# Ubuntu 向け Cisco ACI with OpenStack OpFlex 展開ガイド

**初版**：2016 年 02 月 11 日

**最終更新**：2016 年 07 月 07 日

**シスコシステムズ合同会社**
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

# 目 次

# はじめに

この前書きは、次の項で構成されています。



## 対象読者

このガイドは、次の 1 つ以上に責任を持つ、専門知識を備えたデータセンター管理者を主な対象にしています。

- 仮想マシンのインストールと管理
- サーバ管理
- スイッチおよびネットワークの管理

## 表記法

コマンドの説明には、次のような表記法が使用されます。

| 表記法 | 説明 |
|---|---|
| **bold** | 太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。 |
| *italic* | イタリック体の文字は、ユーザが値を入力する引数です。 |

| 表記法 | 説明 |
|---|---|
| [x] | 省略可能な要素（キーワードまたは引数）は、角カッコで囲んで示しています。 |
| [x \| y] | いずれか 1 つを選択できる省略可能なキーワードや引数は、角カッコで囲み、縦棒で区切って示しています。 |
| {x \| y} | 必ずいずれか1つを選択しなければならない必須キーワードや引数は、波カッコで囲み、縦棒で区切って示しています。 |
| [x {y \| z}] | 角カッコまたは波カッコが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角カッコ内の波カッコと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。 |
| `variable` | ユーザが値を入力する変数であることを表します。イタリック体が使用できない場合に使用されます。 |
| string | 引用符を付けない一組の文字。string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。 |

例では、次の表記法を使用しています。

| 表記法 | 説明 |
|---|---|
| `screen フォント` | スイッチが表示する端末セッションおよび情報は、screen フォントで示しています。 |
| **太字の screen フォント** | ユーザが入力しなければならない情報は、太字の screen フォントで示しています。 |
| イタリック体の *screen* フォント | ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。 |
| < > | パスワードのように出力されない文字は、山カッコ（< >）で囲んで示しています。 |
| [ ] | システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。 |
| !、# | コードの先頭に感嘆符（!）またはポンド記号（#）がある場合には、コメント行であることを示します。 |

このマニュアルでは、次の表記法を使用しています。

（注） 「注釈」です。役立つ情報やこのマニュアルに記載されていない参照資料を紹介しています。

注意 「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

警告 IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger.You are in a situation that could cause bodily injury.Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents.Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

# 関連資料

### シスコ アプリケーション セントリック インフラストラクチャ（ACI）のマニュアル

ACI のマニュアルは、次の URL で入手できます。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

### シスコ アプリケーション セントリック インフラストラクチャ（ACI）シミュレータのマニュアル

Cisco ACI Simulator のマニュアルは、次の URL から入手できます。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-centric-infrastructure-simulator/tsd-products-support-series-home.html。

### Cisco Nexus 9000 シリーズ スイッチのマニュアル

Cisco Nexus 9000 シリーズ スイッチのマニュアルは、次の URL で入手できます。http://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/tsd-products-support-series-home.html

### Cisco Application Virtual Switch のマニュアル

Cisco Application Virtual Switch（AVS）のマニュアルは、次の URL で入手できます。http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/tsd-products-support-series-home.html

### シスコ アプリケーション セントリック インフラストラクチャ（ACI）と OpenStack の統合に関するマニュアル

Cisco ACI と OpenStack の統合に関するマニュアルは、次の URL から入手できます。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

# マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、apic-docfeedback@cisco.com までご連絡ください。ご協力をよろしくお願いいたします。

# マニュアルの入手方法およびテクニカル サポート

マニュアルの入手、Cisco Bug Search Tool（BST）の使用、サービス要求の送信、追加情報の収集の詳細については、『*What's New in Cisco Product Documentation*』を参照してください。このドキュメントは、次から入手できます。 http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html

『*What's New in Cisco Product Documentation*』では、シスコの新規および改訂版の技術マニュアルの一覧を、RSS フィードとして購読できます。また、リーダー アプリケーションを使用して、コンテンツをデスクトップに配信することもできます。RSS フィードは無料のサービスです。

第 1 章

# 新機能および変更された機能に関する情報

この章の内容は、次のとおりです。



## 新機能および変更された機能に関する情報

次の表は、この最新リリースまでのガイドでの主な変更点の概要を示したものです。ただし、このリリースまでのこのガイドの変更点や新機能の中には一部、この表に記載されていないものもあります。

**表 *1*：*Ubuntu* 向け *Cisco ACI with OpenStack OpFlex* 展開ガイドの新機能と変更された機能**

| Cisco APIC のリリース バージョン | 機能 | 説明 | 参照先 |
|---|---|---|---|
| 1.2(2g) | -- | 物理ドメインを使用する ACI での OpenStack の展開に対するサポートが追加されました。 | 詳細については、物理ドメインを使用する ACI での OpenStack の展開，（31 ページ）を参照してください。 |
| 1.2(2g) | -- | OpenStack の Liberty リリースに対するサポートが追加されました。 | -- |

新機能および変更された機能に関する情報

| Cisco APIC のリリースバージョン | 機能 | 説明 | 参照先 |
|---|---|---|---|
| 1.2(2g) | ACI サポート、アウトバウンドおよびインバウンドのプレフィックスリスト、ルートマップベースのフィルタリング | OpenStack 経由で作成される可能性のあるサブネットの制約指定に対するサポートが導入されました。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。 | 詳細については、OpenStack Neutron サーバのアップデート, （10 ページ）およびネットワーク制約テンプレート ファイル, （50 ページ）を参照してください。 |
| 1.2(1i) | -- | このマニュアルの大きな変更はありません。 | -- |
| 1.1(4e) | -- | このガイドがリリースされました。 | -- |

第 2 章

# 概要

この章の内容は、次のとおりです。



## OpenStack と Cisco ACI について

Cisco Application Centric Infrastructure（ACI）は、インテリジェントなコントローラベースのネットワーク スイッチング ファブリックを実現する包括的なポリシーベースのアーキテクチャです。このファブリックは、OpenStack など、複数のオーケストレーション、自動化、管理ツールに直接統合可能な API インターフェイスからプログラムによって管理されるように設計されています。ACI を OpenStack と統合することによって、ネットワーク構造の動的な作成を OpenStack 要件に従って直接駆動するだけでなく、ACI アプリケーション ポリシー インフラストラクチャ コントローラ（APIC）内のさらなる可視性を個別の VM インスタンスのレベルに至るまで実現できます。

OpenStack は、クラウド コンピューティング環境を構築するための柔軟なソフトウェア アーキテクチャを定義します。OpenStack のリファレンス ソフトウェアベースの実装により、VLAN、GRE、VXLAN など、複数のレイヤ 2 転送が実現されます。OpenStack 内の Neutron プロジェクトでは、ソフトウェア ベースのレイヤ 3 フォワーディングも提供できます。ACI と連携して使用することにより、ACI ファブリックは、レイヤ 2 およびレイヤ 3 が統合された VXLAN ベースのオーバーレイ ネットワーキング機能を提供します。この機能により、ネットワーク カプセル化の処理を、コンピューティング ノードからトップオブラック（TOR）または ACI リーフ スイッチにオフロードできます。このアーキテクチャは、ソフトウェア オーバーレイ ネットワーキングの柔軟性を提供するとともに、ハードウェアベースのネットワーキングのパフォーマンス上および動作上の利点も提供します。

Cisco ACI OpenStack プラグインは、ML2 モードまたは GBP モードで展開できます。ML2（モジュラ レイヤ 2）モードでは、ネットワークの作成に標準の Neutron API が使用されます。これは OpenStack に VM およびサービスを導入するための従来の方法です。GBP（グループ ベースのポリシー）モードでは、アプリケーションの説明、作成、展開を行うための新しい API がポリシー グループとして提供され、ネットワーク固有の詳細を考慮する必要がなくなります。詳細については、次の URL にある『*OpenStack Group-Based Policy User Guide*』を参照してください。 http://

www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/openstack/b_OpenStack_Group-Based_Policy_User_Guide.html

第 3 章

# 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

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, （39 ページ）」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：Ubuntu 14.4 以降に展開された Juno または Kilo リリースのインストール済みバージョン。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 Linux サーバが必要です。Cisco OpFlex ソフトウェアの apt 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
  ```

  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

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、apt 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 インストレーション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が、コンピューティング ノードで必要なパケット 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。

Ubuntu または Debian 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、`apt` および `dpkg` パッケージ管理によって実施されます。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のみにアクセスできます。

### 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 conref 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 Web サイトから Repo Server にウンロードします。

**ステップ 2** `apache2` 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、開始、有効化を次のように行います。

例：
```
apt-get install apache2
```

**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、アーカイブ ファイルに un-tar を実行して、apt 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 www-data 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>
cd /var/www/html/opflex
dpkg-scanpackages . > Packages
cd ..
chown -R www-data opflex
```

