INSTALLATION RUNBOOK FOR Infoblox vNIOS

Product Name: **vNIOS IPAM driver**

Product Version: **[2.0.1]**

MOS Version: **[8.0]**

OpenStack Version: **[Liberty]**

Product Type: **[Network Service Appliance]**

Contents

Document History

Version	Revision Date	Description
0.1	31st-May-2016	Initial Version

1. Introduction

This document is to serve as a detailed Deployment Guide for Infoblox vNIOS and IPAM solution to OpenStack Neutron to provide DHCP and DNS services by Infoblox vNIOS appliances.

This document describes the reference architecture; installation steps for validated, KVM based MOS and vNIOS deployment, limitations and testing procedure.

It also describes how to install the Infoblox vNIOS virtual appliance on KVM-based Mirantis OpenStack.

1.1 Target Audience

- 1.1.1 Network Administrator
- 1.1.2 Information Technology
- 1.1.3 System Administrators

2. Product Overview

Infoblox appliances deliver core network services—including DNS, DHCP, IPAM, NTP, and TFTP—on a reliable, secure, easy-to-deploy, and manageable platform.

Infoblox delivers a fully integrated and robust DNS, DHCP, and IPAM solution that enables network administrators to centrally manage the entire solution, infrastructure and data easily.

Infoblox Openstack Adapter is created to demonstrate the ability to plug in IPAM solution to OpenStack Neutron to provide DHCP and DNS services by Infoblox NIOS appliances.

The vNIOS provides integrated, secure, and easy-to-manage DNS (Domain Name System), DHCP (Dynamic Host Configuration Protocol) and IPAM (IP address management) services.

The Infoblox OpenStack driver along with vNIOS, provides centralized and automated DNS, DHCP, and IP address management (DDI) services for OpenStack environments.
By using this solution, any network, subnet, or port IP Address created through the OpenStack Horizon UI, Neutron CLI, or Neutron APIs is provisioned directly from the Infoblox Grid Master(vNIOS) along with the corresponding DNS entries (zones/sub zones).

When VMs (virtual machines) are created in OpenStack, fixed IP addresses are allocated directly from the Infoblox Grid through the driver to the VMs and DNS entries (A and PTR records) are automatically created.

The driver also manages floating IP address allocation and corresponding DNS entry creation, providing a comprehensive automated DDI solution for OpenStack.

After you install the Infoblox OpenStack driver in an OpenStack environment, you can configure the driver to connect to a NIOS or vNIOS Grid Master or stand-alone Infoblox appliance.

Depending on the tasks you want to perform in OpenStack through the Horizon UI or Neutron CLI or APIs, the NIOS or vNIOS appliance automatically creates networks and the corresponding DNS zones, obtains the next available IPv4 or IPv6 addresses for VMs, creates DNS A and PTR records (individually or using NIOS host records) for VMs, and stores the associated meta data in the NIOS database.
In addition, the Infoblox Grid members are dynamically allocated to serve DNS and DHCP directly to OpenStack VMs with the support for both overlapping and non-overlapping OpenStack networks.

3. Joint reference architecture

Figure: OpenStack and Infoblox IPAM

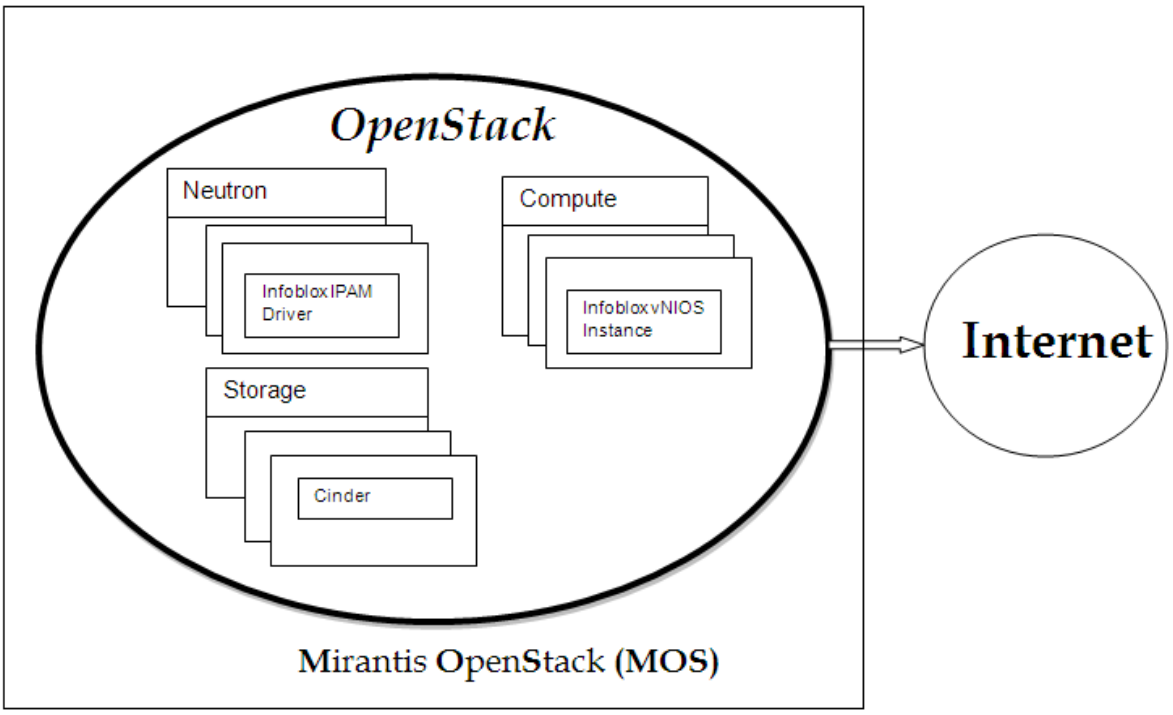
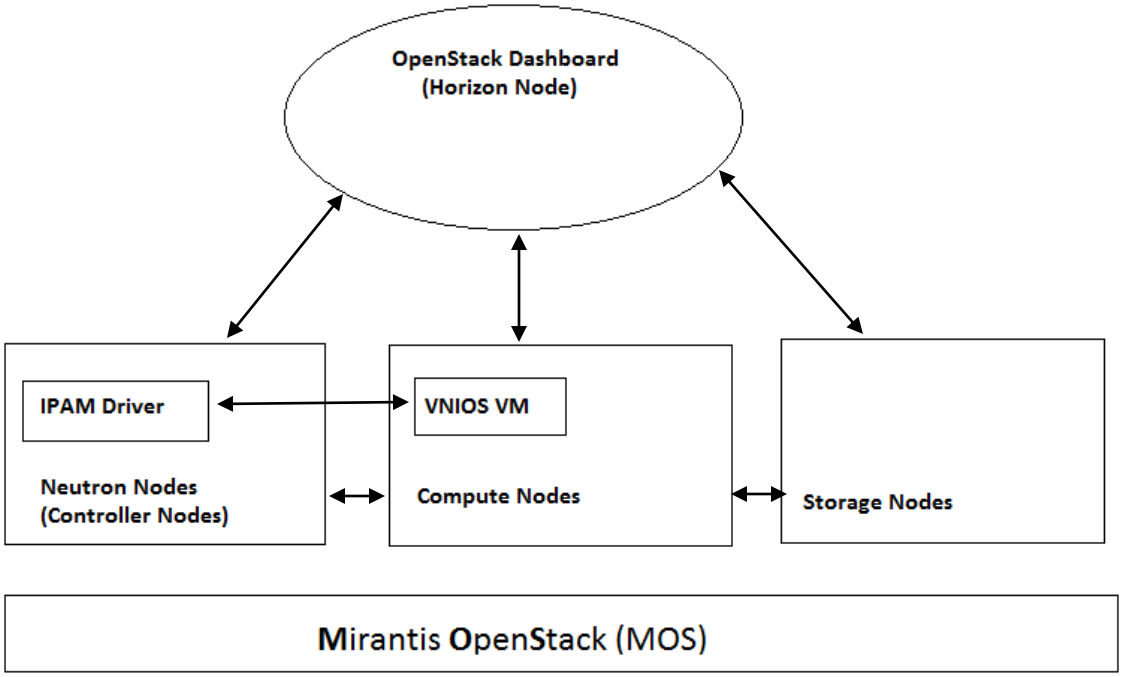


