

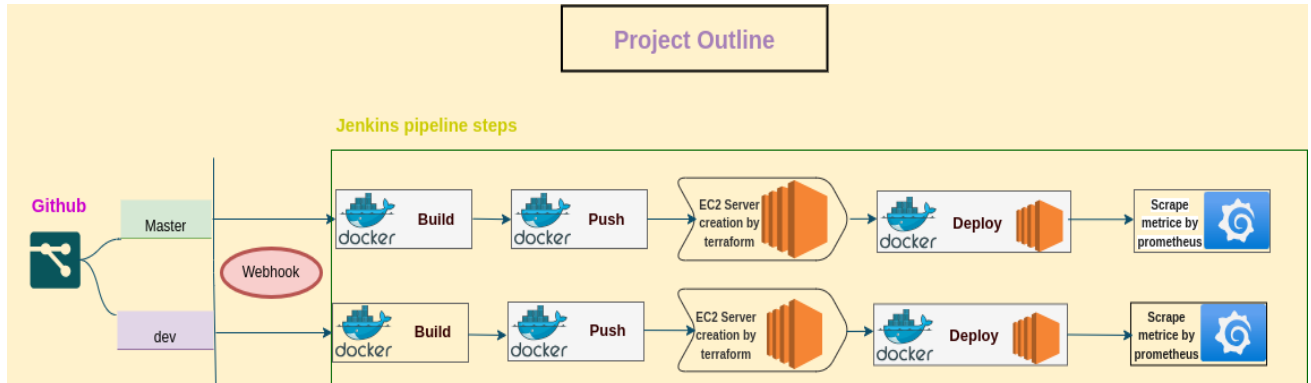
# DEVOPS- CAPESTONE - PROJECT REACT APPLICATION DEPLOYMENT

## 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.

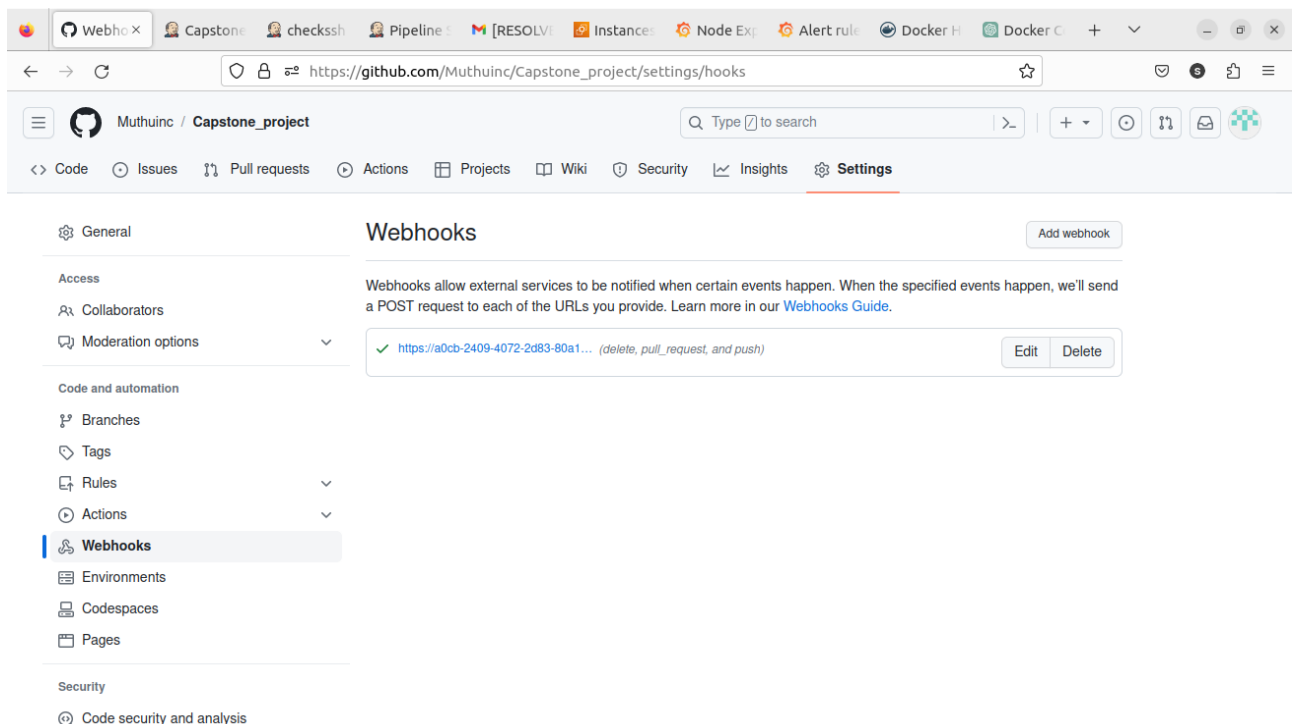
# Github link

GitHub repository of the project: [https://github.com/Muthuinc/Capstone\\_project.git](https://github.com/Muthuinc/Capstone_project.git)



## Local repository

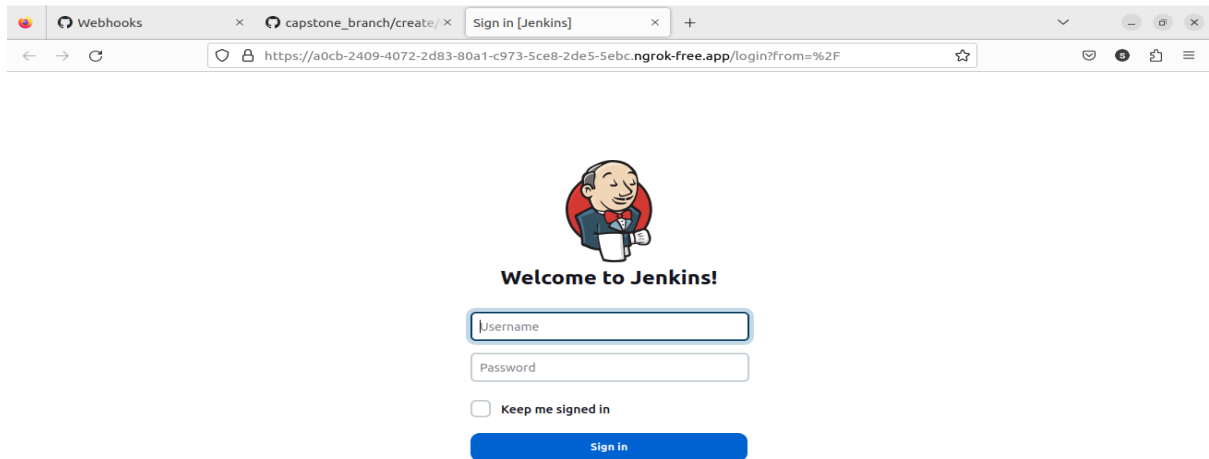
- Application code has been downloaded from the provided github repo to the local machine.
- A new directory created for the project.
- Script files for all the steps like push deploy create were made for each branch and then push to the remote repository.
- Github webhook for **push, branch delete and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



