TASK 1

Pre-requisites

1. An EC2 instance (ex: ubuntu) with port number 8080 open in security group as inbound rule with your IP (for security reasons)
2. Java 11 or above, Maven, Jenkins AWS CLI, Docker,
3. An ECR repo.
4. SpringbootApp repo

Actions

1. Download the Jenkins repo using wget

Wget curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \ /usr/share/keyrings/jenkins-keyring.asc > /dev/null

echo deb [signedby=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update

sudo apt-get install Jenkins

1. Start Jenkins

sudo systemctl start Jenkins

Enable Jenkins

sudo systemctl enable Jenkins

Check the status of Jenkins

sudo systemctl status Jenkins

(Note: Jenkins not starting up or taking time to start, check the machine you are working on if t2.micro upgrade it to t2.medium)

1. Have the IP address of the EC2 and hit it in the new tab with port number 8080

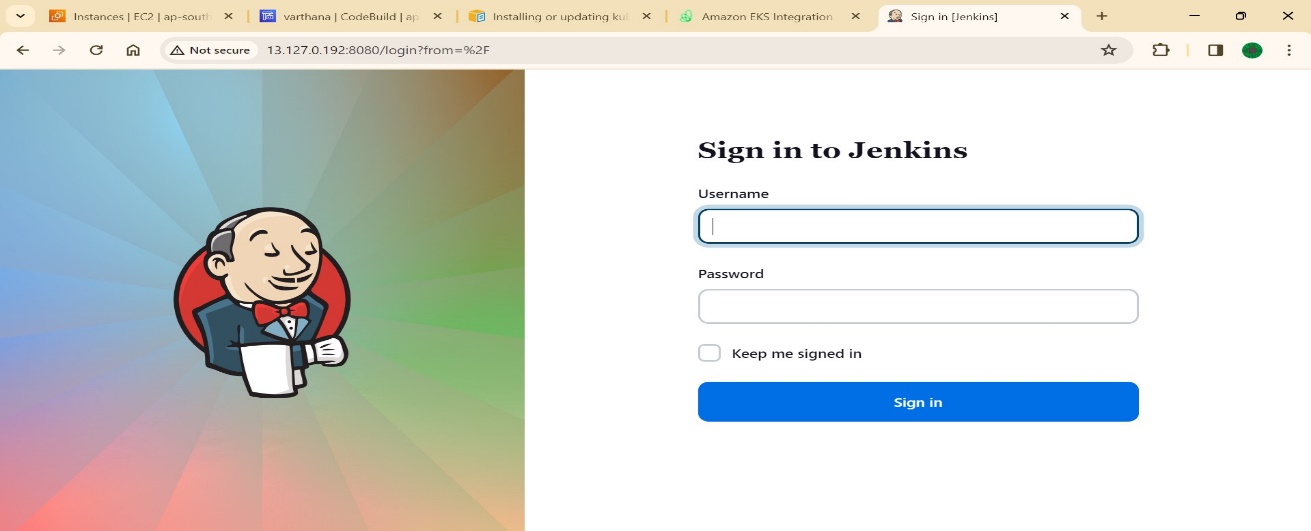
Ex: ip-addrress:8080

1. Asks for administrator password

Copy the above path into your machine with cat

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

copy and paste it into your Jenkins server

1. Asks for Jenkins plugins installation click on suggested plugins and proceed
2. Create a first admin user with password and save (abuateef-ab1234)
3. Jenkins is up and running

AWS CLI for AWS resources

1. Install AWS CLI into your machine using

sudo apt install awscli

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

sudo apt install unzip

unzip awscliv2.zip

sudo ./aws/install --update

aws –version, these commands will install aws cli by adding the aws cli repo



1. Install eksctl

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)

amd64.tar.gz" | tar xz -C /tmp

sudo mv /tmp/eksctl /usr/local/bin

eksctl version

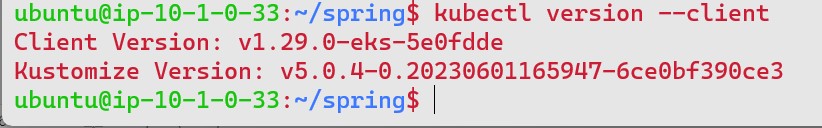


1. Install kubectl

sudo curl --silent --location -o /usr/local/bin/kubectl https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.6/2022-03-09/bin/linux/amd64/kubectl

