Never Lost in Havana -multinode installation-

OpenStack Workshop
in WIDE Cloud WG

2014 WIDE Spring Camp Meeting @ Hamanako

Reference Equipments

- 4 Virtual nodes on VirtualBox
- 1 VirtualBox NAT segment
- 2 VirtualBox Host-only segments

Roles of Nodes

- Controller
  - nova/neutron/cinder controllers, keystone, glance, dashboard
- Network
  - External network gateway
- Storage
  - LVM based virtual volumes
- Compute
  - qemu virtualization function

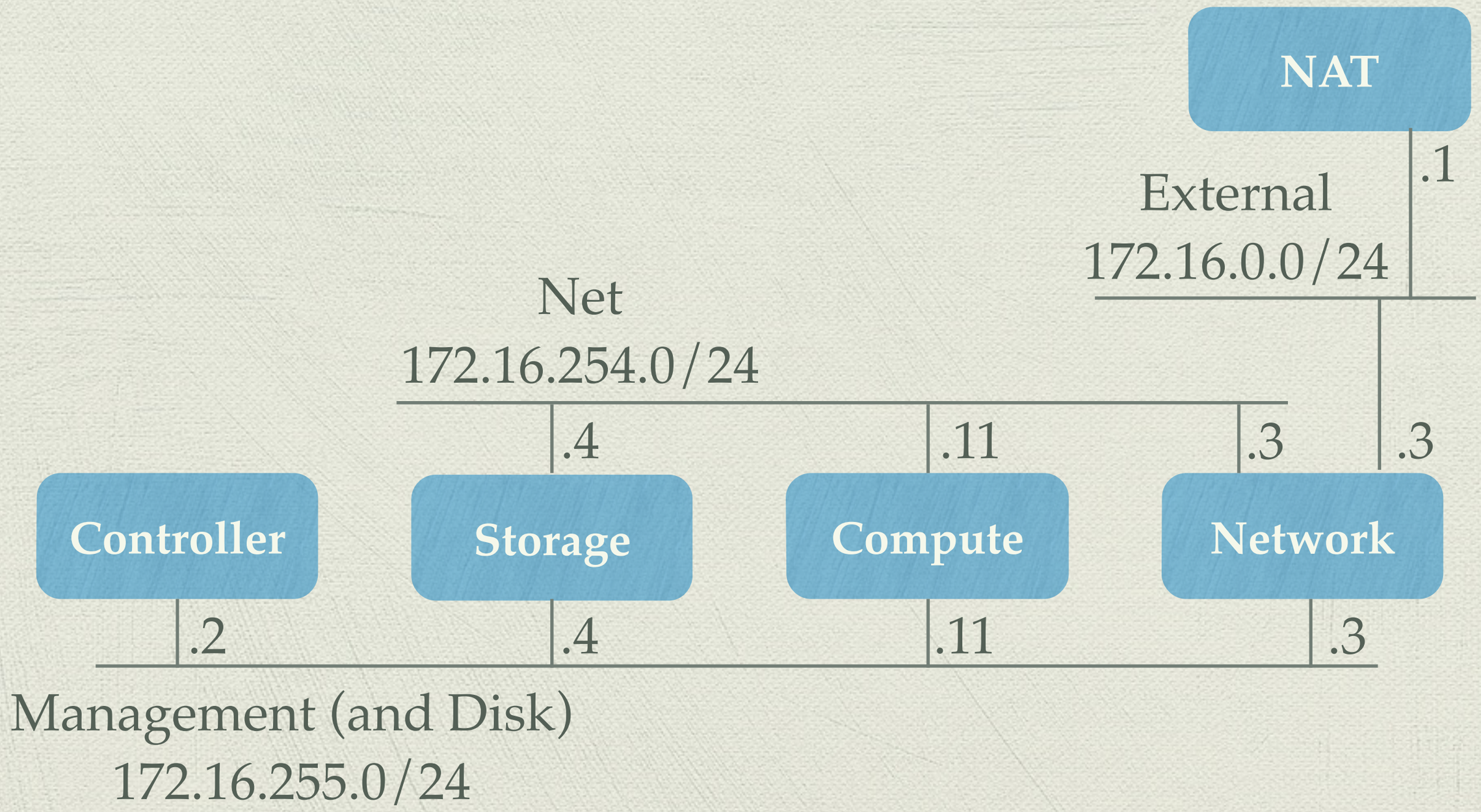
Special Requirements

- Storage
  - Secondary HDD with enough size
- Compute
  - Enough memory to run VMs

Address Specification

	External 172.16.0.0/24	Net 172.16.254.0/24	Management 172.16.255.0/24
Controller	2	-	-
Network	3	3	3
Storage	4	-	-
Compute1	11	11	-

Reference Topology



Original Document

- OpenStack Installation Guide for Ubuntu 12.04 (LTS)

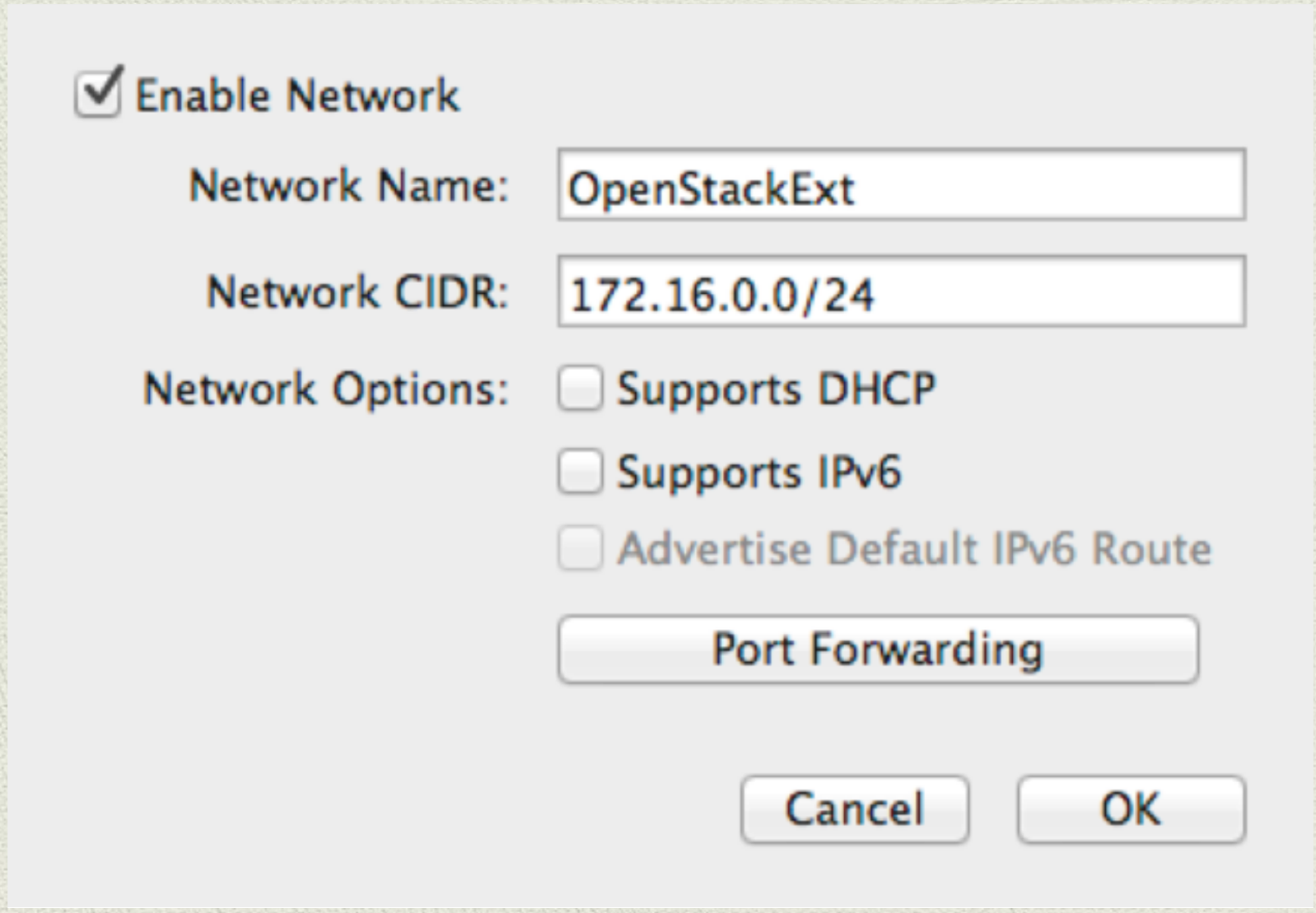
- http://docs.openstack.org/havana/install-guide/install/apt/content/

Network Preparation

- VirtualBox NAT
- Host-only Network x 2

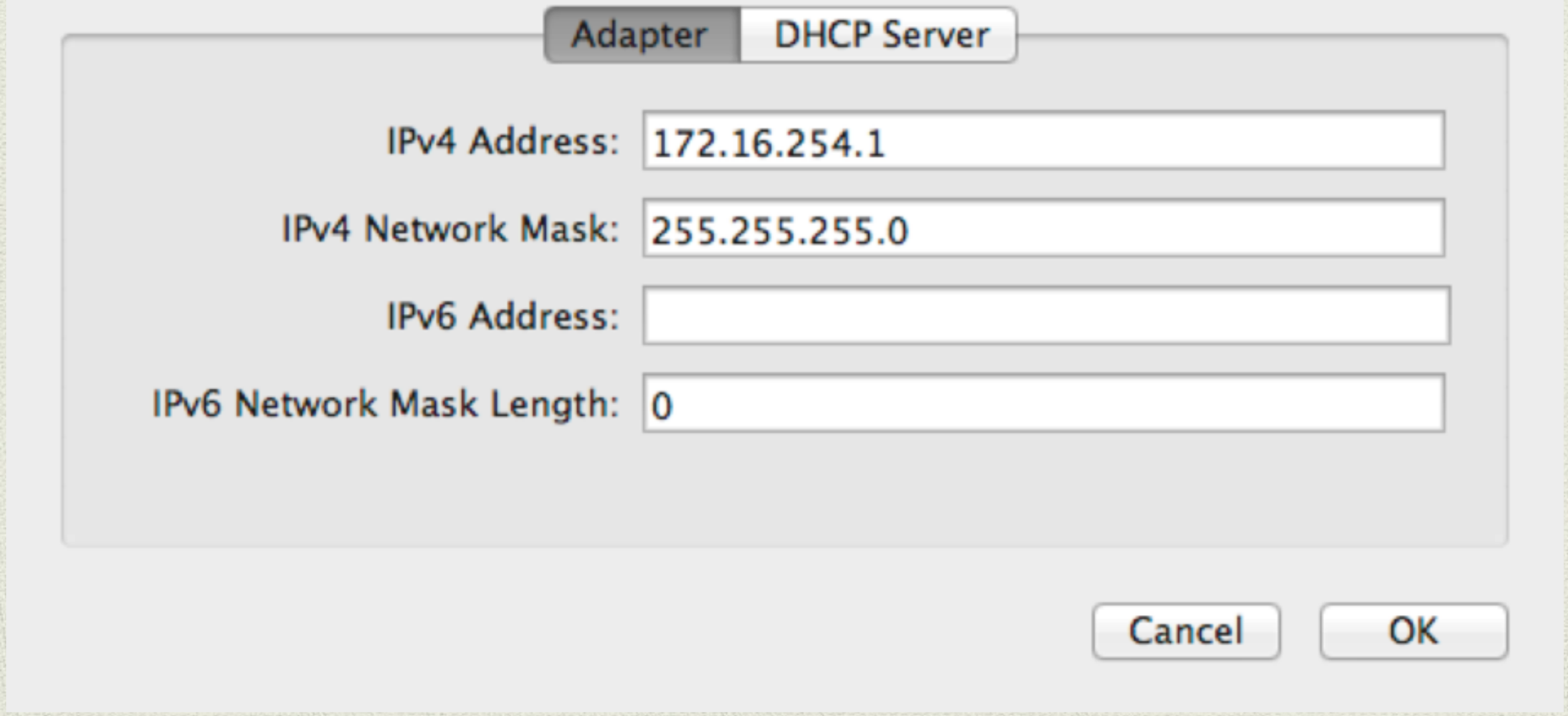
VirtualBox NAT (External)

