Split the screen Horizantal

Screen

Ctrl + a (Then type caps S)

Ctrl + a : split (it will show 3 pieces of screen )

Vertical split

Ctrl + a | (pipe)

Ctrl + a : split –v (3 vertical line )

Ctrl + a : focus

Write inside the vertical line

Ctlr+a c

Move one panel to another panel

Ctrl+a tab

Remove the split

Ctrl +a (Then caps X) or go to that panel Ctrl + a : focus , then ctrl+a :remove

Work normal screen

Ctrl+a : only

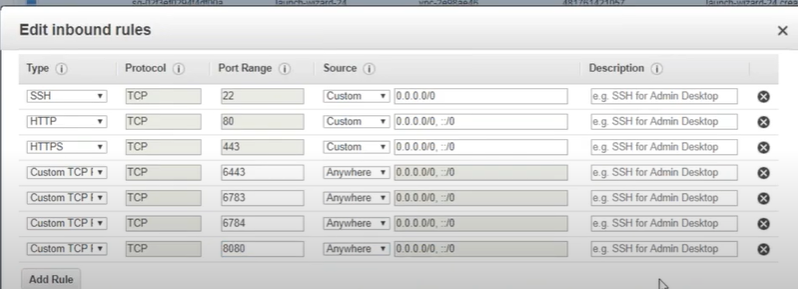
Kubernetes installation on Kubuntu 18.04 on AWS cloud (Date-2.5.2020)

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

Step-1

**Install 3 instance on aws cloud and make one master and 2 worker node**

**Step-2 Open the ports**

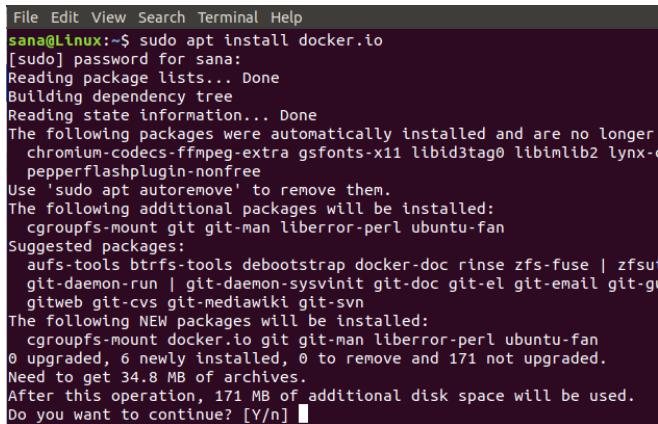
****

**Give the unique name to the both host**

hostnamectl set-hostname master-node

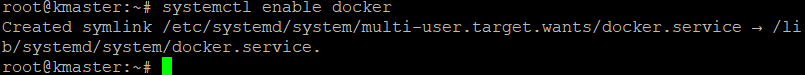
Install Docker on both the nodes

**Sudo apt install docker.io**

****

****

### Enable Docker on both the nodes

****

### Add the Kubernetes signing key on both the nodes

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

****

**It will show**

****

If Curl is not installed on your system, you can install it through the following command as root:

Apt install curl

### Add Xenial Kubernetes Repository on both the nodes

****

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

****

### Install Kubeadm both the node

### apt install kubeadm

### 

### Check the version

### 

### Disable swap memory (if running) on both the nodes

### 

### Initialize Kubernetes on the master node

### Run only this command into the master node

### sudo kubeadm init --pod-network-cidr=10.244.0.0/16

### 

### Please note down the following information from the output

### 

### Now run the commands suggested in the output in order to start using the cluster:

### 

### 

### Deploy a Pod Network through the master node

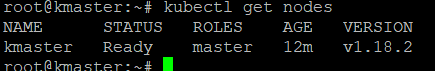
### kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

### 

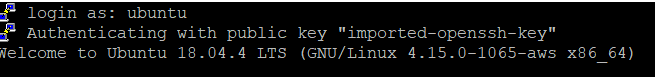
### Use the following command in order to view the status of the network:

### 

### Now when you see the status of the nodes, you will see that the master-node is ready:

****

### Add the slave node to the network in order to form a cluster

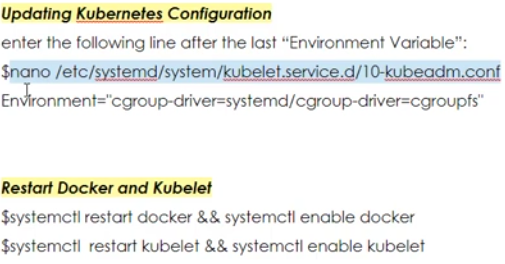
****

**Update kuubernetes configuration**

**Vim /etc/systemd/system/kubelet.service.d/10-kubeadm.conf**

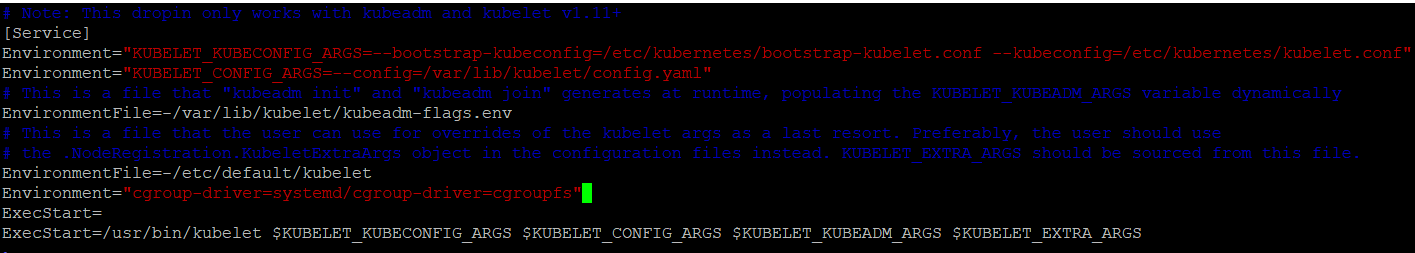
**Environment=”cgroup-driver=systemd/cgroup-driver=cgroupfs”**

