DEPLOYMENT GUIDE

Zero Touch Provisioning of NIOS on OpenStack using Red Hat Ansible

NIOS version 8.3+

Contents

© 2020 Infoblox Inc. All rights reserved.

Overview

Introduction

Deploying an Infoblox Grid manually on OpenStack and configuring it could be a time-consuming task. In a dynamic environment like a Service Provider setup, cutting short the deployment time is the key to success. Zero touch provisioning (ZTP) of NIOS on OpenStack using Ansible enables infrastructure teams to deploy Infoblox Grid on OpenStack without any manual intervention. It reduces the deployment time and automates the initial configuration like setting up licenses, IP addresses and starting the services.

Ansible is an open source software that automates software provisioning, configuration management, and application deployment. Ansible connects via SSH, remote PowerShell or via other remote APIs.

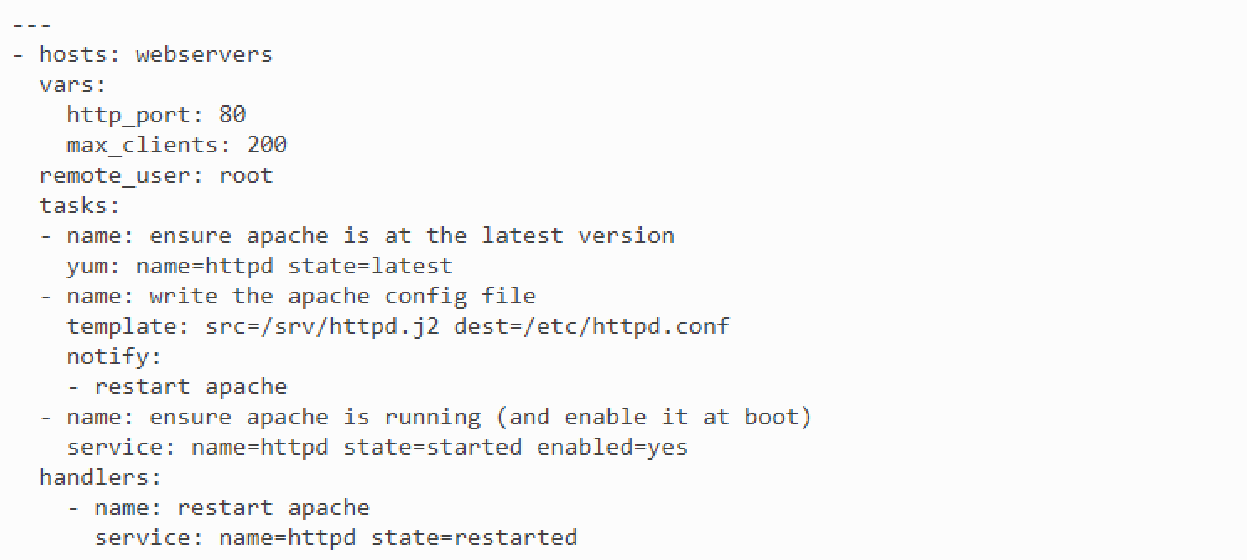
Ansible Playbooks (a.k.a Playbooks)

An Ansible playbook contains one or multiple plays, each of which define the work to be done for a configuration on a managed server. Ansible plays are written in YAML. Every play is created by an administrator with environment-specific parameters for the target machines or servers.

Ansible plays are flexible due to modules, which pertain to various aspects of the target managed servers. The module script is written in Ruby. Modules exist for many parts of system configuration, including software installation and user management.

The playbook is therefore composed of plays, which are composed of modules. It executes when the administrator runs the `ansible-playbook` command against target machines. The administrator must use an inventory file to specify the hosts under the playbook's management. The inventory file contains a list of all hosts that are managed by Ansible, and it offers an option to group hosts according to their functionality.

