



# Red Hat OpenStack Platform 11

## オーバークラウド向けの **IPv6** ネットワーク

オーバークラウドで IPv6 ネットワークを使用するための設定



# Red Hat OpenStack Platform 11 オーバークラウド向けの IPv6 ネットワーク

---

オーバークラウドで IPv6 ネットワークを使用するための設定

OpenStack Team  
rhos-docs@redhat.com

## 法律上の通知

Copyright © 2018 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, 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 Software Collections 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

本ガイドは、Red Hat OpenStack Platform director で、エンドポイントに IPv6 を使用するオーバークラウドを作成する方法について説明します。内容には、**director** で IPv6 ベースのオーバークラウドをデプロイする方法と、そのための設定オプションが含まれます。

---

目次

第1章 はじめに .....	3
1.1. IPV6 ネットワークの定義 .....	3
1.2. RED HAT OPENSTACK PLATFORM における IPV6 の使用 .....	5
1.3. 設定要件 .....	6
1.4. シナリオの定義 .....	7
第2章 オーバークラウドの作成前の設定 .....	8
2.1. STACK ユーザーの初期化 .....	8
2.2. アンダークラウド上での IPV6 アドレスの設定 .....	8
2.3. 環境の設定 .....	9
2.3.1. ノードの登録 .....	9
2.3.2. ノードのハードウェアの検査 .....	11
2.3.3. ノードの手動でのタグ付け .....	11
2.4. ネットワークの設定 .....	12
2.4.1. インターフェースの設定 .....	12
2.4.2. 分離された IPV6 ネットワークの設定 .....	13
2.4.3. IPV6 と IPv4 のハイブリッド設定の使用 .....	14
2.5. オーバークラウドの設定の完了 .....	16
第3章 オーバークラウドの作成 .....	17
3.1. オーバークラウドへのアクセス .....	18
第4章 オーバークラウドの作成後の設定 .....	19
4.1. オーバークラウドのテナントネットワークの作成 .....	19
4.2. オーバークラウドのパブリックネットワークの作成 .....	19
第5章 結果 .....	20



# 第1章 はじめに

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のバージョン 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が、オーバークラウドはインターネットプロトコルのバージョン 6 (IPv6) のエンドポイントもサポートしています。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。本ガイドには、オーバークラウドで IPv6 を使用するための情報と設定例を記載します。

## 1.1. IPV6 ネットワークの定義

IPv6 は、インターネットプロトコル基準の最新バージョンです。Engineering Task Force (IETF) では、現在一般的な IPv4 基準で IP アドレスが使い果たされてしまわないようにするための手段として、IPv6 を開発しました。IPv6 と IPv4 の間では、以下を含むさまざまな相違点があります。

### より大きな IP アドレス範囲

IPv6 のアドレス範囲は、IPv4 の範囲よりもはるかに大きくなります。

### より優れたエンドツーエンドの接続性

IP の範囲が大きいので、ネットワークアドレスの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されます。

### ブロードキャストなし

IPv6 は、従来の IP ブロードキャストをサポートしません。その代わりに、IPv6 はマルチキャストを使用して、階層的な方法で対象のホストにパケットを送信します。

### Stateless Address Autoconfiguration (SLAAC)

IPv6 は、IP アドレスを自動的に設定して、1つのネットワーク上で重複したアドレスを検出する機能を提供します。これにより、アドレスの割り当てでの DHCP サーバーへの依存度が軽減されます。

IPv6 では 128 ビット (16 ビットのグループを使用して、4 つの 16 進数で表現) を使用してアドレスを定義しますが、IPv4 では 32 ビットのみ (8 ビットのグループを使用して、10 進数の桁で表現) を使用します。たとえば、IPv4 アドレス (192.168.0.1) の表現は以下のようになります。

ビット	表現
11000000	192
10101000	168
00000000	0
00000001	1

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

ビット	表現
0010 0000 0000 0001	2001

ビット	表現
0000 1101 1011 1000	0db8
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

