Application Deployment

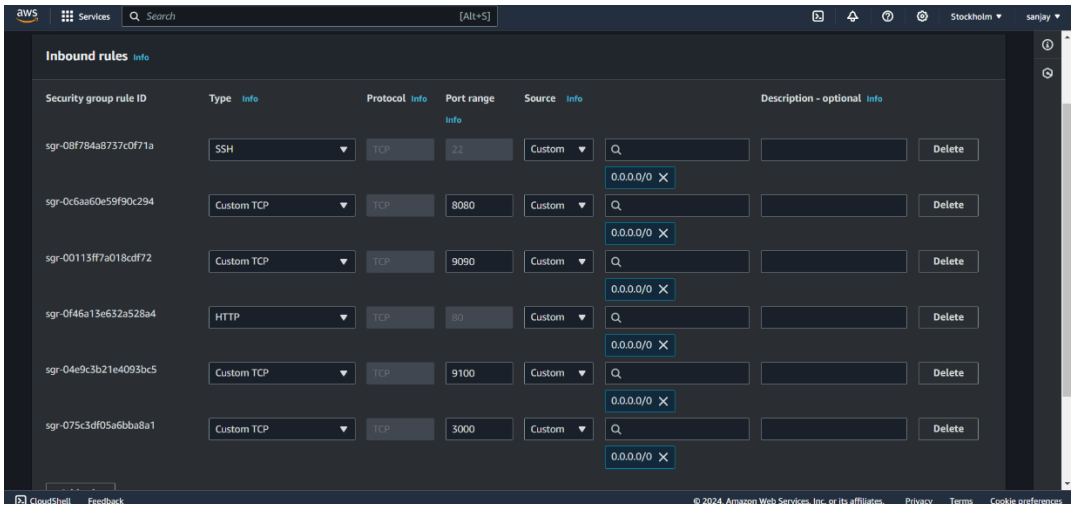
(Deploy the given React application to a production ready state.)

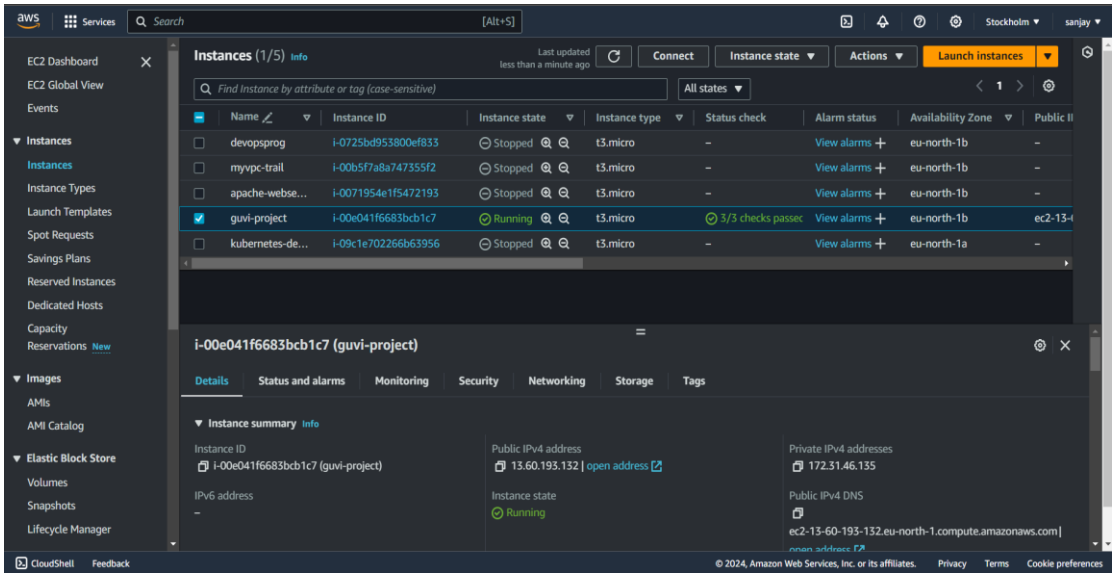
AWS:

Launched an Ubuntu, T2.micro instance and configured Security Group as below:

Configure Inbound Rules:

- **Rule 1: Allow SSH Access from Your IP Address:** Login to server can should be made only from your Ip address
  - Type: SSH, Protocol: TCP, Port Range: 22
  - Source: My IP (Use the current IP of your machine)
- **Rule 2: Allow HTTP/HTTPS Access from Anywhere:** Whoever has the IP address can access the application
  - Type: HTTP, Protocol: TCP, Port Range: 80
  - Source: Anywhere (0.0.0.0/0)
- Rest all settings have remained the same or proceeded with the default settings and Launched the EC2 instance naming “PROJECT”.

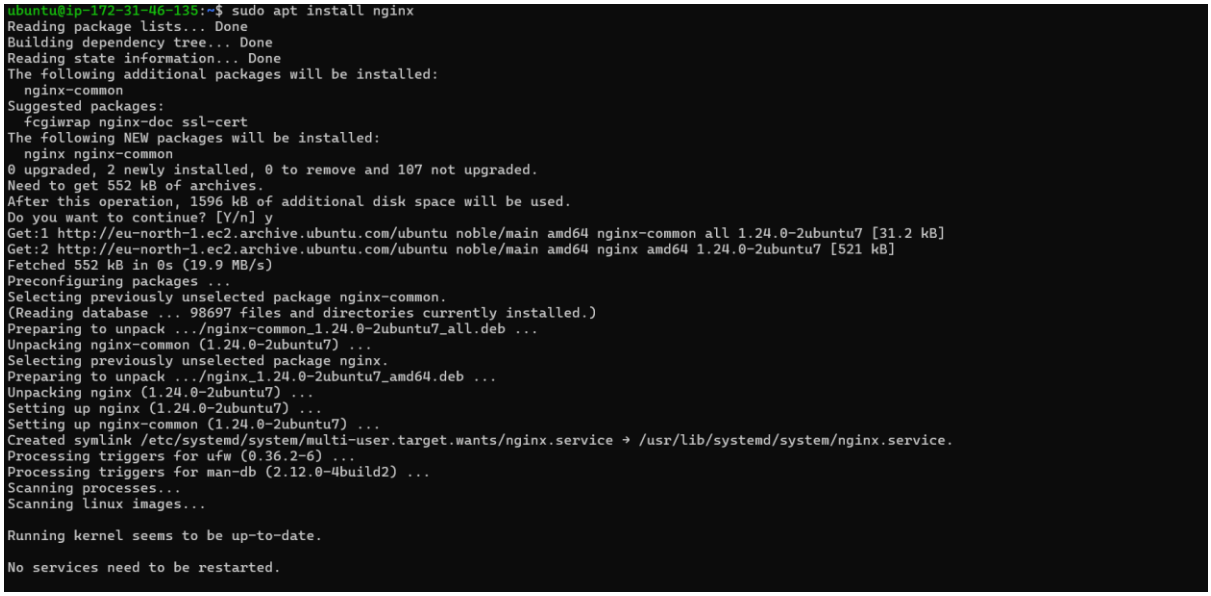


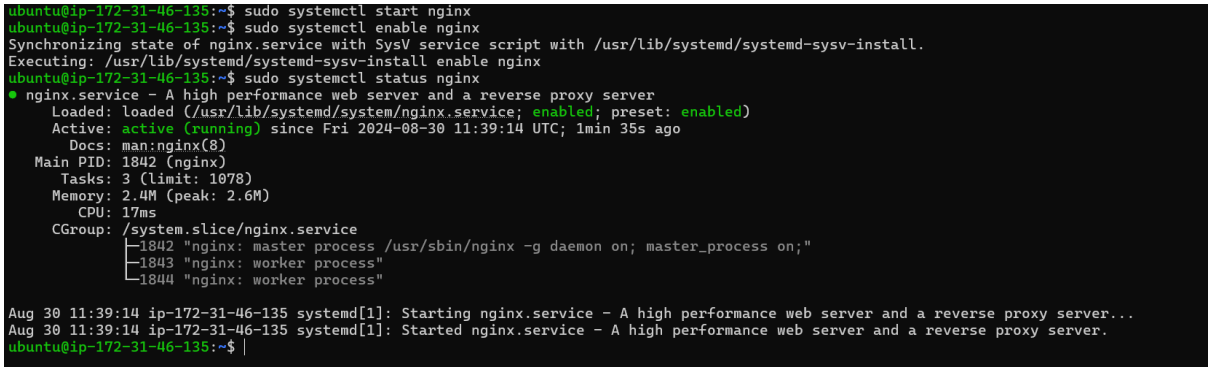


Once the Instance is created and running, Launched the terminal and to execute the given application we need Nginx application on the machine. Here are the commands used to update and install Nginx.

```
sudo apt update
sudo apt install nginx -y
sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl status nginx
```

- As we have to run the application on port 80 I have made changes to the Nginx configuration file to listen to port 3000 as default for Nginx.





