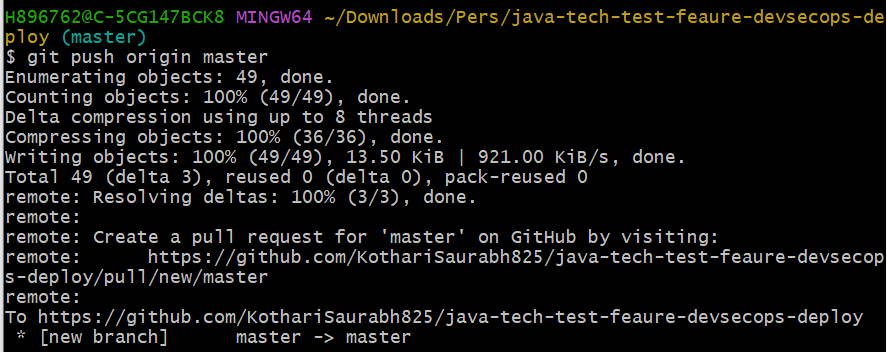
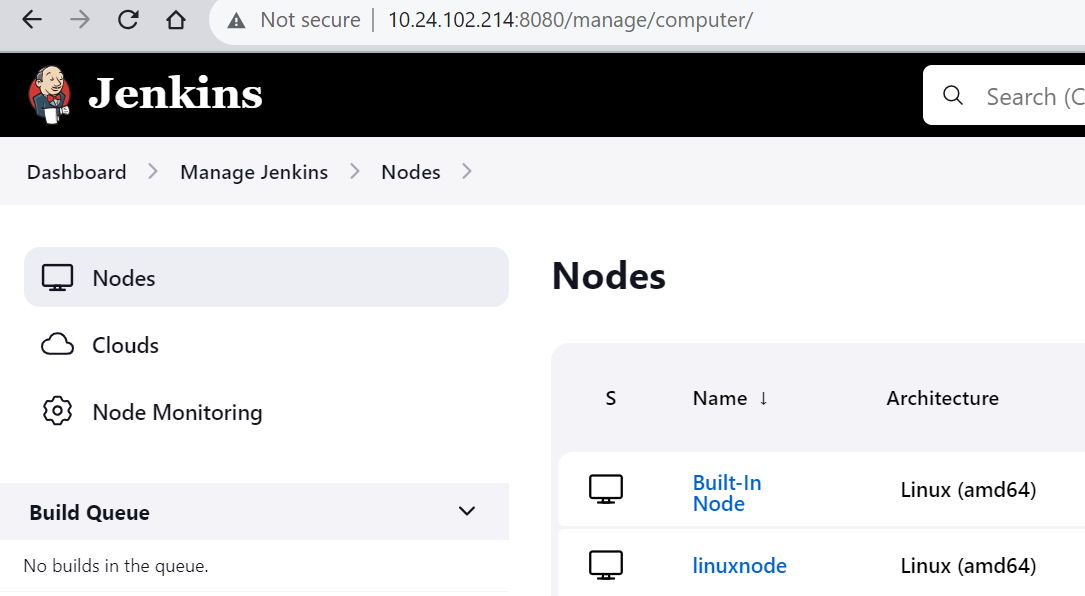
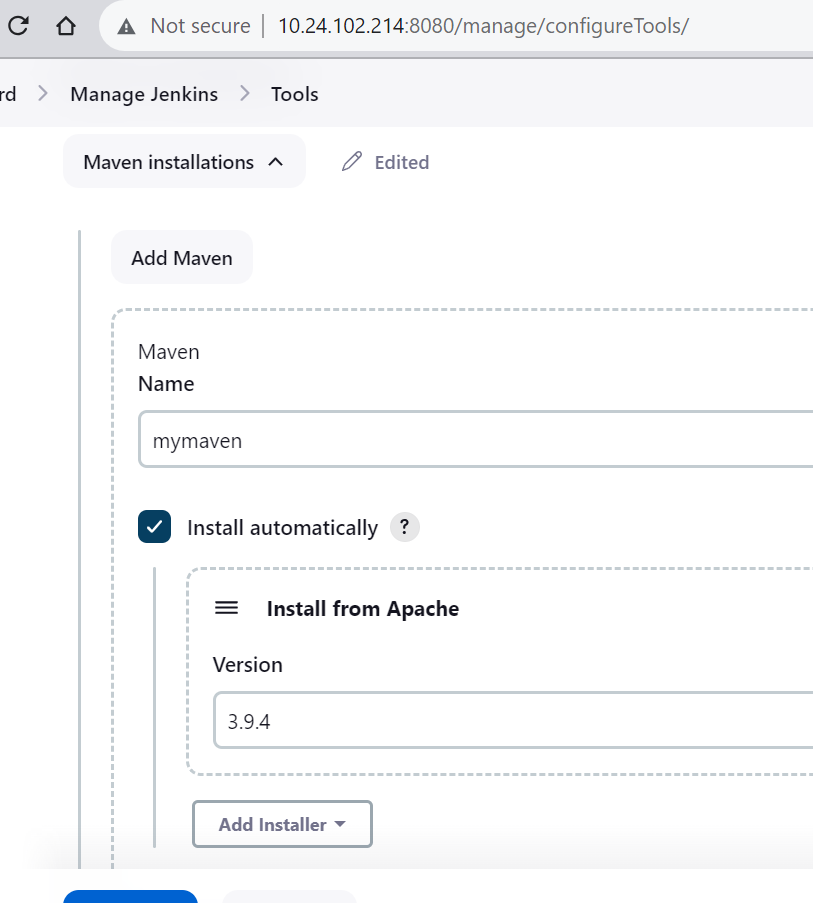
**Java Tech Test Feature devsecops deploy project:**

