Infoblox
CONTROL YOUR NETWORK

DEPLOYMENT GUIDE

Deploying IB-FLEX

NIOS version 8.2 | March 2018

Contents

Overview

Introduction

IB-FLEX is a virtual platform that is scalable based on the resource that you allocate to the virtual machine. NIOS automatically detects the capacity of the virtual machine and scales it to the appropriate platform after you provision the IB-FLEX member.

You must first install the Grid license on a non IB-FLEX appliance that is designated as the Grid Master to allow members to join the Grid, even if you have already installed a Flex Grid Activation license. This license does not affect a non IB-FLEX Grid Master.

An IB-FLEX appliance designated as a member does not require any license, either Grid or vNIOS, while joining the Grid. When you register an IB-FLEX member, the appliance checks for the Grid (enterprise) license and changes it to a non IB-FLEX member. For an IB-FLEX appliance, it checks for a Flex Grid Activation Grid-wide license before node registration.

IB-FLEX members can join the Grid through the MGMT interface when Software ADP is enabled. You can configure an IB-FLEX appliance to function as a Grid Master or a member. To enable reporting for a Grid member that is running Software ADP, you must configure the MGMT interface.

A non IB-FLEX appliance designated as a member requires either a Grid and/or vNIOS/NIOS licenses installed to join the Grid. Similarly, for a reporting appliance to join the Grid, you must install a Grid and/or vNIOS/NIOS licenses. You cannot assign pool licenses to an IB-FLEX appliance. IB-FLEX supports HA for appliances that are running Software ADP.

Infoblox supports elastic scaling on IB-FLEX members that use the **Flex Grid Activation** Grid-wide license. It also supports pre-provisioning for Software ADP on the supported platforms. You must add the new IB-FLEX model to the list of supported pre-provisioning hardware types, so that you can select it during the member pre-provisioning. To pre-provision a non IB-FLEX Grid member, you must have valid pool licenses and pre-provisioned those members in the Grid.

NIOS, vNIOS and IB-FLEX

NIOS

Network Identity Operating System (NIOS) is an Infoblox's proprietary system that powers Infoblox appliances with an embedded processor that delivers nonstop core network services. A security-hardened, real-time set of appliances built to ensure the non-stop operation of network infrastructure, NIOS automates the error-prone and time-consuming manual tasks associated with deploying and managing DNS, DHCP, and IP address management (IPAM) required for continuous IP network availability and business uptime.

vNIOS

vNIOS is Infoblox's virtual offering for the customers who do not wish to have a hardware-based appliance. vNIOS is available for multiple hypervisors like ESXi, KVM, and Hyper-V. Infoblox also supports virtual appliances for various cloud platforms like AWS, Azure, and Openstack.

IB-FLEX

IB-FLEX is an extension of the virtual offering of the Infoblox. It supports elastic scaling and can be scaled easily based on the usage. It makes customer's life easy, since now they don't have to worry about upgrading the appliances in case of increased usage. They can simply activate the IB-FLEX license and scale up the appliances as per the requirement.

Pre-requisites

1. VMware vSphere environment (5.5, 6.0 or 6.5) or an Openstack environment (Mitaka, Newton, Ocata release)
2. Grid Master (deployed either on VMware or Openstack with Flex activation license)
3. DDI images (.ova for VMware or qcow2 for Openstack). Images can be downloaded from https://support.infoblox.com.
4. Reporting server deployed and added to the grid.

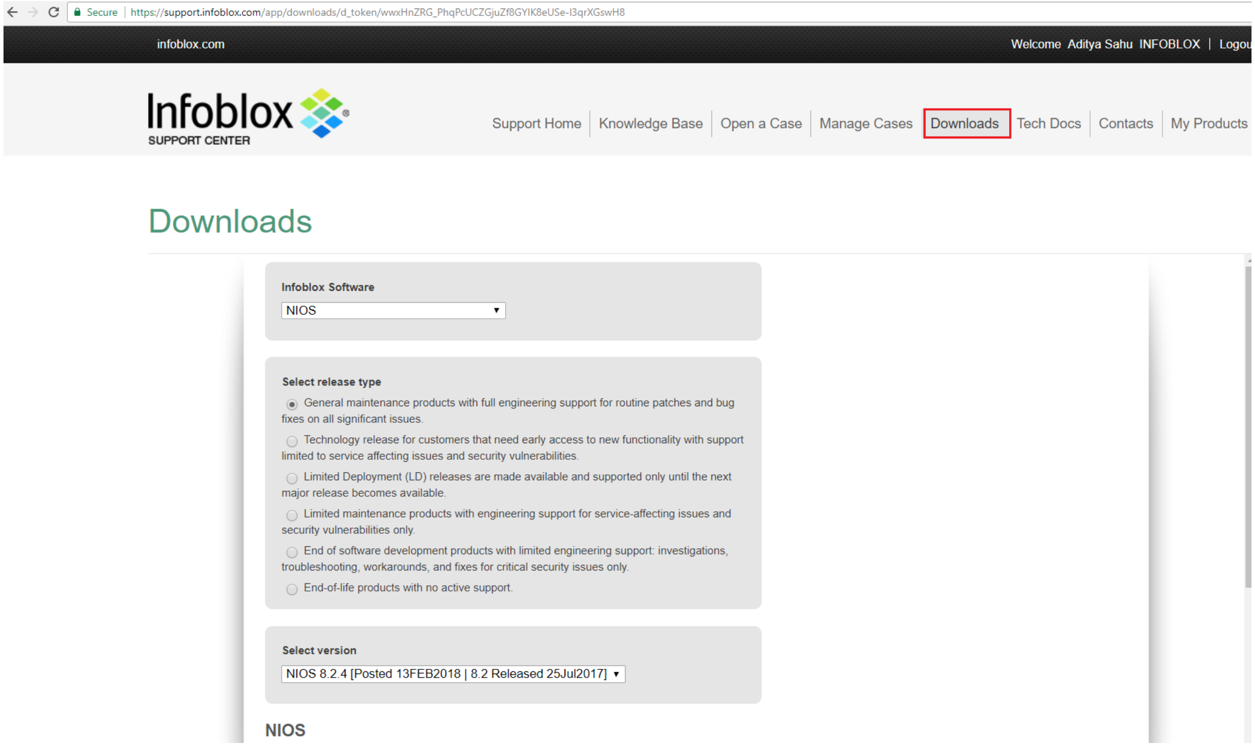
Installing IB-FLEX

Depending on your network environment, you can install the IB-FLEX appliance just like how you install other Infoblox virtual appliances. Before you deploy an IB-FLEX, ensure that you set the hardware type of the appliance to IB-FLEX. You can do so either through the cloud-init process during deployment or manually through the **set hardware-type** CLI command.

Downloading the required images (For VMware and Openstack)

IB-FLEX images can be downloaded from support.infoblox.com portal.

1. Navigate to the **Downloads** tab
2. Select **Infoblox Software** as NIOS
3. Under **Select version,** select NIOS 8.2.4


Secure | https://support.infoblox.com/app/downloads/d_token/wwxHnZRG_PhqPcUCZGjuZf8GYIK8eUSe-l3qrXGswH8
infoblox.com
Welcome Aditya Sahu INFOBLOX | Logou
Infoblox
SUPPORT CENTER
Support Home
Knowledge Base
Open a Case
Manage Cases
Downloads
Tech Docs
Contacts
My Products

Downloads

Infoblox Software
NIOS ▾

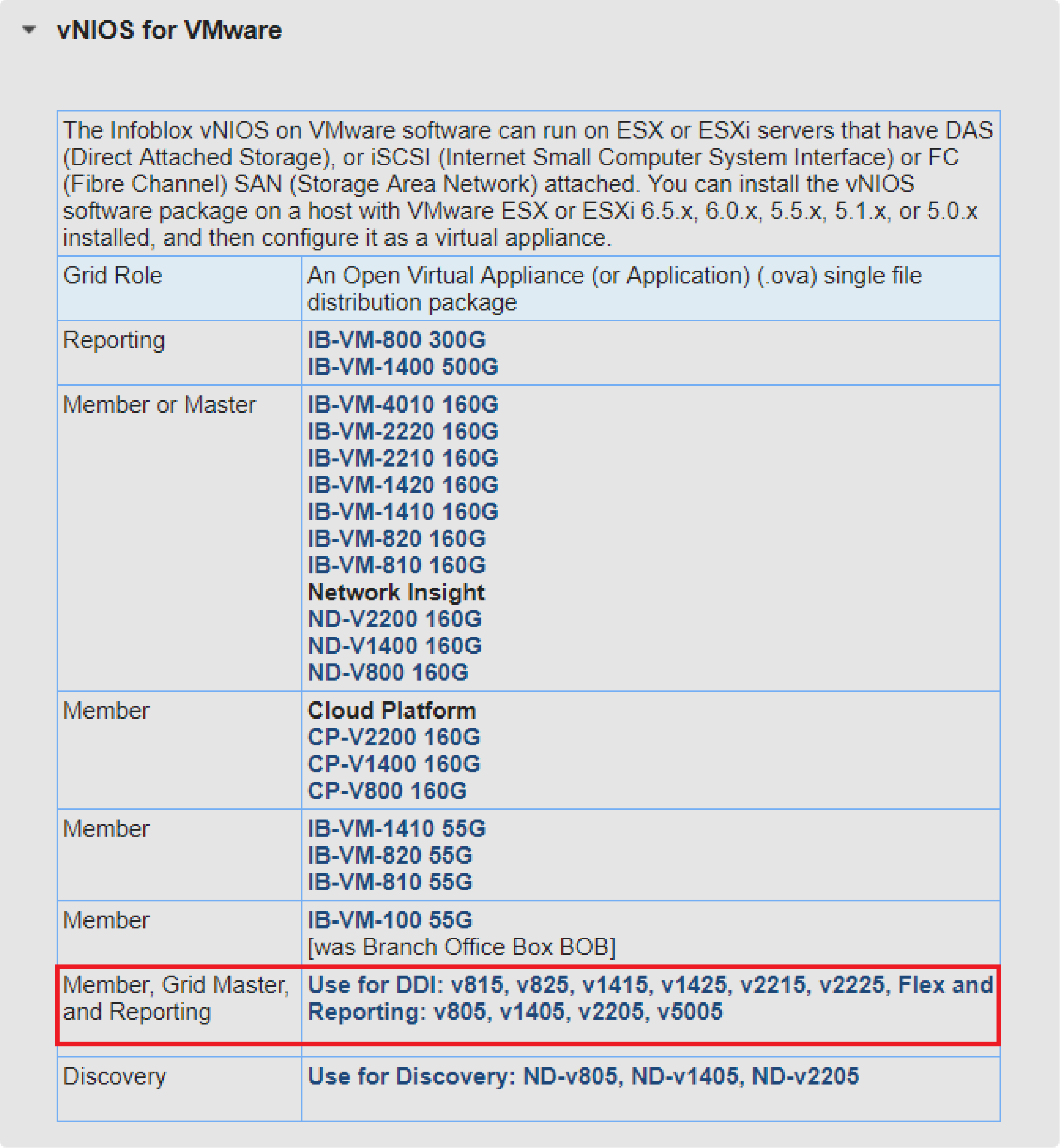
Select release type
● General maintenance products with full engineering support for routine patches and bug fixes on all significant issues.
○ Technology release for customers that need early access to new functionality with support limited to service affecting issues and security vulnerabilities.
○ Limited Deployment (LD) releases are made available and supported only until the next major release becomes available.
○ Limited maintenance products with engineering support for service-affecting issues and security vulnerabilities only.
○ End of software development products with limited engineering support: investigations, troubleshooting, workarounds, and fixes for critical security issues only.
○ End-of-life products with no active support.

Select version
NIOS 8.2.4 [Posted 13FEB2018 | 8.2 Released 25Jul2017] ▾

NIOS

4. Scroll down and expand **vNIOS of Vmware**

5. Go to **Member, Grid Master, and Reporting** row to download the corresponding DDI image.

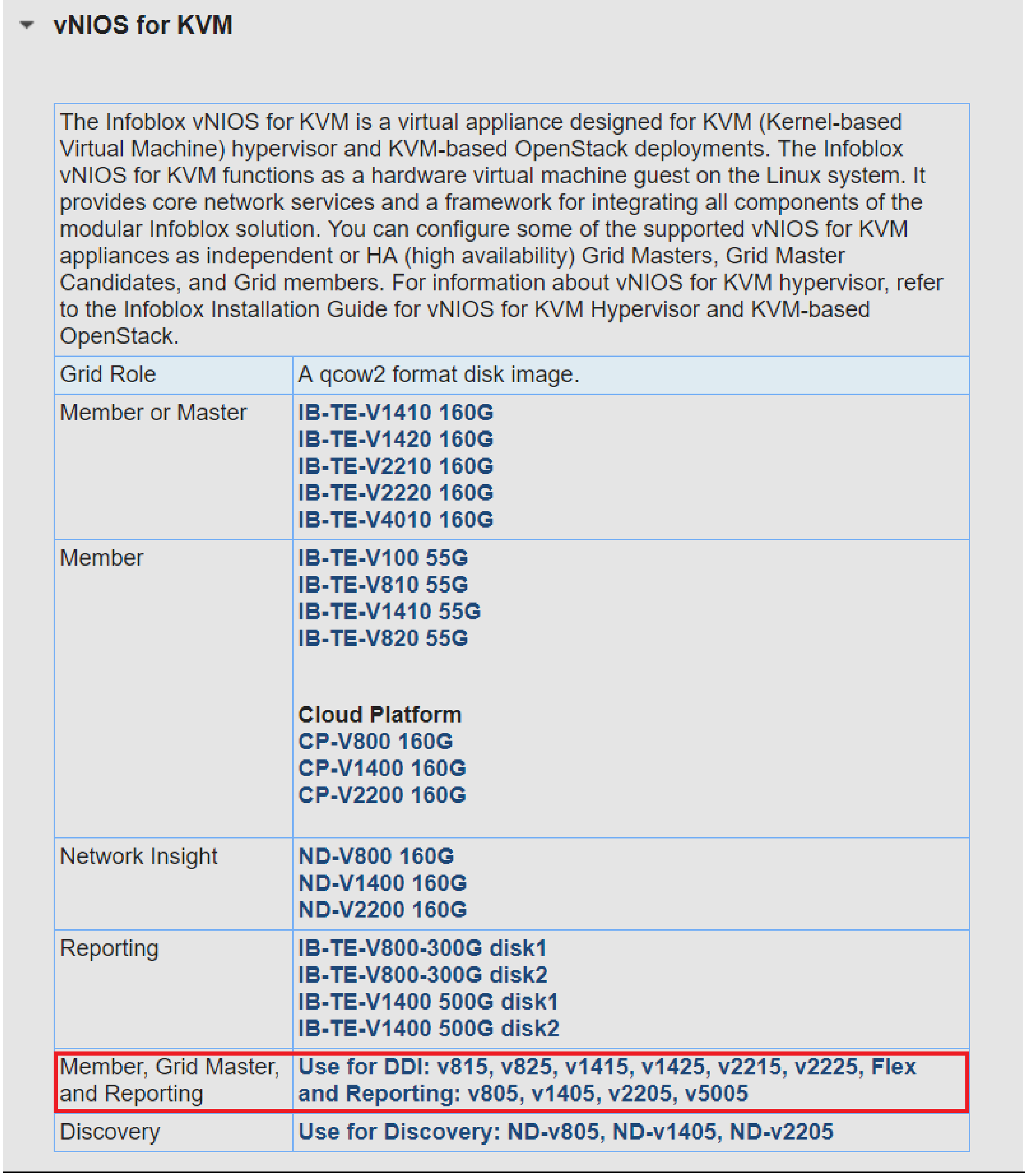


The Infoblox vNIOS on VMware software can run on ESX or ESXi servers that have DAS (Direct Attached Storage), or iSCSI (Internet Small Computer System Interface) or FC (Fibre Channel) SAN (Storage Area Network) attached. You can install the vNIOS software package on a host with VMware ESX or ESXi 6.5.x, 6.0.x, 5.5.x, 5.1.x, or 5.0.x installed, and then configure it as a virtual appliance.

Grid Role	An Open Virtual Appliance (or Application) (.ova) single file distribution package
Reporting	**IB-VM-800 300G** **IB-VM-1400 500G**
Member or Master	**IB-VM-4010 160G** **IB-VM-2220 160G** **IB-VM-2210 160G** **IB-VM-1420 160G** **IB-VM-1410 160G** **IB-VM-820 160G** **IB-VM-810 160G** **Network Insight** **ND-V2200 160G** **ND-V1400 160G** **ND-V800 160G**
Member	**Cloud Platform** **CP-V2200 160G** **CP-V1400 160G** **CP-V800 160G**
Member	**IB-VM-1410 55G** **IB-VM-820 55G** **IB-VM-810 55G**
Member	**IB-VM-100 55G** [was Branch Office Box BOB]
Member, Grid Master, and Reporting	**Use for DDI: v815, v825, v1415, v1425, v2215, v2225, Flex and Reporting: v805, v1405, v2205, v5005**
Discovery	**Use for Discovery: ND-v805, ND-v1405, ND-v2205**

6. To download image for Openstack platform, scroll down and expand **vNIOS for KVM**

7. Go to **Member, Grid Master, and Reporting** row to download the corresponding DDI image.



▾ **vNIOS for KVM**

The Infoblox vNIOS for KVM is a virtual appliance designed for KVM (Kernel-based Virtual Machine) hypervisor and KVM-based OpenStack deployments. The Infoblox vNIOS for KVM functions as a hardware virtual machine guest on the Linux system. It provides core network services and a framework for integrating all components of the modular Infoblox solution. You can configure some of the supported vNIOS for KVM appliances as independent or HA (high availability) Grid Masters, Grid Master Candidates, and Grid members. For information about vNIOS for KVM hypervisor, refer to the Infoblox Installation Guide for vNIOS for KVM Hypervisor and KVM-based OpenStack.

Grid Role	A qcow2 format disk image.
Member or Master	**IB-TE-V1410 160G** **IB-TE-V1420 160G** **IB-TE-V2210 160G** **IB-TE-V2220 160G** **IB-TE-V4010 160G**
Member	**IB-TE-V100 55G** **IB-TE-V810 55G** **IB-TE-V1410 55G** **IB-TE-V820 55G** **Cloud Platform** **CP-V800 160G** **CP-V1400 160G** **CP-V2200 160G**
Network Insight	**ND-V800 160G** **ND-V1400 160G** **ND-V2200 160G**
Reporting	**IB-TE-V800-300G disk1** **IB-TE-V800-300G disk2** **IB-TE-V1400 500G disk1** **IB-TE-V1400 500G disk2**
Member, Grid Master, and Reporting	**Use for DDI: v815, v825, v1415, v1425, v2215, v2225, Flex and Reporting: v805, v1405, v2205, v5005**
Discovery	**Use for Discovery: ND-v805, ND-v1405, ND-v2205**

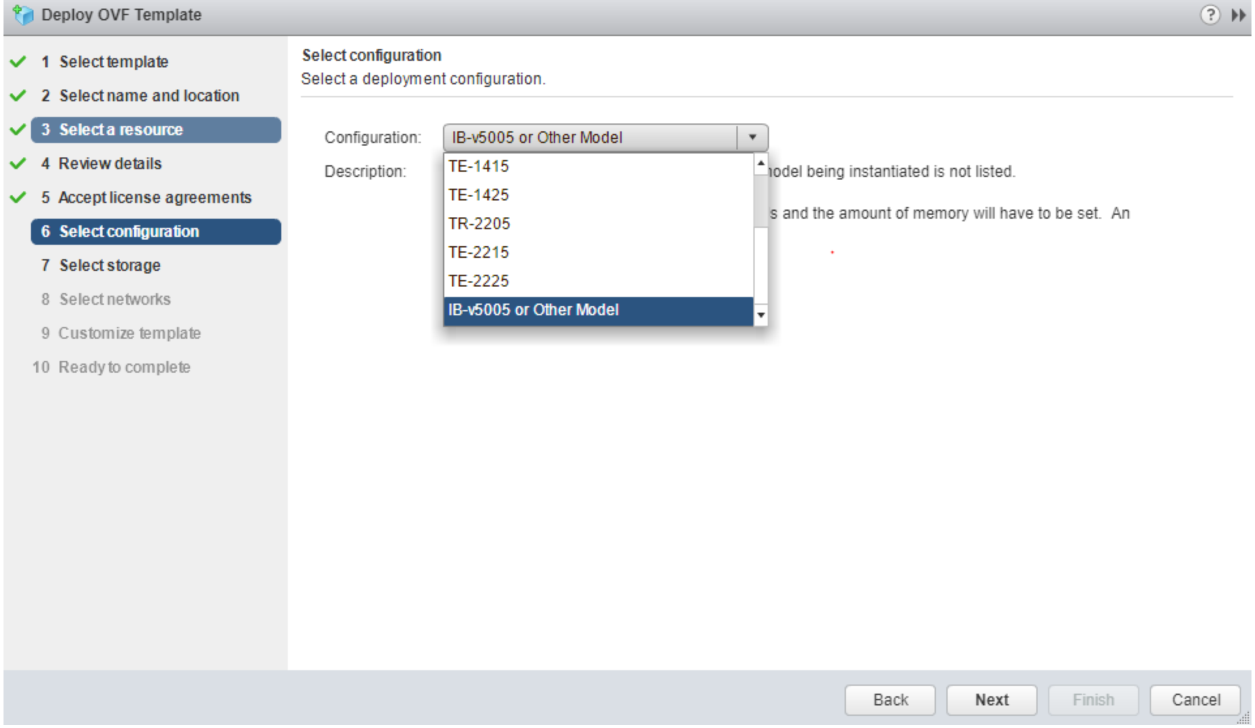
Installing IB-FLEX on VMware

This section walks you through installing and managing IB-FLEX appliances on VMware.