## Jenkins

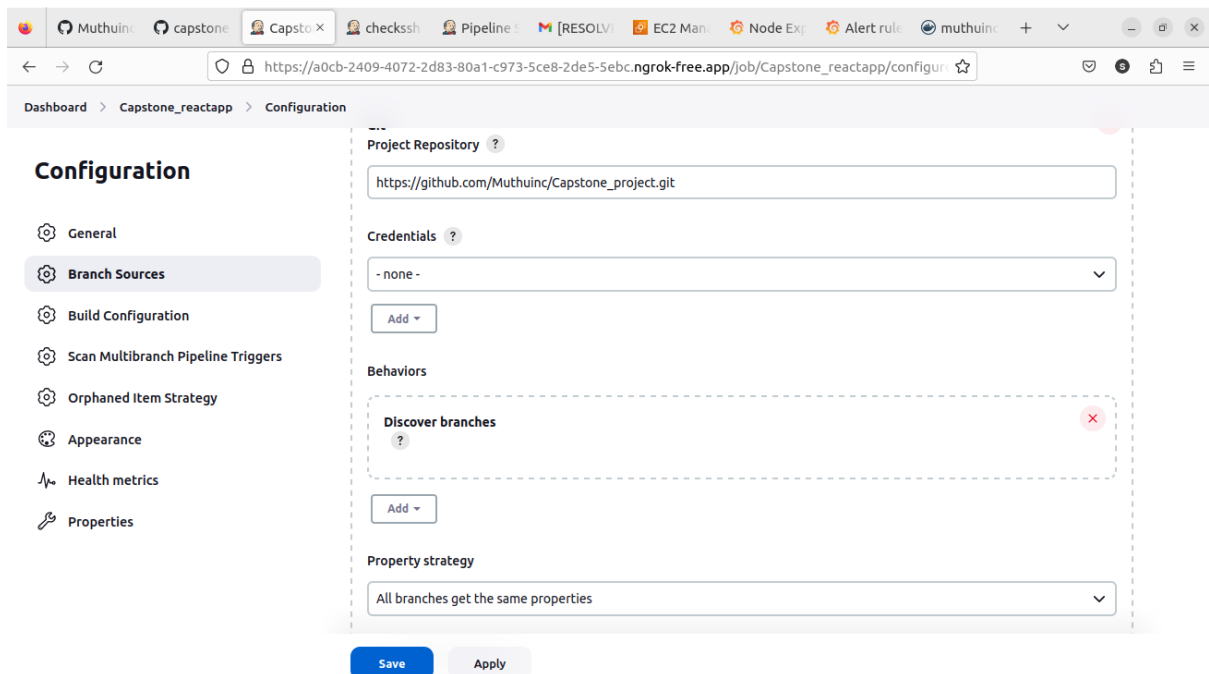
- I installed Jenkins locally in my Ubuntu machine using the official documentation.
- Created a tunnel using **ngrok** to make it accessible for web hook actions.
- New multi-branch pipeline project Capstone\_reactapp created.
- Configurations like adding the remote repository and web hook triggers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

### JENKINS LOGIN PAGE



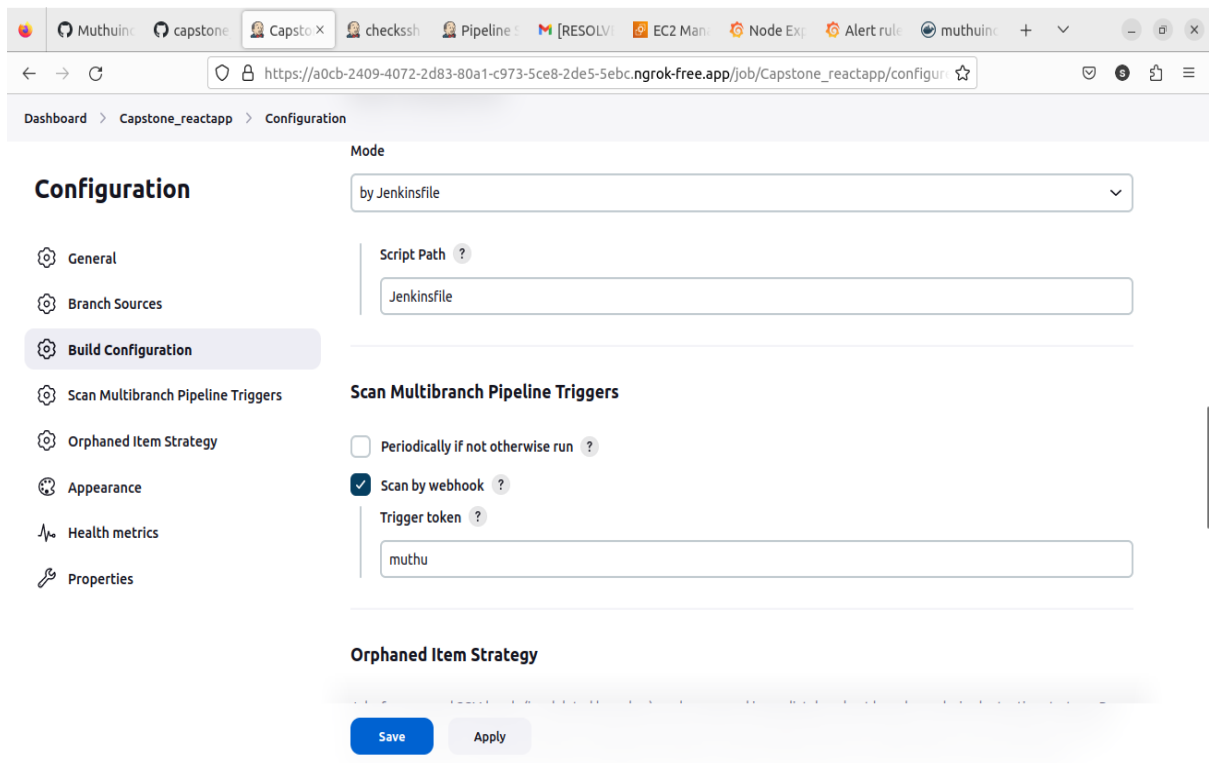
The image shows the Jenkins login page in a web browser. The browser's address bar displays the URL: `https://a0cb-2409-4072-2d83-80a1-c973-5ce8-2de5-5ebc.ngrok-free.app/login?from=%2F`. The page features the Jenkins logo (a cartoon man with a red bow tie) and the text "Welcome to Jenkins!". Below this, there are input fields for "Username" and "Password", a checkbox labeled "Keep me signed in", and a blue "Sign in" button.

### CONFIGURATION PAGE



The image shows the Jenkins configuration page for a project named "Capstone\_reactapp". The browser's address bar displays the URL: `https://a0cb-2409-4072-2d83-80a1-c973-5ce8-2de5-5ebc.ngrok-free.app/job/Capstone_reactapp/configuration`. The page has a sidebar on the left with a "Configuration" section containing links for "General", "Branch Sources", "Build Configuration", "Scan Multibranch Pipeline Triggers", "Orphaned Item Strategy", "Appearance", "Health metrics", and "Properties". The main content area is titled "Configuration" and contains several sections: "Project Repository" with a text input field containing `https://github.com/Muthuinc/Capstone_project.git`; "Credentials" with a dropdown menu showing "- none -" and an "Add" button; "Behaviors" with a dashed box containing a "Discover branches" section with a question mark icon and an "Add" button; and "Property strategy" with a dropdown menu showing "All branches get the same properties". At the bottom of the configuration area are two buttons: "Save" and "Apply".

## WEBHOOK TRIGGER - CONFIGURATION



The screenshot shows the Jenkins Configuration page for the 'Capstone\_reactapp' job. The left sidebar contains a 'Configuration' section with a list of tabs: General, Branch Sources, Build Configuration (selected), Scan Multibranch Pipeline Triggers, Orphaned Item Strategy, Appearance, Health metrics, and Properties. The main content area is titled 'Configuration' and includes a 'Mode' dropdown set to 'by Jenkinsfile'. Below this is a 'Script Path' input field containing 'Jenkinsfile'. The 'Scan Multibranch Pipeline Triggers' section has two checkboxes: 'Periodically if not otherwise run' (unchecked) and 'Scan by webhook' (checked). The 'Trigger token' input field contains 'muthu'. At the bottom, there is an 'Orphaned Item Strategy' section and two buttons: 'Save' and 'Apply'.

Dashboard > Capstone\_reactapp > Configuration

### Configuration

Mode: by Jenkinsfile

Script Path: Jenkinsfile

### Scan Multibranch Pipeline Triggers

☐ Periodically if not otherwise run ?

