



OpFlex ML2 および GBP の展開

この章の内容は、次のとおりです。

- [OpFlex ML2 および GBP の展開, 1 ページ](#)
- [前提条件, 1 ページ](#)
- [展開の概要, 3 ページ](#)
- [Repo Server でのソフトウェア アーカイブのステージング, 3 ページ](#)
- [OpenStack サーバの準備, 4 ページ](#)
- [OpenStack Neutron サーバのアップデート, 7 ページ](#)
- [OpFlex エージェントおよびホストのインストールと設定, 13 ページ](#)
- [VXLAN カプセル化を使用する OpFlex エージェント ポートの設定, 15 ページ](#)
- [VLAN カプセル化を使用する OpFlex エージェント ポートの設定, 17 ページ](#)
- [エージェント サービスの開始と有効化, 18 ページ](#)
- [ACI テナントの初期化, 19 ページ](#)

OpFlex ML2 および GBP の展開

ここでは、Red Hat OpenStack ディストリビューションでの Cisco ACI OpenStack プラグインのインストールおよび設定の方法について説明します。

これらの手順の例では、OpenStack の Juno、Kilo、および Liberty リリースで検証を行いました。OpenStack システムは、インストール方法によって大きく異なる可能性があります。したがって、ここで示した例は、特定のインストール状態に適応するための基本として使用してください。

前提条件

ここでは、前提条件について説明します。

- 対象読者：Linux、Red Hat OpenStack ディストリビューション、ACI ポリシー モデル、GUI ベースの APIC 設定に関する実際的な知識があること。
- ACI ファブリック：ACI ファブリックがインストール済みであり、1.1(4e) および 11.1(4e) バージョン以上を使用して初期化されていること。新規 ACI ファブリックの初期化に関する基本ガイドラインについては、「[ACI ファブリック初期化の例](#)」を参照してください。複数のリーフ ペア間の通信については、OpenStack 外部ネットワークが使用できるように、ファブリックの BGP ルート リフレクタが有効化されている必要があります。
- OpenStack：RHEL 7 以降の Red Hat RDO/OSP Juno または Kilo リリースのインストール済みバージョンで Extra Packages for Enterprise Linux (EPEL) が有効化されていること。場合によっては EPEL-testing repos が有効化されていること。詳細については、「[Repo Server](#)」を参照してください。Packstack ベースのインストールでは、応答ファイルの CONFIG_USE_EPEL=y を使用して EPEL を追加できます。Juno リリースの場合、カーネルバージョン 3.10.0-229.14.1.el7.x86_64 がシステムで要求されます。Kilo リリースおよびそれ以降のリリースの場合、このカーネルバージョンは必要ありません。代わりにアップストリーム OVS/OpenvSwitch カーネル モジュールが使用されるためです。これは、OpenStack のクリーンインストールであり、既存ネットワークやルータが存在しないことが必要です。すべてのネットワーク サービスは ACI によって提供されるためです。
- Repo Server：OpenStack サーバの管理ネットワークからアクセス可能な RHEL または CentOS サーバが必要です。Cisco OpFlex ソフトウェアの yum ベースのインストール用リポジトリを収容するために使用されます。

インストール済みサーバの現在のカーネルバージョンを確認するには、次のコマンドを入力します。

```
uname -a
```

カーネルバージョンの変更が必要な場合は、次の構文を使用します。

```
yum -y install <kernel version>
```

次に例を示します。

```
yum -y install kernel-3.10.0-229.14.1.el7.x86_64
```



(注) EPEL-testing repo は、OpenStack の Kilo バージョンには不要です。

APIC でのルート リフレクタの作成は、システムのマニュアルに従って Web インターフェイス経由で完了できます。また、ACI OpenStack プラグイン ソフトウェアのインストール後に OpenStack コントローラ上で次のコマンドを使用して完了することもできます。

```
apic route-reflector-create --apic-ip <ip address> \
--apic-password <password> --apic-username \
<username> --no-secure
```

展開の概要

ここでは、展開の概要について説明します。

前提条件が満たされている状態で、Cisco ACI OpenStack プラグインのインストールと設定を開始できます。インストールプロセスの概要は次のとおりです。

- Repo サーバ上での OpFlex プラグイン ソフトウェア アーカイブのステージング
- OpenStack サーバの準備 (DHCP および LLDP、yum repo の有効化、ACI インフラストラクチャ VLAN 用 NIC の準備)
- OpenStack Neutron ネットワーク ノードの設定
- OpFlex エージェントのインストールおよびホスト設定
- 次のいずれかを選択します。
 - VXLAN カプセル化対応の OpFlex ポート設定
 - VLAN カプセル化対応の OpFlex ポート設定
- エージェント サービスの開始
- ACI OpFlex テナントの初期化

インストール時に、OpenStack サーバと ACI リーフ スイッチ間における VLAN カプセル化または VXLAN カプセル化を選択できます。リーフ スイッチ上のすべてのトラフィックは、ACI ファブリックで VXLAN にカプセル化されます。サーバとリーフ スイッチ間の VXLAN カプセル化により、OpenStack インストールで使用可能なネットワーク数を拡大できる可能性があります。ただし、コンピューティング ノードに必要なパケット トラフィックの処理も確実に増大します。VXLAN のオフロード機能を利用することにより、ある特定のネットワーク インターフェイスカードを使用してこの影響を相殺できます。OpenStack サーバ上の VLAN モード カプセル化により、膨大な数のテナントネットワークは必要としないシステムのコンピューティング ノードでのオーバーヘッドが緩和されます。

