How to autoprovision a NetScaler VPX instance on OpenStack Nova

Introduction

The on-demand consumption model has become a de facto standard in cloud computing. To support this model in the OpenStack cloud environment, NetScaler MAS supports on-demand autoprovisioning of NetScaler VPX instances on OpenStack Nova. The provisioned instances are used to load balance (including SSL and content switching functionality) the applications that are deployed in OpenStack clouds or outside OpenStack Cloud.

For an example, if a cloud service provider has built a cloud using OpenStack and providing customers with on-demand computing, storage, and networking services, the networking services can include on-demand allocation of NetScaler instances to customers who are deploying applications in or outside the OpenStack Cloud. Using NetScalers VPX instances that are provisioned on OpenStack Nova leverages the existing OpenStack infrastructure.

Preconfiguration

Before autoprovisioning a NetScaler VPX instance on OpenStack Nova, verify that the following prerequisites have been met:

1. NetScaler LBaaS v2 drivers are installed in OpenStack.
2. NetScaler MAS is installed and registered with OpenStack.
3. The VPX image has been uploaded to OpenStack Glance.
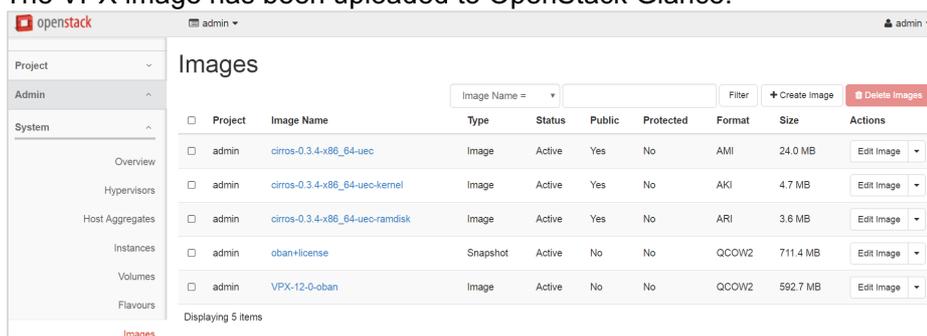
4. In MAS, navigate to **Orchestration** > **OpenStack** > **Deployment** and select the management network of the OpenStack stack. Enter the license activation code and select the VPX image that was uploaded to OpenStack Glance to be used for autoprovisioning.

Configuration

1. Log on to MAS, **Go to Orchestration, Cloud Orchestration, Service Packages**, and click **Add** to create a new service package.
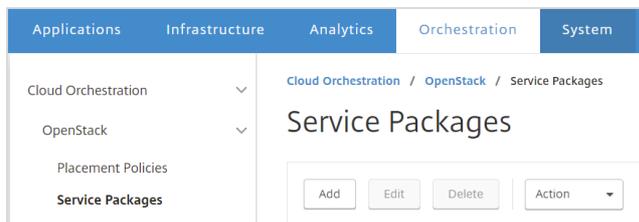
2. Enter the name of the package, set **NetScaler Instance Allocation** to either `Dedicated` or `Partition`, and set **NetScaler Instance Provisioning** to `Create Instance OnDemand`. Also set **Auto Provision Platform** to `OpenStack Compute`.

3. Enter the Maximum number of instances to Auto-provision, and select the flavour and capacity of the instances from the **Flavor** drop-down list. If the autoprovisioned VPX instances are to be a high availability pair, select the **High Availability** option. Click **Continue**, and select the **install license** option if LAC code is available.

4. Under Assign placement policies/OpenStack Tenants, select **OpenStack Tenants** and click **Add** to populate the list of tenants.

5. For each tenant, click the plus (**+**) icon and assign the tenant to a service package. Click **Continue**, and then **Done** to finish the creation of a service package.

