**Capstone Project**

**DevOps ReactJSShoppingApp**

Contents

[Softwares & Tools Used: 3](#_Toc171422652)

[GitHub: 4](#_Toc171422653)

[Jenkins: 5](#_Toc171422654)

[Docker hub: 8](#_Toc171422655)

[Terraform: 10](#_Toc171422656)

[Ansible: 12](#_Toc171422657)

[Docker & Docker Compose: 12](#_Toc171422658)

[AWS: 12](#_Toc171422659)

[Monitoring: 15](#_Toc171422660)

[Work Flow: 18](#_Toc171422661)

[Deployed Site : 19](#_Toc171422662)

[Challenges Faced: 20](#_Toc171422663)

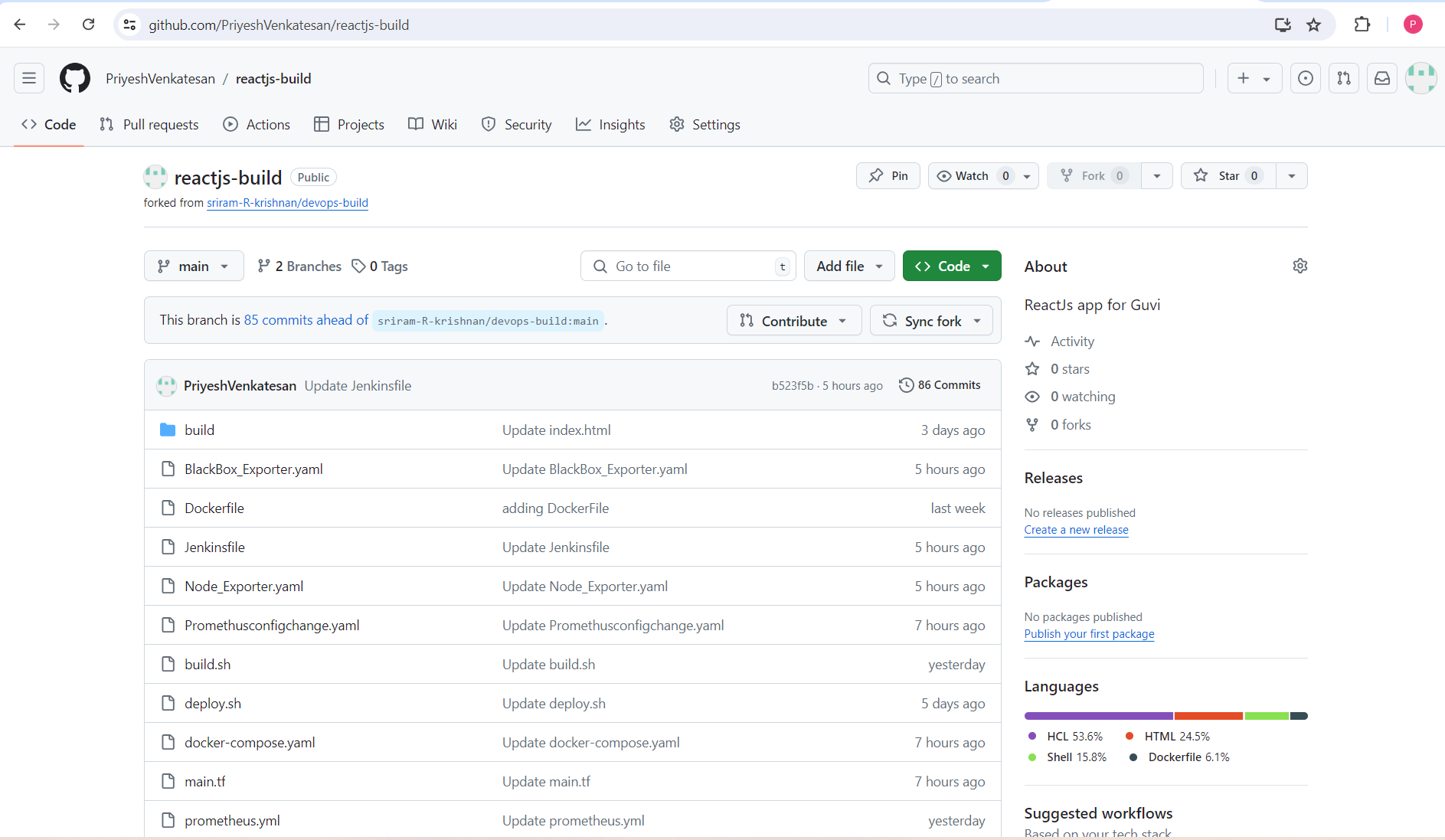
[Limitations: 20](#_Toc171422664)

# Softwares & Tools Used:

* AWS EC2
* Jenkins
* Docker
* Docker Compose
* DockerHub
* Git & GitHub
* Terraform
* Ansible
* Bash scripting
* Prometheus & Alert Manager
* Slack

# GitHub:

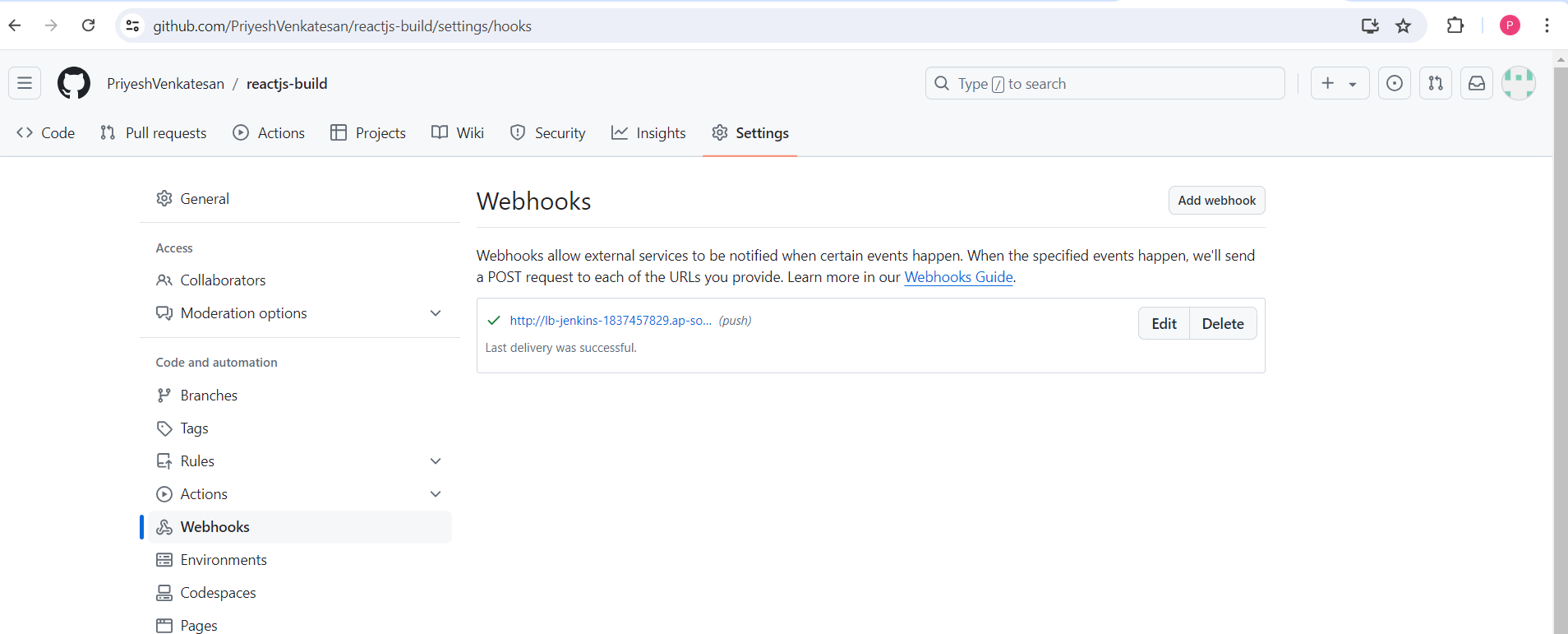
Repo: <https://github.com/PriyeshVenkatesan/reactjs-build>



GitHub Branches:

* Dev
* main

GitHub WebHook:



##### Notes:

1) Github repo was forked from parent

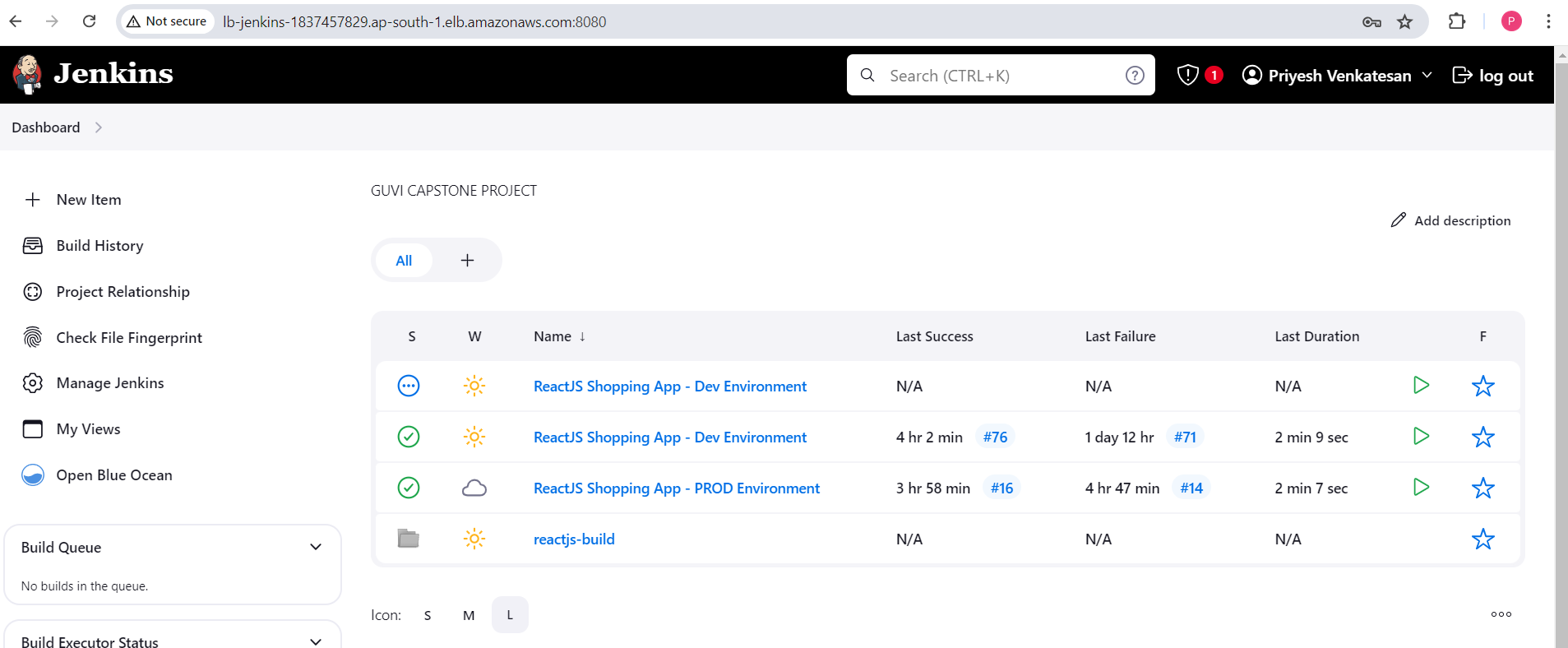


2) Webhook configured so that Dev and main branch can trigger pipeline,once pushed to remote repo

