How to autoprovision a NetScaler VPX on SDX for load balancing OpenStack workloads

Introduction

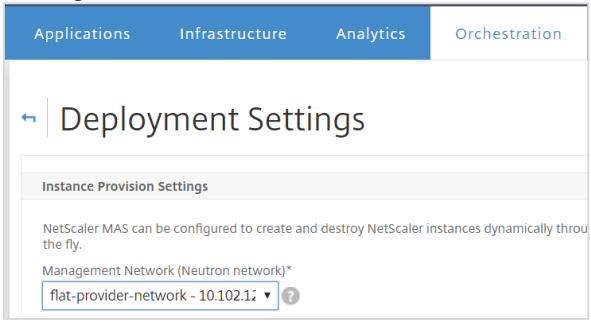
The on demand consumption model has become a de facto standard in cloud computing. To support this model in the OpenStack cloud environment, NetScaler MAS supports autoprovisioning of NetScaler VPX instances on a NetScaler SDX appliance to load balance applications deployed in OpenStack clouds.

For example, if a cloud service provider has built a cloud using OpenStack and serving their customers by providing on demand computing, storage and networking services, the networking services can include allocation of NetScaler instance on demand to customers who are deploying applications in the cloud. With NetScaler VPX instances provisioned on NetScaler SDX, customers can get the benefits of NetScaler MPX, such as high scale and performance, and the benefits of NetScaler VPX, such as flexibility.

Preconfiguration

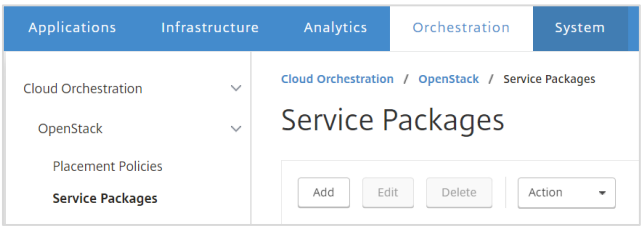
Before autoprovisioning a NetScaler VPX instance on NetScaler SDX, verify that the following prerequisites have been met:

1. NetScaler LBaaS v2 drivers are installed in OpenStack.
2. NetScaler MAS is installed and registered with OpenStack.
3. NetScaler SDX has been added to MAS.
4. In MAS, navigate to **Orchestration** > **OpenStack** > **Deployment** and select the management network to which NetScaler SDX is connected.

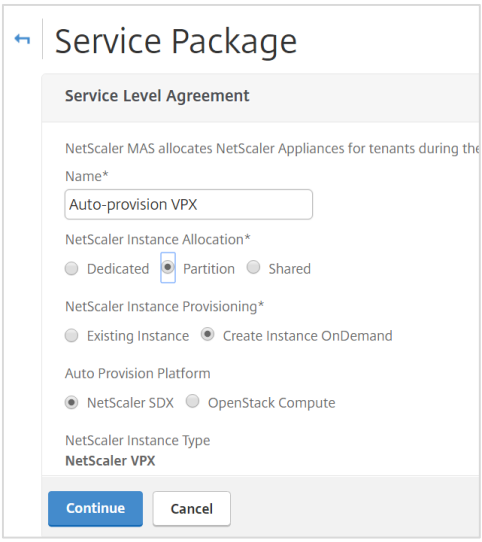


Configuration Steps:

