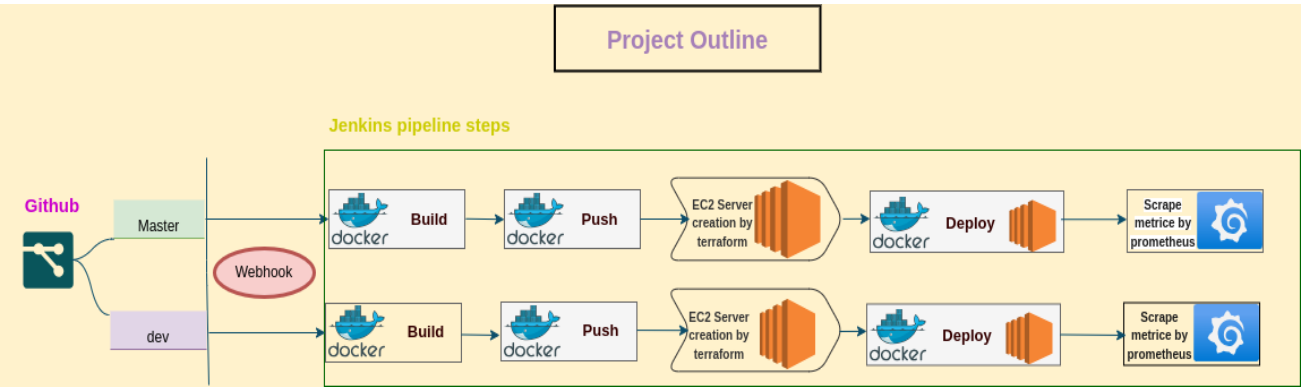
DEVOPS- CAPESTONE - PROJECT REACT APPLICATION DEPLOYMENT

Objective

Automate the SDLC of the given react application using the DevOps tools like GitHub, Docker, Jenkins and Terraform and monitor the Application using grafana after deployment.

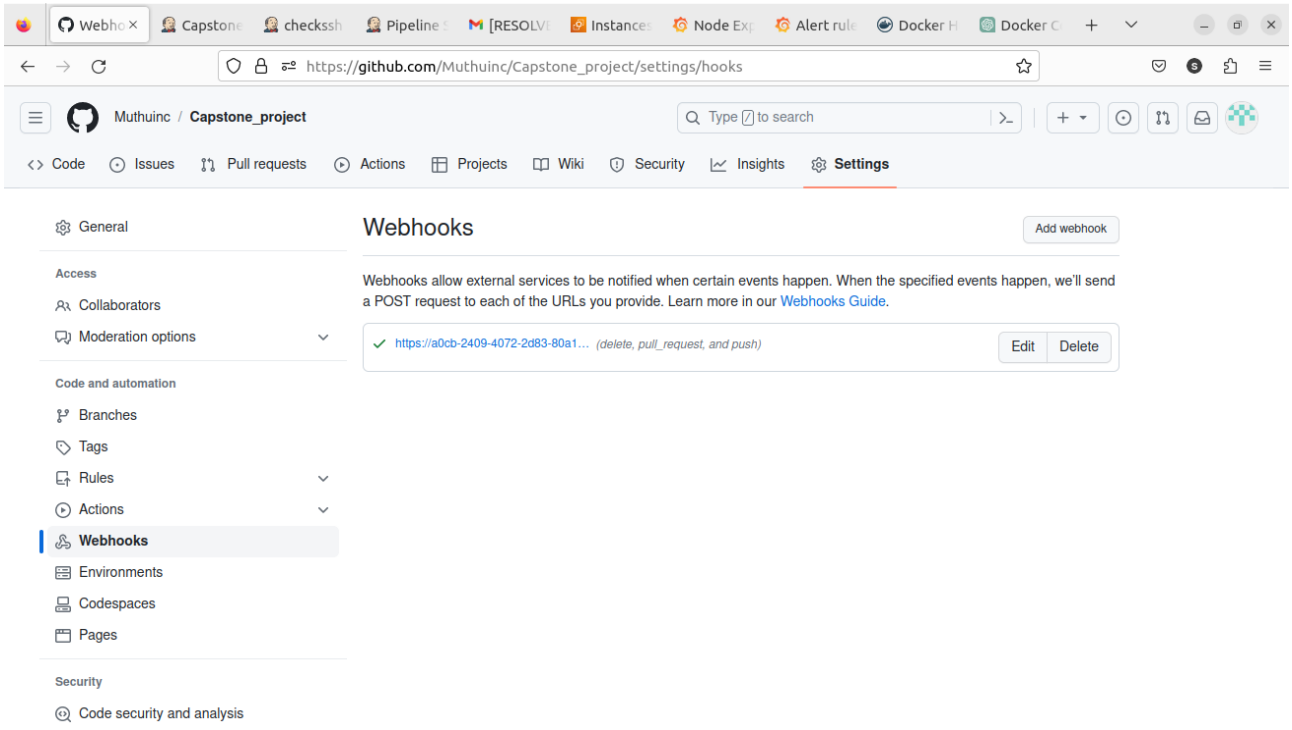
Github link

GitHub repository of the project: https://github.com/Muthuinc/Capstone_project.git



Local repository

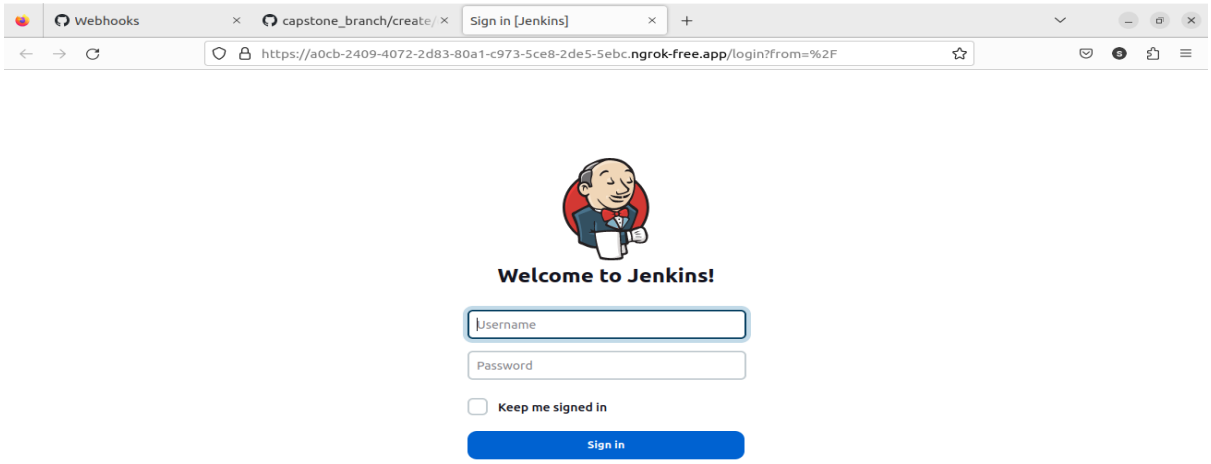
- Application code has been downloaded from the provided github repo to the local machine.
- A new directory created for the project.
- Script files for all the steps like push deploy create were made for each branch and then push to the remote repository.
- Github webhook for **push, branch delete and pull request** set-up were made. it triggers the Jenkins pipeline step automatically.



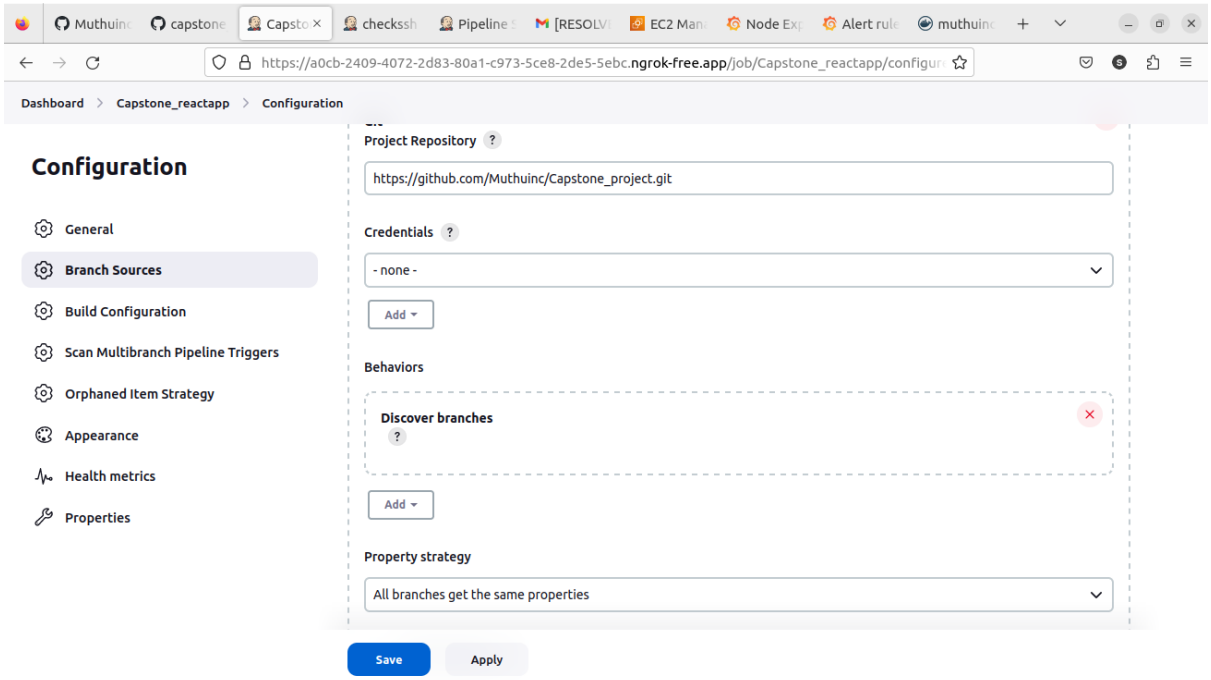
Jenkins

- I installed Jenkins locally in my Ubuntu machine using the official documentation.
- Created a tunnel using **ngrok** to make it accessible for web hook actions.
- New multi-branch pipeline project Capestone_reactapp created.
- Configurations like adding the remote repository and web hook triggers were done.

