MASTER NODE:-

Point1 :- make 1 Master node centOS7on ec2 Instance type T2 small ( 1 GB cpu and 2 gb momory ) and storage 10GB. ( allow all port )

2:- sudo -i

3:- hostnamectl set-hostname master.example.com

4:- bash

5:- 1 yum install net-tools

5:- 2 ifconfig eth0 ( copy the IP address 172.31.87.32)

6:- vi /etc/hosts

( 172.31.1.185 master.example.com )

7:- setenforce 0

8:-

( SELINUX=disabled )

8.1:- swapon -s

9:- yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

#######(( Set the dbashocker repo ) ( website :- https://docs.docker.com/engine/install/centos/ ))

10:- yum install docker-ce -y

11:- systemctl start docker

12:- systemctl enable docker

13:- vi /etc/yum.repos.d/kube.repo

( [kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=0 )

#############(( website :- https://packages.cloud.google.com/yum/repos ))

14:- yum list kubeadm ( check the kubeadm version 1.22.0-0 but we have to take lower version 1.20.0-0 new version is not working )

14.1:- yum install kubeadm-1.20.0-0 kubelet-1.20.0-0 kubectl-1.20.0-0 -y

15:- systemctl start kubelet

16:- systemctl enable kubelet

17:- kubeadm init (or) kubeadm init --ignore-preflight-errors=all ((incase error shows because of low space. if you were take 1 gb cpu, need is 2 gb cpu) (onetime in one node))

18:- rpm -qa kubectl ( To check kubectl available or not )

18.1:- rpm -qa kubectl ( To check the components whether its installed or not )

19:- kubectl get nodes ( To check how many nodes available )

20:- ls -l /etc/kubernetes/admin.conf ( path of the API )

21:- kubectl get nodes --kubeconfig=/etc/kubernetes/admin.conf ( always we need to authenticate API in this method)

or:- export KUBECONFIG=/etc/kubernetes/admin.conf ( export the file to "KUBECONFIG" under the variable in home account )

:- unset KUBECONFIG

or:- mkdir .kube

:- cp /etc/kubernetes/admin.conf .kube/config

incase {kubectl get nodes --kubeconfig /etc/kubernetes/kubelet.conf

Rename kubelet.conf to admin.conf for your convenience at this point.}

22:- kubectl get nodes ( To check how many nodes available )

23:- kubectl get ns ( To check nodes are healthy or not )

24:- kubectl get pod -n kube-system

25:- kubectl get pod

=========================================

Master node setup finish

==========================================

=================================================================== =========================================

WORKER NODE:-

Create Two centOS instances at a time ( ram 2gb type T2 medium and storage 10gb)

worker node 1:-

1:- sudo -i

2:- hostnamectl set-hostname worker1.example.com

3:- bash

yum install net-tools

4:- ifconfig eth0 ( copy the IP )

5:- vi /etc/hosts

( 172.31.94.58 worker1.example.com)

6:- swapon -a (its already close )

7:- yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

8:- yum install docker-ce -y

9:- systemctl start docker

10:- systemctl enable docker

11:- vi /etc/yum.repos.d/kube.repo

( [kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=0 ) ( This is the kubelet package )

12:- yum install kubeadm-1.20.0-0 kubelet-1.20.0-0 -y

13:- systemctl start kubelet

systemctl enable kubelet

15:- kubeadm join 172.31.10.93.6443 --token u1dn7s.kcka...... --discovery-token-ca-cart sha256:c7e7f95.....

( In this command we can attach the worker node with our master node,,, 'kubeadm token' command we can get that or we can create new token 'kubeadm token create' )

-------------------

worker node 2:-

-------------------

Repeat the same step like node 1

======================================================================================

after that run 'calico' command which give ip address to containers. (it should be in master node)

1:- curl https://projectcalico.docs.tigera.io/manifests/calico.yaml -O ( ( website - https://projectcalico.docs.tigera.io/getting-started/kubernetes/self-managed-onprem/onpremises ) ( we can get the calico package ) )

2:- ls -l calico.yaml

3:- kubectl create -f calico.yaml ( it will provide ip to containers )

4:- kubectl get nodes ( if status is ready then the cluster is ready )

kubeadm join 172.31.87.32:6443 --token akgrnj.8wtbplrasg3ynwz9 \

End: