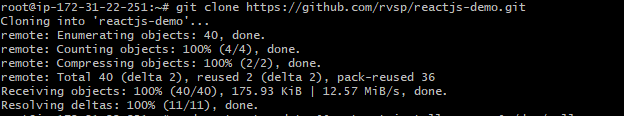
1.Application:

Clone the below mentioned repo and deploy the application. (Run the application in port 80[HTTP])

Repo URL: <https://github.com/rvsp/reactjs-demo.git>

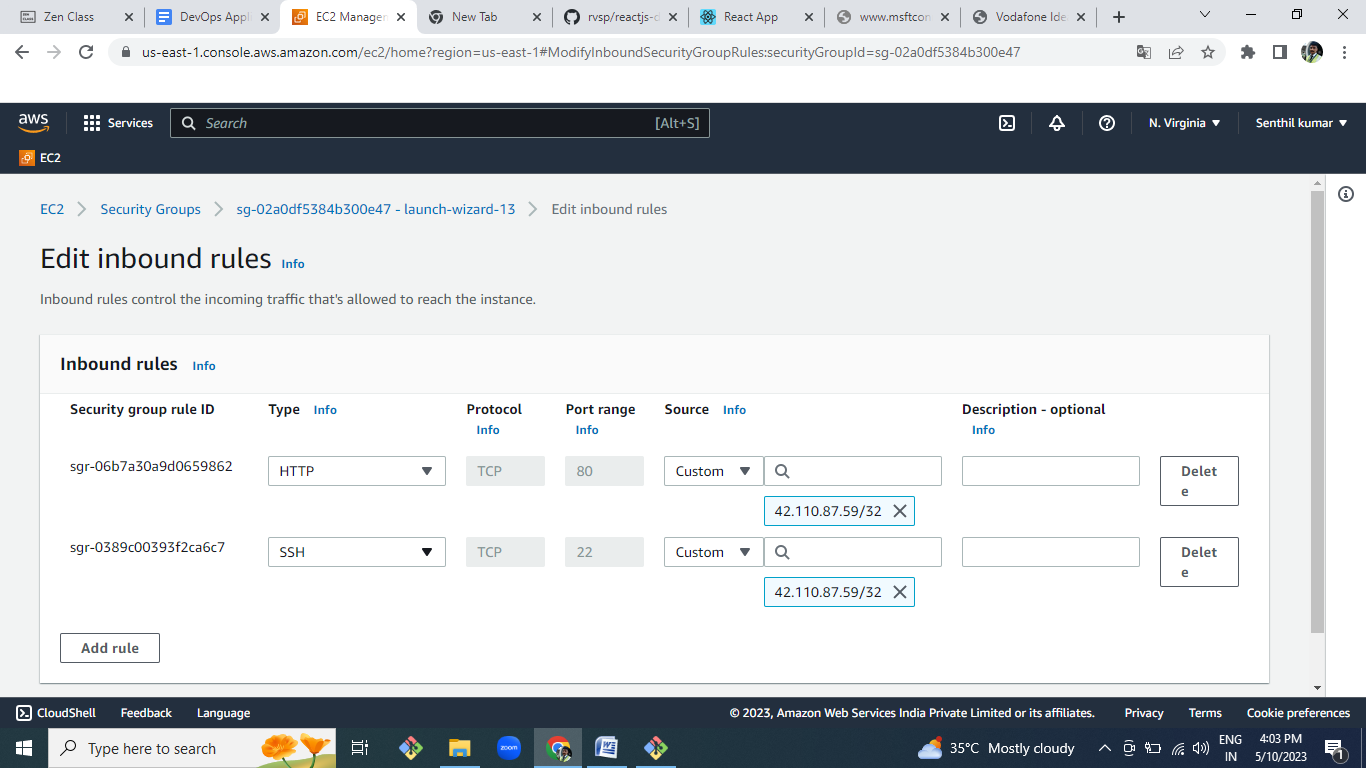
Git clone



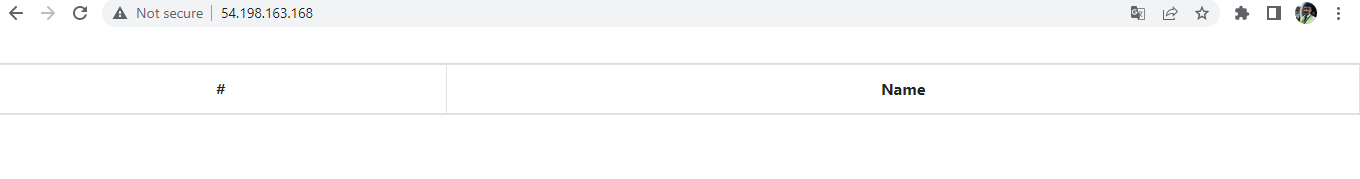
Artiacts



SG-(HTTP ) PORT 80



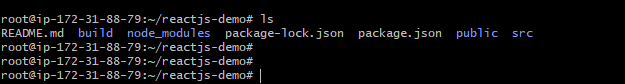
APPLICATION DEPLOYED

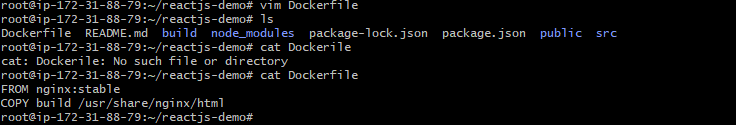


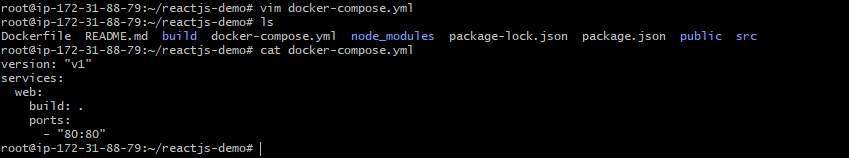
2.Docker:

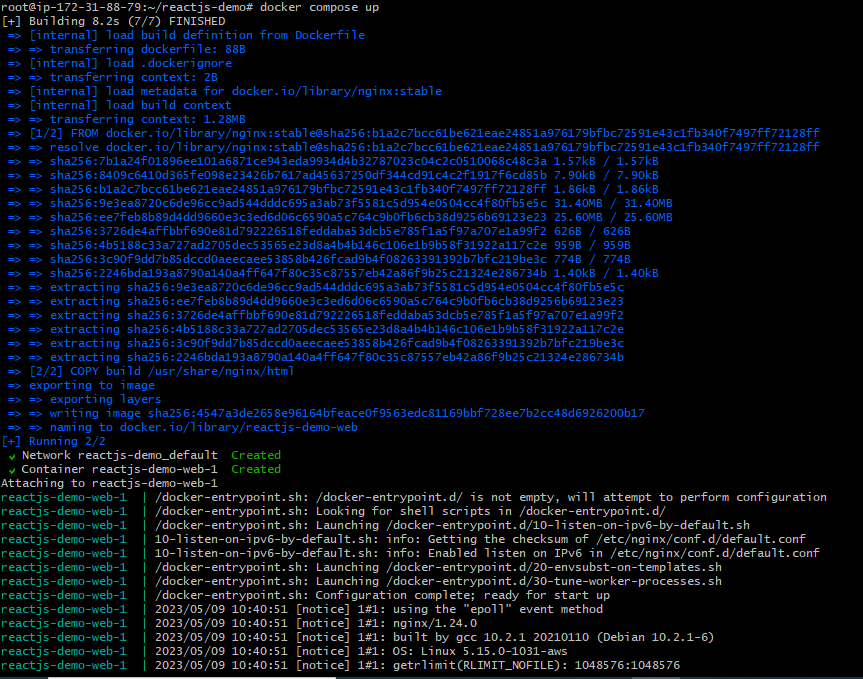
Dockerize the application by creating a Dockerfile

Create a docker-compose file to use the above image











3.**Bash Scripting:**

**Write 2 scripts**

**build.sh for building docker images**

**deploy.sh - for deploying the image to server**

#!/bin/bash

sudo apt-get update &&>/dev/null

sudo git clone https://github.com/rvsp/reactjs-demo.git

sudo apt-get update && apt-get install npm -y &>/dev/null

cd reactjs-demo

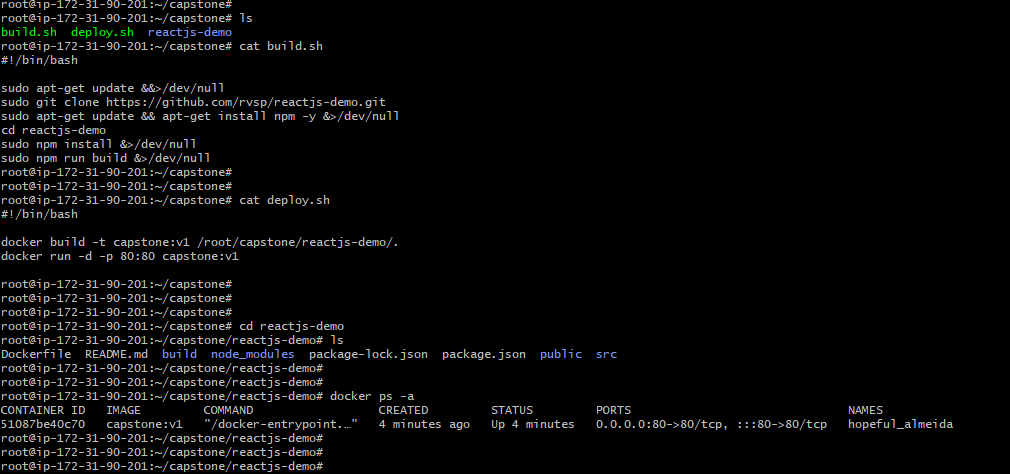
sudo npm install &>/dev/null

sudo npm run build &>/dev/null

#!/bin/bash

docker build -t capstone:v1 .