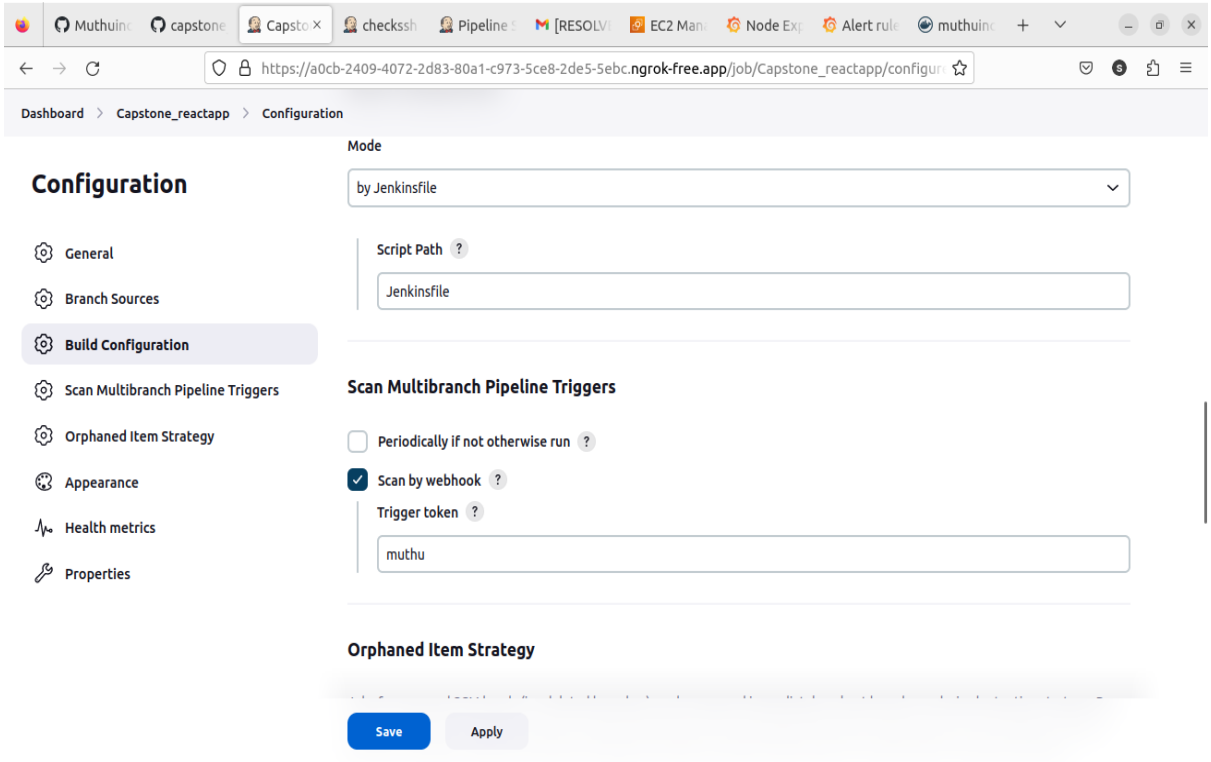
JENKINS LOGIN PAGE



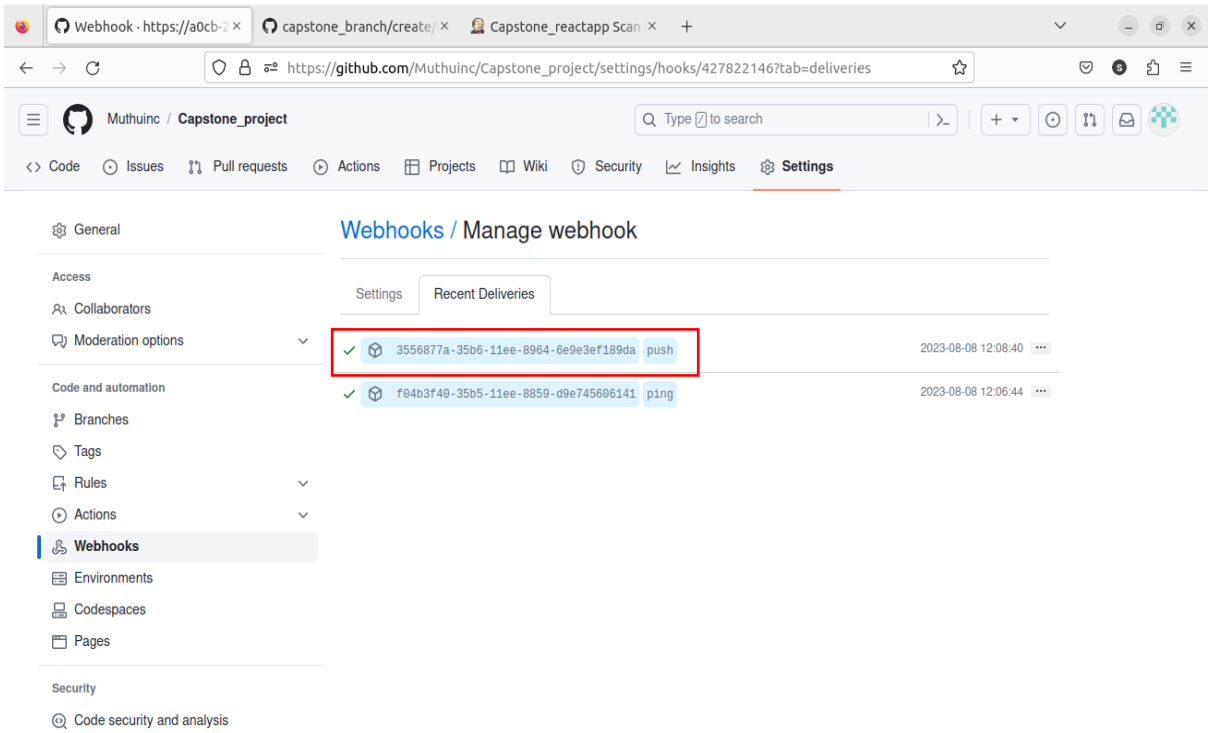
CONFIGURATION PAGE



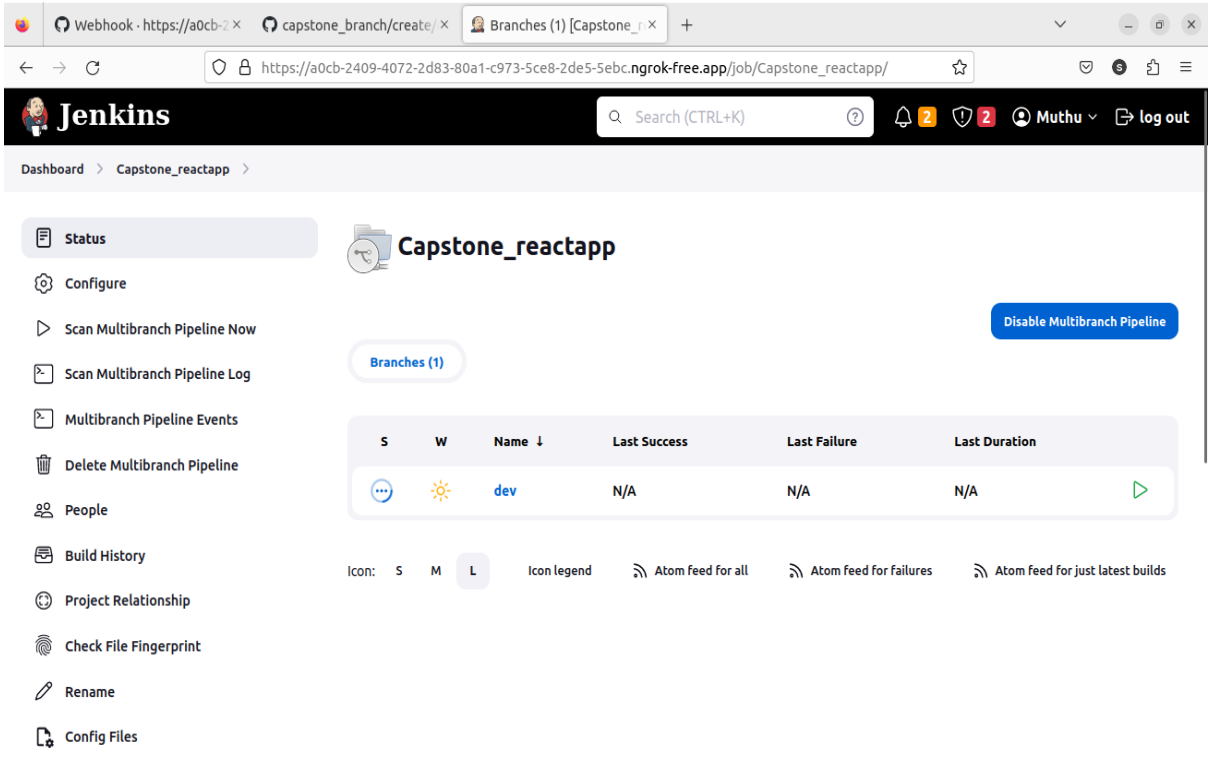
WEBHOOK TRIGGER -CONFIGURATION



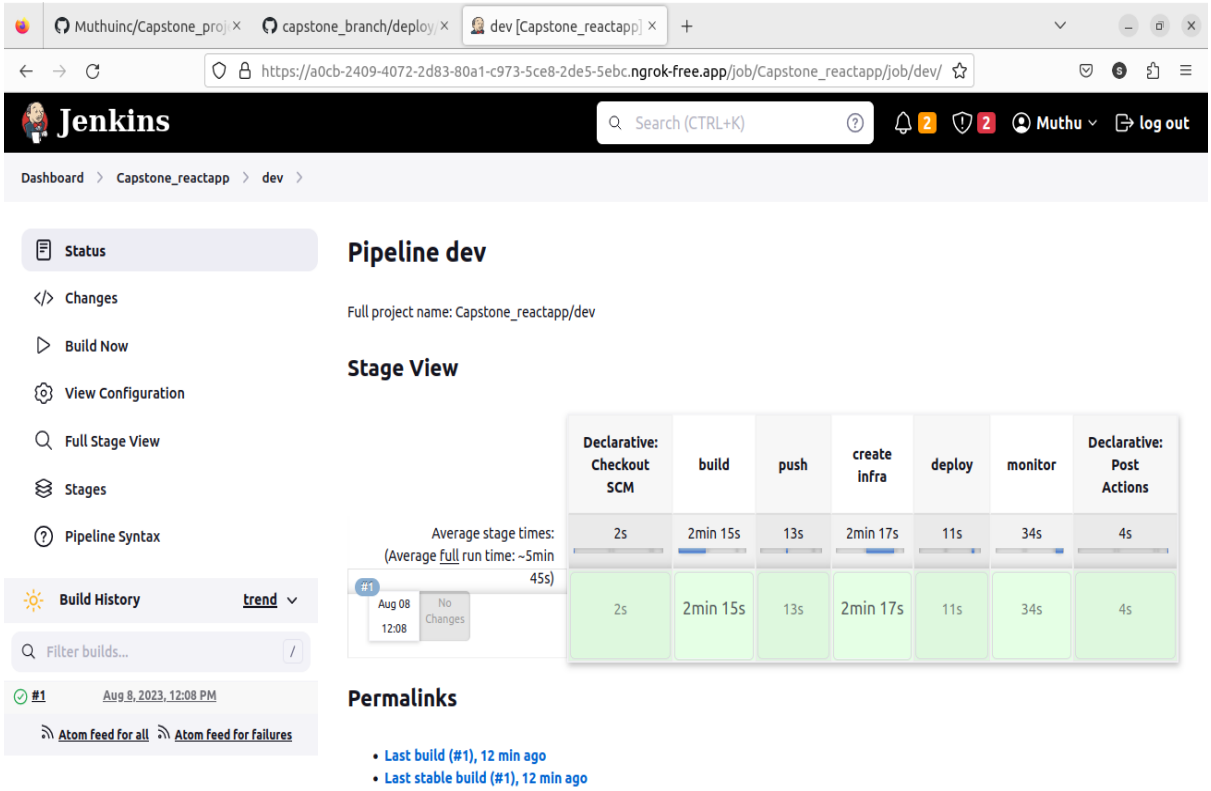
GITHUB SENT TRIGGER TO JENKINS AS SOON AS DEV BRANCH PUSHES THE CODE



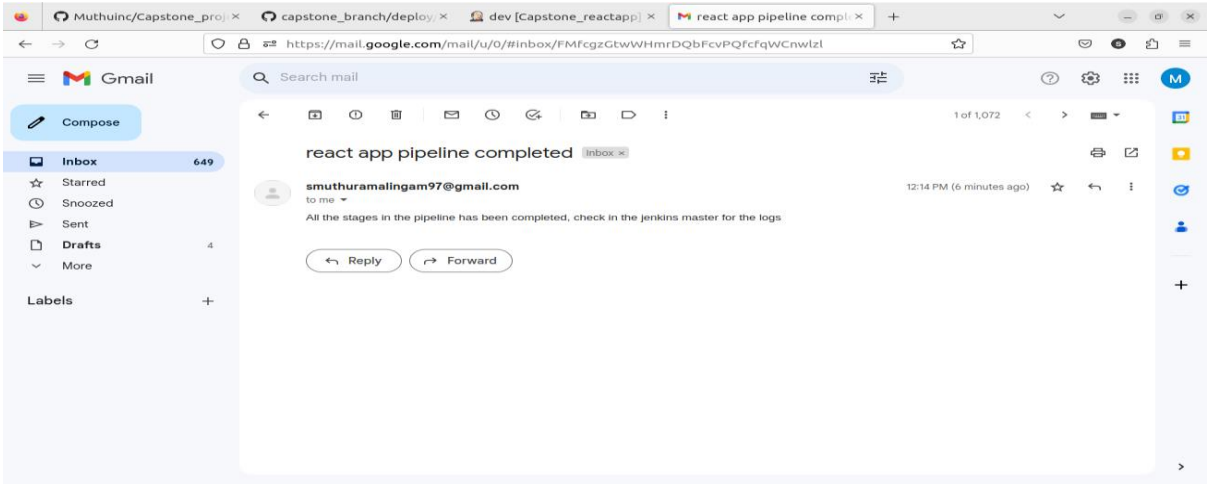
