***Kubernetes cluster setup with worker nodes***

***1 . Create 2 ubuntu machine***

*Master Machine: 4GB RAM,2 CORE Processor(t2.medium)*

*Worker Machine: 2GB RAM,1 CORE Processor(t2.micro)*

*2. Execute below commands in both master and slave machine that needs to be a part of Kubernetes clusters.*

***#Turn off the memory swaping***

sudo swapoff -a

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

***modprobe br\_netfilter***

***sysctl -p***

***sudo sysctl net.bridge.bridge-nf-call-iptables=1***

**Also comment out the reference to swap in /etc/fstab. Start by editing the below file:**

sudo vim /etc/fstab

**Reboot the system to take effect**

sudo reboot

**Update the system Packages**

sudo apt-get update

**#Install Docker Container Runtime on All node (Master and Worker Nodes)**

**Uninstall old versions**

sudo apt-get remove docker docker-engine docker.io containerd runc

**Update the apt package index and install packages to allow apt to use a repository over HTTPS:**

sudo apt-get update

sudo apt-get install \

ca-certificates \

curl \

gnupg \

lsb-release

**Add Docker’s official GPG key:**

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

**#Use the following command to set up the stable repository. To add the nightly or test #repository, add the word**

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

**Install Docker Engine**

*sudo apt-get update*

*sudo apt-get install docker-ce docker-ce-cli containerd.io*

**Add the Docker Daemon configurations to use systemd as the cgroup driver.**

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

**Check docker images**

*docker images*

**Add the docker user in group and give permission for docker.sock**

*sudo usermod -aG docker ubuntu*

*sudo chmod 666 /var/run/docker.sock*

*sudo systemctl start docker.service*

*sudo systemctl status docker.service*

***Enable Docker service at startup***

*sudo systemctl enable docker.service*

*sudo systemctl restart docker*

***#Add Kubernetes GPG Key on All node***

*sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg* [*https://packages.cloud.google.com/apt/doc/apt-key.gpg*](https://packages.cloud.google.com/apt/doc/apt-key.gpg)

***#Add Kubernetes APT Repository on All node***

*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*

***#update the system packages***

*sudo apt-get update*

***# Install Kubeadm,Kubelet and Kubectl on All Node***

*sudo apt-get install -y kubelet kubeadm kubectl*

*sudo systemctl start kubelet*

*sudo systemctl enable kubelet.service*

sudo su -

rm /etc/containerd/config.toml

systemctl restart containerd

**ERROR:**

*sudo kubeadm reset*

***==================Common Commands for master and slaves END HERE====================***

***Execute below commands only on master of Kubernetes clusters.***

***==================Commands that has to run only on master starts here====================***

***# Execute below command as root user***

sudo kubeadm init

# Exit root user & execute as normal user

su ubuntu

mkdir -p $HOME/.kube

curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

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

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

kubectl apply -f [https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')](https://cloud.weave.works/k8s/net?k8s-version=$(kubectl%20version%20|%20base64%20|%20tr%20-d%20'\n'))

kubectl get nodes

kubectl get pods --all-namespaces

***kubeadm token create --print-join-command***

***============*========*Commands that has to run on master ends here========================***

***========================IN Worker Node Command Starts here==================***

***copy the kubeadm join token and execute in worker Nodes to join to cluster***

***========================IN Worker Node Command Ends here====================***