Author : Raghavendra Pothula

Kubernetes Installation Instructions using kubeadm on Ubuntu 20.04

Note: process is similar to both Master and Node until the installation of kubeadm;

Steps To Disable Swap

```
#  sudo apt-get update && sudo apt-get upgrade
```

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

```
# sudo swapoff -a
```

```
# sudo reboot
```

Steps Install Docker

```
#  sudo apt install -y curl gnupg2 software-properties-common apt-transport-https ca-certificates
```

```
# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –
```

```
# sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
```

```
# sudo apt update
```

```
# sudo apt install -y containerd.io docker-ce docker-ce-cli
```

Create required directories for docker

```
# sudo mkdir -p /etc/systemd/system/docker.service.d
```

--- *One complete step*

```
# 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
```

Start and enable Services.

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

```
sudo systemctl enable docker
```

Steps Install kubernets ; *source from*

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

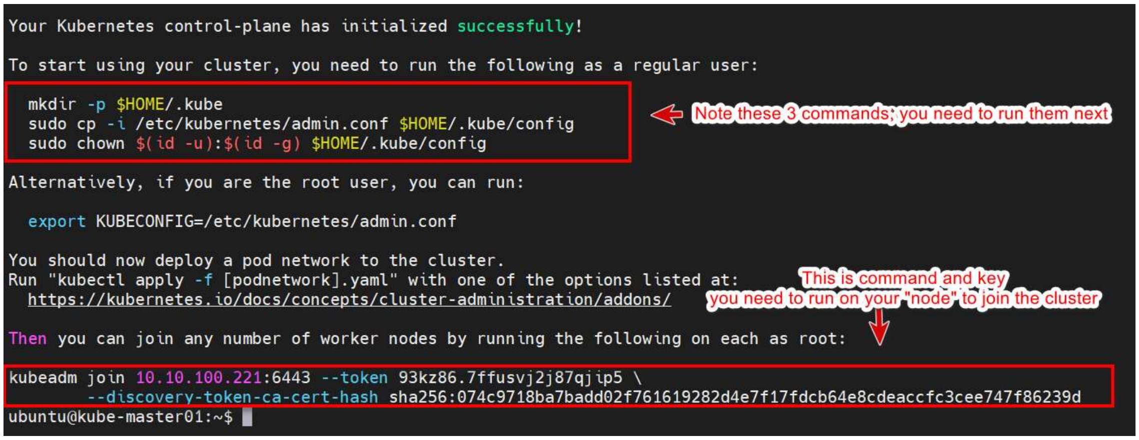
```
# sudo apt-get update
# sudo apt-get install -y apt-transport-https ca-certificates curl

# 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

# sudo apt-get update
# sudo apt-get install -y kubelet kubeadm kubectl
# sudo apt-mark hold kubelet kubeadm kubectl  # don’t use if not required
```

Till Here, Process is similar to both Master and Node

- ➔ To initialize the kubeadm
- ➔ Advertise address is server (master) IP
- ➔ Pod network is Calico (192.168.0.0/16)

# sudo kubeadm init --apiserver-advertise-address=10.10.100.221 --pod-network-cidr=192.168.0.0/16

- ➔ *your screen tail needs to as below*

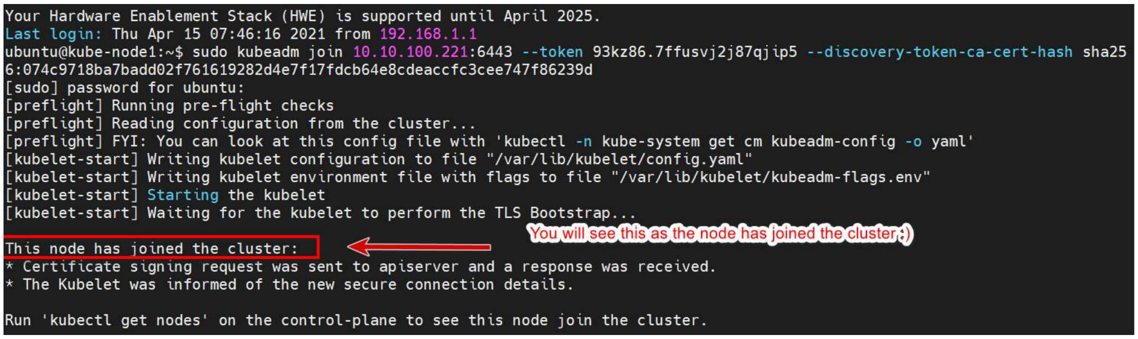


```
Your Kubernetes control-plane has initialized successfully!

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

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

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 10.10.100.221:6443 --token 93kz86.7ffusvj2j87qjip5 \
        --discovery-token-ca-cert-hash sha256:074c9718ba7badd02f761619282d4e7f17fdcb64e8cdeaccfc3cee747f86239d
ubuntu@kube-master01:~$
```