Deploying Grid Master

1. Deploy the NIOS OVF template downloaded from the Infoblox Support site.

2. During the OVF deployment, please select the IB-Appliance model



Deploy OVF Template

1 Select template
2 Select name and location
3 Select a resource
4 Review details
5 Accept license agreements
6 Select configuration
7 Select storage
8 Select networks
9 Customize template
10 Ready to complete

Select configuration
Select a deployment configuration.

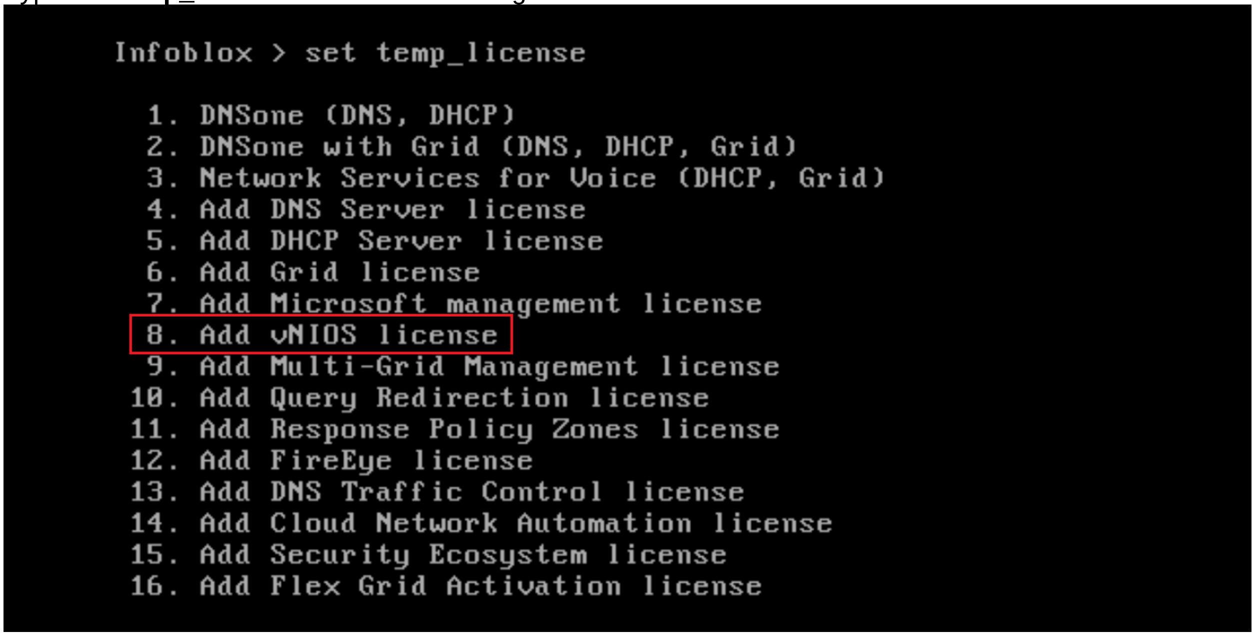
Configuration: IB-v5005 or Other Model

Description: ıodel being instantiated is not listed.

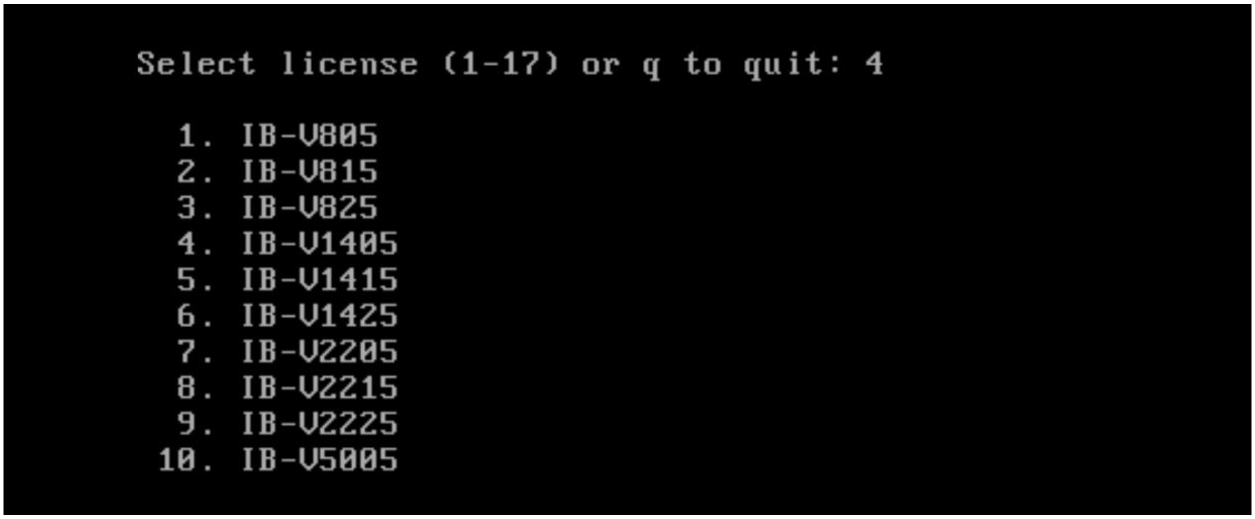
s and the amount of memory will have to be set. An

TE-1415
TE-1425
TR-2205
TE-2215
TE-2225
IB-v5005 or Other Model

3. Type **set temp_license** and enter **8** to assign vNIOS license.

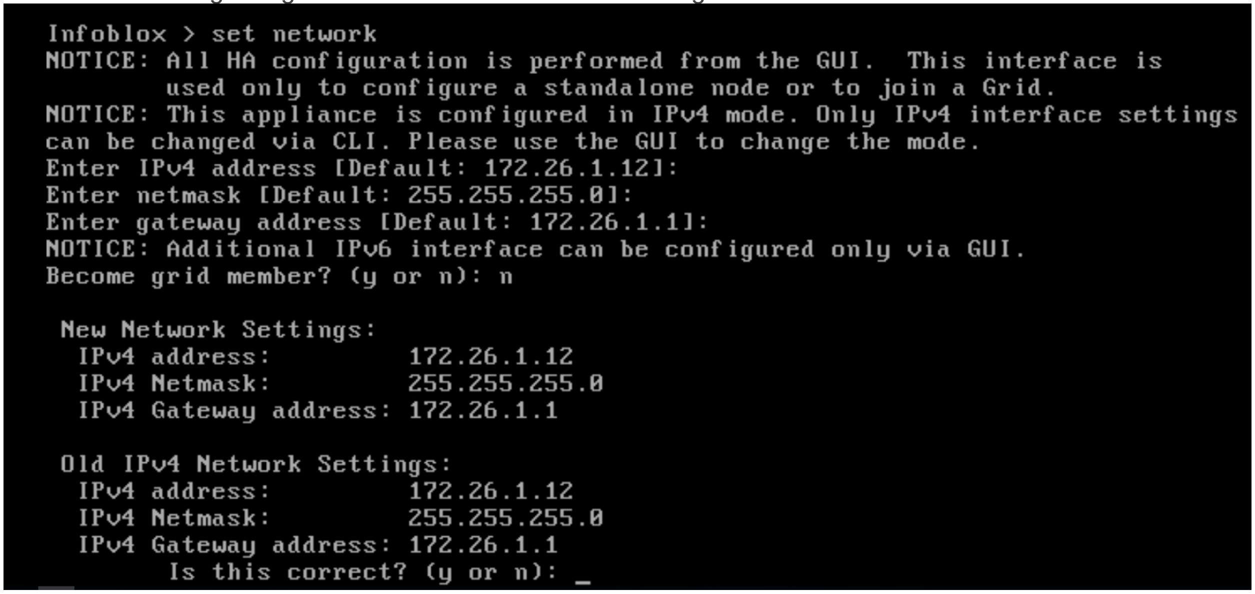


```
Infoblox > set temp_license

  1. DNSone (DNS, DHCP)
  2. DNSone with Grid (DNS, DHCP, Grid)
  3. Network Services for Voice (DHCP, Grid)
  4. Add DNS Server license
  5. Add DHCP Server license
  6. Add Grid license
  7. Add Microsoft management license
  8. Add vNIOS license
  9. Add Multi-Grid Management license
 10. Add Query Redirection license
 11. Add Response Policy Zones license
 12. Add FireEye license
 13. Add DNS Traffic Control license
 14. Add Cloud Network Automation license
 15. Add Security Ecosystem license
 16. Add Flex Grid Activation license
```



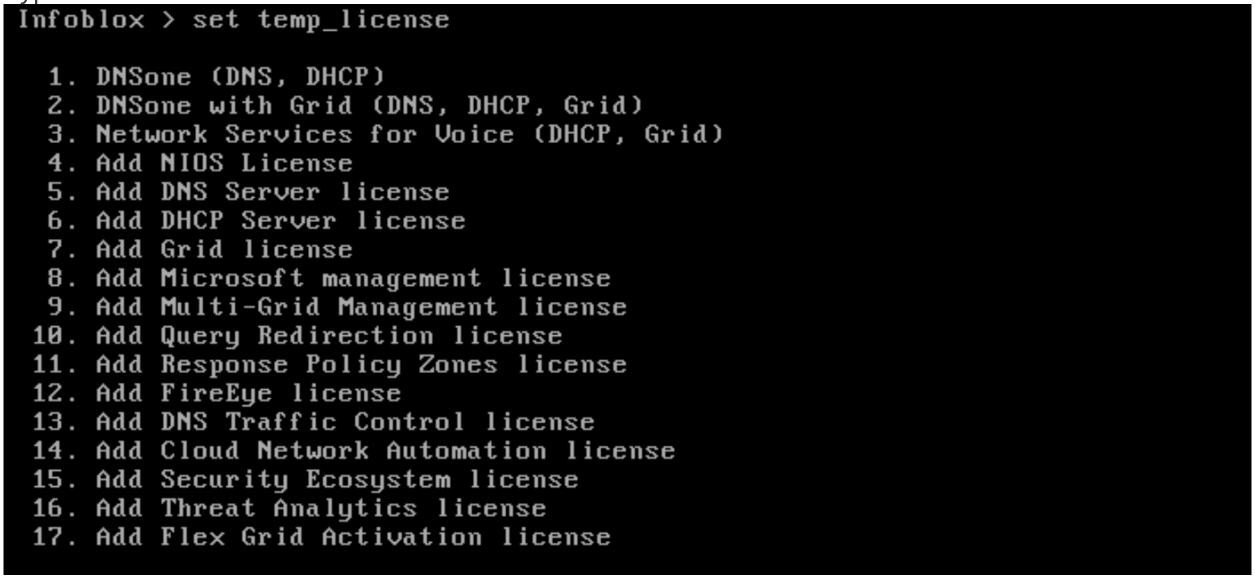
```
Select license (1-17) or q to quit: 4

  1. IB-V805
  2. IB-V815
  3. IB-V825
  4. IB-V1405
  5. IB-V1415
  6. IB-V1425
  7. IB-V2205
  8. IB-V2215
  9. IB-V2225
 10. IB-V5005
```

4. Set the networking using **set network** command and configure it as Grid Master



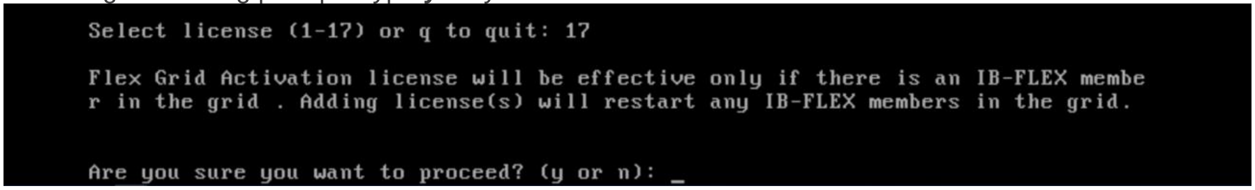
```
Infoblox > set network
NOTICE: All HA configuration is performed from the GUI.  This interface is
        used only to configure a standalone node or to join a Grid.
NOTICE: This appliance is configured in IPv4 mode. Only IPv4 interface settings
can be changed via CLI. Please use the GUI to change the mode.
Enter IPv4 address [Default: 172.26.1.12]:
Enter netmask [Default: 255.255.255.0]:
Enter gateway address [Default: 172.26.1.1]:
NOTICE: Additional IPv6 interface can be configured only via GUI.
Become grid member? (y or n): n

 New Network Settings:
  IPv4 address:         172.26.1.12
  IPv4 Netmask:         255.255.255.0
  IPv4 Gateway address: 172.26.1.1

 Old IPv4 Network Settings:
  IPv4 address:         172.26.1.12
  IPv4 Netmask:         255.255.255.0
  IPv4 Gateway address: 172.26.1.1
        Is this correct? (y or n): _
```

5. Navigate to the licensing option again by typing **set temp_license**.
   Type **17** to activate Flex Grid Activation License



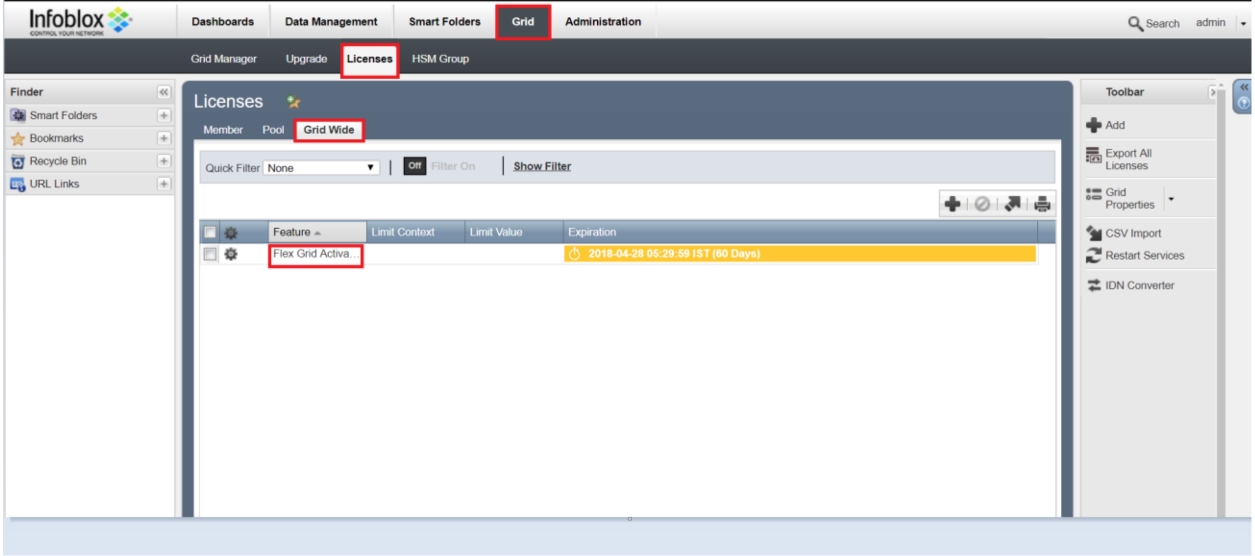
```
Infoblox > set temp_license

   1. DNSone (DNS, DHCP)
   2. DNSone with Grid (DNS, DHCP, Grid)
   3. Network Services for Voice (DHCP, Grid)
   4. Add NIOS License
   5. Add DNS Server license
   6. Add DHCP Server license
   7. Add Grid license
   8. Add Microsoft management license
   9. Add Multi-Grid Management license
  10. Add Query Redirection license
  11. Add Response Policy Zones license
  12. Add FireEye license
  13. Add DNS Traffic Control license
  14. Add Cloud Network Automation license
  15. Add Security Ecosystem license
  16. Add Threat Analytics license
  17. Add Flex Grid Activation license
```

6. You will get following prompt. Type **y** for yes and hit enter.



```
     Select license (1-17) or q to quit: 17

     Flex Grid Activation license will be effective only if there is an IB-FLEX membe
     r in the grid . Adding license(s) will restart any IB-FLEX members in the grid.


     Are you sure you want to proceed? (y or n): _
```

7. To verify the Flex Grid Activation license, login to the Grid GUI and navigate to
   Grid → Licenses → Grid Wide

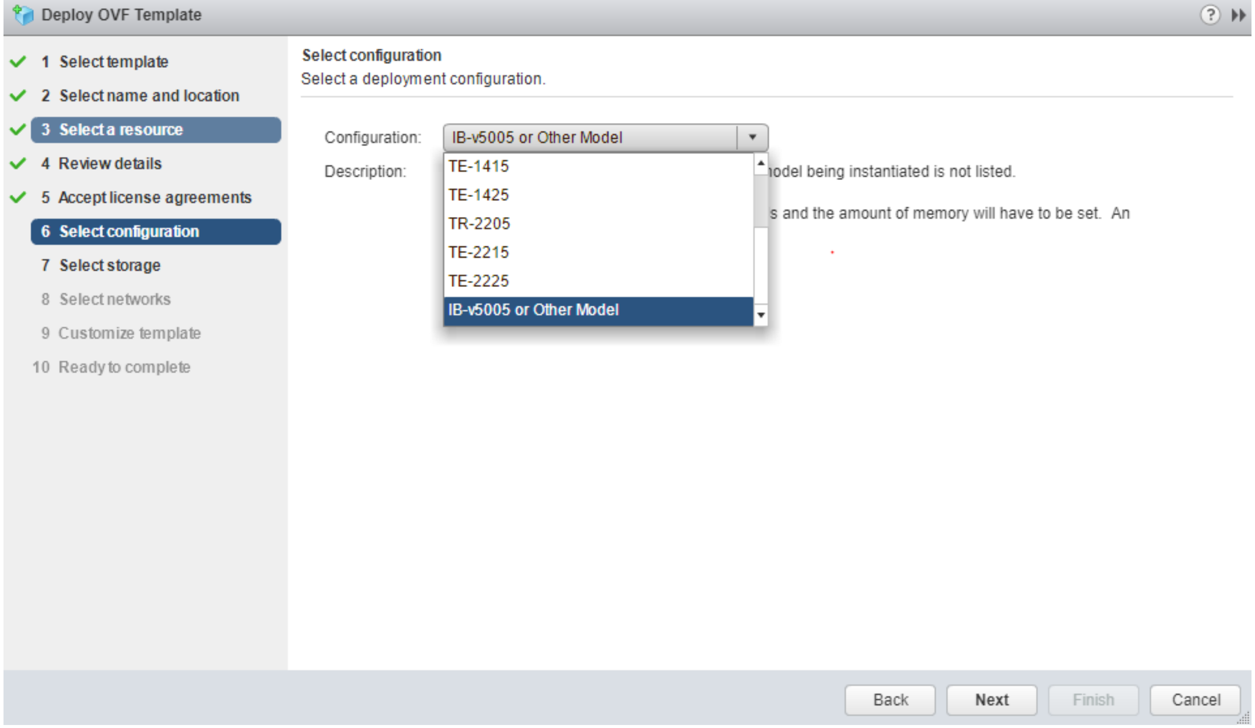


Feature	Limit Context	Limit Value	Expiration
Flex Grid Activa...			2018-04-28 05:29:59 IST (60 Days)

Adding a IB-FLEX member to the Grid

1. Deploy a NIOS instance from the previously downloaded NIOS OVF template.

2. During the deployment, select the desired NIOS model.

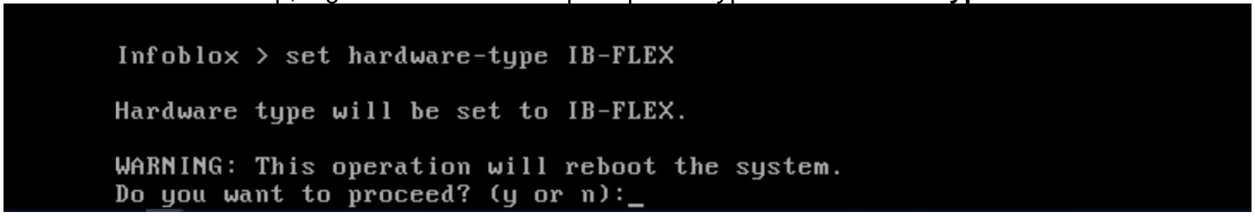


Select configuration
Select a deployment configuration.