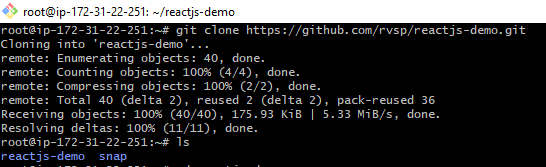
docer run -d -p 80:80 capstone:v1



**Version Control:**

**Push the code to github to dev branch (use dockerignore & gitignore files)**

**Note: Use only CLI for related git commands**

****

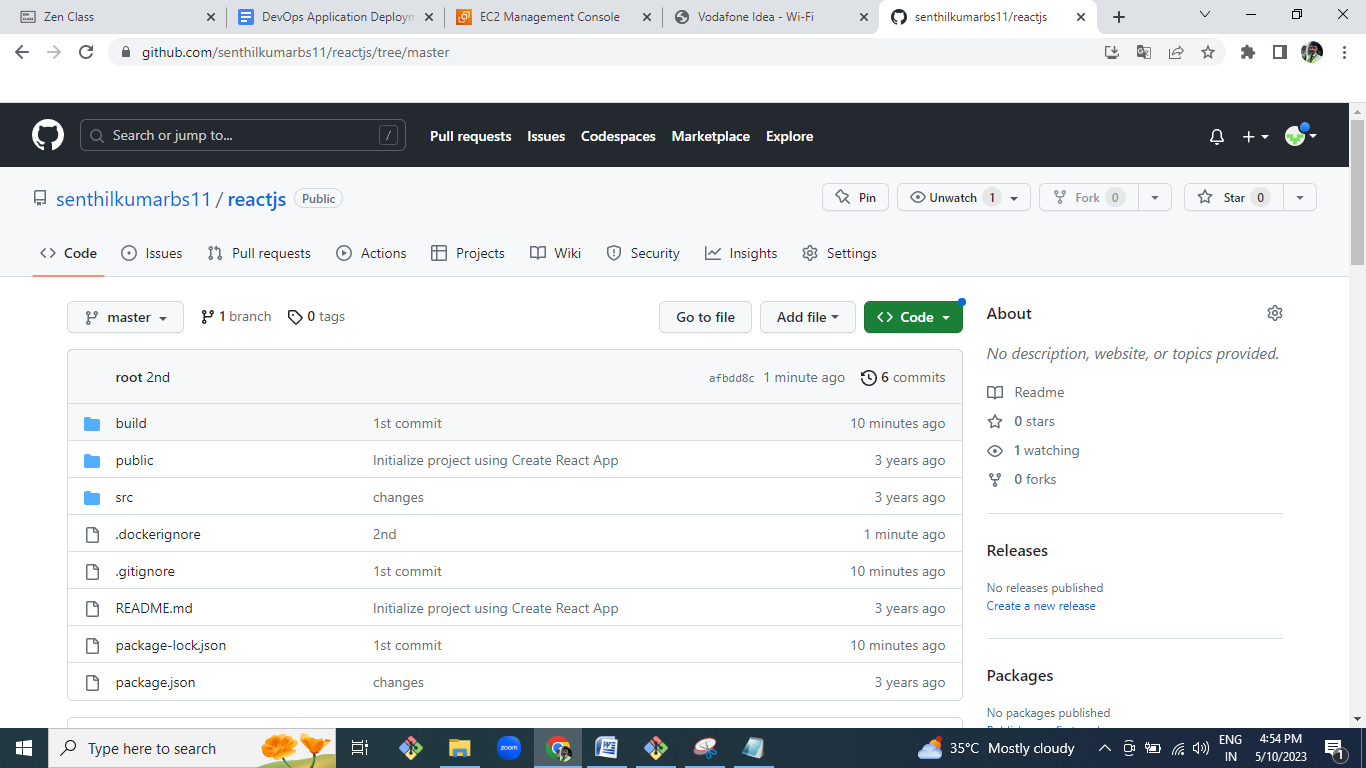
****

****

****

****

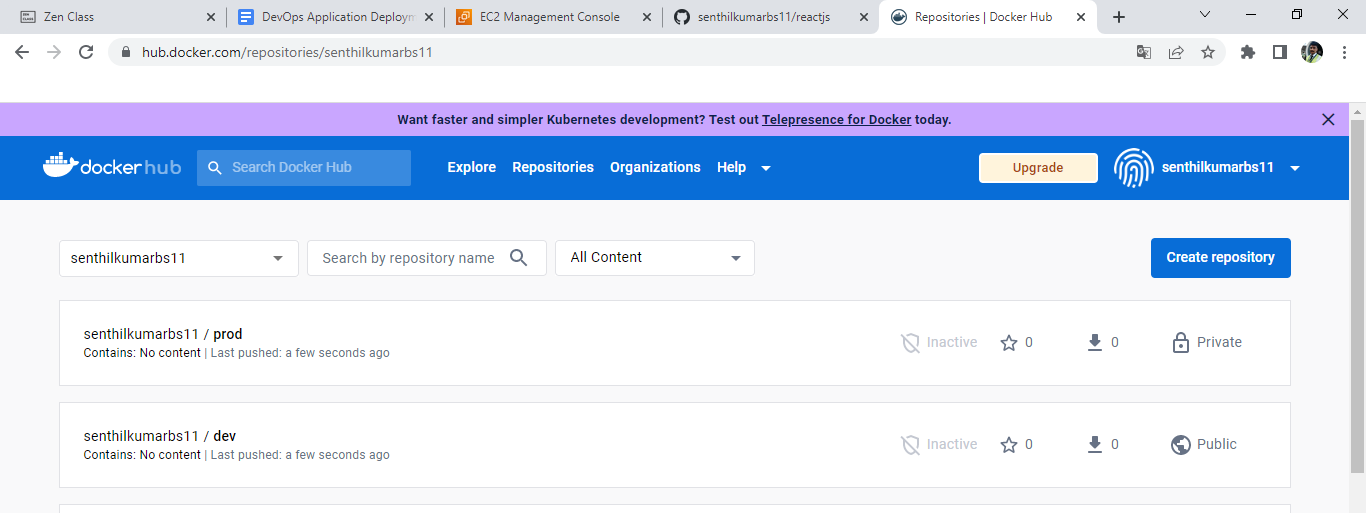
****

****

**Docker hub:**

**• Create 2 repos "dev" and "prod" to push images.**

**"Prod" repo must be private and "dev" repo can be public**



**Jenkins:**

Install and configure jenkins build step as per needs to build, push & deploy the

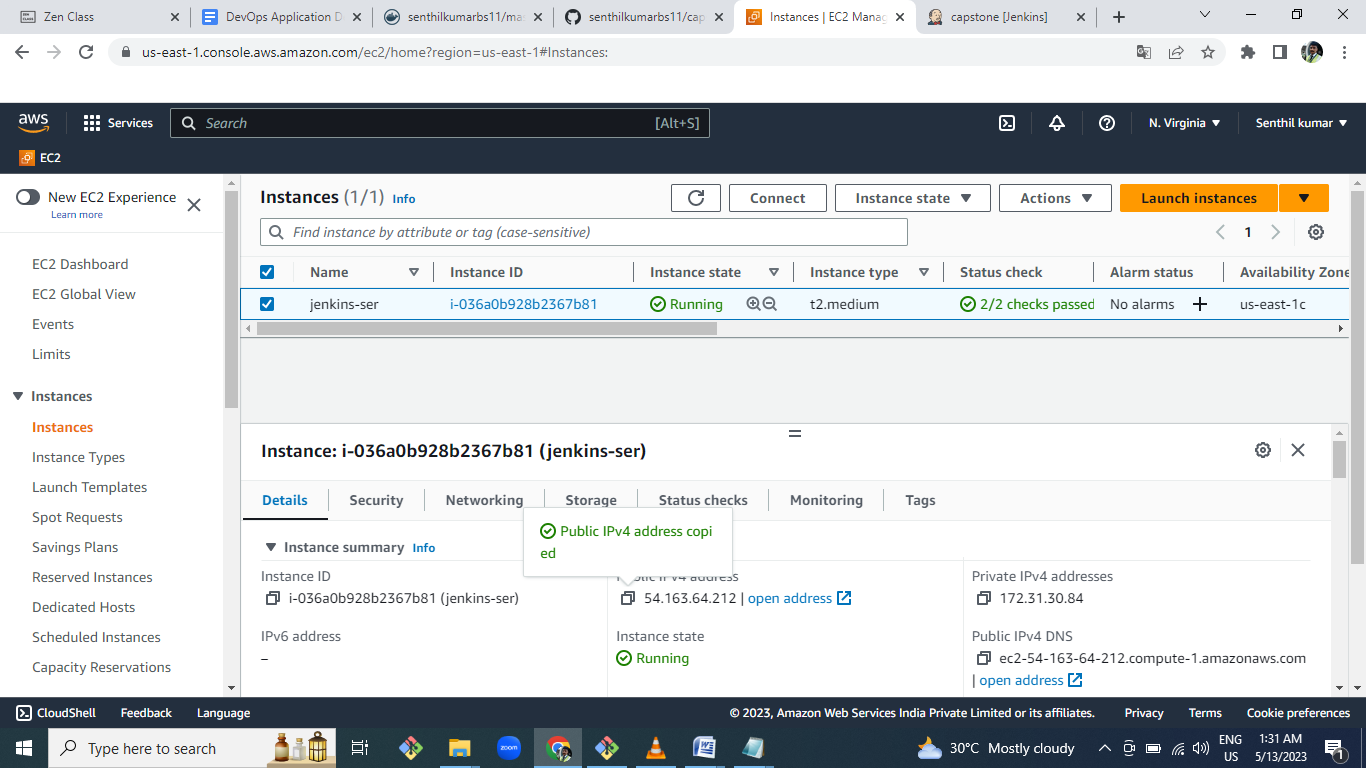
application

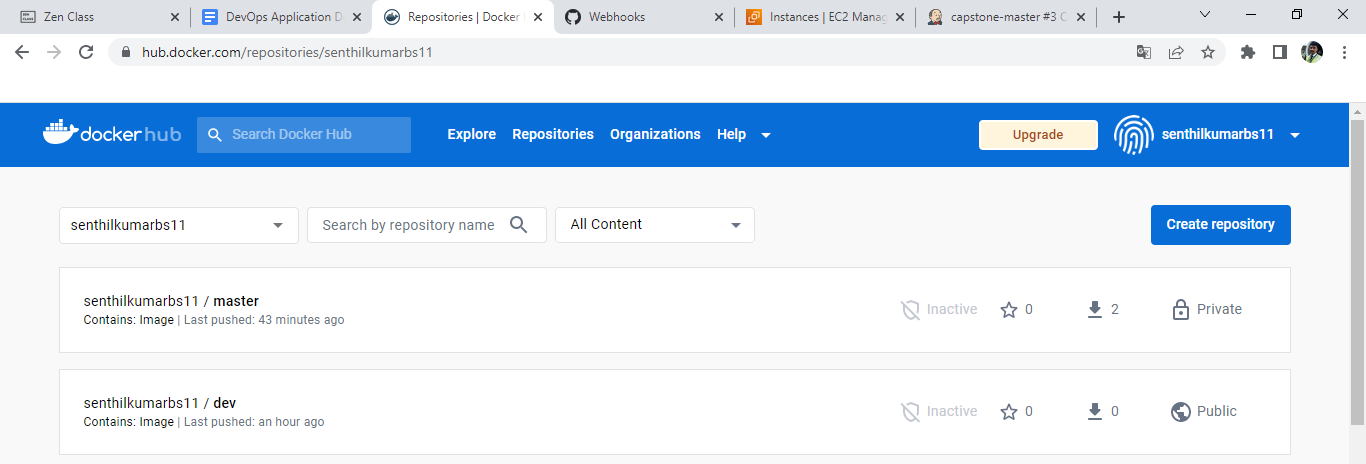
• Connect jenkins to the github repo with auto build trigger from both dev & master branch