IPv6 アドレスは、各ビットグループで先頭にゼロを付けずに表現し、IP アドレスごとのゼロのビットグループセットを省略することも可能である点に注意してください。たとえば、**0683** ビットグループを **683** として表現し、**0000** のビットグループセットを **3** つ省くことが可能です。これにより、表現は、「**2001:0db8:88ec:9fb3:0000:0000:0000:0001**」から「**2001:db8:88ec:9fb3::1**」に短くすることができます。

## IPv6 でのサブネット化

IPv4 と同様に、IPv6 アドレスはビットマスクを使用して、アドレスプレフィックスをネットワークとして定義します。たとえば、**/64** ビットマスクをサンプルの IP アドレス (例: **2001:db8:88ec:9fb3::1/64**) に追加すると、そのビットマスクは最初の **64** ビット (**2001:db8:88ec:9fb3**) をネットワークとして定義するプレフィックスとして機能します。残りのビット (**0000:0000:0000:0001**) はホストを定義します。

IPv6 は、以下を含む特殊なアドレスタイプもいくつか使用します。

### ループバック

ループバックデバイスは、ホスト内における内部の通信に IPv6 を使用します。このデバイスは常に **::1/128** です。

### 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1つ割り当てられ、プレフィックス **fe80::/10** を使用する必要があります。ただし、大半の場合には、このアドレスには **fe80::/64** のプレフィックスが付きます。

### ユニークローカル

ローカル通信向けの一意なローカルアドレス。このアドレスは、**fc00::/7** プレフィックスを使用します。

### 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。このアドレスは、**ff00::/8** のプレフィックスを使用します。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** は全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



このアドレスは、通常パブリック IP アドレス用に予約され、2000::/3 のプレフィックスを使用します。

## 1.2. RED HAT OPENSTACK PLATFORM における IPV6 の使用

Red Hat OpenStack Platform director は、OpenStack サービスを分離されたネットワークにマッピングする手段を提供します。これらのネットワークには、以下が含まれます。

- 内部 API
- ストレージ
- ストレージ管理
- 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のタイプについての説明は、『[director のインストールと使用方法](#)』を参照してください。

Red Hat OpenStack Platform director は、これらのネットワークに IPv6 通信を使用する手段も提供します。これは、必要な OpenStack サービス、データベース、その他の関連サービスが通信に IPv6 アドレスを使用することを意味し、複数のコントローラーを伴う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。この機能は、組織が Red Hat OpenStack Platform を IPv6 インフラストラクチャーに統合するのに役立ちます。

Red Hat OpenStack Platform で IPv6 をサポートしているネットワークについては、以下の表を参考にしてください。

ネットワーク種別	デュアルスタック (IPv4/v6)	シングルスタック (IPv6)	シングルスタック (IPv4)	備考
内部 API		はい	はい	
ストレージ		はい	はい	
ストレージ管理		はい	はい	
テナントネットワーク	はい	はい	はい	
テナントネットワークエンドポイント			はい	これは、テナントネットワークトンネルをホストしているネットワークの IP アドレスを指します。
外部 - パブリック API (および Horizon)		はい	はい	

ネットワーク種別	デュアルスタック (IPv4/v6)	シングルスタック (IPv6)	シングルスタック (IPv4)	備考
外部 - Floating IP	はい [1]	はい [1]	はい	
プロバイダーネットワーク	はい	はい	はい	IPv6 のサポートは、テナントの OS によって異なります。
プロビジョニング (PXE/DHCP)			はい	このネットワークのインターフェースは、IPv4 のみです。
IPMI またはその他の BMC			はい	OpenStack Platform は、プロビジョニングネットワーク上で BMC インターフェースと通信します。このネットワークは IPv4 ですが、BMC インターフェースがデュアルスタック IPv4/IPv6 をサポートしている場合には、OpenStack Platform 以外のツールは IPv6 を使用して BMC と通信することができます。
オーバークラウド プロビジョニング ネットワーク				オーバークラウド内の Ironic 用の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				マシンを再使用する前にクリーニングするための分離されたネットワーク

