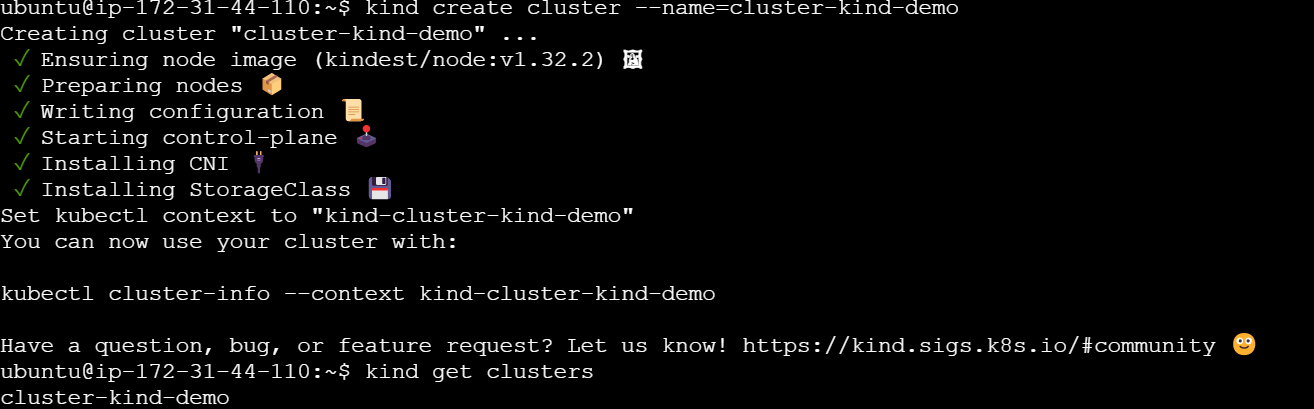
**Create Cluster**

kind create cluster --name=cluster-kind-demo

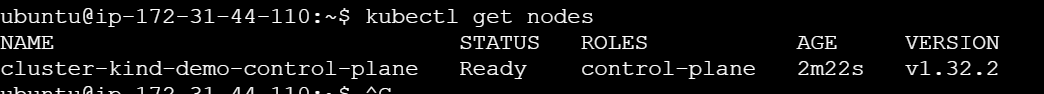
**get all clusters**

kind get clusters

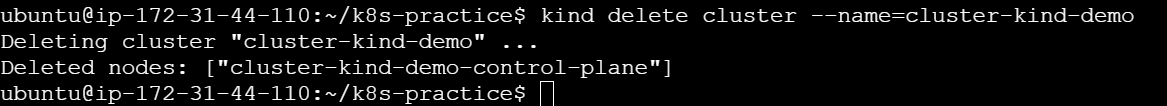


**get all nodes—**here only control pane created which contals scheduled, api server, etcd,

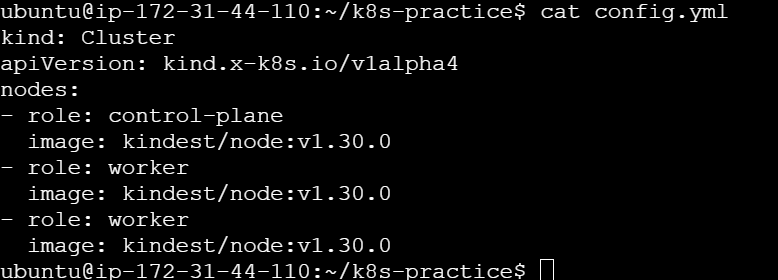
kubectl get nodes



Delete Cluster-



Cluster creation using yml file



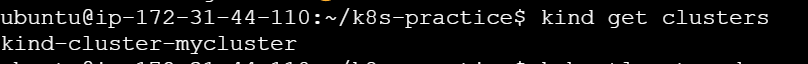
Cmd-

kind create cluster --name=kind-cluster-mycluster --config=config.yml

here preparing 3 nodes

A screenshot of a computer

AI-generated content may be incorrect.



Pods not displayed because pods not created yet, pod is smallest unit of Kubernetes container in which we ran the docker containers

only nodes are displayed one is control plane and 2 are workers

A screen shot of a computer screen

AI-generated content may be incorrect.

**Namespace – is group**

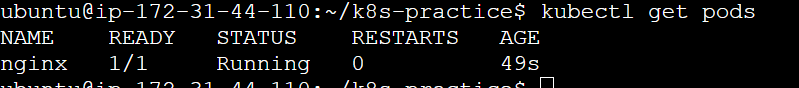
A screen shot of a computer

AI-generated content may be incorrect.