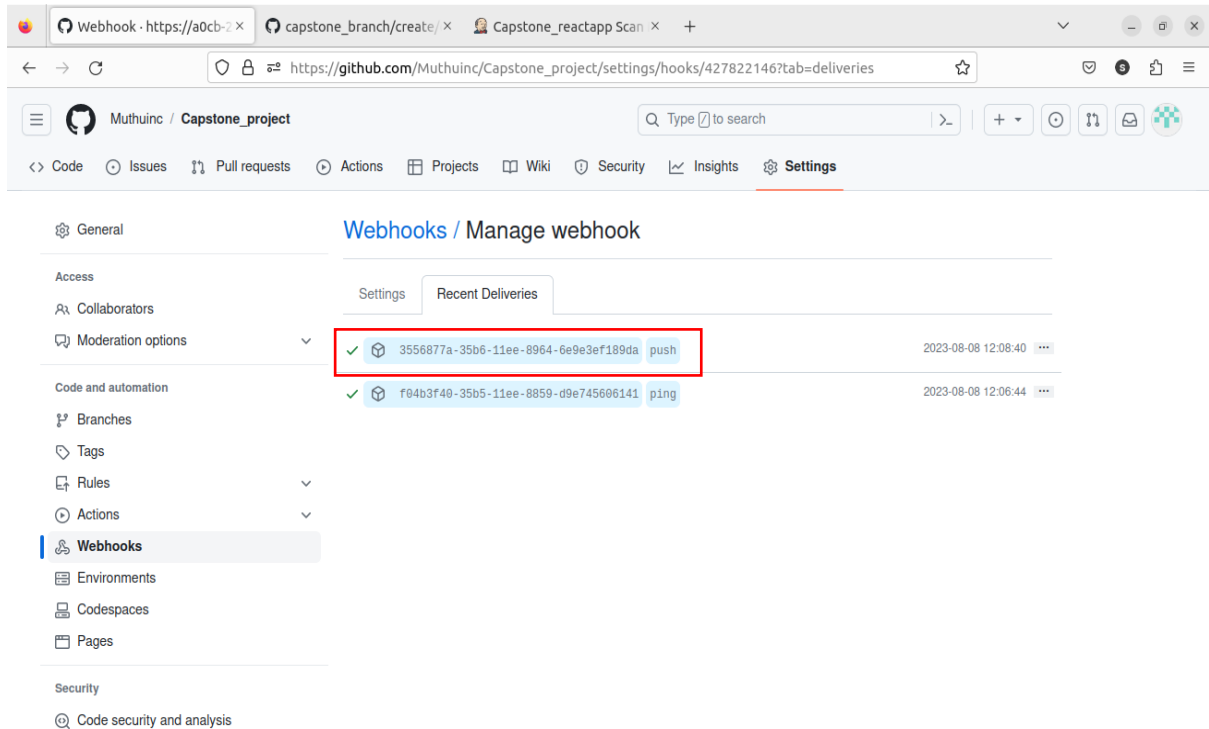
☒ Scan by webhook ?

Trigger token: muthu

### Orphaned Item Strategy

Save Apply

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



The screenshot shows the GitHub repository settings for 'Muthuinc / Capstone\_project'. The left sidebar contains a list of settings categories: General, Access, Collaborators, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Webhooks (selected), Environments, Codespaces, Pages, Security, and Code security and analysis. The main content area is titled 'Webhooks / Manage webhook' and has two tabs: 'Settings' and 'Recent Deliveries'. The 'Recent Deliveries' tab is active, showing a table of webhook deliveries. The first delivery is highlighted with a red box, showing a successful 'push' event. The second delivery is a 'ping' event.

Webhook · https://a0cb-2409-4072-2d83-80a1-c973-5ce8-2de5-5ebc.ngrok-free.app/job/Capstone\_reactapp/configure

https://github.com/Muthuinc/Capstone\_project/settings/hooks/427822146?tab=deliveries

Muthuinc / Capstone\_project

Webhooks / Manage webhook

Settings Recent Deliveries

✓	3556877a-35b6-11ee-8964-6e9e3ef189da	push	2023-08-08 12:08:40
✓	f04b3f40-35b5-11ee-8859-d9e745606141	ping	2023-08-08 12:06:44

## DEV-BRANCH STARTED TO BUILD

The Jenkins dashboard for the 'Capstone\_reactapp' project shows the 'dev' branch in a 'Building' state. The left sidebar contains navigation links for Status, Configure, Scan Multibranch Pipeline Now, Scan Multibranch Pipeline Log, Multibranch Pipeline Events, Delete Multibranch Pipeline, People, Build History, Project Relationship, Check File Fingerprint, Rename, and Config Files. The main area displays the 'Branches (1)' table with one entry for the 'dev' branch. Below the table, there are options for 'Icon: S M L', 'Icon legend', and three 'Atom feed' links for all, failures, and latest builds.

S	W	Name ↓	Last Success	Last Failure	Last Duration
...	☀	dev	N/A	N/A	N/A

## DEV-BRANCH SUCCESSFULLY BUILT

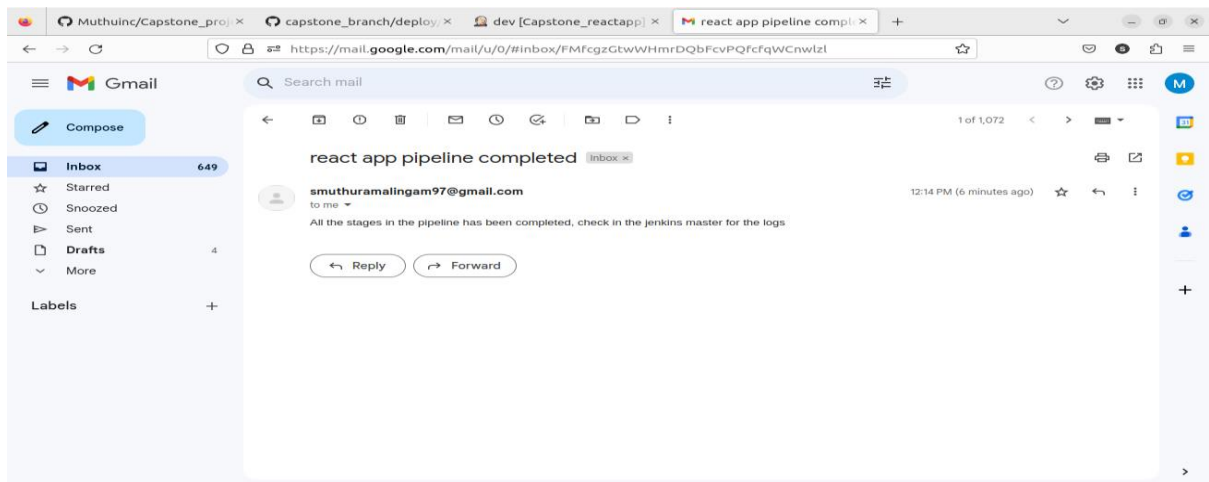
The Jenkins Pipeline View for the 'dev' branch shows a successful build. The left sidebar includes links for Status, Changes, Build Now, View Configuration, Full Stage View, Stages, and Pipeline Syntax. The 'Build History' section shows a single build (#1) from Aug 8, 2023, at 12:08 PM. The 'Stages View' table shows the following stages and their durations:

Declarative: Checkout SCM	build	push	create infra	deploy	monitor	Declarative: Post Actions
2s	2min 15s	13s	2min 17s	11s	34s	4s

The 'Permalinks' section provides links to the last build and the last stable build.