****

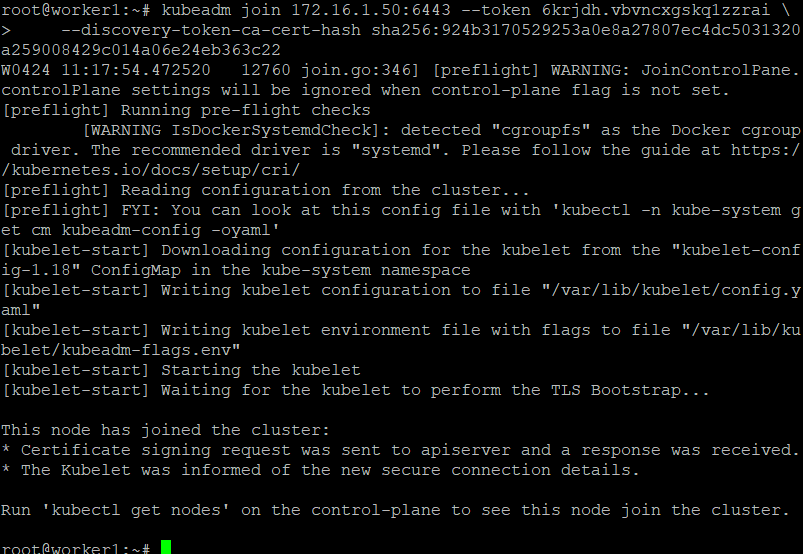
****

**Enter the following line the Last “environment variable”**

**Environment="cgroup-driver=systemd/cgroup-driver=cgroupfs"**

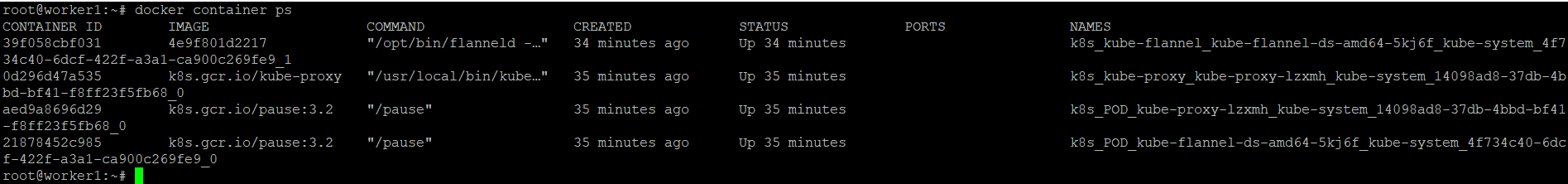
****

**Now join the worker node to master using the token**

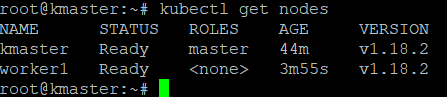
****

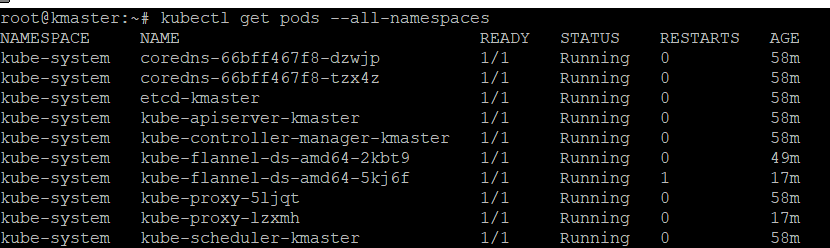
**Check in worker node what are the pods are running**

