

KVM 인스턴스 및 Kubernetes 플랫폼 설치

2023.10.19

KVM 인스턴스 설치 초기 설정

- OS : Ubuntu 22.04 LTS
- sudo apt-get update
- sudo apt-get upgrade
- sudo vi /etc/netplan/~~~.yaml 내용을 아래 내용대로 편집하여 브릿지 설정
- sudo netplan apply 입력하여 변경사항 반영

```
# Let NetworkManager manage all devices on this system

network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp216s0f1: → 이더넷 인터페이스 확인하여 이름 변경
      dhcp4: false
  bridges:
    br0:
      interfaces: [enp216s0f1] → 이더넷 인터페이스 확인하여 이름 변경
      dhcp4: true
      dhcp6: false
      addresses: [192.168.240.201/24] → 사용할 IP주소 입력
      routes:
        to: default
        via: 192.168.240.1 → 기본 게이트웨이
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4] → DNS 서버 설정
      parameters:
        stp: true
        forward-delay: 4
      dhcp4: no
```

KVM 인스턴스 설치 초기 설정

- 필수 패키지 설치
- 모든 설치 및 구동 작업은 root 계정으로 진행 (sudo su 명령어 이용하여 root 계정 접속)
- `sudo apt-get install qemu qemu-kvm libvirt-clients libvirt-daemon-system virtinst bridge-utils virt-manager libvirt-bin cpu-checker`
- `kvm-ok` 명령어 입력하여 아래 내용 확인

```
root@server1:~/kvm# kvm-ok  
INFO: /dev/kvm exists  
KVM acceleration can be used
```

KVM 인스턴스 생성

- KVM 인스턴스 생성 스크립트 작성 후 실행권한 부여 (chmod +x ./~~.sh)

```
#!/bin/sh

if [ -z "$1" ];
then
    echo Specify a virtual-machine name.
    exit 1
fi

sudo virt-install \W
--name $1 \W
--ram 65536 \W      메모리 크기(MB)
--disk path=/var/lib/libvirt/images/$1.img,size=100 \W 기본 디스크 사이즈(GB)
--vcpus 36 \W vcpu 할당 수
--os-variant ubuntu20.04 \W
--network bridge:br0,model=virtio \W bridge 이름 입력(br0)
--graphics none \W
--console pty,target_type=serial \W
--location 'http://ftp.ubuntu.com/ubuntu/dists/focal/main/installer-amd64/' \W
--extra-args 'console=ttyS0,115200n8 serial'
```

- Master / Worker / Bootstrap용 가상머신 각각 다르게 설정하여 생성 (Master x1, Worker x2, Bootstrap x1)

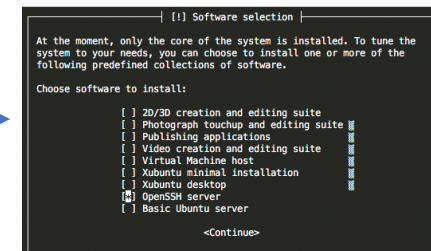
- ex. ./create-vm_master.sh k8s-master

- Hostname : k8s-master / k8s-worker / k8s-bootstrap , Username : ubuntu

- (중요) Software selection 단계에서 OpenSSH server 선택 (스페이스바 이용 선택)

생성된 가상머신은 virsh 명령어를 이용하여 제어 가능 (virsh help 명령어로 옵션 조회)

- ex. 가상머신 정지/시작/재부팅 : virsh destroy/start/reboot k8s-master



KVM 인스턴스 생성

- 각 가상머신 생성 후 `sudo nmap -sP 192.168.x.0/24` 명령어 입력하여 KVM으로 생성된 가상머신 주소 확인 (QEMU virtual NIC)

```
root@server1:~/kvm# sudo nmap -sP 192.168.240.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2023-08-22 17:32 KST
Nmap scan report for _gateway (192.168.240.1)
Host is up (0.00013s latency).
MAC Address: BC:CF:4F:FF:B6:72 (Zyxel Communications)
Nmap scan report for 192.168.240.110
Host is up (0.00017s latency).
MAC Address: 52:54:00:40:EC:A2 (QEMU virtual NIC)
Nmap scan report for 192.168.240.202
Host is up (0.000094s latency).
MAC Address: BA:32:47:C5:02:30 (Unknown)
Nmap scan report for 192.168.240.211
Host is up (0.000030s latency).
MAC Address: 52:54:00:10:7F:74 (QEMU virtual NIC)
Nmap scan report for 192.168.240.212
Host is up (0.000035s latency).
MAC Address: 52:54:00:B0:49:45 (QEMU virtual NIC)
Nmap scan report for 192.168.240.213
Host is up (0.000041s latency).
MAC Address: 52:54:00:90:0F:B7 (QEMU virtual NIC)
Nmap scan report for 192.168.240.214
Host is up (0.000044s latency).
MAC Address: 52:54:00:74:62:9F (QEMU virtual NIC)
Nmap scan report for server1 (192.168.240.201)
Host is up.
Nmap done: 256 IP addresses (8 hosts up) scanned in 2.16 seconds
```