Configuration: IB-v5005 or Other Model

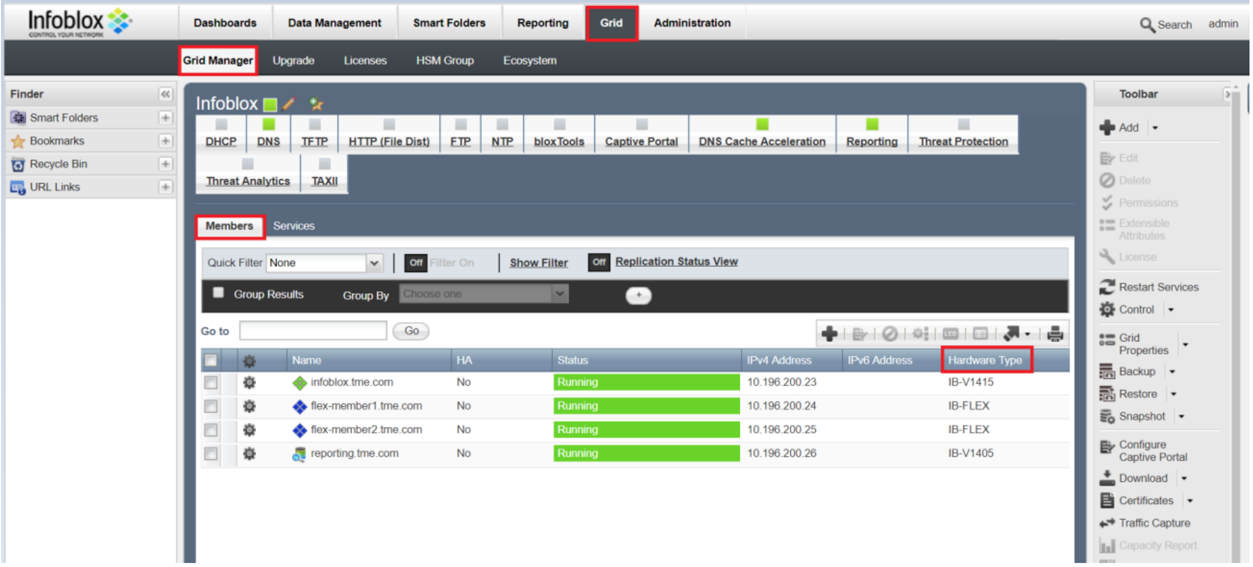
Description: ...odel being instantiated is not listed. ...s and the amount of memory will have to be set. An

3. After NIOS VM boots up, login at the command prompt and type **set hardware-type IB-FLEX**



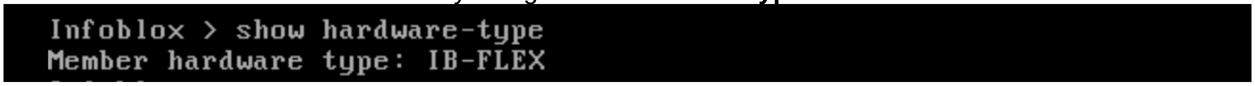
```
Infoblox > set hardware-type IB-FLEX

Hardware type will be set to IB-FLEX.

WARNING: This operation will reboot the system.
Do you want to proceed? (y or n):_
```

4. After reboot set the networking and add the NIOS to the grid using **set network** command.
5. You can verify the IB-FLEX member type in the grid by navigating to **Grid → Grid Manager → Members→ Hardware Type**



Name	HA	Status	IPv4 Address	IPv6 Address	Hardware Type
infoblox.tme.com	No	Running	10.196.200.23		IB-V1415
flex-member1.tme.com	No	Running	10.196.200.24		IB-FLEX
flex-member2.tme.com	No	Running	10.196.200.25		IB-FLEX
reporting.tme.com	No	Running	10.196.200.26		IB-V1405

6. IB-FLEX license can also be verified by using **show hardware-type** command.

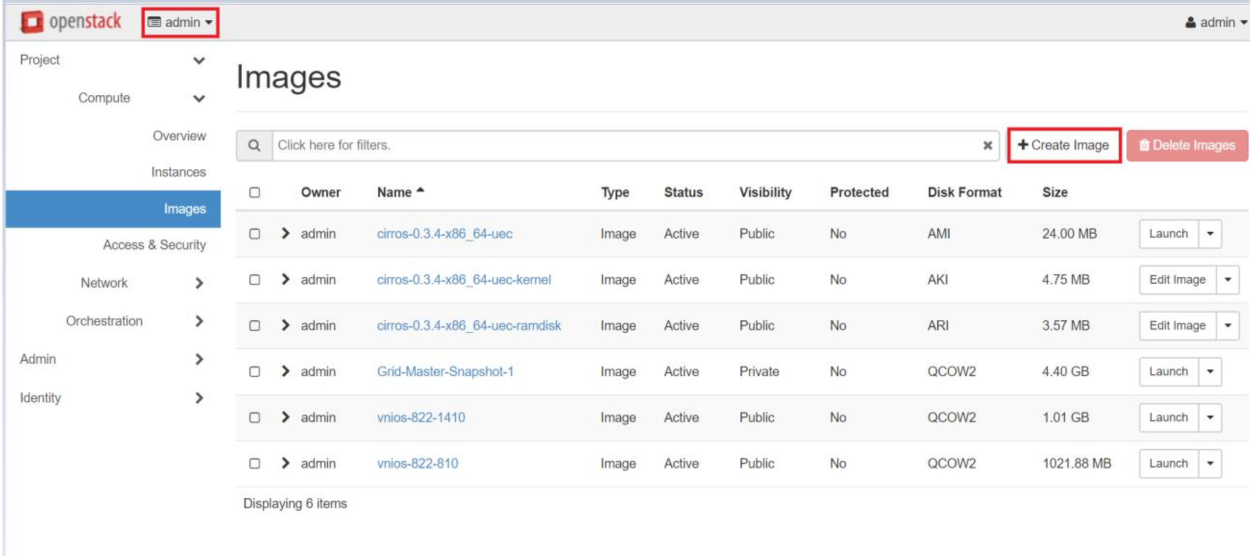


```
Infoblox > show hardware-type
Member hardware type: IB-FLEX
```

Installing IB-FLEX on Openstack

Creating a DDI image

1. Login to the admin project of Openstack and navigate to **Project → Compute → Images.**
2. Click on **Create Image**.



Images

	Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size	
□	admin	cirros-0.3.4-x86_64-uec	Image	Active	Public	No	AMI	24.00 MB	Launch
□	admin	cirros-0.3.4-x86_64-uec-kernel	Image	Active	Public	No	AKI	4.75 MB	Edit Image
□	admin	cirros-0.3.4-x86_64-uec-ramdisk	Image	Active	Public	No	ARI	3.57 MB	Edit Image
□	admin	Grid-Master-Snapshot-1	Image	Active	Private	No	QCOW2	4.40 GB	Launch
□	admin	vnios-822-1410	Image	Active	Public	No	QCOW2	1.01 GB	Launch
□	admin	vnios-822-810	Image	Active	Public	No	QCOW2	1021.88 MB	Launch

Displaying 6 items

3. Enter an image name.
4. Browse for the previously downloaded NIOS DDI qcow2 image.
5. In the **Format** option select QCOW2-QEMU Emulator. Click on **Create Image**

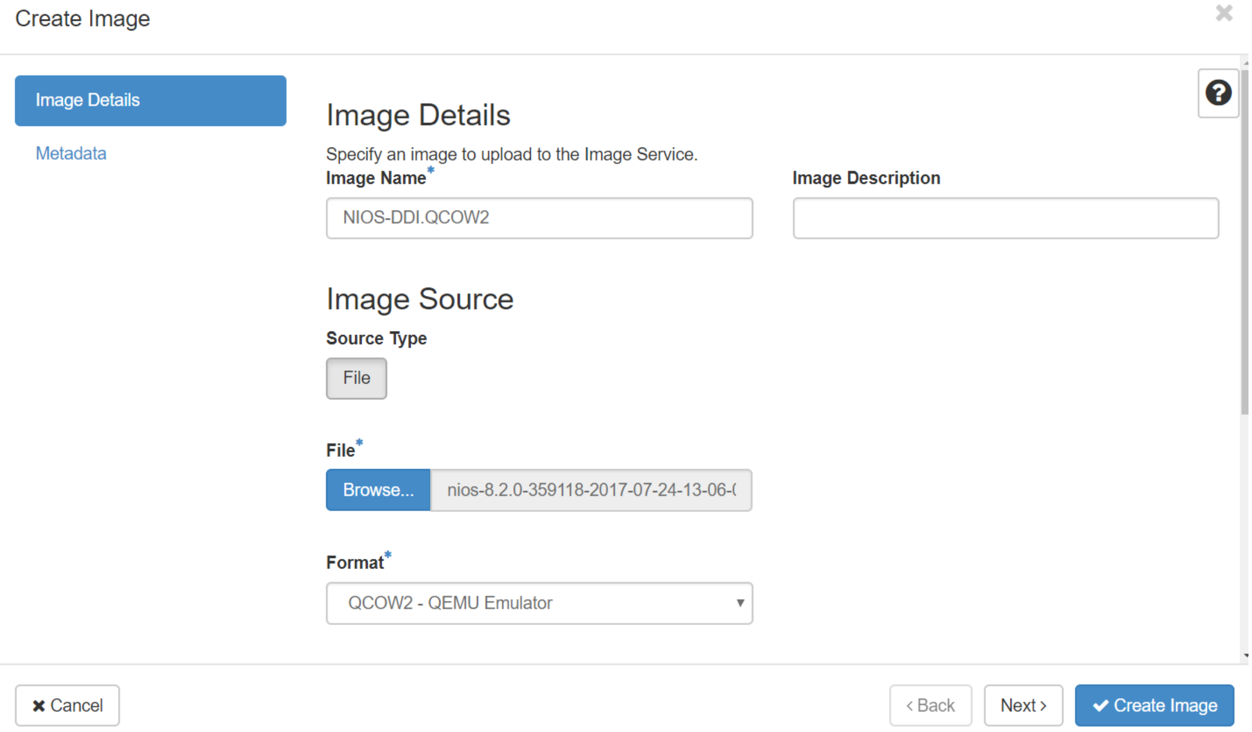


Image Details

Specify an image to upload to the Image Service.

Image Name*: NIOS-DDI.QCOW2

Image Description:

Image Source

Source Type: File

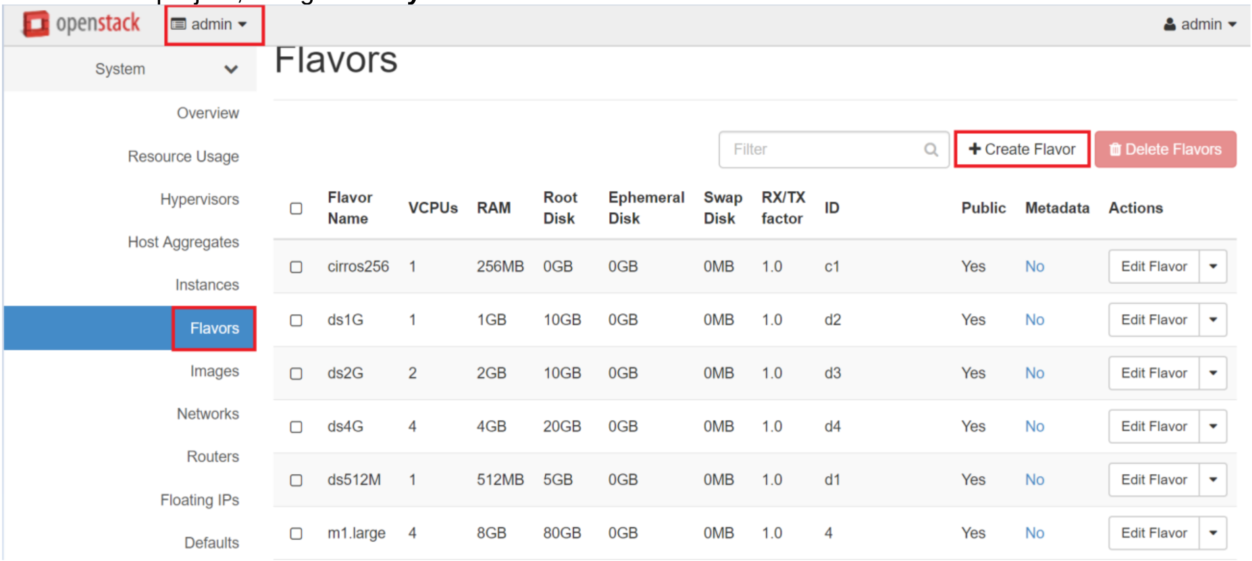
File*: nios-8.2.0-359118-2017-07-24-13-06-(

Format*: QCOW2 - QEMU Emulator

Creating a Flavor

Flavor is required to create an instance. Flavor contains the resource information required by the instance. This deployment guide uses NIOS 1415 which requires 32 GB RAM and 4 vCPUs.

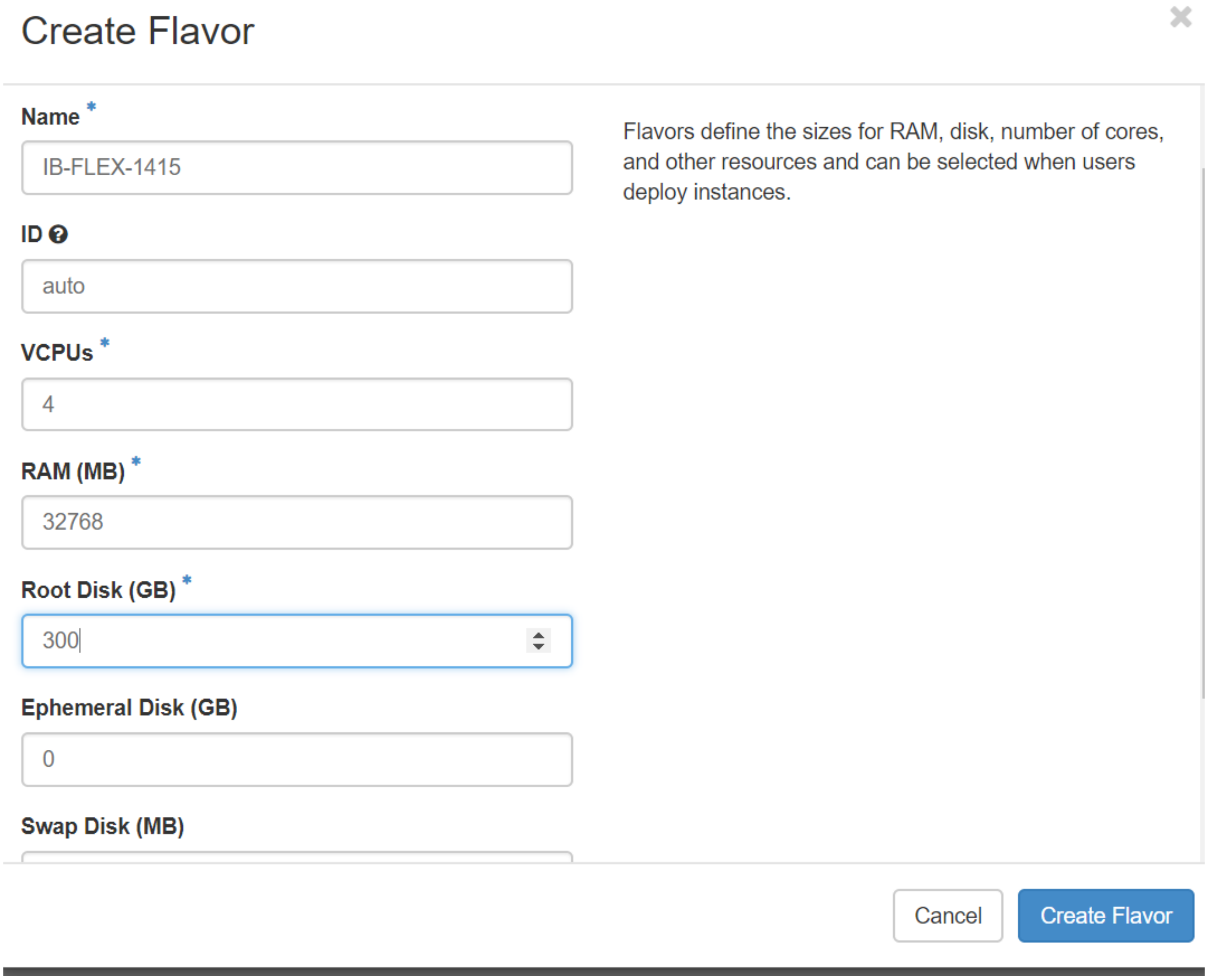
1. In the admin project, navigate to **System→Flavors→ Create Flavor**



□	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	Edit Flavor ▾
□	ds1G	1	1GB	10GB	0GB	0MB	1.0	d2	Yes	No	Edit Flavor ▾
□	ds2G	2	2GB	10GB	0GB	0MB	1.0	d3	Yes	No	Edit Flavor ▾
□	ds4G	4	4GB	20GB	0GB	0MB	1.0	d4	Yes	No	Edit Flavor ▾
□	ds512M	1	512MB	5GB	0GB	0MB	1.0	d1	Yes	No	Edit Flavor ▾
□	m1.large	4	8GB	80GB	0GB	0MB	1.0	4	Yes	No	Edit Flavor ▾

2. Enter a Flavor name.
3. In the VCPUs enter 4 and in the RAM (MB) enter 32768.

4. Root disk for this flavor is 300 (GB)



Create Flavor

Name *

IB-FLEX-1415

ID

auto

VCPUs *

4

RAM (MB) *

32768

Root Disk (GB) *

300

Ephemeral Disk (GB)

0

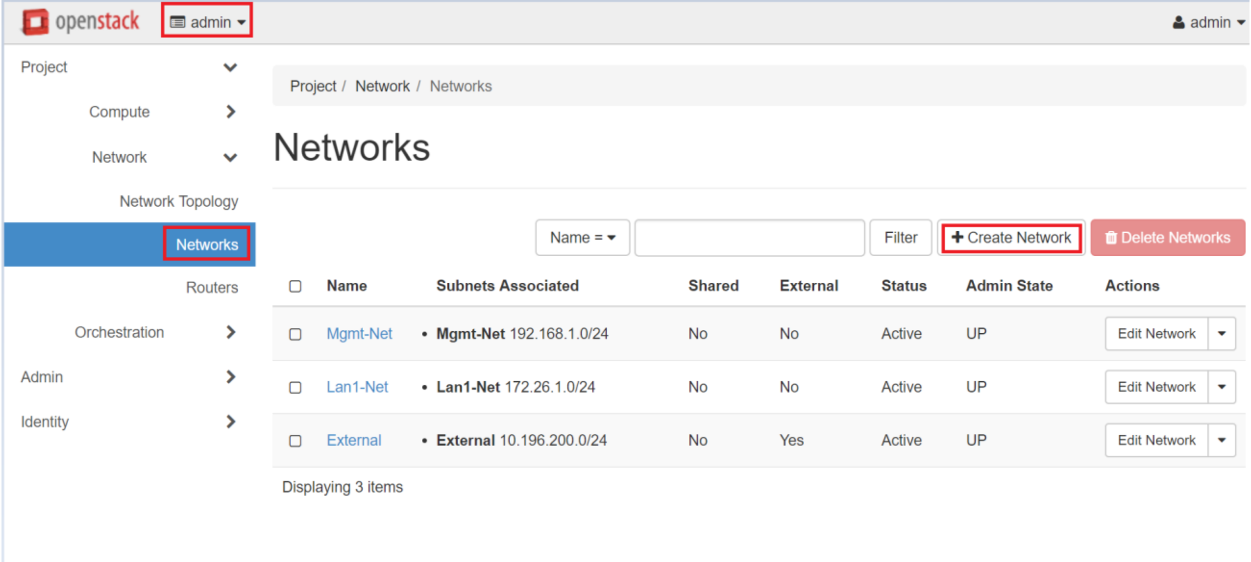
Swap Disk (MB)

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

Creating Networks

NIOS instances require 2 networks (Mgmt and Lan1) to be connected, to boot up successfully.

