**Lab 1**

**Install a single master Kubernetes cluster**

**Step 1 – Check and configuring prerequisites (Both on master and worker nodes)**

* 1. **Check OS versions**

|  |
| --- |
| Ensure both master and worker nodes have the identical OS versions |
| *cat /etc/os-release* |

* 1. **Add cluster node names**

|  |
| --- |
| Add cluster node names to every node in the cluster |
| *vi /etc/hosts*  *192.168.56.201 master-1*  *192.168.56.211 worker-1*  *192.168.56.212 worker-2*  *192.168.56.213 worker-3* |

* 1. **Check network connectivity**

|  |
| --- |
| Check connectivity to internet and between all nodes |
| *ping 4.2.2.2*  *192.168.56.201*  *192.168.56.211*  *192.168.56.212*  *192.168.56.213* |

* 1. **Disable swap**

|  |
| --- |
| Disable swap permanently |
| *vi /etc/fstab # Comment out swap file system entry* |

* 1. **Disable SELinux**

|  |
| --- |
| Disable SELinux. You can however run Kubernetes with SELinux. |
| *vi /etc/hosts*  *192.168.56.201 master-1*  *192.168.56.211 worker-1*  *192.168.56.212 worker-2*  *192.168.56.213 worker-3* |

* 1. **Enable firewall ports**

|  |
| --- |
| Firewall ports must be opened permanently for both master and workers nodes |
| *# On master nodes*  *firewall-cmd --permanent --add-port=6443/tcp*  *firewall-cmd --permanent --add-port=2379/tcp*  *firewall-cmd --permanent --add-port=2380/tcp*  *firewall-cmd --permanent --add-port=10250/tcp*  *firewall-cmd --permanent --add-port=10251/tcp*  *firewall-cmd --permanent --add-port=10252/tcp*  *firewall-cmd --permanent --add-port=10255/tcp*  *firewall-cmd --reload*  *# on worker nodes*  *firewall-cmd --permanent --add-port=10251/tcp*  *firewall-cmd --permanent --add-port=10255/tcp*  *firewall-cmd --reload* |

* 1. **Verify and load kernel modules**

|  |
| --- |
| Following kernel modules must be loaded. If not loaded already configure the systems to load them at startup.  **br\_netfilter** - This module is required to enable transparent masquerading and to facilitate Virtual Extensible LAN (VxLAN)  **overlay** - For the overlay network |
| *lsmod | grep br\_netfilter*  *lsmod | grep overlay*  *modprobe br\_netfilter; lsmod | grep br\_netfilter*  *modprobe overlay; lsmod | grep overlay*  vi /etc/modules-load.d/k8s.conf  br\_netfilter  overlay |

* 1. **Add kernel parameters**

|  |
| --- |
| Add the following kernel parameters permanently |
| *vi /etc/sysctl.d/k8s.conf*  *net.bridge.bridge-nf-call-ip6tables = 1*  *net.bridge.bridge-nf-call-iptables = 1*  *net.ipv4.ip\_forward = 1*  *sysctl --system* |

* 1. **Reboot**

|  |
| --- |
| Reboot the system to make sure systems are configured and clean before installation |
| *reboot* |

**Step 2 – Install Kubernetes (Both on master and worker nodes)**

* 1. **Install container runtime - containerd**

|  |
| --- |
| Install the containerd runtime in all nodes, enable and start the service |
| *dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo*  *dnf install --allowerasing -y containerd*  *mkdir -p /etc/containerd*  *containerd config default | tee /etc/containerd/config.toml*  *vi /etc/containerd/config.toml*  *Find the line in the flie and make it to true*  *[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options] ---> And change the value of SystemdCgroup to true*  *systemctl restart containerd*  *systemctl enable containerd*  *ps -ef | grep containerd* |

* 1. **Install Kubernetes packages - kubelet, kubectl, kubeadm**

|  |
| --- |
| Every node requires the following 3 packages.  **kubeadm**: the command to bootstrap the cluster  **kubelet**: the component that runs on all of the machines in your cluster and does things like starting pods and containers  **kubectl**: the command line util to talk to your cluster  *Official installation guide : https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/*  *Google repos : https://packages.cloud.google.com/yum/repos/* |
| *vi /etc/yum.repos.d/kubernetes.repo*  *[kubernetes]*  *name=Kubernetes*  *baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64*  *enabled=1*  *gpgcheck=1*  *repo\_gpgcheck=1*  *gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg*  *dnf install -y kubelet kubeadm kubectl*  *systemctl enable kubelet* |

**Step 3 – Initialize the cluster**

* 1. **Initialize the cluster (Only on master node)**

|  |
| --- |
| Install the containerd runtime in all nodes, enable and start the service |
| *kubeadm init --pod-network-cidr=10.10.0.0/16 --control-plane-endpoint 192.168.56.201:6443 --apiserver-advertise-address=192.168.56.201*  *mkdir -p $HOME/.kube*  *cp -i /etc/kubernetes/admin.conf $HOME/.kube/config*  *chown $(id -u):$(id -g) $HOME/.kube/config*  *kubectl get nodes* |

* 1. **Join the worker nodes (Only on worker nodes)**

|  |
| --- |
| Once the cluster is initialized join the worker nodes to the cluster |
| *kubeadm join 192.168.56.201:6443 --token ux1el0.8owgc4afkkk2obrx \*  *--discovery-token-ca-cert-hash sha256:ec046f3eddc7b2b0ee90b47b3b6a09791de3b0055f4942da20def05aceace3b3*  *kubectl get nodes*  *kubectl get pods -A*  *#Nodes should be in a NotReady State* |

**Step 4 – Initialize the cluster (Only on master node)**

* 1. **Create the pod network**

|  |
| --- |
| Create the pod overlay network |
| *curl https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml -O*  *kubectl apply -f calico.yaml*  *kubectl get pods -A ### Wait until all pods are running*  *kubectl get nodes ### Master node should be in Ready state* |

**Step 5 – Verify the installation**

* 1. **xxx**

|  |
| --- |
| xxx |
| *xxx ### Master node should be in Ready state* |