OpenStackExt



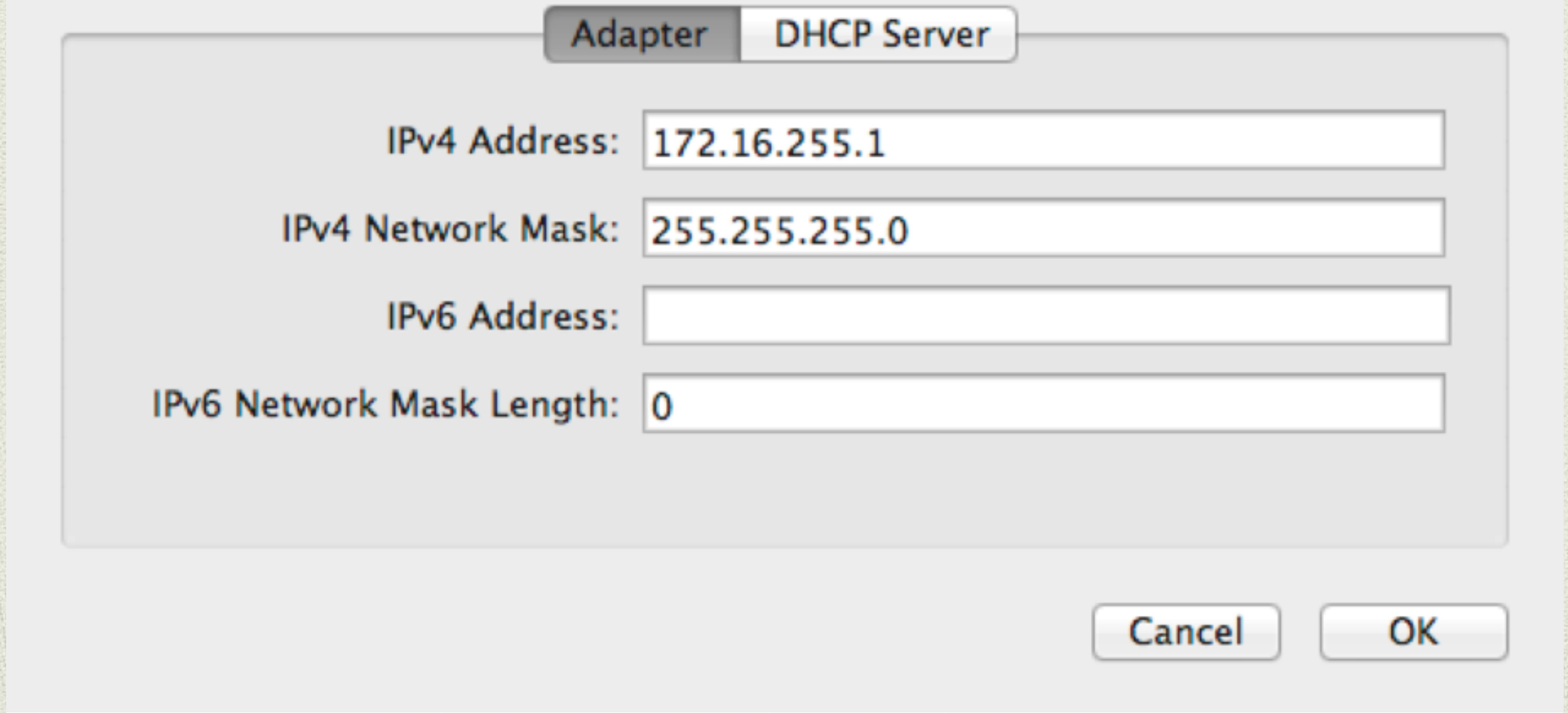
Host-only (Data)

vboxnet0



Host-only (Management)

vboxnet1



Create a Virtual Machine Template

- Name: template
- Memory: 1024MB
- HDD: 8GB
- Network (Promiscuous mode -> Allow all)
  - Adapter 1: OpenStackExt (172.16.0.0/24)
  - Adapter 2: vboxnet0 (172.16.254.0/24)
  - Adapter 3: vboxnet1 (172.16.255.0/24)
  - Adapter 4: NAT