これにより、trivial apt repo が作成されます。

高度な設定については、Ubuntu のマニュアルを参照してください。

**ステップ 4** これで apt 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 apt 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, （41 ページ）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/network/interfaces`.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。

```
auto eth1.4093
iface eth1.4093 inet dhcp
hwaddress ether <eth1 mac address, or self created mac, see note>
vlan-raw-device eth1
pre-up /sbin/ip link set dev eth1.4093 mtu 1600
post-up /sbin/route -nv add -net 224.0.0.0/4 dev eth1.4093
```

**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。これを確認するには、`eth1` インターフェイスの設定に pre-up ステートメントを追加します。以下に例を示します。

**例：**

```
auto eth1
iface eth1 inet manual
pre-up /sbin/ip link set dev eth1 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, （41 ページ） 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。

/etc/dhcp/dhclient.conf ファイルを編集して以下の内容を追加し、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：
```
interface "eth1.4093" {send host-name "<hostname>";
send dhcp-client-identifier 01:<interface MAC address>; }
```

**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、上記のように、post-up ステートメントをインターフェイス設定ファイルに追加することで達成できます。

例：
```
post-up /sbin/route -nv add -net 224.0.0.0/4 dev eth1.4093
```

**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：
```
apt-get install lldpd
```

（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 ファイルから手動で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, （46 ページ） を参照してください。

**ステップ 7** 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 apt 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/apt/sources.list.d/opflex.list ファイルを作成し、deb 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：
```
deb http://10.10.225.2:8080/plugins/aci_opflex-0.2/repositories/ubuntu /
```

**ステップ 8** 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：
```
apt-get update
```

# 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 通信に必要となる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：
```
service neutron-l3-agent stop
mv /etc/init/neutron-l3-agent.conf \
/etc/init/neutron-l3-agent.disabled
```

**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、APIC API、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：
```
apt-get install python-pip
apt-get install python-pbr
```

**ステップ 3** opflex エージェント、apicapi、ml2 ドライバをインストールします。

例：
```
apt-get install neutron-opflex-agent python-apicapi \
neutron-ml2-driver-apic
```

**ステップ 4** 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。

- group-based-policy
- python-group-based-policy-client
- group-based-policy-ui
- group-based-policy-automation

例：
```
apt-get install group-based-policy \
python-group-based-policy-client group-based-policy-ui \
group-based-policy-automation
```

Python-click-cli に対する依存度に関するエラーが表示されたら、Ubuntu パッケージの Web サイトから python-click-cli をインストールしてください。詳細については、http://packages.ubuntu.com/wily/all/python-click-cli/download を参照してください。

**ステップ 5** 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：

例：
```
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 6** 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 7** 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 8** 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 9** 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 10** 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_mapppings = 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 11** /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 12** 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：
```
service neutron-dhcp-agent restart
```

**ステップ 13** 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 ACIAPIC 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，（46 ページ）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_sync_interval = 0
apic_agent_report_interval = 30
apic_agent_poll_interval = 2
enable_aci_routing = True
enable_arp_flooding = True
apic_provision_infra = True
apic_provision_hostlinks = False
enable_optimized_dhcp = True
enable_optimized_metadata = True
integrated_topology_service = 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。

integrated_topology_service = True により、LLDP の検出が合理化されます。

**ステップ 14** 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 15** 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。/etc/init/neutron-server.conf ファイルを編集し、--config-file /etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini を exec 行に追加してください。

**例：**

```
exec start-stop-daemon --start --chuid neutron --exec
/usr/bin/neutron-server -- \
    --config-file /etc/neutron/neutron.conf \
    --config-file /etc/neutron/plugins/ml2/ml2_conf.ini \
    --config-file /etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini \
    --log-file /var/log/neutron/server.log $CONF_ARG
```

**ステップ 16** （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, （50 ページ）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 17** 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、次のコマンドを使用して Neutron サーバを再起動します。

例：
```
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，（10 ページ）で実行）。

## 手順

**ステップ 1** これらのエージェントをapt opflex リポジトリからインストールします。

例：
```
apt-get install neutron-opflex-agent
apt-get 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 ファイルを使用してください。

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 を停止および無効化して、次のコマンドを入力します。

例：
```
service neutron-plugin-openvswitch-agent stop
mv /etc/init/neutron-plugin-openvswitch-agent.conf
/etc/init/neutron-plugin-openvswitch-agent.disabled
```

**ステップ 4** Liberty 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。

Kilo 以前のバージョンを実行している場合は、以下のように動作します。

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 設定では使用されません。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：

```
apt-get install openvswitch-datapath-dkms=2.4.1\*
apt-get install openvswitch-common=2.4.1\*
apt-get install openvswitch-switch=2.4.1\*
apt-get 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 に割り当てられたエニーキャスト 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, （18 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, （20 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す。

**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, （41 ページ） 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
```

**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するには、次のように実行します。

例：
```
service neutron-cisco-apic-host-agent restart
```

**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る状態です。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。

第 4 章

# 最適化 Neutron サービスの有効化

この章の内容は、次のとおりです。



# 最適化 Neutron サービスの有効化

ここでは、分散 DHCP 機能およびメタデータ プロキシ機能の設定について説明します。

ACI OpenStack プラグイン ソフトウェア スタックでは、ローカル レイヤ 3、NAT、DHCP、メタデータ プロキシに対して最適化された機能を有効化できます。ローカル レイヤ 3 転送は、システムに組み込まれています。そのため、設定は不要です。

コンピューティング ノード上の分散 NATサービスは、Neutron の外部ネットワークの有効化と連動しています（「OpenStack 外部ネットワークの追加，（27 ページ）」を参照）。

## 最適化 DHCP サービス

ファイル `/etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini` に `enable_optimized_dhcp` という設定行があり、ファイルに明記されていない場合のデフォルト設定は「`True`」になります。デフォルトを使用する場合または「True」に設定する場合、VM インスタンスと対話する Discovery、Offer、Response、Acknowledgement（DORA）の機能は、各コンピューティング ノードにローカルに保持されます。

ローカルの `agent-ovs` サービスは、コンピューティング ノードごとにこの対話を処理します。アドレス割り当ては、`neutron-dhcp` エージェントによって引き続き Neutron サーバで処理され、管理ネットワーク上で `agent-ovs` インスタンスに伝達されます。「`False`」に設定すると、システムは Neutron サーバ上でのすべてのアドレス割り当て機能および DORA 機能に関して集中型の DHCP 機能に戻ります。

# 最適化メタデータ プロキシ

ここでは、最適化メタデータ プロキシを有効化する方法について説明します。

最適化されたメタデータ プロキシを有効にするには、/etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini ファイル内の enable_optimized_metadata と呼ばれる設定が必要です。デフォルト設定は「False」です。したがって、ファイル内で参照されない場合は、テナント ネットワーク上で一元化された従来のメタデータ プロキシが使用されます。

**手順**

**ステップ 1** /etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini ファイルを編集し、enable_optimized_metadata=True を設定します。

**ステップ 2** /etc/neutron/metadata-agent.ini ファイルが存在し、正しいことを確認します。ファイルが存在しない場合は、コントローラからこのファイルをコピーする必要があります。

**ステップ 3** neutron-ml2-driver-apic パッケージもコンピューティング ノードにインストールされていることを確認します。

**ステップ 4** Neutron サーバ上のメタデータ エージェントを無効化し、neutron-server を再起動して、次のコマンドを入力します。

**例：**

```
service neutron-metadata-agent stop
mv /etc/init/neutron-metadata-agent.conf /etc/init/neutron-metadata-agent.disabled
service neutron-server restart
```

**ステップ 5** コンピューティング ノードで、neutron-ml2-driver-apic パッケージをインストールする必要があります。このパッケージが利用可能になり、opflex エージェント サービスが再起動されると、システムはメタデータ プロキシの分散モードで機能を開始します。次のコマンドを入力してください。

