Kubernetes Setup using Kubeadm

~Start - Execute the below commands in both Master/worker nodes

Login to both instances execute the below commands:

```
sudo apt-get update -y  && sudo apt-get install apt-transport-https -y
```

Change to root user

```
sudo su -
sudo curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -

cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF

sudo apt-get update
```

#Disable swap memory for better performance

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

Enable IP tables

#We need to enable IT tables for pod to pod communication.

```
modprobe br_netfilter
sysctl -p
sudo sysctl net.bridge.bridge-nf-call-iptables=1
```

Install Docker on both Master and Worker nodes

```
apt-get install docker.io -y
```

Add ubuntu user to Docker group

```
usermod -aG docker ubuntu
systemctl restart docker
systemctl enable docker.service
```

Type exit to come out of root user.

Install Kubernetes Modules

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

sudo systemctl daemon-reload
sudo systemctl start kubelet
sudo systemctl enable kubelet.service
sudo systemctl status docker
```

#End - Execute the above commands in both Master/worker nodes##########

```
cd /etc/docker/
vi daemon.json

add this below commands:-
{
   "exec-opts": ["native.cgroupdriver=systemd"]
}
```

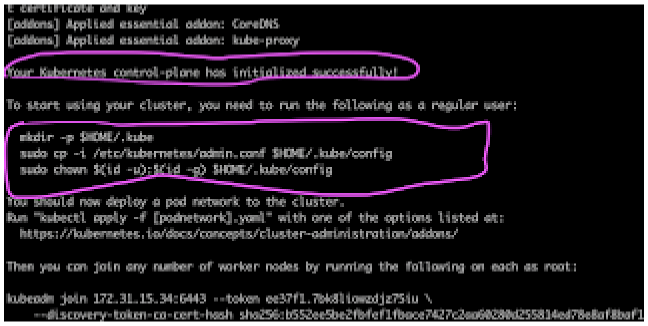
sudo systemctl daemon-reload
sudo systemctl restart docker
sudo systemctl restart kubelet

Wait for Sometime, It will take some time

Initialize Kubeadm on Master Node(only on Master Node)

#Execute the below command as root user to initialize Kubernetes Master node.

```
sudo su -
kubeadm init
```



```
t certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

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

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.15.34:6443 --token ee37f1.7bk8liowzdjz75iu \
    --discovery-token-ca-cert-hash sha256:b552ee5be2fbfef1fbace7427c2ae60280d255814ed78e8af8baf1
```

Make sure you see the above message to confirm master node is up.

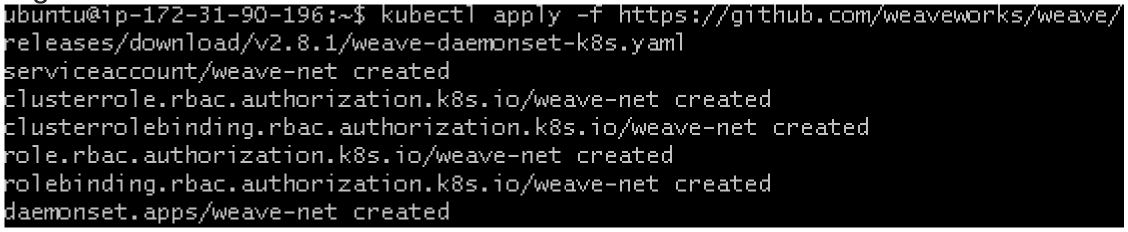
#Now type exit to exit from root user and execute below commands as normal user

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

Installing the Weave Net Add-On

```
kubectl apply -f https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml
```

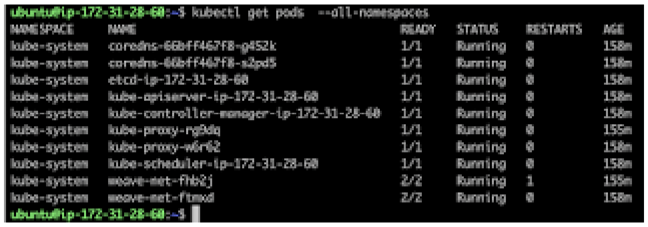
It make take a few mins to execute the above command and show show the below message.