Create a Virtual Machine Template

- Install Ubuntu 12.04 Server
  - Use Adapter 4 (eth3) for networking
  - Install OpenSSH server
- After installation
  - Copy your `authorized_keys` to `/root/.ssh/`
  - Setup Ubuntu Cloud Archive

```
# apt-get install python-software-properties
# add-apt-repository cloud-archive:havana
# apt-get update && apt-get dist-upgrade
# reboot
```

Create a Virtual Machine Template

- Remove `/etc/udev/rules.d/70-persistent-net.rules` before shutting down

Add `/etc/hosts` entries

- Add the following lines to `/etc/hosts`

```
172.16.255.2        controller
172.16.255.3        network
172.16.255.4        storage
172.16.255.11       compute1
```

Clone controller

- Update `/etc/hostname`
- Assign an IP address on eth2 (Management)

```
auto eth2
iface eth2 inet static
    address 172.16.255.2
    netmask 255.255.255.0
```

Clone network

- Update `/etc/hostname`
- Assign IP addresses on eth1 (Net), eth2 (Management)

```
auto eth1
iface eth1 inet static
   address 172.16.254.3
   netmask 255.255.255.0

auth eth2
iface eth2 inet static
   address 172.16.255.3
   netmask 255.255.255.0
```

Clone storage

- Add a secondary HDD with size e.g. 128GB
- Update `/etc/hostname`
- Assign IP addresses on eth2 (Management)

```
auth eth2
iface eth2 inet static
    address 172.16.255.4
    netmask 255.255.255.0
```

Clone compute1

- Increase memory to e.g. 2048MB
- Update `/etc/hostname`
- Assign IP addresses on eth1 (Net), eth2 (Management)

```
auto eth1
iface eth1 inet static
   address 172.16.254.11
   netmask 255.255.255.0

auth eth2
iface eth2 inet static
   address 172.16.255.11
   netmask 255.255.255.0
```

Reference IDs

	User	Password
MySQL	root	osmysql
RabbitMQ	guest	osrabbit
Keystone	keystone	oskeystone
Nova	nova	osnova
Glance	glance	osglance
Cinder	cinder	oscinder
Neutron	neutron	osneutron
Neutron Metadata	-	osneutronmeta

Backend Database (controller)

- Install MySQL server and Python client library on controller

```
# apt-get install python-mysqldb mysql-server
```

- Update `/etc/msql/my.cnf`

```
bind-address            172.16.255.2
```

- Restart MySql

```
# service mysql restart
```

Backend Database (controller)

- Secure MySQL

```
# mysql_secure_installation
```

Backend Database (network, storage, compute1)

- Install Python client library on network, storage, and compute1

```
# apt-get install python-mysqldb
```

Message Queue (controller)

- Install RabbitMQ on controller

```
# apt-get install rabbitmq-server
```

- Change password

```
# rabbitmqctl change_password guest osrabbit
```

Keystone (controller)

- Install keystone

```
# apt-get install keystone
```

- Update database location in `/etc/keystone/keystone.conf`

```
connection = mysql://keystone:oskeystone@controller/keystone
```

- Update `admin_token` in `/etc/keystone/keystone.conf`

```
admin_token    = ostoken
```

Keystone (controller)

- Create keystone database

```
# mysql -u root -p
mysql> CREATE DATABASE keystone;
mysql> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' \
IDENTIFIED BY 'oskeystone';
mysql> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'172.16.255.%' \
IDENTIFIED BY 'oskeystone';
```

```
# keystone-manage db_sync
```

- Restart keystone

```
# service keystone restart
```

Keystone (controller)

- Setup ENV for initial setup

```
# export OS_SERVICE_TOKEN=ostoken
# export OS_SERVICE_ENDPOINT=http://controller:35357/v2.0
```

- Create tenants

```
# keystone tenant-create --name=admin --description="Admin Tenant"
# keystone tenant-create --name=service --description="Service Tenant"
```

Keystone (controller)

- Create admin user

```
# keystone user-create --name=admin --pass=admin \
   --email=admin@example.com
```

- Create an admin role and add it to admin

```
# keystone role-create --name=admin
```

```
# keystone user-role-add --user=admin --tenant=admin \
 --role=admin
```

Keystone (controller)

- Create a service entry

```
# keystone service-create --name=keystone --type=identity \
  --description="Keystone Identity Service"
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description | Keystone Identity Service        |
| id          | 15c11a23667e427e91bc31335b45f4bd |
| name        | keystone                         |
| type        | identity                         |
+-------------+----------------------------------+
```

- Create a service endpoint

```
# keystone endpoint-create \
  --service-id=15c11a23667e427e91bc31335b45f4bd \
  --publicurl=http://controller:5000/v2.0 \
  --internalurl=http://controller:5000/v2.0 \
  --adminurl=http://controller:35357/v2.0
```

