

VMware NSX-T Data Center Plugin for OpenStack Neutron インストールガイド

2019 年 9 月 19 日

VMware NSX-T Data Center 2.5



vmware®

最新の技術ドキュメントは、VMware の Web サイト (<https://docs.vmware.com/jp/>) でご確認ください。このドキュメントに関するご意見および感想は、docfeedback@vmware.com までお送りください。

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

ヴァイエムウェア株式会社
105-0013 東京都港区浜松町 1-30-5
浜松町スクエア 13F
www.vmware.com/jp

Copyright © 2019 VMware, Inc. All rights reserved. [著作権および商標情報](#)。

目次

1	VMware NSX-T Data Center Plugin for OpenStack Neutron インストール ガイド	4
2	NSX-T Data Center Plugin for OpenStack のインストール準備	5
	前提条件	5
	システム要件	5
	Neutron プラグインの比較	6
3	NSX-T Data Center Plug-In での Neutron 基本サービスのインストール	9
	Ubuntu システムへの NSX-T Data Center Plugin for OpenStack のインストール	9
	Red Hat システムへの NSX-T Data Center Plug-In for OpenStack のインストール	10
4	NSX-T Data Center プラグインでの OpenStack の設定	11
	OpenStack Neutron ネットワーク ノードの設定	11
	neutron.conf と nsx.ini の編集	12
	クライアント証明書認証の有効化	13
	DHCP とメタデータ プロキシ サービスの有効化	13
	NSX-T Data Center Plugin for OpenStack の設定ファイルの例	15
	OpenStack Nova コントローラの設定	16
	OpenStack Nova コンピューティングの設定ファイル	16
5	NSX-T Data Center Plugin for OpenStack での Neutron 拡張サービスの設定	18
	OpenStack Octavia の Load Balancer as a Service (LBaaS) の設定	19
	OpenStack Neutron の Load Balancer as a Service (LBaaS) の設定	19
	OpenStack Neutron の Firewall as a Service (FWaaS) の設定	20
	OpenStack Neutron の IPSec VPN as a Service (VPNaaS) の設定	21
	Neutron 拡張サービスの設定ファイルのサンプル	22
6	付録 : NSX-T Data Center Plugin for OpenStack の設定プロパティ	25

VMware NSX-T Data Center Plugin for OpenStack Neutron インストール ガイド

1

このガイドでは、NSX-T Data Center Plugin for OpenStack Neutron のインストール方法と設定方法について説明します。詳細な設定手順についても記載しています。

VMware NSX-T Data Center Plugin を設定して実行すると、OpenStack Neutron が NSX-T Data Center 環境内の仮想ネットワーク リソースを認識し、管理できるようになります。このガイドに従って操作を行う前に、NSX-T Data Center と OpenStack のコンポーネントと機能について理解しておく必要があります。

このガイドでは、NSX-T Policy と NSX-T Manager の Openstack Neutron プラグインについて説明します。これは、NSX-T Policy 用の Openstack プラグインを提供する最初の NSX-T Data Center リリースです。このリリースでは、新規インストールでのみ NSX-T Policy プラグインを使用できます。

これらのトピックについては、次を参照してください。

- NSX-T Data Center 管理ガイド
- OpenStack ドキュメント

NSX-T Data Center Plugin for OpenStack のインストール準備

2

この章には、次のトピックが含まれています。

- 前提条件
- システム要件
- Neutron プラグインの比較

前提条件

VMware から提供される OpenStack Neutron プラグイン用のサポート パッケージには、NSX-T 固有のアーティファクトのみが含まれています。そのため、このインストール プロセスを行う前に、選択した OpenStack サービスをインストールする必要があります。

このドキュメントの手順に従って、NSX-T Data Center Plugin for OpenStack Neutron をインストールし、NSX 環境との統合を設定します。以下の手順では、NSX-T トランスポート ノードに VMware NSX-T Data Center がインストールされ、設定されていることを前提としています。

適切な依存関係をダウンロードしてインストールし、インストール プロセスで設定するには、Neutron サービスのインストール時にインターネットに接続するか、ローカルのディストリビューション リポジトリ ミラーへのアクセスが必要になります。

システム要件

NSX-T Data Center Plugin for OpenStack サポートは、Neutron プラグインとして実装されています。Neutron の設定時に使用される VMware NSX プラグイン クラスは、使用している NSX のバージョンによって異なります。

vSphere および KVM (Ubuntu、Red Hat Enterprise Linux、CentOS など) でサポートされるハイパーバイザーのバージョンは、NSX-T Data Center インストール ガイド に記載されています。

NSX-T Data Center Plugin for OpenStack には、互換性のある OpenStack ソフトウェア バージョンに関する次の要件があります。

NSX-T Policy プラグインの OpenStack ディストリビューション	バージョン
オープン ソース エディション	Stein

NSX-T Manager プラグインの OpenStack ディストリビューション	バージョン
オープンソースエディション	Rocky
オープンソースエディション	Stein
Red Hat OpenStack プラットフォーム	Red Hat OpenStack バージョン 13 (Red Hat Enterprise Linux 関連バージョンを含む)

Neutron プラグインの比較

VMware NSX-T Datacenter 2.5 から、Openstack Neutron と NSX-T を統合するために 2 つのプラグインが使用できるようになりました。

- NSX-T Policy プラグインは、インテント ベースの API 抽象化を使用して、NSX-T ポリシー マネージャと通信します。これは新しいプラグインで、新規インストールの場合に推奨されます。
- NSX-T Manager プラグインは、強制 API を使用して、NSX-T Manager と通信します。これは既存の NSX-T プラグインであり、既存のインストール環境で使用する必要があります。また、NSX-T Policy プラグインが対応していないユースケースで使用できます。