```
ubuntu@ip-172-31-90-196:~$ kubectl apply -f https://github.com/weaveworks/weave/
releases/download/v2.8.1/weave-daemonset-k8s.yaml
serviceaccount/weave-net created
clusterrole.rbac.authorization.k8s.io/weave-net created
clusterrolebinding.rbac.authorization.k8s.io/weave-net created
role.rbac.authorization.k8s.io/weave-net created
rolebinding.rbac.authorization.k8s.io/weave-net created
daemonset.apps/weave-net created
```

Now execute the below command to see the pods.

kubectl get pods --all-namespaces



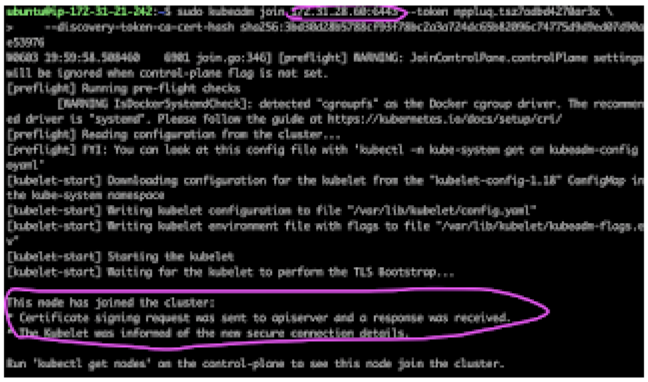
```
ubuntu@ip-172-31-28-60:~$ kubectl get pods  --all-namespaces
NAMESPACE     NAME                                       READY   STATUS    RESTARTS   AGE
kube-system   coredns-66bff467f8-g452k                   1/1     Running   0          158m
kube-system   coredns-66bff467f8-s2pd5                   1/1     Running   0          158m
kube-system   etcd-ip-172-31-28-60                       1/1     Running   0          158m
kube-system   kube-apiserver-ip-172-31-28-60             1/1     Running   0          158m
kube-system   kube-controller-manager-ip-172-31-28-60    1/1     Running   0          158m
kube-system   kube-proxy-rg9dq                           1/1     Running   0          155m
kube-system   kube-proxy-w6r62                           1/1     Running   0          158m
kube-system   kube-scheduler-ip-172-31-28-60             1/1     Running   0          158m
kube-system   weave-net-fhb2j                            2/2     Running   1          155m
kube-system   weave-net-ftmxd                            2/2     Running   0          158m
ubuntu@ip-172-31-28-60:~$
```

Now login to Worker Node

Join worker node to Master Node

The below command will join worker node to master node, execute this a normal user by putting sudo before:

sudo kubeadm join <master_node_ip>:6443 --token xrvked.s0n9771cd9x8a9oc \
    --discovery-token-ca-cert-hash sha256:288084720b5aad132787665cb73b9c530763cd1cba10e12574b4e97452137b4a



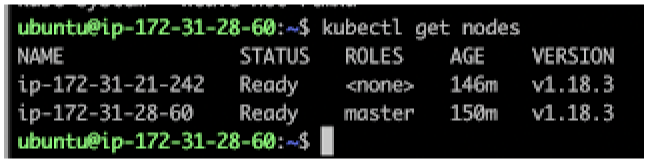
```
ubuntu@ip-172-31-21-242:~$ sudo kubeadm join 172.31.28.60:6443 --token mppluq.tsz7odbd4270ar3x \
>     --discovery-token-ca-cert-hash sha256:3b0383d2805788cf93f78bc2a3a724dc65b82096c74775d9d9ed07d90e
e53976
W0603 19:59:58.508460    6901 join.go:346] [preflight] WARNING: JoinControlPane.controlPlane settings
will be ignored when control-plane flag is not set.
[preflight] Running pre-flight checks
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommen
ed driver is "systemd". Please follow the guide at https://kubernetes.io/docs/setup/cri/
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config
-oyaml'
[kubelet-start] Downloading configuration for the kubelet from the "kubelet-config-1.18" ConfigMap in
the kube-system namespace
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.e
v"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

Go to Master and type the below command

kubectl get nodes
the above command should display both Master and worker nodes.



```
ubuntu@ip-172-31-28-60:~$ kubectl get nodes
NAME               STATUS   ROLES    AGE    VERSION
ip-172-31-21-242   Ready    <none>   146m   v1.18.3
ip-172-31-28-60    Ready    master   150m   v1.18.3
ubuntu@ip-172-31-28-60:~$
```

It means Kubernetes Cluster - both Master and worker nodes are setup successfully and up and running!!!