Installation of Kubernetes

(Ubuntu 18.04)

Pre-requisite Of Kubernetes Installation

- 2 GB or more of RAM per machine
- 2 CPUs or more.
- Swap disabled.
- Selinux policy should be disabled.
- Docker Should be installed on system.

Steps for Installation:

Master Node:

- Login should be root user
- Update package repository so system should be updated with latest package

```
apt-get update
```

- Curl Package Install it will be needed to use Kuberenetes apt repository

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

- Download the Google Cloud Public Signing Key

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

- Add the Kubernetes apt repository

```
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 the apt package after adding Kubernetes apt repository

```
apt-get update
```

- Installed the package of Kubectl Kubeadm kubelete

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

- Kubelet is restarting every few seconds as it waits in crash loop for Kubeadm to tell it what to do hence we hold the version due to which they are not activated just installed it on the system.

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

- Make directory for Docker for installation

```
mkdir /etc/docker
```

- Configure the Docker daemon, in particular to use systemd for the management of the container’s cgroups.

```
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
```

- Install docker package

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

- Enable docker Service

```
systemctl enable docker
```

- Reload deamon service

```
systemctl daemon-reload
```

- Restart Docker Service

```
systemctl restart docker
```

- first runs a series of prechecks to ensure that the machine is ready to run Kubernetes. These prechecks expose warnings and exit on errors. kubeadm init then downloads and installs the cluster control plane components.

```
kubeadm init
```

- After Initialized the Kubeadm in prechecks we get one join node link we need to copy that link on notepad and use it when we configure the Node.

```
kubeadm join 172.31.16.135:6443 --token
sr4fn3.8448hosdjhkzzo3b \

        --discovery-token-ca-cert-hash
sha256:90bdb4e3a12fdbf907c766f41959537ee80e80a314ab7f
27b7965c058e37d88f
```

- To make kubectl work for your non-root user, run these commands

```
mkdir -p $HOME/.kube

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

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

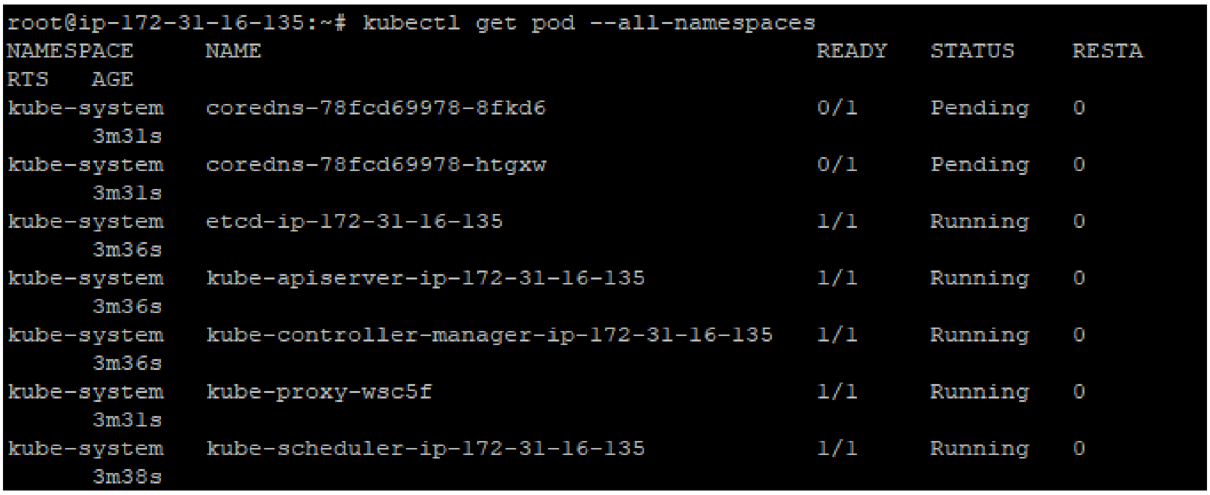
- To check nod has been active or not

```
kubectl get node
```

- To check all pod has running correctly or not

```
kubectl get pod --all-namespaces
```

- Here we observed Core DNS Pod has been not running its showing pending due to network pluggin is not installed in system.