1. In the admin project, navigate to **Project → Network → Networks → Create Network**



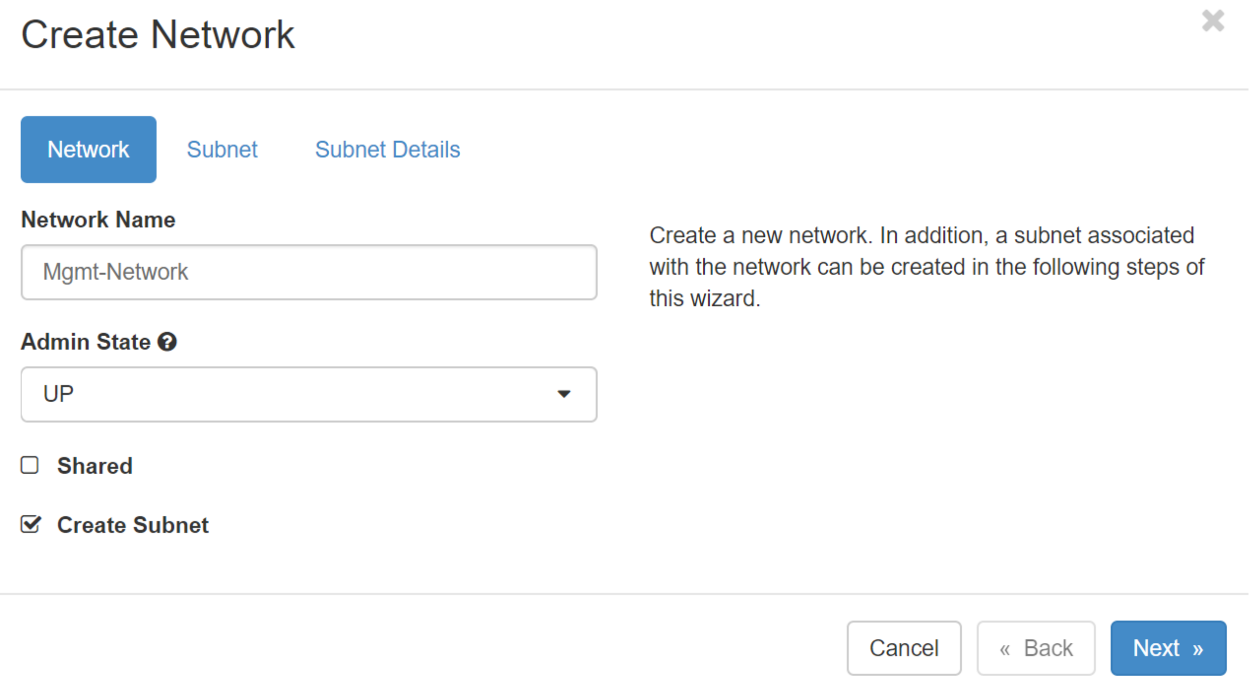
Project / Network / Networks

Networks

☐	Name	Subnets Associated	Shared	External	Status	Admin State	Actions
☐	Mgmt-Net	• **Mgmt-Net** 192.168.1.0/24	No	No	Active	UP	Edit Network
☐	Lan1-Net	• **Lan1-Net** 172.26.1.0/24	No	No	Active	UP	Edit Network
☐	External	• **External** 10.196.200.0/24	No	Yes	Active	UP	Edit Network

Displaying 3 items

2. Specify a name for this network and click on Next



Create Network

Network Name

Mgmt-Network

Admin State

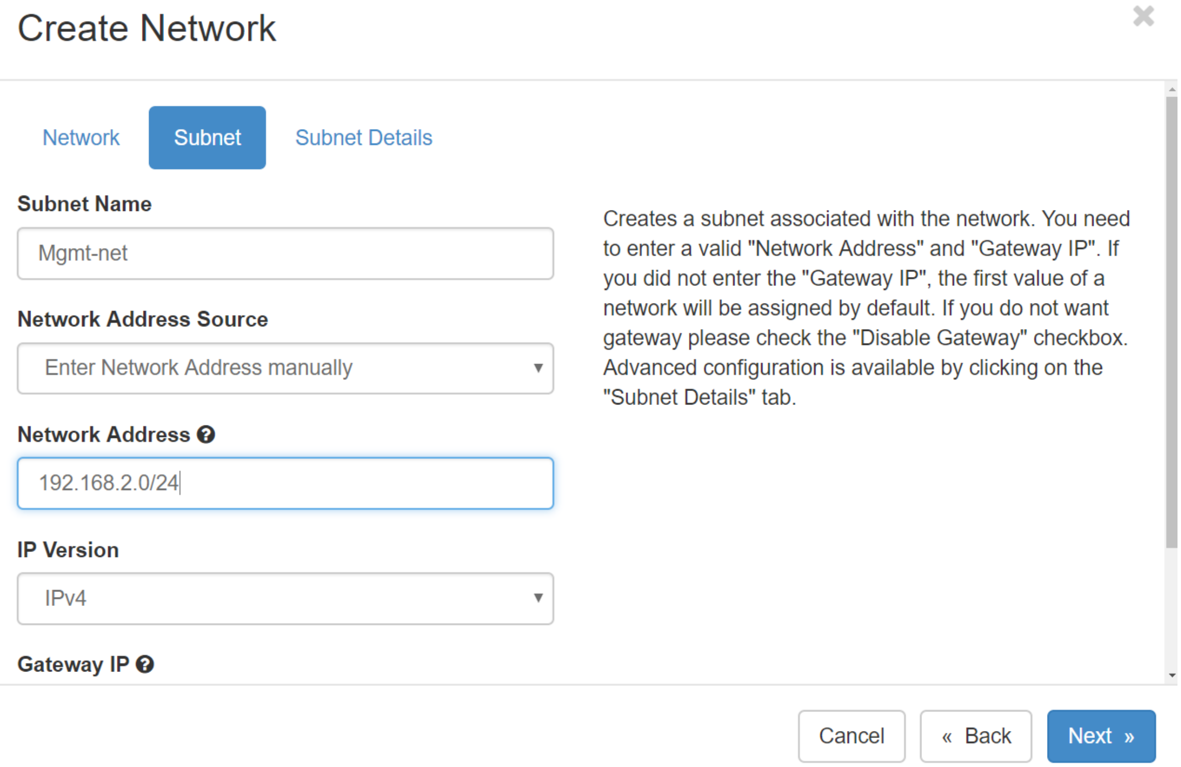
UP

☐ **Shared**

☑ **Create Subnet**

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

3. Enter the subnet information and click on Next.



Create Network

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.

Subnet Name: Mgmt-net

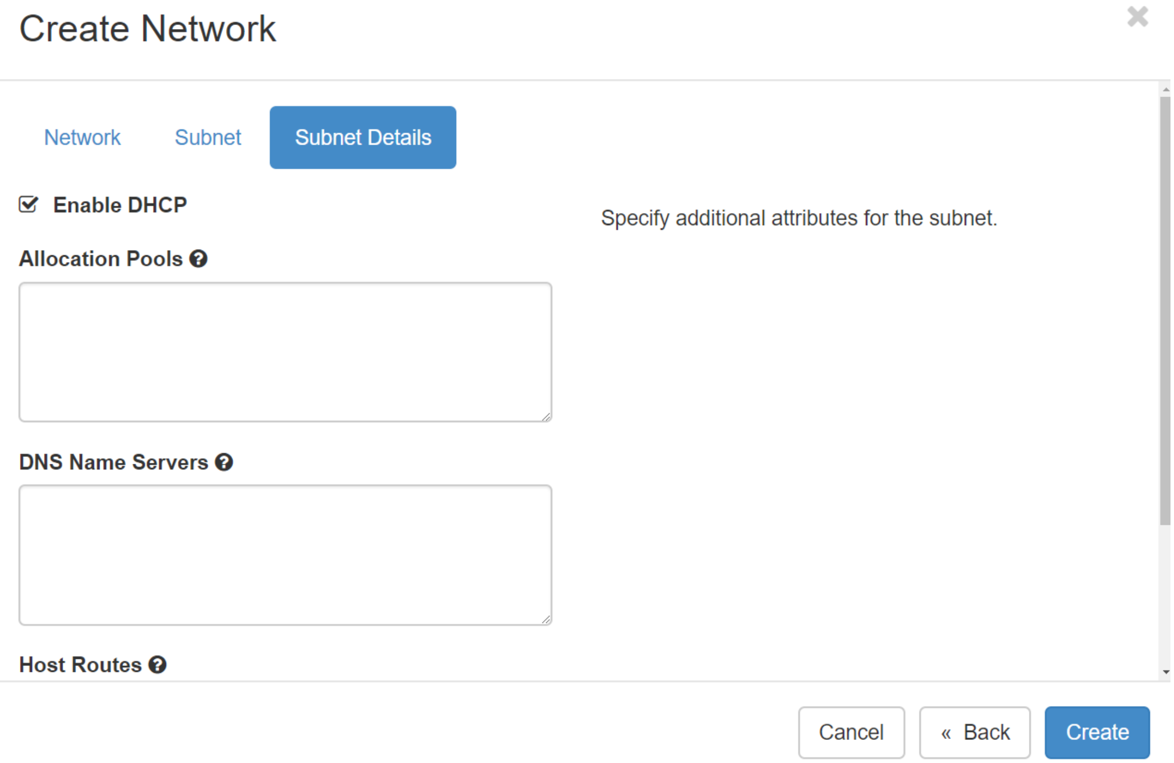
Network Address Source: Enter Network Address manually

Network Address: 192.168.2.0/24

IP Version: IPv4

Gateway IP

4. Leave all the values to default in this page and click on create.



Create Network

Specify additional attributes for the subnet.

☑ Enable DHCP

Allocation Pools

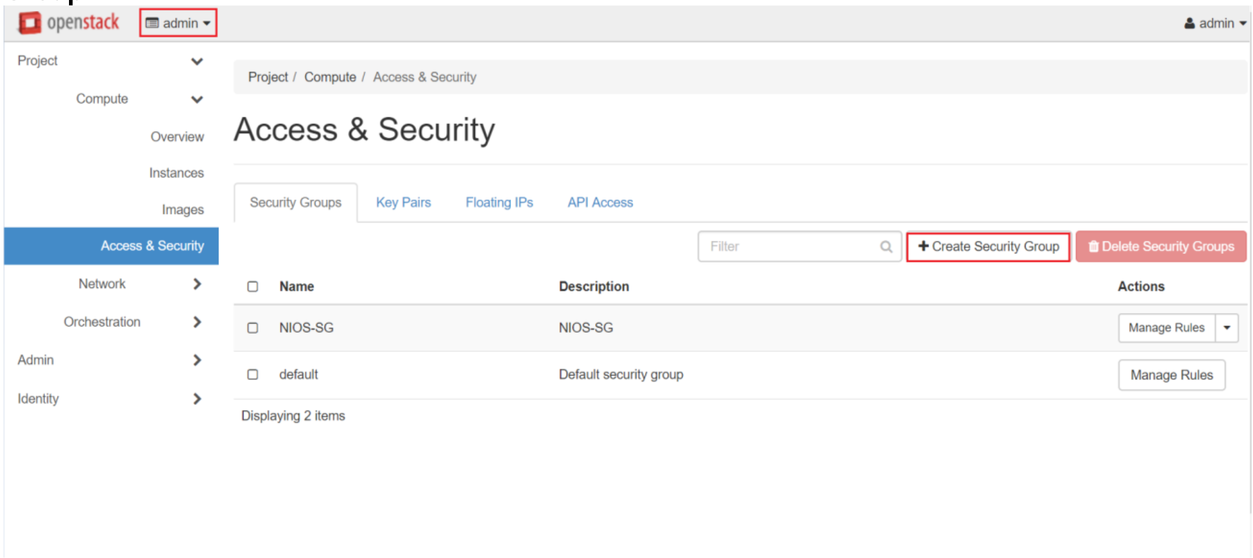
DNS Name Servers

Host Routes

5. Repeat these steps to create a second network (Lan1-Network).

Creating Security Groups

1. In the admin project, navigate to **Project → Compute → Access & Security → Create Security Group**

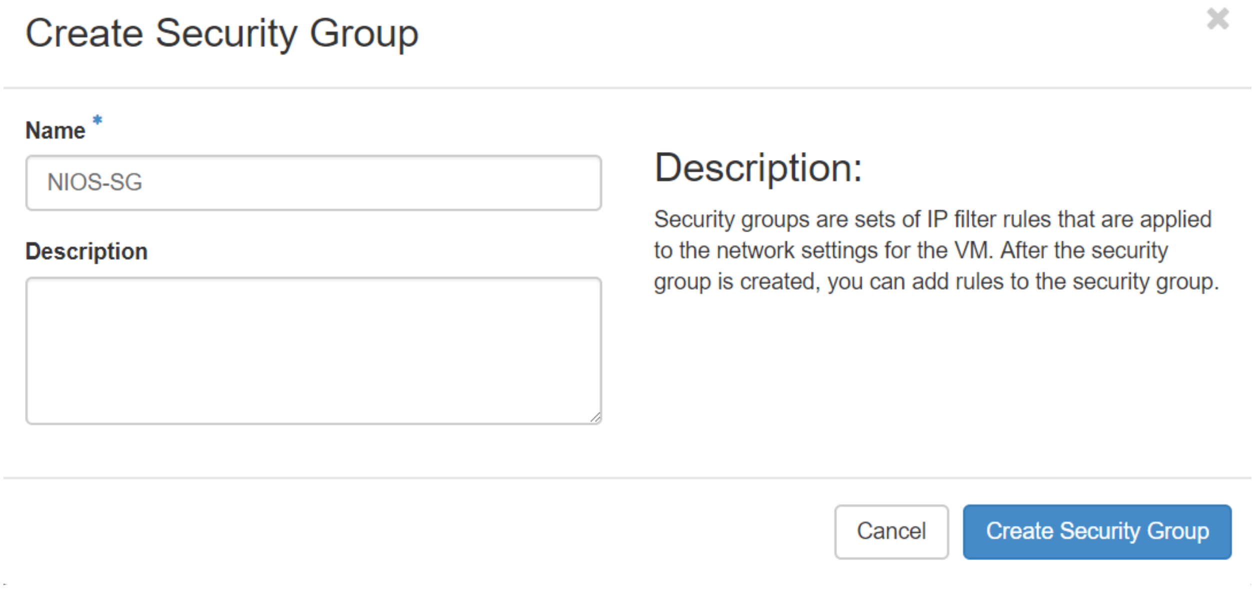


Access & Security

□	Name	Description	Actions
□	NIOS-SG	NIOS-SG	Manage Rules
□	default	Default security group	Manage Rules

Displaying 2 items

2. Enter a name for this security group and click on **Create Security Group.**



Create Security Group

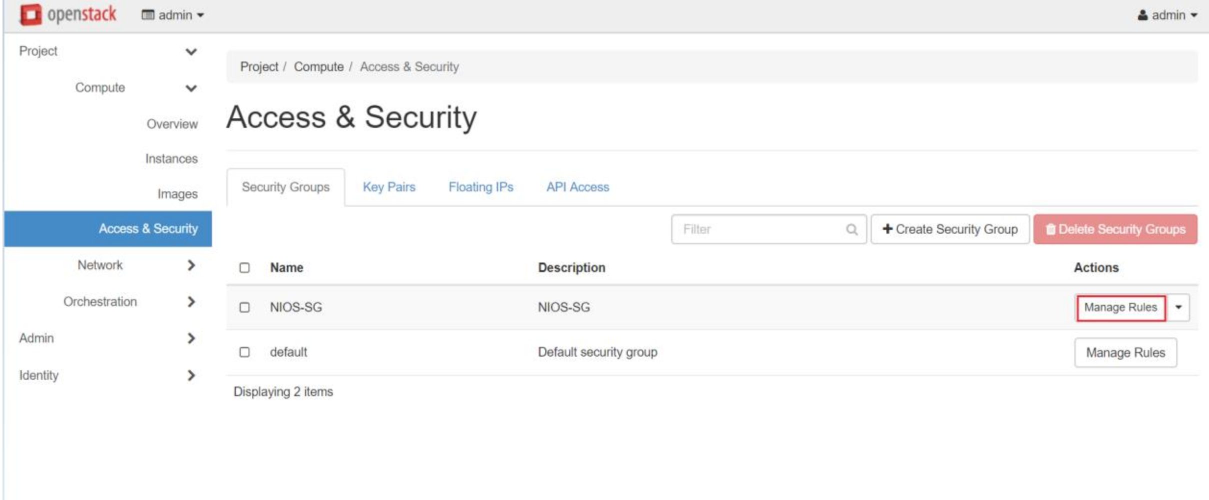
Name *: NIOS-SG

Description:

Description:

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

3. After the security group gets created, click on **Manage Rules** to edit the rules.

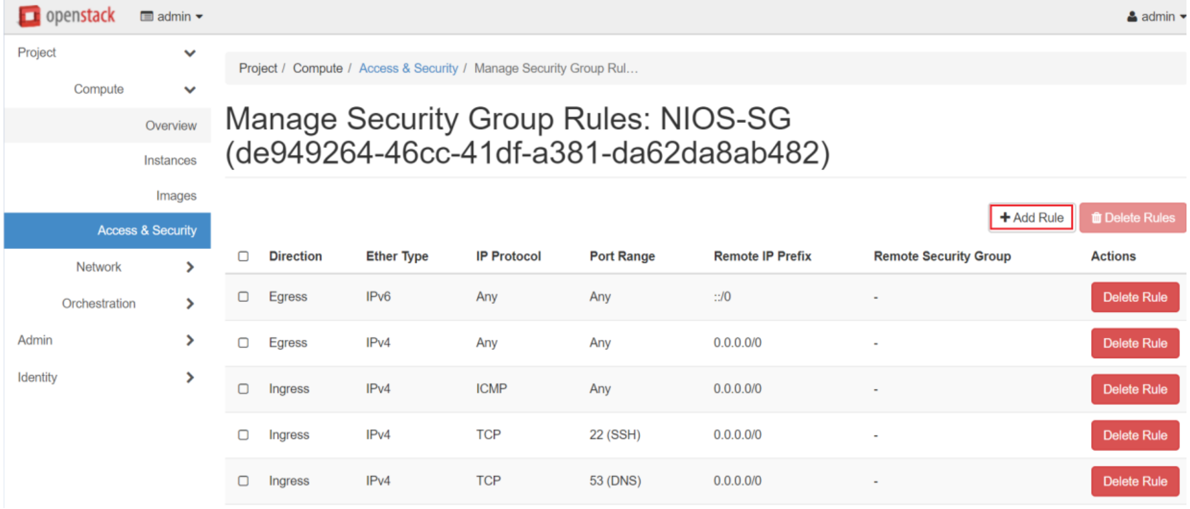


Access & Security

□	Name	Description	Actions
□	NIOS-SG	NIOS-SG	Manage Rules ▾
□	default	Default security group	Manage Rules

Displaying 2 items

4. Add following rules to the security group by clicking on **Add rule**



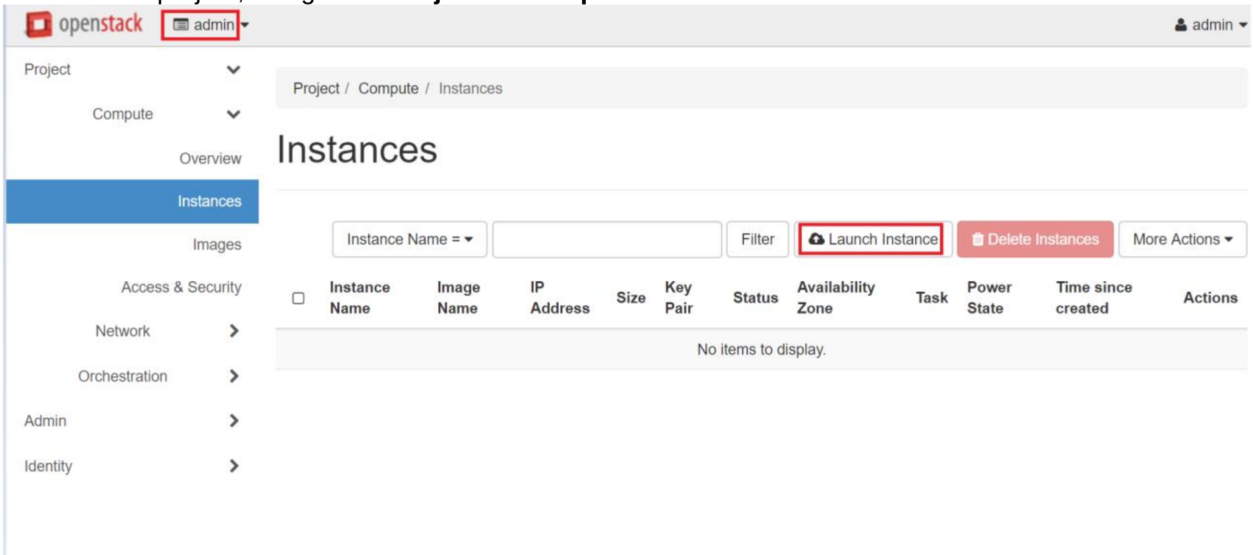
Manage Security Group Rules: NIOS-SG (de949264-46cc-41df-a381-da62da8ab482)

□	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
□	Egress	IPv6	Any	Any	::/0	-	Delete Rule
□	Egress	IPv4	Any	Any	0.0.0.0/0	-	Delete Rule
□	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
□	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	Delete Rule
□	Ingress	IPv4	TCP	53 (DNS)	0.0.0.0/0	-	Delete Rule

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix
Egress	IPv6	Any	Any	::/0
Egress	IPv4	Any	Any	0.0.0.0/0
Ingress	IPv4	ICMP	Any	0.0.0.0/0
Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0
Ingress	IPv4	TCP	53(DNS)	0.0.0.0/0
Ingress	IPv4	TCP	161	0.0.0.0/0
Ingress	IPv4	TCP	443(HTTPS)	0.0.0.0/0
Ingress	IPv4	UDP	53	0.0.0.0/0
Ingress	IPv4	UDP	161	0.0.0.0/0
Ingress	IPv4	UDP	514	0.0.0.0/0
Ingress	IPv4	UDP	1194	0.0.0.0/0
Ingress	IPv4	UDP	2114	0.0.0.0/0

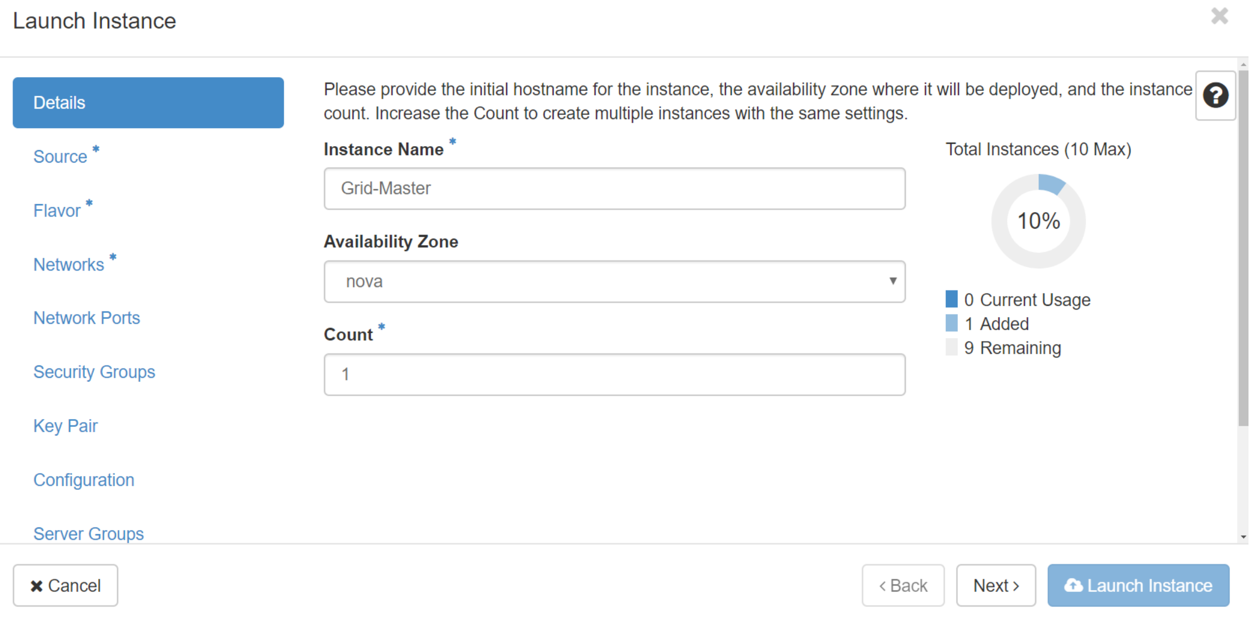
Deploying Grid Master

1. In the admin project, navigate to **Project → Compute → Instances → Launch Instance**



□	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
No items to display.											

