# OpenStack Deployment with Ansible & Kolla

## Project Structure

openstack-deploy/

├── ansible.cfg # Ansible configuration file

├── group\_vars/

│ └── all.yml # Global variables for all hosts

├── inventory/

│ └── multinode # Inventory of controller/compute nodes

├── playbooks/

│ └── site.yml # Master playbook triggering all roles

├── roles/ # Modular roles for each deployment phase

│ ├── bootstrap\_kolla/

│ ├── create\_venv/

│ ├── deploy\_openstack/

│ ├── install\_deps/

│ ├── install\_kolla/

│ ├── post\_deploy/

│ ├── prepare\_configs/

│ └── run\_prechecks/

├── precheck.log # Output log from pre-deployment checks

└── requirements.yml # Ansible Galaxy dependency definitions

## Setup Prerequisites

### 1️⃣ Install Ansible Core

Use a virtual environment for isolation:

python3 -m venv kolla-venv

source kolla-venv/bin/activate

pip install 'ansible-core>=2.13,<2.15'

### 2️⃣ Install Required Ansible Collections

ansible-galaxy collection install -r requirements.yml

#### Sample requirements.yml:

collections:

- name: ansible.posix

- name: ansible.utils

- name: community.general

### 3️⃣ Update ansible.cfg

Ensure it includes:

[defaults]

collections\_paths = ~/.ansible/collections:/usr/share/ansible/collections

## Deployment Workflow

### 1. Install System Dependencies

Install Docker, Python packages, SSH, and other required tools:

ansible-playbook playbooks/site.yml --tags install\_deps

### 2. (Optional) Create Python Virtual Environment

ansible-playbook playbooks/site.yml --tags create\_venv

### 3. Install Kolla-Ansible

ansible-playbook playbooks/site.yml --tags install\_kolla

### 4. Prepare Configuration Files

ansible-playbook playbooks/site.yml --tags prepare\_configs

This:

* Copies passwords.yml
* Renders globals.yml from globals.yml.j2
* Copies Ceph config/keyrings if applicable

Confirm /etc/kolla/globals.yml is fully rendered (no {{ }} placeholders).

### 5. Define Host Group Mappings (Octavia & Others)

In inventory/multinode, add:

[octavia-api]

controller1

[octavia-worker]

controller1

[octavia-housekeeping]

controller1

[octavia-health-manager]

controller1

Repeat for any other services you've enabled in globals.yml (e.g., zun, vitrage).

### 6. Generate Octavia TLS Certificates

Required for enable\_octavia: true:

kolla-ansible octavia-certificates

This creates a local CA and drops certs into /etc/kolla/certificates/.

### 7. Re-Prepare Configuration (Now Including TLS Paths)

ansible-playbook playbooks/site.yml --tags prepare\_configs

### 8. Bootstrap Kolla Runtime Environment

ansible-playbook playbooks/site.yml --tags bootstrap\_kolla

### 9. Run Pre-Deployment Checks

ansible-playbook playbooks/site.yml --tags run\_prechecks

Review precheck.log for failures like:

* Hostname-to-IP mismatch
* Octavia resolution errors
* Missing certificates
* DNS cache issues

### 10. Deploy OpenStack

ansible-playbook playbooks/site.yml --tags deploy\_openstack or

/root/kolla-venv/bin/kolla-ansible -i /root/openstack-deploy/inventory/multinode prechecks

Or the manual equivalent:

kolla-ansible -i inventory/multinode deploy

### 11. Post-Deployment Setup

ansible-playbook playbooks/site.yml --tags post\_deploy or

kolla-ansible -i multinode post-deploy

This includes:

* kolla-ansible post-deploy
* Appending OS\_CACERT to admin-openrc.sh
* Installing python-openstackclient

## Validate the Deployment

### Load OpenStack CLI Environment

source /etc/kolla/admin-openrc.sh

### Basic Service Tests

openstack service list

openstack compute service list

openstack network agent list

## Horizon Dashboard Access

Visit:

http://<controller-vip> or http://<controller-ip>

Login:

* Username: admin
* Password: (check /etc/kolla/passwords.yml under keystone\_admin\_password)