INSTALLATION RUNBOOK FOR NetScaler Management & Analytics System with Mirantis OpenStack 9.0

Application Type:	**[NetScaler Management Platform]**
Application Version:	**[NetScaler MAS 11.1]**
MOS Version:	**[9.0]**
OpenStack version:	**[Mitaka]**

Content

Document History
1 Introduction
    1.1 Target Audience
2 Application overview
3 Joint Reference Architecture
4 Physical & Logical Network Topology
5 Installation & Configuration
    5.1 Environment preparation
    5.2 MOS installation
        5.2.1Health Check Results
    5.3 <Application name> installation steps
    5.4 Limitations
    5.5 Testing
        5.5.1 Integration Test cases and test results
        5.5.2 Target Use cases

Document History

Version	Revision Date	Description
1.0	**04-10-2016**	**Initial Version**

2.0	**13-10-2016**	**Reworked**
3.0	**19-10-2016**	**Reworked**

1 Introduction

This document is to serve as a detailed Deployment Guide for NetScaler Management & Analytics System. Citrix offers NetScaler Management & Analytics System. This document describes the reference architecture, installation steps for certified MOS+ NetScaler Management & Analytics System, limitations and testing procedures.

1.1 Target Audience

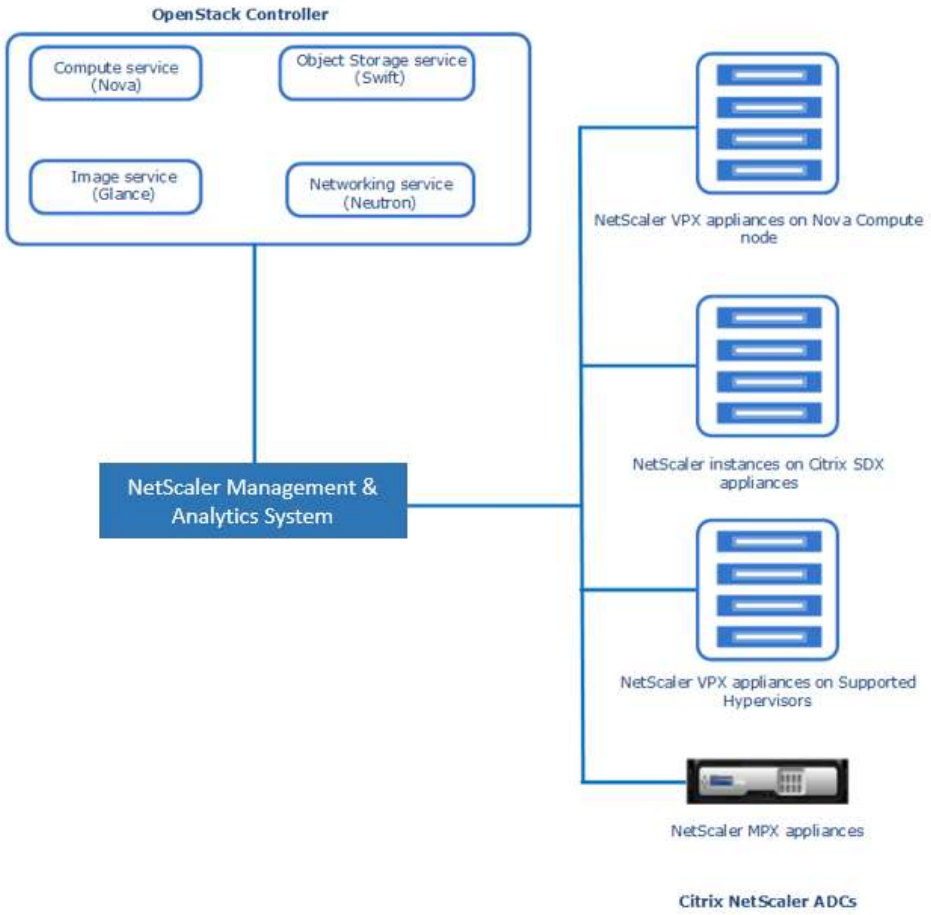
Customers interested in deploying Citrix NetScaler products (release 10.5 and later) with Mirantis OpenStack 9.0 (Mitaka release and later)

2 Application overview

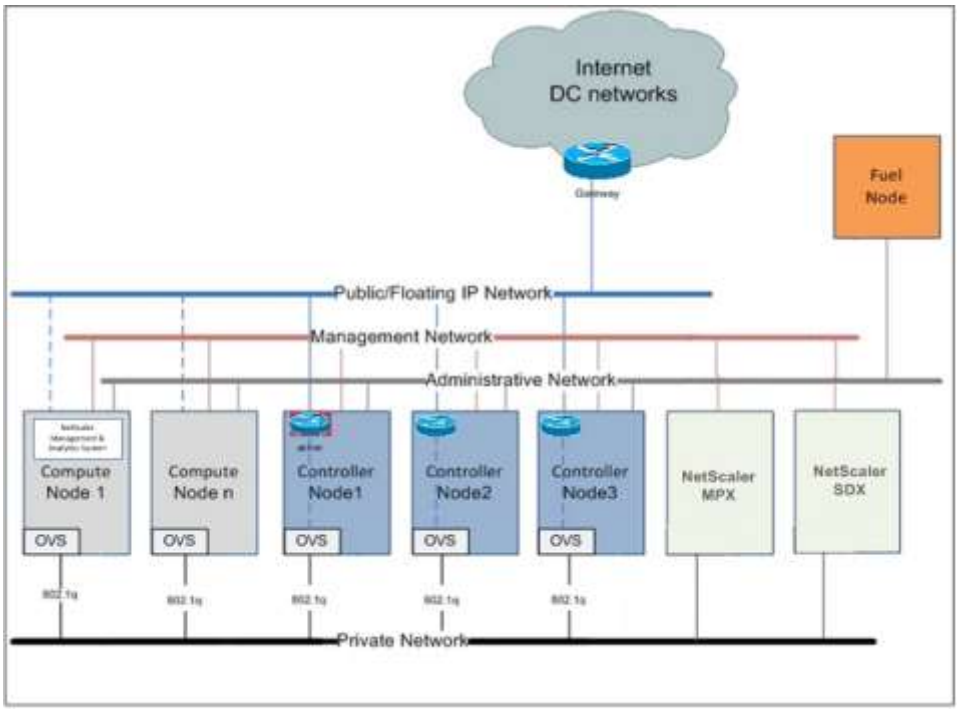
NetScaler Management and Analytics System (MAS) is a centralized management solution that simplifies operations by providing administrators with enterprise-wide visibility and automating management jobs that need to be executed across multiple NetScaler instances. You can manage and monitor Citrix application networking products that include Citrix NetScaler MPX, Citrix NetScaler VPX, Citrix NetScaler Gateway, Citrix NetScaler SDX, Citrix NetScaler CPX, and Citrix NetScaler SD-WAN. You can use NetScaler MAS to manage, monitor, and troubleshoot the entire global application delivery infrastructure from a single, unified console.

The Cloud Orchestration feature of NetScaler Management and Analytics System (MAS) enables integration of Citrix NetScaler products with Mirantis OpenStack platform. By using this feature with Mirantis OpenStack platform, the Mirantis OpenStack users are able to avail the load balancing feature (LBaaS) of the NetScaler. After this, the Mirantis OpenStack users can deploy their load balancer configurations from Mirantis OpenStack in NetScaler instance.

3 Joint Reference Architecture



4 Physical & Logical Network Topology



5 Installation & Configuration

5.1 Environment preparation

Hardware Required:

- 3 bare metal servers with at least 4 core CPU, 48GB RAM, 1 TB HDD for controller nodes
- 2 bare metal servers with at least 12 core CPU, 96GB RAM, 1 TB HDD for compute nodes
- 1 bare metal server with at least 2 core CPU, 3 GB RAM, 350 GB HDD for Fuel node
- NetScaler MPX
- NetScaler SDX

Software Required:

- Mirantis OpenStack 9.0 ISO file
- Neutron LBaaS package from https://github.com/openstack/neutron-lbaas.git
- NetScaler Management & Analytics System 11.1
- NetScaler Software version 10.5 and above

