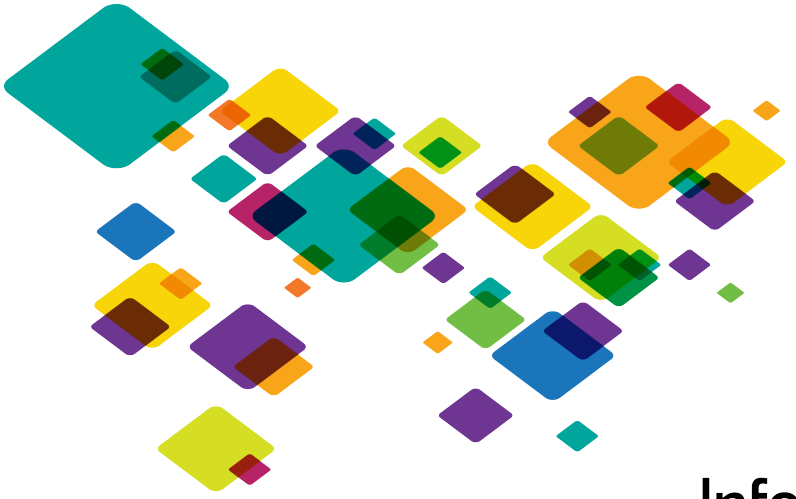

Infoblox
CONTROL YOUR NETWORK

Infoblox IPAM Driver for Openstack Neutron

Version 10.0.1 for OpenStack Ocata
Version 11.0.1 for OpenStack Pike
Version 12.0.0 for OpenStack Queens

IPAM Driver for OpenStack Neutron

Error rendering macro 'content-column' : null

Installing Infoblox IPAM Driver for OpenStack Neutron

The Infoblox IPAM adapter for OpenStack Neutron should be installed on the controller nodes that are running your `neutron-server`. The installation consists of the following steps:

1. Configure the Infoblox Grid, as described in *Configuring the Infoblox Grid*.
2. Install the adapter module on the controller nodes, as described in *Installing the IPAM Driver*.
3. Run database migrations to create the Infoblox tables, as described in *Creating the Infoblox Neutron Database*.
4. Modify neutron.conf and nova.conf, as described in *Modifying the OpenStack Configuration*.
5. Start the Infoblox IPAM Agent, as described in *Starting the Infoblox IPAM Agent*.
6. Restart the neutron-server and nova-compute services, as described in *Restarting the Services*.
7. Optional: Migrate networks, subnets, and ports from OpenStack to the Infoblox Grid, as described in *Creating Extensible Attribute Definitions and Network View Associations*.

Configuring the Infoblox Grid

To set up your Infoblox Grid to work with the Infoblox IPAM Driver for OpenStack Neutron, perform the steps described below.

Note: To get a holistic view of your cloud network environment through the NIOS management system — the Infoblox Grid Manager, enable the Cloud Network Automation license in the Grid Manager. This allows you to manage all the DNS, DHCP, and IPAM (DDI) information organized by tenant, network, and virtual machine for cloud objects in the Infoblox Grid Manager's Cloud tab. For more information, see *Infoblox NIOS Administrator Guide*.

Do the following:

1. Create a non-administrative Infoblox user (if required) and, optionally a group, as described in *Configuring a User*.
2. Create Extensible Attribute (EA) definitions and associate network views with OpenStack, as described in *Creating Extensible Attribute Definitions and Network View Associations*.
3. Set the EAs to values representing the desired behavior, as described in *Setting EA values to Configure the Integration*.
4. Identify the Grid members for serving OpenStack. For information, see the section Identifying Members for Serving OpenStack in the *Configuration Guide for Infoblox IPAM Driver for OpenStack Neutron*.

Configuring a User

You can use the admin user or create a separate user for the Infoblox integration. If you have a Cloud Network Automation license, a cloud-api-only group is created automatically. You can create a user in this group. Otherwise, you may want to create a group specifically for this integration.