[1] Global Unicast Address (GUA) のプレフィックスおよびアドレスを割り当てられた **Neutron** テナントネットワークは、**neutron** ルーターの外部ゲートウェイポートが外部にアクセスできる必要はありません。

### 1.3. 設定要件

本ガイドは、『[Red Hat OpenStack Platform 11 director のインストールと使用方法](#)』ガイドの補足情報としての役割を果たします。これは、『[Red Hat OpenStack Platform 11 director のインストールと使用方法](#)』で指定されているのと同じ要件が本ガイドにも適用されることを意味します。必要に応じて、この要件を実装してください。

本ガイドでは、以下の要件も満たす必要があります。

- Red Hat OpenStack Platform director がインストールされたアンダークラウドホスト。『[Red Hat OpenStack Platform 11 director のインストールと使用方法](#)』ガイドを参照してください。
- お使いのネットワークが IPv6 のネイティブ VLAN と IPv4 のネイティブ VLAN をサポートしていること。デプロイメントには、両方が使用されます。

## 1.4. シナリオの定義

本ガイドのシナリオでは、オーバークラウドを作成して、IPv6 を使用する分離されたネットワークを設定します。本書では、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の分離を設定することにより、この目的を達成します。本シナリオでは、Heat テンプレートと環境ファイルのいくつかの異なる実例をあげて、設定の具体的な相違点について説明します。



### 注記

このシナリオでは、アンダークラウドは、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は IPv4 の接続を引き続き使用します。

本ガイドでは、『[Red Hat OpenStack Platform 11 director のインストールと使用方法](#)』ガイドの高度なオーバークラウドのシナリオと同様のシナリオを使用します。主な相違点は、Ceph ストレージノードが省略されていることです。

このシナリオに関する詳しい情報は、『[Red Hat OpenStack Platform 11 director のインストールと使用方法](#)』を参照してください。



### 重要

本ガイドでは、説明の目的で、[RFC 3849](#) で定義されている 2001:DB8::/32 IPv6 プレフィックスを使用します。これらの IPv6 アドレスの例は、お使いのネットワークに応じて置き換えるようにしてください。

## 第2章 オーバークラウドの作成前の設定

本章では、**openstack overcloud deploy** コマンドを実行する前に必要な設定について説明します。これには、プロビジョニングに向けたノードの準備、アンダークラウド上での **IPv6** アドレスの設定、オーバークラウド用の **IPv6** パラメーターを定義するネットワーク環境ファイルの作成が含まれます。

### 2.1. STACK ユーザーの初期化

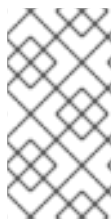
**stack** ユーザーとして **director** ホストにログインし、以下のコマンドを実行して **director** の設定を初期化します。

```
$ source ~/stackrc
```

このコマンドでは、**director** の CLI ツールにアクセスする認証情報が含まれる環境変数を設定します。

### 2.2. アンダークラウド上での **IPv6** アドレスの設定

アンダークラウドは、外部のネットワーク上にあるオーバークラウドのパブリック **API** にアクセスする必要があります。このアクセスを可能にするには、アンダークラウドホストで、外部ネットワークにアクセスするインターフェースに **IPv6** アドレスが必要となります。



#### 注記

プロビジョニングネットワークには、全ノードに対する **IPv4** 接続が引き続き必要です。アンダークラウドとオーバークラウドは、このネットワークを **PXE** ブート、イントロスペクション、デプロイメントに使用します。また、ノードはこのネットワークを使用して、**IPv4** で **DNS** および **NTP** サービスにアクセスします。

#### ネイティブ **VLAN** または専用のインターフェース

アンダークラウドが、外部ネットワークにアタッチされたネイティブ **VLAN** または専用のインターフェースを使用する場合には、**ip** コマンドでそのインターフェースに **IPv6** アドレスを追加します。以下の例では専用のインターフェースは **eth0** です。

