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, （16 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，（7 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），（7 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，（7 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，（8 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。

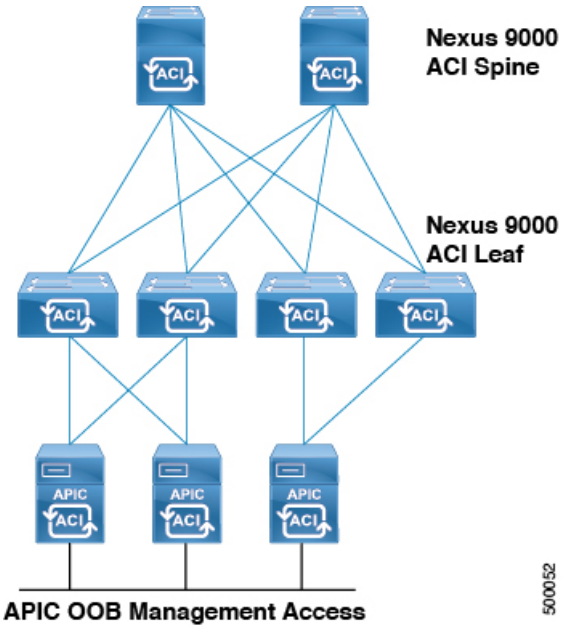
```
[ml2_cisco_apic]
per_tenant_context = False
shared_context_name=my_shared_context

[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。

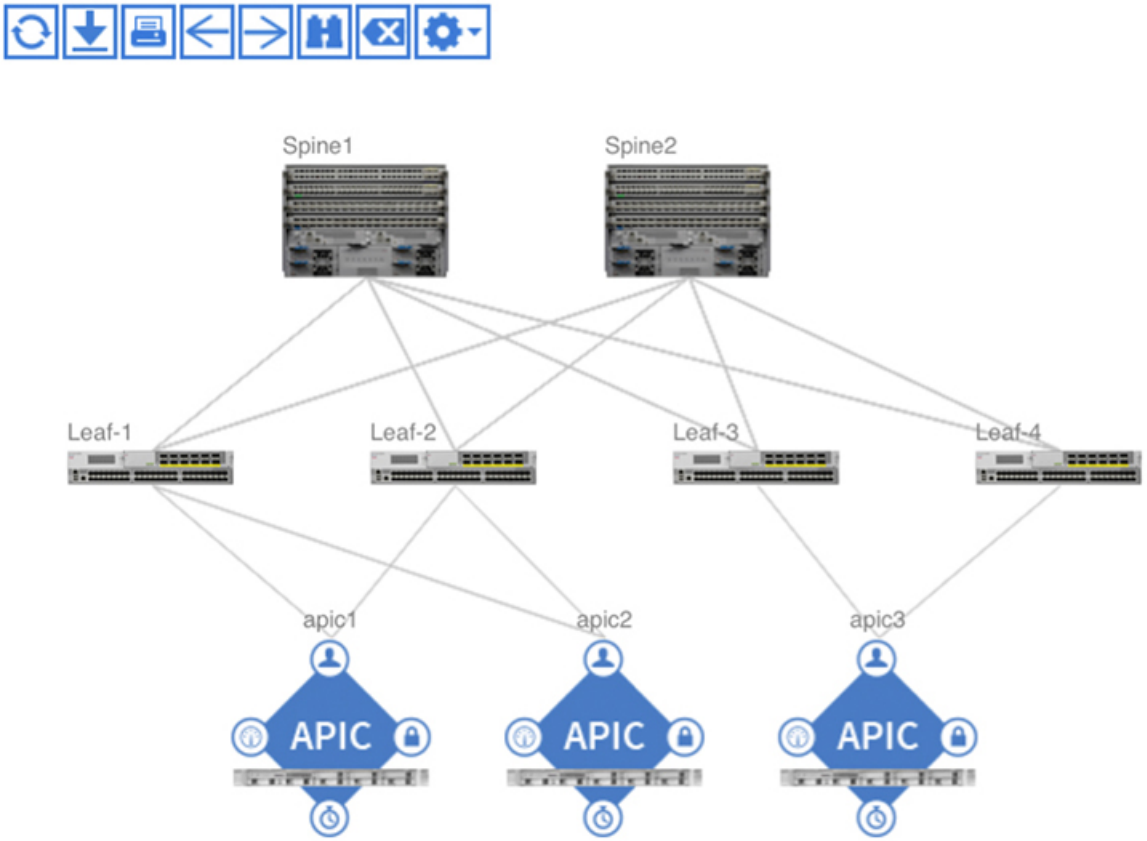
ステップ 3 最初のリーフが検出されると、システムはそのリーフ経由でスパイン スイッチを検出し、そのスパイン スイッチを使用して残りのリーフ スイッチを検出します。スパイン/リーフ ファブリックのレイアウトに従って、論理ノード ID 番号と名前を割り当てる追加ノードを登録します。

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