sudo chmod +x /usr/local/bin/kubectl

kubectl version –client

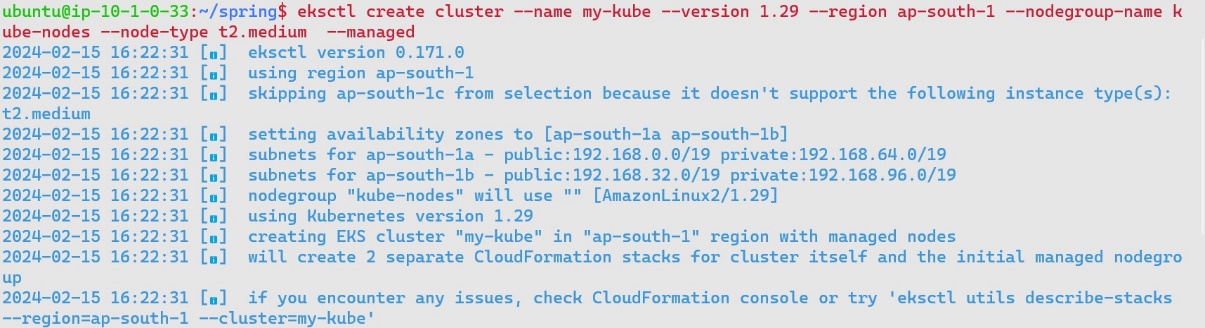


1. Shift to the Jenkins user

sudo su jenkins

1. After installing eksctl and kubectl, now to create a cluster on EKS using command

eksctl create cluster --name my-kube --version 1.29 --region ap-south-1 --nodegroup-name kube-nodes --node-type t2.medium --managed