For example, following sample playbook ensures that the apache is at the latest release, writes the config file and enables the auto start feature.



```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
  - name: ensure apache is at the latest version
    yum: name=httpd state=latest
  - name: write the apache config file
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running (and enable it at boot)
    service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

Use Cases

The Ansible playbook discussed in the document addresses the following use cases:

- Deploying Grid Master
    - Deployment
    - License and network initialization using cloud-init
- Deploying a Member
    - Deployment
    - License and network initialization using cloud-init
    - Adding the Member to the Grid.
- Starting DNS and DHCP services

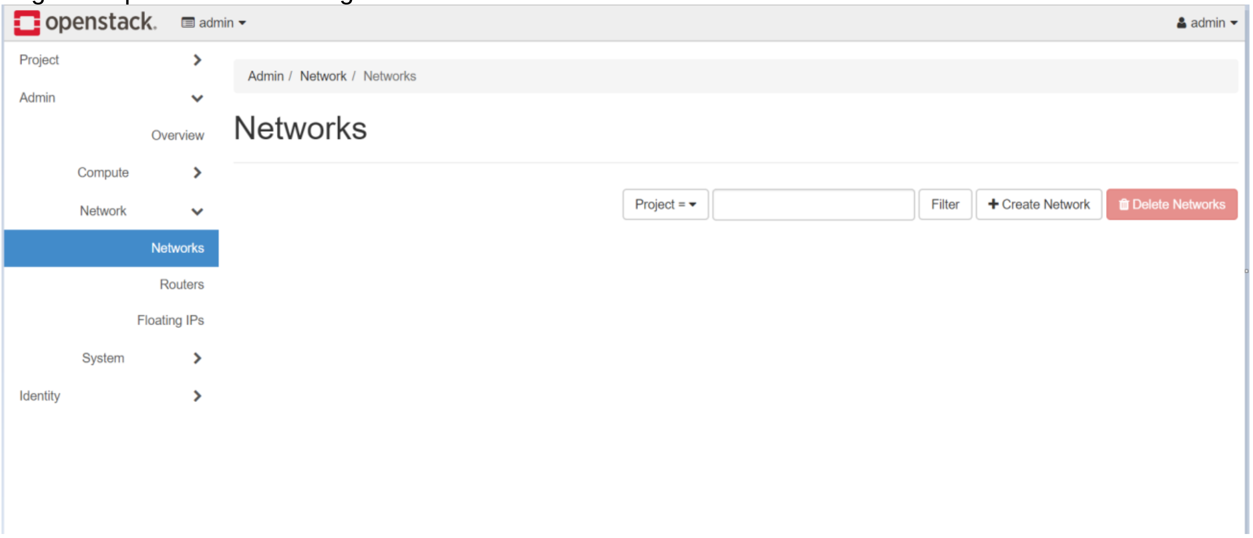
Pre-requisites

- A working OpenStack (Newton release onwards) setup with sufficient resources to host Infoblox grid (a minimum of 24 GB RAM, 8 vCPUs and 600 GB hard disk space).
- The vNIOS image for KVM (version specific or DDI) in the qcow2 format.
- A Linux machine with Ansible (**2.6.2** onwards) installed and access to the OpenStack setup.(Please refer → https://docs.ansible.com/ansible/2.5/installation_guide/intro_installation.html#id15)
- Download the Ansible playbooks from https://github.com/infobloxopen/nios-ztp-anisble.git to the above-mentioned Linux machine with ansible installed.

Setting up the OpenStack for deploying NIOS through Ansible

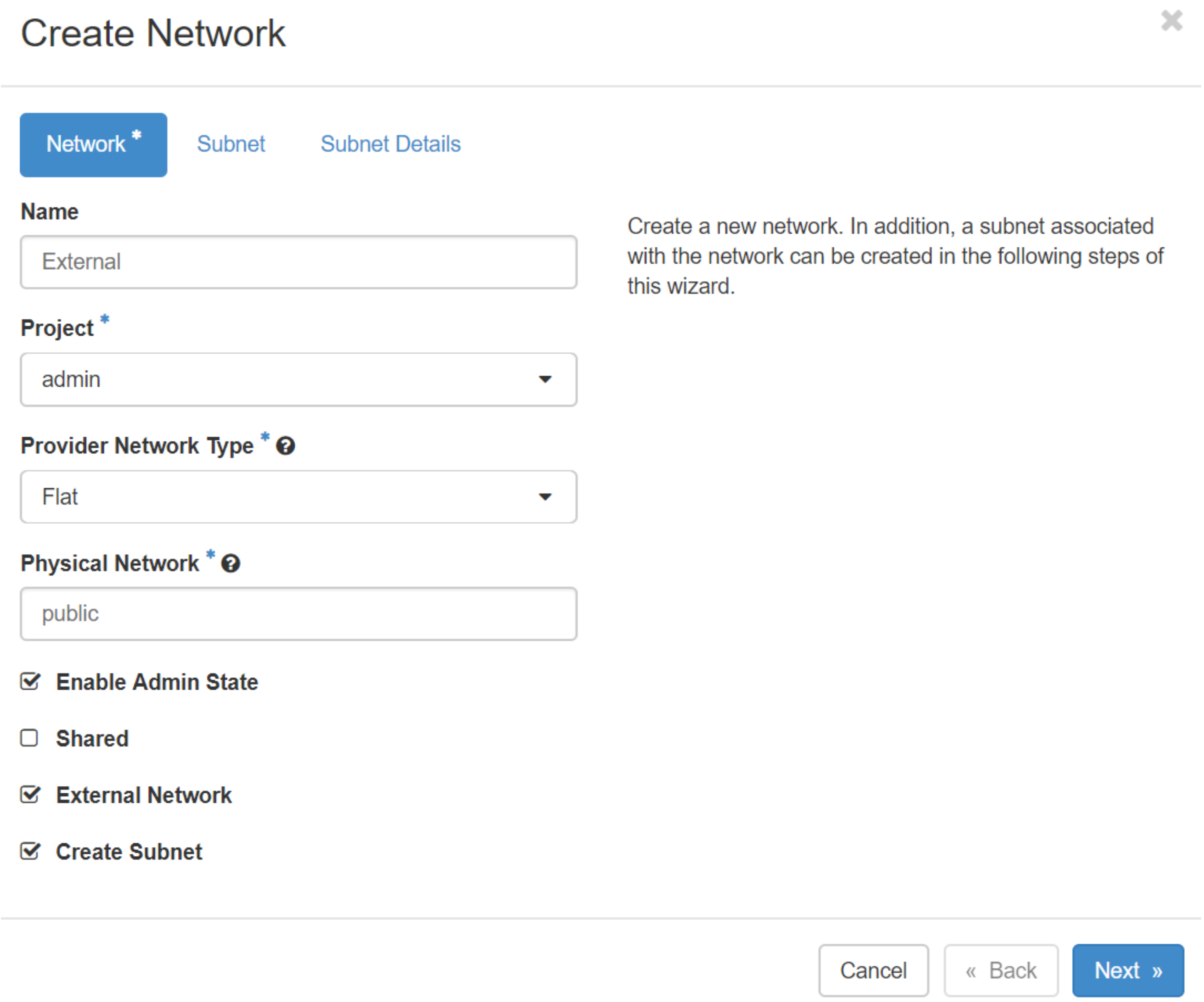
Creating External Network

1. Login to Openstack and navigate to **Admin → Network → Create Network**



2. Enter the name for this external network.
3. In the **Project** dropdown box, select the project name.
4. In the **Provider Network Type** drop down, select your provider network type.
   In this deployment guide, we use Flat as provider network type and the name of the physical network is public.
   (For more information on OpenStack networking please refer → https://docs.openstack.org/mitaka/networking-guide/intro-os-networking.html)

5. Check the External Network check box and click on next.



Create Network

Name: External

Project *: admin

Provider Network Type *: Flat

Physical Network *: public

☑ Enable Admin State

□ Shared

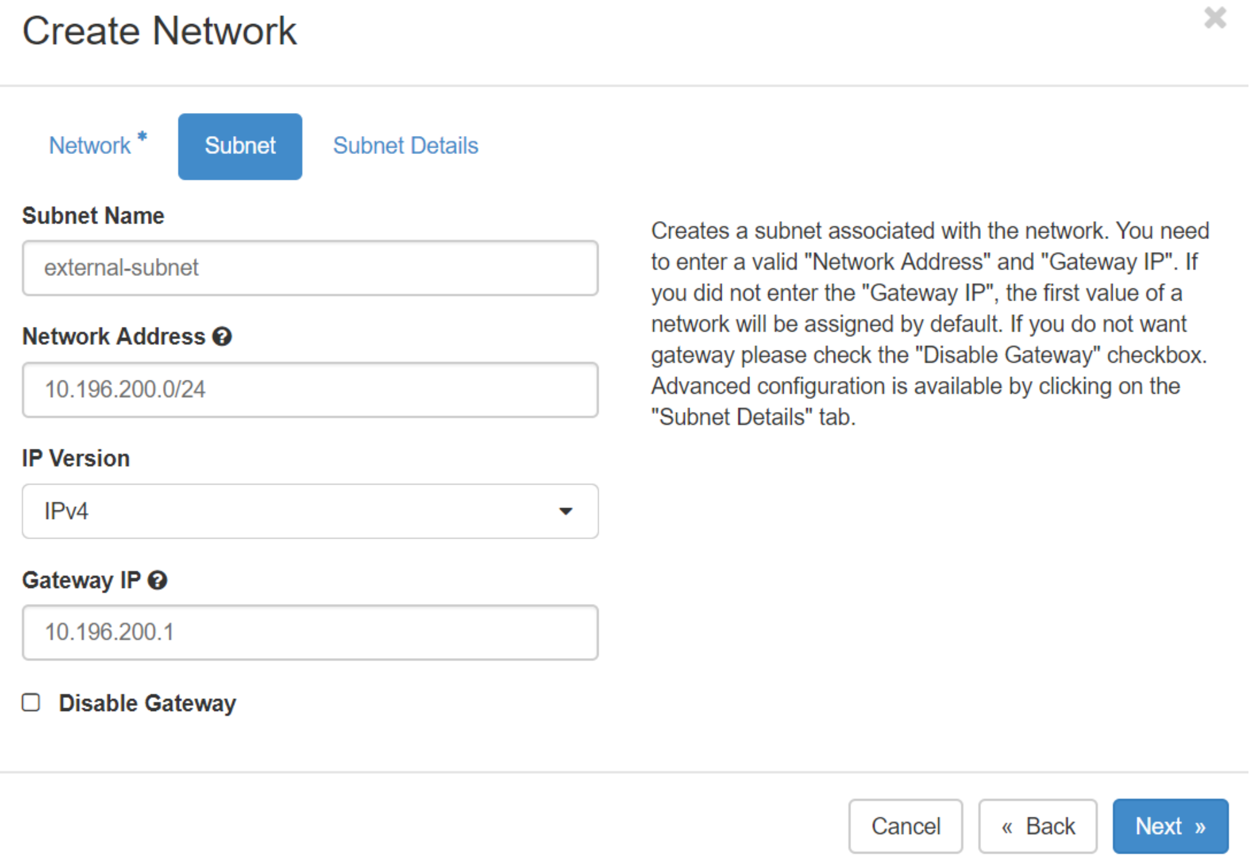
☑ External Network

☑ Create Subnet

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

6. In the **Subnet Name** option enter the external network subnet name.
7. In the **Network Address** option enter the external network address in CIDR notation.

8. You can enter the gateway. If you don’t, OpenStack by default takes the first IP Address as the Gateway IP. Click on next.



Create Network

Subnet Name

external-subnet

Network Address

10.196.200.0/24

IP Version

IPv4

Gateway IP

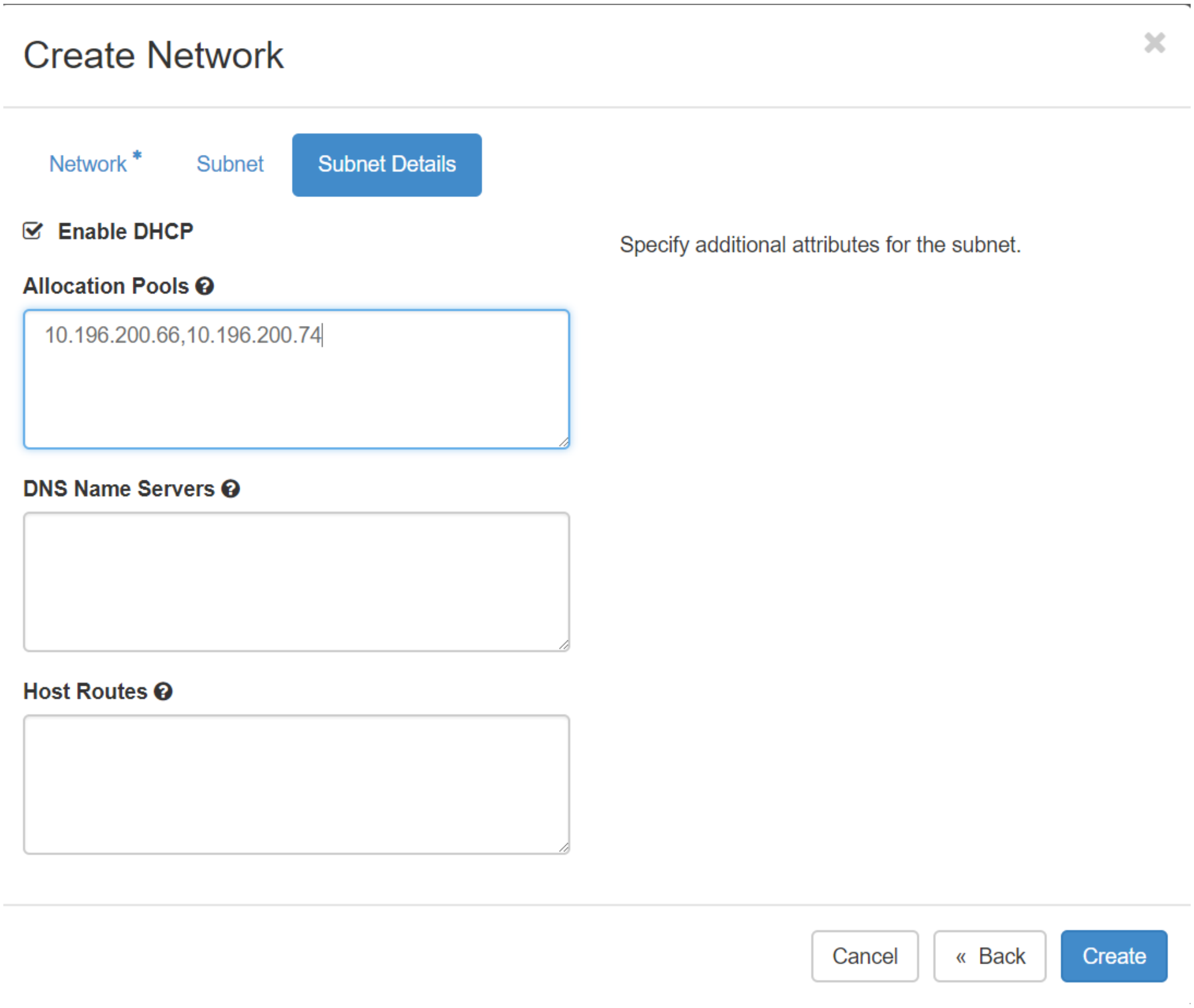
10.196.200.1

□ Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

9. In the next screen, check the **Enable DHCP** option.
10. In the allocation pool enter the IPs from the external subnet which you would like Openstack to use as floating IPs. For example, 10.196.200.66,10.196.200.74 will allocate 9 IPs.

11. Click on create to create the external network.



Create Network

Network * | Subnet | Subnet Details

☑ Enable DHCP

Specify additional attributes for the subnet.

Allocation Pools: 10.196.200.66,10.196.200.74

DNS Name Servers:

Host Routes:

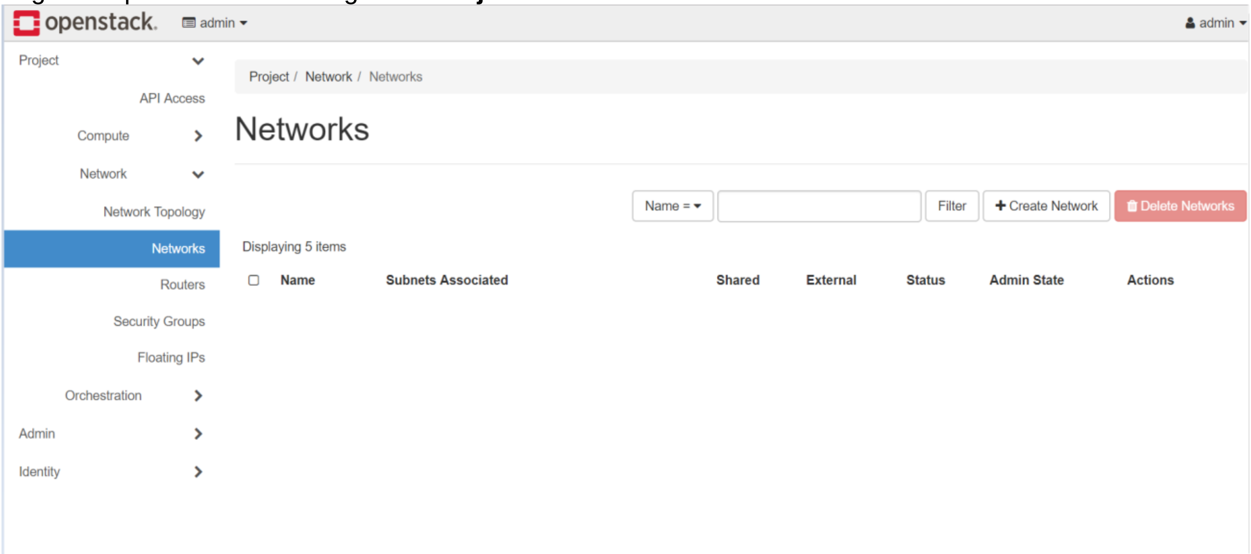
Creating Mgmt and Lan1 networks

NIOS requires a minimum of 2 networks to boot up successfully. In case it does not detect atleast 2 networks, it throws a Fatal Error during boot process.

In this deployment guide, we first create 2 networks for NIOS, called Mgmt and Lan1. Once the networks are created, we create 2 ports in each network. Port of Lan1 network will be associated with the floating IP so that Infoblox Grid can be accessed from outside of OpenStack.

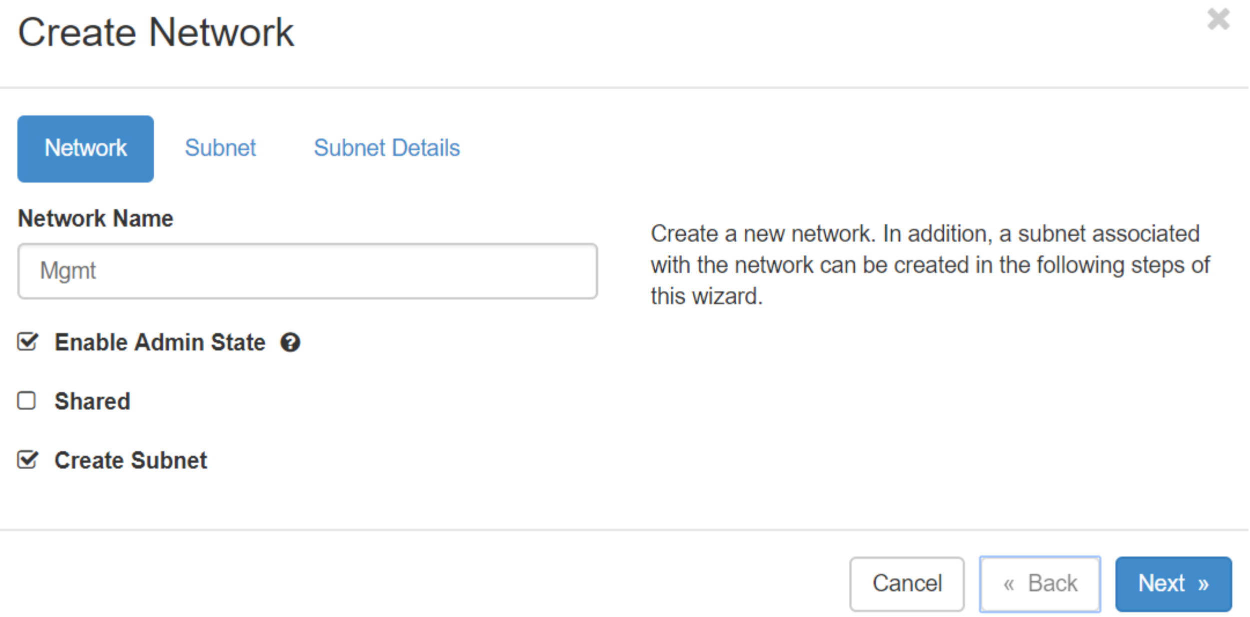
Creating Mgmt Network

1. Login to OpenStack and navigate to **Project → Networks** and click on **Create Network.**



□	Name	Subnets Associated	Shared	External	Status	Admin State	Actions

2. Enter the **Network Name** and click on next.



Network Name: Mgmt

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

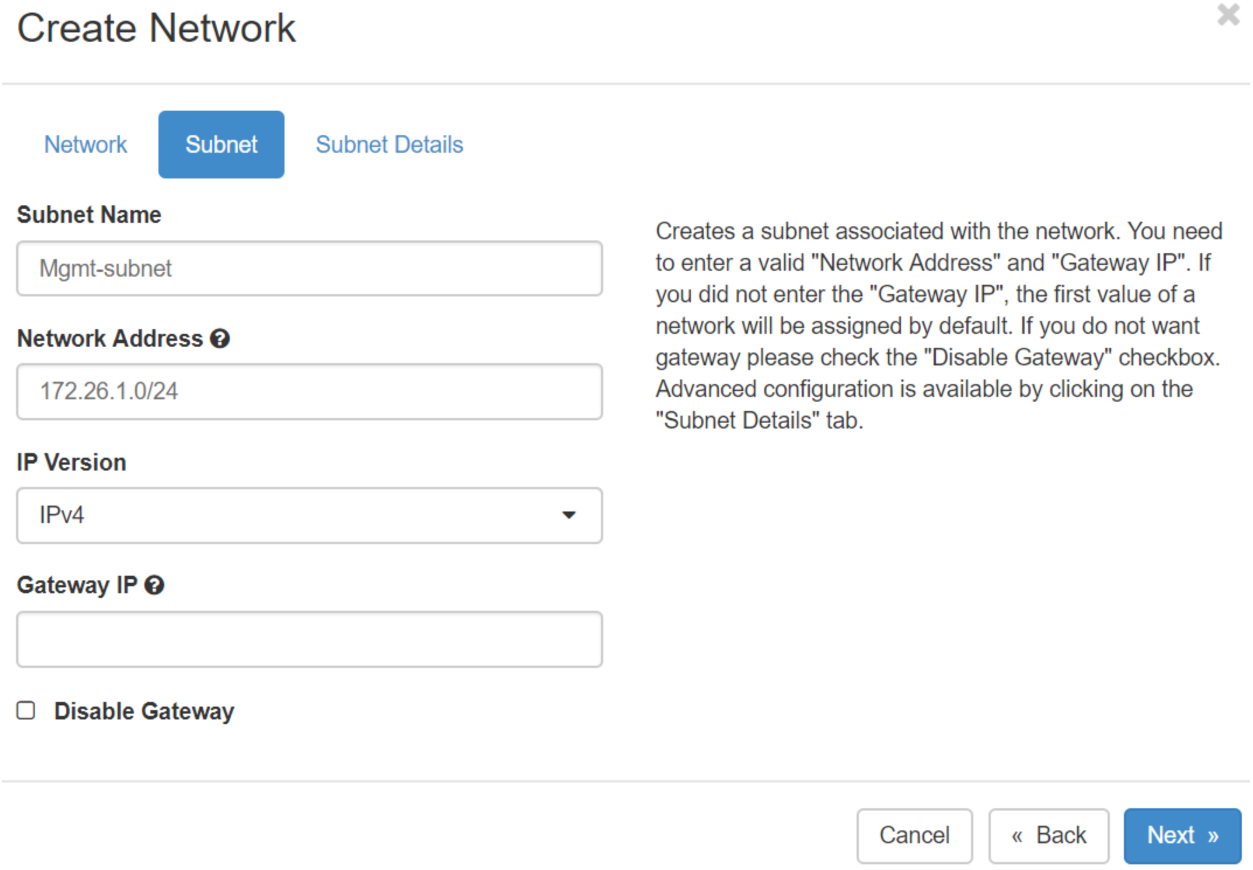
☑ Enable Admin State

□ Shared

☑ Create Subnet

3. In the **Subnet Name** option enter the Mgmt network subnet name.
4. In the **Network Address** option enter the Mgmt network address in CIDR notation.

5. You can enter the gateway. If you don’t, OpenStack by default takes the first IP Address as the Gateway IP. Click on next



Create Network

Subnet Name

Mgmt-subnet

Network Address

172.26.1.0/24