In any scenario, the user or group must have the following minimum set of permissions for full DDI functionality to work the Infoblox IPAM Driver for OpenStack Neutron:

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
[DNS, DHCP, IPAM]	All IPv4 host addresses	IPv4 host address	RW
[DNS, DHCP, IPAM]	All IPv6 host addresses	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 the 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 views	Network view	RW	
[DHCP, IPAM]	All IPv4 networks	IPv4 network	RW	
[DHCP, IPAM]	All IPv6 networks	IPv6 network	RW	

Installing the IPAM Driver

The Infoblox IPAM Driver for OpenStack Neutron needs to be installed on each controller node that is running the Neutron service. The Infoblox IPAM Driver for OpenStack Neutron is available from PyPi and can be installed using the `pip install` command.

Installing the Infoblox IPAM Agent

The `infoblox-ipam-agent` init script that is used to start the Infoblox IPAM Agent will be installed as part of the `pip install` command. By default, the `infoblox-ipam-agent` init script is installed as `/usr/local/etc/init.d/infoblox-ipam-agent`. To install the script in `/etc/init.d`, specify `--install-option` as follows:

```
$ sudo pip install --install-option="--install-data=/" networking-infoblox
```

Latest Releases

To install the most recent production release, use the following command:

```
$ sudo pip install networking-infoblox
```

For compatibility between the releases of the Infoblox IPAM Driver and OpenStack Neutron, see the following table:

Release of Infoblox IPAM Driver for OpenStack Neutron	OpenStack Neutron Version
10.0.1	Ocata
11.0.1	Pike
12.0.0	Queens

You can install the correct version of the Driver using the following command:

```
$ sudo pip install networking-infoblox==X.0.Y
```
where X represents the IPAM Driver version.

Note: Infoblox strongly recommends you use 10.0.1, 11.0.1, 12.0.0 and later versions of the IPAM Driver as they include critical bug fixes.

Creating the Infoblox Neutron Database

The Infoblox IPAM Driver for OpenStack Neutron 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.

Modifying the OpenStack Configuration

The `neutron.conf` files on each controller node, as well as the `nova.conf` files on each compute node, must be updated as described in the following section:

- *Neutron*

- *Nova*

Neutron

The Grid connectivity and credentials configuration must be added to the `neutron.conf` file in infoblox and infoblox-dc stanzas. The infoblox stanza contains keystone authentication and a list of Grids, and then in each there is an infoblox-dc stanza containing the appropriate configuration for each Grid. Support for multiple Grids is not yet available.

For keystone authentication, add an entry for the following configuration:

```
keystone_auth_uri = <auth_uri>
keystone_admin_username = <username>
keystone_admin_password = <password>
```

If `keystone_auth_uri` does not include keystone version, then configure `keystone_auth_version` or it will take version 2.0 by default.

```
keystone_auth_version = <auth_version>
```

For keystone version 2.0, add:

```
keystone_admin_tenant_name = <tenant_name>
```

For keystone version 3, add:

```
keystone_admin_user_domain_id = <user_domain_id>
```

Note: For keystone version 3, you can set any one of the following scopes: project level or domain level.

```
# if authorization is project-level scope add:
keystone_admin_project_name = <project_name>
keystone_admin_project_domain_id = <project_domain_id>

# if authorization is domain-level scope add:
keystone_admin_domain_id = <domain_id>
```

For TLS support, add the following keystone configuration:

```
cafile = <cafile>
insecure = <True/False> # default value: False
cert = <cert>
key = <key>
```

The following table lists general configuration options for the Infoblox IPAM Driver for OpenStack Neutron.

Option	Description
keystone_auth_uri	OpenStack keystone authentication URI
keystone_admin_username	OpenStack keystone admin user name
keystone_admin_password	Password of keystone admin user
keystone_auth_version	OpenStack keystone version
keystone_admin_tenant_name	Tenant name of keystone admin user
keystone_admin_user_domain_id	User Domain Id of keystone admin user
keystone_admin_project_name	Project name of keystone admin user
keystone_admin_project_domain_id	Project Domain Id of keystone admin user
keystone_admin_domain_id	Domain Id of keystone admin user
cafile	CA certificate bundle file for keystone authentication
insecure	Disable server certificate verification
cert	Client certificate bundle file for keystone authentication
key	Client certificate key file for keystone authentication
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. The ID should begin with 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. Supported versions: WAPI version 2.5 and later ; NIOS 8.1.0 and later
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 the “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.