1. sudo apt update

sudo apt install docker.io

docker --version to verify the installation of docker



1. apply these commands and update

sudo usermod -a -G docker jenkins

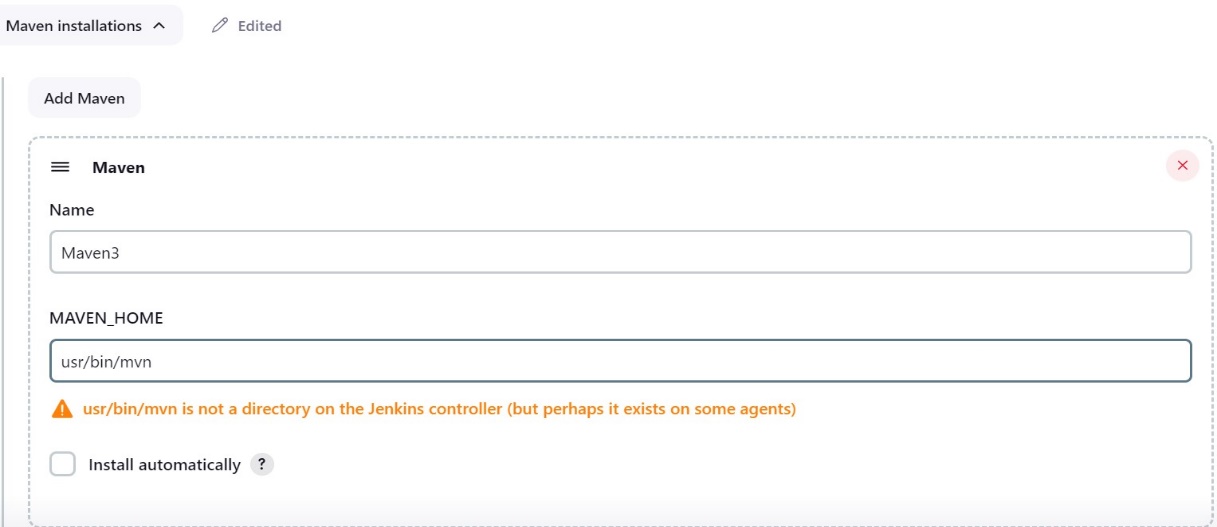
sudo service jenkins restart

sudo systemctl daemon-reload

sudo systemctl start docker

sudo systemctl enable docker

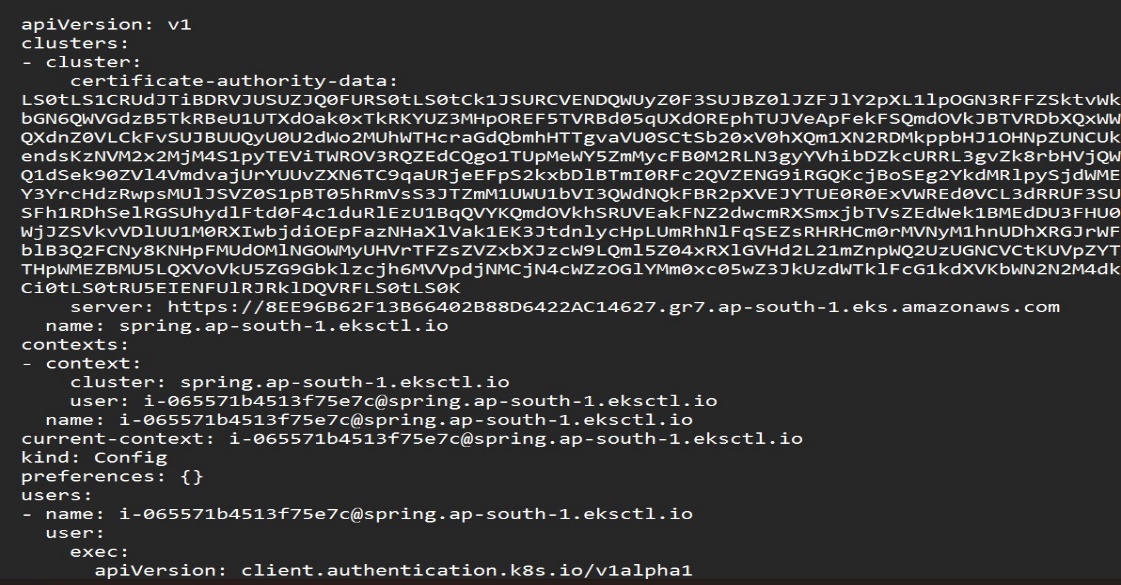
1. Navigate to Jenkins > plugins > available plugins and search for the following
2. Docker. Docker pipeline, Kubernetes cli and Kubernetes if required install accordingly
3. Navigate to Jenkins > manage Jenkins > tool configuration search for maven and register Maven3 as maven installation and save

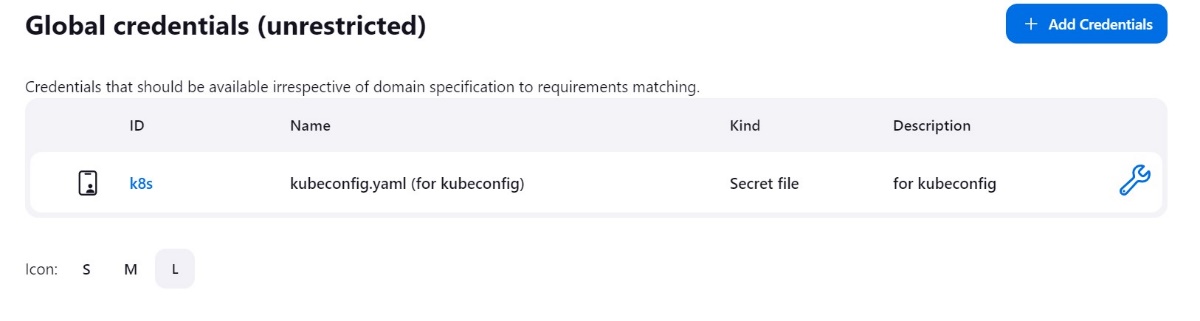


1. A configuration is required of the cluster to use as credentials in the Jenkins pipeline for Kubernetes, to obtain need to cat the filr config file of the cluster as