DEV-BRANCH STARTED TO BUILD



DEV-BRANCH SUCCESSFULLY BUILT

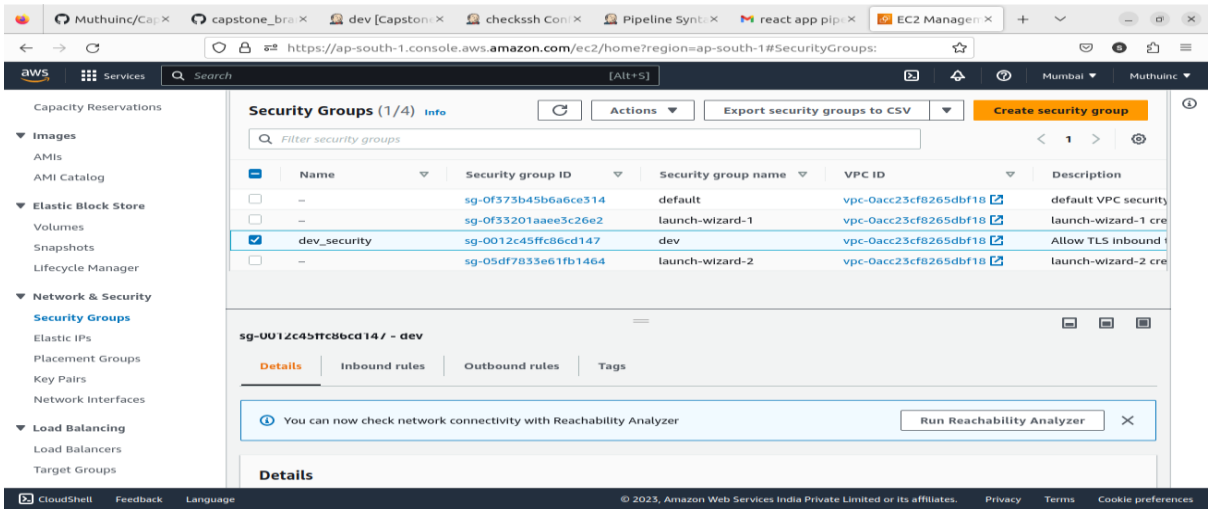


COMPLETION MAIL RECEIVED AS A RESULT OF POST BUILD ACTION

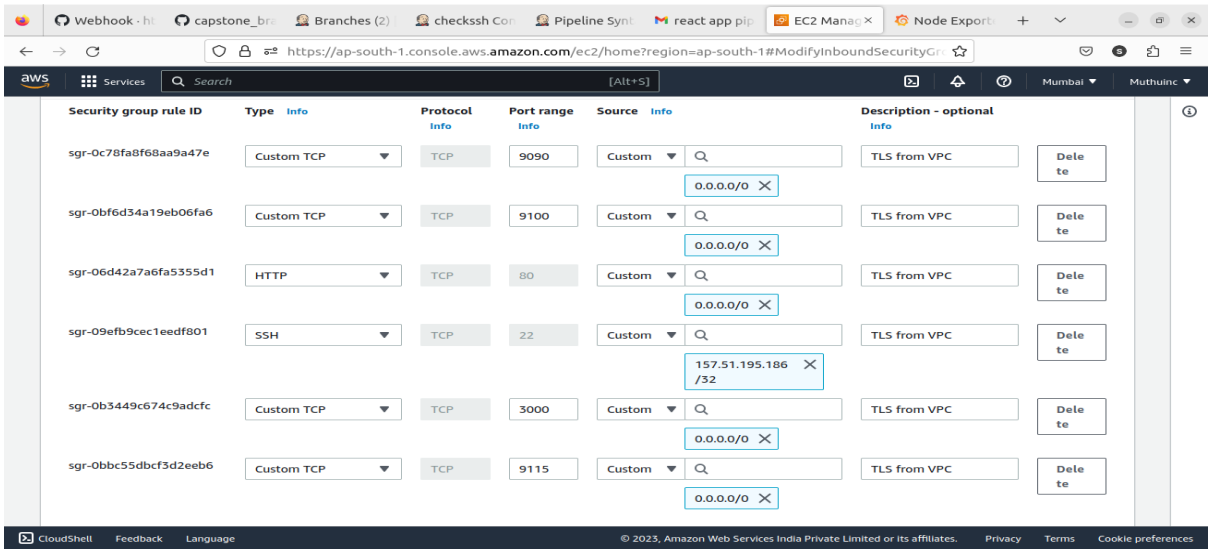


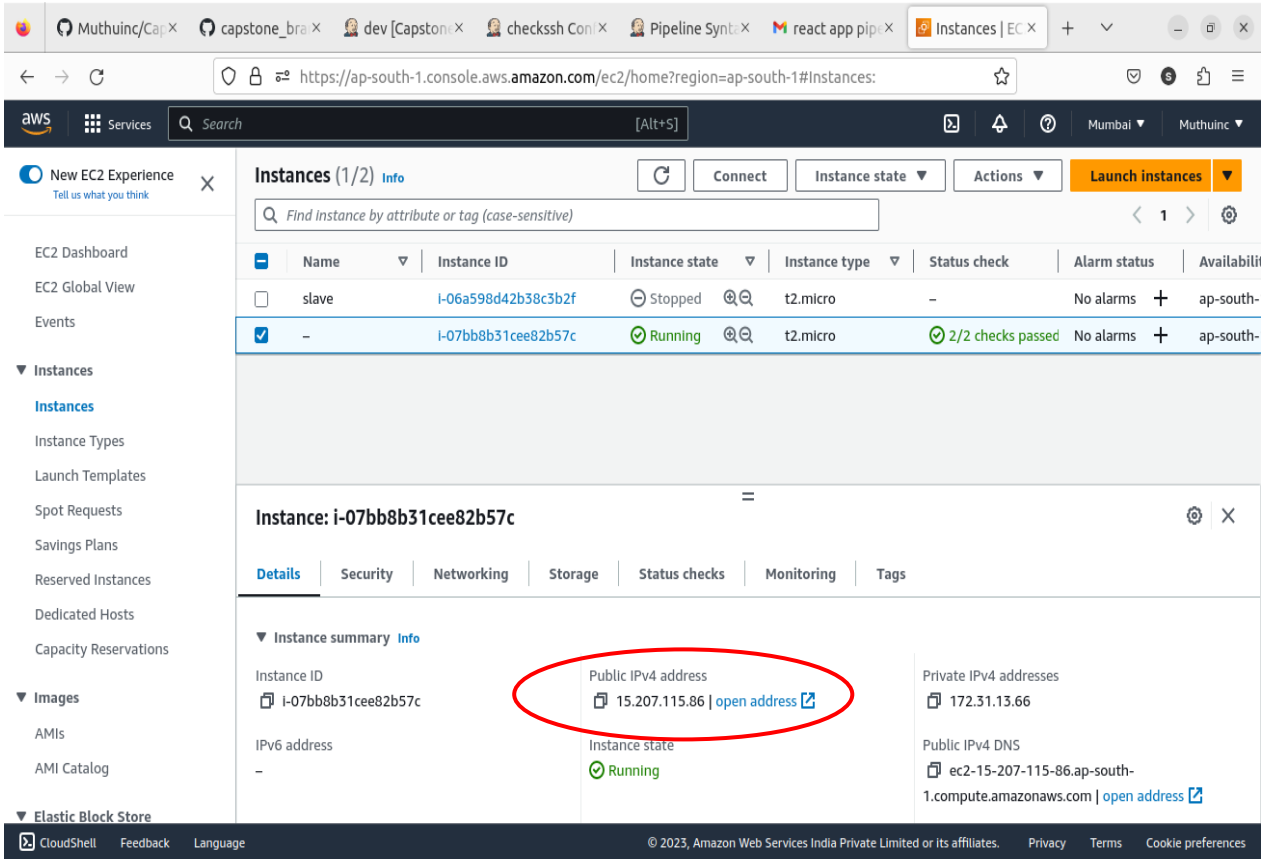
IN THE PIPELINE, TERRAFORM IS USED TO CREATE INFRASTRUCTURE FOR THE APP IN AWS

