Step1: Both master & worker

1) TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`

2) hostnamectl set-hostname $(curl -s http://169.254.169.254/latest/meta-data/local-hostname -H "X-aws-ec2-metadata-token: $TOKEN")

3) Create the script [vi container.sh]

```
#!/bin/bash

sudo apt update

sudo swapoff -a

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

sudo tee /etc/modules-load.d/containerd.conf <<EOF
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

sudo tee /etc/sysctl.d/kubernetes.conf <<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
sudo sysctl --system

sudo apt install -y ca-certificates curl gnupg lsb-release

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

sudo apt update

sudo apt install -y containerd.io

containerd config default | sudo tee
/etc/containerd/config.toml >/dev/null 2>&1
sudo sed -i 's/SystemdCgroup \= false/SystemdCgroup \
= true/g' /etc/containerd/config.toml
sudo systemctl restart containerd
sudo systemctl enable containerd

4) Create the script for install [vi kube.sh]

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-
certificates curl gpg
curl -fsSL
https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.k
ey | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-
apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' |
sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet=1.28.* kubeadm=1.28.*
kubectl=1.28.*
sudo apt-mark hold kubelet kubeadm kubectl
sudo systemctl enable --now kubelet

5) Create a file [vi /etc/kubernetes/aws.yaml]
```

```
---
apiVersion: kubeadm.k8s.io/v1beta3
kind: ClusterConfiguration
apiServer:
  extraArgs:
    cloud-provider: external
controllerManager:
  extraArgs:
    cloud-provider: external
kubernetesVersion: v1.28.15
networking:
  podSubnet: 192.168.0.0/16
  serviceSubnet: 10.96.0.0/12


---
apiVersion: kubeadm.k8s.io/v1beta3
kind: InitConfiguration
nodeRegistration:
  kubeletExtraArgs:
    cloud-provider: external
```

Step2: Master :

```
1) # init cluster
kubeadm init --config /etc/kubernetes/aws.yaml

[you'll get joint token here:
kubeadm join 10.0.0.95:6443 --token
1mwt1r.bspbr63rs50nsyn2 \
        --discovery-token-ca-cert-hash
sha256:c878b65c4a06c666cb3707e2bfcb54cf8c2dc47d86a62e
a696c8def652eba116]

2) mkdir -p $HOME/.kube

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

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

```
5) export KUBECONFIG=/etc/kubernetes/admin.conf

6) kubectl apply -k 'github.com/kubernetes/cloud-
provider-aws/examples/existing-cluster/base/?
ref=master'

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

8) kubectl get nodes

9) kubectl get pods -A
```

Step3: Worker :

```
1) cat << EOF > /etc/kubernetes/node.yml
---
apiVersion: kubeadm.k8s.io/v1beta3
kind: JoinConfiguration
discovery:
  bootstrapToken:
    token: "1mwt1r.bspbr63rs50nsyn2"
    apiServerEndpoint: "10.0.0.95:6443"
    caCertHashes:
      -
"sha256:c878b65c4a06c666cb3707e2bfcb54cf8c2dc47d86a62
ea696c8def652eba116"
nodeRegistration:
  name: ip-10-0-0-124.ap-south-1.compute.internal
  kubeletExtraArgs:
    cloud-provider: external
EOF


[ NOTE: Replace hash,token,apiserver endpoint and
name with the details you get from joint cmd in STEP2
,youll get name by running cmd  #hostname -f in
worker]

2) kubeadm join --config /etc/kubernetes/node.yml
```