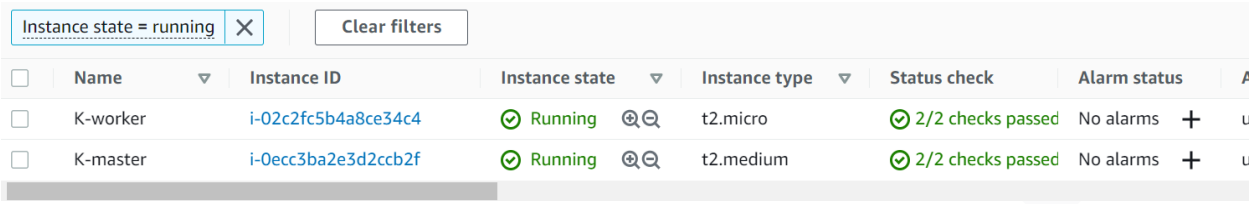
Module 7: Hands-On: Kubernetes Installation

Step 1: Launch 2 instances with the following configuration: **ubuntu 20.04 ami, t2.medium, sg: all traffic. ubuntu 20.04 ami, t2.micro , sg: all traffic**



To Install Kubernetes use the following commands:

On Master and Worker node:

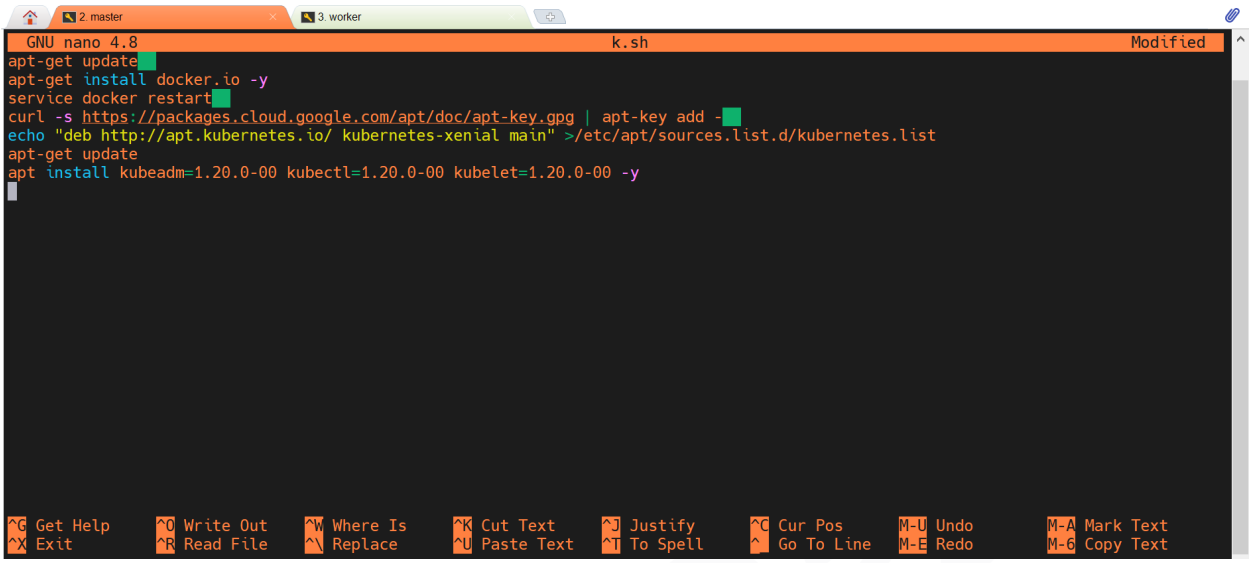
```
sudo su

apt-get update

apt-get install docker.io -y

service docker restart

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

echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list

apt-get update

apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y
```

Step 2: On both master and worker nodes run the above command:

2.1. **sudo su**

2.2. **create a script file k.sh**

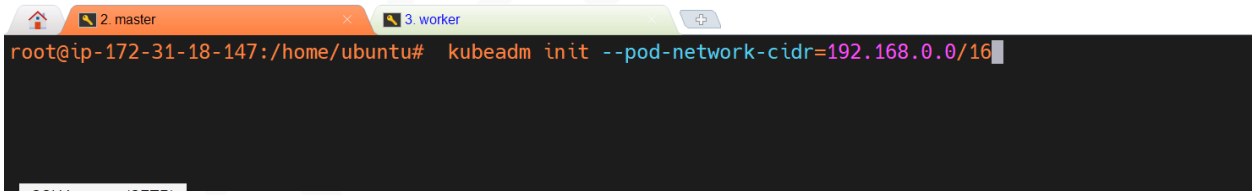
2.3. **to execute the script file: bash k.sh**



On Master:

Step 3:

kubeadm init --pod-network-cidr=192.168.0.0/16



```
[bootstrap-token] configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client 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

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 172.31.18.147:6443 --token rb42wf.rli4srhyw2cark8s \
    --discovery-token-ca-cert-hash sha256:bc23c6c2902c4c5f514ad86075b474e9c5b7060cd42e275499297c959aa8bd41
root@ip-172-31-18-147:/home/ubuntu#
```

Copy the token and paste it into the worker node

```
2 master   3 worker
root@ip-172-31-13-193:/home/ubuntu# kubeadm join 172.31.18.147:6443 --token rb42wf.rli4srhyw2cark8s \
>     --discovery-token-ca-cert-hash sha256:bc23c6c2902c4c5f514ad86075b474e9c5b7060cd42e275499297c959aa8bd41
[preflight] Running pre-flight checks
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is "systemd". Pleas
e follow the guide at https://kubernetes.io/docs/setup/cri/
        [WARNING SystemVerification]: this Docker version is not on the list of validated versions: 20.10.7. Latest validated versi
on: 19.03
[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.

root@ip-172-31-13-193:/home/ubuntu#
```

Step 4:

On Master:

exit

mkdir -p $HOME/.kube

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

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

Note: In case we want to retrieve the join token use the below-mentioned command.

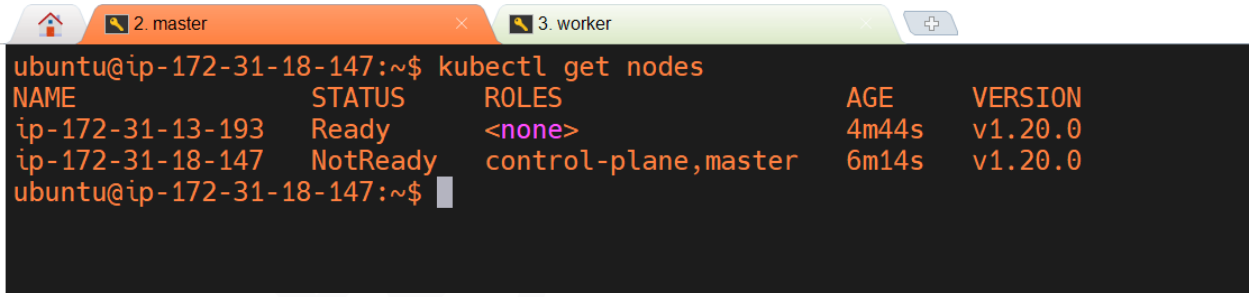
```
kubeadm token create --print-join-command
```

Step 5:

On Master:

```
kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml
```



Our Kubernetes installation and configuration is complete.