Breaking the Ice with OpenStack Neutron

Muneeb Ahmad

About me..

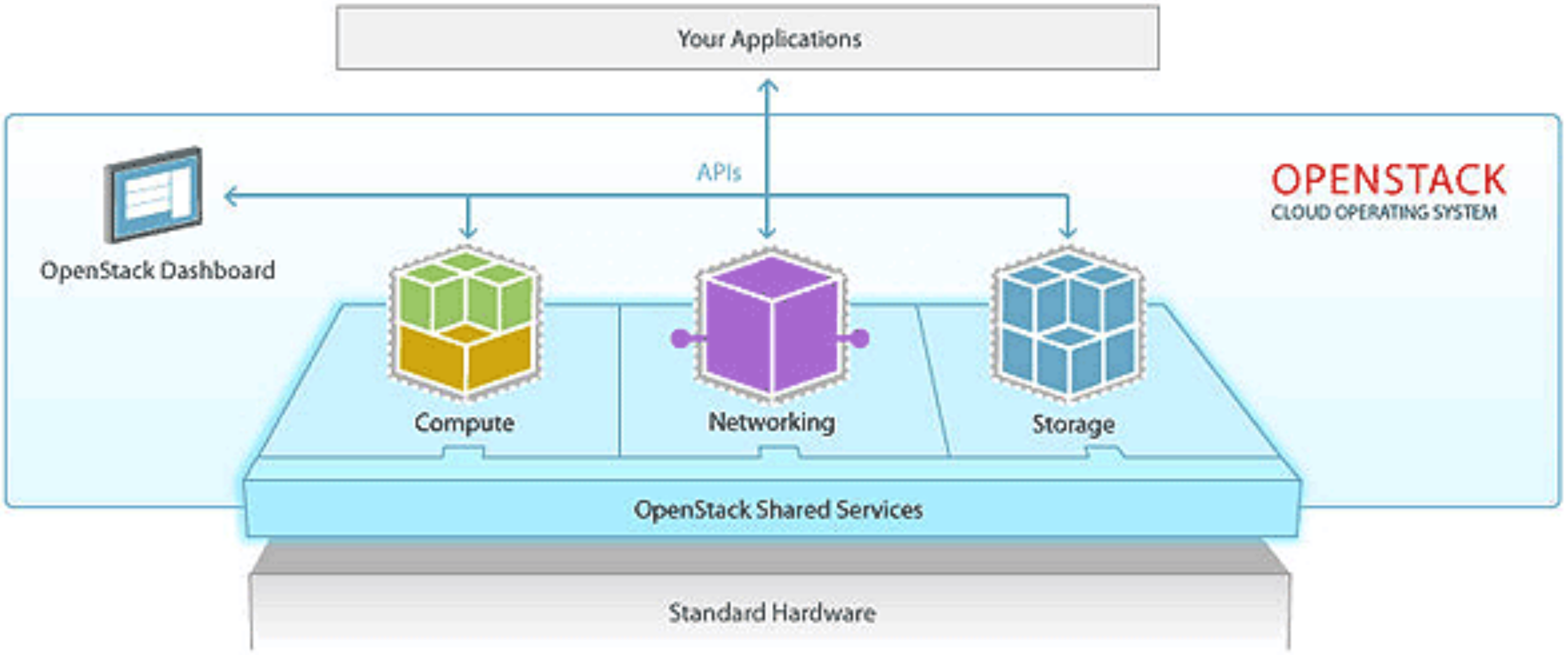
Senior Design Engineer at PLUMgrid Inc with over three years of experience in implementing software-defined networking (SDN), high availability, distributed systems and APIs. An active member of the OpenStack community, with contributions in the networking ecosystem, focused primarily on Neutron project in OpenStack.

muneeb@plumgrid.com

@smuneebahmed

What is OpenStack?

- **OpenStack** is a cloud management system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.