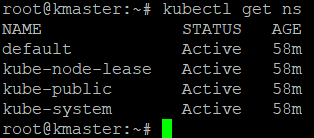
**Docker container ps**

****

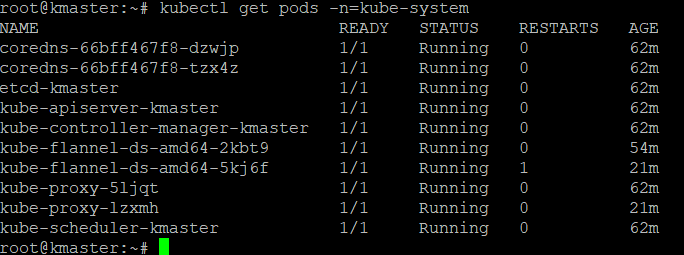
**Now check in master node**

****

****

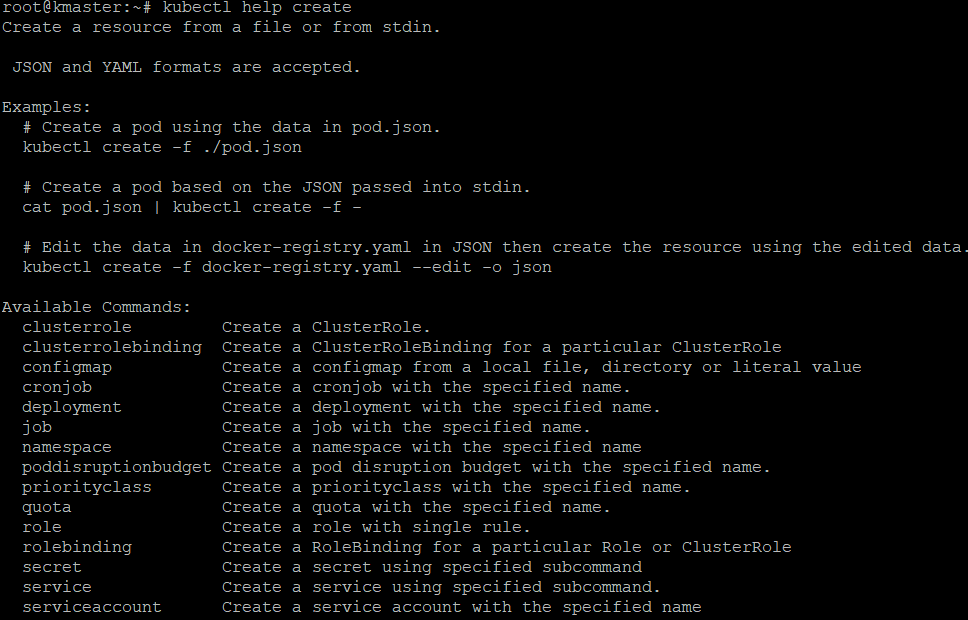
****

**These are below pods are running managing the cluster**

****

**Now create the pod using the cmd liner & manifest file**

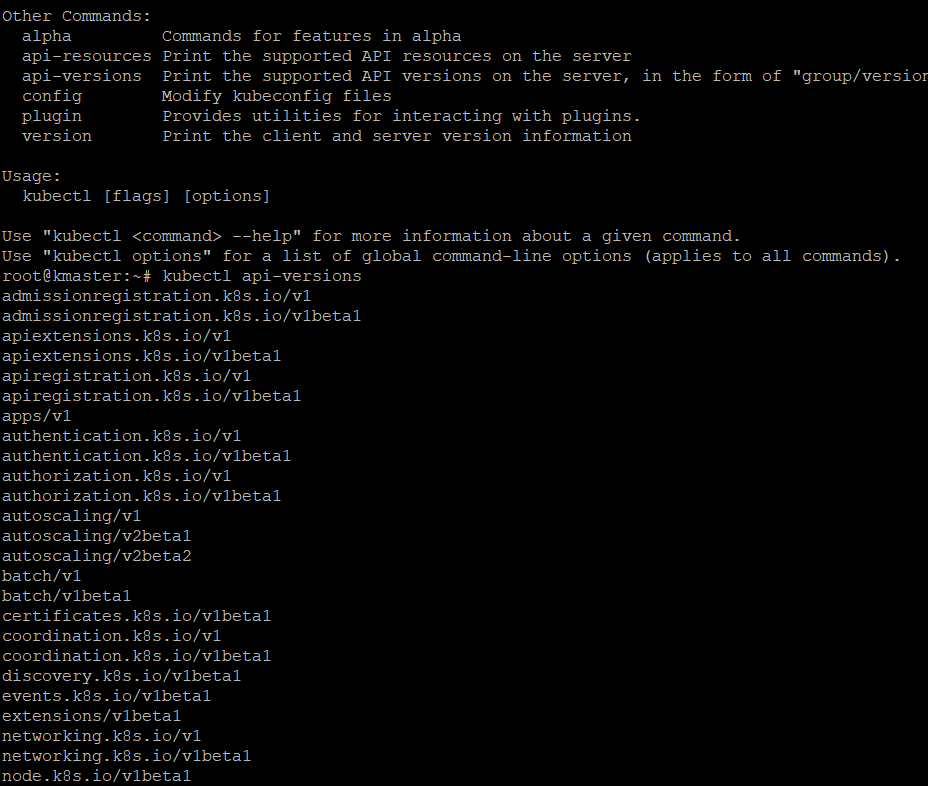
**How to know**

****

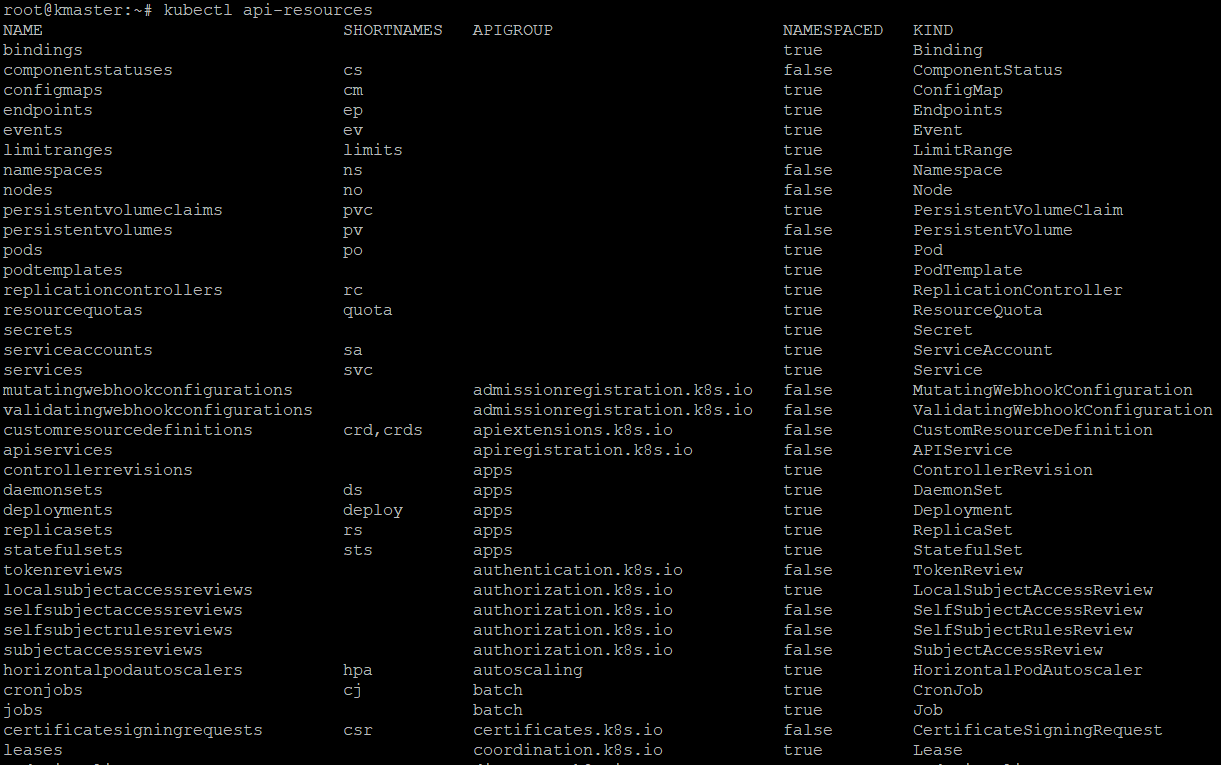
**Pod is not available so that is the deployment ,using deployment we will create the pod**