EC2 INSTANCE AND SECURITY GROUP CREATED – DEV BRANCH



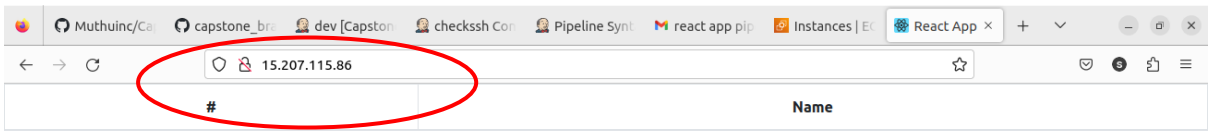
THESE ARE FOR DEV BRANCH, HERE SSH CAN BE DONE ONLY BY MY IP. OTHERS PORT OPENING IS FOR MONITORING PURPOSES, GRAFANA – 3000, PROMETHEUS – 9090, NODE EXPORTER – 9100 ETC



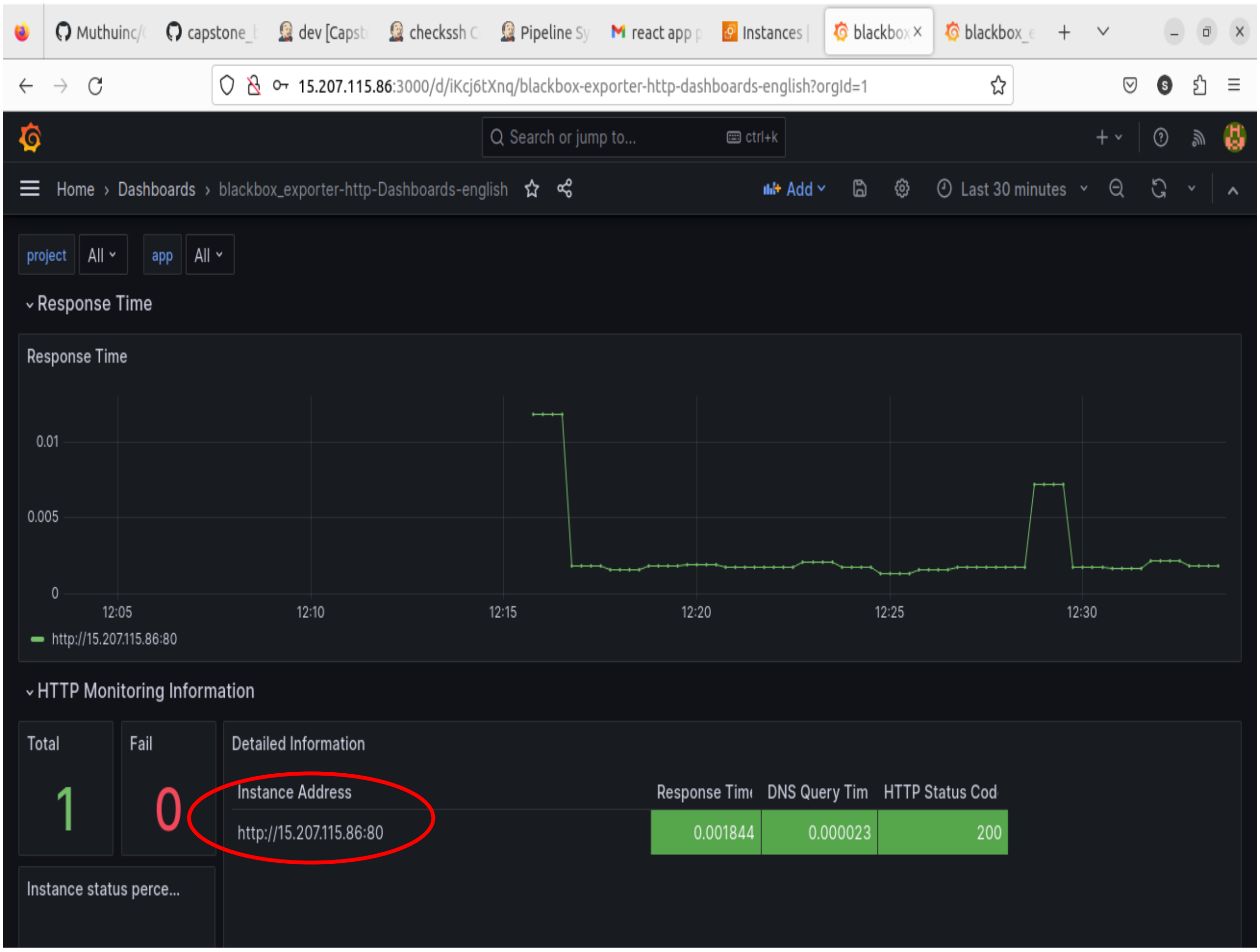


NOTE THE IP ADDRESS OF THE EC2 SERVER (FOR DEV BRANCH)