IP Version

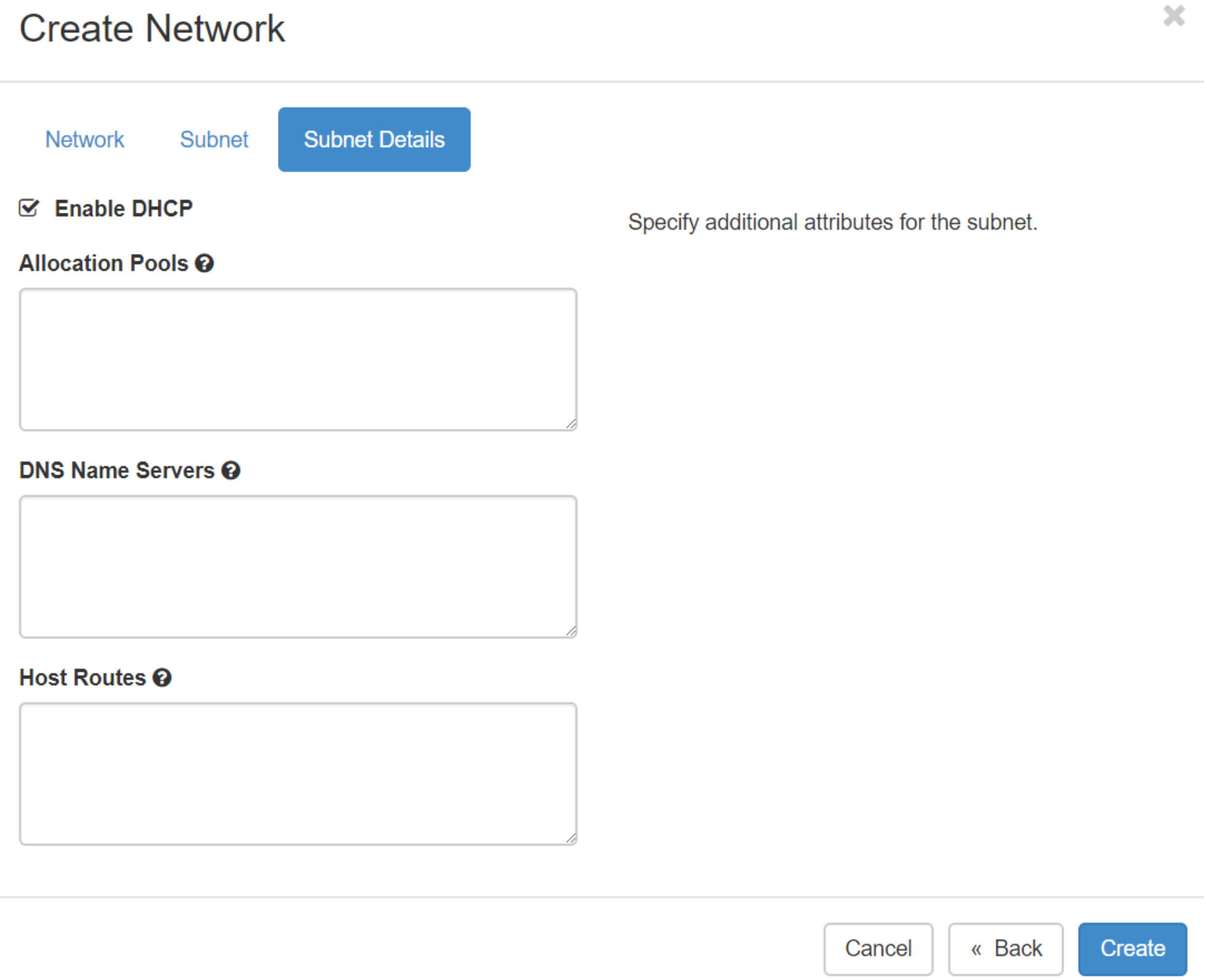
IPv4

Gateway IP

□ Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

6. Check the **Enable DHCP** option and click on create.

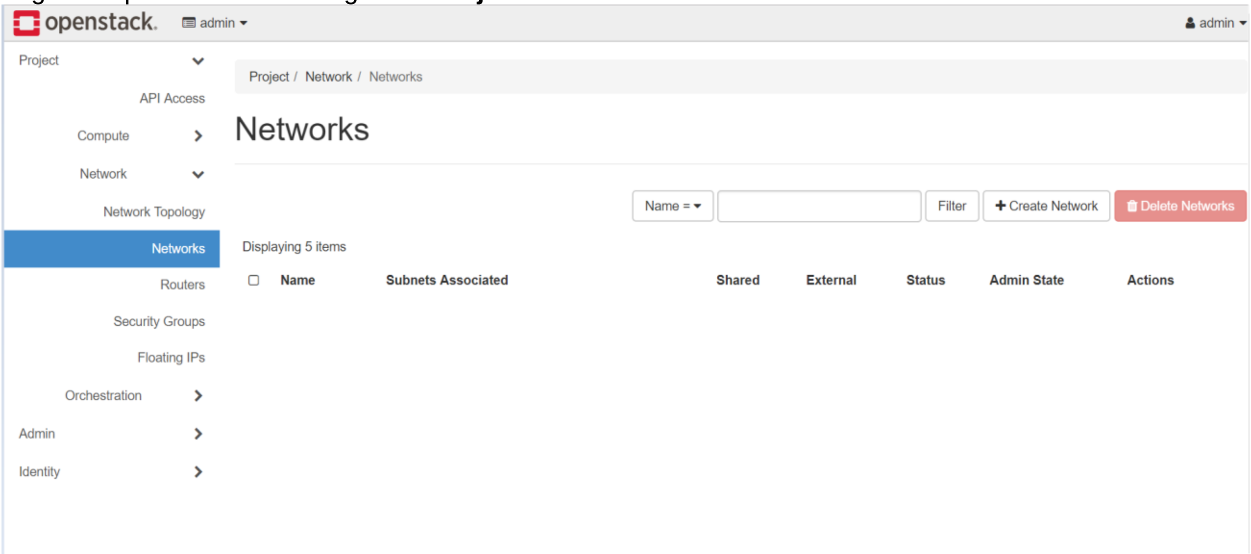


☑ Enable DHCP

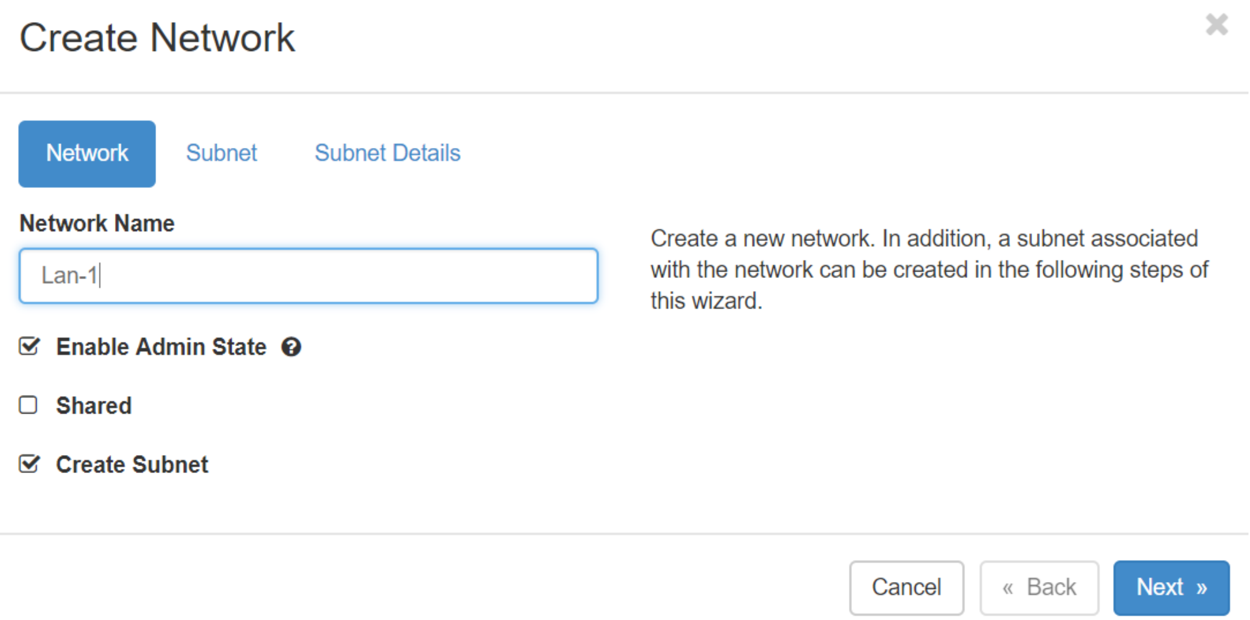
Specify additional attributes for the subnet.

Creating Lan1 Network

1. Login to OpenStack and navigate to **Project → Networks** and click on **Create Network.**



2. Enter the **Network Name** and click on next



Network Name: Lan-1

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

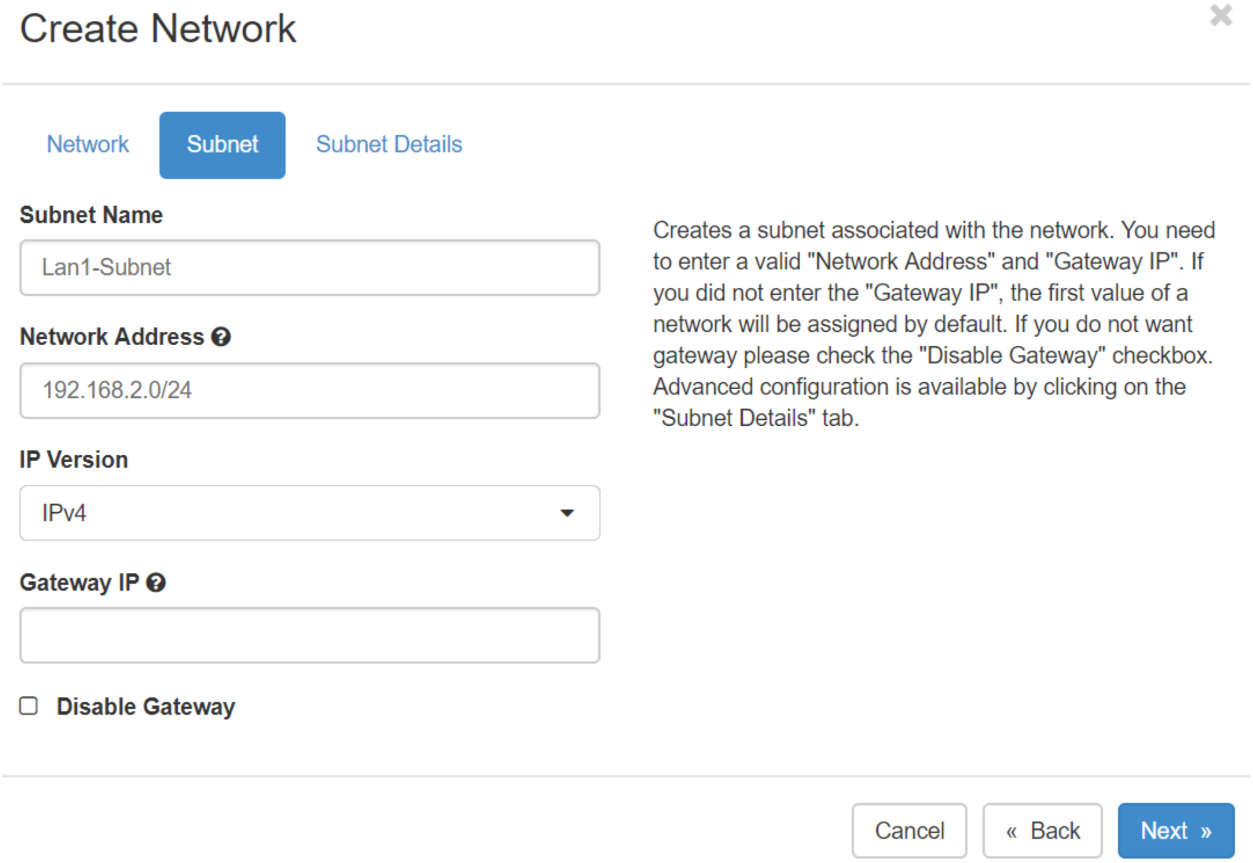
☑ Enable Admin State

□ Shared

☑ Create Subnet

3. In the **Subnet Name** option enter the Lan 1 network subnet name.
4. In the **Network Address** option enter the Lan 1 network address in CIDR notation.

5. You can enter the gateway. If you don’t, OpenStack by default takes the first IP Address as the Gateway IP. Click on next



Create Network

Subnet Name: Lan1-Subnet

Network Address: 192.168.2.0/24

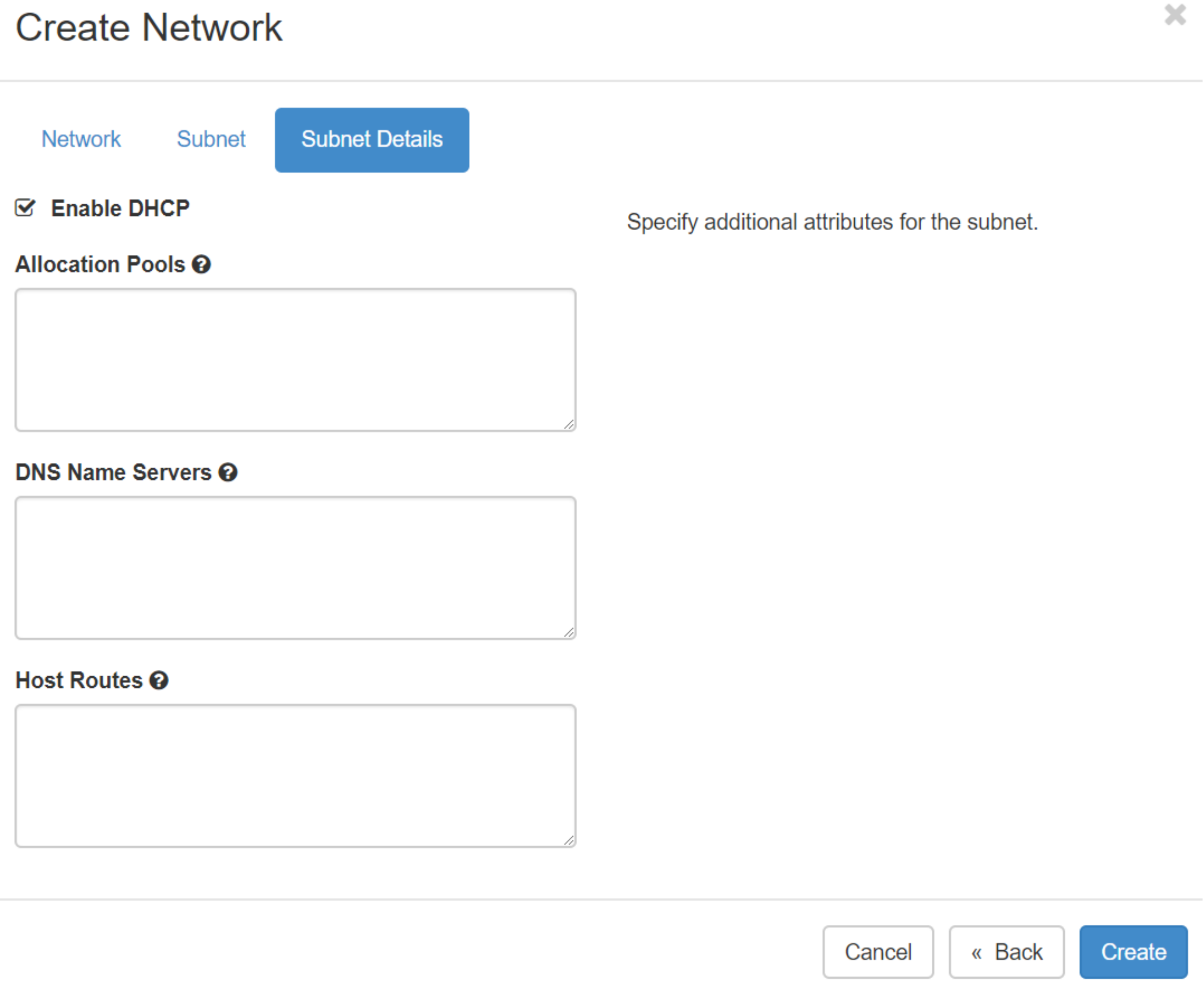
IP Version: IPv4

Gateway IP:

□ Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

6. Check the **Enable DHCP** option and click on create.



Create Network

☑ Enable DHCP

Specify additional attributes for the subnet.

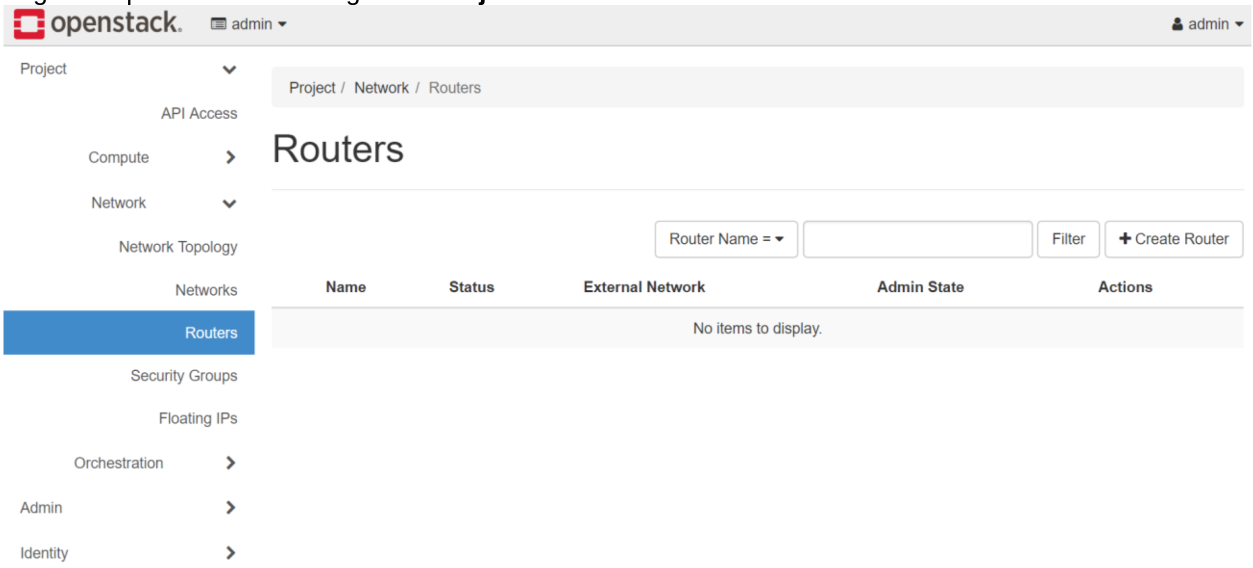
Allocation Pools

DNS Name Servers

Host Routes

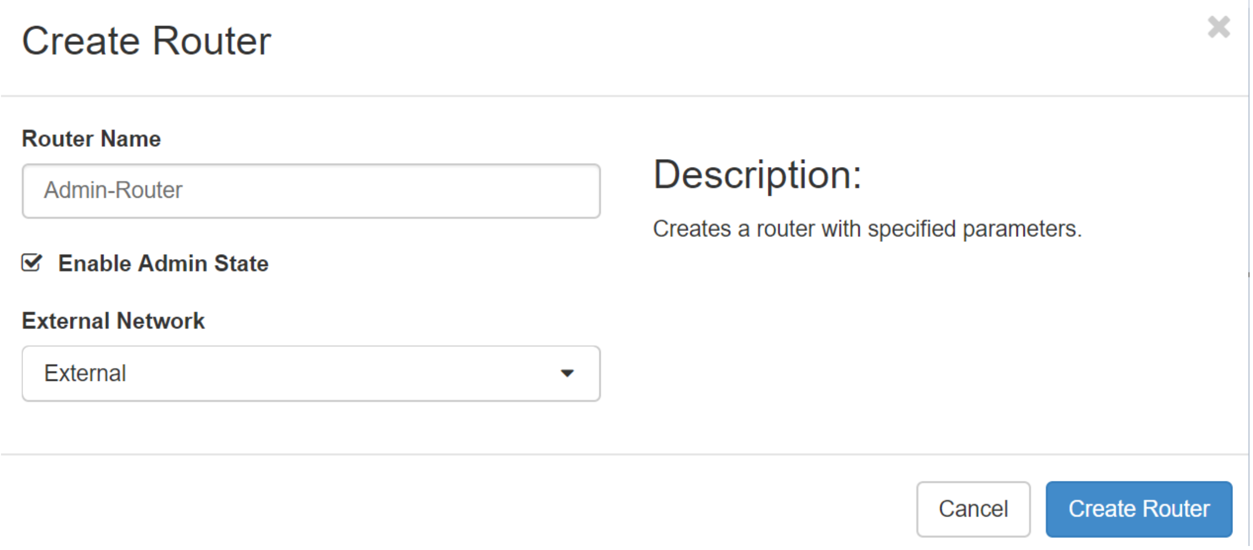
Creating a Router

1. Login to OpenStack and navigate to **Project → Network → Routers → Create Router**



Name	Status	External Network	Admin State	Actions
No items to display.				

2. Enter the **Router Name** and in **External Network** drop down option, select the external network which you previously created.
3. Click on **Create Router** to create the router.



Router Name

Admin-Router

☑ **Enable Admin State**

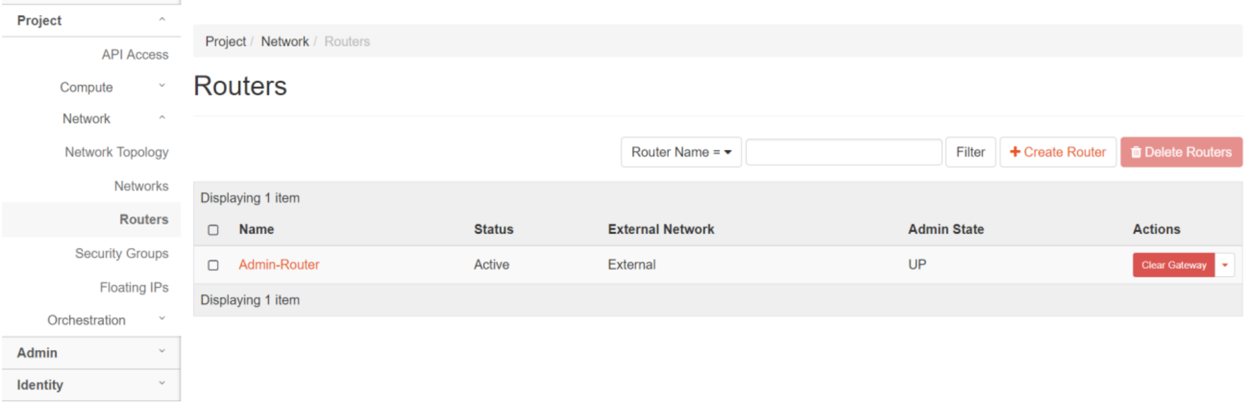
External Network

External

Description:

Creates a router with specified parameters.

4. Once the router is created click on the name of the router.



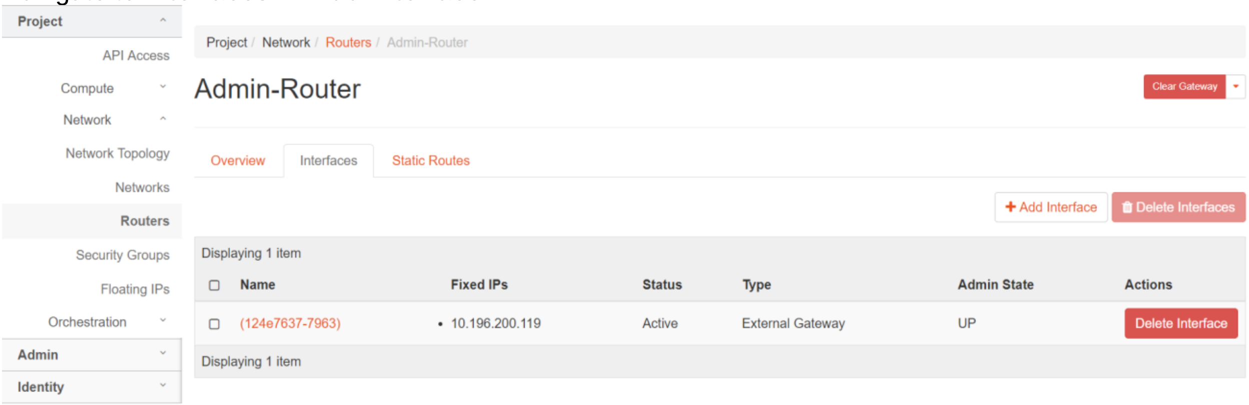
Routers

Displaying 1 item

□	Name	Status	External Network	Admin State	Actions
□	Admin-Router	Active	External	UP	Clear Gateway

Displaying 1 item

5. Navigate to **Interfaces → Add Interface**



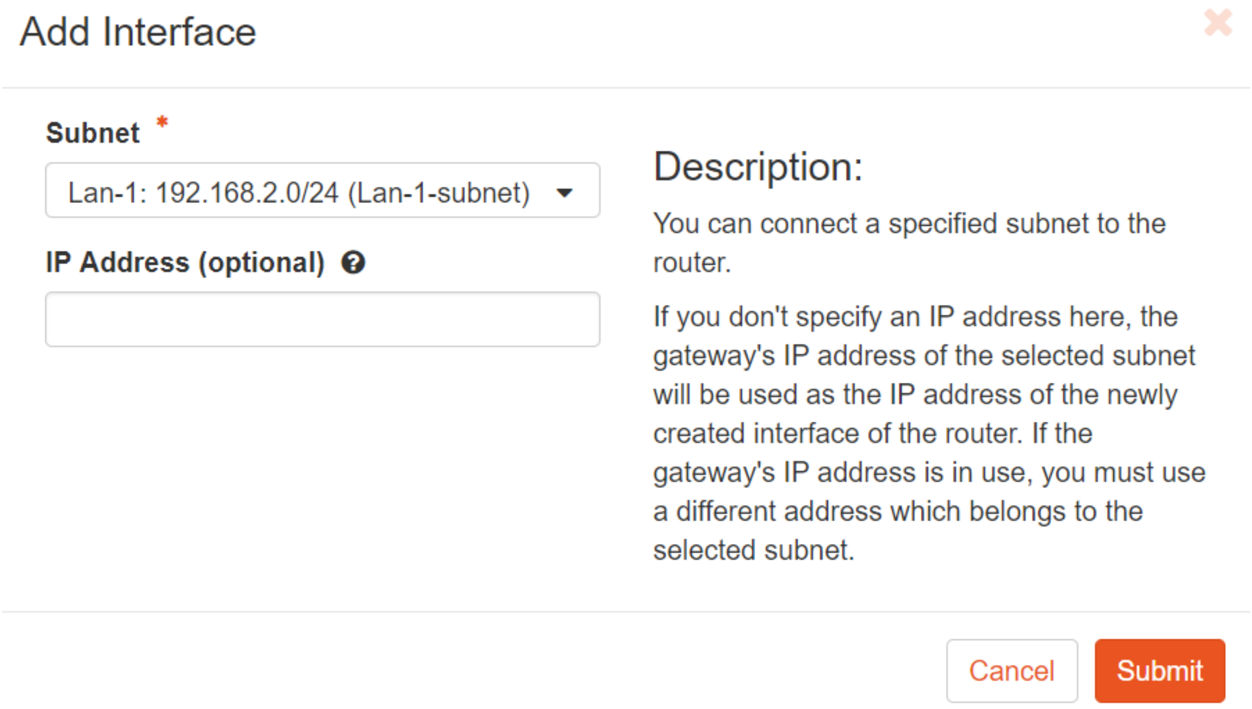
Admin-Router

Displaying 1 item

□	Name	Fixed IPs	Status	Type	Admin State	Actions
□	(124e7637-7963)	• 10.196.200.119	Active	External Gateway	UP	Delete Interface

Displaying 1 item

6. In the **Subnet** drop down option, select Lan-1 network and click on submit.



Add Interface

Subnet *

Lan-1: 192.168.2.0/24 (Lan-1-subnet)

IP Address (optional)

Description:

You can connect a specified subnet to the router.

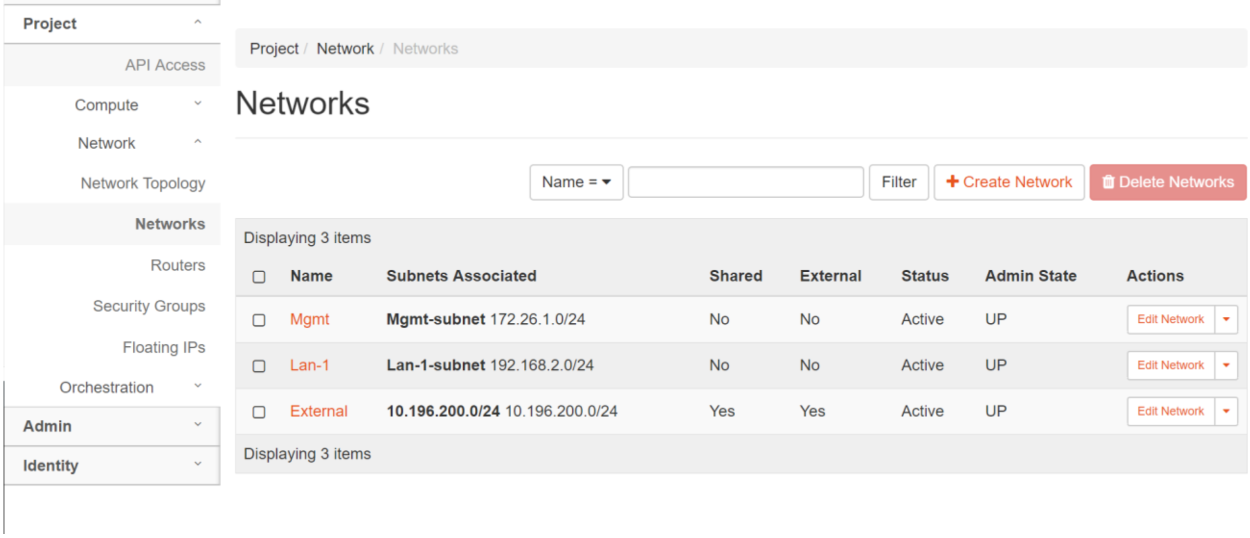
If you don't specify an IP address here, the gateway's IP address of the selected subnet will be used as the IP address of the newly created interface of the router. If the gateway's IP address is in use, you must use a different address which belongs to the selected subnet.

Creating Ports

We create ports with fixed IP addresses for both Lan1 and Mgmt networks. Once the ports are created we bind the Lan-1 ports with the floating IP.

Creating ports in Lan-1 network

1. Navigate to **Project → Networks** and click on **Lan-1**



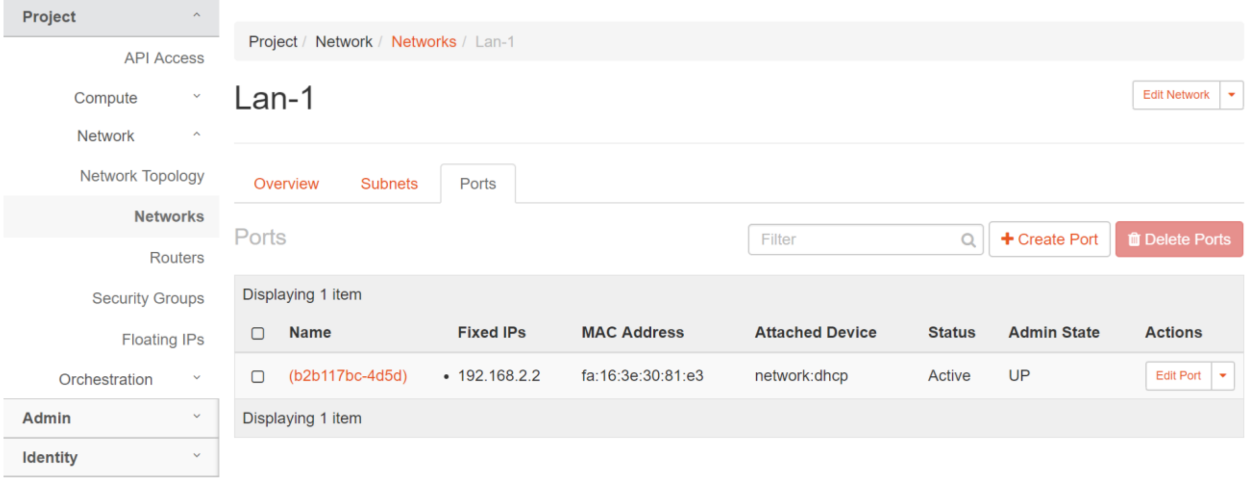
Networks

Displaying 3 items

☐	Name	Subnets Associated	Shared	External	Status	Admin State	Actions
☐	Mgmt	**Mgmt-subnet** 172.26.1.0/24	No	No	Active	UP	Edit Network ▾
☐	Lan-1	**Lan-1-subnet** 192.168.2.0/24	No	No	Active	UP	Edit Network ▾
☐	External	**10.196.200.0/24** 10.196.200.0/24	Yes	Yes	Active	UP	Edit Network ▾

Displaying 3 items

2. Navigate to **Ports → Create Port**



Lan-1

Ports

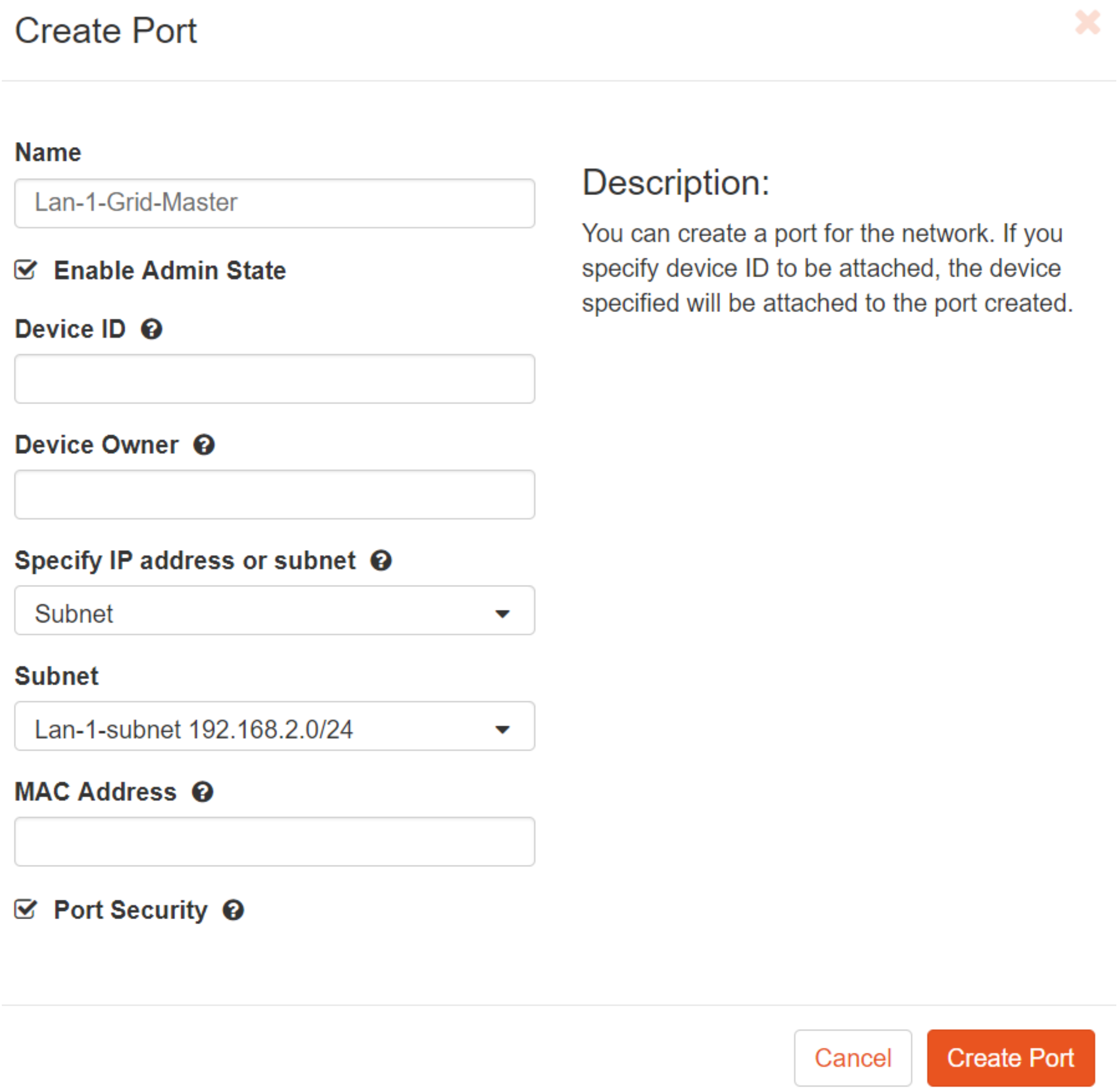
Displaying 1 item

☐	Name	Fixed IPs	MAC Address	Attached Device	Status	Admin State	Actions
☐	(b2b117bc-4d5d)	• 192.168.2.2	fa:16:3e:30:81:e3	network:dhcp	Active	UP	Edit Port ▾

Displaying 1 item

3. Enter a **Name** for this port.
4. In the **Specify IP address or subnet** drop down option select subnet.

5. In the **Subnet** drop down option select Lan-1-subnet 192.168.2.0/24.



Create Port

Name

Lan-1-Grid-Master

☑ Enable Admin State

Device ID

Device Owner

Specify IP address or subnet

Subnet

Subnet

Lan-1-subnet 192.168.2.0/24

MAC Address

☑ Port Security

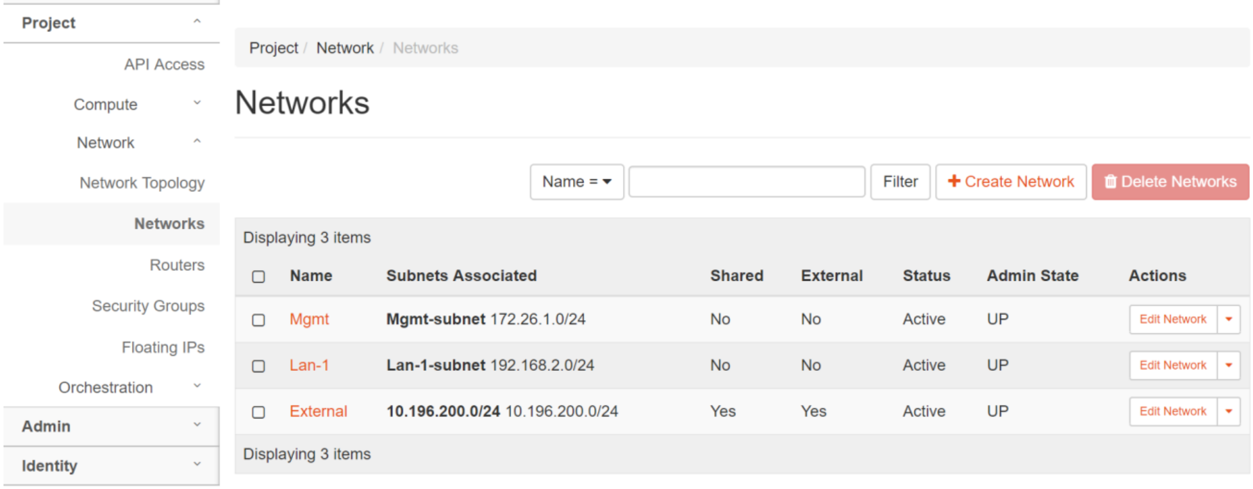
Description:

You can create a port for the network. If you specify device ID to be attached, the device specified will be attached to the port created.

6. Once the port gets created make a note of the port IP address
7. Repeat the same steps to create a port by the name Lan-1-Member for the grid member.

Creating Ports in Mgmt Network

1. Navigate to **Project → Networks** and click on **Mgmt**

