行されて正常に終了したことを確認してください。ベアメタルノードの場合は、通常 15 分ほどかかります。

2.3. IPV6 デプロイメント用ノードのタグ付け

各ノードのハードウェアを登録および検査した後、特定のプロファイルに各ノードをタグ付けします。このプロファイルタグにより、ノードがフレーバーにマッピングされ、次にそのフレーバーがデプロイメントロールに割り当てられます。

前提条件

- アンダークラウドの正常なインストール。詳細は、[Installing director on the undercloud](#) を参照してください。

手順

1. ノードリストを取得して UUID を把握します。

```
$ ironic node-list
```

2. 各ノードの **properties/capabilities** パラメーターに profile オプションを追加します。たとえば、3つのノードがコントローラープロファイルを使用し、3つのノードがコンピュートプロファイルを使用するには、以下のコマンドを実行します。

```
$ ironic node-update 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 6faba1a9-e2d8-4b7c-95a2-c7fbc12129a add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 5e3b2f50-fcd9-4404-b0a2-59d79924b38e add
properties/capabilities='profile:control,boot_option:local'
```

```
$ ironic node-update 484587b2-b3b3-40d5-925b-a26a2fa3036f add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d010460b-38f2-4800-9cc4-d69f0d067efe add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d930e613-3e14-44b9-8240-4f3559801ea6 add
properties/capabilities='profile:compute,boot_option:local'
```

profile:control と **profile:compute** オプションを追加することで、これらのノードがそれぞれのプロファイルにタグ付けされます。



注記

手でタグを付ける代わりに、自動プロファイルタグ付け機能を使用し、ベンチマークデータに基づいて、多数のノードに自動でタグ付けします。

2.4. IPV6 ネットワークの設定

本項では、オーバークラウドのネットワーク設定を検証します。これには、OpenStack サービスを分離して、特定のネットワークトラフィックを使用し、IPv6 オプションを使用したオーバークラウドの設定が含まれます。

2.4.1. コンポーザブル IPv6 ネットワークの設定

前提条件

- アンダークラウドの正常なインストール。詳細は、[アンダークラウドへの director のインストール](#) を参照してください。
- 使用しているネットワークが IPv6 ネイティブ VLAN と IPv4 ネイティブ VLAN をサポートしている。

手順

1. デフォルトの **network_data** ファイルをコピーします。

```
$ cp /usr/share/openstack-tripleo-heat-templates/network_data.yaml /home/stack/.
```

2. **network_data.yaml** ファイルのローカルコピーを編集し、IPv6 ネットワーク要件に応じてパラメーターを変更します。たとえば、外部ネットワークには以下のデフォルトネットワーク情報が含まれます。

```
- name: External
  vip: true
  name_lower: external
  vlan: 10
  ipv6: true
  ipv6_subnet: '2001:db8:fd00:1000::/64'
  ipv6_allocation_pools: [{'start': '2001:db8:fd00:1000::10', 'end':
'2001:db8:fd00:1000:ffff:ffff:ffff:ffff'}]
  gateway_ipv6: '2001:db8:fd00:1000::1'
```

- **name** は、唯一の必須の値です。ただし、**name_lower** を使用して名前を正規化し、読みやすくできます。たとえば、**InternalApi** を **internal_api** に変更します。

- **vip: true** は、新規ネットワークのデフォルトを設定する残りのパラメーターを使用して、新規ネットワーク上に仮想 IP アドレス (VIP) を作成します。
 - **ipv6** は、IPv6 を有効にするかどうかを定義します。
 - **ipv6_subnet** と **ipv6_allocation_pools**、および **gateway_ip6** は、ネットワークのデフォルト IPv6 サブネットおよび IP 範囲を設定します。
3. **-n** オプションを使用して、カスタム **network_data** ファイルをデプロイメントに含めます。**-n** オプションを設定しない場合、デプロイメントコマンドはデフォルトのネットワーク情報を使用します。

2.4.2. オーバークラウドでの IPv6 ネットワーク分離

デフォルトでは、オーバークラウドはサービスをプロビジョニングネットワークに割り当てます。ただし、director はオーバークラウドのネットワークトラフィックを分離したネットワークに分割することができます。これらのネットワークは、デプロイメントコマンドラインに追加するファイルで定義されます。デフォルトのファイル名は **network_data.yaml** です。