cat /var/lib/jenkins/.kube/config

save the above output as yaml extension to use in credentials



1. Open Jenkins and navigate to the manage Jenkins > credentials > global credentials > create credentials
2. Use secret file
3. Upload the saved file from the Jenkins path /var/lib/Jenkins/workspace/
4. Save it
5. Create a pipeline and run the declarative script after necessary changes such as
6. Source as your github repo
7. The ECR registry url
8. The docker login command
9. The docker push command
10. Save the pipeline and build, an image is created and pushed to the ECR
11. Include a stage as deploy and add the commands

stage('K8S Deploy') {

steps{

script {

withKubeConfig([credentialsId: 'k8s', serverUrl: 'https://8EE96B62F13B66402B88D6422AC14627.gr7.ap-south-1.eks.amazonaws.com']) {

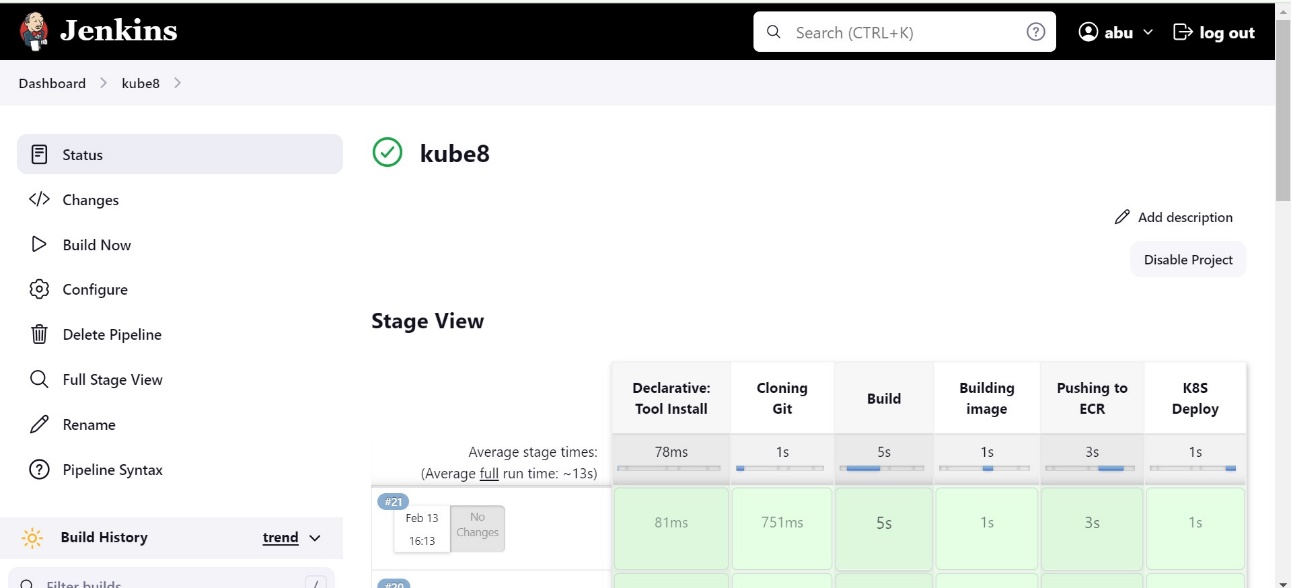
sh ('kubectl apply -f eks-deploy-k8s.yaml')