2. Enter a name for this instance.



Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

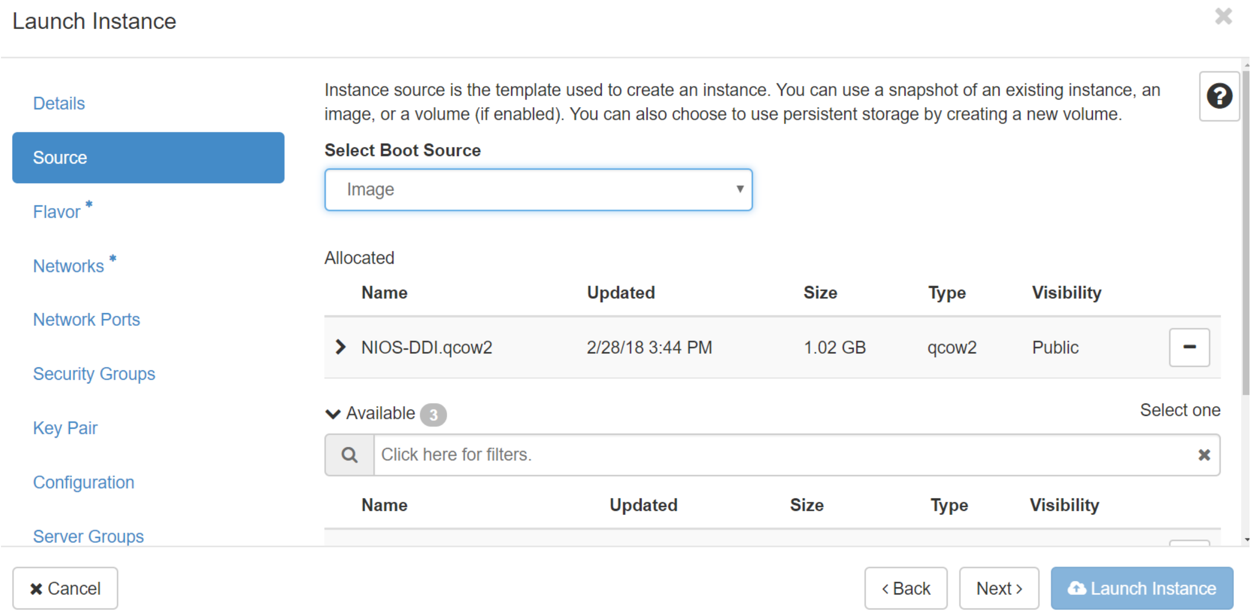
Instance Name *: Grid-Master

Availability Zone: nova

Count *: 1

- ▇ 0 Current Usage
- ▇ 1 Added
- □ 9 Remaining

3. In Source tab, select the previously created NIOS-DDI.qcow2 image and click on next



Launch Instance

Instance source is the template used to create an instance. You can use a snapshot of an existing instance, an image, or a volume (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source: Image

Allocated

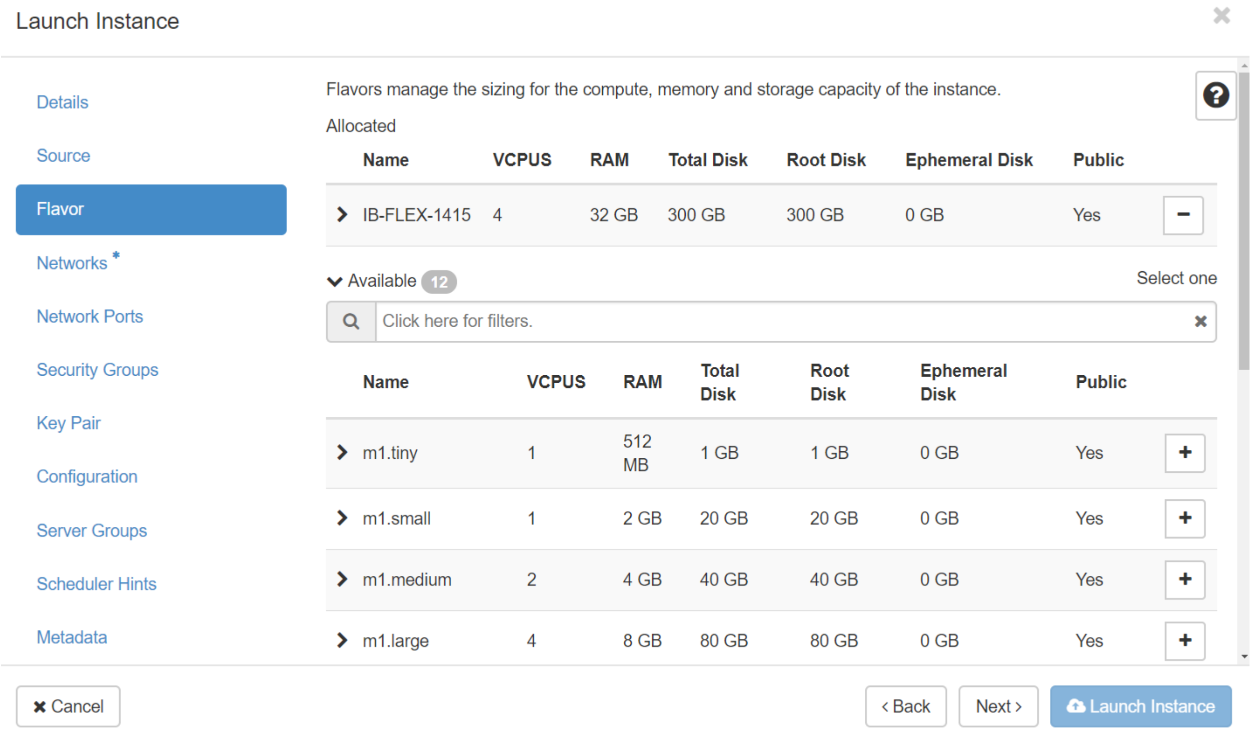
Name	Updated	Size	Type	Visibility
NIOS-DDI.qcow2	2/28/18 3:44 PM	1.02 GB	qcow2	Public

Available 3 — Select one

Click here for filters.

Name	Updated	Size	Type	Visibility

4. In Flavor tab, select the flavor which we created (IB-FLEX-1415) and click on next.



Launch Instance

Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

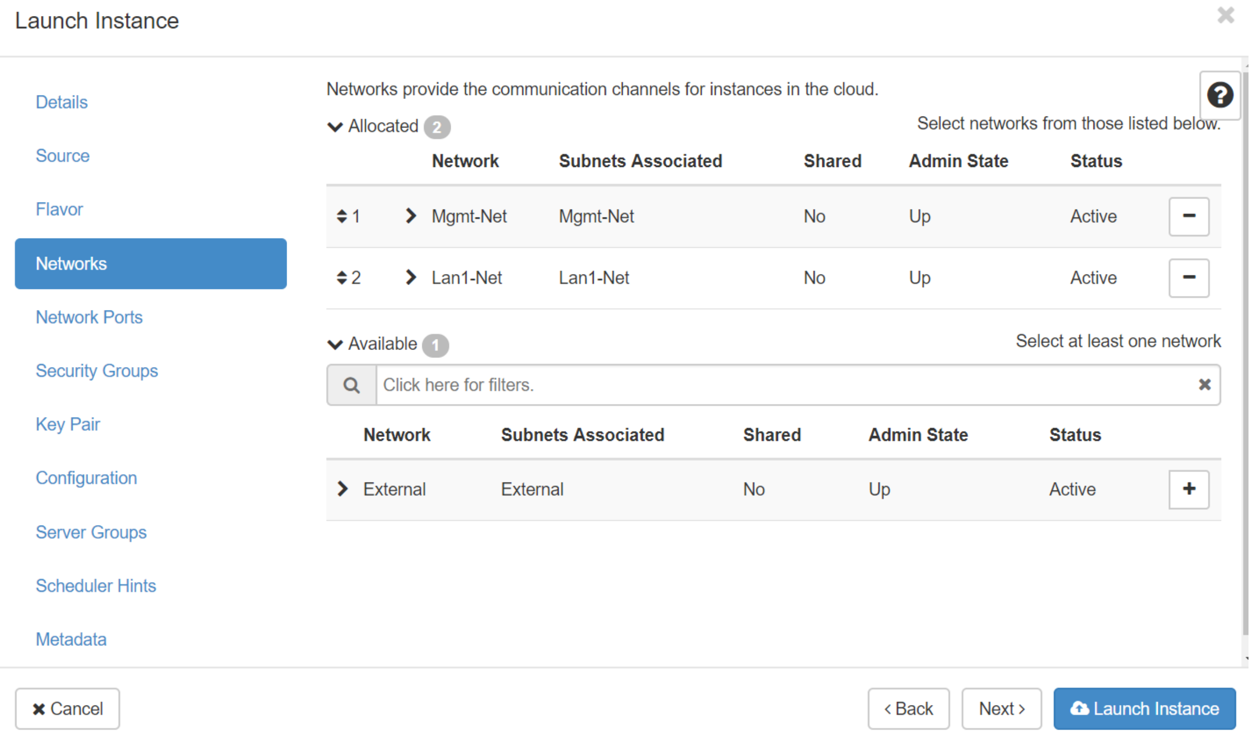
Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
IB-FLEX-1415	4	32 GB	300 GB	300 GB	0 GB	Yes

Available 12 — Select one

Click here for filters.

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes
m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes
m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes
m1.large	4	8 GB	80 GB	80 GB	0 GB	Yes

5. In the Networks the, select the 2 networks which we created and click on next. Ensure that they are selected in the correct order.



Networks provide the communication channels for instances in the cloud.

Allocated 2 — Select networks from those listed below.

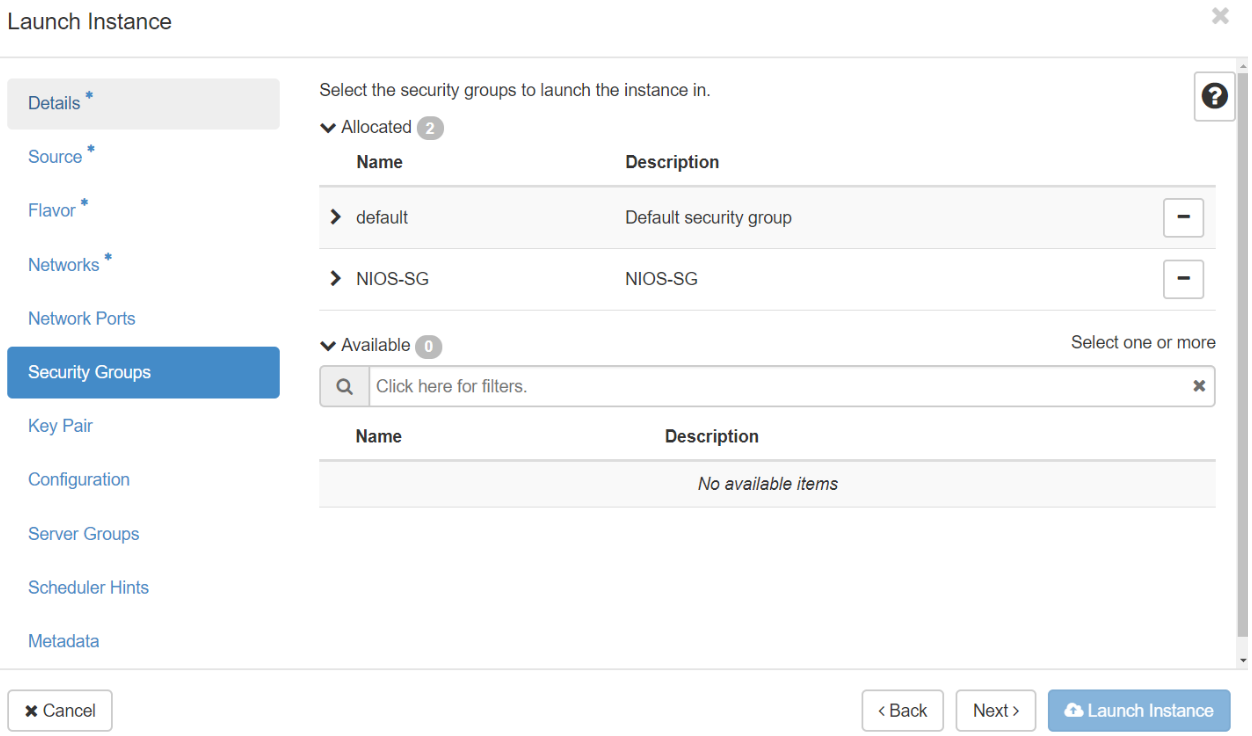
	Network	Subnets Associated	Shared	Admin State	Status	
1	Mgmt-Net	Mgmt-Net	No	Up	Active	−
2	Lan1-Net	Lan1-Net	No	Up	Active	−

Available 1 — Select at least one network

Click here for filters.

Network	Subnets Associated	Shared	Admin State	Status	
External	External	No	Up	Active	+

6. In the Security groups window, select the security group which we created and click on Launch instance.



Select the security groups to launch the instance in.

Allocated 2

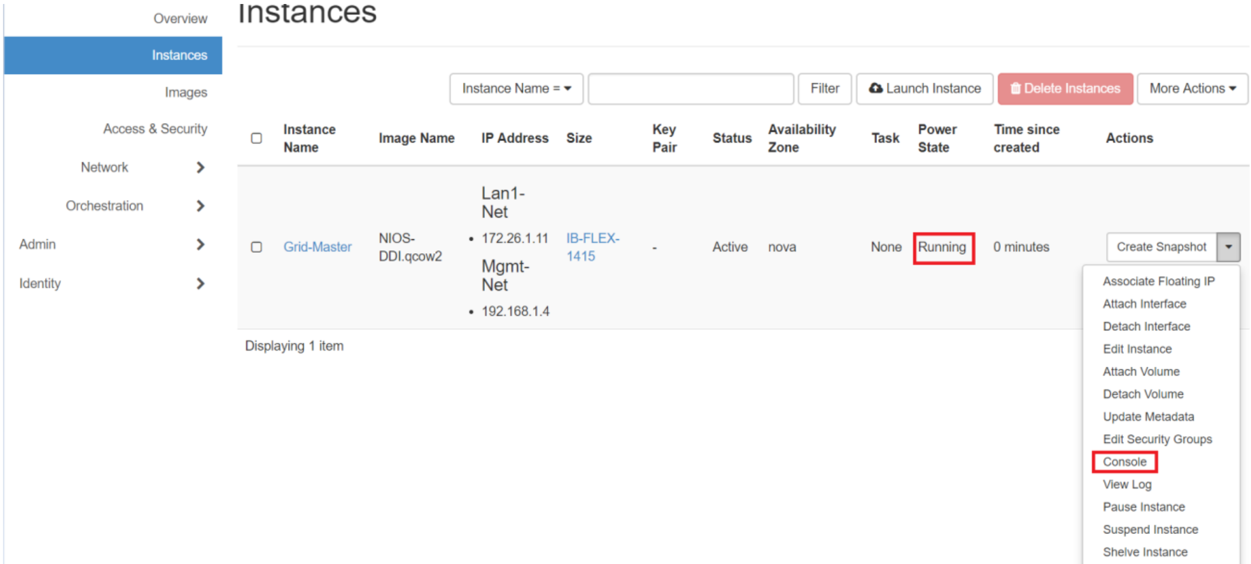
Name	Description	
default	Default security group	−
NIOS-SG	NIOS-SG	−

Available 0 — Select one or more

Click here for filters.

Name	Description
No available items	

7. Once the instance enters Running state, click on drop down menu and select the console to access the console of the instance.

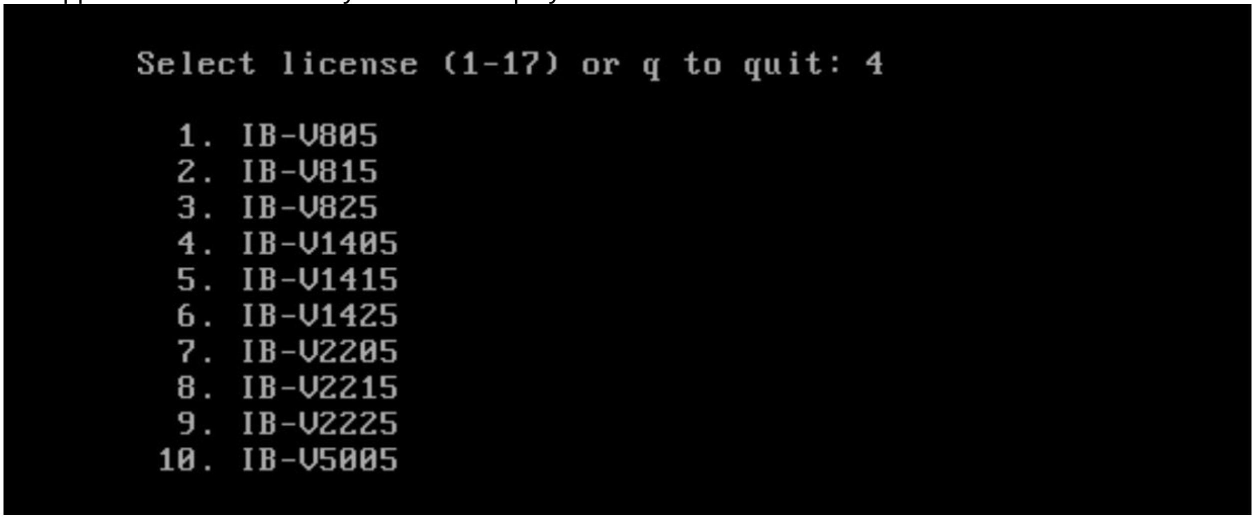


Instances

□	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	Grid-Master	NIOS-DDI.qcow2	Lan1-Net • 172.26.1.11 Mgmt-Net • 192.168.1.4	IB-FLEX-1415	-	Active	nova	None	Running	0 minutes	Create Snapshot

Displaying 1 item

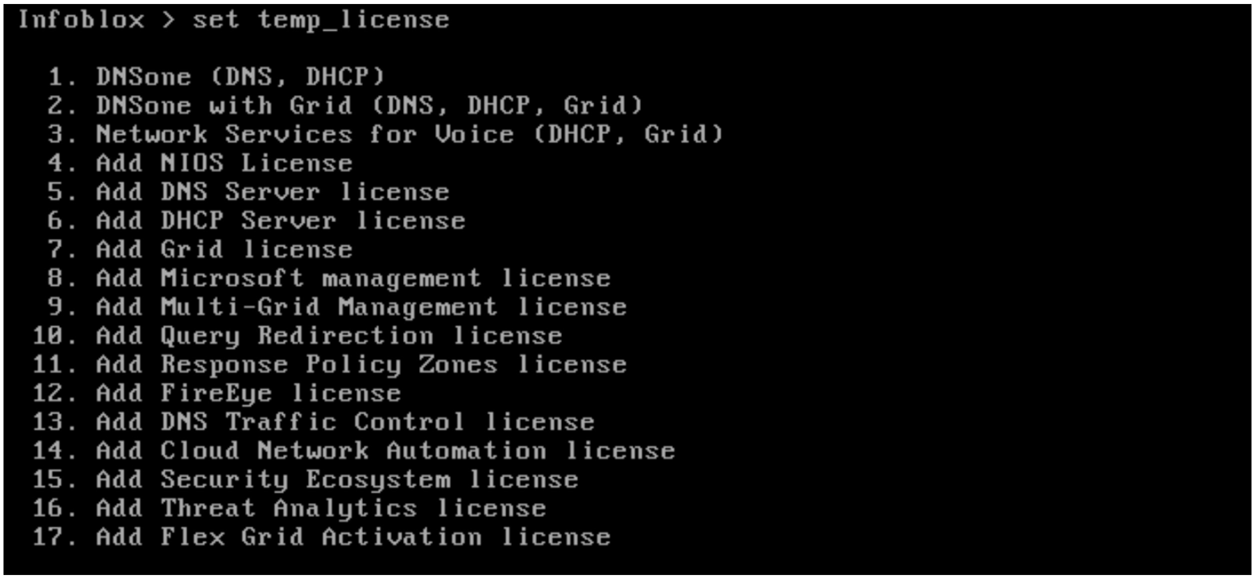
8. Login to the NIOS instance. During the license assignment step (using **set temp_license)**, select the appliance number which you wish to deploy. In our case it is 1415.



```
Select license (1-17) or q to quit: 4

   1. IB-V805
   2. IB-V815
   3. IB-V825
   4. IB-V1405
   5. IB-V1415
   6. IB-V1425
   7. IB-V2205
   8. IB-V2215
   9. IB-V2225
  10. IB-V5005
```

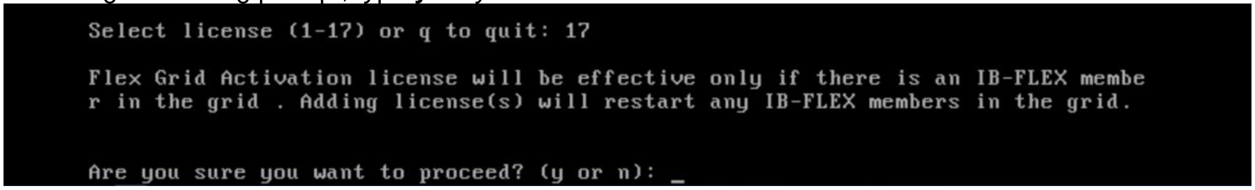
9. Set the networking using **set network** command and configure it as Grid Master

10. Navigate to the licensing option again by typing **set temp_license**. Type 17 to activate Flex Grid Activation License