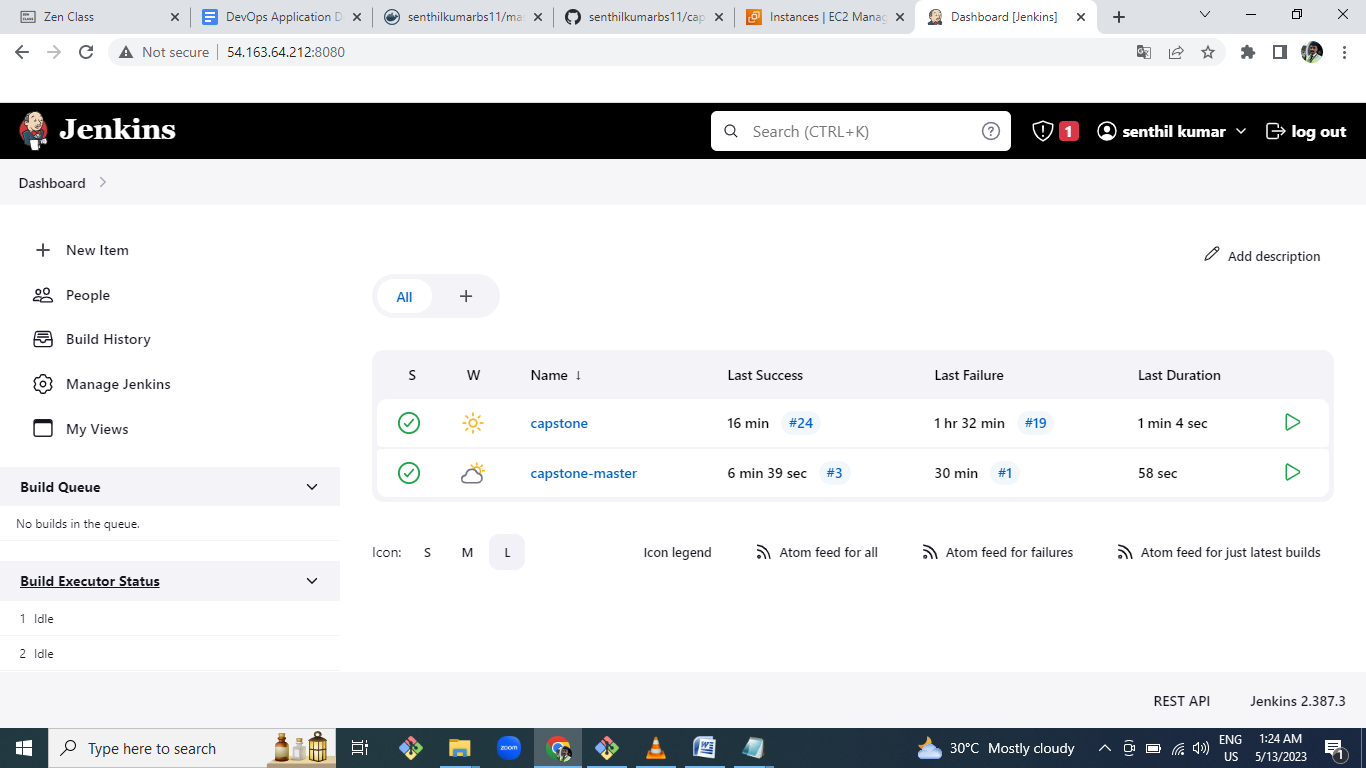
• If code pushed to dev branch, docker image must build and pushed to dev repo in docker

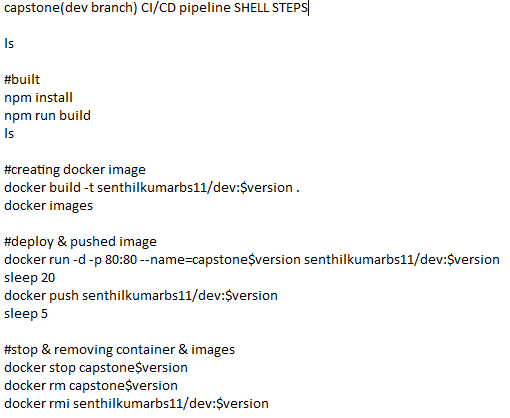
hub

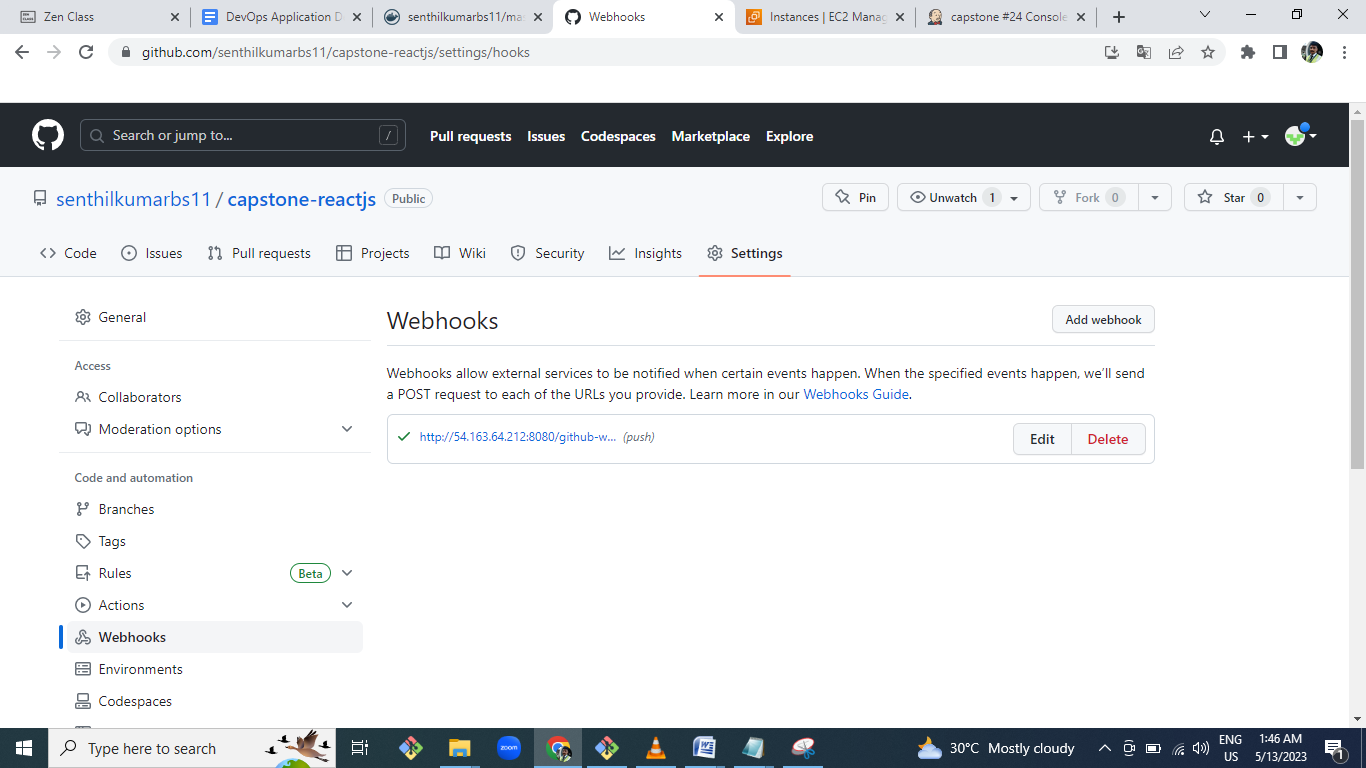
• If dev merged to master, then docker image must be pushed to prod repo in docker hub