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

#### トランキングされた **VLAN** インターフェース

アンダークラウドは、同じインターフェース上のトランキングされた **VLAN** を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**) として使用する場合には、新しい **VLAN** インターフェー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。たとえば、以下のシナリオでは外部ネットワークの **VLAN ID** に **100** を使用しています。

```
$ 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** コマンドで、**IPv6** アドレスが追加されたことを確認します。

```
$ ip addr
```

選択したインターフェースに IPv6 アドレスが表示されます。

### 永続的な IPv6 アドレスの設定

上記の手順に加えて、IPv6 アドレスを永続的にする必要がある場合があります。その場合には、`/etc/sysconfig/network-scripts/` で適切なインターフェース (本ガイドの例では `ifcfg-eth0` または `ifcfg-vlan100`) のファイルを編集または作成して、以下の行を追記してください。

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

詳しい説明は、Red Hat カスタマーポータルの「[IPv6 にネットワークインターフェイスを設定する](#)」のソリューションを参照してください。

## 2.3. 環境の設定

本項では、『Red Hat OpenStack Platform 11 director のインストールと使用方法』ガイドの「[CLI ツールでの基本的なオーバークラウド要件の設定](#)」に記載のプロセスを簡略にまとめたバージョンを使用します。

以下のワークフローに従って環境を設定します。

- ノード定義のテンプレートを作成して `director` で空のノードを登録します。
- 全ノードのハードウェアを検査します。
- 手動でロールにノードをタグ付けします。
- フレーバーを作成してロールにタグ付けします。

### 2.3.1. ノードの登録

ノード定義のテンプレート (`instackenv.json`) は JSON ファイル形式で、このテンプレートには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": "pxe_ipmitool",
      "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": "pxe_ipmitool",
    "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": "pxe_ipmitool",
    "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": "pxe_ipmitool",
    "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": "pxe_ipmitool",
    "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": "pxe_ipmitool",
        "pm_user": "admin",
        "pm_password": "p@55w0rd!",
        "pm_addr": "192.0.2.210"
    }
}

```



### 注記

プロジェクトネットワークは IPv4 アドレスを使用します。IPMI アドレスは IPv4 アドレスで、直接アタッチされているか、プロビジョニングネットワークでルーティング到達可能である必要があります。

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**) にファイルを保存してから、**director** にインポートします。これには、以下のコマンドを実行します。

```
$ openstack baremetal import --json ~/instackenv.json
```

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

カーネルと **ramdisk** イメージを全ノードに割り当てます。

```
$ openstack baremetal configure boot
```

**director** でのノードの登録、設定が完了しました。

### 2.3.2. ノードのハードウェアの検査

ノードの登録後には、各ノードのハードウェア属性を検査します。各ノードのハードウェア属性を検査するには、以下のコマンドを実行します。

```
$ openstack baremetal introspection bulk start
```



### 重要

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。ベアメタルの場合には、通常 15 分ほどかかります。

### 2.3.3. ノードの手動でのタグ付け

各ノードのハードウェアを登録、検査した後は、特定のプロファイルにノードをタグ付けします。これらのプロファイルタグによりノードとフレーバーが照合され、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。

ノード一覧を取得して **UUID** を特定します。

```
$ ironic node-list
```

特定のプロファイルにノードを手動でタグ付けする場合には、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、2 台のノードをタグ付けしてコントローラープロファイルとコンピュータープロファイルをそれぞれ使用するには、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-c7fdbc12129a 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:compute** と **profile:control** オプションを追加すると、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

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するには、手動でのタグ付けの代わりに、プロファイルの自動タグ付けを使用します。

## 2.4. 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について考察します。これには、特定のネットワークトラフィックを使用するようにサービスを分離し、IPv6 オプションでオーバークラウドを設定する手順が含まれます。