**Pod creation using cmd-**

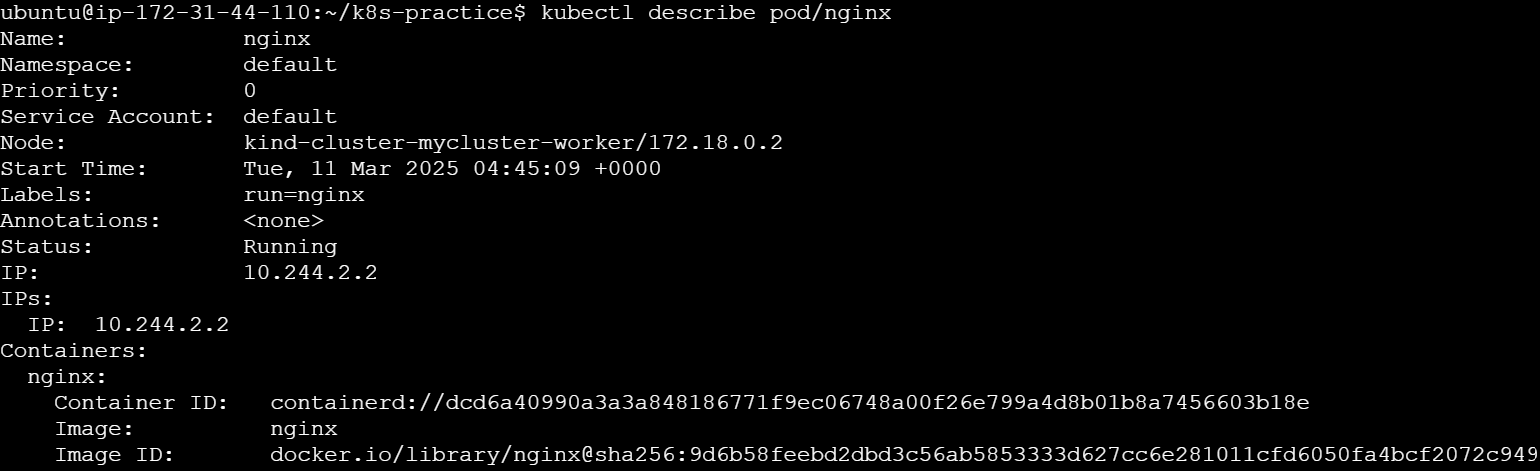
kubectl run nginx --image=nginx





To see all infor-

kubectl describe pod/nginx



Events performed while creating pod

A black screen with white text

AI-generated content may be incorrect.

**Namespace –**

**A black screen with white text

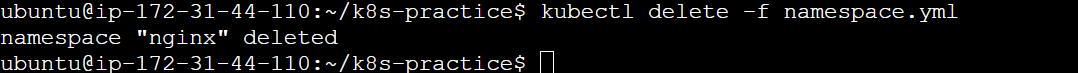
AI-generated content may be incorrect.**

kubectl apply -f namespace.yml



**Delete namespace-**

kubectl delete -f namespace.yml



**Pod-***Pods* are the smallest deployable units of computing that you can create and manage in Kubernetes.

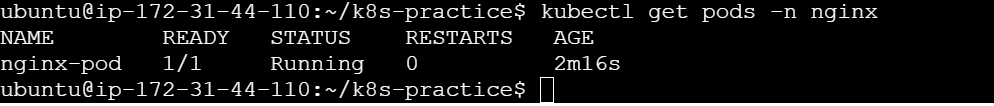
**A screen shot of a computer

AI-generated content may be incorrect.**

To check pod.yml file correct so doing dry run-

****

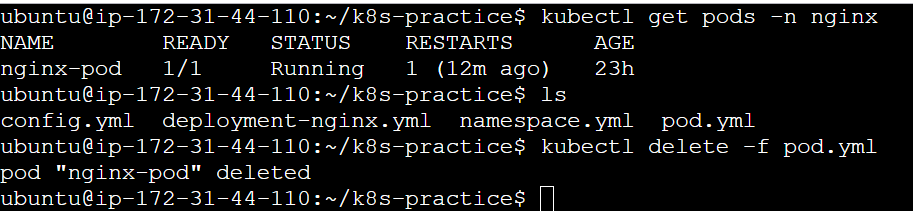
****

****

**Deployment (desired set of pod)-**

A Deployment manages a set of Pods to run an application workload, usually one that doesn't maintain state. A *Deployment* provides declarative updates for [Pods](https://kubernetes.io/docs/concepts/workloads/pods/) and [ReplicaSets](https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/).

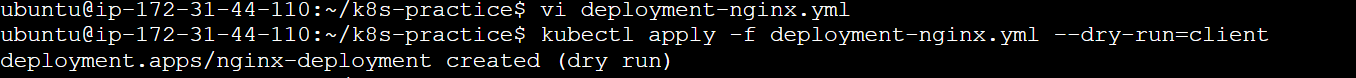
In deployment pod template written so earlier pods we have to create in namespace



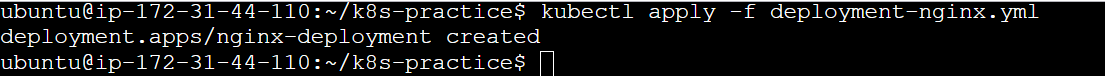
Deplyement yml file

A screenshot of a computer

AI-generated content may be incorrect.



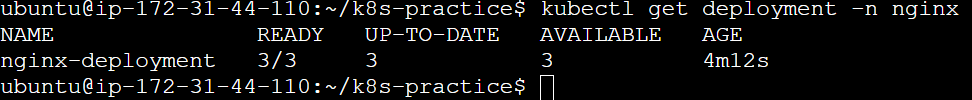
kubectl apply -f deployment-nginx.yml



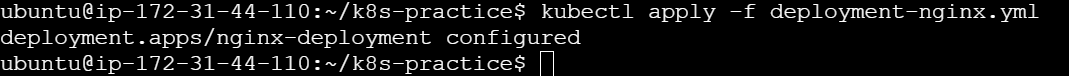
3 replica written in yml file so created 3 pods replica

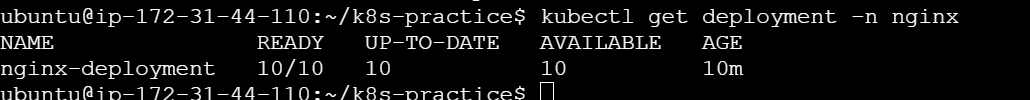
A screen shot of a computer

AI-generated content may be incorrect.



If changing replica in yml file as 10 then



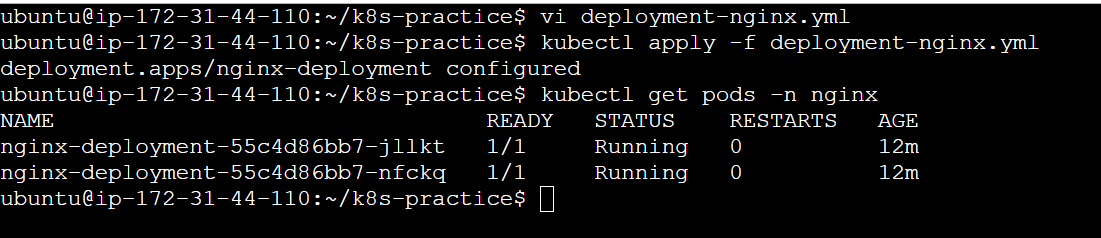


Earlier 3 have added new 7

A screen shot of a computer program

AI-generated content may be incorrect.

Again configured into 2

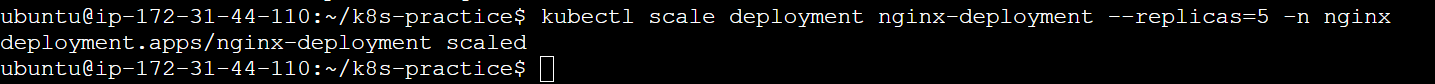


If one of pod is auto delete or manually deleted , new pod created and running called **auto healing**



We can scale deployment, **auto scale**

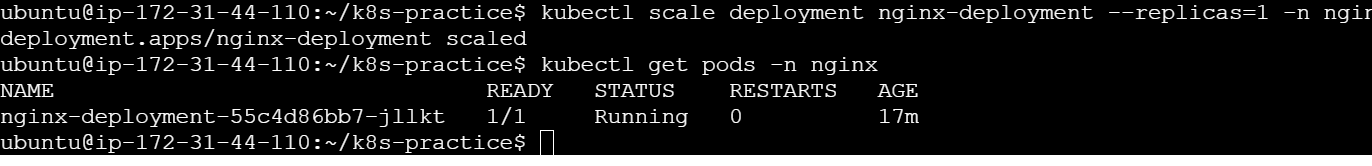
kubectl scale deployment nginx-deployment --replicas=5 -n nginx

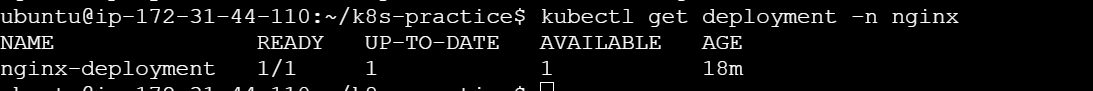


A screen shot of a computer

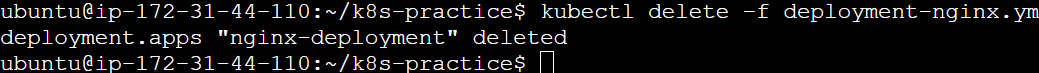
AI-generated content may be incorrect.

Again scaled to 1 replica

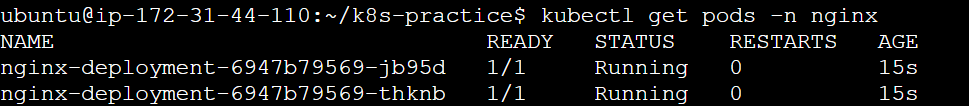




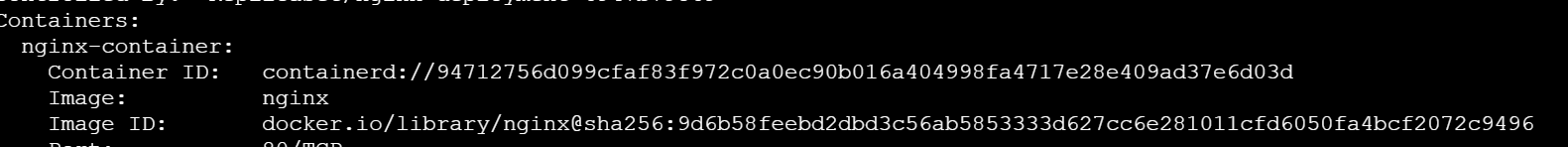
Delete deployment-

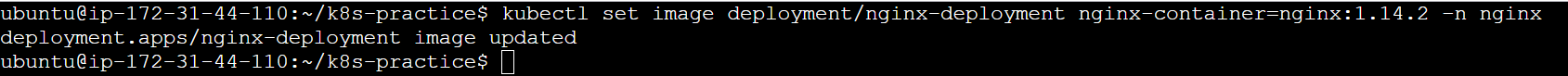


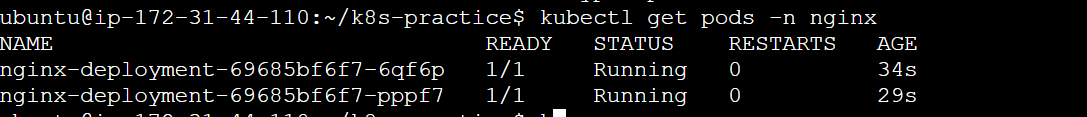
If updated imgae

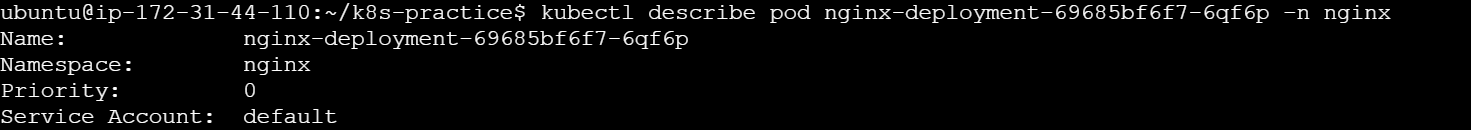


Changed container name- nginx-container

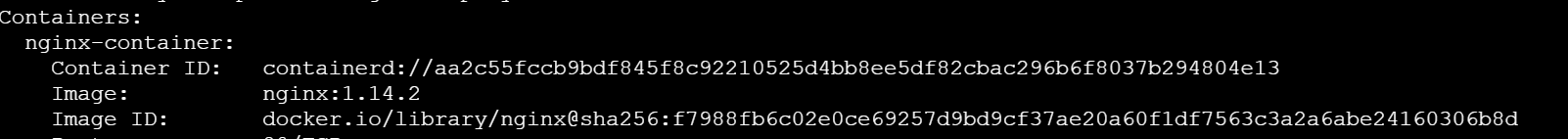








This image is done one by one on pod this will not on parallel or directly on two pod because this will increase downtime. This method is called rolling update



-w is watch mode

To run nginx in pod

kubectl exec --stdin --tty nginx-deployment-69685bf6f7-pppf7 -n nginx -- /bin/bash

or

kubectl exec --i –t nginx-deployment-69685bf6f7-pppf7 -n nginx -- /bin/bash



**Replica Set-**

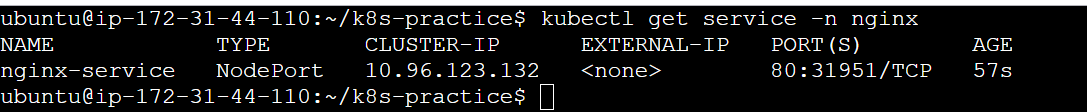
Similar to deployment and replicaset is lower version of deployment. Create the replica of pods

**Service-**

Service will be run outside the cluster

****

****

****

31951 – nodeport have specific range between 30000 to 32000

If not given in range then by default picked here 31951 picked

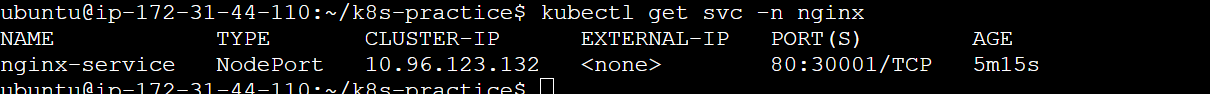
We can also provide range port in yml file

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

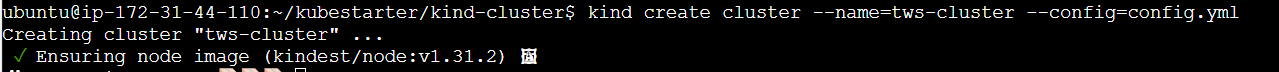


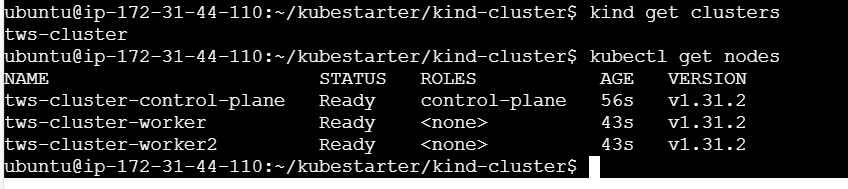
This is in docker container , not running in EC2 instance port we have to port forward

kubectl port-forward service/nginx-service -n nginx 80:80 --address=0.0.0.0

**BANK APP**

**Persistentvolumn and persistent volumn claim**

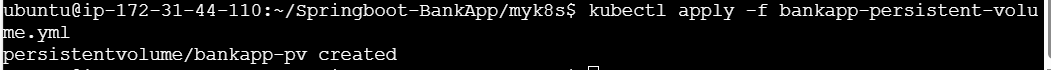




Created namespace-



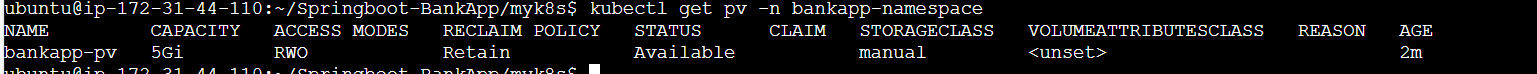
Persistent volume



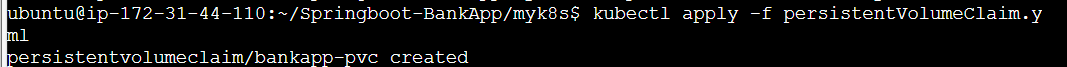
To ruan to files at a time



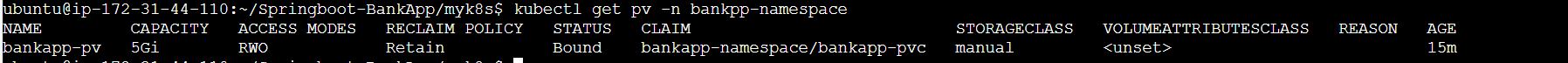
Persistent volume will be created in bankapp-namespace their status is available. Yet space is not allocated to the pod, to allocate space to pod we have to claim that space.



