vagrant

노트북: AWS EKS강의
만든 날짜: 2022-10-07 오전 6:36 **수정한 날짜:** 2022-10-07 오전 7:00
작성자: pro_4701@naver.com

---

1. 초기 플러그인 설치

```
CMD > vagrant plugin install vagrant-hostmanager
CMD > vagrant plugin install vagrant-disksize
```

2. 파일구성 (Vagrantfile 포트 기억 , 60047)
2-1. Vagrantfile

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

  ### Master Node ####

Vagrant.configure("2") do |config|
  config.vm.define "mine-control1" do |config|
    config.vm.box = "ubuntu/focal64"
    config.vm.hostname = "kube-control1"
    config.vm.network "private_network", ip: "192.168.56.47"
    config.vm.network "forwarded_port", guest: 22, host: 60047, auto_correct:
true, id: "ssh"  #호스트 60047로 접속시 가상머신 22번으로  포트포워딩
    config.vm.provision "shell", path: "config.sh"
    config.vm.provision "shell", path: "install.sh"
    config.vm.provider "virtualbox" do |vb|
      vb.name = "mine-control1"
      vb.cpus = 2
      vb.memory = 3072
      if !File.exist?("disk00.vdi")
        vb.customize ["createmedium", "disk", "--filename", "disk00.vdi", "--
size", 10240]
      end
      vb.customize ["storageattach", :id, "--storagectl", "SCSI", "--port", 2,
"--device", 0, "--type", "hdd", "--medium", "disk00.vdi"]
    end
  end
end
```

2-2. config.sh

```
echo 'source <(kubectl completion bash)' >>~/.bashrc
echo 'alias k=kubectl' >>~/.bashrc
echo 'complete -F __start_kubectl k' >>~/.bashrc

# swapoff -a to disable swapping
swapoff -a && sed -i '/ swap / s/^/#/' /etc/fstab

# alias
echo 'source <(kubectl completion bash)' >>~/.bashrc
echo 'alias k=kubectl' >>~/.bashrc
echo 'complete -F __start_kubectl k' >>~/.bashrc