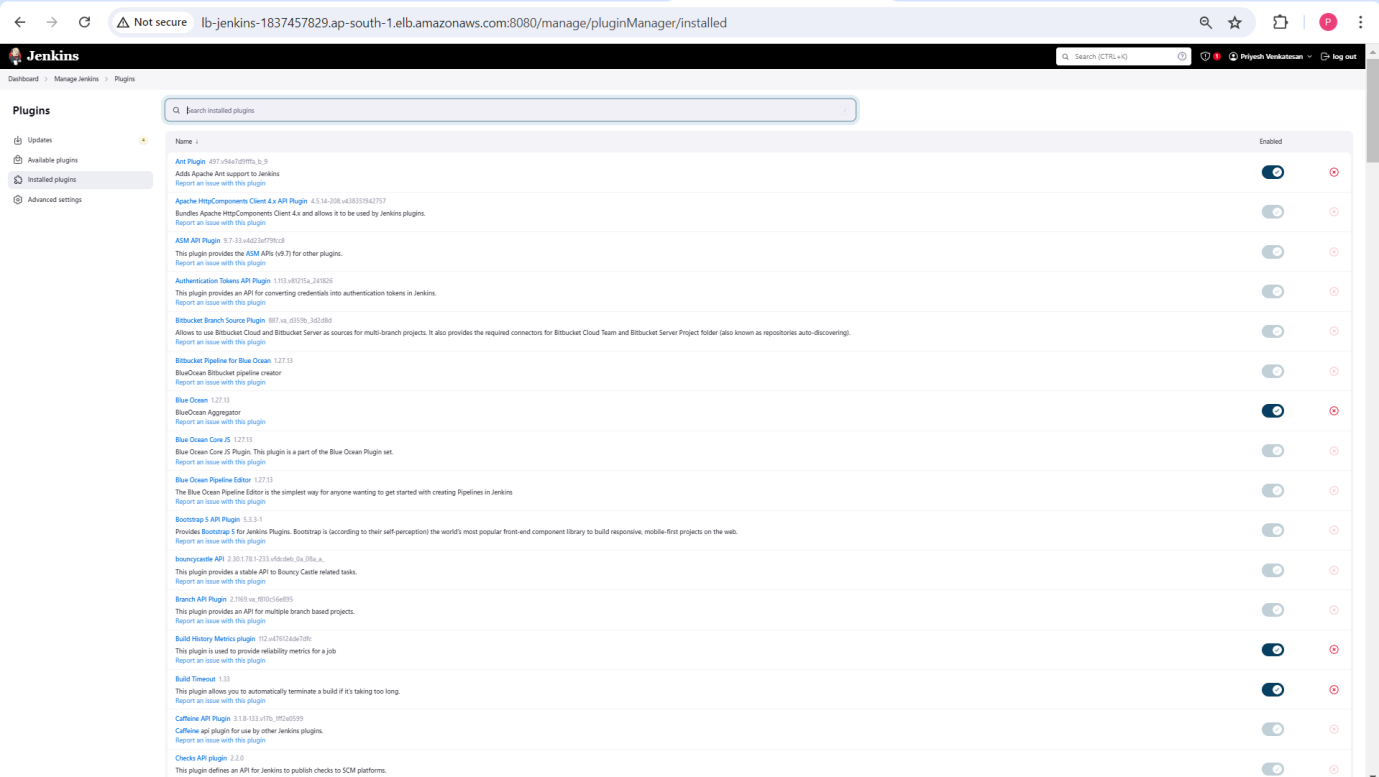
# Jenkins:

URL: <http://lb-jenkins-1837457829.ap-south-1.elb.amazonaws.com:8080/login>

Jenkins Home Page:



Jenkins Plugin Page:

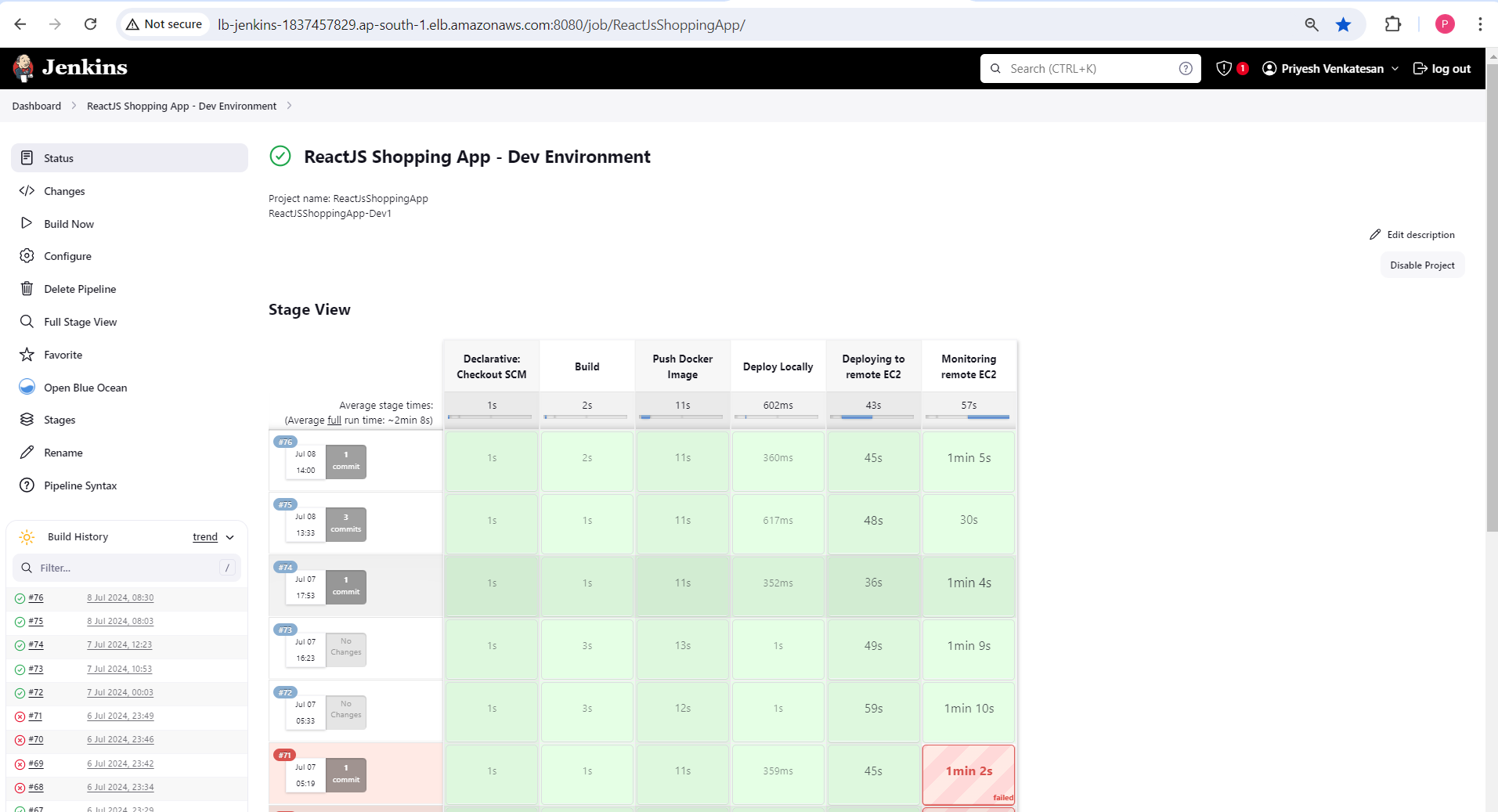


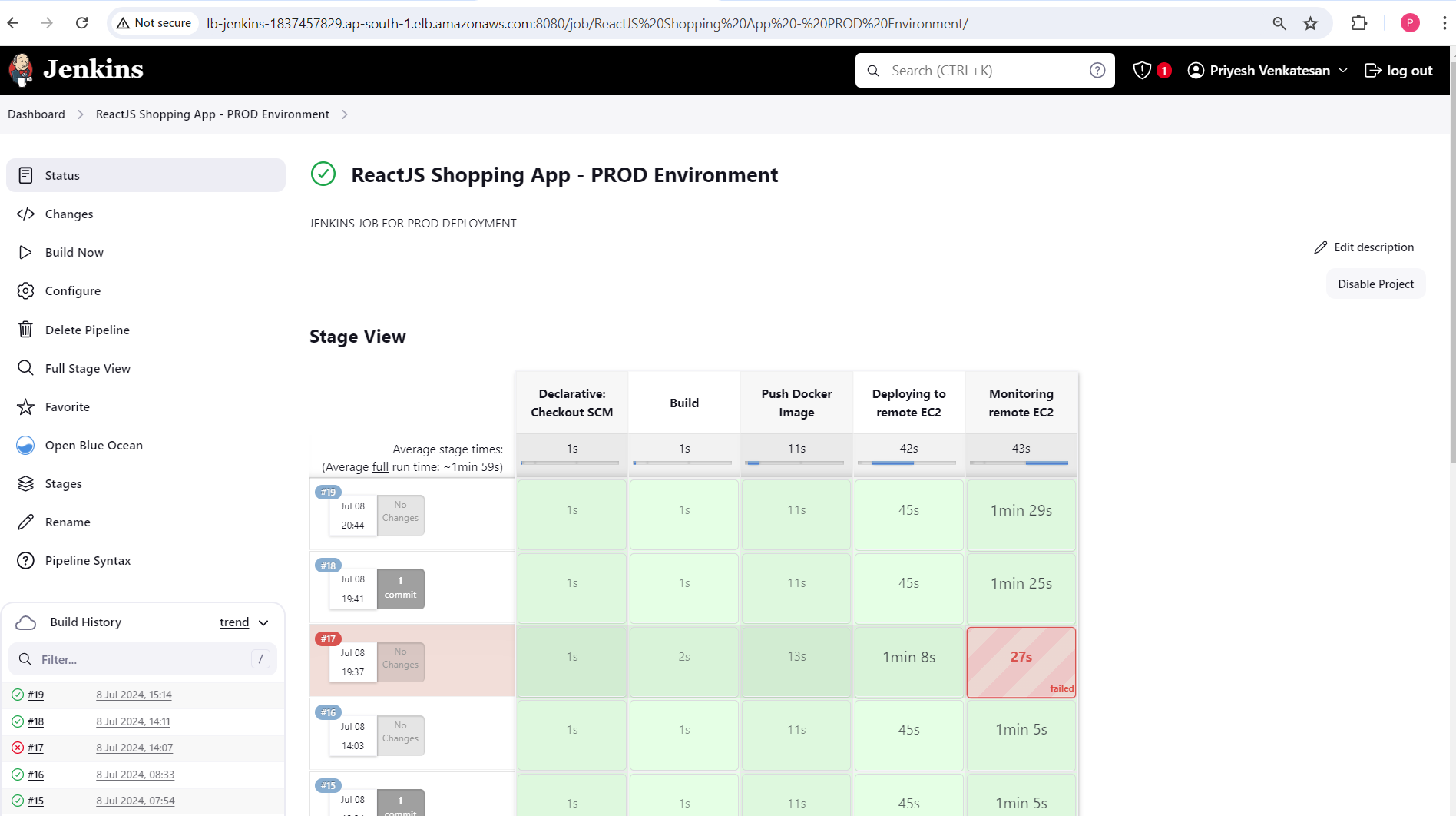
Jenkins Plugins Installed:

