



Day 2 オペレーション

この章の内容は、次のとおりです。

- [データプレーン検証 \(1 ページ\)](#)

データ プレーン検証

データ プレーン検証の概要

この章では、OpenStack 向け Cisco ACI プラグインが正しくインストールされていることを検証する方法と、OpenStack インスタンスとそのデフォルトゲートウェイおよび外部で作成されたネットワークとの基本的な接続をテストする方法について説明します。

前提条件

開始する前に、次の前提条件を満たしていることを確認してください。

- コマンドを実行して、ネットワーク構造とインスタンスを作成するプロジェクト用の keystone ファイルを読み取ることを確認します。
- external-network-shared という 1 つの外部ネットワークが OpenStack プロジェクトの共有リソースとしてすでに作成されていることを確認します。
- インスタンスを作成するために、Nova でフレーバーが事前に定義されていることを確認します。
- インスタンスをブートするイメージが Glance に定義済みであることを確認します。

Neutron ネットワークの作成

ここでは、Neutron ネットワークを作成する方法について説明します。

手順

ステップ1 次のコマンドを入力して Neutron ネットワークを作成します。

```
$ openstack network create test_net
```

サンプル出力：

| Field | Value |
|--------------------------|--------------------------------------|
| admin_state_up | UP |
| availability_zone_hints | |
| availability_zone | |
| created_at | 2018-07-24T20:32:27z |
| description | |
| dns_domain | None |
| id | 016b9885-c8ac-4a2d-be7e-e5203c945ba4 |
| ipv4_address_scope | None |
| ipv6_address_scope | None |
| is_default | None |
| mtu | 1500 |
| name | test_net |
| port_security_enabled | True |
| project_id | 7d0be879a12c47ae9c0a26d3fd4407d1 |
| provider:network_type | opflex |
| provider:physical_type | physnet1 |
| provider:segmentation_id | None |
| qos_policy_id | None |
| revision_number | 3 |
| router:external | Internal |
| segments | None |
| shared | False |
| status | ACTIVE |
| subnets | |
| updated_at | 2018-07-24T20:32:27Z |

同様に ACI ファブリックでも、対応するテナントに Neutron ネットワーク名が付いた EPG と BD が定義されていることを確認できます。

ステップ2 Neutron ネットワーク名が付いた EPG と BD が定義されていることを確認します。

- APIC GUI のメニューバーで、**[Tenants] > [tenant_name] > [Application Profiles] > [OpenStack] > [Application EPGs] > [EPG_name]** を選択します。EPG が定義されているかどうかを確認します。
- APIC GUI のメニューバーで、**[Tenants] > [tenant_name] > [Networking] > [Bridge Domains] > [BD_name]** を選択します。BD が定義されているかどうかを確認します。

Neutron サブネットの作成

ここでは、Neutron サブネットを作成する方法について説明します。

手順

次の CLI コマンドを入力して Neutron サブネットを作成し、以前に作成したネットワークにバインドします。

例：

```
openstack subnet create --network test_net --gateway 192.168.1.254 \
--subnet-range 192.168.1.0/24 subnet01
```

サンプル出力：

| Field | Value |
|-------------------|--------------------------------------|
| allocation_pools | 192.168.1.1-192.168.1.253 |
| cidr | 192.168.1.0/24 |
| created_at | 2018-07-24T20:37:03Z |
| description | |
| dns_nameservers | |
| enable_dhcp | True |
| gateway_ip | 192.168.1.254 |
| host_routes | |
| id | d3341f6d-5fbe-476e-a0b7-d0e1b546eba4 |
| ip_version | 4 |
| ipv6_address_mode | None |
| ipv6_ra_mode | None |
| name | subnet01 |
| network_id | 016b9885-c8ac-4a2d-be7e-e5203c945ba4 |
| project_id | 7d0be879a12c47ae9c0a26d3fd4407d1 |
| revision_number | 2 |
| segment_id | None |
| service_types | |
| subnetpool_id | None |
| updated_at | 2018-07-24T20:37:03Z |

このコマンドによって ACI に変更が加えられることはありません。サブネットはどのルータにもまだ接続されておらず、L3 ルーティングでは有効になっていません。したがって、ACI ブリッジドメインにはまだサブネットが関連付けられていません。