**How to know the API version**

**Kubectl**

****

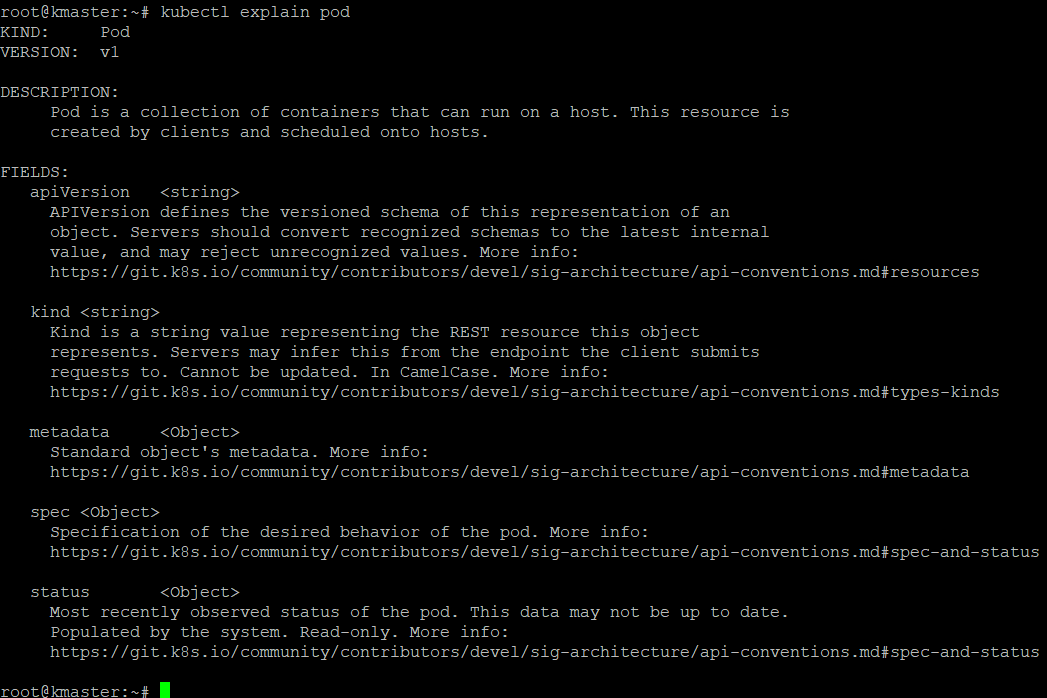
**API-resources (print the supported API resources on the server )**

****

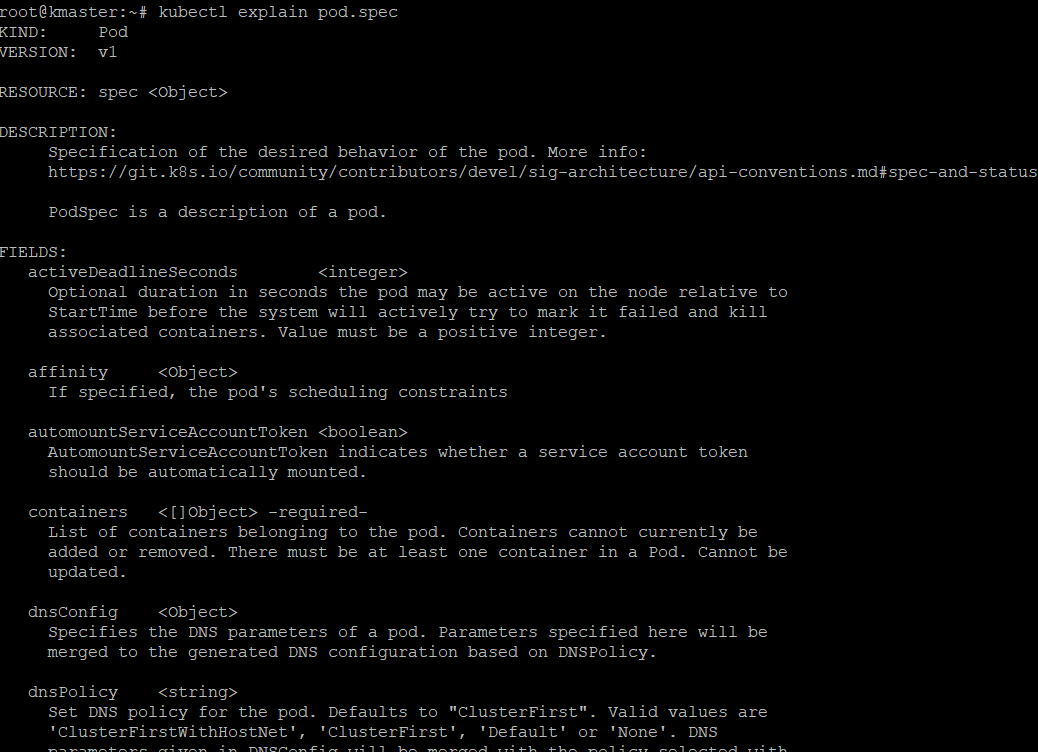
**Namespace & kind is important here ,u find here which resources is which one**