常に VXLAN または VLAN カプセル化の手順に従ってください。OpFlex エージェント コンフィギュレーション タスクについては、それぞれのカプセル化ごとに個別の項に記載されています。

OpFlex プラグインは、インストールおよび設定が完了すると、自動の OpenStack テナント ネットワーキングに対して動作します。このガイドの別の項には、送信元 NAT (sNAT) およびフローティング IP アドレスを使用して OpenStack クラウドを外部ネットワークに拡張する方法についての説明があります。

Repo Server でのソフトウェアアーカイブのステージング

ここでは、Repo Server でのソフトウェアのアーカイブのステージング方法について説明します。

RHEL または CentOS システムでの ACI OpenStack プラグインおよび関連ソフトウェアのインストールは、yum および RPM パッケージ管理を通じて行います。Repo Server は、システム内のすべての OpenStack サーバ ノードに対して一貫したコードバージョンを一元的に分散するポイントとして機能します。このサーバとしては、実際の環境内の任意の多目的サーバを使用できます。また緊急時には、この機能は OpenStack サーバ ノードのいずれかで提供できます。Repo Server は、ACI OpenStack プラグインのインストールとアップグレード時にのみアクセスできます。

手順

-
- ステップ 1** Juno または Kilo の正しいバージョンおよび Cisco ACI OpenStack プラグイン リリース アーカイブをシスコの Web サイト [Download Software](#) から Repo Server にダウンロードします。
- ステップ 2** httpd サービスがインストールされ起動していることを確認します。必要に応じて、サービスのインストール、起動、有効化を行います。

例 :

```
yum install httpd
systemctl start httpd
systemctl enable httpd
```

- ステップ 3** Repo Server の /var/www/html ディレクトリの下に「opflex」などの名前を選択してディレクトリを作成します。archive tar ファイルを新しいディレクトリに移動し、ファイルに un-tar を実行して、yum 用の repo を作成します。作成後、repo から提供されるすべてのファイルの所有者が apache ユーザに設定されていることを chown コマンドで確認します。

例 :

```
mv <release-archive-name> /var/www/html/opflex
cd /var/www/html/opflex
tar xvf <release-archive-name>
createrepo /var/www/html/opflex
cd ..
chown -R apache:apache opflex
```

- ステップ 4** これで yum repo を OpenStack サーバに配信する準備が完了しました。管理/SSH インターフェイス上の Repo Server に接続する IP がこれらのサーバに設定されていることを確認してください。
-

OpenStack サーバの準備

ここでは、OpenStack サーバの準備方法について説明します。

OpFlex の ACI ファブリックと正しく対話できるように OpenStack サーバ ノードを準備する必要があります。これには、ACI インフラストラクチャ VLAN 上のインターフェイスの DHCP 設定、および LLDP 通信が含まれます。また、OpFlex ソフトウェア repo の Repo Server 上の yum リポジトリを指すようにサーバをセットアップする必要があります。

手順

ステップ 1 OpenStack Neutron およびコンピューティング サーバは、ACI ファブリックのインフラストラクチャ (infra) VLAN に一致する 802.1Q VLAN タグ付きトラフィック用に設定された ACI 接続ネットワーク インターフェイスを必要とします。Cisco VIC カードを搭載した Cisco UCS サーバを使用する場合、上述のインターフェイスをオペレーティング システムの Linux サブインターフェイスとして設定することや、タギング機能を仮想 NIC (vNIC) にオフロードすることができます。

(注) Cisco VIC カードを使用する場合、VIC 上のローカル LLDP 機能を無効にしてください。VLAN タギングを持つ Cisco VIC カードを使用する ACI に VPC を接続するための Cisco C シリーズ サーバの設定方法の詳細については、[ホスト vPC の手動設定](#)を参照してください。

infra VLAN 用に Linux レベルのサブインターフェイスを使用するには、インターフェイス設定ファイルを作成します。そのファイルには、親物理インターフェイスまたはボンドインターフェイスの名前を指定し、その後にピリオドと ACI infra VLAN の番号を指定します。ACI ファブリック上のデフォルト infra VLAN は 4093 です。たとえば、親インターフェイスが eth1 と命名されている場合、サブインターフェイスの設定ファイルは /etc/sysconfig/network-scripts/ifcfg-eth1.4093 となります。この設定ファイルの内容例を以下に示します。

```
PERSISTENT_DHCLIENT=1
DHCPRELEASE=1
DEVICE=eth1.4093
ONBOOT=yes
PEERDNS=yes
NM_CONTROLLED=no
MACADDR= <unique MAC address>
TYPE=Ethernet
BOOTPROTO=dhcp
VLAN=yes
ONPARENT=yes
MTU=1600
```

ステップ 2 親インターフェイスの MAC アドレスとは異なる一意な MAC アドレスをサブインターフェイスが持ち、これが設定ファイルの MACADDR= 行に設定されていることを確認します。このアドレスが親インターフェイスと重複する場合には、アップストリーム スイッチとの LLDP 通信に問題が発生する可能性があります。また、この MAC アドレスが VLAN でも一意であることを確認します。親インターフェイスには MTU を 1600 に設定することが必要です。そうでない場合、サブインターフェイスの MTU が大きくなりません。ifcfg-eth1 ファイルを編集し、以下の内容を追加します。

例：
MTU=1600

ステップ 3 インターフェイスをバウンスします。

例 :

```
ifdown eth1
ifup eth1
```

(注) Cisco VIC カードを使用して OpFlex infra VLAN 通信の仮想インターフェイスを提供している場合は、一意のアドレスが CIMC によって自動的に生成されるため、一意のアドレスをファイルに追加する必要はありません。