* BlueOcean
* Thin Backup
* Docker Plugin
* StageViewPlugin

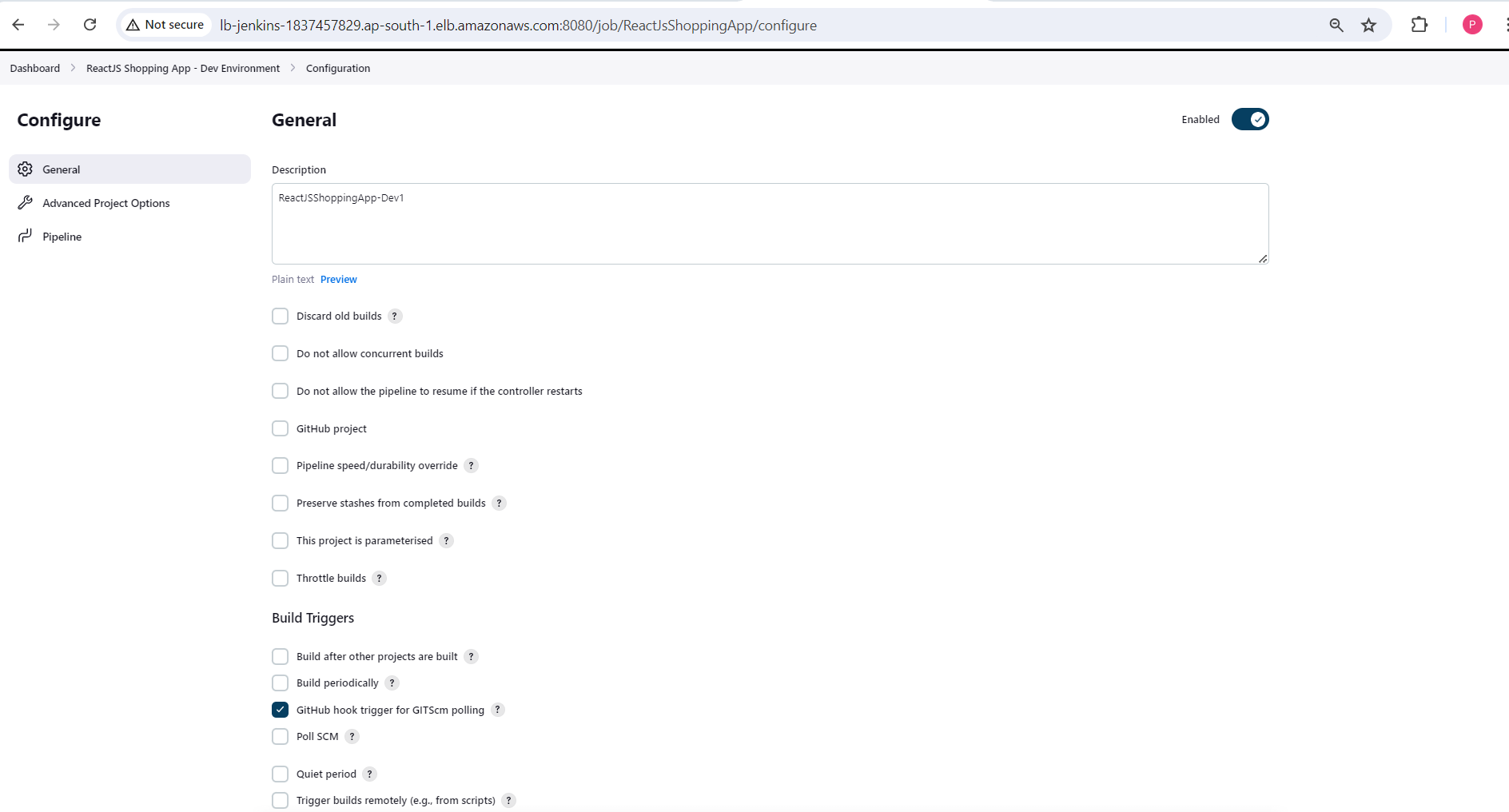
Jenkins PipeLine Configuration:

* Dev:
* Build History & Stage view

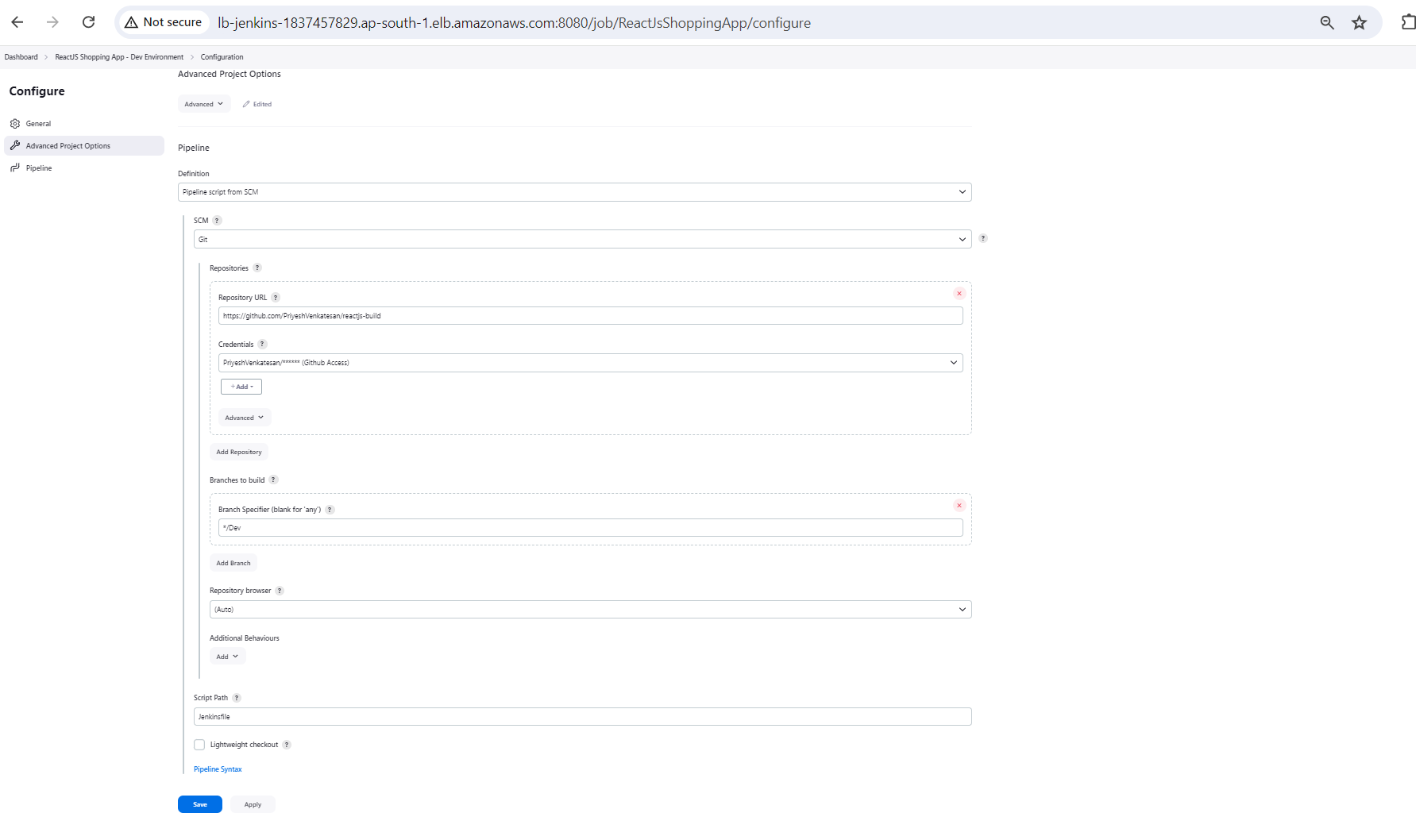




Pipleline Config View\_1:



Pipeline ConfigView\_2



Declarative Pipeline/Jenkins Files:

Using Jenkins file we are performing below stages:

Stage 1: Build the docker image with Dockerfile using bash scripting (build.sh)

Stage 2: using docker plugin, we are pushing the docker image to the repository

Stage 3: using Terraform we are launching new EC2 instance and deploying the site

Stage 4: we are installing exporters for Prometheus monitoring and configuring Prometheus config to update settings in Jenkinsserver

##### **Notes**:

1) We have to manually change github hook URL everytime,when server restarts,so configured LoadBalancer to avoid this issue

2) Jenkins env variable ${BUILD\_NUMBER} is used as tag for DockerImages

3) Jenkins host servers, we have also installed Prometheus,AlertManager,Ansible,Docker,DockerCompose

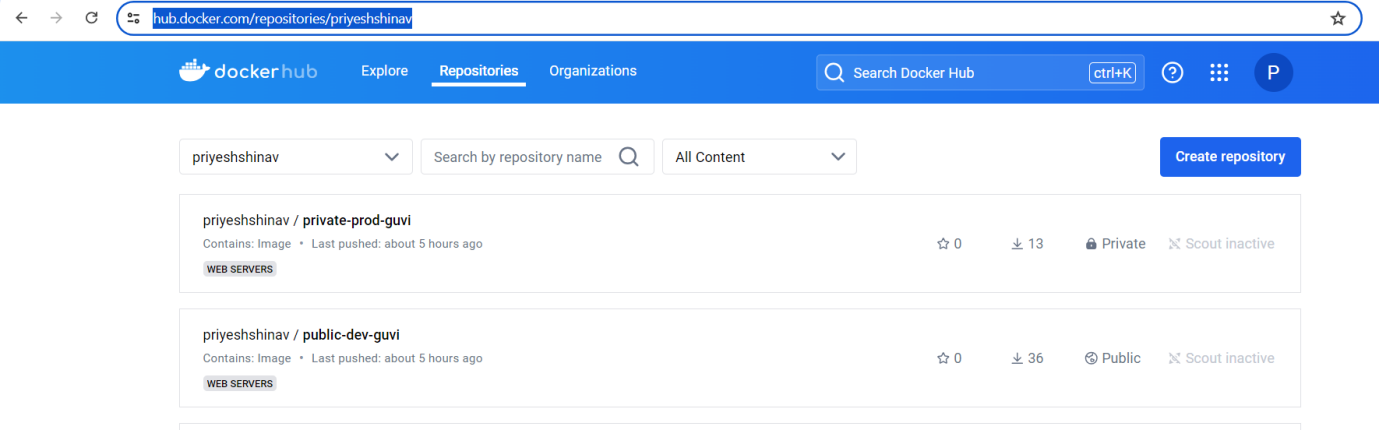
# Docker hub:

Dockerhub UI:

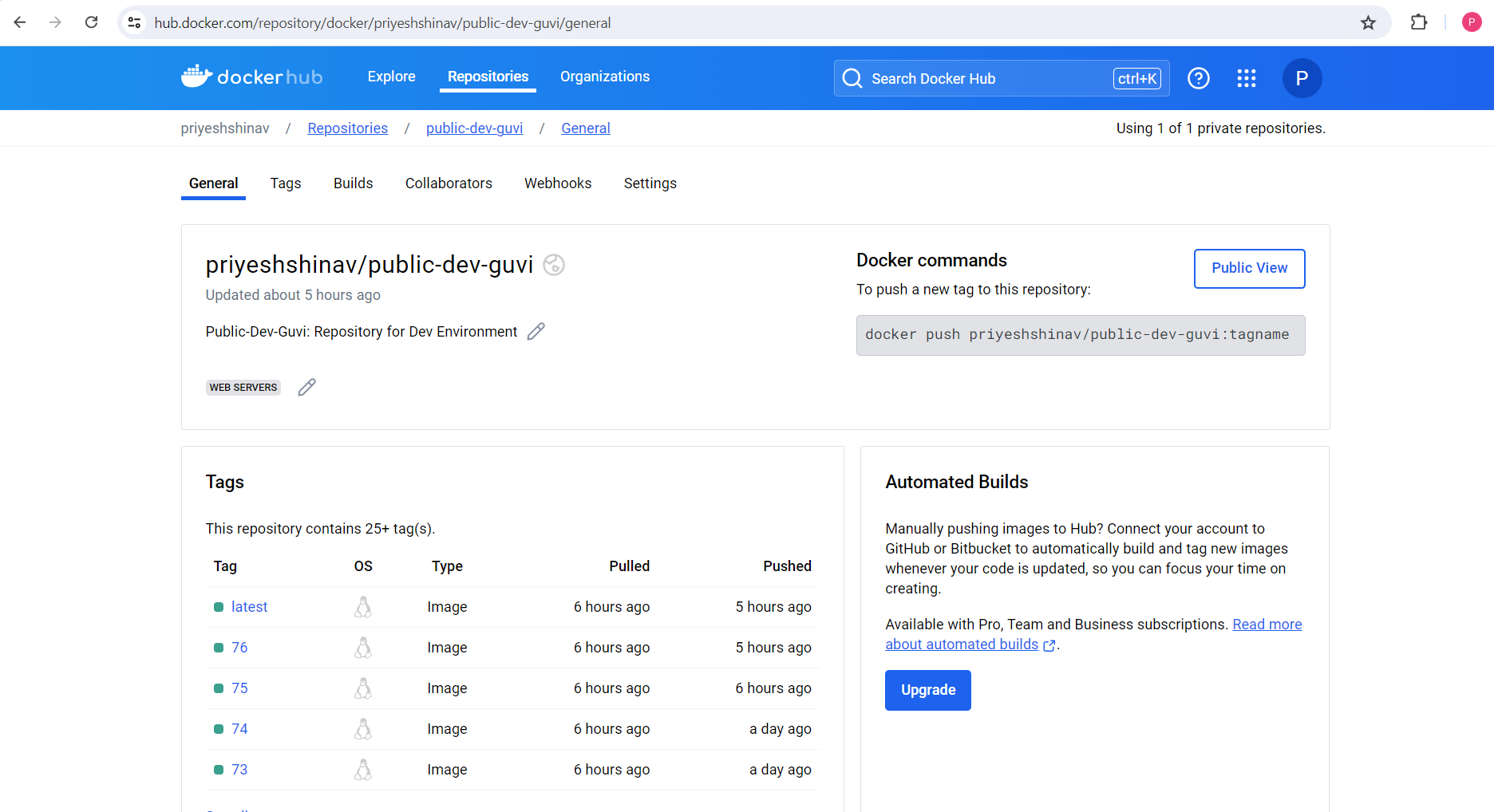
<https://hub.docker.com/repositories/priyeshshinav>