サービスが IPv6 アドレスを使用してネットワークをリッスンしている場合は、パラメーターのデフォルト値を指定して、そのサービスが IPv6 ネットワーク上で実行されていることを示す必要があります。各サービスが実行されるネットワークは、**network/service_net_map.yaml** のファイルで定義され、**ServiceNetMap** エントリーのパラメーターのデフォルトを個別に宣言することで上書きできます。サービスでは、パラメーターのデフォルトを環境ファイルで設定する必要があります。

```
parameter_defaults:
  # Enable IPv6 for Ceph.
  CephIPv6: True
  # Enable IPv6 for Corosync. This is required when Corosync is using an IPv6 IP in the cluster.
  CorosyncIPv6: True
  # Enable IPv6 for MongoDB. This is required when MongoDB is using an IPv6 IP.
  MongoDbIPv6: True
  # Enable various IPv6 features in Nova.
  NovalIPv6: True
  # Enable IPv6 environment for RabbitMQ.
  RabbitIPv6: True
  # Enable IPv6 environment for Memcached.
  MemcachedIPv6: True
  # Enable IPv6 environment for MySQL.
  MysqlIPv6: True
  # Enable IPv6 environment for Manila
  ManilIPv6: True
  # Enable IPv6 environment for Redis.
  RedisIPv6: True
```

コア heat テンプレートの **environments/network-isolation.j2.yaml** ファイルは Jinja2 形式のファイルで、コンポーザブルネットワークファイル内の各 IPv6 ネットワークのポートおよび仮想 IP をすべて定義します。レンダリングすると、すべてのリソースレジストリーと共に **network-isolation.yaml** ファイルが同じ場所に生成されます。

2.4.3. IPv6 分離ネットワークの設定

デフォルトの heat テンプレートコレクションには、デフォルトのネットワーク設定用に Jinja2 ベースの環境ファイルが含まれます。このファイルは **environments/network-environment.j2.yaml** です。**network_data** ファイルでレンダリングすると、**network-environment.yaml** という標準の YAML

ファイルが作成されます。このファイルの一部ではオーバーライドが必要となる可能性があります。

前提条件

- アンダークラウドの正常なインストール。詳細は、[アンダークラウドへの director のインストール](#) を参照してください。
- 使用しているネットワークが IPv6 ネイティブ VLAN と IPv4 ネイティブ VLAN をサポートしている。

手順

- カスタム環境ファイル (`/home/stack/network-environment.yaml`) を以下の内容で作成します。

```
parameter_defaults:
  DnsServers: ["8.8.8.8","8.8.4.4"]
  ControlPlaneDefaultRoute: 192.0.2.1
  ControlPlaneSubnetCidr: "24"
```

parameter_defaults セクションには、IPv4 上に残る特定のサービスのカスタマイズが含まれます。

2.4.4. IPv6 ネットワークインターフェイスのテンプレート

オーバークラウドには、ネットワークインターフェイステンプレートのセットが必要です。director には、**network_data** ファイルをベースにレンダリングされる Jinja2 ベースの Heat テンプレートセットが含まれています。

NIC ディレクトリー	説明	環境ファイル
single-nic-vlans	単一の NIC (nic1) がコントロールプレーンネットワークにアタッチされ、VLAN 経由でデフォルトの Open vSwitch ブリッジにアタッチされます。	environments/net-single-nic-with-vlans-v6.j2.yaml
single-nic-linux-bridge-vlans	単一の NIC (nic1) がコントロールプレーンネットワークにアタッチされ、VLAN 経由でデフォルトの Linux ブリッジにアタッチされます。	environments/net-single-nic-linux-bridge-with-vlans-v6.yaml
bond-with-vlans	コントロールプレーンネットワークが nic1 にアタッチされます。ボンディング設定の NIC (nic2 および nic3) が VLAN 経由でデフォルトの Open vSwitch ブリッジにアタッチされます。	environments/net-bond-with-vlans-v6.yaml

NIC ディレクトリー	説明	環境ファイル
multiple-nics	コントロールプレーンネットワークが nic1 にアタッチされます。それ以降の NIC は network_data ファイルで定義されるネットワークに割り当てられます。デフォルトでは、Storage が nic2 に、Storage Management が nic3 に、Internal API が nic4 に、Tenant が br-tenant ブリッジ上の nic5 に、External がデフォルトの Open vSwitch ブリッジ上の nic6 に割り当てられます。	environments/net-multiple-nics-v6.yaml

2.5. IPV6 オーバークラウドのデプロイ

IPv6 ネットワークを使用するオーバークラウドをデプロイするには、デプロイメントコマンドに別の引数を追加する必要があります。

前提条件

- アンダークラウドの正常なインストール。詳細は、[Installing director on the undercloud](#) を参照してください。

手順

```
$ openstack overcloud deploy --templates \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/net-single-nic-with-vlans.yaml \
-e /home/stack/templates/network-environment.yaml \
--ntp-server pool.ntp.org \
[ADDITIONAL OPTIONS]
```

上記のコマンドは、以下のオプションを使用します。

- **--templates** - デフォルトの Heat テンプレートコレクションからオーバークラウドを作成します。
- **-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml** - オーバークラウドデプロイメントに別の環境ファイルを追加します。ここでは、IPv6 向けのネットワーク分離設定を初期化する環境ファイルを追加します。
- **-e /usr/share/openstack-tripleo-heat-templates/environments/net-single-nic-with-vlans.yaml** - オーバークラウドのデプロイメントに別の環境ファイルを追加します。ここでは、IPv6 向けのネットワーク分離設定を初期化する環境ファイルを追加します。
- **-e /home/stack/network-environment.yaml** - オーバークラウドのデプロイメントに別の環境ファイルを追加します。この場合は、IPv6 に関するオーバーライドが含まれます。**network_data.yaml** ファイルに **ipv6: true** の設定が含まれていることを確認します。以前のバージョンの Red Hat OpenStack Platform director には、2つのデフォルトルートが含まれていました (外部ネットワーク上の IPv6 用ルートおよびコントロールプレーン上の IPv4 用ルー

