Related link troubleshooting used –

**[Solved] kubelet isn't running or healthy.The HTTP call equal to 'curl -sSL http://localhost:10248/healthz' failed with error**

<https://www.unixcloudfusion.in/2021/08/solved-kubelet-isnt-running-or.html>

for swap -- <https://www.devopsschool.com/blog/kubeadm-error-kubelet-isnt-running-or-healthy-and-connection-refused/>

instllation link -- <https://thenewstack.io/how-to-deploy-a-kubernetes-cluster-with-ubuntu-server-18-04/>

Deployment link on Kube cluster. <https://www.tecmint.com/deploy-nginx-on-a-kubernetes-cluster/>

Create 2 ubuntu servers 18 –

Then done the update on both –

apt update

apt-get install docker.io

perform this on both servers

[root@k8smaster ~]# vi /etc/docker/daemon.json

Add this code

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

Then run below commands

systemctl restart docker

systemctl status docker

then

sudo swapoff -a

sudo sed -i '/ swap / s/^/#/' /etc/fstab

then reboot both servers.

Now follow the installation process of Kubernetes.

[**https://thenewstack.io/how-to-deploy-a-kubernetes-cluster-with-ubuntu-server-18-04/**](https://thenewstack.io/how-to-deploy-a-kubernetes-cluster-with-ubuntu-server-18-04/)

1

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add

sudo apt-get install curl -y

sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"

sudo apt-get install kubeadm kubelet kubectl -y

Skip the hostname set ..

## Initializing the Master Only

sudo kubeadm init --apiserver-advertise-address=add master ip here

--pod-network-cidr=192.168.0.0/16

sudo kubeadm init --pod-network-cidr=replace here you master ip/16

If you will get any error please see in stack over flow or google it will be helpful.

Once Initializing done on master copy the Token and join command on finished screen

Like below :i:e

kubeadm join 172.31.14.140:6443 --token eigb36.v6ahmbys7jpguea9 --discovery-token-ca-cert-hash sha256:ebbd02226f6add35e32d87a4b07c4228cc5c6b2f315129db01552cc0f419e9cf

kubeadm join 172.31.11.78:6443 --token nu8h7c.rsowkr1xpgns5cm7 --discovery-token-ca-cert-hash sha256:33b9f023476ad0a86f2d16db843e0cb0c424b06047b1e8c355eade87caaa1b55

mkdir -p $HOME/.kube

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

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

now run kubectl get nodes (it will throw error of local host refused)

**kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml**

curl https://docs.projectcalico.org/manifests/calico.yaml -O

kubectl apply -f calico.yaml

Next run again

kubectl get nodes. On master only

you will see status ready for all nodes.

Also run this command on – Master only

Run this command in node only

kubeadm join 172.31.14.140:6443 --token eigb36.v6ahmbys7jpguea9 --discovery-token-ca-cert-hash sha256:ebbd02226f6add35e32d87a4b07c4228cc5c6b2f315129db01552cc0f419e9cf

these command will be on master only

**kubectl get nodes**

kubectl get pods --all-namespaces

mkdir -p $HOME/.kube

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

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

Continue happy learning !!!!!!!.. will continue deployment on created cluster..

Deploying NGINX on a Kubernetes Cluster

kubectl get nodes

We create a deployment of **NGINX** using the **NGINX** image.

kubectl create deployment nginx --image=nginx

Output will below..

Text

Description automatically generated

You can now see the state of your deployment.

kubectl get deployments

Text

Description automatically generated with medium confidence

If you’d like to see more detail about your deployment, you can run the **describe** command. For example, it is possible to determine how many replicas of the deployment are running. In our case, we expect to see a replica of 1 running (i.e **1/1** replicas).

kubectl describe deployment nginx

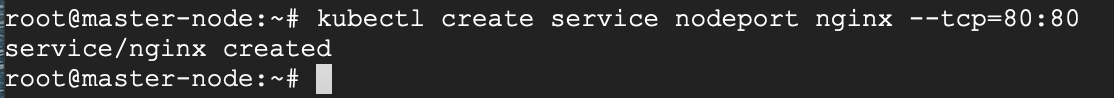
out will be Text

Description automatically generated

Now your Nginx deployment is active, you may want to expose the **NGINX** service to a public IP reachable on the internet.

#### Exposing Your Nginx Service to Public Network

kubectl create service nodeport nginx --tcp=80:80



Run the **get svc** command to see a summary of the service and the ports exposed.

Text

Description automatically generated

Now you can verify that the **Nginx** page is reachable on all nodes using the [curl command](https://www.tecmint.com/linux-curl-command-examples/).

Localhost:30386 or port you see in your screen.

As you can see, the “**WELCOME TO NGINX!**” page can be reached.

Next check with your Public IP and port number

Graphical user interface, text, application, email

Description automatically generated

we have successfully deployed **NGINX** on our Kubernetes cluster.