



Add an OpenStack External Network

- [Add an OpenStack External Network, on page 1](#)

Add an OpenStack External Network

This section describes how to add an OpenStack external network.



Note Execute the commands in this procedure sourcing the keystone file for the project where you want to create the network constructs and the instance.

Before you begin

You must have done the following before adding an OpenStack external network:

- Created a Layer 3 outside connection (L3Out) in Cisco Application Centric Infrastructure (ACI).
The L3Out can be in the OpenStack-created tenant (dedicated L3out for the OpenStack tenant) or in the Common tenant (Shared L3out across multiple OpenStack tenants). This procedure assumes that a dedicated L3out called *l3out1* is configured in the OpenStack tenant.
- Specified the following in the L3Out:
 - Interfaces and their IP address information.
 - Dynamic routing, if used.
 - An external endpoint group (EPG).
This procedure uses an external EPG named *extEpg*.



Note Do not add any contracts; the plug-in adds them automatically.



Important If you require Source Network Address Translation (SNAT) or a floating IP (FIP) address, you must define the L3Out in a different VRF from the one created by OpenStack.

Procedure

Step 1 Create the Neutron external network and provide the distinguished name of the L3Out.

Example:

```
neutron net-create network_name --router:external --apic:distinguished_names type=dict
ExternalNetwork=uni/tn-ACI_tenant_name/out-ACI_L3out_name/instP-ACI_externalEPG_name
(--apic:nat_type "")
```

--apic:nat_type "" is optional. Use it only if you do not use NAT for the specific external Neutron network.

The following shows an example of the creation of the external network with NAT enabled:

```
neutron net-create external-net-dedicated --router:external --apic:distinguished_names
type=dict ExternalNetwork=uni/tn-prj_$demo01/out-l3out1/instP-extEpg
Created a new network:
```

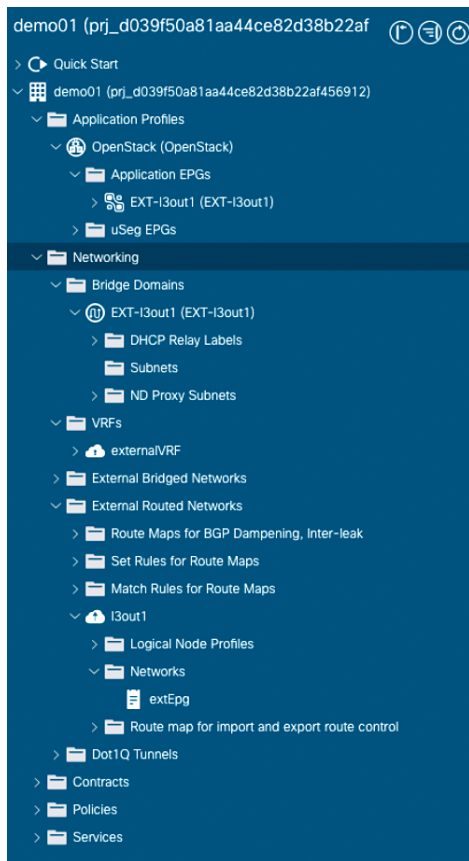
Field	Value
admin_state_up	True
apic:bgp_asn	0
apic:bgp_enable	False
apic:bgp_type	default_export
apic:distinguished_names	{"EndpointGroup": "uni/tn-prj_cdeda9c674a94394a09e86a2fea498c2/ap-OpenStack/epg-EXT-l3out1", "ExternalNetwork": "uni/tn-prj_cdeda9c674a94394a09e86a2fea498c2/out-l3out1/instP-extEpg", "VRF": "uni/tn-prj_cdeda9c674a94394a09e86a2fea498c2/ctx-externalVRF", "BridgeDomain": "uni/tn-prj_cdeda9c674a94394a09e86a2fea498c2/BD-EXT-l3out1"}
apic:external_cidrs	0.0.0.0/0
apic:nat_type	distributed
apic:nested_domain_allowed_vlans	
apic:nested_domain_infra_vlan	
apic:nested_domain_name	
apic:nested_domain_node_network_vlan	
apic:nested_domain_service_vlan	
apic:nested_domain_type	
apic:svi	False
apic:synchronization_state	build
availability_zone_hints	

```

|
| availability_zones                    |
| created_at                          | 2019-05-22T13:38:32Z
| description                          |
| id                                   | 635623ed-5dba-42ec-b3f8-3cff18f925c6
| ipv4_address_scope                  |
| ipv6_address_scope                  |
| is_default                           | False
| mtu                                  | 9000
| name                                 | external-net-dedicated
| port_security_enabled                | True
| project_id                           | cdeda9c674a94394a09e86a2fea498c2
| provider:network_type                | opflex
| provider:physical_network            | physnet1
| provider:segmentation_id             |
| revision_number                      | 6
| router:external                      | True
| shared                               | False
| status                               | ACTIVE
| subnets                             |
| tags                                  |
| tenant_id                            | cdeda9c674a94394a09e86a2fea498c2
| updated_at                           | 2019-05-22T13:38:33Z
+-----+

```

In Cisco ACI, the command creates a new EPG—*EXT-l3out1*— and a new bridge domain—*EXT-l3out1*, as shown in the following screen capture of the Cisco Application Policy Infrastructure Controller (APIC) GUI:



Step 2 Create a Neutron subnet that will be used for SNAT and the floating IP address.

This step is not required if you used `--apic:nat_type ""` when you created the Neutron external network (because NAT is disabled).

