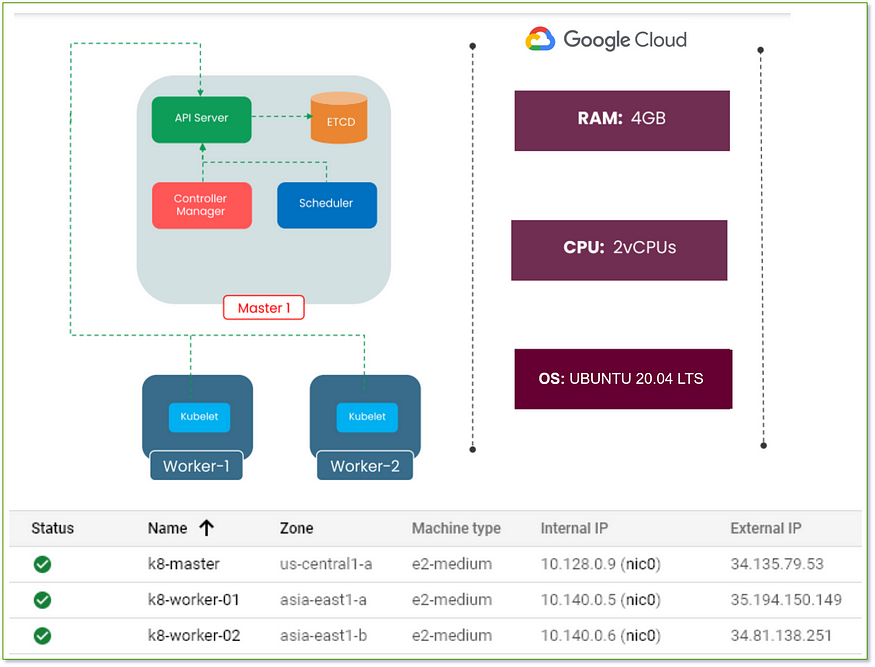
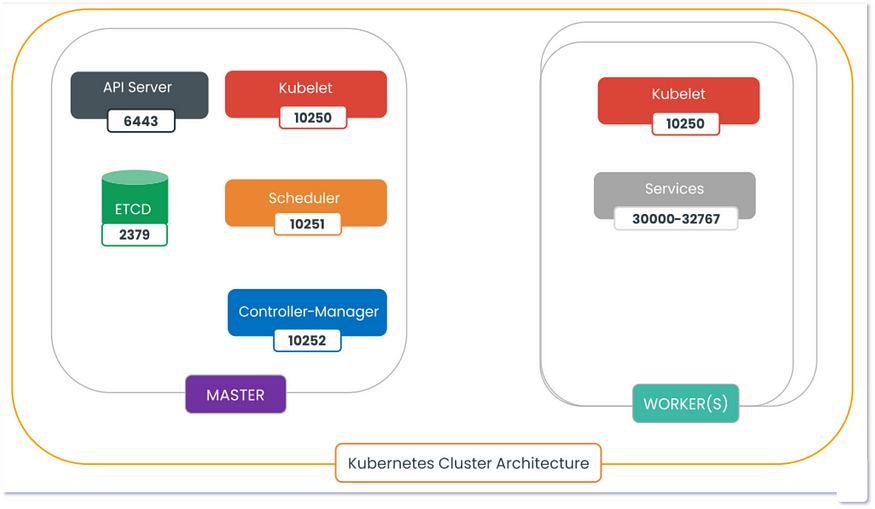
Create a Kubernetes cluster using kubeadm

**Prerequisites**

* Three Ubuntu servers 20.04 with at least 4GB RAM and 2 vCPUs each.