Keystone (controller)

- Clear ENV for initial setup

```
# unset OS_SERVICE_TOKEN OS_SERVICE_ENDPOINT
```

- Create a rc file at `/root/keystonerc`

```
export OS_USERNAME=admin
export OS_PASSWORD=admin
export OS_TENANT_NAME=admin
export OS_AUTH_URL=http://controller:35357/v2.0
```

- Now you can issue keystone commands

```
# source /root/keystonerc
# keystone user-list
```

Glance (controller)

- Install glance

```
# apt-get install glance python-glanceclient
```

- Update database location in `/etc/glance/glance-api.conf` and `/etc/glance-registry.conf`

```
sql_connection = mysql://glance:osglance@controller/glance
```

Glance (controller)

- Create glance database

```
# mysql -u root -p
mysql> CREATE DATABASE glance;
mysql> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' \
IDENTIFIED BY 'osglance';
mysql> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'172.16.255.%' \
IDENTIFIED BY 'osglance';
```

```
# glance-manage db_sync
```

- Restart glance

```
# service glance-api restart
# service glance-registry restart
```

Glance (controller)

- Create glance user

```
# keystone user-create --name=glance --pass=osglance \
  --email=glance@example.com
```

```
# keystone user-role-add --user=glance --tenant=service \
 --role=admin
```

Glance (controller)

- Setup service authentication in `/etc/glance/glance-api.conf` and `/etc/glance-registry.conf`

```
[keystone_authtoken]
...
auth_uri = http://controller:5000
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = osglance
```

```
[paste_deploy]
...
flavor = keystone
```

Glance (controller)

- Setup service authentication in `/etc/glance/glance-api-paste.init` and `/etc/glance-registry-paste.ini`

```
[filter:authtoken]
...
paste.filter_factory=keystoneclient.middleware.auth_token:filter_factory
auth_host=controller
admin_user=glance
admin_tenant_name=service
admin_password=osglance
```

Glance (controller)

- Create a service entry and a service endpoint

```
# keystone service-create --name=glance --type=image \
  --description="Glance Image Service"
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |       Glance Image Service       |
|      id     | 4d66a66b8c14434f8c39251a7f68da81 |
|     name    |              glance              |
|     type    |              image               |
+-------------+----------------------------------+
```

```
# keystone endpoint-create \
  --service-id=4d66a66b8c14434f8c39251a7f68da81 \
  --publicurl=http://controller:9292 \
  --internalurl=http://controller:9292 \
  --adminurl=http://controller:9292
```

Glance (controller)

- Configure RabbitMQ in `/etc/glance/glance-api.conf`

```
rabbit_host = controller
rabbit_password = osrabbit
```

- Restart glance

```
# service glance-registry restart
# service glance-api restart
```

Compute (controller)

- Install compute controller component on controller

```
# apt-get install nova-novncproxy novnc nova-api \
  nova-ajax-console-proxy nova-cert nova-conductor \
  nova-consoleauth nova-doc nova-scheduler \
  python-novaclient
```

- Add [database] and [keystone_authtoken] sections in `/etc/nova/nova.conf`

```
[database]
connection = mysql://nova:osnova@controller/nova
[keystone_authtoken]
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = osnova
```

Compute (controller)

- Configure RabbitMQ in [DEFAULT] section of `/etc/nova/nova.conf`

```
rpc_backend = nova.rpc.impl_kombu
rabbit_host = controller
rabbit_password = osrabbit
```

Compute (controller)

- Create nova user

```
# keystone user-create --name=nova --pass=osnova \
  --email=nova@example.com
```

```
# keystone user-role-add --user=nova --tenant=service \
 --role=admin
```

Compute (controller)

- Create nova database

```
# mysql -u root -p
mysql> CREATE DATABASE nova;
mysql> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' \
IDENTIFIED BY 'osnova';
mysql> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'172.16.255.%' \
IDENTIFIED BY 'osnova';
```

```
# nova-manage db sync
```

Compute (controller)

- Configure VNC in `/etc/nova/nova.conf`

```
[DEFAULT]
...
my_ip=172.16.255.2
vncserver_listen=172.16.255.2
vncserver_proxyclient_address=172.16.255.2
```

- Configure auth strategy

```
[DEFAULT]
...
auth_strategy=keystone
```

Compute (controller)

- Configure authtoken in `/etc/nova/api-paste.ini`

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = controller
auth_port = 35357
auth_protocol = http
auth_uri = http://controller:5000/v2.0
admin_tenant_name = service
admin_user = nova
admin_password = osnova
```

Compute (controller)

- Create a service entry and a service endpoint

```
# keystone service-create --name=nova --type=compute \
  --description="Nova Compute service"
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |       Nova Compute service      |
|      id     | 0adcc6692e974d849994823503b1fa5f |
|     name    |               nova               |
|     type    |              compute             |
+-------------+----------------------------------+
```

```
# keystone endpoint-create \
  --service-id=0adcc6692e974d849994823503b1fa5f \
  --publicurl=http://controller:8774/v2/%\(tenant_id\)s \
  --internalurl=http://controller:8774/v2/%\(tenant_id\)s \
  --adminurl=http://controller:8774/v2/%\(tenant_id\)s
```

Compute (controller)

- Restart nova