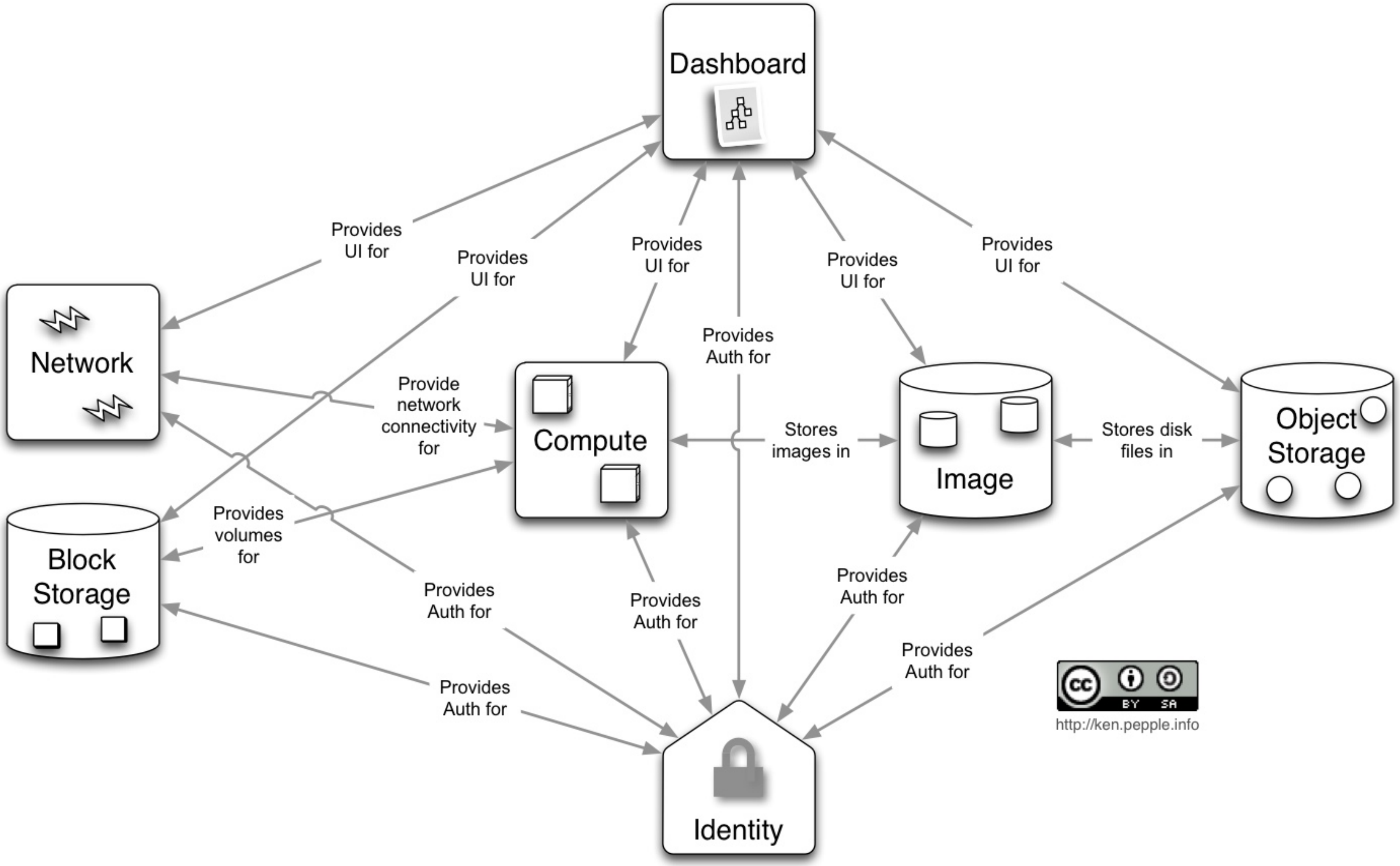
OpenStack Core Services

Compute (Nova) provides virtual servers on demand. Compute resources are accessible via APIs for developers building cloud applications and via web interfaces for admins and users. The compute architecture is designed to scale horizontally on standard hardware.

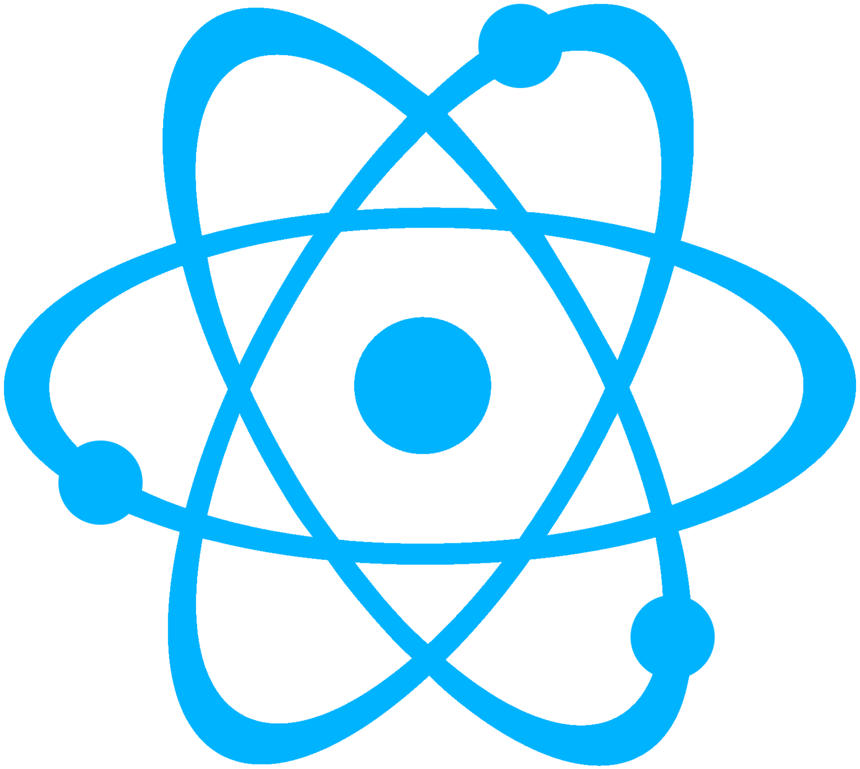
Network (Neutron) is a pluggable, scalable and API-driven system for managing networks and IP addresses. It allows users to customize network settings, then provision different network types (such as flat networks, VLANs or virtual private networks) on demand.

Block Storage (Cinder) provides persistent block storage guest to guest VMs. They're meant to be assigned to compute instances to allow for expanded storage. The Cinder software manages the creation of these blocks, plus the operations for attaching and detaching the blocks to compute servers.

OpenStack Core Services (Conceptual)



Neutron



Neutron Overview

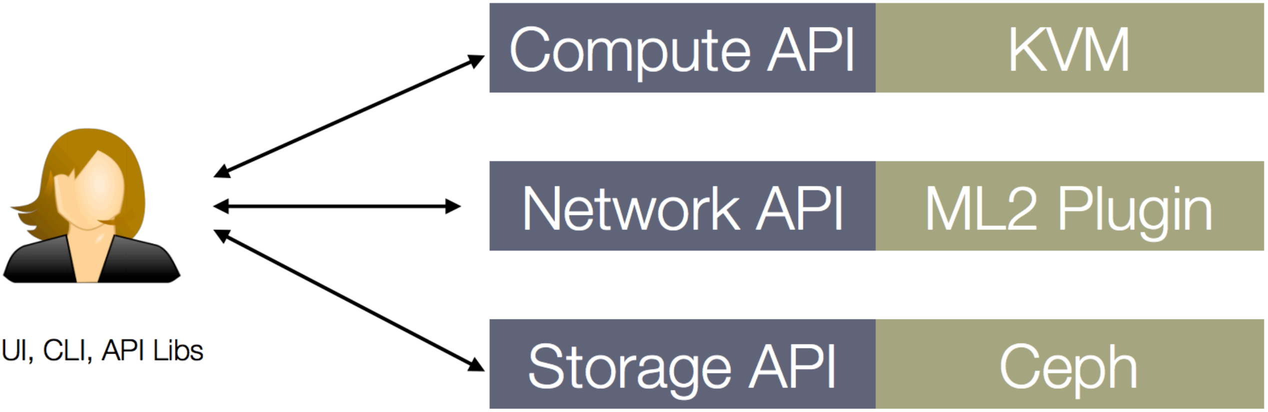
- Incubation project in April 2011
- Promoted to Core Project at Folsom Summit (April 2012)
- Neutron solves two main problems in Nova – Network:
  1. Limited networking technology
     - Basic linux bridging-based implementation
     - Limited features (ACL, QoS, VPN?)
     - Limited multi-tenancy isolation – 802.1q VLAN tags

  2. Limited User/Tenant control over the network
     - Tenant cannot create their own network topologies
     - Tenant cannot leverage different network virtualization technologies

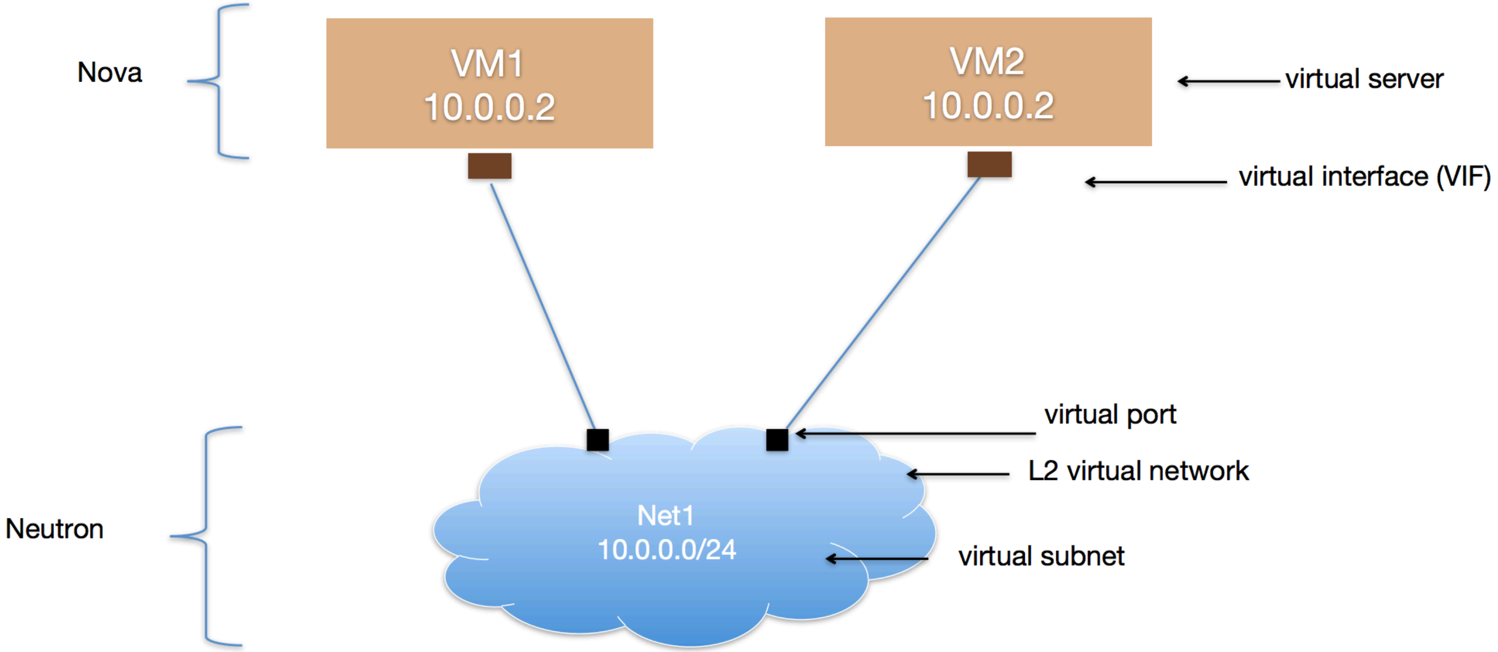
Why use Neutron?

- Provides REST APIs to manage network connections for the resources managed by other OpenStack services (e.g. Nova)
- Create rich topologies
- Technology agnostic
- Extensible
- Multi-tenancy: Isolation, Abstraction, full control over virtual networks
- Modular design: API specifies service, vendor provides its implementation
- Advance Services Support (LB, VPN, Firewall)

What does the user see?