Example:

```
[DEFAULT]
ipam_driver = infoblox
notification_driver = messagingv2
notification_topics = notifications

[infoblox]
cloud_data_center_id = 1
keystone_admin_project_domain_id = default
keystone_admin_user_domain_id = default
keystone_admin_domain_id = default
keystone_admin_username = admin
keystone_admin_password = infoblox
keystone_auth_uri = http://10.39.12.101/identity
keystone_auth_version = v3
cafile = /opt/stack/data/ca-bundle.pem
insecure = False
key = <key>
cert = <cert>

[infoblox-dc:1]
grid_master_host = 10.35.114.2
grid_master_name = gmc.com
admin_user_name = admin
admin_password = infoblox
wapi_version = 2.7
wapi_max_results = -50000
```

If Cloud Platform (CP) member is part of the NIOS Grid, then create a user with `cloud-api-only` user group and configure the user as described in the section *Configuring a User* and update neutron.conf as follows:
Example:

```
[infoblox-dc:1]
grid_master_host = 10.35.114.2
grid_master_name = gmc.com
admin_user_name = cloud-api-only_user_name
admin_password = cloud-api-only_user_password
wapi_version = 2.7
wapi_max_results = -50000
```

If you want to use CP members only, use the CP Members Host IP address in the `grid_master_host` field.
Example:

```
[infoblox-dc:1]
grid_master_host = x.x.x.x(CP Host IP)
grid_master_name = gmc.com (grid_master_hostname)
admin_user_name = cloud-api-only_user_name
admin_password = cloud-api-only_user_password
wapi_version = 2.7
wapi_max_results = -50000
```

Nova

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:

```
notification_driver = messagingv2
notification_topics = notifications
notify_on_state_change = vm_state
```

Starting the Infoblox IPAM Agent

Depending on your distribution, you will need to create and configure init.d and/or systemd service definitions for the `infoblox-ipam-agent`. Once that is done, you should start the agent.

To start it manually, without any init.d or systemd setup, you run the following command as the same user that runs neutron-server:

```
$ /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
```

Restarting the Services

The appropriate services must be restarted so that the changes to the configuration files take effect.

Neutron

Restart neutron-server on each node running it. The exact command may vary based upon your distribution. In Ubuntu, the command is as follows:

```
$ sudo service neutron-server restart
```

Nova

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 as follows:

```
$ sudo service nova-compute restart
```

Creating keystone_admin File

In order to run the script to configure the integration, 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.

Note: `networking-infoblox` should be successfully configured before running the migration script.

Following is the script that needs to be run to set the environment variables:

```
$ 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
```

For keystone behind TLS:

```
$ cat keystone_admin
unset OS_SERVICE_TOKEN
export OS_USERNAME=admin
export OS_PASSWORD=mysecret
export OS_AUTH_URL=https://controller:5000/v3
export PS1='[\u@\h \W(keystone_admin)]\$ '
```

```
export OS_TENANT_NAME=admin
export OS_PROJECT_NAME=admin
export OS_REGION_NAME=RegionOne
export OS_PROJECT_DOMAIN_ID=default
export OS_USER_DOMAIN_ID=default
export OS_DOMAIN_ID=default
export SERVICE_ENDPOINT=https://controller:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_CACERT=/etc/ssl/certs/apache-selfsigned.crt
export OS_INSECURE=False
export OS_KEY=<key>
export OS_CERT=<cert>
```

```
$ source keystone_admin
```

Creating Extensible Attribute Definitions and Network View Associations

The Infoblox IPAM Driver for OpenStack Neutron uses a variety of Extensible Attributes (EAs) to manage its configuration. The needed extensible attributes may be created automatically using the `create_ea_defs.py` script that can be found at location `/usr/local/lib/python2.7/dist-packages/networking_infoblox` under the `tools` directory in the distribution:

```
$ python create_ea_defs.py
```

The script will prompt you for the user name and password of an Infoblox NIOS admin user, which is needed to create the EA definitions.

The script will also prompt you for association or un-association of network views to OpenStack. This is an important step. You can use this script to select network views explicitly to use in OpenStack. An associated network view will have Cloud Adapter ID EAs stored on that network view. The Cloud Adapter ID is equivalent to `cloud_data_center_id` defined in neutron.conf.

If you use the Driver with Cloud Platform (CP), you should run `create_ea_defs.py` on the Grid Master:

1. Configure neutron.conf with the Grid Master details and administrator user.
2. Run `create_ea_defs.py`.
3. Change neutron.conf config to point to the CP member. User details can be changed to the Cloud User. If you have multiple OpenStack deployments for the same Grid Master, then this is needed only for the first OpenStack deployment.

