**Docker& Kubernets Assignment**

* For the REST API, I created an endpoint using the Spring Boot framework, added the necessary dependencies, and configured the controller class to handle the API endpoints.
* Regarding Docker, I created a Docker file for the REST API, and after setting up a Docker Hub account, I built and pushed the Docker image to Docker Hub. The image has been made public and can be accessed through this link: [Insert Docker Hub image path here].
* For Kubernetes, I set up a local Kubernetes cluster using Minikube and created YAML files for the deployment and service of the REST API. The deployment YAML includes two replicas of the Docker image.

**Deliverables**

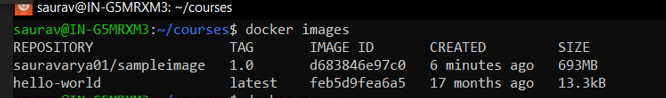
**1**.**Git Hub public repository:**

[**https://github.com/sauravarya01/courses/blob/master/Dockerfile**](https://github.com/sauravarya01/courses/blob/master/Dockerfile)

**2**.**Docker File:**

From openjdk:11  
 **WORKDIR /myapp  
 COPY target/courses-0.0.1-SNAPSHOT.jar /myapp**  
 **CMD [ "java","-jar","courses-0.0.1-SNAPSHOT.jar" ]**  
 **EXPOSE 9001**

**3.Docker Image**



**4.Image path from Docker Hub Repo:**

[**https://hub.docker.com/r/sauravarya01/sampleimage**](https://hub.docker.com/r/sauravarya01/sampleimage)

**5 Yaml File**

Deployment.yaml

apiVersion : apps/v1

kind: Deployment

metadata:

name: docker-assignment-deployment

labels:

app: docker-assignment

spec:

selector:

matchLabels:

app: docker-assignment

replicas: 2

template:

metadata:

labels:

app: docker-assignment

spec:

containers:

- name: docker-assignment

image: sauravarya01/sampleimage:1.0

ports:

- containerPort: 9001

Service.yaml

apiVersion : v1

kind: Service

metadata:

  name: docker-assignment-service

spec:

  type: NodePort

  selector:

    app: docker-assignment

  ports:

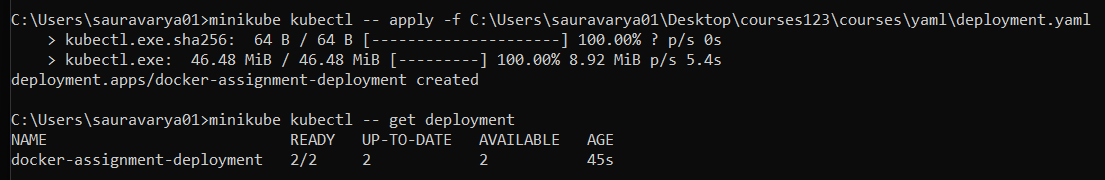
  - port: 9001

    targetPort: 9001

    nodePort: 30002

**Screeshots:**

6. One for Deployment Created Successfully and Second for Total Number of Deployments.



7. One for Screenshot of Service Created Successfully and Second for Total Number of Running services.

**A screenshot of a computer

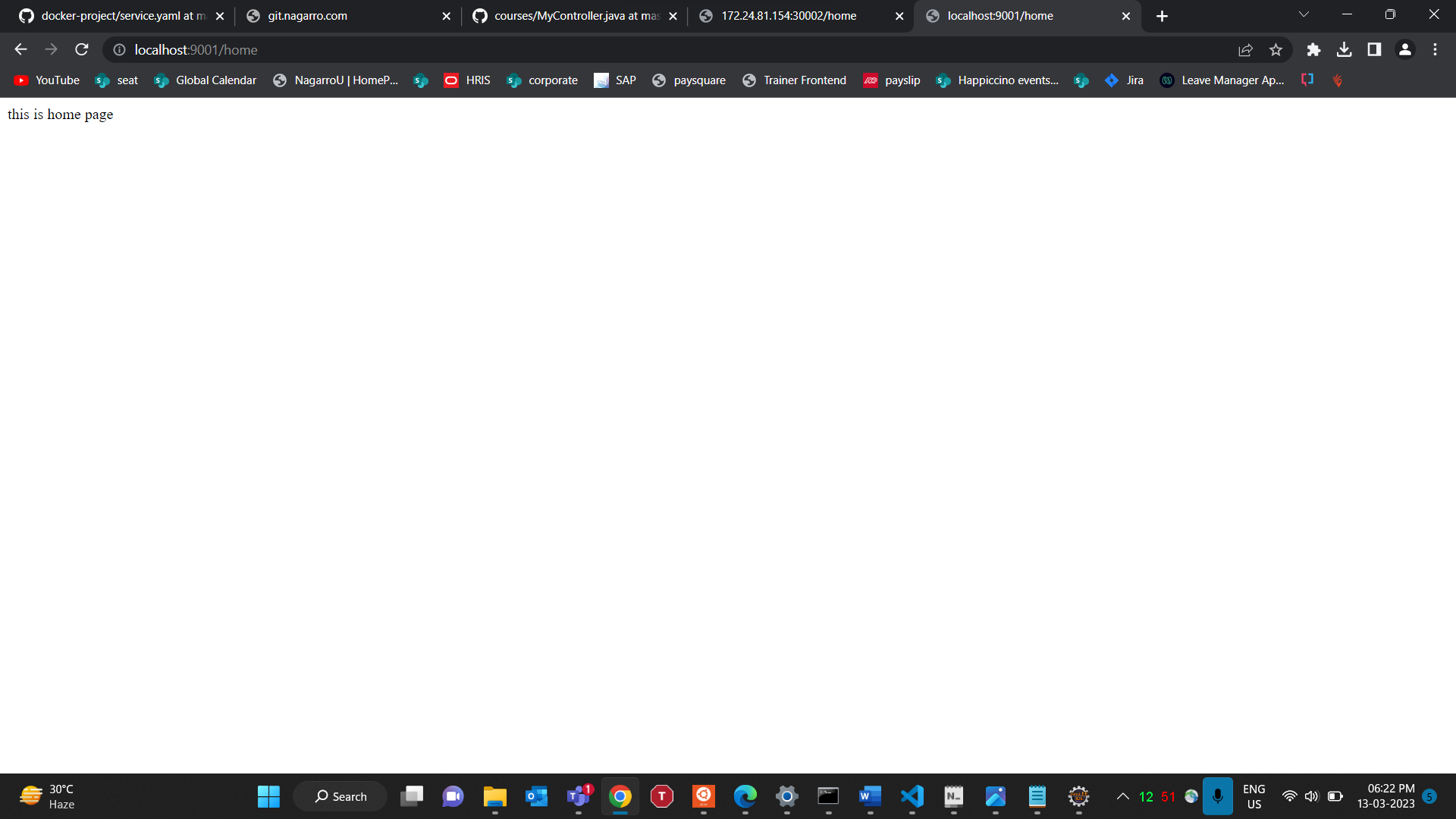
Description automatically generated with medium confidence**

8. Screenshot for Number of Running Pods.

**Text

Description automatically generated**

9. Screenshot of Running Application. (Show Endpoint is accessible successfully through browser)

****

10. Screenshot - Access pod through the service endpoint only

**A screenshot of a computer

Description automatically generated**