1. **Downloaded the source code shared by Sakshi Malik over gmail to my local system.**
2. **Uploaded the code to my github account using Git Bash(installed on my local system).**



1. **I already have 2 linux servers (built during devops training) where I have Java and Jenkins Installed (Jenkins Master & Client(‘linuxnode’) ). On Jenkins I have Maven plug-in Enabled which helps to compile, code review(pmd), unit-test, package and install java application.**



1. **Created a pipeline with below groovy code:**

\*\*\*\*CODE START \*\*\*\*

pipeline{

tools{

maven 'mymaven'

}

agent {

label 'linuxnode'

}

environment {

DOCKERHUB\_CREDENTIALS=credentials('dockerloginid')

}

stages{

stage('Clone a Repo'){

steps{

git 'https://github.com/KothariSaurabh825/java-tech-test-feaure-devsecops-deploy.git'

}

}

stage('Compile the code'){

steps{

sh 'mvn compile'

}

}

stage('CodeReview'){

steps{

sh 'mvn pmd:pmd'

}

}

stage('Unit Test'){

steps{

sh 'mvn test'

}

}

stage('Package'){

steps{

sh 'mvn clean install package'

}

}

stage('Copy jar file to /tmp/jenkinsdir'){

steps{

sh 'mv /tmp/jenkinsdir/workspace/java-tech-test/target/tdd-supermarket-1.0.0-SNAPSHOT.jar /tmp/jenkinsdir/workspace/java-tech-test/tddsupermarket.jar'

sh 'chmod -R 777 /tmp/jenkinsdir/workspace/java-tech-test/tddsupermarket.jar'

}

}

stage('Build the image'){

steps{

sh 'docker version'

sh "docker build -t kotharisaurabh/tddsupermarket ."

// sh "docker build -t kotharisaurabh/java-tech-test:latest ."

// sh "docker build -t kotharisaurabh/java-tech-test:${BUILD\_NUMBER} ."

sh 'docker image list'

// sh "docker tag kotharisaurabh/java-tech-test:${BUILD\_NUMBER} kotharisaurabh/java-tech-test:latest"

}

}

stage('Login2DockerHub') {

steps {

sh 'echo $DOCKERHUB\_CREDENTIALS\_PSW | docker login -u $DOCKERHUB\_CREDENTIALS\_USR --password-stdin'

}

}

stage('Publish\_to\_Docker\_Registry') {

steps {

sh "docker push kotharisaurabh/tddsupermarket"

//sh "docker push kotharisaurabh/java-tech-test:latest"

}

}

stage('Run Docker Image') {

steps {

sh "docker run -d -p 8080:8080 kotharisaurabh/tddsupermarket"