```
# service nova-api restart
# service nova-cert restart
# service nova-consoleauth restart
# service nova-scheduler restart
# service nova-conductor restart
# service nova-novncproxy restart
```

Compute (compute1)

- Install compute components on compute1
  - When using VirtualBox (no nested VT support)

```
# apt-get install nova-compute-qemu python-guestfs
```

  - When a nested VT function is supported

```
# apt-get install nova-compute-kvm python-guestfs
```

- When asked whether to build supermin kernel, say yes

Compute (compute1)

- Kernel image permission fix

```
# dpkg-statoverride  --update --add root root 0644 \
    /boot/vmlinuz-$(uname -r)
```

- Create a permission fix script at `/etc/kernel/postinst.d/stateoverride` with the following lines

```
#!/bin/sh
version="$1"
# passing the kernel version is required
[ -z "${version}" ] && exit 0
dpkg-statoverride --update --add root root 0644 /boot/vmlinuz-${version}
```

- Make it executable

```
# chmod +x /etc/kernel/postinst.d/stateoverride
```

Compute (compute1)

- Update database location and RabbitMQ configuration in `/etc/nova/nova.conf`

```
[DEFAULT]
...
auth_strategy=keystone
[database]
connection=mysql://nova:osnova@controller/nova
```

```
[DEFAULT]
...
rpc_backend = nova.rpc.impl_kombu
rabbit_host = controller
rabbit_password = osrabbit
```

Compute (compute1)

- Configure VNC in `/etc/nova/nova.conf`

```
[DEFAULT]
...
my_ip=172.16.255.11
vncserver_listen=0.0.0.0
vncserver_proxyclient_address=172.16.255.11
novncproxy_base_url=http://172.16.255.2:6080/vnc_auto.html
```

- Configure glance host

```
[DEFAULT]
...
glance_host=controller
```

Compute (compute1)

- Configure authtoken in `/etc/nova/api-paste.ini`

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = osnova
```

- Restart nova-compute

```
# service nova-compute restart
```

Dashboard (controller)

- Install dashboard related components

```
# apt-get install memcached libapache2-mod-wsgi openstack-dashboard
```

- Uninstall Ubuntu specific unnecessary component

```
# apt-get remove --purge openstack-dashboard-ubuntu-theme
```

Dashboard (controller)

- Update `/etc/openstack-dashboard/local_settings.py`

```
OPENSTACK_HOST = "controller"
```

```
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "_member_"
```

```
TIME_ZONE = "Asia/Tokyo"
```

Dashboard (controller)

- Restart apache2 and memcached

```
# service apache2 restart
# service memcached restart
```

Cinder (controller)

- Install cinder controller components on controller

```
# apt-get install cinder-api cinder-scheduler
```

- Update database location in `/etc/cinder/cinder.conf`

```
[database]
connection=mysql://cinder:oscinder@controller/cinder
```

Cinder (controller)

- Configure RabbitMQ in `/etc/cinder/cinder.conf`

```
[DEFAULT]
...
rpc_backend = cinder.openstack.common.rpc.impl_kombu
rabbit_host = controller
rabbit_port = 5672
rabbit_password = osrabbit
```

Cinder (controller)

- Specify volume API in `/etc/nova/nova.conf`

```
[DEFAULT]
...
volume_api_class=nova.volume.cinder.API
```

Cinder (controller)

- Create a cinder database

```
# mysql -u root -p
mysql> CREATE DATABASE cinder;
mysql> GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'localhost' \
IDENTIFIED BY 'oscinder';
mysql> GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'172.16.255.%' \
IDENTIFIED BY 'oscinder';
```

```
# cinder-manage db sync
```

Cinder (controller)

- Create a cinder user

```
# keystone user-create --name=cinder --pass=oscinder \
    --email=cinder@example.com
# keystone user-role-add --user=cinder --tenant=service \
    --role=admin
```

Cinder (controller)

- Configure authtoken in `/etc/cinder/api-paste.ini`

```
[filter:authtoken]
paste.filter_factory=keystoneclient.middleware.auth_token:filter_factory
auth_host=controller
auth_port = 35357
auth_protocol = http
admin_tenant_name=service
admin_user=cinder
admin_password=oscinder
```

Cinder (controller)

- Create a service entry and a service endpoint

```
# keystone service-create --name=cinder --type=volume \
  --description="Cinder Volume Service"
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |      Cinder Volume Service       |
|      id     | 7960ee6feffb4ff2a9ce508287b5bcff |
|     name    |              cinder              |
|     type    |              volume              |
+-------------+----------------------------------+
```

```
# keystone endpoint-create \
  --service-id=7960ee6feffb4ff2a9ce508287b5bcff \
  --publicurl=http://controller:8776/v1/%\(tenant_id\)s \
  --internalurl=http://controller:8776/v1/%\(tenant_id\)s \
  --adminurl=http://controller:8776/v1/%\(tenant_id\)s
```

Cinder (controller)

- Create another service entry and endpoint

```
# keystone service-create --name=cinderv2 --type=volumev2 \
  --description="Cinder Volume Service V2"
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |     Cinder Volume Service V2     |
|      id     | b406b5bb23c3480290e62529e2781dca |
|     name    |             cinderv2             |
|     type    |             volumev2             |
+-------------+----------------------------------+
```

```
# keystone endpoint-create \
  --service-id=b406b5bb23c3480290e62529e2781dca \
  --publicurl=http://controller:8776/v2/%\(tenant_id\)s \
  --internalurl=http://controller:8776/v2/%\(tenant_id\)s \
  --adminurl=http://controller:8776/v2/%\(tenant_id\)s
```