Following are the lists of Extensible Attributes:

Name	Type	Comment
Address Scope ID Mapping	String	Mapping
Address Scope Name Mapping	String	Mapping
Admin Network Deletion	List	Grid Configuration
Allow Service Restart	List	Grid Configuration
Allow Static Zone Deletion	List	Grid Configuration
Cloud Adapter ID	String	Grid Configuration
DHCP Support	List	Grid Configuration
DNS Record Binding Types	String	Grid Configuration
DNS Record Removable Types	String	Grid Configuration
DNS Record Unbinding Types	String	Grid Configuration
DNS Support	List	Grid Configuration
DNS View	String	Grid Configuration
Default Domain Name Pattern	String	Grid Configuration
Default Host Name Pattern	String	Grid Configuration
Default Network View Scope	List	Grid Configuration
Default Network View	String	Grid Configuration
External Domain Name Pattern	String	Grid Configuration

External Host Name Pattern	String	Grid Configuration
Grid Sync Maximum Wait Time	Integer	Grid Configuration
Grid Sync Minimum Wait Time	Integer	Grid Configuration
Grid Sync Support	List	Grid Configuration
IP Allocation Strategy	List	Grid Configuration
Is Cloud Member	List	Grid Membership
Last Grid Sync Time	String	Grid Sync Report
NS Group	String	Grid Configuration
Network ID Mapping	String	Mapping
Network Name Mapping	String	Mapping
Network Template	String	Grid Configuration
Relay Support	List	Grid Configuration
Report Grid Sync Time	List	Grid Configuration
Subnet CIDR Mapping	String	Mapping
Subnet ID Mapping	String	Mapping
Tenant ID Mapping	String	Mapping
Tenant Name Mapping	String	Mapping
Tenant Name Persistence	List	Grid Configuration
Tenant Name	String	Tenant Name in OpenStack
Use Grid Master for DHCP	List	Grid Configuration
Zone Creation Strategy	List	Grid Configuration

Setting EA values to Configure the Integration

You must decide on the configuration you would like to use. For details on the configuration options, please refer to the *Configuration Guide for Infoblox IPAM Driver for OpenStack Neutron*.

The configuration is captured within the various EAs that were created in the previous step. In general, these EAs are set on the Grid Master member. To do this, you navigate to **Grid** -> **Grid Manager** -> **Members** and click on the Gear icon next to the Grid Master member. Choose the **Extensible Attributes** option. From there you can create and modify various EA values that will apply to the entire IPAM Driver integration.

```
# 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
```

Note: If you make changes to the Grid EA values in NIOS, wait for the minimum sync time for the values to get reflected or work. The minimum sync time is also an EA whose value can be specified in the Grid EA.

You can re-run the migration script as many times as needed.

If you want to use any member (CP or Grid Member), set the Extensible Attributes **Is Cloud Member** to **True**.

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_infoblox `tools` folder.

You need to run the sync neutron script once the admin file is sourced:

```
$ networking-infoblox(keystone_admin)]# python
```

```
networking_infoblox/tools/sync_neutron_to_infoblox.py
```

Upgrading Infoblox IPAM Driver for OpenStack Neutron

To upgrade the Driver from version 10.0.0 to 10.0.1, use the following command:
```
sudo pip install networking-infoblox==10.0.1
```

To upgrade the Driver from version 11.0.0 to 11.0.1, use the following command:
```
sudo pip install networking-infoblox==11.0.1
```

To upgrade the Driver to version 12.0.0, use the following command:
```
sudo pip install network-infoblox==12.0.0
```

Configuring Infoblox IPAM Driver for OpenStack Neutron

The Configuration Guide for the Infoblox IPAM Driver for OpenStack Neutron describes additional options for configuring the Infoblox side of the OpenStack Neutron and Infoblox Grids integration. The configuration options for Neutron and Nova, as well as the most common Infoblox configuration options, are described in the *Installation Guide for Infoblox Driver for OpenStack Neutron*.