サブインターフェイスと親インターフェイスはともに、VXLAN ヘッダーがパケットに追加されるように MTU を大きくする必要があります。設定例の MTU=1600 の行がこれに対応しています。同じ行を親インターフェイスの設定ファイルに追加してください。

ステップ 4 Infra VLAN 上のネットワーク インターフェイスは、OpFlex 通信用に APIC インフラストラクチャネットワークからの DHCP アドレスを要求します。サーバがリースに関する DHCP オプションのすべてを正しく受け取るためには、このインターフェイスに関する `dhclient` 設定がサーバに必要です。OpenStack サーバの VPC インターフェイスを設定する方法については、[ホスト vPC の手動設定](#)を参照してください。

(注) この項のインターフェイス例では、このマニュアルの「付録」の例に示すように、ACI Infra VLAN トラフィックを伝送するインターフェイスには、「ten-bond」という名前を参照します。実環境における infra VLAN インターフェイスは、「eth0.4093」など、基本的な Linux レベルのサブインターフェイスにすることもできます。以下の内容を使用して `/etc/dhcp/dhclient-ten-bond.conf` ファイルを作成し、ファイルの最初の行の各サーバのイーサネット インターフェイスの MAC アドレスを挿入します。

例 :

```
send dhcp-client-identifier 01:<mac-address of infra VLAN interface>;
request subnet-mask, domain-name, domain-name-servers, host-name;
send host-name <server-host-name>;

option rfc3442-classless-static-routes code 121 = array of unsigned integer 8;
option ms-classless-static-routes code 249 = array of unsigned integer 8;
option wpad code 252 = string;

also request rfc3442-classless-static-routes;
also request ms-classless-static-routes;
also request static-routes;
also request wpad;
also request ntp-servers;
```

ステップ 5 マルチキャストルートは特に `opflex` インフラストラクチャ VLAN インターフェイスに適用する必要があります。これは、次の内容を含む `/etc/sysconfig/network-scripts/route-ten-bond` ファイルを作成することにより達成できます。

例 :

```
ADDRESS0=224.0.0.0
NETMASK0=240.0.0.0
GATEWAY0=0.0.0.0
METRIC0=1000
```

ステップ 6 ACI ファブリックが OpenStack ノードの動的な検出を使用できるようにするには、サーバ上にソフトウェア LLDP スタックが必要です。LLDP パッケージをインストールするには、次のコマンドを実行します。

```
例 :
cd /etc/yum.repos.d/
yum install wget
wget http://download.opensuse.org/repositories/home:vbernat/RHEL_7/home:vbernat.repo
yum install lldpd
```

ステップ 7 サーバのブート後に lldpd サービスをリロードする機能を開始および有効化します。

```
例 :
service lldpd start
chkconfig lldpd on
```

(注) ホストオペレーティングシステムのバージョンにもよりますが、ACI ファブリックが動的にサーバノードを検出できる場合に限り、代替ソフトウェア LLDP スタックを使用できます。コンピューティングノードで LLDP が有効化されていない場合は、ml2_conf_cisco_apic.ini から手動で LLDP を設定する必要があります。構文例については、[ホストリンクの自動設定のセットアップ](#) を参照してください。

ステップ 8 OpenStack ネットワーキングおよびコンピューティングノードは、それらの yum 設定にポインタが追加されていることが必要です。それにより、Repo Server から OpFlex ソフトウェアをプルできるようになります。次の内容を含む /etc/yum.repos.d/opflex.repo ファイルを作成し、baseurl ステートメントの行に Repo Server の IP アドレスを代入してください。

```
例 :
[opflex]
name=opflex repo
baseurl=http://<ip address>/opflex
failovermethod=priority
enabled=1
gpgcheck=0
```

この設定が完了したら、repo が正しく動作しており、エラーがないことを確認します。

```
例 :
yum makecache
```

OpenStack Neutron サーバのアップデート

ここでは、OpenStack Neutron サーバのアップデート方法について説明します。

OpenStack システムの Neutron サーバは、ACI ファブリックと間での OpenStack テナントのダイナミックプロビジョニングに関する主要なやり取りを APIC 経由で提供します。ここでは、OpFlex エージェントとドライバのインストールについて説明するとともに、APIC 通信に必要な特定の設定ファイルの編集についても説明します。ACI OpenStack プラグインを使用することにより、通常は OpenStack Neutron L3 エージェントサービスによって提供されるレイヤ 3 転送機能レイヤを、ACI ファブリックが置き換えることができます。このサービスは、今後使用されなくなります。

手順

ステップ 1 次のコマンドを使用して、Neutron サーバ上のサービスを無効にする必要があります。

例：
`systemctl stop neutron-l3-agent`
`systemctl disable neutron-l3-agent`

ステップ 2 OpenStack コントローラ ノードで、必要なサポート モジュールとともに、neutron-opflex-agent、APICAPI、ML2/GBP ドライバをインストールします。これらのパッケージは EPEL repo から取得され、インストールに成功するには、ノードで EPEL が有効化されている必要があります。サポート モジュールである python-pip と python-pbr も前提条件として必要です。

例：
`yum install python-pip`
`yum install python-pbr`

ステップ 3 Juno を実行している場合、epel-testing repo が有効化されていることを確認するための手順として、/etc/yum.repos.d/epel-testing.repo ファイルを変更し、[epel-testing] セクションの下に enabled=1 が設定されていることを確認します（次の例を参照）。