**How to know a fields of each Api resources ? ( want to know th pod )**

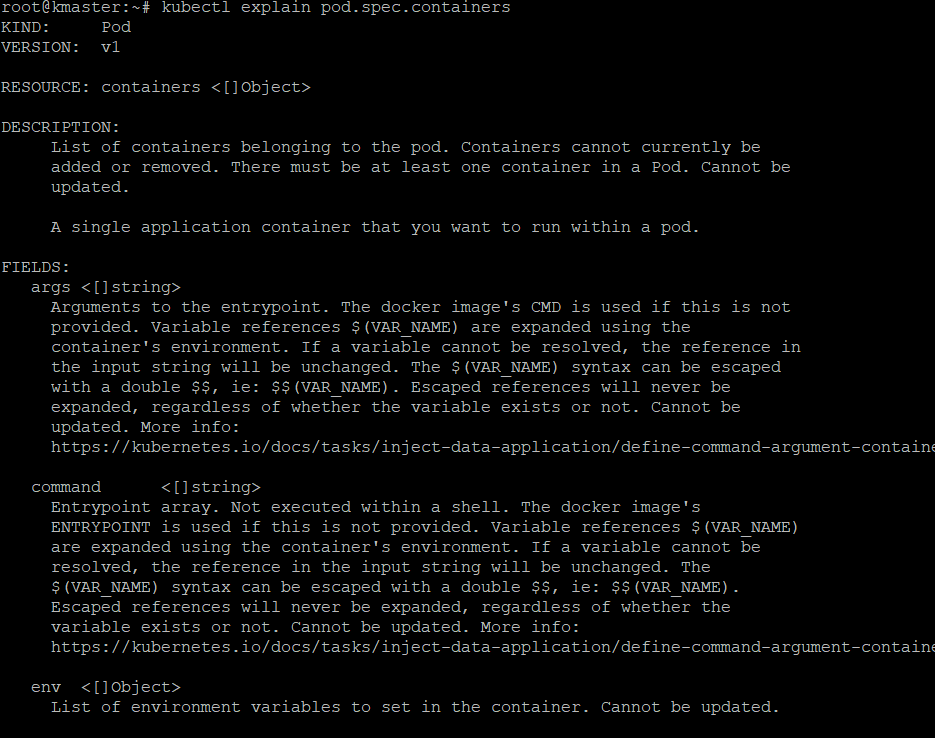
**Kubectl explain pod**

****

**If u want to know more behavior of Spec (kubectl describe pod.spec)**

****

**If u want to know more inside the spec ( kubectl explain pod.spec.containers )**

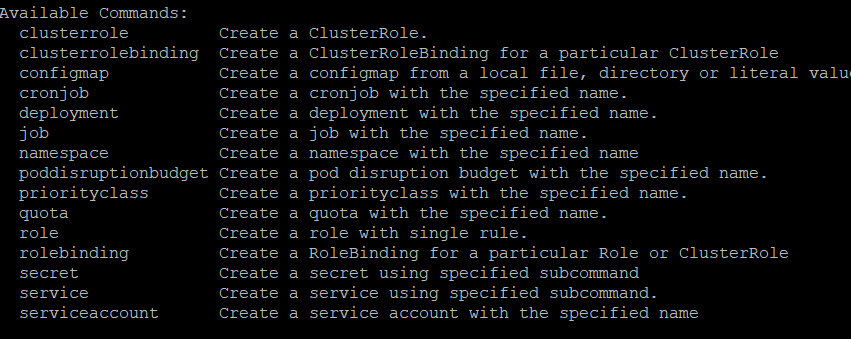
****

**How is very imp go get the object**

**DEPLOYMENT**

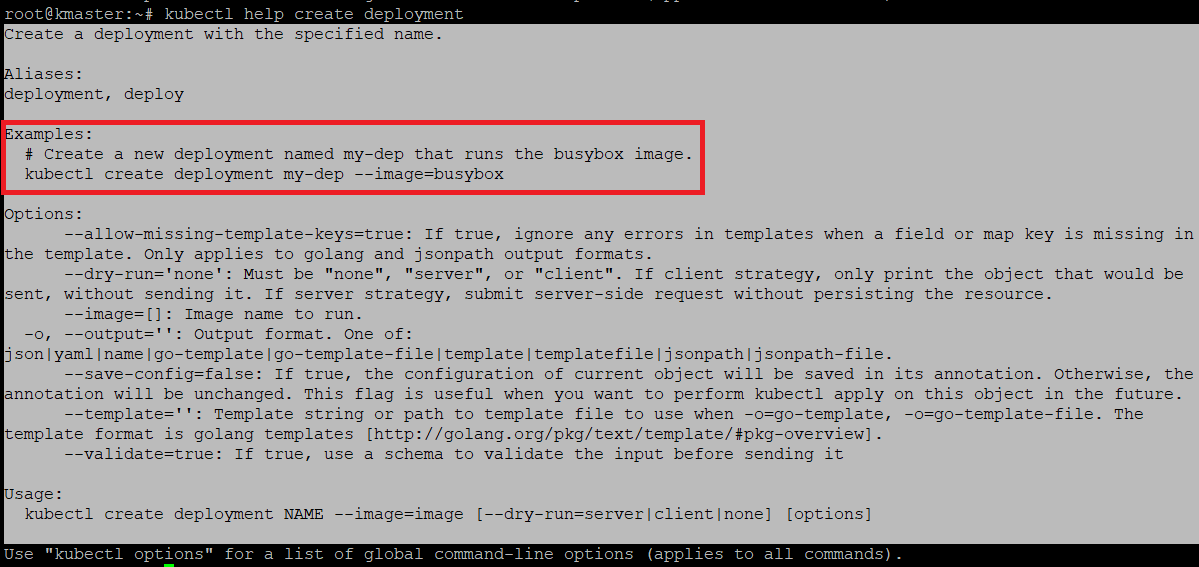
**Find the create approach**

**Kubectl help create**

****

**These are thiung we can create**

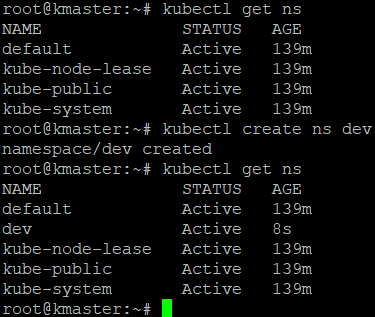
**My requirement here is deployment (kubectl help create deployment )**

****

**We can Deploy 2 ways 1- cmd 2- yaml file**

**Now I am using cmd line – kubectl create deployment my-nging --image=nginx**

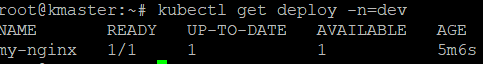
**Create namespace & inside namespace create pod**