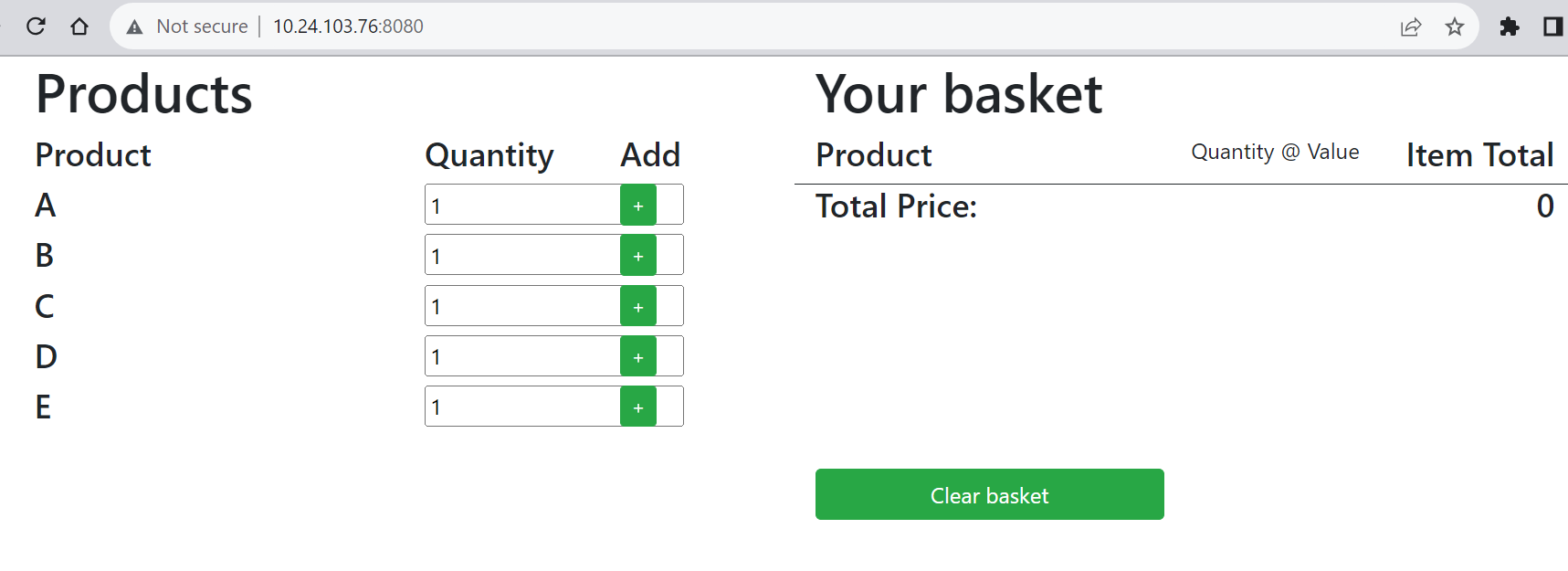
}

}

}

} \*\*\*\*CODE END \*\*\*\*

1. **Now browse** [**http://10.24.103.76:8080/**](http://10.24.103.76:8080/) **to see the application console.**



**I have also deployed this application on my kubernetes cluster(open source) which I configured in a server on my personal account.**

**# 1. Deployment YAML file**

**# tddsupermarket-deploy.yaml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: tddsupermarket-deploy**

**labels:**

**app: tddsupermarket**

**spec:**

**replicas: 3**

**template:**

**metadata:**

**labels:**

**app: tddsupermarket**

**spec:**

**containers:**

**- name: tddsupermarket-container**

**image: kotharisaurabh/tddsupermarket:latest**

**ports:**

**- containerPort: 8080**

**selector:**

**matchLabels:**

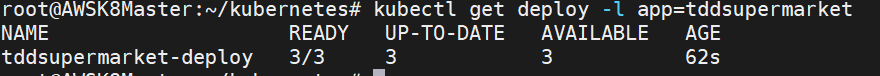
**app: tddsupermarket**

**# 2. Create and Display Deployment**

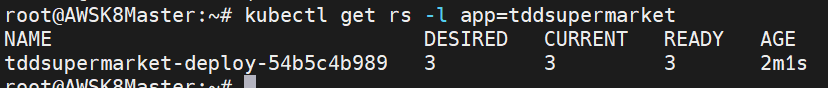
**kubectl create -f tddsupermarket-deploy.yaml**



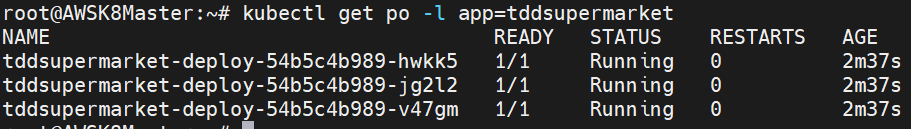
**kubectl get deploy -l app=tddsupermarket**



