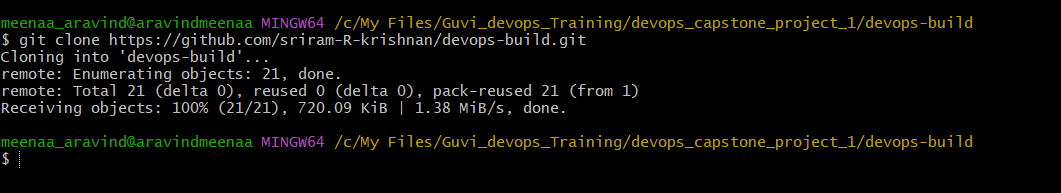
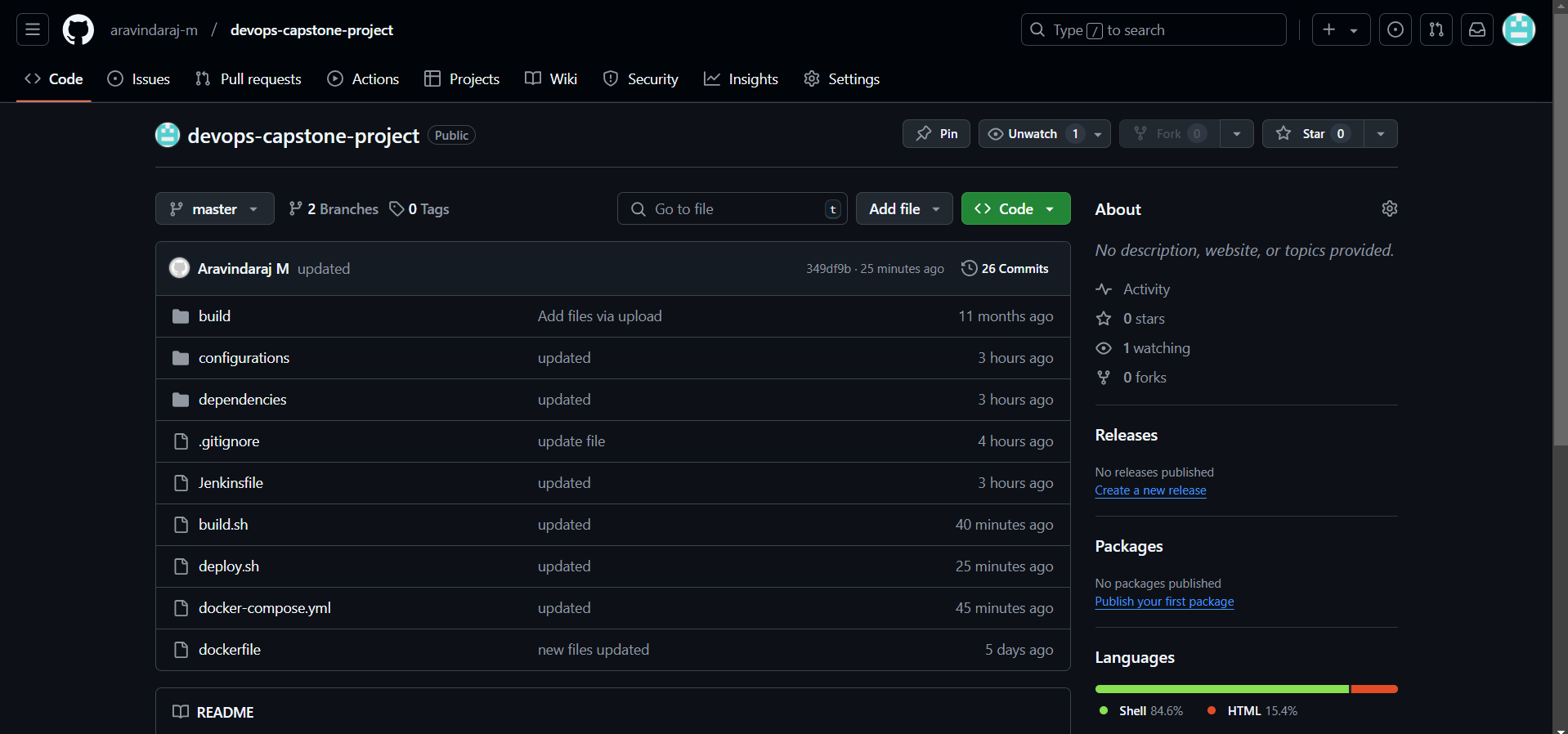
**DevOps Application Deployment Capstone – Reactjs E-commerce Application**

**🡪Cloning the Pre-build website application:**

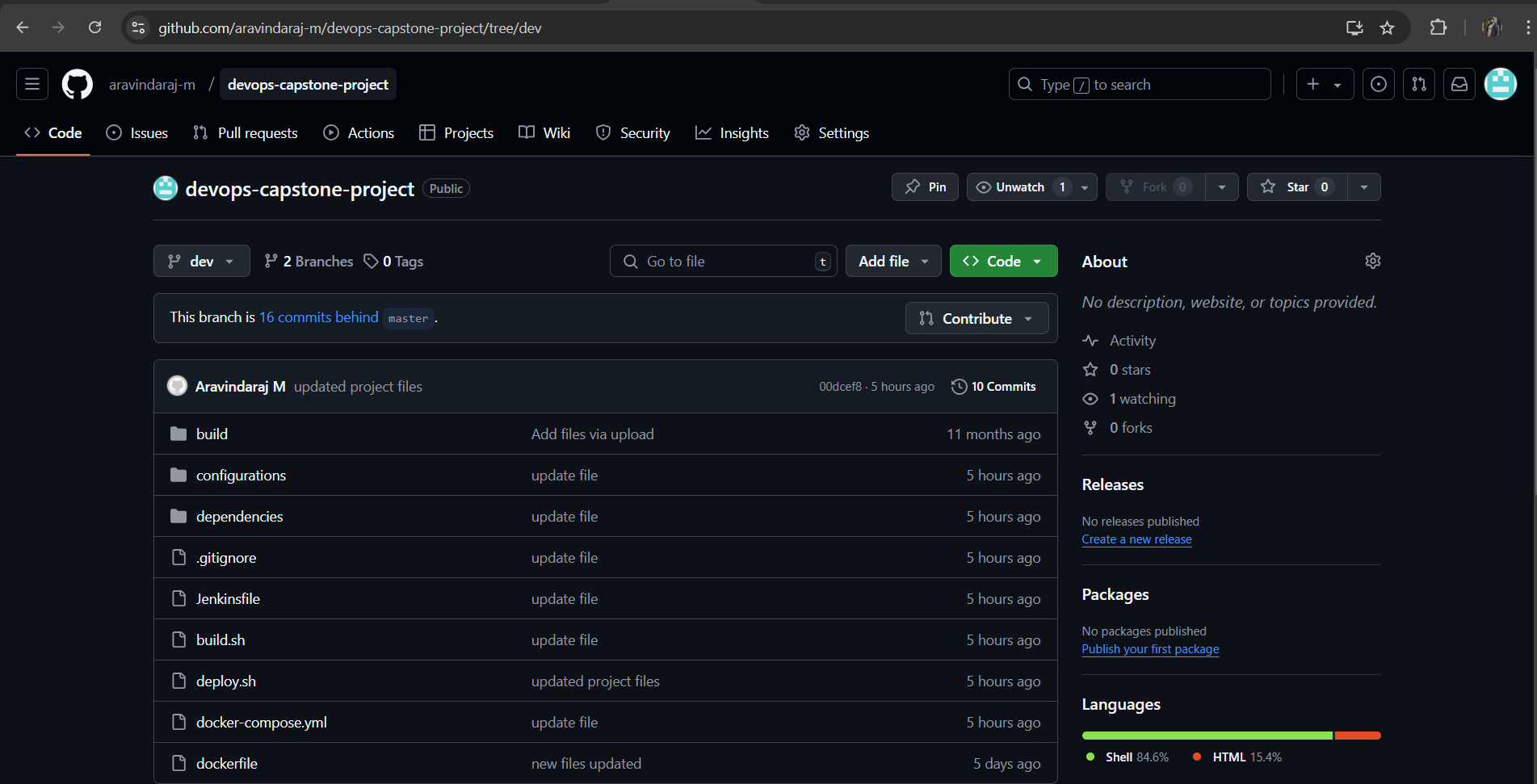
Create a folder in your local path and clone the application in public user repository