例：

```
[epel-testing]
name=Extra Packages for Enterprise Linux 7 - Testing - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/testing/7/$basearch
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=testing-epel7&arch=$basearch
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
```

ステップ 4 opflex エージェント、apicapi、ml2 ドライバをインストールします。

例：
`yum install neutron-opflex-agent apicapi neutron-ml2-driver-apic`

ステップ 5 GBP ベースのインストールの場合には、以下の追加パッケージをインストールする必要があります。

- openstack-neutron-gbp
- python-gbpclient
- openstack-dashboard-gbp
- openstack-heat-gbp

例 :

```
yum install openstack-neutron-gbp python-gbpclient openstack-dashboard-gbp \
openstack-heat-gbp
```

- ステップ 6** インストールが完了したら、ネットワーク サービスの APIC を指すように /etc/neutron/neutron.conf ファイルを更新する必要があります。ファイル内のサービスプラグインの既存リストを次のように変更します。
ML2 の場合 :

例 :

```
service_plugins = cisco_apic_l3, metering, lbaas
```

GBP の場合 :

```
service_plugins = group_policy, servicechain, apic_gbp_l3, metering
```

(注) このプラグインに必要なサービスと競合しないサービスを除去しないように注意する必要があります。たとえば、lbaas や計測サービスが有効化されている場合、上述の例に示すように、それらを引き続き有効化しておく必要があります。

- ステップ 7** GBP ベースのインストールの場合、GBP のヒートプラグインを有効化する必要があります。その操作は、次に示すように、/etc/heat/heat.conf ファイルの DEFAULT セクションの plugin_dirs に必ず GBP パスを含めることによって実行できます。

例 :

```
plugin_dirs = /usr/lib/python2.7/site-packages/gbpautomation/heat
```

- ステップ 8** ML2 設定ファイル /etc/neutron/plugins/ml2/ml2_conf.ini で次の変更を実施して、APIC 用のメカニズム ドライバを有効化し、OpFlex を新しいネットワーク タイプとして追加することも必要です。

例 :

```
type_drivers = opflex, local, flat, vlan, gre, vxlan
tenant_network_types = opflex
```

ML2 の場合 :

例 :

```
mechanism_drivers = cisco_apic_ml2
```

GBP の場合 :

例 :

```
mechanism_drivers = apic_gbp
```

- ステップ 9** VXLAN カプセル化を使用している場合は、/etc/neutron/plugins/ml2/ml2_conf.ini ファイルを編集し、以下の行をコメントアウトします。

例：
network_vlan_ranges =

ステップ 10 VLAN カプセル化を使用している場合は、`/etc/neutron/plugins/ml2/ml2_conf.ini` ファイルを編集し、以下の行を使用して VLAN の範囲を `[ml2_type_vlan]` セクションに追加します。

例：
network_vlan_ranges = **physnet1:1000:2000**

ステップ 11 キーワード `physnet1` は、同じセクションの `bridge_mappings` で定義されたものです。以下のように `bridge_mappings` を定義するステートメントが同じセクションに存在することを確認してください。

例：
bridge_mappings = **physnet1:br-prv**

このファイルで定義された VLAN の範囲は、ACI OpenStack プラグインが APIC 上に VLAN プールを作成するために使用します (`ml2_conf_cisco_apic.ini` ファイルの `apic_provision_infra` が `True` に設定されている場合)。

ステップ 12 `/etc/neutron/dhcp_agent.ini` ファイルを編集し、`dhcp_driver` を変更した後、他の値を確認します。

例：
dhcp_driver = **apic_ml2.neutron.agent.linux.apic_dhcp.ApicDnsmasq**
ovs_integration_bridge = br-int
enable_isolated_metadata = True

`ovs_integration_bridge = br-int` 行がコメントアウトされていないことを確認します。

ステップ 13 OpFlex agent-ovs コンポーネントは、デフォルトで各コンピューティング ノード上の VM インスタンスにローカル DHCP リース配信を提供します。分散動作を制御するための設定は、`ml2_conf_cisco_apic.ini` ファイルの `enable_optimized_dhcp` で指定できます。このデフォルト設定 (ファイルで上書きされていない場合の設定) は「true」です。
`neutron-dhcp-agent` プロセスは引き続き Neutron サーバで必要とされます。これは、agent-ovs DHCP 機能に対する dnsmasq プロセスの IP アドレス管理と適切な通信を処理するために使用されます。すべての設定変更を確実に適用をするために、`neutron-dhcp-agent` を再起動します。

例：
systemctl restart neutron-dhcp-agent

ステップ 14 `ml2_conf_cisco_apic.ini` ファイルは、Neutron サーバ上の主要な設定ファイルであり、ACI OpenStack プラグインと ACI APIC との対話をカスタマイズするために使用されます。APIC IP アドレス、クレデンシャル、ACI ポリシー モデルにおけるオブジェクトのデフォルト命名法はここで設定します。次のファイル例では、使用する ACI 環境に合わせるために必要な関連設定を示し、説明しています。

(注) この例は、LLDP ベースのホスト検出で使用するために設計されたものです。手動でホスト検出を設定する場合には、エントリを `ml2_conf_cisco_apic.ini` ファイルにも適用してください。手動設定の詳細については、[ホストリンクの自動設定のセットアップ](#)を参照してください。