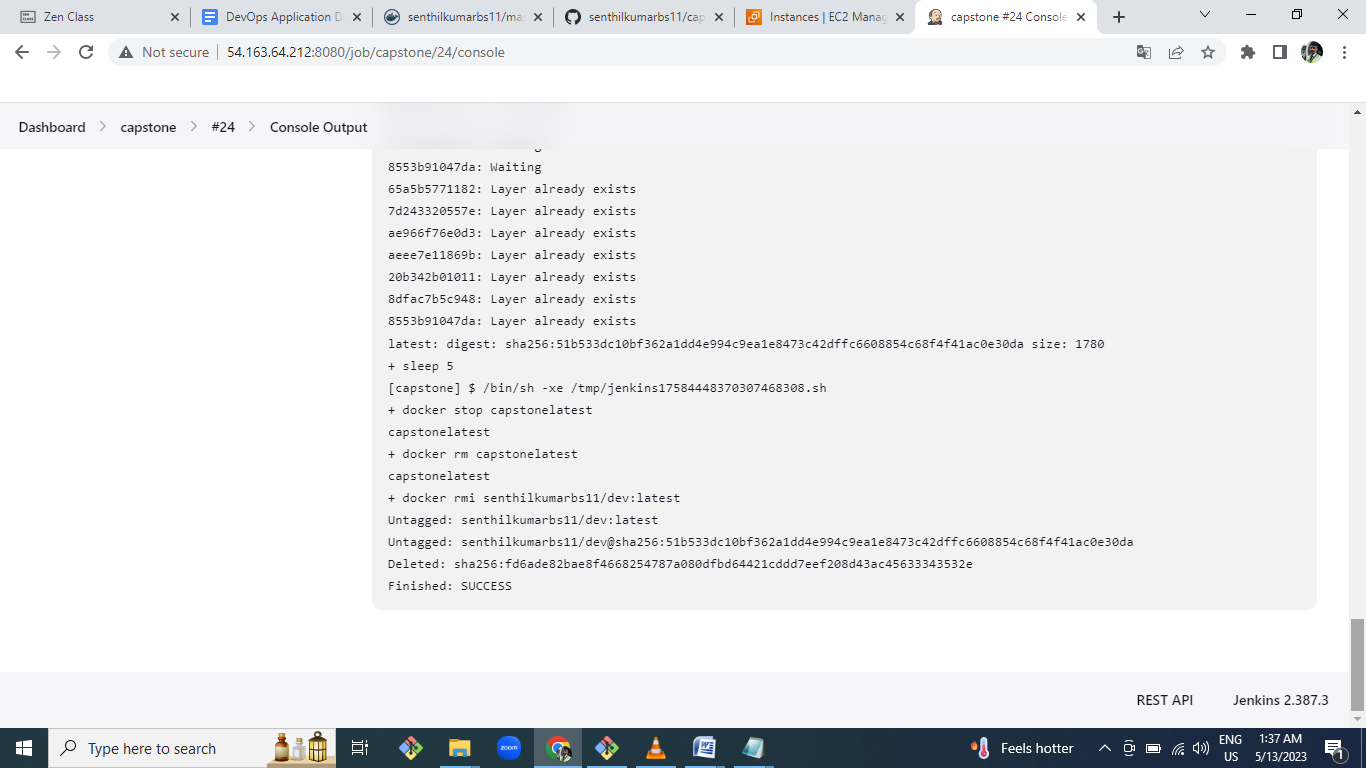


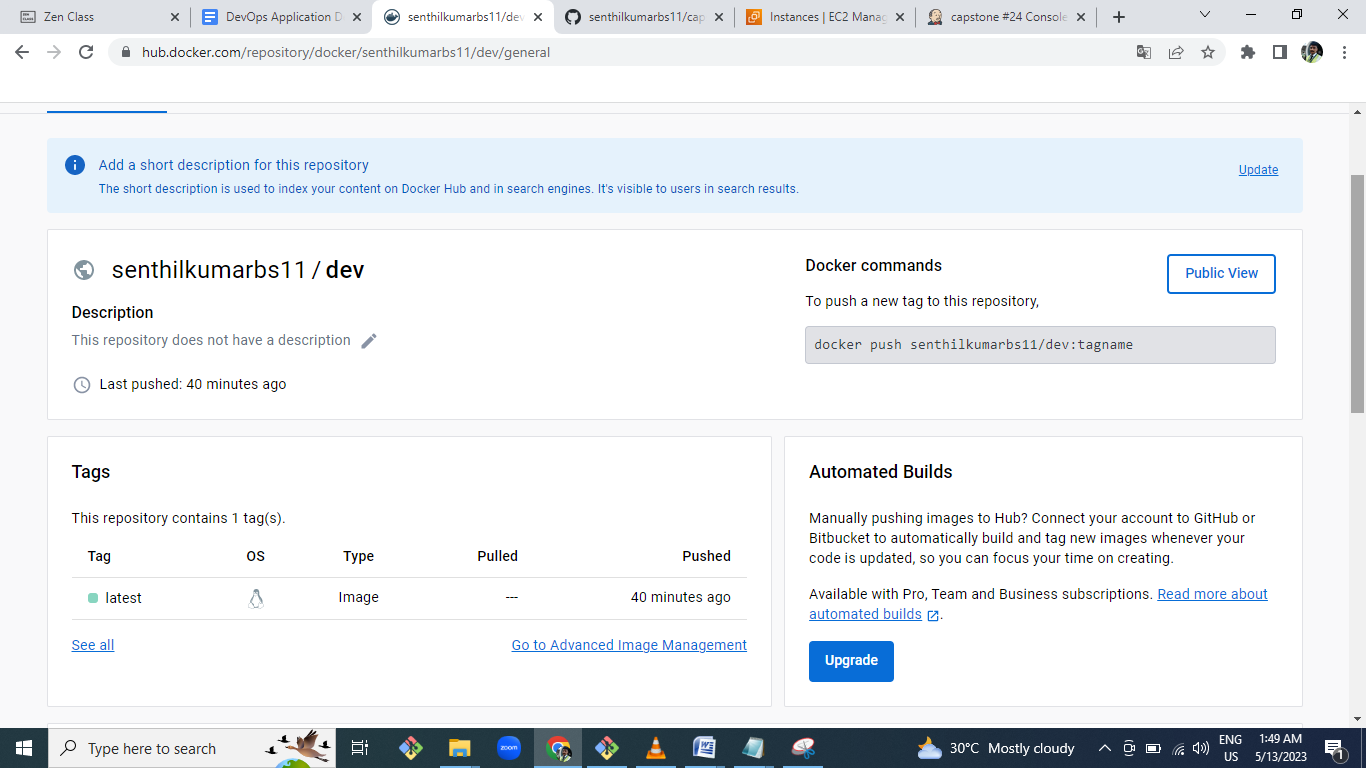


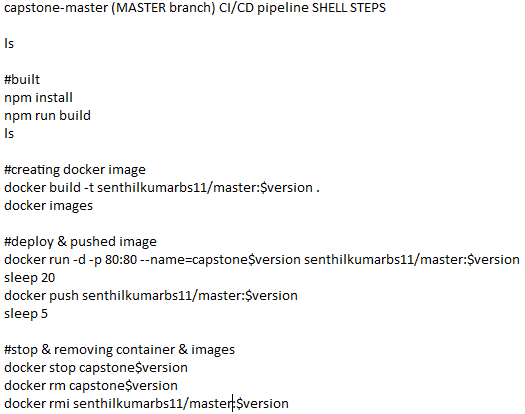


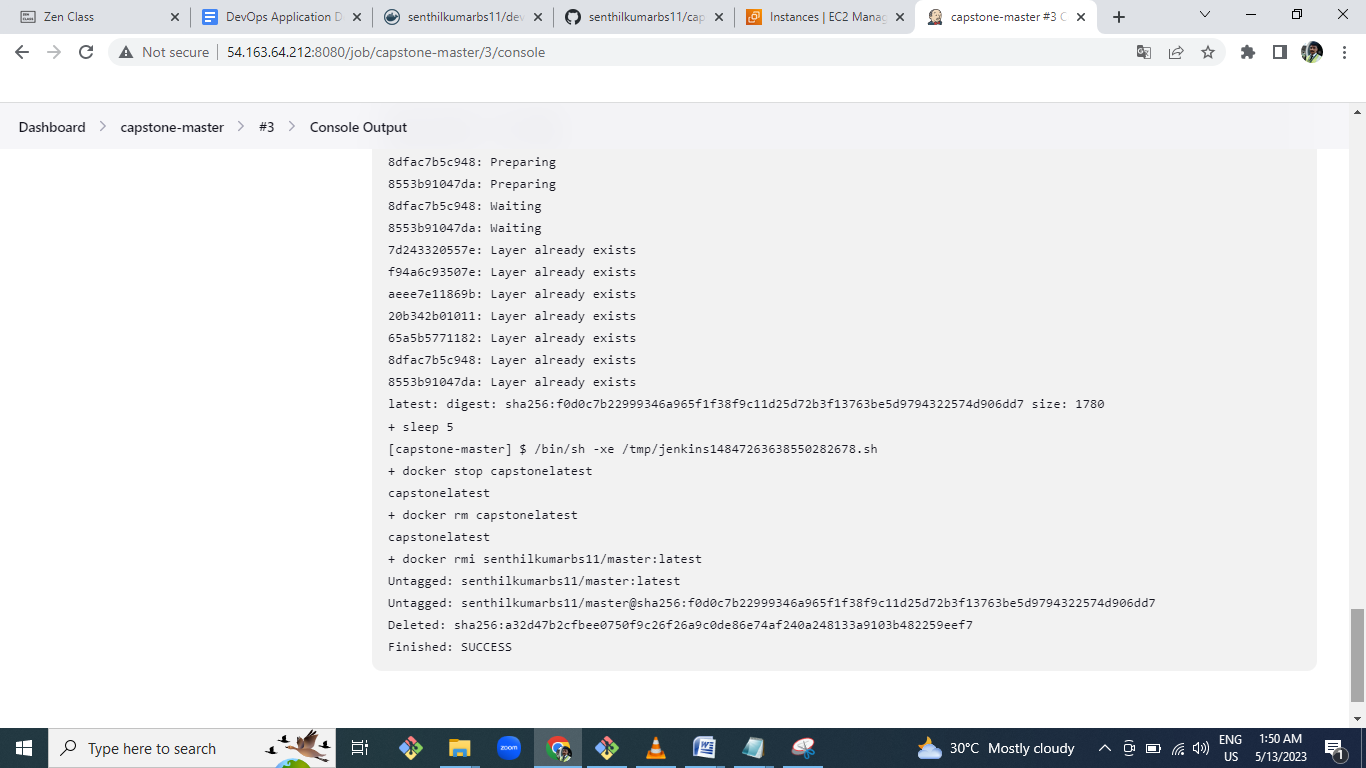


