Assignment 1 For Deployment of Microservices (FSE\_Oct\_2022\_C6\_CFSEZG561) Developing microservices with minikube

By,

Jeetendra Prasad

(2021CFSE032@wilp.bits-pilani.ac.in)

Contents

[Purpose: 3](#_Toc140954566)

[Notes/Legends: 3](#_Toc140954567)

[Assumptions: 3](#_Toc140954568)

[Steps :- 3](#_Toc140954569)

[Multi-service deployment and service discovery 15](#_Toc140954570)

[References:- 20](#_Toc140954571)

# Purpose:

We will go through basics of managing micro-services using Kubernetes using minikube, which allows us to run Kubernetes locally.

For demo we create threes basic microservices:-

* **gateway\_svc : Simple NodeJS backend application using express.js. Refer:** https://github.com/jeetendraprasad/MicroserviceAssignment/tree/master/MS2. **NOTE : We are using local copy of this project for this project for minikube/docker deployment**
* **auth\_svc : Simple NodeJS backend application using express.js. Refer:** https://github.com/jeetendraprasad/MicroserviceAssignment/tree/master/MS3. **NOTE : We are using local copy of this project for this project for minikube/docker deployment**
* **books\_svc : Simple NodeJS backend application using express.js. Refer:** https://github.com/jeetendraprasad/MicroserviceAssignment/tree/master/MS4. **NOTE : We are using local copy of this project for this project for minikube/docker deployment**

# Notes/Legends:

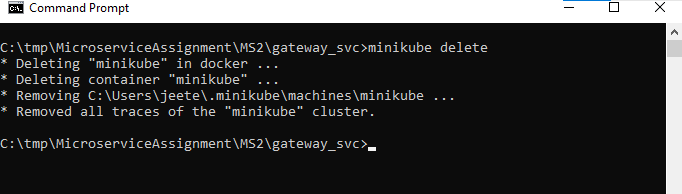
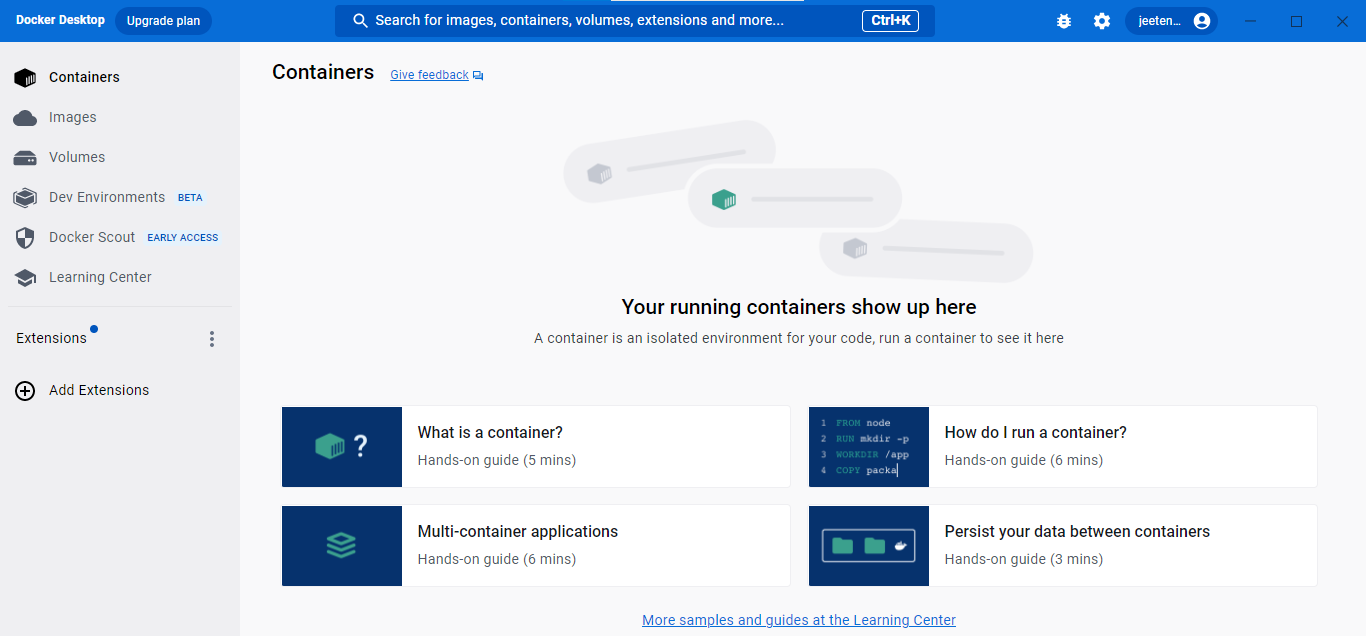
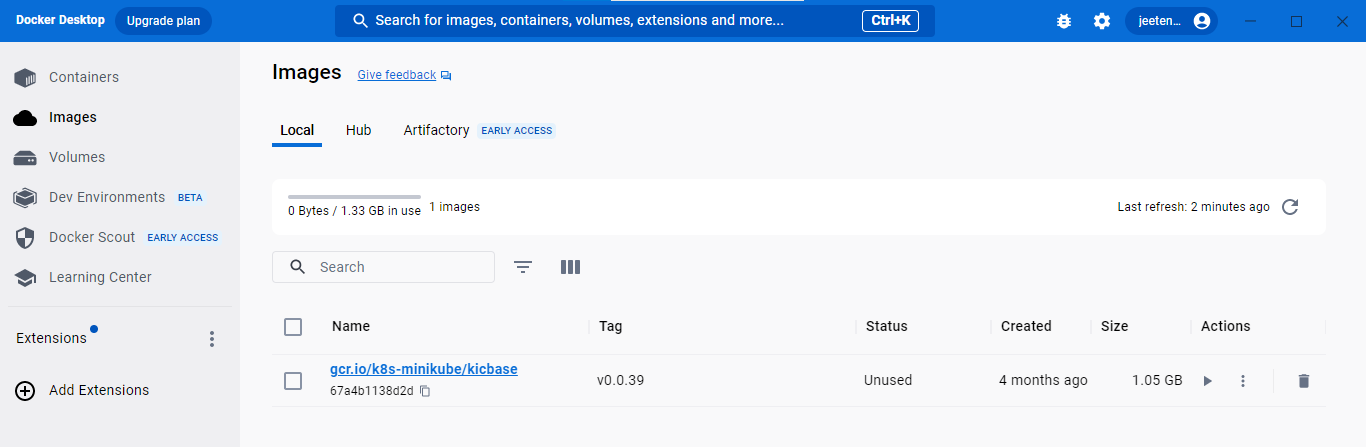
All the commands are in light blue colour.