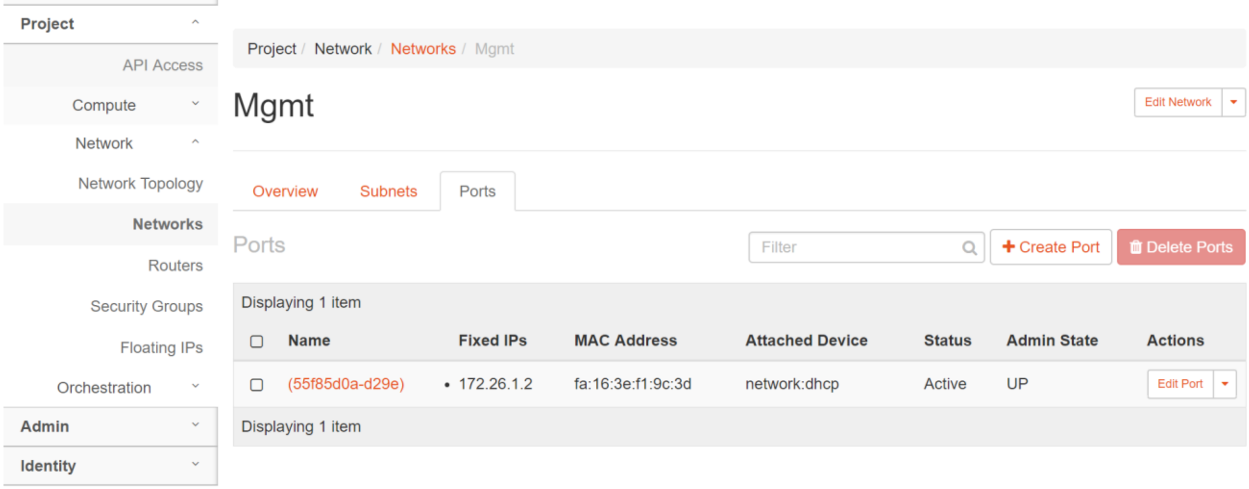


Displaying 3 items

□	Name	Subnets Associated	Shared	External	Status	Admin State	Actions
□	Mgmt	**Mgmt-subnet** 172.26.1.0/24	No	No	Active	UP	Edit Network
□	Lan-1	**Lan-1-subnet** 192.168.2.0/24	No	No	Active	UP	Edit Network
□	External	**10.196.200.0/24** 10.196.200.0/24	Yes	Yes	Active	UP	Edit Network

Displaying 3 items

2. Navigate to **Ports → Create Port**



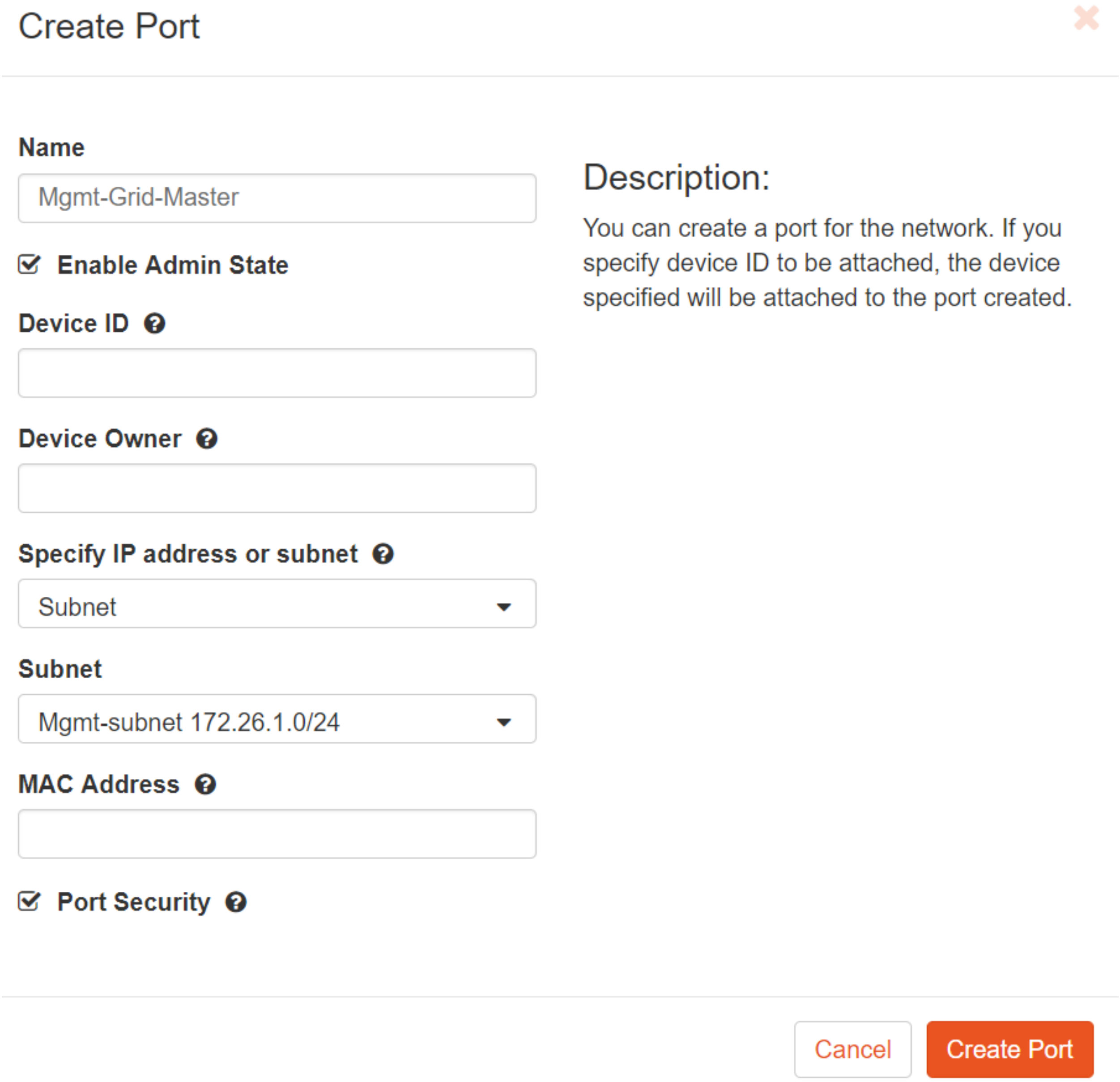
Displaying 1 item

□	Name	Fixed IPs	MAC Address	Attached Device	Status	Admin State	Actions
□	(55f85d0a-d29e)	• 172.26.1.2	fa:16:3e:f1:9c:3d	network:dhcp	Active	UP	Edit Port

Displaying 1 item

3. Enter the **Name** for this port.
4. In the **Specify IP address or subnet** drop down option select subnet.

5. In the **Subnet** drop down option select Mgmt-subnet 172.26.1.0/24.



Create Port

Name: Mgmt-Grid-Master

☑ Enable Admin State

Device ID:

Device Owner:

Specify IP address or subnet: Subnet

Subnet: Mgmt-subnet 172.26.1.0/24

MAC Address:

☑ Port Security

Description:

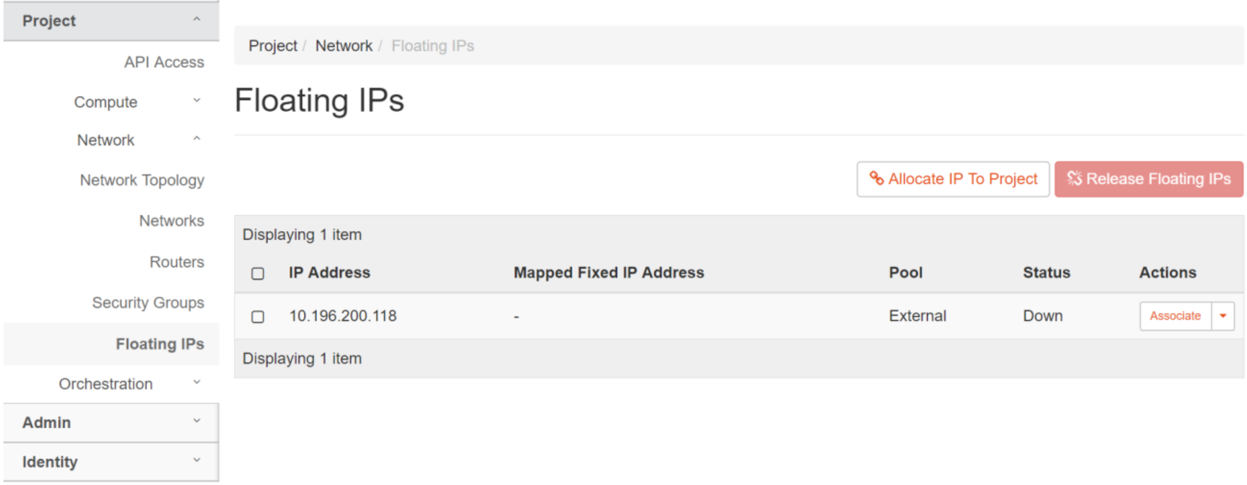
You can create a port for the network. If you specify device ID to be attached, the device specified will be attached to the port created.

6. Once the port gets created make a note of the port IP address
7. Repeat the same steps to create a port by the name Mgmt-Member for the Grid Member.

Allocating and associating floating IPs to the ports present in Lan-1 network

Allocating floating IPs

1. Navigate to Project → **Floating IPs** → **Allocate IP to Project**

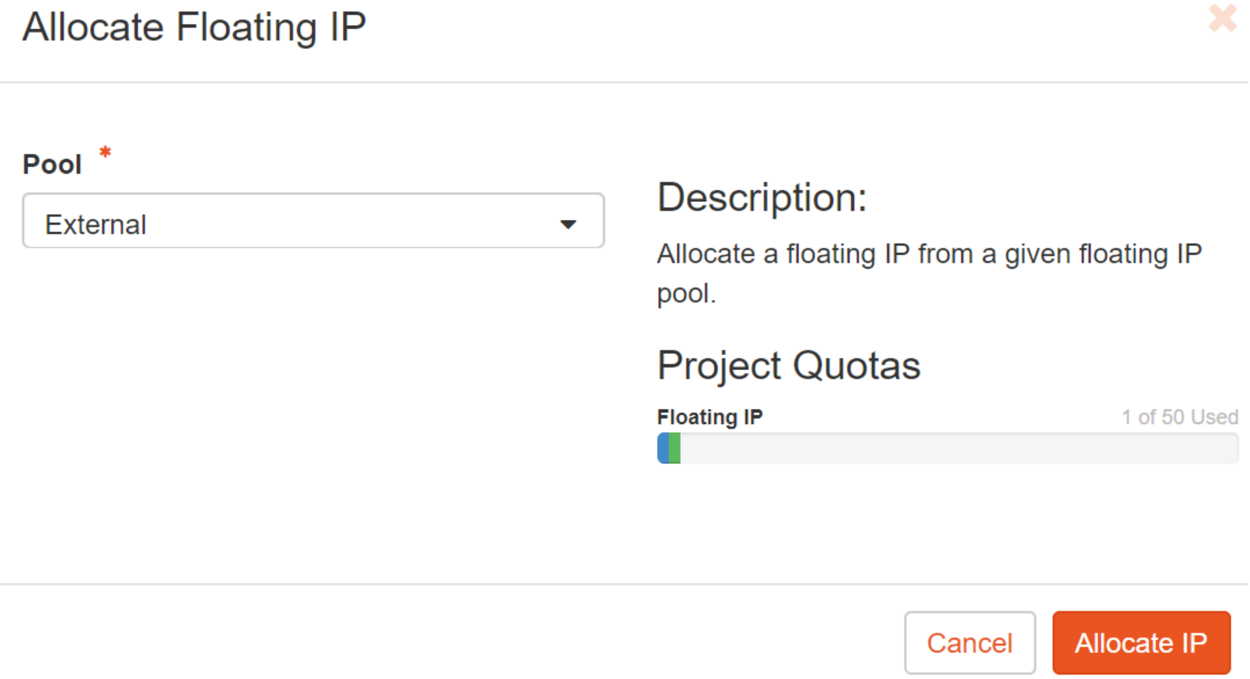


Displaying 1 item

□	IP Address	Mapped Fixed IP Address	Pool	Status	Actions
□	10.196.200.118	-	External	Down	Associate

Displaying 1 item

2. In the **Pool** drop down box select External and click on **Allocate IP**



Pool * External

Description:

Allocate a floating IP from a given floating IP pool.

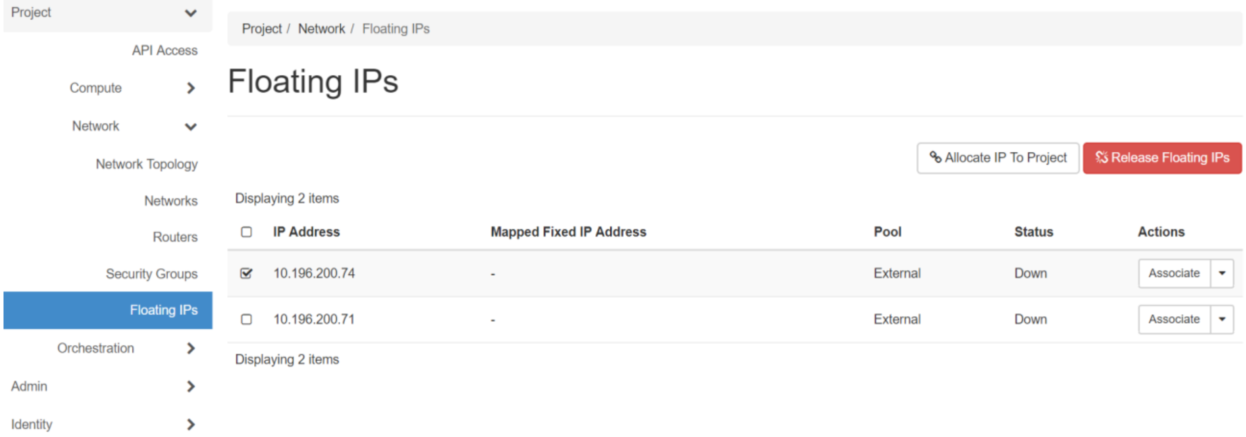
Project Quotas

Floating IP 1 of 50 Used

3. Repeat the same step again to allocate one more floating IP.

Associating floating IP to the Port

1. Navigate to **Project → Floating IPs.** Check the first floating and click on **Associate**.



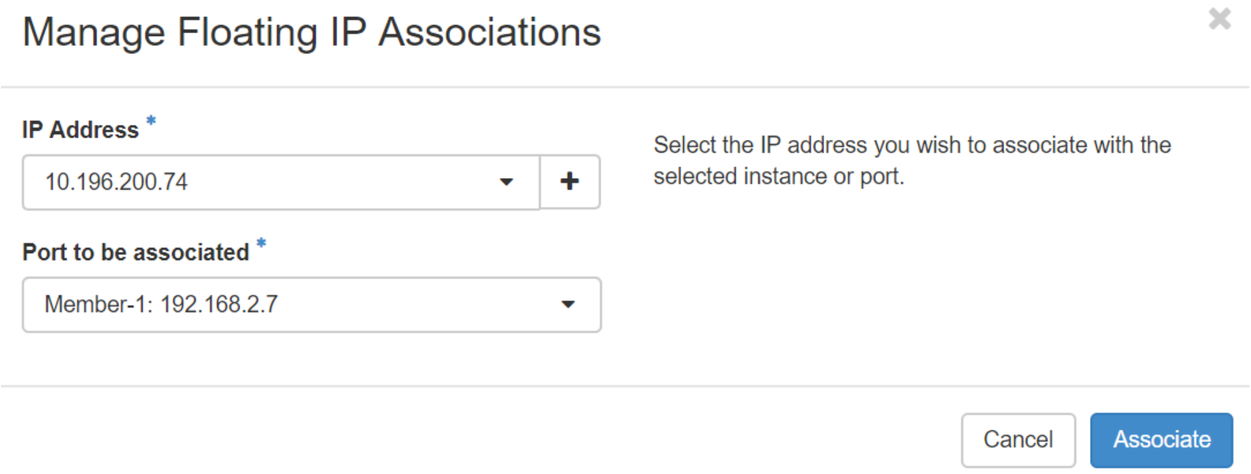
Floating IPs

Displaying 2 items

	IP Address	Mapped Fixed IP Address	Pool	Status	Actions
☑	10.196.200.74	-	External	Down	Associate ▾
□	10.196.200.71	-	External	Down	Associate ▾

Displaying 2 items

2. In **Port to be Associated** drop down box select the Member port and click on associate.



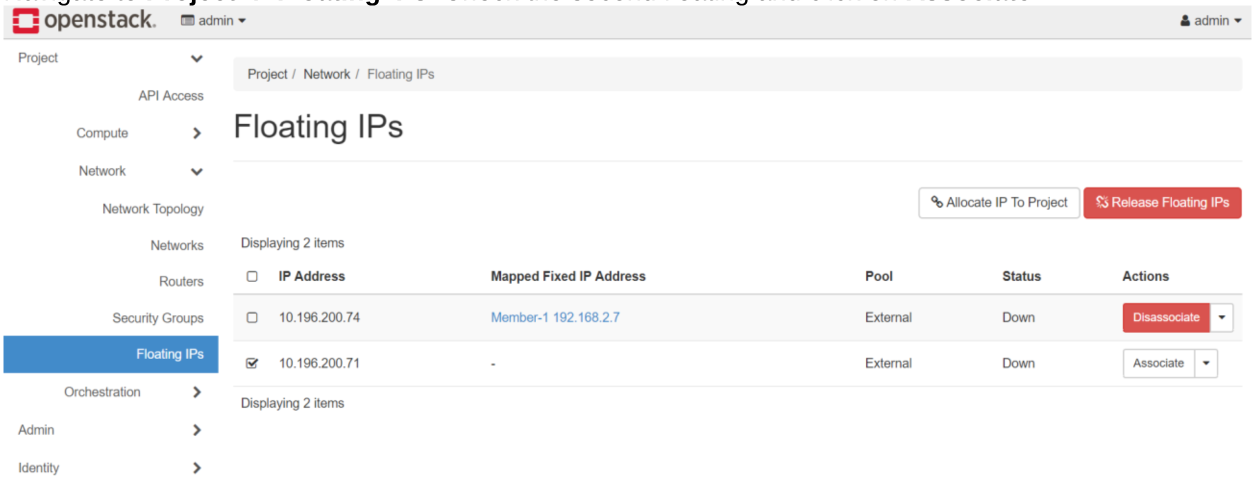
Manage Floating IP Associations

IP Address *: 10.196.200.74

Port to be associated *: Member-1: 192.168.2.7

Select the IP address you wish to associate with the selected instance or port.

3. Navigate to **Project → Floating IPs.** Check the second floating and click on **Associate**.



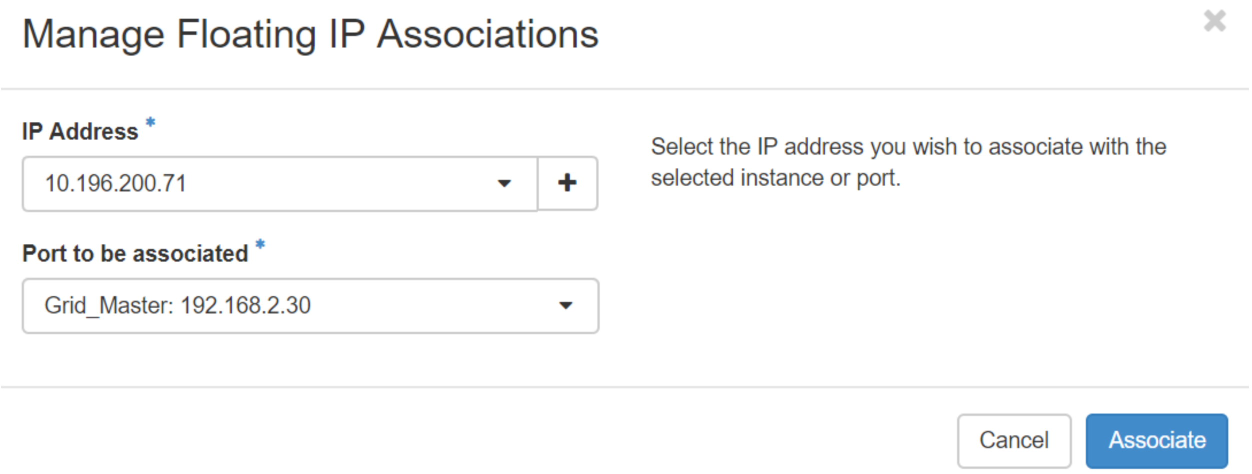
Floating IPs

Displaying 2 items

	IP Address	Mapped Fixed IP Address	Pool	Status	Actions
□	10.196.200.74	Member-1 192.168.2.7	External	Down	Disassociate ▾
☑	10.196.200.71	-	External	Down	Associate ▾

Displaying 2 items

4. In **Port to be Associated** drop down box select the Grid-Master port and click on associate.



Manage Floating IP Associations

IP Address *

10.196.200.71

Port to be associated *

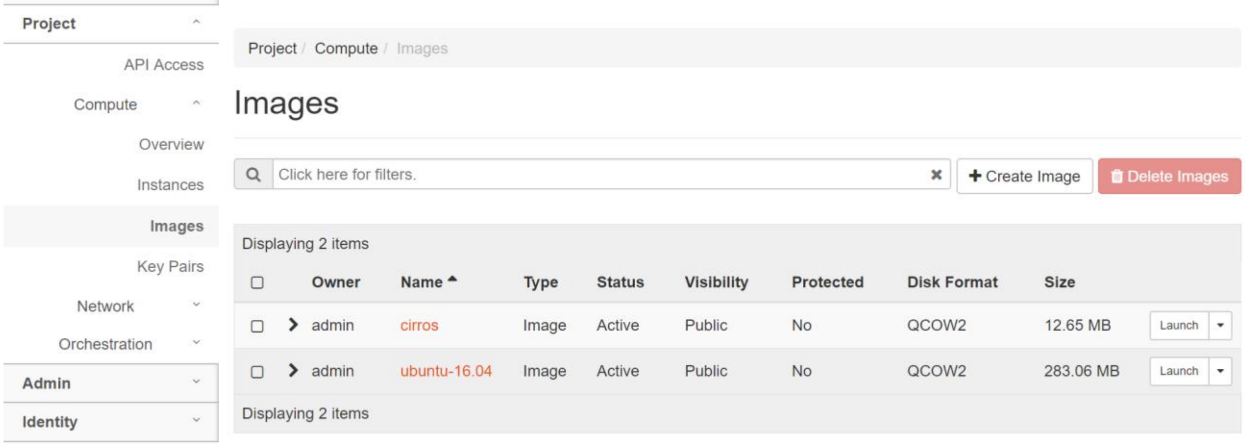
Grid_Master: 192.168.2.30

Select the IP address you wish to associate with the selected instance or port.

Uploading NIOS image to OpenStack

NIOS QCOW2 image can be downloaded from https://support-internal.infoblox.com/support-downloads#

1. Login to OpenStack and navigate to **Project → Compute → Images → Create Image**



Displaying 2 items

☐		Owner	Name ▲	Type	Status	Visibility	Protected	Disk Format	Size	
☐	›	admin	cirros	Image	Active	Public	No	QCOW2	12.65 MB	Launch ▾
☐	›	admin	ubuntu-16.04	Image	Active	Public	No	QCOW2	283.06 MB	Launch ▾

Displaying 2 items

2. Enter the **Image Name.**
3. In the **Format** drop down option select QCOW2 – QEMU Emulator.

4. Click on Browse to select the image.
   In this deployment guide, we use NIOS-810 QCOW2 image.

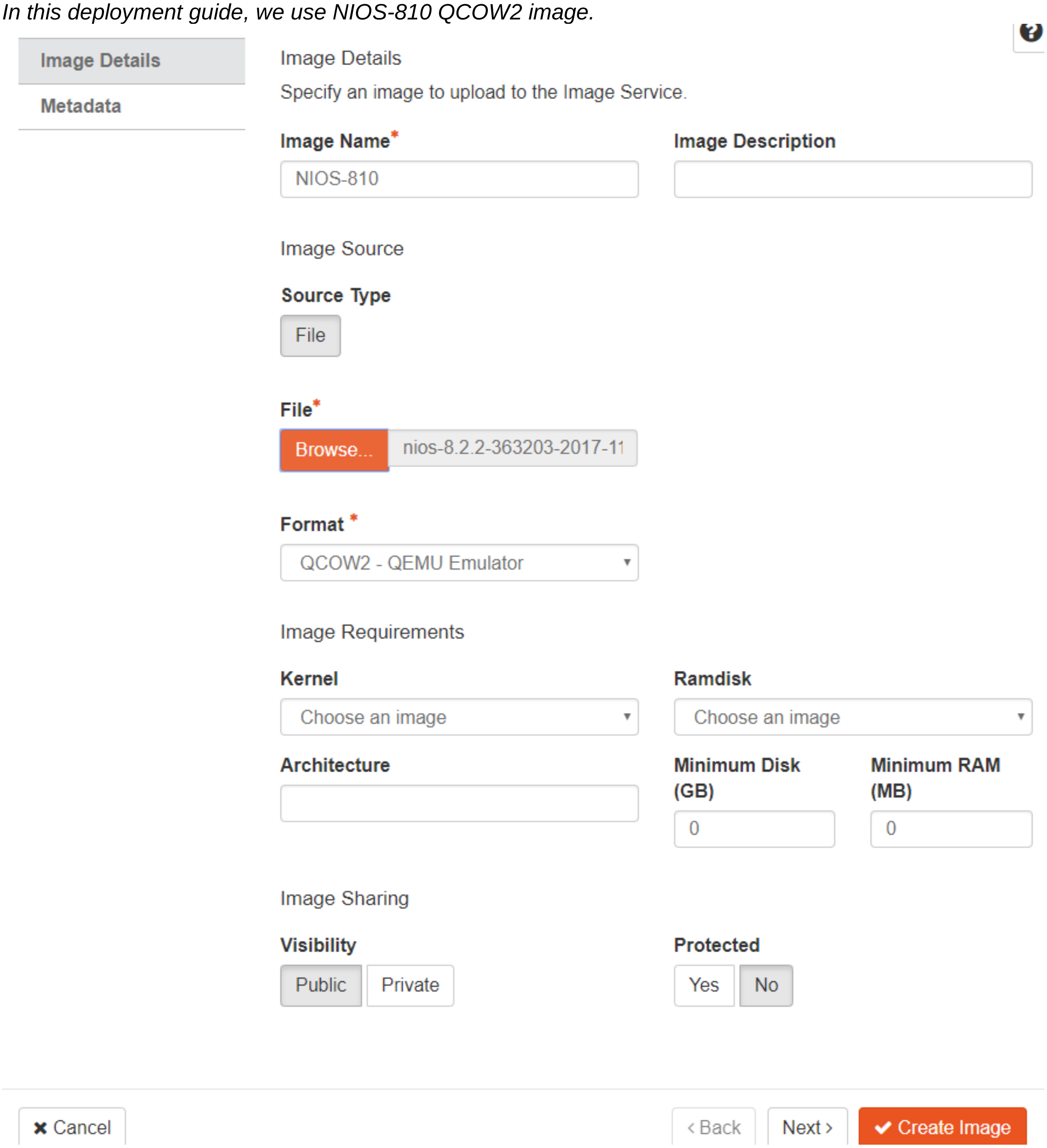


Image Details

Specify an image to upload to the Image Service.

Image Name*: NIOS-810

Image Description:

Image Source

Source Type: File

File*: nios-8.2.2-363203-2017-11

Format *: QCOW2 - QEMU Emulator

Image Requirements

Kernel: Choose an image

Ramdisk: Choose an image

Architecture:

Minimum Disk (GB): 0

Minimum RAM (MB): 0

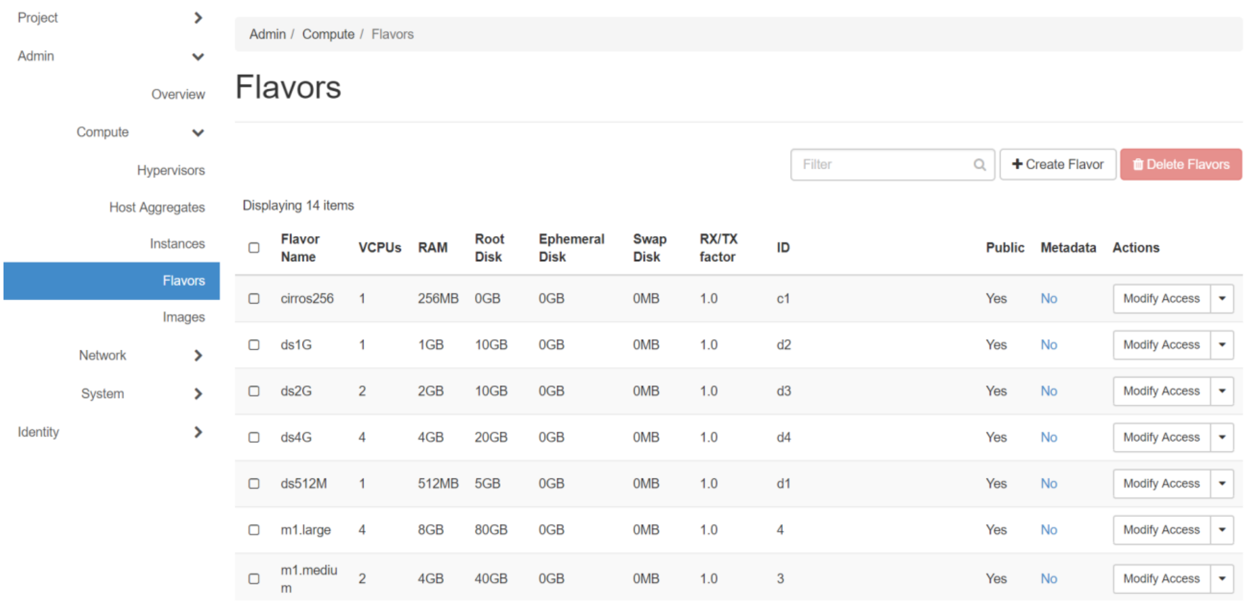
Image Sharing

Visibility: [✓ Public] Private

Protected: Yes [✓ No]

Creating a Flavor

1. Login to OpenStack and navigate to **Admin → Compute → Flavors → Create Flavor**

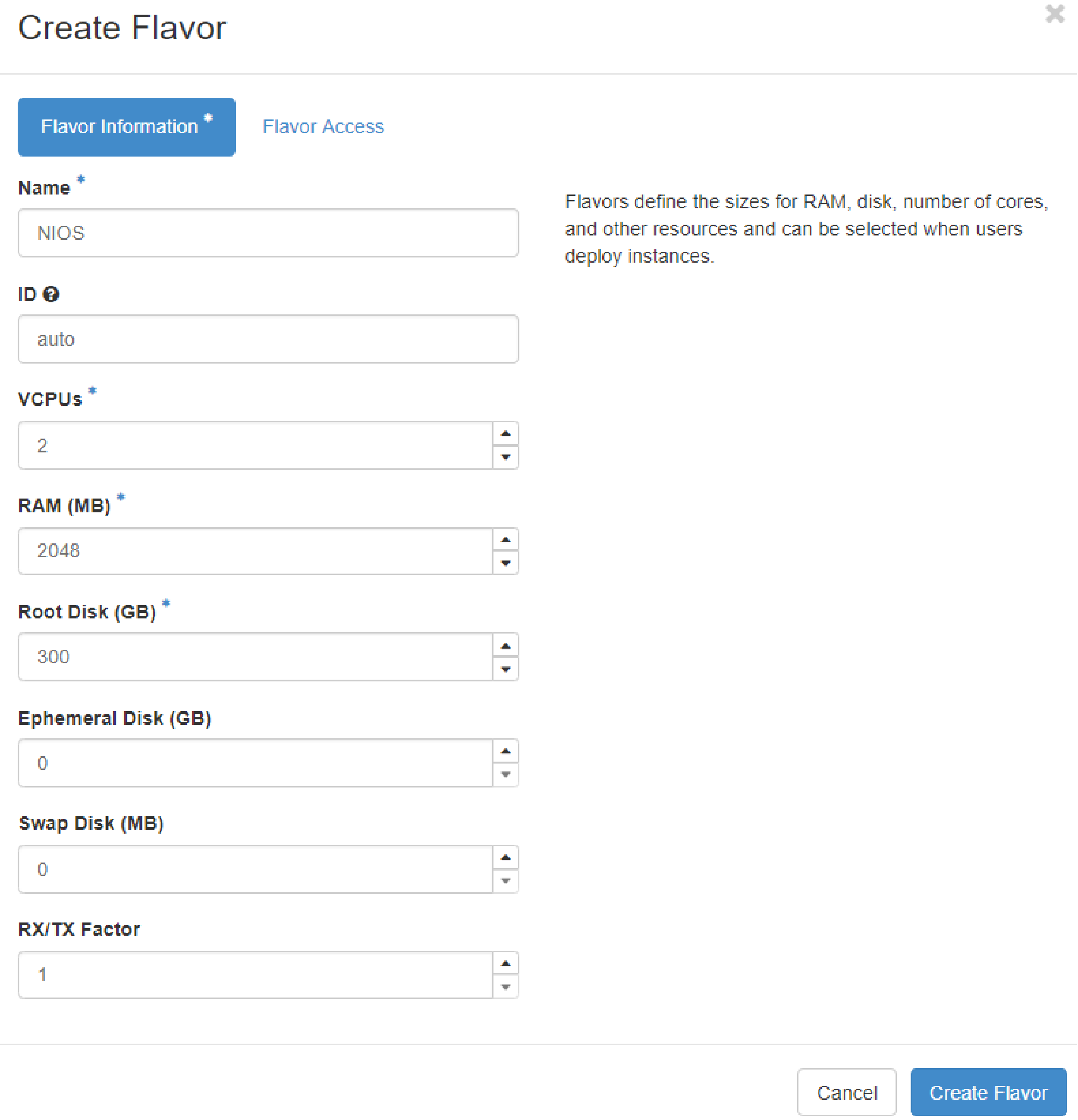


Displaying 14 items