Neutron ルータの作成

ここでは、Neutron ルータを作成する方法について説明します。

手順

ステップ 1 次の CLI コマンドを入力して Neutron ルータを作成します。

例：

```
openstack router create router01
```

サンプル出力：

| Field | Value |
|-------------------------|--------------------------------------|
| admin_state_up | UP |
| availability_zone_hints | None |
| availability_zone | None |
| created_at | 2018-07-24T20:44:11Z |
| description | |
| distributed | False |
| external_gateway_info | None |
| flavor_id | None |
| ha | False |
| id | 236734ab-c39e-4ad7-a9ab-c0d1fb03691a |
| name | router01 |
| project_id | 7d0be879a12c47ae9c0a26d3fd4407d1 |
| revision_number | None |
| routes | None |
| status | ACTIVE |
| updated_at | 2018-07-24T20:41:11Z |

このコマンドは、ACI 共通テナントに ACI 契約を作成します。実際には、OpenStack ルータは ACI で契約の「permit IP any any」タイプとしてレンダリングされます。契約は常に共通テナントに配置され、コンシューマおよびプロバイダーとして、サブネットがそのルータにバインドされている EPG で作成されたすべての Neutron ネットワークに適用されます。

ステップ 2 共通 ACI テナントに契約が作成されていることを確認します。

APIC GUI のメニューバーで、**[Tenants] > [common] > [Tenant Common] > [Contracts] > [Standards] > [router_name]** を選択します。ルータが定義されているかどうかを確認します。

ルータへのサブネットのバインディング

ここでは、サブネットをルータにバインドする方法について説明します。

手順

ステップ 1 次の CLI コマンドを入力して、作成した neutron ネットワーク上でルーティングを有効にします。

例：

```
openstack router add subnet router01 subnet01
```

この結果、APIC で DefaultRouterVRF という VRF が作成されます。この VRF に BD がバインドされ、BD サブネットとして Neutron サブネットも作成されます。

ステップ 2 DefaultRouterVRF という VRF が作成されて BD が VRF にバインドされ、BD サブネットとして Neutron サブネットも作成されていることを確認します。

a) APIC GUI のメニューバーで、**[Tenants] > [tenant_name] > [Networking] > [Bridge Domains] > BD 名 > [Subnets] > [subnet]** を選択します。サブネットが定義されているかどうかを確認します。

- b) APIC GUI のメニュー バーで、[Tenants] > [tenant_name] > [VRFs] > [DefaultRoutedVRF (DefaultVRF)] を選択します。DefaultRouterVRF が定義されているかどうかを確認します。

ルータへのゲートウェイの設定

ここでは、ルータにゲートウェイを設定する方法について説明します。

手順

- ステップ 1** OpenStack ドメインから外部ルータへの外部接続を実現するには、以前に作成した OpenStack ルータにゲートウェイを設定する必要があります。次のコマンドでは、external-net-shared として定義された外部ネットワークがすでに存在し、OpenStack プロジェクトで消費可能であることを前提としています。

例：

```
openstack router set --external-gateway external-net-shared router01
```

- ステップ 2** L3out が作成されていることを確認します。

APIC GUI のメニュー バーで、[Tenants] > [tenant_name] > [Networking] > [External Routed Networks] > [l3out1-DefaultVRF (l3out1-DefaultVRF)] を選択します。l3out1-DefaultVRF が定義されているかどうかを確認します。

インスタンスを作成して作成済みのネットワークに NIC を接続する

ここでは、インスタンスを作成し、以前に作成したネットワークにその NIC を接続する方法について説明します。

手順

- ステップ 1** ネットワークを作成して外部ルータにルーティングできるように設定したので、OpenStack インスタンスを作成して Neutron ネットワークに接続し、接続を確認できるようになりました。次の CLI コマンドを入力して Nova VM を作成します。

例：

```
NET1=$(openstack network list | awk '/test_net/ {print $2}')
```

```
nova boot --flavor ml.tiny --image cirros --nic net-id=$NET1 vm1
```

- ステップ 2** VM vm1 が [EPG test_net Operational] タブに表示されることを確認します。

APIC GUI のメニュー バーで、[Tenants] > [tenant_name] > [Application Profiles] > EPG 名 > [Application EPGs] > EPG を選択します。ペインにある [Operational] タブをクリックします。

ICMP を使用して VM がインフラストラクチャの他の部分に正しく接続していることを確認する

VM が表示されているかどうかを確認します。APIC によって IP アドレスが正しく検知される必要があります。

ICMP を使用して VM がインフラストラクチャの他の部分に正しく接続していることを確認する

ここでは、ICMP を使用して VM がインフラストラクチャの他の部分に正しく接続されていることを確認する方法について説明します。

手順

次の CLI コマンドを入力して、VM からそのデフォルト ゲートウェイおよび外部 IP への ICMP 接続に L3out 経由で到達可能であることを確認します。

例 :

```
$ ifconfig eth0
$ ping 192.168.1.254
```