### 2.4.1. インターフェースの設定

まず最初に、オーバークラウドには、ネットワークインターフェースのテンプレートセットが必要です。このテンプレートは YAML 形式の標準の Heat テンプレートです。director にはテンプレートサンプルが含まれているので、すぐに使用を開始することができます。

- **/usr/share/openstack-tripleo-heat-templates/network/config/single-nic-vlans:** このディレクトリーには、ロールごとに VLAN が設定された単一 NIC のテンプレートが含まれます。
- **/usr/share/openstack-tripleo-heat-templates/network/config/bond-with-vlans:** このディレクトリーには、ロール別のボンディング NIC 設定のテンプレートが含まれます。

これらのテンプレートコレクションの1つを **stack** ユーザーの **templates** ディレクトリーにコピーします。以下に例を示します。

```
$ sudo cp -r /usr/share/openstack-tripleo-heat-
templates/network/config/single-nic-vlans ~/templates/nic-configs
```

ネットワークインターフェースの設定に関する詳細は、『[Red Hat OpenStack Platform 11 director のインストールと使用方法](#)』を参照してください。





### 注記

いずれのインターフェーステンプレートコレクションにも、コントローラーノードを設定するための2つのファイル(**controller.yaml**と**controller-v6.yaml**)が含まれます。IPv6の設定には**controller-v6.yaml**を使用します。

## 2.4.2. 分離された IPv6 ネットワークの設定

オーバークラウドには、IPv6を設定するためのネットワーク環境ファイルが必要です。このファイルは、オーバークラウドのネットワーク環境を記述し、ネットワークインターフェースの設定テンプレートを参照する Heat 環境ファイルです。本シナリオでは、環境ファイル(**/home/stack/network-environment.yaml**)を作成して、以下のセクションを追加します。

```
resource_registry:
  OS::Triple0::BlockStorage::Net::SoftwareConfig:
    /home/stack/templates/nic-configs/cinder-storage.yaml
  OS::Triple0::Compute::Net::SoftwareConfig: /home/stack/templates/nic-configs/compute.yaml
  OS::Triple0::Controller::Net::SoftwareConfig: /home/stack/templates/nic-configs/controller-v6.yaml
  OS::Triple0::ObjectStorage::Net::SoftwareConfig:
    /home/stack/templates/nic-configs/swift-storage.yaml
  OS::Triple0::CephStorage::Net::SoftwareConfig:
    /home/stack/templates/nic-configs/ceph-storage.yaml
```

このセクションにより、インターフェーステンプレートがそれぞれリソースとして登録されます。オーバークラウドの作成で、ノード種別ごとにネットワークを設定する際に、メインの Heat テンプレートコレクションが適切な **OS::Triple0::\*:Net::SoftwareConfig** リソースを呼び出します。このリソースがない場合には、メインの Heat テンプレートコレクションは、メインリソースレジストリーからデフォルトのネットワーク設定セットを使用します。