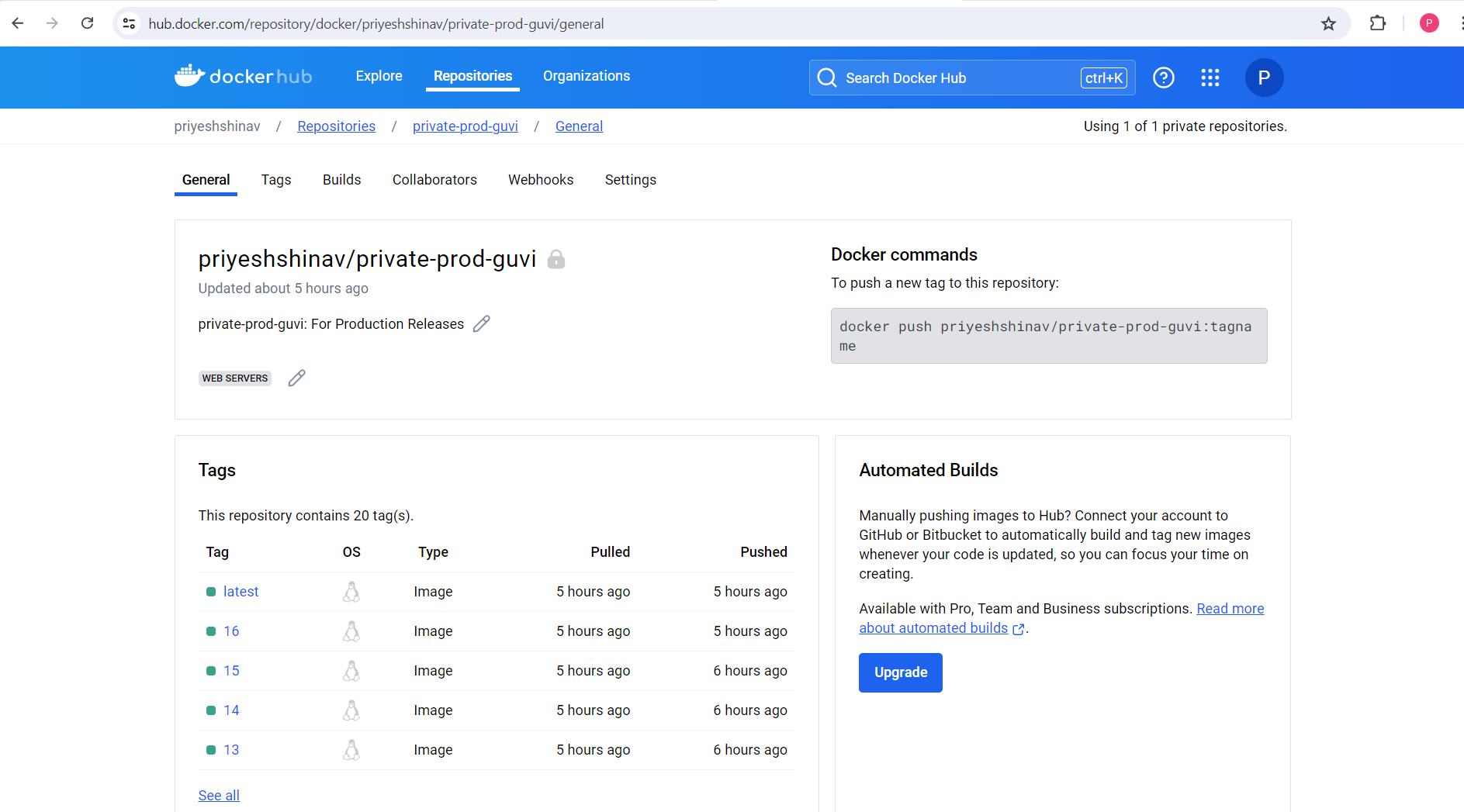
Dockerhub Repos:

* priyeshshinav/private-prod-guvi – Private repo
* priyeshshinav/public-dev-guvi -Public repo



Docker repos with Tags:





##### Notes:

For Private repos, docker login is mandatory, it was done with the help of --password-stdin

# Terraform:

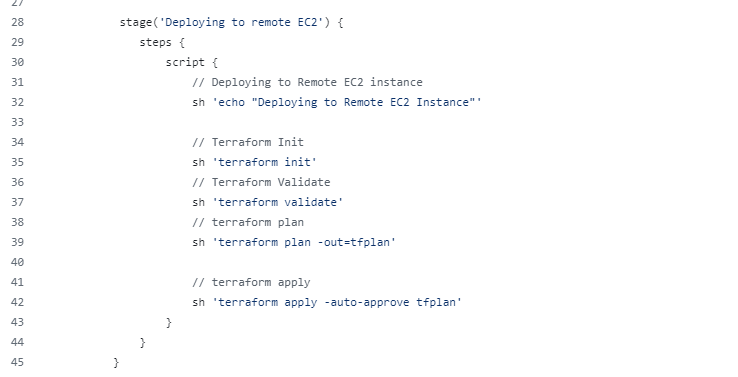
We have configured main.tf, in such a way that , EC2 instance t2.micro is created and using user data we are installing docker/dockercompose and then running dockercompose via deploy.sh



##### Userdata:



Then using Jenkins file, we are running below stage block to launch Ec2 instance and deploy app into remote instance



# Ansible:

We have installed ansible core on Jenkins Server, using ansible playbook we are installing

* NodeExporter
* BlackBoxExporter
* Modifying Prometheus.yml in Jenkins server wrt to remote EC2 instance and restart

We have used modules like copy,service,replace,command,shell for this activity as well as using aws ec2 plugin for dynamic inventory

# Docker & Docker Compose:

Docker Image is built with DockerFile , we have used following commands COPY,EXPOSE,WORKDIR,CMD

The build image is tagged wrt to Public and Private Repo and Jenkins Build Number

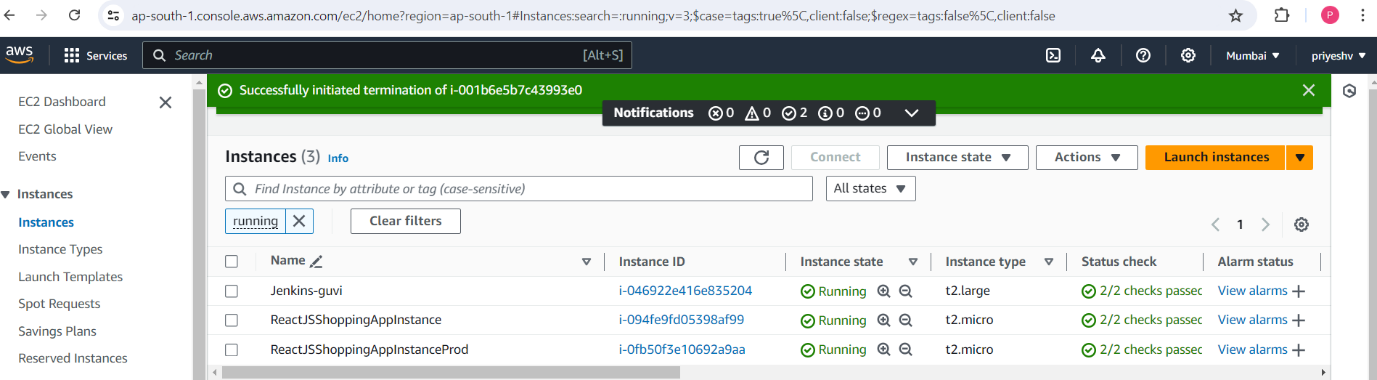
Then we use docker compose to run the container and following commands were used in dockercompose.yml like image,port

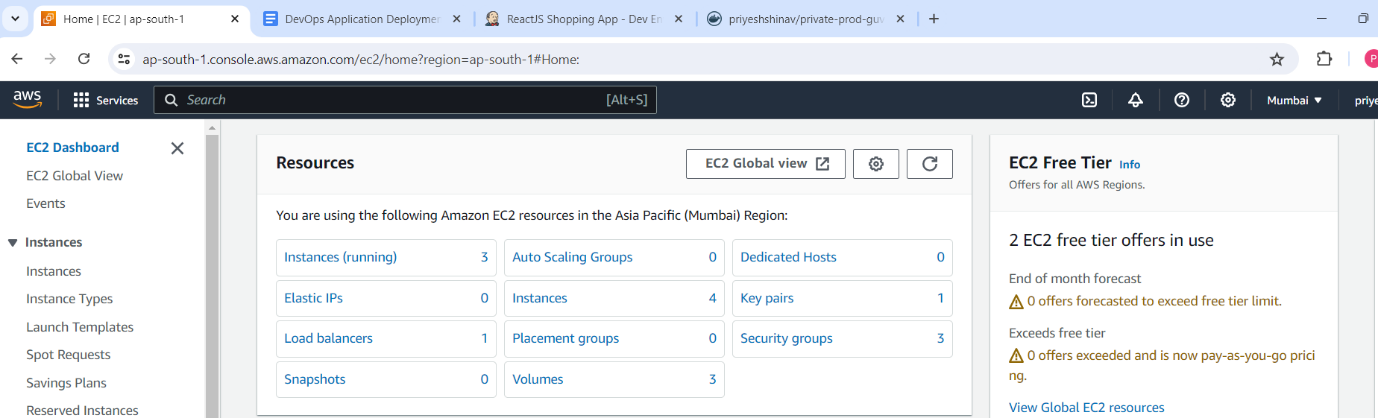
Dev ImageName: priyeshshinav/public-dev-guvi:latest -t priyeshshinav/public-dev-guvi:$1 .  
Main Image Name: priyeshshinav/private-prod-guvi:latest -t priyeshshinav/private-prod-guvi:$1