Figure: MOS and Infoblox IPAM,VNIOS



4. Physical & Logical network topology

Fig: Infoblox IPAM and OpenStack Overview:

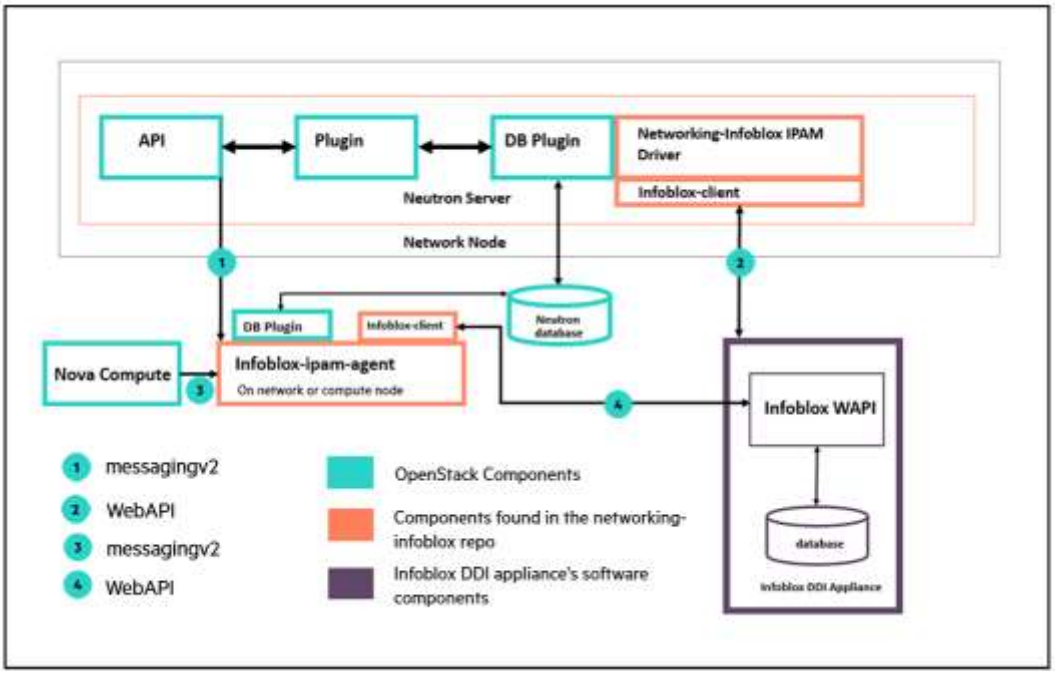
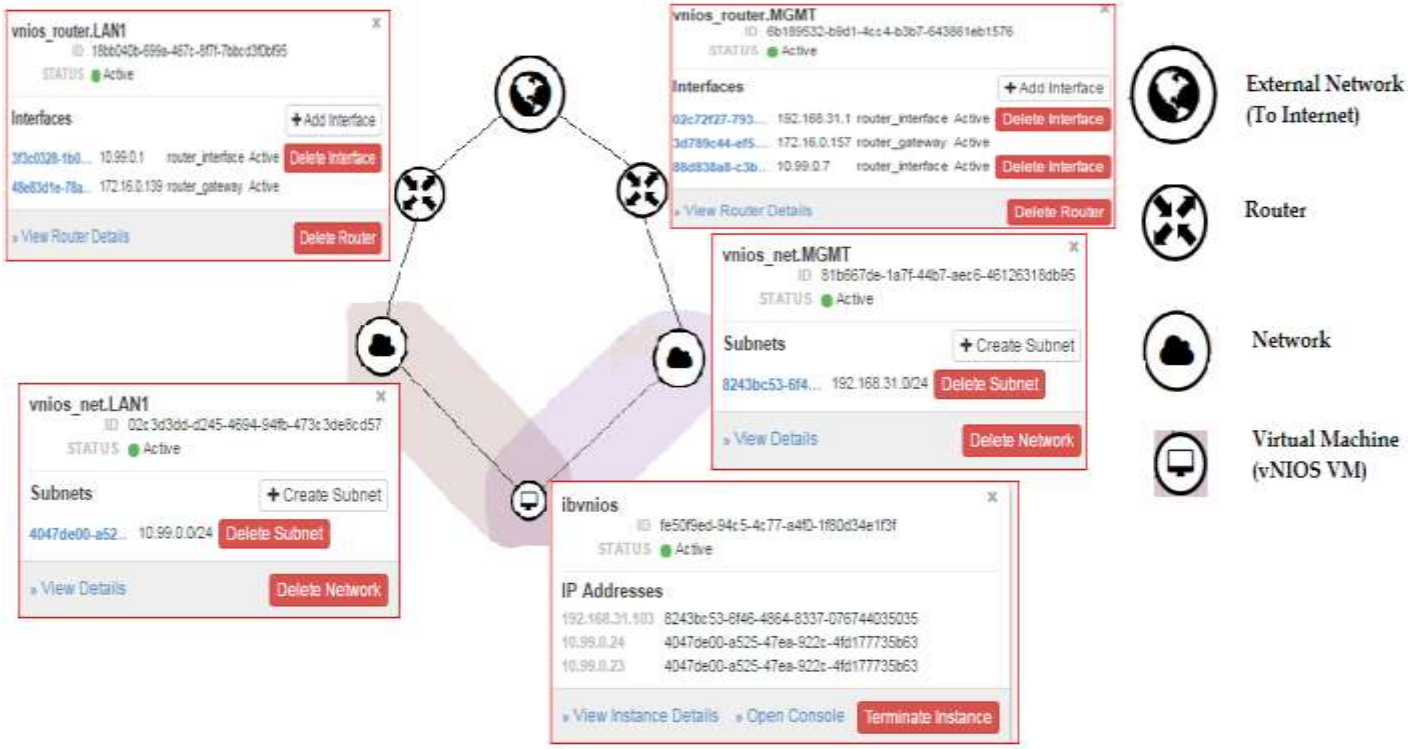


Fig: Infoblox OpenStack Network topology



Note- The IP address and ID's from above figure will change depends upon network/subnet address.

5. Installation and Configuration

5.1 Environment preparation

An MOS deployment that includes the following services
- Compute
- Network
- Storage

The Minimum number of nodes required -

- 1 host machine for the MOS Fuel master node.
  For a production environment:
  Quad-core CPU 4 GB RAM
  10 Gigabit network port
  128 GB SAS Disk
  IPMI access through an independent management network
  For a testing environment:
  Dual-core CPU 2 GB RAM
  1 Gigabit network port
  50 GB disk
  Physical console access
- MOS Compute node.
  The number and hardware configuration of the compute nodes depend on the following:
  Number of virtual machines
  Applications that you plan to run on these virtual machines
  for standalone Infoblox appliance
  6 CPU, 12 GB RAM, 250 GB storage
- MOS Controller node
  For a production environment:
  Use at least three controller nodes for high availability

  For a testing environment:
  1 host machine for MOS Controller node in a cluster
  Dual-Core CPU, 8GB RAM, 200 GB storage

Required Infoblox Packages:

- RPM-GPG-KEY- Infoblox file
- vnios_kvm-1.0.1-*.el6.x86_64.rpm package

- nios-7.3.*.160G-1420-disk1.qcow2 - This is the vNIOS software package.
- *Neutron drivers for integration with Infoblox grids for IPAM and DNS.* https://pypi.python.org/pypi/networking-infoblox

You can download the vNIOS software from theInfoblox Technical Support site. To download the software, you must have a valid login account on the Infoblox Support site. Register your product at https://support.infoblox.com if you do not already have an account.