□	Flavor Name	VCPUs	RAM	Root Disk	Ephemeral Disk	Swap Disk	RX/TX factor	ID	Public	Metadata	Actions
□	cirros256	1	256MB	0GB	0GB	0MB	1.0	c1	Yes	No	Modify Access ▾
□	ds1G	1	1GB	10GB	0GB	0MB	1.0	d2	Yes	No	Modify Access ▾
□	ds2G	2	2GB	10GB	0GB	0MB	1.0	d3	Yes	No	Modify Access ▾
□	ds4G	4	4GB	20GB	0GB	0MB	1.0	d4	Yes	No	Modify Access ▾
□	ds512M	1	512MB	5GB	0GB	0MB	1.0	d1	Yes	No	Modify Access ▾
□	m1.large	4	8GB	80GB	0GB	0MB	1.0	4	Yes	No	Modify Access ▾
□	m1.medium	2	4GB	40GB	0GB	0MB	1.0	3	Yes	No	Modify Access ▾

2. Give a **Name** to this flavor.
3. In the **VCPU** option enter 2.
4. In **RAM** option enter 2048

5. In the **Root Disk** option enter 300.



Create Flavor

Name *: NIOS

ID: auto

VCPUs *: 2

RAM (MB) *: 2048

Root Disk (GB) *: 300

Ephemeral Disk (GB): 0

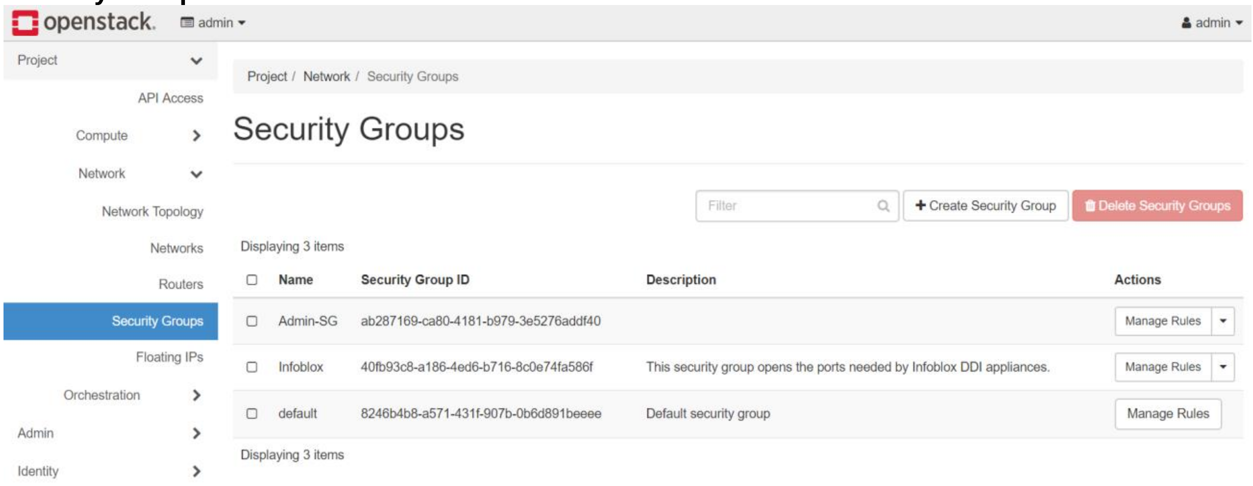
Swap Disk (MB): 0

RX/TX Factor: 1

Flavors define the sizes for RAM, disk, number of cores, and other resources and can be selected when users deploy instances.

Creating a Security Group

1. Login to OpenStack and navigate to **Project → Network → Security Groups → Create Security Group**



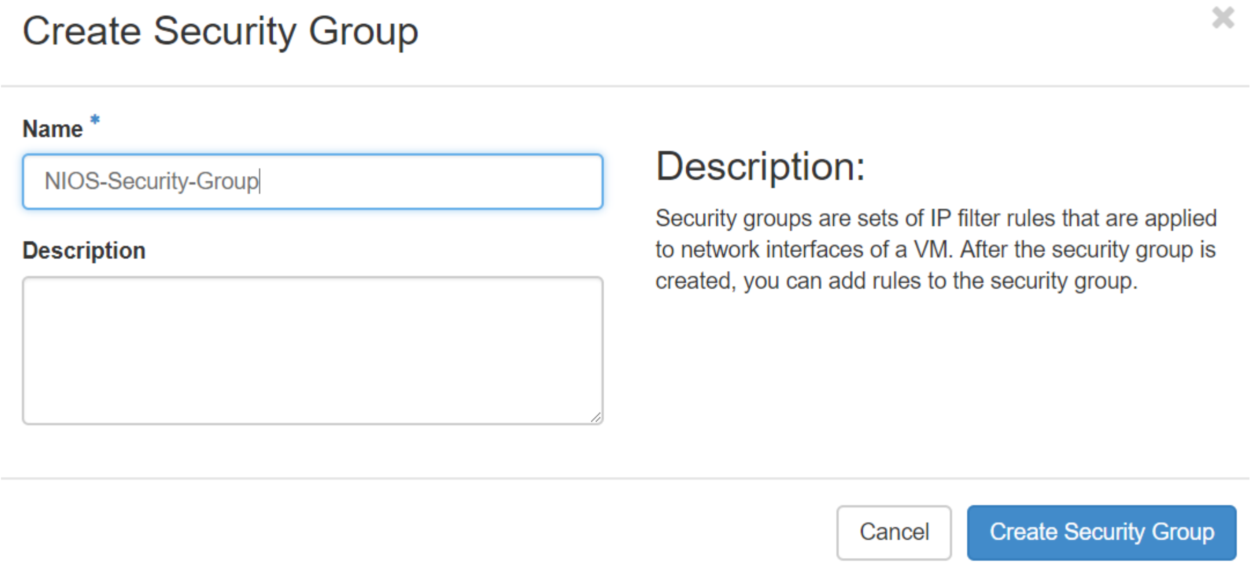
Security Groups

Displaying 3 items

□	Name	Security Group ID	Description	Actions
□	Admin-SG	ab287169-ca80-4181-b979-3e5276addf40		Manage Rules ▾
□	Infoblox	40fb93c8-a186-4ed6-b716-8c0e74fa586f	This security group opens the ports needed by Infoblox DDI appliances.	Manage Rules ▾
□	default	8246b4b8-a571-431f-907b-0b6d891beeee	Default security group	Manage Rules

Displaying 3 items

2. Enter a **Name** for this security group.



Name *

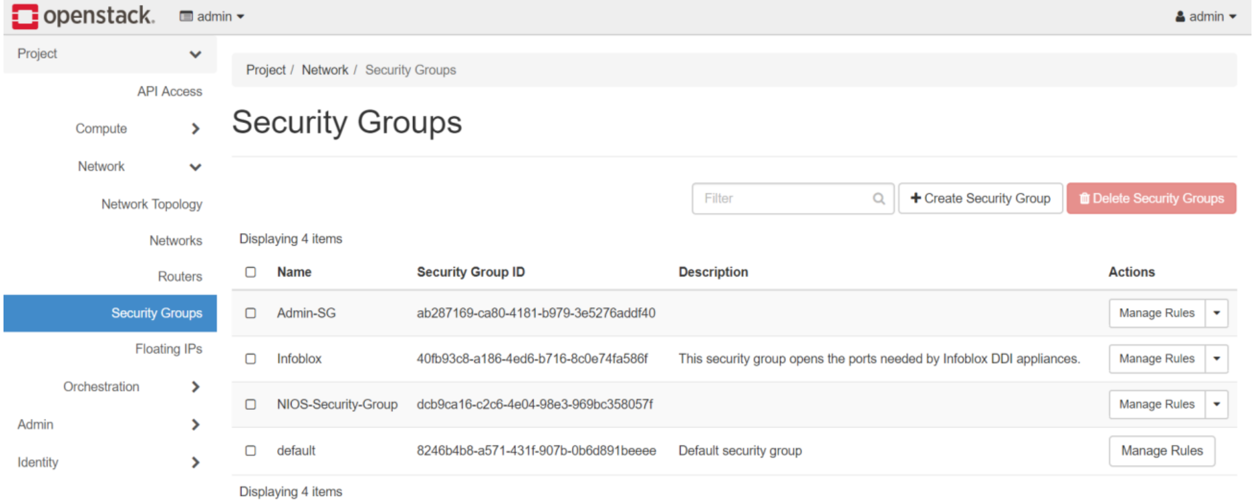
NIOS-Security-Group

Description

Description:

Security groups are sets of IP filter rules that are applied to network interfaces of a VM. After the security group is created, you can add rules to the security group.

3. Once the security group is created click on **Manage Rules**



Project / Network / Security Groups

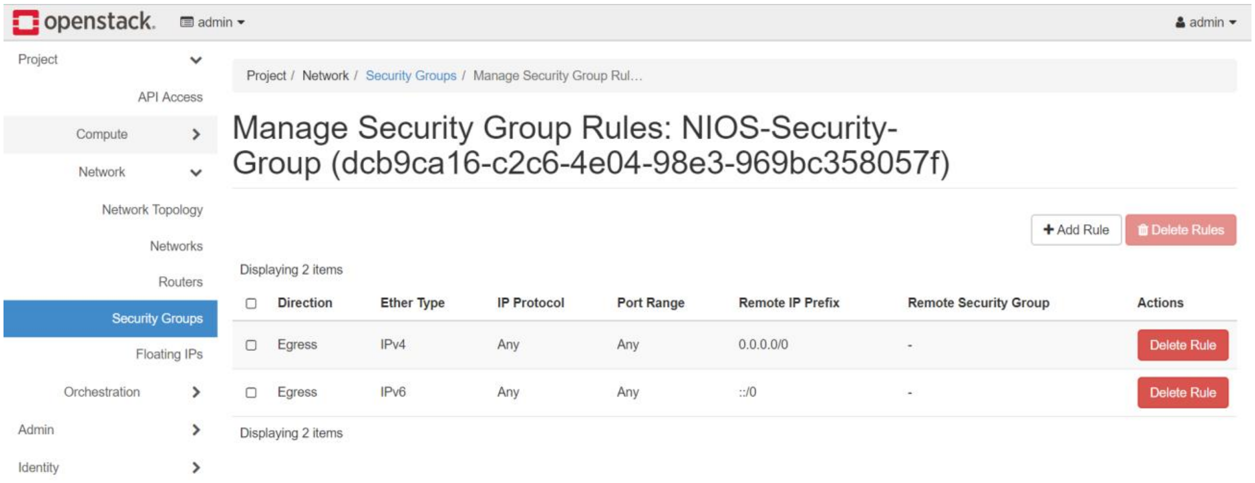
Security Groups

Displaying 4 items

□	Name	Security Group ID	Description	Actions
□	Admin-SG	ab287169-ca80-4181-b979-3e5276addf40		Manage Rules
□	Infoblox	40fb93c8-a186-4ed6-b716-8c0e74fa586f	This security group opens the ports needed by Infoblox DDI appliances.	Manage Rules
□	NIOS-Security-Group	dcb9ca16-c2c6-4e04-98e3-969bc358057f		Manage Rules
□	default	8246b4b8-a571-431f-907b-0b6d891beeee	Default security group	Manage Rules

Displaying 4 items

4. Click on **Add Rule**



Project / Network / Security Groups / Manage Security Group Rul…

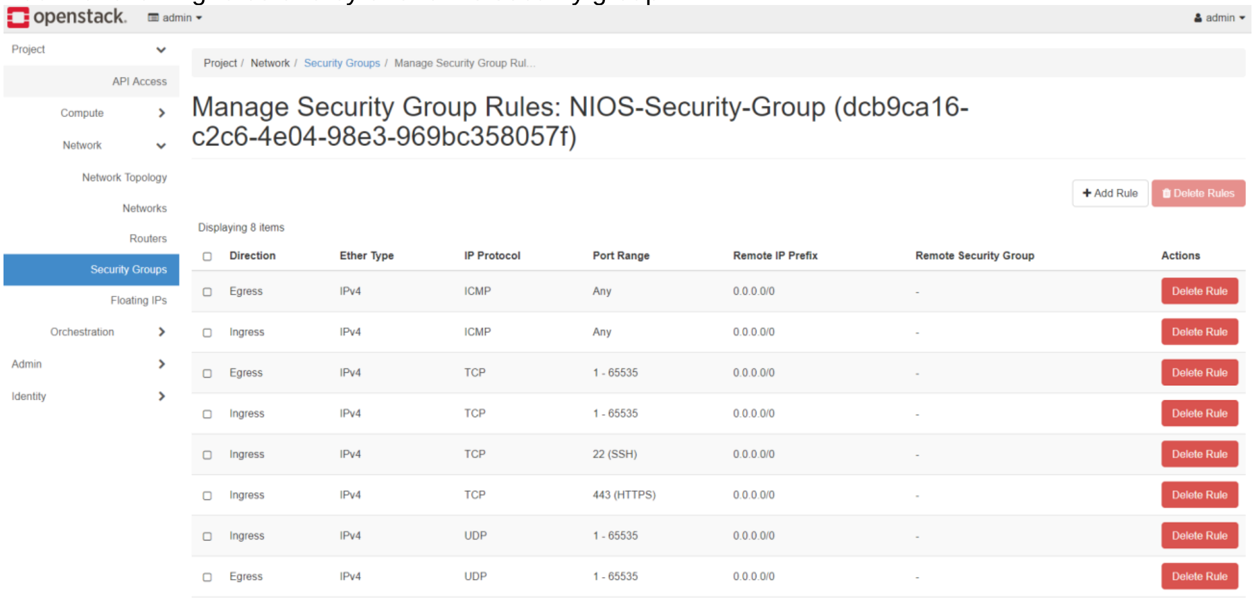
Manage Security Group Rules: NIOS-Security-Group (dcb9ca16-c2c6-4e04-98e3-969bc358057f)

Displaying 2 items

□	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
□	Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
□	Egress	IPv6	Any	Any	::/0	-	Delete Rule

Displaying 2 items

5. Add the following rules one by one to the security group.



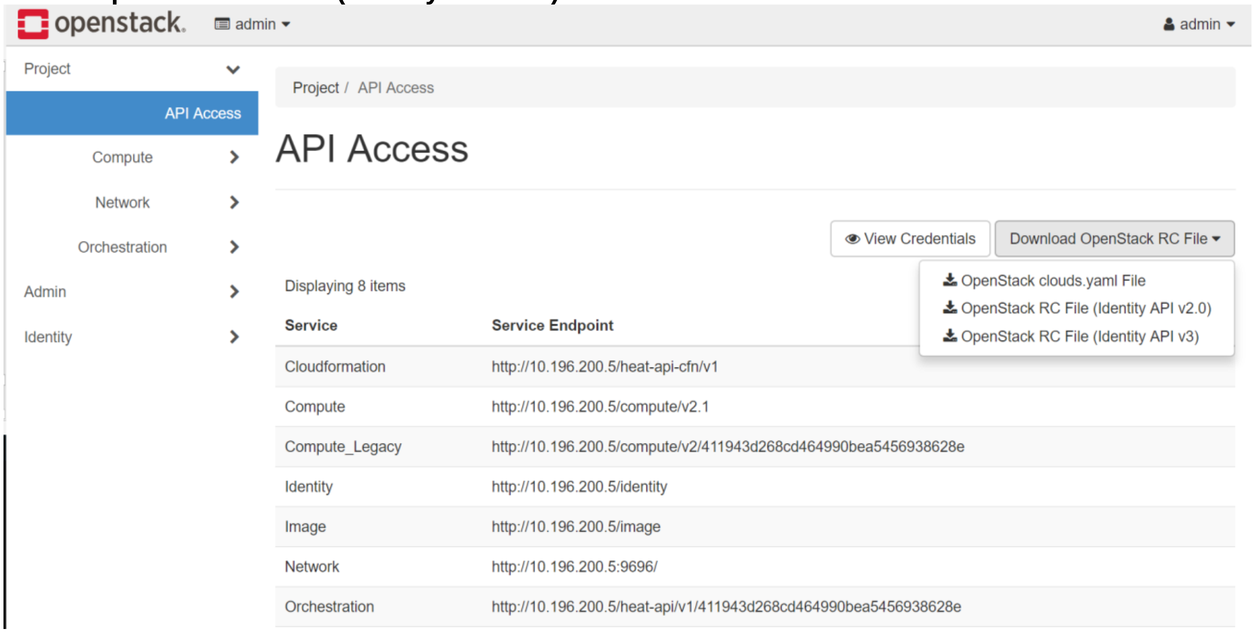
Manage Security Group Rules: NIOS-Security-Group (dcb9ca16-c2c6-4e04-98e3-969bc358057f)

Displaying 8 items

□	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
□	Egress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
□	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
□	Egress	IPv4	TCP	1 - 65535	0.0.0.0/0	-	Delete Rule
□	Ingress	IPv4	TCP	1 - 65535	0.0.0.0/0	-	Delete Rule
□	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule
□	Ingress	IPv4	TCP	443 (HTTPS)	0.0.0.0/0	-	Delete Rule
□	Ingress	IPv4	UDP	1 - 65535	0.0.0.0/0	-	Delete Rule
□	Egress	IPv4	UDP	1 - 65535	0.0.0.0/0	-	Delete Rule

Downloading the admin-openrc.sh

1. Login to the OpenStack and navigate to **Project → API Access → Download OpenStack RC File → OpenStack RC File (Identity API v2.0)**


openstack.
admin ▾
admin ▾
Project
API Access
Compute
Network
Orchestration
Admin
Identity
Project / API Access
View Credentials
Download OpenStack RC File ▾
OpenStack clouds.yaml File
OpenStack RC File (Identity API v2.0)
OpenStack RC File (Identity API v3)

API Access

Displaying 8 items

Service	Service Endpoint
Cloudformation	http://10.196.200.5/heat-api-cfn/v1
Compute	http://10.196.200.5/compute/v2.1
Compute_Legacy	http://10.196.200.5/compute/v2/411943d268cd464990bea5456938628e
Identity	http://10.196.200.5/identity
Image	http://10.196.200.5/image
Network	http://10.196.200.5:9696/
Orchestration	http://10.196.200.5/heat-api/v1/411943d268cd464990bea5456938628e

2. OpenStack RC file shows up as **admin-openrc** once downloaded. If you are using windows machine you can use WinSCP software to move admin-openrc file to the ansible machine.

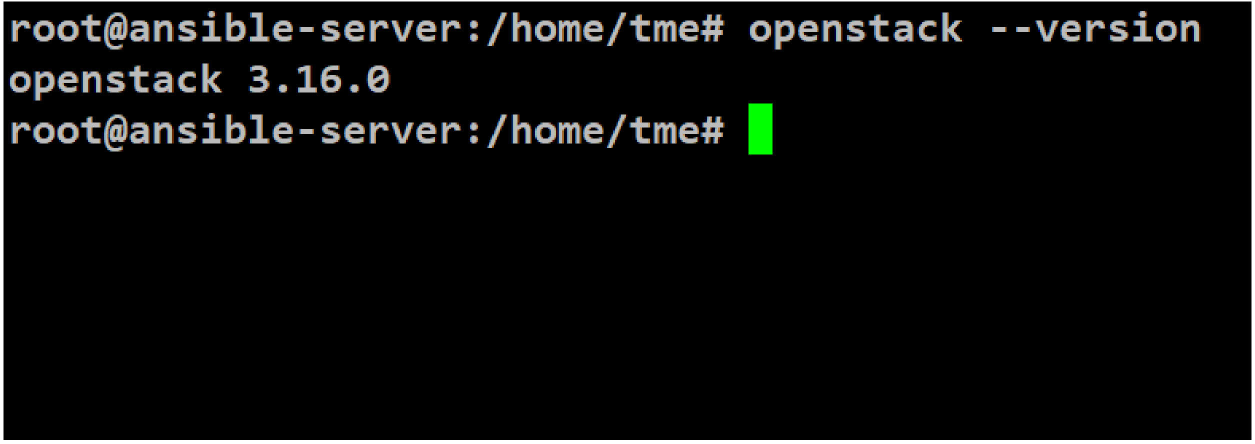
Setting up the Ansible machine

1. Login to the Linux machine in which you have installed ansible.
2. Run following commands one by one to install openstack client.