the serverURL is the url you will get from the config file you obtained doing

cat /var/lib/jenkins/.kube/config

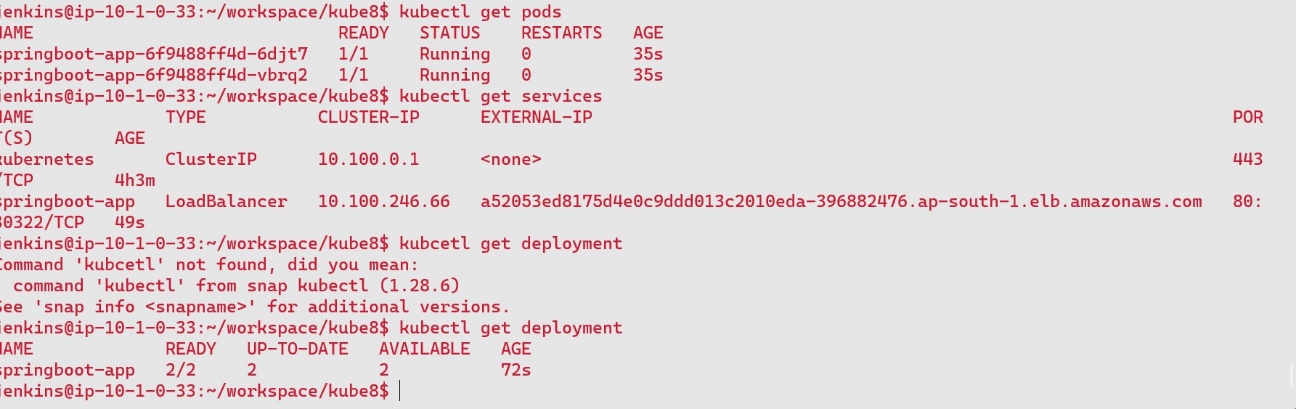
copy the URL and paste it in the serverURL, and path o fthe deployment.yaml file and build again

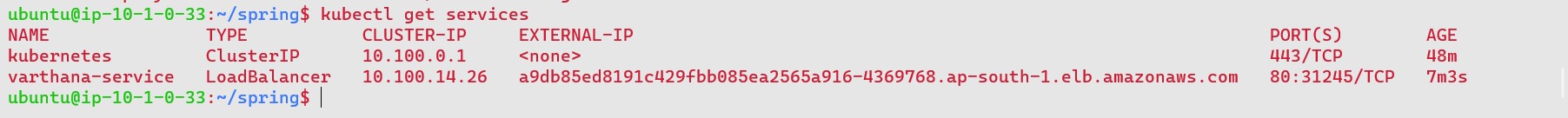
1. Build successful refers to the deployment also