Note these 3 commands; you need to run them next

This is command and key
you need to run on your "node" to join the cluster

- ➔ switch to Node; Join the node using above command which as bottom of the screenshot

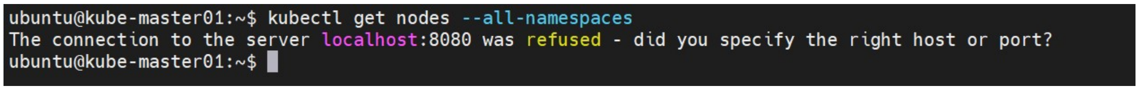


```
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Thu Apr 15 07:46:16 2021 from 192.168.1.1
ubuntu@kube-node1:~$ sudo kubeadm join 10.10.100.221:6443 --token 93kz86.7ffusvj2j87qjip5 --discovery-token-ca-cert-hash sha25
6:074c9718ba7badd02f761619282d4e7f17fdcb64e8cdeaccfc3cee747f86239d
[sudo] password for ubuntu:
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[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.env"
[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.
```

You will see this as the node has joined the cluster :)

- ➔ Switch back to Master

# kubectl get nodes –all-namespaces *# try to see nodes you have to see as below*



```
ubuntu@kube-master01:~$ kubectl get nodes --all-namespaces
The connection to the server localhost:8080 was refused - did you specify the right host or port?
ubuntu@kube-master01:~$
```

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

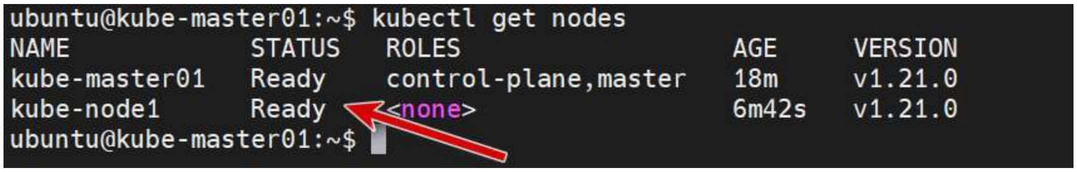
- ➔ Till this you are done with installation of kubeadm now you need to install POD network
- ➔ As mentioned above ; we are using calico so , I would be installing the calico network (192.168.0.0/16)

How to install calico network

# curl https://docs.projectcalico.org/manifests/calico.yaml -O
# kubectl apply -f calico.yaml

kubectl get nodes *# try this to check nodes*

If your calico network is created you should see the status as Ready



```
ubuntu@kube-master01:~$ kubectl get nodes
NAME            STATUS   ROLES                  AGE     VERSION
kube-master01   Ready    control-plane,master   18m     v1.21.0
kube-node1      Ready    <none>                 6m42s   v1.21.0
ubuntu@kube-master01:~$
```

How to create a dashboard

# kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yaml

# touch dashboard-admin.yaml

# nano dashoboard-admin.yaml

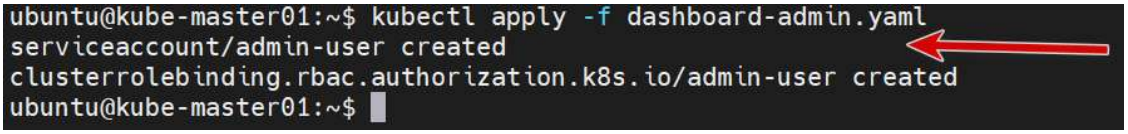
Copy – paste below script and save it on above "dashboard-admin.yaml" file

