
OpenStack and Containers
Installing OpenStack-Ansible

AIO Host

Minimum Hardware Requirements

8 vCPUs
50GB free disk space on /
8GB RAM

Deploying the AIO

Prepare the host

Configure SSH keys from local machine

ssh-copy-id –i .ssh/id_rsa.pub <user>@<IP>

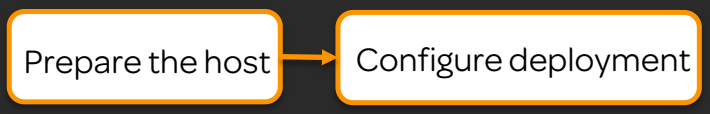
Configure the Operating System

sudo su -
apt-get update
apt-get dist-upgrade
apt-get install aptitude build-essential git ntp ntpdate python-dev

Install the source and dependencies

git clone -b stable/queens https://git.openstack.org/openstack/openstack-ansible /opt/openstack-ansible
scripts/bootstrap-ansible.sh

Deploying the AIO



Configure the Deployment

scripts/bootstrap-aio.sh
nano /etc/openstack_deploy/user_secrets.yml

Deploying the AIO

Run the Playbooks

```
cd /opt/openstack-ansible/playbooks
openstack-ansible setup-hosts.yml
openstack-ansible setup-infrastructure.yml
openstack-ansible setup-openstack.yml
```

Deploying the AIO

Verify OpenStack Operation

```
cd ~
. openrc
openstack service list
openstack network list
openstack flavor list
openstack image list
openstack server create --image cirros
--flavor tempest1 --network private test
openstack server list
```


OpenStack and Containers
Managing OpenStack-Ansible

LXC Commands

```
ls /usr/bin/lxc-*
man lxc-ls
```

The Containers

Finding the Containers

less /etc/hosts
lxc-ls

Accessing the Containers

ssh <container>
lxc-attach --name <container>

Managing OpenStack

Restarting a Service

service nova-scheduler restart

Looking at Logs

ls /var/log/nova
tail /var/log/nova/nova-scheduler.log
exit
ls /var/log/nova
tail /var/log/nova/nova-compute.log

Rebooting an AIO

Restarting Galera

```
cd /opt/openstack-ansible/playbooks
openstack-ansible –e
galera_ignore_cluster_state=true galera-
install.yml
```

Verify OpenStack

```
cd ~
. openrc
openstack service list
```

OpenStack and Containers

Installing Kolla

Network Configuration

Create a Host-only Network

Global Tools
Host Network Manager
Create Network
Enable Server
Adapter
- IPv4 – 10.10.20.1

DHCP
- Server Address: 10.10.20.2
- Server Mask: 255.255.255.0
- Lower Address Bound: 10.10.20.3
- Upper Address Bound: 10.10.20.254

AIO Host

Minimum Hardware Requirements

4 vCPUs
40GB free disk space on /
8GB RAM
1 NIC on NAT
2 NICs both on vboxnet<#>

Deploying the AIO

Prepare the host

Configure the Network Interfaces

```
sudo su -
nano /etc/network/interfaces
auto enp0s8
iface enp0s8 inet static
address 10.10.20.31
netmask 255.255.255.0
auto enp0s9
iface enp0s9 inet static
address 10.10.20.32
netmask 255.255.255.0
```

Restart the Networking Service

```
service networking restart
```

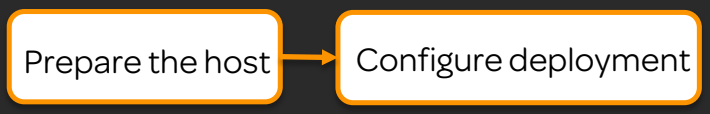
Deploying the AIO

Prepare the host

Configure the Operating System

apt-get update
apt-get dist-upgrade
apt-get install python-jinja2 python-pip libssl-dev
pip install ansible

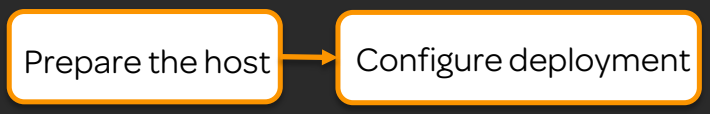
Deploying the AIO



Install the Deployment

```
pip install kolla-ansible==6.1.0
cp -r /usr/local/share/kolla-
ansible/etc_examples/kolla /etc/kolla
```