Version Control:

- Prerequisites for creating git ignore and docker ignore files are to install Git and Docker and you should have access to the repository and necessary permissions. Below is Git install & version

```
sudo apt-get update
sudo apt-get install git
git config --global user.name "Your Name"
git config --global user.email
"your.email@example.com"
```

Go to GitHub and create a new repository.

```
root@ip-172-31-46-135:/home/ubuntu# git clonehttps://github.com/Sanjai9000/Guvi-project.git
git: 'clonehttps://github.com/Sanjai9000/Guvi-project.git' is not a git command. See 'git --help'.
root@ip-172-31-46-135:/home/ubuntu# git clone https://github.com/Sanjai9000/Guvi-project.git
Cloning into 'Guvi-project'...
warning: You appear to have cloned an empty repository.
root@ip-172-31-46-135:/home/ubuntu#
```

- Docker install & version:

```
sudo apt update
sudo apt install docker-ce -y
sudo systemctl start docker
sudo systemctl enable docker
sudo docker –version
sudo usermod -aG docker $USER
docker login
```

```
ubuntu@ip-172-31-46-135:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 25 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 runc amd64 1.1.12-0ubuntu3 [8599 kB]
Get:4 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 containerd amd64 1.7.12-0ubuntu4 [38.6 MB]
Get:5 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702~willsync1 [4450 B]
Get:6 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:7 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 docker.io amd64 24.0.7-0ubuntu4 [29.1 MB]
Get:8 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 76.8 MB in 1s (69.2 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 98330 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1-1ubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1-1ubuntu2) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.12-0ubuntu3_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu3) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.7.12-0ubuntu4_amd64.deb ...
Unpacking containerd (1.7.12-0ubuntu4) ...
Selecting previously unselected package dns-root-data.
Preparing to unpack .../4-dns-root-data_2023112702~willsync1_all.deb ...
Unpacking dns-root-data (2023112702~willsync1) ...
Selecting previously unselected package dnsmasq-base.
```

```
ubuntu@ip-172-31-46-135:~$ sudo usermod -aG docker ubuntu
ubuntu@ip-172-31-46-135:~$ docker version
Client:
 Version:           24.0.7
 API version:       1.43
 Go version:        go1.22.2
 Git commit:        24.0.7-0ubuntu4
 Built:             Wed Apr 17 20:08:25 2024
 OS/Arch:           linux/amd64
 Context:           default
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version":
 dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-46-135:~$
```

- Now open the terminal and Clone the given Repository with the build application. And navigate to the directory and located all the files.

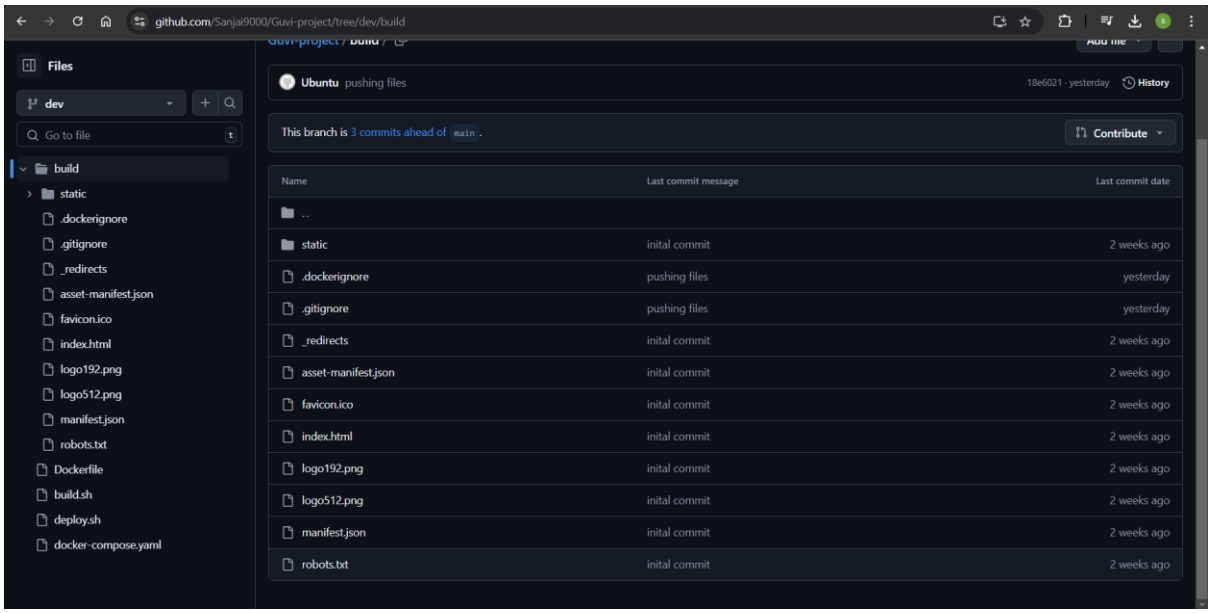
```
ubuntu@ip-172-31-46-135:~$ ls
Guvi-project  devops-build
ubuntu@ip-172-31-46-135:~$ cd Guvi-project/
ubuntu@ip-172-31-46-135:~/Guvi-project$ ls
Dockerfile  build  build.sh  deploy.sh  docker-compose.yaml
ubuntu@ip-172-31-46-135:~/Guvi-project$
```

- Now firstly creating the Git ignore file and the Docker ignore file .After creating both the files pushed the given repository to my newly created repository with the same name 'devops-build' to new 'dev' branch.

```
ubuntu@ip-172-31-46-135: ~/
npm-debug.log*
yarn-debug.log*
yarn-error.log*
*.log
node_modules
dist/
.git/
.gitignore
.DS_Store
.env
.env.local
*.swp
*.tmp
.vscode/
.idea/
.cache
~
                                                     16,6          All
```

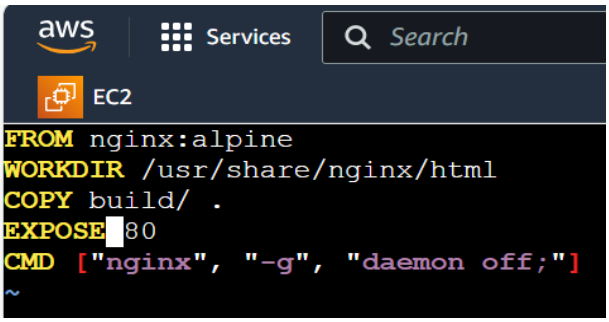
```
#logs
npm-debug.log*
yarn-debug.log*
*.log
#dependencies
/node_modules
/.pnp
.pnp.js
# misc
.DS_Store
.vscode/
#environment variables
.env.local
.env.development.local
.env.test.local
.env.production.local
.env*
# testing
/coverage
# production
/build
/dist
~
                                                     22,5          All
```

- Here we can see the '.gitignore' & '.dockerignore' files which have been created.



Docker:

- Dockerize the application by creating the Dockerfile.