5.2 MOS installation

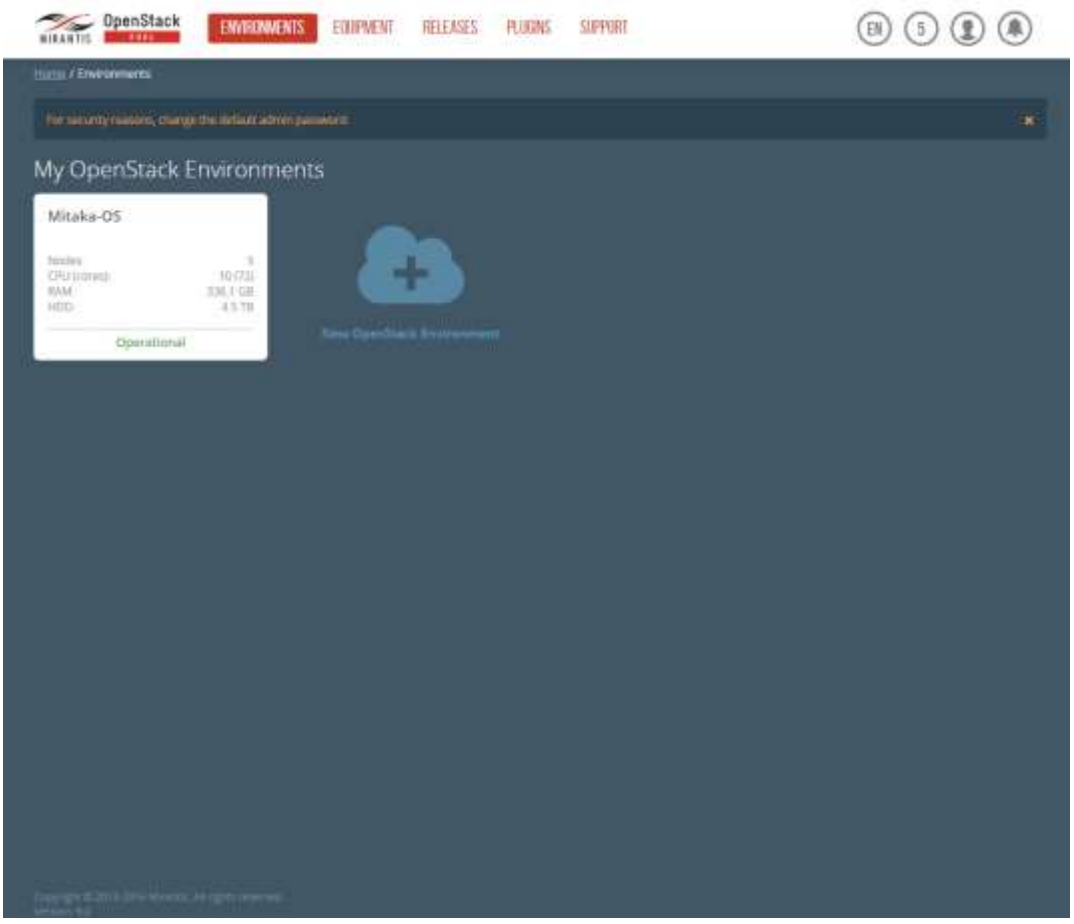
Fuel server is used to deploy and manage the OpenStack environment. Fuel acts as a DHCP server. The OpenStack nodes are configured to network boot, using PXE. It assigns IP addresses to the OpenStack nodes, performs PXE boot and initial configuration, and provisions the nodes according to their roles in the environment. In the testing environment, Fuel has

deployed Mirantis OpenStack with an operating system based on Ubuntu Linux. Mirantis OpenStack uses the MySQL/Galera for database replication in HA deployments that use the Ubuntu kernel.

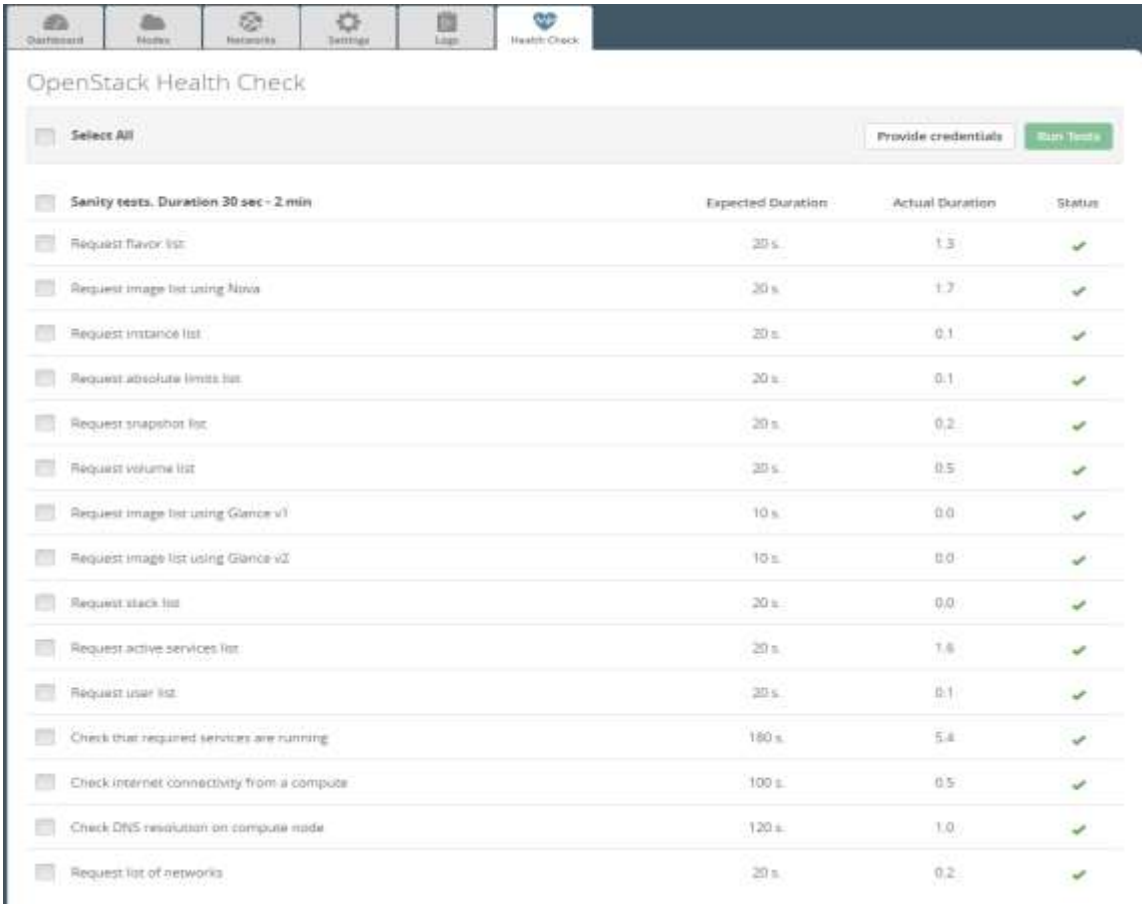
For more information about Mirantis OpenStack Installation, see https://docs.mirantis.com/openstack/fuel/fuel-9.0/

http://docs.openstack.org/developer/fuel-docs/userdocs/fuel-install-guide.html

Creating OpenStack Environment: http://docs.openstack.org/developer/fuel-docs/userdocs/fuel-user-guide/create-environment.html



5.2.1Health Check Results



OpenStack Health Check

Select All | Provide credentials | Run Tests

Sanity tests. Duration 30 sec - 2 min	Expected Duration	Actual Duration	Status
Request flavor list	20 s.	1.3	✔
Request image list using Nova	20 s.	1.7	✔
Request instance list	20 s.	0.1	✔
Request absolute limits list	20 s.	0.1	✔
Request snapshot list	20 s.	0.2	✔
Request volume list	20 s.	0.5	✔
Request image list using Glance v1	10 s.	0.0	✔
Request image list using Glance v2	10 s.	0.0	✔
Request stack list	20 s.	0.0	✔
Request active services list	20 s.	1.6	✔
Request user list	20 s.	0.1	✔
Check that required services are running	180 s.	5.4	✔
Check internet connectivity from a compute	100 s.	0.5	✔
Check DNS resolution on compute node	120 s.	1.0	✔
Request list of networks	20 s.	0.2	✔

Functional tests. Duration 3 min - 14 min	Expected Duration	Actual Duration	Status
Create instance flavor	30 s.	0.8	✔
Check create, update and delete image actions using Glance v2	70 s.	5.5	✔
Create volume and boot instance from it	350 s.	66.3	✔
Create volume and attach it to instance	350 s.	73.8	✔
Check network connectivity from instance via floating IP	300 s.	41.4	✔
Create keypair	25 s.	0.4	✔
Create security group	25 s.	1.0	✔
Check network parameters	50 s.	0.2	✔
Launch instance	200 s.	25.0	✔
Launch instance with file injection	200 s.	31.5	✔
Launch instance, create snapshot, launch instance from snapshot	300 s.	63.4	✔
Create user and authenticate with it	80 s.	5.3	✔