### 注記

いずれのインターフェーステンプレートコレクションにも、コントローラーノードを設定するための2つのファイル(**controller.yaml**と**controller-v6.yaml**)が含まれます。IPv6の設定には**controller-v6.yaml**を使用します。

```
parameter_defaults:
  DnsServers: ["8.8.8.8", "8.8.4.4"]
  ControlPlaneSubnetCidr: "24"
  EC2MetadataIp: 192.0.2.1
  ControlPlaneDefaultRoute: 192.0.2.1
  ExternalInterfaceDefaultRoute: 2001:db8::1

  ExternalNetworkVlanID: 100
  ExternalNetCidr: '2001:db8:0:1::/64'
  ExternalAllocationPools: [{'start': '2001:db8:0:1::10', 'end':
    '2001:db8:0:1:ffff:ffff:ffff:fffe'}]

  InternalApiNetworkVlanID: 201
  InternalApiNetCidr: 'fd00:fd00:fd00:2000::/64'
  InternalApiAllocationPools: [{'start': 'fd00:fd00:fd00:2000::10', 'end':
    'fd00:fd00:fd00:2000:ffff:ffff:ffff:fffe'}]
```

```

TenantNetworkVlanID: 202
TenantNetCidr: 172.17.0.0/24
TenantAllocationPools: [{ 'start': '172.17.0.10', 'end': '172.17.0.200' }]

StorageNetworkVlanID: 203
StorageNetCidr: 'fd00:fd00:fd00:4000::/64'
StorageAllocationPools: [{ 'start':
'fd00:fd00:fd00:4000:0000:0000:0000:0000', 'end':
'fd00:fd00:fd00:4000:ffff:ffff:ffff:ffff' }]

StorageMgmtNetworkVlanID: 204
StorageMgmtNetCidr: 'fd00:fd00:fd00:5000::/64'
StorageMgmtAllocationPools: [{ 'start':
'fd00:fd00:fd00:5000:0000:0000:0000:0000', 'end':
'fd00:fd00:fd00:5000:ffff:ffff:ffff:ffff' }]

```

**parameter\_defaults** セクションには、環境のカスタマイズが記述されます。IPv6 の設定では、以下のパラメーターを確認してください。

#### ExternalInterfaceDefaultRoute

外部インターフェースのデフォルトルート用の IP アドレス。本シナリオでは、アンダークラウドをデフォルトルートとして使用し、「[アンダークラウド上での IPv6 アドレスの設定](#)」で作成した IP アドレスを指定します。

#### ExternalNetworkVlanID

外部ネットワークの VLAN ID。アンダークラウドで外部ネットワークへのアクセスに VLAN を使用する場合には、「[アンダークラウド上での IPv6 アドレスの設定](#)」を参照)、この値に同じ VLAN をマッピングするようにしてください。

#### ExternalNetCidr、InternalApiNetCidr、TenantNetCid、StorageNetCidr、StorageMgmtNetCidr

各ネットワークの IPv6 CIDR プレフィックス

#### ExternalAllocationPools、InternalApiAllocationPools、TenantAllocationPools、StorageAllocationPools、StorageMgmtAllocationPools

ノードに割り当てる IPv6 アドレスの範囲。これは、IPv6 の競合が発生しないようにするのに役立ちます。

### 2.4.3. IPv6 と IPv4 のハイブリッド設定の使用

オーバークラウドでは、さまざまなサービスに、IPv4 と IPv6 ネットワークを組み合わせで使用することが可能です。これには、オーバークラウドのネットワークとポートを変更する必要があります。たとえば、ストレージおよびストレージ管理のネットワークを IPv4 にデプロイする一方で、他のネットワークに IPv6 を使用することができます。

ネットワーク分離の初期化ファイル (**network-isolation-v6.yaml**) をコピーします。

```

$ sudo cp /usr/share/openstack-tripleo-heat-
templates/environments/network-isolation-v6.yaml ~/templates/nic-
configs/network-isolation-v6.yaml

```

このファイルを編集して、ストレージおよびストレージ管理リソースが IPv4 バージョンのテンプレートを使用するようにマッピングします。以下に例を示します。

```

resource_registry:
  OS::TripleO::Network::External: ../network/external_v6.yaml
  OS::TripleO::Network::InternalApi: ../network/internal_api_v6.yaml