Note: Infoblox recommends that you install the Cloud Network Automation license on your Grid before configuring the Infoblox IPAM Driver and using the `create_ea_defs.py` script to define Extensible Attributes (EAs). If you install the Cloud Network Automation license after running the `create_ea_defs.py` script, you might notice the following error message in the infoblox.log: “Hellfire initialization failed with Failed to initialize cloud-specific EAs.” Please note that this error does not affect any functionality in your Grid or IPAM Driver.

The configuration options should be set on the Grid Master using EAs. You can also pre-define mappings for OpenStack entities; those EAs will be set on the specific network view in Infoblox NIOS. This is discussed in more detail below.

Configuring additional options of the Infoblox IPAM Driver for OpenStack Neutron includes the following:

- *Grid Synchronization Settings*
- *Network View Mapping*
- *Domain and Host Name Patterns*
- *Identifying Members to Use*
- *Miscellaneous Grid Configurations*

Grid Synchronization Settings

When you make a change in NIOS to the EAs that represent the Driver configuration, that change must be synchronized to the Driver's local storage. Set on the Grid Master object, the following EAs define Grid synchronization settings:

- **Grid Sync Support**. This EA is used to choose whether to enable synchronization of Grid configuration from Infoblox NIOS. The default is True.
- **Grid Sync Minimum Wait Time**. This EA is used to define the minimum wait time, in seconds, before a synchronization is allowed to take place. The default is 60.
- **Grid Sync Maximum Wait Time**. This EA is used to define the maximum wait time, in seconds, between synchronizations. The default is 300.
- **Report Grid Sync Time**. This EA is used to allow reporting of Grid synchronization time. The default is False. If this is set to True, Last Grid Sync Time EA is used to store last Grid sync time. The infoblox-ipam-agent updates the Grid synchronization time. It is important to note that setting this EA to True requires a WRITE permission on the Grid member.

Network View Mapping

When creating a new object in Infoblox NIOS, the Infoblox IPAM Driver for OpenStack Neutron must know the network view in which to create the object. This is determined using a number of EAs.

In the simpliest form you can configure the Driver to automatically create network views as needed. The first EA that needs to be set is the Default Network View Scope. This EA defines the default mapping to network view when no mapping already exists within the Infoblox system. This can be any of the following values:

- **Single**. This means that any time a pre-existing mapping cannot be found, the resulting object should be placed within a single, specific network view. That view should be specified with another EA, Default Network View.
- **Tenant**. This means that any time a pre-existing mapping cannot be found, the resulting object should be placed within a network view determined by the OpenStack tenant that owns the object. If no network view tagged with that Tenant ID exists, then a new network view is created with the name `tenant_name.tenant_id`.
- **Address Scope**. This means that any time a pre-existing mapping cannot be found, the resulting object should be placed within a network view determined by the OpenStack address scope associated with the object.
- **Network**. This means that any time a pre-existing mapping cannot be found, the resulting object should be placed within a network view determined by the OpenStack network. This is rarely used and primarily is provided for use in automated testing, where the same tenant may create multiple OpenStack Network entities with overlapping subnets.
- **Subnet**. This means that any time a pre-existing mapping cannot be found, the resulting object should be placed within a network view determined by the OpenStack subnet. This is rarely used, but can be necessary in certain deployments that utilize SDN plug-ins that allow spanning subnets across OpenStack Neutron installations.

Alternatively, You can pre-define mappings by creating a network view and then tagging it with the name of a tenant, address scope, or network, in addition to CIDR of a subnet. This can be done by creating the following EAs on a network view object. Each of these EAs allows multiple values to be specified. This includes:

- **Subnet CIDR Mapping**. If a subnet created matches one of the CIDR values specified in this EA, the subnet is created under this network view.
- **Subnet ID Mapping**. If the ID of a subnet created matches one of the values specified in this EA, the subnet is created under this network view.
- **Network Name Mapping**. If the name of a network matches one of the values specified in this EA, the subnets within the network are created under this network view.
- **Network ID Mapping**. If the ID of a network matches one of the values specified in this EA, the subnets within the network are created under this network view.
- **Tenant Name Mapping**. If the name of a tenant matches one of the values specified in this EA, objects within the tenant are created under this network view.
- **Tenant ID Mapping**. If the ID of a tenant matches one of the values specified in this EA, objects within the tenant are created under this network view.
- **Address Scope Name Mapping**. If the name of an address scope matches one of the values specified in this EA, objects within the address scope are created under this network view.
- **Address Scope ID Mapping**. If the ID of an address scope matches one of the values specified in this EA, objects within the address scope are created under this network view.

