**Check kops installed or not using**

kops version

**check kubectl installed or not using**

kubectl version –short

**Why ssh keys need to be generated using ssh-keygen command ?**

ssh-keygen command creates a public and private key which are used while creating cluster and using these keys we can do ssh/connect to the server. It internally uses these private and public keys.

**Creating cluster :**

**Below command gives the information as it is dry run**

kops create cluster --name=k8sb14.xyz \

--state=s3://k8sb14.xyz --zones=us-east-1a,us-east-1b,us-east-1c \

--node-count=3 --master-count=3 --node-size=t3.medium --master-size=t3.medium \

--master-zones=us-east-1a,us-east-1b,us-east-1c --master-volume-size 10 --node-volume-size 10 \

--dns-zone=k8sb14.xyz --dry-run --output yaml

**Actual command:**

kops create cluster --name=k8sb14.xyz \

--state=s3://k8sb14.xyz --zones=us-east-1a,us-east-1b,us-east-1c \

--node-count=3 --master-count=3 --node-size=t3.medium --master-size=t3.medium \

--master-zones=us-east-1a,us-east-1b,us-east-1c --master-volume-size 10 --node-volume-size=10 \

--dns-zone=k8sb14.xyz --yes

kops create cluster --name=k8sb14.xyz \

--state=s3://k8sb14.xyz --zones=us-east-1a,us-east-1b,us-east-1c \

--node-count=3 --master-count=1 --node-size=t3.medium --master-size=t3.medium \

--master-zones=us-east-1a --master-volume-size=10 --node-volume-size=10 \

--dns-zone=k8sb14.xyz --yes

kops create cluster --name=k8sb14.xyz \

--state=s3://k8sb14.xyz --zones=us-east-1a,us-east-1b,us-east-1c \

--node-count=3 --master-count=1 --node-size=t3.medium --master-size=t3.medium \

--master-zones=us-east-1a --master-volume-size=10 --node-volume-size=10 \

--dns-zone=k8sb14.xyz --dry-run --output yaml

kops create cluster --name=k8sb14.xyz \

--state=s3://k8sb14.xyz --zones=us-east-1a,us-east-1b,us-east-1c \

--node-count=3 --master-count=1 --node-size=t3.medium --master-size=t3.medium \

--master-zones=us-east-1a --master-volume-size=10 --node-volume-size=10 \

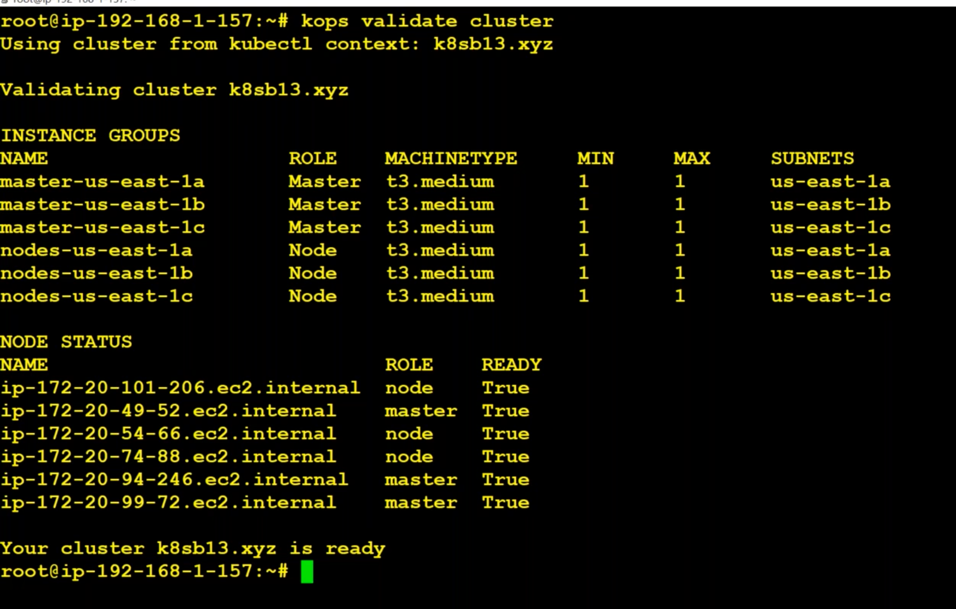
--dns-zone=k8sb14.xyz --yes

**Command to validate cluster :**

kops validate cluster

output after successful cluster creation:

It is showing 3 master and 3 workers nodes have been created.