Deploying the AIO

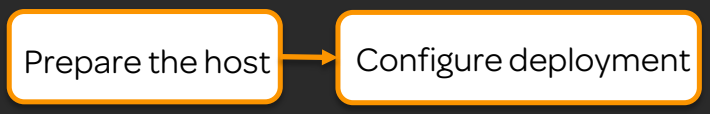


Configure the Deployment

nano /etc/kolla/globals.yml

```
kolla_base_distro: "ubuntu"
kolla_install_type: "binary"
openstack_release: "queens"
kolla_internal_vip_address: "10.10.20.33"
kolla_external_vip_address: "10.10.20.30"
network_interface: "enp0s8"
neutron_external_interface: "enp0s9"
kolla_enable_tls_external: "yes"
nodeconfigdirectory: "/etc/kolla"
kolla_external_fqdn_cert: "{{ nodeconfigdirectory
}}/certificates/haproxy.pem"
api_interface: "{{ network_interface }}"
```

Deploying the AIO



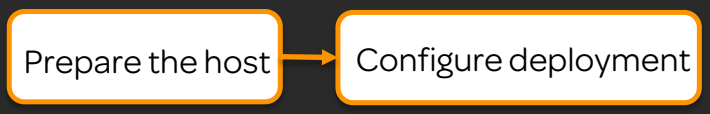
Create the Certs

kolla-ansible certificates

Configure Nova

mkdir -p /etc/kolla/config/nova
nano /etc/kolla/config/nova/nova-compute.conf
[libvirt]
virt_type = qemu
cpu_mode = none

Deploying the AIO



Generate and Set Passwords

kolla-genpwd
nano /etc/kolla/passwords.yml
keystone_admin_password: openstack

Bootstrap the Server

kolla-ansible -i /usr/local/share/kolla-ansible/ansible/inventory/all-in-one bootstrap-servers

Pull Docker Images

kolla-ansible pull

Deploying the AIO

Check the Environment

kolla-ansible prechecks -i /usr/local/share/kolla-ansible/ansible/inventory/all-in-one

Deploy OpenStack

kolla-ansible deploy -i /usr/local/share/kolla-ansible/ansible/inventory/all-in-one

Verify Docker

docker ps -a

Deploying the AIO

Generate Creds FIle

kolla-ansible post-deploy

Install the OpenStack CLI

pip install python-openstackclient

Edit the init-runonce File

nano /usr/local/share/kolla-ansible/init-runonce
EXT_NET_CIDR='10.10.20.0/24'
EXT_NET_RANGE='start=10.10.20.110,end=10.10.20.254'
EXT_NET_GATEWAY='10.10.20.1'

Deploying the AIO

Source Creds FIle

source /etc/kolla/admin-openrc.sh

Initialize the Deployment

cd /usr/local/share/kolla-ansible/ && ./init-runonce

Deploying the AIO

Verify OpenStack Operation

```
openstack service list
openstack network list
openstack flavor list
openstack image list
openstack server create --image cirros
--flavor m1.tiny --network demo-net test
openstack server list
```

OpenStack and Containers

Managing OpenStack with Kolla

Docker Commands

```
man docker
man docker-exec
docker ps | grep nova
docker exec nova_api ls /etc/nova
```

The Containers

Finding the Containers

```
docker ps -a
```

Accessing the Containers

```
docker exec –it nova_api bash
ls /etc/nova
exit
```

Managing OpenStack

Restarting a Service

```
docker ps -a | grep nova_scheduler
docker restart nova_scheduler
docker ps -a | grep nova_scheduler
```

Looking at Logs

```
docker exec -it fluentd bash
tail /var/log/kolla/nova/nova-scheduler.log
tail /var/log/kolla/nova/nova-compute.log
exit
```

Rebooting an AIO

Rebooting the AIO

reboot
ssh kolla@10.10.20.31
sudo su -

Verify OpenStack

source /etc/kolla/admin-openrc.sh
openstack service list


OpenStack and Containers
Installing Zun with Devstack

Devstack Host

Minimum Hardware Requirements

8 vCPUs
50GB free disk space on /
8GB RAM

Prepping the Host

Prepare the host

Update and Install Packages

```
apt update
apt dist-upgrade -y
apt install bridge-utils -y
```

Create stack User

```
useradd -m  -s /bin/bash stack
passwd stack
echo "stack ALL=(ALL) NOPASSWD: ALL" >>
/etc/sudoers
```