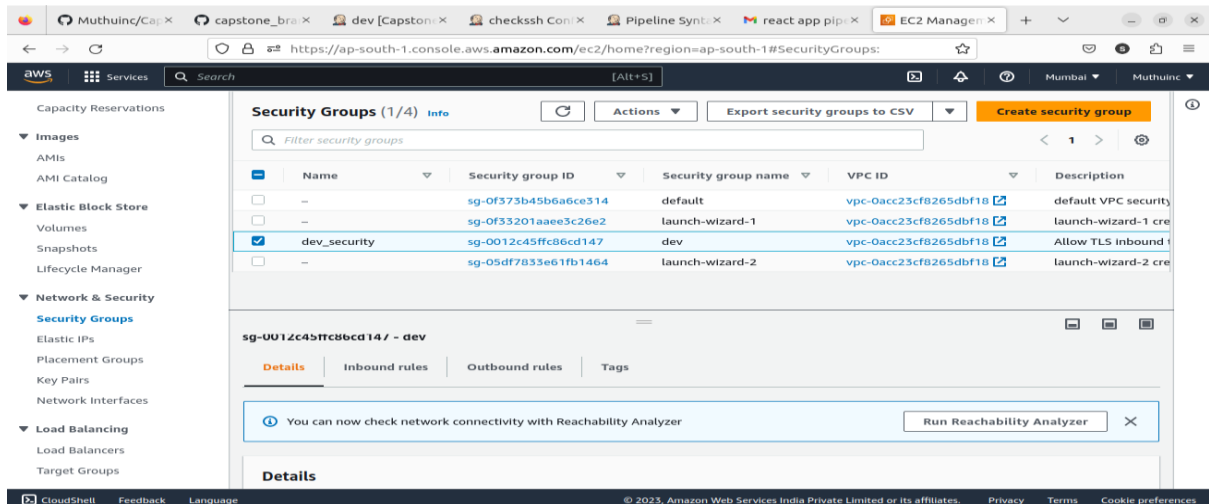
- Last build (#1), 12 min ago
- Last stable build (#1), 12 min ago

## COMPLETION MAIL RECEIVED AS A RESULT OF POST BUILD ACTION

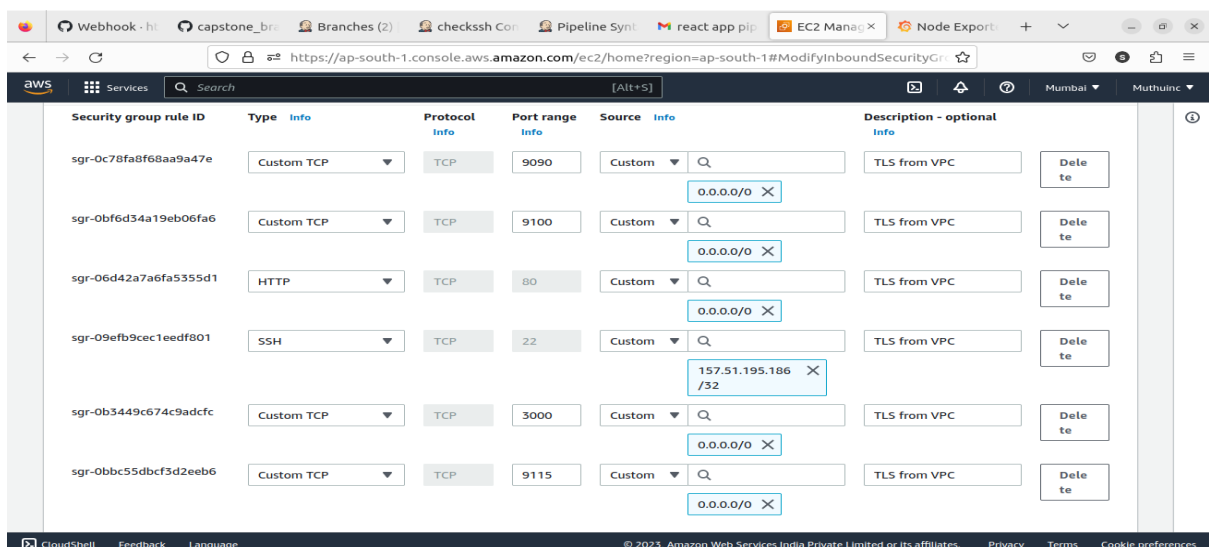


IN THE PIPELINE, TERRAFORM IS USED TO CREATE INFRASTRUCTURE FOR THE APP IN AWS

## 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



Instances (1/2) Info

Find instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	slave	i-06a598d42b38c3b2f	Stopped	t2.micro	-	No alarms	ap-south-
<input checked="" type="checkbox"/>	-	i-07bb8b31cee82b57c	Running	t2.micro	2/2 checks passed	No alarms	ap-south-

Instance: i-07bb8b31cee82b57c

Details Security Networking Storage Status checks Monitoring Tags

▼ Instance summary Info

Instance ID i-07bb8b31cee82b57c	Public IPv4 address 15.207.115.86   <a href="#">open address</a>	Private IPv4 addresses 172.31.13.66
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-15-207-115-86.ap-south-1.compute.amazonaws.com   <a href="#">open address</a>

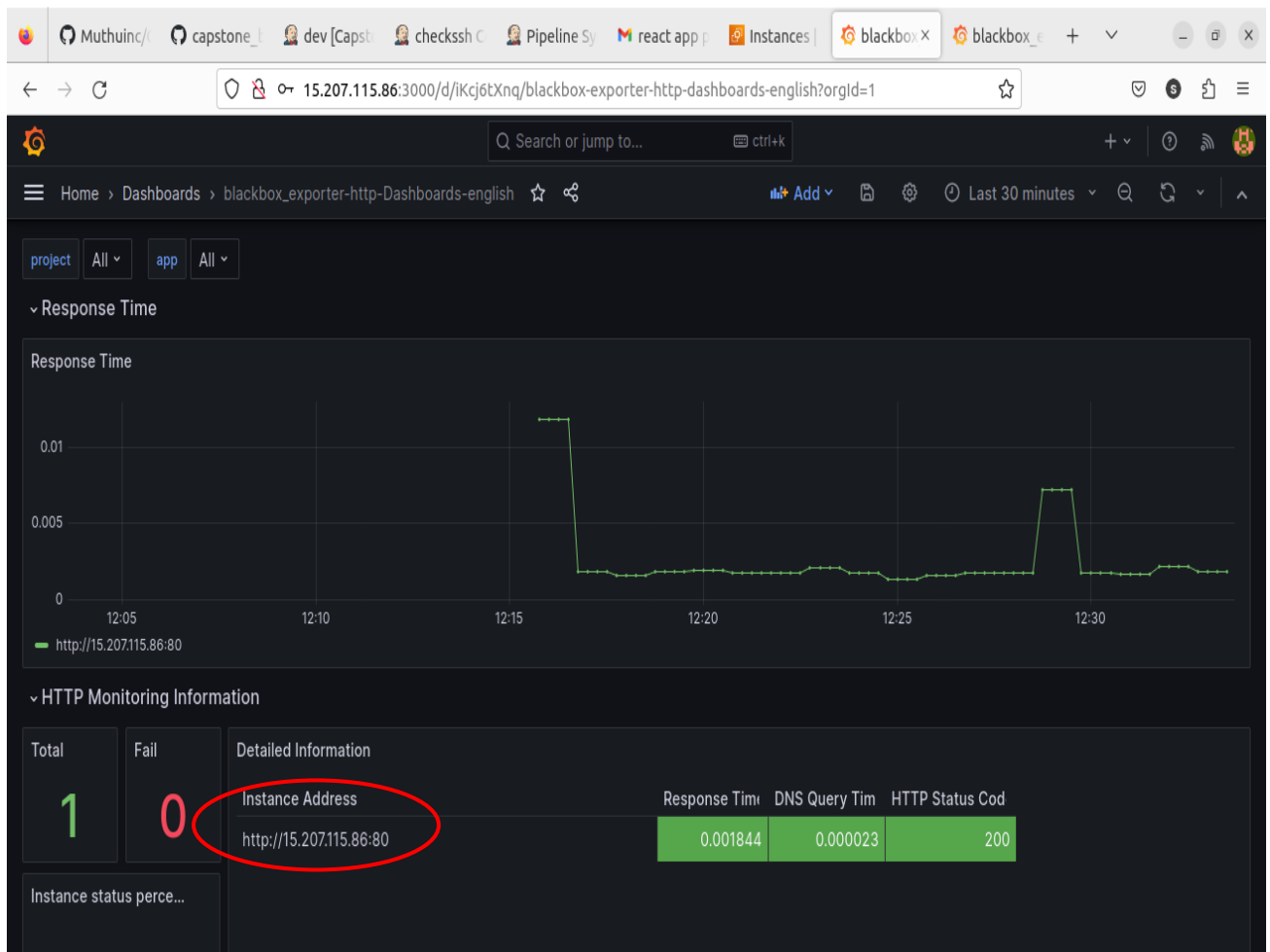
NOTE THE IP ADDRESS OF THE EC2 SERVER (FOR DEV BRANCH)