**例：**

```
apt-get install neutron-ml2-driver-apic
service neutron-opflex-agent restart
service agent-ovs restart
```

第 5 章

# OpenStack 外部ネットワークの追加

この章の内容は、次のとおりです。



# OpenStack 外部ネットワークの追加

ここでは、OpenStack 外部ネットワークを追加する方法について説明します。

外部 OpenStack ネットワークのための OpFlex 設定には、外部ルーテッド ネットワークまたはレイヤ 3 Out が APIC テナントまたは共通設定に存在することが必要です。このレイヤ 3 Out は、ACI ファブリック外の通信用の外部ルーティング エンティティまでのパスを提供します。レイヤ 3 Out 設定には多数の種類があります。OSPF、BGP、スタティック ルーティングを使用できます。ルーテッド インターフェイス、SVI を持つ vPC、ルーテッド サブインターフェイスも使用できます。ACI OpenStack プラグインは、論理構造としての名前を使って既存のレイヤ 3 Out と対話できます。実際の環境に適したルーティングの設定は ACI システム管理者が担います。レイヤ 3 Out がまだ存在していない場合、簡単な設定に使用できる手順についてACI 外部ルーテッド ネットワークの例，（47 ページ） を参照するか、またはホワイト ペーパー『*Connecting Application Centric Infrastructure (ACI) to Outside Layer 2 and 3 Networks*』を参照してください。入手先： http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c07-732033.html

## はじめる前に

- 外部ネットワーク上の SNAT およびフローティング IP コミュニケーションには、2 つの独立した IP サブネットが必要です。ACI ファブリック外のアップストリーム ルータは、これら 2 つのサブネットの IP ルートを使用して設定する必要があります。その際、使用中のルーティング プロトコル経由で実行する方法と静的に実行する方法があります。
- この設定を実行するには、送信元 NAT およびフローティング IP 機能をサポートするための IP サブネットの要件に精通している必要があります。

OpenStack 外部ネットワークの追加

### 手順

**ステップ 1** Neutron サーバ上で外部ルーテッド ネットワークと通信するように ACI OpenStack プラグインを設定し、/etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini ファイルを編集して、次のセクションを追加します。

例：

```
[apic_external_network: <name of L3-Out> ]
preexisting=True
enable_nat=False
external_epg= <name of EPG>
host_pool_cidr= <ip of SNAT default gateway/prefix bits>
```

ここで、*<name of L3-Out>* は、APIC のテナントまたは共通ネットワーキング フォルダに定義された外部ルーテッド ネットワークの名前です。

enable_nat=False は、このネットワーク上での NAT の動作を無効化します。デフォルトは enable_nat=True であるため、NAT の動作を使用する場合は、このパラメータを省略できます。

複数の L3-Out を使用して NAT と no NAT の動作を組み合わせることができます。各 L3-Out は独自の apic_external_network セクションを必要とするため、同じ名前で Neutron プロバイダー ネットワークを作成する必要があります。

ここで、*<name of EPG>* は、APIC の Layer 3 Out の下の Networks フォルダで定義された名前です。これは、最初に作成されたときに「EPG Network」と呼ばれています。

*<ip of SNAT default gateway/prefix bits>* は、SNAT に使用されるサブネット上のデフォルト ゲートウェイであり、/prefix の表記で識別されます（例：10.1.2.1/24）。これは、ACI OpenStack プラグインがそのサブネットを APICの正しいブリッジ ドメインの下に追加するために使用するアドレスです。

**ステップ 2** すべての Neutron ノードで、neutron-server サービスを再起動して新しい設定をアクティブにします。

**ステップ 3** OpenStack で外部ネットワークを作成します。ネットワークの名前は、ml2_conf_cisco_apic.ini ファイルの apic_external_network 値の名前と一致する必要があります。これは、Neutron CLI から実行することにより、新しい「OpFlex」タイプのネットワークを管理用に作成できます。Neutron サーバ上で管理クレデンシャルを確保した後、Neutron で net-create コマンドを使用して新しいネットワークを外部として追加し、共有を指定します。

例：

```
neutron net-create OS-L3Out --router:external --shared
```

**ステップ 4** 最初に net-create コマンドで作成した外部ネットワークにサブネットを追加します。ここで追加したサブネットはフローティング IP の範囲として使用され、ACI OpenStack プラグインによって APIC の正しいブリッジ ドメインにも追加されます。

**ステップ 5** neutron net-list コマンドを実行します。
名前の最初に「host-nat-network-for-internal-use」が付けられたドライバによって作成されたセカンダリ ネットワークが表示されます。OpenStack と ACI はこのネットワークを使用

OpenStack 外部ネットワークの追加

することにより、フローティング IP アドレスが割り当てられていない VM インスタンスの SNAT トラフィックを正しく処理することができます。OpFlex システムは、SNAT アドレスを `host_pool_cidr` サブネットから OpenStack クラスタ内の各コンピューティング ホストに自動的に割り当てます。

OpenStack 外部ネットワークの追加

付 録 A

# 参考資料

この章の内容は、次のとおりです。



# 物理ドメインを使用する ACI での OpenStack の展開

ここでは、物理ドメインを使用する ACI での OpenStack の展開方法を説明します。

**手順**

**ステップ 1** 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。OpenStack コントローラ ノードで、LLDP パッケージをインストールし、次のコマンドを実行します。

例：

```
apt-get install lldpd
```

（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` ファイルから手動で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, （46 ページ） を参照してください。

**ステップ 2** 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 apt 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。OpenStack コントローラ ノードで、次の内容を含む `/etc/apt/sources.list.d/opflex.list` ファイルを作成し、deb 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：
```
deb http://10.10.225.2:8080/plugins/aci_opflex-0.2/repositories/ubuntu /
```

**ステップ 3** この設定が完了したら、OpenStack コントローラ ノードで、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：
```
apt-get update
```

**ステップ 4** 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、APIC API、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：
```
apt-get install python-pip
apt-get install python-pbr
```

**ステップ 5** OpenStack コントローラ ノードで、opflex エージェント、apicapi、ml2 ドライバをインストールします。

例：
```
apt-get install neutron-opflex-agent python-apicapi \
neutron-ml2-driver-apic
```

**ステップ 6** GBP ベースのインストールの場合には、OpenStack コントローラ ノードで、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。

- group-based-policy
- python-group-based-policy-client
- group-based-policy-ui
- group-based-policy-automation

例：
```
apt-get install group-based-policy \
python-group-based-policy-client group-based-policy-ui \
group-based-policy-automation
```

Python-click-cli に対する依存度に関するエラーが表示されたら、Ubuntu パッケージの Web サイトから python-click-cli をインストールしてください。詳細については、http://packages.ubuntu.com/wily/all/python-click-cli/download を参照してください。

**ステップ 7** 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。OpenStack コントローラ ノードで、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編集します。
ML2 の場合：

例：
```
service_plugins = cisco_apic_l3,metering,lbaas
```

GBP の場合：

例：
```
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 8** OpenStack コントローラ ノードで、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。
GBP の場合には、次のドライバを使用します：openvswitch、apic_gbp

例：
```
[ml2]
type_drivers = local,flat,vlan,gre,vxlan
tenant_network_types = vlan
mechanism_drivers = openvswitch,cisco_apic_ml2

[ml2_type_vlan]
network_vlan_ranges = physnet1:2500:3000

[securitygroup]
enable_security_group = True
```

**ステップ 9** OpenStack コントローラ ノードで、/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
```

**ステップ 10** OpenStack コントローラ ノードで、dhcp エージェントを再起動します。

例：
```
service neutron-dhcp-agent restart
```

**ステップ 11** OpenStack コントローラ ノードで、/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 ファイルを使用してください。

例：
```
[ovs]
integration_bridge = br-int
local_ip = <Management IP of the server>
enable_tunneling = False
bridge_mappings = physnet1:br-eth

[agent]
polling_interval = 2
l2_population = False
arp_responder = False

[securitygroup]
enable_security_group = True
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
```

**ステップ 12** OpenStack コントローラ ノードで、br-eth ブリッジを作成し、アップリンク インターフェイスを追加します。

例：
```
ovs-vsctl add-br br-eth
ovs-vsctl add-port br-eth <Name of the uplink interface>
```

**ステップ 13** OpenStack コントローラ ノードで、openvswitch エージェントを再起動します。

例：
```
service neutron-openvswitch-agent restart
```

**ステップ 14** OpenStack コントローラ ノードで、/etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini ファイルを編集します。

例：
```
[ml2_cisco_apic]

apic_hosts = 172.31.218.136
apic_username = admin
apic_password = cisco123
apic_use_ssl = True
apic_name_mapping = use_name
enable_optimized_metadata = False
enable_optimized_dhcp = False
enable_aci_routing = True
apic_arp_flooding = True
```

```
apic_provision_hostlinks = True
apic_clear_node_profiles = True
apic_provision_infra = True
use_vmm = False


[apic_switch:101]
bm1.sys.cisco.com = 1/19

[apic_switch:102]
bm2.sys.cisco.com = 1/19


[DEFAULT]
apic_system_id = liberty-perf
```

