1. [All nodes] Ubuntu Install

❶ 소프트웨어 업데이트 비활성화

❷ Ubuntu 저장소 설정

❸ 네트워크 툴 설치

```
apt install net-tools
```

❹ 방화벽 설치와 서비스 비활성화

```
apt install firewalld
systemctl stop firewalld

systemctl disable firewalld
```

❺ 텔넷 설치와 서비스 활성화

```
apt-get install telnetd xinetd -y
```

```
nano /etc/xinetd.d/telnet
service telnet
{
        disable             = no
        flags               = REUSE
        socket_type         = stream
        wait                 = no
        user                 = root
        server                = /usr/sbin/in.telnetd
        log_on_failure    += USERID
}
systemctl restart xinetd

systemctl enable xinetd

systemctl status xinetd
```

❻ sudo su – root

```
****

passwd
```

2. [All nodes] Docker Install

❶ Docker 설치에 필요한 툴 설치

```
apt-get install ca-certificates curl gnupg lsb-release -y
```

❷ 도커 GPG Key 등록 (도커 인증서 저장)

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

❸ Docker Repository URL 등록

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null
```

❹ 패키지 업데이트

```
apt-get update
```

❺ Docker Engine 설치

```
apt-get install docker-ce docker-ce-cli containerd.io
```

❻ Docker 활성화

```
systemctl enable docker
systemctl start docker

docker version
```

3. [All nodes] Before K8s Install

❶ Swap 비활성화

```
swapoff -a && sed -i '/swap/s/^/#/' /etc/fstab
```

❷ Master와 Workernode 브릿지(스위치) 연결

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

❸ 방화벽 비활성화

```
systemctl stop firewalld
systemctl disable firewalld
```

4. [All nodes] Kubeadm, kubect, kubelet Install

❶ K8S 설치에 필요한 툴 설치

```
apt-get install apt-transport-https ca-certificates
```

❷ K8S GPG Key 등록 (K8S 인증서 저장)

```
curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

❸ K8S Repository URL 등록

```
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" |
tee /etc/apt/sources.list.d/kubernetes.list
```

❹ kubelet kubeadm kubectl 설치

```
apt-get update

apt-get install kubelet kubeadm kubectl

apt-mark hold kubelet kubeadm kubectl
```

❺ json 을 이용한 cgroupdriver 지정

```
nano /etc/docker/daemon.json
{
    "exec-opts": ["native.cgroupdriver=systemd"]
 }
```

❽ K8S 서비스 활성화

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

❾ 시스템 재부팅

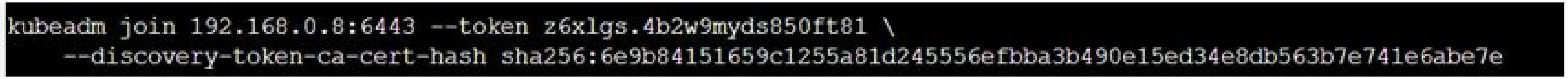
```
reboot
```

5. [Master node] Kubeadmn을 이용한 cluster 구성

❶ Control plane node 초기화(initializing)

kubeadm init

kubeadm join~~copy



```
kubeadm join 192.168.0.8:6443 --token z6xlgs.4b2w9myds850ft81 \
    --discovery-token-ca-cert-hash sha256:6e9b84151659c1255a81d245556efbba3b490e15ed34e8db563b7e741e6abe7e
```

❷ CNI 구성

kubectl apply -f https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')

❸ 클러스터 구성에 필요한 환경 설정 파일 생성

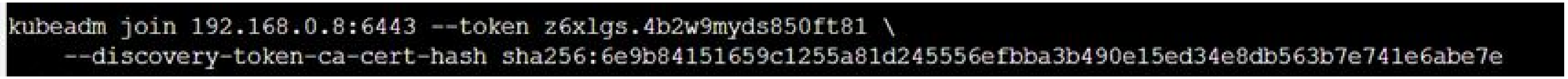
export KUBECONFIG=/etc/kubernetes/admin.conf

❹ 클러스터 노드 확인

kubectl get nodes

6. [Worker Node] Master node와 Worker node join

kubeadm join~~copy



```
kubeadm join 192.168.0.8:6443 --token z6xlgs.4b2w9myds850ft81 \
    --discovery-token-ca-cert-hash sha256:6e9b84151659c1255a81d245556efbba3b490e15ed34e8db563b7e741e6abe7e
```

7. [Master node] Cluster 설정 확인

kubectl get nodes

8. [All nodes] 글자 자동 완성 기능 활성화

source <(kubectl completion bash)

source <(kubeadm completion bash)

echo 'source <(kubectl completion bash)' >>~/.bashrc

echo 'source <(kubeadm completion bash)' >>~/.bashrc