# Assumptions:

We have already done following software installations:

* Docker desktop
* Minikube
* Kubectl

# Steps :-

1. We delete all existing containers of minikube by “**minikube delete**”  and also delete all images and containers in ‘Docker Desktop’ as  
2. Goto base folder (e.g., C:\tmp\)
3. Create a folder MicroserviceAssignment
4. Check necessary softwares: Open command Prompt at folder ‘C:\tmp\MicroserviceAssignment\’ and give following commands:-

C:\tmp\MicroserviceAssignment>**kubectl version --client**

WARNING: This version information is deprecated and will be replaced with the output from kubectl version --short. Use --output=yaml|json to get the full version.

Client Version: version.Info{Major:"1", Minor:"27", GitVersion:"v1.27.2", GitCommit:"7f6f68fdabc4df88cfea2dcf9a19b2b830f1e647", GitTreeState:"clean", BuildDate:"2023-05-17T14:20:07Z", GoVersion:"go1.20.4", Compiler:"gc", Platform:"windows/amd64"}

Kustomize Version: v5.0.1

C:\tmp\MicroserviceAssignment>**docker --version**

Docker version 24.0.2, build cb74dfc

C:\tmp\MicroserviceAssignment>**docker-compose --version**

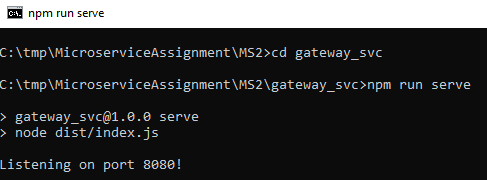
Docker Compose version v2.19.1

C:\tmp\MicroserviceAssignment>**minikube version**

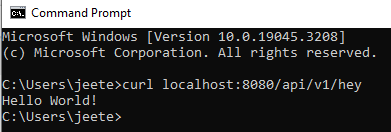
minikube version: v1.30.1

commit: 08896fd1dc362c097c925146c4a0d0dac715ace0

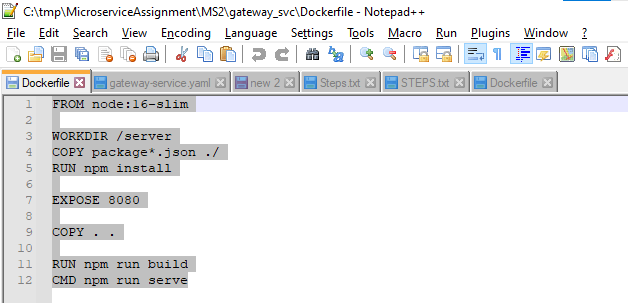
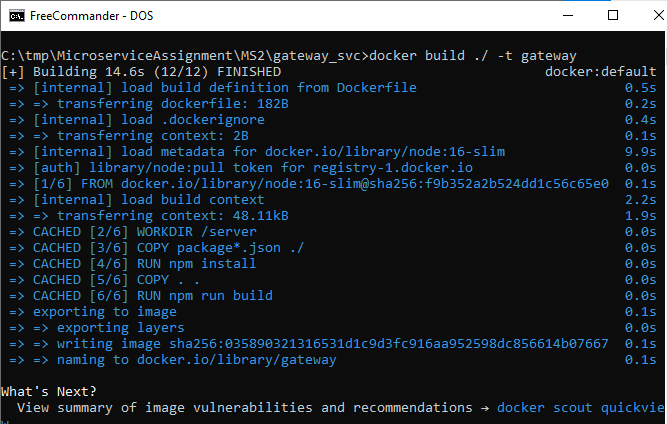
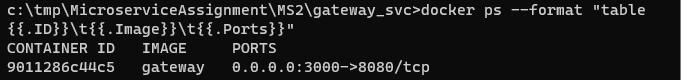
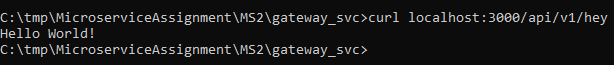
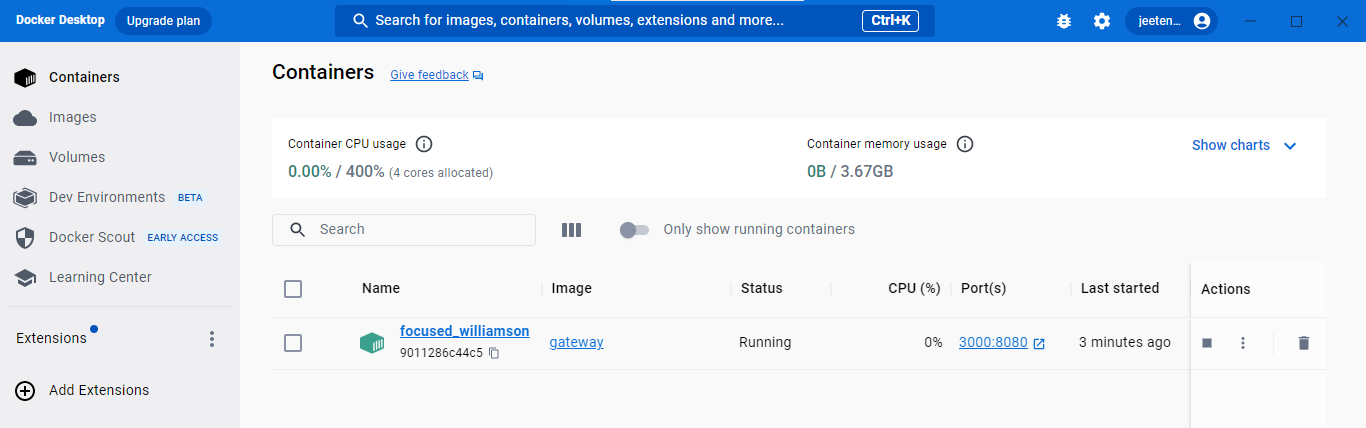
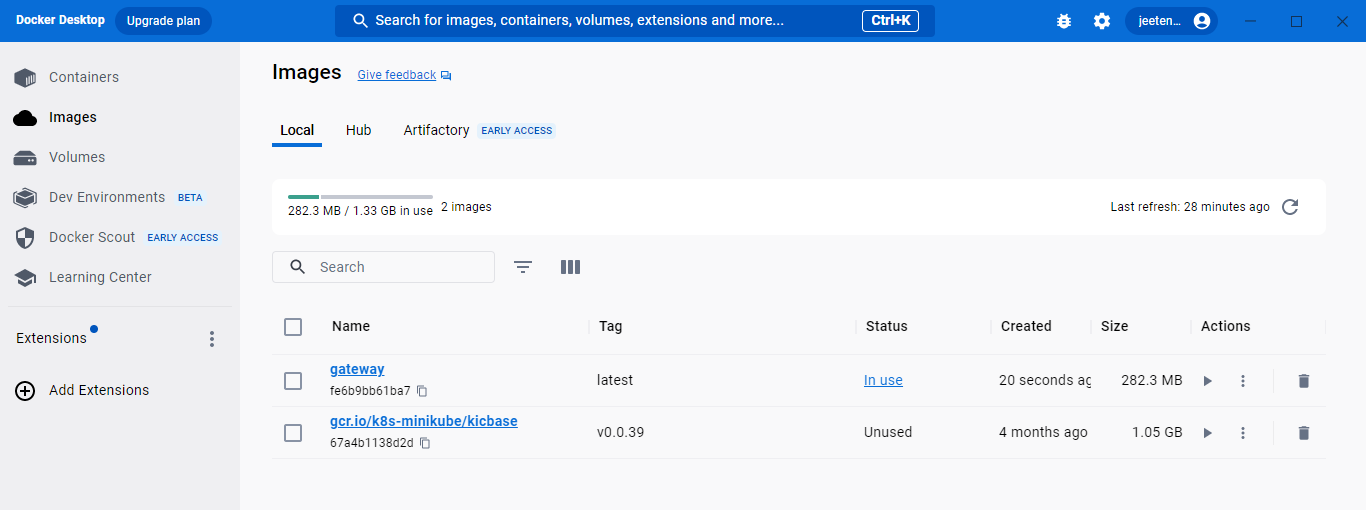
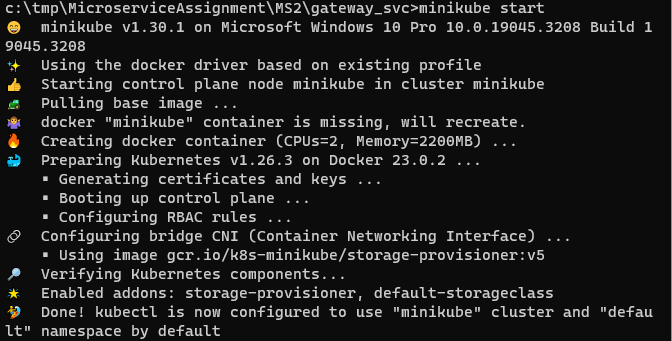
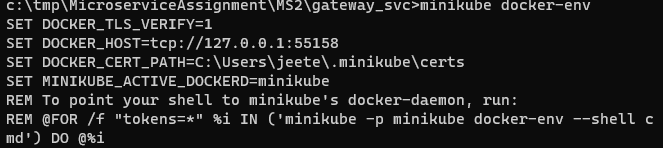
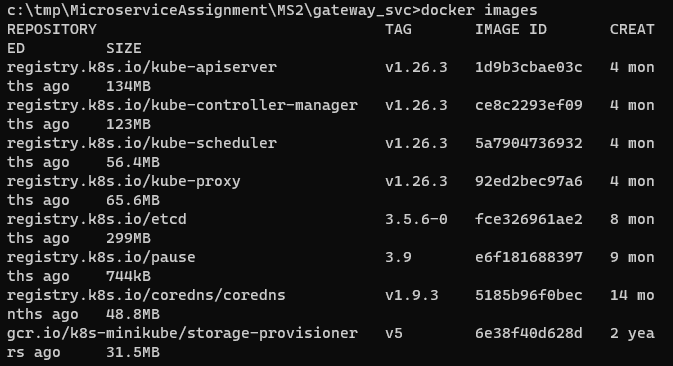
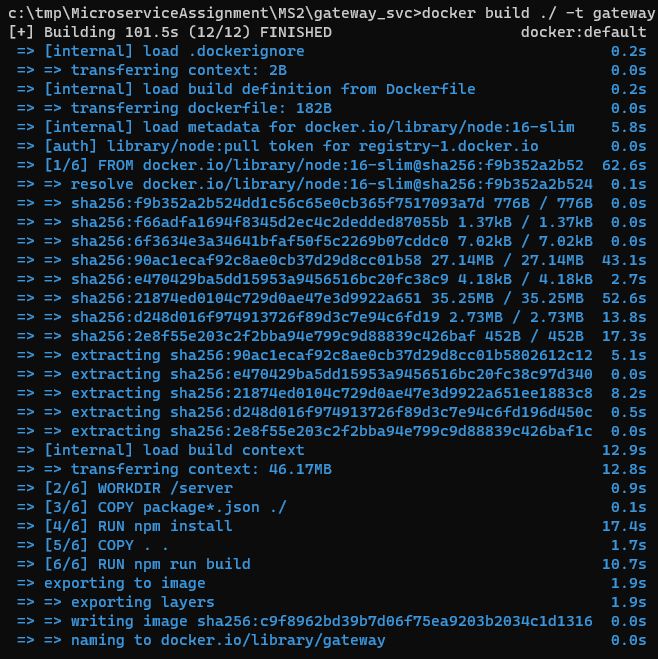
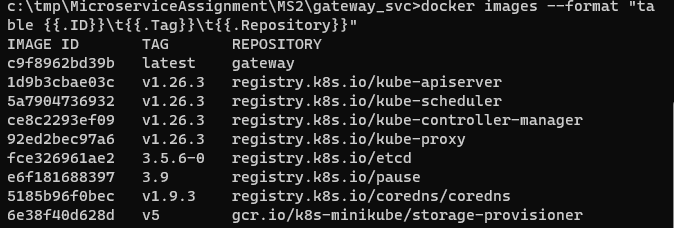
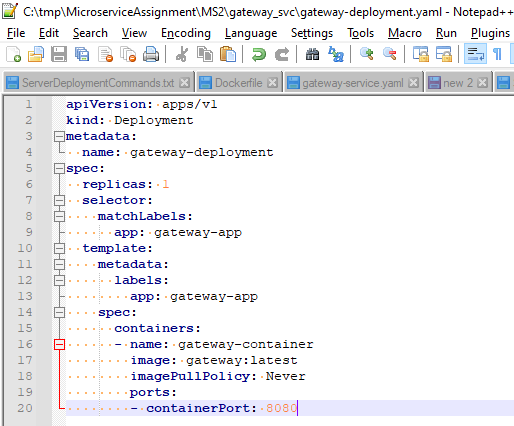
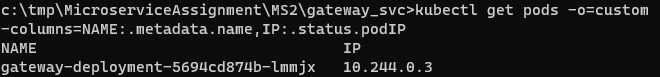
1. Create another folder "C:\tmp\MicroserviceAssignment\MS2" which contains microservice 2 code. The microservice is **auth\_svc** is responsible for user authentication. The code from GitHub (https://github.com/jeetendraprasad/MicroserviceAssignment/tree/master/MS2) can be kept here.
2. We test the project as: **npm run serve**