表 2-1. プラグインの機能の比較

ネットワークとセキュリティ機能	NSX-T MP プラグイン	NSX-T Policy	説明
スイッチング			
重複する IP サブネットのサポート	○	○	各プロジェクトで、プロジェクトのプライベート ネットワークを動的に作成できます。これらのネットワークは、互いに重複する IP サブネットを持つことができます。
DHCP	○	○	DHCP により、インスタンスのアドレスが自動的に割り当てられます。
固定 IPv6 アドレスの割り当て	×	○	
ルーティング			
論理ルーティング	○	○	複数のプライベート論理ネットワーク間のルーティングと、論理ネットワークと外部ネットワーク間のルーティングを有効にします。
IPv6 論理ルーティング	×	○	複数のプライベート IPv6 論理ネットワーク間のルーティングと、論理ネットワークと外部ネットワーク間のルーティングを有効にします。

表 2-1. プラグインの機能の比較 (続き)

ネットワークとセキュリティ機能	NSX-T MP プラグイン	NSX-T Policy	説明
外部ネットワーク	○	○	インスタンスへの外部アクセスを提供するネットワーク。プライベート ネットワーク上のインスタンスへの外部からのアクセスを提供するため、プライベート ネットワークがルーター経由で外部ネットワークにアップリンクされます。
IPv6 外部ネットワーク	○	○	IPv6 を使用する外部ネットワーク。
スタティック ルート	○	○	スタティック ルートを挿入します。
IPv6 スタティック ルート	×	○	IPv6 を使用する外部ネットワーク。
インスタンスのフローティング IP	○	○	インスタンスへの外部アクセスを有効にするため、ルーティング可能なパブリック IP アドレスをインスタンスに割り当てます。
No-NAT ルーター	○	○	No-NAT ルーティング トポロジ。
IPv6 No-NAT ルーター	×	○	No-NAT トポロジは、IPv6 の OpenStack でサポートされている唯一のルーティング トポロジです。IPv6 の NAT はサポートされていません。
Neutron ルーター デュアル スタック インターフェイス	×	○	Neutron ルーターの同一インターフェイスで IPv4 と IPv6 のデュアル スタックをサポートします。
IPv6 SLAAC	×	○	ステートレス アドレスの自動設定をサポートします。
セキュリティ			
ファイアウォール - セキュリティ グループ	○	○	OpenStack セキュリティ グループ (NSX でセキュリティ グループを使用。これらの SG を使用して作成された DFW ルールも使用。これにより、マイクロセグメンテーションが可能になります)。
IPv6 ファイアウォール(セキュリティ グループ)	×	○	IPv6 の Neutron セキュリティ グループ。
ポートのセキュリティ	○	○	Neutron ポートのセキュリティは、NSX SpoofGuard 機能を使用して実装されます。
IPv6 ポートのセキュリティ	×	○	Neutron ポートのセキュリティは、NSX SpoofGuard 機能を使用して実装されます。これにより、allowed_address_pairs と IPv6 サブネットとポートのマッピングが可能になります。

表 2-1. プラグインの機能の比較 (続き)

ネットワークとセキュリティ機能	NSX-T MP プラグイン	NSX-T Policy	説明
ファイアウォール (L3 FWaaS)	○	○	
IPv6 ファイアウォール (L3 FWaaS)	×	○	
その他のサービス			
ロード バランシング	○	○	
サービス品質	○	○	
DNS	○	○	
VPNaaS	○	×	

アップグレード

NSX-T Manager プラグインを使用する Openstack Neutron から、NSX-T Policy プラグインを使用する Openstack Neutron に移行することはできません。アップグレードを行う場合は、既存のインストールで NSX-T Manager プラグインを引き続き実行する必要があります。NSX Manager から NSX-T Policy への移行は、今後のリリースで対応する予定です。新規インストールの場合は、NSX-T Policy プラグインをおすすめします。このソリューションには独自の機能 (IPv6) が含まれています。新しい機能への移行は、NSX-T プラグインのみを対象としています。

NSX-T Data Center Plug-In での Neutron 基本サービスのインストール

3

この章には、次のトピックが含まれています。

- [Ubuntu システムへの NSX-T Data Center Plugin for OpenStack のインストール](#)
- [Red Hat システムへの NSX-T Data Center Plug-In for OpenStack のインストール](#)

Ubuntu システムへの NSX-T Data Center Plugin for OpenStack の インストール

NSX-T Data Center Plugin は、Ubuntu ベースの Linux ディストリビューションの debian (deb) パッケージとして配布されます。

前提条件

NSX-T Data Center Plugin は、Ubuntu ベースの Linux ディストリビューションの debian (deb) パッケージとして配布されます。

次の手順は、NSX Manager と NSX-T Policy の両方のプラグインに適用されます。

- 依存関係のエラーで debian パッケージのインストールが失敗する場合は、python-tooz パッケージと python-oslo.vmware パッケージのインストールが必要になる場合があります。tooz は、分散調整プリミティブの抽象化を提供する Python ライブラリです。その主な目的は、分散システムでグループとそのメンバーシップを処理することです。Oslo VMware ライブラリは、一般的な VMware の運用と API をサポートします。例：
`sudo apt-get install python-oslo.vmware`

手順

- 1 .deb ファイル（NSX Neutron プラグインと NSX Neutron の共通ライブラリ）をダウンロードします。
- 2 Neutron ネットワーク ノードにファイルをコピーします。
- 3 .deb ファイルと同じディレクトリで dpkg コマンドを実行して、パッケージをインストールします。

ダウンロード時に選択したリリースによっては、次の例にあるバージョン番号が異なる場合があります。

- `sudo dpkg -i python-vmware-nsxlib_12.0.0.9797177-1_all.deb`
- `sudo dpkg -i openstack-vmware-nsx_12.0.0.9797177-1_all.deb`