`/etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini` ファイルを編集します。

例 :

```
[DEFAULT]
apic_system_id= <any string>

[opflex]
networks = '*'

[ml2_cisco_apic]

# Hostname:port list of APIC controllers
apic_hosts = <comma-separated list of APIC IP addresses>

# Username for the APIC controller
apic_username= <username with administrative access to APIC>

# Password for the APIC controller
apic_password= <password for apic_username>

# Whether use SSL for connecting to the APIC controller or not
apic_use_ssl = True

# How to map names to APIC: use_uuid or use_name.
apic_name_mapping = use_name

# Agent timers for State reporting and topology discovery
apic_agent_report_interval = 30
apic_agent_poll_interval = 2
apic_provision_infra = True
apic_provision_hostlinks = False
enable_optimized_dhcp = True
enable_optimized_metadata = True
```

ここで、`<any string>` は、ドライバによる APIC オブジェクトの自動作成において、OpenStack システム用 ACI テナントとして使用される名前です。

`apic_provision_infra = True` は、APIC で VMM ドメインを作成するために最初にシステムが起動されるときに必要とされます。既存のサーバ接続が使用されており、すでに APIC で定義されている場合、作成された VMM ドメインも、これらの接続の作成時に使用された AEP に手動で関連付ける必要があります。True の設定により、テナント ネットワーク用に VLAN プールを作成する機能も有効化されています (VLAN カプセル化モードが使用されている場合)。

`apic_provision_hostlinks = False` は、手動サーバポートプロビジョニングです。

`enable_optimized_dhcp = True` は、デフォルトで `true` です。

`enable_optimized_metadata = True` は、メタデータを分散する場合に使用します。

ステップ 15 GBP の場合、`[group_policy]` セクションを `ml2_conf_cisco_apic.ini` ファイルに追加します。ターゲットポリシーグループのサブネットは、`192.168.0.0/16` アドレス空間から切り分けられます。

例 :

```
[group_policy]
policy_drivers=implicit_policy,apic
```

```
[group_policy_implicit_policy]
default_ip_pool=192.168.0.0/16
```

ステップ 16 Ml2_conf_cisco_apic.ini ファイルの編集が完了したら、これを OpenStack neutron-server サービスのサービス定義に追加して、サービスの起動時にオプション用に読み取られるようにする必要があります。/usr/lib/systemd/system/neutron-server.service ファイルを編集し、--config-file /etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini を ExecStart 行に追加します。

例：

```
ExecStart=/usr/bin/neutron-server \
--config-file /usr/share/neutron/neutron-dist.conf \
--config-file /etc/neutron/neutron.conf \
--config-file /etc/neutron/plugin.ini
--config-file /etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini \
--log-file /var/log/neutron/server.log
```

ステップ 17 (オプション) リリース 1.2(2x) では、OpenStack 経由で作成される可能性のあるサブネットの制約を追加で指定できます。これらの制約により、特定サブネットの作成拒否、パブリック指定、プライベート指定などを APIC で行えます。

a) 制約ファイル cisco_apic_network_constraints.ini を指すように、ml2_conf_cisco_apic.ini ファイルの [Ml2_cisco_apic] セクションを編集します。以下の行を ml2_conf_cisco_apic.ini ファイルに追加します。

例：

```
[ml2_cisco_apic]
network_constraints_filename = /etc/neutron/plugins/ml2/cisco_apic_network_constraints.ini
```

b) ネットワーク制約ファイル

/etc/neutron/plugins/ml2/cisco_apic_network_constraints.ini を編集して、制約を記述します。詳細については、[ネットワーク制約テンプレートファイル](#)を参照してください。

ネットワーク制約ファイルは、後からいつでも変更でき、変更を有効にするために Neutron サーバを再起動する必要はありません。

(注) 複数の Neutron コントローラを高可用性モード使用する展開においては、正しく動作するために制約ファイルがすべてのコントローラで同一であることが必要です (他の設定ファイルと同様)。

ステップ 18 Neutron サーバ サービス定義をアップデートして設定ファイルを読み込んだら、systemd manager 設定をリロードした後、次のコマンドを使用して Neutron サーバを再起動します。

例：

```
systemctl daemon-reload
service neutron-server restart
```

OpFlex エージェントおよびホストのインストールと設定

ここでは、OpFlex エージェントおよびホストのインストールと設定の方法について説明します。
Neutron サーバノードおよびコンピューティングノードはともに、Neutron OpFlex エージェント、および OVS をプログラムする OpFlex エージェント (agent-ovs) のインストールと設定が必要です。

はじめる前に

Neutron ノードには、すでに neutron-opflex-agent がインストールされている必要があります (OpenStack Neutron サーバのアップデート、(7 ページ) で実行)。

手順

ステップ 1 neutron-opflex-agent および agent-ovs を yum リポジトリからインストールします。

例：

```
yum install neutron-opflex-agent
yum install agent-ovs
```

ステップ 2 /etc/neutron/plugins/ml2/openvswitch_agent.ini ファイルに次の例に示す設定が含まれていることを確認します。Liberty より以前のリリースを使用している場合は、代わりに /etc/neutron/plugins/openvswitch/ovs_neutron_plugin.in ファイルを使用します。

例：

```
[ovs]
enable_tunneling = False
integration_bridge = br-int
```

また、tunnel_bridge、vxlan_udp_port、tunnel_types の設定行が削除またはコメントアウトされていることも確認してください。

ステップ 3 Neutron-openvswitch-agent を無効化および停止して、次のコマンドを入力します

例：

```
systemctl disable neutron-openvswitch-agent
service neutron-openvswitch-agent stop
```

ステップ 4 Kilo 以降を実行している場合は、このステップを省略してステップ 5 に進みます。

(注) CentOS 上で Kilo を実行している場合は、Open vSwitch のバージョンが 2.4.0 以降であることを確認する必要があります。バージョンを調べるには、次のコマンドを実行します。