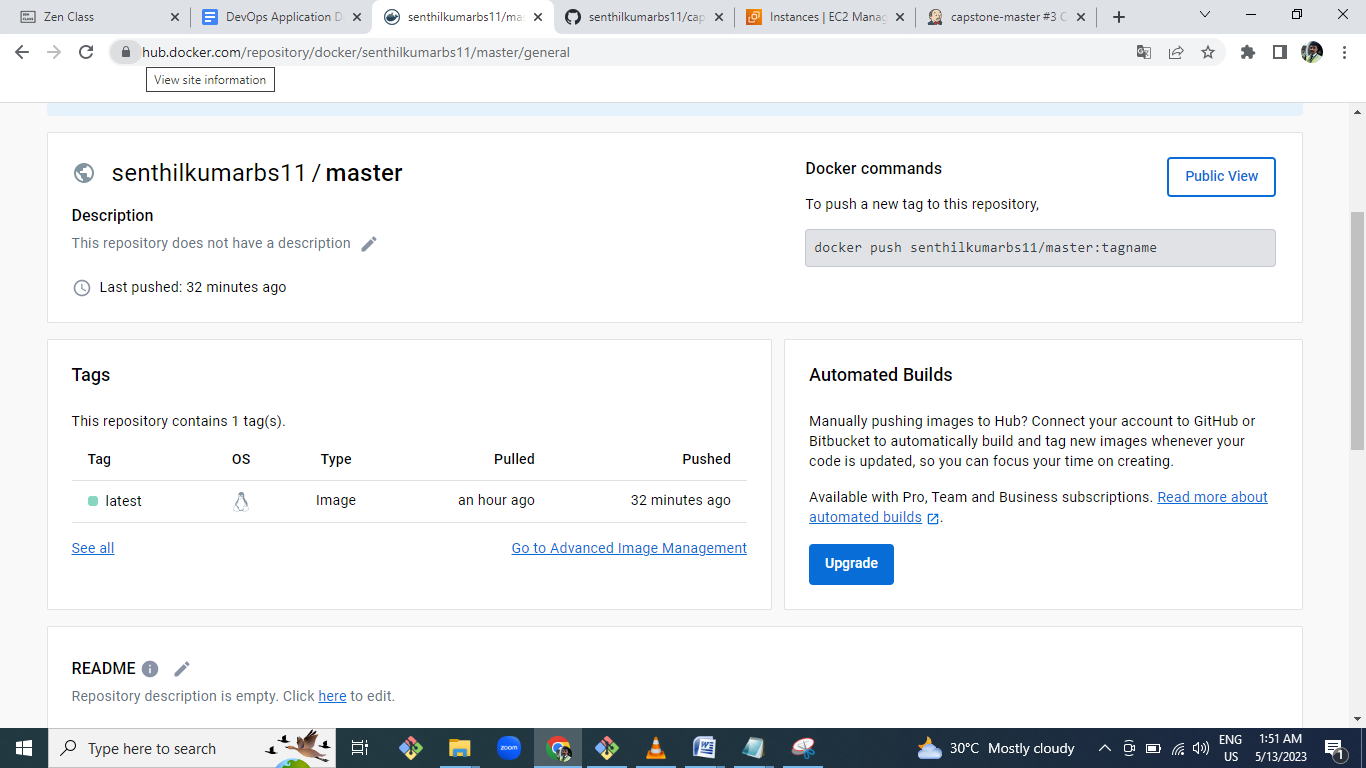












**AWS:**

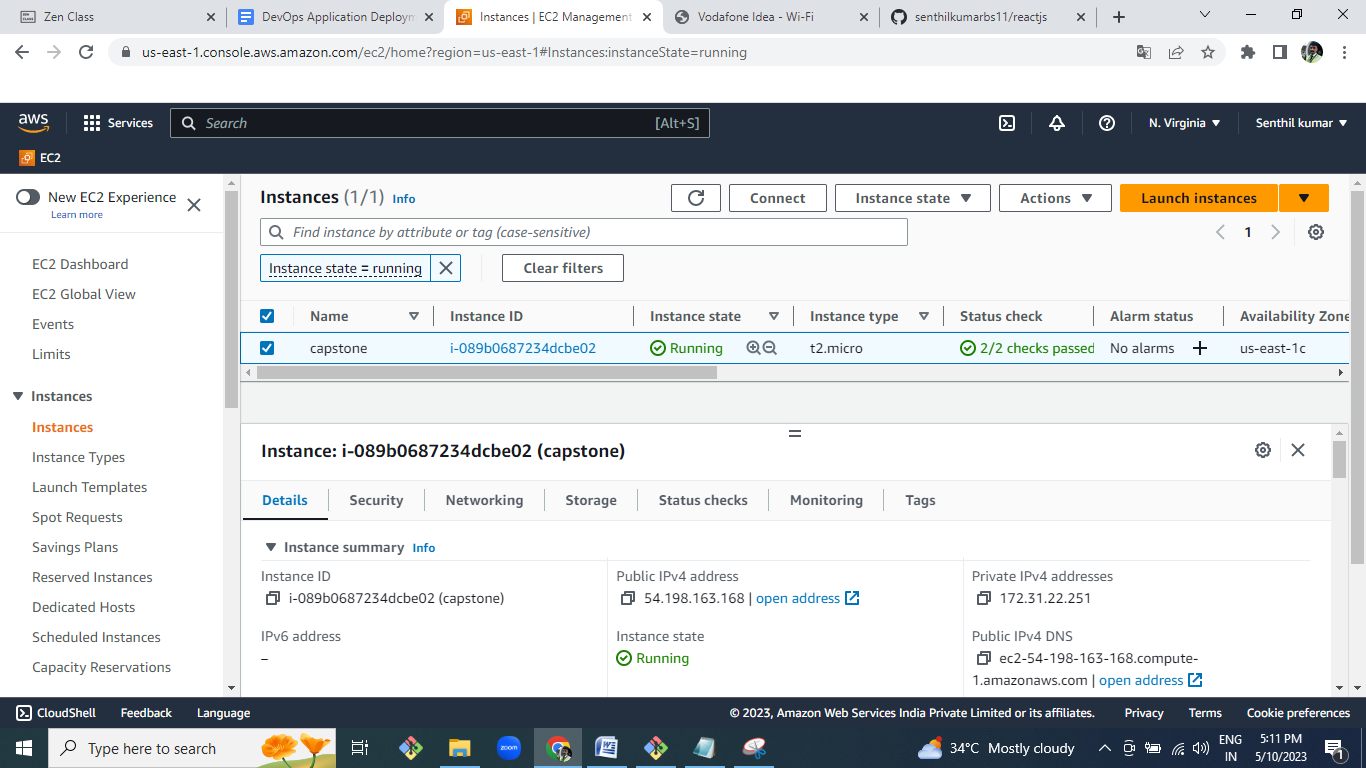
**Launch t2.micro instance and deploy the create application.**

