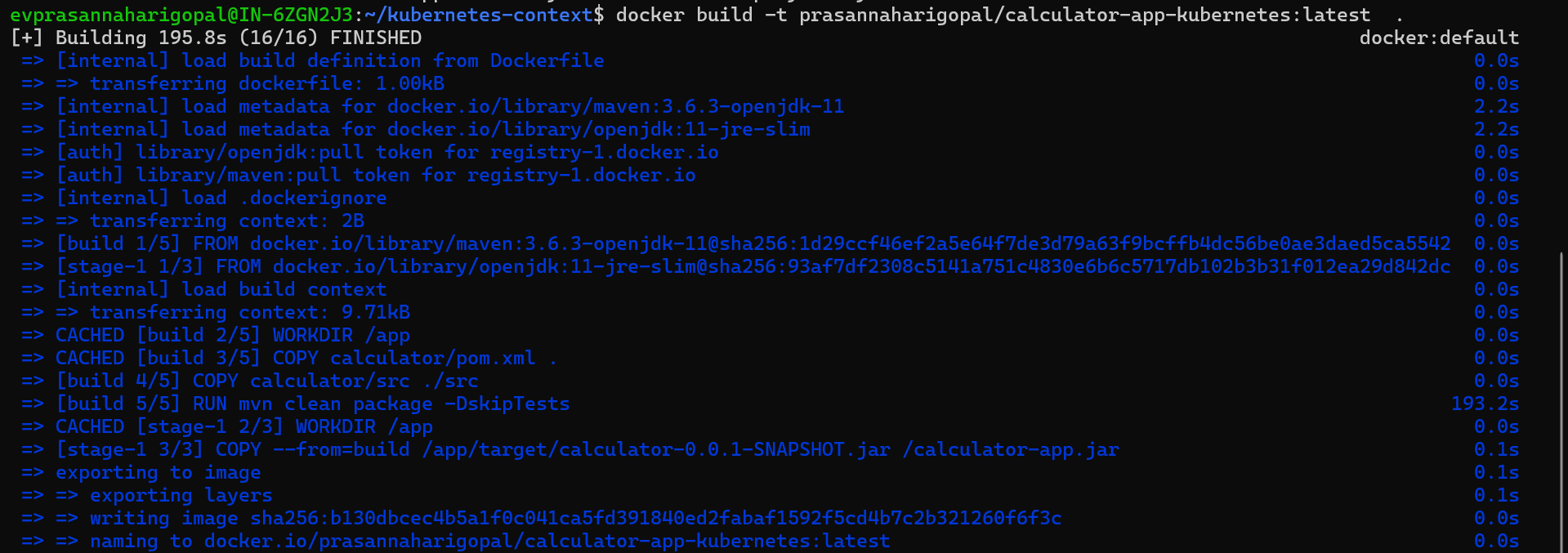
**Kubernetes Assignment Submission**

Here I have created a Folder called “kubernetes-context” and placed the source code and necessary files required for kubernetes.Here I have a create a sample Java Springboot Calculator application,and Created a Docker file and Build an Image from that Dockerfile and Pushed that image to Dockerhub  
Collect that image and build the Replicas and done deployments into Kubernetes

I created a Deployment.yaml and Service .yaml file

Build the Docker image with the following command   
Now pushed the Docker image into public repository -DockerHub