HA tests. Duration 30 sec - 8 min	Expected Duration	Actual Duration	Status
Check state of haproxy backends on controllers	10 s.	1.3	✔
Check data replication over mysql	10 s.	4.7	✔
Check if amount of tables in databases is the same on each node	10 s.	3.8	✔
Check galera environment state	10 s.	1.5	✔
Check pacemaker status	10 s.	2.0	✔
RabbitMQ availability	100 s.	13.9	✔
RabbitMQ replication	100 s.	27.9	✔

Platform services functional tests. Duration 3 min - 60 min	Expected Duration	Actual Duration	Status
Typical stack actions: create, delete, show details, etc.	720 s.	47.2	✔
Advanced stack actions: suspend, resume and check	900 s.	78.8	✔
Check stack rollback	470 s.	101.3	✔
Update stack actions: inplace, replace and update whole template	1300 s.	103.9	✔
Check creation of stack with Wait Condition/Handle resources Time limit exceeded while waiting for stack status becoming "CREATE_COMPLETE" to finish. Please refer to OpenStack logs for more details. Target component: Heat Scenario: 1. Create test flavor. 2. Create a keypair. 3. Save generated private key to file on Controller node. 4. Create a stack using template. **5. Wait for the stack status to change to 'CREATE_COMPLETE'.** 6. Delete the file with private key. 7. Delete the stack. 8. Wait for the stack to be deleted.	820 s.	620.9	✖

Cloud validation tests. Duration 30 sec - 2 min	Expected Duration	Actual Duration	Status
Check disk space outage on controller and compute nodes	20 s.	1.8	✔
Check log rotation configuration on all nodes	20 s.	1.9	✔

Configuration tests. Duration 30 sec - 2 min	Expected Duration	Actual Duration	Status
Check usage of default credentials on master node	20 s.	—	—
Check if default credentials for OpenStack cluster have changed	20 s.	—	—
Check usage of default credentials for keystone on master node	20 s.	—	—

5.3 NetScaler MAS installation steps

NetScaler MAS can be downloaded from
https://www.citrix.com/downloads/netscaler-mas.html
Please download the build 50.XX

For NetScaler MAS installation please refer
http://docs.citrix.com/en-us/netscaler-mas/11-1/single-server-deployment.html
NetScaler MAS can be deployed on any of the supported hypervisor that are listed in the link above

To register NetScaler MOS with MAS please refer to the below link.
http://docs.citrix.com/en-us/netscaler-mas/11-1/integrating-netscaler-mas-with-openstack-platform/preconfiguration-tasks-mas-openstack.html#par_anchortitle_b14e

Note: For MAS builds earlier to 50.XX please follow the below work around for registering MOS with MAS.
Using any of the API tools, please fire the below API.

POST call with URL: http://<MAS IP>/oca/v1/openstacks

Copy the below payload as body:

```
{
        "openstack": {
                "name": "Openstack",
                "username": "<OpenStack Admin Username>",
                "password": "<password>",
                "admin_tenant_name": "<Openstack Tenant name>",
                "driver_username": "openstack_driver",
                "driver_password": "<Driver Password>",
                "keystone_uri": "<Keystone IP:port>/",
                "keystone_admin_uri": "<Keystone admin IP:port>",
                "neutron_uri": "<Neutron IP:port>",
                "glance_uri": "<Glance IP:port>",
                "nova_uri": "<Nova IP:port>"
        }
}
```

Payload content type: JSON

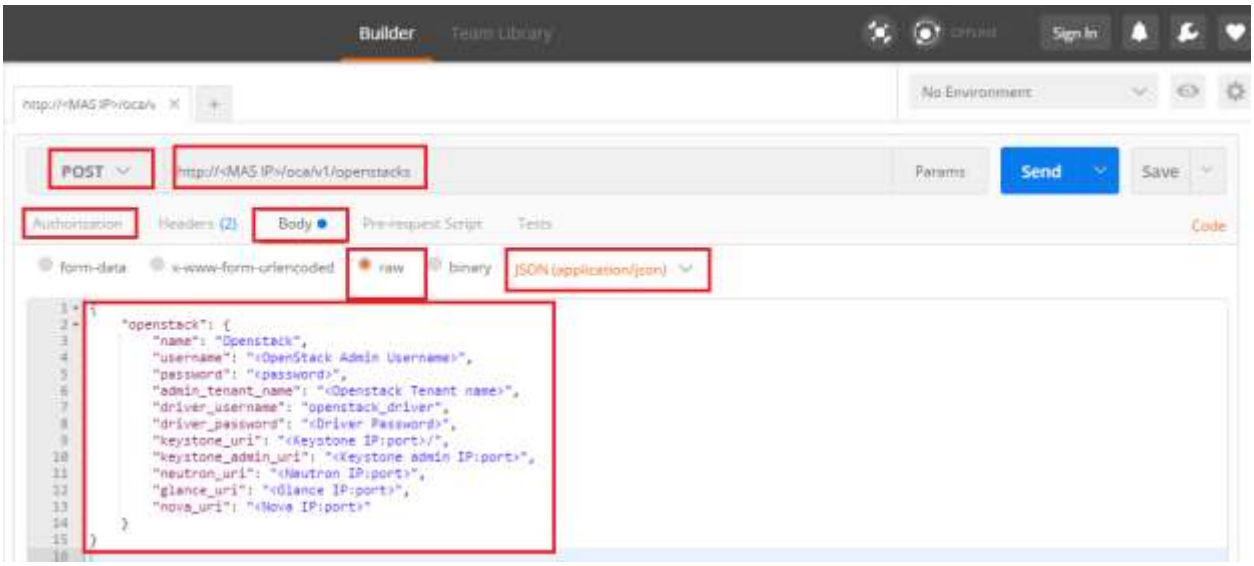
Authentication type: Basic Auth

For example Postman REST client can be used to fire the API.

Postman download link:

https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?hl=en

Configurations in Postman



5.4 Limitations

http://docs.citrix.com/en-us/netscaler-mas/11-1/Before-You-Begin.html
http://docs.citrix.com/en-us/netscaler-mas/11-1/single-server-deployment/install-mas-on-xenserver.html

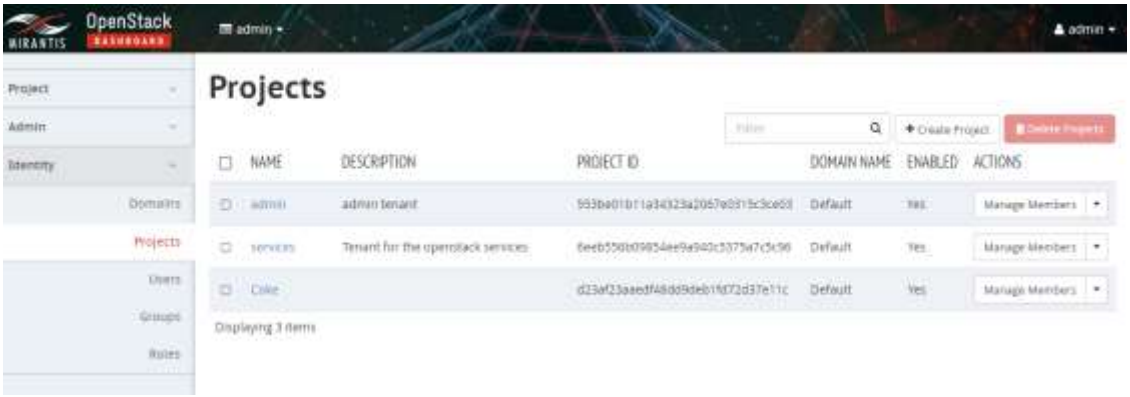
5.5 Testing

5.5.1 Integration Test cases and test results

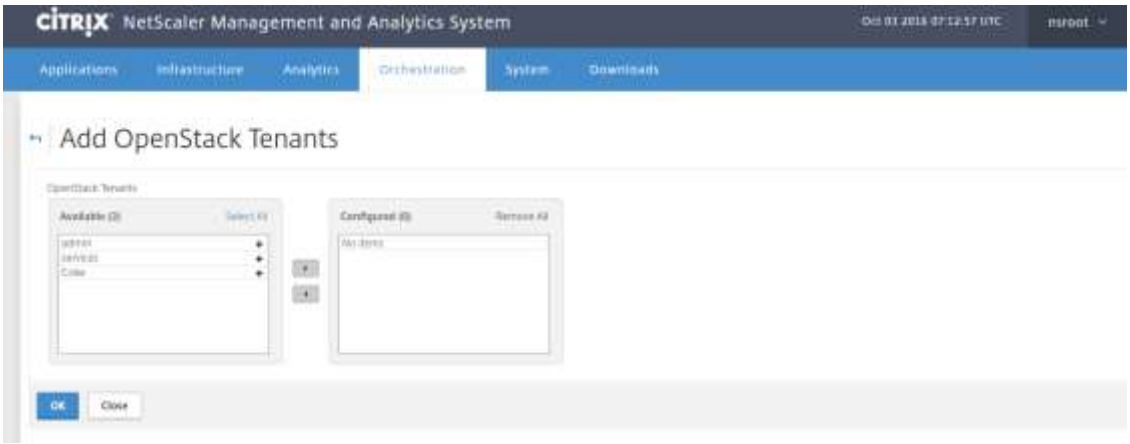
Verify that NetScaler MAS is able to get the Tenants, Images, Networks and Flavors information configured in OpenStack. In NetScaler MAS only Flavors applicable to NetScaler Control Center (vCPU equal to 2 or more) will be displayed. (Screenshots are below)

1. Compare the tenants present in MOS and MAS.

Tenants present in OpenStack.