- 4 Firewall as a Service (FWaaS) パッケージをインストールします。これは、インストール後に有効にする必要はありません。

ダウンロード時に選択したリリースによっては、次の例にあるバージョン番号が異なる場合があります。

- `sudo apt-get install python-neutron-fwaas`

Red Hat システムへの NSX-T Data Center Plug-In for OpenStack のインストール

Red Hat ベースの Linux ディストリビューションでは、NSX-T Data Center Plug-Ins for OpenStack は .rpm ファイルとしてパッケージ化されています。

.rpm パッケージは、NSX-T Data Center のダウンロード ページの [ドライバとツール] からダウンロードできます。ここでは、TripleO を利用しない場合のインストール手順を説明します。それ以外の場合は、Red Hat OpenStack 専用のガイドを参照してください。

手順

- 1 .rpm ファイル（NSX Neutron プラグインと NSX Neutron の共通ライブラリ）をダウンロードします。
- 2 プラグインをインストールする Neutron ネットワーク ノードにコピーします。
- 3 .rpm ファイルと同じディレクトリで rpm コマンドを実行して、パッケージをインストールします。

ダウンロード時に選択したリリースによっては、次の例にあるバージョン番号が異なる場合があります。

- `sudo rpm -i python-vmware-nsxlib_12.0.0.9797177-1_all.rpm`
- `sudo rpm -i vmware-nsx-12.0.0.9797177-1.noarch.rpm`

- 4 Firewall as a Service (FWaaS) パッケージをインストールします。これは、インストール後に有効にする必要はありません。

- `sudo yum install python-neutron-fwaas`

NSX-T Data Center プラグインでの OpenStack の設定

- [OpenStack Neutron ネットワーク ノードの設定](#)

このセクションでは、Neutron ネットワーク ノードの設定について説明します。

- [NSX-T Data Center Plugin for OpenStack の設定ファイルの例](#)

設定ファイルは通常、`/etc/neutron/plugins/vmware/nsx.ini` にあります。

- [OpenStack Nova コントローラの設定](#)

このセクションで説明する情報は、Nova コントローラ ノードの設定を補足するもので、NSX-T Data Center に関する情報を追加しています。

- [OpenStack Nova コンピューティングの設定ファイル](#)

Nova 設定ファイルを編集するには NSX-T を使用します。

OpenStack Neutron ネットワーク ノードの設定

このセクションでは、Neutron ネットワーク ノードの設定について説明します。

このドキュメントでは、NSX-T Policy プラグインについて説明していますが、このセクションでは NSX-T Manager プラグインに固有の設定についても説明します。

デフォルトでは、この 2 つ設定ファイルは次のパスにあります。

- `/etc/neutron/neutron.conf` - Neutron の設定ファイル。

- `/etc/neutron/plugin/vmware/nsx.ini` - VMware NSX Neutron プラグインの設定ファイル

- [neutron.conf と nsx.ini の編集](#)

Neutron プラグインが NSX-T 環境と通信できるようにするため、これらのファイルを編集して、NSX-T 環境に関連する設定を行う必要があります。

- [クライアント証明書認証の有効化](#)

NSX Manager への Neutron クライアント証明書認証がサポートされています。

- [DHCP とメタデータ プロキシ サービスの有効化](#)

NSX-T Data Center プラグインを使用すると、OpenStack DHCP のリファレンス実装が NSX-T Data Center ネイティブ DHCP サーバに置き換わります。また、NSX-T Data Center プラットフォームが Nova メタデータにアクセスするためのプロキシ サーバを提供します。

neutron.conf と nsx.ini の編集

Neutron プラグインが NSX-T 環境と通信できるようにするため、これらのファイルを編集して、NSX-T 環境に関連する設定を行う必要があります。

手順

- 1 `neutron.conf` ファイルに **[DEFAULT] core_plugin = vmware_nsxp** を記述して、Neutron コア プラグインを設定します。NSX-T Manager プラグインを有効にする場合：
[DEFAULT] core_plugin = vmware_nsxv3

- 2 `nsx.ini` 設定ファイルを編集して、NSX 展開用のプラグインを設定します。

NSX-T OpenStack プラグインのプロパティは、`nsx.ini` 設定ファイルの `[nsx_p]` セクションに指定します。

次の設定プロパティも NSX-T Manager プラグインに適用されます。これらのプロパティは `[nsx_v3]` 設定セクションに指定します。

少なくとも次の設定プロパティを定義する必要があります。

変数	説明
<code>nsx_api_managers</code>	このパラメータには、カンマ区切りされたマネージャ エンドポイントのリストを指定できます。
<code>nsx_api_user</code>	管理者の NSX-T Manager のユーザー名（通常は admin）。
<code>nsx_api_password</code>	管理者の NSX-T Manager のパスワード。
<code>insecure</code>	NSX Manager サーバ証明書の検証を行うには、False に設定します。デフォルトは True です。
<code>ca_file</code>	NSX Manager サーバ証明書の検証で使用する CA バンドル ファイル。 insecure が True に設定されている場合、このオプションは無視されます。insecure が False に設定され、このオプションが未設定の場合、システムのルート CA がサーバ証明書の検証に使用されます。
<code>nsx_api_managers</code>	Neutron ネットワークの作成に使用されるデフォルトの NSX オーバーレイ トランスポート ゾーンの名前または UUID。Neutron を開始する前に NSX で作成しておく必要があります。
<code>default_tier0_router</code>	これは、OpenStack 論理ルーター（NSX-T Tier-1）に接続する Policy Manager NSX-T Tier-0 ネーム ゲートウェイ ルーターまたは UUID にする必要があります（「ルーティング/ルーター」にあります）。
<code>dhcp_profile</code>	UUID または名前のいずれかを入力します。 NSX Manager での DHCP プロファイルの作成 を参照してください。
<code>metadata_proxy</code>	UUID または名前のいずれかを入力します。 メタデータ プロキシの作成 を参照してください。