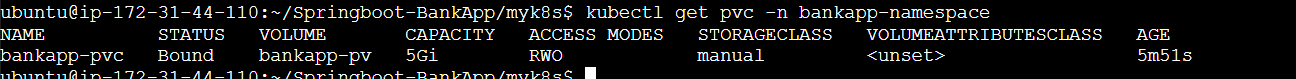
persistentVolumeClaim

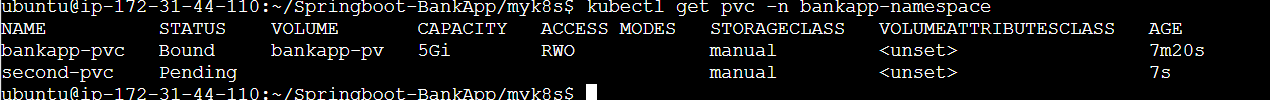


Now status changed from avaailbe to bound

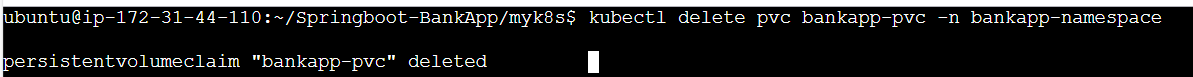


Checking Pvc

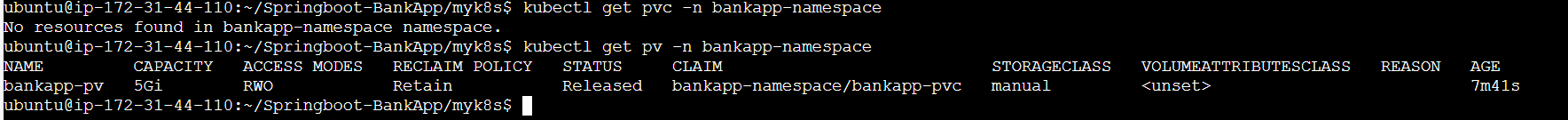


If created second which again required 2Gi storage, that is created but their status is pending because of pv don’t have spce available

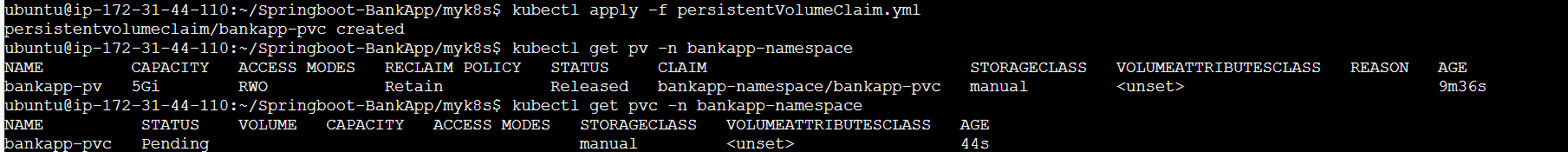
IF delete both pvc bank app-pvc and secondpvc



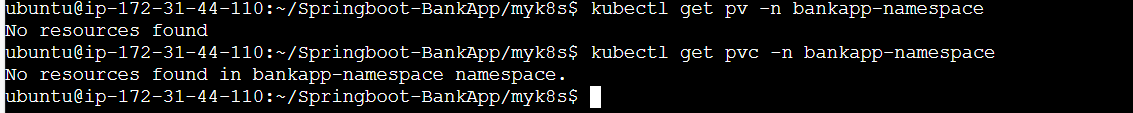
Earlier storage not deleted its status still retain not avilable



Again created pvc, its status will be pending because of space



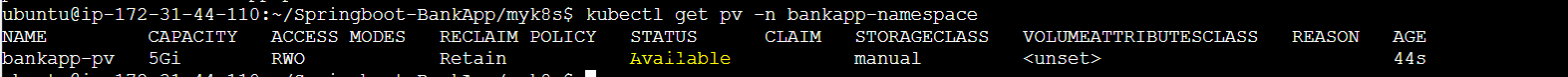
Again deleted both pvc and pv





Created pv

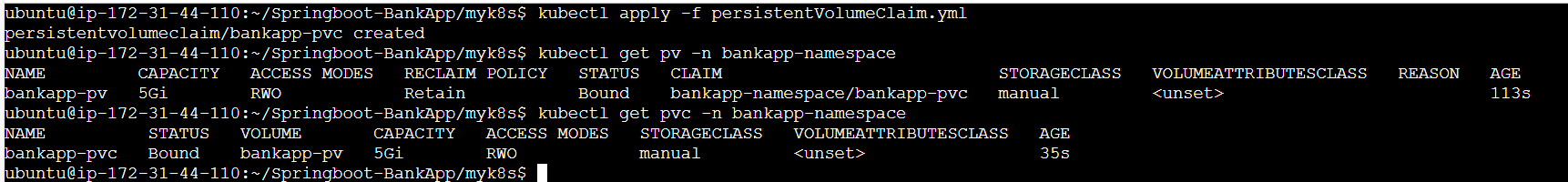




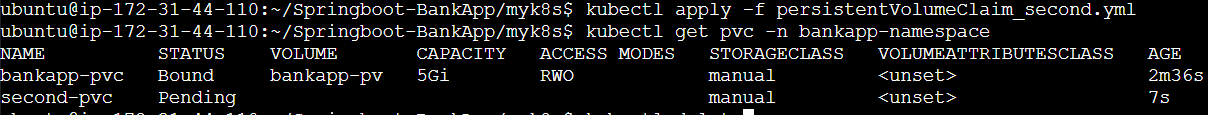
Pvc-

If pvc is 3 Gi still allocated 5Gi





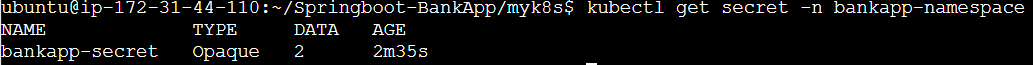
If creted second their status will be pending-