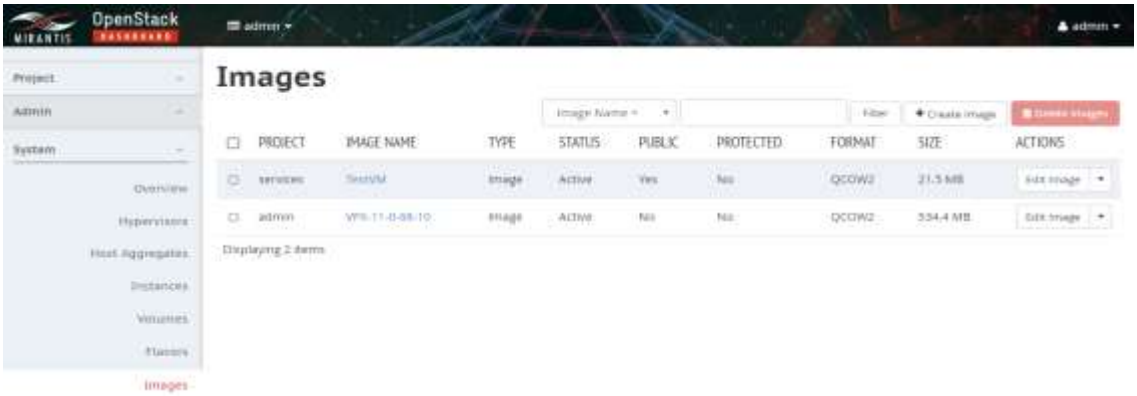


MAS fetches the Tenants from OpenStack.
Note: MAS can be accessed by typing the IP address of the NetScaler MAS server (IP address provided during MAS installation) in the address bar of the browser.

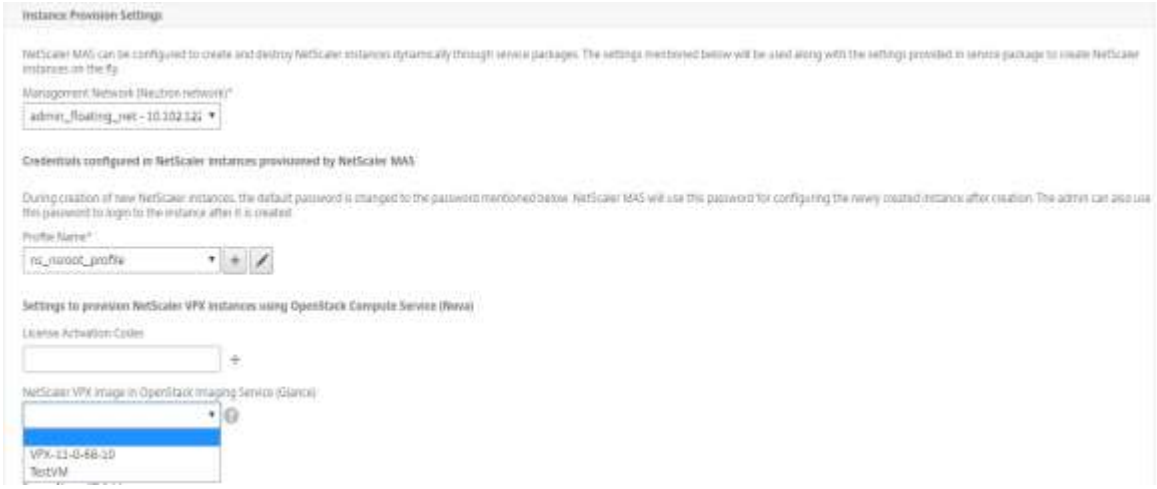


2. Compare the Images present in MOS and MAS.

Images present in OpenStack

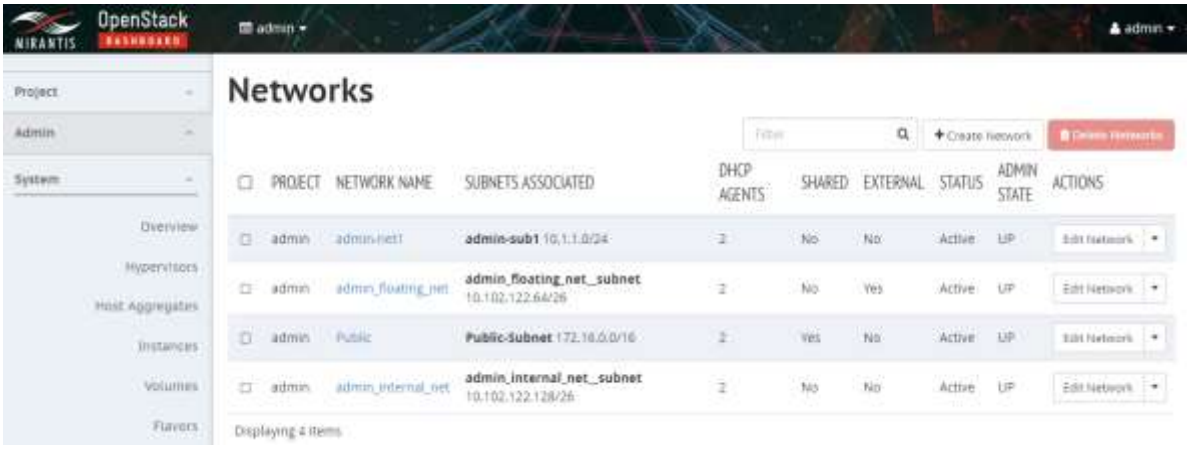


MAS fetches the Images from OpenStack

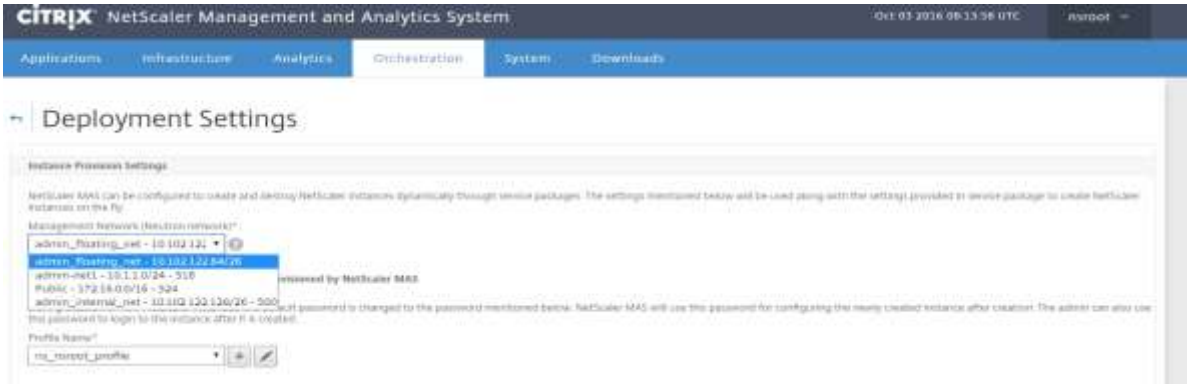


3. Compare the Networks present in MOS and MAS.

Networks present in OpenStack.