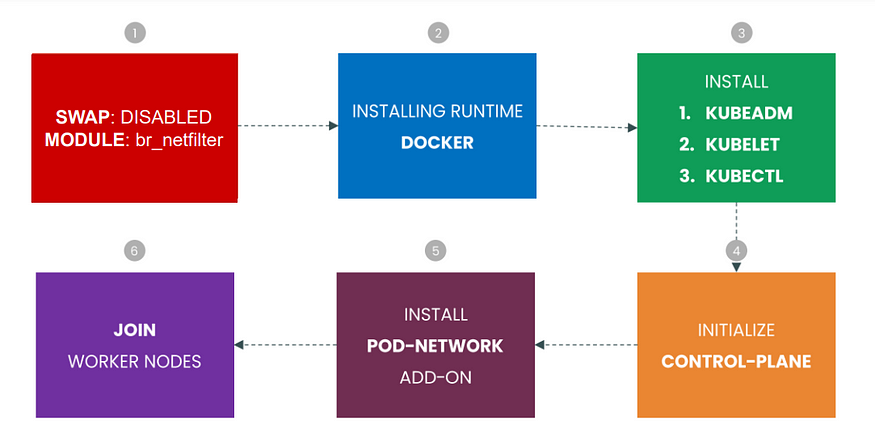


* SSH Access with sudo privileges.
* Firewall Ports/Inbound Traffic Ports should open for Kubernetes Cluster.



* Master Node Ports: 2379,6443,10250,10251,10252
* Worker Node Ports: 10250,30000–32767.
* Default port range for NodePort Services -30000–32767.

**Setup Steps:**



1. Disable the swap and make sure be a net filter module is installed.
2. we will need to install the container runtime interface ie. docker
3. Install kubeadm, kubelet, and kubectl: **kubeadm** is building tools that help to bootstrap the cluster,  
   **kubelet** is an agent that runs on each node to make sure that containers are running in a Pod,  
   **kubectl** allows you to run commands against Kubernetes clusters.
4. Initialize the Kubernetes cluster which creates certificates, pods, services, and other resources.
5. Installing wave network add-on.
6. Finally, join the worker nodes to the Kubernetes cluster.

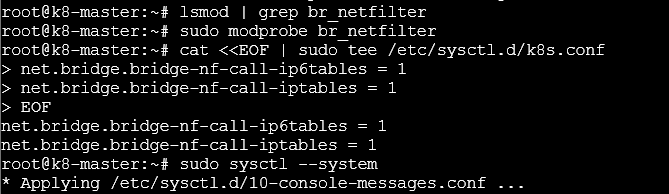
**Step1) Disable Swap (Run it on MASTER & WORKER Nodes)**

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

https://miro.medium.com/v2/resize:fit:875/1*NG1bz8U4-BIIv1nTk6LugA.png

1a) Bridge Traffic

$ lsmod | grep br\_netfilter   
$ sudo modprobe br\_netfilter$ cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf  
net.bridge.bridge-nf-call-ip6tables = 1  
net.bridge.bridge-nf-call-iptables = 1  
EOF$ sudo sysctl --system



* lsmod | grep br\_netfilter will load the module.
* To load it explicitly calls sudo modprobe br\_netfilter.
* As a requirement for your Linux Node’s iptables to correctly see bridged traffic.
* You should ensure net.bridge.bridge-nf-call-iptables is set to 1 in your sysctl configBridge Traffic.

**Step2) Install Docker (Run it on MASTER & WORKER Nodes)**

$ apt-get update   
$ apt install docker.io  
$ systemctl start docker

If you facing any issues, [Click here](https://bikramat.medium.com/docker-installation-on-ubuntu-88193b135b25) to install docker.

Get MrDevSecOps’s stories in your inbox

Join Medium for free to get updates from this writer.



