Keystone

Prerequisites:

1. Use the database access client to connect to the database server as the `root` user:

   ```
   # mysql
   ```

2. Create the `keystone` database:

   ```
   MariaDB [(none)]> CREATE DATABASE keystone;
   ```

3. Grant proper access to the `keystone` database:

   ```
   MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO
   'keystone'@'localhost' \
   IDENTIFIED BY 'KEYSTONE_DBPASS';
   MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' \
   IDENTIFIED BY 'KEYSTONE_DBPASS';
   ```

   Replace `KEYSTONE_DBPASS` with a suitable password.

4. Exit the database access client.

Install and configure components:

1. Run the following command to install the packages:

```
# apt install keystone
```

2. Populate the Identity service database:

```
# su -s /bin/sh -c "keystone-manage db_sync" keystone
```

3. Initialize Fernet key repositories:

```
# keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
```

```
# keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
```

4. Bootstrap the Identity service:

```
# keystone-manage bootstrap --bootstrap-password ADMIN_PASS \
--bootstrap-admin-url http://controller:5000/v3/ \
--bootstrap-internal-url http://controller:5000/v3/ \
--bootstrap-public-url http://controller:5000/v3/ \
 --bootstrap-region-id RegionOne
```

Replace `ADMIN_PASS` with a suitable password for an administrative user.

Configure the Apache HTTP server

1. Edit the `/etc/apache2/apache2.conf` file and configure the `ServerName` option to reference the controller node:

   ```
   ServerName controller
   ```

   The `ServerName` entry will need to be added if it does not already exist.

SSL

A secure deployment should have the web server configured to use SSL or running behind an SSL terminator.

Finalize the installation

1. Restart the Apache service:

   ```
   # service apache2 restart
   ```

2. Configure the administrative account by setting the proper environmental variables:

   ```
   $ export OS_USERNAME=admin
   $ export OS_PASSWORD=ADMIN_PASS
   $ export OS_PROJECT_NAME=admin
   $ export OS_USER_DOMAIN_NAME=Default
   $ export OS_PROJECT_DOMAIN_NAME=Default
   $ export OS_AUTH_URL=http://controller:5000/v3
   $ export OS_IDENTITY_API_VERSION=3
   ```

   These values shown here are the default ones created from `keystone-manage bootstrap`.

   Replace `ADMIN_PASS` with the password used in the `keystone-manage bootstrap` command in keystone-install-configure-ubuntu.

NOVA

Prerequisites:

1. Create the databases

2. Create the nova_api, nova, and nova_cell0 databases

3. Grant proper access to the databases

4. Source the `admin` credentials to gain access to admin-only CLI commands:

```
$ . admin-openrc
```

5. Create the Compute service credentials:

- Create the `nova` user:

```
$ openstack user create --domain default --password-prompt nova
```

6. Add the admin role to the nova user:

```
$ openstack role add --project service --user nova admin
```

7. Create the nova service entity:

```
$ openstack service create --name nova
```

8. reate the Compute API service endpoints:

```
$ openstack endpoint create --region RegionOne
```

Install and configure components - Controller:

1. Install the packages:

```
# apt install nova-api nova-conductor nova-novncproxy nova-scheduler
```

2. Edit the `/etc/nova/nova.conf` file

3. Populate the `nova-api` database:

```
# su -s /bin/sh -c "nova-manage api_db sync" nova
```

4. Register the `cell0` database:

```
      # su -s /bin/sh -c "nova-manage cell_v2 map_cell0" nova
```

5. Create the cell1 cell:

```
      # su -s /bin/sh -c "nova-manage cell_v2 create_cell --name=cell1 --
verbose" nova
```

6. Populate the nova database:

```
      # su -s /bin/sh -c "nova-manage db sync" nova
```

7. Verify nova cell0 and cell1 are registered correctly:

```
      # su -s /bin/sh -c "nova-manage cell_v2 list_cells" nova
```

Install and configure a compute node:

1. Install the packages:

```
      # apt install nova-compute
```

2. Edit the /etc/nova/nova.conf file

3. Determine whether your compute node supports hardware acceleration for virtual machines:

```
      $ egrep -c '(vmx|svm)' /proc/cpuinfo
```

4. Restart the Compute service:

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

Add the compute node to the cell database

Neutron

Controller node

Configure network interfaces

Configure name resolution

1. Set the hostname of the node to `controller`

2. Edit the `/etc/hosts` file

Compute node

Configure network interfaces

Configure name resolution

1. Set the hostname of the node to `compute1`.
2. Edit the `/etc/hosts` file

Verify connectivity

```
# ping -c 4 openstack.org
```

Prerequisites

1. To create the database

2. Source the `admin` credentials to gain access to admin-only CLI commands:

```
$ . admin-openrc
```

3. To create the service credentials, complete these steps:

- Create the `neutron` user:

```
$ openstack user create --domain default --password-prompt neutron
```

. Add the `admin` role to the `neutron` user:

```
$ openstack role add --project service --user neutron admin
```

- Create the `neutron` service entity:

  ```
  $ openstack service create --name neutron
  ```

4. Create the Networking service API endpoints:

```
$ openstack endpoint create --region RegionOne
```

Configure the Compute service to use the Networking service

Finalize installation and Populate the database

Glance

Prerequisites

1.To create the database

2. Source the `admin` credentials to gain access to admin-only CLI commands:

```
$ . admin-openrc
```

3. To create the service credentials, complete these steps:

- Create the `glance` user:

```
$ openstack user create --domain default --password-prompt glance
```

- Add the `admin` role to the `glance` user and `service` project:

```
$ openstack role add --project service --user glance
admin
```

- Create the `glance` service entity:

```
$ openstack service create --name glance \
```

4. Create the Image service API endpoints:

```
$ openstack endpoint create --region RegionOne
```

Install and configure components

1. Install the packages:

```
# apt install glance
```

```
2. Edit the /etc/glance/glance-api.conf file
```

3. Populate the Image service database:

```
# su -s /bin/sh -c "glance-manage db_sync" glance
```

Finalize installation

1. Restart the Image services:

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

Horizon

Install and configure for Ubuntu

1. Install the packages:

```
# apt install openstack-dashboard
```

2. Edit the `/etc/openstack-dashboard/local_settings.py` file and complete the following actions:

- Configure the dashboard to use OpenStack services on the `controller` node:

```
OPENSTACK_HOST = "controller"
```

- In the Dashboard configuration section, allow your hosts to access Dashboard:

```
ALLOWED_HOSTS = ['one.example.com', 'two.example.com']
```

information.

- Configure the `memcached` session storage service

- Enable the Identity API version 3:

```
OPENSTACK_KEYSTONE_URL = "http://%s/identity/v3" % OPENSTACK_HOST
```

- Enable support for domains:

```
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
```

- Configure API versions:

```
OPENSTACK_API_VERSIONS = {
    "identity": 3,
    "image": 2,
    "volume": 3,
}
```

- Configure `Default` as the default domain for users that you create via the dashboard:

```
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
```

- Configure `user` as the default role for users that you create via the dashboard:

```
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"
```

`Optionally, configure the time zone:`

```
TIME_ZONE = "TIME_ZONE"
```

3. Add the following line to `/etc/apache2/conf-available/openstack-dashboard.conf` if not included.

```
$ WSGIApplicationGroup %{GLOBAL}
```

Finalize installation

- Reload the web server configuration:

  ```
  # systemctl reload apache2.service
  ```