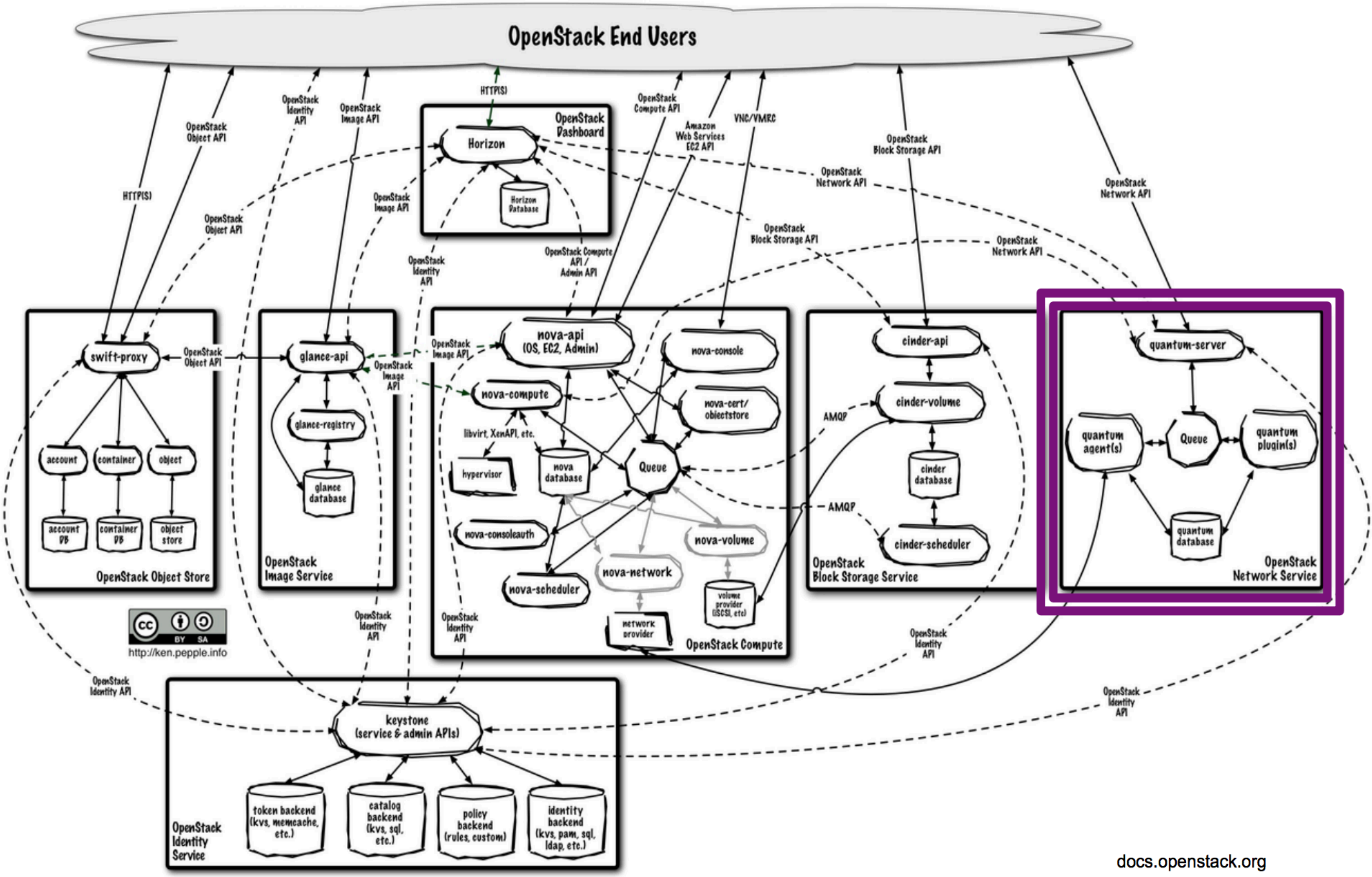
Abstractions



Design Goals

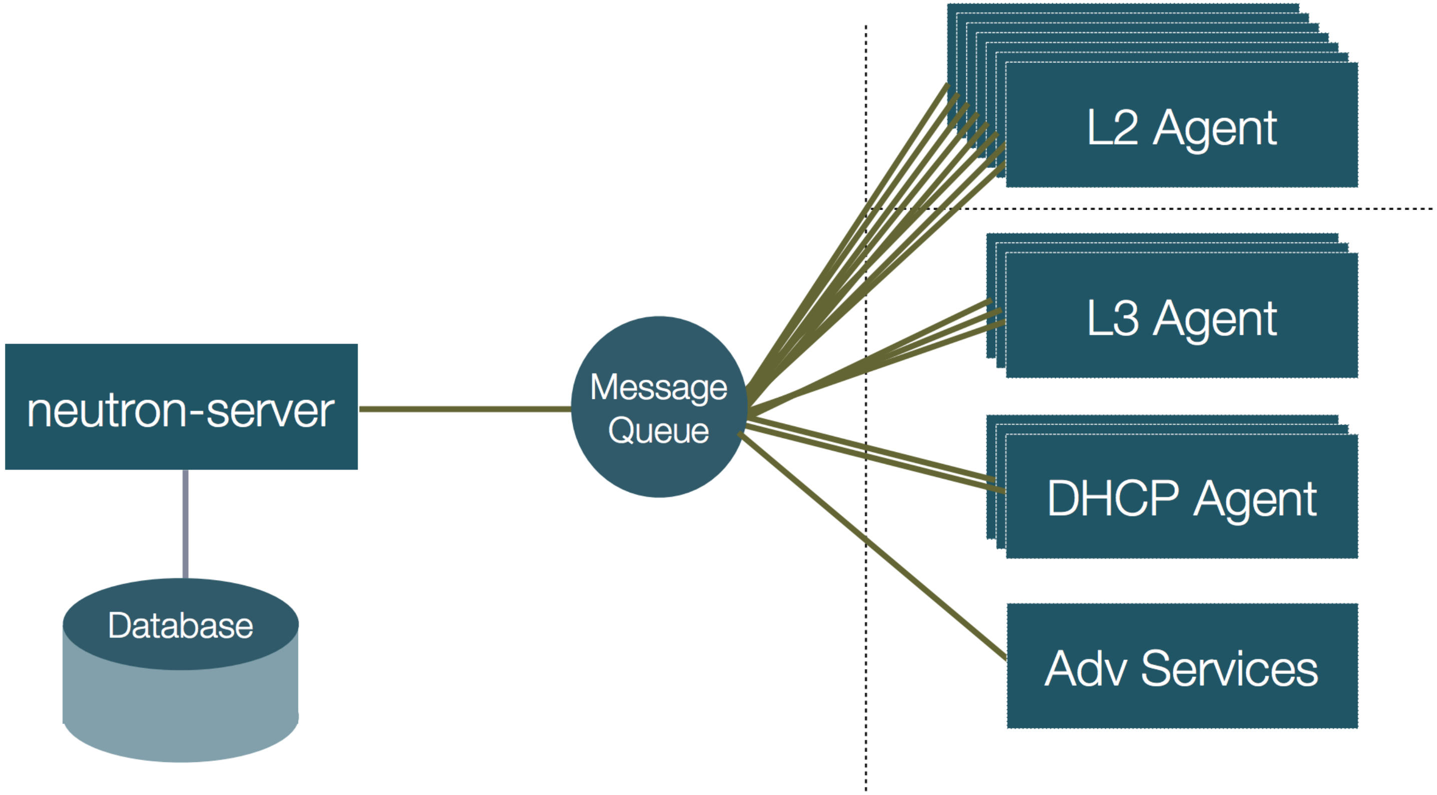
- Unified API
- Small Core
- Pluggable Open Architecture
- Extensible

Architecture



docs.openstack.org

Basic Deployment


neutron-server

Database

Message Queue

L2 Agent

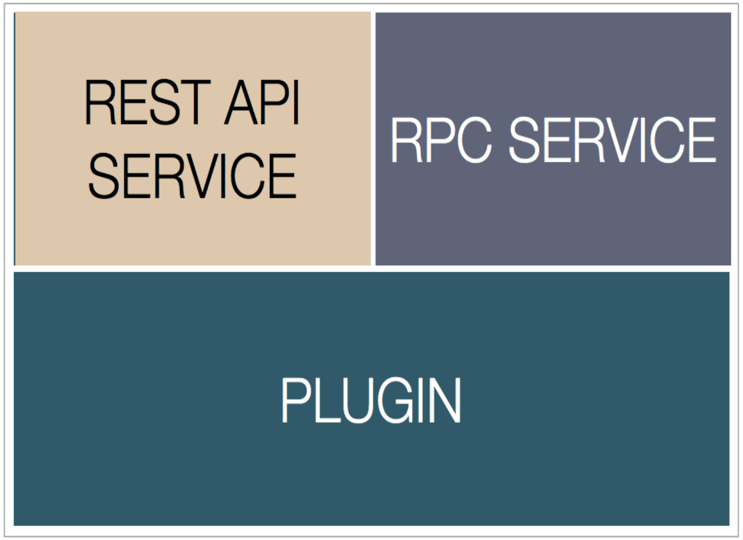
L3 Agent

DHCP Agent

Adv Services

neutron-server

- REST API
  - HTTP(S) Python WSGI Application
  - Customary TCP port is 9696
  - Exposes logical resources
    - networks
    - subnets
    - ports
  - Request/Response Serialization