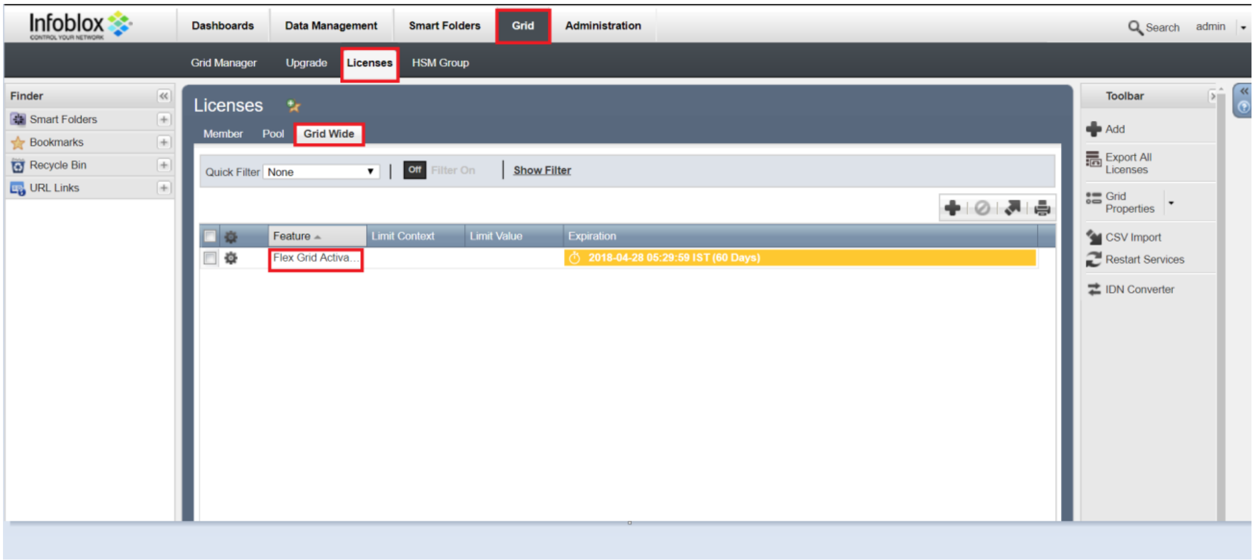
```
Infoblox > set temp_license

  1. DNSone (DNS, DHCP)
  2. DNSone with Grid (DNS, DHCP, Grid)
  3. Network Services for Voice (DHCP, Grid)
  4. Add NIOS License
  5. Add DNS Server license
  6. Add DHCP Server license
  7. Add Grid license
  8. Add Microsoft management license
  9. Add Multi-Grid Management license
 10. Add Query Redirection license
 11. Add Response Policy Zones license
 12. Add FireEye license
 13. Add DNS Traffic Control license
 14. Add Cloud Network Automation license
 15. Add Security Ecosystem license
 16. Add Threat Analytics license
 17. Add Flex Grid Activation license
```

11. You will get following prompt, type **y** for yes and hit enter.



```
     Select license (1-17) or q to quit: 17

     Flex Grid Activation license will be effective only if there is an IB-FLEX membe
     r in the grid . Adding license(s) will restart any IB-FLEX members in the grid.


     Are you sure you want to proceed? (y or n): _
```

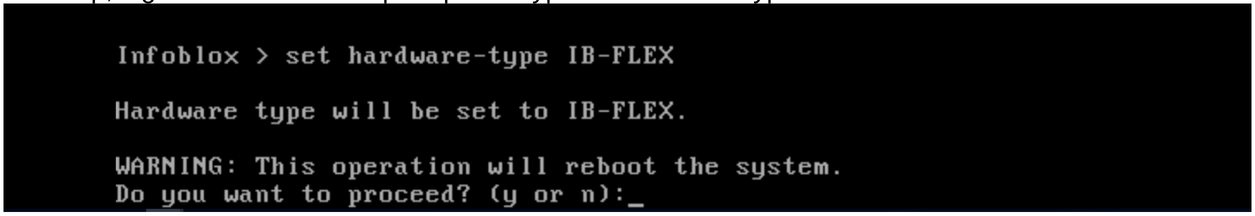
12. To verify the Flex Grid Activation license, login to the Grid GUI and navigate to **Grid → Licenses → Grid Wide**



Feature	Limit Context	Limit Value	Expiration
Flex Grid Activa...			2018-04-28 05:29:59 IST (60 Days)

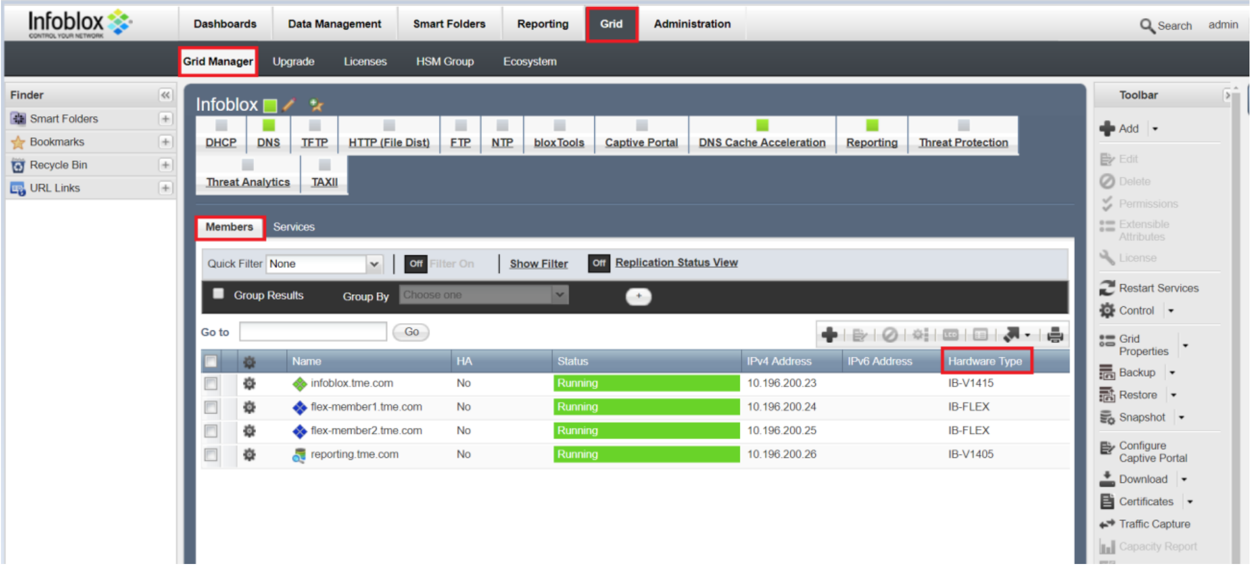
Adding a IB-FLEX member to the Grid

1. Follow the same steps (Deploying a Grid Master) and deploy a grid member. After NIOS VM boots up, login at the command prompt and type set hardware-type IB-FLEX



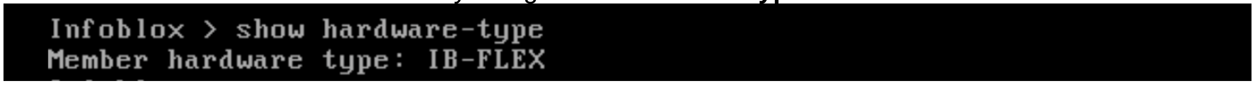
```
Infoblox > set hardware-type IB-FLEX

Hardware type will be set to IB-FLEX.

WARNING: This operation will reboot the system.
Do you want to proceed? (y or n):_
```

2. Type y for yes to install the IB-FLEX license. NIOS VM will reboot post license installation.
3. After reboot set the networking and add the NIOS to the grid using **set network** command.
4. You can verify the IB-FLEX member type in the grid by navigating to **Grid → Grid Manager → Members → Hardware Type**



Name	HA	Status	IPv4 Address	IPv6 Address	Hardware Type
infoblox.tme.com	No	Running	10.196.200.23		IB-V1415
flex-member1.tme.com	No	Running	10.196.200.24		IB-FLEX
flex-member2.tme.com	No	Running	10.196.200.25		IB-FLEX
reporting.tme.com	No	Running	10.196.200.26		IB-V1405

5. IB-FLEX license can also be verified by using **show hardware-type** command.

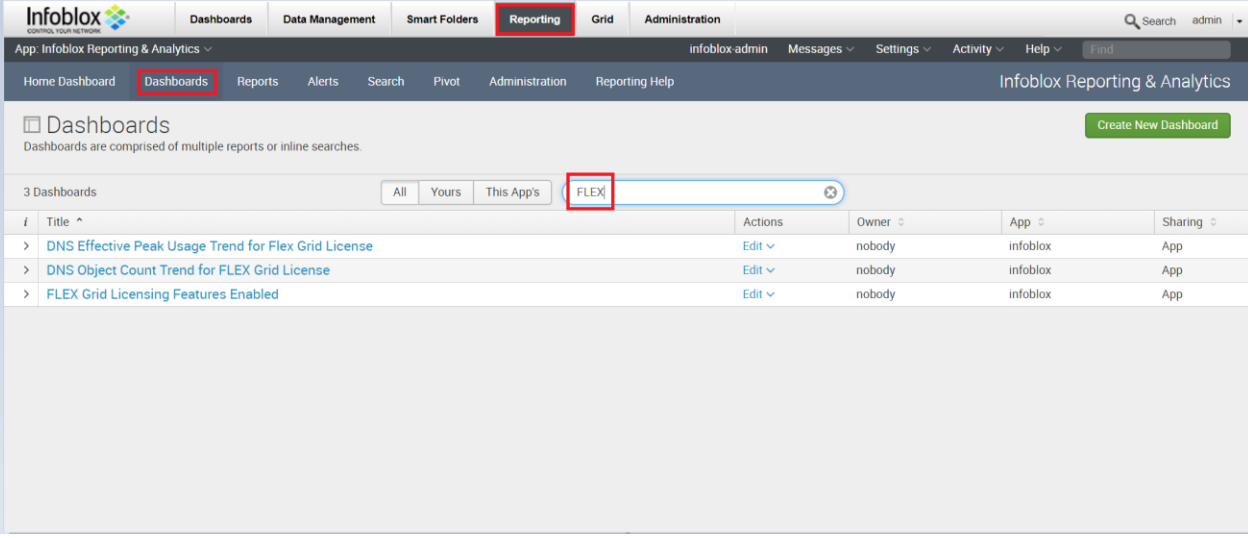


```
Infoblox > show hardware-type
Member hardware type: IB-FLEX
```

Using Reporting appliance to get IB-FLEX reports

The Infoblox reporting appliance has multiple in-built reports which gives insights of IB-FLEX members. To leverage IB-FLEX reports deploy a reporting appliance and add it to the grid.

1. Navigate to **Reporting → Dashboards** and type Flex in the search bar.



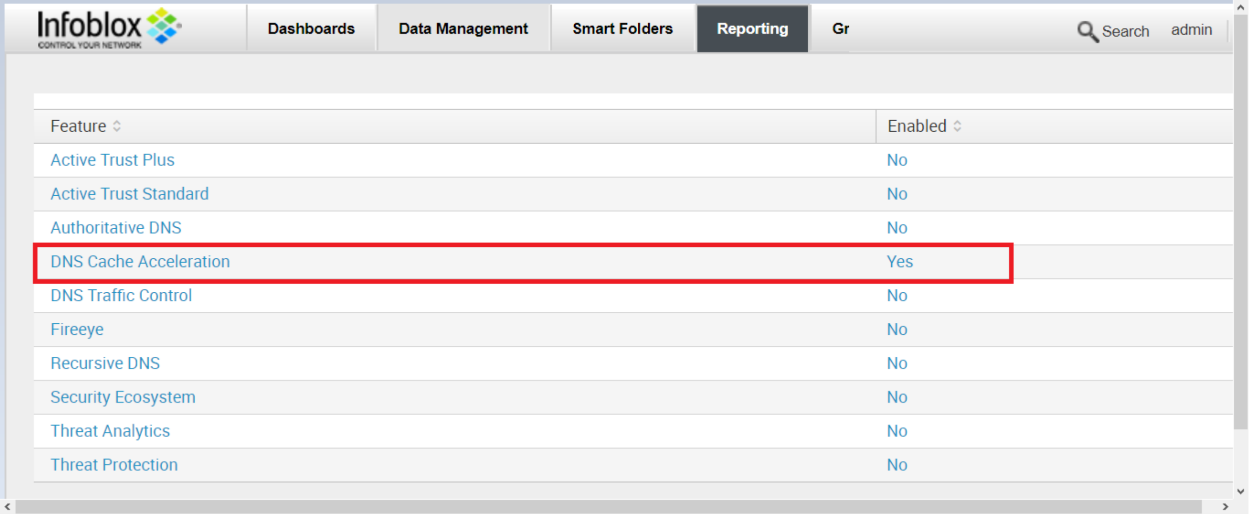
Dashboards

Dashboards are comprised of multiple reports or inline searches.

3 Dashboards

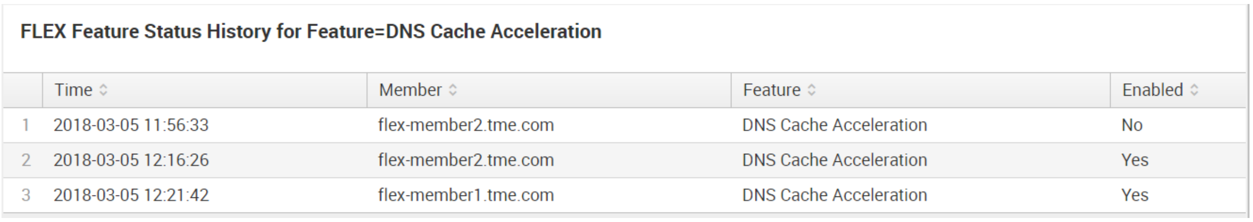
i	Title	Actions	Owner	App	Sharing
>	DNS Effective Peak Usage Trend for Flex Grid License	Edit	nobody	infoblox	App
>	DNS Object Count Trend for FLEX Grid License	Edit	nobody	infoblox	App
>	FLEX Grid Licensing Features Enabled	Edit	nobody	infoblox	App

2. Select the 3rd report (**Flex Grid Licensing Features Enabled**) to get details of IB-FLEX members.
3. This report will list all the features. Further you can check which features are enabled and on which member.



Feature	Enabled
Active Trust Plus	No
Active Trust Standard	No
Authoritative DNS	No
DNS Cache Acceleration	Yes
DNS Traffic Control	No
Fireeye	No
Recursive DNS	No
Security Ecosystem	No
Threat Analytics	No
Threat Protection	No

4. For example, click on DNS Cache Acceleration to get the list of members on which this feature is enabled.

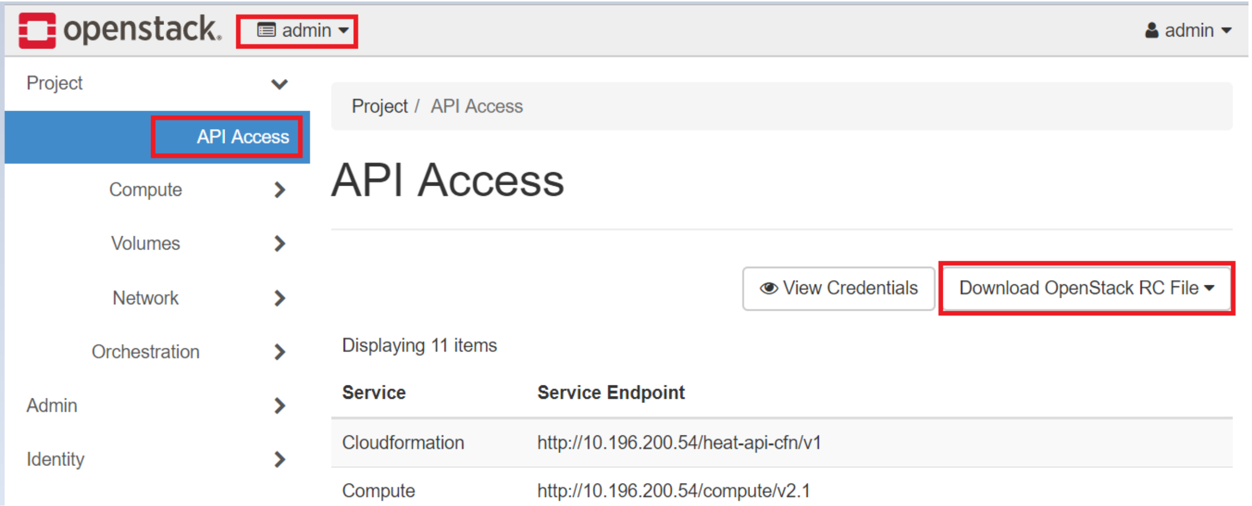


FLEX Feature Status History for Feature=DNS Cache Acceleration

	Time	Member	Feature	Enabled
1	2018-03-05 11:56:33	flex-member2.tme.com	DNS Cache Acceleration	No
2	2018-03-05 12:16:26	flex-member2.tme.com	DNS Cache Acceleration	Yes
3	2018-03-05 12:21:42	flex-member1.tme.com	DNS Cache Acceleration	Yes

Deploying IB-FLEX appliance on OpenStack using Ansible

1. Download the OpenStack RC file from the horizon by navigating to **admin → Project → API access**

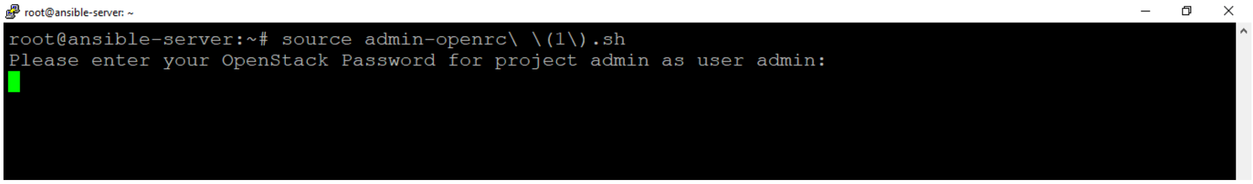


API Access

Displaying 11 items

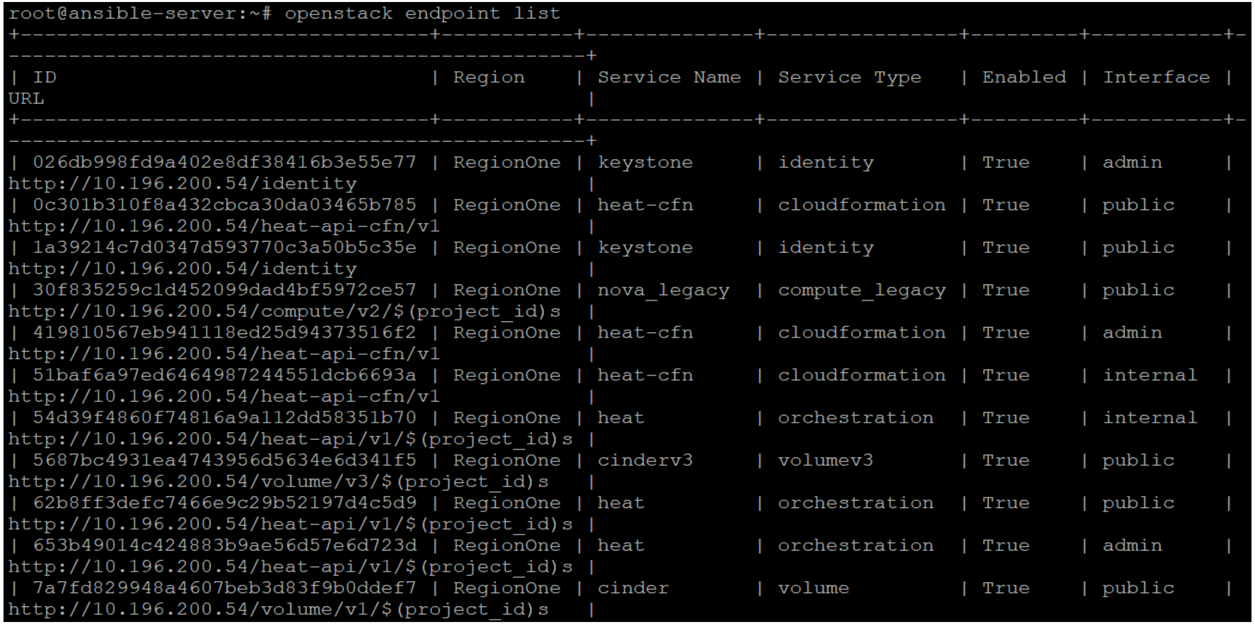
Service	Service Endpoint
Cloudformation	http://10.196.200.54/heat-api-cfn/v1
Compute	http://10.196.200.54/compute/v2.1

2. Copy the OpenStack RCv2 file to a Linux machine with Ansible installed and source it. You will be asked to enter your admin tenant password.



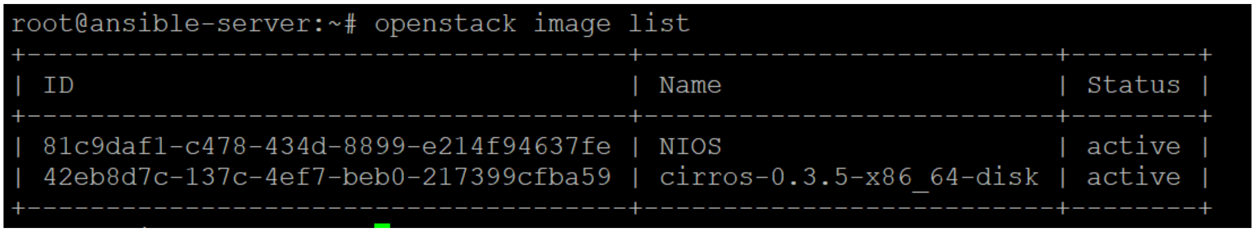
```
root@ansible-server:~# source admin-openrc\ \(1\).sh
Please enter your OpenStack Password for project admin as user admin:
```

3. To verify, OpenStack connectivity,
    a. first install openstack client by running **`apt-get install openstack-client`**
4. Execute **`openstack endpoint list`** command to get the list of openstack endpoints.