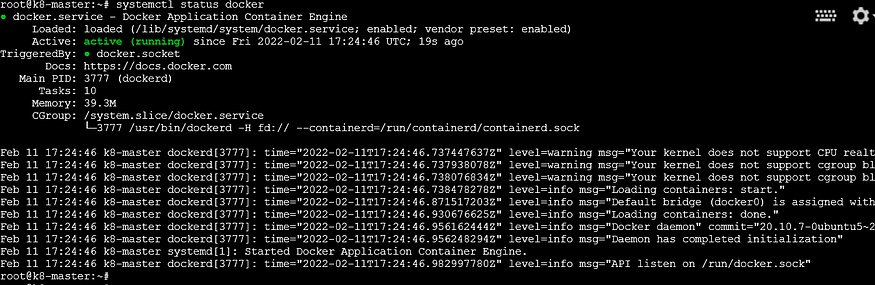
Subscribe

2a) Setting up the Docker daemon

$ 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

2b) Reload, enable and restart the docker service

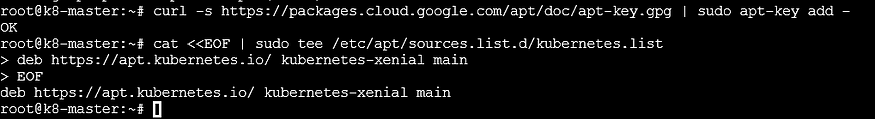
$ systemctl daemon-reload  
$ systemctl enable docker  
$ systemctl restart docker  
$ systemctl status docker



Make sure the docker service is running.

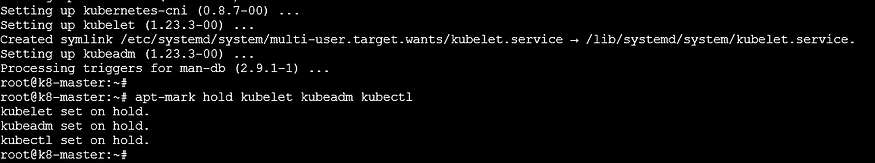
**Step3) Install kubeadm, kubelet, and kubectl (Run it on MASTER & WORKER Nodes)**

$ apt-get update && sudo apt-get install -y apt-transport-https curl  
$ curl -s <https://packages.cloud.google.com/apt/doc/apt-key.gpg> | sudo apt-key add -  
$ cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list  
deb <https://apt.kubernetes.io/> kubernetes-xenial main  
EOF



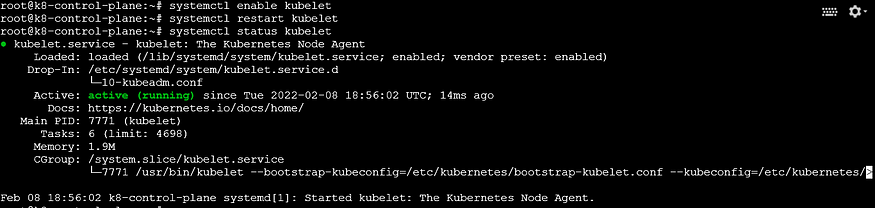
3a) Installing Kubeadm, Kubelet, Kubectl:

$ apt-get update  
$ apt-get install -y kubelet kubeadm kubectl  
$ apt-mark hold kubelet kubeadm kubectl



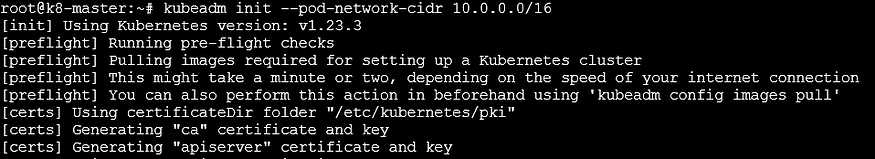
3b) Start and enable Kubelet

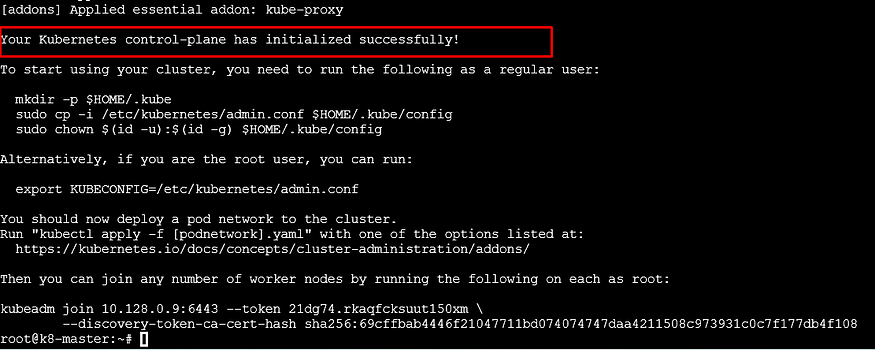
$ systemctl daemon-reload  
$ systemctl enable kubelet  
$ systemctl restart kubelet  
$ systemctl status kubelet



**Step4) Initializing CONTROL-PLANE (Run it on MASTER Node only)**

$ kubeadm init --pod-network-cidr 10.0.0.0/16





As the above output mentioned copy the token in your notepad, we will need to join worker/slave to the master node.