5.2 MOS Installation

MOS Environment Details –

1. Number of controller nodes: 3
2. Number of compute nodes: 3
3. Number of Storage-Cinder nodes: 3
4. Compute Hypervisor type: KVM
5. Storage Backends: Cinder LVM over iSCSI for volumes
   Default: Use qcow format for images
6. Network : Neutron with VLAN segmentation

Installing Mirantis OpenStack Manually:

Configuring Virtual Machines

Before installing Fuel, you must configure the Fuel Master node and Fuel Slave nodes virtual machines.

The virtual machine configuration includes:

1. **Configuring the Network**
2. **Creating Virtual Machines**
3. **Mounting the Mirantis OpenStack ISO Image**

1. Configuring the Network

Configure the VirtualBox Host-Only Ethernet Adapters for the Fuel Master node and Fuel Slave nodes.

Procedure:

1. In VirtualBox, click **File ▸ Preferences ▸ Network**.

2. Select **Host-only Networks**.
3. Create three VirtualBox Host-Only Ethernet Adapters by clicking the **Adds new host-only network** icon.

VirtualBox creates three new Ethernet adapters. For the purpose of example, Ethernet adapters' names are:

- For Linux and Mac OS X:
  - vboxnet0
  - vboxnet1
  - vboxnet2
- For Windows with Cygwin:
  - VirtualBox Host-Only Ethernet Adapter
  - VirtualBox Host-Only Ethernet Adapter #2
  - VirtualBox Host-Only Ethernet Adapter #3

4. Modify the settings of the first Ethernet adapter:
   - IPv4 Address: 10.20.0.1
   - IPv4 Network mask: 255.255.255.0
   - DHCP Server: disabled
5. Modify the settings of the second Ethernet adapter:
   - IPv4 Address: 172.16.0.254
   - IPv4 Network mask: 255.255.255.0
   - DHCP Server: disabled
6. Modify the settings for the third Ethernet adapter:
   - IPv4 Address: 172.16.1.1
   - IPv4 Network mask: 255.255.255.0
   - DHCP Server: disabled
7. Proceed to Creating Virtual Machines

2. Creating Virtual Machines

You must manually configure virtual machines for the Fuel installation. Create one virtual machine for the Fuel Master node and at least three virtual machines for Fuel Slave Nodes.

Procedure:

1. In VirtualBox, configure the Fuel Master node virtual machine according to the Virtual Machine Requirements.
2. In the Fuel Master node network settings, configure the following network adapters:
    - For Windows with Cygwin:
        - **Adapter 1:** Host-only adapter "VirtualBox Host-Only Ethernet Adapter"
        - **Adapter 2:** Host-only adapter "VirtualBox Host-Only Ethernet Adapter #2"
        - **Adapter 3:** NAT
    - For Linux:
        - **Adapter 1:** Host-only adapter vboxnet0
        - **Adapter 2:** Host-only adapter vboxnet1
        - **Adapter 3:** NAT
3. Specify the following parameters to the Fuel Master node network adapters:
    - Promiscuous mode: **Allow All**
    - Adapter Type: **Intel PRO/1000 MT Desktop**
    - Select the **Cable Connected** checkbox
4. Select the Fuel Master node virtual machine and click **Settings**.
5. Select **System ‣ Processor**.
6. Select **Enable PAE/NX**.
7. Adjust the number of CPU to 2.
8. Click **OK**.
9. Configure at least three Fuel Slave nodes virtual machines according to the Virtual Machine Requirements.
10. Select a Fuel Slave node VM and click **Settings ‣ System**.
11. In **Boot Order**, select **Network**.
12. Unselect **Floppy** and **Optical**.
13. Set the following booting order:
        - Network
        - Hard drive
14. Click **OK**.
15. Click on a Fuel Slave node VM and select **Settings ‣ Network**.

16. Configure the following network adapters:
    - For Windows with Cygwin:
        - **Adapter 1:** Host-only adapter "VirtualBox Host-Only Ethernet Adapter"
        - **Adapter 2:** Host-only adapter "VirtualBox Host-Only Ethernet Adapter #2"
        - **Adapter 3:** Host-only adapter "VirtualBox Host-Only Ethernet Adapter #3
    - For Linux:
        - **Adapter 1:** Host-only adapter vboxnet0
        - **Adapter 2:** Host-only adapter vboxnet1
        - **Adapter 3:** Host-only adapter vboxnet2
17. Specify the following parameters to the Fuel Slave node network adapters:
    - Promiscuous mode: **Allow All**
    - Adapter Type: **Intel PRO/1000 MT Desktop**
    - Select the **Cable Connected** checkbox
18. Click **Settings ▸ Storage**.
19. Select **Controller SATA**
20. Click **Create Hard Disk**.
21. In the **Create New Virtual Disk** wizard, select:
    - File type: VDI
    - Storage details: Dynamically allocated
    - Size: 64 GB
22. Click **Create**.
23. Create another disk as described in **Step 18** - **Step 22**.
24. Repeat **Step 10** - **Step 23** for each Fuel Slave node.
25. Proceed to Mounting the Mirantis OpenStack ISO Image.

3. Mounting the Mirantis OpenStack ISO Image

To install Fuel, mount the Mirantis OpenStack ISO image in the virtual machine settings.

Procedure:

1. Right-click the Fuel Master node.
2. Select **Storage**.

3. Select the empty optical drive.
4. Click the optical drive icon.
5. Select **Choose Virtual Optical Disk File**.
6. Open the Fuel ISO image.
7. Proceed to Installing Fuel.

See also

- Downloading the Mirantis OpenStack Image

Installing Fuel

After you complete the steps described in Configuring Virtual Machines, install Fuel.

Procedure:

1. Power on the Fuel Master node VM to start the installation.
2. When prompted, select **1. Fuel Install (Static IP)**.

   Fuel installs on the virtual machine. It may take some time.

3. Optionally, enter the Fuel Setup screen when the following message displays:

```
Press a key to enter Fuel Setup (or press ESC to skip)...
```

4. Press F8.

   System response:

```
Loading docker images. (This may take a while)
```

   When Fuel completes the installation, the following message displays:

```
Welcome to the Fuel server
...
fuel login:
```

5. After the Fuel Master node installs, power on the Fuel Slave nodes. When the Fuel Slave nodes boot, the Fuel Master node automatically discovers them.
6. Log in to the Fuel Master Node CLI using the default credentials.

7. Configure network interfaces:

    1. Prepare the network configuration files:

    ```
    sed -i.orig \
    '/^UUID=\|^NM_CONTROLLED=/d;s/^\(.*\)=yes/\1=no/g;' \
    /etc/sysconfig/network-scripts/ifcfg-eth{0,1,2}
    sed -i.orig \
    's/^ONBOOT=.*/ONBOOT=yes/;/^ONBOOT=/iNM_CONTROLLED=no' \
    /etc/sysconfig/network-scripts/ifcfg-eth{0,1,2}
    ```

    These commands create a backup of network configuration, removes the network manager options, disables default settings, enables network interface activation at boot time, and disables the network manager.