Note: When the Grid Master is not connected while using the Driver with a CP member, make sure that the “Default Network View” EA value is set to “Delegated Network View” for the corresponding CP member.

Domain and Host Name Patterns

This includes the following EAs:

- **Default Domain Name Pattern**. This EA is used to control how domain names for IP address allocations are determined. Typically this pattern is used for private networks (not external), but if External Domain Name Pattern is not set, it applies to all network types. This EA can be set to a fixed string, or can use patterns to generate unique zone names. For example, you may set this to `cloud.example.com` to have all DNS entries within that domain. Or, you can use substitution patterns: `{tenant_name}.cloud.example.com` would place IPs associated with each tenant in their own domain.
  For domain names, the following patterns are supported:
  — `{network_name}` is replaced with the OpenStack Network Name.
  — `{network_id}` is replaced with the OpenStack Network ID.
  — `{tenant_name}` is replaced with the OpenStack Tenant Name. Note that for this to work, the Tenant Name Persistence EA must be set to True.
  — `{tenant_id}` is replaced with the OpenStack Tenant ID.
  — `{subnet_name}` is replaced with the OpenStack Subnet Name.

  Note: In the Ocata releases, the network, tenant, and subnet name must not contain uppercase letters, otherwise there is an exception. This does not apply to the Pike release.

  — {subnet_id} is replaced with the OpenStack Subnet ID.
  The DNS zones are created under a DNS View, the name of which is constructed using the DNS View EA.
- **External Domain Name Pattern**. This EA is used to control how domain names for IP address allocations are determined for external networks. If this EA is not set, then Default Domain Name Pattern is used for external networks. The same patterns are supported as for Default Domain Name Pattern.
- **Default Host Name Pattern**. This EA controls host names in a manner similar to the way Default Domain Name Pattern controls domain names. In addition to the patterns supported for domain names, this EA supports these:
  — `{port_id}`. The port ID of the port associated with the IP.
  — `{instance_id}`. The Nova instance ID of the VM associated with the port.
  — `{instance_name}`. The Nova instance name of the VM associated with the port.

  Note: Special characters are not allowed in the instance name.

  — `{ip_address}`. The IP address for this port or host, with dots replaced by dashes.
  — `{ip_address_octet{n}}` where n is a number 1-4. This is for IPv4 addresses only. For example, if the pattern is `host-{ip_address_octet2}-{ip_address_octet3}` and the IP is `10.1.2.3`, then the resulting hostname is `host-1-2`.

  Note: If the host name pattern is set to {instance_name.constant}, then you should not create two instances with the same name in OpenStack as this will create the same DNS host record for both instances.

- **External Host Name Pattern**. This EA controls host names in the same way as Default Host Name Pattern, but applies only to hosts allocated in external network. If External Host Name Pattern is not set, Default Host Name Pattern is used for external networks.
  Per NIOS restriction, the domain label must not be longer than 63 characters.

- **Tenant Name Persistence**. Since Neutron does not have direct access to tenant names (they are part of Keystone), the Infoblox IPAM agent can cache those names it receives from the message bus. This reduces the Keystone API calls needed to retrieve tenant name. This EA controls this behavior; it must be set to True for tenant name support in domain or host names.IPAM and DHCP/DNS Support IPAM and DHCP/DNS support can be configured by tuning DHCP Support and DNS Support EAs. These include the following:
- **DHCP Support**. When DHCP support is set to False, and **Enable DHCP on Openstack** is set to True, dnmasq-based DHCP is used.

Note: For NIOS DHCP service to function, you should disable the Openstack DHCP agent using the following command:
`$ openstack network agent set --disable <OpenStack DHCP Agent ID>`

- **DNS Support.** When set to False, DNS support is disabled. Enabling it allows DNS record generation and DNS protocol. The default is False.

Currently the following configurations are supported.