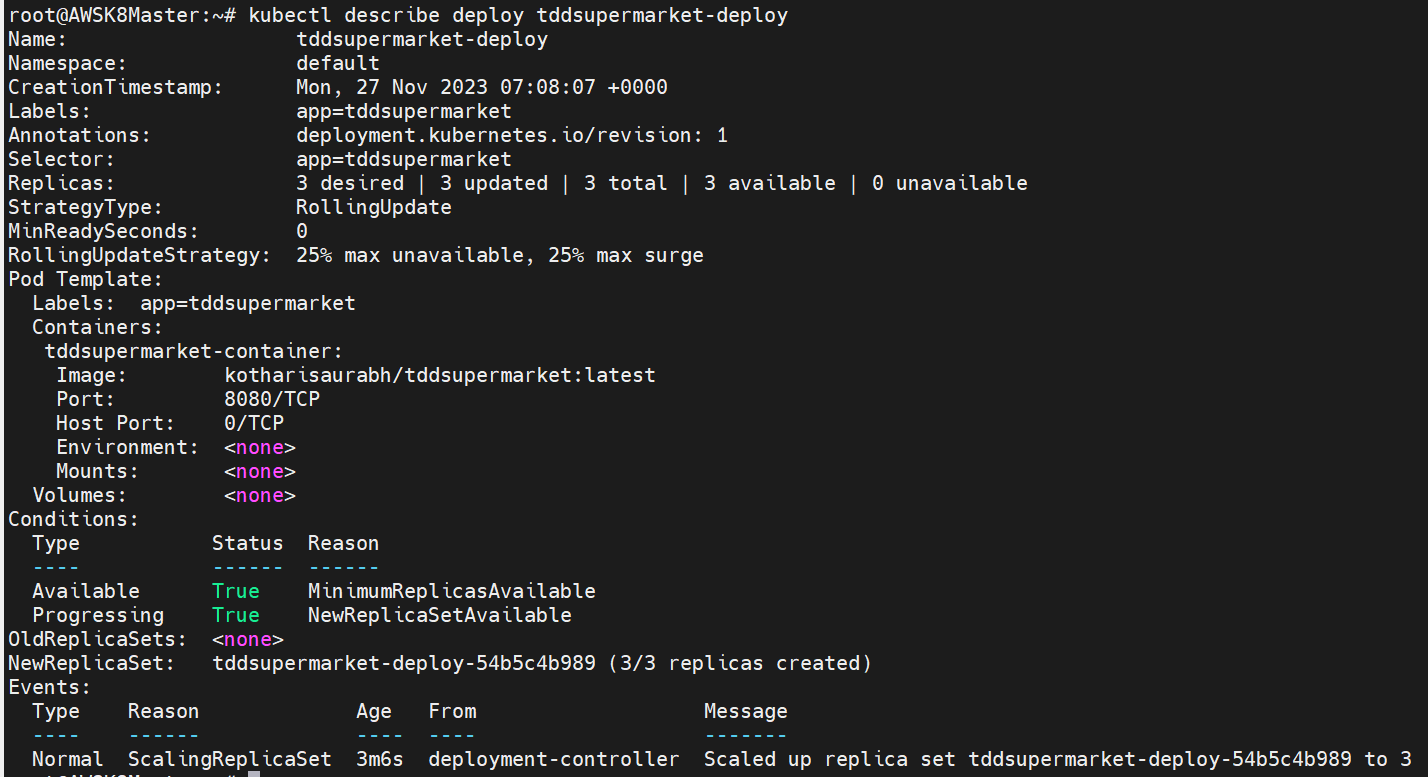
**kubectl get rs -l app=tddsupermarket**



**kubectl get po -l app=tddsupermarket**



**kubectl describe deploy tddsupermarket-deploy**



**All 3 pods are running now. To make it accessible over internet, I have created a Nodeport service.**

**# 3. Create and NodePort Deployment**

**# tddsupermarket-svc-np.yaml**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: tddsupermarket-service**

**labels:**

**app: tddsupermarket**

**spec:**

**selector:**

**app: tddsupermarket**

**type: NodePort**

**ports:**

**- nodePort: 31112**

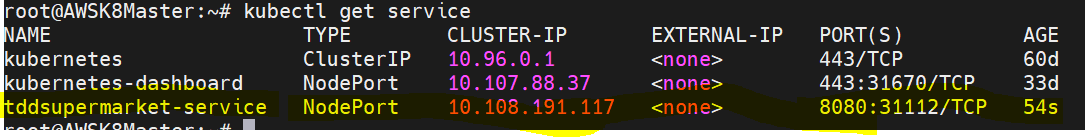
**port: 8080**

**targetPort: 8080**

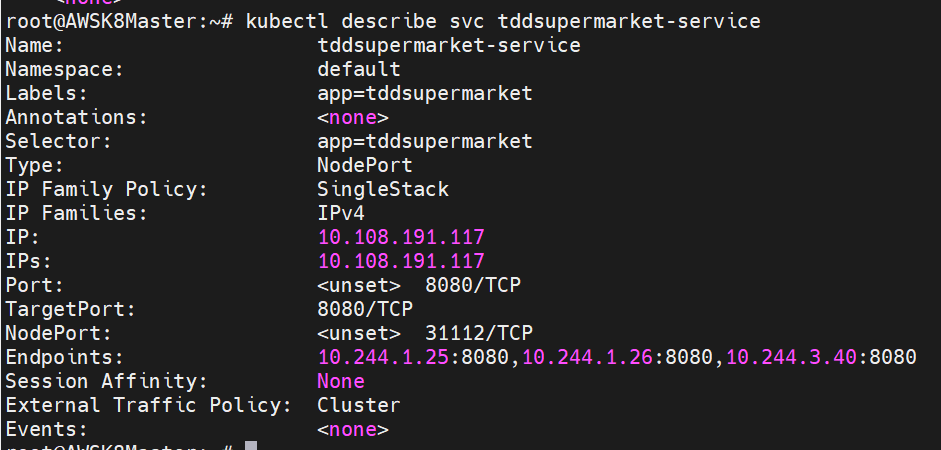
**kubectl create -f tddsupermarket-svc-np.yaml**