**ステップ 15** GBP の場合、OpenStack コントローラ ノードで、[group_policy] セクションおよび [group_policy_implicit_policy] セクションを ml2_conf_cisco_apic.ini ファイルに追加します。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。OpenStack コントローラ ノードで、/usr/lib/systemd/system/neutron-server.service ファイルを編集し、[1] 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** OpenStack コントローラ ノードで、neutron サーバを再起動します。

**例：**

```
service neutron-server restart
```

**ステップ 18** コントローラ ノードで、/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 ファイルを使用してください。

**例：**

```
[ovs]
integration_bridge = br-int
local_ip = <Management IP of the server>
```

```
enable_tunneling = False
bridge_mappings = physnet1:br-eth

[agent]
polling_interval = 2
l2_population = False
arp_responder = False

[securitygroup]
enable_security_group = True
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
```

**ステップ 19** コンピューティング サーバで、br-eth ブリッジを作成し、アップリンク インターフェイスを追加します。

例：

```
ovs-vsctl add-br br-eth
ovs-vsctl add-port br-eth <Name of the uplink interface>
```

**ステップ 20** コンピューティング サーバで、openvswitch エージェントを再起動します。

例：

```
service neutron-openvswitch-agent restart
```

# 仮想ルーティングと転送およびネットワークアドレス変換

仮想ルーティングと転送（VRF）、またはネットワーク アドレス変換（NAT）を使用して Cisco Application Centric Infrastructure （ACI）を展開できます。次のいずれかの方法を使用してください。

- 構成上、IP アドレスの重複が必要な場合は、テナントごとに 1 つの VRF を使用します。

  詳細については、複数の仮想ルーティングと転送およびネットワーク アドレス変換の同時使用，（37 ページ）を参照してください。

- 構成上、IP アドレスの重複が必要ない場合は、OpenStack クラウドに対して単一の共有 VRF を使用します。

  詳細については、単一の共有仮想ルーティングと転送（ネットワーク アドレス変換なし），（37 ページ）を参照してください。

- フローティング IP アドレスを使用する予定がある場合は、複数の VRF または単一の共有 VRF による NAT が必要です。

  詳細については、複数の仮想ルーティングと転送およびネットワーク アドレス変換の同時使用，（37 ページ）および単一の共有仮想ルーティングと転送およびネットワーク アドレス変換の同時使用，（38 ページ）を参照してください。

# 複数の仮想ルーティングと転送およびネットワークアドレス変換の同時使用

仮想ルーティングと転送（VRF）およびネットワーク アドレス変換（NAT）の同時使用はデフォルトの動作です。このシナリオでは、Cisco Application Centric Infrastructure（ACI）管理者は、Common テナントの VRF に接続された Common テナントに L3Out を作成します。OpenStack プロジェクトが作成されると、追加の Application Policy Infrastructure Controller（APIC）テナントが作成され、それぞれが専用プライベート VRF とシャドウ L3Out を持ちます。Common テナント内の VRF に NAT 機能を示すために、シャドウ L3out が使用されます。

次の手順は、NATと複数の VRF 設定の概要を示しています。

**手順**

**ステップ 1** L3Out を作成する。

**ステップ 2** インターネット VRF を作成する。

**ステップ 3** L3Out をインターネット VRF に接続する。
追加の VRF が必要な場合は、システムによって作成されます。

このシナリオの設定ファイルには、次のエントリが含まれています。

```
[ml2_cisco_apic]
per_tenant_context = True
[apic_external_DC-Out]
preexisting = True
enable_nat = True
external_epg=DC-Out-EPG
host_pool_cidr=1.2.3.1/24
```

# 単一の共有仮想ルーティングと転送（ネットワーク アドレス変換なし）

ネットワーク アドレス変換（NAT）を使用せずに、単一の仮想ルーティングと転送（VRF）を Common テナント内で使用できます。NAT ありの単一 VRF との主な違いは、OpenStack ネットワークが L3Out 上で直接アドバタイズされることです。

（注） ダイナミック ルーティングを使用している場合、Neutronによって作成されたブリッジドメインは、L3Out に接続される必要があります。

次に、NAT なしの単一共有 VRF を設定する手順の概要を示します。

手順

**ステップ 1** L3Out を作成する。

**ステップ 2** 共有コンテキスト VRF（インターネット）を作成し、設定ファイルでこの VRF を *shared_context_name* として指定します。

**ステップ 3** L3Out を VRF に接続する。

このシナリオの設定ファイルには、次のエントリが含まれています。

```
[ml2_cisco_apic]
per_tenant_context = False
shared_context_name=my_shared_context

[apic_external_DC-Out]
preexisting = True
enable_nat = False
external_epg=DC-Out-EPG
```

# 単一の共有仮想ルーティングと転送およびネットワークアドレス変換の同時使用

単一の仮想ルーティングと転送（VRF）を Common テナント内で使用できます。OpenStack プロジェクトごとに別々の VRF を作成する必要はありません。単一の VRF を使用しているため、OpenStack プロジェクト間で IP アドレスの重複はありえません。

この設定には、ネットワーク アドレス変換（NAT）が引き続き使用されます。フローティング IP アドレスおよび SNAT のサブネットのみが外部にアドバタイズされ、外部ネットワークとの間を行き来するすべてのトラフィックは NAT によって変換されます。

単一 VRF モードは、per_tenant_context パラメータによってトリガーされます。使用するコンテキストの名前を指定する必要もありますが、指定しない場合は、デフォルトで「shared」という単一名を使用して作成されます。

次に、単一の共有 VRF を NAT とともに設定する手順の概要を示します。

手順

**ステップ 1** L3Out を作成する。

**ステップ 2** インターネット VRF を作成する。

**ステップ 3** L3Out をインターネット VRF に接続する。

**ステップ 4** 共有コンテキスト VRF を作成し、設定ファイルでこの VRF を *shared_context_name* として指定します。

このシナリオの設定ファイルには、次のエントリが含まれています。

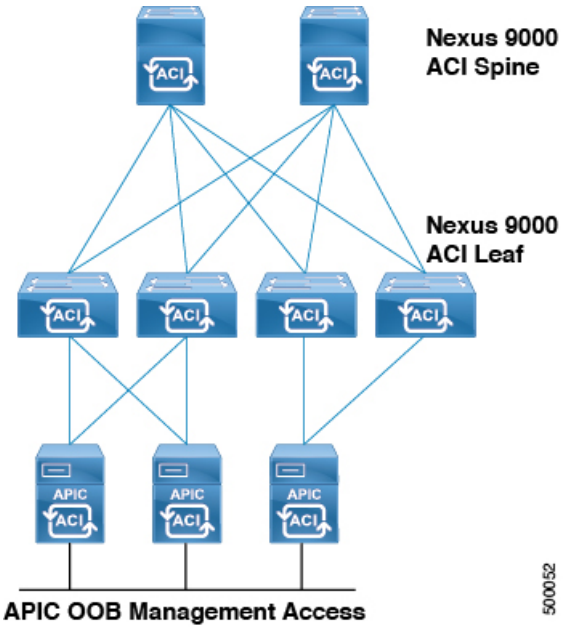
```
[ml2_cisco_apic]
per_tenant_context = False
shared_context_name=my_shared_context

[apic_external_DC-Out]
preexisting = True
enable_nat = True
external_epg=DC-Out-EPG
host_pool_cidr=1.2.3.1/24
```

# ACI ファブリック初期化の例

このソリューション例は、ファブリック名とコントローラ IP アドレッシング以外のすべての設定がデフォルト状態で APIC にインストールされた基本的なスパイン/リーフ スイッチング ファブリックに基づいています。可用性の高いクラスタを形成するために、3 つの APIC が使用されています。それぞれの APIC は、ファブリック内の複数のリーフ スイッチに接続されています。コントローラ サービスの可用性を向上させるには、多様なリーフ スイッチを使用して複数の APIC を接続することが最良の方法です。

スイッチング システムは、APIC クラスタの有無に関わらずトラフィックを転送し続けます。ファブリックのすべての設定はクラスタによって推進されるため、APIC の接続が正しく確立されていない状態では、設定の追加、変更、削除は一切できません。ファブリックの管理制御がファブリック自体に依存しないことを保証するためには、次の図に示すように、APIC のそれぞれにアウトオブバンド（OOB）ネットワーク接続が必要です。

**図 *1*：*APIC* クラスタ接続**



### 手順

**ステップ 1** ACI ファブリックの設定では、ファブリックを検出する前にファブリック内の各スイッチのシリアル番号をメモすると効果的です。理想的には、各スイッチのコンソール ポートにもターミナル サーバを接続して、ACIファブリックの状態に関わらず、常に管理制御が存在する状態にします。ACI ソフトウェア イメージを実行するスイッチにログインする際にシリアル番号を回復するには、show inventory コマンドを ACI スイッチの CLI から入力し、プライマリ システムのシリアル番号をメモします。この番号は、ファブリック検出時に APIC に表示され、この番号を使用することにより、スキーム内の正しい名前とノードの番号付けをデバイスに割り当てることができます。