Cinder (controller)

- Restart cinder

```
# service cinder-scheduler restart
# service cinder-api restart
```

Cinder (storage)

- Install LVM and create a cinder volume

```
# apt-get install lvm2
# pvcreate /dev/sdb
# vgcreate cinder-volumes /dev/sdb
```

- Update LVM scan filter in /etc/lvm/lvm.conf

```
devices {
...
filter = [ "a/sda5/", "a/sdb/", "r/.*/"]
...
}
```

Cinder (storage)

- Install cinder-volume

```
# apt-get install cinder-volume qemu-utils
```

- Configure RabbitMQ in `/etc/cinder/cinder.conf`

```
[DEFAULT]
...
my_ip = 172.16.255.4
rpc_backend = cinder.openstack.common.rpc.impl_kombu
rabbit_host = controller
rabbit_port = 5672
rabbit_password = osrabbit
glance_host = controller
```

Cinder (storage)

- Update database location in `/etc/cinder/cinder.conf`

```
[database]
connection = mysql://cinder:oscinder@controller/cinder
```

- Configure authtoken in /etc/cinder/api-paste.ini

```
[filter:authtoken]
paste.filter_factory=keystoneclient.middleware.auth_token:filter_factory
auth_host=controller
auth_port = 35357
auth_protocol = http
admin_tenant_name=service
admin_user=cinder
admin_password=oscinder
```

Cinder (storage)

- Restart cinder-volume

```
# service cinder-volume restart
# service tgt restart
```

Neutron (controller)

- Create a neutron database

```
# mysql -u root -p
mysql> CREATE DATABASE neutron;
mysql> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' \
IDENTIFIED BY 'osneutron';
mysql> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'172.16.255.%' \
IDENTIFIED BY 'osneutron';
```

Neutron (controller)

- Create a neutron user

```
# keystone user-create --name=neutron --pass=osneutron \
  --email=neutron@example.com
# keystone user-role-add --user=neutron --tenant=service --role=admin
```

Neutron (controller)

- Create a service entry and a service endpoint

```
# keystone service-create --name=neutron --type=network \
     --description="OpenStack Networking Service"
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |   OpenStack Networking Service   |
|      id     | 0aa067dad34247f0a2ec6e1627922c58 |
|     name    |             neutron              |
|     type    |             network              |
+-------------+----------------------------------+
```

```
# keystone endpoint-create \
     --service-id 0aa067dad34247f0a2ec6e1627922c58 \
     --publicurl http://controller:9696 \
     --adminurl http://controller:9696 \
     --internalurl http://controller:9696
```

Neutron (controller)

- Install neutron controller components on controller

```
# apt-get install neutron-server
```

Neutron (controller)

- Update database location in `/etc/neutron/neutron.conf`

```
[database]
connection = mysql://neutron:osneutron@controller/neutron
```

Neutron (controller)

- Update authentication information in `/etc/neutron/neutron.conf`

```
[DEFAULT]
...
auth_strategy = keystone
...
rpc_backend = neutron.openstack.common.rpc.impl_kombu
rabbit_host = controller
rabbit_port = 5672
rabbit_password = osrabbit
...
allow_overlapping_ips = True
...
[keystone_authtoken]
auth_host = controller
admin_tenant_name = service
admin_user = neutron
admin_password = osneutron
auth_url = http://controller:35357/v2.0
...
```

Neutron (controller)

- Specify OpenvSwitch plugin in `/etc/neutron/neutron.conf`

```
core_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
```

Neutron (controller)

- Update authtoken in `/etc/neutron/api-paste.ini`

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
admin_tenant_name = service
admin_user = neutron
admin_password = osneutron
```

Neutron (controller)

- Configure nova in `/etc/nova/nova.conf`

```
[DEFAULT]
...
network_api_class=nova.network.neutronv2.api.API
neutron_url=http://controller:9696
neutron_auth_strategy=keystone
neutron_admin_tenant_name=service
neutron_admin_username=neutron
neutron_admin_password=osneutron
neutron_admin_auth_url=http://controller:35357/v2.0
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSInterfaceDriver
firewall_driver=nova.virt.firewall.NoopFirewallDriver
security_group_api=neutron
neutron_metadata_proxy_shared_secret=osneutronmeta
service_neutron_metadata_proxy=true
```

Neutron (controller)

- Install OVS plugin

```
# apt-get install neutron-plugin-openvswitch
```

Neutron (controller)

- Configure OVS plugin information in `/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini`

```
tenant_network_type = gre
```

```
enable_tunneling = True
```

```
tunnel_id_ranges = 1:1000
```

```
[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewal
lDriver
```

Neutron (controller)

- Restart neutron service

```
# service nova-api restart
# service nova-scheduler restart
# service nova-conductor restart
# service neutron-server restart
```

Neutron (network)

- Install neutron components on network

```
# apt-get install neutron-server neutron-dhcp-agent \
  neutron-plugin-openvswitch-agent neutron-l3-agent
```

- Enable forwarding and disable reverse path filter in `/etc/sysctl.conf`

```
net.ipv4.ip_forward=1
net.ipv4.conf.all.rp_filter=0
net.ipv4.conf.default.rp_filter=0
```

- Reload `/etc/sysctl.conf`