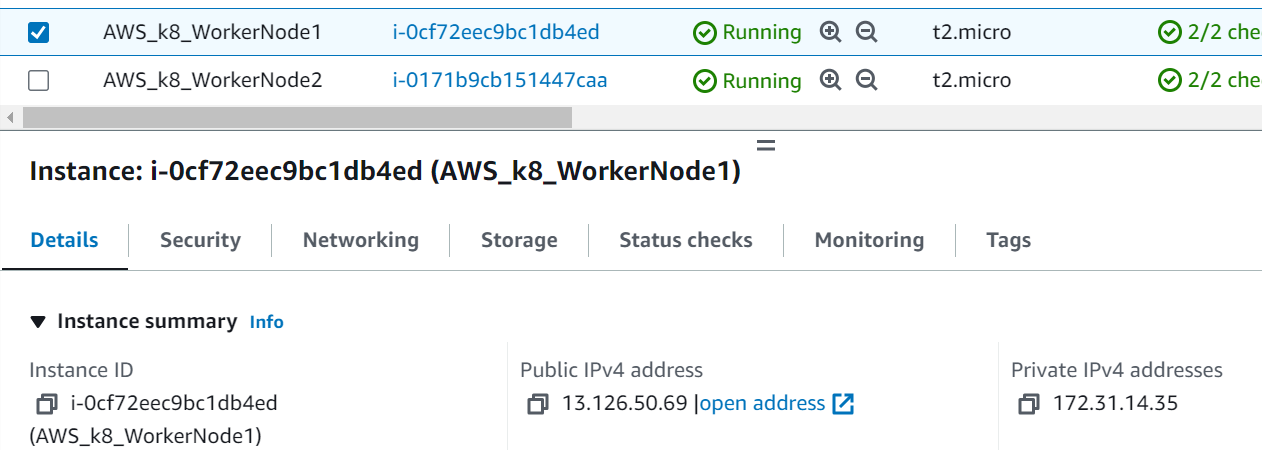
**kubectl get service**

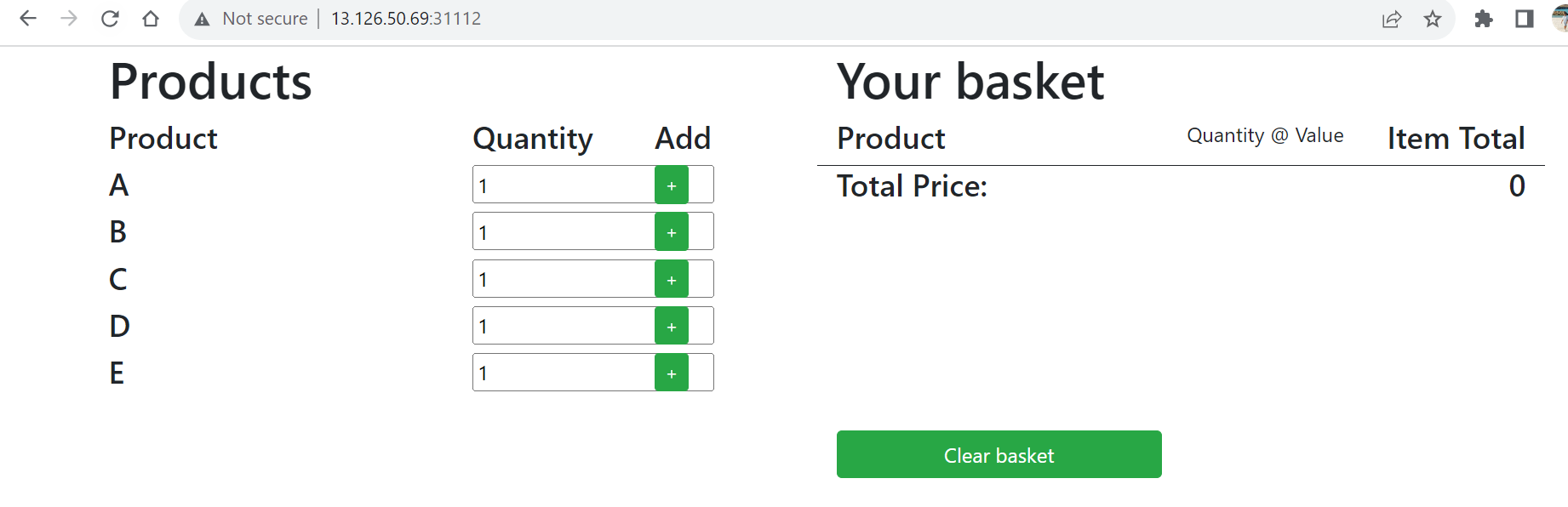


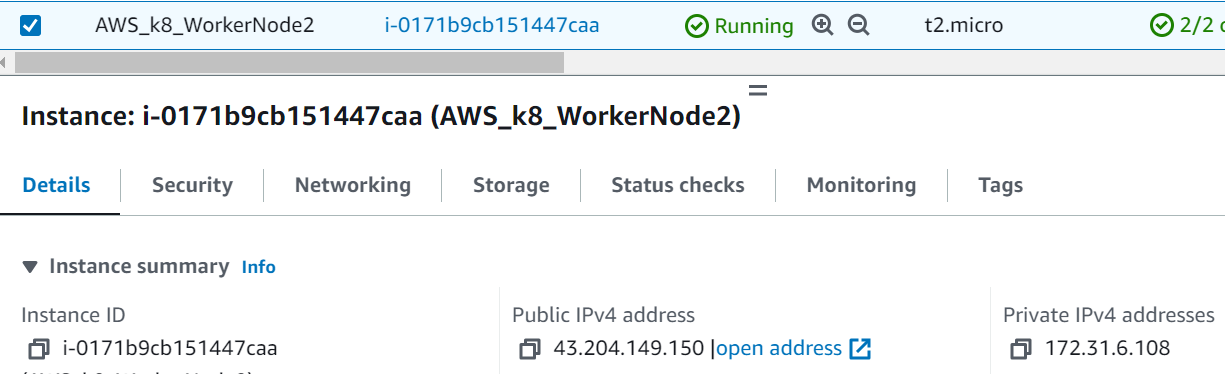
**kubectl describe svc tddsupermarket-service**

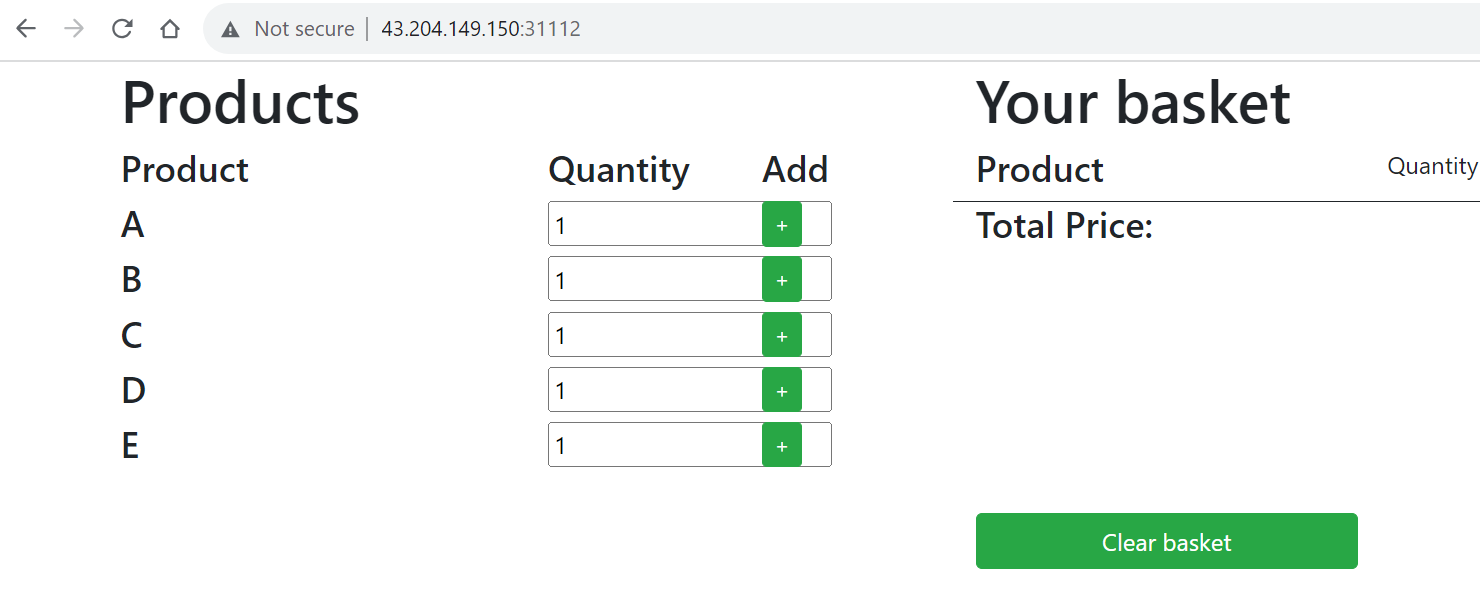


**The Public IP of my kubernetes worker nodes are below and using which I am able to access the application on port 31112(External Node Port) which is redirecting to Endpoints on port 8080(Internal Port).**