A screen shot of a computer

Description automatically generated

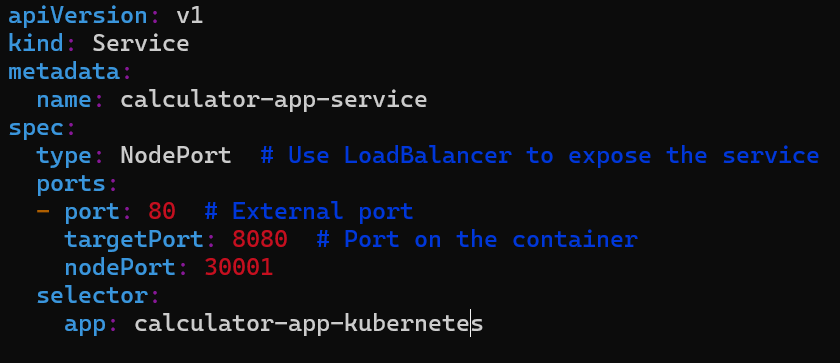
**This is my Deployment.yaml file**

Calculator-deployment.yml

A screen shot of a computer

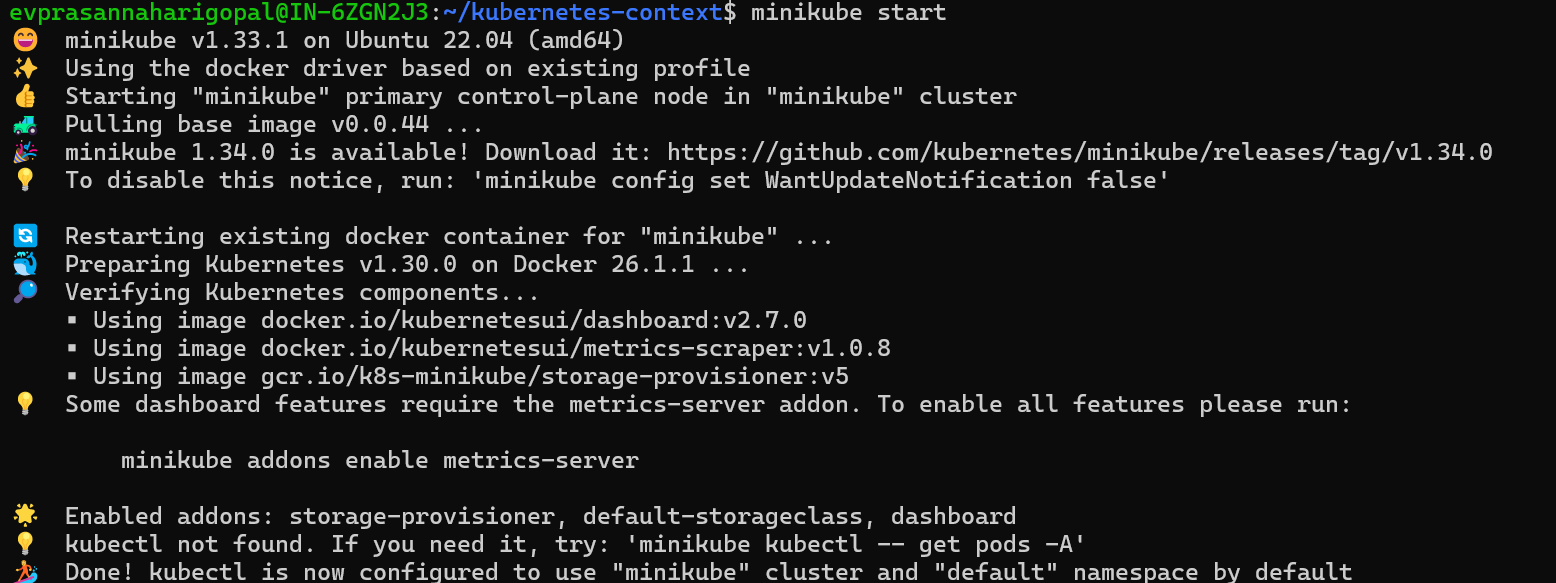
Description automatically generated

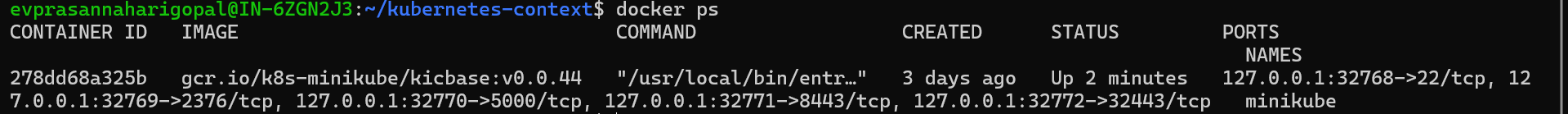
**This is my service.yml file**  
calculator-app-service.yml