**ステップ 2** APIC がファブリック内のスイッチを検出および登録できるようにするために、APIC GUI にログインします。

a) メニュー バーで、[FABRIC] > [INVENTORY] を選択します。

b) [Navigation] ペインで、[Fabric Membership] を選択します。

c) [Work] ペインには、APIC によって検出された最初のスイッチのエントリが表示されます。

d) これがクラスタ内の最初の APIC に対して想定される最初のスイッチであることを、シリアル番号に基づいて確認します。

e) [Work] ペインでスイッチを選択し、右クリックして [Register Switch] を選択します。

（注） その後のトラブルシューティングおよびバーチャル ポート チャネル（vPC）のペアリング プランに適した論理ノード ID 番号とノード名を割り当てます。たとえば、最初の 2 つのリーフのノード ID が 101/102、名前が leaf1/leaf2 など。

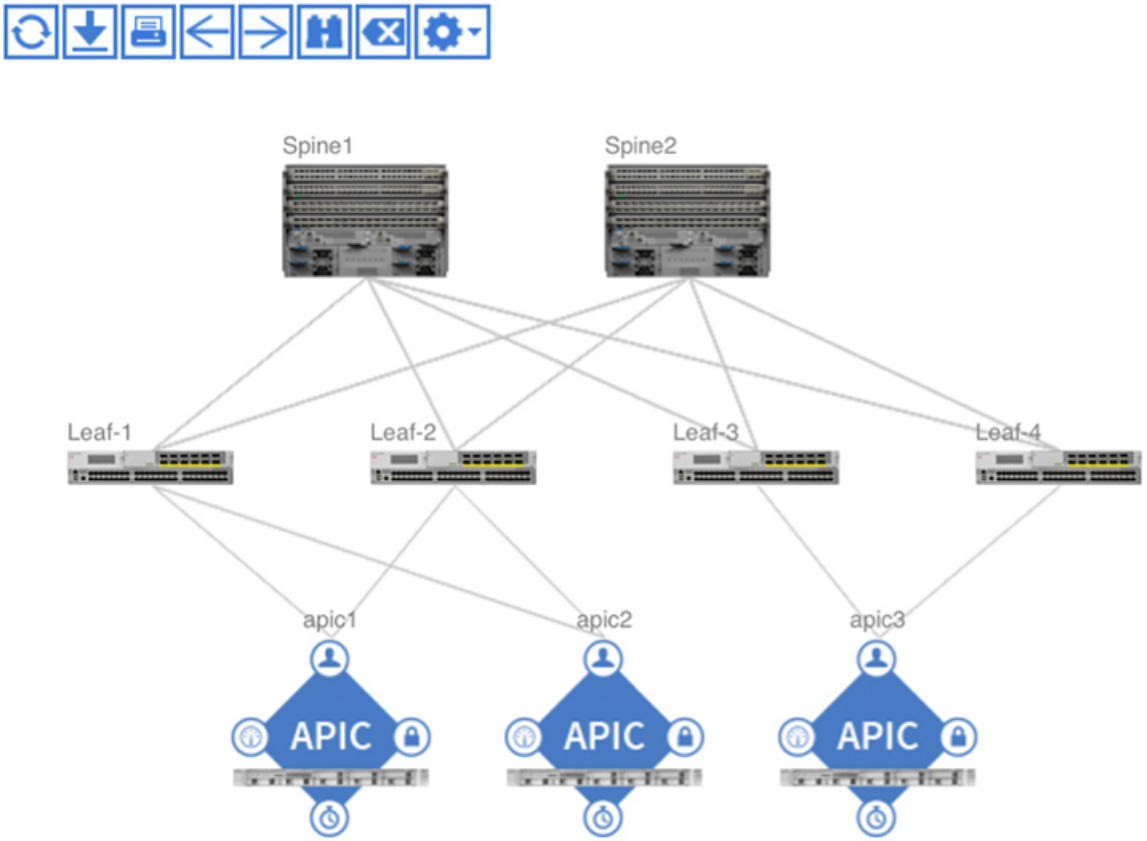
**ステップ 3** 最初のリーフが検出されると、システムはそのリーフ経由でスパイン スイッチを検出し、そのスパイン スイッチを使用して残りのリーフ スイッチを検出します。スパイン/リーフ ファブリックのレイアウトに従って、論理ノード ID 番号と名前を割り当てる追加ノードを登録します。

**ステップ 4** 想定どおりにトポロジが検出され、物理的に接続されていることを視覚的に確認し、次の操作を実行します。

a) メニュー バーで、[FABRIC] > [INVENTORY] を選択します。

b) [Navigation] ペインで [Topology] を選択します。

図 *2* : 検出されたスパイン/リーフ トポロジ




c) ファブリックが検出されたら、[Admin] > [Firmware] を選択し、すべての APIC とファブリック ノード（スイッチ）上で稼働しているファームウェア バージョンを検証します。必要に応じて、初期設定を開始する前に、最新バージョンまたは一貫したバージョンにアップグレードします。

# ホスト vPC の手動設定

ここでは、ホスト vPC を手動で設定する方法について説明します。

OpenStack サーバの vPC 接続を手動で設定することにより、ACI 管理者がファブリック アクセス ポリシー オブジェクトの命名法をより細かく制御できます。また、単一の物理 vPC を使用して、OpenStack のインストールに必要な複数のネットワーク タイプを伝送できます。

通常、OpenStack ホストは、管理/SSH およびテナントの ネットワーキングに個別の物理インターフェイスまたは論理インターフェイスを使用します。さらに、個別のインターフェイスは API、

ストレージ、プロバイダーネットワークなどの固有の目的に割り当てられることもあります。ACI OpenStack プラグインにより、Neutron 外部ネットワークに必要な要件が排除されます。これは、SNAT およびフローティング IP アクセスが ACI ファブリックおよび、OVS を使用する OpFlex によって処理されるためです。

この項の設定例では、802.1Q VLAN ヘッダーを使用して管理/SSH トラフィックおよびテナント トラフィック用の個別インターフェイスを準備する方法を示しています。これらの個別インターフェイスは、次の要素を使用して Linux オペレーティング システムに渡されます。

- 仮想インターフェイス カード（VIC ）1225 ネットワーク アダプタを搭載した Cisco UCS C シリーズ スタンドアロン サーバ。
- VLAN ヘッダーのアプリケーションを処理し、各 VLAN の個別の論理インターフェイスを Linux オペレーティング システムに渡すために、VIC アダプタで仮想ネットワーク インターフェイス カード（vNIC）が使用されます。
- 可用性の高いポート チャネル接続を OpenStack サーバ ノードに渡すために、ACI リーフ スイッチのペアで仮想ポート チャネル（vPC）が使用されます。
- ポート チャネル設定を作成し、ACI リーフ スイッチ ペアで設定されている vPC とメイティングするために、Linux オペレーティング システムで Bond インターフェイスが使用されます。

同様の設定は、Linux オペレーティング システム レベルで VLAN サブインターフェイス設定を使用することにより、基本的なデュアル ポート 10 ギガビット イーサネット アダプタでも実現できます。

### 手順

**ステップ 1** それぞれの OpenStack サーバの Cisco Integrated Management Controller（CIMC）インターフェイスにアクセスし、[Server] タブで次の操作を実行します。

a) [Inventory] を選択し、[Cisco VIC Adapters] タブに移動します。
b) [General] タブを選択した後、[Modify Adapter Properties] を選択します。
c) [Modify Adapter Properties] ダイアログボックスで、アダプタの [Enable FIP Mode] ボックスおよび [Enable LLDP] ボックスがオフであることを確認します。
d) [Save Changes] をクリックします。

**ステップ 2** アダプタ カードの [vNICs] タブを選択し、次の操作を実行します。

a) デフォルトでは、eth0 および eth1 という名前の 2 つの vNIC が一般的な Cisco VIC 上に存在し、トランク モードで動作します。これら 2 つの vNIC は個別の物理アップリンクに割り当てられています。特定の VLAN のトラフィックにタグを付ける vNIC を追加するには、[Properties] を選択します。
b) [Name] フィールドに、新規 vNIC の名前（eth4）を入力します。
c) [Uplink Port] フィールドに、アップリンク ポート（1）を入力します。
d) [VLAN Mode] フィールドで、ドロップダウン リストから [ACCESS] を選択します。
e) [Default VLAN] フィールドで、ラジオ ボタンを選択して、VLAN 番号（168）を入力します。