ト)。両方のデフォルトルートを使用するには、**roles_data.yaml** ファイルの Controller 定義に **default_route_networks** パラメーターの両方のネットワークが含まれるようにしてください。
例: **default_route_networks: ['External', 'ControlPlane']**

- **--ntp-server pool.ntp.org** - NTP サーバーを設定します。

オーバークラウドの作成プロセスが開始され、director によりオーバークラウドノードがプロビジョニングされます。このプロセスは完了するまで多少時間がかかります。オーバークラウドの作成のステータスを確認するには、**stack** ユーザーとして別のターミナルを開き、以下のコマンドを実行します。

```
$ source ~/stackrc
$ heat stack-list --show-nested
```

オーバークラウドへのアクセス

director は、director ホストからオーバークラウドに対話するための設定を行い、認証をサポートするスクリプトを作成します。director は、このファイル (**overcloudrc**) を **stack** ユーザーのホームディレクトリーに保存します。このファイルを使用するには、以下のコマンドを実行します。

```
$ source ~/overcloudrc
```

これで、director のホストの CLI からオーバークラウドと対話するために必要な環境変数が読み込まれます。director のホストとの対話に戻るには、以下のコマンドを実行します。

```
$ source ~/stackrc
```

第3章 デプロイメント後の IPV6 操作

IPv6 ネットワークを使用してオーバークラウドをデプロイしたら、追加の設定を行う必要があります。

前提条件

- アンダークラウドの正常なインストール。詳細は、[Installing director on the undercloud](#) を参照してください。
- オーバークラウドの正常なデプロイメント。詳細は、[Creating a basic overcloud with CLI tools](#) を参照してください。

3.1. オーバークラウドでの IPV6 プロジェクトネットワークの作成

オーバークラウドには、インスタンス用の IPv6 ベースのプロジェクトネットワークが必要です。source コマンドで **overcloudrc** ファイルを読み込み、**Neutron** で初期プロジェクトネットワークを作成します。

前提条件

- アンダークラウドの正常なインストール。詳細は、[Installing director on the undercloud](#) を参照してください。
- オーバークラウドの正常なデプロイメント。詳細は、[Creating a basic overcloud with CLI tools](#) を参照してください。
- 使用しているネットワークが IPv6 ネイティブ VLAN と IPv4 ネイティブ VLAN をサポートしている。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/overcloudrc
```

2. カスタムネットワークおよびサブネットを作成します。

```
$ openstack network create default --external --provider-physical-network datacentre --  
provider-network-type vlan --provider-segment 101
```

```
$ openstack subnet create default --subnet-range 2001:db8:fd00:6000::/64 --ipv6-address-  
mode slaac --ipv6-ra-mode slaac --ip-version 6 --network default
```

上記のステップにより、**default** という名前の基本的な **neutron** ネットワークが作成されます。

検証手順

- ネットワークが正常に作成されたことを確認します。

```
$ openstack network list  
$ openstack subnet list
```

3.2. オーバークラウドでの IPv6 パブリックネットワークの作成

ノードインターフェイスが外部ネットワークを使用するように設定したら、オーバークラウドでこのネットワークを作成して、ネットワークアクセスを有効にする必要があります。

前提条件

- アンダークラウドの正常なインストール。詳細は、[Installing director on the undercloud](#) を参照してください。
- オーバークラウドの正常なデプロイメント。詳細は、[Creating a basic overcloud with CLI tools](#) を参照してください。
- 使用しているネットワークが IPv6 ネイティブ VLAN と IPv4 ネイティブ VLAN をサポートしている。

手順

1. 外部ネットワークおよびサブネットを作成します。

```
$ openstack network create public --external --provider-physical-network datacentre --  
provider-network-type vlan --provider-segment 100
```

```
$ openstack subnet create public --network public --subnet-range 2001:db8:0:2::/64 --ip-  
version 6 --gateway 2001:db8::1 --allocation-pool  
start=2001:db8:0:2::2,end=2001:db8:0:2::ffff --ipv6-address-mode slaac --ipv6-ra-mode slaac
```

このコマンドは、**public** と呼ばれるネットワークを作成します。このネットワークは、インスタンスに 65000 個以上 IPv6 アドレスが含まれる割当プールを提供します。

2. インスタンストラフィックを外部ネットワークにルーティングするためのルーターを作成します。

```
$ openstack router create public-router  
$ openstack router set public-router --external-gateway public
```