- To create a Docker-compose file for the created image firstly we need to install Docker-compose:

```
sudo apt update
sudo apt install -y docker-compose
```

- Apply executable permissions to the Docker Compose binary:

```
sudo chmod +x /usr/local/bin/docker-compose
```

- Verify the installation:

```
docker-compose –version
```

```
ubuntu@ip-172-31-46-135:~/Guvi-project$ sudo apt install docker-ce -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils dns-root-data dnsmasq-base ubuntu-fan
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following packages will be REMOVED:
  containerd docker.io runc
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 slirp4netns
0 upgraded, 9 newly installed, 3 to remove and 101 not upgraded.
Need to get 121 MB of archives.
After this operation, 149 MB of additional disk space will be used.
Get:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:4 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.21-1 [29.5 MB]
Get:5 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.16.2-1~ubuntu.24.04~noble [29.9 MB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:27.2.0-1~ubuntu.24.04~noble [14.8 MB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:27.2.0-1~ubuntu.24.04~noble [25.3 MB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:27.2.0-1~ubuntu.24.04~noble [9318 kB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 2.29.2-1~ubuntu.24.04~noble [12.5 MB]
Fetched 121 MB in 2s (69.7 MB/s)
(Reading database ... 98747 files and directories currently installed.)
Removing docker.io (24.0.7-0ubuntu4) ...
'/usr/share/docker.io/contrib/nuke-graph-directory.sh' -> '/var/lib/docker/nuke-graph-directory.sh'
```

```
ubuntu@ip-172-31-46-135:~/Guvi-project$ sudo systemctl start docker
ubuntu@ip-172-31-46-135:~/Guvi-project$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-08-30 12:17:03 UTC; 9s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 3706 (dockerd)
      Tasks: 8
     Memory: 37.1M (peak: 37.4M)
        CPU: 371ms
     CGroup: /system.slice/docker.service
             └─3706 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Aug 30 12:17:02 ip-172-31-46-135 dockerd[3706]: time="2024-08-30T12:17:02.456459929Z" level=info msg="Starting up"
Aug 30 12:17:02 ip-172-31-46-135 dockerd[3706]: time="2024-08-30T12:17:02.466420279Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-reso>
Aug 30 12:17:02 ip-172-31-46-135 dockerd[3706]: time="2024-08-30T12:17:02.604469989Z" level=info msg="[graphdriver] using prior storage driver: overlay2"
Aug 30 12:17:02 ip-172-31-46-135 dockerd[3706]: time="2024-08-30T12:17:02.608090512Z" level=info msg="Loading containers: start."
Aug 30 12:17:02 ip-172-31-46-135 dockerd[3706]: time="2024-08-30T12:17:02.873909633Z" level=info msg="Default bridge (docker0) is assigned with an IP addre>
Aug 30 12:17:02 ip-172-31-46-135 dockerd[3706]: time="2024-08-30T12:17:02.942827798Z" level=info msg="Loading containers: done."
Aug 30 12:17:02 ip-172-31-46-135 dockerd[3706]: time="2024-08-30T12:17:02.969033944Z" level=info msg="Docker daemon" commit=3ab5c7d containerd-snapshotter=>
Aug 30 12:17:02 ip-172-31-46-135 dockerd[3706]: time="2024-08-30T12:17:02.969489910Z" level=info msg="Daemon has completed initialization"
Aug 30 12:17:03 ip-172-31-46-135 systemd[1]: Started docker.service - Docker Application Container Engine.
Aug 30 12:17:03 ip-172-31-46-135 dockerd[3706]: time="2024-08-30T12:17:03.031307087Z" level=info msg="API listen on /run/docker.sock"
lines 1-22/22 (END)
```

```
ubuntu@ip-172-31-46-135:~/Guvi-project$ sudo usermod -aG docker ubuntu
ubuntu@ip-172-31-46-135:~/Guvi-project$ docker login

USING WEB BASED LOGIN
To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: BFTC-TZWN
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser…
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
ubuntu@ip-172-31-46-135:~/Guvi-project$ |
```

```
ubuntu@ip-172-31-46-135:~/Guvi-project$ sudo curl -L "https://github.com/docker/compose/releases/download/v2.29.2/docker-compose-$(uname -s)-$(uname -m)" -o
 /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 60.2M  100 60.2M    0     0  69.5M      0 --:--:-- --:--:-- --:--:--  340M
ubuntu@ip-172-31-46-135:~/Guvi-project$ sudo chmod +x /usr/local/bin/docker-compose
ubuntu@ip-172-31-46-135:~/Guvi-project$ docker-compose --version
Docker Compose version v2.29.2
ubuntu@ip-172-31-46-135:~/Guvi-project$ |
```

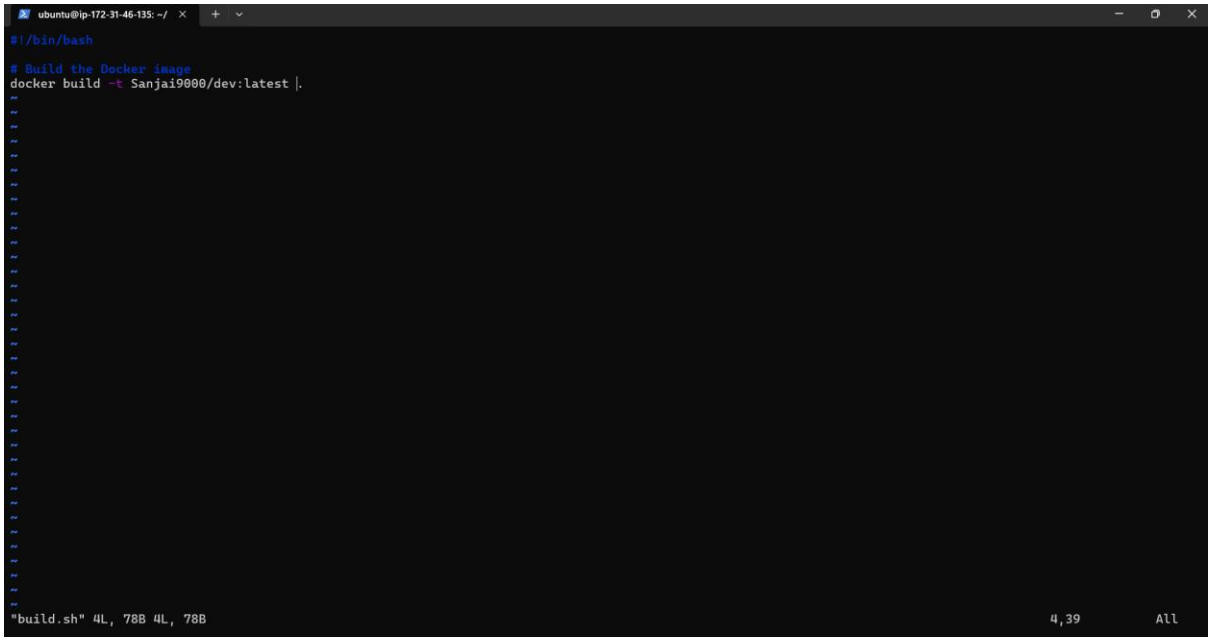
- Below is the docker-compose.yaml file.

```
version: '3'
services:
  app:
    build: .
    ports:
      - "80:80"
~
"docker-compose.yaml" 6L, 70B                                        6,15          All
```

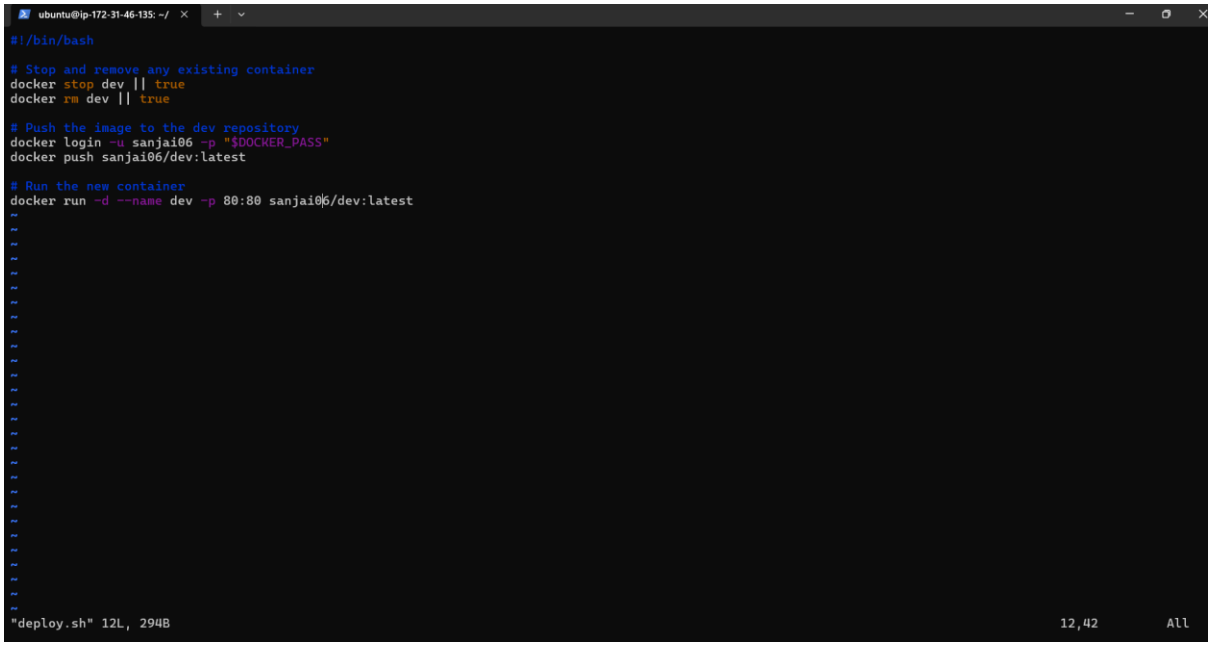
Bash Scripting:

Here are the created 2 scripts

- build.sh - for building docker images and given chmod +x build.sh to execute the file we can use the command ‘./build.sh’

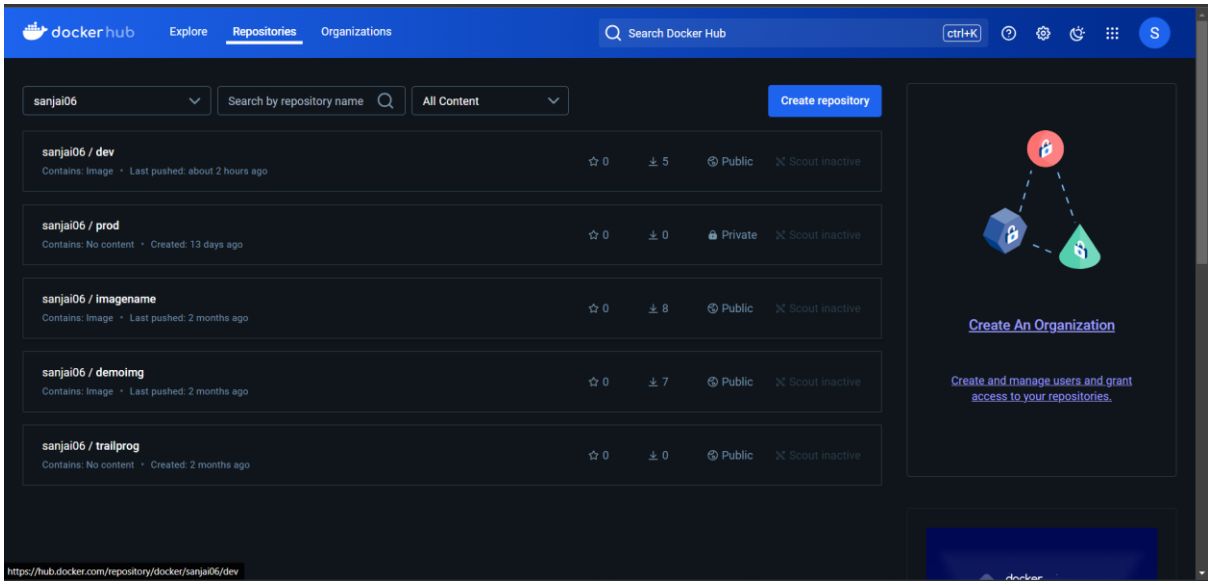


- deploy.sh - for deploying the image to the server , chmod +x deploy.sh and ./deploy.sh



Docker hub:

- Created 2 repos “dev” and “prod” to push images with the rules “Prod” repo must be private and “dev” repo can be public.



Jenkins:

- Install and configure jenkins build step as per needs to build, push and deploy the application:

```
sudo apt update
sudo apt install openjdk-17-jdk -y
java -version
```

- Jenkins is not included in the default Ubuntu repositories, so you'll need to add the official Jenkins repository:

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

echo deb [signed-by=/usr/share/keyrings/jenkins-
keyring.asc] \https://pkg.jenkins.io/debian binary/
| sudo tee \/etc/apt/sources.list.d/jenkins.list >
/dev/null
```

- Installing Jenkinks

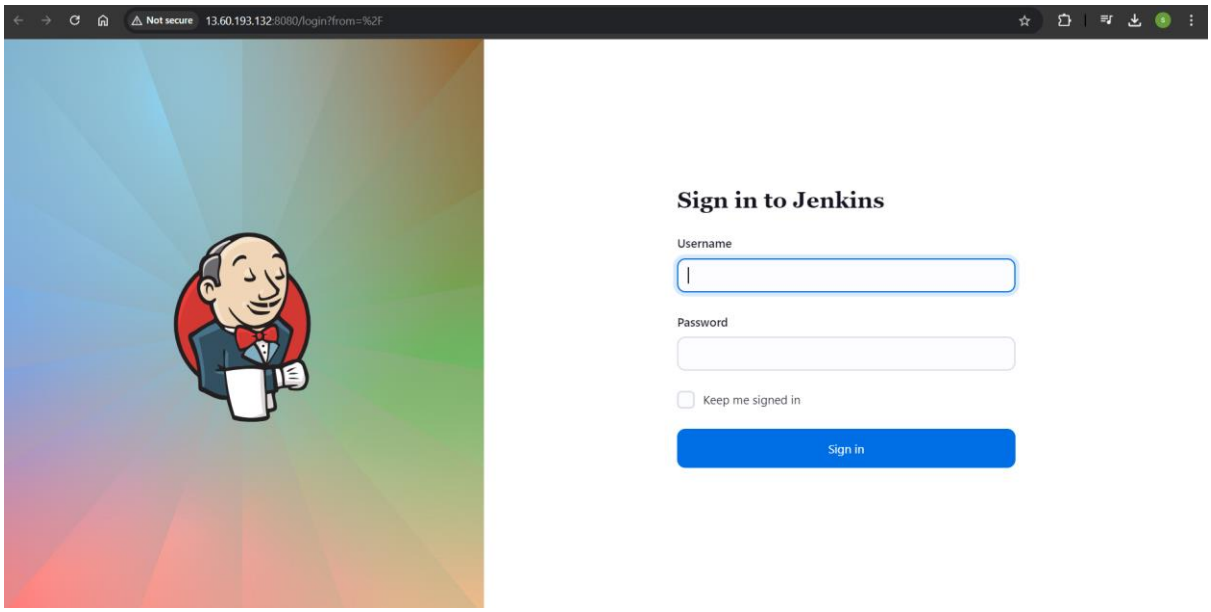
```
sudo apt update
sudo apt install jenkins -y
sudo systemctl start Jenkins
sudo systemctl enable Jenkins
sudo systemctl status Jenkins
```

```
ubuntu@ip-172-31-46-135:~/Guvi-project$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-08-30 12:34:37 UTC; 1min 28s ago
   Main PID: 7550 (java)
      Tasks: 38 (limit: 1078)
     Memory: 285.6M (peak: 327.1M)
        CPU: 29.006s
     CGroup: /system.slice/jenkins.service
             └─7550 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Aug 30 12:34:31 ip-172-31-46-135 jenkins[7550]: 2bb9f2fc512c4cddbf50693dd079bd09
Aug 30 12:34:31 ip-172-31-46-135 jenkins[7550]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Aug 30 12:34:31 ip-172-31-46-135 jenkins[7550]: ************************************************************
Aug 30 12:34:31 ip-172-31-46-135 jenkins[7550]: ************************************************************
Aug 30 12:34:31 ip-172-31-46-135 jenkins[7550]: ************************************************************
Aug 30 12:34:37 ip-172-31-46-135 jenkins[7550]: 2024-08-30 12:34:37.655+0000 [id=30]        INFO        jenkins.InitReactorRunner$1#onAttained: Completed i>
Aug 30 12:34:37 ip-172-31-46-135 jenkins[7550]: 2024-08-30 12:34:37.672+0000 [id=23]        INFO        hudson.lifecycle.Lifecycle#onReady: Jenkins is full>
Aug 30 12:34:37 ip-172-31-46-135 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Aug 30 12:34:38 ip-172-31-46-135 jenkins[7550]: 2024-08-30 12:34:38.110+0000 [id=48]        INFO        h.m.DownloadService$Downloadable#load: Obtained the>
Aug 30 12:34:38 ip-172-31-46-135 jenkins[7550]: 2024-08-30 12:34:38.111+0000 [id=48]        INFO        hudson.util.Retrier#start: Performed the action che>
lines 1-20/20 (END)
```

- To Access Jenkins complete the installation, you need to set up Jenkins by accessing it in a web browser. Open your web browser and navigate to `http://<your_server_ip>:8080`, here is my Jenkins URL http://13.60.193.132:8080/
- You'll see the "Unlock Jenkins" screen. To retrieve the initial password, run:

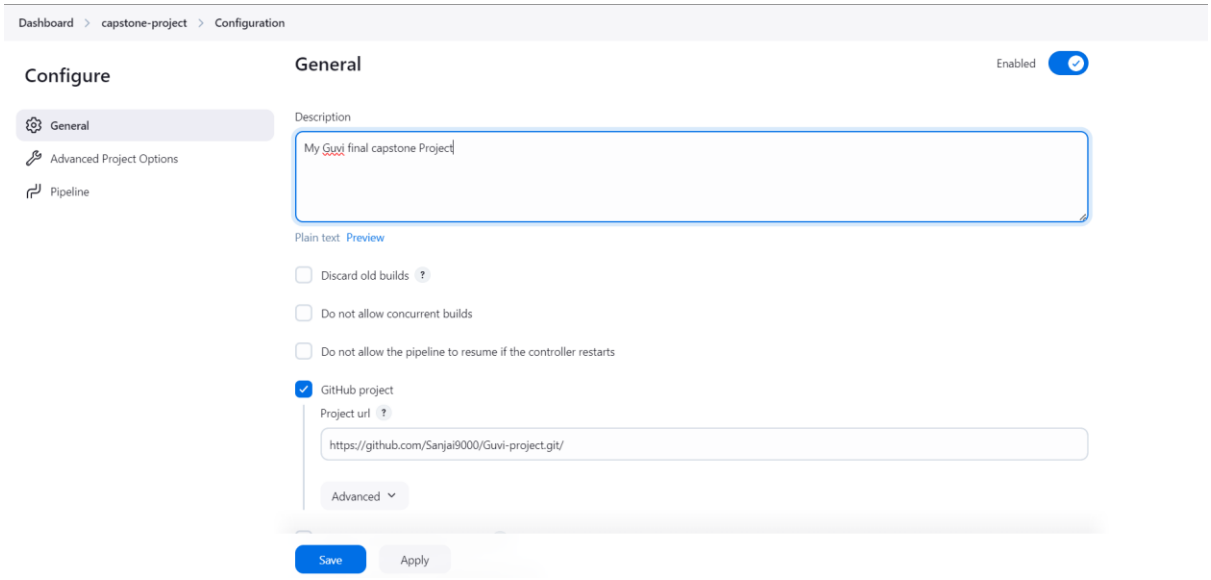
```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