VIEW THE APPLICATION BELOW

15.207.115.86

#	Name
---	------

## 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**

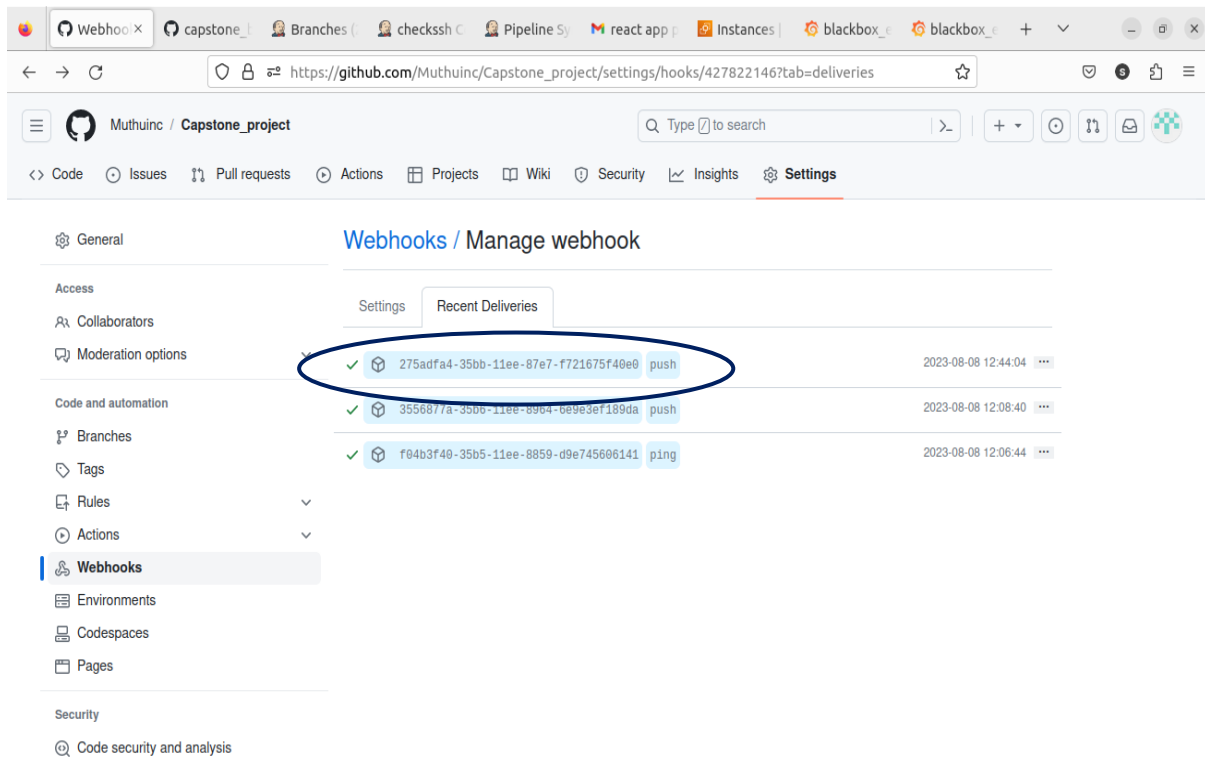
```
Incubus@Incubus-ThinkCentre-M93p: ~/Videos/Capstone_react_project
create mode 100755 deploy/deploy.sh
create mode 100755 push/push.sh
Incubus@Incubus-ThinkCentre-M93p:~/Videos/Capstone_react_project$ git merge dev
Auto-merging Jenkinsfile
Auto-merging build.sh
Auto-merging create/create.sh
Auto-merging create/docker-compose.yml
Auto-merging create/main.tf
Auto-merging deploy/deploy.sh
Auto-merging push/push.sh
Merge made by the 'ort' strategy.
 .dockerignore      | 12 +
 .gitignore          | 15 +
 Dockerfile          | 34 +
 monitor/blackbox.yml | 17 +
 monitor/docker-compose.yml | 32 +
 monitor/grafana.ini | 1469 ++++++
 monitor/monitor.sh  | 23 +
 monitor/prometheus.yml | 27 +
 package-lock.json   | 13944 ++++++
 package.json        | 36 +
 public/favicon.ico   | Bin 0 -> 3150 bytes
 public/index.html    | 43 +
 public/logo192.png  | Bin 0 -> 5347 bytes
 public/logo512.png  | Bin 0 -> 9664 bytes
 public/manifest.json | 25 +
 public/robots.txt    | 3 +
 src/App.css          | 38 +
 src/App.js           | 58 +
 src/App.test.js      | 9 +
 src/index.css        | 13 +
 src/index.js         | 17 +
 src/logo.svg        | 7 +
 src/serviceWorker.js | 141 +
 src/setupTests.js    | 5 +
 src/user-data/UserDetail.js | 41 +
 25 files changed, 16009 insertions(+)
create mode 100644 .dockerignore
```



## PUSHING THE CHANGES TO THE REMOTE REPO

```
Incubus@Incubus-ThinkCentre-M93p: ~/Videos/Capstone_react_project
create mode 100644 .dockerignore
create mode 100644 .gitignore
create mode 100644 Dockerfile
create mode 100644 monitor/blackbox.yml
create mode 100644 monitor/docker-compose1.yml
create mode 100644 monitor/grafana.ini
create mode 100755 monitor/monitor.sh
create mode 100644 monitor/prometheus.yml
create mode 100644 package-lock.json
create mode 100644 package.json
create mode 100644 public/favicon.ico
create mode 100644 public/index.html
create mode 100644 public/logo192.png
create mode 100644 public/logo512.png
create mode 100644 public/manifest.json
create mode 100644 public/robots.txt
create mode 100644 src/App.css
create mode 100644 src/App.js
create mode 100644 src/App.test.js
create mode 100644 src/index.css
create mode 100644 src/index.js
create mode 100644 src/logo.svg
create mode 100644 src/serviceWorker.js
create mode 100644 src/setupTests.js
create mode 100644 src/user-data/UserDetail.js
Incubus@Incubus-ThinkCentre-M93p:~/Videos/Capstone_react_project$ git push
Username for 'https://github.com': Muthuinc
Password for 'https://Muthuinc@github.com':
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 4 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 1.79 KiB | 1.79 MiB/s, done.
Total 14 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), completed with 6 local objects.
To https://github.com/Muthuinc/Capstone_project.git
  306c8f9..c88882f master -> master
Incubus@Incubus-ThinkCentre-M93p:~/Videos/Capstone_react_project$
```