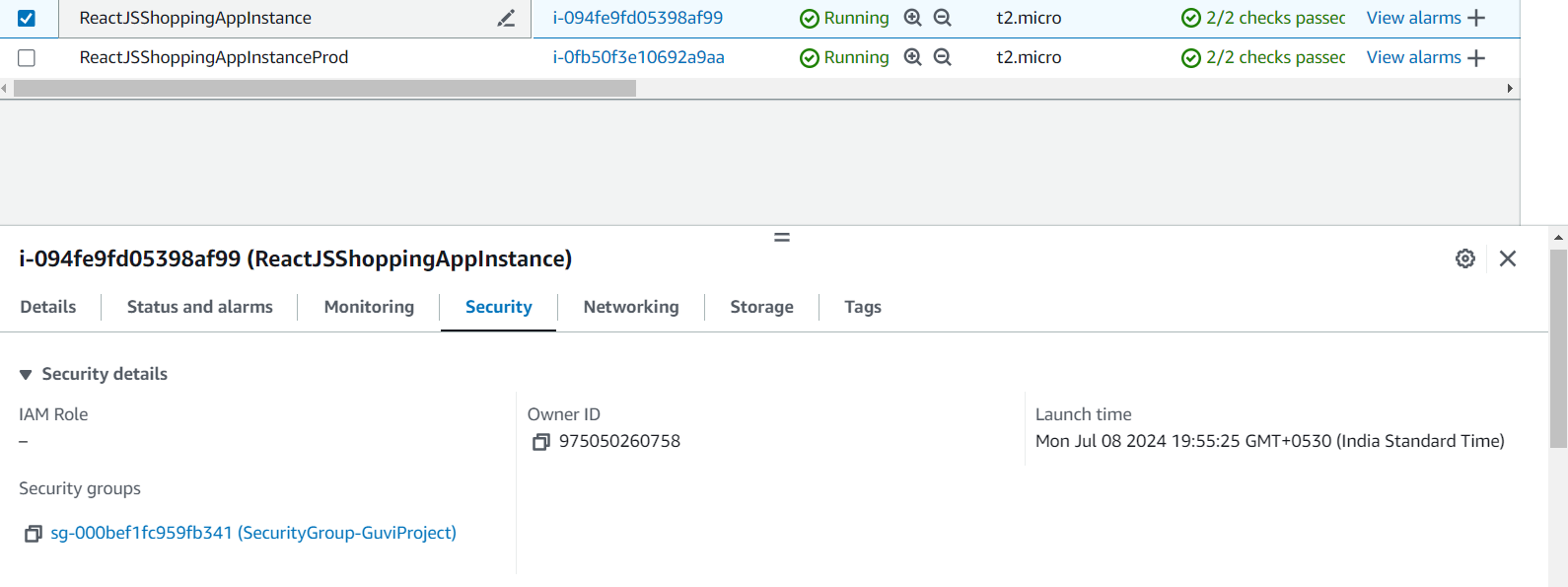
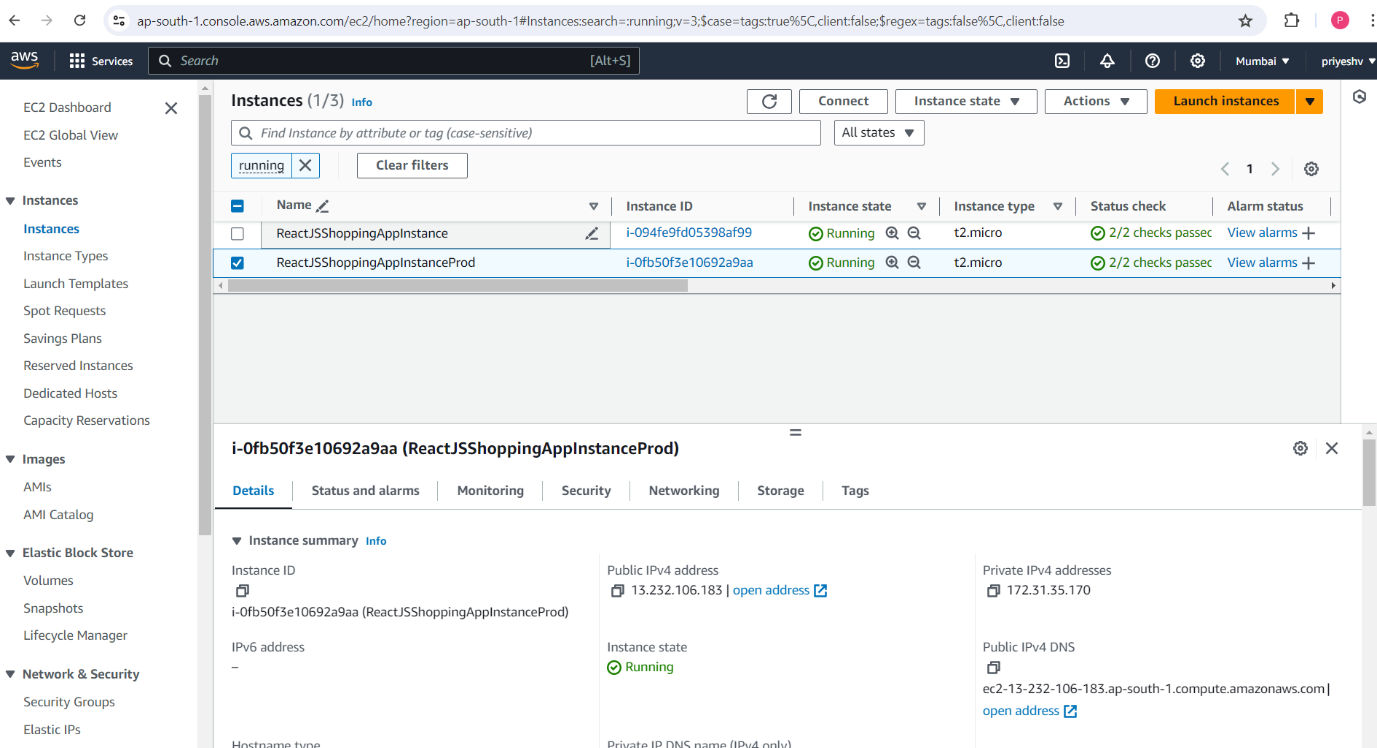
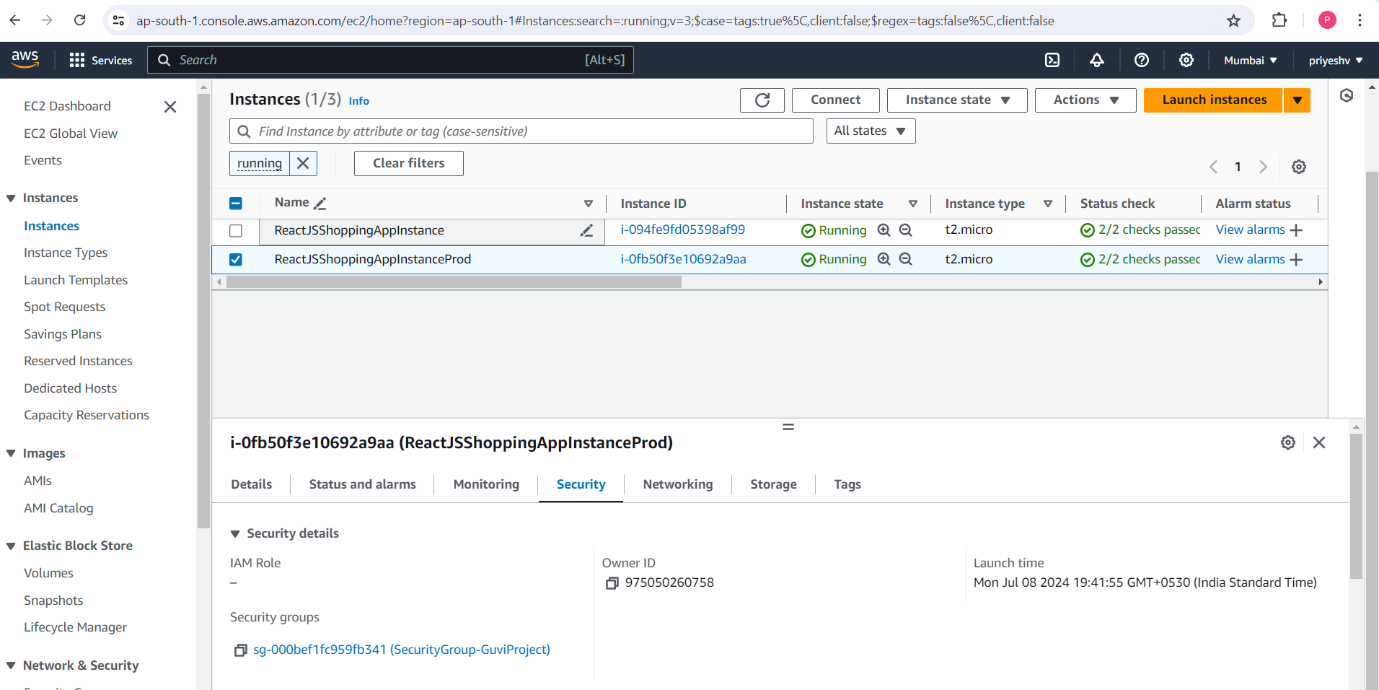
Where $1 is Jenkins Build Number

# AWS:

Ec2 Console:



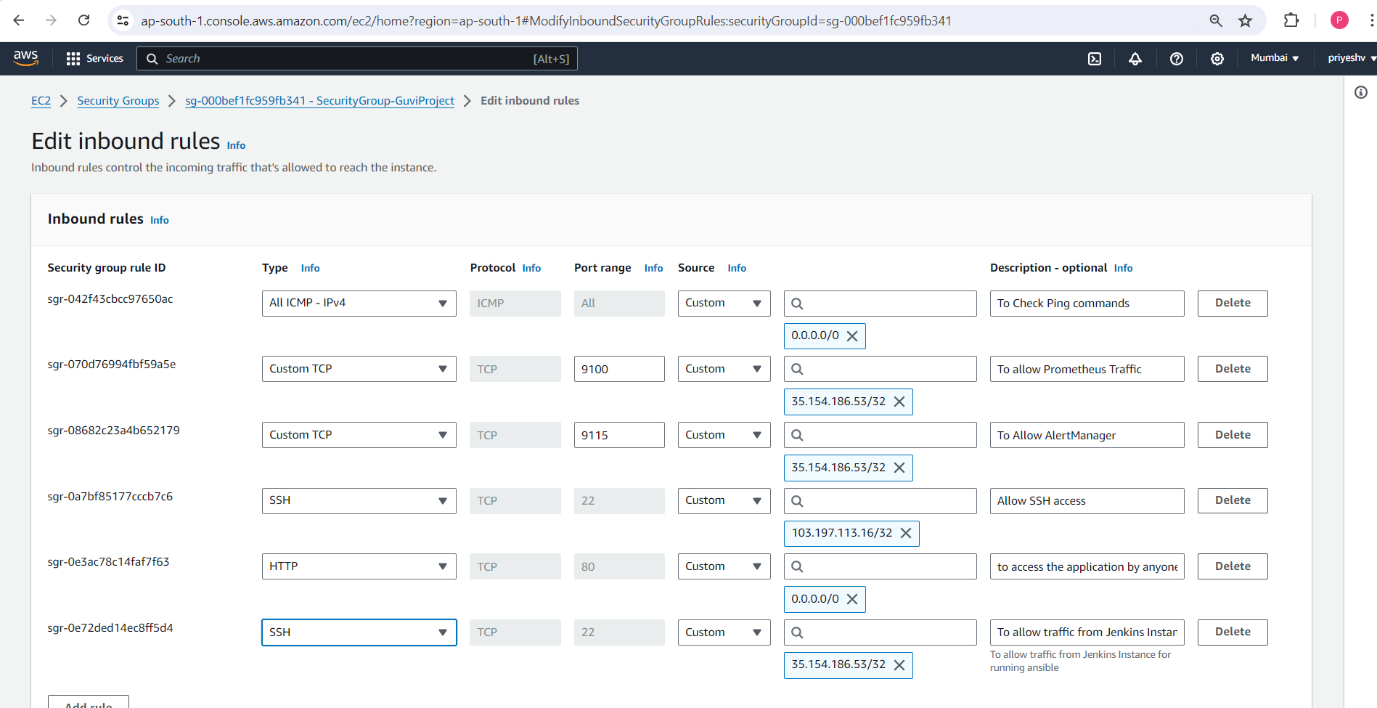
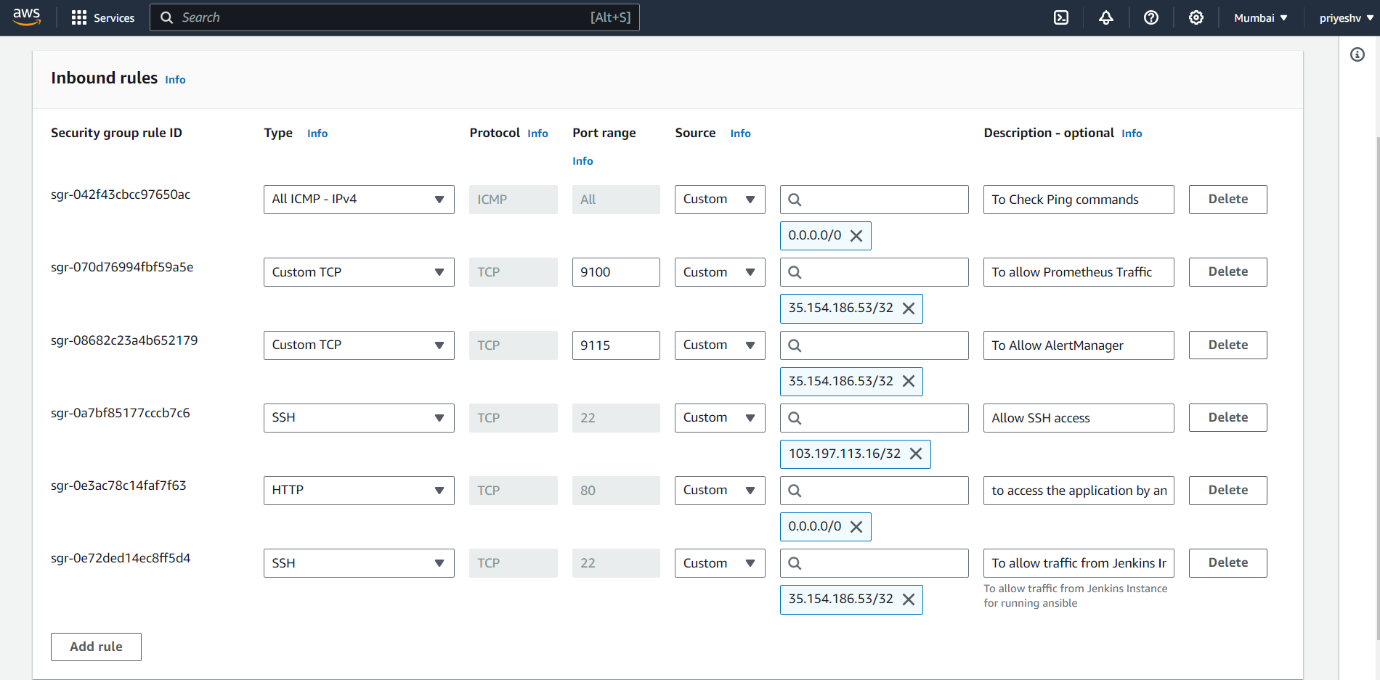