and that can be tested as curl localhost:8080/api/v1/hey

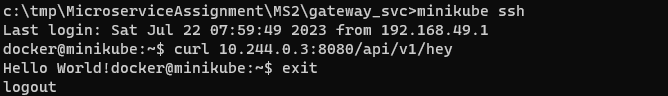


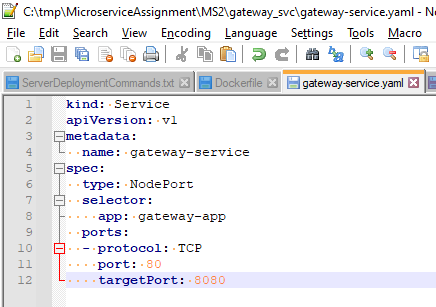
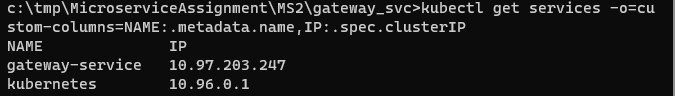
Post test we close “npm run serve” by Ctrl+C

1. Now we start deployment by creating a docker file. The name of file is simply ‘Dockerfile’. The content of docker file is 
2. Build the docker image using command “**docker build ./ -t gateway**” 
3. Run image: **docker run -d -it -p 3000:8080 gateway** 
4. Check “**docker ps --format "table {{.ID}}\t{{.Image}}\t{{.Ports}}"**” . NOTE: container id.
5. Command “**curl localhost:3000/api/v1/hey**” and we get response as “Hello world.” 
6. We delete docker image with “container id” found above. The command is “**docker stop 9011286c44c5**”
7. In docker we can see containers as and image as 
8. Now we start minikube as “**minikube start**” 
9. “**minikube docker-env**” 
10. **@FOR /f "tokens=\*" %i IN ('minikube -p minikube docker-env --shell cmd') DO @%i** 
11. **docker images**
12. **docker build ./ -t gateway**
13. **docker images --format "table {{.ID}}\t{{.Tag}}\t{{.Repository}}"**  NOTE: Some images are controlled by minikube that can be ignored.
14. To deploy this image using **kubectl**, we’ll use the deployment file. To do this we create file **gateway-deployment.yaml with content:- **
15. **kubectl create -f gateway-deployment.yaml** 
16. Check ‘**kubectl get pods**’
17. Get ip using command ‘**kubectl get pods -o=custom-columns=NAME:.metadata.name,IP:.status.podIP**’ . NOTE: IP as this will be used later.
18. **minikube ssh**