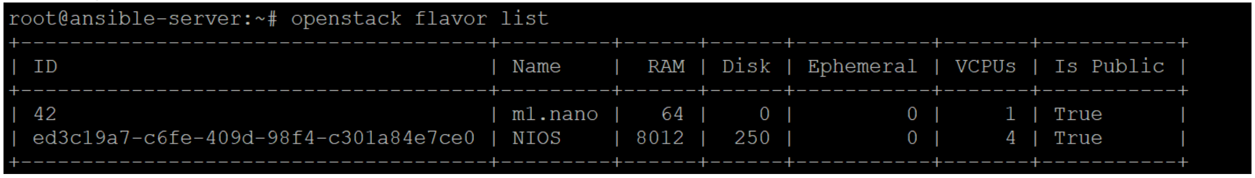
```
root@ansible-server:~# openstack endpoint list
+----------------------------------+-----------+--------------+----------------+---------+-----------+-
-----------------------------------------------+
| ID                               | Region    | Service Name | Service Type   | Enabled | Interface |
URL                                            |
+----------------------------------+-----------+--------------+----------------+---------+-----------+-
-----------------------------------------------+
| 026db998fd9a402e8df38416b3e55e77 | RegionOne | keystone     | identity       | True    | admin     |
http://10.196.200.54/identity                  |
| 0c301b310f8a432cbca30da03465b785 | RegionOne | heat-cfn     | cloudformation | True    | public    |
http://10.196.200.54/heat-api-cfn/v1           |
| 1a39214c7d0347d593770c3a50b5c35e | RegionOne | keystone     | identity       | True    | public    |
http://10.196.200.54/identity                  |
| 30f835259c1d452099dad4bf5972ce57 | RegionOne | nova_legacy  | compute_legacy | True    | public    |
http://10.196.200.54/compute/v2/$(project_id)s |
| 419810567eb941118ed25d94373516f2 | RegionOne | heat-cfn     | cloudformation | True    | admin     |
http://10.196.200.54/heat-api-cfn/v1           |
| 51baf6a97ed6464987244551dcb6693a | RegionOne | heat-cfn     | cloudformation | True    | internal  |
http://10.196.200.54/heat-api-cfn/v1           |
| 54d39f4860f74816a9a112dd58351b70 | RegionOne | heat         | orchestration  | True    | internal  |
http://10.196.200.54/heat-api/v1/$(project_id)s |
| 5687bc4931ea4743956d5634e6d341f5 | RegionOne | cinderv3     | volumev3       | True    | public    |
http://10.196.200.54/volume/v3/$(project_id)s  |
| 62b8ff3defc7466e9c29b52197d4c5d9 | RegionOne | heat         | orchestration  | True    | public    |
http://10.196.200.54/heat-api/v1/$(project_id)s |
| 653b49014c424883b9ae56d57e6d723d | RegionOne | heat         | orchestration  | True    | admin     |
http://10.196.200.54/heat-api/v1/$(project_id)s |
| 7a7fd829948a4607beb3d83f9b0ddef7 | RegionOne | cinder       | volume         | True    | public    |
http://10.196.200.54/volume/v1/$(project_id)s  |
```

5. Run following commands one by one and copy the output to a document
    a. To get image id use `#openstack image list` command



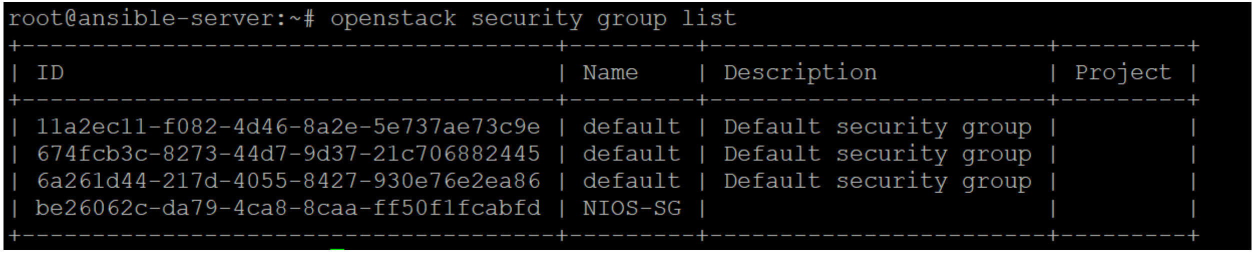
```
root@ansible-server:~# openstack image list
+--------------------------------------+--------------------------+--------+
| ID                                   | Name                     | Status |
+--------------------------------------+--------------------------+--------+
| 81c9daf1-c478-434d-8899-e214f94637fe | NIOS                     | active |
| 42eb8d7c-137c-4ef7-beb0-217399cfba59 | cirros-0.3.5-x86_64-disk | active |
+--------------------------------------+--------------------------+--------+
```

b. To get flavor id use #`openstack flavor list` command



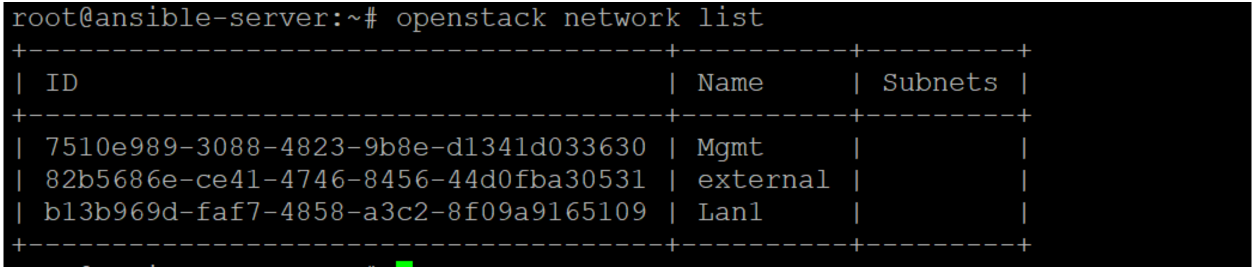
```
root@ansible-server:~# openstack flavor list
+--------------------------------------+---------+------+------+-----------+-------+-----------+
| ID                                   | Name    |  RAM | Disk | Ephemeral | VCPUs | Is Public |
+--------------------------------------+---------+------+------+-----------+-------+-----------+
| 42                                   | m1.nano |   64 |    0 |         0 |     1 | True      |
| ed3c19a7-c6fe-409d-98f4-c301a84e7ce0 | NIOS    | 8012 |  250 |         0 |     4 | True      |
+--------------------------------------+---------+------+------+-----------+-------+-----------+
```

c. To get security group id use #`openstack security group list`



```
root@ansible-server:~# openstack security group list
+--------------------------------------+---------+------------------------+---------+
| ID                                   | Name    | Description            | Project |
+--------------------------------------+---------+------------------------+---------+
| 11a2ec11-f082-4d46-8a2e-5e737ae73c9e | default | Default security group |         |
| 674fcb3c-8273-44d7-9d37-21c706882445 | default | Default security group |         |
| 6a261d44-217d-4055-8427-930e76e2ea86 | default | Default security group |         |
| be26062c-da79-4ca8-8caa-ff50f1fcabfd | NIOS-SG |                        |         |
+--------------------------------------+---------+------------------------+---------+
```

d. To get nic ids use #`openstack network list`



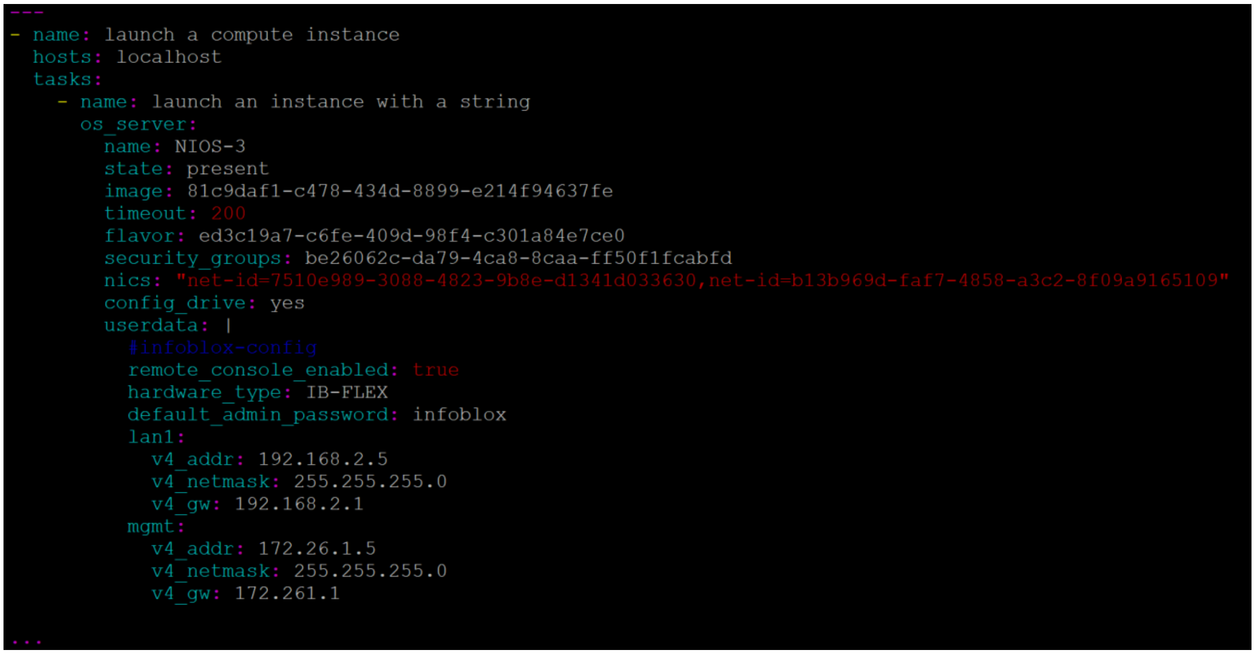
```
root@ansible-server:~# openstack network list
+--------------------------------------+----------+---------+
| ID                                   | Name     | Subnets |
+--------------------------------------+----------+---------+
| 7510e989-3088-4823-9b8e-d1341d033630 | Mgmt     |         |
| 82b5686e-ce41-4746-8456-44d0fba30531 | external |         |
| b13b969d-faf7-4858-a3c2-8f09a9165109 | Lan1     |         |
+--------------------------------------+----------+---------+
```

6. Create a NIOS_deploy.yaml file with following contents. Replace the value in red with the output of the commands discussed above.

   This yaml file spins a NIOS instance and does initial configuration like

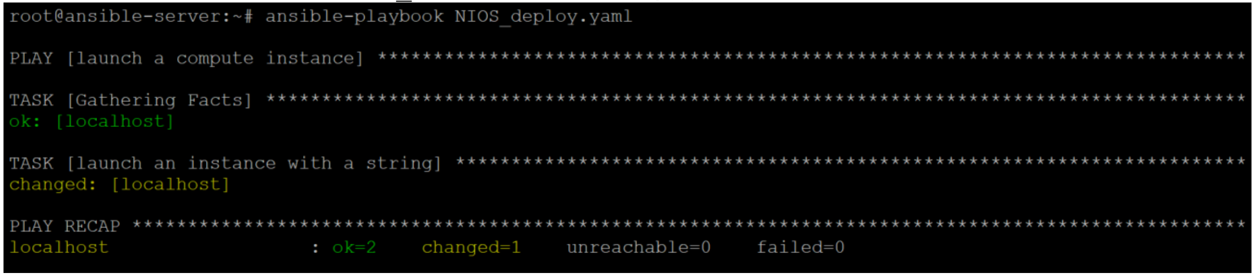
   - Setting the Hardware type to IB-FLEX.
   - Configures the lan1 and mgmt IP addresses.

```
---

- name: launch a compute instance
  hosts: localhost
  tasks:
    - name: launch an instance with a string
      os_server:
        name: NIOS-3
        state: present
        image: 81c9daf1-c478-434d-8899-e214f94637fe
        timeout: 200
        flavor: ed3c19a7-c6fe-409d-98f4-c301a84e7ce0
        security_groups: be26062c-da79-4ca8-8caa-ff50f1fcabfd
        nics: "net-id=1c12a43f-fd29-4c0c-bc51-b52f4d59f757,net-id=e4469c64-49cd-4109-91e9-be1e9c271155"
        config_drive: yes
        userdata: |
          #infoblox-config
          remote_console_enabled: true
          hardware_type: IB-FLEX
          default_admin_password: infoblox
          lan1:
            v4_addr: lan1_ip_address
            v4_netmask: lan1_subnet_mask
            v4_gw: lan1_gw
          mgmt:
            v4_addr:Mgmt_ip_address
            v4_netmask:Mgmt_subnet_mask
            v4_gw:Mgmt_gw
…
```



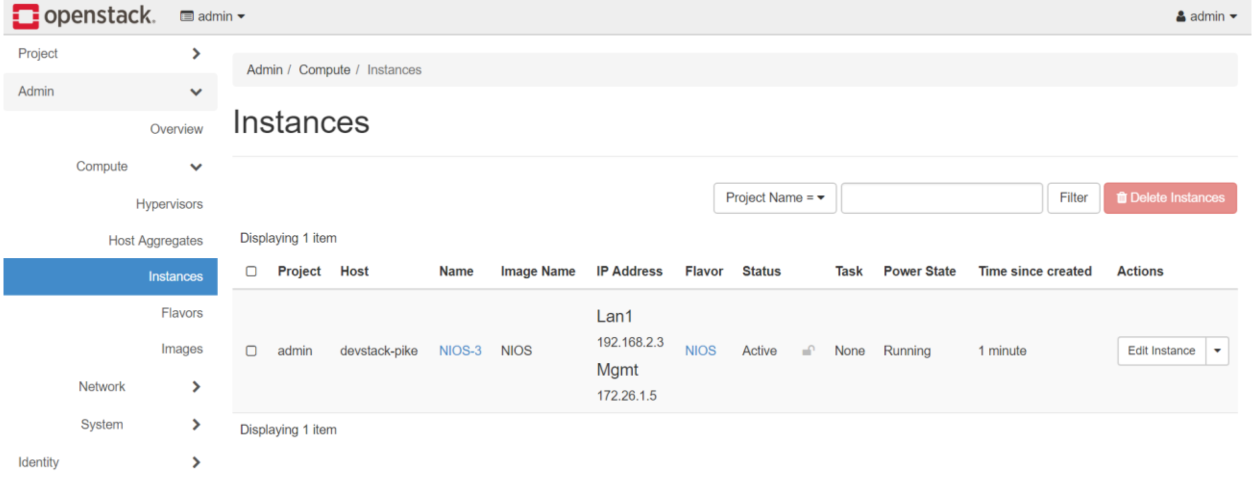
```
---
- name: launch a compute instance
  hosts: localhost
  tasks:
    - name: launch an instance with a string
      os_server:
        name: NIOS-3
        state: present
        image: 81c9daf1-c478-434d-8899-e214f94637fe
        timeout: 200
        flavor: ed3c19a7-c6fe-409d-98f4-c301a84e7ce0
        security_groups: be26062c-da79-4ca8-8caa-ff50f1fcabfd
        nics: "net-id=7510e989-3088-4823-9b8e-d1341d033630,net-id=b13b969d-faf7-4858-a3c2-8f09a9165109"
        config_drive: yes
        userdata: |
          #infoblox-config
          remote_console_enabled: true
          hardware_type: IB-FLEX
          default_admin_password: infoblox
          lan1:
            v4_addr: 192.168.2.5
            v4_netmask: 255.255.255.0
            v4_gw: 192.168.2.1
          mgmt:
            v4_addr: 172.26.1.5
            v4_netmask: 255.255.255.0
            v4_gw: 172.261.1

...
```

7. Run `ansible-playbook NIOS_deploy.yaml` to deploy NIOS instance



```
root@ansible-server:~# ansible-playbook NIOS_deploy.yaml

PLAY [launch a compute instance] *****************************************************************************

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

TASK [launch an instance with a string] **********************************************************************
changed: [localhost]

PLAY RECAP ***************************************************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0
```

8. Login to the Horizon dashboard to verify that the instance has been created.



Admin / Compute / Instances

Instances

Displaying 1 item

□	Project	Host	Name	Image Name	IP Address	Flavor	Status		Task	Power State	Time since created	Actions
□	admin	devstack-pike	NIOS-3	NIOS	Lan1 192.168.2.3 Mgmt 172.26.1.5	NIOS	Active		None	Running	1 minute	Edit Instance ▾

Displaying 1 item

Deploying IB-FLEX appliance on KVM

Infoblox vNIOS for KVM is a virtual appliance designed for KVM (Kernel-based Virtual Machine) hypervisor. Infoblox vNIOS for KVM enables you to deploy large, robust, manageable and cost effective Infoblox Grids. Infoblox vNIOS is supported only on RHEL and Centos based KVM.

Configuring KVM

1. Install following packages on either CentOS or RHEL to install and configure KVM

```
# yum install qemu-kvm qemu-img libvirt libvirt-python libvirt-client
virt-install virt-viewer bridge-utils
```

2. Start and enable the libvirtd service

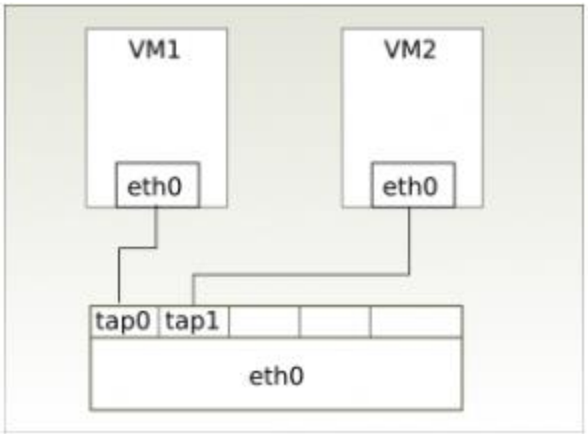
```
# systemctl start libvirtd
# systemctl enable libvirtd
```

3. Configure Macvtap interfaces

   The **Macvlan** driver is a separate Linux kernel driver that the Macvtap driver depends on. Macvlan makes it possible to create virtual network interfaces that "cling on" a physical network interface. Each virtual interface has its own MAC address distinct from the physical interface's MAC address. Frames sent to or from the virtual interfaces are mapped to the physical interface, which is called the lower interface.

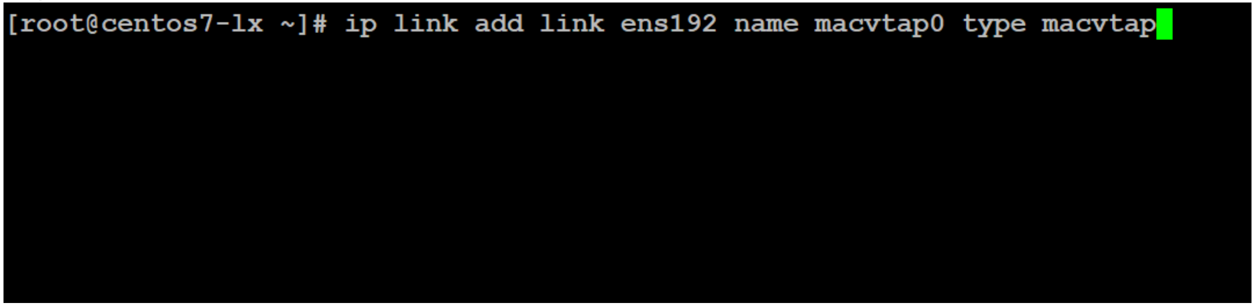
   A Tap interface is a software-only interface. Instead of passing frames to and from a physical Ethernet card, the frames are read and written by a user space program. The kernel makes the Tap interface available via the /dev/tapN device file, where N is the index of the network interface.

   A Macvtap interface combines the properties of these two; it is a virtual interface with a tap-like software interface. A Macvtap interface can be created using the `ip` command



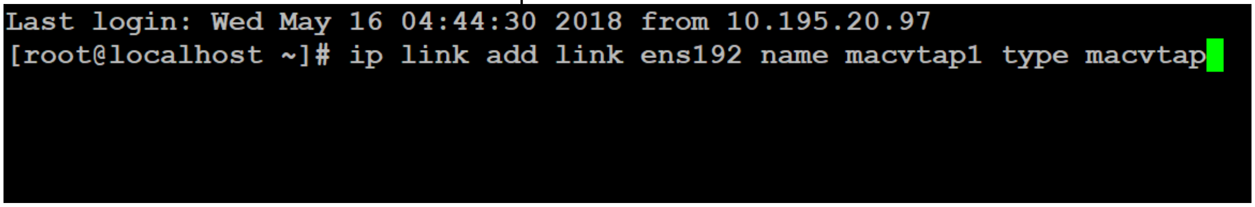
Tap interfaces

4. Use # `ip link add link ens192 name macvtap0 type macvtap` command to create macvtap interface.
   Replace `ens192` with the interface name.