- 3 Neutron を再起動して `ps-aux |grep neutron` コマンドを実行し、`nsx.ini` ファイルの変更を取得します。

- 4 出力に `nsx.ini` と `neutron.conf` が含まれていることを確認します。Neutron は、コマンドラインで 1 つ以上の設定ファイルを受け入れます。設定ファイルの構造が特定のユーザーの環境設定を反映するように、これらの設定ファイルは解析時にマージされます。

```
ps -aux |grep neutron
stack      7688  0.0  1.8 311332 148904 ?        Ss   Nov26  21:10
/usr/bin/python /usr/local/bin/neutron-server --config-file
/etc/neutron/neutron.conf --config-file
/etc/neutron/plugins/vmware/nsx.ini
```

クライアント証明書認証の有効化

NSX Manager への Neutron クライアント証明書認証がサポートされています。

クライアント証明書認証により、Neutron プラグインは、エンタープライズ管理者ロールを持つプリンシパル ID としてログインできるようになります。他のプリンシパル ID は、neutron プリンシパル ID によって作成されたリソースを編集できないため、neutron ルーターに関連付けられている論理ルーターの削除などのエラーを防ぐことができます。詳細については、『NSX-T Data Center 管理ガイド』の「プリンシパル ID の表示」を参照してください。

手順

- 1 クライアント証明書認証を有効にするには、`nsx.ini` ファイルで次のように定義します。
 - `nsx_use_client_auth = True`
 - `nsx_client_cert_storage = nsx-db`
 - `nsx_client_cert_file = <file to store certificate and private key>`
- 2 Neutron を再起動して `service neutron-server restart` コマンドを実行し、`nsx.ini` ファイルの変更を取得します。

次のコマンドを実行して、Neutron サーバが `neutron.conf` と `nsx.ini` の両方を使用していることを確認します。

```
■ ps -aux |grep neutron
```

出力に `nsx.ini` と `neutron.conf` が含まれていることを確認します。次はその例です。

```
ps -aux |grep neutron
stack      7688  0.0  1.8 311332 148904 ?        Ss   Nov26  21:10
/usr/bin/python /usr/local/bin/neutron-server --config-file
/etc/neutron/neutron.conf --config-file
/etc/neutron/plugins/vmware/nsx.ini
```

DHCP とメタデータ プロキシ サービスの有効化

NSX-T Data Center プラグインを使用すると、OpenStack DHCP のリファレンス実装が NSX-T Data Center ネイティブ DHCP サーバに置き換わります。また、NSX-T Data Center プラットフォームが Nova メタデータにアクセスするためのプロキシ サーバを提供します。

前の手順で設定した NSX-T プラグインに関係なく、これらの操作を行う必要があります。

■ NSX Manager での DHCP プロファイルの作成

DHCP サーバ プロファイルは、NSX Edge クラスタまたは NSX Edge クラスタのメンバーを指定します。このプロファイルを持つ DHCP サーバは、プロファイルで指定された NSX Edge ノードに接続されている論理スイッチ上の仮想マシンからの DHCP 要求を処理します。

■ メタデータ プロキシの作成

メタデータ プロキシ サーバでは、仮想マシン インスタンスは OpenStack Nova API サーバからインスタンス固有のメタデータを取得することができます。

■ nsx.ini ファイルの編集

これらのプロファイルを使用するには、新しい変数 `native_dhcp_metadata`、`metadata_proxy`、`dhcp_profile` を `nsx.ini` に指定する必要があります。

NSX Manager での DHCP プロファイルの作成

DHCP サーバ プロファイルは、NSX Edge クラスタまたは NSX Edge クラスタのメンバーを指定します。このプロファイルを持つ DHCP サーバは、プロファイルで指定された NSX Edge ノードに接続されている論理スイッチ上の仮想マシンからの DHCP 要求を処理します。

前提条件

NSX-T Data Center のネイティブ DHCP サーバを有効にするには、NSX-T Data Center で DHCP プロファイルを作成し、`nsx.ini` の Neutron プラグイン設定に指定する必要があります。neutron dhcp サービス (devstack の q-dhcp) とメタデータ エージェント (devstack の q-meta) が実行されていないことを確認します。`neutron.conf` ファイルで、`enable_dhcp` を `False` に設定します。

手順

- 1 ブラウザから NSX Manager (<https://nsx-manager-ip-address>) に管理者権限でログインします。
- 2 ナビゲーション パネルから [ネットワークとセキュリティの詳細設定] > [DHCP] の順に選択します。
- 3 [サーバ プロファイル] を選択して、[追加] をクリックします。
- 4 名前を入力します。オプションで説明を入力できます。
- 5 ドロップダウン メニューから [Edge クラスタ] を選択します。
- 6 [追加] をクリックします。

メタデータ プロキシの作成

メタデータ プロキシ サーバでは、仮想マシン インスタンスは OpenStack Nova API サーバからインスタンス固有のメタデータを取得することができます。

NSX プラットフォームは、Nova メタデータにアクセスするためのプロキシ サーバを提供します。プロキシは、アドレス 169.254.269.254 で行われたすべての要求をキャプチャし、NSX-T メタデータ プロキシの設定で指定されている nova メタデータ サーバエンドポイントに転送します。