Assign Placement Policies/OpenStack Tenants

◉ OpenStack Tenants ○ Placement Policies

Available (1)	Select All
demo	+

Configured (0)	Remove All
No items	

Continue   Cancel

Assign Placement Policies/OpenStack Tenants

◉ OpenStack Tenants ○ Placement Policies

Available (0)	Select All
No items	

Configured (1)	Remove All
demo	−

Continue   Cancel

↩ Service Package

Service Level Agreement

Name	Auto-provision Nova VPX		NetScaler Instance Allocation	dedicated
			NetScaler Instance Type	**NetScaler VPX**
			Platform Type	**OpenStack Compute**

Auto Provision Settings   ✎

Flavor	Model	Provision pair of NetScaler appliances for high availabilty.
m1.medium, 2 vcpus, 4096 RAM	Pre-Licensed	false

Assign Tenant   ✎

◉ OpenStack Tenants ○ Placement Policies

Configured (1)	Remove All
demo	− ⇕

Done

6. The tenants assigned to the service package will try to consume NetScaler load balancing as a service from OpenStack. Before any LBaaS commands can be executed, you have to set the environment variables. For each tenant, open an SSH session and enter the following commands:

export OS_USERNAME=<username>
export OS_PASSWORD=<password>
export OS_TENANT_NAME=<projectName>
export OS_AUTH_URL=https://<KeystoneIP:portNumber>**/v2.0**

```
root@ubuntu:~# export OS_USERNAME=
root@ubuntu:~# export OS_PASSWORD=
root@ubuntu:~# export OS_TENANT_NAME=
root@ubuntu:~# export OS_AUTH_URL=http://                    /v2.0
root@ubuntu:~#
```

7: Enter the following command to create a load balancer:

$neutron lbaas-loadbalancer-create --name <loadbalancer-name> <subnet-name> **--provider netscaler**

```
root@ubuntu:~# neutron lbaas-loadbalancer-create --name demolb  Sub1-demo --provider netscaler
Created a new loadbalancer:
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| admin_state_up      | True                                 |
| description         |                                      |
| id                  | 3effd95b-9c7a-46c5-95cf-007c2ecf4a3d |
| listeners           |                                      |
| name                | demolb                               |
| operating_status    | OFFLINE                              |
| pools               |                                      |
| provider            | netscaler                            |
| provisioning_status | PENDING_CREATE                       |
| tenant_id           | 97dcf754ad494897b55296c50ab8ad18     |
| vip_address         | 30.0.0.23                            |
| vip_port_id         | 4c942f77-3a30-42a8-a6f0-b6b6917333e0 |
| vip_subnet_id       | a66c991d-db7c-4ddb-bbdb-a9bbd53642d2 |
+---------------------+--------------------------------------+
```

8. Enter the following command to check the provisioning status of the newly created load balancer

$neutron lbaas-loadbalancer-list

```
root@ubuntu:~# neutron lbaas-loadbalancer-list
+--------------------------------------+--------+-------------+---------------------+-----------+
| id                                   | name   | vip_address | provisioning_status | provider  |
+--------------------------------------+--------+-------------+---------------------+-----------+
| 3effd95b-9c7a-46c5-95cf-007c2ecf4a3d | demolb | 30.0.0.23   | ACTIVE              | netscaler |
+--------------------------------------+--------+-------------+---------------------+-----------+
```

Note: You can verify the status of the load balancer at each stage, after the creation of listener, pool, or members.