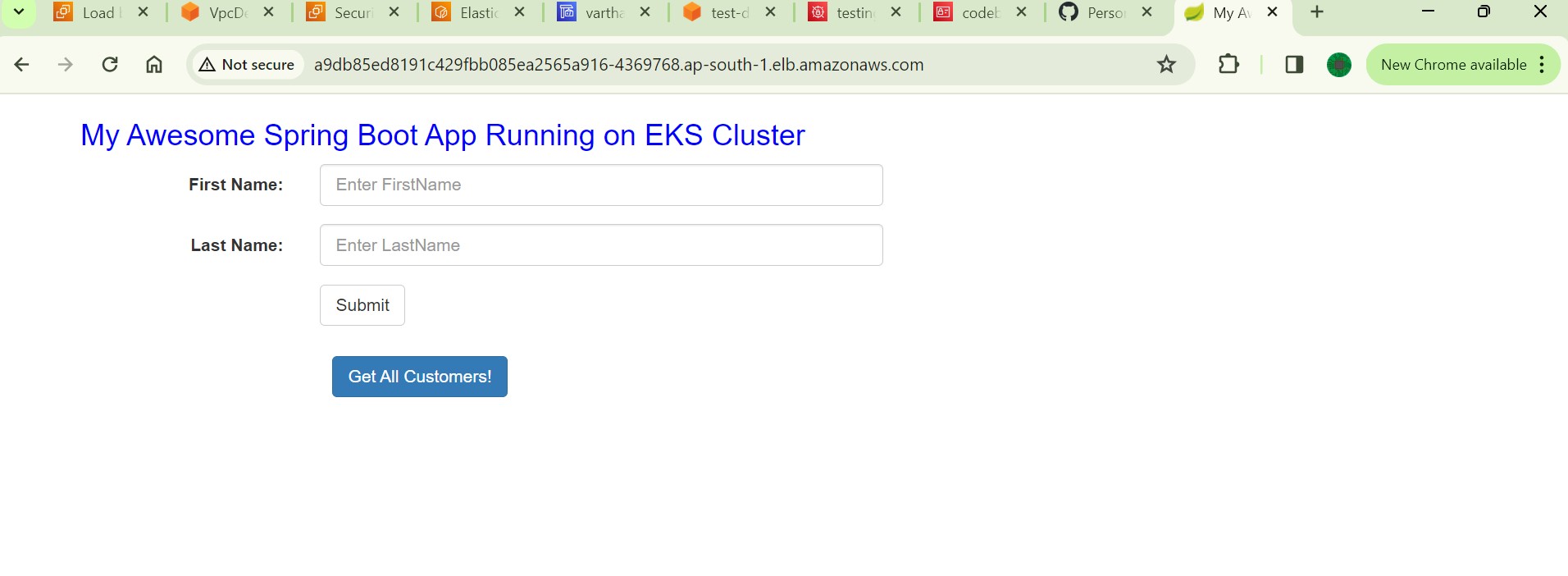


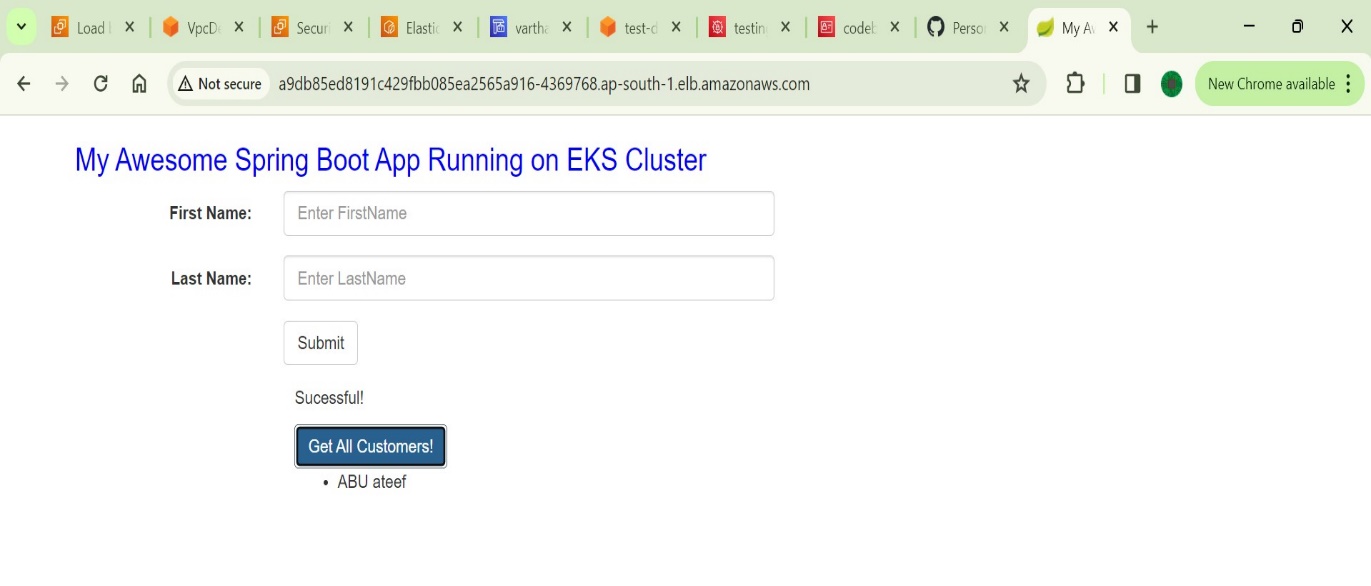
1. Run kubectl get pods to verify pods running



Kubectl get services for the loadbalancer service

1. Copy the load balancer URL and hit it in the browser if errors occurred the springbootApp will be accessible.





Done

Link for files https://drive.google.com/drive/folders/1oy7eqM\_8x8gvWF\_Uvl13SUggZg9t0S\_n?usp=drive\_link