****

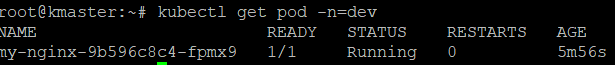
**Create**

****

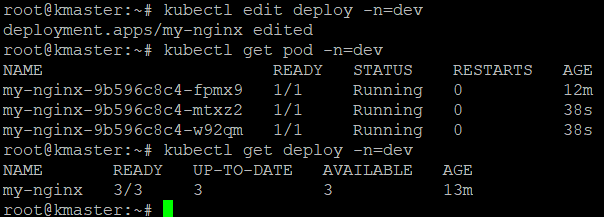
**Now check**

****

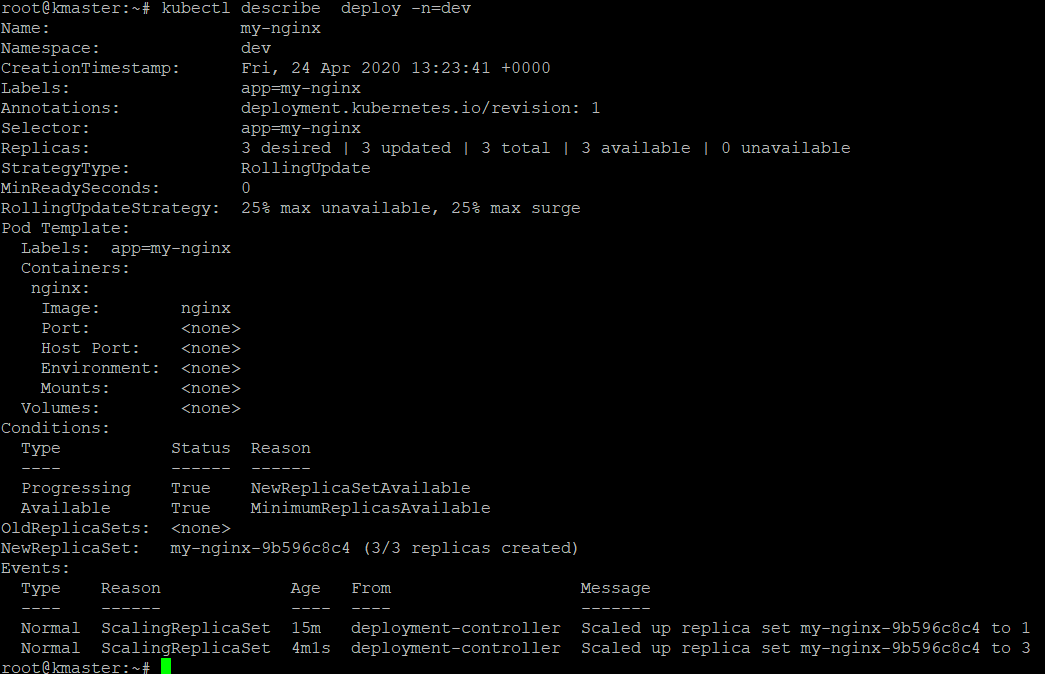
**Now check the pod**

****

**Now edit the deployment , I edited here replica**

****

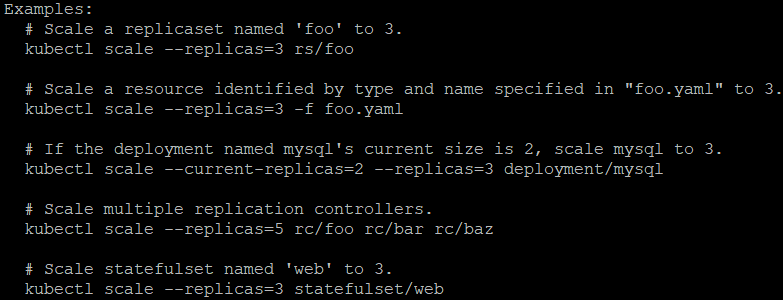
**Now describe**

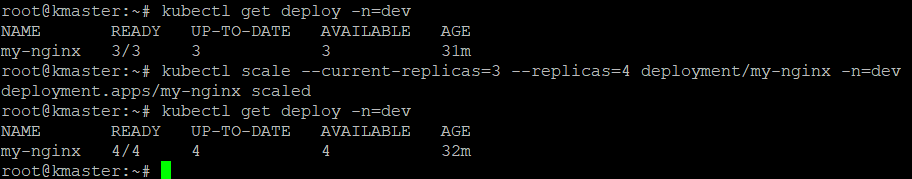
****

**Update the deployment ( I am skiping her because I did not created it using yaml file )**

**SCALE**

****

****

****

**Check controller is working or not for that delete one pod**

****

**Rolling Update**

**Before rolling update check the version**

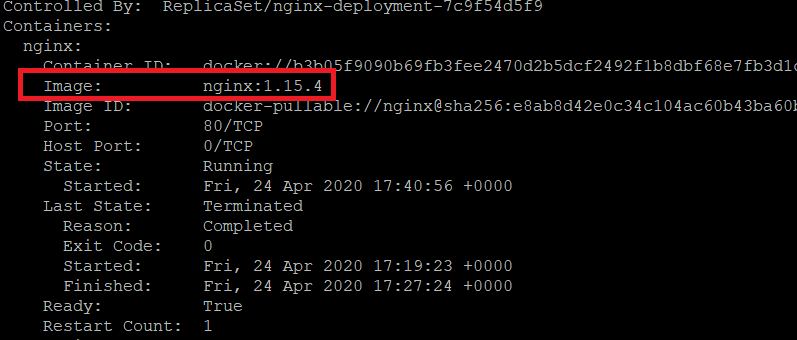
**Checking versioning steps**

****

**Curl** <http://10.244.1.17>

**Or**

****

****

**It will show the version**

**Now roll update**

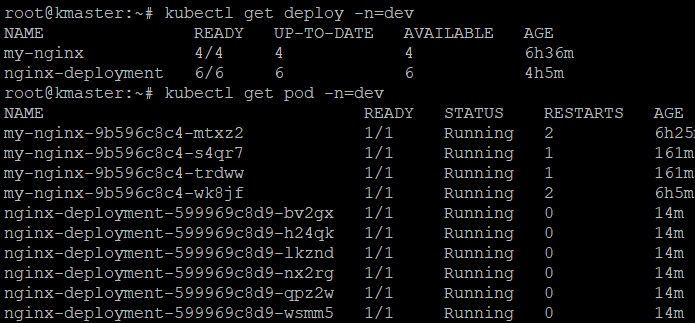
