

OpenStack 구축(22.08 Yoga)

[Quick Start – kolla-ansible 14.1.0.dev143 documentation](#) 참고

- 외부망
 - ost-master : 200.200.0.11
 - node1 : 200.200.0.12
 - node2 : 200.200.0.13
- SSH Port = 22
- 계정정보
 - ID : miruware (Sudo User)
 - PW : miruware!

서버 정보

```
# ost-master (VM)
eth0 : 200.200.0.11/24
eth1 : x (neutron external interface로 사용)
internal : 50.50.0.11/24

# node1
eth0 : 200.200.0.12/24
internal : 50.50.0.12/24

# node2
eth0 : 200.200.0.13/24
internal : 50.50.0.13/24
```

Network 장치명 고정 (모든 노드 실행)

모든 작업은 root 계정으로 진행한다.

/etc/default/grub 수정

```

$ vim /etc/default/grub

#####

# 아래 라인을 수정한다.

GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"

#####

$ update-grub

$ reboot

```

/etc/udev/rules.d/70-persistent-net.rules 생성

master

```

$ vim /etc/udev/rules.d/70-persistent-net.rules

#####

# ATTR{address}== 부분에 mac address를 입력하고 NAME에 원하는 장치명을 입력한다.

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}
=="00:0c:29:0e:f6:cd", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth0"

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}
=="00:0c:29:0e:f6:d7", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth1"

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}
=="00:0c:29:0e:f6:e1", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",
NAME="internal"

#####

$ reboot

```

node1

```

$ vim /etc/udev/rules.d/70-persistent-net.rules

#####

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}
=="a4:bf:01:29:44:b7", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth0"

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}

```

```

=="a4:bf:01:29:44:b8", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth1"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}
=="00:e0:63:35:0c:0f", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",
NAME="internal"
#####

$ reboot

```

node2

```

$ vim /etc/udev/rules.d/70-persistent-net.rules
#####
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}
=="00:1e:67:51:30:dc", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth0"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}
=="00:1e:67:51:30:df", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",
NAME="internal"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}
=="00:1e:67:51:30:dd", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth1"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}
=="00:1e:67:51:30:de", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth2"
#####

$ reboot

```

netplan 으로 IP Static 설정

master

```

$ vim /etc/netplan/00-installer-config.yaml
#####
network:
  ethernets:

```

```

eth0:
  dhcp4: no
  addresses: [200.200.0.11/24]
  gateway4: 200.200.0.1
  nameservers:
    addresses: [8.8.8.8]

eth1:
  dhcp4: no

internal:
  addresses: [50.50.0.11/24]

version: 2

#####

```

node1

```

$ vim /etc/netplan/00-installer-config.yaml

#####

network:
  ethernets:
    eth0:
      dhcp4: no
      addresses: [200.200.0.12/24]
      gateway4: 200.200.0.1
      nameservers:
        addresses: [8.8.8.8]

    internal:
      addresses: [50.50.0.12/24]

  version: 2

#####

```

node2

```

$ vim /etc/netplan/00-installer-config.yaml

#####

network:
  ethernets:
    eth0:

```

```
dhcp4: no
addresses: [200.200.0.13/24]
gateway4: 200.200.0.1
nameservers:
    addresses: [8.8.8.8]
internal:
    addresses: [50.50.0.13/24]
version: 2
#####

$ reboot
```

업데이트 및 기본 패키지 설치

apt update && apt upgrade (모든 노드 실행)

```
$ apt update && apt upgrade -y
```

자동 업데이트 끄기 (모든 노드 실행)

```
$ vim /etc/apt/apt.conf.d/10periodic
#####
# 아래 줄 마지막 1을 0으로 변경
APT::Periodic::Unattended-Upgrade "0";
#####
```

방화벽 끄기 (모든 노드 실행)

```
$ ufw disable
>>> Firewall stopped and disabled on system startup

# 제대로 비활성화 되었는지 확인
```

```
$ ufw status
>>> Status: inactive
```

root 계정 활성화 및 SSH root 접속 허용 (모든 노드 실행)

```
# root 계정 활성화
$ passwd
(password 입력)

# ssh root 접속 허용
$ vim /etc/ssh/sshd_config
#####
# 내용 수정
PermitRootLogin yes
#####

$ systemctl restart sshd.service
```

/etc/hosts 파일 수정 (Master 노드에서 실행)

```
#####
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

# MW-OST
50.50.0.11 master
50.50.0.12 node1
50.50.0.13 node2
#####
```

#####SSH KEY 배포 (마스터에서 실행)

```
# ssh-keygen
$ ssh-keygen -t rsa

# root 계정에서 비밀번호 없이 접속 가능하도록 설정
$ ssh-copy-id node1
$ ssh-copy-id node2
```