Secretes and configMap

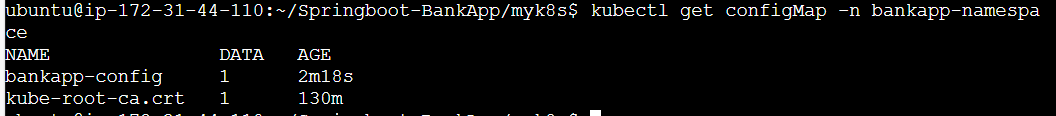
Sercret containts sercrete data like passwords, encode it into base64 cmd- echo”Tect@123” | base64





configMap containts no sercretes means database name, url





Volume Mount

Worker node(/tmp/bankapp-data)

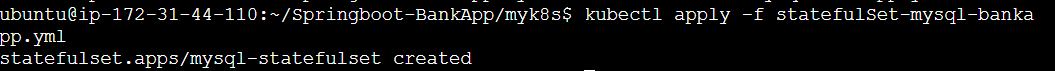
Pod(/var/lib/mysql)

Mysql-data

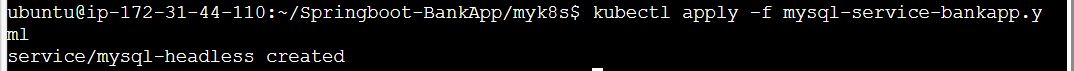
pvc

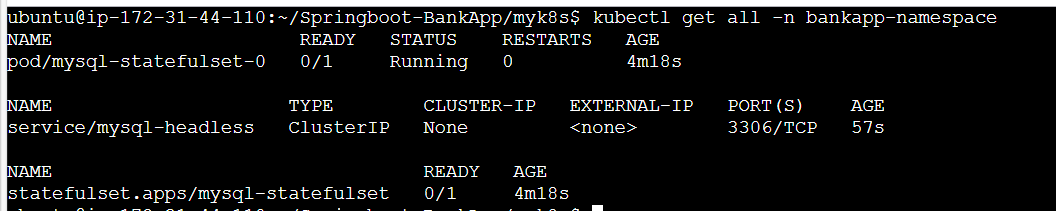
pv

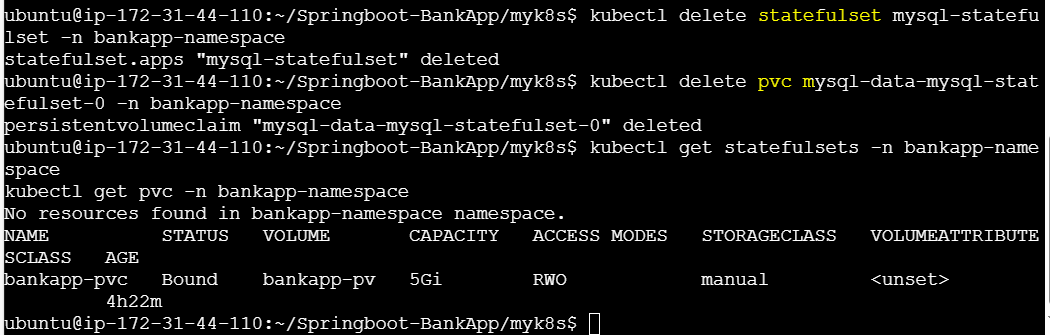
Stateful state



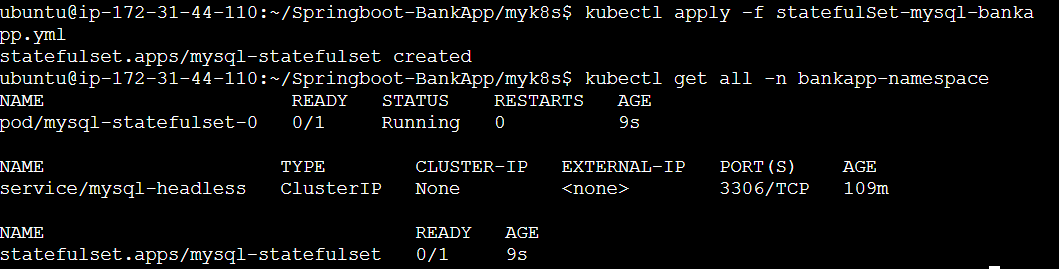
Service mysql-

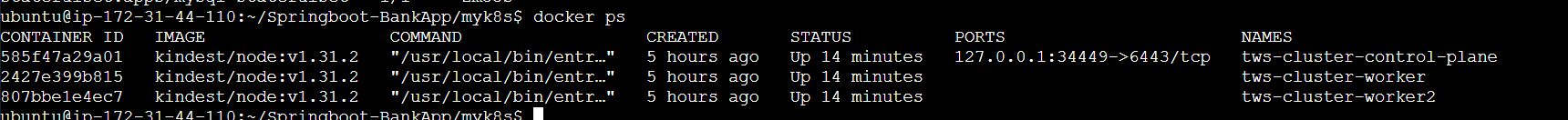




To delete stateful state we have to delete statefulset and pvc

If deleted stateful state, serice not deleted because it created separately not exposed outside cluster, only od and stateful set deleted. Here after deleted stateful set and pod recreated stateful set then new pod created and stateful set newloy created