例：

```
# ovs-vsctl show

285c1281-1e63-49fd-a7c7-828e31b071b0
    Bridge br-tun
        Port br-tun
```

```

        Interface br-tun
            type: internal
    Bridge br-int
        fail_mode: secure
    Port of-svc-ovsport
        Interface of-svc-ovsport
    Port "br-int_vxlan0"
        Interface "br-int_vxlan0"
            type: vxlan
            options: {dst_port="8472", key=flow, remote_ip=flow}
    Port br-int
        Interface br-int
            type: internal
    ovs_version: "2.4.0"

```

ovs_version が 2.4.0 より前である場合は、Open vSwitch パッケージバージョン 2.4.0 以降を取得する必要があります。1つの方法として、Open vSwitch パッケージを次の CentOS ミラーから取得できます：http://mirrors.mit.edu/centos/7.2.1511/virt/x86_64/ovirt-3.6/openvswitch-2.4.0-1.el7.x86_64.rpm

以前のバージョンを実行している場合、OpenStack とともにインストールされたデフォルトの Open vSwitch エージェントは OpFlex 設定に適していません。ACI ファブリックと正しくやり取りできるように修正されたシスコ専用の Open vSwitch パッケージをインストールする必要があります。次のコマンドを入力してください。

例：

```
yum install openvswitch-gbp
```

- ステップ 5** 新しい設定および現時点までの OVS モジュールを使用してシステムがクリーンな状態で実行することを保証するために、各サーバをリブートします。
- ステップ 6** サーバのリブートプロセスが完了したら、ログインし、ディレクトリを /etc/opflex-agent-ovs/conf.d に変更します。
- ステップ 7** agent-ovs サービスがその設定を /etc/opflex-agent-ovs/opflex-agent-ovs.conf ファイルから読み込み、その conf.d サブディレクトリでは、より小さな JSON 形式ファイルを使用して、そのファイル内の特定の設定を細かく上書きできます。以下に示した例の内容を使用して、新規に /etc/opflex-agent-ovs/conf.d/10-opflex-connection.conf ファイルを作成します。

例：

```

{
    "opflex": {
        "domain":
"comp/prov-OpenStack/ctrlr- [<apic_system_id>]-<apic_system_id>/sw-InsiemeLSOid",
        "name": " <hostname of this system> ",
        "peers": [
            {"hostname": "10.0.0.30", "port": "8009"}
        ],
        "ssl": {
            "mode": "encrypted"
        }
    }
}

```

ここで、<apic_system_id> は、Neutron サーバの /etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini ファイルで使用したものと同じです。

<hostname of this system> は、OpenStack ホストのそれぞれにローカルな Linux サーバ ホスト名です。

- ステップ 8** 新しい `/etc/opflex-agent-ovs/conf.d/10-opflex-connection.conf` ファイルを保存し、JSON 構造の他の形式や括弧表記が変更されていないことを確認します。
- ステップ 9** ACI ファブリックがトンネル エンドポイントに対するデフォルトの IP アドレス プールを使用してインストールされている場合、この例のホスト名の隣の IP アドレスが OpFlex 通信のデフォルトのファブリック インターフェイスです (10.0.0.0/16)。この IP アドレス プールがファブリックのインストール中に変更されている場合は、ここで使用されるアドレッシングをファブリックに合わせて変更します。SSH でリーフ スイッチに接続し、`show ip interface` コマンドを使用して、ファブリックで使用されているアドレスを特定します。OpFlex ピアのホスト名アドレスは、リーフ スイッチの上の `infra VLAN` の SVI に割り当てられたユニキャスト IP アドレスです。
- ステップ 10** サーバとリーフ スイッチ間に VXLAN カプセル化を使用している場合は、次の [VXLAN カプセル化を使用する OpFlex エージェント ポートの設定](#)、(15 ページ) に進んでください。VLAN カプセル化を使用している場合は、[VLAN カプセル化を使用する OpFlex エージェント ポートの設定](#)、(17 ページ) までスキップしてください。

VXLAN カプセル化を使用する OpFlex エージェント ポートの設定

ここでは、VXLAN カプセル化を使用する OpFlex エージェント ポートの設定方法について説明します。

この項は、OpenStack サーバと ACI リーフ スイッチ間で VXLAN カプセル化を使用することを選択した場合に適用されます。

手順

- ステップ 1** OpFlex の設定には、ホストとリーフ スイッチ間の VXLAN 設定に固有な 2 番目の上書き値セットが必要です。次に示す例の内容を使用して、新規に `/etc/opflex-agent-ovs/conf.d/20-vxlan-aci-renderer.conf` ファイルを作成します。

例:

```
{
  "renderers": {
    "stitched-mode": {
      "ovs-bridge-name": "br-int",
      "encap": {
        "vxlan": {
          "encap-iface": "br-int_vxlan0",
          "uplink-iface": "eth1.4093",
          "uplink-vlan": 4093,

```

```

        "remote-ip": "10.0.0.32",
        "remote-port": 8472
    },
    "flowid-cache-dir": "/var/lib/opflex-agent-ovs/ids"
}
}
}

```

ここで、**eth1.4093** は、実際に使用する OpFlex infra VLAN インターフェイスのインターフェイス名と VLAN 番号です。

4093 は、実際に使用する OpFlex infra VLAN インターフェイスの VLAN 番号です。

ステップ 2 新しい `/etc/opflex-agent-ovs/conf.d/20-vxlan-aci-renderer.conf` ファイルを保存し、JSON 構造の他の形式や括弧表記が変更されていないことを確認します。