```

```

    OS::Triple0::Network::StorageMgmt: ../network/storage_mgmt.yaml #
Changed for IPv4
    OS::Triple0::Network::Storage: ../network/storage.yaml #
Changed for IPv4
    OS::Triple0::Network::Tenant: ../network/tenant_v6.yaml

# Port assignments for the VIPs
    OS::Triple0::Network::Ports::ExternalVipPort:
../network/ports/external_v6.yaml
    OS::Triple0::Network::Ports::InternalApiVipPort:
../network/ports/internal_api_v6.yaml
    OS::Triple0::Network::Ports::StorageVipPort:
../network/ports/storage.yaml # Changed for IPv4
    OS::Triple0::Network::Ports::StorageMgmtVipPort:
../network/ports/storage_mgmt.yaml # Changed for IPv4
    OS::Triple0::Network::Ports::RedisVipPort: ../network/ports/vip_v6.yaml

# Port assignments for the controller role
    OS::Triple0::Controller::Ports::ExternalPort:
../network/ports/external_v6.yaml
    OS::Triple0::Controller::Ports::InternalApiPort:
../network/ports/internal_api_v6.yaml
    OS::Triple0::Controller::Ports::StoragePort:
../network/ports/storage.yaml # Changed for IPv4
    OS::Triple0::Controller::Ports::StorageMgmtPort:
../network/ports/storage_mgmt.yaml # Changed for IPv4
    OS::Triple0::Controller::Ports::TenantPort:
../network/ports/tenant_v6.yaml

...

```

**openstack overcloud deploy** を実行する際には、元のファイルではなく、このカスタム **network-isolation-v6.yaml** を指定します。

さらに、**/home/stack/network-environment.yaml** の **parameter\_defaults** セクションを編集して、**IPv6** を割り当てるネットワークと **IPv4** を割り当てるネットワークが混在するように定義します。以下に例を示します。

```

parameter_defaults:
  DnsServers: ["8.8.8.8", "8.8.4.4"]
  ControlPlaneSubnetCidr: "24"
  EC2MetadataIp: 192.0.2.1
  ControlPlaneDefaultRoute: 192.0.2.1
  ExternalInterfaceDefaultRoute: 2001:db8::1

  ExternalNetworkVlanID: 100
  ExternalNetCidr: '2001:db8:0:1::/64'
  ExternalAllocationPools: [{'start': '2001:db8:0:1::10', 'end':
'2001:db8:0:1:ffff:ffff:ffff:fffe'}]

  InternalApiNetworkVlanID: 201
  InternalApiNetCidr: 'fd00:fd00:fd00:2000::/64'
  InternalApiAllocationPools: [{'start': 'fd00:fd00:fd00:2000::10', 'end':
'fd00:fd00:fd00:2000:ffff:ffff:ffff:fffe'}]

  TenantNetworkVlanID: 202

```

```
TenantNetCidr: 172.17.0.0/24
TenantAllocationPools: [{'start': '172.17.0.10', 'end': '172.17.0.200'}]

StorageNetworkVlanID: 203
StorageNetCidr: 172.18.0.0/24
StorageAllocationPools: [{'start': '172.18.0.10', 'end':
'172.18.0.200'}]

StorageMgmtNetworkVlanID: 204
StorageMgmtNetCidr: 172.19.0.0/24
StorageMgmtAllocationPools: [{'start': '172.19.0.10', 'end':
'172.19.0.200'}]
```



### 重要

オーバークラウドで **VXLAN** ネットワークを使用する場合には、テナントネットワークに **IPv4** を使用するように設定します。テナントネットワークでは、**IPv6 VXLAN** はサポートされていません。

## 2.5. オーバークラウドの設定の完了

これで、IPv6 ベースのオーバークラウドの設定に必要なステップは完了です。次章では、**openstack overcloud deploy** コマンドで、本章で作成した設定を使用してオーバークラウドを作成します。

