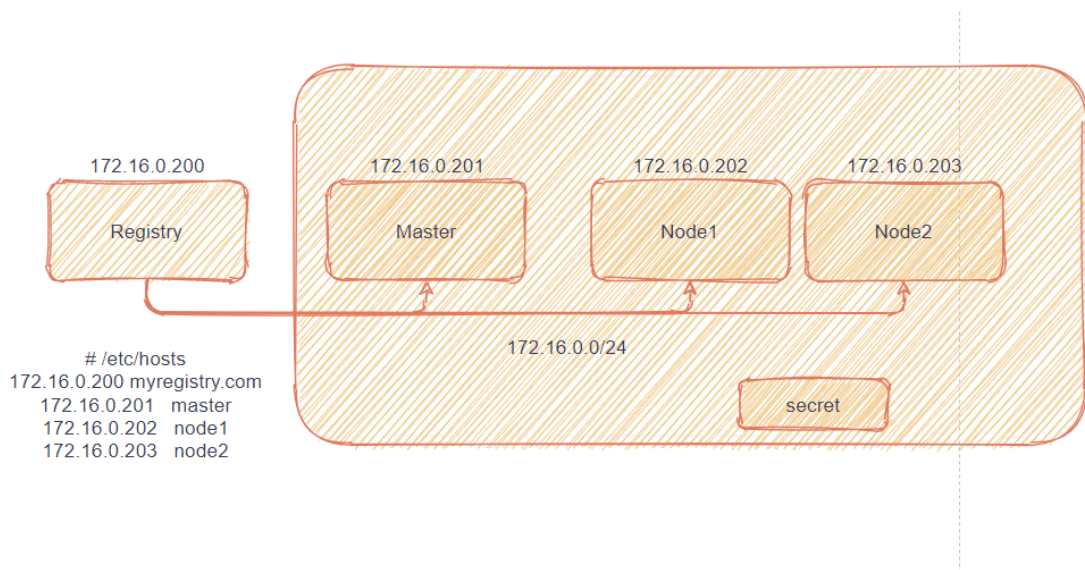


# Install Kubernetes

kubernetes 는 가상 시스템 (VM) 호스트 혹은 물리적 호스트에 설치 할 수 있습니다. 미리 설치할 호스트에 다음의 내용을 준비해 주시기 바랍니다.

## ▼ 준비사항

- 호환 가능한 Debian , Redhat등의 Linux 호스트 3대 (master, node1, node2)
- 2 GB 이상의 RAM 메모리
- 2 Core 이상의 CPU
- 모든 호스트들의 네트워크 연결
- 고유 호스트명 , MAC 주소 그리고 product\_uuid
- Swap 영역 비활성화



## 원격 호스트에서 실행 (myregistry)

```
# On Remote Host
cat << EOF >> /etc/hosts

172.16.0.200 myregistry.com
172.16.0.201 master
172.16.0.202 node1
172.16.0.203 node2
EOF

scp /etc/hosts master:/etc/hosts
scp /etc/hosts node1:/etc/hosts
scp /etc/hosts node2:/etc/hosts

ssh master 'hostnamectl set-hostname master'
ssh node1 'hostnamectl set-hostname node1'
ssh node2 'hostnamectl set-hostname node2'

sed -i '/^#\allow /a\allow 172.16.0.0/24' /etc/chrony.conf

systemctl restart chronyd
```

## 각 노드에서 실행 (Master , Node1, Node2)

```
# disable Service
systemctl disable NetworkManager; systemctl stop NetworkManager
```

```
systemctl disable firewalld ; systemctl stop firewalld
systemctl restart network
```

```
# Set NTP
sed -i 's/^server /\#server/g' /etc/chrony.conf
echo 'server myregistry.com iburst' >> /etc/chrony.conf
systemctl restart chronyd
chronyc sources -v
```

```
(OUTPUT)
210 Number of sources = 1

... Source mode '^' = server, '=' = peer, '#' = local clock.
/.. Source state '*' = current synced, '+' = combined , '-' = not combined,
| / '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||
|| Reachability register (octal) --. | xxxx [ yyyy ] +/- zzzz
|| Log2(Polling interval) --. | | xxxx = adjusted offset,
|| | | yyyy = measured offset,
|| | | zzzz = estimated error.
||
||
MS Name/IP address Stratum Poll Reach LastRx Last sample
=====
^* myregistry.com 3 6 17 5 +7440ns[ +65us] +/- 27ms
```

```
# Install kubeadm

# uninstall docker previous version
yum remove docker \
    docker-client \
    docker-client-latest \
    docker-common \
    docker-latest \
    docker-latest-logrotate \
    docker-logrotate \
    docker-selinux \
    docker-engine-selinux \
    docker-engine

swapoff -a
sed -i 's/^(\.*)$/\1g' /etc/fstab

cat /etc/fstab
swapon -s

yum install wget -y
wget https://download.docker.com/linux/centos/docker-ce.repo -O /etc/yum.repos.d/docker-ce.repo

yum install -y docker-ce

systemctl start docker && systemctl enable docker

cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=0
repo_gpgcheck=0
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kube*
EOF
setenforce 0
sed -i s/SELINUX=enforcing/SELINUX=permissive/ /etc/selinux/config

yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
systemctl enable kubelet && systemctl start kubelet

# lsmod | grep br_netfilter
# modprobe br_netfilter

cat <<EOF > /etc/modules-load.d/k8s.conf
br_netfilter
```