**To wait and see until cluster gets created**

kops validate cluster –wait 10

**To delete a cluster:**

kops delete cluster <cluster\_name> --yes

kops delete cluster k8sb14.xyz --yes

**How we came to know when the cluster gets created?**

We can check the ips of instances are there in the Hosted zone details like below

Graphical user interface, application

Description automatically generated

what are the alias for kubectl ?

to know more go to Kubernetes docs and type bash completion in search bar.

echo 'source <(kubectl completion bash)' >>~/.bashrc

echo 'alias ku=kubectl' >>~/.bashrc

echo 'complete -F \_\_start\_kubectl ku' >>~/.bashrc

Here we are just adding some commands/lines in .bashrc file

what are default namespaces?



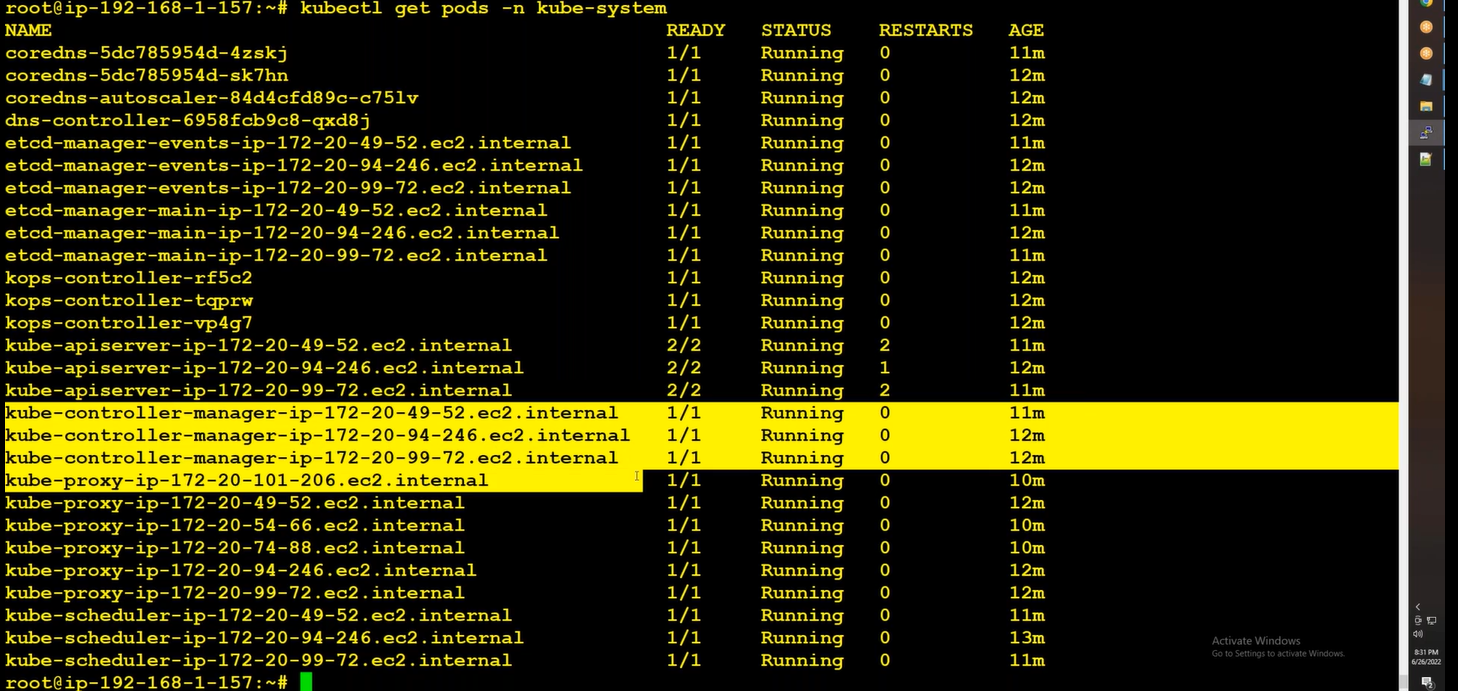
How to solve the issues ?

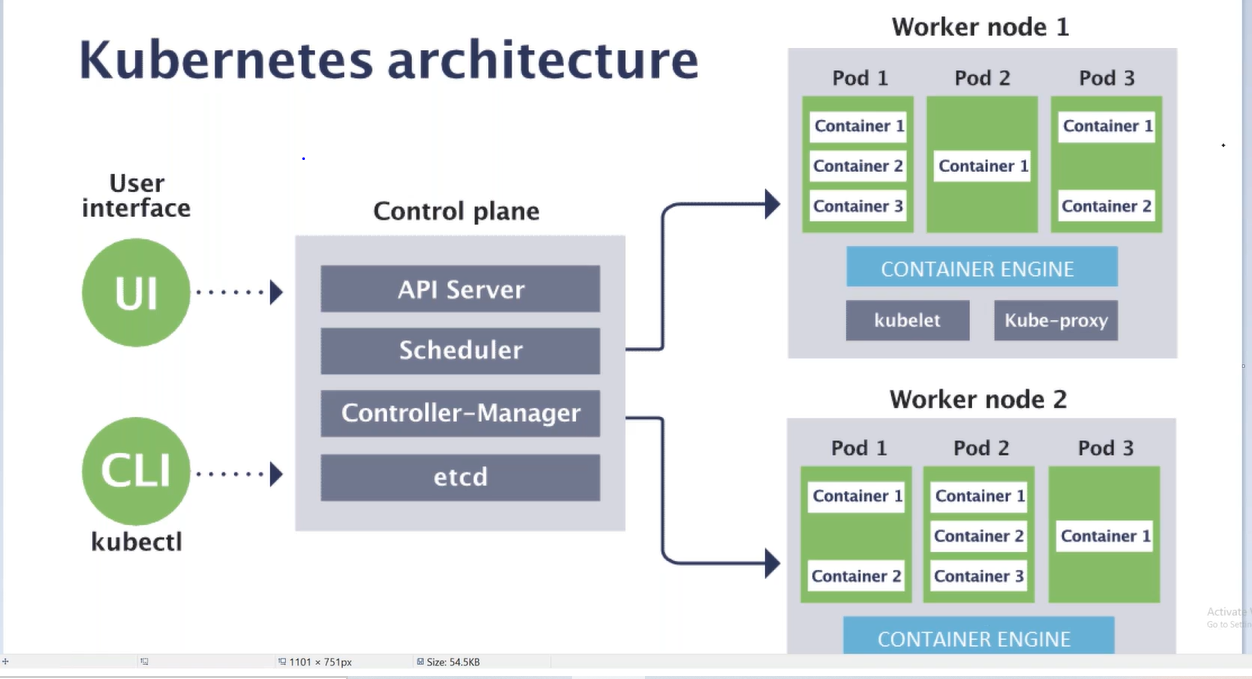
We raise a ticket and sent dump to them.

Kubectl cluster-info dump

What are running on control plane and worker node ?

kubectl get pods -n kube-system





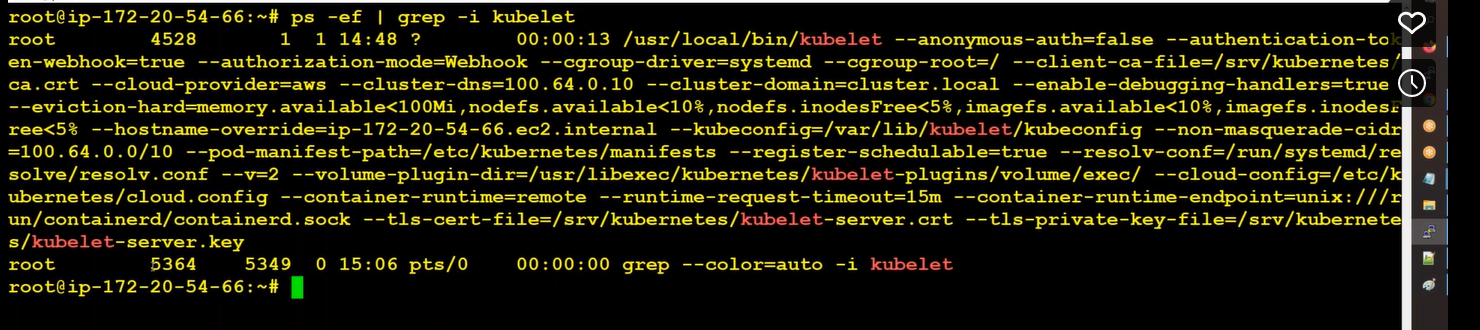
We have three masters and three worker nodes.