前提条件

メタデータ プロキシに使用される Edge ノードは、管理 IP アドレスからメタデータ サーバに接続できるようにする必要があります。

手順

- 1 ブラウザから NSX Manager (<https://nsx-manager-ip-address>) に管理者権限でログインします。
- 2 ナビゲーション パネルから [ネットワークの詳細設定] > [DHCP] の順に選択します。
- 3 [メタデータ プロキシ] を選択して、[追加] をクリックします。
- 4 [名前] に名前を入力します。オプションで説明を入力できます。
- 5 [Nova サーバの URL] に `http://<openstack_controller>:8775` を入力します。メタデータ プロキシ サーバがデフォルトの 8775 以外のポートで待機している場合は、正しいポートを使用して URL を更新します。ポートは、Nova API 設定ファイル (`/etc/nova.conf`) のコントローラ ノードにあります。
`metadata_listen_port` パラメータを検索してください。設定の変更が必要な場合は、n-api または nova サーバを再起動します。
- 6 [シークレット キー] パラメータを入力します。
- 7 ドロップダウン メニューから [Edge クラスタ] を選択します。
- 8 [追加] をクリックします。

nsx.ini ファイルの編集

これらのプロファイルを使用するには、新しい変数 `native_dhcp_metadata`、`metadata_proxy`、`dhcp_profile` を `nsx.ini` に指定する必要があります。

- `dhcp_profile = <UUID or name - DHCP>`
- `native_dhcp_metadata = True`
- `metadata_proxy = <UUID or name - MetaData Proxy>`
- `native_metadata_route = 169.254.169.254/31`

NSX-T Data Center Plugin for OpenStack の設定ファイルの例

設定ファイルは通常、`/etc/neutron/plugins/vmware/nsx.ini` にあります。

次に、設定ファイルの例を示します。

```
[nsx_p]
# NSX-T credentials
nsx_api_managers = 192.168.10.5
nsx_api_user = admin
nsx_api_password = VMware1!
insecure = True
# NSX-T objects information
default_tier0_router = 0fd8b97f-315d-4461-a80b-adb489b6cfbc
default_overlay_tz_ = 4d3fcd4f-0946-4b08-ab6b-5463c571463d
```

```
default_vlan_tz = f74b5dab-dad3-47d2-b46e-57a1eeb5fde3
# DHCP and Metadata Proxy offered by NSX-T
dhcp_profile = 153637ce-657a-4ff9-a2f2-ffab62441abc
metadata_proxy = 32cf4708-7b1f-4932-b4ca-9f7029c9a7a2
```

```
[nsx_v3]
# NSX-T credentials
nsx_api_managers = 192.168.10.5
nsx_api_user = admin
nsx_api_password = VMware1!
insecure = True
# NSX-T objects information
default_tier0_router_uuid = 0fd8b97f-315d-4461-a80b-adb489b6cfbc
default_overlay_tz_uuid = 4d3fcd4f-0946-4b08-ab6b-5463c571463d
# DHCP and Metadata Proxy offered by NSX-T
dhcp_profile = 153637ce-657a-4ff9-a2f2-ffab62441abc
metadata_proxy = 32cf4708-7b1f-4932-b4ca-9f7029c9a7a2
```

OpenStack Nova コントローラの設定

このセクションで説明する情報は、Nova コントローラ ノードの設定を補足するもので、NSX-T Data Center に関する情報を追加しています。

以下に、Nova コントローラの設定ファイルの例を示します。このファイルは通常、コントロール ノードの `/etc/nova/nova.conf` にあります。

```
[DEFAULT]
firewall_driver = nova.virt.firewall.NoopFirewallDriver
use_neutron = True

[neutron]
metadata_proxy_shared_secret = VMware1!
service_metadata_proxy = True
```

OpenStack Nova コンピューティングの設定ファイル

Nova 設定ファイルを編集するには NSX-T を使用します。

このセクションで説明する情報は、Nova コンピューティング ノードの設定を補足するもので、NSX-T Data Center に関する情報を追加しています。

以下に、Nova コンピューティングの設定ファイルの例を示します。このファイルは通常、/etc/nova/nova.conf にあります。

```
[DEFAULT]
firewall_driver = nova.virt.firewall.NoopFirewallDriver
use_neutron = True

[neutron]
#for KVM
ovs_bridge = nsx-managed
```

NSX-T Data Center Plugin for OpenStack での Neutron 拡張サービスの設定

5

OpenStack Neutron の Load Balancer as a Service (LBaaS)、Firewall as a service (FWaaS)、IPSec VPN as a Service (VPNaaS) は、Neutron 拡張サービスともいいます。

NSX-T Data Center 2.5 NSX-T プラグイン (Policy と Manager の両方) から、Octavia ロード バランシング サービスがサポートされます。これは、非推奨の Neutron LBaaS サービスに代わるものです。

以下の説明は、NSX-T Data Center 2.4 と OpenStack Stein 14.0 を使用していることを前提としています。

現在サポートされているバージョンについては、[システム要件](#)を参照してください。

Neutron 拡張サービス NSX-T Data Center を有効にするには、Neutron の設定で次のように指定する必要があります。

- 特定のサービスでサービス プラグインを有効にする
- サービスのサービス プロバイダを設定する
- 必要であれば、NSX-T Data Center ドライバと NSX-T Data Center 固有の設定エントリを指定します。

これらのオプションは、neutron 設定ファイルで指定されます。このファイルは通常、`/etc/neutron` にあります (注: プラグイン固有の設定ファイルは `/etc/neutron/plugins/vmware` にあります)。Neutron は、コマンドラインで 1 つ以上の設定ファイルを受け入れます。設定ファイルの構造が特定のユーザーの環境設定を反映するように、これらの設定ファイルは解析時にマージされます。通常、次のような構造になっています。