----

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
```

# kubectl apply -f dashboard-admin.yaml

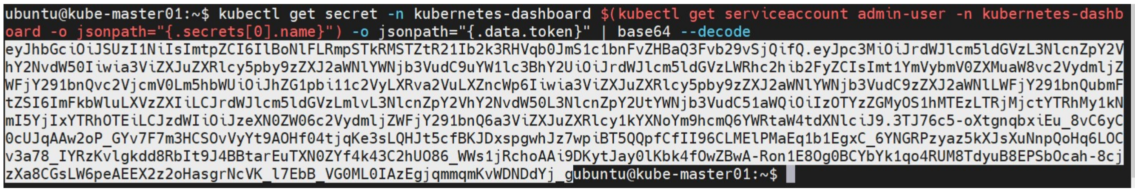
- ➔ *You will see the output as in the below screenshot*



```
ubuntu@kube-master01:~$ kubectl apply -f dashboard-admin.yaml
serviceaccount/admin-user created
clusterrolebinding.rbac.authorization.k8s.io/admin-user created
ubuntu@kube-master01:~$
```

# kubectl get secret -n kubernetes-dashboard $(kubectl get serviceaccount admin-user -n kubernetes-dashboard -o jsonpath="{.secrets[0].name}") -o jsonpath="{.data.token}" | base64 –decode

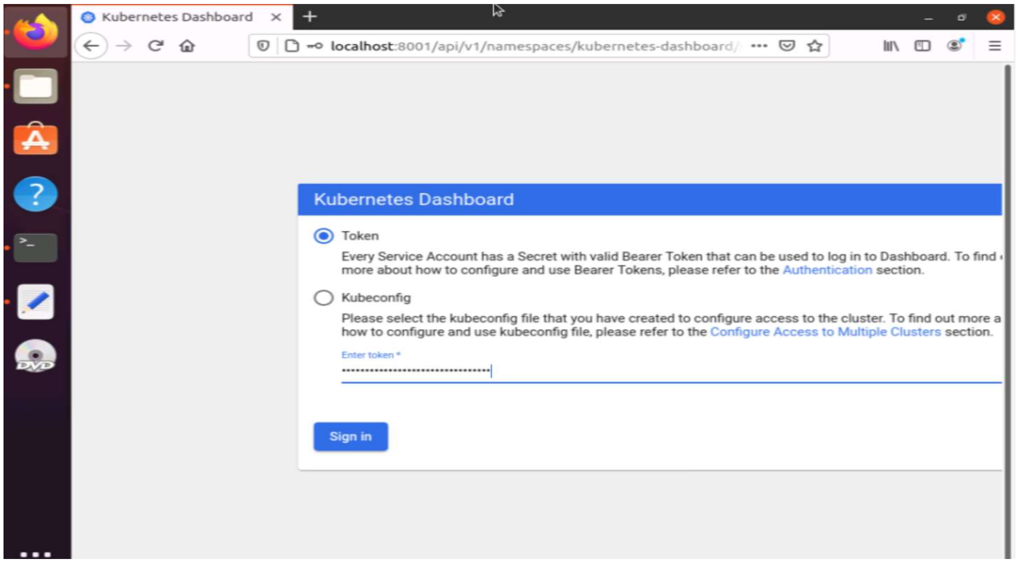
- ➔ Using above command you will be generating a token to login into your Kubernetes webgui
- ➔ It will as below