ステップ 3 ACI ファブリックがトンネルエンドポイントに対するデフォルトの IP アドレス プールを使用してインストールされている場合、`20-vxlan-aci-renderer.conf` ファイルの `remote-ip` の IP アドレスが OpFlex 通信のデフォルトのファブリック インターフェイスになります (10.0.0.0/16)。この IP アドレス プールがファブリックのインストール中に変更されている場合は、ここで使用されるアドレッシングをファブリックに合わせて変更します。SSH でリーフスイッチに接続し、`show vlan extended` コマンドおよび `show ip interface` コマンドを使用して、ファブリックで使用されているアドレスを特定します。`remote-ip` アドレスは、リーフスイッチ上でインターフェイス ループバック 1023 に割り当てられたエニーキャスト IP アドレスと一致しません。

ステップ 4 OpenStack サーバと ACI リーフスイッチ間で VXLAN カプセル化を使用するには、VXLAN インターフェイスを OVS で定義する必要があります。このインターフェイス名は、`opflex-agent-ovs.conf` ファイル内の `encap-iface` 設定に一致する必要があります。次のコマンドを入力します。

例 :

```

ovs-vsctl add-port br-int br-int_vxlan0 -- set Interface br-int_vxlan0 type=vxlan
options:remote_ip=flow options:key=flow options:dst_port=8472

```

ステップ 5 OpenStack のプロビジョニングに使用したインストール ツールによっては、OVS セットアップで必要ではないポートやブリッジが設定されていることがあります。たとえば、`br-ex` と呼ばれる OVS ブリッジは、通常、Neutron ノード上の外部ネットワーク用にプロビジョニングされるものであり、不要になります。`br-ethX` などのインターフェイスブリッジは、通常、VLAN トラフィックを伝送するために、VLAN モードの `packstack` インストールによってプロビジョニングされます。その機能は、`br-int` に直接追加されたテナント ネットワーク インターフェイスに置き換えられています。`ovs-vsctl` コマンドの `del-br` および `del-port` を使用して、不要なブリッジやパッチ接続を削除できます。シンプルになった OVS 設定は、次に示す `ovs-vsctl show` の出力のようになります。

例 :

```

Bridge br-int
    fail_mode: secure
    Port br-int
        Interface br-int
            type: internal
    Port "br-int_vxlan0"
        Interface "br-int_vxlan0"

```



```

    type: vxlan
    options: {dst_port="8472", key=flow, remote_ip=flow}
    ovs_version: "2.4.1.gbp"

```

VM インスタンスがコンピューティング ノード上で起動されると、システムは動的に OVS インターフェイスを `qvo` から順に `br-int` に追加して、各 VM の接続に使用される個々の Linux ブリッジにそれらをリンクします。VM トラフィックは、`br-int` を通過し、`agent-ovs` によるプログラムに従って、テナント VXLAN インターフェイスから ACI ファブリックに横断します。

VLAN カプセル化を使用する OpFlex エージェント ポートの設定

ここでは、VLAN カプセル化を使用する OpFlex エージェント ポートの設定方法について説明します。

この項は、OpenStack サーバと ACI リーフ スイッチ間で VLAN カプセル化を使用することを選択した場合に適用されます。

手順

- ステップ 1** OpFlex の設定には、ホストとリーフ スイッチ間の VLAN 設定に固有な 2 番目の上書き値セットが必要です。次に示す例の内容を使用して、新規に `/etc/opflex-agent-ovs/conf.d/20-vlan-aci-renderer.conf` ファイルを作成してください。

例：

```

{
  "renderers": {
    "stitched-mode": {
      "ovs-bridge-name": "br-int",

      "encap": {
        "vlan": {
          "encap-iface": "<tenant-VLAN-trunk>"
        }
      },
      "flowid-cache-dir": "/var/lib/opflex-agent-ovs/ids"
    }
  }
}

```

ここで、`<tenant-VLAN-trunk>` は、実際に使用するテナント VLAN トランク インターフェイスのインターフェイス名です。

- ステップ 2** 新しい `/etc/opflex-agent-ovs/conf.d/20-vlan-aci-renderer.conf` ファイルを保存し、JSON 構造の他の形式や括弧表記が変更されていないことを確認します。
- ステップ 3** コンピューティング ノードからの OpenStack テナント ネットワーキング用のインターフェイスは、VLAN トランッキングをサポートする物理インターフェイスです。場合によっては、これが

infra VLAN サブインターフェイスの親インターフェイスになります。VPC の場合、Cisco VIC ベースの設定について [ホスト vPC の手動設定](#) の説明を参照してください。これは、LACP トラフィックが送信される独立した main-bond インターフェイスです。このインターフェイス名は、opflex-agent-ovs.conf ファイル内の encap-iface 設定に一致する必要があります。次のコマンド構文を使用して、テナント VLAN トランク インターフェイスを OVS ブリッジ br-int に追加します。

例：

```
ovs-vsctl add-port br-int <tenant-VLAN-trunk>
```

ステップ 4

OpenStack のプロビジョニングに使用したインストールツールによっては、OVS セットアップで必要ではないポートやブリッジが設定されていることがあります。たとえば、br-ex と呼ばれる OVS ブリッジは、通常、Neutron ノード上の外部ネットワーク用にプロビジョニングされるものであり、不要になります。br-ethX などのインターフェイスブリッジは、通常、VLAN トラフィックを伝送するために、VLAN モードの packstack インストールによってプロビジョニングされます。その機能は、br-int に直接追加されたテナント ネットワーク インターフェイスに置き換えられています。ovs-vsctl del-br コマンドおよび ovs-vsctl del-port コマンドを使用して、不要なブリッジやパッチ接続を削除できます。シンプルになった OVS 設定は、次に示す ovs-vsctl show の出力のようになります。