- `virsh list` 명령어로 설치된 가상머신 조회

```
root@server1:~/kvm# virsh list
  Id    Name               State
-----
  7     k8s-worker1        running
  9     k8s-worker2        running
  11    k8s-bootstrap      running
  15    k8s-master          running
  16    dockerbuilder       running
```

Kubernetes 설치를 위한 기본 설정

※ 생성한 모든 노드에 대하여 아래 작업 수행

- 쿠버네티스는 Pod 생성 시 메모리 Swap을 고려하지 않고 필요한 만큼의 리소스를 할당받는 구조로 설계되어, 모든 노드는 Swap 메모리 Off 필요
- 각 가상머신에 접속하여 아래 명령어로 Swap 메모리 비활성화
 - `sudo swapoff`
 - `sudo sed -i '/swap/d' /etc/fstab`
- 각 가상머신 별 IP 설정
 - `/etc/netplan/****.yaml` 파일 수정 등 OS에서 제공하는 방법 이용
- `sudo apt-get update`
- `sudo vi /etc/ssh/sshd_config` 명령어 실행하여 `PermitRootLogin` 항목에 대해 `yes`로 설정, `sudo systemctl restart sshd`

※ 마스터노드에 대하여 아래 작업 수행

- `sudo apt-get update`
- `sudo apt-get install -y apt-transport-https ca-certificates curl`
- `curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg`
- `echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list`

Kubespray 설치 및 설정 (1)

- bootstrap 가상머신으로 접속 (설치 과정에서 입력한 username과 password 입력) 후 root 계정 진입
- ssh-keygen -t rsa -b 2048
- ssh-copy-id root@192.168.xxx.xxx [master / worker 노드 별로 각각 IP 입력]
- git clone -b v2.16.0 <https://github.com/Kubernetes-sigs/kubespray.git>
- cd kubespray
- sudo apt-get update && sudo apt-get install python3 python3-pip
- sudo pip3 install -r requirements.txt
- cp -rfp inventory/sample inventory/testcluster
- vi inventory/testcluster/inventory.ini 명령어 입력 후 파일 내용 작성

```
# ## Configure 'ip' variable to bind kubernetes services on a
# ## different ip than the default iface
# ## We should set etcd_member_name for etcd cluster. The node that is not a etcd member do not need to set the value, or can set the empty string value.
[all]
k8s-master ansible_host=192.168.240.211 ip=192.168.240.211
k8s-worker1 ansible_host=192.168.240.212 ip=192.168.240.212
k8s-worker2 ansible_host=192.168.240.213 ip=192.168.240.213

[kube_control_plane]
k8s-master

[etcd]
k8s-master

[kube_node]
k8s-worker1
k8s-worker2

[calico_rr]

[k8s_cluster:children]
kube_control_plane
kube_node
calico_rr
```

모든 노드 정보(이름 및 IP주소) 입력

마스터노드 이름 입력

마스터노드 이름 입력

워커노드 이름 입력

Kubespray 설치 및 설정 (2)

- `ansible all -m ping -i inventory/testcluster/inventory.ini` 입력하여 모든 노드와 ping 테스트 진행

```
k8s-worker1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
k8s-worker2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
k8s-master | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

- `ansible-playbook -i inventory/testcluster/inventory.ini cluster.yml -b --become-user root -e ansible_ssh_timeout=50 -vvv` 입력하여 Kubernetes 클러스터 설치
- 마스터노드 접속하여 `kubectl get node`

```
root@k8s-master:/home/ubuntu# kubectl get nodes
NAME           STATUS    ROLES                  AGE     VERSION
k8s-master     Ready    control-plane,master   15d     v1.20.7
k8s-worker1    Ready    worker                 15d     v1.20.7
k8s-worker2    Ready    worker                 15d     v1.20.7
```

- 라벨 변경은 `kubectl label node k8s-worker1 node-role.kubernetes.io/worker=` 명령어 입력하여 수행