```
apt-get install python-pip
```

```
pip install python-openstackclient
apt-get install python-openstackclient
```

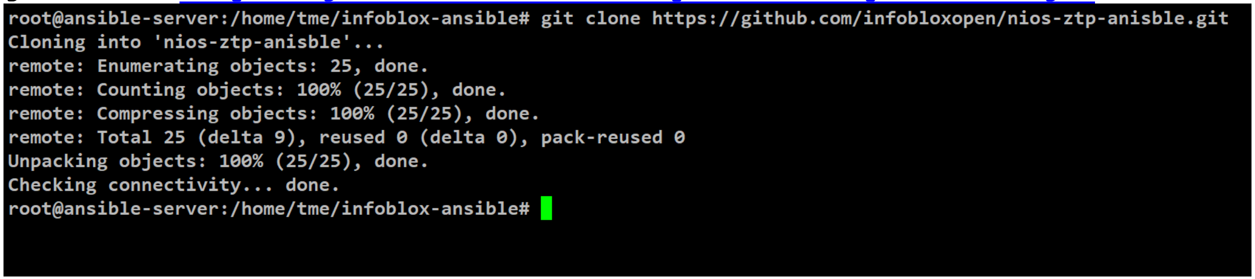
3. Check for the python-openstack client version by running `openstack –version`



```
root@ansible-server:/home/tme# openstack --version
openstack 3.16.0
root@ansible-server:/home/tme#
```

Note: python-openstackclient recently got updated. Its ok if the version shows as 3.16.1 or 3.16.2.

4. Download the ansible playbooks by running
`git clone https://github.com/infobloxopen/nios-ztp-anisble.git`

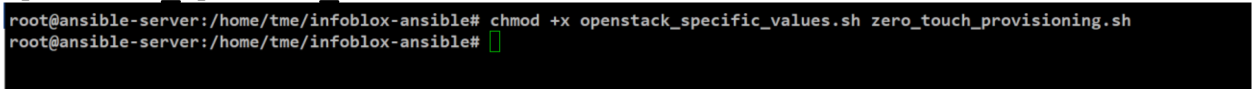


```
root@ansible-server:/home/tme/infoblox-ansible# git clone https://github.com/infobloxopen/nios-ztp-anisble.git
Cloning into 'nios-ztp-anisble'...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 25 (delta 9), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (25/25), done.
Checking connectivity... done.
root@ansible-server:/home/tme/infoblox-ansible#
```

5. Once the clone process is over you will see folder nios-ztp-anisble
6. cd to this folder. You should see following files.
   - `auth1.json`
   - `auth.json`
   - `deploy_grid_master_and_member.yml`
   - `join_member.yml`
   - `openstack_specific_values.sh`
   - `pre_provision_member.yml`
   - `wait_for_grid_master.yml`
   - `zero_touch_provisioning.sh`

   Note: Folder will have LICENSE, README.md file as well. These files can be ignored

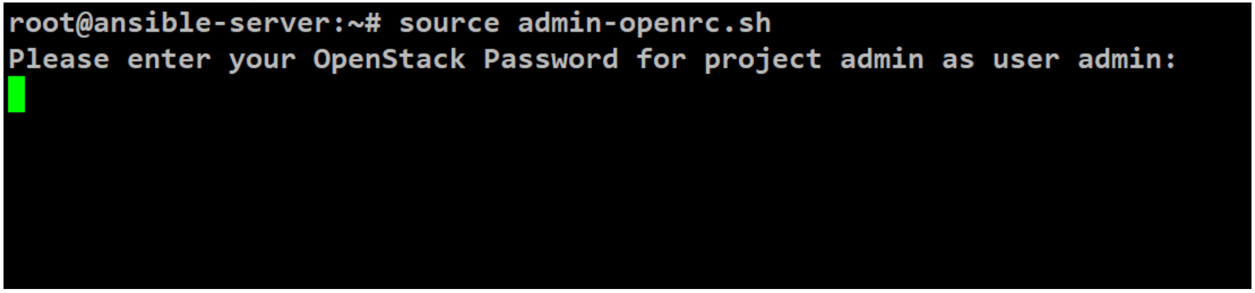
7. Change the permission of the `openstack_specific_values.sh` and `zero_touch_provisioning.sh` file make them executable by running following command.
```
chmod +x openstack_specific_values.sh
openstack_specific_values.sh
```