EOF

```
cat <<EOF > /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sysctl --system
```

```
systemctl daemon-reload
systemctl restart kubelet
```

```
# vi /etc/sysctl.conf
# net.ipv4.ip_forward = 1
# sysctl -p
# sysctl -a | grep ip_forward
```

```
# Change Cgroup Driver
sed -i 'ExecStart/ s/$/ --exec-opt native.cgroupdriver=systemd/g' \
/usr/lib/systemd/system/docker.service
```

```
systemctl daemon-reload
systemctl restart docker
docker info | grep Cgroup
```

(OUTPUT)

```
Cgroup Driver: systemd
Cgroup Version: 1
```



kubernetes 기본 cgroup은 systemd 이며 docker 기본 cgroup은 cgroupfs입니다.  
kubernetes 1.22 이후 버전에서는 cgroup이 일치하지 않으면 충돌이 발생하여 설치가 실패 합니다.

## Initialize Control Plane by kubeadm

```
# master node에서만 실행합니다.
kubeadm init --pod-network-cidr=192.168.0.0/16 --apiserver-advertise-address=172.16.0.201
```

## Join worker node to cluster

```
# Worker Node 에서만 실행합니다.
# master 노드에서 출력된 마지막 라인을 복사해서 실행합니다.
kubeadm join --token <token> <master-ip>:<master-port> --discovery-token-ca-cert-hash sha256:<hash>
```

## Install kubectl to remote host

```
# On Remote Host myregistry 에서만 실행합니다.
mkdir ~/.kube
scp master:/etc/kubernetes/admin.conf ~/.kube/config

curl -LO https://dl.k8s.io/release/\
$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl

# wget https://storage.googleapis.com/kubernetes-release/release/v1.22.0/bin/linux/amd64/kubectl

chmod +x kubectl
mv kubectl /usr/local/bin/

kubectl get nodes
```

## Install Network Plugin - ProjectCalico

```
# Network Plugin - Calico
kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml

# kubectl create -f https://docs.projectcalico.org/manifests/tigera-operator.yaml
# kubectl create -f https://docs.projectcalico.org/manifests/custom-resources.yaml
watch kubectl get pods -n calico-system

# kubectl taint nodes --all node-role.kubernetes.io/master-

# install calicoctl
# curl -O -L https://github.com/projectcalico/calicoctl/releases/download/v3.17.1/calicoctl

Ref:
https://docs.projectcalico.org/getting-started/kubernetes/quickstart
```

## Create New Namespace for default NS

```
kubectl create ns demo
kubectl config set-context --current --namespace demo
```



처음 설치 후 자동으로 생성된 토큰은 만료 되기 전까지 계속 사용할 수 있지만, 만료된 이후에는 새로운 토큰을 만들어서 worker node를 추가할 수 있습니다. 다음과 같이 토큰을 새로 생성 할 수 있습니다.

```
# create new token and hash value
kubeadm token list
kubeadm token create
openssl x509 -pubkey -in /etc/kubernetes/pki/ca.crt | \
openssl rsa -pubin -outform der 2>/dev/null | openssl dgst -sha256 -hex | sed 's/^.* //'
```

## Node IP 강제 지정

이전 버전의 버그로 kubeadm으로 설치한 node의 IP가 잘못 지정이 될수가 있습니다. 이때는 아래의 내용을 참고하여 수정할 수 있습니다.

```
# ubuntu 인 경우 /etc/systemd/system/kubelet.service.d/10-kubeadm.conf
# CentOS 인 경우 /usr/lib/systemd/system/kubelet.service.d/10-kubeadm.conf
# ExecStart=/usr/bin/kubelet $KUBELET_KUBECONFIG_ARGS $KUBELET_CONFIG_ARGS $KUBELET_KUBEADM_ARGS $KUBELET_EXTRA_ARGS --node-ip [NODE_IP]
systemctl daemon-reload
systemctl restart kubelet
```