    2. Configure eth1 to use as a static IP address with the corresponding netmask.

    Example:

    ```
    sed -i 's/^BOOTPROTO=.*/BOOTPROTO=static/' \
    /etc/sysconfig/network-scripts/ifcfg-eth1

    sed -i '/^BOOTPROTO/aIPADDR=172.16.0.1\nNETMASK=255.255.255.0' \
    /etc/sysconfig/network-scripts/ifcfg-eth1
    ```

    Therefore, eth1 will have a static IP address *172.16.0.1* with the netmask *255.255.255.0*.

    3. Configure eth2 to obtain an IP address from the VirtualBox DHCP server and use a default route:

    ```
    sed -i 's/^BOOTPROTO=.*/BOOTPROTO=dhcp/;s/^DEFROUTE=.*/DEFROUTE=yes/' \
    /etc/sysconfig/network-scripts/ifcfg-eth2

    sed -i '/^BOOTPROTO/aPERSISTENT_DHCLIENT=yes' \
    /etc/sysconfig/network-scripts/ifcfg-eth2
    ```

    4. Create a backup of network configuration and disable zero-configuration networking:

    ```
    sed -i.orig '/NOZEROCONF/d;aNOZEROCONF=yes' /etc/sysconfig/network
    ```

    Therefore, eth2 will use DHCP only.

    5. Remove the default route and system-wide settings from eth0:

```
sed -i '/^GATEWAY=/d' /etc/sysconfig/network \
/etc/sysconfig/network-scripts/ifcfg-eth0
```

6. Add the `aType Loopback` parameter to the `ifcfg-lo` configuration file:

```
sed -i.orig '/^DEVICE=lo/aTYPE=Loopback' \
/etc/sysconfig/network-scripts/ifcfg-lo
```

7. Enable NAT (MASQUERADE) and IP forwarding for the Public network:

   Example:

```
iptables -I FORWARD 1 --dst 172.16.0.0/24 -j ACCEPT
iptables -I FORWARD 1 --src 172.16.0.0/24 -j ACCEPT
iptables -t nat -A POSTROUTING -s 172.16.0.0/24 \! -d 172.16.0.0/24 \
-j MASQUERADE
service iptables save
```

8. Disable NetworkManager and apply the new network settings:

```
nmcli networking off &>/dev/null ; service network restart
```

9. Verify the Internet connection on the Fuel Master node:

```
ping -c 3 google.com
```

   Example of system response:

```
PING google.com (216.58.214.206) 56(84) bytes of data.
64 bytes from bud02s23-in-f14.1e100.net (216.58.214.206): icmp_seq=1
ttl=54 time=31.0 ms
64 bytes from bud02s23-in-f14.1e100.net (216.58.214.206): icmp_seq=2
ttl=54 time=30.1 ms
64 bytes from bud02s23-in-f14.1e100.net (216.58.214.206): icmp_seq=3
ttl=54 time=30.0 ms
```

10. Create a bootstrap image for Fuel Slave nodes:

```
fuel-bootstrap -v --debug build --activate
```

11. Verify the bootstrap images:

```
fuel-bootstrap list
```

Example of system response:

```
+--------------------------------------+---------------+--------+
| uuid                                 | label         | status |
+--------------------------------------+---------------+--------+
| dd2f45bf-08c2-4c39-bd2d-6d00f26d6540 | dd2f45bf-08c2 | active |
| centos                               | deprecated    |        |
+--------------------------------------+---------------+--------+
```

Log in to the Fuel UI by pointing your browser to the URL specified in the command prompt. Use the default login and password.

Proceed to *Create an OpenStack environment* in *Fuel User Guide*.

https://docs.mirantis.com/openstack/fuel/fuel-8.0/pdf/Fuel-8.0-UserGuide.pdf

5.2.1 Health Check Results

Environments Settings:

Name: Infoblox
Status: Operational
OpenStack Release: Liberty on Ubuntu 14.04
Compute: KVM
Network: Neutron with VLAN segmentation
Storage Backend's: Cinder LVM over iSCSI for volumes

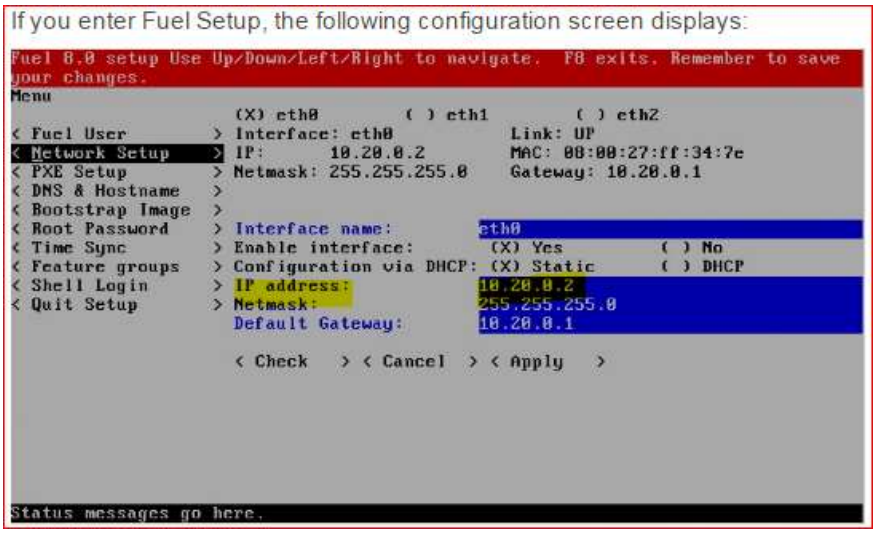
5.3 Infoblox vNIOS Installation Steps

Installation steps divided into two parts –

1. vNIOS package installation
2. IPAM driver for neutron installation

5.3.1 Installing vNIOS for KVM in the OpenStack Environment

1. Connect (SSH) to OpenStack controller node
   To ssh to controller node –
   1.1 SSH to fuel master node :
   Use Fuel master IP address mentioned during the Fuel setup i.e specified in the command prompt.



   1.2 ssh root@10.20.0.2 (default username/password: admin/admin)
   1.3 run $ fuel node list
   This will list all openstack nodes(controller/compute etc)
   1.4 Use controller IP from above(#1.3) list to ssh to controller
   1.5 For more details about fuel setup refer:
   https://docs.mirantis.com/openstack/fuel/fuel-8.0/quickstart-

guide.html#installing-mirantis-openstack-manually

2. Install the device-mapper packages
   sudo apt-get install libdevmapper-dev
   sudo apt-get install libguestfs-tools

3. Download the keystonerc_admin file from Openstack horizon portal.
   Login to portal -> Select Project -> Compute -> Access & Security -> API Access tab -> Download OpenStack RC File.
   OR
   Refer to the section on Getting Credentials for a CLI in the OpenStack CLI Guide.
   Refer: http://docs.openstack.org/cli-reference/common/cli_set_environment_variables_using_openstack_rc.html
   $ source keystonerc_admin

4. Download the *.qcow2 file on OpenStack controller node
5. Upload the *.qcow2 file for the specified vNIOS for KVM model to OpenStack
   $ glance image-create --name vnios-1420 --visibility public --container-format bare --disk-format qcow2 --file /tmp/nios-7.3.4.160G-1420-disk1.qcow2

6. Setting up the OpenStack flavors
   After you upload the qcow2 file, set up the OpenStack flavors for your vNIOS models. Each flavor corresponds to different vCPU, RAM, disk size, and functionality
   vNIOS OpenStack Flavors

Name	Memory(MB)	Disk(GB)	Swap	vCPU
vnios800.300G	8192	300	0	2
vnios820.55G	4096	55	0	2
vnios1420.160G	8192	160	0	4
vnios2220.160G.CP	12288	160	0	4
vnios1400.160G.CP	8192	160	0	4

   To setup flavor for particular vNIOS appliance use -
   $ nova flavor-create --is-public true <name><ID><Memory><disk><cpu> --swap 0 --ephemeral 0