```
root@ansible-server:/home/tme/infoblox-ansible# chmod +x openstack_specific_values.sh zero_touch_provisioning.sh
root@ansible-server:/home/tme/infoblox-ansible#
```

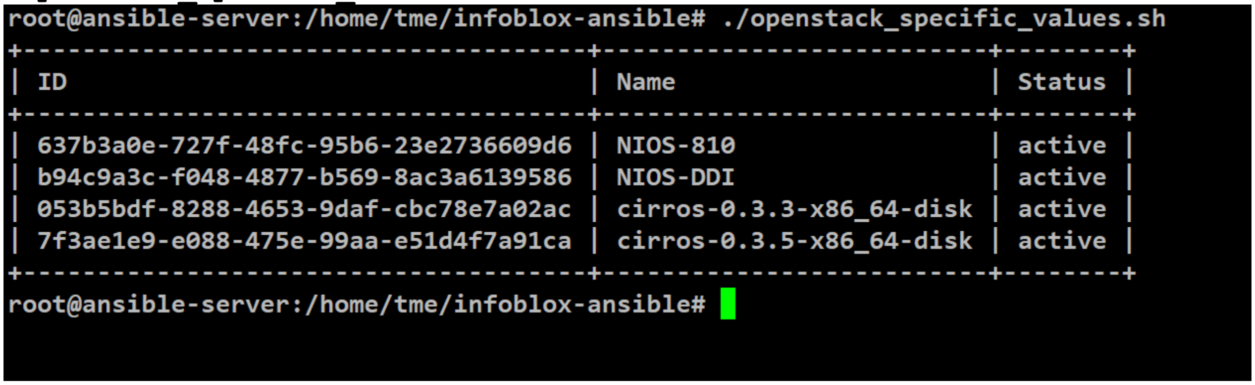
8. Move the admin-openrc file to this folder. Source it by running following command. You will be prompted to enter your OpenStack password.

source admin-openrc



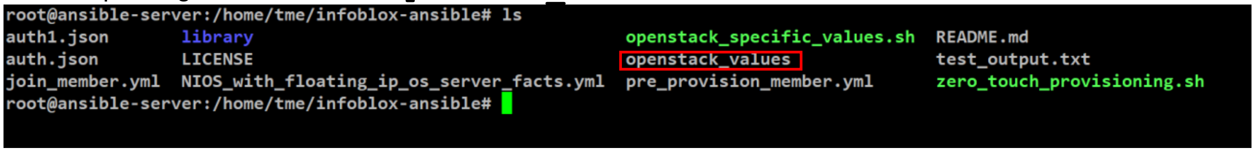
```
root@ansible-server:~# source admin-openrc.sh
Please enter your OpenStack Password for project admin as user admin:
```

9. To get parameters required to run ansible playbook execute **openstack_specific_values.sh** script by running following command
./openstack_specific_values.sh



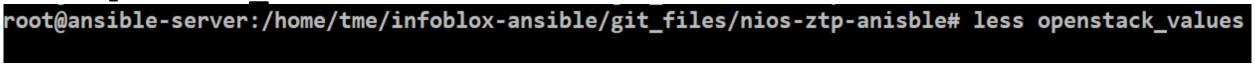
```
root@ansible-server:/home/tme/infoblox-ansible# ./openstack_specific_values.sh
+--------------------------------------+--------------------------+--------+
| ID                                   | Name                     | Status |
+--------------------------------------+--------------------------+--------+
| 637b3a0e-727f-48fc-95b6-23e2736609d6 | NIOS-810                 | active |
| b94c9a3c-f048-4877-b569-8ac3a6139586 | NIOS-DDI                 | active |
| 053b5bdf-8288-4653-9daf-cbc78e7a02ac | cirros-0.3.3-x86_64-disk | active |
| 7f3ae1e9-e088-475e-99aa-e51d4f7a91ca | cirros-0.3.5-x86_64-disk | active |
+--------------------------------------+--------------------------+--------+
root@ansible-server:/home/tme/infoblox-ansible#
```

10. This script will generate a file → **openstack_values**



```
root@ansible-server:/home/tme/infoblox-ansible# ls
auth1.json      library                                openstack_specific_values.sh  README.md
auth.json       LICENSE                                openstack_values              test_output.txt
join_member.yml NIOS_with_floating_ip_os_server_facts.yml pre_provision_member.yml   zero_touch_provisioning.sh
root@ansible-server:/home/tme/infoblox-ansible#
```

11. **openstack_values** file has list of all the parameters which we will need to run the ansible playbook for NIOS deployment on OpenStack.
12. Verify the contents of **openstack_values** file by running following command
less openstack_values



```
root@ansible-server:/home/tme/infoblox-ansible/git_files/nios-ztp-anisble# less openstack_values
```

13. To deploy NIOS using ansible we will need IDs for following parameters

Parameter Name	Description	Sample
image	QCOW2 image for Grid Master and Member deployment	637b3a0e-727f-48fc-95b6-23e2736609d6
flavor	Resource details (RAM, CPU and Hard Disk) for Grid Master and Member	92aff88f-839f-479b-9156-4be2a3c3594b
nics	Port details for the Mgmt network	port-name=84da4e8c-3af9-41b3-bfc0-927095e420cd,port-name=0f12f865-9c56-4cc0-9f85-83047e240163
	Port details for the Lan-1 network	
security_groups	Outgoing and Incoming traffic policies for Grid-Master and Member	ab287169-ca80-4181-b979-3e5276addf40

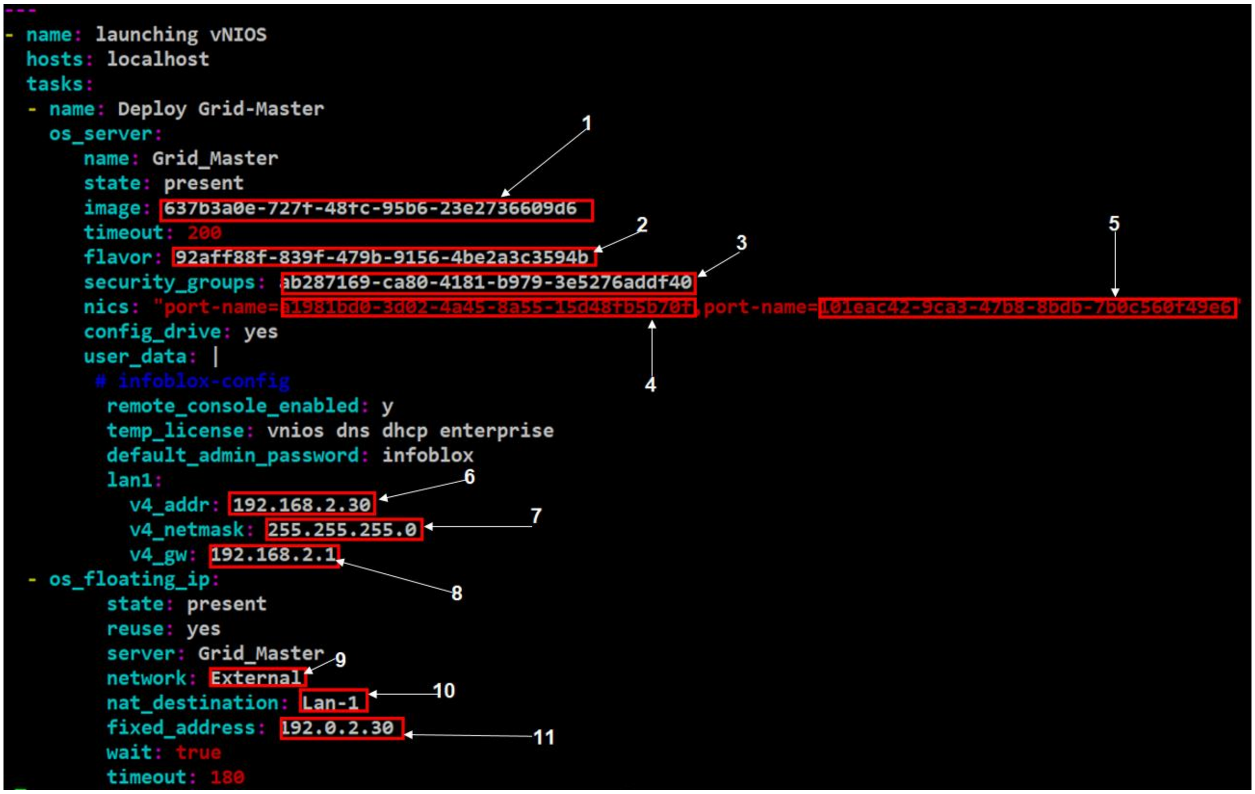
Editing Ansible Playbooks

Edit the deploy_grid_master_and_member.yml file using vim editor

Editing **`Deploy Grid-Master`** section
Replace the values in red box with the values you will get from **`openstack_values`** file

```
1. Enter the image id
2. Enter the flavor id
3. Enter the security_groups id
4. Enter the Grid-Master-Mgmt port id
5. Enter the Grid-Master-Lan1 port id
6. Enter the IP address of the Lan1 port for Grid master
7. Enter the Lan-1 subnet in CIDR
8. Enter the Lan-1 default gateway
9. Enter the name of the external network
10. Enter the name of Lan-1 network. (Name which you gave while
     creating it in Horizon)
11. Enter the IP address of the Grid-Master-lan1 port.
```

Note: Do not change the order of port IDs. First port id will be of Mgmt port and second port id will be of Lan-1 port.



```
---
- name: launching vNIOS
  hosts: localhost
  tasks:
  - name: Deploy Grid-Master
    os_server:
       name: Grid_Master
       state: present
       image: 637b3a0e-727f-48fc-95b6-23e2736609d6
       timeout: 200
       flavor: 92aff88f-839f-479b-9156-4be2a3c3594b
       security_groups: ab287169-ca80-4181-b979-3e5276addf40
       nics: "port-name=a1981bd0-3d02-4a45-8a55-15d48fb5b70f,port-name=101eac42-9ca3-47b8-8bdb-7b0c560f49e6"
       config_drive: yes
       user_data: |
        # infoblox-config
         remote_console_enabled: y
         temp_license: vnios dns dhcp enterprise
         default_admin_password: infoblox
         lan1:
           v4_addr: 192.168.2.30
           v4_netmask: 255.255.255.0
           v4_gw: 192.168.2.1
  - os_floating_ip:
         state: present
         reuse: yes
         server: Grid_Master
         network: External
         nat_destination: Lan-1
         fixed_address: 192.0.2.30
         wait: true
         timeout: 180
```

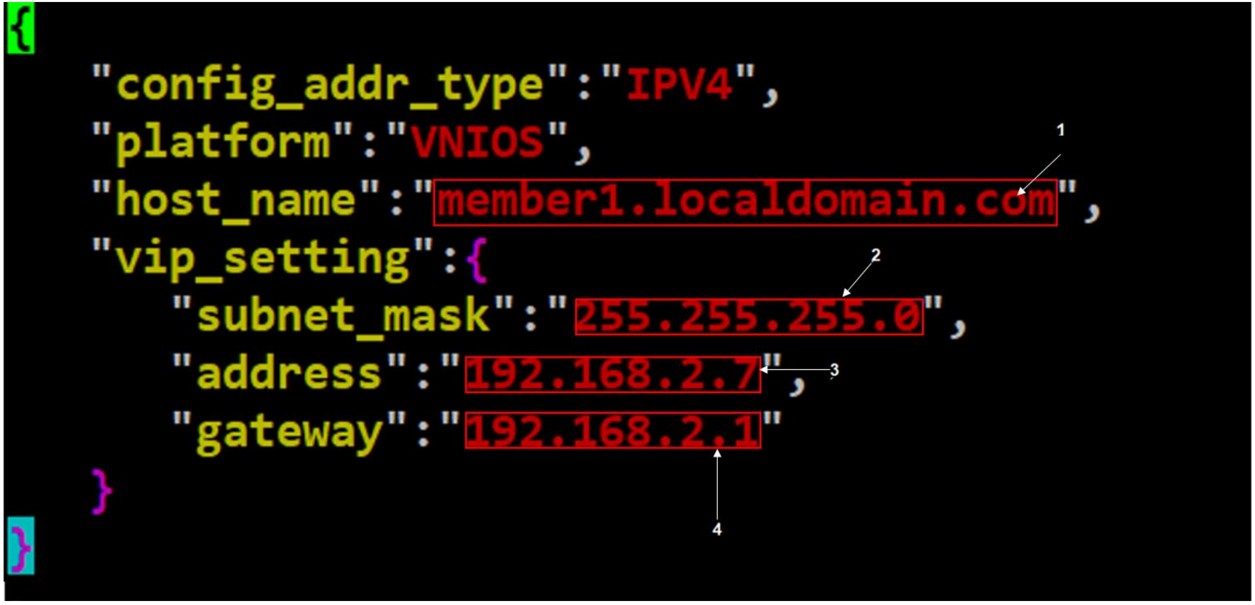
Similarly edit the Deploy Member-1 section of the file.

Editing the auth.json file using vim editor

1. **`auth.json`** file contains information about the member which will be pre-provisioned in the grid in json format. This file is used pre_provision_member.yml file.

2. Edit the values which are highlighted in red box.

    1. **Enter the name of the member**
    2. **Enter the Lan-1 subnet**
    3. **Enter the Lan-1 IP address of the member.(This will be the IP address of the port created in Lan-1 network for member)**
    4. **Enter the Lan-1 network gateway.**

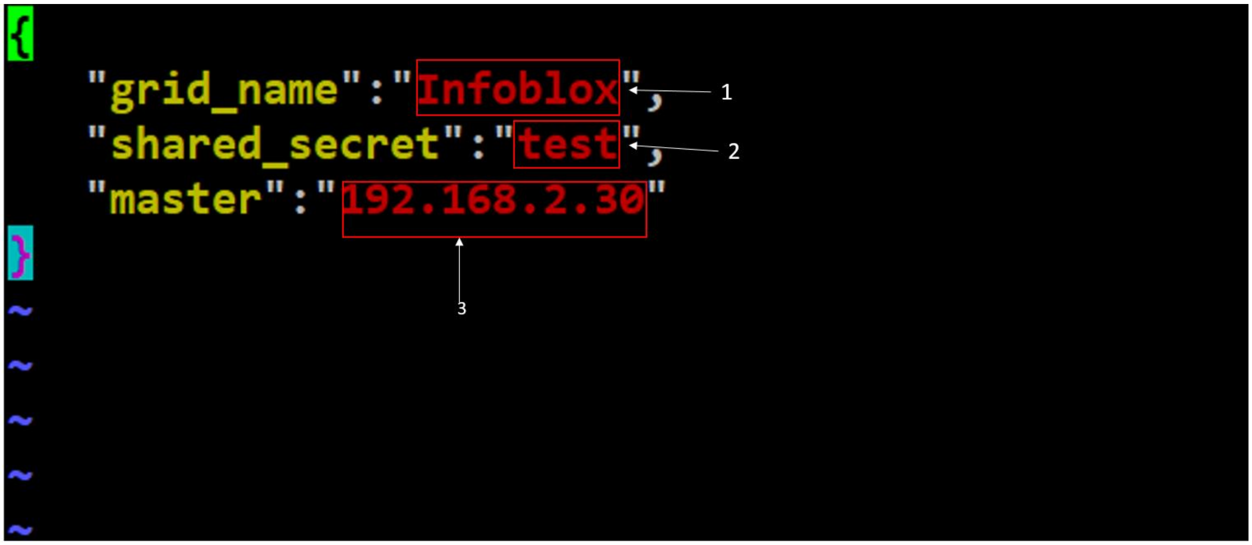


```
{
   "config_addr_type":"IPV4",
   "platform":"VNIOS",
   "host_name":"member1.localdomain.com",
   "vip_setting":{
      "subnet_mask":"255.255.255.0",
      "address":"192.168.2.7",
      "gateway":"192.168.2.1"
   }
}
```

Editing auth1.json file using vim editor

1. **`auth1.json`** file contains information about the Grid-master in json format. This file is used in join_member.yml file.
2. Edit the values which are highlighted in red box.

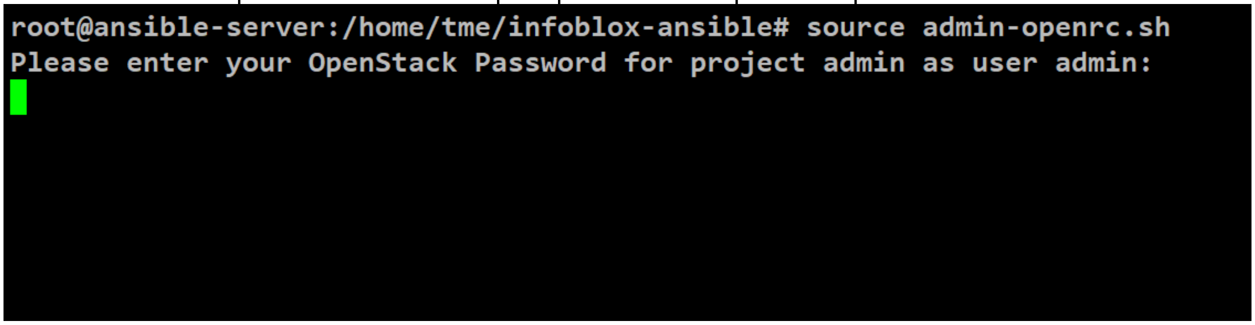
    1. **Enter the name of the Grid. Default value is Infoblox**
    2. **Enter the shared secret. Default value is test**
    3. **Enter the Grid-Master Lan-1 port IP address.**



```
{
    "grid_name":"Infoblox",
    "shared_secret":"test",
    "master":"192.168.2.30"
}
~
~
~
~
~
```

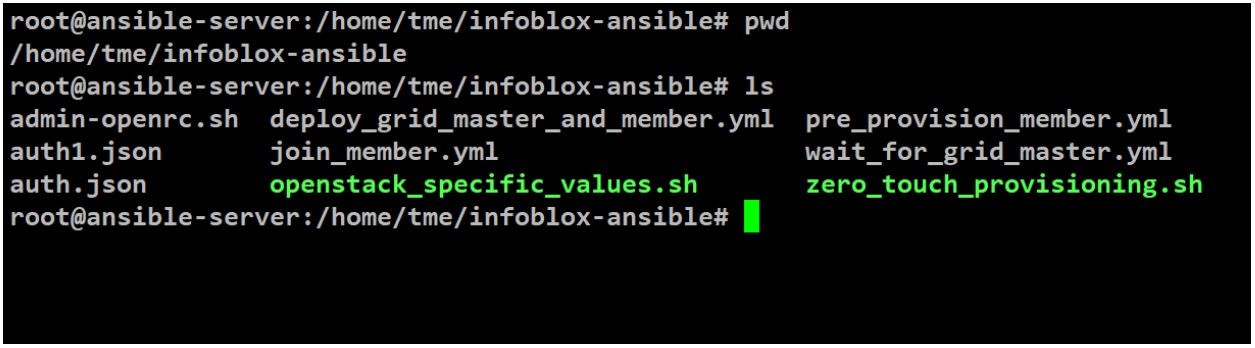
Invoking zero_touch_provisioning.sh script

1. Source the admin-openrc file. You will be prompted to enter OpenStack password.



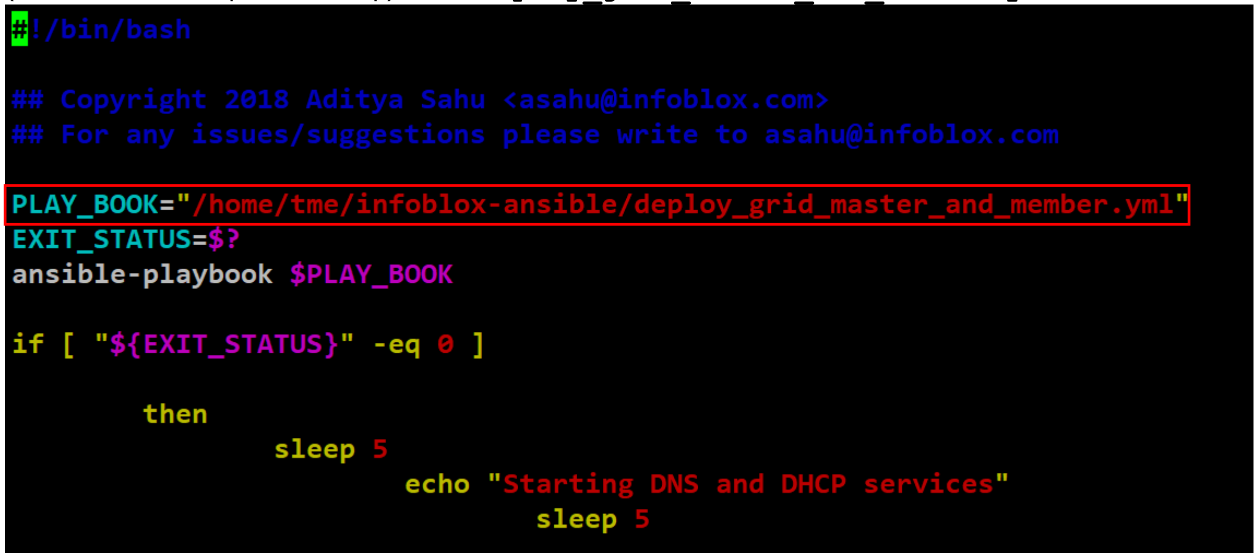
```
root@ansible-server:/home/tme/infoblox-ansible# source admin-openrc.sh
Please enter your OpenStack Password for project admin as user admin:
```

2. Run **pwd** command to find out the current directory where all the playbooks are present



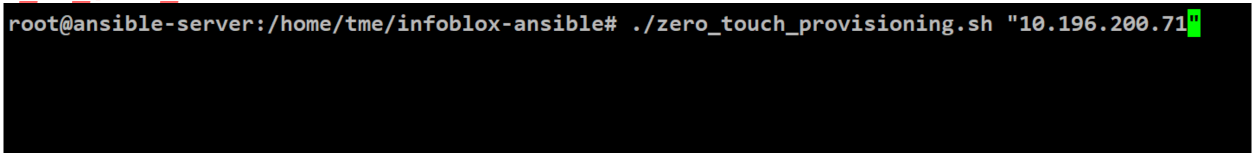
```
root@ansible-server:/home/tme/infoblox-ansible# pwd
/home/tme/infoblox-ansible
root@ansible-server:/home/tme/infoblox-ansible# ls
admin-openrc.sh  deploy_grid_master_and_member.yml  pre_provision_member.yml
auth1.json       join_member.yml                    wait_for_grid_master.yml
auth.json        openstack_specific_values.sh       zero_touch_provisioning.sh
root@ansible-server:/home/tme/infoblox-ansible#
```

3. Edit the **zero_touch_provisioning.sh** file using vim editor and update the absolute path (obtained from the previous step) of the **deploy_grid_master_and_member.yml file**



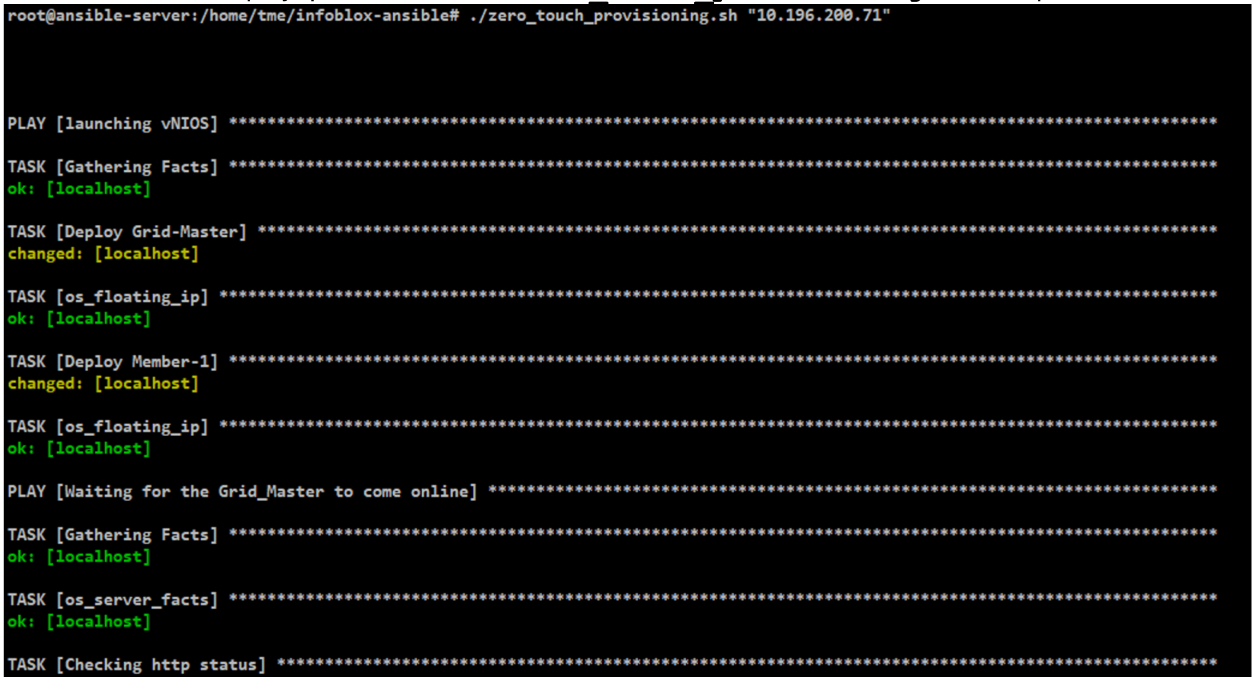
```
#!/bin/bash