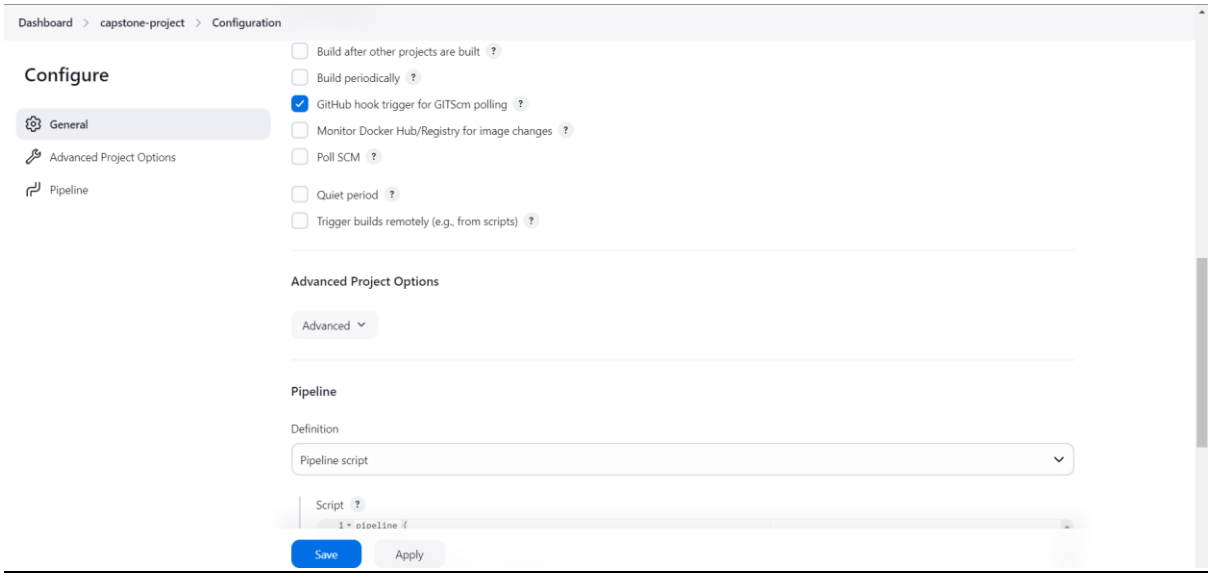


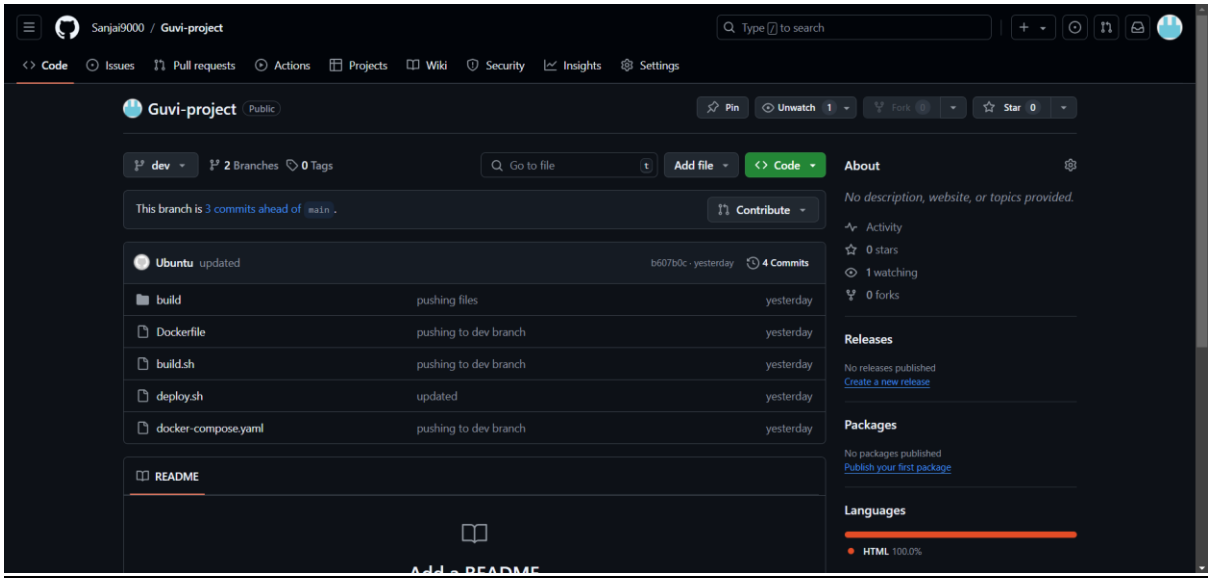
- Now checked and installed the required plugins for Github, Docker and Dockerhub. And also provided the required credentials in Jenkins. Jenkins dashboard->manage Jenkins->security->credentials->global

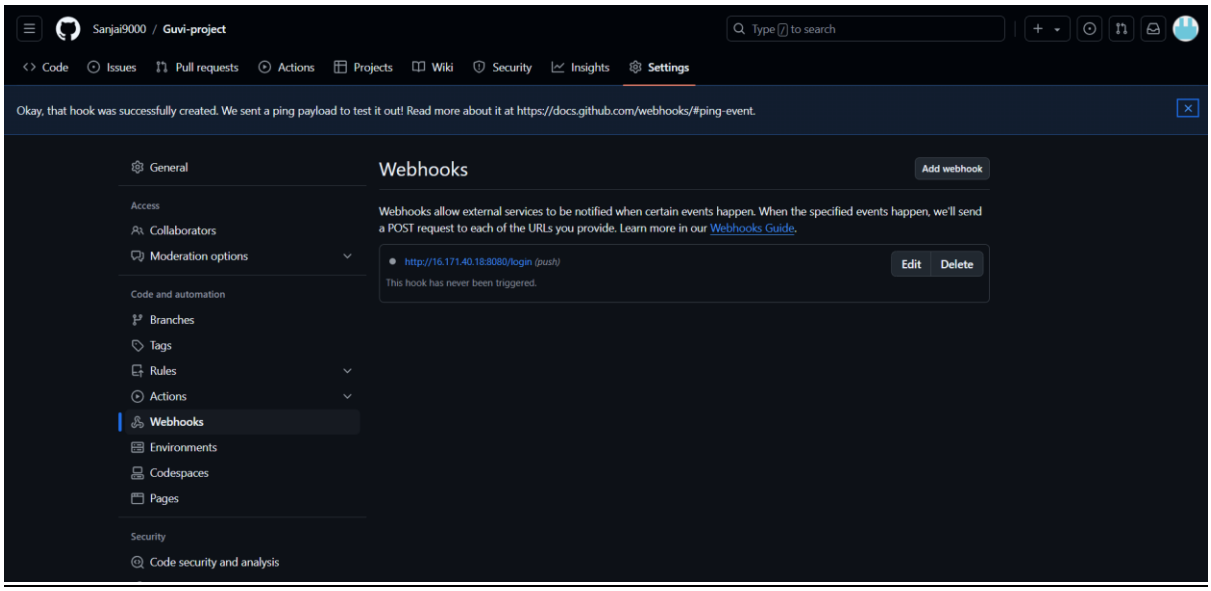
- To connect Jenkins to the Github repo with auto build trigger from both “dev” and master branch we have to create a new job in the Jenkins. Navigate to the Jenkins dashboard on left ->select ‘New Item’ and give a name for the pipeline->select item type as Pipeline-> OK to save the new pipeline.