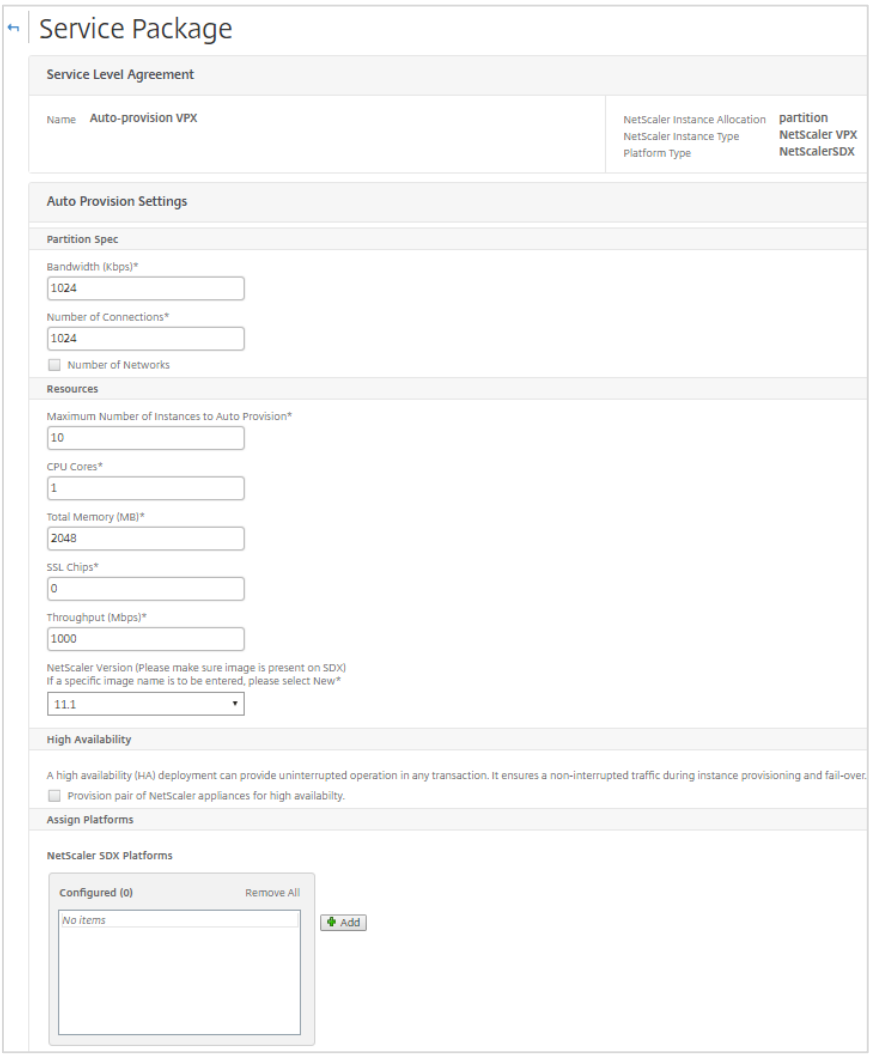
1. Log on to MAS, go to **Orchestration** > **Cloud Orchestration** > **Service Packages**, and click **Add** to create a new Service Package.



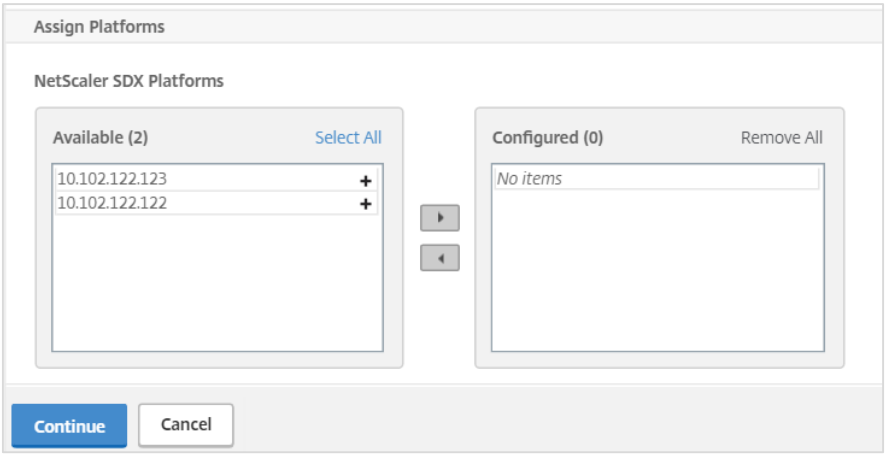
2. Enter the name of the package, and set **NetScaler Instance Allocation** to **`Dedicated`** or **`Partition`**, **NetScaler Instance Provisioning** to **`create instance OnDemand`**, and **Auto-provision platform** to **`NetScaler SDX`**.