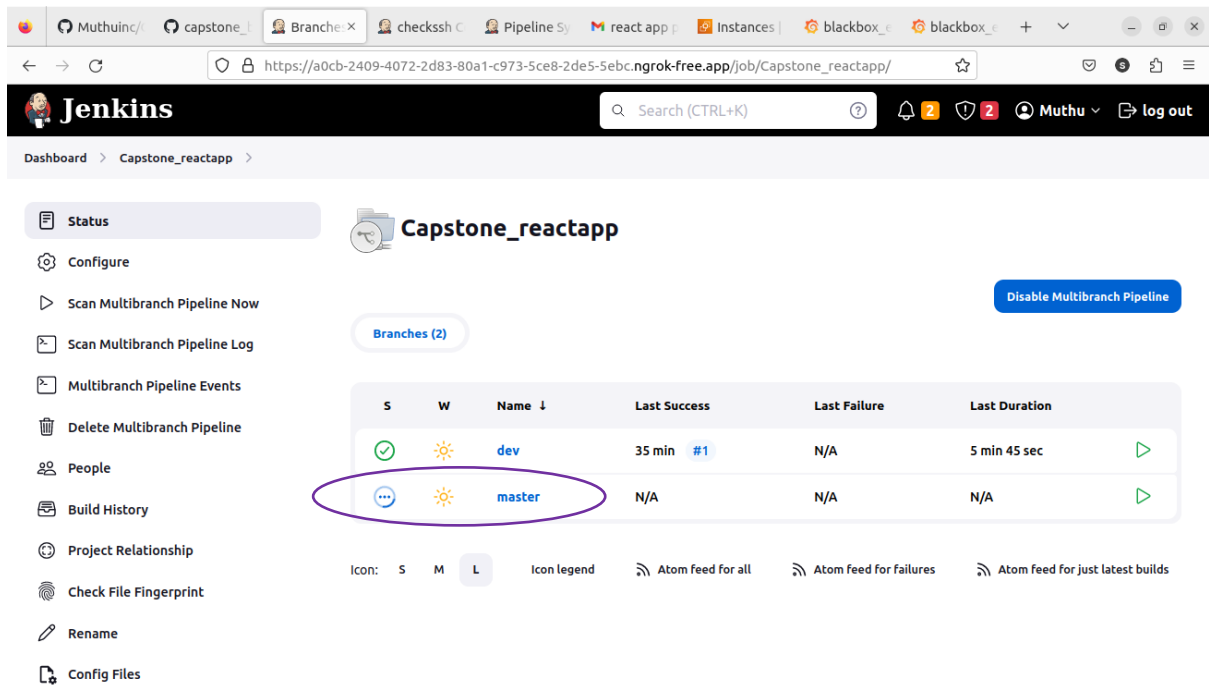
## YOU CAN SEE THE WEBHOOK SENDS ANOTHER TRIGGER AS A RESULT OF MASTER BRANCH



The screenshot shows the GitHub repository settings for 'Muthuinc / Capstone\_project'. The 'Webhooks' tab is selected, displaying a list of recent deliveries. A blue circle highlights the first delivery, which is a 'push' event triggered by the master branch. The delivery is successful, indicated by a green checkmark.

SHA	Event	Time	Status
275adfa4-35bb-11ee-87e7-f721675f40e0	push	2023-08-08 12:44:04	Success
3556877a-35bb-11ee-89c4-b89e3ef189da	push	2023-08-08 12:08:40	Success
f04b3f40-35b5-11ee-8859-d9e745606141	ping	2023-08-08 12:06:44	Success

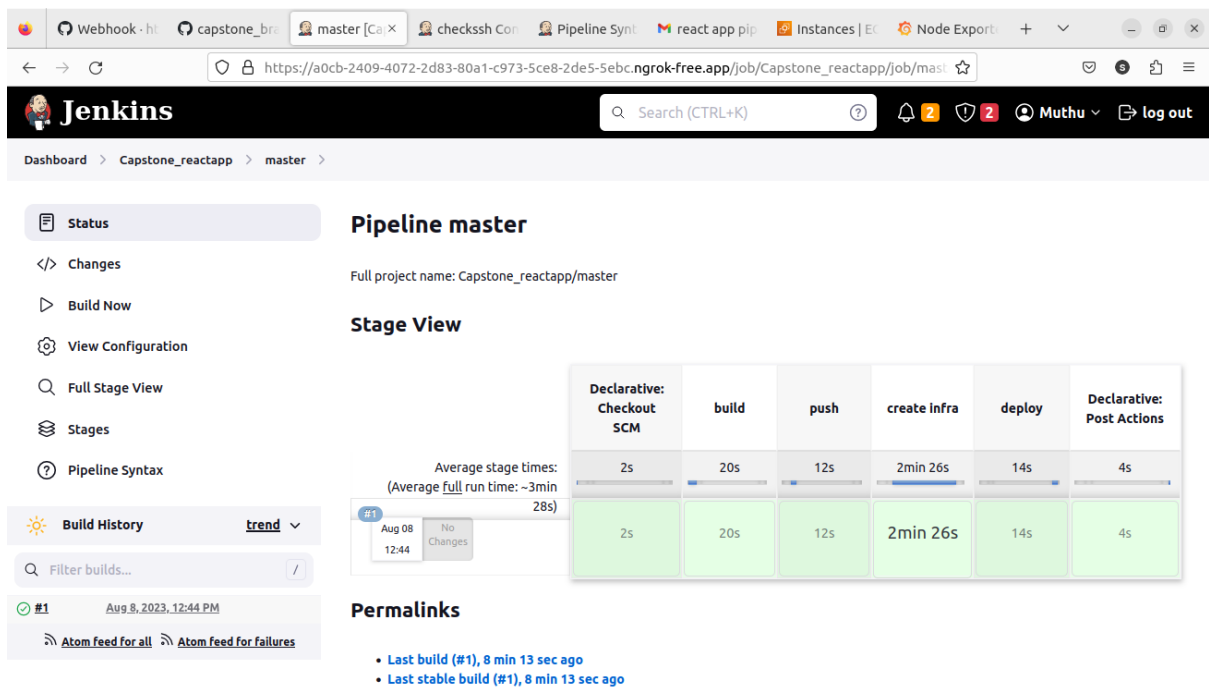
## JENKINS SERVER DETECT THE MASTER BRANCH AUTOMATICALLY AND STARTED TO WORK OUT



The screenshot shows the Jenkins dashboard for the 'Capstone\_reactapp' job. The 'master' branch is highlighted with a red circle. The dashboard includes a sidebar with navigation options like Status, Configure, and Scan Multibranch Pipeline Now. The main area displays a table of branches with columns for Status, Name, Last Success, Last Failure, and Last Duration.

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	dev	35 min #1	N/A	5 min 45 sec
⋮	☀	master	N/A	N/A	N/A

## MASTER BRANCH PIPELINE STEPS SUCCESSFULLY EXECUTED



The screenshot shows the Jenkins pipeline view for the 'master' branch. The pipeline is titled 'Pipeline master' and shows a successful build. The 'Stage View' section displays a table of stages with their respective durations. The 'Build History' section shows the build status and time.

Declarative: Checkout SCM	build	push	create Infra	deploy	Declarative: Post Actions
2s	20s	12s	2min 26s	14s	4s

**Permalinks**

- Last build (#1), 8 min 13 sec ago
- Last stable build (#1), 8 min 13 sec ago

## NEW SECURITY GROUP FOR THE PROD SERVER IS CREATED BY TERRAFORM

The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Security Groups' page is displayed, showing a list of security groups. The 'Prod\_security' group is selected, and its ID 'sg-03d0bdd43f37f6bf2' is circled in red. A red arrow points from this ID to the 'Edit inbound rules' page in the second screenshot.