（注） アクセス モード vNIC は、タグなしの仮想 PCI インターフェイスをサーバのオペレーティング システムに渡し、スイッチング ファブリックのデフォルト VLAN のパケットにタグを付けます。

**ステップ 3** 設定を完了するには、アクセス モードの vNIC 2 つを ACI インフラストラクチャ（infra）VLAN 上に追加し、アクセス モードのインターフェイス 2 つを、管理/SSH トラフィックに使用される VLAN 上に追加します。この例では、使用中の ACI infra VLAN はデフォルト設定の 4093、管理/SSH は VLAN 168 です。

**ステップ 4** リブート後、サーバの Linux オペレーティング システムによって 6 つの仮想ネットワーク インターフェイスが検出されるようになります。この例では、これら 6 つのインターフェイスを使用して 3 つの Linux ボンド インターフェイスが作成されることにより、vPC アップリンク上で高可用性が提供されます。

- Main-bond：このインターフェイスは VIC 上でオリジナルの eth0/eth1 トランク モード vNIC から構築されており、LACP をアップストリーム スイッチ ペアに送信して vPC を起動するために使用されます。VLAN カプセル化が OpenStack サーバと ACI リーフ スイッチ間で使用されている場合、これが OpenStack テナント VLAN ネットワークのインターフェイスになります。
- Ten-bond：OpFlex の通信に使用される ACI infra VLAN を伝送する VIC 上で設定された 2 つの vNIC から構築されています。VXLAN カプセル化が OpenStack サーバと ACI リーフ スイッチ間で使用されている場合、このインターフェイスが OpenStack テナント VXLAN ネットワークの伝送に使用されます（このインターフェイスは、VLAN と VXLAN の両方のモード動作において ACI ファブリックへの OpFlex 通信の伝送に必要です）。
- Mgt-bond：管理/ssh VLAN を伝送する 2 つの vNIC から構築されます。管理トラフィックが ACI ファブリック上を伝送されている場合は、これがサーバの管理と更新および OpenStack のインストールに使用されるインターフェイスです。

**ステップ 5** OpenStack サーバをサポートし、main-bond インターフェイスからの LACP 通信に一致させるために必要な vPC 設定を手動でプロビジョニングします。サーバ vPCは、基本的なエンド ノード vPC 接続です。システムの設定作業の参照用に以下の手順を示します。
ファブリック アクセス ポリシーは、基盤となるスイッチ ファブリックのポリシー設定のグループを構成します。これらのポリシーは、後からテナントアプリケーションポリシーによって参照できます。次の操作を実行します。

a) APIC GUI のメニューバーで [FABRIC] > [ACCESS POLICIES] を選択します。
b) [Navigation] ペインで [Pools] を選択します。
c) 管理インターフェイスを備えた OpenStack サーバの初期設定に使用する VLAN プールを追加します。OpenStack ノードにマッピングされた VLAN を備え、静的に設定された EPG の場合、プールに [Static Allocation] を選択します。
d) メニュー バーで、[FABRIC] > [FABRIC POLICIES] を選択します。
e) [Navigation] ペインで、[Global Policies] を選択し、アタッチャブル アクセス エンティティ プロファイルを作成します。このプロファイルは、共通のアクセス要件を持つインターフェイスのグループについて説明します。名前を AEP に割り当て、[Enable Infrastructure VLAN] ボックスをオンにして、OpenStack サーバ ノードへの OpFlex 通信を有効化します。
f) メニュー バーで、[FABRIC] > [FABRIC POLICIES] を選択します。

g) [Navigation] ペインで、[Physical and External domains] を選択し、OpenStack ノード管理通信用の物理ドメインを作成します。VLAN プールおよび以前に作成した AEP に物理ドメインを関連付けます。

**ステップ 6** [Navigation] ペインで、[Interface Policies] > [Policies] を展開し、ポート チャネルの設定（LACP アクティブ）、CDP、LLCP の有効/無効ステータスをインターフェイス上で簡単に制御できるようにするための新しいインターフェイスを作成します。名前付きポリシー（CDP 有効または CDP 無効オプションを定義することにより、システム全体のデフォルトを更新しなくても、管理者が他の画面からこれらの設定を選択できるようになります。

**ステップ 7** [Navigation] ペインで、[Interface Policies] > [Policies] を展開し、OpenStack サーバを接続している TOR スイッチのノード プロファイルを作成します。この TOR スイッチは、リーフ スイッチのペア上で接続されているポート ID を参照しているインターフェイス セレクタを備えています。デュアルホーム サーバ接続には、各スイッチで同じポート番号が使用されます。

a) アクセス ポート セレクタのアイデンティティを指定するには、[Interface Policy Group] メニューをプルダウンして、[Create VPC Interface Policy Group] を選択します。CDP、LLDP、および以前に作成したポート チャネル ポリシーを活用し、AEP（OS-AEP-1）の特定も行います。
たとえば、インターフェイス プロファイルには、リーフ スイッチ 101 に接続する OpenStack コントローラおよび Compute-1 サーバに対して定義されたセレクタがあります。

**ステップ 8** [Navigation] ペインで、[Switch Policies] > [Profiles] を展開し、リーフ スイッチごとに新しいスイッチ プロファイルを作成します。

a) [Work] ペインで、[Blocks] からリーフ スイッチ番号を選択します。
b) 設定された最初のスイッチで、ドロップダウン リストから [Create Access Switch Policy Group] を選択し、OpenStack リーフ スイッチに使用される新規ポリシー グループの名前を割り当てます。
c) [Update] をクリックします。

**ステップ 9** プロファイルにスイッチを割り当てた後、[Next] を選択してアソシエーションの画面に移動し、以前にスイッチ用に作成したポート プロファイルを選択します。たとえば、[OS-PProfile-101] のような名前です。VPC ペアの 2 番目のリーフ スイッチに対してこのプロセスを繰り返します。

**ステップ 10** これで VPC に使用するポートおよびスイッチの定義が完了しましたが、VPC 関係自体の作成およびドメイン ID の割り当てが必要です（まだ VPC 関係が存在しない場合）。

a) APIC GUI のメニューバーで [FABRIC] > [ACCESS POLICIES] を選択します。
b) [Navigation] ペインで、[Quick Start] を選択します。
c) [Work] ペインで、[Configure an interface, PC, and VPC] を選択します。
d) VPC スイッチ ペアの下の [+] アイコンをクリックして、新しいペアリングを定義します。
e) vPC ドメインのドメイン ID を数字で入力し、VPC スイッチ ペアとして関連付ける 2 つのスイッチを選択します。
f) [Save] をクリックします。
たとえば、VPC ドメイン 10 の物理ドメインは、スイッチ 101 と 102 の間で作成され、現在は VLAN プールからサポートされる管理 VLAN のみを備えています。その後、ACI OpenStack プラグインの設定が完了したら、OpenStack VMM ドメインを AEP に関連付けます。すると、仮想ネットワーク カプセル化が vPC に追加されます。

**ステップ 11** vPC 設定が適切に実行されると、ポート チャネルの状態を APIC 上で確認できます。次の操作を実行します。

a) メニュー バーで、[FABRIC] > [INVENTORY] を選択します。

b) [Navigation] ペインで、[Pod] > [Leaf] > [Interfaces] > [VPC interfaces] > *[port channel assigned on the given leaf for the vPC]*を展開します。

c) [PROPERTIES] ペインで、ポート チャネルが [lacp-active]、[up]、[connected] であることを確認します。
ポートチャネルが起動していない場合は、物理接続を確認してください。また、LACP の Linux オペレーティング システム上で設定された main-bond インターフェイスが起動し、動作していることを確認してください。

（注） ポート チャネルに表示される VLAN 番号は、システム内部で使用される VLAN です。インターフェイス上のエンド ノード カプセル化に使用される VLAN タグではありません。

**ステップ 12** OpenStack サーバには、サーバの接続に必要な vPC ポート属性を特定するために ACI で設定された基本的なファブリック アクセス ポリシーが追加されています。サーバ管理目的のトラフィックは、ACI テナントでエンド ポイント グループ（EPG）を定義して、通信を許可するポリシーを定義するまで、ファブリックを流れることができません。APIC の [Tenants] で、OpenStack システムに使用される ACI テナントに EPG を追加してください。
このテナントが存在しない場合は、[Tenants] セクションの [Add Tenant] を選択してテナントを作成してください。

（注） このテナント名は、後述の OpenStack コントローラやネットワーク ノードのドライバ設定で定義する `apic_system_id` と一致する必要があります。