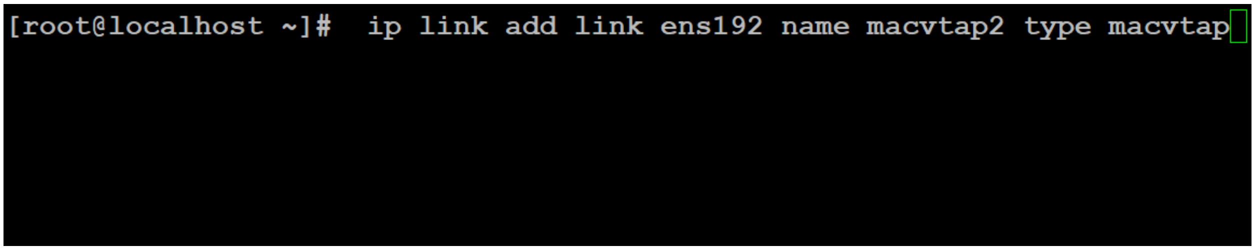


```
[root@centos7-lx ~]# ip link add link ens192 name macvtap0 type macvtap
```

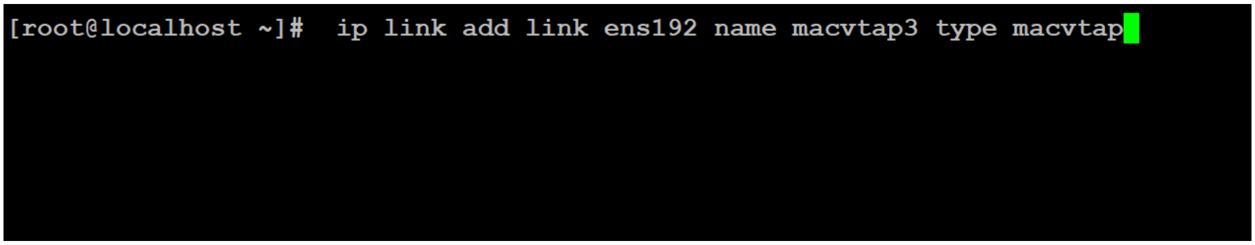
5. Use the same command to create 3 macvtap interfaces



```
Last login: Wed May 16 04:44:30 2018 from 10.195.20.97
[root@localhost ~]# ip link add link ens192 name macvtap1 type macvtap
```

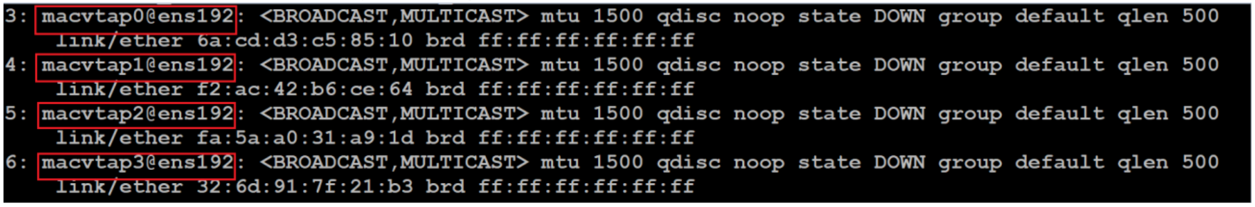


```
[root@localhost ~]#  ip link add link ens192 name macvtap2 type macvtap
```



```
[root@localhost ~]#  ip link add link ens192 name macvtap3 type macvtap
```

6. Run `#ip a` command to verify macvtap interfaces have been created.

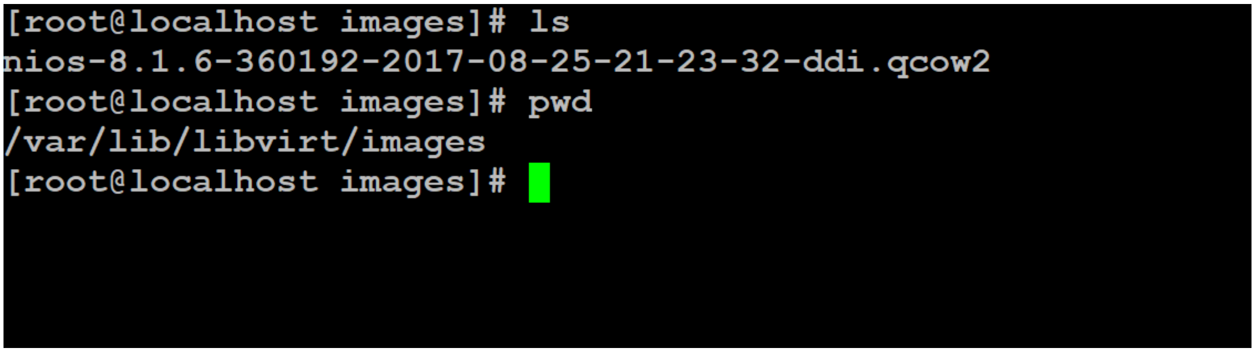


```
3: macvtap0@ens192: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/ether 6a:cd:d3:c5:85:10 brd ff:ff:ff:ff:ff:ff
4: macvtap1@ens192: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/ether f2:ac:42:b6:ce:64 brd ff:ff:ff:ff:ff:ff
5: macvtap2@ens192: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/ether fa:5a:a0:31:a9:1d brd ff:ff:ff:ff:ff:ff
6: macvtap3@ens192: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/ether 32:6d:91:7f:21:b3 brd ff:ff:ff:ff:ff:ff
```

7. Make a note of the MAC addresses of the macvtap interfaces.

Deploying VNIOS on KVM

1. Downloading and uploading the NIOS image
   Depending on which KVM Hypervisor you are using, download the NIOS qcow2 image from the Infoblox Support site and upload the qcow2 file(s) for the specified vNIOS virtual appliance model to the KVM/libvirt environment. This deployment guide assumes that vNIOS will be copied to **/var/lib/libvirt/images** directory.



```
[root@localhost images]# ls
nios-8.1.6-360192-2017-08-25-21-23-32-ddi.qcow2
[root@localhost images]# pwd
/var/lib/libvirt/images
[root@localhost images]# 
```

2. Creating Domain.

   Instances (VMs) are defined in Libvirt via XML and referred as domain. A domain is an instance of an operating system running on a virtualized machine. A guest domain can refer to either a running virtual machine or a configuration which can be used to launch a virtual machine.

   Following is a sample XML file for defining a vNIOS virtual appliance in KVM. Note that the VM name, memory, vCPU, MAC address of macvtap interfaces and location of the qcow2 file (highlighted in red in the following example) may vary. You can change these parameters according to your deployment.

   Create **vNIOS.xml** file under /var/lib/libvirt/images directory with the following contents.

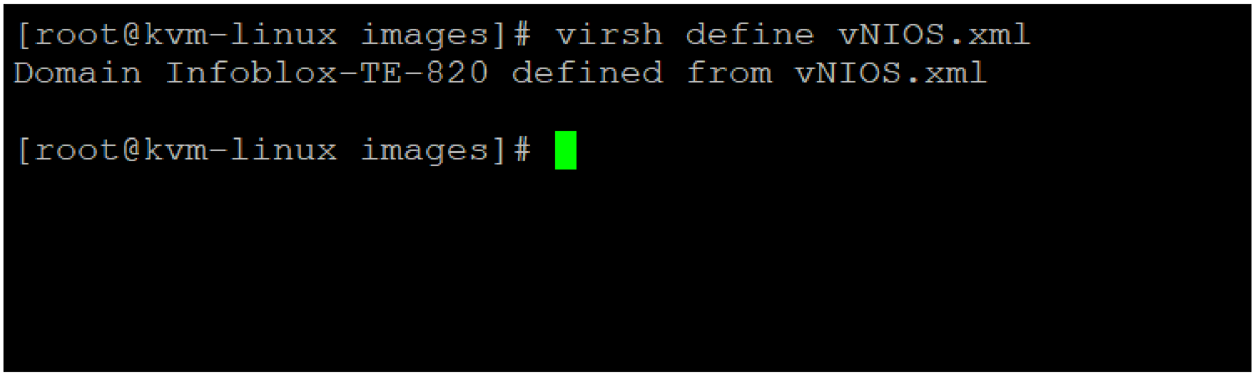
```
<domain type='kvm' id='1'>
  <name>Infoblox-TE-820</name>

  <memory unit='KiB'>21299200</memory>
  <currentMemory unit='KiB'>21299200</currentMemory>
  <vcpu placement='static'>8</vcpu>
  <resource>
    <partition>/machine</partition>
  </resource>
  <os>
    <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <cpu mode='custom' match='exact' check='full'>
    <model fallback='forbid'>IvyBridge</model>
    <feature policy='require' name='hypervisor'/>
    <feature policy='require' name='xsaveopt'/>
  </cpu>
  <clock offset='utc'>
    <timer name='rtc' tickpolicy='catchup'/>
    <timer name='pit' tickpolicy='delay'/>
    <timer name='hpet' present='no'/>
  </clock>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <pm>
    <suspend-to-mem enabled='no'/>
    <suspend-to-disk enabled='no'/>
  </pm>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2'/>
      <source file='/var/lib/libvirt/images/nios-8.1.6-360192-2017-08-25-21-23-32-ddi.qcow2'/>
      <backingStore/>
      <target dev='hda' bus='ide'/>
      <alias name='ide0-0-0'/>
      <address type='drive' controller='0' bus='0' target='0' unit='0'/>
    </disk>
    <controller type='usb' index='0' model='ich9-ehci1'>
      <alias name='usb'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x7'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci1'>
      <alias name='usb'/>
      <master startport='0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x0'
multifunction='on'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci2'>
      <alias name='usb'/>
      <master startport='2'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x1'/>
    </controller>
    <controller type='usb' index='0' model='ich9-uhci3'>
      <alias name='usb'/>
      <master startport='4'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x2'/>
    </controller>
    <controller type='pci' index='0' model='pci-root'>
      <alias name='pci.0'/>
    </controller>
```

```
<controller type='ide' index='0'>
  <alias name='ide'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
</controller>
<controller type='virtio-serial' index='0'>
  <alias name='virtio-serial0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>
</controller>
<interface type='direct'>
  <mac address='6a:cd:d3:c5:85:10'/>
  <source dev='ens192' mode='bridge'/>
  <target dev='macvtap0'/>
  <model type='virtio'/>
  <alias name='net0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
<interface type='direct'>
  <mac address='f2:ac:42:b6:ce:64'/>
  <source dev='ens192' mode='bridge'/>
  <target dev='macvtap1'/>
  <model type='virtio'/>
  <alias name='net1'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
</interface>
<interface type='direct'>
  <mac address='fa:5a:a0:31:a9:1d'/>
  <source dev='ens192' mode='bridge'/>
  <target dev='macvtap2'/>
  <model type='virtio'/>
  <alias name='net2'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
</interface>
<interface type='direct'>
  <mac address='32:6d:91:7f:21:b3'/>
  <source dev='ens192' mode='bridge'/>
  <target dev='macvtap3'/>
  <model type='virtio'/>
  <alias name='net3'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
</interface>
<serial type='pty'>
  <source path='/dev/pts/1'/>
  <target type='isa-serial' port='0'>
    <model name='isa-serial'/>
  </target>
  <alias name='serial0'/>
</serial>
<console type='pty' tty='/dev/pts/1'>
  <source path='/dev/pts/1'/>
  <target type='serial' port='0'/>
  <alias name='serial0'/>
</console>
<channel type='spicevmc'>
  <target type='virtio' name='com.redhat.spice.0' state='disconnected'/>
```

```
      <alias name='channel0'/>
      <address type='virtio-serial' controller='0' bus='0' port='1'/>
    </channel>
    <input type='mouse' bus='ps2'>
      <alias name='input0'/>
    </input>
    <input type='keyboard' bus='ps2'>
      <alias name='input1'/>
    </input>
    <graphics type='spice' port='5900' autoport='yes' listen='127.0.0.1'>
      <listen type='address' address='127.0.0.1'/>
      <image compression='off'/>
    </graphics>
    <sound model='ich6'>
      <alias name='sound0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
    </sound>
    <video>
      <model type='qxl' ram='65536' vram='65536' vgamem='16384' heads='1' primary='yes'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <redirdev bus='usb' type='spicevmc'>
      <alias name='redir0'/>
      <address type='usb' bus='0' port='1'/>
    </redirdev>
    <redirdev bus='usb' type='spicevmc'>
      <alias name='redir1'/>
      <address type='usb' bus='0' port='2'/>
    </redirdev>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0a' function='0x0'/>
    </memballoon>
  </devices>
  <seclabel type='dynamic' model='selinux' relabel='yes'>
    <label>system_u:system_r:svirt_t:s0:c100,c932</label>
    <imagelabel>system_u:object_r:svirt_image_t:s0:c100,c932</imagelabel>
  </seclabel>
  <seclabel type='dynamic' model='dac' relabel='yes'>
    <label>+107:+107</label>
    <imagelabel>+107:+107</imagelabel>
  </seclabel>
</domain>
```

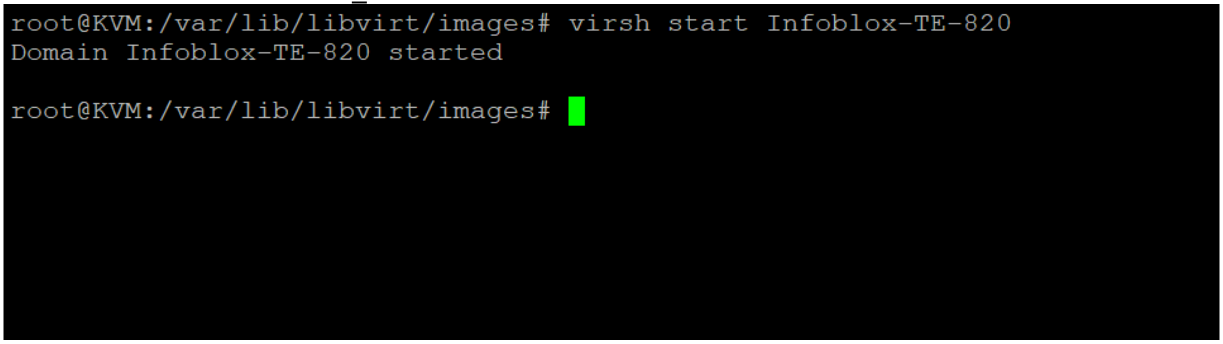
3. Defining a domain
   Change the directory to **/var/lib/libvirt/images**.

Use # `virsh define vNIOS.xml` command to define the domain.



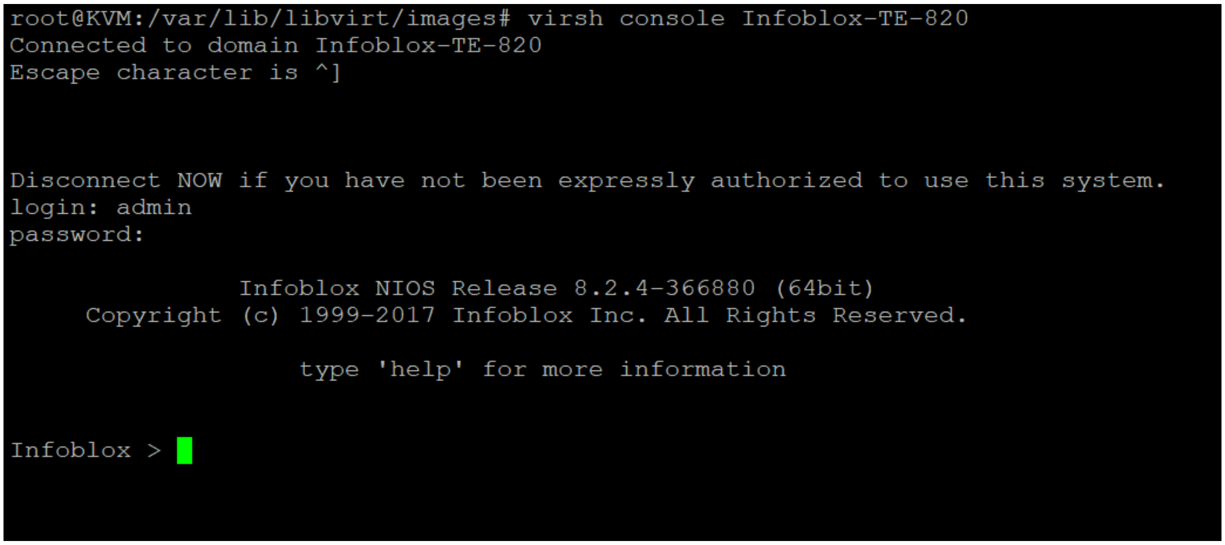
```
[root@kvm-linux images]# virsh define vNIOS.xml
Domain Infoblox-TE-820 defined from vNIOS.xml

[root@kvm-linux images]#
```

4. Starting an instance
   Use #`virsh start <domain_name>` command to start the instance



```
root@KVM:/var/lib/libvirt/images# virsh start Infoblox-TE-820
Domain Infoblox-TE-820 started

root@KVM:/var/lib/libvirt/images#
```

5. Connecting to the console
   You can connect to the console of the vNIOS by # `virsh console <domain_name>`
   Please note that the instance takes a few minute to boot.



```
root@KVM:/var/lib/libvirt/images# virsh console Infoblox-TE-820
Connected to domain Infoblox-TE-820
Escape character is ^]


Disconnect NOW if you have not been expressly authorized to use this system.
login: admin
password:

               Infoblox NIOS Release 8.2.4-366880 (64bit)
     Copyright (c) 1999-2017 Infoblox Inc. All Rights Reserved.

                   type 'help' for more information


Infoblox >
```

6. To exit out from the console use the ctlrl 5 key combination.

Using cloud-init to do initial vNIOS configuration

Cloud-init is an open-source package that is commonly used to perform configuration of cloud instances based on key-value pairs provided by the user as part of the instance launch request. vNIOS uses cloud-init to configure initial settings like defining IP address, hardware type etc.

Creating cloud-init file

1. Create a directory **cloud-config** in **/var/lib/libvirt/images** folder and create a file **ovf-env.xml** and in this directory add the following contents to it.

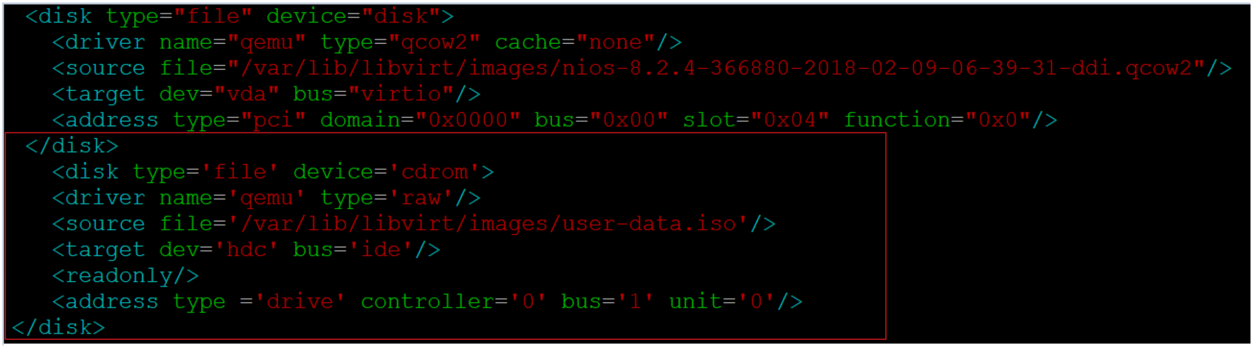
   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <Environment xmlns="http://schemas.dmtf.org/ovf/environment/1"
                xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
     <PropertySection>
       <Property oe:key="remote_console_enabled" oe:value="y"/>
       <Property oe:key="hardware_type" oe:value="IB-FLEX"/>
       <Property oe:key="temp_license" oe:value="flex_grid"/>
       <Property oe:key="lan1-v4_addr" oe:value="ip_address"/>
       <Property oe:key="lan1-v4_netmask" oe:value="subnet_mask"/>
       <Property oe:key="lan1-v4_gw" oe:value="default_gateway"/>
     </PropertySection>
   </Environment>
   ```

2. Use the genisoimage utility to generate the iso file from the cloud-config folder under /var/lib/libvirt/images directory

   ```
   #genisoimage -V OVF-TRANSPORT -o user-data.iso -R cloud-config
   ```

   where user-data.iso is the name of the iso file which will be generated.

3. Add a section about user-data.iso file under <device> tag in the in original domain vNIOS.xml file.



```
 <disk type="file" device="disk">
   <driver name="qemu" type="qcow2" cache="none"/>
   <source file="/var/lib/libvirt/images/nios-8.2.4-366880-2018-02-09-06-39-31-ddi.qcow2"/>
   <target dev="vda" bus="virtio"/>
   <address type="pci" domain="0x0000" bus="0x00" slot="0x04" function="0x0"/>
 </disk>
   <disk type='file' device='cdrom'>
   <driver name='qemu' type='raw'/>
   <source file='/var/lib/libvirt/images/user-data.iso'/>
   <target dev='hdc' bus='ide'/>
   <readonly/>
   <address type ='drive' controller='0' bus='1' unit='0'/>
</disk>
```

4. Follow the same steps as mentioned under Deploying vNIOS on KVM section to create an instance with cloud-init.
5. Newly created instance should have a predefined lan1 IP address and hardware type should be set as IB-FLEX.

Some useful Information

1. If you are planning to enable DNS Cache Acceleration(DDCA), make sure that VM has sufficient resources (atleast 65GB RAM and 8vCPUs). In the absence of sufficient resources DDCA license will not get activated.

2. After the deploying grid member, first enable the IB-FLEX license before making any other configuration changes. Enabling IB-FLEX license lets the VM to adjust the resources.

3. Make sure the Grid NTP is pointing to right NTP server. (time.apple.com or pool.ntp.org). In case if NTP is not synced DNS, queries may get timed out or drop.

4. For a grid with IB-FLEX members, it is recommended to have reporting member → v1405, v2205 or v5005 as the grid generates lots of data.

5. IB-FLEX feature is qualified and tested on Openstack → Mitaka, Newton and Ocata release.

6. Infoblox vNIOS is supported only on RHEL and Centos based KVM.

7. Do not change the name of the cloud-init file and use it as it is mentioned in the deployment guide.

8. genisoimage utility can be downloaded by `# yum install genisoimage`

9. Standalone vNIOS on KVM uses only .ovf format and not the .yaml format.