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0f373b45b6a6ce314	default	vpc-0acc23cf8265dbf18	default VPC security
-	sg-0f33201aeee3c26e2	launch-wizard-1	vpc-0acc23cf8265dbf18	launch-wizard-1 cre
Prod_security	sg-03d0bdd43f37f6bf2	prod	vpc-0acc23cf8265dbf18	Allow TLS inbound t
dev_security	sg-001245ffc86cd147	dev	vpc-0acc23cf8265dbf18	Allow TLS inbound t
-	sg-05d7833e61fb1464	launch-wizard-2	vpc-0acc23cf8265dbf18	launch-wizard-2 cre

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

The screenshot shows the 'Edit inbound rules' page for the 'Prod\_security' group. The 'SSH' rule is highlighted, and its source is being edited to '157.51.195.186 /32'. The 'HTTP' rule is also visible, with its source set to '0.0.0.0/0'.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-002db91ae17a9b54d	HTTP	TCP	80	Custom	TLS from VPC
sgr-0e71cdea2f4a6c399	SSH	TCP	22	Custom	TLS from VPC

## EC2 SERVER FOR THE PROD

The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Instances' page displays a list of three EC2 instances. The second instance, with ID 'i-04127916bfef940f8', is in a 'Running' state. Below the list, the details for this instance are shown, including its public IPv4 address, 52.66.237.204, which is circled in red. The console also shows the instance's state as 'Running' and its public IPv4 DNS as 'ec2-52-66-237-204.ap-south-1.compute.amazonaws.com'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
slave	i-06a598d42b38c5b2f	Stopped	t2.micro	-	No alarms	ap-south-
-	i-04127916bfef940f8	Running	t2.micro	2/2 checks passed	No alarms	ap-south-
-	i-07bb8b31cee82b57c	Running	t2.micro	2/2 checks passed	No alarms	ap-south-

**Instance: i-04127916bfef940f8**

**Instance summary**

Instance ID	Public IPv4 address	Private IPv4 addresses
i-04127916bfef940f8	52.66.237.204   <a href="#">open address</a>	172.31.0.249

**Instance state**

Running

**Public IPv4 DNS**

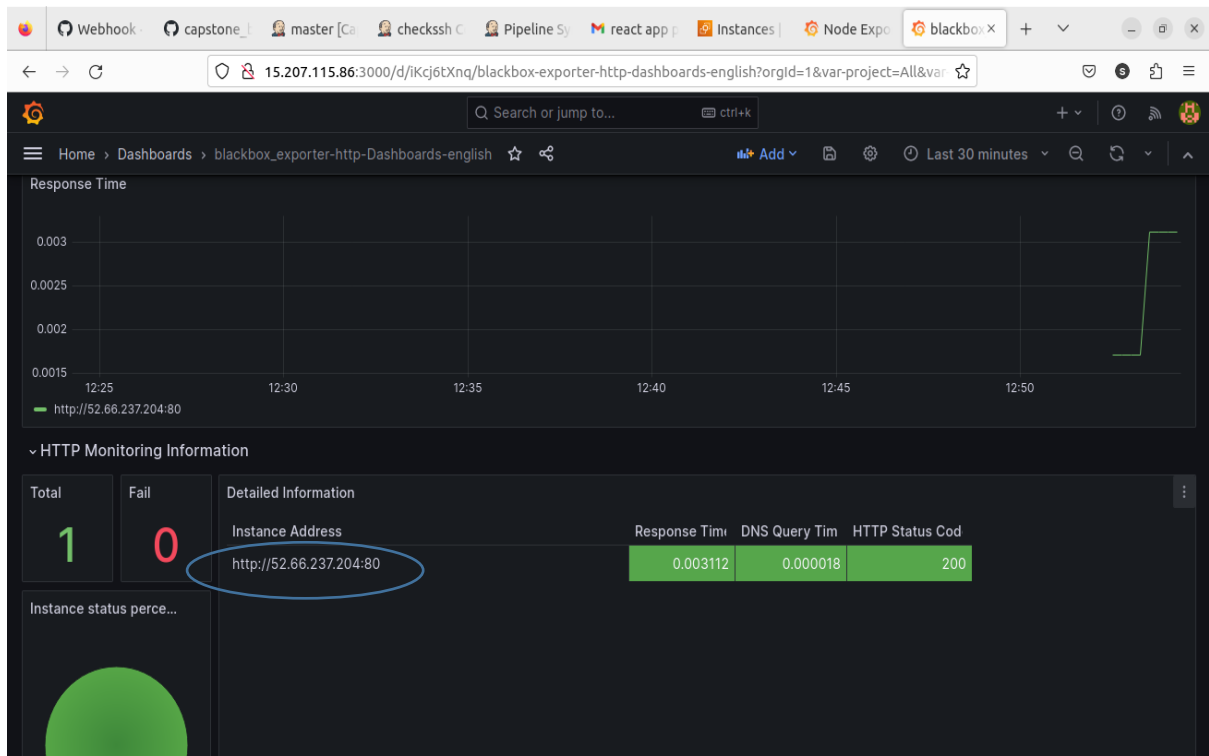
ec2-52-66-237-204.ap-south-1.compute.amazonaws.com | [open address](#)

## APPLICATION IS RUNNING SUCCESSFULLY - PROD ENVIRONMENT

The screenshot shows a web browser with the URL '52.66.237.204' entered in the address bar, which is circled in red. Below the address bar, there is a table with three columns: '#', 'Name', and 'React App'. The table is currently empty.