**curl 10.244.0.3:8080/api/v1/hey**

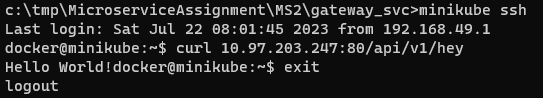
**exit**

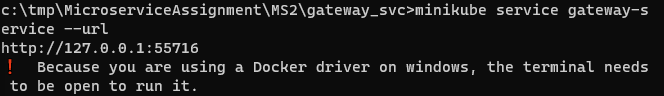
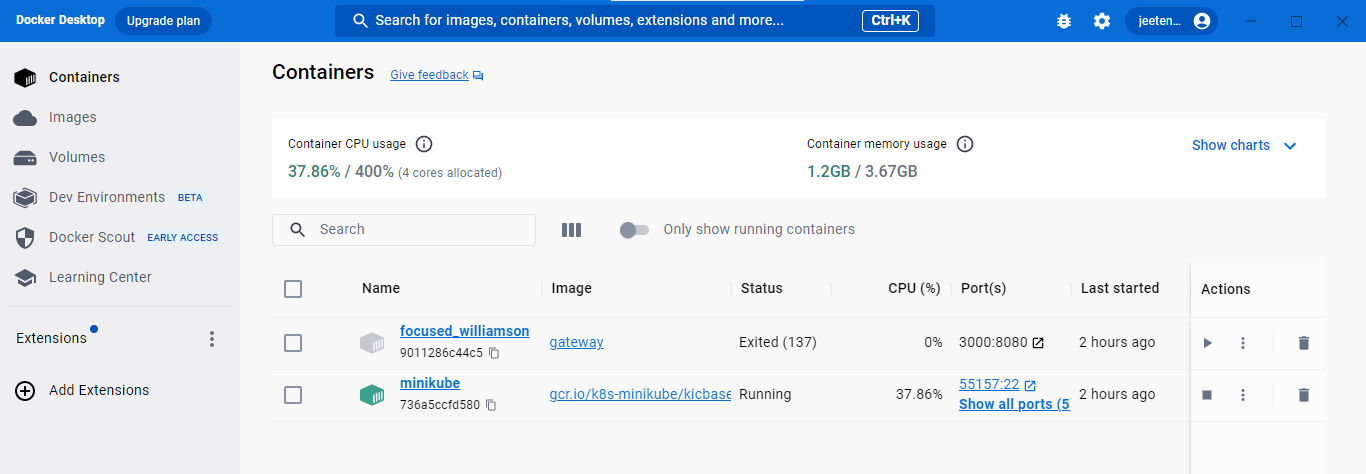
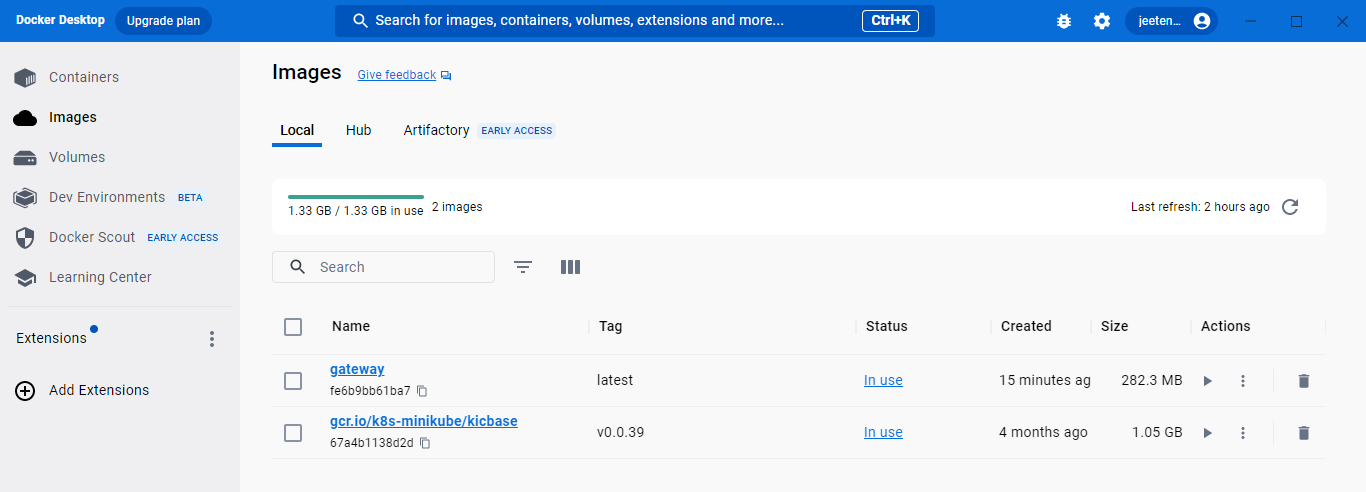
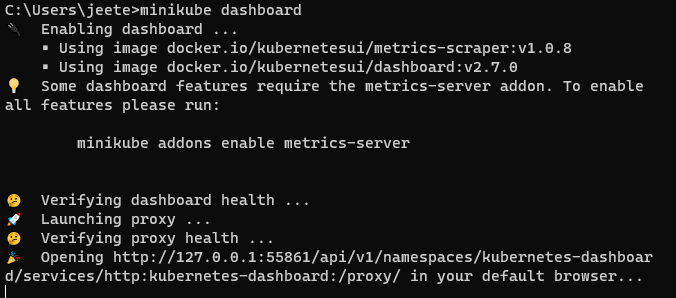
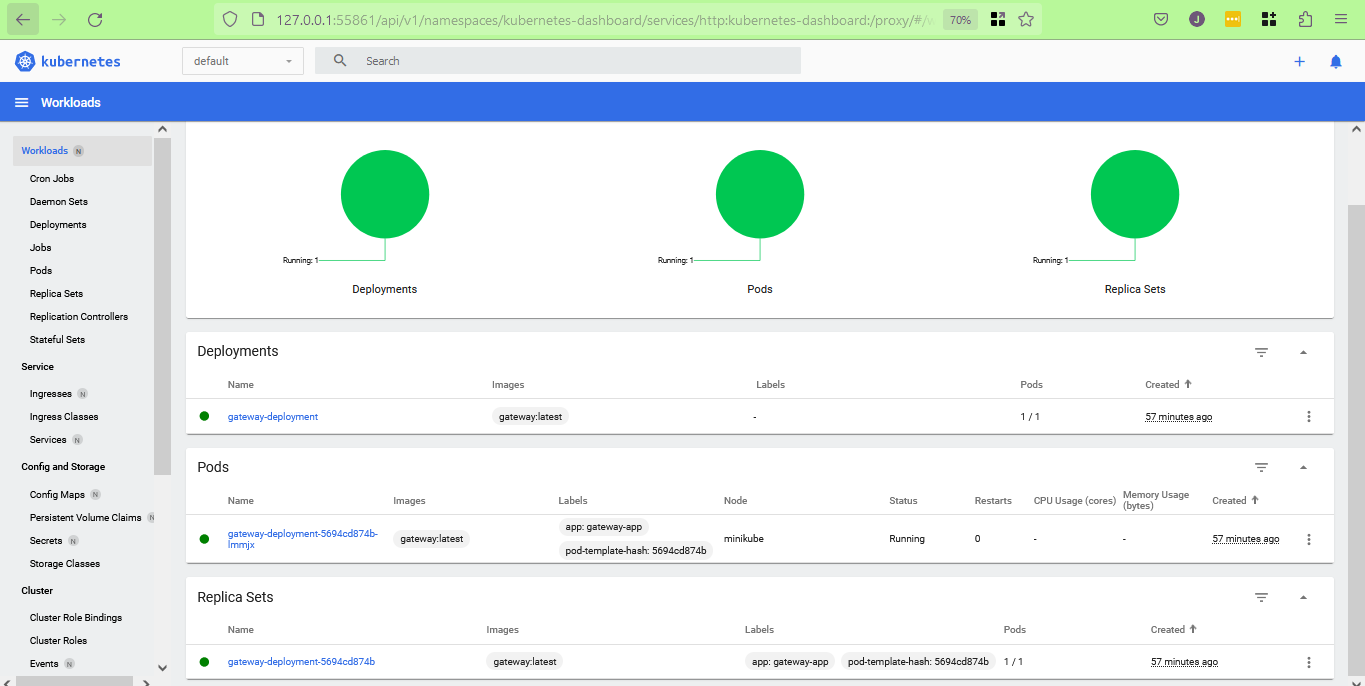


