**High-Level Setup Overview**

**Requirements**

* OS: Ubuntu 22.04 LTS (Jammy)
* Virtualization: Vmware Workstation
* Deployment Tool: Kolla-Ansible
* Services:
  + **Compute**: Nova
  + **Storage**: Swift
  + **Database as a Service**: Trove
* Controller: 3 Server
* Compute : 3 Server
* Storage: Ceph Storage

Due to the limited resources I have created 3 VM machines on Vmware workstation and attached 3 extra drives on them. I have used these 3 VM machines for Controller, Compute and Ceph Storage Cluster.

**Project Directory Structure**

openstack-ansible-deployment/

├── inventory/

│ └── multinode

│ └── inventory.ini

├── playbooks/

│ ├── 01-prerequisites.yml

│ ├── 02-install-kolla-ansible.yml

│ ├── 03-bootstrap.yml

│ ├── 04-deploy-openstack.yml

│ └── 05-post-deploy.yml

└── group\_vars/

└── all.yml

**1. Inventory File (inventory/multinode/inventory.ini)**

# These initial groups are the only groups required to be modified. The

# additional groups are for more control of the environment.

[control]

# These hostname must be resolvable from your deployment host

controller1

controller2

controller3

# The above can also be specified as follows:

#control[01:03] ansible\_user=kolla

# The network nodes are where your l3-agent and loadbalancers will run

# This can be the same as a host in the control group

[network]

controller1

controller2

controller3

[compute]

controller1

controller2

controller3

[monitoring]

controller1

# When compute nodes and control nodes use different interfaces,

# you need to comment out "api\_interface" and other interfaces from the globals.yml

# and specify like below:

#compute01 neutron\_external\_interface=eth0 api\_interface=em1 tunnel\_interface=em1

[storage]

controller1

controller2

controller3

[deployment]

localhost ansible\_connection=local

[baremetal:children]

control

network

compute

storage

monitoring

[tls-backend:children]

control

# You can explicitly specify which hosts run each project by updating the

# groups in the sections below. Common services are grouped together.

[common:children]

control

network

compute

storage

monitoring

[collectd:children]

compute

[grafana:children]

monitoring

[etcd:children]

control

[influxdb:children]

monitoring

[prometheus:children]

monitoring

[telegraf:children]

compute

control

monitoring

network

storage

[hacluster:children]

control

[hacluster-remote:children]

compute

[loadbalancer:children]

network

[mariadb:children]

control

[rabbitmq:children]

control

[keystone:children]

control

[glance:children]

control

[nova:children]

control

[neutron:children]

network

[openvswitch:children]

network

compute

manila-share

[cinder:children]

control

[cloudkitty:children]

control

[memcached:children]

control

[horizon:children]

control

[barbican:children]

control

[heat:children]

control

[ironic:children]

control

[magnum:children]

control

[mistral:children]

control

[manila:children]

control

[ceilometer:children]

control

[aodh:children]

control

[cyborg:children]

control

compute

[gnocchi:children]

control

[tacker:children]

control

[trove:children]

control

[watcher:children]

control

[octavia:children]

control

[designate:children]

control

[placement:children]

control

[bifrost:children]

deployment

[zun:children]

control

[skyline:children]

control

[redis:children]

control

[blazar:children]

control

[venus:children]

monitoring

[letsencrypt:children]

loadbalancer

# Additional control implemented here. These groups allow you to control which

# services run on which hosts at a per-service level.

#

# Word of caution: Some services are required to run on the same host to

# function appropriately. For example, neutron-metadata-agent must run on the

# same host as the l3-agent and (depending on configuration) the dhcp-agent.

# Common

[cron:children]

common

[fluentd:children]

common

[kolla-logs:children]

common

[kolla-toolbox:children]

common

[opensearch:children]

control

# Opensearch dashboards

[opensearch-dashboards:children]

opensearch

# Glance

[glance-api:children]

glance

# Nova

[nova-api:children]

nova

[nova-conductor:children]

nova

[nova-metadata:children]

nova

[nova-super-conductor:children]

nova

[nova-novncproxy:children]

nova

[nova-scheduler:children]

nova

[nova-spicehtml5proxy:children]

nova

[nova-compute-ironic:children]

nova

[nova-serialproxy:children]

nova

# Neutron

[neutron-server:children]

control

[neutron-dhcp-agent:children]

neutron

[neutron-l3-agent:children]

neutron

[neutron-metadata-agent:children]

neutron

[neutron-ovn-metadata-agent:children]

compute

network

[neutron-bgp-dragent:children]

neutron

[neutron-infoblox-ipam-agent:children]

neutron

[neutron-metering-agent:children]

neutron

[ironic-neutron-agent:children]

neutron

[neutron-ovn-agent:children]

compute

network

# Cinder

[cinder-api:children]

cinder

[cinder-backup:children]

storage

[cinder-scheduler:children]

cinder

[cinder-volume:children]

storage

# Cloudkitty

[cloudkitty-api:children]

cloudkitty

[cloudkitty-processor:children]