So we have three schedulers, apiservers, control-managers,

We should have three kube-proxy. But we have 6 kube-proxy because master also treated as node. So it also contains kube-proxy. That why it is showing 6 kube-proxy. Kube-proxy runs on all the nodes in the cluster no matter it is master or worker.

Kubelet will not be shown there since it is a service not a pod/container. It runs as demon.

Check kubelet running on the server node ?

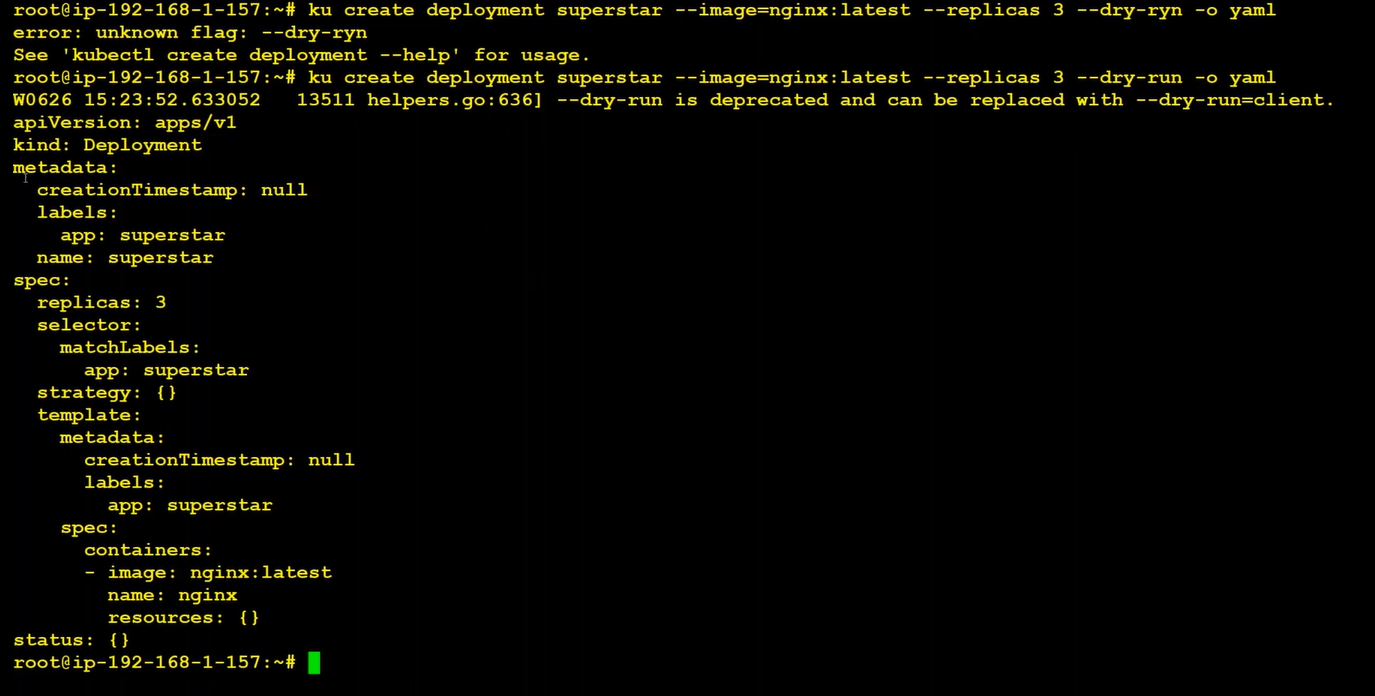


**To switch namespace from default to kube-system:**

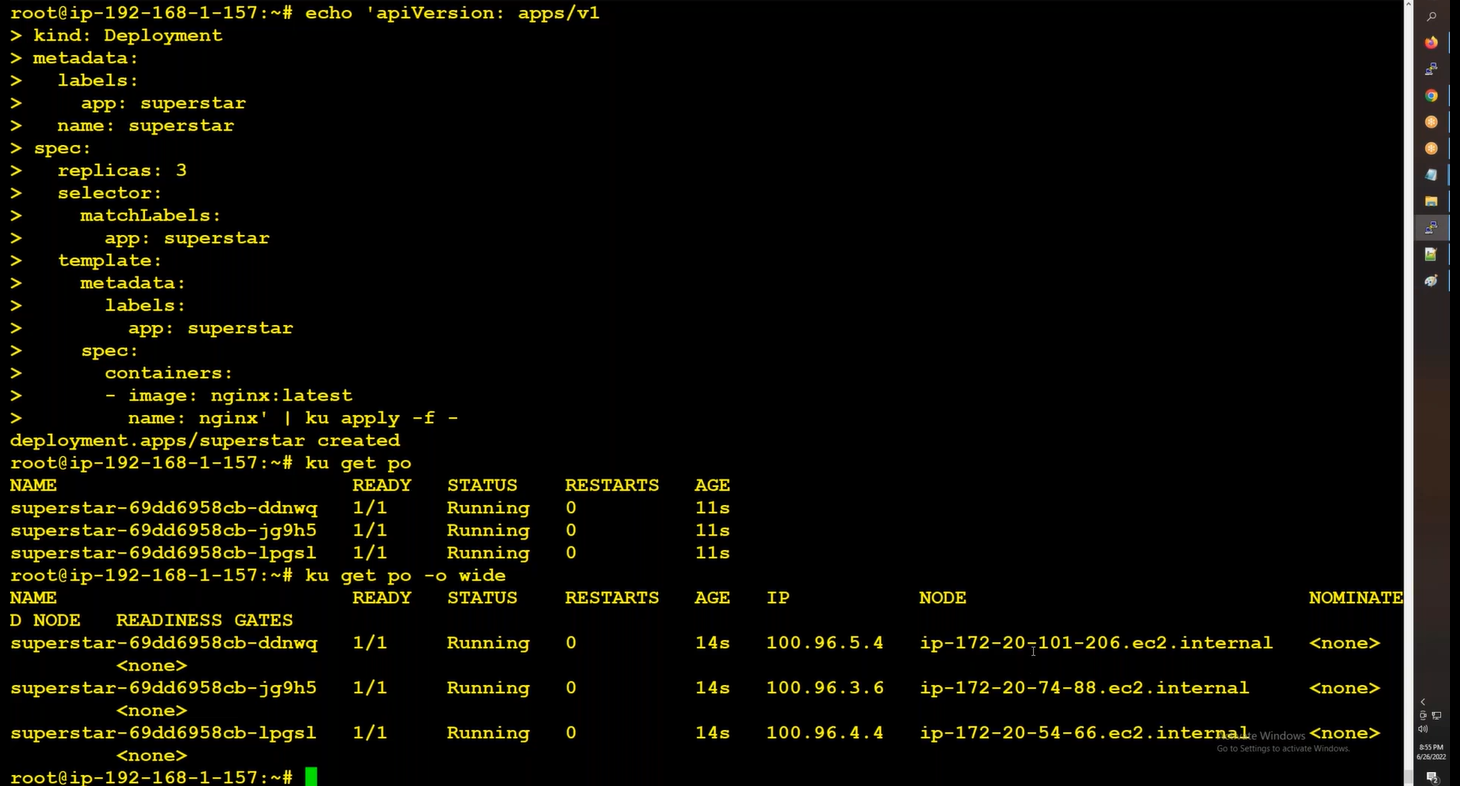
kubectl config set-context –current –namespace=kube-system

Kubens is used to switch namespaces

If we were given an image to deployvia yaml, we should choose easy way instead of writing yaml.



Copy the content into yaml file and then running it also takes time. So we can go with