VIEW THE APPLICATION BELOW



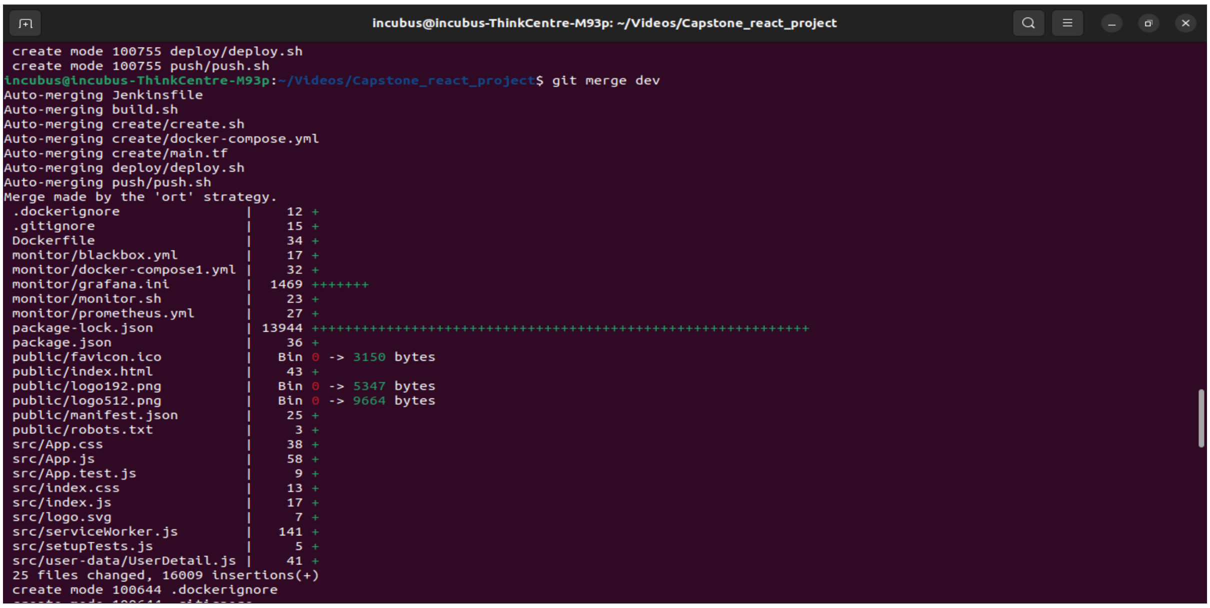
MONITORING BY GRAFANA – DEV SERVER



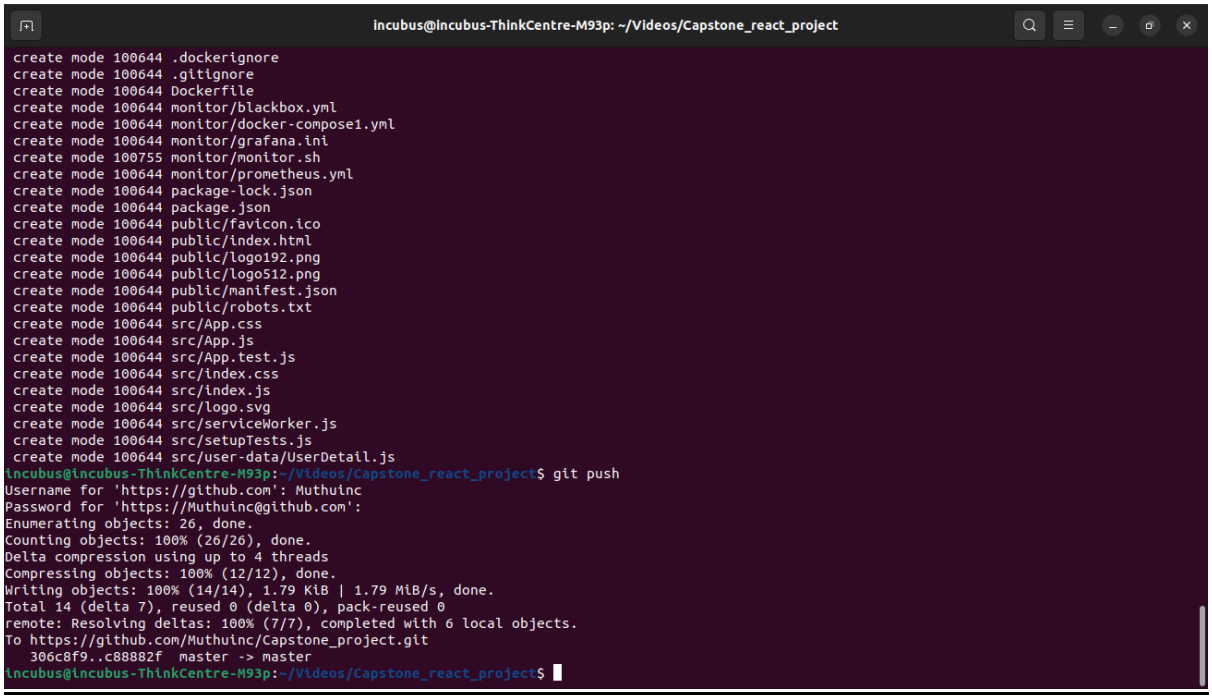
SET UP FOR THE DEV BRANCH IS OVER

WE VERIFIED THE APPLICATION IS RUNNING

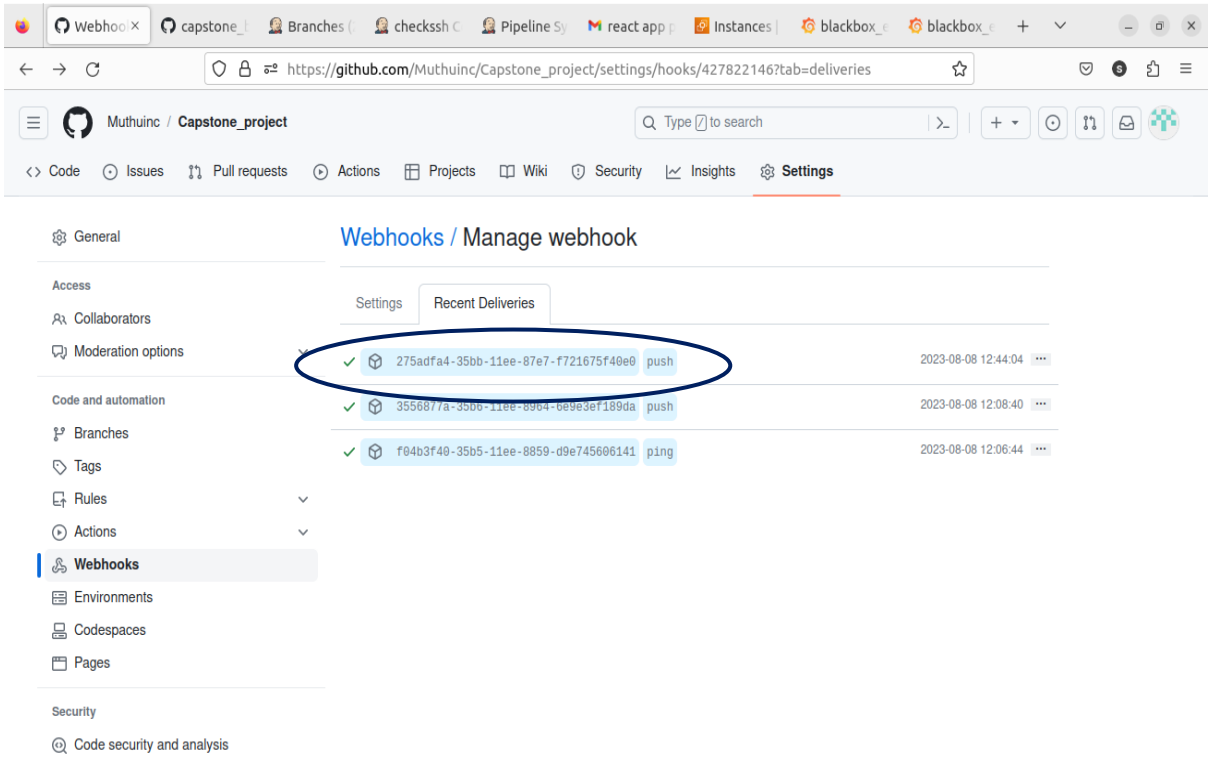
MERGING THE DEV BRANCH TO THE MASTER FOR PRODUCTION RELEASE



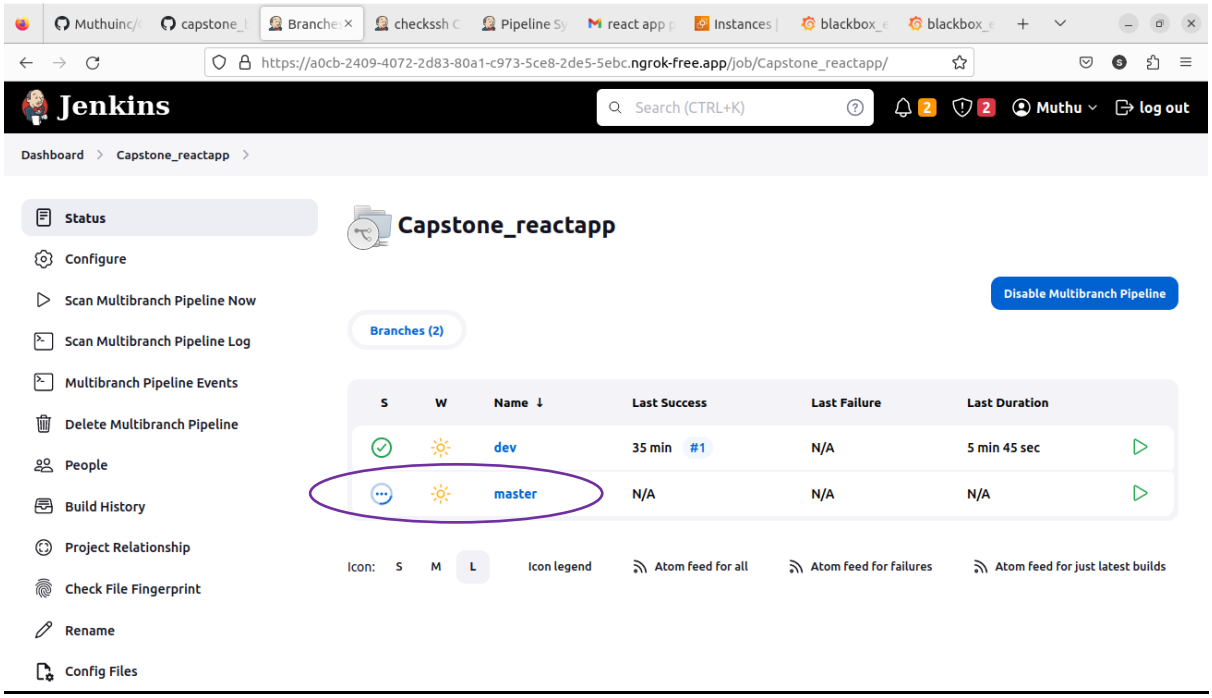
PUSHING THE CHANGES TO THE REMOTE REPO



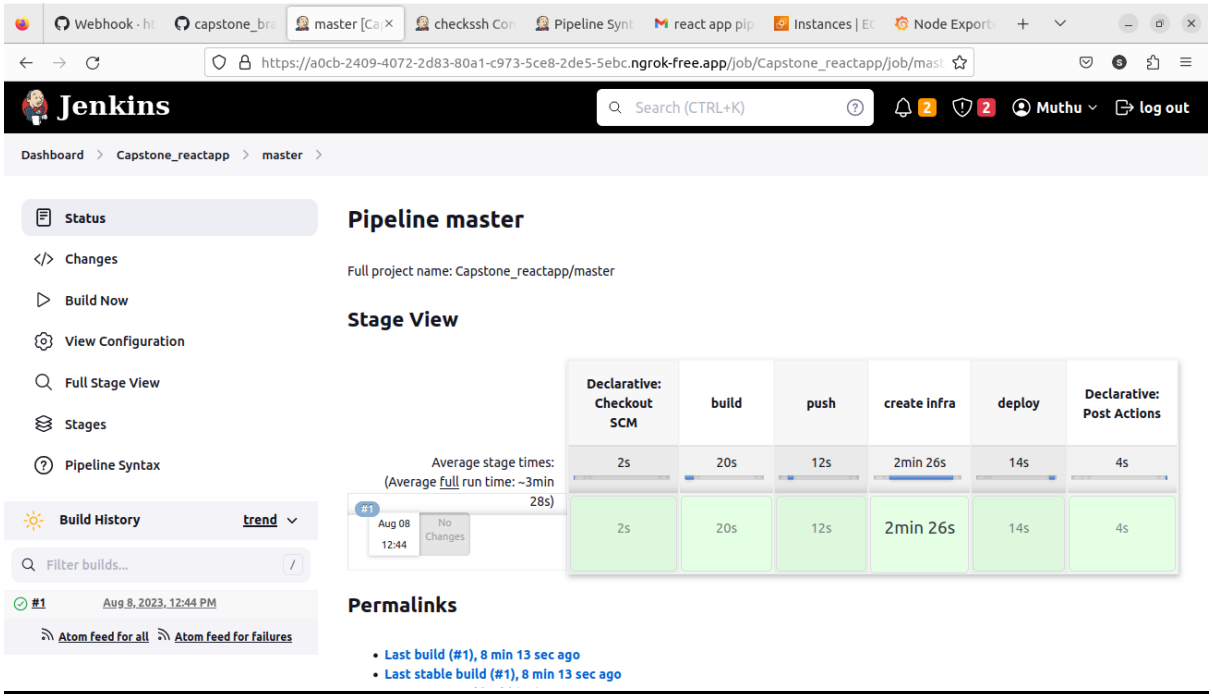
YOU CAN SEE THE WEBHOOK SENDS ANOTHER TRIGGER AS A RESULT OF MASTER BRANCH



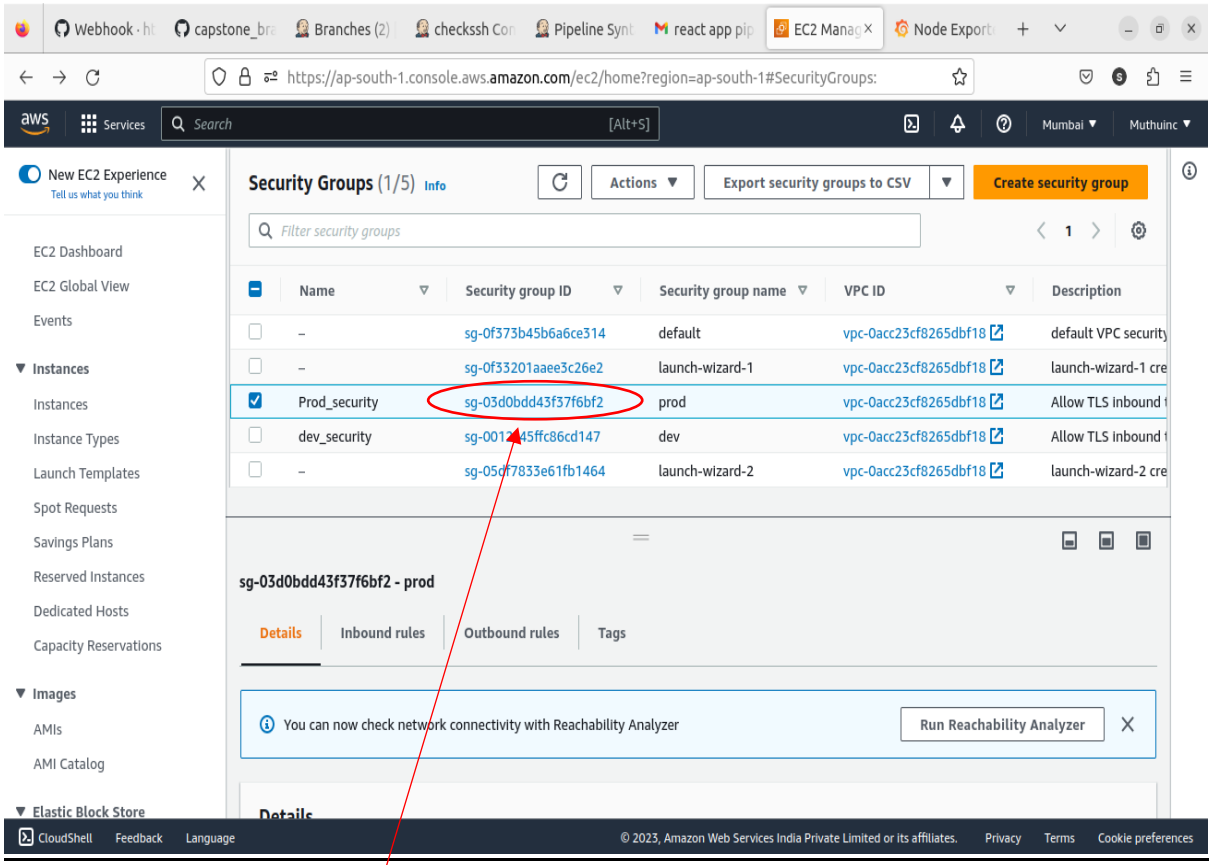
JENKINS SERVER DETECT THE MASTER BRANCH AUTOMATICALLY AND STARTED TO WORK OUT