## 第3章 オーバークラウドの作成

IPv6 ネットワークを使用するオーバークラウドを作成するには、**openstack overcloud deploy** コマンドに追加の引数を指定する必要があります。以下に例を示します。

```
$ openstack overcloud deploy --templates -e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation-v6.yaml -e /home/stack/templates/network-environment.yaml --control-scale 3 --compute-scale 3 --control-flavor control --compute-flavor compute --neutron-disable-tunneling --neutron-network-type vlan --neutron-tunnel-types vlan --neutron-network-vlan-ranges datacenter:1:1000 --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-v6.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。この場合は、IPv6 用にネットワーク分離の設定を初期化する環境ファイルです。
- **-e /home/stack/templates/network-environment.yaml:** オーバークラウドデプロイメントに追加の環境ファイルを追加します。この場合は、以前に作成したネットワーク環境ファイルです。
- **--control-scale 3:** コントローラーノードを 3 台にスケーリングします。
- **--compute-scale 3:** コンピュートノードを 3 台にスケーリングします。
- **--control-flavor control:** 対象のコントローラーノードに特定のフレーバーを使用します。
- **--compute-flavor compute:** コンピュートノードに特定のフレーバーを使用します。
- **--neutron-disable-tunneling:** オーバークラウド内のトンネリングを無効にします。IPv6 ではトンネリングはサポートされていません。テナントネットワークの設定に IPv4 を使用して、トンネリングを有効にする場合には、このオプションは追加しないでください。
- **--neutron-network-type vlan:** neutron のネットワーク種別を設定します。テナントネットワークに IPv6 を使用する場合には VLAN モードを選択します。VXLAN を使用する場合には、テナントネットワークが IPv4 を使用するように変更します。詳しい情報は、「[IPv6 と IPv4 のハイブリッド設定の使用](#)」を参照してください。
- **--neutron-network-vlan-ranges datacenter:1:1000:** neutron がサポートするマッピングの範囲を設定します。
- **--ntp-server pool.ntp.org:** NTP サーバーを設定します。



## 注記

オプションの完全な一覧を表示するには、以下のコマンドを実行します。

```
$ openstack help overcloud deploy
```

パラメーターの例については、『[Red Hat OpenStack Platform 11 director インストールと使用方法](#)』ガイドを参照してください。

オーバークラウドの作成プロセスが開始され、**director** により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、以下を実行します。

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

### 3.1. オーバークラウドへのアクセス

**director** は、**director** ホストがオーバークラウドと対話するための設定と認証を行うスクリプトを生成します。このファイル (**overcloudrc**) は、**stack** ユーザーのホームディレクトリーに保存されます。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4章 オーバークラウドの作成後の設定

上記のプロセスで、IPv6 ネットワークを使用するように設定された、完全に機能するオーバークラウドが作成されましたが、そのオーバークラウドには、作成後の設定をいくつか行う必要があります。

### 4.1. オーバークラウドのテナントネットワークの作成

オーバークラウドには、インスタンス用に IPv6 ベースのテナン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** に初期テナントネットワークを作成します。以下に例を示します。

```
$ source ~/overcloudrc
$ neutron net-create default --provider:physical_network datacentre --
  provider:network_type vlan --provider:segmentation_id 101
$ neutron subnet-create default 2001:db8:fd00:6000::/64 --ipv6-ra-mode
  slaac --ipv6-address-mode slaac --ip-version 6 --name default
```

これで、**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。「**neutron net-list**」コマンドで、作成したネットワークを確認します。

```
$ neutron net-list
```

### 4.2. オーバークラウドのパブリックネットワークの作成

本シナリオでは、ノードのインターフェースが外部ネットワークを使用するように設定しましたが、オーバークラウド上にこのネットワークを作成して、ネットワークアクセスを提供できるようにする必要があります。

```
$ neutron net-create public --router:external --provider:physical_network
  datacentre --provider:network_type vlan --provider:segmentation_id 100
$ neutron subnet-create public 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
  --ip-version 6 --ipv6_address_mode=slaac --ipv6_ra_mode=slaac
```

このステップにより、**public** という名前のネットワークが作成され、インスタンス用に **65000** を超える IPv6 アドレスの割り当てプールが提供されます。

ルーターを作成して、インスタンスのトラフィックを外部ネットワークにルーティングします。

```
neutron router-create public-router
neutron router-gateway-set public-router public
```

## 第5章 結果

これで、IPv6 ベースのオーバークラウドの作成および設定が完了しました。オーバークラウド作成後の一般的な機能については、『[Red Hat OpenStack Platform 11 director のインストールと使用方法](#)』を参照してください。