```
ubuntu@kube-master01:~$ kubectl get secret -n kubernetes-dashboard $(kubectl get serviceaccount admin-user -n kubernetes-dashb
oard -o jsonpath="{.secrets[0].name}") -o jsonpath="{.data.token}" | base64 --decode
eyJhbGciOiJSUzI1NiIsImtpZCI6IlBoNlFLRmpSTkRMSTZtR21Ib2k3RHVqb0JmS1c1bnFvZHBaQ3Fvb29vSjQifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2V
hY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsImt1YmVybmV0ZXMuaW8vc2VydmljZ
WFjY291bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi11c2VyLXRva2VuLXZncWp6Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmF
tZSI6ImFkbWluLXVzZXIiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiIzOTYzZGMyOS1hMTEzLTRjMjctYTRhMy1kN
mI5YjIxYTRhOTEiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6a3ViZXJuZXRlcy1kYXNoYm9hcmQ6YWRtaW4tdXNlciJ9.3TJ76c5-oXtgnqbxiEu_8vC6yC
0cUJqAAw2oP_GYv7F7m3HCSOvVyYt9AOHf04tjqKe3sLQHJt5cfBKJDxspgwhJz7wpiBT5QQpfCfII96CLMElPMaEq1b1EgxC_6YNGRPzyaz5kXJsXuNnpQoHq6LOC
v3a78_IYRzKvlgkdd8RbIt9J4BBtarEuTXN0ZYf4k43C2hUO86_WWs1jRchoAAi9DKytJay0lKbk4fOwZBwA-Ron1E8Og0BCYbYk1qo4RUM8TdyuB8EPSbOcah-8cj
zXa8CGsLW6peAEEX2z2oHasgrNcVK_l7EbB_VG0ML0IAzEgjqmmqmKvWDNDdYj_gubuntu@kube-master01:~$
```

- ➔ You can now access the webfrom localhost which mean same computer (master) using below link, and above token

# http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/



Kubernetes Dashboard

● Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find
more about how to configure and use Bearer Tokens, please refer to the Authentication section.

○ Kubeconfig

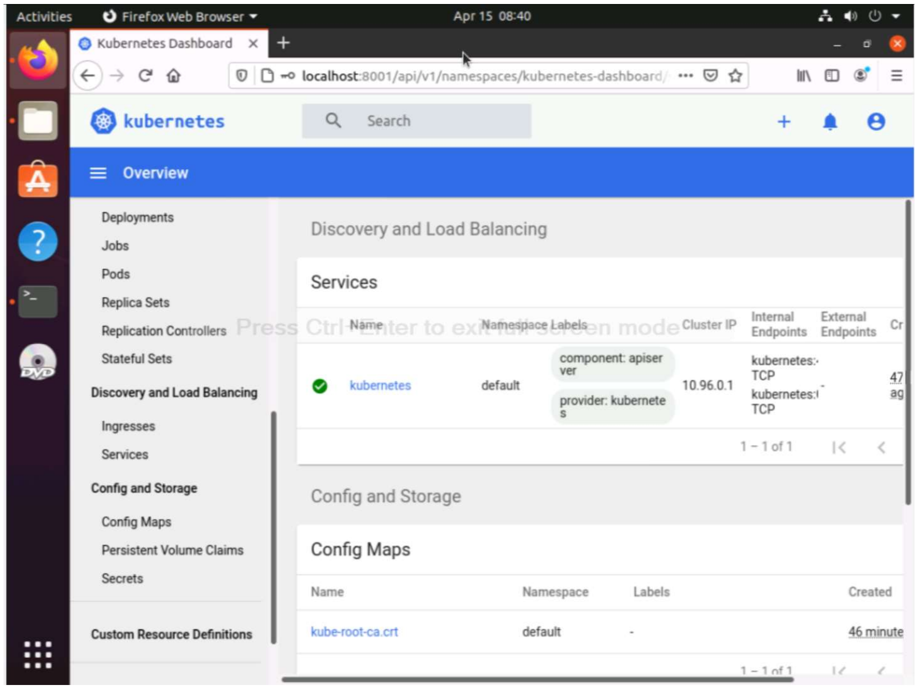
Please select the kubeconfig file that you have created to configure access to the cluster. To find out more a
how to configure and use kubeconfig file, please refer to the Configure Access to Multiple Clusters section.

Enter token *

································

Sign in

Your screen will look as mentioned below



Discovery and Load Balancing

Services

	Name	Namespace	Labels	Cluster IP	Internal Endpoints	External Endpoints	Cr
✓	kubernetes	default	component: apiserver provider: kubernetes	10.96.0.1	kubernetes: TCP kubernetes: TCP	-	47 ag

1 – 1 of 1

Config and Storage

Config Maps

Name	Namespace	Labels	Created
kube-root-ca.crt	default	-	46 minute

1 – 1 of 1

Thank you!