

Kubernetes 1.24 + cri-docker Installation (kubeadm)

Kubernetes

18.04

Controll Node (node01) , Worker Node (node02 ,node03) 3EA

root 가 user(worker) sudo

(Pre-Option)

```
sudo kill -9 $(lsof -t /var/lib/dpkg/lock)
sudo kill -9 $(lsof -t /var/lib/apt/lists/lock)
sudo kill -9 $(lsof -t /var/cache/apt/archives/lock)
sudo rm /var/lib/apt/lists/lock
sudo rm /var/cache/apt/archives/lock
sudo rm /var/lib/dpkg/lock
sudo dpkg --configure -a
```

(Require) SWAPOFF

```
# SWAP-On 가
sudo swapoff /swap.img
sudo sed -i -e '/swap.img/d' /etc/fstab
```

() Container Runtime Install

k8s 1.24 (2022/05)

k8s dockershim

cri-docker 가 k8s

Pre-Setting - All node

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo systemctl enable --now docker && sudo systemctl status
docker --no-pager
sudo usermod -aG docker worker
```

```
sudo docker container ls
```

```
# cri-docker Install
```

```
VER=$(curl -s https://api.github.com/repos/Mirantis/cri-dockerd/releases/latest|grep tag_name | cut -d '"' -f 4|sed 's/v//g')
echo $VER
wget https://github.com/Mirantis/cri-dockerd/releases/download/v${VER}/cri-dockerd-${VER}.amd64.tgz
tar xvf cri-dockerd-${VER}.amd64.tgz
sudo mv cri-dockerd/cri-dockerd /usr/local/bin/
```

```
# cri-docker Version Check
```

```
cri-dockerd --version
```

```
wget
```

```
https://raw.githubusercontent.com/Mirantis/cri-dockerd/master/packaging/systemd/cri-docker.service
```

```
wget
```

```
https://raw.githubusercontent.com/Mirantis/cri-dockerd/master/packaging/systemd/cri-docker.socket
```

```
sudo mv cri-docker.socket cri-docker.service /etc/systemd/system/
```

```
sudo sed -i -e 's,/usr/bin/cri-dockerd,/usr/local/bin/cri-dockerd,' /etc/systemd/system/cri-docker.service
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable cri-docker.service
```

```
sudo systemctl enable --now cri-docker.socket
```

```
# cri-docker Active Check
```

```
sudo systemctl restart docker && sudo systemctl restart cri-docker
```

```
sudo systemctl status cri-docker.socket --no-pager
```

```
# Docker cgroup Change Require to Systemd
```

```
sudo mkdir /etc/docker
```

```
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 restart docker && sudo systemctl restart cri-
docker
sudo docker info | grep Cgroup
```

```
# Kernel Forwarding
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
br_netfilter
EOF
```

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

```
sudo sysctl --system
```

Packages Install - All node

```
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates
curl
sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-
keyring.gpg
https://packages.cloud.google.com/apt/doc/apt-key.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
```

```
# Update
sudo apt-get update
```

```
# k8s
```

```
sudo apt-get install -y kubelet kubeadm kubectl
```

```
#  
kubectl version --short
```

```
Client Version: v1.24.3  
Kustomize Version: v4.5.4
```

```
#  
sudo apt-mark hold kubelet kubeadm kubectl
```

k8s Init - Controller Node (Node01)

```
# Controller Node  
sudo kubeadm config images pull --cri-socket unix:///run/cri-dockerd.sock
```

```
sudo kubeadm init --ignore-preflight-errors=all --pod-network-cidr=192.168.0.0/16 --apiserver-advertise-address=211.115.207.136 --cri-socket /var/run/cri-dockerd.sock  
## --apiserver-advertise-address=203.248.23.161 -> Controller IP.
```

```
# join Command  
kubeadm join 211.115.207.136:6443 --token r8gfco.6s7f60dns4vgwcc0 \\\n--discovery-token-ca-cert-hash sha256:7b0f82be076748e67f8615eab0b86a61317bac397f94b2921810231ab14afdcc
```

```
# kubeadm root 가  
mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
# Token  
kubeadm token create --print-join-command
```

```
# ( Ready , Running )  
kubectl get nodes -o wide  
kubectl get pod -A
```

```
# Worker Node      kubectl      admin.conf
sudo      scp      /etc/kubernetes/admin.conf
worker@node02:/home/worker/admin.conf
sudo      scp      /etc/kubernetes/admin.conf
worker@node03:/home/worker/admin.conf
```

Node Join - Worker Node (Node02 , Node03)