**ステップ 13** テナント内で [Application Profiles] を選択し、次の操作を実行します。

a) OpenStack 管理トラフィックのプロファイル名を追加します。

b) このプロファイル内で、実際の環境に該当するブリッジ ドメインを使用して EPG を作成します。
この設定例では、シンプルなレイヤ 2 ブリッジ ドメインを使用して、ACI ファブリックの外側に存在する管理トラフィックのデフォルト ゲートウェイに接続しています。レイヤ 3 ハードウェア プロキシおよびユニキャスト ルーティングは、このレイヤ 2 トランスポート ドメインに対して無効化されています。

（注） 実際の環境に該当する場合、OpenStack 管理トラフィックに関する別のオプションでレイヤ 3 対応ブリッジ ドメインや、後で ACI 契約によってレイヤ 3 外部ネットワーク接続にリンクされた EPG が使用されます。この接続を設計する方法は数多くありますが、この例では説明を簡単にするためにレイヤ 2 設定が使用されています。

**ステップ 14** EPG を作成した後、次の操作を実行します。

a) [Static Bindings] フォルダをハイライトし、[Actions] の [Deploy Static EPG] を選択して、VPC インターフェイスを EPG に追加します。この例では、コンピューティング ノード、コントローラ、外部レイヤ 2 TOR 接続を追加し、管理 VLAN 168 上ですべてがタグ付きモードで直接通信するようにします。

管理設定が適切に実行されると、Linux がインストールされた OpenStack サーバは、使用するシステムに該当する OpenStack ディストリビューションのインストール準備が整います。この設定例

では、OpenStack 機能の間での管理/SSH および API トラフィックに対して単一のインターフェイスが使用されています。この項では、Cisco VIC/vNIC のアプローチが示されています。さらに細かい操作も簡単に適用できます。たとえば、API インターフェイスの準備、別の VLAN タグを使用するストレージ インターフェイスなどの個別要件の追加が可能です。

# ホスト リンクの自動設定のセットアップ

Cisco ACI OpenStack プラグイン ソフトウェアには、OpenStack サーバのホスト リンクや vPC 接続を自動的にプロビジョニングする機能があります。自動設定は、OpenStack のインストールに使用される管理/SSH ネットワークがサーバ ノードの個別の物理ネットワーク インターフェイス上で維持される場合に非常に便利です。自動設定は、インストール済みの OpenStack システムと連携して動作することが必要です。通常、OpenStack はインストール時に、少なくとも管理/SSH/APIインターフェイスがすでに機能していることを必要とします。APIC でのポートの設定が完了していない場合、自動設定ではテナント ネットワーキングのインターフェイスのみが設定されます。

詳細については、『*Cisco Application Centric Infrastructure Fundamentals Guide*』を参照してください。

**手順**

**ステップ 1** ホスト リンクの自動プロビジョニングを有効にする設定は、`/etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini` ファイルにあります。インストールパッケージで提供されるテンプレートファイルには、コメント処理されたセクションがいくつかあり、ポート定義を設定するための構文が提供されています。`/etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini` ファイルを編集し、`apic_provision_hostlinks` 設定を `True` に変更します。

**ステップ 2** OpenStack ノードが接続されたリーフ スイッチごとに、設定ファイル `/etc/neutron/plugins/ml2/ml2_conf_cisco_apic.ini` のホスト定義ブロックを設定します。個々のポートに関するホストの構文は、以下の設定ファイル概要に従います。

**例**：

```
# Specify your network topology.
# This section indicates how your compute nodes are connected to the fabric's
# switches and ports. The format is as follows:
#
# [apic_switch:<swich_id_from_the_apic>]
# <compute_host>,<compute_host> = <switchport_the_host(s)_are_connected_to>
#
# You can have multiple sections, one for each switch in your fabric that is
# participating in Openstack.
# An example, note you can list more than one host name per physical port
# if your topology has virtual elements:

[apic_switch:18]
ubuntu6 = 1/1
ubuntu7,ubuntu8 = 1/2
```

または、サーバがポート チャネルから 2 つのリーフ スイッチまでのデュアルホーム接続の場合は、最初に VPC 用のスイッチのペアリングを含める必要があります。

例：
```
apic_vpc_pairs = 101:102,103:104
```

**ステップ 3** 次の例に従って、ポートの詳細を追加します。

例：
```
[apic_switch:101]
server1 = vpc-1-1/bundle-101-1-1-and-102-1-1
server2 = vpc-1-2/bundle-101-1-2-and-102-1-2
[apic_switch:102]
server1 = vpc-1-1/bundle-101-1-1-and-102-1-1
server2 = vpc-1-2/bundle-101-1-2-and-102-1-2
```

ここで、`bundle-101-1-1` はリーフ スイッチ 101 とポート ethernet1/1 を表しています。

**ステップ 4** これらの設定が `ml2_conf_cisco_apic.ini` ファイルに存在する場合、ACI OpenStack プラグインでは、neutron-server サービスが再起動するたびに、それらが APIC に適切に反映されることを確認します。ファイルのホストポートを手動で設定した場合は、ハイパーバイザ ノードの自動検出が、LLDP が提供する VMM ドメインに戻されます。LLDP のアプローチは、ノードの動的な検出が可能な優れた柔軟性を提供します。特にサーバ ホスト リンクがすでに APIC で定義されている場合に、ACI OpenStack のプラグインでプロビジョニングする必要がありません。

# ACI 外部ルーテッド ネットワークの例

ACI における外部ルーテッド ネットワーク（L3-Out）は、OpenStack で Neutron 外部ネットワークとして動作することが必要です。共有 L3-Out は、APIC の共通テナントの下で設定できます。または、プライベート L3-Out を OpenStack インスタンスに割り当てられた APIC テナント専用に追加することもできます。このルーテッド接続は、スタティック ルーティングを使用して設定できます。または、環境の要件に応じて、OSPF や BGP などのダイナミック プロトコルを使用して設定することもできます。

ACI には、L3-Out 用の多くの設定オプションがあります。以降の手順では、単一のインターフェイス上でスタティック ルートを設定する例を示します。このアプローチは、ACI による ACI OpenStack プラグインの試験展開に便利です。実稼働展開では、特定の ACI 環境のその他の部分と一貫性のある L3-Out 設定を使用する必要があります。

## はじめる前に

L3-Out に使用するインターフェイスは、APIC の [FABRIC] > [ACCESS POLICIES] で、基本的なポリシー グループを使用して設定されており、L3-Out に使用する前に、リーフ スイッチに関連付けられている必要があります。

### 手順

**ステップ 1** APIC の ACI 共通テナント、または APIC GUI で OpenStack に関連付けられた ACI テナントの下で、次のアクションを実行します。

a) [Navigation] ペインで、[Networking] > [External Routed Networks] > [Actions] を展開します。
b) [Create Routed Outside] を選択して、新しい L3-Out を追加します。
c) [Work] ペインに、L3-Out の名前を入力します（Example-L3-Out）。このプライマリ パネルがプライベート ネットワーク、ドメインの設定の起動、L3-Out へのノードおよびインターフェイスの追加に使用されます。
d) [Private Network] フィールドで、ドロップダウン リストから [Create Private Network] を選択し、プライベート ネットワークの名前を入力します。
e) [External Routed Domain] フィールドで、ドロップダウン リストから [Create Layer 3 Domain] を選択し、レイヤ 3 ドメインの名前を入力します。
L3-Out に使用するインターフェイスのセットアップ方法に応じて、既存の AEP をドメイン割り当てるか、新しい AEP を作成するかを選択できます。また、外部接続に VLAN タギングが必要とされる場合は、既存の VLAN プールを作成または選択することもできます。
f) プライベート ネットワークと外部ルーテッド ドメインを割り当てたら、ノード プロファイルを作成し、以下の操作を実行します。
g) [Nodes and Interface Protocol Profiles] セクションで、[+] アイコンをクリックします。
h) プロファイルの名前を入力します。この名前は、特定のノード（リーフ スイッチ）およびインターフェイスを L3-Out に関連付けるために使用されます。
i) [Nodes] フィールドで、[+] アイコンをクリックして ACI リーフ スイッチ ノードをプロファイルに追加します。この [Select Node] ペインは、L3-Out に使用されるリーフ スイッチの設定を定義するために使用されます。
j) [Select Node] ペインの [Node ID] フィールドで、ドロップダウン リストから ACI トポロジの有効なノード ID を選択します（topology/pod-1/node-101）。
k) [Router ID] フィールドは OSPF および BGP プロトコル通信に使用されます。これは、スタティック ルーテッド セットアップを選択するためのアドレスに設定できます。
l) [+] アイコンをクリックしてスタティック ルート内のルートを追加し、デフォルト ルート（0.0.0.0/0）にネクスト ホップ IP（192.168.100.1）を指定します。これは、リンク サブネット上の ACI ファブリック外の外部ルータに割り当てられたアドレスです。
m) [Interface Profiles] セクションで、[+] アイコンをクリックしてインターフェイスを追加します。
n) [Select Routed Interface] ペインでは、ルーテッド インターフェイスまたはルーテッド サブインターフェイス（VLAN タギングを使用中の場合）を設定できます。または、レイヤ 3 Out に vPC 接続が使用されている場合は、ルーテッド SVI を使用します。インターフェイスに割り当てられた IP アドレスとプレフィクスは、L3-Out のリンク サブネットの ACI ファブリック アドレスになります。ルーテッド インターフェイスを設定する場合は、以下の操作を実行します。