- After clicking ‘Ok’ we will be lead to new page called configuration where we ca see all the settings under->General.

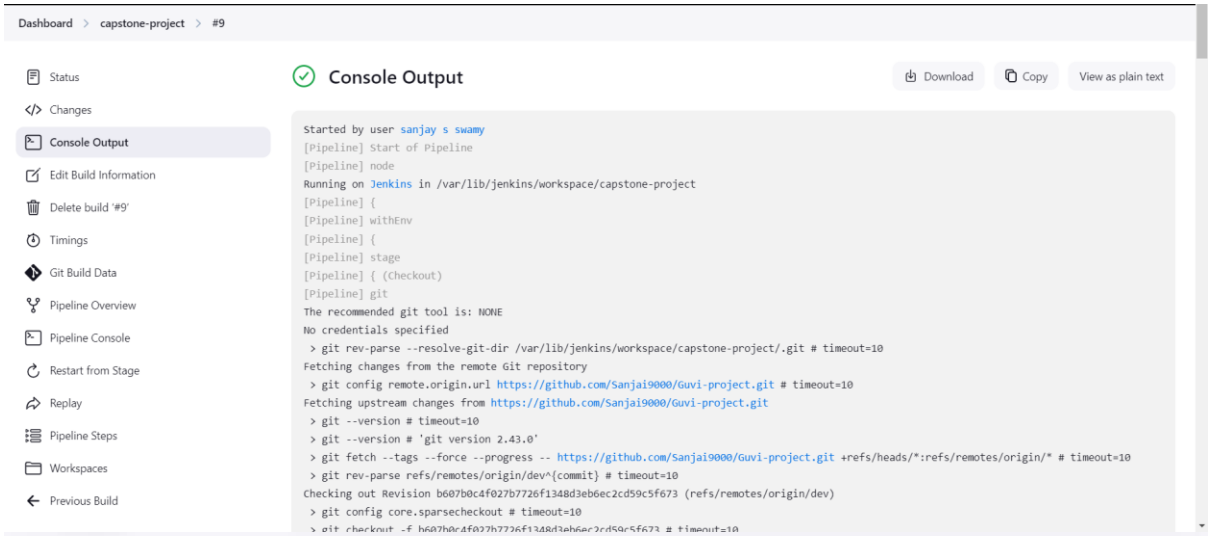

Dashboard > capstone-project > Configuration
Configure
General
Advanced Project Options
Pipeline
General
Enabled
Description
My Guvi final capstone Project
Plain text Preview
Discard old builds
Do not allow concurrent builds
Do not allow the pipeline to resume if the controller restarts
GitHub project
Project url
https://github.com/Sanjai9000/Guvi-project.git/
Advanced
Save
Apply

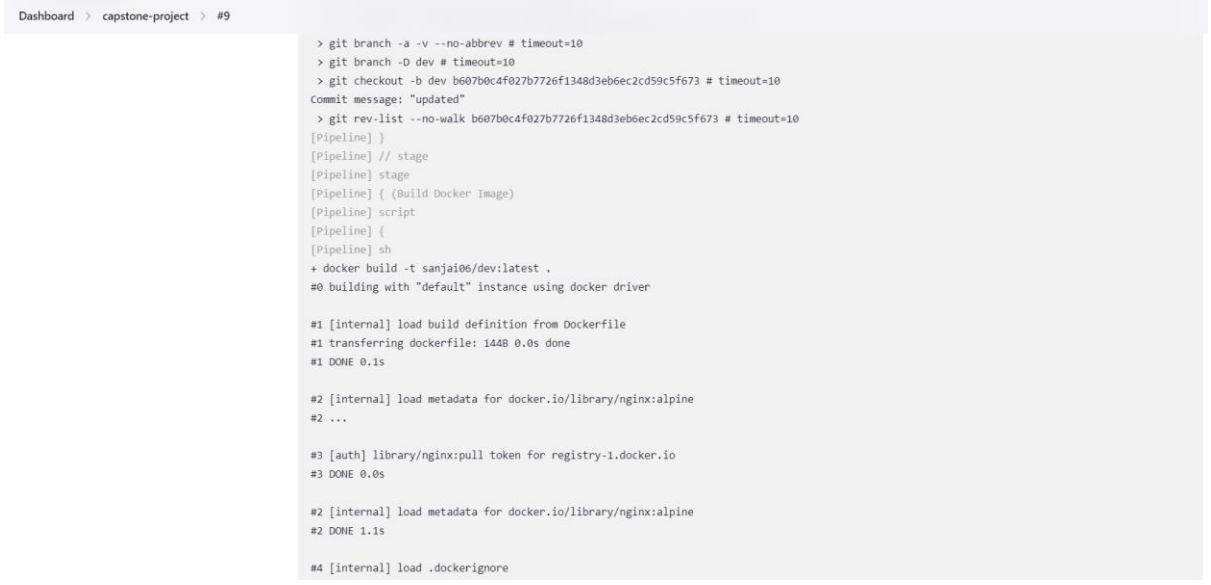

Dashboard > capstone-project > Configuration
Configure
General
Advanced Project Options
Pipeline
Build after other projects are built
Build periodically
GitHub hook trigger for GITScm polling
Monitor Docker Hub/Registry for image changes
Poll SCM
Quiet period
Trigger builds remotely (e.g., from scripts)
Advanced Project Options
Advanced
Pipeline
Definition
Pipeline script
Script
Save
Apply



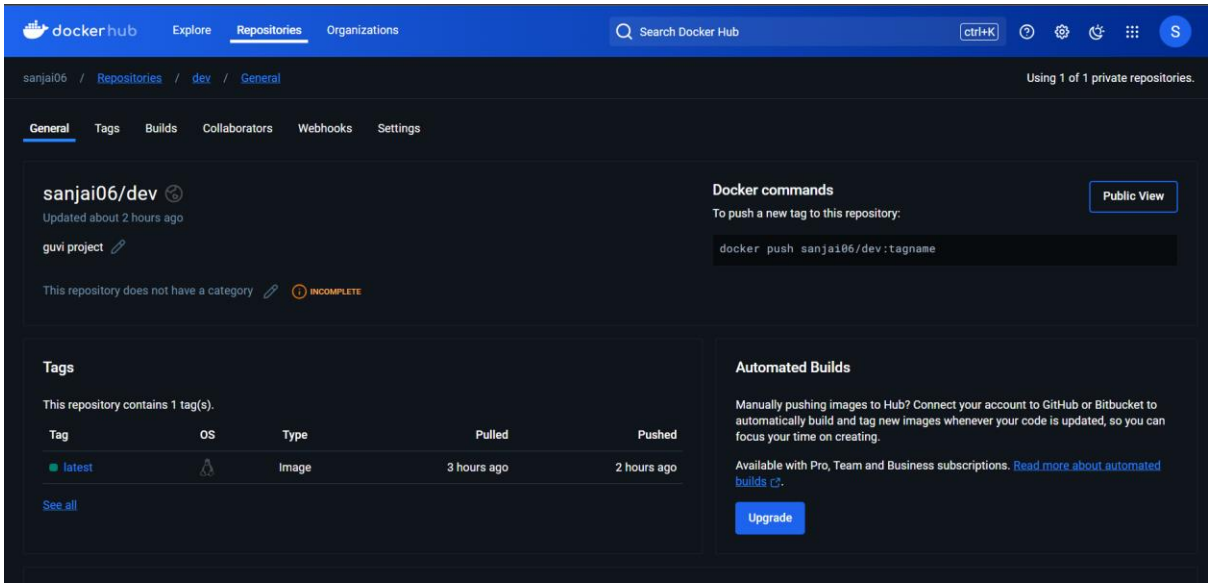


- Once code pushed to “dev” branch, docker image must build and pushed to dev repo in docker hub