#	Name	React App
---	------	-----------

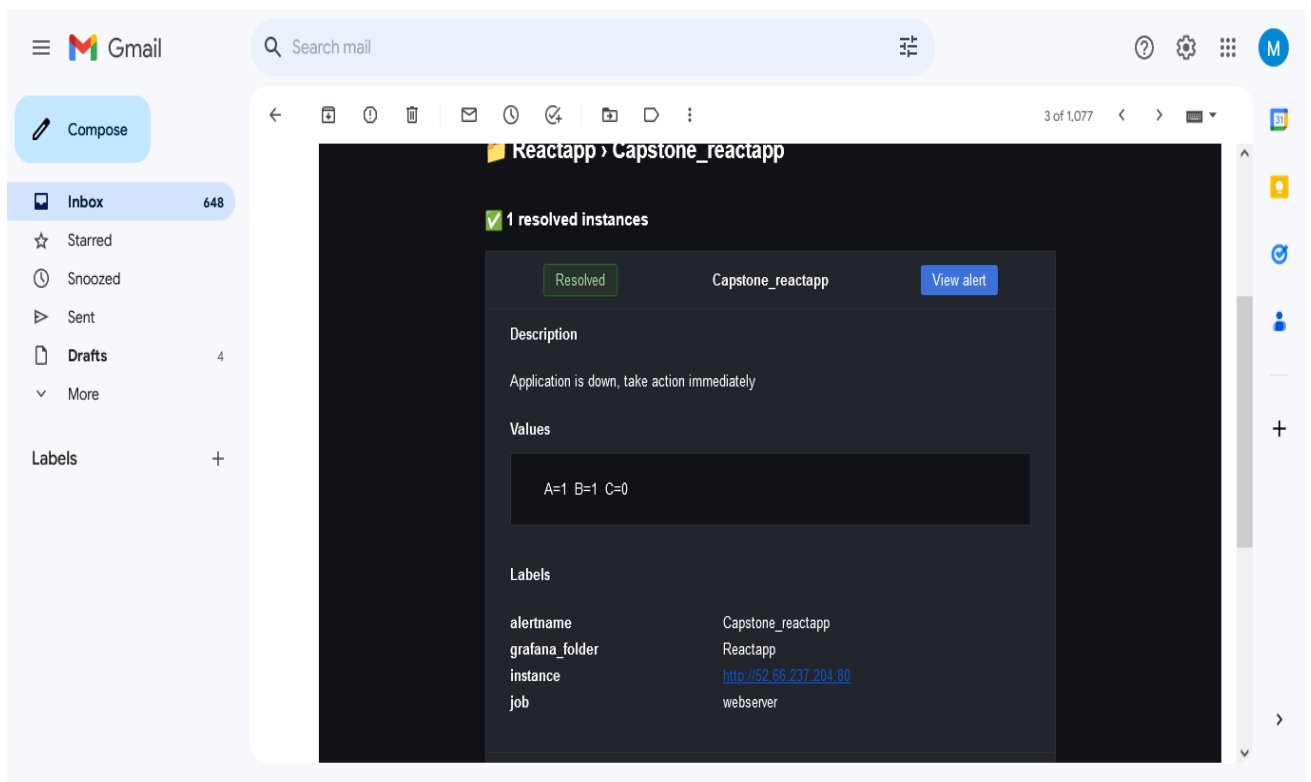
## GRAFANA MONITORING IS SET UP



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

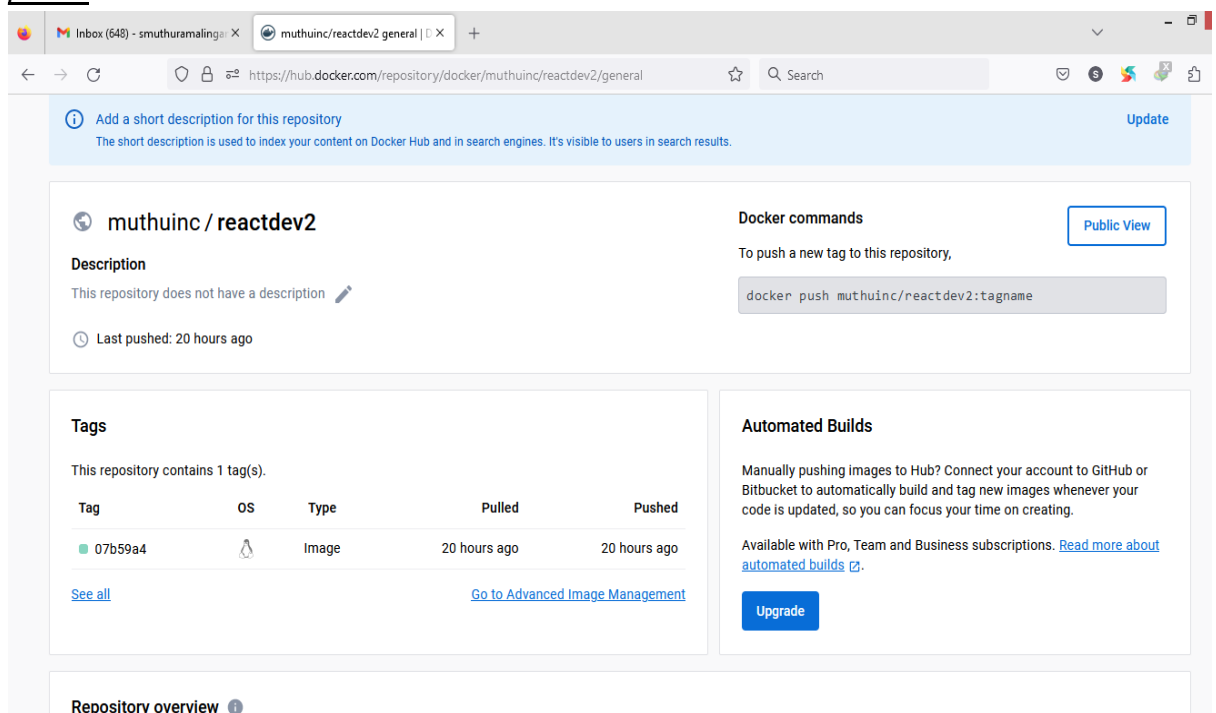
The image is a screenshot of a Gmail interface. The main content is an email from 'Reactapp > Capstone\_reactapp' with the subject '1 firing instances'. The email body shows a 'Firing' alert for 'Capstone\_reactapp'. The description states 'Application is down, take action immediately'. The 'Values' section shows 'A=0 B=0 C=1'. The 'Labels' section lists 'alertname: Capstone\_reactapp', 'grafana\_folder: Reactapp', 'instance: <http://52.66.237.204:80>', and 'job: webserver'. The instance URL is circled in blue.

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



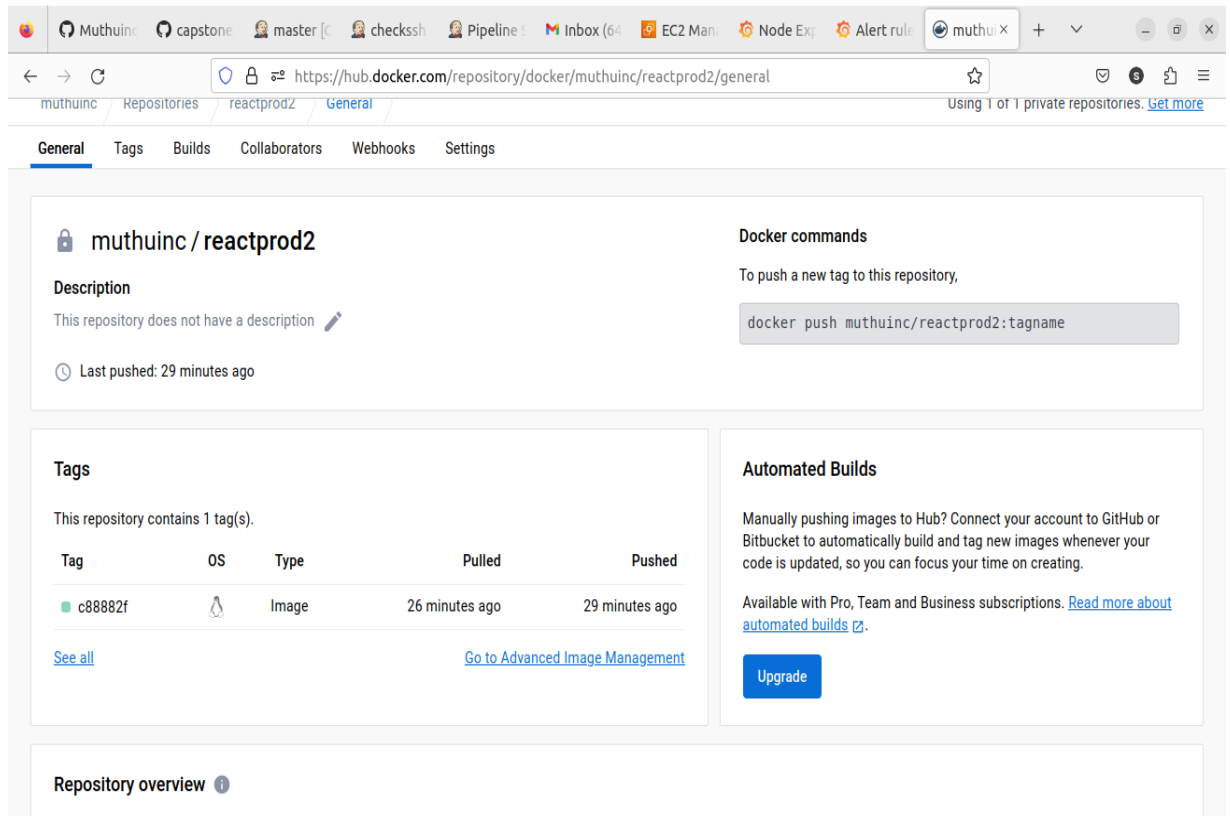
- THE OBJECTIVES OF THE PROJECT HAS BEEN DONE SUCCESSFULLY.
- REFER THE GIT HUB LINK FOR THE SCRIPT FILES AND OTHER DETAILS.

**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>      the terraform main.tf file is available in  
Dev server = <http://15.207.115.86:80>      both the master and dev branches.

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