1. Now we create file ‘**gateway-service.yaml**’ with content 
2. “**kubectl apply -f gateway-service.yaml**” 
3. **kubectl get services -o=custom-columns=NAME:.metadata.name,IP:.spec.clusterIP**  NOTE : IP this will be used later
4. **minikube ssh**

**curl 10.97.203.247:80/api/v1/hey**

**exit**



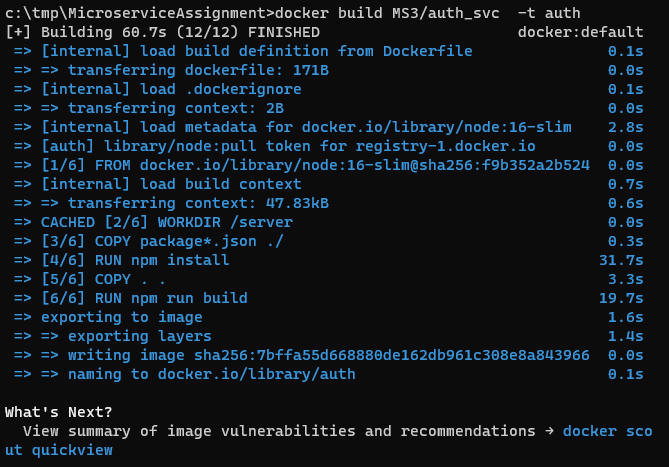
1. Check minicube status as “**minikube status**”. 
2. Get gateway url as “**minikube service gateway-service --url**” . NOTE : The url which needs to be used later
3. **curl http://127.0.0.1:55716/api/v1/hey **
4. In browser also we can access this url ****
5. Post the steps our docker desktop looks as: containers:-images:- 
6. And we can see minikube using command “**minikube dashboard**” which opens dashboard in browser as  
7. **This completes our 1st microservice**.

# Multi-service deployment and service discovery

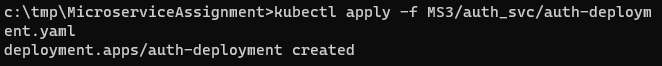
We have two more services **auth\_svc and books\_svc which** will add to minikube.

We will use following commands:-

**docker build MS3/auth\_svc -t auth**



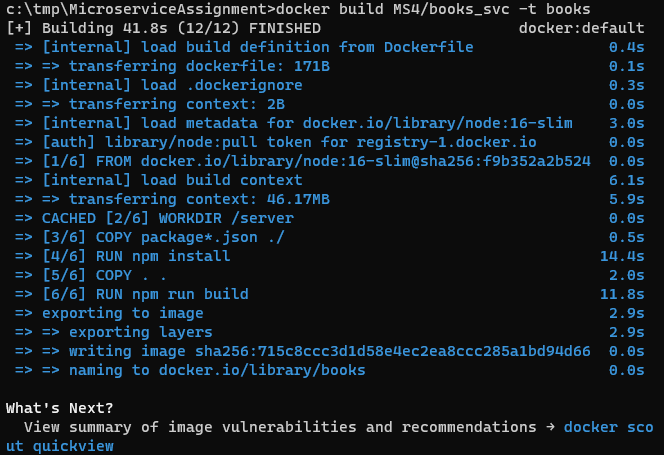
**kubectl apply -f MS3/auth\_svc/auth-deployment.yaml**