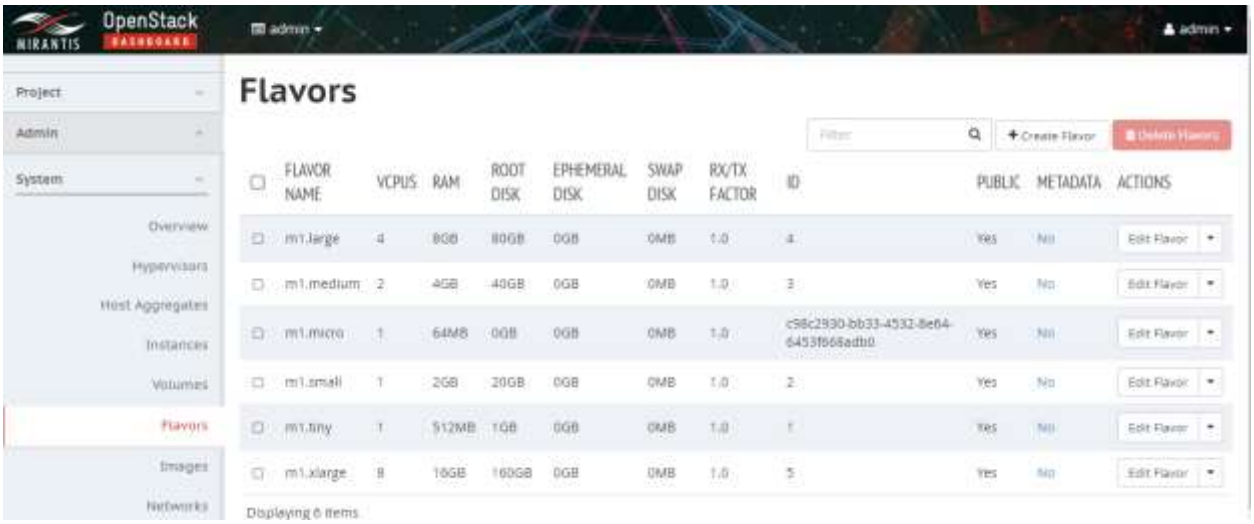


MAS fetches Networks from OpenStack

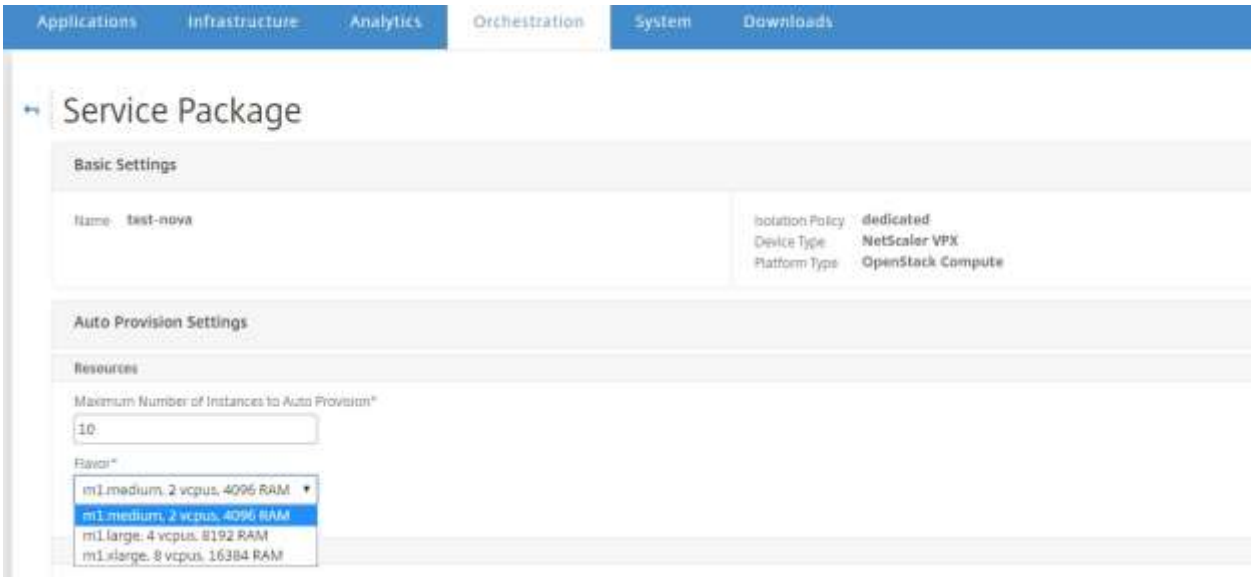


4. Compare the Flavors present in MOS and MAS.

Flavors present in OpenStack



MAS fetches the Flavors from OpenStack.



5.5.2 Target Use case(s)

Provide LBaaS service through Mirantis OpenStack by integrating Citrix NetScaler devices through NetScaler MAS.

- Provide LBaaS with pre-provisioned Dedicated NetScaler VPX.
- Provide LBaaS with pre-provisioned NetScaler MPX.
- Provide LBaaS with VPX instances auto-provisioned(both in standalone and HA mode) on a NetScaler SDX appliance,

Deployment modes and configuration options

The following table lists the minimum combination of Mirantis OpenStack deployment options. Example:

OS	Mode	HV	Network	Storage	
			VLAN	Object	Block
Ubuntu (14.04.5	HA	KVM	X	Ceph	Cinder

LTS)					

NS Release version	Pre-Provisioned VPX	Pre-Provisioned MPX	SDX	Nova based VPX
10.5	Yes		Yes	
10.5.e	Yes		Yes	
11.0	Yes		Yes	
11.1	Yes		Yes	

Functional testing

1) Auto provision on SDX :
    a. In MAS, add the SDX devices under infrastructure.
    b. Create a Service package with **SDX** as the platform and **NetScaler VPX** as the device type. Specify the values (integer only) for cores, memory, SSL chips, and throughput.
    c. Assign the Device and Tenant to the Service package.
    d. The tenant then creates the Loadbalancer, Listener, pool and VIP from OpenStack.

2) LBaaS for Pre provisioned MPX device with dedicated isolation policy :
    a. In NSMAS, add the MPX device under infrastructure
    b. Create a Service Package with isolation policy as “dedicated.”
    c. Assign the Device and Tenant to the Service Package.
    d. The tenant creates the Loadbalancer, Listener, pool and VIP from OpenStack

3) LBaaS for Pre provisioned VPX device with dedicated isolation policy
    a. In NSMAS, add the VPX device under infrastructure
    b. Create a Service Package with isolation policy as ”dedicated.”
    c. Assign the Device and Tenant to the Service Package.
    d. The tenant creates the Loadbalancer, Listener, pool and VIP from OpenStack

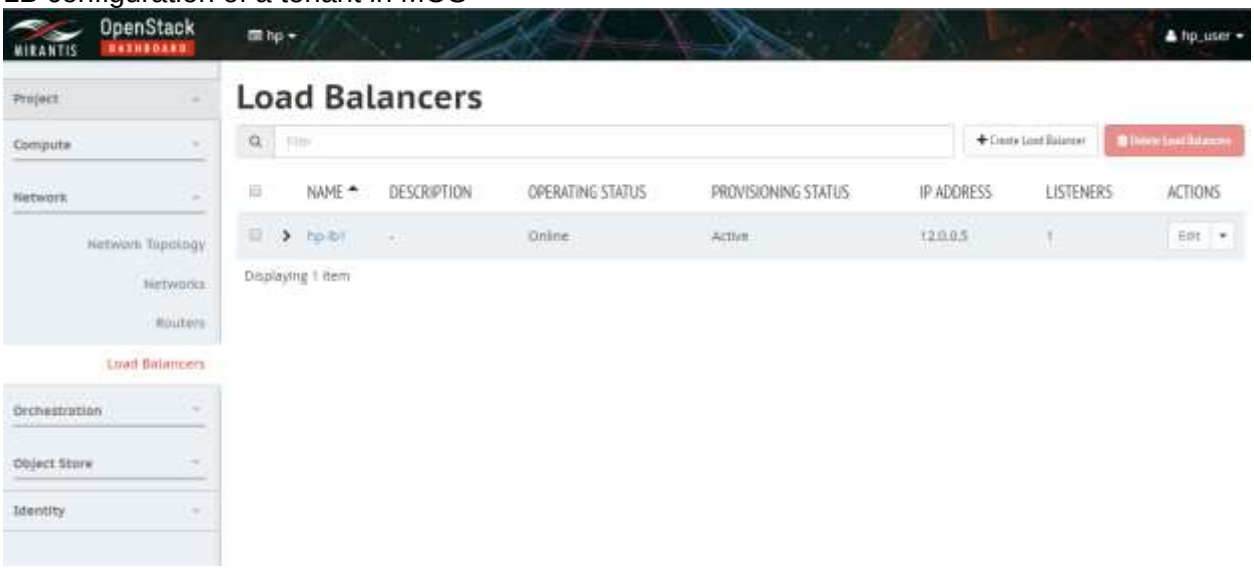
Negative testing

1) Verify registering Mirantis OpenStack with NetScaler MAS fails if the OpenStack credentials are incorrect.
2) Verify a tenant that does not exist in OpenStack keystone cannot log on to the NetScaler MAS user interface.
3) Verify the OpenStack driver cannot communicate with NetScaler MAS without proper credentials in the neutron.conf file.
4) Verify VPX provisioning fails if the OpenStack Nova service is down.
5) Verify VPX provisioning fails if the OpenStack Keystone service is down.
6) Verify if OpenStack Neutron service is down after provisioning the device, VPX device is not connected to the network.
7) Verify that insufficient resources on the Compute node cause VPX provisioning to fail.
8) Verify that insufficient resources on the SDX device cause VPX provisioning to fail.

