# yum install docker -y

# systemctl enable docker --now

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

# vim /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

exclude=kubelet kubeadm kubectl

# yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

# systemctl enable --now kubelet

# kubeadm config images pull

# kubeadm init --pod-network-cidr=10.240.0.0/16

# docker info | grep -i cgroup

Cgroup Driver: cgroupfs

# systemctl restart docker

# cat /etc/docker/daemon.json

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

# docker info | grep -i cgroup

Cgroup Driver: systemd

# yum install iproute-tc

# kubeadm init --pod-network-cidr=10.240.0.0/16 --ignore-preflight-errors=NumCPU --ignore-preflight-errors=Mem

# mkdir -p $HOME/.kube

# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

# sudo chown $(id -u):$(id -g) $HOME/.kube/config

# kubectl get pods --all-namespaces

# echo 3 > /proc/sys/vm/drop\_caches

# kubeadm token create --print-join-command

**Setup Worker Node:**

# yum install docker -y

# systemctl enable docker --now

<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

# vim /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

exclude=kubelet kubeadm kubectl

# yum install -y kubectl kubelet kubeadm --disableexcludes=kubernetes

# systemctl enable kubelet --now

# docker info | grep -i cgroup

Cgroup Driver: cgroupfs

# cat /etc/docker/daemon.json

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

# systemctl restart docker

# docker info | grep -i cgroup

Cgroup Driver: systemd

# yum install iproute-tc

# vim /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-ip6tables = 1

net.bridge.bridge-nf-call-iptables = 1

# sysctl --system

# kubeadm join 172.31.40.209:6443 --token 3joamn.y8asi8hw2jb41xx5 --discovery-token-ca-cert-hash sha256:86dd3714b1168b8de0abcb63300a458baf838860ff9c6f53bd85707853ff4db7

# systemctl status kubelet -l

# echo 3 > /proc/sys/vm/drop\_caches

**Set at Master Node:**

<https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/network-plugins/#cni>

<https://kubernetes.io/docs/concepts/cluster-administration/addons/>

# kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

# kubectl get pods --all-namespaces

# kubectl get nodes

**TroubleShooting of CoreDNS pods:**

**At master node:**

# kubectl get pods --all-namespaces

# kubectl logs coredns-74ff55c5b-c45jz -n kube-system

# kubectl edit configmap kube-flannel-cfg -n kube-system

net-conf.json: |

{

"Network": "10.240.0.0/16",

"Backend": {

"Type": "vxlan"

}

}

# kubectl delete pods -l app=flannel -n kube-system

# cat /var/run/flannel/subnet.env

FLANNEL\_NETWORK=10.240.0.0/16

FLANNEL\_SUBNET=10.240.0.1/24

FLANNEL\_MTU=8951

FLANNEL\_IPMASQ=true

# kubectl get pods -l k8s-app=kube-dns -n kube-system

<https://github.com/coreos/flannel-cni/tree/v0.3.0#readme>