```
# Worker Node Join. --cri-docker sorket      가      .
init      .
```

```
sudo kubeadm join --token <token> <controlplane-
host>:<controlplane-port> --ignore-preflight-errors=all --cri-
socket unix:///var/run/cri-dockerd.sock
# sudo kubeadm join [Controller IP]:6443 --token
3nvrgw.33k750dq9klm5omi --discovery-token-ca-cert-hash
sha256:b10158bcea37aae0e92ed6b68b4dd1e8213623cc7d406e77eef55fe
6196fe346 --cri-socket /var/run/cri-dockerd.sock
```

```
# Join
sudo kubeadm join [Controller IP]:6443 --token
r8gfco.6s7f60dns4vgwcc0 \
--discovery-token-ca-cert-hash
sha256:7b0f82be076748e67f8615eab0b86a61317bac397f94b2921810231
ab14afdcc --cri-socket /var/run/cri-dockerd.sock
```

```
# Check
systemctl status kubelet
```

```
# worker      kubectl      가
cd ~
mkdir -p $HOME/.kube
sudo cp -i ./admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

```
가      node      kubectl get nodes -o wide
```

Pod Network Install - Controller Node (Node01)

```
#      CNI Plugin      CoreDNS 가 Pending
      kube-system-proxy      .
```

```

# CNI Mode
# Check - kube-proxy . kube-proxy Node
3
kubectl get pod -A
kubectl logs -f pod/kube-proxy-[name] -n kube-system

# Log - Linux iptables Mode
kubectl logs -f pod/kube-proxy-6dqp8 -n kube-system | grep
mode
I0805 06:32:48.444077 1 server_others.go:578] "Unknown
proxy mode, assuming iptables proxy" proxyMode=""
I0805 06:32:48.469028 1 server_others.go:213] "kube-
proxy running in dual-stack mod
" ipFamily=IPv4

## Pod Network Plugin Install
, CKA Callico
Plugin
Calico Network IPIP , VXLAN IPVS

kubectl create -f
https://projectcalico.docs.tigera.io/manifests/tigera-operator
.yaml
curl
https://projectcalico.docs.tigera.io/manifests/custom-resource
s.yaml -O
kubectl create -f custom-resources.yaml
kubectl get pods -A

## Control-Plane ( Controller ) Node Taint (
) calico-controller-pod가 Pending

## Taint Check
kubectl describe node node01 | grep Taints

kubectl taint nodes --all node-role.kubernetes.io/master-
kubectl taint nodes node01 node-role.kubernetes.io/control-
plane:NoSchedule-

## PoD check
kubectl get pod -A

```

```
#kubectl describe pods/calico-kube-controllers-657d56796-xxxxx  
-n calico-system
```

```
## Calicoctl Status
```

```
cd /usr/local/bin
```

```
sudo curl -L  
https://github.com/projectcalico/calico/releases/download/v3.2  
3.3/calicoctl-linux-amd64 -o calicoctl  
sudo chmod +x calicoctl
```

```
## CNI Type Check
```

```
calicoctl get ippool -o wide
```

```
## Block Check
```

```
sudo calicoctl ipam show --show-blocks
```

```
## BGP Protocol Check
```

```
sudo calicoctl node status
```

```
## Node Endpoint Check
```

```
calicoctl get workloadendpoint -A
```

Rejoin or Reset

(Trouble)

```
# All Node
```

```
sudo systemctl stop kubelet
```

```
sudo kubeadm reset -f --cri-socket /var/run/cri-dockerd.sock
```

```
sudo rm -rf ~/.kube
```

```
sudo rm -rf /root/.kube
```

```
sudo rm -rf /var/lib/etcd
```

```
sudo rm -rf /etc/kubernetes
```

Calico Network

```
# ALL Node
```

```
kubectl delete -f custom-resources.yaml
```

```
kubectl delete -f tigera-operator.yaml
```

```
sudo rm -rf /var/run/calico/
```

```
sudo rm -rf /var/lib/calico/
```

```
sudo rm -rf /etc/cni/net.d/  
sudo rm -rf /var/lib/cni/  
sudo rm -rf /opt/cni  
sudo reboot
```