9. Enter the following command to create a listener:

$neutron lbaas-listener-create -loadbalancer <loadbalancer-name> **--name** <listener-name> **--protocol** <protocol_type> **-protocol-port** <port_number>

```
root@ubuntu:~# neutron lbaas-listener-create --loadbalancer demolb --name demolistener --protocol http --protocol-port
80
Created a new listener:
+------------------------+-------------------------------------------------+
| Field                  | Value                                           |
+------------------------+-------------------------------------------------+
| admin_state_up         | True                                            |
| connection_limit       | -1                                              |
| default_pool_id        |                                                 |
| default_tls_container_ref |                                              |
| description            |                                                 |
| id                     | bda03133-29ab-45a1-abfb-3ea52bf23b66            |
| loadbalancers          | {"id": "3effd95b-9c7a-46c5-95cf-007c2ecf4a3d"}  |
| name                   | demolistener                                    |
| protocol               | HTTP                                            |
| protocol_port          | 80                                              |
| sni_container_refs     |                                                 |
| tenant_id              | 97dcf754ad494897b55296c50ab8ad18                |
+------------------------+-------------------------------------------------+
```

10. Enter the following command to create a pool:

$neutron lbaas-pool-create --lb-algorithm <algorithm_type> **--listener** <listener-name> **--protocol** <protocol_type> **--name** <pool-name>

```
root@ubuntu:~# neutron lbaas-pool-create --lb-algorithm LEAST_CONNECTIONS --listener demolistener --protocol http --nam
e demopool
Created a new pool:
+---------------------+---------------------------------------------------+
| Field               | Value                                             |
+---------------------+---------------------------------------------------+
| admin_state_up      | True                                              |
| description         |                                                   |
| healthmonitor_id    |                                                   |
| id                  | d78b40eb-9d5e-4531-9c32-81600a321b85              |
| lb_algorithm        | LEAST_CONNECTIONS                                 |
| listeners           | {"id": "d7153e9e-225f-46b0-bad9-a7cdf92e99e5"}    |
| loadbalancers       | {"id": "44a6b466-851b-4ff3-947c-b9ede069c631"}    |
| members             |                                                   |
| name                | demopool                                          |
| protocol            | HTTP                                              |
| session_persistence |                                                   |
| tenant_id           | 97dcf754ad494897b55296c50ab8ad18                  |
+---------------------+---------------------------------------------------+
```

11. Add members to the pool. To add a member, enter the following command:

$neutron lbaas-member-create --subnet <subnet-name> **--address** <ip-address of the web server> **--protocol-port** <port_number> <pool-name>

```
root@ubuntu:~# neutron lbaas-member-create  --subnet Sub1-admin --address 10.0.0.18 --protocol-port 80 demopool
Created a new member:
+----------------+--------------------------------------+
| Field          | Value                                |
+----------------+--------------------------------------+
| address        | 10.0.0.18                            |
| admin_state_up | True                                 |
| id             | 1fc18cfd-0a5d-45da-834c-7d04f738d056 |
| name           |                                      |
| protocol_port  | 80                                   |
| subnet_id      | d6a449bf-1410-4f71-a126-788555e0cdf6 |
| tenant_id      | 97dcf754ad494897b55296c50ab8ad18     |
| weight         | 1                                    |
+----------------+--------------------------------------+
```

```
root@ubuntu:~# neutron lbaas-member-create  --subnet Sub1-admin --address 10.0.0.19 --protocol-port 80 demopool
Created a new member:
+----------------+--------------------------------------+
| Field          | Value                                |
+----------------+--------------------------------------+
| address        | 10.0.0.19                            |
| admin_state_up | True                                 |
| id             | 86170dac-1ff3-4c64-ae33-d3c92e4249a5 |
| name           |                                      |
| protocol_port  | 80                                   |
| subnet_id      | d6a449bf-1410-4f71-a126-788555e0cdf6 |
| tenant_id      | 97dcf754ad494897b55296c50ab8ad18     |
| weight         | 1                                    |
+----------------+--------------------------------------+
```

12. Log on to the NetScaler GUI and go to **Infrastructure** > **Instances** > **NetScaler VPX** to see the newly created VPX instance and the service package to which it is bound.

Scroll bar to left to see the service package information.

14. Go to **Applications** > **Load Balancing** and verify the newly created virtual servers, service groups, and content switching virtual servers on the VPX instance.

Once the NetScaler VPX is created and completely configured, it is ready to serve the traffic.