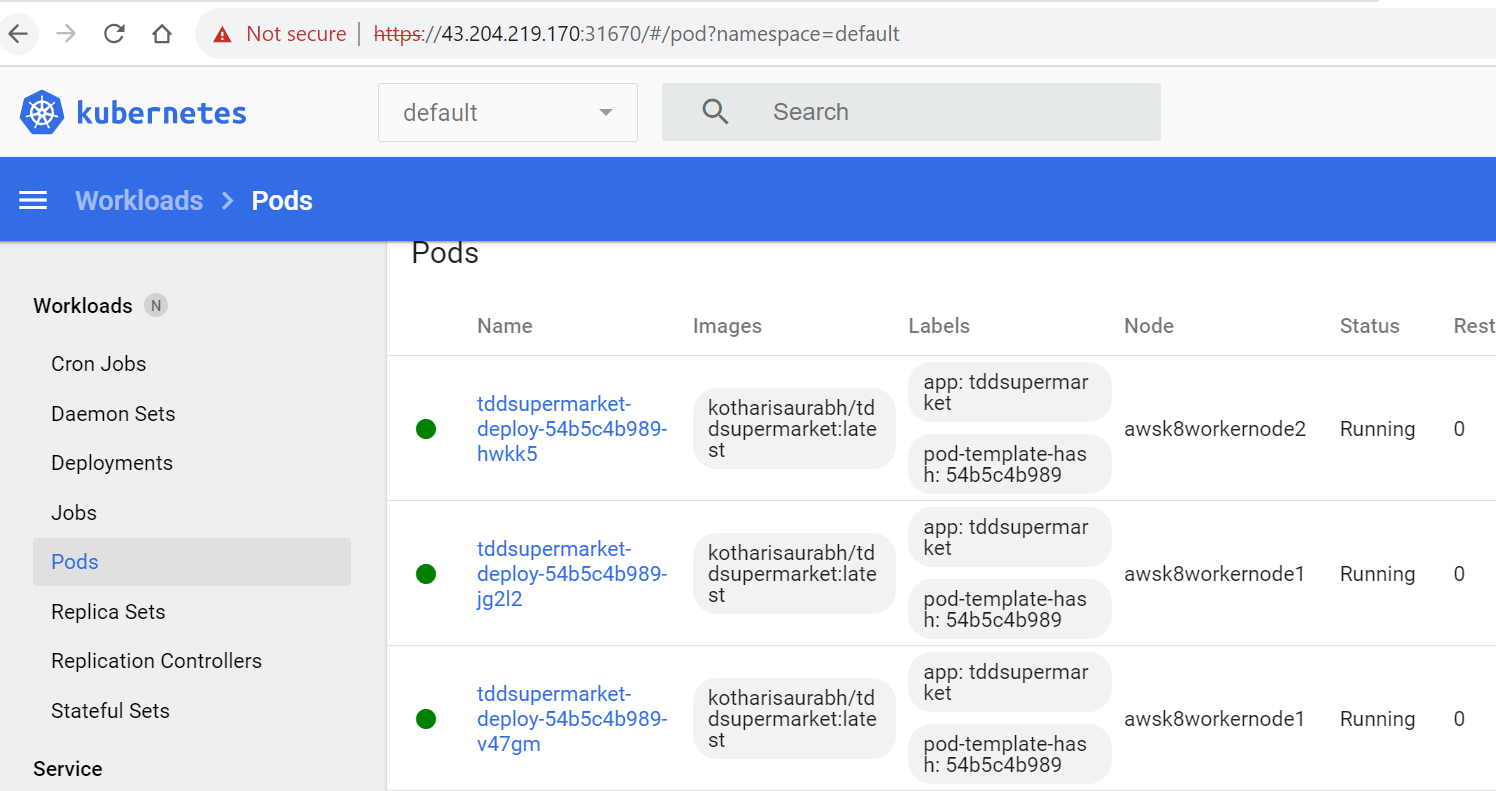




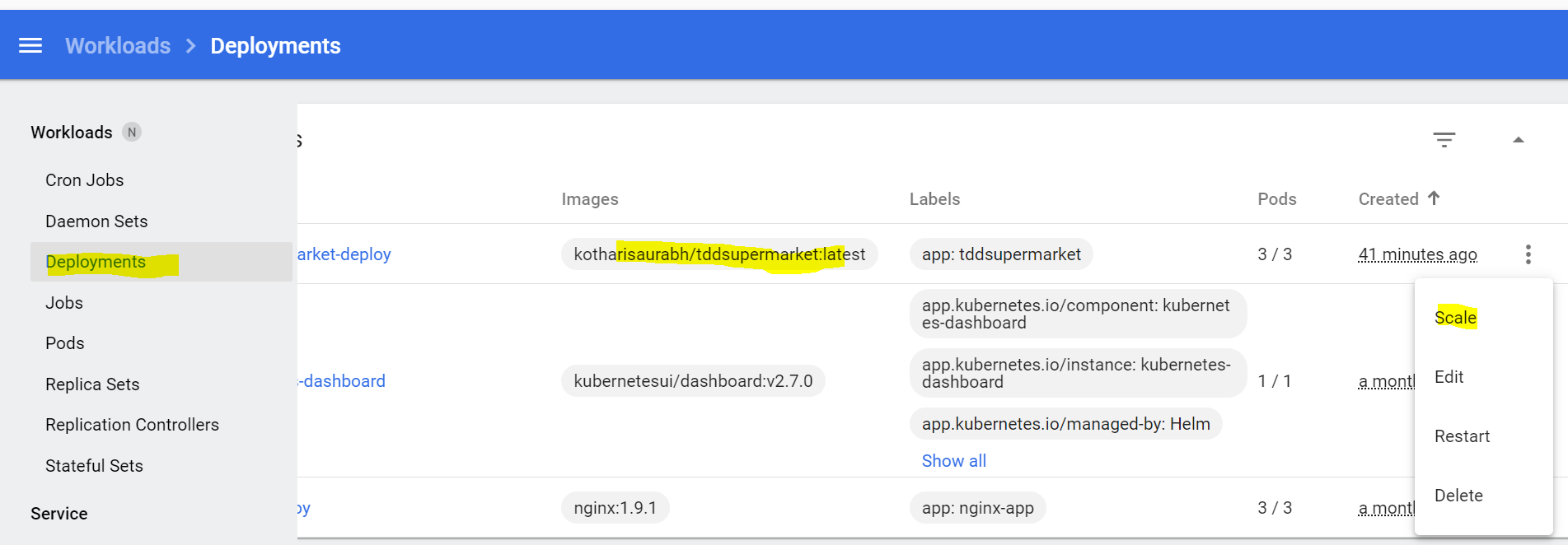


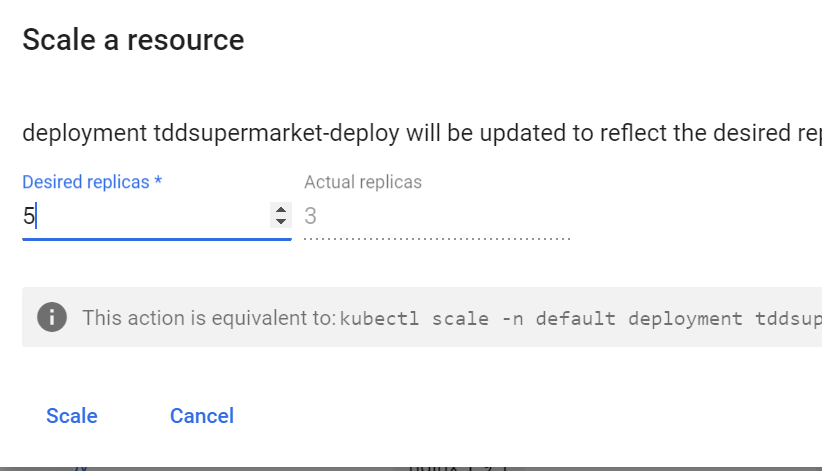
**I have also configured kubernetes dashboard, using which I can manage the kubernetes cluster.**

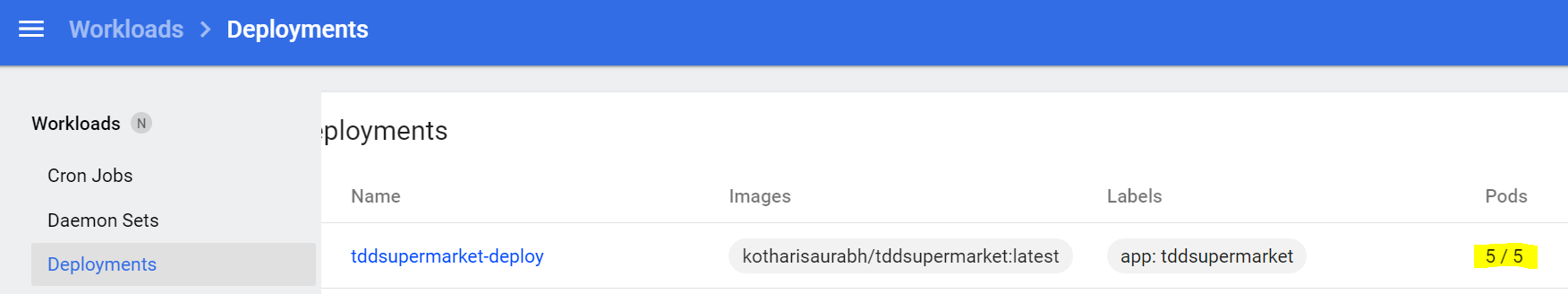