## Copyright 2018 Aditya Sahu <asahu@infoblox.com>
## For any issues/suggestions please write to asahu@infoblox.com

PLAY_BOOK="/home/tme/infoblox-ansible/deploy_grid_master_and_member.yml"
EXIT_STATUS=$?
ansible-playbook $PLAY_BOOK

if [ "${EXIT_STATUS}" -eq 0 ]

        then
                sleep 5
                        echo "Starting DNS and DHCP services"
                                sleep 5
```

4. Invoke the **zero_touch_provisioning.sh** script and pass the floating IP, associated with the Lan-1 port of Grid-Master as a positional parameter using following command.
   ./ zero_touch_provisioning.sh “*floating_ip_associated_with_Lan-1_of_grid_master*”



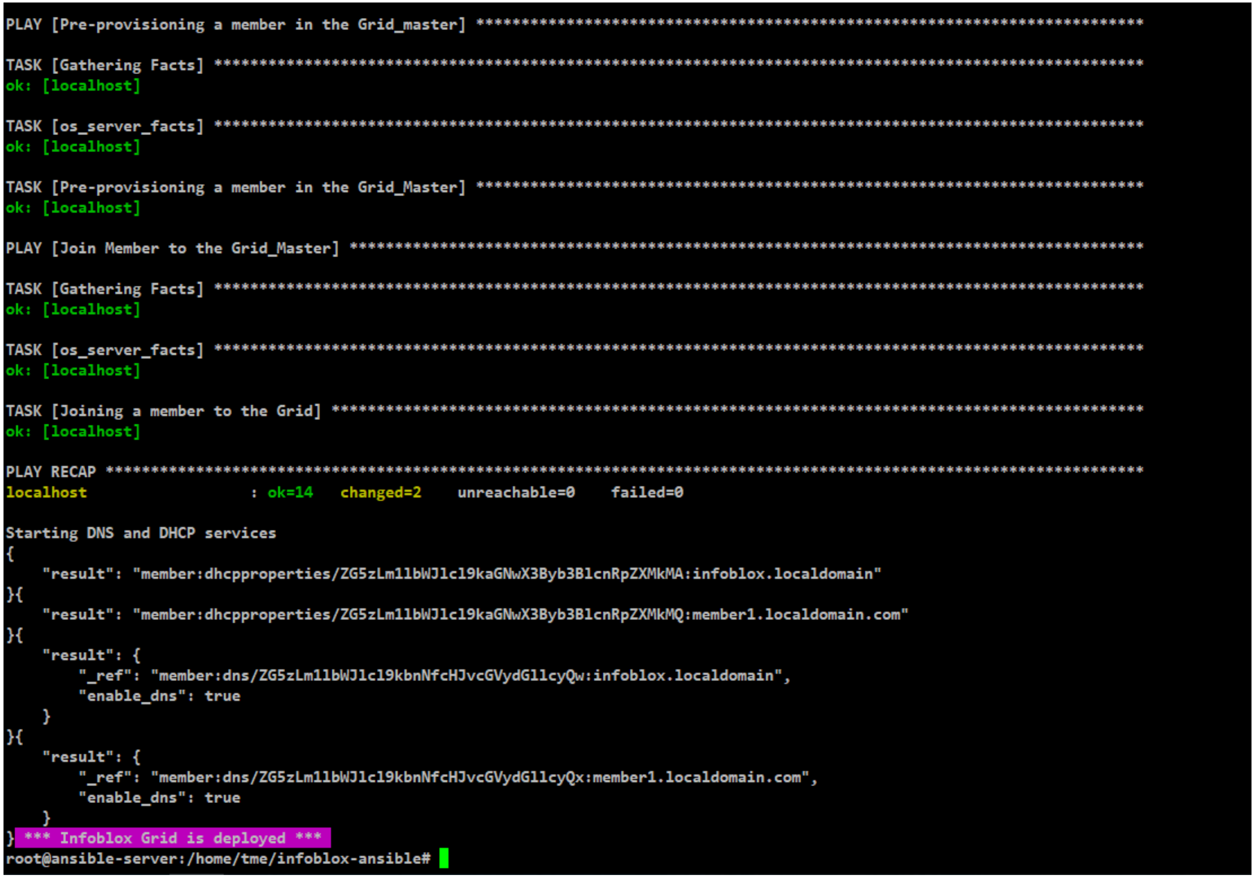
```
root@ansible-server:/home/tme/infoblox-ansible# ./zero_touch_provisioning.sh "10.196.200.71"
```

5. Ansible will start the play, post execution of **zero_touch_provisioning.sh** script.



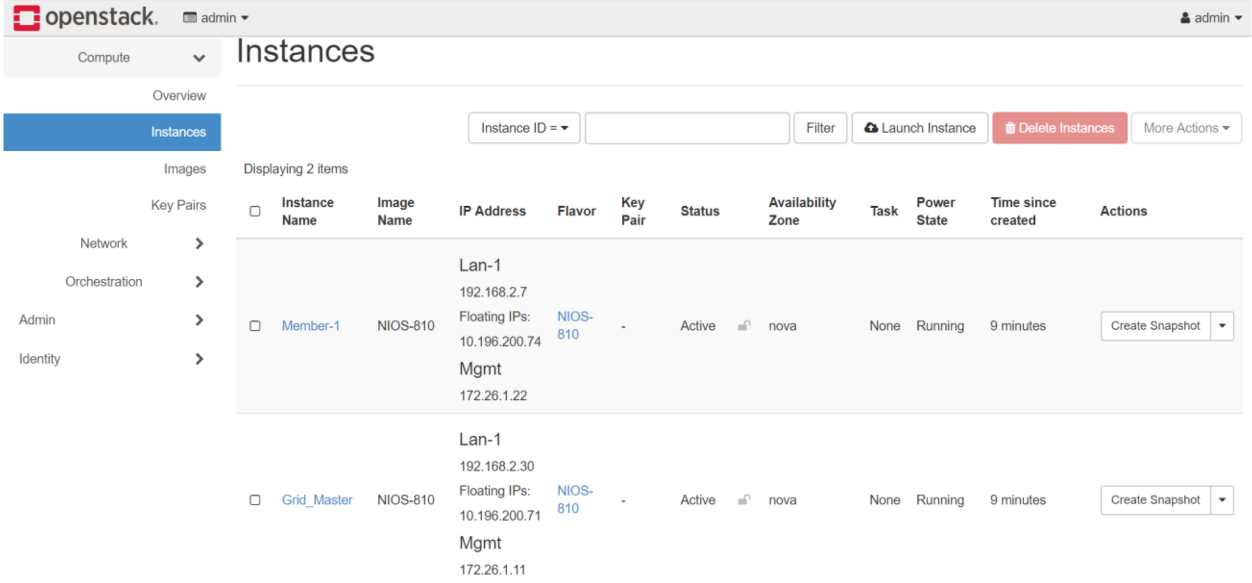
```
root@ansible-server:/home/tme/infoblox-ansible# ./zero_touch_provisioning.sh "10.196.200.71"



PLAY [launching vNIOS] *******************************************************************************************************

TASK [Gathering Facts] *******************************************************************************************************
ok: [localhost]

TASK [Deploy Grid-Master] ****************************************************************************************************
changed: [localhost]

TASK [os_floating_ip] ********************************************************************************************************
ok: [localhost]

TASK [Deploy Member-1] *******************************************************************************************************
changed: [localhost]

TASK [os_floating_ip] ********************************************************************************************************
ok: [localhost]

PLAY [Waiting for the Grid_Master to come online] ****************************************************************************

TASK [Gathering Facts] *******************************************************************************************************
ok: [localhost]

TASK [os_server_facts] *******************************************************************************************************
ok: [localhost]

TASK [Checking http status] **************************************************************************************************
```

6. Post successful execution of all the playbooks you will get a prompt that **`Infoblox Grid is deployed.`**



```
PLAY [Pre-provisioning a member in the Grid_master] **************************************************************************

TASK [Gathering Facts] *******************************************************************************************************
ok: [localhost]

TASK [os_server_facts] *******************************************************************************************************
ok: [localhost]

TASK [Pre-provisioning a member in the Grid_Master] **************************************************************************
ok: [localhost]

PLAY [Join Member to the Grid_Master] ****************************************************************************************

TASK [Gathering Facts] *******************************************************************************************************
ok: [localhost]

TASK [os_server_facts] *******************************************************************************************************
ok: [localhost]

TASK [Joining a member to the Grid] ******************************************************************************************
ok: [localhost]

PLAY RECAP *******************************************************************************************************************
localhost                  : ok=14   changed=2    unreachable=0    failed=0

Starting DNS and DHCP services
{
    "result": "member:dhcpproperties/ZG5zLm1lbWJlcl9kaGNwX3Byb3BlcnRpZXMkMA:infoblox.localdomain"
}{
    "result": "member:dhcpproperties/ZG5zLm1lbWJlcl9kaGNwX3Byb3BlcnRpZXMkMQ:member1.localdomain.com"
}{
    "result": {
        "_ref": "member:dns/ZG5zLm1lbWJlcl9kbnNfcHJvcGVydGllcyQw:infoblox.localdomain",
        "enable_dns": true
    }
}{
    "result": {
        "_ref": "member:dns/ZG5zLm1lbWJlcl9kbnNfcHJvcGVydGllcyQx:member1.localdomain.com",
        "enable_dns": true
    }
} *** Infoblox Grid is deployed ***
root@ansible-server:/home/tme/infoblox-ansible#
```

7. Login to horizon dashboard and verify that Grid-Master and Member have been deployed and in a running state.

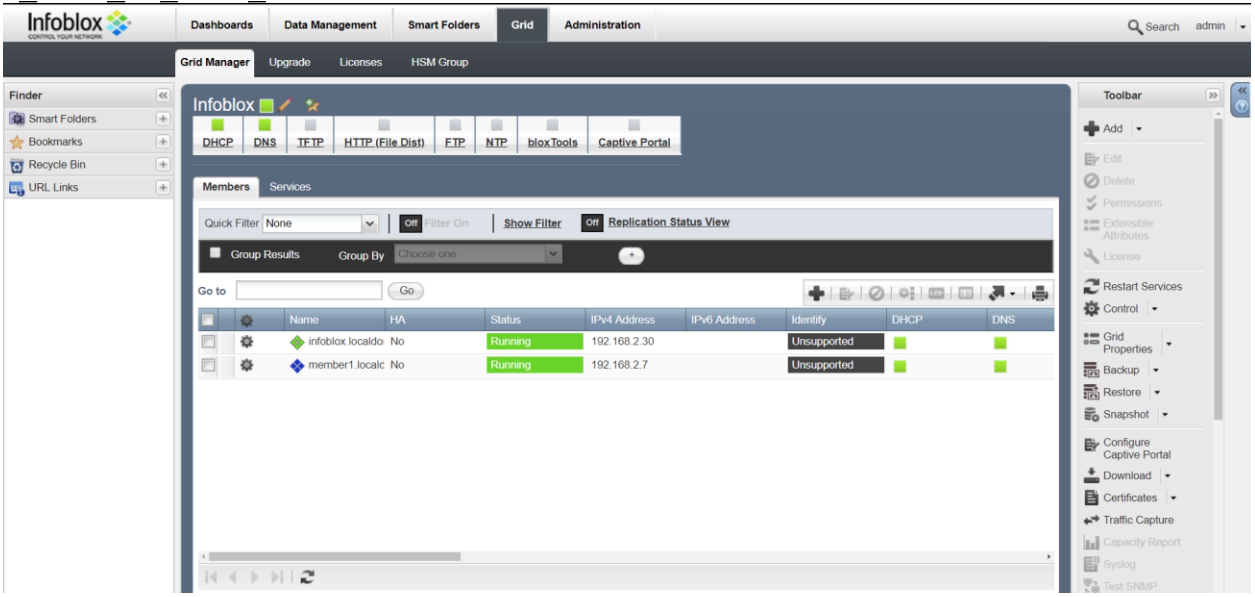


Instances

Displaying 2 items

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status		Availability Zone	Task	Power State	Time since created	Actions
□	Member-1	NIOS-810	Lan-1 192.168.2.7 Floating IPs: 10.196.200.74 Mgmt 172.26.1.22	NIOS-810	-	Active	🔓	nova	None	Running	9 minutes	Create Snapshot ▾
□	Grid_Master	NIOS-810	Lan-1 192.168.2.30 Floating IPs: 10.196.200.71 Mgmt 172.26.1.11	NIOS-810	-	Active	🔓	nova	None	Running	9 minutes	Create Snapshot ▾

8. Login to the Infoblox Grid using `https://floating_ip_associated_with_Lan-1_port_of_grid_master` and verify grid status.



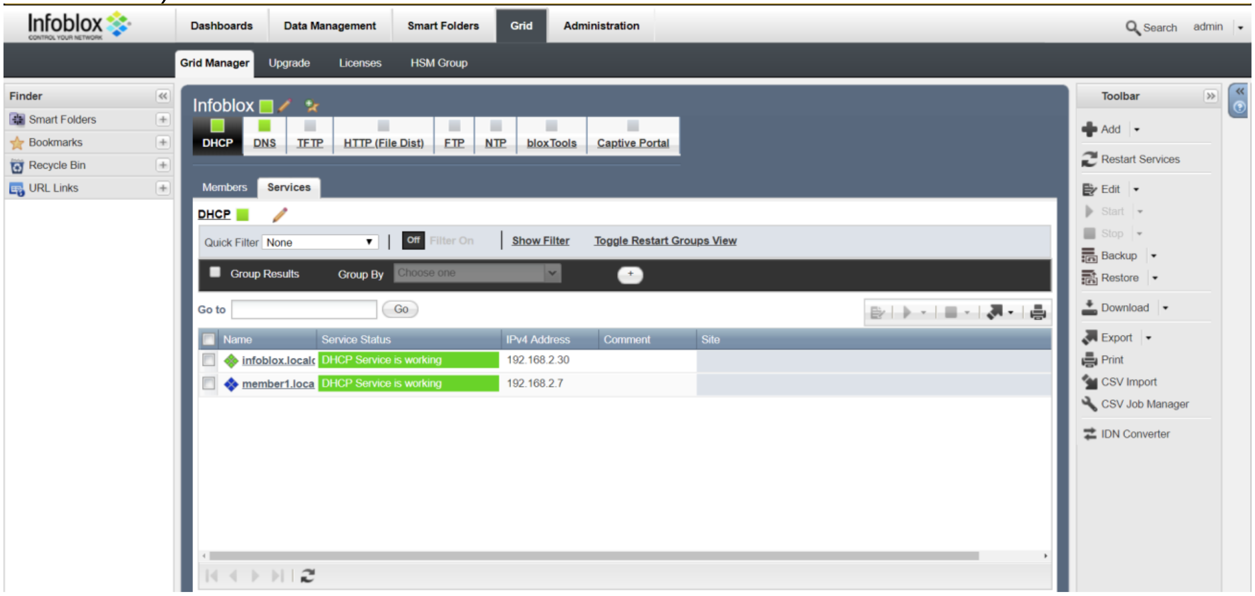
Quick Filter None | Off Filter On | Show Filter | Off Replication Status View

□ Group Results Group By Choose one

Go to Go

□		Name	HA	Status	IPv4 Address	IPv6 Address	Identify	DHCP	DNS
□		infoblox.localdo	No	Running	192.168.2.30		Unsupported	🟩	🟩
□		member1.localc	No	Running	192.168.2.7		Unsupported	🟩	🟩

9. Click on **Services** tab and verify DHCP status. (It should be up and running for both Grid-Master and Member.)



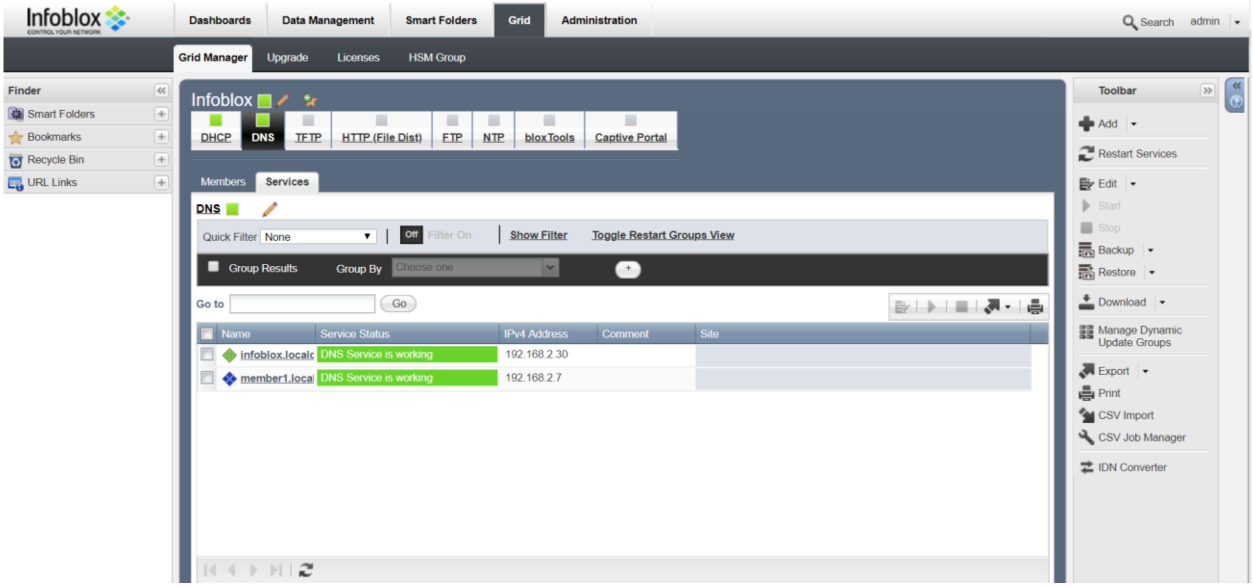
Quick Filter None | Off Filter On | Show Filter Toggle Restart Groups View

□ Group Results Group By Choose one

Go to Go

□	Name	Service Status	IPv4 Address	Comment	Site
□	infoblox.localc	DHCP Service is working	192.168.2.30		
□	member1.loca	DHCP Service is working	192.168.2.7		

10. Click on DNS tab and verify its status. (It should be up and running for both Grid-Master and Member.)



Quick Filter: None

Name	Service Status	IPv4 Address	Comment	Site
infoblox.localc	DNS Service is working	192.168.2.30		
member1.local	DNS Service is working	192.168.2.7		

Infoblox is the leader in modern, cloud-first networking and security services. Through extensive integrations, its solutions empower organizations to realize the full advantages of cloud networking today, while maximizing their existing infrastructure investments. Infoblox has over 12,000 customers, including 70 percent of the Fortune 500.

Corporate Headquarters | 2390 Mission College Boulevard, Ste. 501 | Santa Clara, CA | 95054
+1.408.986.4000 | info@infoblox.com | www.infoblox.com

© 2020 Infoblox, Inc. All rights reserved. Infoblox logo, and other marks appearing herein are property of Infoblox, Inc. All other marks are the property of their respective owner(s).