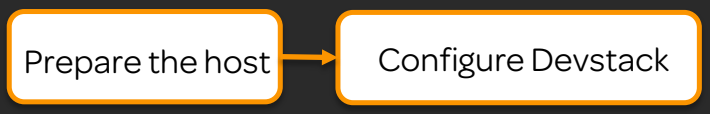
Prepping the Host

Prepare the host

Install the Source

```
sudo su – stack
sudo git clone
https://git.openstack.org/openstack-
dev/devstack /opt/stack/devstack
sudo chown –R stack:stack /opt/stack
cd /opt/stack/devstack
```

Configuring Devstack



Configure Devstack

Checkout Version

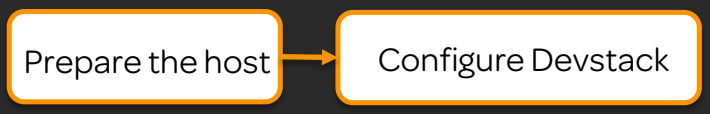
```
git checkout stable/rocky
```

nano local.conf

```
[[local|localrc]]
ADMIN_PASSWORD=openstack
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD

# install python-zunclient from git
LIBS_FROM_GIT="python-zunclient"
```

Configuring Devstack



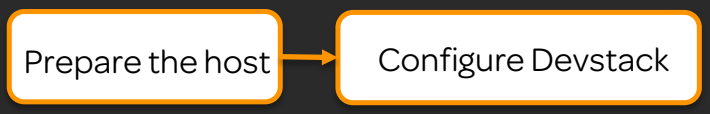
Configure Devstack

/opt/stack/devstack/local.conf

```
#Zun
enable_plugin zun
https://git.openstack.org/openstack/zun
enable_plugin zun-tempest-plugin
https://git.openstack.org/openstack/zun-
tempest-plugin

#Install devstack container plugin
enable_plugin devstack-plugin-container
https://git.openstack.org/openstack/devstack-
plugin-container
```

Configuring Devstack



Configure Devstack

/opt/stack/devstack/local.conf

```
KURYR_CAPABILITY_SCOPE=local
KURYR_ETCD_PORT=2379
KURYR_PROCESS_EXTERNAL_CONNECTIVITY
=False
enable_plugin kuryr-libnetwork
https://git.openstack.org/openstack/kuryr-
libnetwork

enable_plugin zun-ui
https://git.openstack.org/openstack/zun-ui

enable_plugin heat
https://git.openstack.org/openstack/heat
```

Deploying Devstack

Run stack.sh

./stack.sh

Verifying Devstack

Verify OpenStack

```
. openrc admin admin
openstack service list
openstack network list
openstack flavor list
openstack image list
openstack server create --image cirros-0.3.5-x86_64-disk --flavor m1.tiny --network public test
openstack server list
```


OpenStack and Containers
Managing Containers with Zun

Working with Zun with the CLI

Source Credentials

```
ls
. admin-openrc.sh
```

Check the Help

```
openstack appcontainer --help | less
```

Check the System

```
openstack appcontainer host list
openstack appcontainer service list
```

Working with Images with the CLI

Check the Help

```
openstack appcontainer image search --help | less
openstack appcontainer image pull --help | less
openstack appcontainer image delete --help | less
```

Find and Pull an Image

```
openstack appcontainer image search ubuntu
openstack appcontainer image pull ubuntu <host>
openstack appcontainer image list
```

Delete Image

```
openstack appcontainer image delete <UUID>
openstack appcontainer image list
```

Create a Container with the CLI

Check the Help

```
openstack appcontainer create --help | less
openstack appcontainer run --help | less
openstack appcontainer exec --help | less
openstack appcontainer stop --help | less
```

Gather Network Information

```
openstack network list
```

Create and Interact with Container

```
openstack appcontainer run --name test
--net network=public cirros ping 8.8.8.8
openstack appcontainer list
openstack appcontainer exec --interactive
test /bin/sh
ping 8.8.8.8
exit
openstack appcontainer stop test
openstack appcontainer list
```

Working with Zun in the Dashboard

Create Container

Name: test2
Image: cirros
Network: public

Interact with Containers

Start test
Stop and Delete test and test2

OpenStack and Containers

Installing Magnum with Devstack

Devstack Host

Minimum Hardware Requirements

Bare Metal not VM
8 CPUs
50GB free disk space on /
8GB RAM

Prepping the Host

Prepare the host

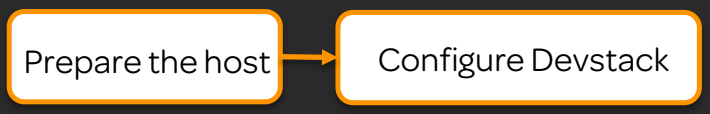
Create stack User

```
useradd -m  -s /bin/bash stack
passwd stack
echo "stack ALL=(ALL) NOPASSWD: ALL" >>
/etc/sudoers
```