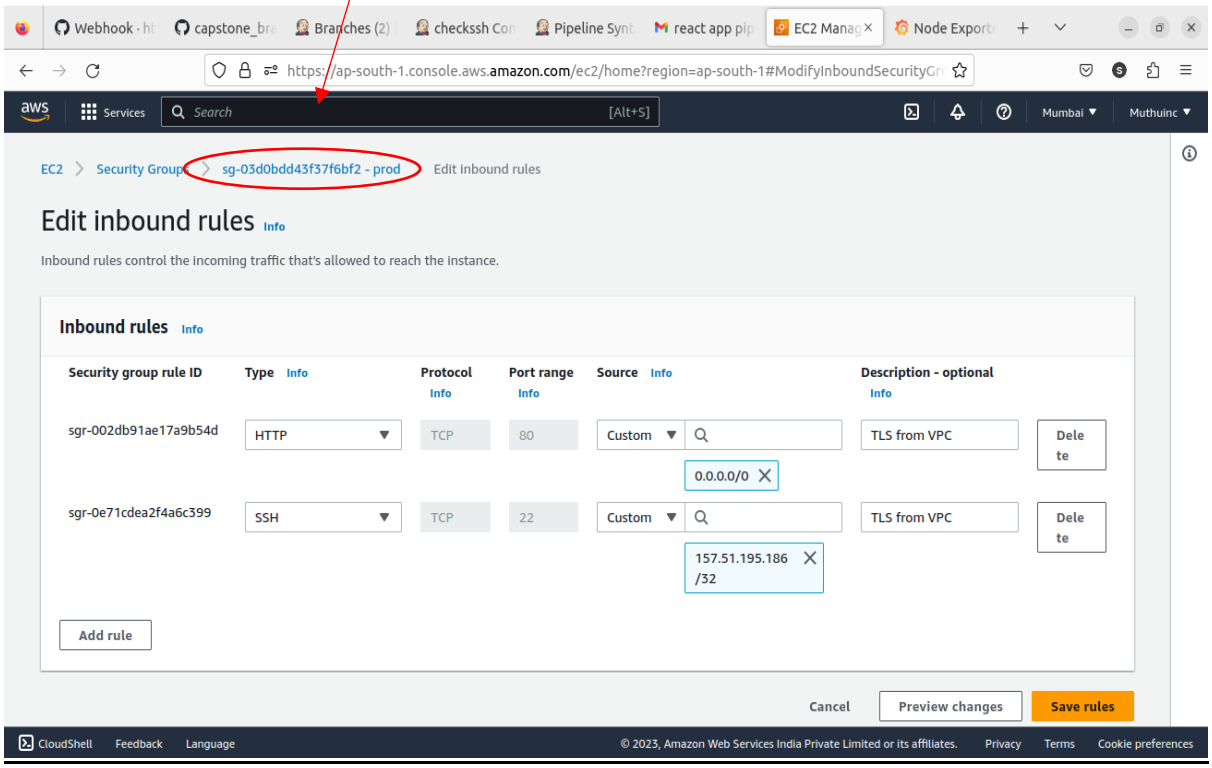
MASTER BRANCH PIPELINE STEPS SUCCESSFULLY EXECUTED



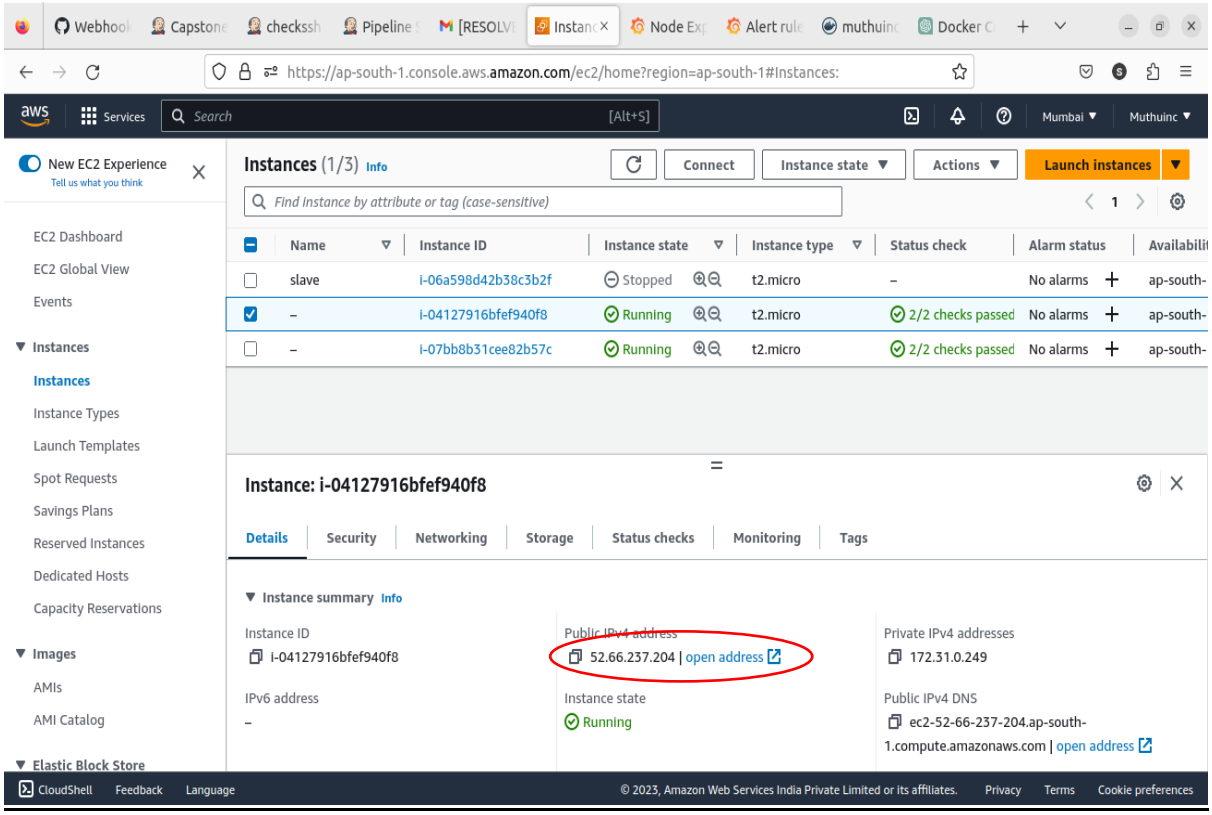
NEW SECURITY GROUP FOR THE PROD SERVER IS CREATED BY TERRAFORM



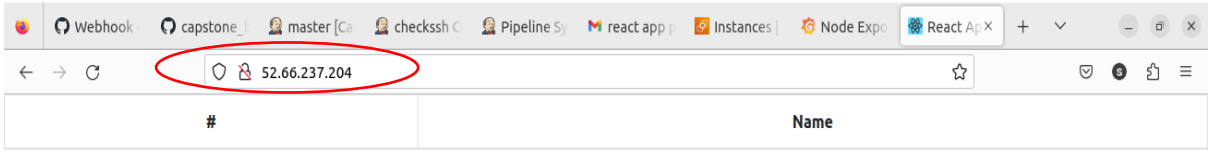
SSH CAN ONLY BE DONE FROM MY IP ADDRESS, APPLICATION CAN BE VIEW FROM ANYWHERE,



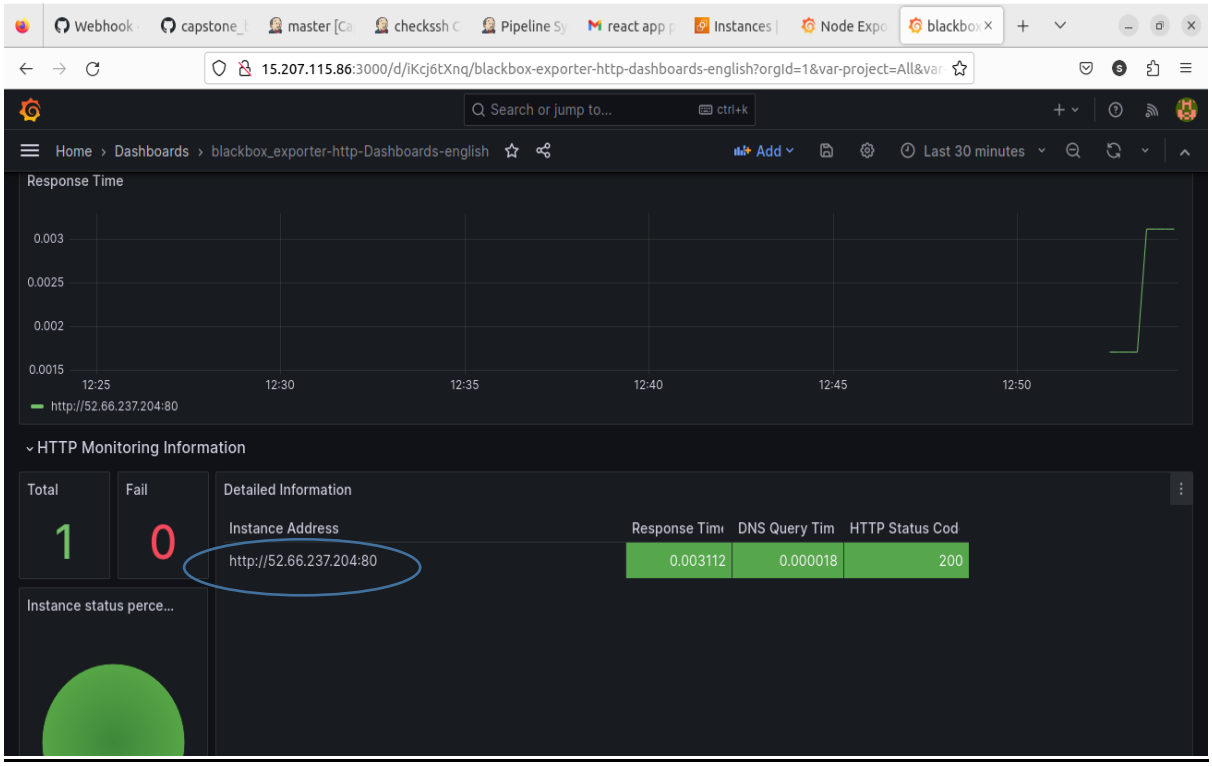
EC2 SERVER FOR THE PROD



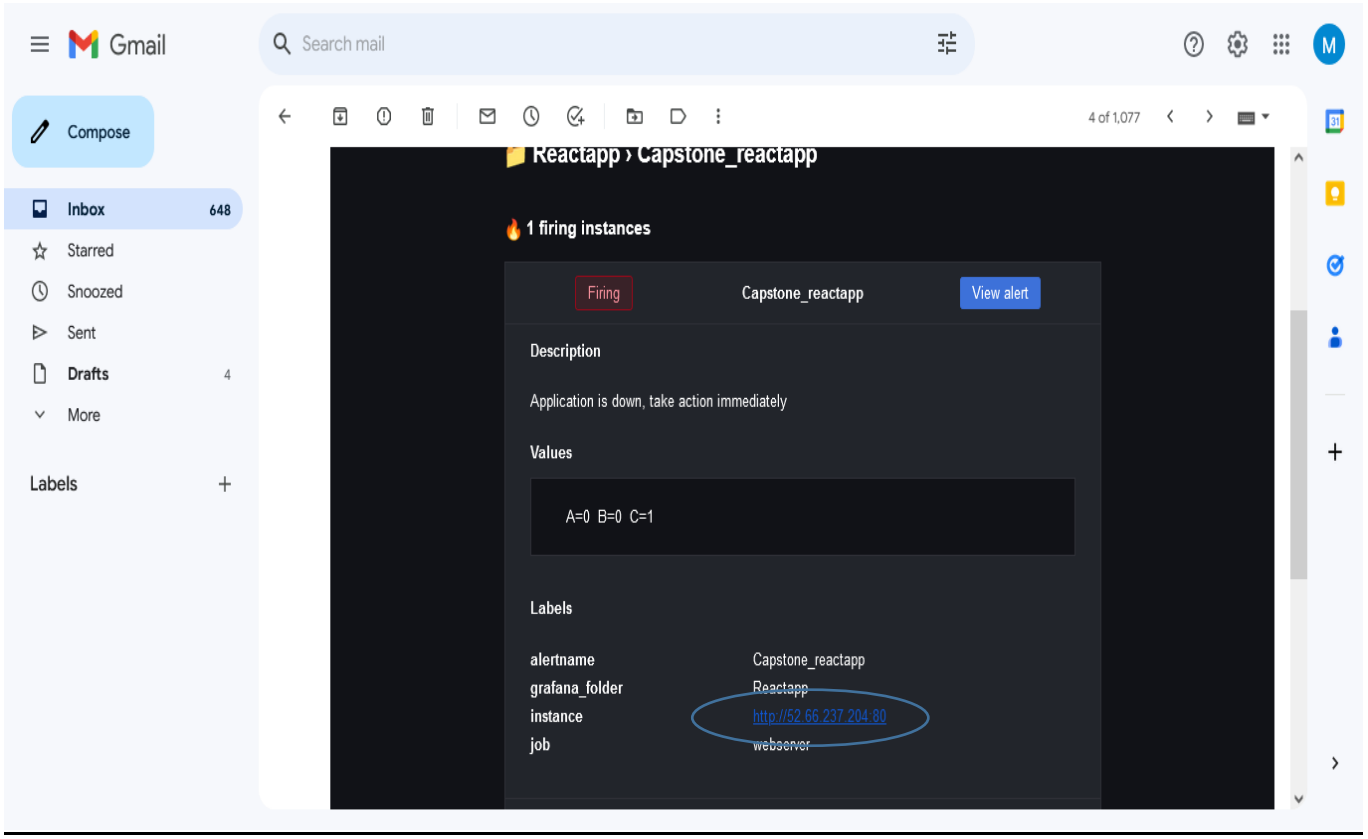
APPLICATION IS RUNNING SUCCESSFULLY - PROD ENVIRONMENT



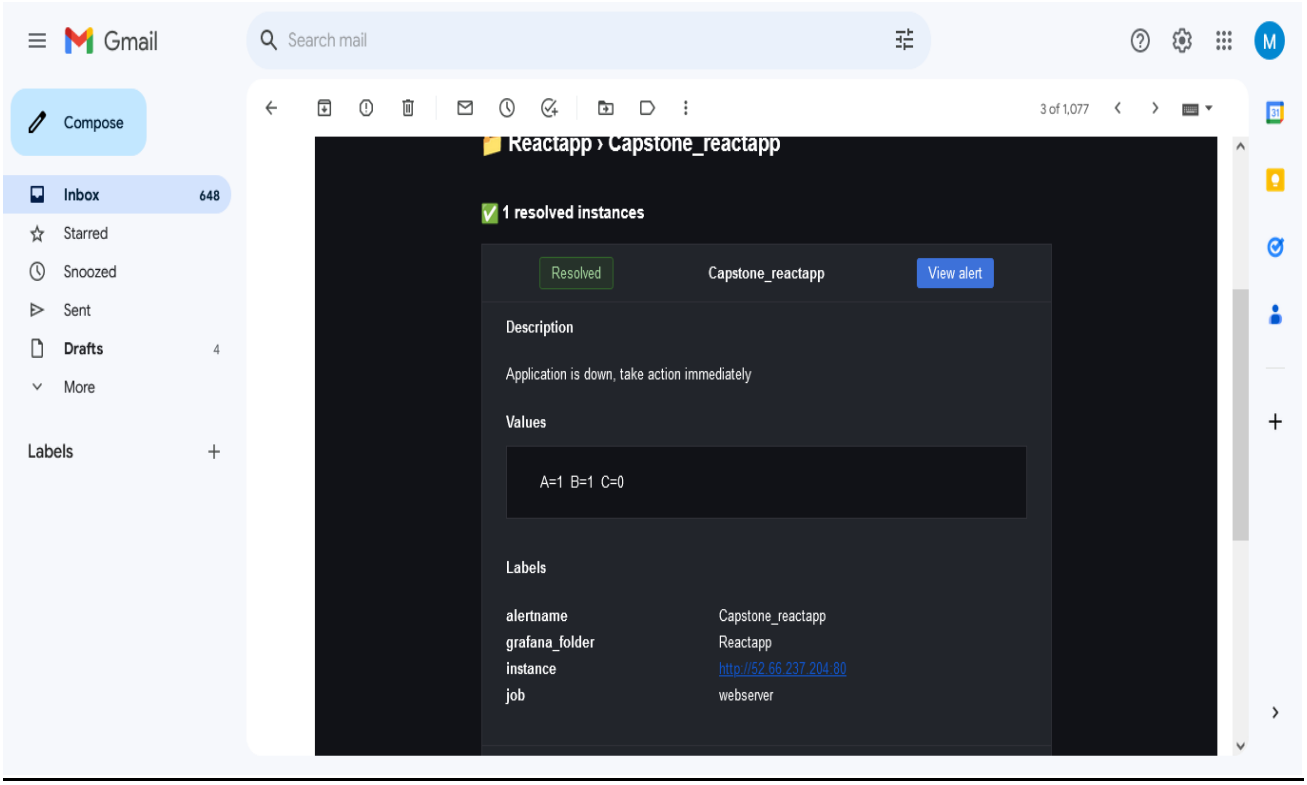
GRAFANA MONITORING IS SET UP