**Create a new yaml file or edit in that exist file ,Here I created the new manifest file (nginx1.yaml)**

**Kubectl apply –f**

****

**It will give Warning message leave that message**

**Check now**

****

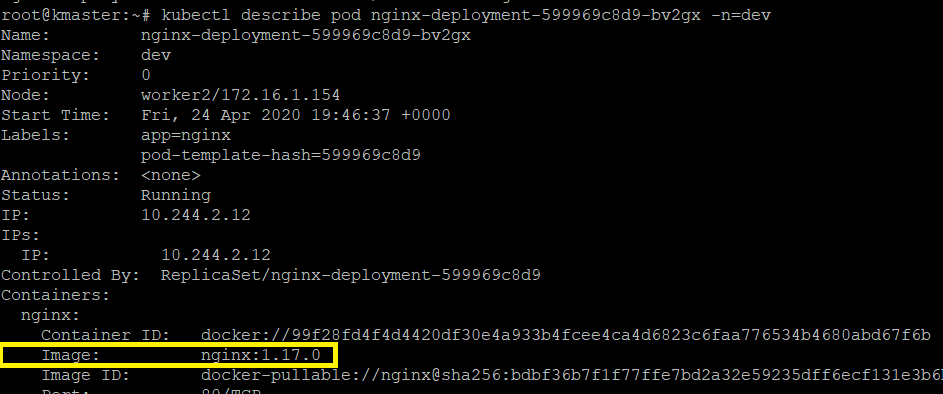
**Now check versioning**

**Checking versioning steps**

****

**Curl** <http://10.244.1.17>

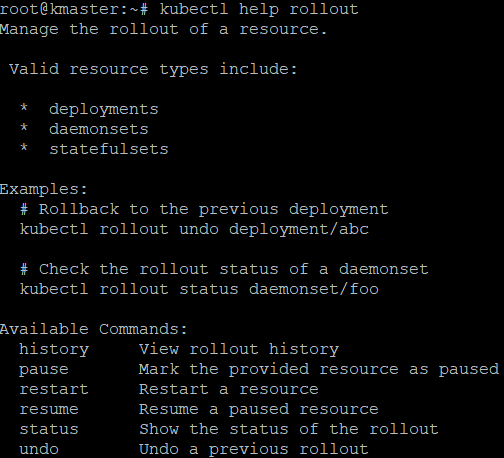
**Or**

****

**How I can check the version & rollback**

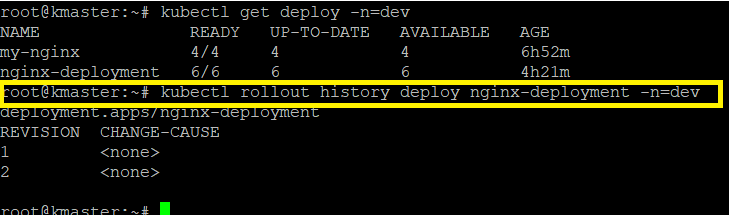
**Kubectl**

**Kubectl help rollout**

****

**Below command will help me show the version**

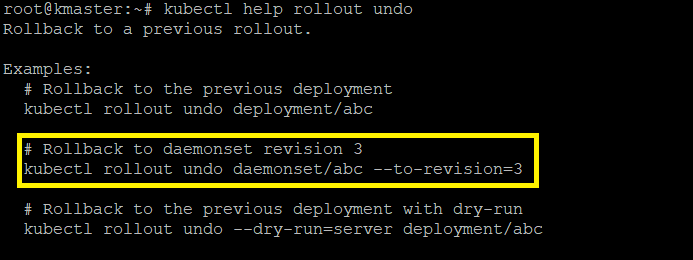
**Kubectl rollout history deploy nginx-deployment –n=dev**