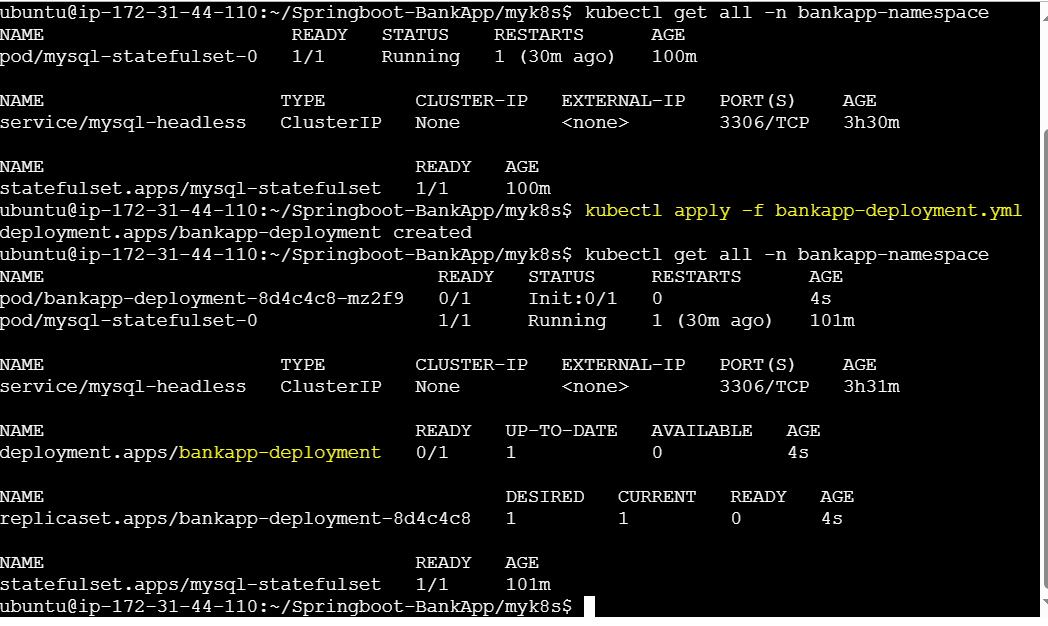


3 container running

Added database details in config map file-

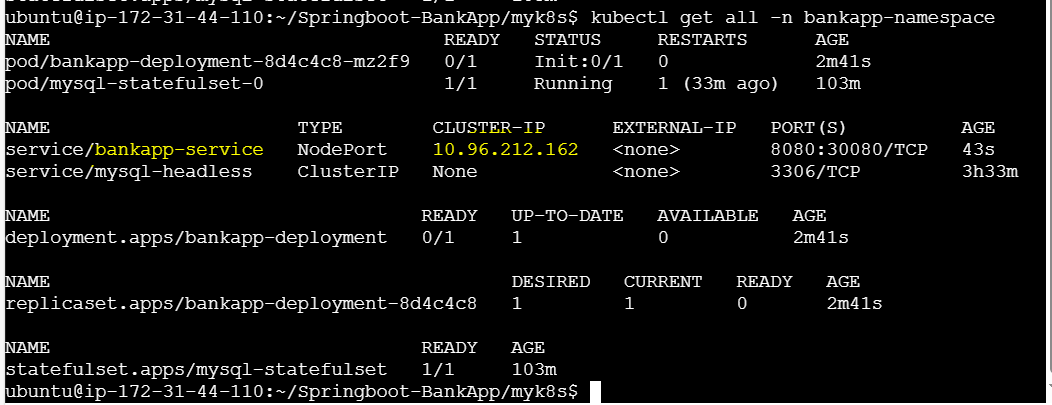


Earlier mysql service created for mysql image now one more service for bankapp with bankapp image we have to create to expose



Service-





kubectl port-forward service/bankapp-ser

vice -n bankapp-namespace 8080:8080 --address:0.0.0.0

**Persistent Volume (PV) & Persistent Volume Claim (PVC)**

PV is storage which is given from host machine support hots machine have 10 GB. Then Kubernetes PV will pick the 5GB for their use

Persistent volume is Binding with Kubernetes cluster and host machine with a particular read write policy or storage. Persistent volume claim (PVC) has to claim that PV.

Node

As per diagram suppose 10GB storage present in host machine cluster persisted the 5GB from host and pods required 3 GB so claimed to PVC and PVC get that storage from PV. Now PV have 2GB available. 3 GB mounted/ Bounded for pod 1 &2.

