* **Kubernetes Cluster installation using kubeadm  
    
  Follow this documentation to set up a Kubernetes cluster on CentOS machines.  
    
  This documentation guides you in setting up a cluster with one master node and two worker nodes.  
    
  Prerequisite:  
    
    
    
  1) System Requirements  
    
  Master: t2.medium (2 CPUs and 2GB Memory)  
  Worker Nodes: t2.micro  
    
  2) Open Below ports in the Security Group.  
  Master node:  
  Kubernetes Cluster installation using kubeadm  
    
  Follow this documentation to set up a Kubernetes cluster on CentOS machines.  
    
  This documentation guides you in setting up a cluster with one master node and two worker nodes.  
    
  Prerequisite:  
    
    
    
  1) System Requirements  
    
  Master: t2.medium (2 CPUs and 2GB Memory)  
  Worker Nodes: t2.micro  
    
  2) Open Below ports in the Security Group.  
  Master node:  
    
  6443 3275@ 10250 4443 443 8080  
    
    
    
  On Master node and Worker node:  
    
  3) Perform all the commands as root user unless otherwise specified  
    
  sudo su -  
    
  Install, Enable and start docker service. Use the Docker repository to install docker.  
    
  If you use docker from CentOS OS repository, the docker version might be old to work with Kubernetes v1.13.@ and above**  
  yum install -y -q yum-utils device-mapper-persistent-data lvm2 > /dev/null 2>&1

yum-config-manager –add-repo <https://download.docker.com/linux/centos/docker-ce.repo> > /dev/null 2>&1   
yum install -y -q docker-ce >/dev/null 2>&1  
  
4) Start Docker services  
  
systemctl enable docker  
  
systemctl start docker  
  
service docker status  
  
5) Disable SELinux:  
  
setenforce 0

* sed -i --follow-symlinks 's/^SELINUX=enforcing/SELINUX=disabled/' /etc/sysconfig/selinux  
    
  6) Disable Firewall  
    
  systemctl disable firewalld  
  systemctl stop firewalld|  
  7) Disable swap space  
    
  sed -i '/swap/d' /etc/fstab  
  swapoff -a  
    
  8) Update sysctl settings for Kubernetes networking  
    
  cat >> /etc/sysctl.d/kubernetes.conf <<EOF

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

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

EOF

sysct1 –system

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

EOF

sysct1 –system

K8S SETUP

1. Add yum repository for kubernetes packages  
     
   cat >>/etc/yum.repos.d/kubernetes.repo<[kubernetes]  
   name=Kubernetes  
   baseurl=https: //[packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64](http://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](http://packages.cloud.google.com/yum/doc/rpm-package-key.gpg)  
   EOF  
     
   2) Install Kubernetes  
     
     
     
   yum install -y kubeadm-1.15.6-0.x86\_64 kubelet-1.15.6-0.x86\_64 kubectl-1.15.6-0.x86\_64  
     
     
   3) Enable and Start kubelet service  
   systemctl enable kubelet  
   systemctl start kubelet  
     
   On Master Node:  
     
   1) Initialize Kubernetes Cluster  
   kubeadm init --apiserver-advertise-address= --pod-network-cidr=[192.168.0.0/16](http://192.168.0.0/16)
2. Create a user for kubernetes administration and copy kube config file.  
     
   To be able to use kubectl command to connect and interact with the cluster, the user needs kube config file.  
     
   In this case, we are creating a user called kubeadmin  
     
   useradd kubeadmin  
     
   mkdir /home/kubeadmin/.kube  
     
   cp /etc/kubernetes/admin.conf /home/kubeadmin/.kube/config  
     
   chown -R kubeadmin:kubeadmin /home/kubeadmin/.kube  
     
   3) Deploy Calico network as a kubeadmin user.  
     
   This should be executed as a user (heare as a kubeadmin )  
     
   sudo su - kubeadmin  
     
   kubectl create -f <https://docs.projectcalico.org/v3.9/manifests/calico.yaml>  
     
   4) Cluster join command  
     
   kubeadm token create --print-join-command  
     
   On Worker Node:  
     
   Add worker nodes to cluster  
     
     
     
   Use the output from kubeadm token create command in previous step from the master server and run here.

Verifying the cluster To Get Nodes status

Kubect1 get nodes

To get component status

Kubectl get pods

Kubectl get pods -n kube-system