Example:

```
neutron subnet-create net_name subnet/mask --name subnet_name --disable-dhcp --gateway gateway_ip --apic:snat_host_pool True
```

The OpFlex agent automatically assigns one IP address for every compute node from the subnet. Virtual machines (VMs) connecting to the external use this IP address (one-to-many NAT) unless they have been assigned with a floating IP address.

The following shows an example of the creation of the external network with NAT enabled

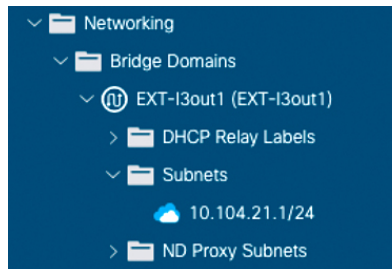
```
neutron subnet-create external-net-dedicated 10.104.21.0/24 --name ext-subnet --disable-dhcp --gateway 10.104.21.1 --apic:snat_host_pool True
```

Created a new subnet:

Field	Value
allocation_pools	{"start": "10.104.21.2", "end": "10.104.21.254"}
apic:distinguished_names	{}
apic:snat_host_pool	True
apic:synchronization_state	N/A
cidr	10.104.21.0/24
created_at	2019-05-22T13:38:35Z
description	
dns_nameservers	

enable_dhcp	False
gateway_ip	10.104.21.1
host_routes	
id	238aa55d-1537-4f01-86c9-5f6fc4bde625
ip_version	4
ipv6_address_mode	
ipv6_ra_mode	
name	ext-subnet
network_id	635623ed-5dba-42ec-b3f8-3cff18f925c6
project_id	cdeda9c674a94394a09e86a2fea498c2
revision_number	0
service_types	
subnetpool_id	
tags	
tenant_id	cdeda9c674a94394a09e86a2fea498c2
updated_at	2019-05-22T13:38:35Z

Creating a SNAT subnet generates a new subnet under the bridge domain, as shown in the following screen capture of the Cisco APIC GUI:



Step 3 (Optional) Assign one or more floating subnets to the external Neutron network:

Example:

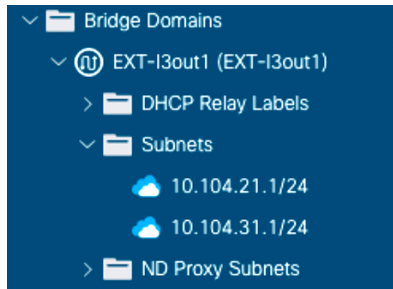
```
neutron subnet-create net_name fip_subnet/mask --name subnet_name --allocation-pool
start=start_ip,end=end_ip --disable-dhcp --gateway gateway_ip
```

The following output and screen capture in the Cisco APIC GUI show an example of the creation of a floating IP subnet:

```
neutron subnet-create external-net-dedicated 10.104.31.0/24 --name ext-subnet-FIP
--allocation-pool start=10.104.31.10,end=10.104.31.100 --disable-dhcp --gateway 10.104.31.1
Created a new subnet:
```

Field	Value
allocation_pools	{"start": "10.104.31.10", "end": "10.104.31.100"}
apic:distinguished_names	{}
apic:snat_host_pool	False
apic:synchronization_state	N/A
cidr	10.104.31.0/24
created_at	2019-05-22T13:38:38Z
description	
dns_nameservers	
enable_dhcp	False
gateway_ip	10.104.31.1
host_routes	
id	107c2714-2ace-44a7-9cb0-1a7f40ba2833
ip_version	4
ipv6_address_mode	
ipv6_ra_mode	
name	ext-subnet-FIP
network_id	635623ed-5dba-42ec-b3f8-3cff18f925c6

project_id	cdeda9c674a94394a09e86a2fea498c2
revision_number	0
service_types	
subnetpool_id	
tags	
tenant_id	cdeda9c674a94394a09e86a2fea498c2
updated_at	2019-05-22T13:38:38Z

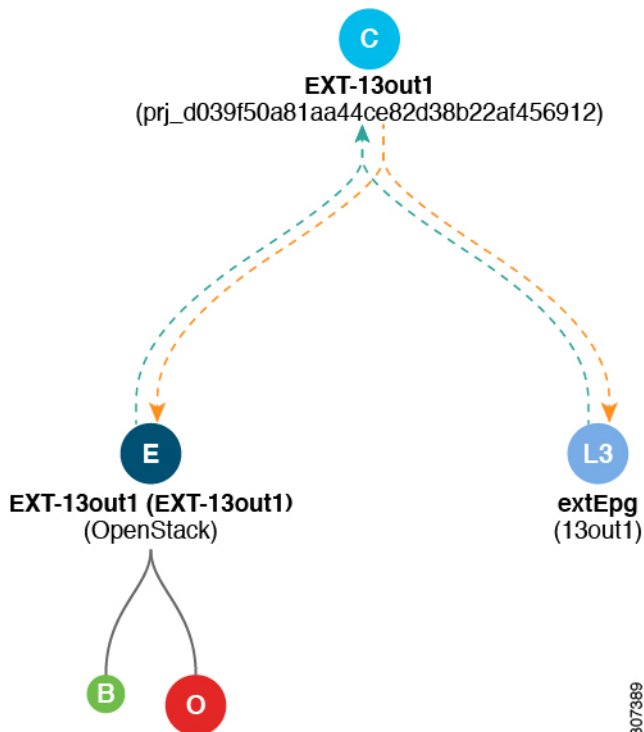


Step 4 Attach the Neutron external network to one OpenStack router as a gateway.

Example:

```
openstack router set --external-gateway external_net_name router_name
```

The command creates a contract that allows external connectivity for tenant networks attached to the OpenStack router of the external Neutron network, as shown in the following image:



What to do next