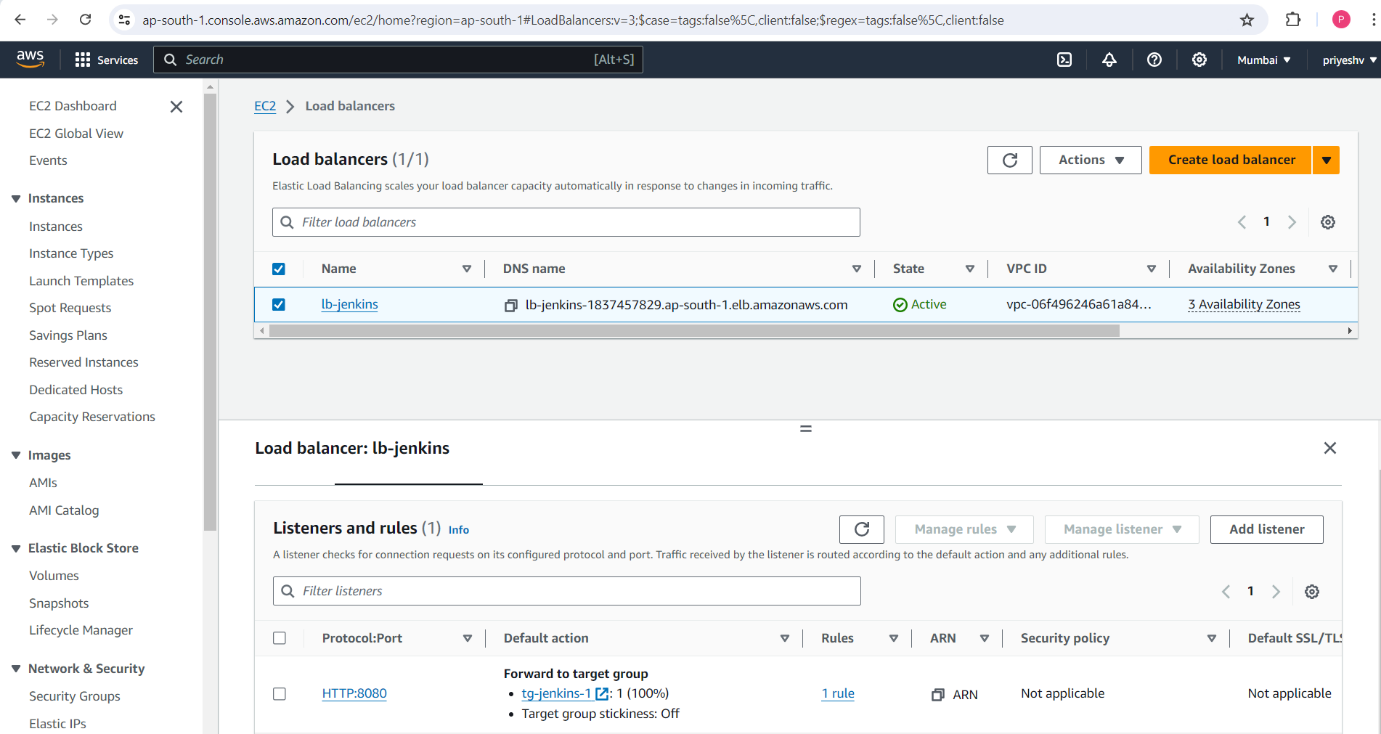
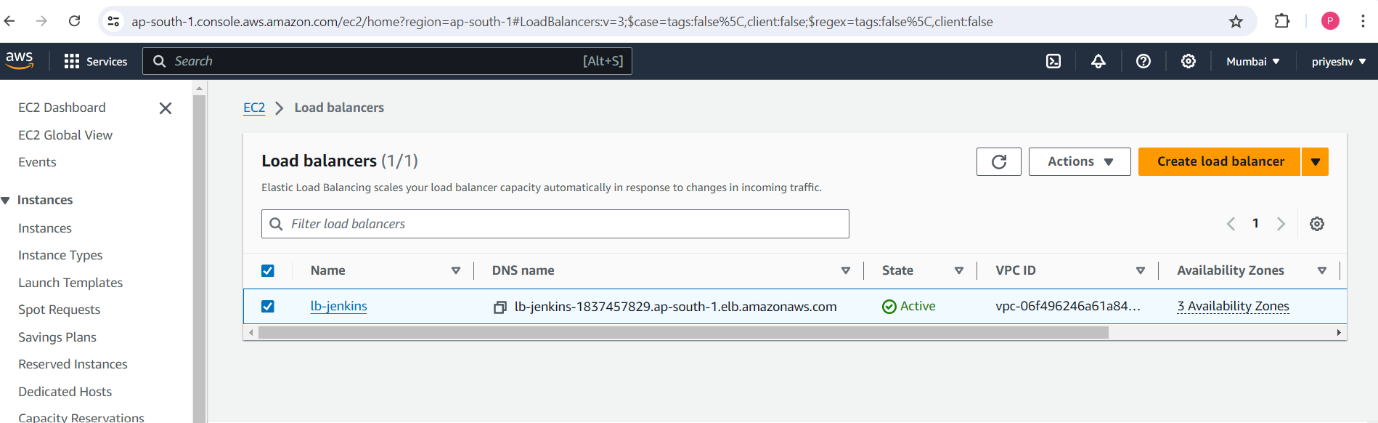


Security Group:

Apart from 2 rules mentioned, we have added few more rules to integrate with ansible /Prometheus and AlertManager



Load Balancer:



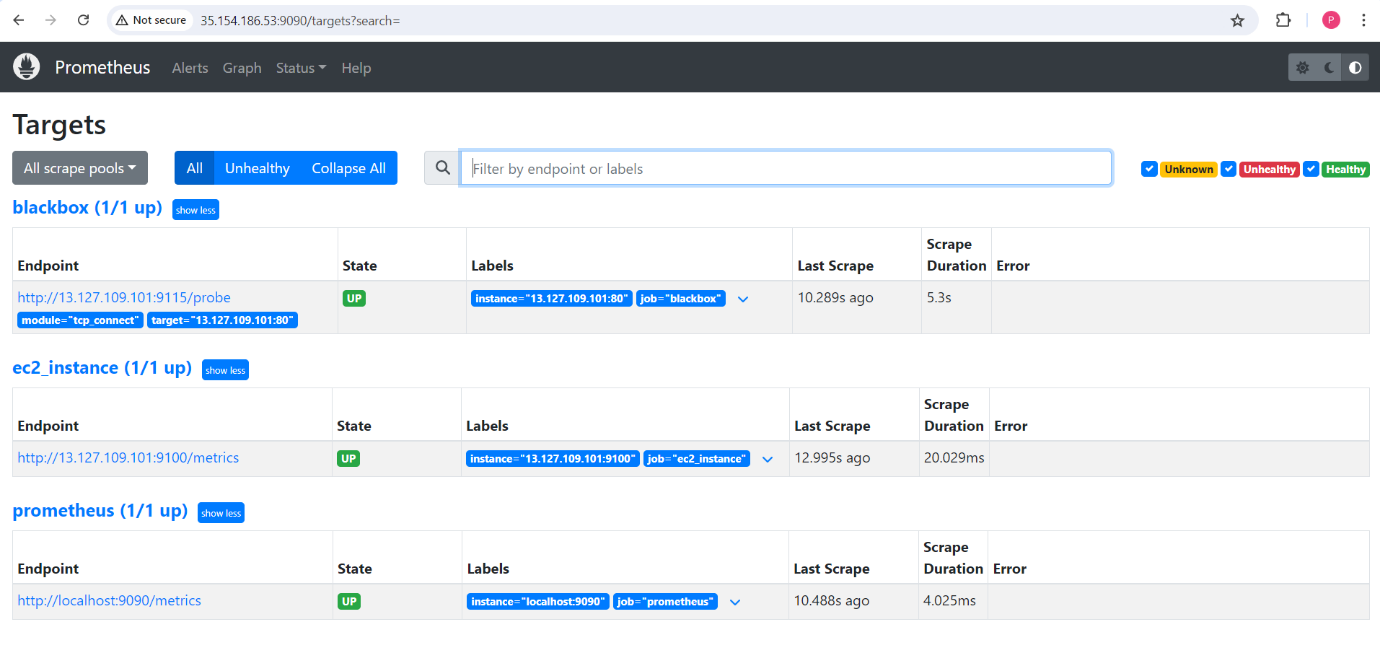
# Monitoring:

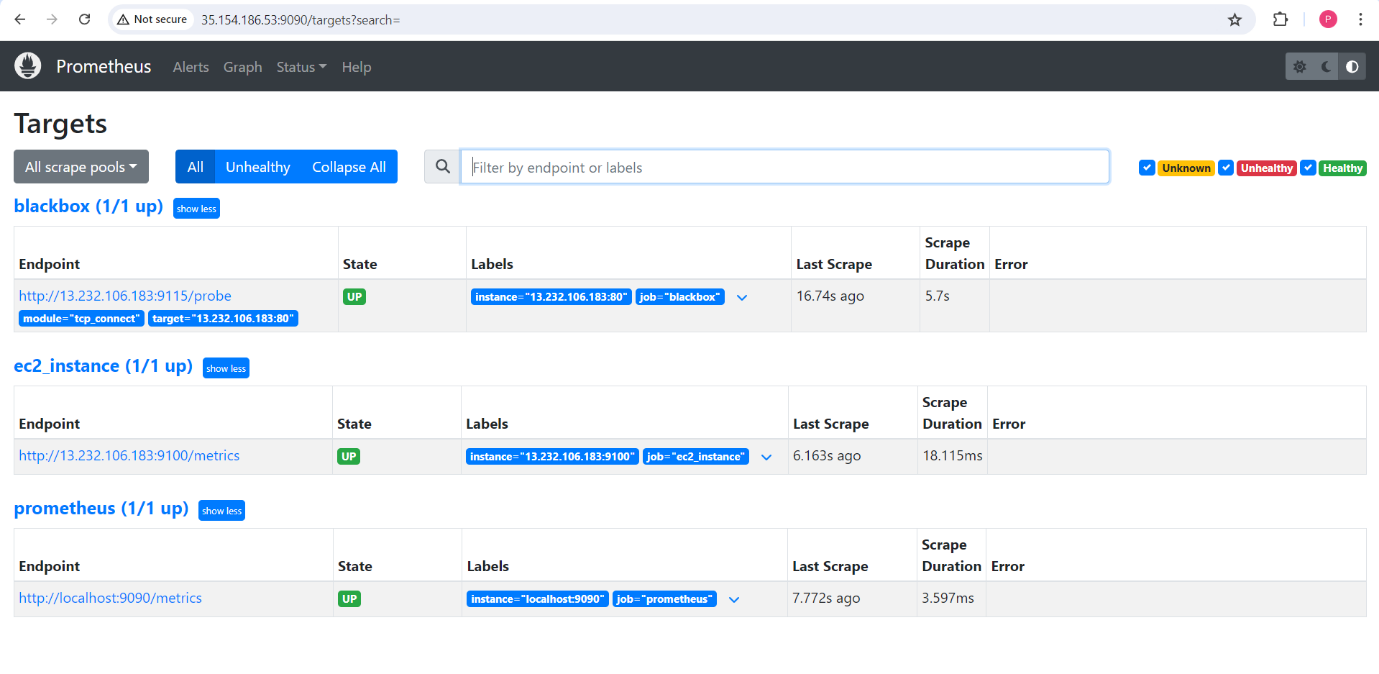
We have installed and configured Prometheus and AlertManager in Jenkins Server, we everytime install Blackbox exporter and nodexporter when new Ec2 instances gets created

