**Creation of k8s cluster:-**

* **Installing the pre requisites**
* **Setting up the container run time**
* **Installing the kubeadm**
* **Initializing the cluster**

**Installing the pre requisites:-**

Forwarding IPv4 and letting iptables see bridged traffic Execute the below mentioned instructions:

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

overlay

br\_netfilter

EOF

sudo modprobe overlay

sudo modprobe br\_netfilter

*# sysctl params required by setup, params persist across reboots*

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-iptables = 1

net.bridge.bridge-nf-call-ip6tables = 1

net.ipv4.ip\_forward = 1

EOF

*# Apply sysctl params without reboot*

sudo sysctl --system

Verify that the br\_netfilter, overlay modules are loaded by running below instructions:

lsmod | grep br\_netfilter

lsmod | grep overlay

Verify that the net.bridge.bridge-nf-call-iptables, net.bridge.bridge-nf-call-ip6tables, net.ipv4.ip\_forward system variables are set to 1 in your sysctl config by running below instruction

sysctl net.bridge.bridge-nf-call-iptables net.bridge.bridge-nf-call-ip6tables net.ipv4.ip\_forward

**Setting up the container run time :-**

Install the necessary dependencies with:

**sudo apt install curl gnupg2 software-properties-common apt-transport-https ca-certificates -y**

Add the GPG key with:

**curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –**

Add the required repository with:

**sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"**

Install containerd with the commands:+

**sudo apt update**

**sudo apt install containerd.io -y**

Change to the root user with:

**sudo su -**

Create a new directory for containerd with:

**mkdir -p /etc/containerd**

Generate the configuration file with:

**containerd config default>/etc/containerd/config.toml**

Exit from the root user with:

**Exit**

#### **Configuring the systemd cgroup driver**

To use the systemd cgroup driver in /etc/containerd/config.toml with runc, set

[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc]

...

[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]

SystemdCgroup = true

Note :- the above change must be done

Restart containerd with the command:

**sudo systemctl restart containerd**

Enable containerd to run at startup with:

**sudo systemctl enable containerd**

**Installing the kubeadm:-**

1. Update the apt package index and install packages needed to use the Kubernetes apt repository:
2. sudo apt-get update
3. sudo apt-get install -y apt-transport-https ca-certificates curl
4. Download the Google Cloud public signing key:
5. sudo curl -fsSLo /etc/apt/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
6. Add the Kubernetes apt repository:
7. echo "deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
8. Update apt package index, install kubelet, kubeadm and kubectl, and pin their version:
9. sudo apt-get update
10. sudo apt-get install -y kubelet kubeadm kubectl

sudo apt-mark hold kubelet kubeadm kubectl

**Initializing the cluster**

**On Master:**

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

**>Copy the token and paste it into the worker node.**

**exit**

**mkdir -p $HOME/.kube**

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

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

**kubectl apply -f** [**https://docs.projectcalico.org/manifests/calico.yaml**](https://docs.projectcalico.org/manifests/calico.yaml)