   Where,

- name defines the name for the vNIOS for KVM instance.
- ID defines the unique OpenStack flavor ID for the KVM instance.
- Memory disk and cpu specify the flavors of the vNIOS for KVM instance

Following is a sample command for vnios 1420:

$ nova flavor-create --is-public true vnios-1420.160 6 8192 160 4 --swap 0 --ephemeral 0

7. Setting Up Security Groups

Basic Configuration:

Creating security group “vnios-sec-group”:

#vNIOS security group

$ neutron security-group-create vnios-sec-group

You can add certain protocol

l rules to existing or default security groups to allow specific network traffic

HTTPS communications:

Example-

$ neutron security-group-rule-create --protocol tcp --port-range-min 443 --port-range-max 443 --ethertype IPv4 vnios-sec-group

$ neutron security-group-rule-create --protocol tcp --port-range-min 443 --port-range-max 443 --ethertype IPv6 vnios-sec-group

Deleting security group “vnios-sec-group”:

$ neutron security-group-delete vnios-sec-grp

8. Setting Up vNIOS Networks

For the vNIOS appliance on OpenStack, you must specify at least two networks, MGMT and LAN1.

Infoblox also recommends to set up the HA and LAN2 networks, as once the instance is launched, you cannot attach networks to it.

The Infoblox HA -

You can configure two appliances as an HA (high availability) pair to provide redundancy for core network services and Infoblox External DNS

Security. For information about Infoblox External DNS Security, see Infoblox External DNS Security. An HA pair can be a Grid Master, a Grid Master candidate, a Grid member, or an independent appliance. The two nodes that form an HA pair—identified as Node 1 and Node 2—are in an active/passive configuration. The active node receives, processes, and responds to all service requests. The passive node constantly keeps its database synchronized with that of the active node, so it can take over services if a failover occurs. A failover is the reversal of the active/passive roles of each node; that is, when a failover occurs, the previously active node becomes passive and the previously passive node becomes active. You can configure an HA pair in either IPv4, IPv6, or in dual mode. An IPv4 HA pair uses IPv4 as the communication protocol between the two nodes and an IPv6 HA pair uses IPv6 as the communication protocol between the two nodes. But in a dual mode HA pair, you can select either IPv4 or IPv6 as the communication protocol between the two nodes. Note that when you add a dual mode HA member to a Grid, the communication protocol between the two nodes of an HA pair must the same as the Grid communication protocol.

The network is specified in CIDR notation (e.g. 10.0.0.0/24) with its gateway IP address.

You can use the example scripts to set up networks.
For example:
setup LAN1/HA network
Notice: 201-254 are omitted from the allocation-pool. They are reserved for the public VIP

```
# Create LAN1 network
$ neutron net-create --admin-state-down --shared vnios_net.LAN1
$ neutron net-update vnios_net.LAN1 --admin-state-up True

# Create Subnet
$ neutron subnet-create vnios_net.LAN1 10.99.0.0/24 --name
vnios_subnet.LAN1 --gateway 10.99.0.1 --enable-dhcp --allocation-
pool start=10.99.0.2,end=10.99.0.100

# Create Router
$ neutron router-create vnios_router.LAN1
```

# Create router interface
$ neutron router-interface-add vnios_router.LAN1 vnios_subnet.LAN1

# Set gateway
$ neutron router-gateway-set vnios_router.LAN1 ext_net

setup MGMT/LAN2

Create MGMT network
$neutron net-create --admin-state-down --shared vnios_net.MGMT
$ neutron net-update vnios_net.MGMT--admin-state-up True

# Create SubNet
$ neutron subnet-create vnios_net.MGMT192.168.31.0/24 --name vnios_subnet.MGMT --gateway 192.168.31.1 --enable-dhcp --allocation-pool start=192.168.31.100,end=192.168.31.200

# Create Router
$ neutron router-create vnios_router.MGMT

# Create router interface
$ neutron router-interface-add vnios_router.MGMT vnios_subnet.MGMT

# Set gateway
$ neutron router-gateway-set vnios_router.MGMText_net

9. Setting up neutron port for vnios instance

neutron port-create --name 'ib.vnios' --security-group 'vnios-sec-group' 'vnios_net.LAN1'

Note- This command output will display port details in table format. We need port ID and network ID later, while creating the instance.

neutron port-create --name 'ib.vnios' --security-group 'vnios-sec-group' 'vnios_net.MGMT'

Note- This command output will display port details in table format. We need port ID and network ID later, while creating the instance

10. Create instance

    $ nova boot --config-drive False --image <nios-7.3.0-314352-2016-01-29-05-02-02-160G-1420-disk1.qcow2> --flavor <vnios1410.160> --security-groups <name of the security group> --nic net-id=<the network ID for the MGMT interface> --nic net-id=<the network ID for the LAN1/HA interface only if you are configuring an HA pair> --nic port-id=<the IP address ID for the LAN1 interface><my-vm-grid-master>

    For the vNIOS appliance to run in OpenStack, you must specify at least two networks, MGMT and LAN1.

    To remove networks, use the neutron net-delete command.
    If some of the networks remain, use OpenStack Horizon to manually remove them.

11. Setting up created vNIOS instance
    Go to OpenStack Horizon and select the previously launched instance console.
    When the Infoblox login prompt appears, log in with the default user name and password.
    login: admin
    password: infoblox

    The Infoblox prompt appears: Infoblox >

    You must have valid licenses before you can configure the vNIOS appliance. To obtain permanent licenses, first use the Infoblox > show version command to obtain the serial number of the vNIOS appliance, and then visit the Infoblox Support web site at https://support.infoblox.com. Log in with the user ID and password you receive when you register your product online at http://www.infoblox.com/support/customer/evaluation-and-registration.
    If the vNIOS virtual appliance does not have the Infoblox licenses required to run NIOS services and to join a Grid, you can use the set temp_license command to generate and install a temporary 60-day license.

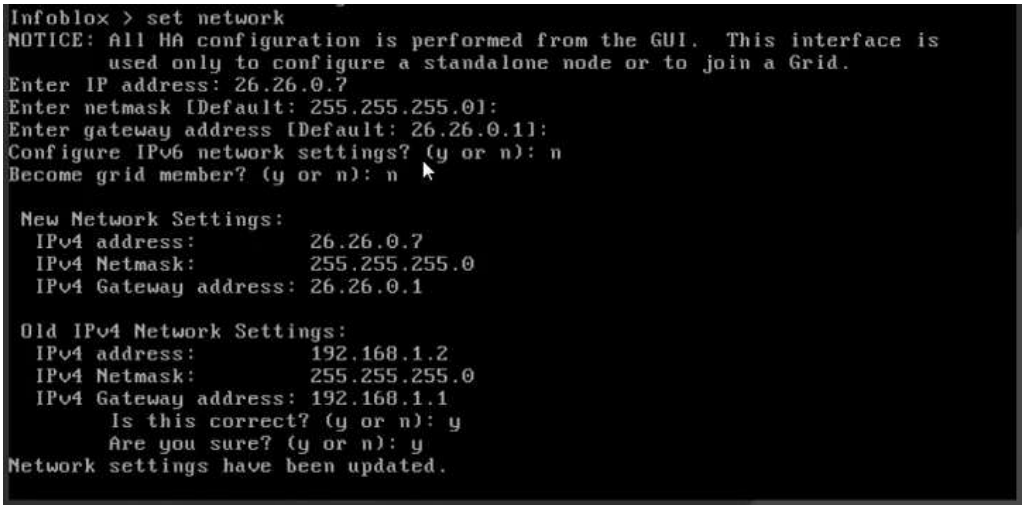
From the list of licenses, select the Grid, vNIOS, and other relevant licenses for your vNIOS virtual appliance.