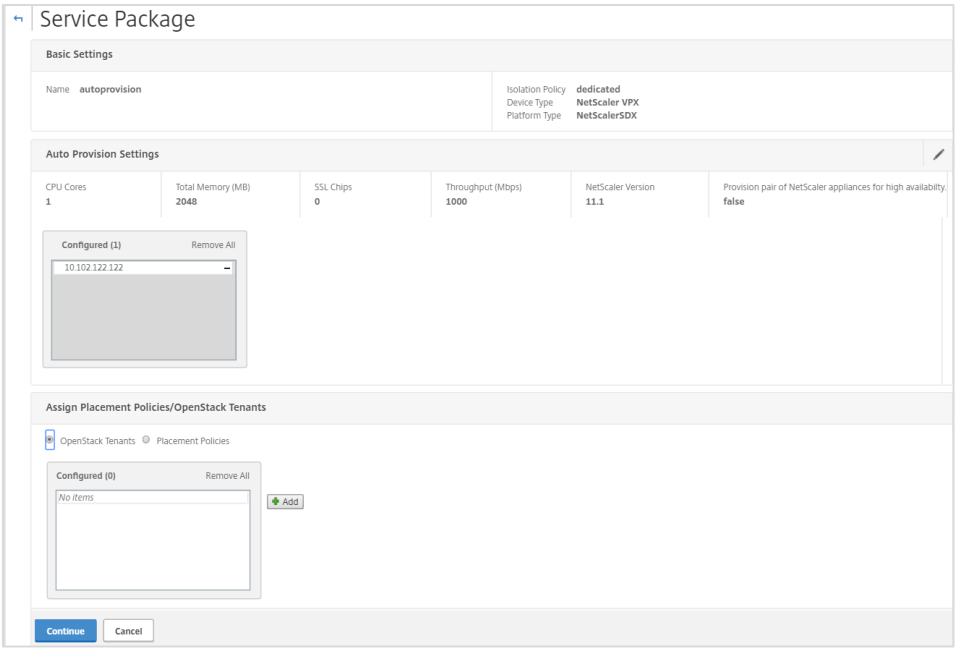
3. Enter the values for **Bandwidth**, **Number of connections**, **Maximum number of instances to Auto provision**, **CPU cores per instance**, **total Memory per instance**, **SSL chips per instance**, **throughput required**, and **NetScaler VPX version**.



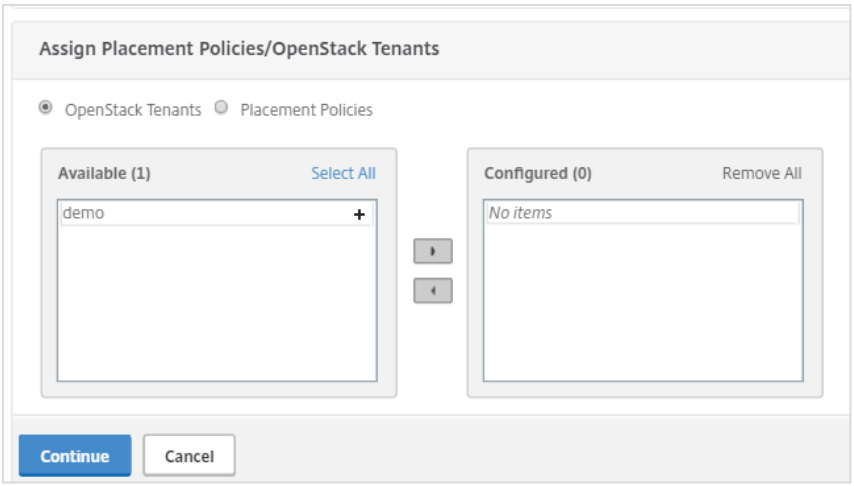
4. Click the **Add** button to populate the available SDX devices. Then, click the plus (**+**) icon next to each IP address, to add that NetScaler SDX appliance to the service package, and then click **Continue**.