Using ansible we are configuring Prometheus to pull configs wrt to remote Ec2 instances

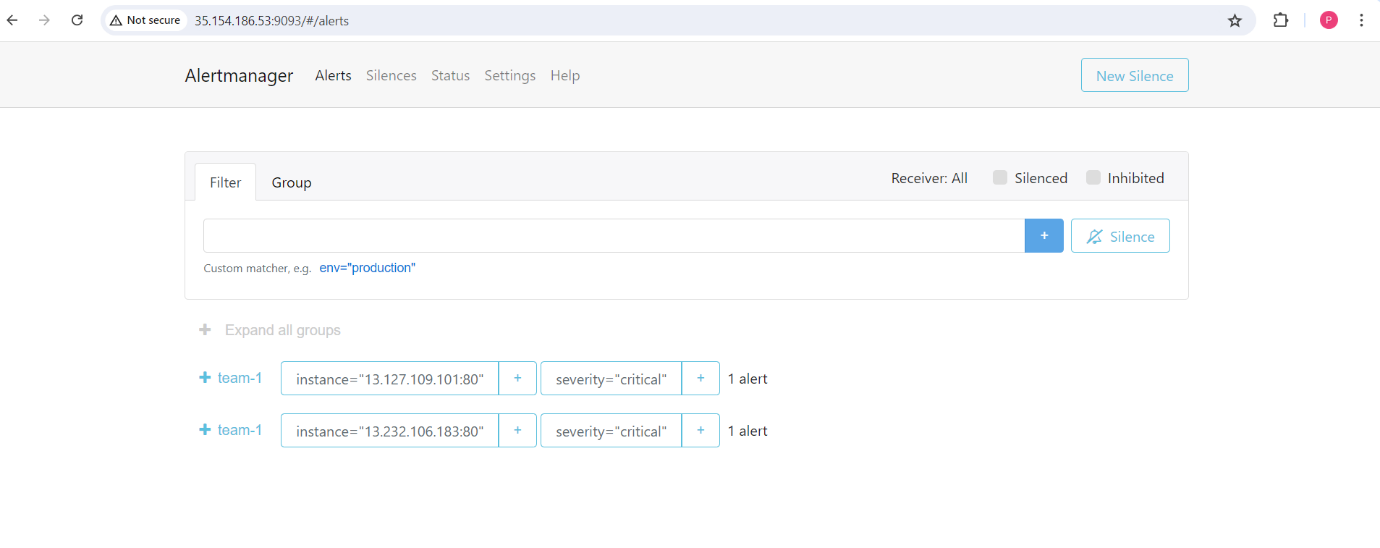
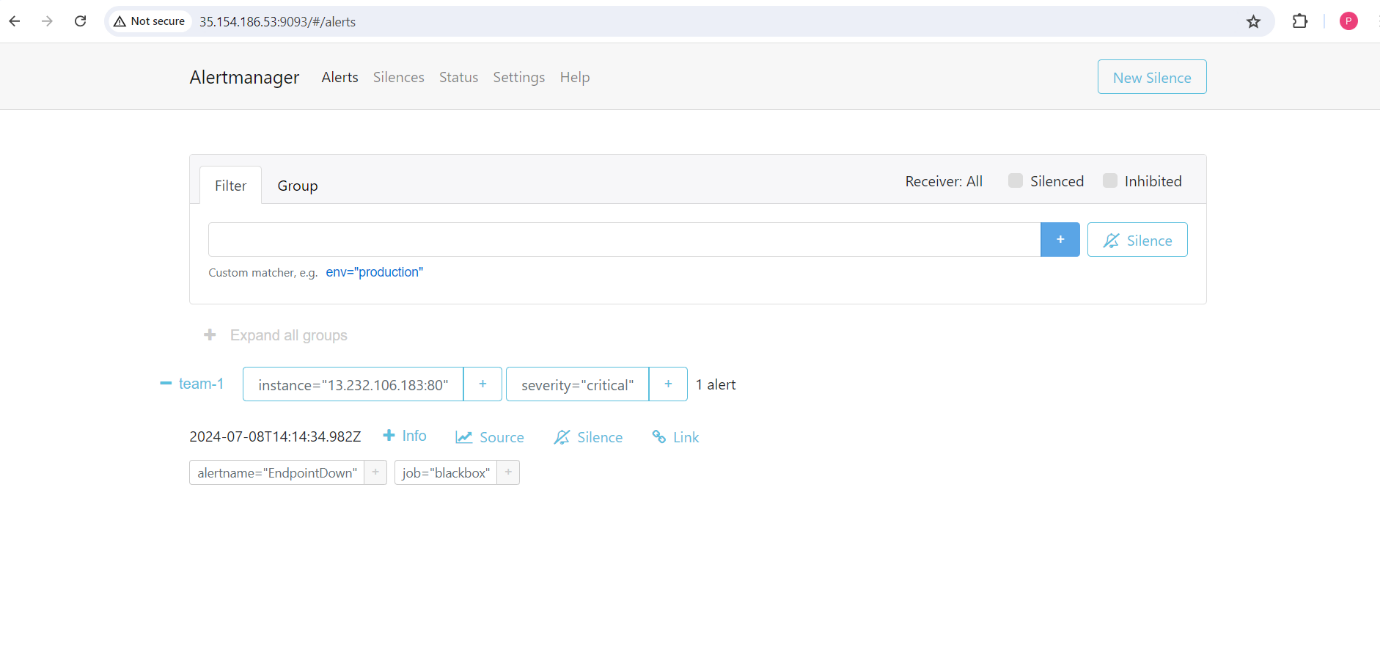
We have installed AlertManager and integrated slack for notifications

Dev & Prod environment on Prometheus :

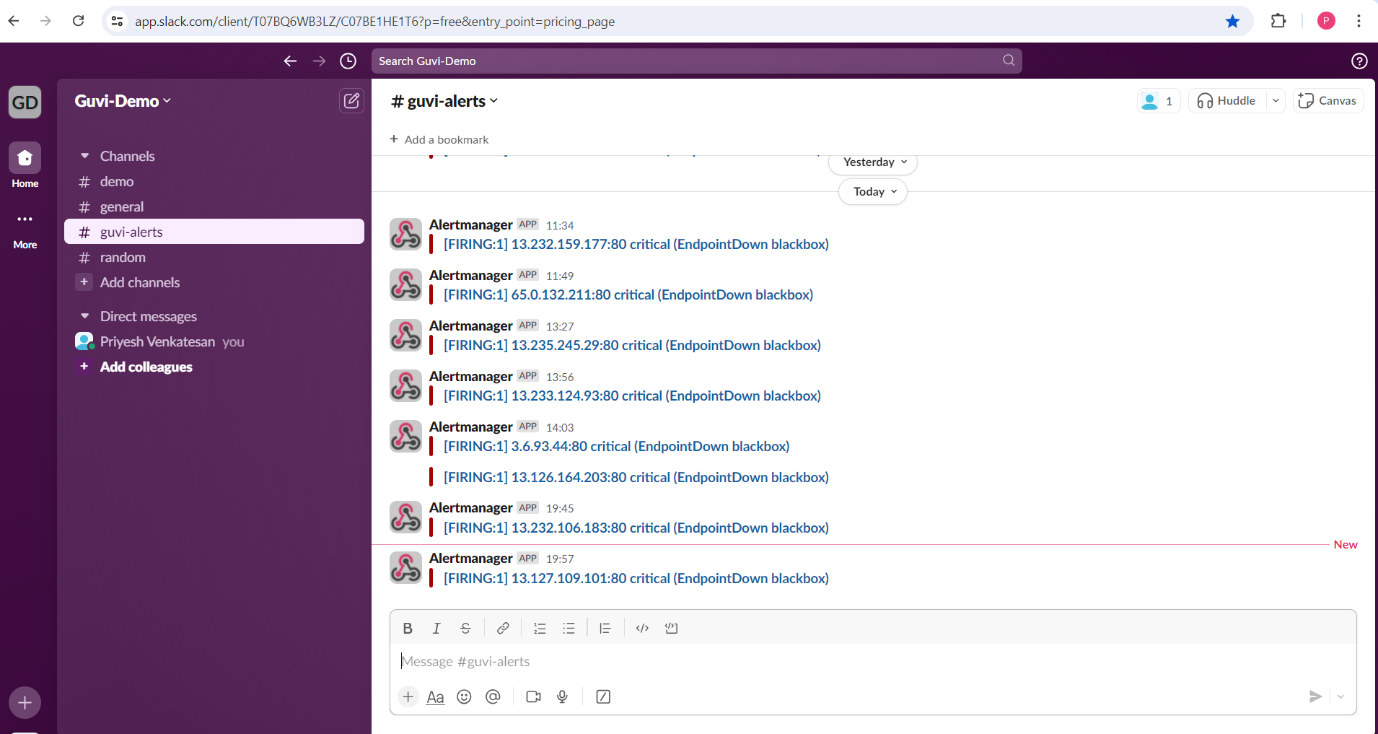
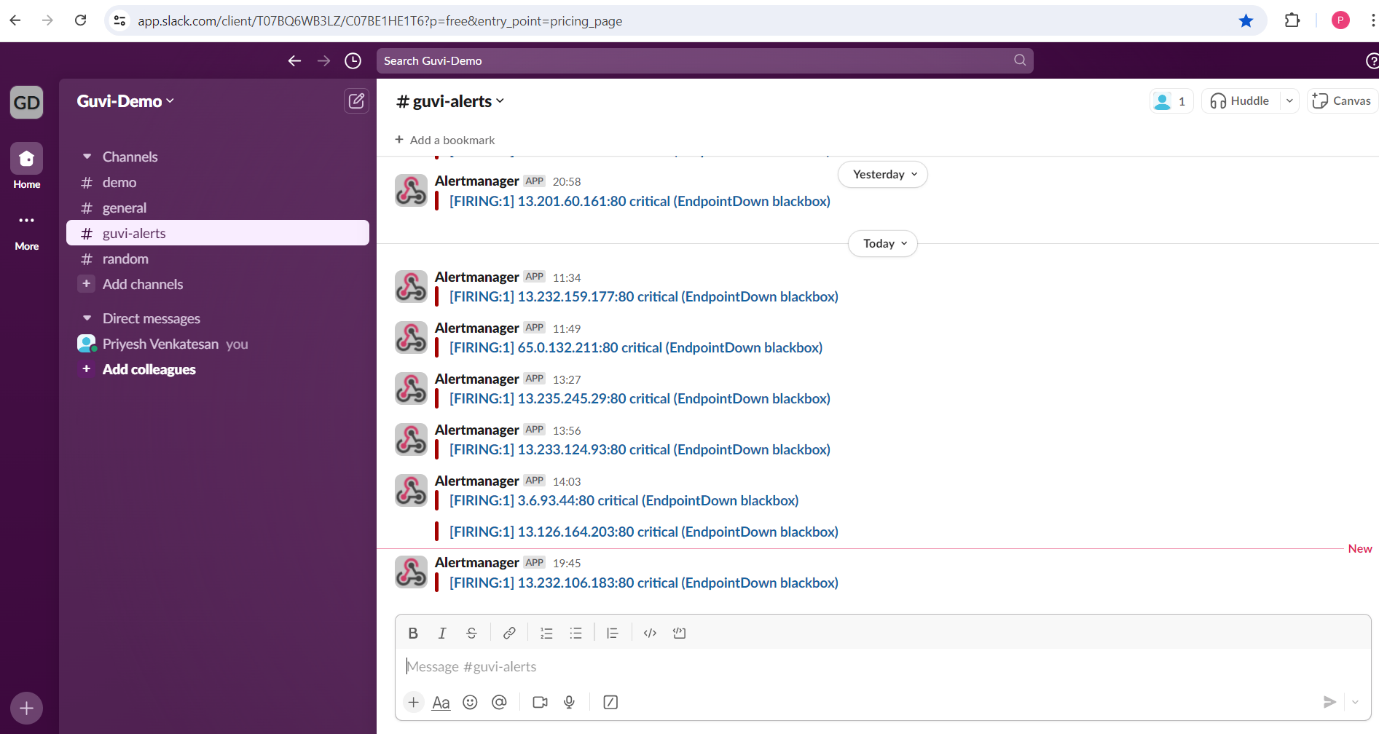




AlertManager for Dev & Prod:



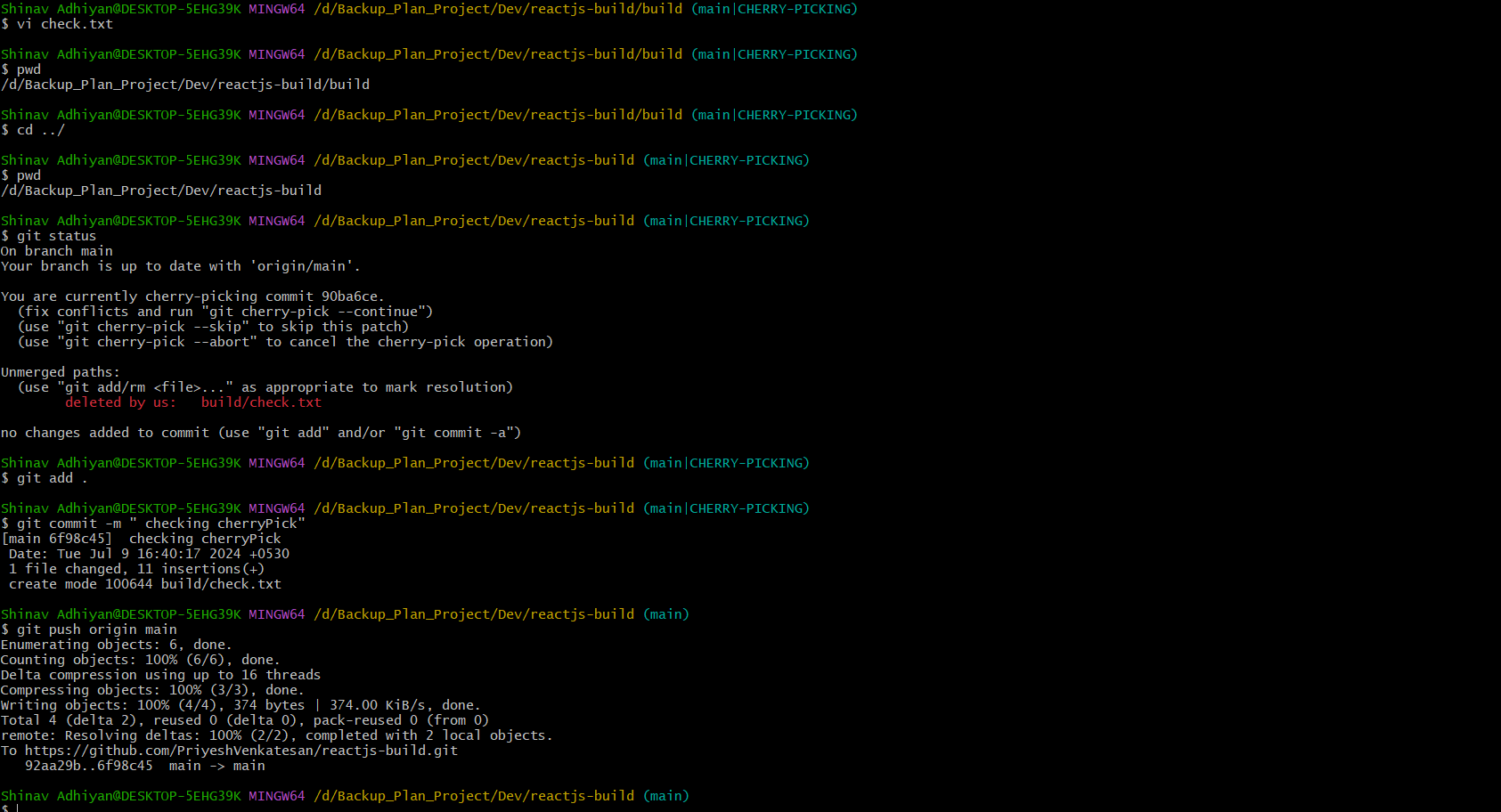
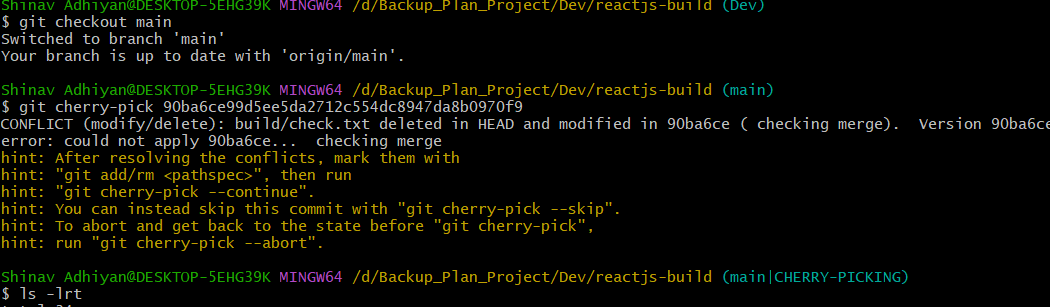
Slack Alerts in Dev & Prod:



# Work Flow:

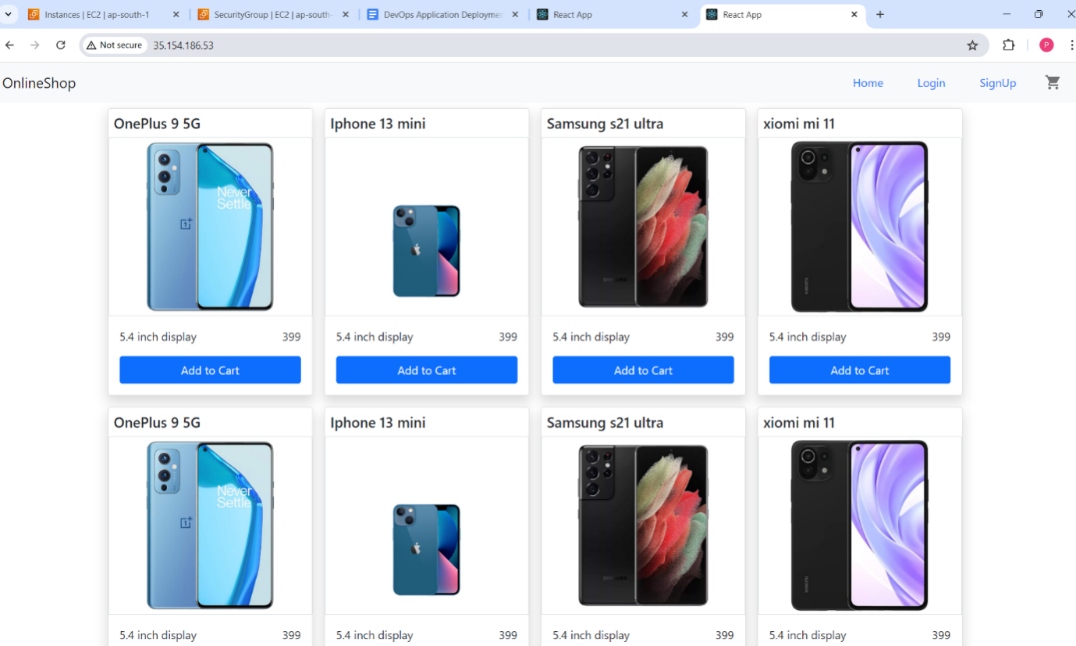
Whenever we add or push changes to Dev branch on remote repo and commit , webhook is triggered and Jenkins Pipeline is called , which builds ,push(public repo) and deploy the site on local and Dev Server (using terraform) and then installs exporters on Dev Server and push configs to Prometheus

As per this project, we have to merge only build directory of Dev branch to main branch using cherrypick , which inturn triggers Jenkins file to build ,push(private repo),and installs and deploy on Prod Servers

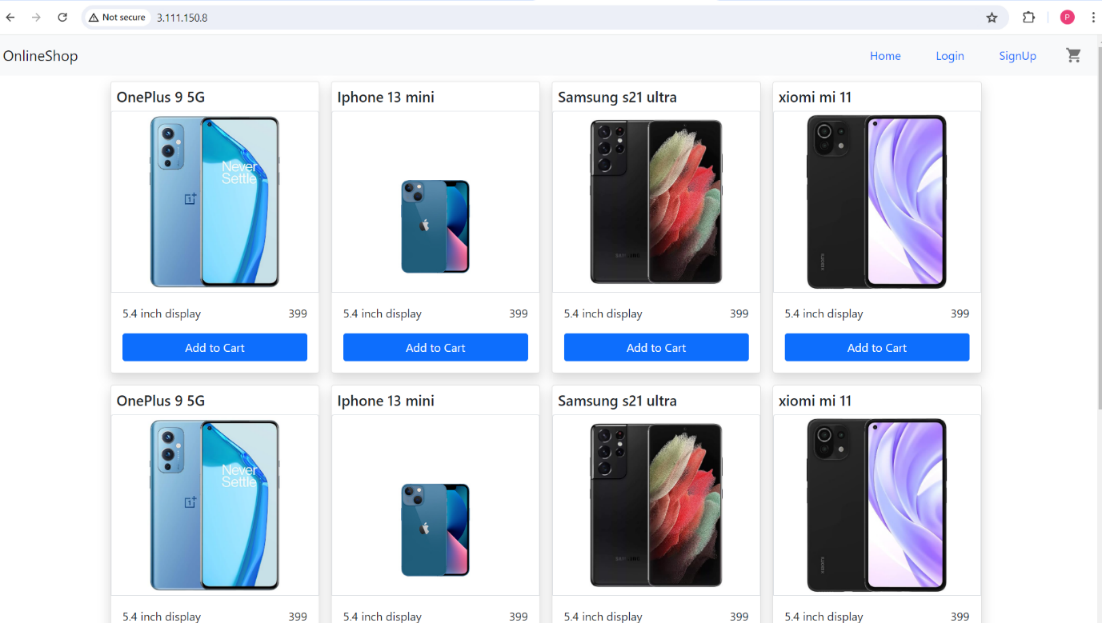


# Deployed Site :

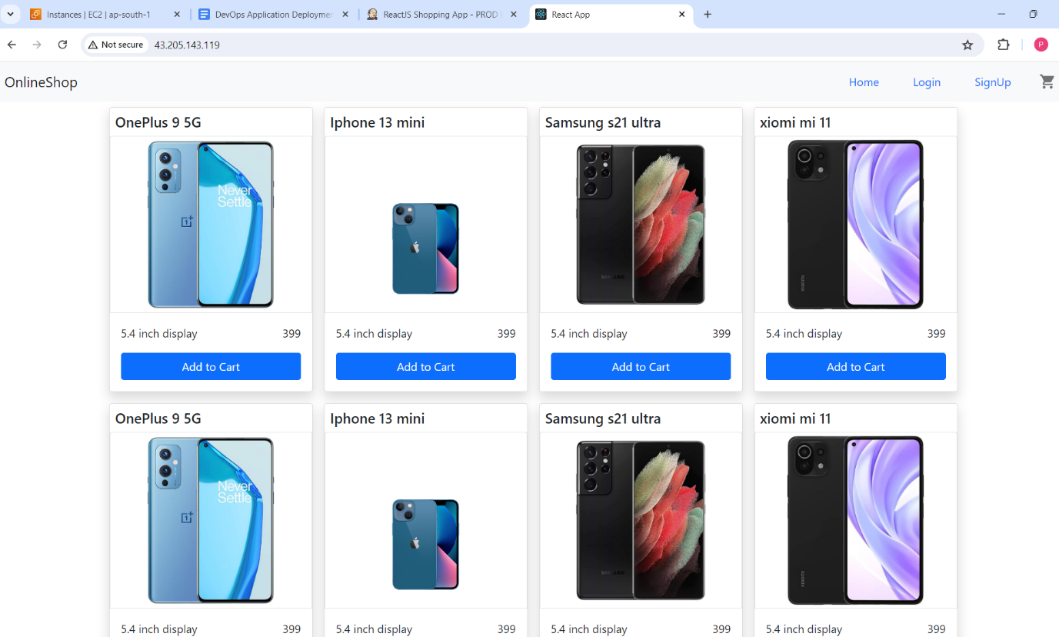
##### Local:



##### Dev:



##### Prod:



# Challenges Faced:

1) Pulling image from Private Repo which required docker login

2) Ansible Dynamic Inventory

3) Configuring Alert Managers

4) Tried and Failed installing nagios for monitoring

5) Pushing and tagging latest in both public and private repo

6) Configuring UserData in EC2

# Limitations:

* AlertManager not configured properly
* Usage of dockerignore not feasible
* This Project support partial merge (build directory) of Dev branch into main branch