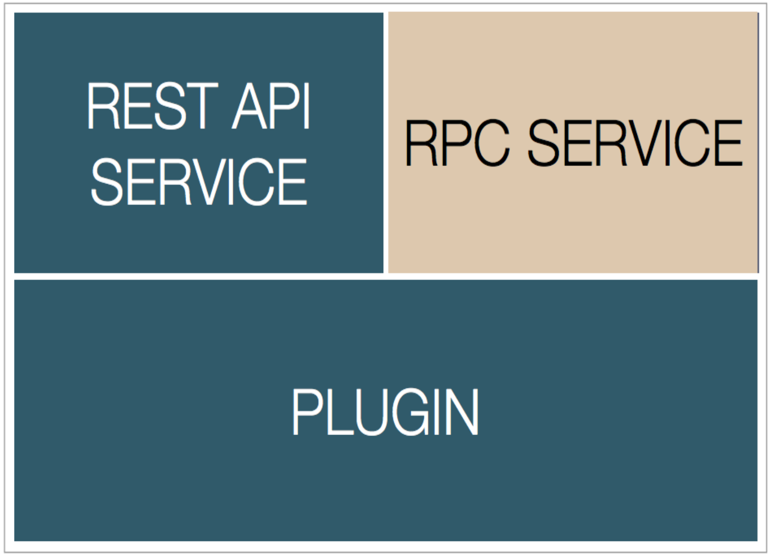

REST API SERVICE

RPC SERVICE

PLUGIN

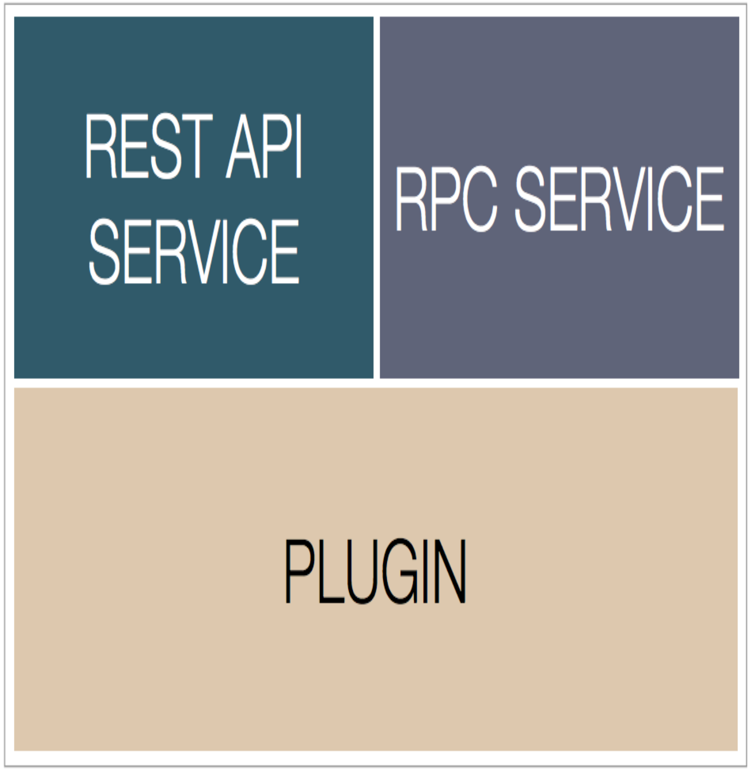
neutron-server

- RPC Service
  - AMQP via Oslo messaging modules
  - Enables bidirectional agent communication
  - Optional



neutron-server

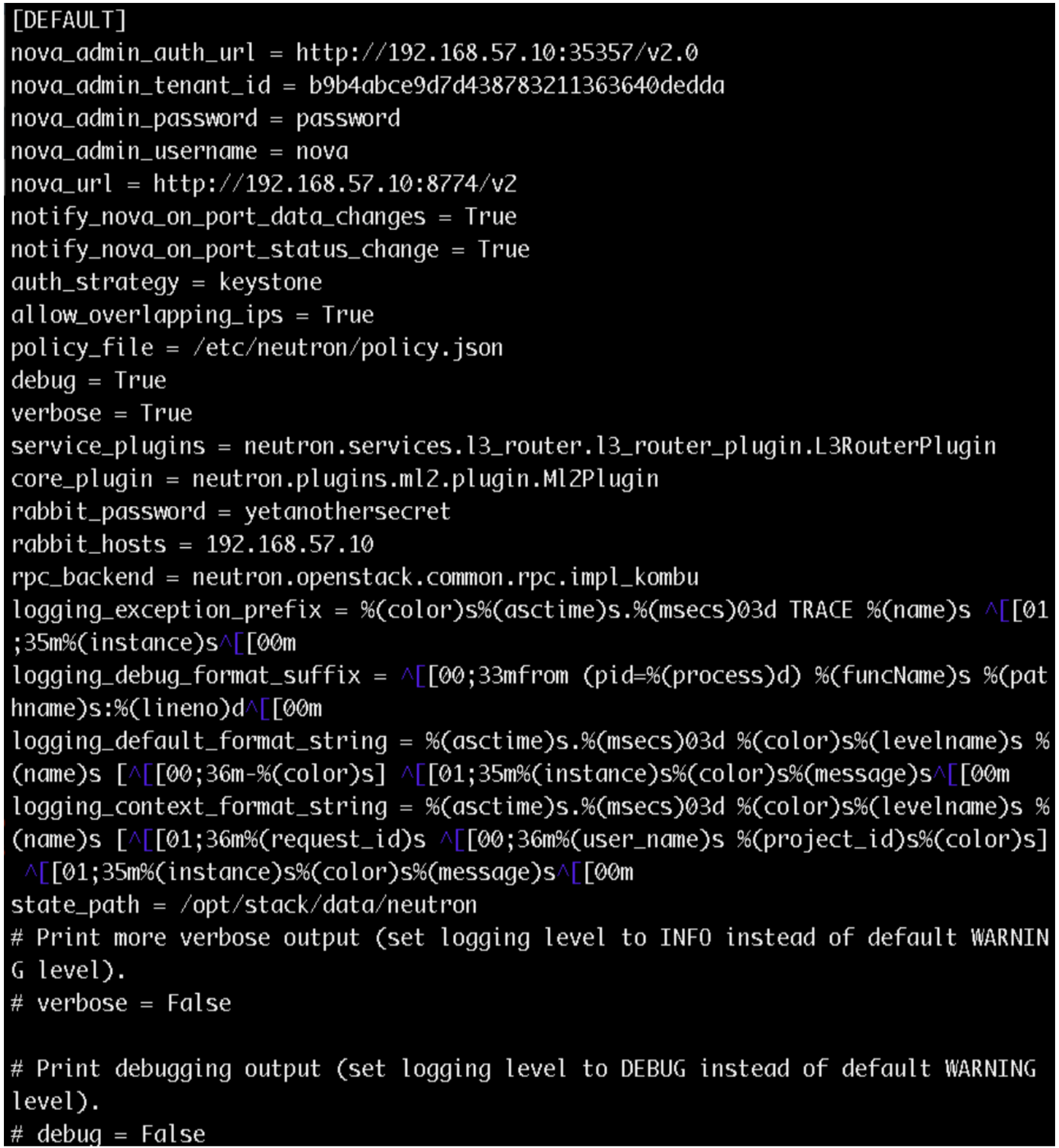
- PLUGIN
  - Written in Python
  - Only one is active
  - Must implement V2 API calls
  - Optional database access
  - Optional extension support



The Plugin

core_plugin = neutron.plugins.ml2.plugin.Ml2Plugin

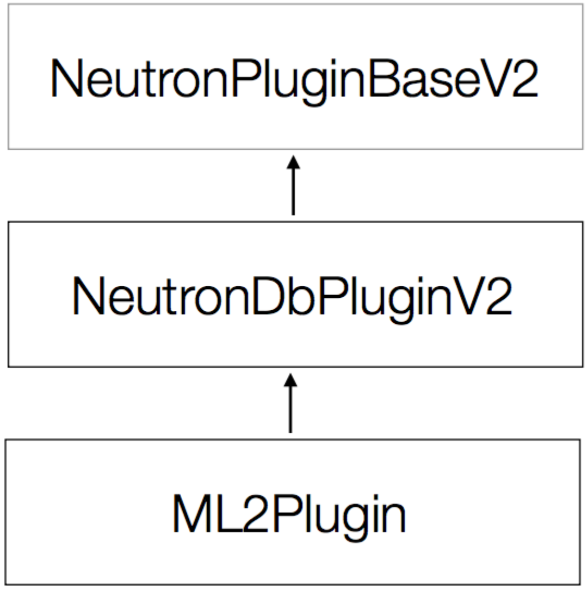
ML2Plugin



```
[DEFAULT]
nova_admin_auth_url = http://192.168.57.10:35357/v2.0
nova_admin_tenant_id = b9b4abce9d7d438783211363640dedda
nova_admin_password = password
nova_admin_username = nova
nova_url = http://192.168.57.10:8774/v2
notify_nova_on_port_data_changes = True
notify_nova_on_port_status_change = True
auth_strategy = keystone
allow_overlapping_ips = True
policy_file = /etc/neutron/policy.json
debug = True
verbose = True
service_plugins = neutron.services.l3_router.l3_router_plugin.L3RouterPlugin
core_plugin = neutron.plugins.ml2.plugin.Ml2Plugin
rabbit_password = yetanothersecret
rabbit_hosts = 192.168.57.10
rpc_backend = neutron.openstack.common.rpc.impl_kombu
logging_exception_prefix = %(color)s%(asctime)s.%(msecs)03d TRACE %(name)s ^[[01
;35m%(instance)s^[[00m
logging_debug_format_suffix = ^[[00;33mfrom (pid=%(process)d) %(funcName)s %(pat
hname)s:%(lineno)d^[[00m
logging_default_format_string = %(asctime)s.%(msecs)03d %(color)s%(levelname)s %
(name)s [^[[00;36m-%(color)s] ^[[01;35m%(instance)s%(color)s%(message)s^[[00m
logging_context_format_string = %(asctime)s.%(msecs)03d %(color)s%(levelname)s %
(name)s [^[[01;36m%(request_id)s ^[[00;36m%(user_name)s %(project_id)s%(color)s]
 ^[[01;35m%(instance)s%(color)s%(message)s^[[00m
state_path = /opt/stack/data/neutron
# Print more verbose output (set logging level to INFO instead of default WARNIN
G level).
# verbose = False

# Print debugging output (set logging level to DEBUG instead of default WARNING
level).
# debug = False
```

The Plugin

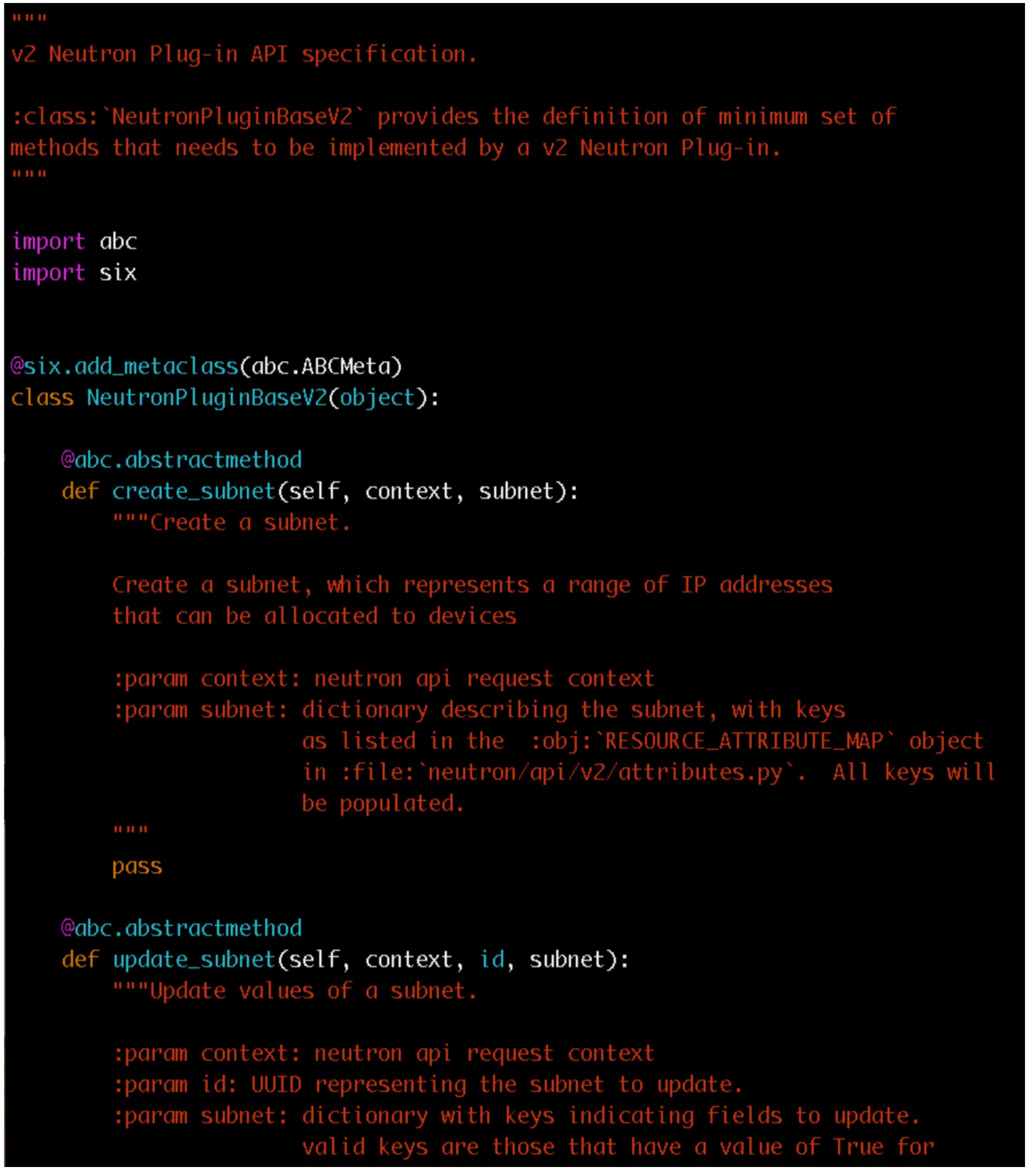
core_plugin = neutron.plugins.ml2.plugin.Ml2Plugin



NeutronPluginBaseV2

NeutronDbPluginV2

ML2Plugin



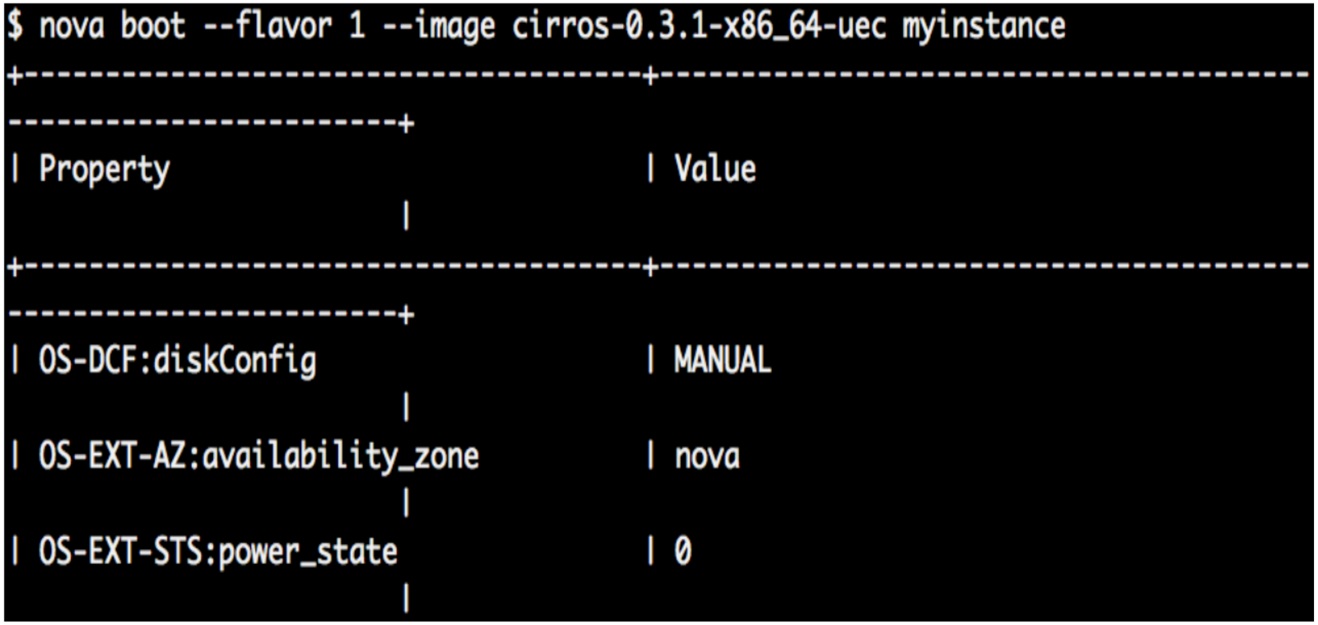
```
"""
v2 Neutron Plug-in API specification.

:class:`NeutronPluginBaseV2` provides the definition of minimum set of
methods that needs to be implemented by a v2 Neutron Plug-in.
"""

import abc
import six


@six.add_metaclass(abc.ABCMeta)
class NeutronPluginBaseV2(object):

    @abc.abstractmethod
    def create_subnet(self, context, subnet):
        """Create a subnet.

        Create a subnet, which represents a range of IP addresses
        that can be allocated to devices

        :param context: neutron api request context
        :param subnet: dictionary describing the subnet, with keys
                       as listed in the  :obj:`RESOURCE_ATTRIBUTE_MAP` object
                       in :file:`neutron/api/v2/attributes.py`.  All keys will
                       be populated.
        """
        pass

    @abc.abstractmethod
    def update_subnet(self, context, id, subnet):
        """Update values of a subnet.

        :param context: neutron api request context
        :param id: UUID representing the subnet to update.
        :param subnet: dictionary with keys indicating fields to update.
                       valid keys are those that have a value of True for
```

Plugin Extensions

- Add logical resources to the REST API
- Discovered by server at startup:
  - REST: /v2.0/extensions
- Common extensions:
  - Binding, DHCP, L3, Provider, Quota, Security Groups
- Other extensions:
  - Extra Routes, Metering