오픈스택 구축

- Requirements

- ☒ Master 서버의 `_etc_hosts` 파일 수정
- ☒ 방화벽 해제
- ☒ 자동 업데이트 비활성화
- ☒ 모든 서버의 root 계정 활성화 및 SSH root 접속 허용 (`ssh-keygen & ssh-copy-id`)
- ☒ 모든 서버의 SSH 22번 포트 활성화
- ☒ 모든 서버의 네트워크 장치명 통일 (Openstack 구축용 장치만 통일해도 됨)
- ☒ Master 서버의 Instance 통신용 Ethernet 케이블 장착 여부 (2번째 네트워크 장치 이용)

모든 작업은 Master Node 에서 Root 계정으로 실행

기본 패키지 설치

```
$ apt-get install software-properties-common
$ apt-add-repository ppa:ansible/ansible
$ apt-get update
$ apt-get install python3-dev libffi-dev gcc libssl-dev
$ apt-get install ansible=5.10.0-1ppa~focal python-pip-whl python3-pip
libguestfs-tools

$ pip3 install pip -U

# pip3 와 pip 헷갈리지 않도록 주의. pip3로 update 한 pip를 사용하는것.
$ pip install kolla==14.1.0 kolla-ansible==14.1.0 tox gitpython pbr requests
```

```
jinja2 oslo_config -U
$ pip install python-openstackclient python-glanceclient python-neutronclient
-U
```

Docker 설치

```
$ sudo apt-get update &&
sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent
software-properties-common net-tools ipmitool -y &&
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - &&
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/
ubuntu $(lsb_release -cs) stable" &&
sudo apt-get update &&
sudo apt-get install docker-ce docker-ce-cli containerd.io -y &&
sudo usermod -aG docker $USER
```

ansible.cfg 파일 수정

```
$ ansible-config init --disabled -t all > /etc/ansible/ansible.cfg
$ vi /etc/ansible/ansible.cfg
#####
# 아래 내용을 찾아 수정
[defaults]
forks=100
host_key_checking=False

[Connection]
pipelining=True
#####
```

kolla-ansible 파일 복사

```
$ mkdir -p ~/kolla-ansible
$ cp /usr/local/share/kolla-ansible/ansible/inventory/* ~/kolla-ansible/
```



```
$ mkdir -p /etc/kolla
$ cp -r /usr/local/share/kolla-ansible/etc_examples/kolla/* /etc/kolla/
```

Kolla-ansible Config 파일 수정

```
$ vi ~/kolla-ansible/multinode

#####
# 파일 내용 수정

[control]
# These hostname must be resolvable from your deployment host
master

# 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]
master

[compute]
node1
node2

[monitoring]
master

# 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
storage_interface=em1 tunnel_interface=em1

[storage]
master

[deployment]
```

```
master      ansible_connection=local
#####
```

Kolla-ansible 웹 접속 비밀번호 생성

```
$ cd /etc/kolla && kolla-genpwd

$ mkdir -p /backup/openstack

$ cp /etc/kolla/passwords.yml /backup/openstack/

$ vi /etc/kolla/passwords.yml
#####
# 검색하여 해당 줄만 수정.

/keystone_admin_password: miruware!
#####
```

Openstack 서비스 리스트 config

```
$ vi /etc/kolla/globals.yml
#####
# 검색하여 주석 제거 또는 수정

kolla_base_distro: "ubuntu"
kolla_install_type: "binary"
openstack_release: "yoga"

#->오픈스택 서비스 네트워크 대역중 비는것 VIP사용 ## 호로보드에서 사용
kolla_internal_vip_address: "50.50.0.200"

network_interface: "internal"
neutron_external_interface: "eth1"
neutron_plugin_agent: "openvswitch"

enable_openstack_core: "yes"
enable_glance: "{{ enable_openstack_core | bool }}"
enable_haproxy: "yes"
```

```
enable_keystone: "{{ enable_openstack_core | bool }}"
enable_mariadb: "yes"
enable_memcached: "yes"
enable_neutron: "{{ enable_openstack_core | bool }}"
enable_nova: "{{ enable_openstack_core | bool }}"

#####
```

Ansible Galaxy 요구 사항 설치 (yoga버전 이후 필수)

```
# Ansible Galaxy 종속성 설치(Yoga 릴리스 이후)
$ kolla-ansible install-deps
```

Openstack 배포

```
# python3 를 python 으로 사용할 수 있도록 설정
$ update-alternatives --install /usr/bin/python python /usr/bin/python3 100

# ssh 기본 22번 포트로 통신 하므로 만약 변경 했을 시 22번 포트 추가해줘야함
$ kolla-ansible -i ~/kolla-ansible/multinode bootstrap-servers

#Host 체크
$ kolla-ansible -i ~/kolla-ansible/multinode prechecks

#openstack 배포
$ kolla-ansible -i ~/kolla-ansible/multinode deploy
```

배포 완료 후 네트워크 설정 진행

Openstack 네트워크 설정

```

$ kolla-ansible post-deploy
$ . /etc/kolla/admin-openrc.sh
$ cp /etc/kolla/admin-openrc.sh ~/
$ . /usr/local/share/kolla-ansible/init-runonce

$ vi /etc/kolla/haproxy/services.d/horizon.cfg
#####
# 변경하지 말고 추가할것.
bind 200.200.0.11:30000 #-> 마스터노트 외부아이피 : 접속할 포트
#####

$ docker restart haproxy
$ docker restart horizon

```

dashboard 접속

http://200.200.0.11:30000

ID : admin

PW : miruware!