例：

```
Bridge br-int
  fail_mode: secure
  Port br-int
    Interface br-int
      type: internal
  Port <tenant-VLAN-trunk>
    Interface <tenant-VLAN-trunk>
  ovs_version: "2.4.1"
```

VM インスタンスがコンピューティング ノード上で起動されると、システムは動的に OVS インターフェイスを qvo から順に br-int に追加して、各 VM の接続に使用される個々の Linux ブリッジにそれらをリンクします。VM トラフィックは、br-int を通過し、agent-ovs によるプログラムに従って、テナント VLAN インターフェイスから ACI ファブリックに横断します。

エージェントサービスの開始と有効化

ここでは、エージェントサービスを開始および有効化する方法について説明します。

手順

- ステップ 1** OpFlex が適切に設定されている状態で、neutron-opflex-agent および agent-ovs サービスを開始および有効化し、次のコマンドを入力します。

例 :

```
service agent-ovs restart
service neutron-opflex-agent restart
systemctl enable agent-ovs
systemctl enable neutron-opflex-agent
```

- ステップ 2** OpenStack サーバと ACI リーフ スイッチの間でホストサーバ接続の LLDP 自動検出を提供するには、APIC ホスト エージェントが必要です。ホスト エージェントは、ACI ファブリックからの LLDP 情報をリッスンして、各コンピューティング ノードに接続されているリーフ スイッチと物理ポートを特定する情報を復号化し、OpenStack コントローラに対してその情報を更新します。エージェントを起動し、その永続性を保証するには、次の手順を実行します。

例 :

```
service neutron-cisco-apic-host-agent restart
systemctl enable neutron-cisco-apic-host-agent
```

- ステップ 3** すべてのサービスが稼働したら、OpFlex infra VLAN のインターフェイスが UP の状態であることを確認するか、`ifup <interface-name>` コマンドを使用してインターフェイスを起動する必要があります。

ACI テナントの初期化

ここでは、ACI テナントの初期化方法について説明します。

手順

- ステップ 1** 現在、ACI OpenStack プラグインソフトウェアは稼働しており、OpenStack のテナントネットワークをプロビジョニングする準備ができています。APIC への OpenStack 設定の読み込みは、ACI OpenStack プラグインがアクティブの状態の OpenStack で最初のネットワーク セグメントが作成されるまで開始されません。ACI テナントおよび APIC の VMM ドメインの最初の作成をトリガーするために、OpenStack の管理者プロジェクトの下にテスト用 Neutron ネットワークを作成します。
- ステップ 2** ネットワークが作成されたら、APIC GUI にログインします。
- (注) Application Policy Infrastructure Controller (APIC) 1.2(1x) リリースの場合、APIC GUI にログインする際に [Advanced] モードを選択します。
- APIC 1.2(1X) リリースの場合、シスコでは、コンフィギュレーションモード (拡張または基本) を混在させないことをお勧めしています。いずれかのモードで設定を作成し、他方のモードを使用して設定を変更すると、意図しない変更が発生する可能性があります。たとえば、拡張モードを使用して 2 つのポートにインターフェイス ポリシーを適用し、次に基本モードを使用して 1 つのポートの設定を変更すると、変更内容が両方のポートに適用される可能性があります。

- a) メニューバーで、[TENANTS] を選択して、新しく作成された ACI テナントが実際の ACI OpenStack プラグイン システム名を使用して命名されていることを確認します。

ステップ 3 メニューバーで、[VM NETWORKING] を選択します。

- a) [navigation] ペインで、[OpenStack] を展開し、実際のシステム用に作成された VMM ドメインが存在することを確認します。
- b) この VMM ドメインは、OpenStack サーバ接続が APIC にプロビジョニングされたときにインターフェイス ポリシー グループによって参照された AEP に関連付ける必要があります。メニューバーで、[FABRIC] > [ACCESS POLICIES] を選択します。
- c) [navigation] ペインで、[Global Policies] > [Attachable Access Entity Profiles] を展開し、使用している OpenStack サーバの [Interface Policies] > [Policy Groups] 定義によって参照される AEP を選択します。
- d) [PROPERTIES] ペインの [Domains] フィールドで、「[+]」アイコンをクリックして、OpenStack VMM ドメインを AEP の関連ドメインのリストに追加します。
- e) [Submit] をクリックします。

ステップ 4 ACI テナントが OpenStack から初期化された状態で、複数のネットワーク、VM インスタンス、OpenStack 内のルータを少しずつずらして起動し、予測される接続を確認することで、インストールの基本機能を確認できます。OpenStack Horizon または CLI インターフェイスを経由してすべての動作をオーケストレートしながら、APIC 内の ACI テナントの下で動的に作成される EPG およびブリッジ ドメインを観察できます。

(注) /Etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini ファイルの `apic_provision_infra=true` 設定は、VMM ドメイン、AEP、VLAN プールの作成をトリガーします (VLAN カプセル化が使用されている場合)。また、APIC の [FABRIC] > [ACCESS POLICIES] の下に表示されるインターフェイスとスイッチ レベルのポリシー グループの作成もトリガーします。手動で設定されたサーバ ホストリンクが使用中の場合でも、アンダースコア文字から始まる特殊なインターフェイスとスイッチのポリシー グループは参照されません。これらのグループは、そのままにしても削除してもかまいません。