Dashboard > capstone-project > #9
Status
Changes
Console Output
Edit Build Information
Delete build '#9'
Timings
Git Build Data
Pipeline Overview
Pipeline Console
Restart from Stage
Replay
Pipeline Steps
Workspaces
Previous Build
Console Output
Download
Copy
View as plain text
Started by user sanjay s swamy
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/capstone-project
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/capstone-project/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Sanjai9000/Guvi-project.git # timeout=10
Fetching upstream changes from https://github.com/Sanjai9000/Guvi-project.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/Sanjai9000/Guvi-project.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/dev^{commit} # timeout=10
Checking out Revision b607b0c4f027b7726f1348d3eb6ec2cd59c5f673 (refs/remotes/origin/dev)
> git config core.sparsecheckout # timeout=10

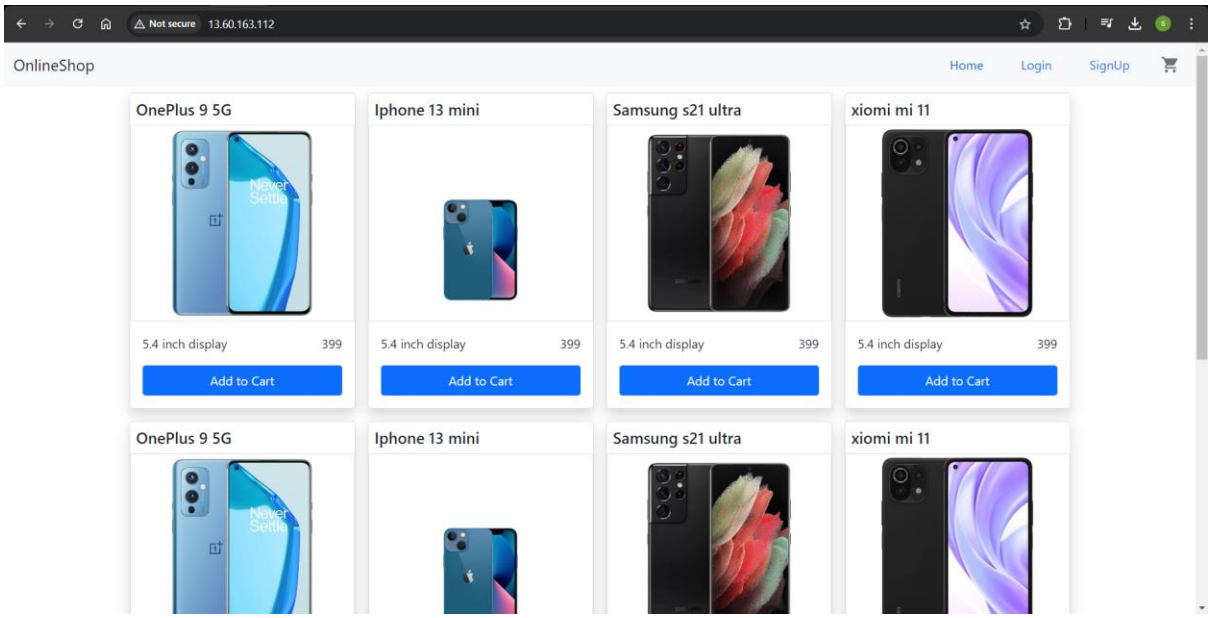

Dashboard > capstone-project > #9
> git branch -a -v --no-abbrev # timeout=10
> git branch -D dev # timeout=10
> git checkout -b dev b607b0c4f027b7726f1348d3eb6ec2cd59c5f673 # timeout=10
Commit message: "updated"
> git rev-list --no-walk b607b0c4f027b7726f1348d3eb6ec2cd59c5f673 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build Docker Image)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ docker build -t sanjai06/dev:latest .
#0 building with "default" instance using docker driver
#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 144B 0.0s done
#1 DONE 0.1s
#2 [internal] load metadata for docker.io/library/nginx:alpine
#2 ...
#3 [auth] library/nginx:pull token for registry-1.docker.io
#3 DONE 0.0s
#2 [internal] load metadata for docker.io/library/nginx:alpine
#2 DONE 1.1s
#4 [internal] load .dockerignore


Dashboard > capstone-project > #9
c028c01f43bc: Layer already exists
9a2d14b22cbe: Layer already exists
b65aff7ee426: Layer already exists
78561cef0761: Layer already exists
latest: digest: sha256:3ad3ee66e448e111a72b05895b225d1eb7590fc6ac7cb2c24d0bfb9f899b750a size: 2405
b6b1a4967c96f0b4451429a647e4748fd45b6489eda97db9fbbdf8dfa8d0b741
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] sh
+ docker system prune -f
Deleted build cache objects:
3p0tynlonu1spgdw7hk9vvi8o
j0wwafbjuk2e4xdkrd28e9hmr
vi108ebacxjcoyzco05mhg0ml
Total reclaimed space: 2.604MB
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS



- Once the image is build and deployed we can see that the image is pushed to docker hub, and to see the site deployed edit the inbound rules and add port 80.

- In the browser using the Public IP along with the port 80 we can see the application running-> React App ->13.60.163.112:80



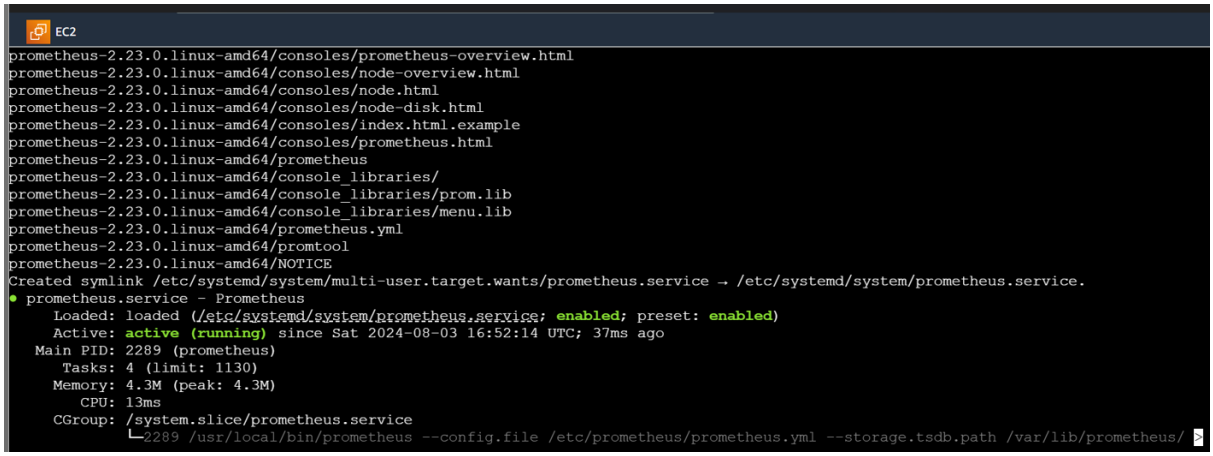
Monitoring:

- For setting up a monitoring system to check the health status of the application I have used Prometheus. Firstly, ensure your system is up to date:

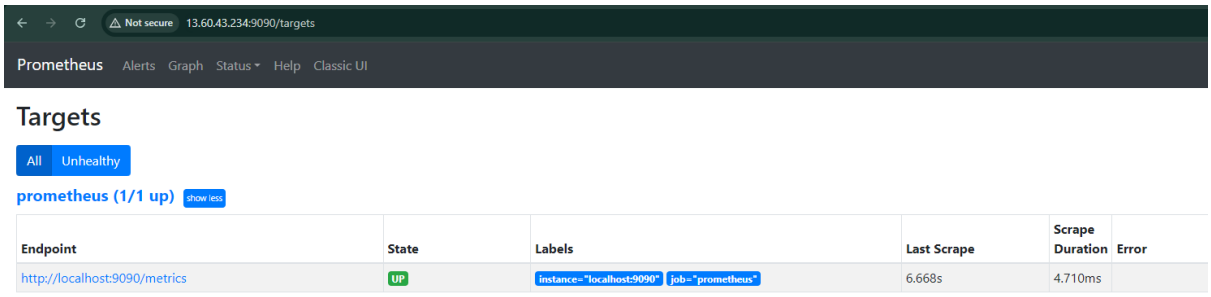
```
sudo apt-get update
```

- Prometheus can be installed by downloading the binary files or using Docker. Here, we have download the binary:

```
cd /opt/
sudo wget https://github.com/prometheus/prometheus/releases/download/v2.44.0/prometheus-2.44.0.linux-amd64.tar.gz
sudo tar -xvf prometheus-2.44.0.linux-amd64.tar.gz
sudo mv prometheus-2.44.0.linux-amd64 prometheus
cd Prometheus
```



- To access Prometheus we have to enable port number 9090 and as we have not assigned any target machine yet it will monitor the same running Monitor machine for now.