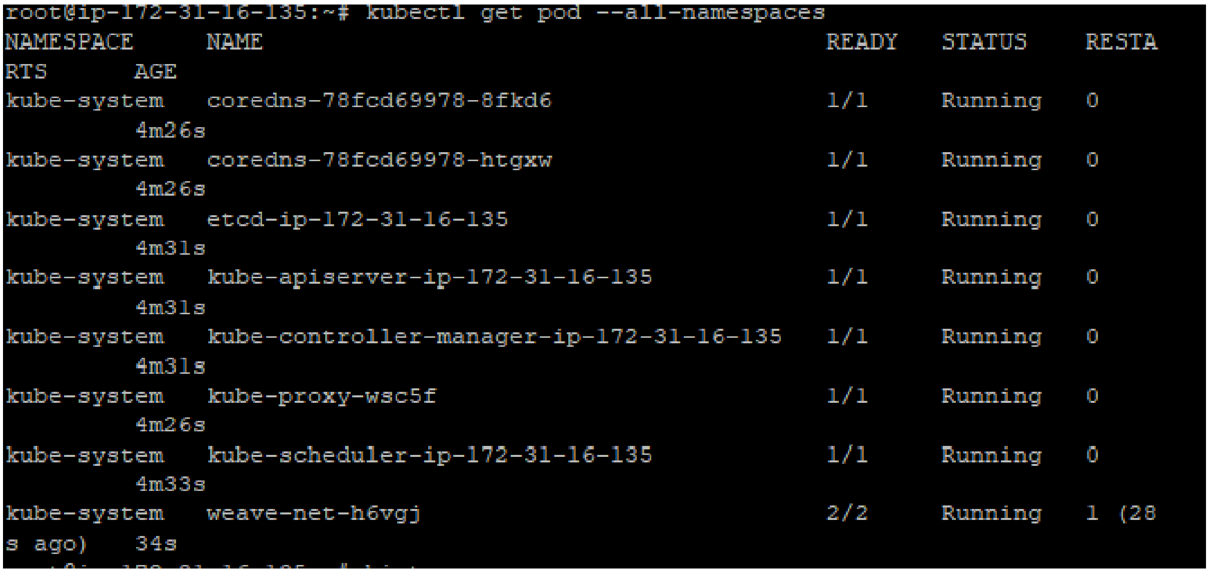
```
root@ip-172-31-16-135:~# kubectl get pod --all-namespaces
NAMESPACE     NAME                                       READY   STATUS    RESTA
RTS   AGE
kube-system   coredns-78fcd69978-8fkd6                   0/1     Pending   0
      3m31s
kube-system   coredns-78fcd69978-htgxw                   0/1     Pending   0
      3m31s
kube-system   etcd-ip-172-31-16-135                      1/1     Running   0
      3m36s
kube-system   kube-apiserver-ip-172-31-16-135            1/1     Running   0
      3m36s
kube-system   kube-controller-manager-ip-172-31-16-135   1/1     Running   0
      3m36s
kube-system   kube-proxy-wsc5f                           1/1     Running   0
      3m31s
kube-system   kube-scheduler-ip-172-31-16-135            1/1     Running   0
      3m38s
```