IPAM Only:

- DNS Support = False
- DHCP Support = False

Full DHCP/DNS Support:

- DNS Support = True
- DHCP Support = True

Creating multiple network views with specific Default Network View Scope EA:

- DNS Support = True
- DHCP Support = False/True
  — If the DHCP Support EA is False:
    - When the Default Network View Scope EA is set to “Single”, the Grid Master or Grid member are not assigned to the network and multiple networks are created in the default or custom network view.
    - When the Default Network View Scope is set to “Tenant”/”Network”/”Subnet”/”Address Scope”, the Grid Master or Grid member are not assigned to the network, and a network view is added in NIOS for each new network.

  — If the DHCP Support EA is True:
    - When the Default Network View Scope is set to “Single”, the Grid Master is assigned to multiple networks in the default or custom network view.
    - When the Grid is standalone and the Default Network View Scope is set to “Tenant”/”Network”/”Subnet”, we can add only one network with the member assigned.
    - When the Grid is standalone with a member and the Default Network View Scope is set to “Tenant”/”Network”/”Subnet”, we can add only two networks: one to the Grid Master and another to the Grid member.

IP Allocation and DNS Record Creation

This includes the following EAs:

- **IP Allocation Strategy.** This EA is used to choose between Host Record and Fixed Address for IP allocation. If chosen for Fixed Address, DNS records associated with a fixed address are controlled by the additional EAs below.
- **DNS Record Binding Types.** List of DNS records to generate and bind to a fixed address during IP allocation. Supported DNS record types are `record:a` (for A records), `record:aaaa` (for AAAA records), and `record:ptr` (for PTR records). This is a multi-value EA, with one of these entries per value.
- **DNS Record Unbinding Types.** List of DNS records to unbind from a fixed address during IP deallocation. Supported DNS record types are the same as DNS Record Binding Types.
- **DNS Record Removable Types**. List of associated DNS records to delete when a fixed address is deleted. This is typically a list of DNS records created independently of the Infoblox IPAM Driver. Supported DNS record types are `record:a, record:aaaa, record:ptr, record:txt`, and `record:cname`.

Note: A DHCP port IP is an exception to this. The DHCP port IP is created as a host record with DHCP disabled to allow IP aliasing, regardless of IP Allocation Strategy configuration. IP aliasing is used in OpenStack when multiple subnets are created in the same network. Each subnet requires a DHCP port IP and those IPs are all assigned to the same DHCP port, but only one MAC address exists. If IPAM only support configuration is used, DNS is disabled as well for the host record.

Identifying Members to Use

In order to serve DHCP and DNS, you must pick Grid members to be registered to Neutron. You should exclude network discovery members and reporting members since they cannot serve DHCP and DNS. For the members to serve DHCP and DNS, the licenses must be properly installed and services must be properly running.

In general in order to utilize Infoblox for DHCP, you will need to use an SDN solution that provides a DHCP relay function. The standard Neutron functions do not provide relay.

To identify a Grid member as available for use by OpenStack, you must set the EA Is Cloud Member to True. If you are running a Grid but the GM is not configured and licensed for DNS or DHCP, set Use Grid Master for DHCP EA on the GM object to False. This will exclude the GM from being selected to serve DHCP or DNS.

Note: To avoid a request to restart DHCP service for updating Fixed IP address refer *Configuring Fixed Addresses without Restarting DHCP Service* in the *Infoblox NIOS Administrator Guide*.

Miscellaneous Grid Configurations

This includes the following options:
• **NS Group**. Name of the Name Server Group that is used for serving DNS for all DNS zones. The default is None, in whichccase DNS service members are selected based on mapping conditions.
• **Network Template**. Name of the Template to use when a network is created. A Template contains predefined network settings. The default is None.
• **Admin Network Deletion**. Specifies whether to delete object from Infoblox when an Admin Network is deleted from OpenStack. A network that is specified as “external” and/or “shared” is considered an Admin Network. The default is False.
• **Relay Support.** Specifies whether a Relay is used. If set to False, then DNS Servers option is set to the DNS Member that the IPAM Driver assigns. If True, DNS Servers option is set to the same IP as DHCP Port for the subnet. However, if the user specifies Nameservers option when the OpenStack subnet is created, then only the user provided nameservers would be used for DNS Servers option, irrespective of the Relay Support flag.

Technical Support

Infoblox Technical Support provides assistance via the Web, e-mail, and telephone. The Infoblox Support web site at *https://support.infoblox.com* provides access to product documentation and release notes, but requires the user ID and password you receive when you register your product online at: *http://www.infoblox.com/support/customer/evaluation-and-registration.*