Booting a VM

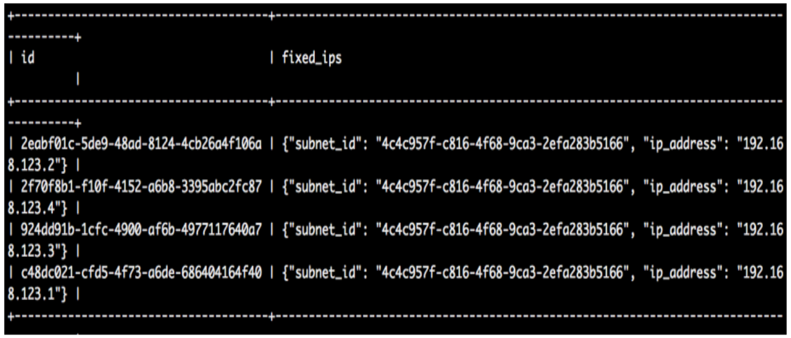
- nova boot



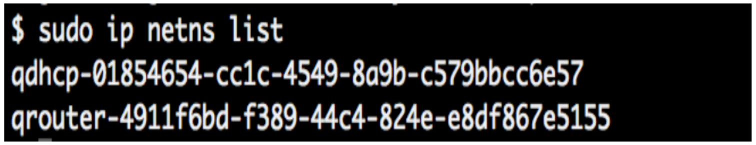
```
$ nova boot --flavor 1 --image cirros-0.3.1-x86_64-uec myinstance
+---------------------------------------+----------------------------------------
------------------------+
| Property                              | Value
                        |
+---------------------------------------+----------------------------------------
------------------------+
| OS-DCF:diskConfig                     | MANUAL
                        |
| OS-EXT-AZ:availability_zone           | nova
                        |
| OS-EXT-STS:power_state                | 0
                        |
```

Booting a VM

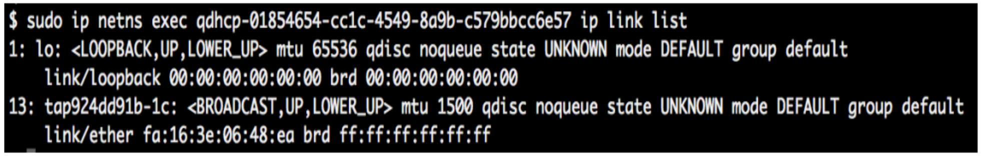
- nova boot
- create port
    - notify DHCP of new port



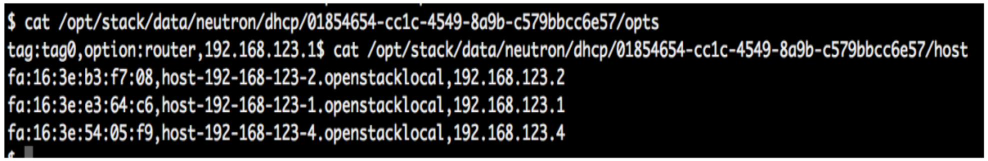
```
+--------------------------------------+----------------------------------------------------------------------------
----------+
| id                                   | fixed_ips
          |
+--------------------------------------+----------------------------------------------------------------------------
----------+
| 2eabf01c-5de9-48ad-8124-4cb26a4f106a | {"subnet_id": "4c4c957f-c816-4f68-9ca3-2efa283b5166", "ip_address": "192.16
8.123.2"} |
| 2f70f8b1-f10f-4152-a6b8-3395abc2fc87 | {"subnet_id": "4c4c957f-c816-4f68-9ca3-2efa283b5166", "ip_address": "192.16
8.123.4"} |
| 924dd91b-1cfc-4900-af6b-4977117640a7 | {"subnet_id": "4c4c957f-c816-4f68-9ca3-2efa283b5166", "ip_address": "192.16
8.123.3"} |
| c48dc021-cfd5-4f73-a6de-686404164f40 | {"subnet_id": "4c4c957f-c816-4f68-9ca3-2efa283b5166", "ip_address": "192.16
8.123.1"} |
+--------------------------------------+----------------------------------------------------------------------------
```



```
$ sudo ip netns list
qdhcp-01854654-cc1c-4549-8a9b-c579bbcc6e57
qrouter-4911f6bd-f389-44c4-824e-e8df867e5155
```



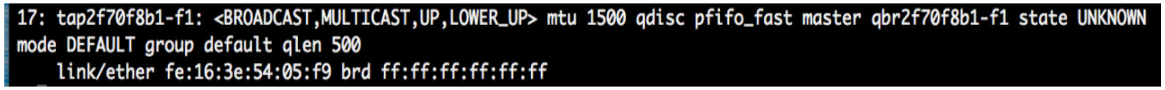
```
$ sudo ip netns exec qdhcp-01854654-cc1c-4549-8a9b-c579bbcc6e57 ip link list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
13: tap924dd91b-1c: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN mode DEFAULT group default
    link/ether fa:16:3e:06:48:ea brd ff:ff:ff:ff:ff:ff
```



```
$ cat /opt/stack/data/neutron/dhcp/01854654-cc1c-4549-8a9b-c579bbcc6e57/opts
tag:tag0,option:router,192.168.123.1$ cat /opt/stack/data/neutron/dhcp/01854654-cc1c-4549-8a9b-c579bbcc6e57/host
fa:16:3e:b3:f7:08,host-192-168-123-2.openstacklocal,192.168.123.2
fa:16:3e:e3:64:c6,host-192-168-123-1.openstacklocal,192.168.123.1
fa:16:3e:54:05:f9,host-192-168-123-4.openstacklocal,192.168.123.4
```

Booting a VM

- nova boot
- create port
  - notify DHCP of new port
- create device
- wire port
- boot



```
17: tap2f70f8b1-f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master qbr2f70f8b1-f1 state UNKNOWN
mode DEFAULT group default qlen 500
    link/ether fe:16:3e:54:05:f9 brd ff:ff:ff:ff:ff:ff
```

Summary

- Unified API
- Small Core
- Pluggable Open Architecture
    - Multiple Vendor Support
- Extensible

vmware
Open vSwitch / Linux Bridge
CISCO
Ryu OpenFlow Controller
NEC
OPEN DAYLIGHT
BROCADE
Mellanox TECHNOLOGIES
Windows Hyper-V
PLUMgrid
big switch networks
midokura
radware
tail-f
embrane
ARISTA
nuagenetworks
One Convergence
IBM

I’ve Got It!

Acknowledgements

- www.docs.openstack.org
- http://developer.openstack.org/api-ref-networking-v2.html
- http://docs.openstack.org/admin-guide-cloud/index.html
- http://www.slideshare.net/markmcclain/inside-architecture-of-neutron