```
# sysctl -p
```

Neutron (network)

- Update database location in `/etc/neutron/neutron.conf`

```
[database]
connection = mysql://neutron:osneutron@controller/neutron
```

Neutron (network)

- Configure authentication in `/etc/neutron/neutron.conf`

```
[DEFAULT]
...
auth_strategy = keystone
...
rpc_backend = neutron.openstack.common.rpc.impl_kombu
rabbit_host = controller
rabbit_port = 5672
rabbit_password = osrabbit
...
[keystone_authtoken]
auth_host = controller
auth_protocol = http
admin_tenant_name = service
admin_user = neutron
admin_password = osneutron
auth_url = http://controller:35357/v2.0
...
```

Neutron (network)

- Update authtoken in `/etc/neutron/api-paste.ini`

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host=controller
auth_uri=http://controller:5000
admin_tenant_name=service
admin_user=neutron
admin_password=osneutron
```

Neutron (network)

- Configure DHCP driver in `/etc/neutron/dhcp_agent.ini`

```
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
```

- Configure metadata agent in `/etc/neutron/metadata_agent.ini`

```
[DEFAULT]
auth_url=http://controller:5000/v2.0
auth_region=regionOne
admin_tenant_name=service
admin_user=neutron
admin_password=osneutron
...
nova_metadata_ip=controller
...
metadata_proxy_shared_secret=osneutronmeta
```

Neutron (network)

- Install OVS plugin

```
# apt-get install neutron-plugin-openvswitch-agent \
  openvswitch-datapath-dkms
```

Neutron (network)

- Create OVS interfaces

```
# ovs-vsctl add-br br-int
# ovs-vsctl add-br br-ex
```

- Add the external interface (eth0) to the OVS external interface (br-ex)

```
# ovs-vsctl add-port br-ex eth0
```

Neutron (network)

- Configure external interfaces in `/etc/network/interfaces`

```
auto eth0
iface eth0 inet manual
    up /sbin/ifconfig eth0 up

auto br-ex
iface br-ex inet static
    address 172.16.0.3
    netmask 255.255.255.0
    gateway 172.16.0.1
    dns-nameservers 172.16.0.1
```

Neutron (network)

- Configure network driver in `/etc/neutron/l3_agent.ini` and `/etc/neutron/dhcp_agent.ini`

```
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
...
use_namespaces = True
```

- Configure plugin in `/etc/neutron/neutron.conf`

```
core_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
```

Neutron (network)

- Configure OVS plugin in `/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini`

```
[ovs]
tenant_network_type = gre
tunnel_id_ranges = 1:1000
enable_tunneling = True
integration_bridge = br-int
tunnel_bridge = br-tun
local_ip = 172.16.254.3

[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
```

Neutron (network)

- Restart neutron service

```
# service neutron-server restart
# service neutron-dhcp-agent restart
# service neutron-l3-agent restart
# service neutron-metadata-agent restart
```

Neutron (compute1)

- Disable reverse path filter in `/etc/sysctl.conf`

```
net.ipv4.conf.all.rp_filter=0
net.ipv4.conf.default.rp_filter=0
```

```
# sysctl -p
```

Neutron (compute1)

- Install neutron OVS plugin

```
# apt-get install neutron-plugin-openvswitch-agent \
  openvswitch-datapath-dkms
```

- Create OVS interfaces

```
# ovs-vsctl add-br br-int
```

Neutron (compute1)

- Configure plugin in `/etc/neutron/neutron.conf`

```
core_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
```

Neutron (compute1)

- Configure OVS plugin in `/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini`

```
[ovs]
tenant_network_type = gre
tunnel_id_ranges = 1:1000
enable_tunneling = True
integration_bridge = br-int
tunnel_bridge = br-tun
local_ip = 172.16.254.11

[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
```

Neutron (compute1)

- Update database location in `/etc/neutron/neutron.conf`

```
[database]
connection = mysql://neutron:osneutron@controller/neutron
```

Neutron (compute1)

- Configure authentication in `/etc/neutron/neutron.conf`

```
[DEFAULT]
...
auth_strategy = keystone
...
rpc_backend = neutron.openstack.common.rpc.impl_kombu
rabbit_host = controller
rabbit_port = 5672
rabbit_password = osrabbit
...
[keystone_authtoken]
auth_host = controller
admin_tenant_name = service
admin_user = neutron
admin_password = osneutron
auth_url = http://controller:35357/v2.0
...
```

Neutron (compute1)

- Update authtoken in `/etc/neutron/api-paste.ini`

```
[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = controller
auth_uri = http://controller:5000
admin_tenant_name = service
admin_user = neutron
admin_password = osneutron
```

Neutron (compute1)

- Configure nova in `/etc/nova/nova.conf`

```
[DEFAULT]
...
network_api_class=nova.network.neutronv2.api.API
neutron_url=http://controller:9696
neutron_auth_strategy=keystone
neutron_admin_tenant_name=service
neutron_admin_username=neutron
neutron_admin_password=osneutron
neutron_admin_auth_url=http://controller:35357/v2.0
linuxnet_interface_driver = nova.network.linux_net.LinuxOVSInterfaceDriver
firewall_driver=nova.virt.firewall.NoopFirewallDriver
security_group_api=neutron
```

Disable NAT adapter

- Disable Adapter 4 (NAT) in all nodes