- `neutron.conf` - neutron のコア オプション、基本的な設定パラメータ (API マネージャのエンドポイント、トランスポート ゾーン ID など)、サービス プラグイン リスト。
- `neutron_lbass.conf` - ロード バランシング サービス プロバイダとオプション。
- `Octavia.conf` - ロード バランシング プロバイダと MQ トピック。Octavia ロード バランサ サービスの場合にのみ。このファイルは通常、`/etc/Octavia/Octavia.conf` にあり、Octavia サービスによって読み込まれます。
- `neutron_fwass.conf` - ファイアウォール サービス プロバイダ、ドライバ、ドライバ オプション。
- `neutron_vpnaas.conf` - VPN サービス プロバイダ、その他のオプション。

サービス プロバイダは、`service_provider` オプションで指定します。neutron の設定で、このオプションを繰り返し使用して、異なる種類のサービスを指定できますが、1 つのサービス タイプに複数のデフォルト サービス プロバイダを設定することはできません。

Service_provider 設定属性の構造 : <SERVICE_TYPE>:<PROVIDER_CLASS>:[<DEFAULT>]

- [OpenStack Octavia の Load Balancer as a Service \(LBaaS\) の設定](#)

ここでは、Octavia を使用して NSX-T Data Center ロード バランサで OpenStack を使用するように設定する手順について説明します。

- [OpenStack Neutron の Load Balancer as a Service \(LBaaS\) の設定](#)

- [OpenStack Neutron の Firewall as a Service \(FWaaS\) の設定](#)

ここでは、FWaaS v2 を設定する場合の手順について説明します。

- [OpenStack Neutron の IPSec VPN as a Service \(VPNaaS\) の設定](#)

このサービスを有効にすると、OpenStack ユーザーは Neutron ネットワークから安全な VPN トンネル経由でリモート サイトにアクセスできるようになります。

- [Neutron 拡張サービスの設定ファイルのサンプル](#)

OpenStack Octavia の Load Balancer as a Service (LBaaS) の設定

ここでは、Octavia を使用して NSX-T Data Center ロード バランサで OpenStack を使用するように設定する手順について説明します。

同じ手順で NSX-T Policy と NSX-T Manager の両方のプラグインを設定できます。

Octavia ロード バランサ サービスの概要については、Octavia の公式ドキュメントを参照してください。

手順

- 1 /etc/neutron/neutron.conf で、NSX-T Data Center のロード バランシング サービス プラグインが設定されていないことを確認します。service_plugins リストに vmware_nsx_lbaasv2 がいないことを確認します。neutron.conf ファイルへの変更が完了したら、neutron サービスを再起動します。
- 2 /etc/octavia/octavia.conf で、次の項目を指定します。
 - a [Api_settings] 設定に default_provider_driver = vmwareedge
enabled_provider_drivers = vmwareedge:NSX を追加し、NSX-T プロバイダを指定します。
 - b [oslo_messaging] セクションで、neutron プロセス空間で実行される Octavia と NSX-T ドライバ間の通信トピックを指定します。
- 3 Octavia API サービスを再起動します。

OpenStack Neutron の Load Balancer as a Service (LBaaS) の設定

ここでは、LBaaSv2 を使用して NSX-T Data Center ロード バランサで OpenStack を使用するように設定する手順について説明します。同じ手順で NSX-T Policy と NSX-T Manager の両方のプラグインを設定できます。

手順

- 1 `/etc/neutron/neutron.conf` のデフォルトの設定セクションに `service_plugins = vmware_nsx_lbaasv2, [...]` を設定し、NSX-T Data Center のロード バランシング サービス プラグインを追加します。

`service_plugins` はリスト オプションです。複数のサービス プラグインを指定するには、完全なクラス名またはショートカットをカンマで区切ります。

- 2 次のように `/etc/neutron/neutron-lbaas.conf` ファイルを編集します。

- a `service_providers` 設定セクションに `service_provider` オプションを設定し (`service_plugins = LOADBALANCERV2:VMWareEdge:neutron_lbaas.drivers.vmware.edge_driver_v2.EdgeLoadBalancerDriverV2:default`)、NSX-T Data Center のロード バランサ サービス ドライバを設定します。
- b まだ設定していない場合は、Keystone 認証パラメータを設定します。これらは `neutron-lbaas` によって使用されます。NSX-T Data Center の連携とは関係ありません。Keystone エンドポイントの場所を必ず指定してください。

```
[service_auth]
auth_version = 3
admin_password = password
admin_user = admin
admin_tenant_name = admin
auth_url = http://<keystone_endpoint>/identity/v3
```

- 3 `/etc/neutron/neutron-lbaas.conf` ファイルが `neutron server` コマンドラインに追加されていることを確認します。`ps -aux | grep neutron` コマンドを実行して、出力に `/etc/neutron/neutron-lbaas.conf` が含まれているかどうか確認します。

ファイルが含まれていない場合は、`neutron` サービス ランチャを編集する必要があります。サービス ランチャの場所と構造は、使用する OpenStack ディストリビューションによって異なります。

- 4 `neutron` サービスを再起動します。使用する OpenStack ディストリビューションによって、サービス名が異なる場合があります。

OpenStack Neutron の Firewall as a Service (FWaaS) の設定

ここでは、FWaaS v2 を設定する場合の手順について説明します。

手順

- 1 `/etc/neutron/neutron.conf` のデフォルトの設定セクションに `service_plugins = firewall_v2` を設定し、NSX-T Data Center のサービス プラグインとしてファイアウォールを追加します。

`service_plugins` は、リスト オプションです。複数のサービス プラグインを指定する場合は、完全なクラス名またはショートカットをカンマで区切って指定します。