- [Path] フィールドにパスを入力します（topology/pod-1/paths-101/pathep-[eth1/12]）。
- [IP Address] フィールドに IP アドレスを入力します（192.168.100.2/24）。

- [Secondary IP Addresses] フィールドは空欄にしておきます。
- [MAC Address] フィールドに MAC アドレスを入力します（00:22:BD:F8:19:FF）。
- [MTU (bytes)] フィールドに MTU バイトを入力します（inherit）。
- [Target DSCP] フィールドに、ターゲット DSCP を入力します（未指定）。

（注） L3-Out に VPC 接続を使用している場合は、VPC ペアの各スイッチに個別の物理 IP アドレスを割り当てた後、共有のセカンダリ IP アドレスを両方に割り当てます。共有アドレスは、外部ルータからの着信トラフィックのスタティック ルート接続先として使用できます。

割り当てられたノードおよび特定のインターフェイスに対して完了した [Create Node Profile] ペインは次のようになります。

- [Name] フィールド：[Example-Leaf1]
- [Target DSCP] フィールド：[unspecified]
- [Nodes] フィールドの [Node ID]：[topology/pod-1/node-101]、[Static Routes]：[0.0.0.0/0]
- [INTERFACE PROFILES] セクションの [Name]：[Example 1-12]、[Interfaces]：[eith1/2]

これで [Create Routed Outside] ペインの [Identity] セクションが完了しました。

**ステップ 2** [Next] をクリックして [External EPG Networks] ペインに進み、次の操作を実行します。

a) [SUBNET] セクションで、[+] アイコンをクリックして外部ネットワークを追加します。

b) ACI OpenStack プラグイン設定ファイルで外部 EPG として参照されるネットワークの名前を入力し、サブネットとして [0.0.0.0/0] を入力します。

（注） 外部 EPG ネットワークは作成されると、APIC の [Networks] フォルダの [External Routed Network] に表示されます。[Networks] フォルダに表示されているときは、EPG として参照されません。

c) L3-Out の設定を完了するには [Finish] をクリックします。

**ステップ 3** OpenStack の APIC テナントの [External Routed Networks] フォルダで作成された構造体を使用して APIC の設定を確認できます。L3-Out の接続を確認するには、接続済みのリーフ スイッチの CLI から次のコマンドを使用する方法もあります。

**例**：

```
show vrf
show ip interface
iping -V <name of external vrf> <ip address of external router>
```

show vrf コマンドは、L3-Out のリーフ スイッチに追加された vrf を表示します。

show ip interface コマンドは、割り当てられた IP アドレスを伝達するインターフェイスを表示します。

また、リンク サブネット上の ACI に割り当てられた IP アドレスに、手順 1n でアドレスを割り当てた外部ルータからインバウンド ping を送信することもできます。

# ネットワーク制約テンプレート ファイル

次を参照して /etc/neutron/plugins/ml2/cisco_apic_network_constraints.ini ファイルを編集し、該当する値を選択してください。

```
[DEFAULT]

# The subnet scope to use on APIC if no other constraint
# has been explicitly specified. Valid values are
# public, private or deny.
# public -> Subnet will be advertised externally
# private -> Subnet is private to VRF
# deny -> Disallow creation of subnet
# subnet_scope = public|private|deny


# Tenant (project)-specific constraints and network-specific
# constraints are described in sections of their own.
#
# A tenant section looks like:
# [tenant-name]
# ...
#
# A network section looks like:
# [tenant-name/network-name]
# ...
#
# Network-specific constraints, when specified, take preference over
# tenant-specific constraints.

# Both sections may have the following configuration keys:
# deny -> Comma-separated list of CIDRs. If the requested
#         subnet overlaps with a deny CIDR, then creation of
#         the subnet is disallowed.
# private -> Comma-separated list of CIDRs. If the requested
#            subnet is contained within a private CIDR, then
#            the subnet will be created with 'private' scope
#            (i.e. private to the corresponding VRF).
# public -> Comma-separated list of CIDRs. If the requested
#            subnet is contained within a public CIDR, then
#            the subnet will be created with 'public' scope
#            (i.e. advertised externally).
# default -> The scope to use if the subnet does not match
#             any of the explicitly specified CIDRs. Valid
#             values are public, private or deny.
#
# When deciding subnet scope, the order of preference is deny,
# private, public. Thus if the requested subnet is present in
# both private and public CIDRs, the scope used will be private.
#
# Example:
#
# [tenant1/network1]
# public = 10.10.10.0/24, 10.10.20.0/28
# deny = 30.10.0.0/16
# default = private
#
# [tenant1]
# private = 50.50.50.0/26
# default = deny
```

# APIC OpenStack プラグインのトラブルシューティング

次のチェックリスト項目は、ACI OpenStack プラグインのインストールが正常に機能していない場合の問題の特定と修正に使用できます。

- OpFlex インターフェイスおよびサブインターフェイスが DHCP を取得しており、OpFlex 設定ファイルに記載されているエニーキャスト IP アドレスに ping を実行できることを確認します。
- `neutron-l3-agent` が Neutron サーバで無効化されていることを確認します。
- `Neutron-openvswitch-agent` がコンピューティング ノードで無効化されていることを確認します。
- APIC をチェックし、OpenStack 向けに作成された各 EPG に関連付けられた [Faults] タブに表示されている障害がないかを確認します。または、APIC でシステム全体の障害がないかをチェックします。
- コントローラ/Neutron サーバがリブートされると、`neutron-server` サービスが再起動することを確認します。リブート後に動作しない場合は、手動で再起動してください。
- `neutron-opflex-agent` および `agent-ovs` が各コンピューティング ノード上で動作していることを確認します。システム設定が変更されている場合は、これらの両方のサービスを再起動します。一元化されたメタデータまたは DHCP が使用中の場合、それらもコントローラに必要です。
- `ml2_conf_cisco_apic.ini` ファイル設定が変更されている場合、ファイルから新しい設定を読み込むために `neutron-server` サービスが再起動されていることを確認します。
- 提供された例に照らして `ml2_conf_cisco_apic.ini` ファイルの設定を確認します。欠落している項目がないことを確認します。
- VLAN および VXLAN ベースの設定に関する個別の例に照らして、`conf.d` ファイル内の `opflex-agent-ovs.conf` ファイルの追加項目の設定を確認します。ファイル内の括弧表記や書式設定が変更されていないことを確認します。
- OpenStack の VMM ドメインが APIC のサーバ ポートの AEP に関連付けられていることを確認します。
- Neutron サーバおよびその他の関連プロセスのロギングを `/var/log` ディレクトリの下でチェックします。
- エンドポイント ファイルが新しい VM インスタンスの `/var/lib/opflex-agent-ovs/endpoints` ディレクトリの下に作成されているかどうかを確認します。これらのファイルは、`neutron-opflex-agent` によってコンピューティング ノード上で作成される必要があります。
- サーバ上および APIC 上のサーバ アップリンクの物理インターフェイスの状態をチェックします。

- Modinfo openvswitch を実行することにより、正しい openvswitch カーネル モジュールがインストールされていることを確認します。バージョンは 2.4.1.gbp でなければなりません。
- VXLAN モードを使用している場合は、「ip routing」を使用してマルチキャスト ルートを確認します。たとえば、infra VLAN サブインターフェイスの場合は「224.0.0.0/4 dev bond0.4093」ようなエントリが含まれる必要があります。

# バージョン情報

このガイドで検証およびキャプチャされた設定例は、次のハードウェアおよびソフトウェアのバージョンを使用して作成されています。

- Ubuntu 14.04.4 上で稼働している Kilo OpenStack
- Cisco ACI/APIC バージョン 1.1(4e) および 11.1(4e).
- Cisco Nexus 9504 ACI スパイン スイッチ
- Cisco Nexus 9396PX ACI リーフ スイッチ
- OpenStack サーバ：UCS C220 M3S、Cisco VIC 1225