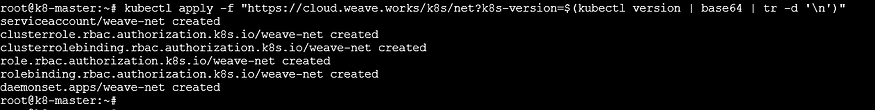
4a) Create new ‘.kube’ configuration directory and copy the configuration ‘admin.conf’ from ‘/etc/kubernetes’ directory.

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

https://miro.medium.com/v2/resize:fit:875/1*ETMjMSkKNV_TDsEsp9w01g.png

**Step5) Installing POD-NETWORK add-on (Run it on MASTER Node only)**

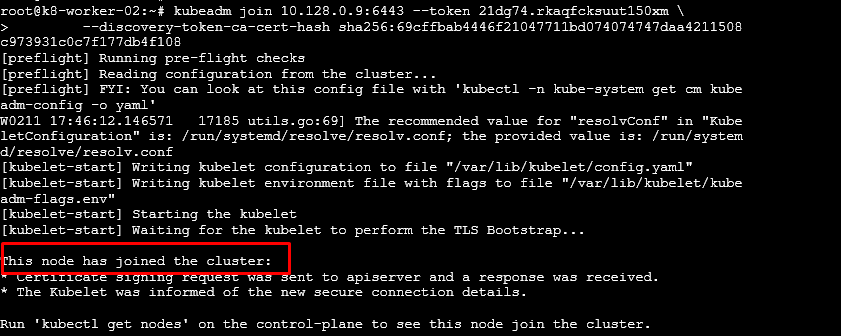
$ kubectl apply -f "[https://cloud.weave.works/k8s/net?k8s-version=$(kubectl](https://cloud.weave.works/k8s/net?k8s-version=%24%28kubectl) version | base64 | tr -d '\n')"

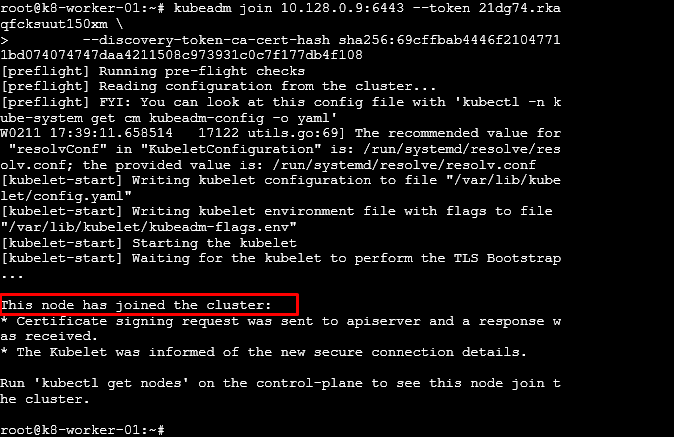


**Step6) Next Join two worker nodes to master (Run it on both worker nodes)**

Paste the Join command from the above kubeadm init output

$kubeadm join 10.128.0.9:6443 --token 21dg74.rkaqfcksuut150xm \  
> --discovery-token-ca-cert-hash sha256:69cffbab4446f21047711bd074074747daa4211508c973931c0c7f177db4f108



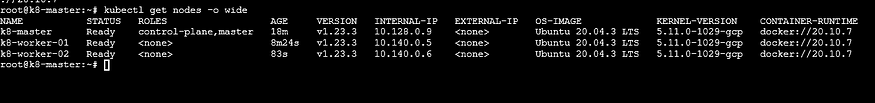


6a) Run this command IF you do not have the above join command.

$ kubeadm token create — print-join-command

6b) Check the joined nodes

$ kubectl get nodes -o wide



Also, check