2 次のように /etc/neutron/neutron-fwaas.conf ファイルを編集します。

- a service_providers 設定セクションに service_provider オプションを設定し、NSX-T Data Center のサービス ドライバとしてファイアウォールを設定します。service_provider =FIREWALL_V2:fwaas_db:neutron_fwaas.services.firewall.service_drivers.agents.agents.FirewallAgentDriver:default

```
[service_auth]
auth_version = 3
admin_password = password
admin_user = admin
admin_tenant_name = admin
auth_url = http://<keystone_endpoint>/identity/v3
```

このオプションの値は、次のような構成とします：

<service_type>:<service_name>:<driver_class>:[<default>]。service_provider は、複数の文字列から構成されます。指定するたびに、オプションの値がリストに追加されます。複数のサービス プロバイダを指定するには、各プロバイダに service_provider オプションを設定します。

- b FWaaS 設定セクションで enabled = True を設定し、ファイアウォールをサービスに切り替えます。
 - c FWaaS 設定セクションに driver = vmware_nsxp_edge_v2 を設定し、NSX-T Data Center の FWaaS デバイス ドライバを設定します。
 - d FWaaS 設定セクションに driver = vmware_nsxv3_edge_v2 を設定し、NSX Manager プラグインの FWaaS デバイス ドライバを設定します。
- ## 3 /etc/neutron/neutron-fwaas.conf ファイルが neutron server コマンドラインに追加されていることを確認します。ps -aux | grep neutron を実行して、出力に /etc/neutron/neutron-fwaas.conf が含まれているかどうか確認します。

ファイルが含まれていない場合は、neutron サービス ランチャを編集する必要があります。サービス ランチャの場所と構造は、使用する OpenStack ディストリビューションによって異なります。

- 4 neutron サービスを再起動します。使用する OpenStack ディストリビューションによって、サービス名が異なる場合があります。

OpenStack Neutron の IPsec VPN as a Service (VPNaaS) の設定

このサービスを有効にすると、OpenStack ユーザーは Neutron ネットワークから安全な VPN トンネル経由でリモート サイトにアクセスできるようになります。

VPNaaS ドライバは、NSX-T Policy プラグインで使用できません。ここでは、NSX-T Manager プラグインの手順について説明します。

手順

- 1 `/etc/neutron/neutron.conf` のデフォルトの設定セクションに `service_plugins = vmware_nsx_vpnaas, [...]` を設定し、NSX-T Data Center のサービス プラグインとして IPsec VPN を追加します。

`service_plugins` は、リスト オプションです。複数のサービス プラグインを指定するには、完全なクラス名またはショートカットをカンマで区切ります。
- 2 次のように `/etc/neutron/neutron-vpnaas.conf` ファイルを編集します。`service_providers` 設定セクションに `service_provider` オプションを設定し、NSX-T Data Center 用の VPN サービス ドライバの読み込みを設定します。 `service_provider = VPN:vmware:vmware_nsx.services.vpnaas.nsxv3.ipsec_driver.NSXv3IPsecVpnDriver:default`

このオプションの値は次のような構成とします：`<service_type>:<service_name>:<driver_class>:<default>`。`service_provider` は、複数の文字列から構成されます。指定するたびに、オプションの値がリストに追加されます。複数のサービス プロバイダを指定する場合は、各サービス プロバイダに `service_provider` オプションを設定します。
- 3 `/etc/neutron/neutron-vpnaas.conf` ファイルが `neutron server` コマンドラインに追加されていることを確認します。 `ps -aux | grep neutron` を実行して、出力に `/etc/neutron/neutron-vpnaas.conf` が含まれているかどうか確認します。

ファイルが含まれていない場合は、`neutron` サービス ランチャを編集する必要があります。サービス ランチャの場所と構造は、使用する OpenStack ディストリビューションによって異なります。
- 4 `neutron` サービスを再起動します。使用する OpenStack ディストリビューションによって、サービス名が異なる場合があります。

Neutron 拡張サービスの設定ファイルのサンプル

```
[DEFAULT]
ovs_integration_bridge = nsxvswitch
dhcp_agent_notification = False
notify_nova_on_port_data_changes = True
notify_nova_on_port_status_changes = True
core_plugin = vmware_nsxv3
service_plugins =
vmware_nsx_lbaasv2,vmware_nsx_vpnaas,neutron_fwaas.services.firewall.fwaas_plugin_v2.FirewallPluginV2
[...]
neutron_vpnaas.conf
[DEFAULT]
[service_providers]
service_provider =
VPN:vmware:vmware_nsx.services.vpnaas.nsxv3.ipsec_driver.NSXv3IPsecVpnDriver:default
neutron_fwaas.conf
[DEFAULT]
[quotas]
# Number of firewalls allowed per tenant. A negative value means unlimited.
```

```
# (integer value)
#quota_firewall = 10
# Number of firewall policies allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_firewall_policy = 10
# Number of firewall rules allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_firewall_rule = 100

[service_providers]
service_provider =
FIREWALL_V2:fwaas-db:neutron_fwaas.services.firewall.service_drivers.agents.agents.FirewallAgentDriver
:default
[fwaas]
enabled = True
driver = vmware_nsxv3_edge_v2
neutron_lbaas.conf
[DEFAULT]
[quotas]
# Number of LoadBalancers allowed per tenant. A negative value
# means unlimited. (integer value)
#quota_loadbalancer = 10

# Number of Loadbalancer Listeners allowed per tenant. A negative
# value means unlimited. (integer value)
#quota_listener = -1
# Number of pools allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_pool = 10
# Number of pool members allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_member = -1
# Number of health monitors allowed per tenant. A negative value
# means unlimited. (integer value)
#quota_healthmonitor = -1
[service_auth]
auth_version = 3
admin_password = password
admin_user = admin
admin_tenant_name = admin
auth_url = http://<keystone_ip>/identity/v3
[service_providers]
service_provider =
LOADBALANCER_V2:VMWareEdge:neutron_lbaas.drivers.vmware.edge_driver_v2.EdgeLoadBalancerDriverV2:default
```

```
Octavia.conf
[DEFAULT]
verbose = True
transport_url = rabbit://<amqp_user>:<amqp_password>@<amqp_node>:5672/
debug = True

[api_settings]
default_provider_driver = vmwareedge
```

```

enabled_provider_drivers = vmwareedge:NSX
bind_port = 9875
api_handler = queue_producer
bind_host = 0.0.0.0

[database]
connection = mysql+pymysql://root:<db_password?@<db_node>:3306/octavia

[keystone_auth_token]
signing_dir =
memcached_servers = <memcached_node>:11211
cafile = <cabundle_path>
project_domain_name = Default
project_name = service
user_domain_name = Default
password = <password>
username = octavia
auth_url = http://<keystone_node>/identity
auth_type = password

[certificates]
server_certs_key_passphrase = insecure-key-do-not-use-this-key
ca_private_key_passphrase = foobar
ca_private_key = /etc/octavia/certs/private/cakey.pem
ca_certificate = /etc/octavia/certs/ca_01.pem

[controller_worker]
amp_ssh_key_name = octavia_ssh_key
amp_image_tag = amphora
network_driver = allowed_address_pairs_driver
compute_driver = compute_nova_driver
amphora_driver = amphora_haproxy_rest_driver
workers = 2
amp_active_retries = 100
amp_active_wait_sec = 2

[oslo_messaging]
topic = vmwarensxv_edge_lb
rpc_thread_pool_size = 2

[house_keeping]
load_balancer_expiry_age = 3600

[service_auth]
memcached_servers = <memcached_node>:11211
cafile = <cabundle_path>
project_domain_name = Default
project_name = admin
user_domain_name = Default
password = openstack
username = admin