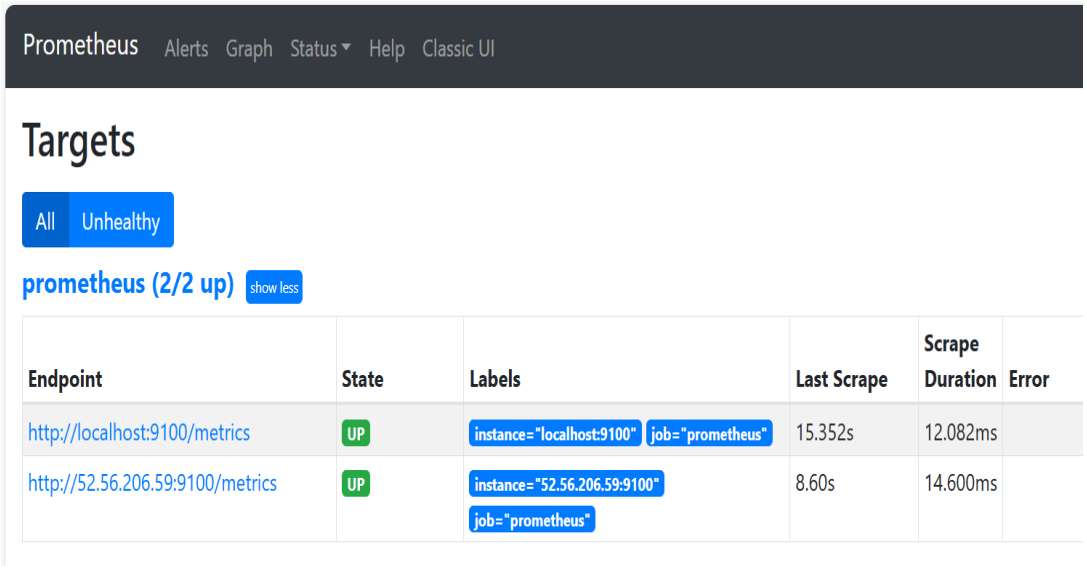


- Once Prometheus is installed, configured and running we have to use Node exporter. Node Exporter is used to collect system-level metrics. To install it:

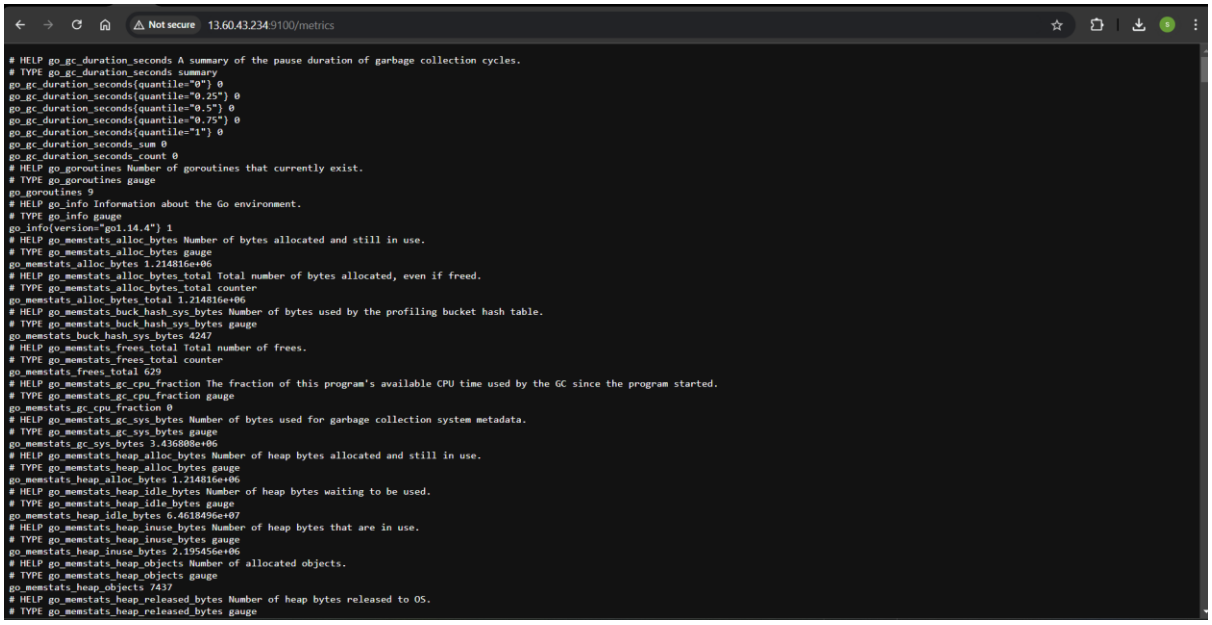
```
cd /opt/
sudo wget
https://github.com/prometheus/node_exporter/releases/download/v1.6.1/node_exporter-1.6.1.linux-amd64.tar.gz
sudo tar -xvf node_exporter-1.6.1.linux-amd64.tar.gz
sudo mv node_exporter-1.6.1.linux-amd64 node_exporter
cd node_exporter
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/node-exporter.service → /etc/systemd/system/node-exporter.service.
● node-exporter.service - Prometheus Node Exporter Service
     Loaded: loaded (/etc/systemd/system/node-exporter.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-08-03 17:05:59 UTC; 35ms ago
   Main PID: 2503 (node_exporter)
      Tasks: 3 (limit: 1130)
     Memory: 1.7M (peak: 2.0M)
        CPU: 8ms
     CGroup: /system.slice/node-exporter.service
             └─2503 /usr/local/bin/node_exporter
```

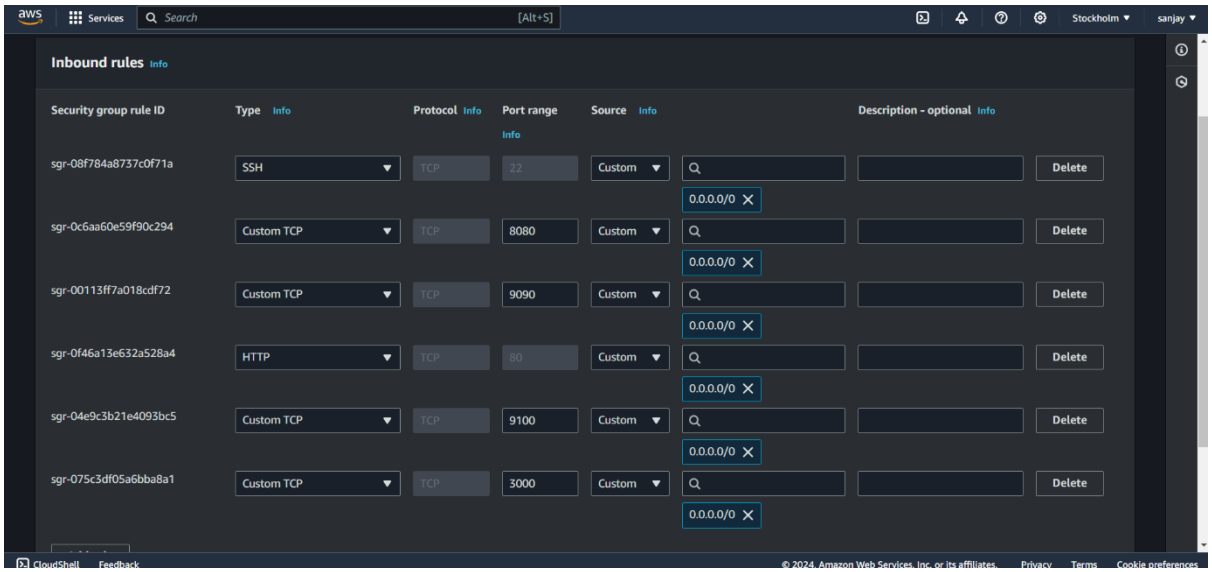
- Now we can also see in the Prometheus the created Monitoring and target machine matrices are visible.



- Inorder to access the Node Exporter we have to enable the port number 9100. It is now collecting the default matrics of the running machine.



- Below are the ports that have been accessed in the main instance for the Project.



Submission:

- Github repo URL :
- Deployed site URL : React App ->13.60.163.112:80
- Docker images name : sanjai06/dev:latest