Installing k8s on Ubuntu 20.04 or ubuntu 22.04

1) Update your system.

sudo apt update

2) Add respository for docker.

sudo mkdir -p /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo "deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

3) Configure IP Address mapping in “/etc/hosts” file
Find IP’s of your vm/systems by ‘ip a” command on enxxx interface

E.g
##
#BELOW LINES NEED CUSTOMIZATION
#echo " " >> /etc/hosts
#echo "192.168.1.11 w1" >> /etc/hosts
#echo "192.168.1.10 cp" >> /etc/hosts

4) Set hostname of the systems

For control plane, on control plane node, execute following command:-
#hostnamectl set-hostname cp

5) On Worker node you will use following command.

#hostnamectl set-hostname w1

6) Load the Modules needed, create respective files as given below

modprobe br_netfilter
modprobe overlay

cat << EOF | tee /etc/modules-load.d/k8s-modules.conf
br_netfilter
overlay
EOF

```
cat << EOF |  tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
```

7) Update system control.

```
sysctl --system
```

8) Install containerd runtime

```
apt-get update ; apt-get install -y containerd.io

mkdir -p /etc/containerd

containerd config default | tee /etc/containerd/config.toml

sed -i "s/SystemdCgroup = false/SystemdCgroup = true/g"
/etc/containerd/config.toml

systemctl restart containerd
```

10) Turn of swap, remove or comment (hash) the entries for swap in "/etc/fstab" file.

```
swapoff -a
```

11) Install prerequisites required, by adding the repositories for k8s

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

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

apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
```

12) Install k8s binaries/ executables

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

13) Install bash completion for command auto completing commands via tab press.

```
# install bash-completion
sudo apt-get install bash-completion

# Add the completion script to your .bashrc file
echo 'source <(kubectl completion bash)' >>~/.bashrc
```

```
# Apply changes
source ~/.bashrc
```

14) After installation, executge kubeadm init on control plane.

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 –apiserver-advertise-address=< IP address of control-plane>
```

15) Step 14, command returns with success – then use following commands on worker node, which is given by kubeadm init command, below is an example..

E.g

```
kubeadm join 192.168.1.251:6443 --token awilnt.qskrvqbkbnq1tzea  --discovery-token-ca-cert-hash sha256:869387b8708df3a0d587c9670f568292bb1aafd4776ee8f1f4fa5096208e5dc0
```

16) There will be following command also needed for using the cluster as 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
```

16) Confirm with command for cluster nodes joining

```
kubectl get nodes
```

17) Configure CNI – flannel, for example for pod networking.

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

18) Confirm again by running 'kubectl get nodes' for "ready status"

E.g

```
kubectl get nodes

NAME     STATUS   ROLES           AGE    VERSION
k8ctrl    Ready    control-plane   5d3h   v1.28.2
u200w1    Ready    <none>          5d3h   v1.28.2
u200w2    Ready    <none>          5d3h   v1.28.2
```

19) Create deployment and check, with following command

e.g

```
kubectl create deployment nginx –image=nginx

kubectl get deployments nginx

kubectl create service nodeport nginx --tcp=80:80

kubectl get svc

curl localhost:30658
```

Note: 30658 port will vary from system to system. Check and use whats configured for your setup.

If all above, your cluster is successfully configured.