```

付録：NSX-T Data Center Plugin for OpenStack の設定プロパティ

6

表 6-1. 設定プロパティ

セクション	変数	説明
nsx_p	nsx_api_managers	NSX Manager の IP アドレス。複数のアドレスを指定する場合は、カンマで区切ります。IP アドレスは [<scheme>://<ip_address>[:<port>] の形式にする必要があります。スキームを指定しないと、https が使用されます。ポートを指定しないと、ポート 80 (http の場合) またはポート 443 (https の場合) が使用されます。
	nsx_use_client_auth	ブール値。クライアント証明書認証を有効にする場合は True に設定します
	nsx_client_cert_file	クライアント証明書とプライベート キーを含むファイルのパスを PEM 形式で指定します。
	nsx_client_cert_pk_password	(オプション) プライベート キーを復号に使用するパスワード。
	nsx_api_user	NSX Manager API にアクセスするためのユーザー名。
	nsx_api_password	NSX Manager API にアクセスするためのパスワード。
	dns_domain	ホスト名の作成に使用されるドメイン。
	default_overlay_tz	default_edge_cluster
	default_vlan_tz	(オプション) VLAN またはフラット プロバイダネットワークを構築する場合にのみ必要です。物理ネットワークが指定されていない場合に、Neutron ネットワーク間のブリッジに使用されるデフォルト NSX VLAN トランスポート ゾーンの UUID または名前。
	edge_cluster	(オプション) Tier-1 ルーターが接続する Edge クラスタを指定します。Tier-1 ルーターに接続されている Edge クラスタは除きます。
	retries	(オプション) 古いリビジョンでエラーが発生したときに API 要求を再試行する最大回数。

表 6-1. 設定プロパティ （続き）

セクション	変数	説明
	ca_file	(オプション) NSX Manager サーバ証明書の検証で使用する CA バンドル ファイルを指定します。insecure が True に設定されている場合、このオプションは無視されます。insecure が False に設定され、ca_file が未設定の場合、システムのルート CA がサーバ証明書の検証に使用されます。
	insecure	(オプション) true の場合、NSX Manager サーバ証明書は検証されません。false の場合、ca_file の CA バンドルが使用されます。未設定の場合は、デフォルトのシステム ルート証明書が使用されます。
	http_timeout	(オプション) NSX Manager との HTTP 接続を中止するまでの時間 (秒)。
	http_read_timeout	(オプション) NSX Manager からの HTTP 読み取り応答を中止するまでの時間 (秒)。
	http_retries	(オプション) HTTP 接続を再試行する最大回数。
	concurrent_connections	(オプション) 各 NSX Manager との最大接続数。
	conn_idle_timeout	(オプション) Manager 接続が使用されていない場合に、NSX Manager との接続が確立するまでの待機時間 (秒)。
	default_tier0_router	(オプション) Tier-1 論理ルーターとの接続と外部ネットワークの設定に使用されるデフォルトの Tier-0 ルーターの UUID または名前。
	metadata_on_demand	(オプション) True の場合、DHCP が無効になっているサブネットに接続している場合にのみ、ルーターの内部メタデータ ネットワークが作成されます。
	dhcp_profile	(オプション) ネイティブ DHCP サービスの有効化に使用される NSX DHCP プロファイルの UUID。これは、NSX プラグインを使用して Neutron を開始する前に NSX で作成しておく必要があります。
	locking_coordinator_url	(オプション) ロック マネージャの分散ロック調整リソースの URL。この値は、パラメータとして tooz コーディネーターに渡されます。デフォルトでは、この値は None で、単一ノードのロック管理に oslo_concurrency が使用されます。
	realization_max_attempts	(オプション) リソース認識の最大再試行回数。 デフォルト: 50
	realization_wait_sec	(オプション) リソース認識の試行間隔 (秒)。デフォルト: 1 秒