****

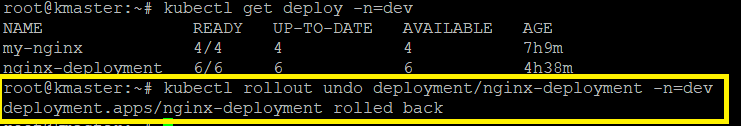
**Here I am seeing two version . Now u are in two version (Curl** <http://10.244.1.17> **)**

**How we can rollback ?**

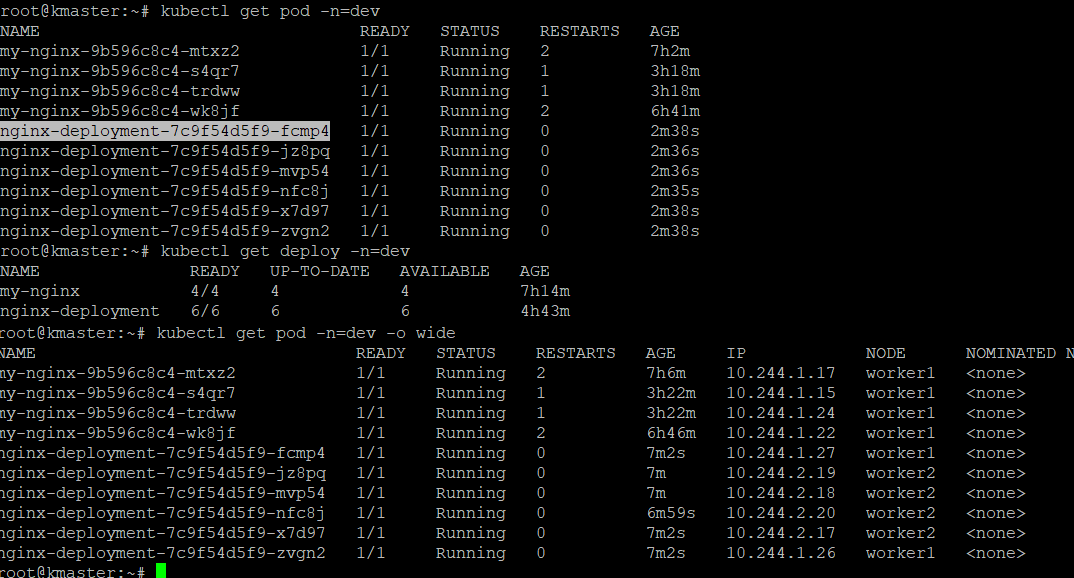
**Kubectl help rollout undo**

****

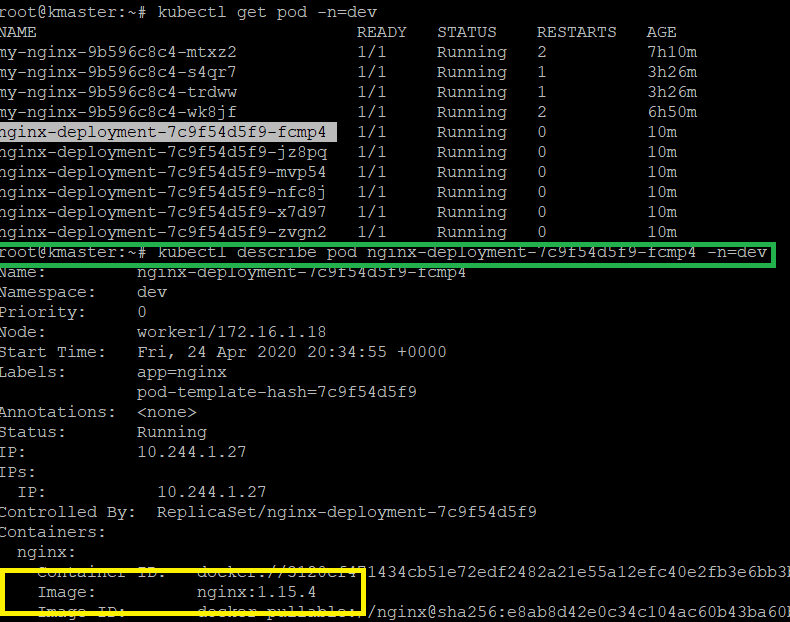
**Kubectl rollout undo deployment/nginx-deployment --to-revision=1 –n=dev**

****

**It is rolled back ,check now**

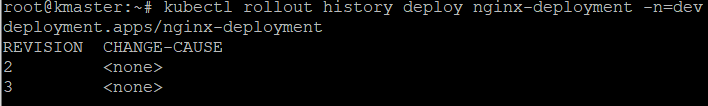
****

**Curl** <http://10.244.1.27> **(it will show the version ) or**

****

**Above snapshot u can find the older version**

**Check history now**

****

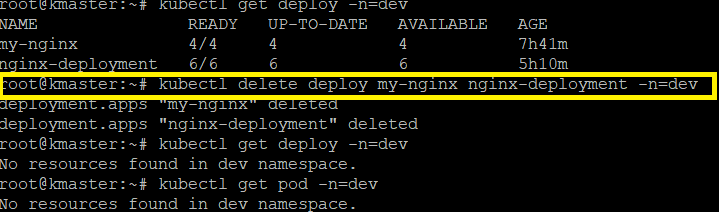
**Right know 3 is working**

**This is the way we can rollback**

**DELETE the deployment**

**Kubectl get deploy –n=dev**

**Kubectl delete deploy <deployname> -n=dev**

****