**🡪master branch**



**🡪dev branch**



**🡪Source code file and Structure**

Source code

|---Build

|---configurations

| |---coinfig.ini

|---dependencies

| |---alert.rules.yml

| |---alertmanager.yml

| |---nginx.conf

| |---prometheus.yml

|---build.sh

|---deploy.sh

|---dockerfile

|---docker-compose.sh

|---Jenkinsfile

|---.gitignore

🡪dockerfile

build a custom image out of nginx image to build the pre-build application and copy build files to default nginx file path

🡪docker-compose.yml

Running multiple container from custom build image, nginx exporter, cadvisor, prom/Prometheus, grafana, alertmanager .

🡪build.sh

Shell script to build custom image tag it dev repo and push it to a docker hub user repository

🡪deploy.sh

shell script to pull docker image from docker hub and retag to prod repo and push it to docker hub repository and run the docker-compose.yml file to deploy the application in server and monitor the application.

🡪Jenkinsfile

For Jenkins CICD pipeline project Jenkinsfile is used

🡪Configurations ---config.ini

All the configuration and settings for build.sh and deploy.sh files are stored in this ini file to pass argument when executing

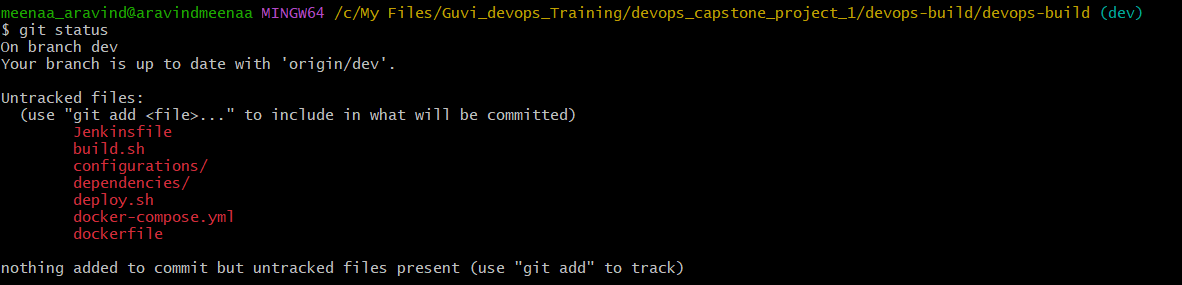
🡪Dependencies

.yml and .conf files that needs to be placed inside the containers are stored in this location, along with docker-compose.yml dependencies will be copied to deployment machine