# config DNS
```

```
cat <<EOF > /etc/resolv.conf
nameserver 8.8.8.8 #Google DNS
EOF

# history , 사용자지정
echo 'HISTTIMEFORMAT="%Y/%m/%d %T"' >> ~/.bashrc
source ~/.bashrc

# ssh password Authentication no to yes
sed -i -e 's/PasswordAuthentication no/PasswordAuthentication yes/g'
/etc/ssh/sshd_config
sed -i 's/archive.ubuntu.com/ftp.daum.net/g' /etc/apt/sources.list
sed -i 's/security.ubuntu.com/ftp.daum.net/g' /etc/apt/sources.list
systemctl restart ssh
systemctl start systemd-timesyncd
timedatectl set-timezone UTC
systemctl restart sshd
```

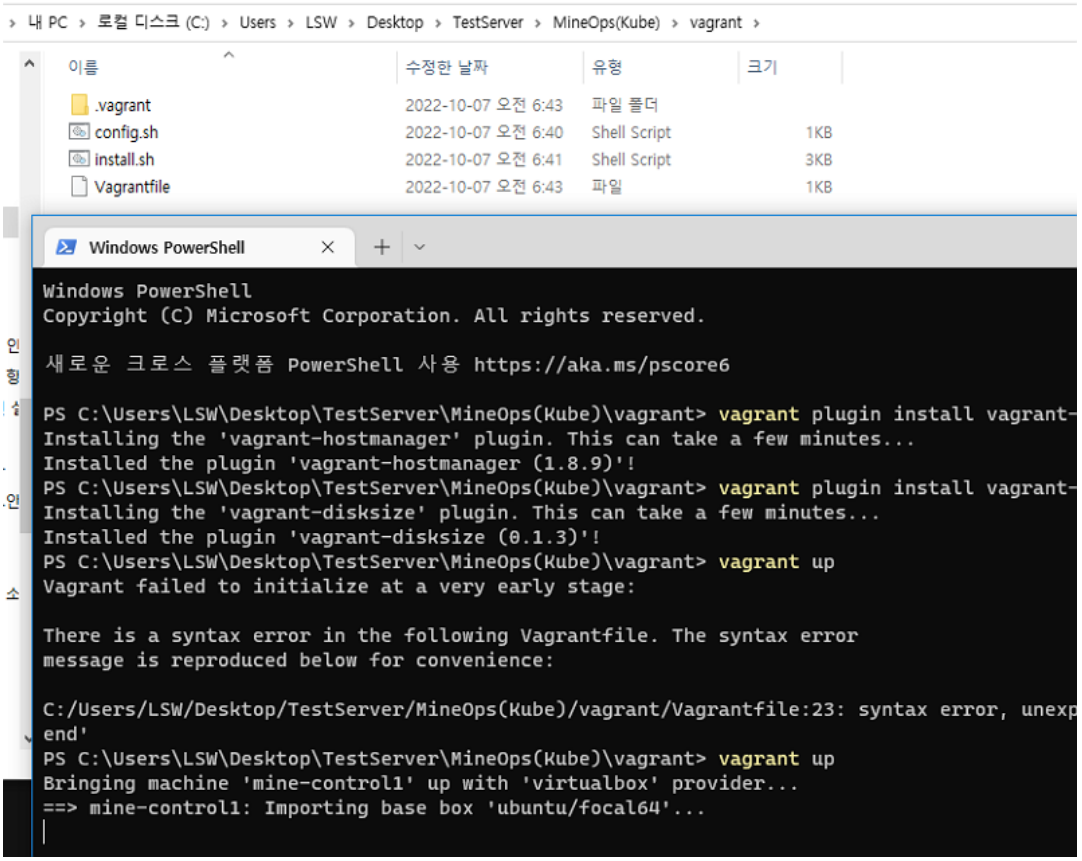
2-3. install.sh

```
sudo apt-get update -y
sudo apt-get install -y curl
sudo apt-get install -y gnupg
sudo apt-get install -y lsb-release
sudo apt-get install -y net-tools
sudo apt-get install -y bind-utils
sudo apt-get install -y vim
sudo apt-get install -y openssh-server -y
sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-
release -y

# 테라폼 설치,  수동설치 한경우 확인용   #  terraform version
curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -
sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com
$(lsb_release -cs) main"
sudo apt-get install -y terraform

# 도커 쿠버네티스 설치
sudo swapoff -a  # SWAP 설정 OFF
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -
o /usr/share/keyrings/docker-archive-keyring.gpg # Docker 추가 Repository 구성
(gpgkey keyring 설치, Repository 추가)
echo \
  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null

# Docker 패키지 설치
sudo apt-get update -y
sudo apt-get install docker-ce docker-ce-cli containerd.io -y

cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF

sudo systemctl daemon-reload
sudo systemctl restart docker.service

# kubeadm, kubelet, kubectl 설치   Kubernetes 추가 Repository 구성(gpgkey
keyring 설치, Repository 추가)
sudo curl -fssL https://packages.cloud.google.com/apt/doc/apt-key.gpg -o
/usr/share/keyrings/kubernetes-archive-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]
https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee
```

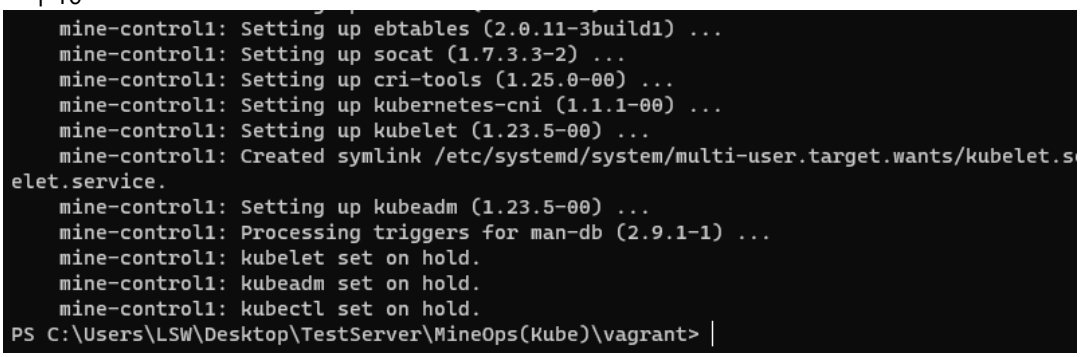
```
/etc/apt/sources.list.d/kubernetes.list