**kubectl apply -f MS3/auth\_svc/auth-service.yaml**



**docker build MS4/books\_svc -t books**



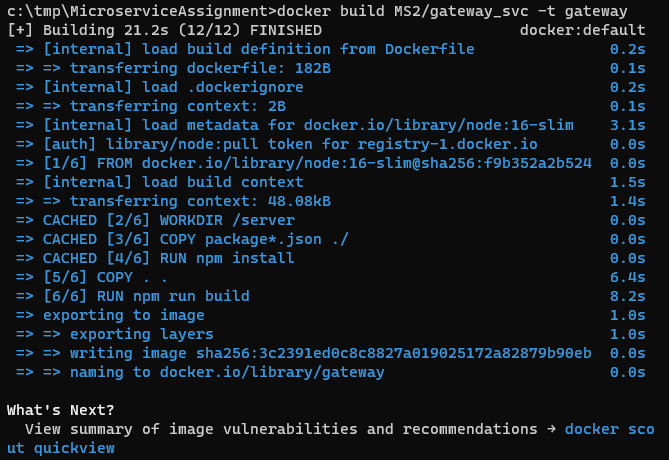
**kubectl apply -f MS4/books\_svc/books-deployment.yaml**



**kubectl apply -f MS4/books\_svc/books-service.yaml**



**docker build MS2/gateway\_svc -t gateway**

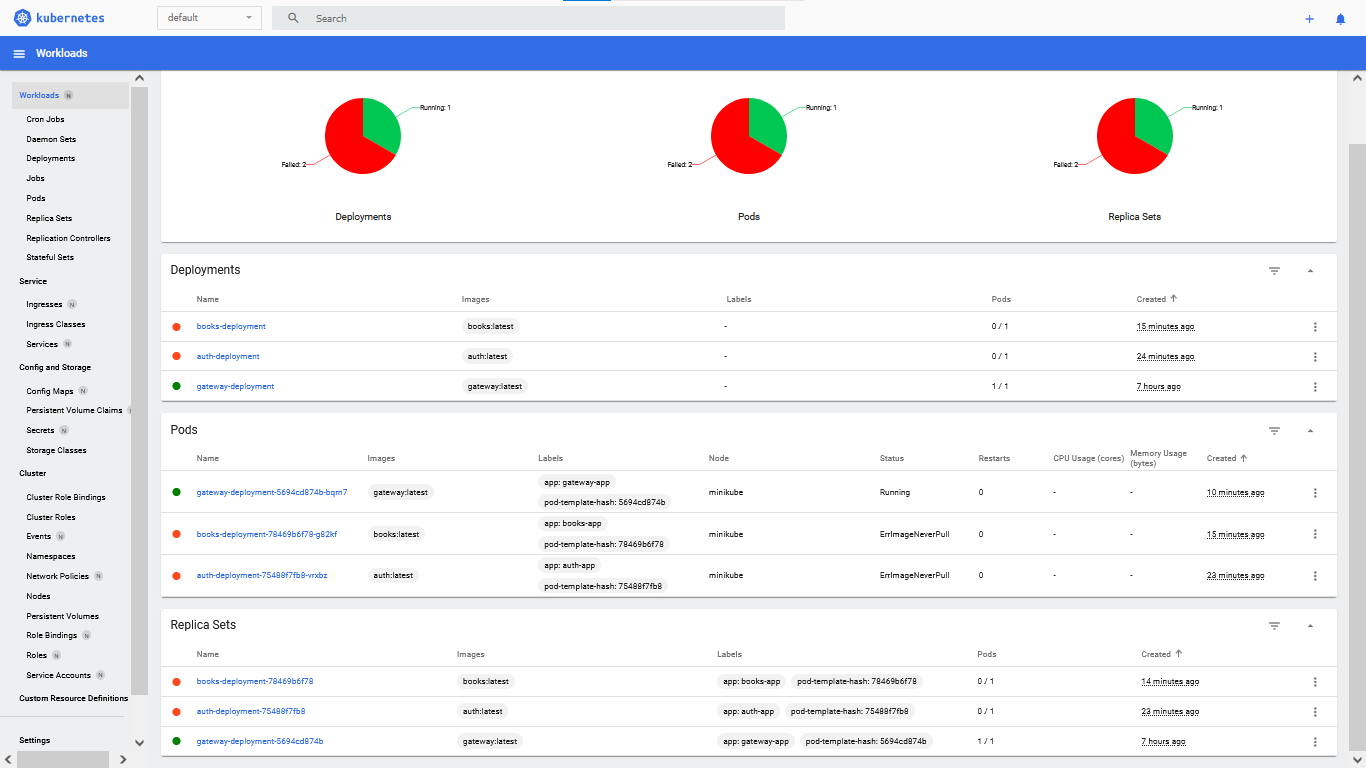
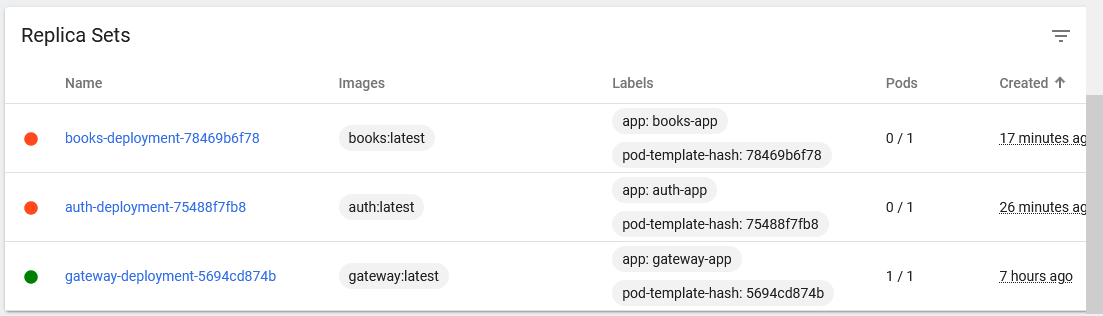
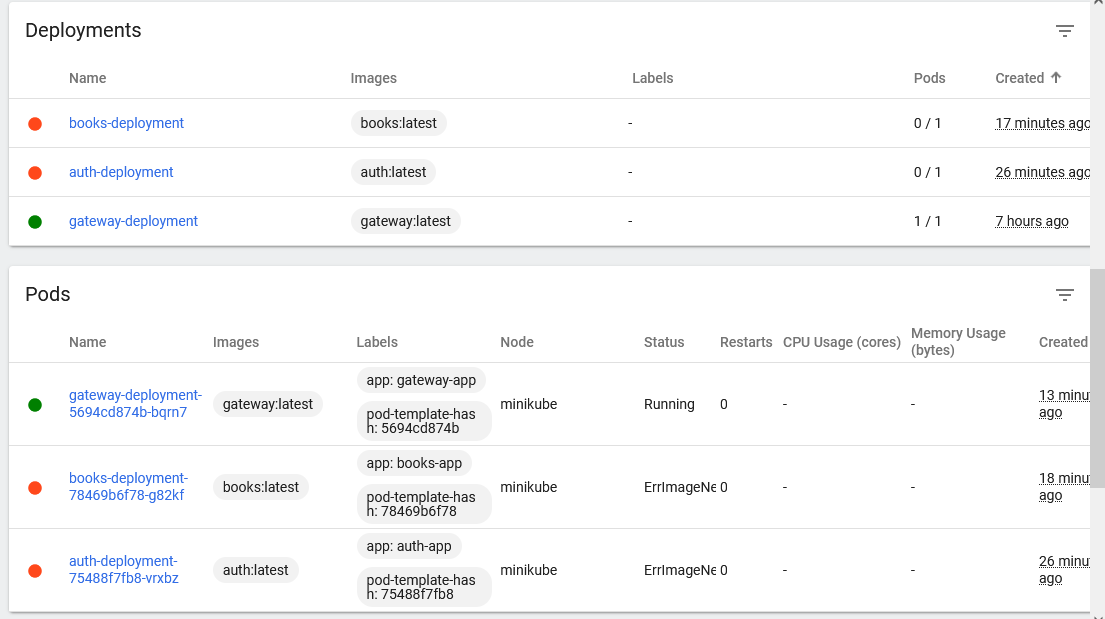
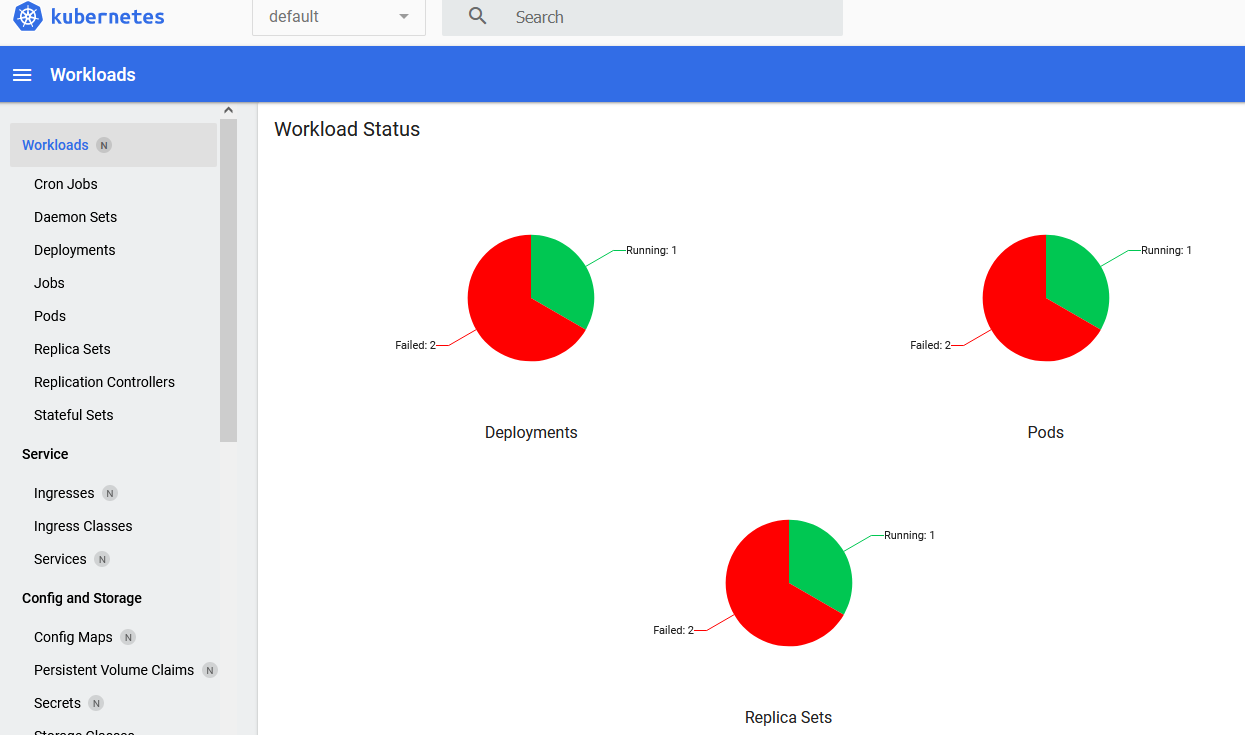


**kubectl delete pods -l app=gateway-app**



Notice that for the **gateway\_svc**, we don’t need to need to recreate the deployment and service. We simply delete the pod, which will then result in a new pod being created with the new docker image.

Final output via minikube dashboard “**minikube dashboard**”

 which when zoom looks as 

We can see various pods, replica sets, deployments. Right now, I need to resolve failed deployments.

**This is how we can use minikube for local**.

# References:-

1. Bits Pilani course,
2. Bits Pilani weekly online classes in MS Teams and its recordings.
3. https://yuchen52.medium.com/developing-microservices-with-minikube-81b31e5366ac and https://github.com/yzhong52/microservices-demo