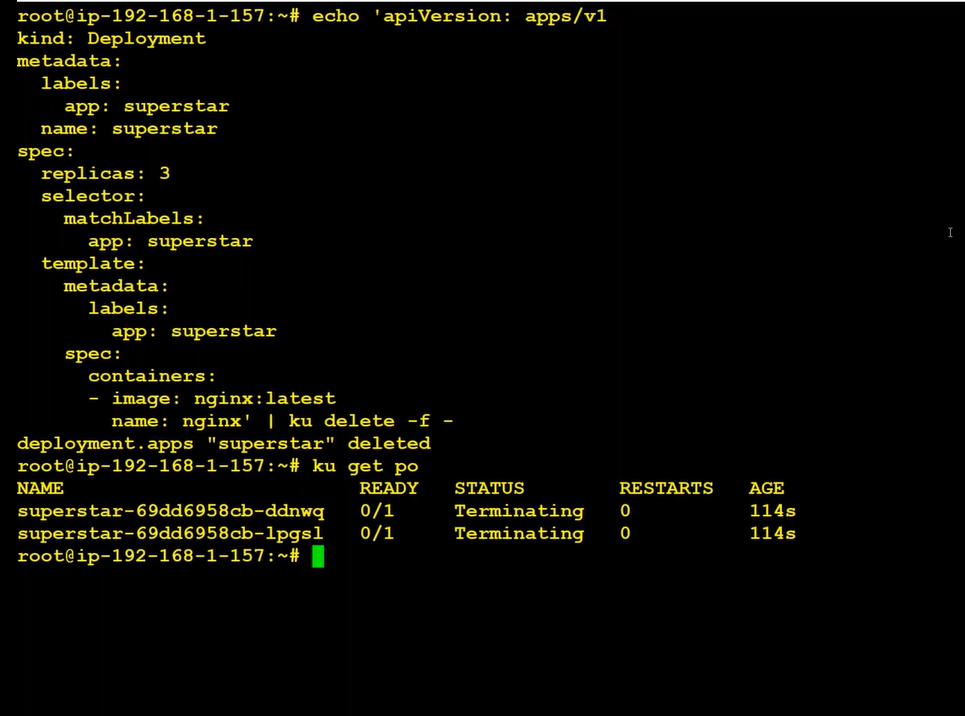


Each pod is running on different worker nodes

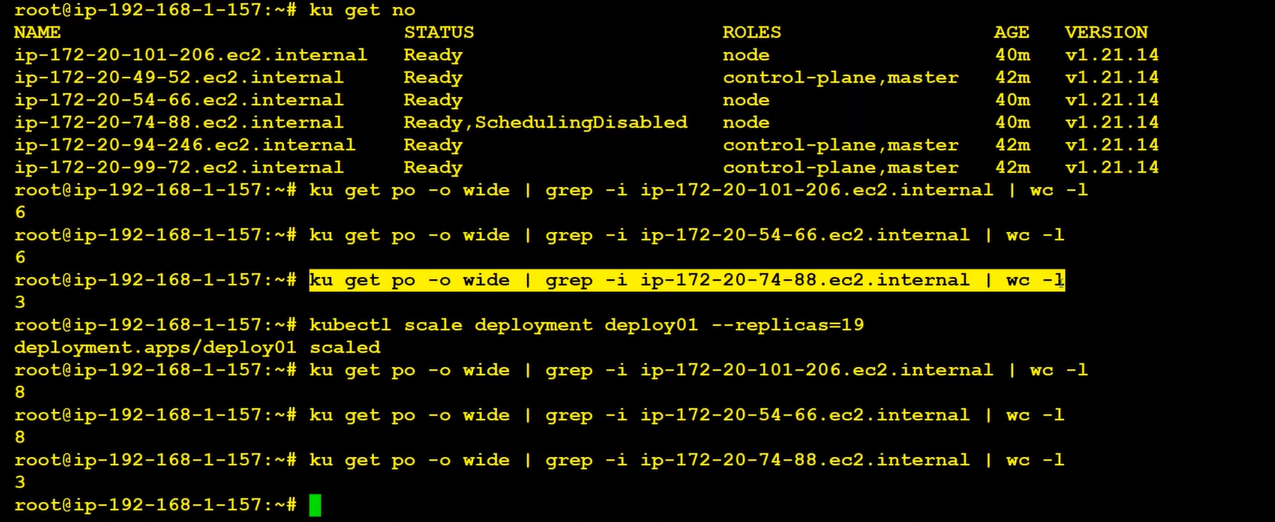
Running through command is Imperative

Running through manifest is Delarative. We prefer Delarative since it gives us the version control and keep track of changes.

To delete the deployment, we can follow the same



If we use cordon on any of the node. We are not allowed to run any new pods on that node



drain deletes the all the pods from the node It won’t allow new pods to create on it

ku uncord <ip-address of the node>

If we delete/terminate the nod emanually the kops-controller is responsibe for keep track of these things. And this will deploy the node again