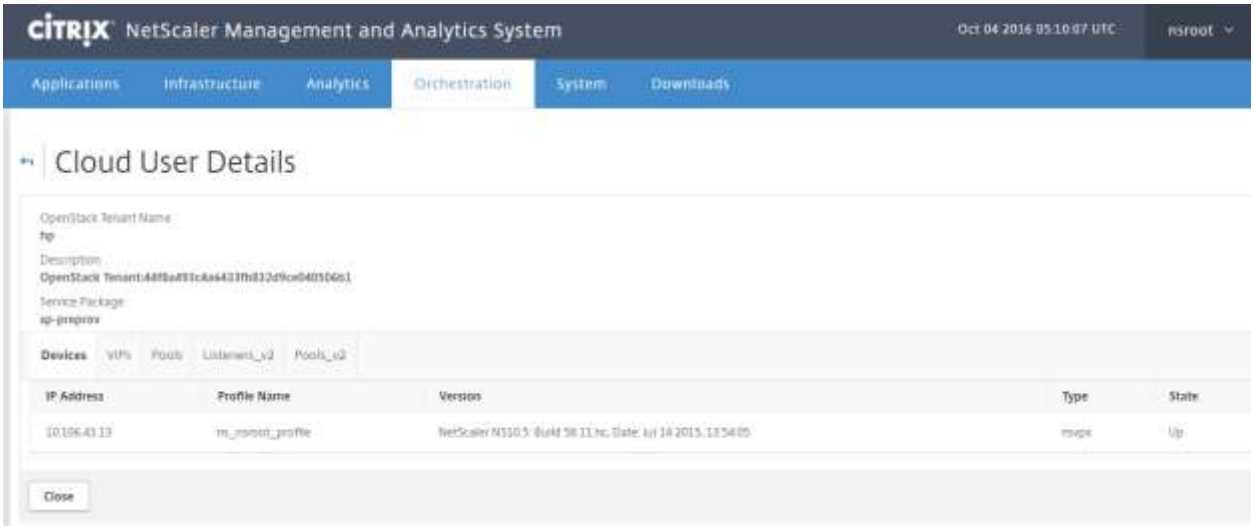
Use Case Test Results

1. **Pre-provision Dedicated VPX Use case:**

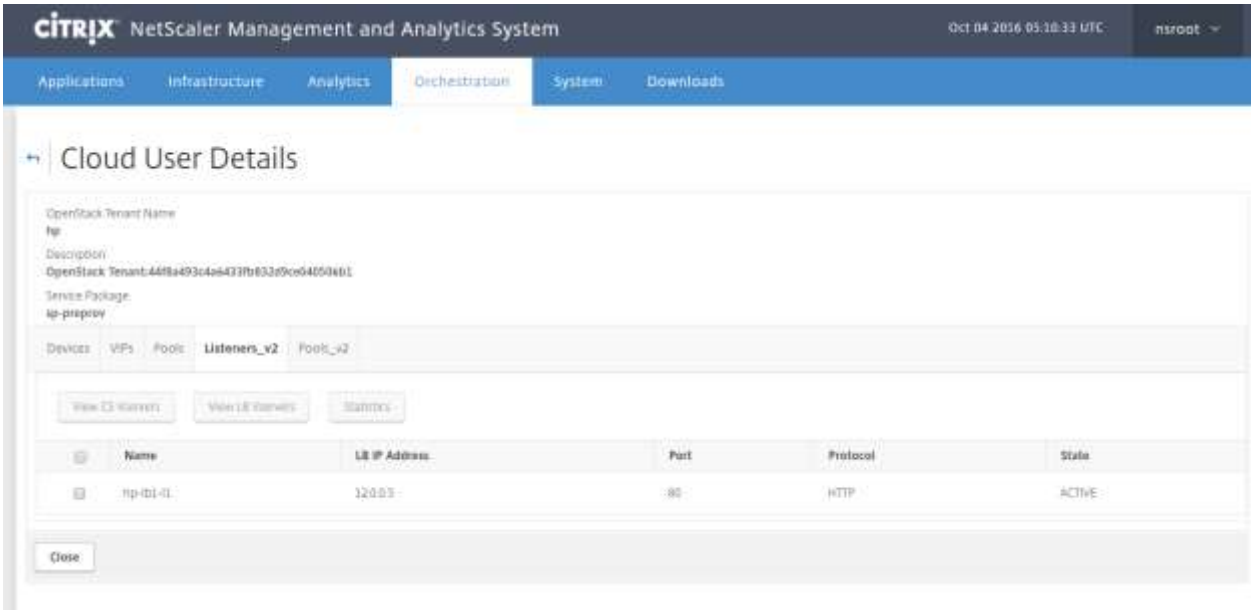
LB configuration of a tenant in MOS



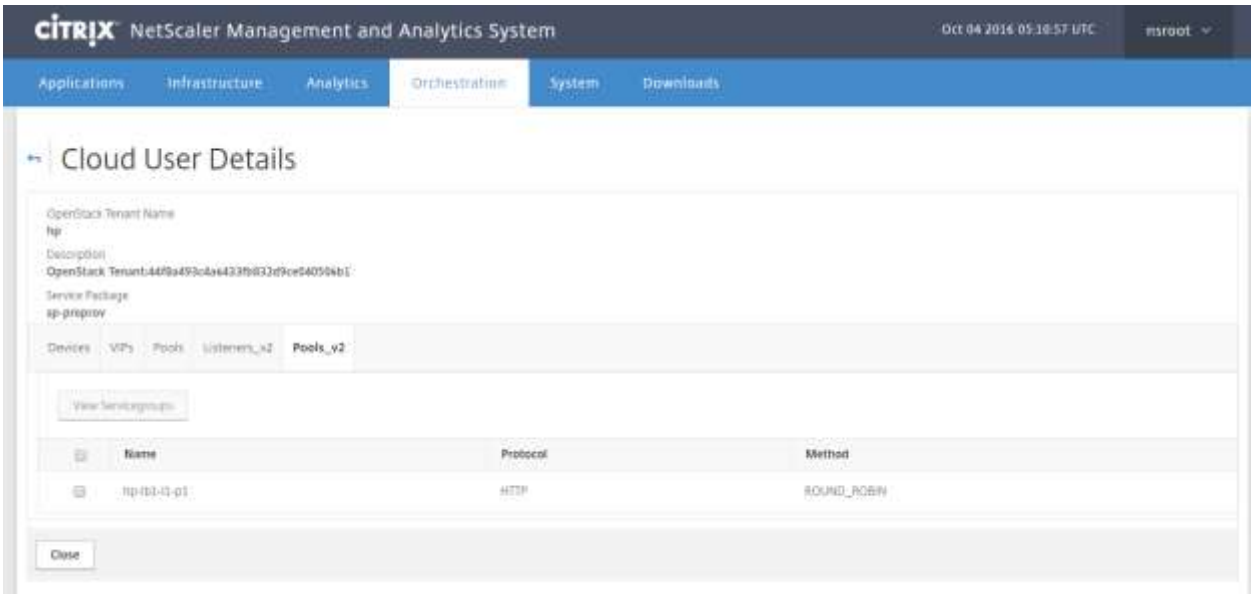
LB configuration of a tenant in MAS



Listener details

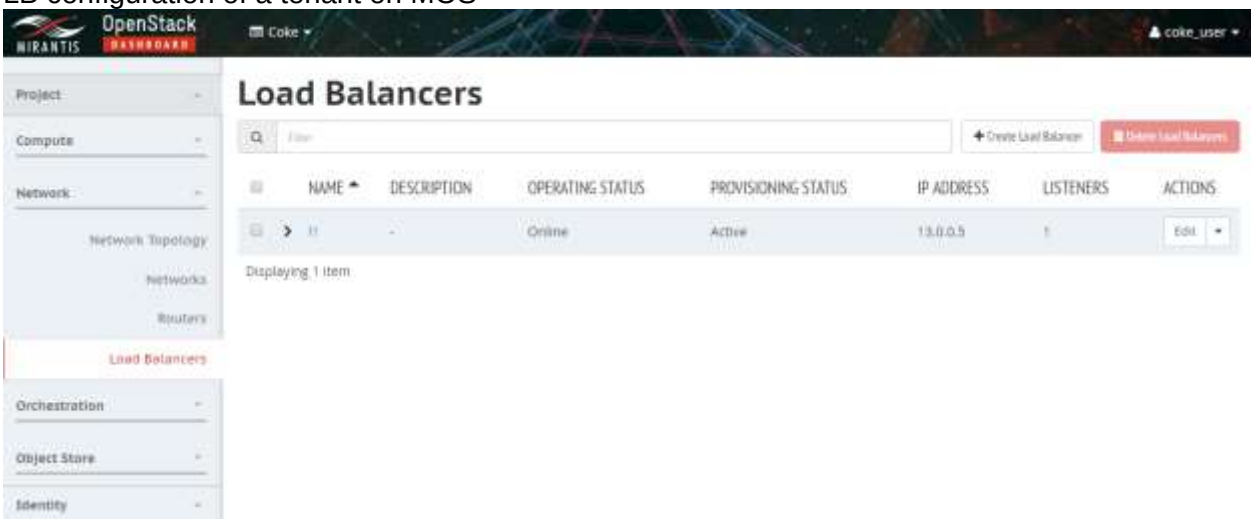


Pool details

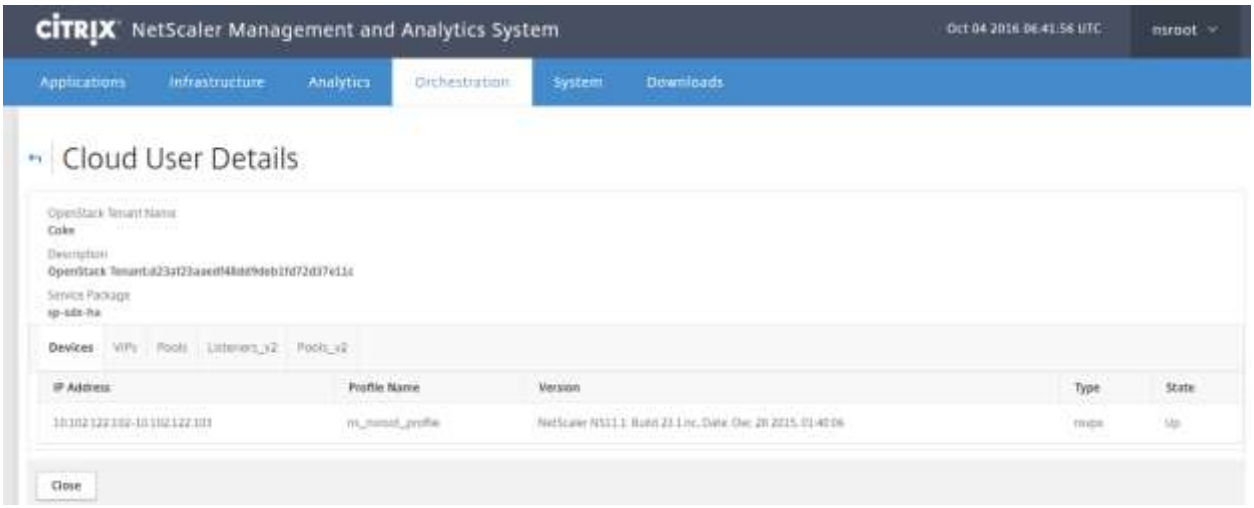


2. Auto-provision of VPX in HA on SDX use case:

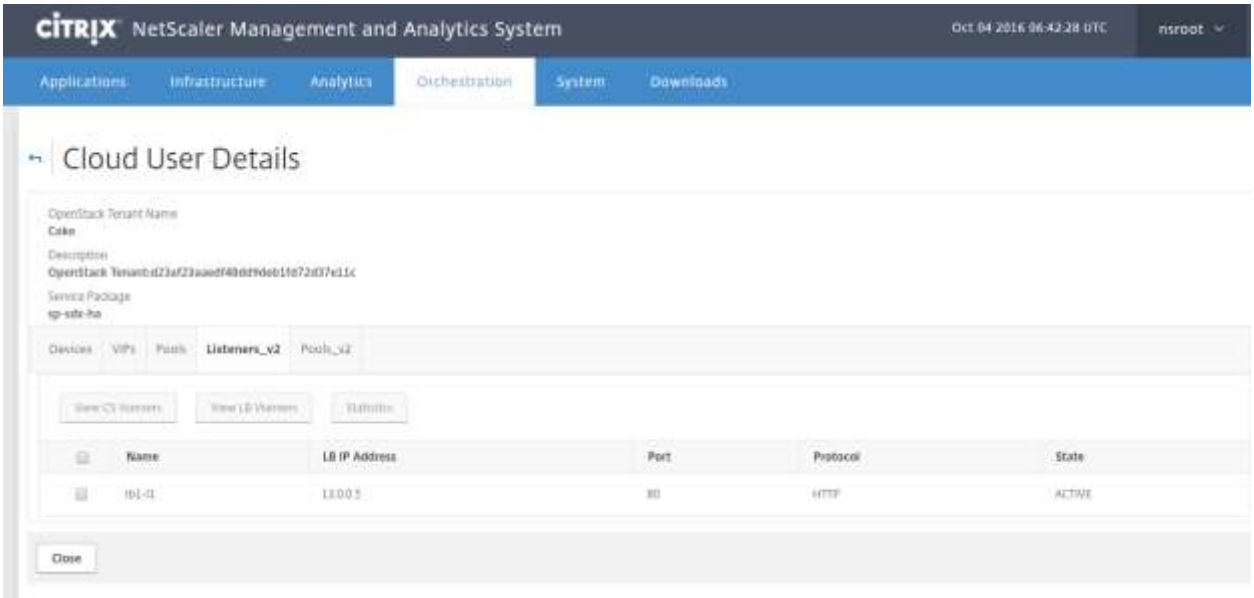
LB configuration of a tenant on MOS



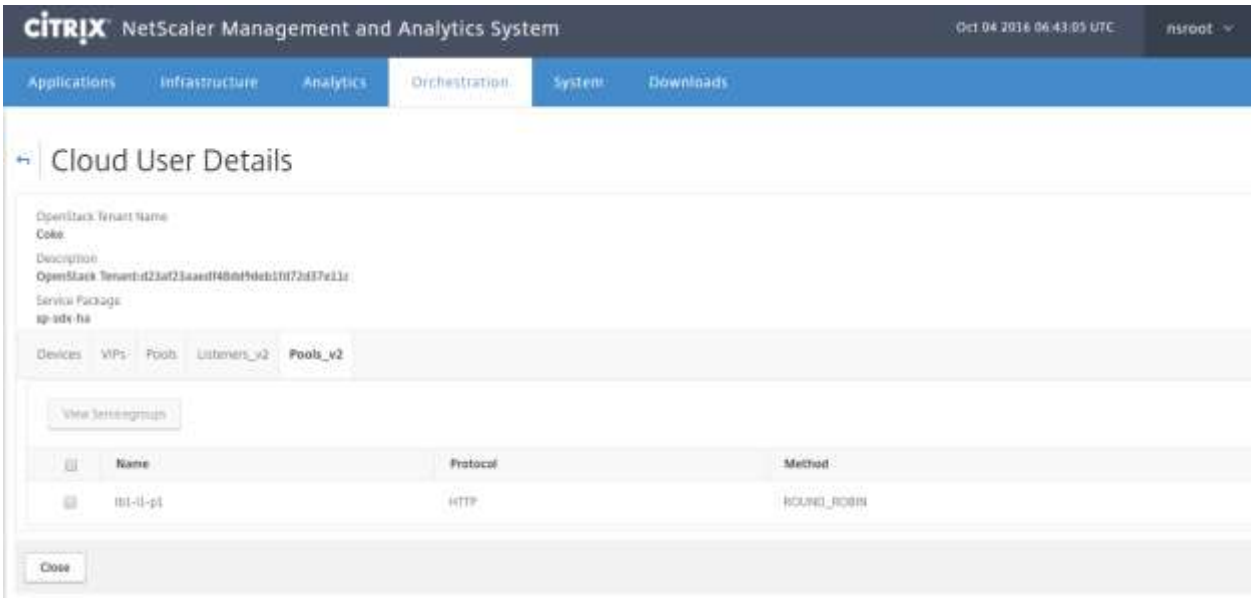
LB configuration of a tenant in MAS



Listener details

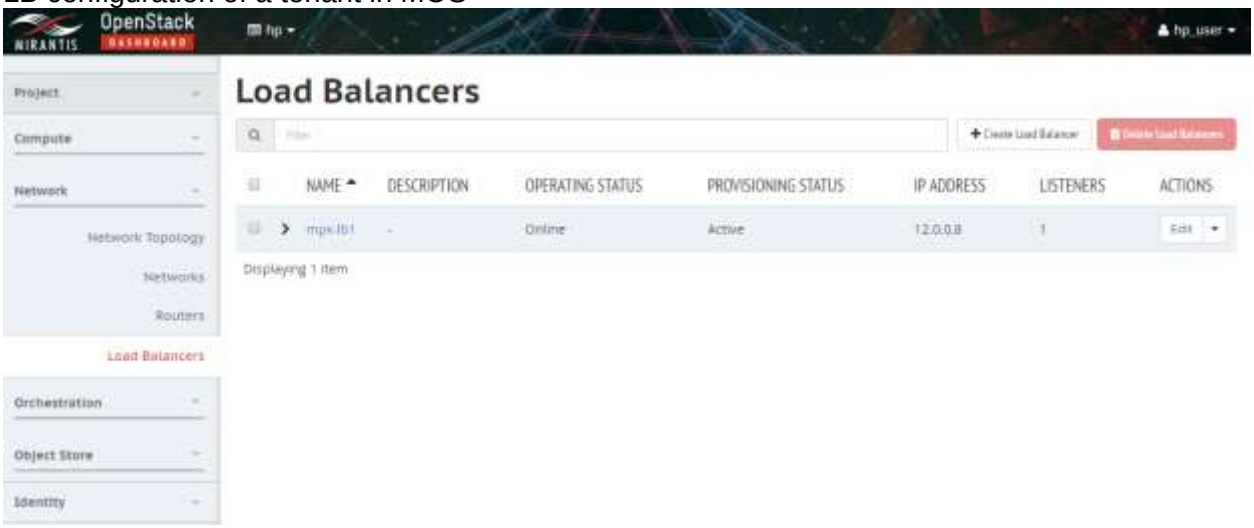


Pool details

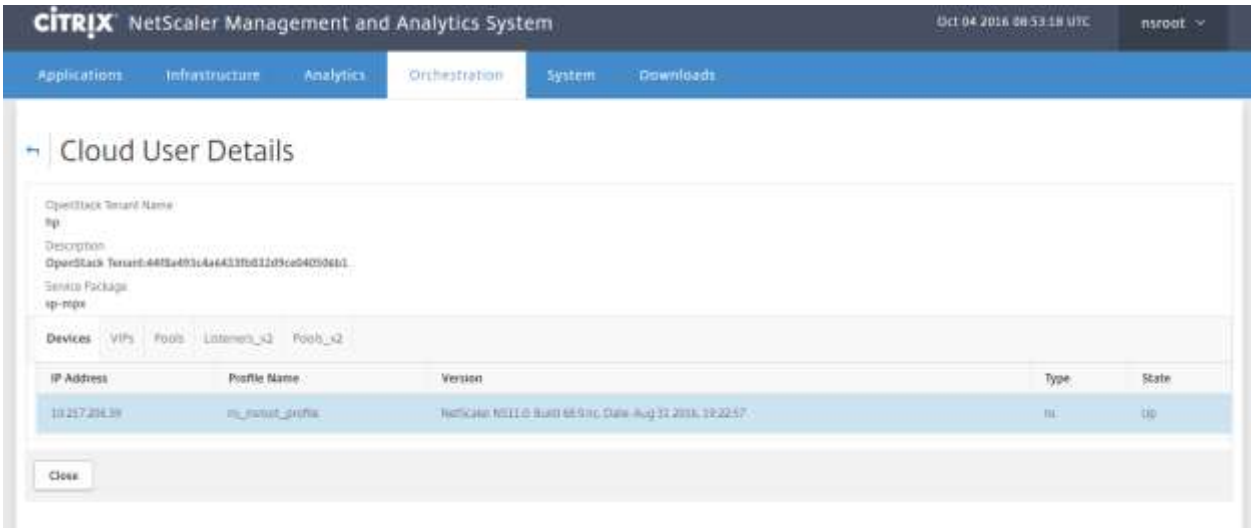


3. Pre-provision Dedicated MPX Use case:

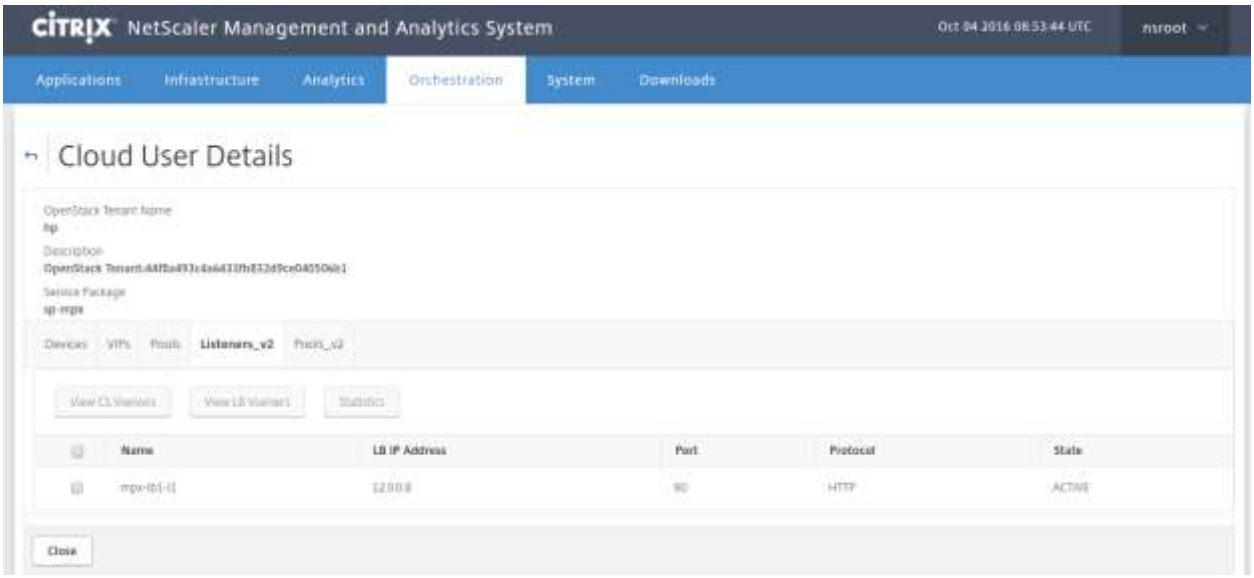
LB configuration of a tenant in MOS



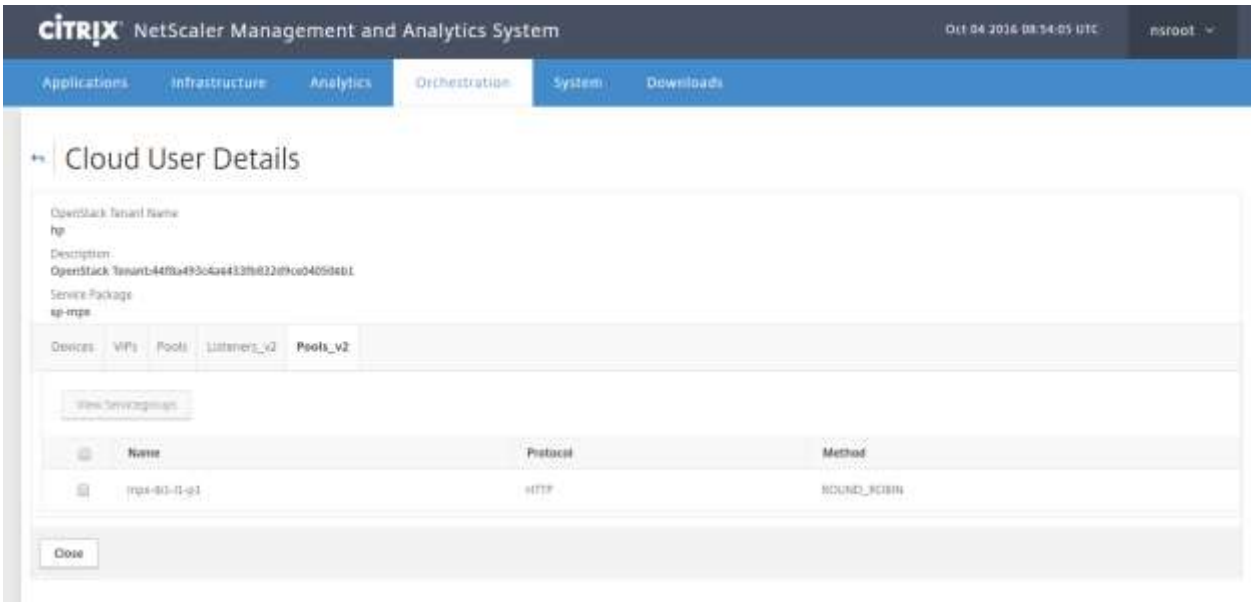
LB configuration of a tenant in MAS



Listener details



Pool details


CITRIX NetScaler Management and Analytics System
Oct 04 2016 08:54:05 UTC
nsroot
Applications
Infrastructure
Analytics
Orchestration
System
Downloads
Cloud User Details
OpenStack Tenant Name
hp
Description
OpenStack Tenant:44ff0a493c4a4433fb822df9ce0405deb1
Service Package
sp-mpx
Devices
VIPs
Pools
Listeners_v2
Pools_v2
Name
Protocol
Method
HTTP
ROUND_ROBIN
Close