Dashboard 설정

네트워크 설정

Dashboard 접속 후 (관리 - 네트워크 - 네트워크) 로 이동하여 external-net 과 internal-net 을 생성한다.

external-net 생성

[+네트워크 생성] 클릭

- 네트워크

```

* 이름 : external-net
* 프로젝트 선택 : admin

```

- * 공급자 네트워크 유형 : flat
- * 물리적인 네트워크 : physnet1 (Default = physnet1)

공급자 네트워크 유형 및 물리적 네트워크는 `_etc_kolla_neutron-server_ml2_conf.ini` 파일에서 정의

- ☒ 관리 상태 활성화
- ☒ 공유
- ☒ 외부 네트워크
- ☒ 서브넷 생성
- * 가용구역 힌트 : nova

• 서브넷

- * 서브넷 이름 : external-sub
- * 네트워크 주소 : 200.200.0.0/24 (인터넷 망의 Subnet mask)
- * IP 버전 : IPv4
- * 게이트웨이 IP : 200.200.0.1 (외부망 게이트웨이)
- ☐ 게이트웨이 비활성

• 서브넷 세부정보

- ☐ DHCP 사용
- * Pools 할당 : 200.200.0.20,200.200.0.30 (range)
- * DNS 네임 서버 : 8.8.8.8
- * 호스트 경로 : x

internal-net 생성

[+네트워크 생성] 클릭

• 네트워크

- * 이름 : internal-net
- * 프로젝트 : admin
- * 공급자 네트워크 유형 : VXLAN
- * 구분 ID : 1
- ☒ 관리 상태 활성화
- ☐ 공유

- [] 외부 네트워크
- [x] 서브넷 생성
- * 가용구역 힌트 : nova

• 서브넷

- * 서브넷 이름 : internal-sub
- * 네트워크 주소 : 30.0.0.0/24
- * IP 버전 : IPv4
- * 게이트웨이 IP : 30.0.0.1
- [] 게이트웨이 비활성

• 서브넷 세부정보

- [x] DHCP 사용
- * Pools 할당 : 30.0.0.2,30.0.0.254
- * DNS 네임 서버 : 8.8.8.8
- * 호스트 경로 : x

router 설정

(관리 - 네트워크 - 네트워크 - 라우터) 로 이동

[+ 라우터 생성] 클릭]

- 라우터 이름 : external-route
- 프로젝트 : admin
- ☒ 관리 상태 활성화
- 외부 네트워크 : external-net
- ☒ SNAT 활성화

(프로젝트 - 네트워크 - 라우터 - external-route - 인터페이스) 로 이동

==[+ 인터페이스 추가] 클릭==

- * 서브넷 : internal-net: 30.0.0.0/24
- * IP 주소 (옵션) : 30.0.0.1

인스턴스와 통신하기 위해 route 추가

```
# 오픈스택 인스턴스망 라우터 추가
```

```
$ route add -net 30.0.0.0/24 gw 200.200.0.27
```

(프로젝트 - Compute - 키페어) 로 이동

기존 생성되어 있는 MyKey 삭제 후 ==[+ 키 페어 생성]== 클릭

- 키 페어 이름 : miruware
- 키 유형 : SSH 키

이미지 생성

(프로젝트 - Compute - 이미지) 로 이동

[+ 이미지 생성] 클릭

- 파일 : ==ISO 파일이 아닌 img 파일로 생성 해야함==
- 포맷 : QCOW2
- 가시성 : 공용

Trouble Shooting

Start Docker Error

kolla-ansible -i ~_kolla-ansible_multinode bootstrap-servers 도중 start docker 에러가 나는 경우

etc_apt_sourcelist.d 에 다른 OS 버전의 Docker repository가 생성되어 발생함.

Master와 각 노드에 Docker를 직접 설치 후 실행하였더니 해결 되었음.

Docker cgroup error

Docker Cgroup Warning 이 발생하는 경우 아래 내용을 수정하여 없앨 수 있다.

```
$ sudo vim /usr/lib/systemd/system/docker.service
#####
# execStart= 라인을 찾아 뒤에 아래 내용 추가
--exec-opt native.cgroupdriver=systemd
#####

$ sudo systemctl daemon-reload
$ sudo systemctl restart docker
$ sudo docker info | grep -i cgroup
>>> Cgroup Driver: systemd
```

haproxy : Waiting for virtual IP to appear

아래 파일 내용을 수정

```
$ sudo vi /etc/kolla/globals.yml
#####
keepalived_virtual_router_id: "251"
#####

$ sudo kolla-genpwd
```

Multinode pull : Could not match supplied host pattern

multinode pull 에서 Could not match supplied host pattern 에러가 발생할 경우 명령어에서 multinode 파일 경로를 절대 경로로 지정한다.

```
$ kolla-ansible -i ~/kolla-ansible/multinode pull
```