$**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 IF-MAP Federation license
10. Add Multi-Grid Management license
11. Add Query Redirection license
12. Add Load Balancer license

Note: You must have both the Grid and vNIOS licenses for the vNIOS virtual appliance to join a Grid (2 and 8 from the list).

12. In OpenStack Horizon, go to Instance Overview and copy the floating IP address of the instance.
    Go back to the console and run the **$ set network** command



13. Go to the Infoblox Grid Manager and enable the NAT mode for the Grid member:
    a. Click Grid -> Grid Manager -> Members -> Network.
    b. Select the Grid member and click Edit.

c. Click Network -> Advanced.
d. Click Enable NAT Compatibility and enter the floating IP address.
e. Click Save & Close.

Note: For an HA Grid Master, ensure that you specify these settings for both nodes. After you confirm your network settings, the Infoblox Grid Manager automatically restarts. You can then proceed to setting up a Grid, as described in Setting Up a Grid.

5.3.2 IPAM driver installation and configuration

The Infoblox driver should be installed on the controller nodes that are running your neutron-server. The installation consists of the following basic steps:

1. Configure Infoblox
   Login to Infoblox Gird Manager and update the user with group and role details.

   First, you should create an Infoblox user for the integration. If you have a Cloud Network Automation license and/or are using Cloud Platform Appliances, this user should be assigned to the Cloud API Only admin group. Otherwise, you may want to create a group specifically for this integration. The group must be given the following permissions for full IPAM/DHCP/DNS functionality to work:

   To add group
   From the Administration tab, select the **Administrators tab** ->**Groups tab**, and then click the **Add** icon

Permission Type	Resource	Resource Type	Permission
[DNS]	All A Records	A record	RW
[DNS]	All AAAA Records	AAAA record	RW
[DNS, DHCP, IPAM]	All Hosts	Host	RW
[DHCP, DNS, IPAM]	All IPv4 Host	IPv4 Host address	RW
[DHCP, DNS, IPAM]	All IPv6 Host	IPv6 Host address	RW
[DHCP, IPAM]	All IPv6 Networks	IPv6 Network	RW

[GRID]	All Members	Member	RW
[DHCP, IPAM]	All IPv4 Networks	IPv4 Network	RW
[DHCP, IPAM]	All Network Views	Network view	RW
[DNS]	All PTR Records	PTR record	RW
[DHCP]	All IPv4 Ranges	IPv4 range	RW
[CLOUD]	All Tenants	Tenant	RW
[DNS]	All DNS Views	DNS View	RW
[DNS]	All Zones	Zone	RW

If you are testing IPAM only case which does not require Infoblox to serve DHCP and DNS, here is the minimum set of required permissions.

Permission Type	**Resource**	**Resource Type**	**Permission**	**Comment**
[GRID]	All Members	Member	RW	This can be set RO if Report Grid Sync Time is set to False.
[CLOUD]	All Tenants	Tenant	RW	
[DHCP, IPAM]	All Network	Network View	RW	
[DHCP, IPAM]	All IPv4 Networks	IPv4 Network	RW	
[DHCP, IPAM]	All IPv6 Networks	IPv6 Network	RW	

2. Install the driver module on the controller nodes
   sudo apt-get install python-pip
   sudo pip install infoblox-client
   sudo pip install networking-infoblox

3. Create Extensible Attribute Definitions and Network View Associations
   Download driver from :
   https://pypi.python.org/pypi/networking-infoblox

tar –xvzfnetworking-infoblox-*.tar.gz

cd ~/networking-infoblox-2.0.1/networking_infoblox/tools
python create_ea_defs.py

run create_ea_defs.py to create EA and provide username and password of Gird Manager/Master

python create_ea_defs.py
#Output:
Creating EA definitions...

In order to create Extensible Attribute definitions,
super user privilege is required.

If the preconfigured credentials already has superuser
Privilege, just hit <ENTER> when prompted for user name.

Otherwise, please enter user name and password of a user that has superuser privilege.

Enter user name: <Enter user name>
Enter password: <Enter Password>

4. Run database migrations to create the Infoblox tables
   Creating the Infoblox Neutron Database
   The driver uses a number of different Infoblox-specific tables to manage the integration. These are created by running the neutron-db-manage after you install the networking_infoblox module:
   $ sudo neutron-db-manage upgrade head
   This should be done on one of the controller nodes, assuming all controller nodes share a common database cluster

5. Modify neutron.conf and nova.conf
   # edit /etc/neutron/neutron.conf from controller node with

```
ipam_driver = infoblox
[infoblox]
cloud_data_center_id = 1

[infoblox-dc:1]
grid_master_host = GRID_MASTER_HOST
grid_master_name = GRID_MASTER_NAME
admin_user_name = USER
admin_password = PASSWORD
wapi_version = 2.2.2
wapi_max_results = -50000
```

cloud_data_center_id: An integer ID used for the data center. This is used to form the stanza name for the rest of the options. If you have multiple instances of OpenStack sharing the same Infoblox grid, this ID needs to be unique across the instances. We recommend the ID starting from 1 and increment by 1 as you add another Openstack instance. This ID is used to generate a unique ID for a network view that is cached in neutron database. Starting it with a very high number may exceed the max length of a network view id.

grid_master_host The IP address, hostname, or FQDN of the Grid Master (GM). Proxying is supported so this does not have to be the exact IP or hostname of the GM if you have a situation where you cannot reach the GM directly in your network. It can be any connection information that proxies to the GM.

grid_master_name The name of the Grid Master (GM) This has to be the exact GM name registered in the Infoblox grid.

admin_user_name The user name to use for the WAPI.

admin_password The password to use for the WAPI.

wapi_version The WAPI version to use. Version 2.2.2 or later is recommended, if your grid supports it (WAPI version 2.3 is supported in NIOS 7.3)

wapi_max_results The maximum number of objects to be returned by

WAPI. If this is set to a negative number, WAPI will return an error when the number of returned objects would exceed the setting. If this is set to a positive number, the results will be truncated when necessary. The default is -1000. If you experience “Result set too large” error, increase this value.

ssl_verify Set to false if you use a self-signed SSL certificate, and true if you use a certificate signed by a known certificate authority. You can also set this to a path to a certificate file so that verification will be done even for a self-signed certificate. Using a value of False in a production environment is not secure.
http_pool_connections, http_pool_maxsize, http_request_timeout
Optional parameters to control the HTTP session pool.

Additionally, the ipam_driver option must be set in neutron.conf to infoblox.
Note- These settings must be done on each controller that runs the Neutron service.

6. Modify nova.conf
   On each controller node running the Nova service, as well as compute node running nova-compute, you must configure Nova to send notifications. These notifications are used by the Infoblox IPAM agent to manage DNS entries and extensible attribute values for VMs. Set the following values in nova.conf, if they are not already set.

   # Edit /etc/nova/nova.conf from compute/controller node

   notification_driver = messagingv2
   notification_topics = notifications
   notify_on_state_change = vm_state
7. Start the Infoblox IPAM Agent :

   /usr/local/bin/infoblox-ipam-agent --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugins/ml2/ml2_conf.ini >/var/log/neutron/infoblox-ipam-agent.log 2>&1

8. Restart the Services
   # neutron service from controller node
   sudo service neutron-server restart

   If you modified the Nova notification settings, you must restart the Nova Compute service on each node running it. The exact command may vary based on your distribution. In Ubuntu the command is:

#nova service
$ sudo service nova-compute restart

9. Running Data Migration

   Before installing networking-infoblox, you may have already created networks, subnets and ports in OpenStack. If you wish to migrate those objects to the Infoblox grid, you can run sync_neutron_to_infoblox.py script under networking_infobloxtools folder.