PV

5GB

PVC

3GB

Pod 2

Pod1

Host Machine

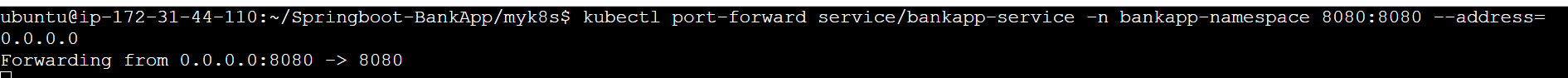
10GB

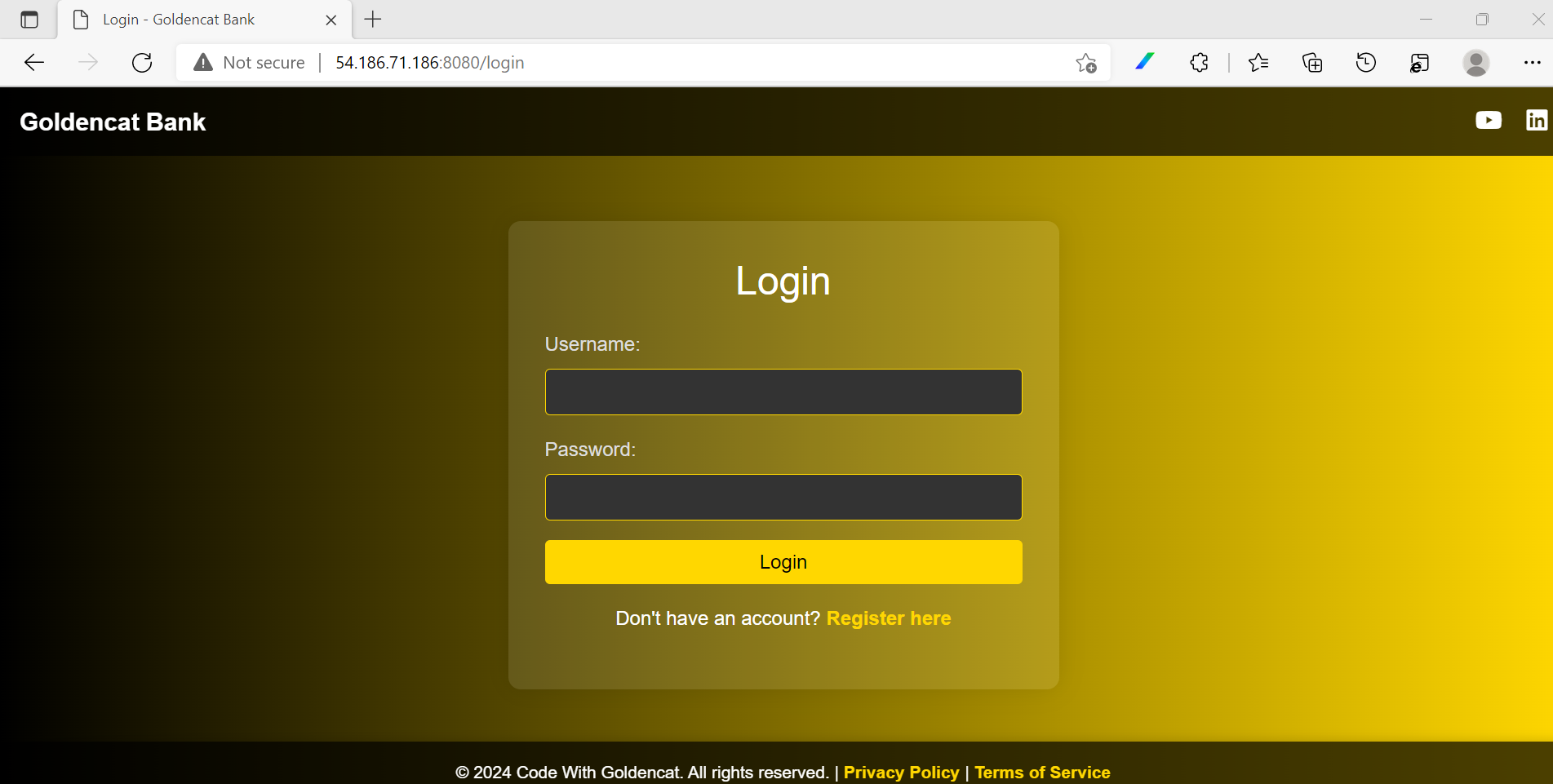
PVC request storage to PV.

**Spring boot bankapp**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

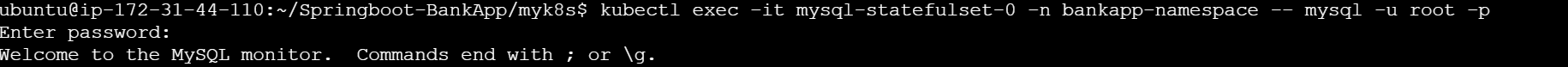
****

****

**A screenshot of a computer

AI-generated content may be incorrect.**

**Checking data in DB**



A black screen with white text

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A black and white rectangle with white text

AI-generated content may be incorrect.

Volume mount storage



A screenshot of a computer

AI-generated content may be incorrect.

Ingress

User account / Service account

**Horizontal pod auto scaling**

Creating new pod based on traffic. Which creating the HPA we must mention min Replicas and Max Replicas based on that pod is created when the more traffic comes. If traffic reduced it automatically reduces the pod means auto deleting pod. Metric pod required to add in kind.

**Vertical Pod Autoscaler**

VPA increasing the size of pod means adding the resources means memory, CPU to run the application. Here in yml file update Policy required which has update Mode Auto.

**Daemon Set**

Daemon set ensure that all nodes run a copy of pod. Suppose while creating cluster we have 2 worker nodes after applying daemon set, daemon set(ds) have 2 desired current means pod are running in both nodes. Not Required to maintain the replicas. As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected. Deleting a Daemon Set will clean up the Pods it created.

**Taints and Tolerance**

Taints means adding some condition to the noes so that they can accept the task or not creating pod. And Tolerance is the even the condition presents they accepting the schedule task and task is running on same node. Suppose we have 3 nodes one is control plane and 2 worker node and added the taints on worker node1 and node 2 also. If we want to create pod both worker nodes not accepting to create pod also control plane not accepting to create pod. That is taint.

**Tolerance** means if while creating pod we are adding some tolerations. Even if the taints are present on worker nodes that worker node that matches the tolerations that will accept this and create that pod.

**HELM**

Help create will create all required yml files like deployment, services and we don’t require to write any yml file. We just need to update **values.yml** file and install that.

**Helm** is a package manager for Kubernetes that simplifies the process of defining, installing, and upgrading applications on Kubernetes clusters.  It uses **Helm Charts**, which are packages of pre-configured Kubernetes resources, to manage these applications.

**RBAC(Role Base Access Control)-**

RBAC is a method of regulating access to computer or network resources based on the **roles** of **individual users** within an organization. In the context of Kubernetes, RBAC is a security feature that controls access to resources within your cluster. It allows you to specify what actions a user or a group of users can and cannot perform.