# kubeadm, kubelet, kubectl 설치(버전 지정 및 고정)
sudo apt-get update -y
sudo apt-get install -y kubelet=1.23.5-00 kubeadm=1.23.5-00 kubectl=1.23.5-00
sudo apt-mark hold kubelet kubeadm kubectl
```

2-4.

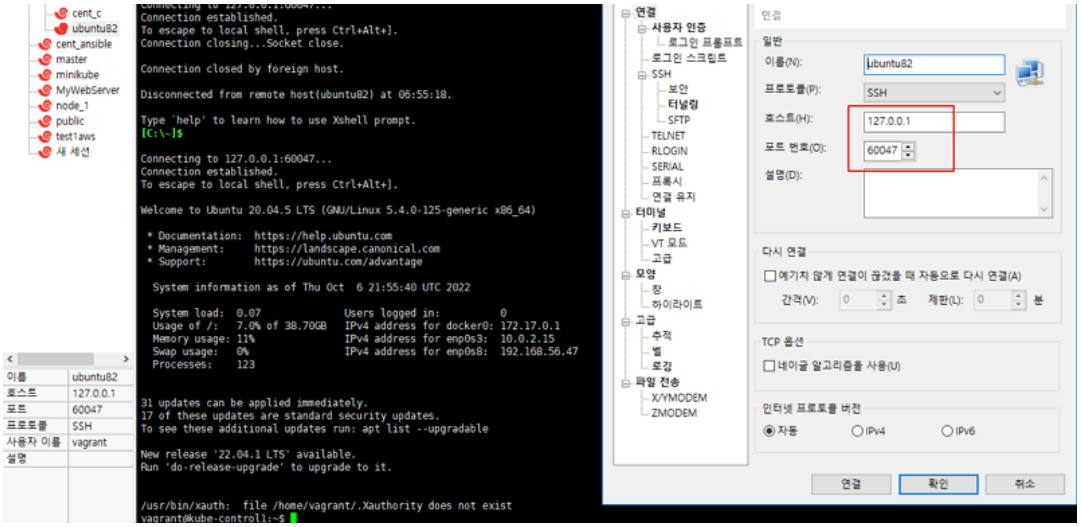
```
CMD > vagrant up
```



3. 설치 완료
- 약 10



4. 접속
- 로컬 또는 사설IP에 60047 포트로 접속 가능



5. aws cli 설치

```
sudo apt install -y unzip


curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install


aws --version

aws configure   # 개인 정보 입력
```

```
vagrant@kube-controll:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
vagrant@kube-controll:~$ aws --version
aws-cli/2.8.1 Python/3.9.11 Linux/5.4.0-125-generic exe/x86_64.ubuntu.20 prompt/off
vagrant@kube-controll:~$ aws configure
AWS Access Key ID [None]:
```

```
aws sts get-caller-identity  # 반드시 아래의 값 출력
```

```
root@jenkins:~/devops_06_03_jenkins# aws sts get-caller-identity
{
    "UserId": "AIDA5643M7CC7W6DPBFVO",
    "Account": "959714228357",
    "Arn": "arn:aws:iam::959714228357:user/aicore0860"
}
root@jenkins:~/devops_06_03_jenkins#
```