Install the source

```
sudo su – stack
sudo git clone
https://git.openstack.org/openstack-
dev/devstack /opt/stack/devstack
sudo chown –R stack:stack /opt/stack
cd /opt/stack/devstack
```

Configuring Devstack



Configure Devstack

Checkout Version

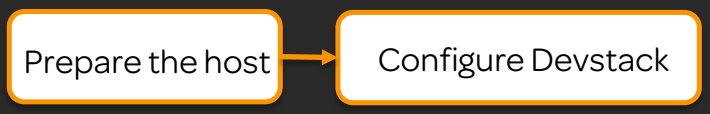
```
git checkout stable/rocky
```

nano local.conf

```
[[local|localrc]]
ADMIN_PASSWORD=openstack
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD

# magnum needs external interface
PUBLIC_INTERFACE=eno1
```

Configuring Devstack

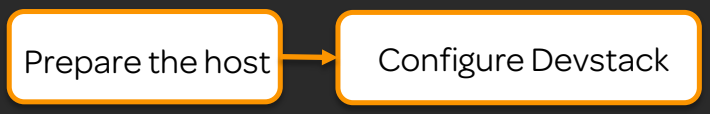


Configure Devstack

/opt/stack/devstack/local.conf

```
# Enable barbican service
enable_plugin barbican
https://git.openstack.org/openstack/barbican
enable_plugin heat
https://git.openstack.org/openstack/heat
enable_plugin heat-dashboard
https://git.openstack.org/openstack/heat-
dashboard
```

Configuring Devstack



Configure Devstack

/opt/stack/devstack/local.conf

```
# Enable magnum
enable_plugin magnum
https://git.openstack.org/openstack/magnum

enable_plugin magnum-ui
https://github.com/openstack/magnum-ui

VOLUME_BACKING_FILE_SIZE=20G
```

Deploying Devstack

Run stack.sh

./stack.sh

Verifying Devstack

Verify OpenStack Operation

```
. openrc admin admin
openstack service list
openstack network list
openstack flavor list
openstack image list
openstack server create --image cirros-0.3.5-x86_64-disk --flavor m1.tiny --network public test
openstack server list
```

OpenStack and Containers

Managing Clusters with Magnum Using the CLI

Managing Magnum

Source Credentials

```
cd /opt/stack/devstack
. openrc admin admin
```

Check the Help

```
openstack coe --help
```

Manage Magnum

```
openstack coe service list
openstack image set --help | less
```

Creating a Cluster Template on the CLI

Create a Key Pair

```
ssh-keygen –q –N ""
openstack keypair create --public-key
~/.ssh/id_rsa.pub mag-key
```

Check the Help

```
openstack coe cluster template create --help | less
```

Create a Cluster Template

```
openstack image list
openstack coe cluster template create kube-
template --image Fedora-Atomic-27-
20180212.2.x86_64 --keypair mag-key
--external-network public --flavor m1.small
--dns-nameserver 8.8.8.8 --coe kubernetes
--docker-volume-size 3 --docker-storage-driver
overlay
openstack coe cluster template list
```

Creating a Cluster on the CLI

Check the Help

```
openstack coe cluster create --help | less
```

Create a Cluster

```
openstack coe cluster create --cluster-template
kube-template --node-count 1 kube-cluster
openstack coe cluster list
openstack coe cluster show kube-cluster
openstack stack list
openstack stack show <stack>
openstack coe cluster list
```

The Cluster

Finding the Container

openstack server list
ping <external IP of container>

Delete the Cluster

openstack coe cluster delete kube-cluster
openstack coe cluster list
openstack stack list

Delete the Cluster Template

openstack coe cluster template delete kube-template
openstack coe cluster template list

OpenStack and Containers

Managing Clusters with Magnum Using the Dashboard

Creating a Cluster Template in the Dashboard

Create a Cluster Template

Cluster Template Name: kube-template
Cluster Orchestration Engine: Kubernetes
Image: Fedora-Atomic-27-20180212.2.x86_64
Keypair: mag-key
Docker Storage Driver: Overlay
Docker Volume Size: 3
External Network ID: public

Creating a Cluster in the Dashboard

Create a Cluster

Cluster Name: kube-cluster
Cluster Template Name: kube-template
Node Count 1:
Node Flavor ID: m1.small

The Cluster

Delete the Cluster

Name: kube-cluster

Delete the Cluster Template

Name: kube-template