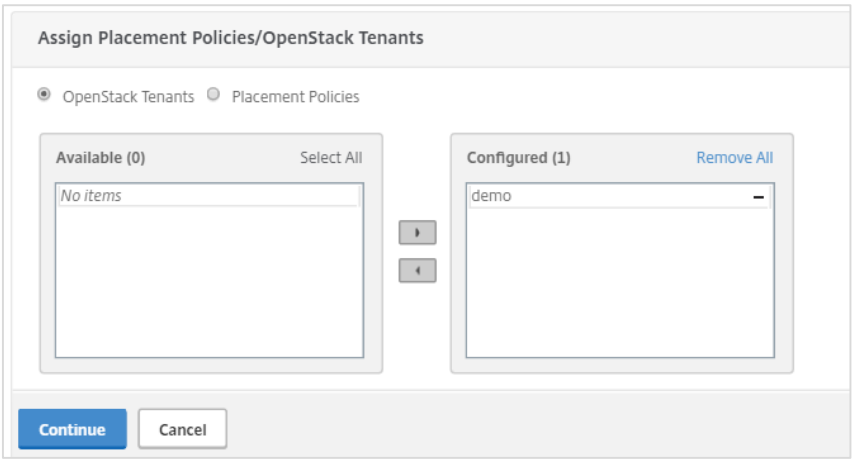


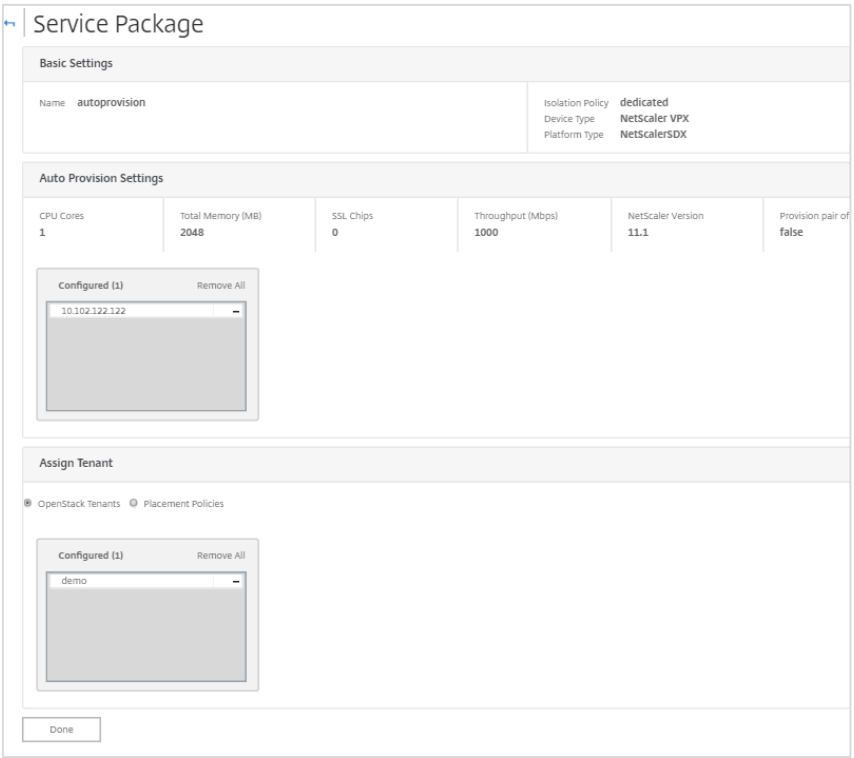
5. Under Assign Placement Policies/OpenStack Tenants, select **OpenStack tenants** and click **Add** to populate the list of tenants