- Pluggin installed using below command

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

- After that all pods are running please confirmed using below command

```
kubectl get pod --all-namespaces
```



```
root@ip-172-31-16-135:~# kubectl get pod --all-namespaces
NAMESPACE     NAME                                       READY   STATUS    RESTA
RTS       AGE
kube-system   coredns-78fcd69978-8fkd6                   1/1     Running   0
          4m26s
kube-system   coredns-78fcd69978-htgxw                   1/1     Running   0
          4m26s
kube-system   etcd-ip-172-31-16-135                      1/1     Running   0
          4m31s
kube-system   kube-apiserver-ip-172-31-16-135            1/1     Running   0
          4m31s
kube-system   kube-controller-manager-ip-172-31-16-135   1/1     Running   0
          4m31s
kube-system   kube-proxy-wsc5f                           1/1     Running   0
          4m26s
kube-system   kube-scheduler-ip-172-31-16-135            1/1     Running   0
          4m33s
kube-system   weave-net-h6vgj                            2/2     Running   1 (28
s ago)    34s
```

- After done the all steps our Master Node of the Kubernetes is Ready.
- Now we are configure our Nodes using below steps.

Node 1:

- Login Should be root user
- Update package repository so system should be updated with latest package

```
apt-get update
```

- Curl Package Install it will be needed to use Kuberenetes apt repository

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

- Download the Google Cloud Public Signing Key

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

- Add the Kubernetes apt repository

```
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 the apt package after adding Kubernetes apt repository

```
apt-get update
```

- Installed the package of Kubectl Kubeadm kubelete

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

- Kubelet is restarting every few seconds as it waits in crash loop for Kubeadm to tell it what to do hence we hold the version due to which they are not activated just installed it on the system.

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

- Make directory for Docker for installation

```
mkdir /etc/docker
```

- Configure the Docker daemon, in particular to use systemd for the management of the container’s cgroups.

```
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
```

- Install docker package

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

- Enable docker Service

```
systemctl enable docker
```

- Reload deamon service

```
systemctl daemon-reload
```

- Restart Docker Service

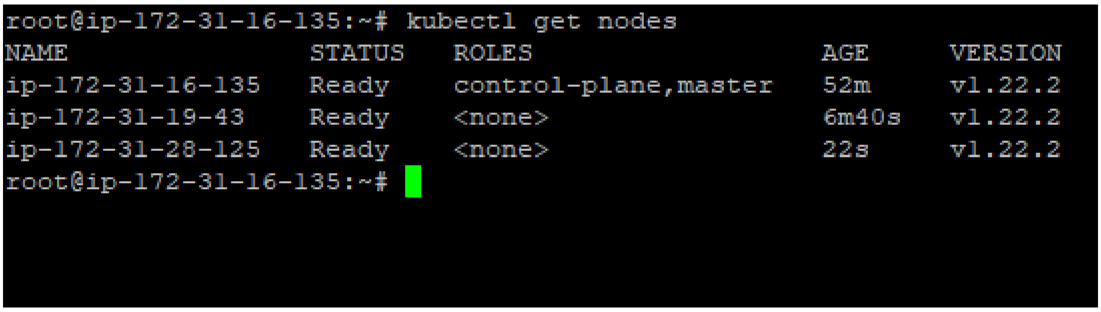
```
systemctl restart docker
```

- After the Docker Service Enable we need to use Node Join link which are we save on notepad just copy and paste it as same that link on node one then your node has join to the cluster.

```
kubeadm join 172.31.16.135:6443 --token
sr4fn3.8448hosdjhkzzo3b        --discovery-token-ca-cert-hash
sha256:90bdb4e3a12fdbf907c766f41959537ee80e80a314ab7f
27b7965c058e37d88f
```

- Same Above Node1 Step should be performed on Node 2 then check on Master Node using below command we get this output then our Kubernetes Installation has been done Successfully.
- Performed below command on Master node

```
kubectl get nodes
```



```
root@ip-172-31-16-135:~# kubectl get nodes
NAME                STATUS   ROLES                  AGE     VERSION
ip-172-31-16-135    Ready    control-plane,master   52m     v1.22.2
ip-172-31-19-43     Ready    <none>                 6m40s   v1.22.2
ip-172-31-28-125    Ready    <none>                 22s     v1.22.2
root@ip-172-31-16-135:~#
```