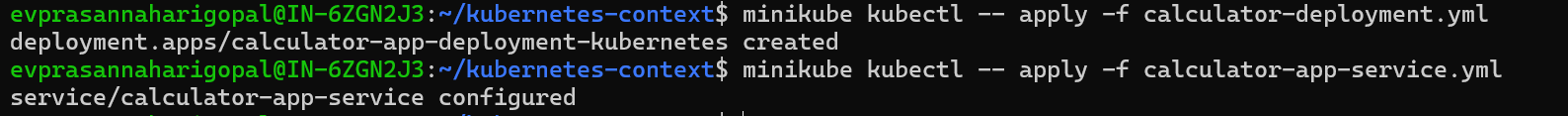
**Minikube Cluster**Minikube is a tool designed to run a single-node Kubernetes cluster on your local machine. It’s ideal for development and testing purposes as it allows you to experiment with Kubernetes without needing a full-scale cluster

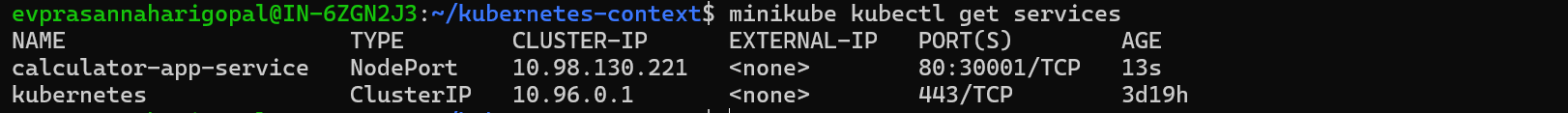
We can start the cluster by giving minikube start

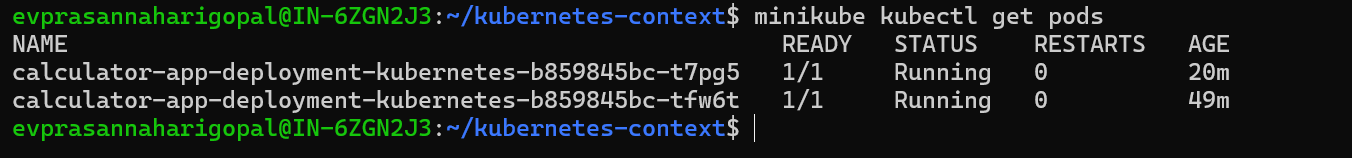
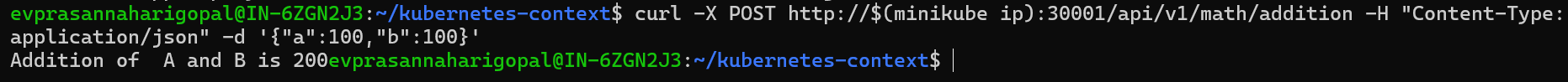
****

It will run a container and we can check them via docker ps command  


We can apply our configurations files with kubectl command



We can check all our running services by following command  


We can check our pods by the following command  
We can access our application with following curl request  


**Access the application externally**

**We can use NodePort or LoadBalancer**

**Use LoadBalancer Type to access it externally in service.yaml file**

**A screenshot of a computer program

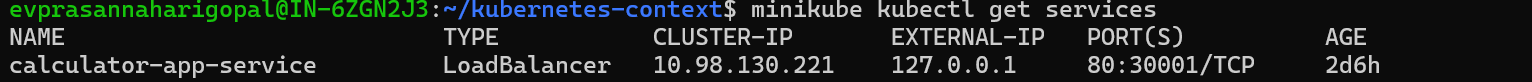
Description automatically generated**

**Minikube Tunnel**

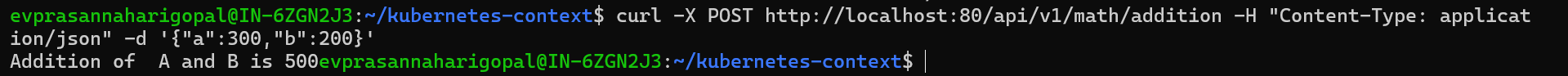
**A screen shot of a computer

Description automatically generated**

**Check the services which are running**

****

**Access it via external IP Address**

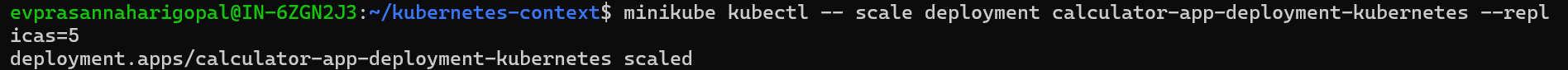
This is in Ubuntu machine ****

This is in Windows machine

**A screenshot of a computer

Description automatically generated**

**Scaling the replicas**

We can scale the application with scale flag and replicas flag and give the count there and the deployment name there   
we can check the pods by giving the following command   
A screen shot of a computer

Description automatically generated

**Bonus Part**

* Enhance the application deployment by adding a health check endpoint and configuring liveness and readiness probes in the Kubernetes deployment.