cloudkitty

# iSCSI

[iscsid:children]

compute

storage

ironic

[tgtd:children]

storage

# Manila

[manila-api:children]

manila

[manila-scheduler:children]

manila

[manila-share:children]

network

[manila-data:children]

manila

# Barbican

[barbican-api:children]

barbican

[barbican-keystone-listener:children]

barbican

[barbican-worker:children]

barbican

# Heat

[heat-api:children]

heat

[heat-api-cfn:children]

heat

[heat-engine:children]

heat

# Ironic

[ironic-api:children]

ironic

[ironic-conductor:children]

ironic

[ironic-inspector:children]

ironic

[ironic-tftp:children]

ironic

[ironic-http:children]

ironic

# Magnum

[magnum-api:children]

magnum

[magnum-conductor:children]

magnum

# Mistral

[mistral-api:children]

mistral

[mistral-executor:children]

mistral

[mistral-engine:children]

mistral

[mistral-event-engine:children]

mistral

# Ceilometer

[ceilometer-central:children]

ceilometer

[ceilometer-notification:children]

ceilometer

[ceilometer-compute:children]

compute

[ceilometer-ipmi:children]

compute

# Aodh

[aodh-api:children]

aodh

[aodh-evaluator:children]

aodh

[aodh-listener:children]

aodh

[aodh-notifier:children]

aodh

# Cyborg

[cyborg-api:children]

cyborg

[cyborg-agent:children]

compute

[cyborg-conductor:children]

cyborg

# Gnocchi

[gnocchi-api:children]

gnocchi

[gnocchi-statsd:children]

gnocchi

[gnocchi-metricd:children]

gnocchi

# Trove

[trove-api:children]

trove

[trove-conductor:children]

trove

[trove-taskmanager:children]

trove

# Multipathd

[multipathd:children]

compute

storage

# Watcher

[watcher-api:children]

watcher

[watcher-engine:children]

watcher

[watcher-applier:children]

watcher

# Octavia

[octavia-api:children]

octavia

[octavia-driver-agent:children]

octavia

[octavia-health-manager:children]

octavia

[octavia-housekeeping:children]

octavia

[octavia-worker:children]

octavia

# Designate

[designate-api:children]

designate

[designate-central:children]

designate

[designate-producer:children]

designate

[designate-mdns:children]

network

[designate-worker:children]

designate

[designate-sink:children]

designate

[designate-backend-bind9:children]

designate

# Placement

[placement-api:children]

placement

# Zun

[zun-api:children]

zun

[zun-wsproxy:children]

zun

[zun-compute:children]

compute

[zun-cni-daemon:children]

compute

# Skyline

[skyline-apiserver:children]

skyline

[skyline-console:children]

skyline

# Tacker

[tacker-server:children]

tacker

[tacker-conductor:children]

tacker

# Blazar

[blazar-api:children]

blazar

[blazar-manager:children]

blazar

# Prometheus

[prometheus-node-exporter:children]

monitoring

control

compute

network

storage

[prometheus-mysqld-exporter:children]

mariadb

[prometheus-memcached-exporter:children]

memcached

[prometheus-cadvisor:children]

monitoring

control

compute

network

storage

[prometheus-alertmanager:children]

monitoring

[prometheus-openstack-exporter:children]

monitoring

[prometheus-elasticsearch-exporter:children]

opensearch

[prometheus-blackbox-exporter:children]

monitoring

[prometheus-libvirt-exporter:children]

compute

[masakari-api:children]

control

[masakari-engine:children]

control

[masakari-hostmonitor:children]

control

[masakari-instancemonitor:children]

compute

[ovn-controller:children]

ovn-controller-compute

ovn-controller-network

[ovn-controller-compute:children]

compute

[ovn-controller-network:children]

network

[ovn-database:children]

control

[ovn-northd:children]

ovn-database

[ovn-nb-db:children]

ovn-database

[ovn-sb-db:children]

ovn-database

[ovn-sb-db-relay:children]

ovn-database

[venus-api:children]

venus

[venus-manager:children]

venus

[letsencrypt-webserver:children]

letsencrypt

[letsencrypt-lego:children]

letsencrypt

(kolla-venv) root@controller1:~#

**2. Playbook: Prerequisites (playbooks/01-prerequisites.yml)**

---

- name: Install system dependencies

hosts: all

become: yes

tasks:

- name: Update APT packages

apt:

update\_cache: yes

- name: Install required packages

apt:

name:

- python3-pip

- python3-dev

- libffi-dev

- gcc

- libssl-dev

- python3-venv

- git

- vim

- net-tools

- bridge-utils

- libvirt-daemon-system

- qemu-kvm

- virt-manager

- libguestfs-tools

state: present

**3. Playbook: Install Kolla-Ansible (playbooks/02-install-kolla-ansible.yml)**

---

- name: Install Kolla-Ansible and dependencies

hosts: all

become: yes

tasks:

- name: Install Docker and dependencies

apt:

name:

- docker.io

state: present

- name: Add user to docker group

user:

name: "{{ ansible\_user }}"

groups: docker

append: yes

- name: Install Kolla-Ansible via pip

pip:

name:

- kolla-ansible

- name: Create necessary directories

file:

path: "{{ item }}"

state: directory

loop:

- /etc/kolla

- /usr/local/share/kolla-ansible

- name: Copy default configs

shell: |

cp -r /usr/local/share/kolla-ansible/etc\_examples/kolla/\* /etc/kolla

cp /usr/local/share/kolla-ansible/ansible/inventory/\* /etc/kolla

args:

creates: /etc/kolla/globals.yml

**4. Playbook: Bootstrap Deployment (playbooks/03-bootstrap.yml)**

---

- name: Bootstrap Kolla-Ansible

hosts: all

become: yes

tasks:

- name: Generate passwords

shell: kolla-genpwd

args:

chdir: /etc/kolla

- name: Bootstrap servers

shell: kolla-ansible -i /etc/kolla/inventory/multinode bootstrap-servers

**5. Playbook: Deploy OpenStack (playbooks/04-deploy-openstack.yml)**

---

- name: Deploy OpenStack using Kolla-Ansible

hosts: all

become: yes

tasks:

- name: Prechecks

shell: kolla-ansible -i /etc/kolla/inventory/multinode prechecks

- name: Deploy OpenStack

shell: kolla-ansible -i /etc/kolla/inventory/multinode deploy

**6. Playbook: Post-Deployment Setup (playbooks/05-post-deploy.yml)**

---

- name: Post deployment steps

hosts: all

become: yes

tasks:

- name: Initialize OpenStack

shell: kolla-ansible post-deploy

- name: Copy OpenRC file

shell: cp /etc/kolla/admin-openrc.sh ~/

**7. Group Vars File (group\_vars/all.yml)**

---

workaround\_ansible\_issue\_8743: yes

kolla\_base\_distro: "ubuntu"

openstack\_release: "master"

kolla\_internal\_vip\_address: "192.168.1.171"

kolla\_external\_vip\_address: "{{ kolla\_internal\_vip\_address }}"

network\_interface: "ens160"

kolla\_external\_vip\_interface: "{{ network\_interface }}"

api\_interface: "{{ network\_interface }}"

octavia\_network\_interface: "{{ api\_interface }}"

neutron\_external\_interface: "ens192"

neutron\_plugin\_agent: "ovn"

enable\_openstack\_core: "yes"

enable\_rabbitmq: "{{ 'yes' if om\_rpc\_transport == 'rabbit' or om\_notify\_transport == 'rabbit' else 'no' }}"

enable\_cinder: "yes"

enable\_cinder\_backup: "yes"

enable\_fluentd: "no"

enable\_redis: "yes"

enable\_horizon\_octavia: "{{ enable\_octavia | bool }}"

enable\_neutron\_qos: "yes"

enable\_neutron\_provider\_networks: "yes"

enable\_octavia: "yes"

enable\_trove: "yes"

enable\_octavia\_driver\_agent: "{{ enable\_octavia | bool and neutron\_plugin\_agent == 'ovn' }}"

ceph\_glance\_user: "glance"

ceph\_glance\_pool\_name: "images"

ceph\_glance\_keyring: "client.glance.keyring"

ceph\_cinder\_user: "cinder"

ceph\_cinder\_pool\_name: "volumes"

ceph\_cinder\_keyring: "client.cinder.keyring"

ceph\_cinder\_backup\_user: "cinder-backup"

ceph\_cinder\_backup\_pool\_name: "backups"

ceph\_cinder\_backup\_keyring: "client.cinder-backup.keyring"

ceph\_nova\_user: "nova"

ceph\_nova\_pool\_name: "vms"

ceph\_nova\_keyring: "client.nova.keyring"

glance\_backend\_ceph: "yes"

cinder\_backend\_ceph: "yes"

cinder\_cluster\_skip\_precheck: true

cinder\_backup\_driver: "ceph"

nova\_backend\_ceph: "yes"

nova\_compute\_virt\_type: "kvm"

neutron\_ovn\_distributed\_fip: "yes"

octavia\_auto\_configure: yes

octavia\_amp\_flavor:

name: "amphora"

is\_public: no

vcpus: 1

ram: 1024

disk: 5

octavia\_amp\_network\_cidr: 10.1.0.0/24

octavia\_amp\_image\_tag: "amphora"

octavia\_loadbalancer\_topology: "ACTIVE\_STANDBY"

**Steps to Run the Playbooks**

cd openstack-ansible-deployment

ansible-playbook -i inventory/multinode/inventory.ini playbooks/01-prerequisites.yml

ansible-playbook -i inventory/multinode/inventory.ini playbooks/02-install-kolla-ansible.yml

ansible-playbook -i inventory/multinode/inventory.ini playbooks/03-bootstrap.yml

ansible-playbook -i inventory/multinode/inventory.ini playbooks/04-deploy-openstack.yml

ansible-playbook -i inventory/multinode/inventory.ini playbooks/05-post-deploy.yml