WE RESTRICT THE PORT 80 OF THE INSTANCE NOT TO BE ACCESSED FROM OUTSIDE – SO WE CREATE THE APPLICATION GOES DOWN SITUATION. AS A RESULT GRAFANA SENT ALERT IN THE EMAIL LOOK BELOW CHECK THE IP ADDRESS

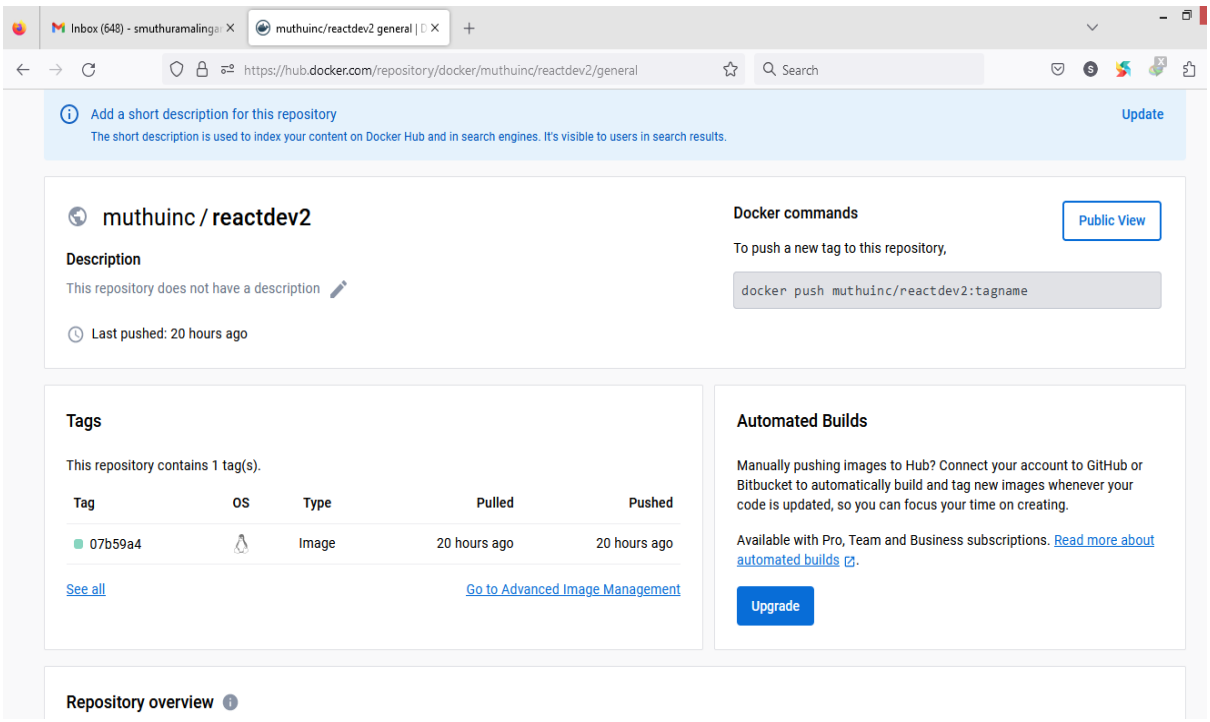


AFTER I OPENED THE PORT 80 IT SENT OUT THE RESOLVED MESSAGE



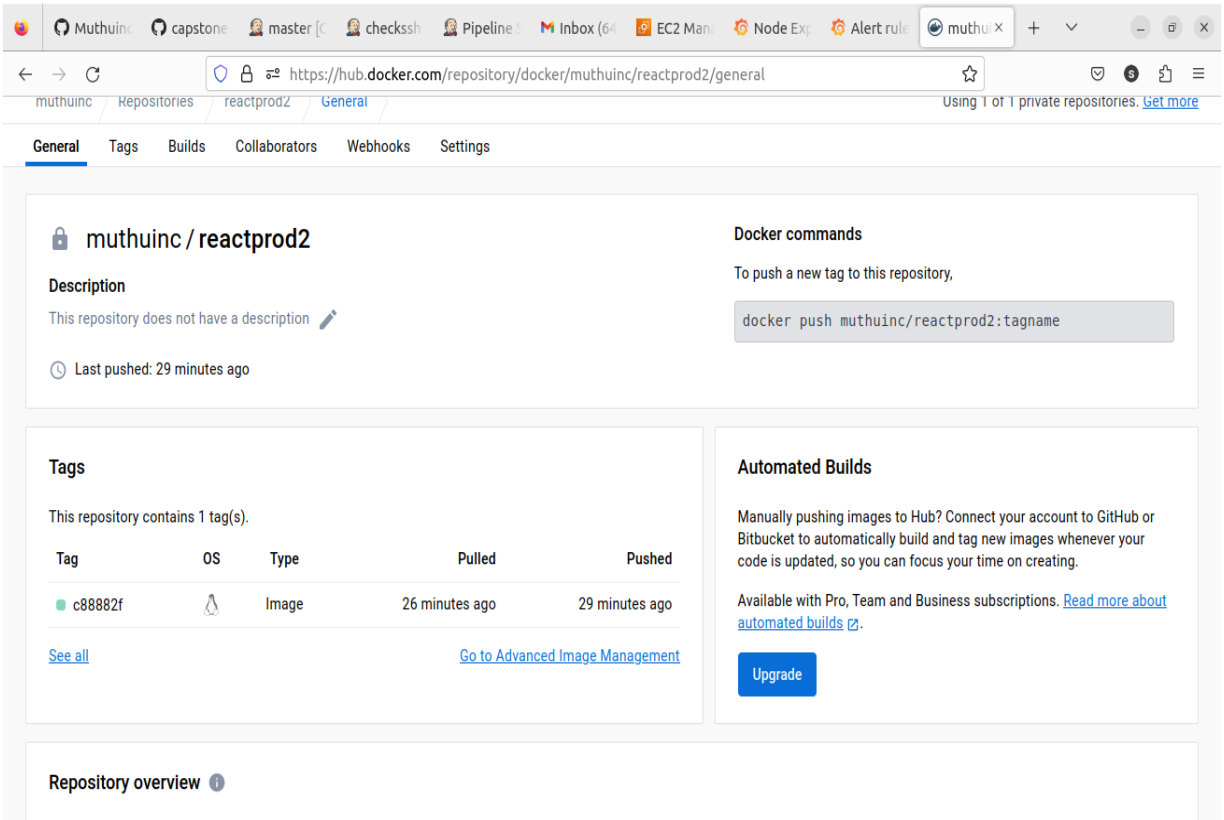
- THE OBJECTIVES OF THE PROJECT HAS BEEN DONE SUCCESSFULLY.
- REFER THE GIT HUB LINK FOR THE SCRIPT FILES AND OTHER DETAILS.

BELOW IS JUST THE DOCKER HUB REPO PICTURE FOR THE DEV – IT IS PUBLIC, you can pull it.



THE TAG IS GETTING FROM THE LAST COMMIT ID OF THE GIT REPO – REFER THE JENKINS FILE

PROD DOCKERHUB REPO – PRIVATE



I CHOSE THE MULTI-BRANCH PIPELINE. BOTH HAVE THE SAME STEPS EXCEPT SOME CHANGES LIKE PUSH TO DIFFERENT REPO AND HAVING DIFFERENT DOCKER IMAGE NAME. DIFFERENT IP ADDRESS OF THE EC2 INSTANCE.

I used *.gitattributes file merge=ours* to avoid conflicts from the same file while merging the branches.

Prod server = http://52.66.237.204:80
Dev server = http://15.207.115.86:80

the terraform main.tf file is available in both the master and dev branches.

I attached the screen shots of docker hub image with tags,
Ec2 instance applicaton page, security configs
I installed Jenkins in my local, also attached the login page,
You can also see the monitoring health status.