6. For each tenant, click the plus (**+**) icon and assign the tenant to a service package. Click **Continue**, and then click **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
Basic Settings
Name autoprovision
Isolation Policy dedicated
Device Type NetScaler VPX
Platform Type NetScalerSDX
Auto Provision Settings
CPU Cores 1
Total Memory (MB) 2048
SSL Chips 0
Throughput (Mbps) 1000
NetScaler Version 11.1
Provision pair of false
Configured (1)
Remove All
10.102.122.122
Assign Tenant
OpenStack Tenants
Placement Policies
Configured (1)
Remove All
demo
Done

7. 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=[redacted]
root@ubuntu:~# export OS_PASSWORD=[redacted]
root@ubuntu:~# export OS_TENANT_NAME=[redacted]
root@ubuntu:~# export OS_AUTH_URL=http://[redacted]/v2.0
root@ubuntu:~#
```

8. 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 |
+---------------------+--------------------------------------+
```

9. 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.

10. 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               |
+---------------------------+------------------------------------------------+
```

11. 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               |
+---------------------+------------------------------------------------+
```

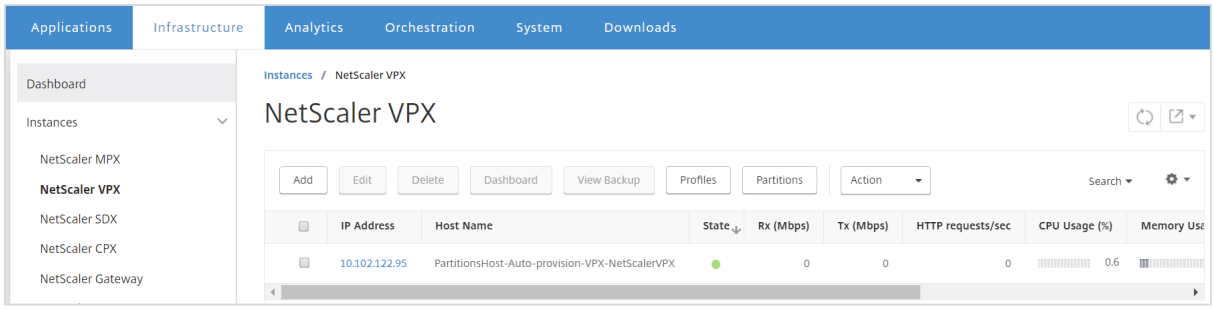
12. Add members to the pool. To add a member, enter the following command:

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

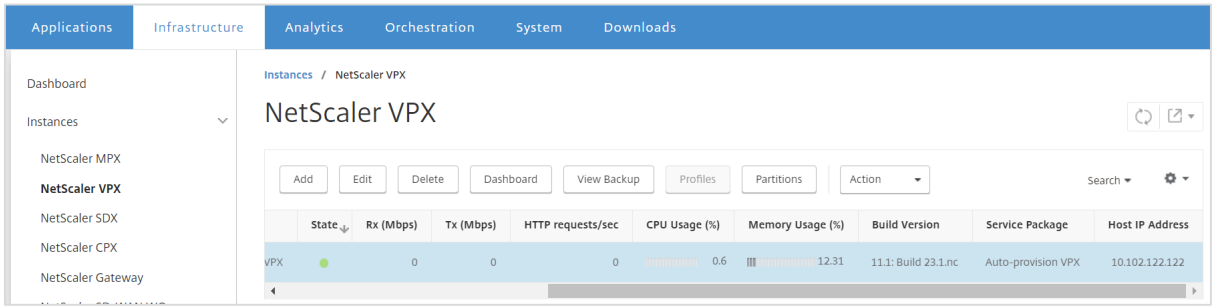
```
root@ubuntu:~# neutron lbaas-member-create  --subnet Sub1-admin --address 10.0.0.18 --protocol-port 80 demopool
Created a new member:
+----------------+--------------------------------------+
| Field          | Value                                |
+----------------+--------------------------------------+
| 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                                    |
+----------------+--------------------------------------+
```