**I can see the application pods that I have deployed now.**

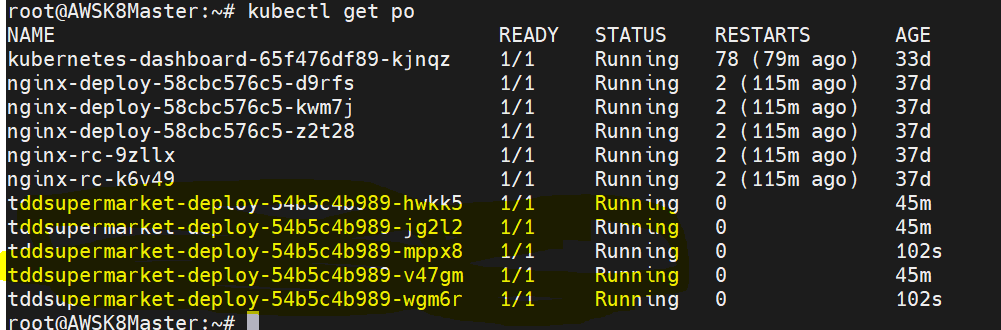


**I am able to scale up the pods from 3 to 5 without any issue.**









**On AWS, I haven’t worked on this deployment yet. But I am sure I will be able to once I get an opportunity. Thanks!**