**Deployment file**

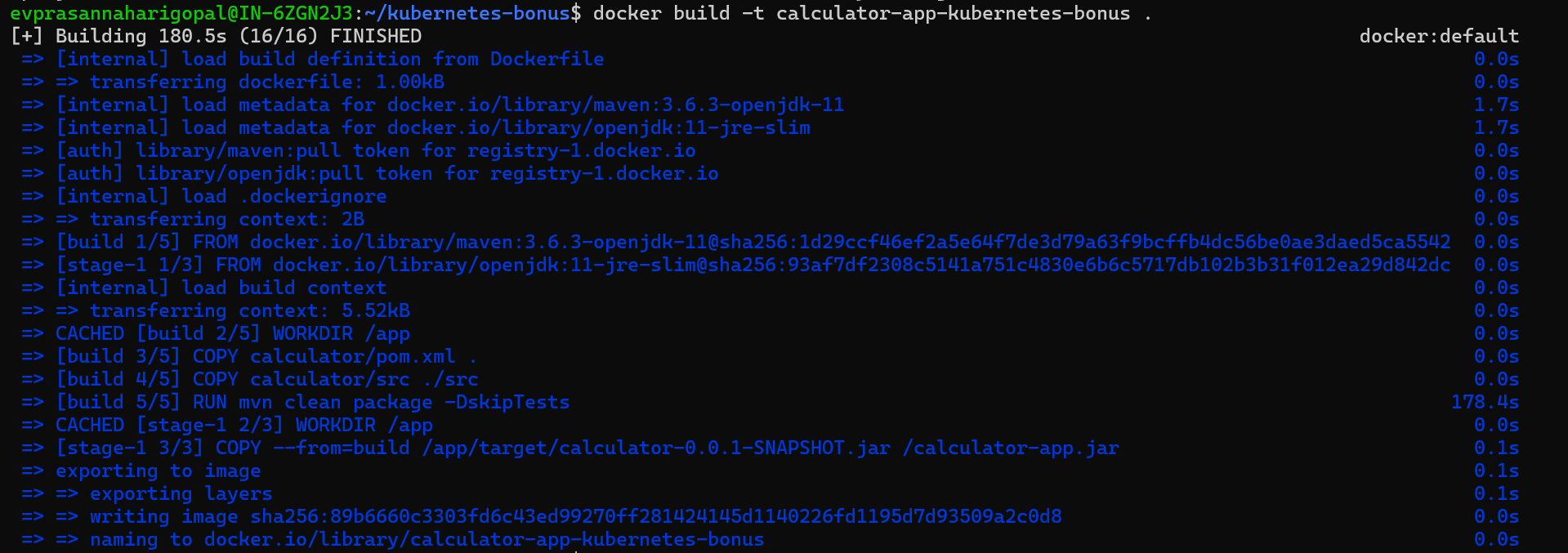
**A screen shot of a computer

Description automatically generated**

Service .yaml file

A screenshot of a computer program

Description automatically generated

Build the Docker Image and pushed into Dockerhub with Docker file  
  


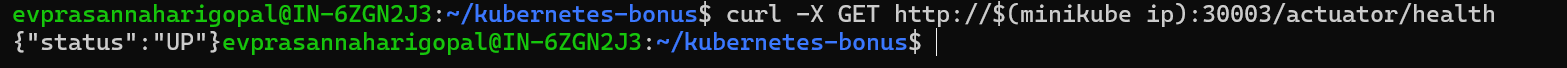
A screen shot of a computer

Description automatically generated

Check the status of the Pods

A screen shot of a computer code

Description automatically generated

Check the health check endpoints