   In order to run the script, you will need to create a keystone_admin file if you don't have one already and source it so that you have the admin credential variables available in the shell environment.

   networking-infoblox should have been successfully configured before running the migration script.

   $ cat keystone_admin
   unset OS_SERVICE_TOKEN
   export OS_USERNAME=admin
   export OS_PASSWORD=admin
   export OS_AUTH_URL=http://10.39.12.161:5000/v2.0
   export PS1='[\u@\h \W(keystone_admin)]\$ '

   export OS_TENANT_NAME=admin
   export OS_REGION_NAME=RegionOne

   $ source keystone_admin

   # If you have not run infoblox-ipam-agent yet, then you need to run
   # infoblox_grid_sync.py to register the Infoblox grid members to Neutron.
   $ networking-infoblox(keystone_admin)]# python
   networking_infoblox/tools/infoblox_grid_sync.py

   $ networking-infoblox(keystone_admin)]# python
   networking_infoblox/tools/sync_neutron_to_infoblox.py
   You can re-run the migration script as many times as needed.

For more details please refer the below link :

http://docs.openstack.org/developer/networking-infoblox/installation.html

5.4 Limitations

1. The current IPAM driver does not support IPv6.
2. You cannot add public/shared network from OpenStack if it already exists on NIOS.
3. We have discovered an issue with A DNS record during the floating association. After a floating IP is associated, infoblox-ipam-agent updates the record name from ‘floating-ip-‘prefixed name to ‘host-ip-‘prefixed name to indicate that the floating IP is now associated with the instance. After the name change happens, sometimes we see that all the EAs are cleared. This happens when WAPI version 2.3 is used against NIOS 7.3.
   The following grid configurations are needed to reproduce the issue:
   IP Allocation Strategy: Fixed Address
   DNS Record Binding Types: record: a, record:aaaaIt requires the vNIOS 1420 appliance and Infoblox IPAM driver.

5.5 Testing

5.5.1 Test cases

The Infoblox IPAM driver uses the **tox** testing framework. Tox is a generic virtualenv management and test command line tool.
Refer: http://tox.readthedocs.io/en/latest/

The module also uses the **oslotest** – OpenStack Testing Framework and Utilities.
Refer: http://docs.openstack.org/developer/oslotest/

Table: Tests details

Test Type	Tests Lists
DnsControllerTestCase	test_create_dns_zones_with_ns_group
	test_delete_dns_zones_for_external_network
	test_delete_dns_zones_for_private_network_with_subnet_pattern
	test_create_dns_zones_without_ns_group
	test_bind_names
	test_delete_dns_zones_for_private_network_with_network_pattern
	test_delete_dns_zones_for_shared_network_with_admin_network_deletable
	test_delete_dns_zones_for_private_network_with_static_zone
	test_delete_dns_zones_for_private_network_with_address_scope_pattern
	test_unbind_names
GridTestCase	test_grid_sync_frequency_check
	test_grid_sync_report_sync_time_multi_nodes
	test_grid_sync_report_sync_time

	test_grid_configuration_with_grid_member
	test_grid_configuration_without_grid_member
Ipam Sync and Async ControllerTestCase	test_delete_subnet_for_external_network_not_deletable
	test_delete_subnet_for_external_network_deletable
	test_allocate_ip_from_pool
	test_update_network_sync_without_subnet
	test_create_subnet_existing_private_network
	test_create_subnet_existing_external_network
	test_update_network_sync_with_network_view_mapping
	test_create_subnet_existing_network_view
	test_allocate_specific_ip
	test_create_subnet_new_network_view
GridMemberTestCase	test_sync_grid
	test_sync_member_with_cloud_support_with_member_licenses
	test_sync_member_without_cloud_support
	test_sync_member_with_cloud_support_without_member_licenses
NotificationTestCase	test_notification_service
	test_notification_endpoint_with_notification_handler
TestIpamEventHandler	test_create_network_sync_tenant_mismatch
	test_update_floatingip_sync
	test_get_instance_name_from_fip
	test_create_network_sync_same_tenant
	test_create_network_alert_should_call_resync
	test_update_network_sync
	test_create_subnet_sync_should_call_resync
	test_create_subnet_alert_should_call_resync
TestUtils	test_db_records_to_obj
	test_get_composite_values_from_records
	test_find_in_list
	test_db_records_to_json
	test_find_in_list_by_condition
	test_generate_duid
	test_exists_in_list
	test_get_dhcp_member_ips_from_ib_network
	test_get_values_from_records
	test_get_dhcp_member_ips_from_network_json
InfobloxContextTestCase	test_network_view_mapping_conditions_with_single_scope
	test_reserve_authority_member_with_dhcp_support
	test_get_dns_members_without_dhcp_support
	test_network_view_mapping_conditions_with_subnet_cidr_condition
	test_network_view_mapping_conditions_with_tenant_scope
	test_network_view_mapping_conditions_with_tenant_id_condition

	test_reserve_authority_member_without_dhcp_support
	test_get_dns_members_with_dhcp_support
	test_reserve_service_members_with_ib_network_with_dhcp_member
	test_reserve_service_members_with_ib_network_without_dhcp_member
EaManagerTestCase	test_get_common_ea
	test_get_ea_for_ip_with_router_gateway_ip
	test_get_common_ea_cloud_api_owned_false
	test_get_default_ea_for_ip
	test_reset_ea_for_network
	test_get_ea_for_ip_with_floatingip_creation
	test_get_ea_for_zone
	test_get_ea_for_ip_with_floatingip_dissociation
	test_reset_ea_for_range
	test_get_ea_for_network
HostRecordAllocatorTestCase	test_creates_host_record_on_allocate_ip_no_dhcp
	test_creates_host_record_range_on_range_allocation_use_dhcp
	test_creates_host_record_on_allocate_ip_use_dhcp
	test_deletes_host_record
FixedAddressAllocatorTestCase	test_creates_fixed_address_range_on_range_allocation
	test_deletes_fixed_address
	test_creates_fixed_address_on_allocate_ip
GridMappingTestCase	test_sync_for_cloud
	test_sync_for_without_cloud
TestIpamEventHandler	test_create_network_sync_tenant_mismatch
	test_update_floatingip_sync
	test_get_instance_name_from_fip
	test_create_network_sync_same_tenan
	test_create_subnet_sync_should_call_resync
	test_update_network_sync
	test_create_network_alert_should_call_resync
	test_create_subnet_alert_should_call_resync
	test_delete_subnet_sync
TestPatternBuilder	test_get_hostname_for_other_device_owners
	test_get_hostname_for_instance_name
	test_get_hostname_for_floating_ip_device_owner
	test_get_zone_name
InfobloxDbTestCase	test_grid_operations
	test_get_next_authority_member_for_ipam

	test_get_next_dhcp_member
	test_get_next_authority_member_for_dhcp_with_one_cpm
	test_add_or_update_tenant
	test_add_and_get_tenant
	test_grid_management
	test_get_next_authority_member_for_dhcp_with_no_cpm
	test_network_view_management
	test_get_next_authority_member_for_dhcp_with_two_cpms
TestDriver	test_update_subnet_zone_change
	test_allocate_specific_ip
	test_get_subnet
	test_allocate_subnet
	test_remove_subnet
	test_allocate_ip_from_pool
	test_deallocate_ip
	test_update_subnet_no_zone_change
	test_floating_address_request
	test_dhcp_port_address_request
	test_fixed_address_request
	test_any_address_request
	test_router_gateway_address_request
	test_auto_address_request
TestWrapper	test_rollback_wrapper_on_delete_failure
	test_rollback_wrapper

5.5.2 Test results

Run tests:

1. Go to the driver path:
   example:
   $ cd /root/networking-infoblox-2.0.1

2. To run all tests
   $ tox –e p27

Table: Tests results

SR.NO	Test case name	Runtime (s)
1.	test_create_dns_zones_with_ns_group	0.930
2.	test_delete_dns_zones_for_external_network	0.911
3.	test_delete_dns_zones_for_private_network_with_subnet_pattern	0.827
4.	test_create_dns_zones_without_ns_group	0.808
5.	test_bind_names	0.775

6.	test_delete_dns_zones_for_private_network_with_network_pattern	0.741
7.	test_delete_dns_zones_for_shared_network_with_admin_network_deletable	0.707
8.	test_delete_dns_zones_for_private_network_with_static_zone	0.653
9.	test_delete_dns_zones_for_private_network_with_address_scope_pattern	0.169
10.	test_unbind_names	0.157
11.	test_grid_sync_frequency_check	1.220
12.	test_grid_sync_report_sync_time_multi_nodes	1.036
13.	test_grid_sync_report_sync_time	1.035
14.	test_grid_configuration_with_grid_member	0.952
15.	test_grid_configuration_without_grid_member	0.873
16.	test_delete_subnet_for_external_network_not_deletable	1.152
17.	test_delete_subnet_for_external_network_deletable	1.124
18.	test_allocate_ip_from_pool	0.898
19.	test_update_network_sync_without_subnet	0.856
20.	test_create_subnet_existing_private_network	0.838
21.	test_create_subnet_existing_external_network	0.795
22.	test_update_network_sync_with_network_view_mapping	0.712
23.	test_create_subnet_existing_network_view	0.705
24.	test_allocate_specific_ip	0.139
25.	test_create_subnet_new_network_view	0. 131
26.	test_sync_grid	0.882
27.	test_sync_member_with_cloud_support_with_member_licenses	0.831
28.	test_sync_member_without_cloud_support	0.795
29.	test_sync_member_with_cloud_support_without_member_licenses	0.795
30.	test_notification_service	0.934
31.	test_notification_endpoint_with_notification_handler	0.770
32.	test_create_network_sync_tenant_mismatch	0.040
33.	test_update_floatingip_sync	0.019
34.	test_get_instance_name_from_fip	0.011
35.	test_create_network_sync_same_tenant	0.004
36.	test_create_network_alert_should_call_resync	0.004
37.	test_update_network_sync	0.003
38.	test_create_subnet_sync_should_call_resync	0.002
39.	test_create_subnet_alert_should_call_resync	0.002
40.	test_db_records_to_obj	1.187
41.	test_get_composite_values_from_records	1.112
42.	test_find_in_list	1.082
43.	test_db_records_to_json	1.075
44.	test_find_in_list_by_condition	1.033
45.	test_generate_duid	0.856
46.	test_exists_in_list	0.822

47.	test_get_dhcp_member_ips_from_ib_network	0.793
48.	test_get_values_from_records	0.357
49.	test_get_dhcp_member_ips_from_network_json	0.184
50.	test_network_view_mapping_conditions_with_single_scope	1.848
51.	test_reserve_authority_member_with_dhcp_support	1.844
52.	test_get_dns_members_without_dhcp_support	1.832
53.	test_network_view_mapping_conditions_with_subnet_cidr_condition	1.800
54.	test_network_view_mapping_conditions_with_tenant_scope	1.785
55.	test_network_view_mapping_conditions_with_tenant_id_condition	1.722
56.	test_reserve_authority_member_without_dhcp_support	1.574
57.	test_get_dns_members_with_dhcp_support	1.550
58.	test_reserve_service_members_with_ib_network_with_dhcp_member	0.655
59.	test_reserve_service_members_with_ib_network_without_dhcp_member	0.637
60.	test_get_common_ea	0.011
61.	test_get_ea_for_ip_with_router_gateway_ip	0.003
62.	test_get_common_ea_cloud_api_owned_false	0.003
63.	test_get_default_ea_for_ip	0.003
64.	test_reset_ea_for_network	0.003
65.	test_get_ea_for_ip_with_floatingip_creation	0.002
66.	test_get_ea_for_zone	0.002
67.	test_get_ea_for_ip_with_floatingip_dissociation	0.002
68.	test_reset_ea_for_range	0.002
69.	test_get_ea_for_network	0.002
70.	test_creates_host_record_on_allocate_ip_no_dhcp	0.012
71.	test_creates_host_record_range_on_range_allocation_use_dhcp	0.005
72.	test_creates_host_record_on_allocate_ip_use_dhcp	0.005
73.	test_deletes_host_record	0.003
74.	test_creates_fixed_address_range_on_range_allocation	0.003
75.	test_deletes_fixed_address	0.002
76.	test_creates_fixed_address_on_allocate_ip	0.002
77.	test_sync_for_cloud	0.831
78.	test_sync_for_without_cloud	0.791
79.	test_create_network_sync_tenant_mismatch	0.038
80.	test_update_floatingip_sync	0.023
81.	test_get_instance_name_from_fip	0.006
82.	test_create_network_sync_same_tenan	0.004
83.	test_create_subnet_sync_should_call_resync	0.004
84.	test_update_network_sync	0.004
85.	test_create_network_alert_should_call_resync	0.003
86.	test_create_subnet_alert_should_call_resync	0.002
87.	test_delete_subnet_sync	0.001

88.	test_get_hostname_for_other_device_owners	0.003
89.	test_get_hostname_for_instance_name	0.003
90.	test_get_hostname_for_floating_ip_device_owner	0.002
91.	test_get_zone_name	0.002
92.	test_grid_operations	1.568
93.	test_get_next_authority_member_for_ipam	1.437
94.	test_get_next_dhcp_member	1.418
95.	test_get_next_authority_member_for_dhcp_with_one_cpm	1.332
96.	test_add_or_update_tenant	1.332
97.	test_add_and_get_tenant	1.322
98.	test_grid_management	1.295
99.	test_get_next_authority_member_for_dhcp_with_no_cpm	1.283
100.	test_network_view_management	0.220
101.	test_get_next_authority_member_for_dhcp_with_two_cpms	0.219
102.	test_update_subnet_zone_change	1.766
103.	test_allocate_specific_ip	1.679
104.	test_get_subnet	1.604
105.	test_allocate_subnet	1.567
106.	test_remove_subnet	1.512
107.	test_allocate_ip_from_pool	1.475
108.	test_deallocate_ip	1.413
109.	test_update_subnet_no_zone_change	1.383
110.	test_floating_address_request	0.003
111.	test_dhcp_port_address_request	0.003
112.	test_fixed_address_request	0.002
113.	test_any_address_request	0.002
114.	test_router_gateway_address_request	0.002
115.	test_auto_address_request	0.002
116.	test_rollback_wrapper_on_delete_failure	0.007
117.	test_rollback_wrapper	0.002

References:

1. Infoblox Administrator Guide [http://dloads.infoblox.com/direct/appliance//NIOS/NIOS_AdminGuide_6.3.pdf]
2. Networking-infoblox [https://pypi.python.org/pypi/networking-infoblox]
3. Infoblox Grid [https://www.infoblox.com/sites/infobloxcom/files/resources/infoblox-datasheet-the-infoblox-grid.pdf]
4. Release Notes [https://infoblox-2.custhelp.com/ci/fattach/get/76653/0/filename/NIOS_7.0.1_ReleaseNotes.pdf]
5. Infoblox Support [https://support.infoblox.com]

6. Openstack Networking Infoblox [http://docs.openstack.org/developer/networking-infoblox/index.html]
7. [Install MOS manually] https://docs.mirantis.com/openstack/fuel/fuel-8.0/quickstart-guide.html#installing-mirantis-openstack-manually
8. [Fuel User Guide] https://docs.mirantis.com/openstack/fuel/fuel-8.0/pdf/Fuel-8.0-UserGuide.pdf