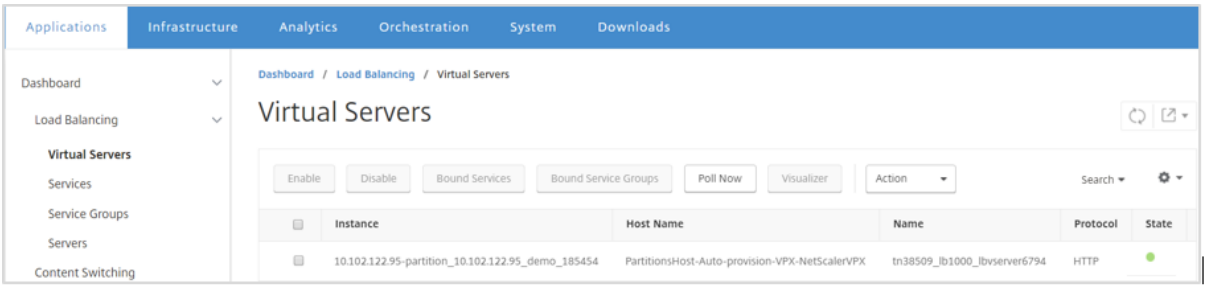
13. Log on to the NetScaler GUI and go to Infrastructure > Instances > NetScaler VPX to see the newly created VPX instance and the service package to which it is bound

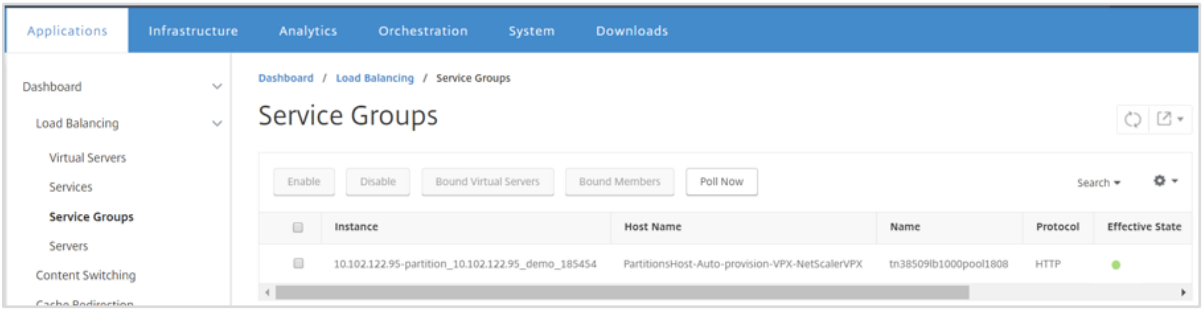


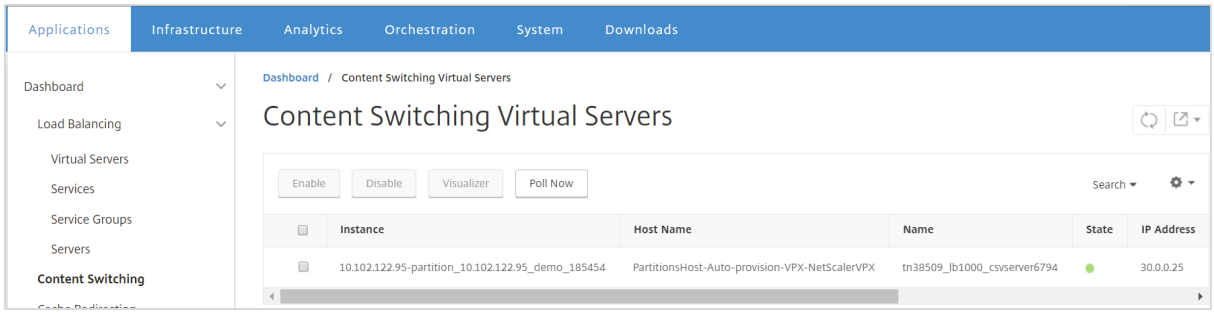
Scroll bar to left to see the service package information.



14. Go to **Applications, Load Balancing** and verify the newly created virtual servers, Service Groups and Content Switching Servers on the VPX instance.







Once the VPX is created and all the configurations are done, it is ready to serve the traffic.