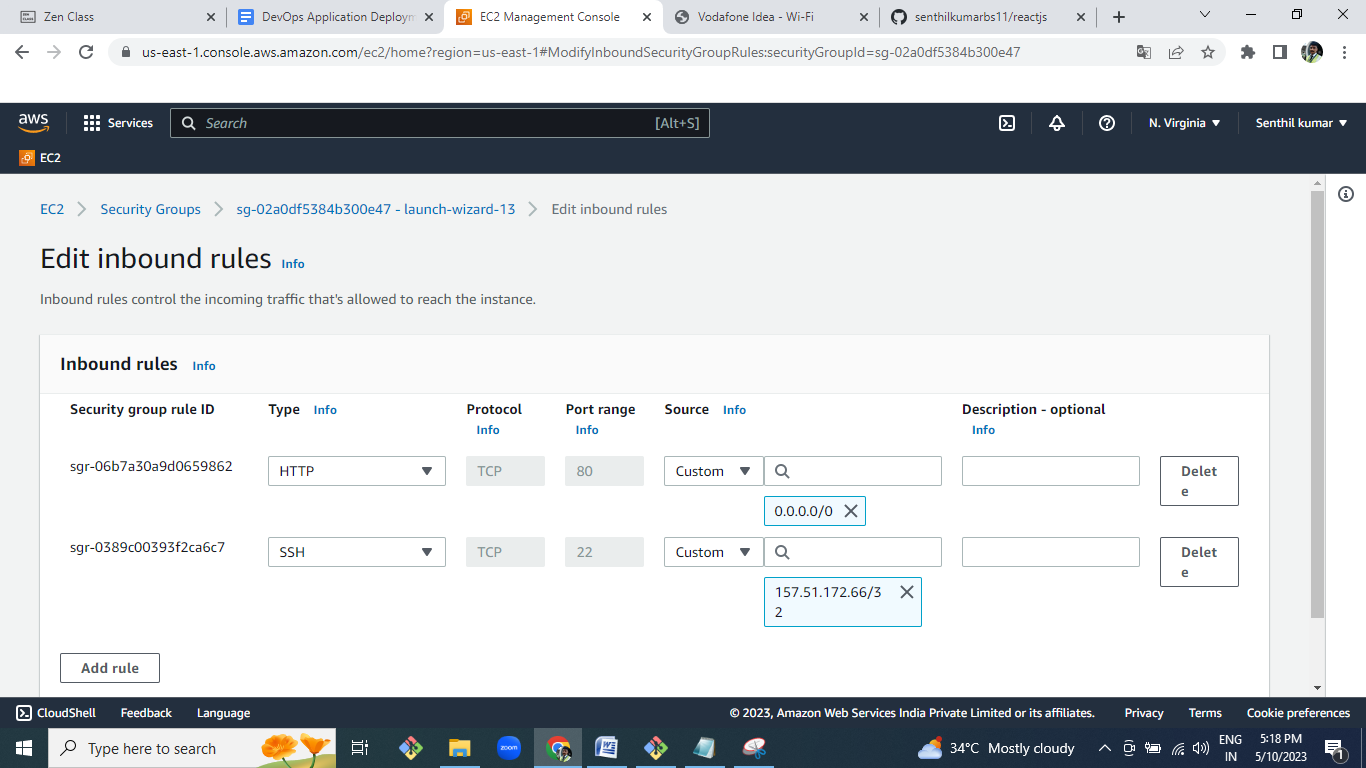
**Configure SG as below:**

**-**

**Whoever has the ip address can access the application**

**Login to server can should be made only from your ip address**

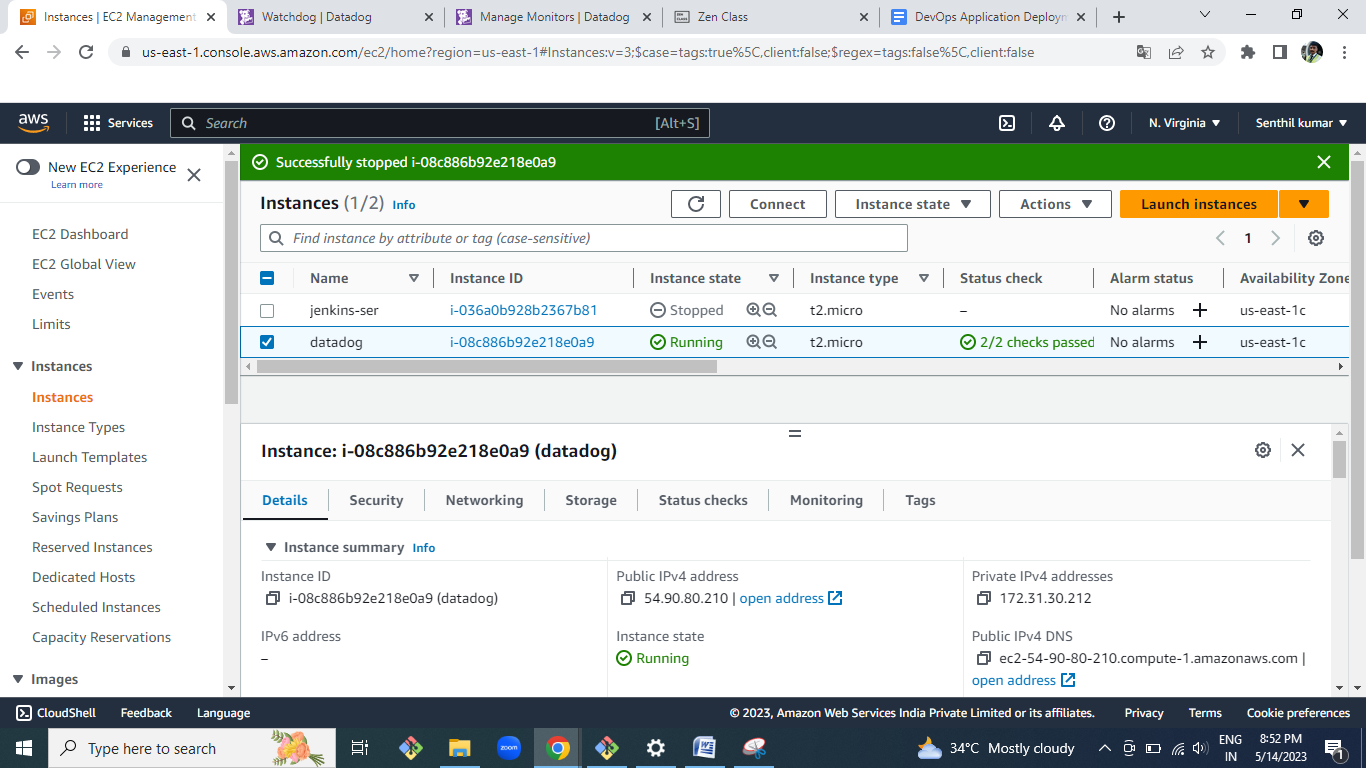
****

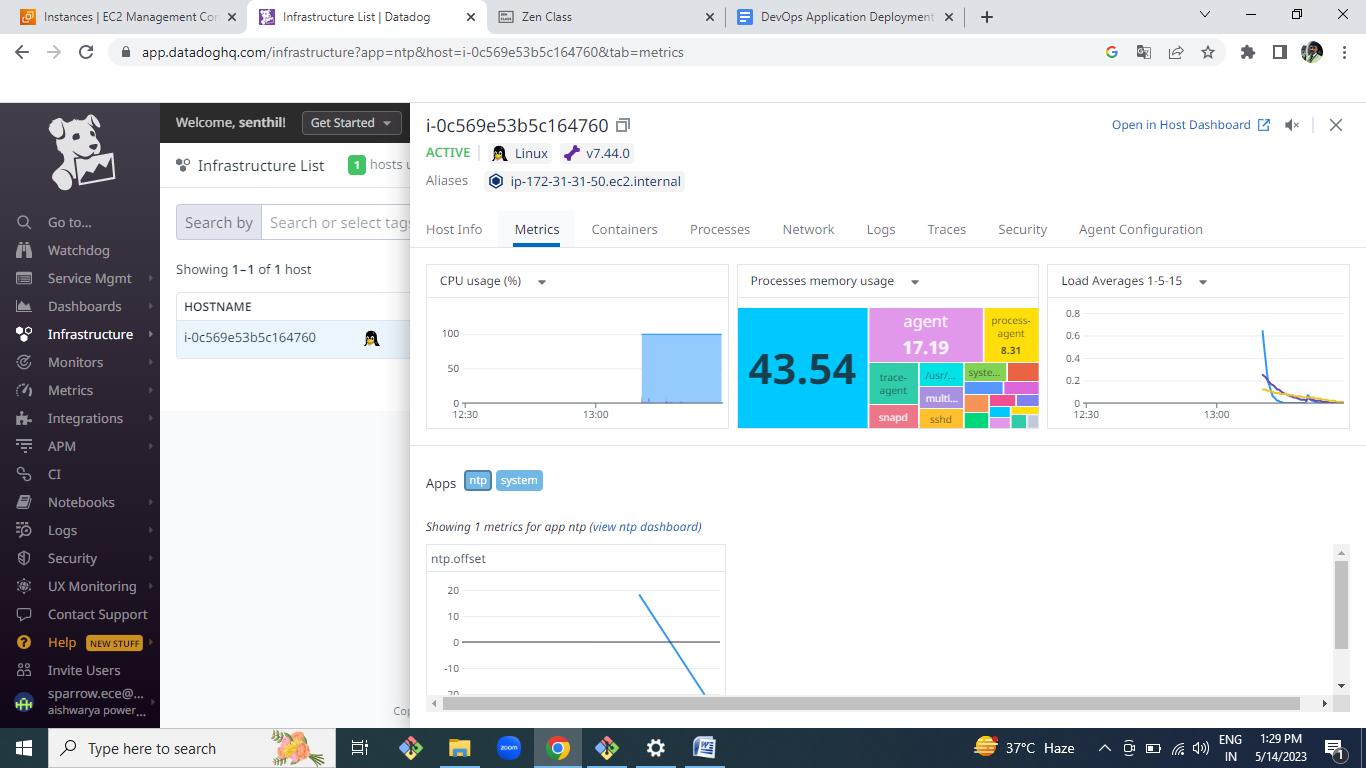


Monitoring:

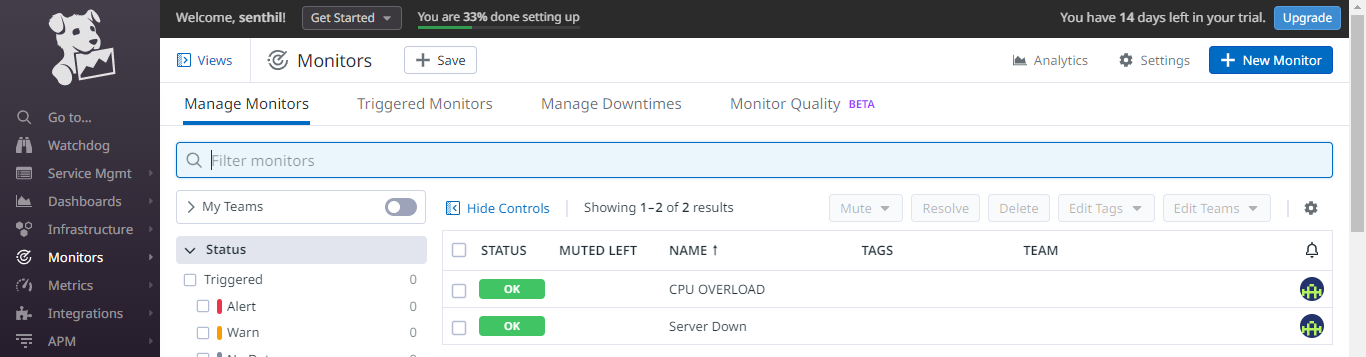
Setup a monitoring system to check the health status of the application. (Open-source)

Sending notifications only if the application goes down is highly appreciable





TWO MONITOR TASK ASSIGNED:



USING STRESS TOOL INCREASED CPU UTILIZATION GOT ALERT NOTIFICATION GOT

AND SERVERDOWN NOTIFICATION ALSO WHEN SERVER WAS STOPPED