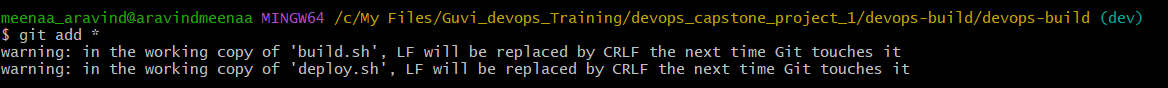
🡪.gitignore

To ignore secret file and confidential files to upload to git repository

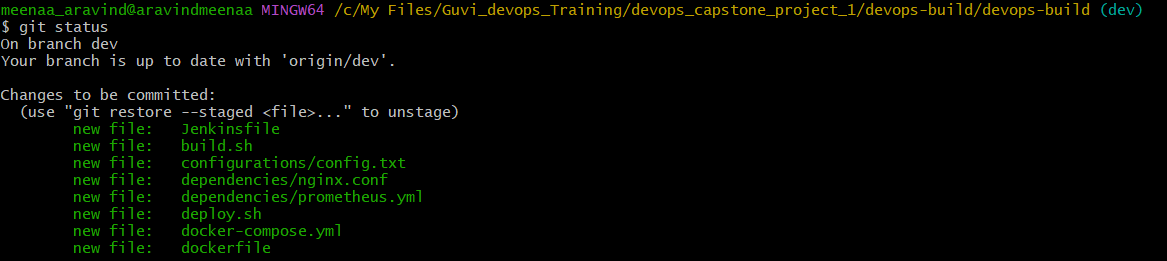
**🡪 push the code to GitHub repository**



git status



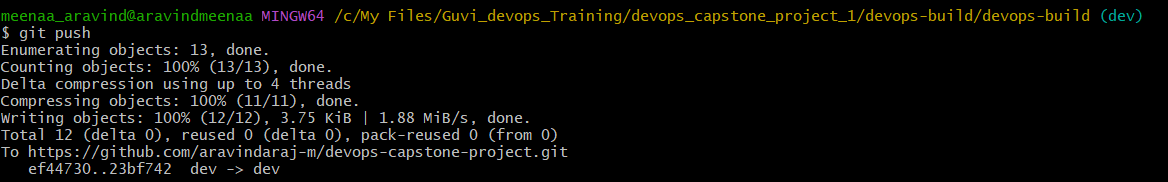
git add \*



Again git status

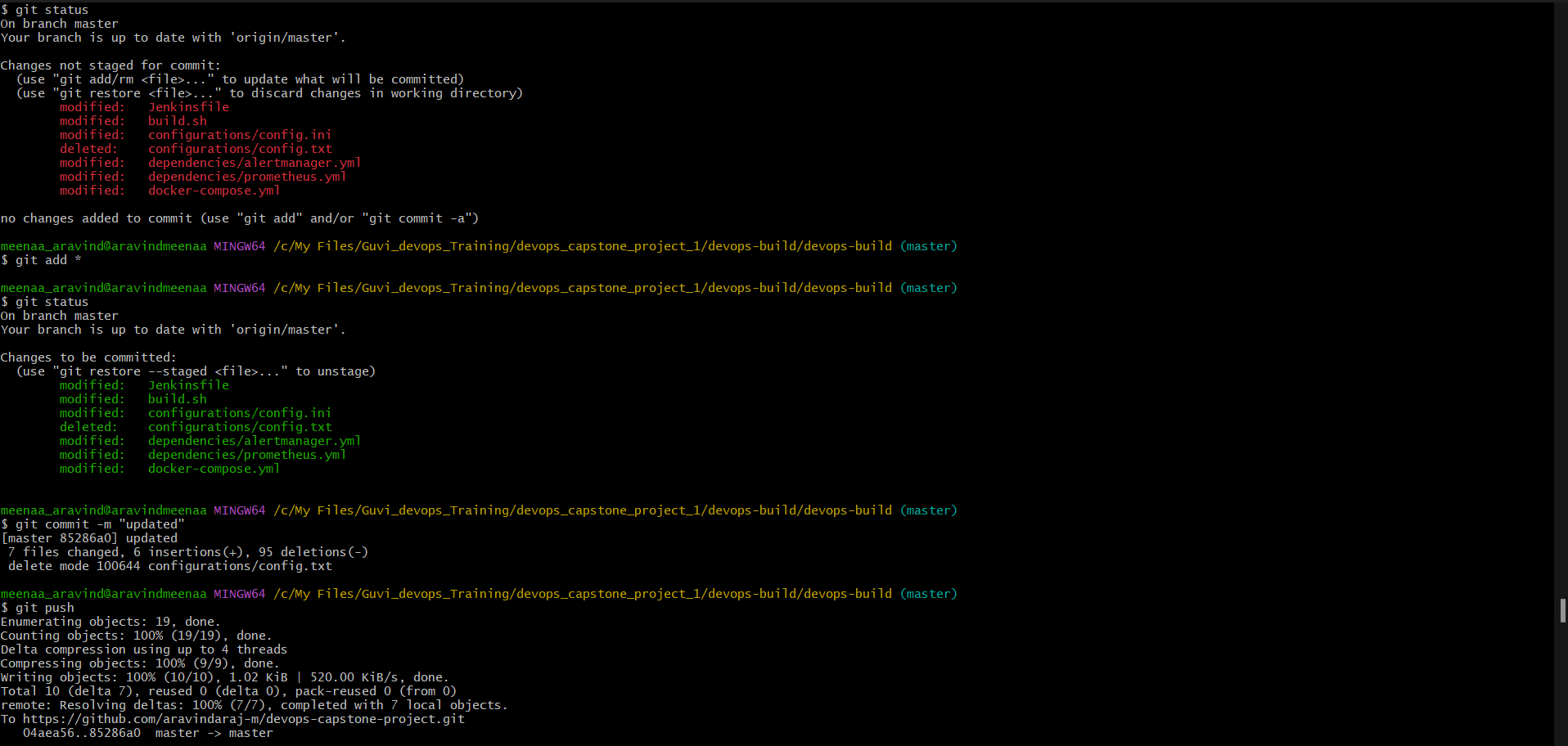


git commit -m “message for commit”

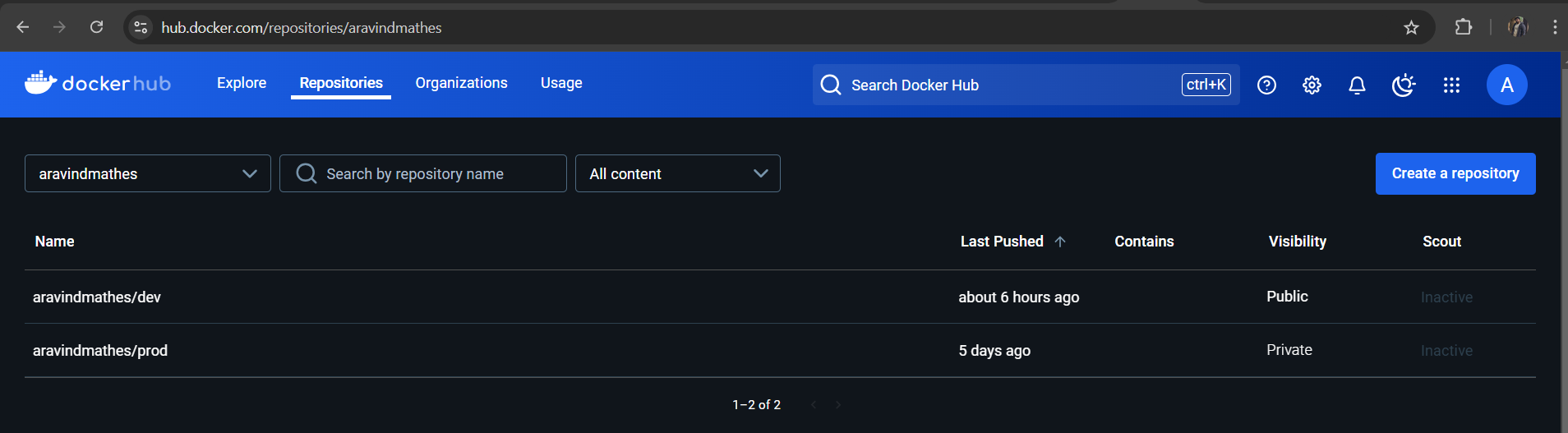


git push

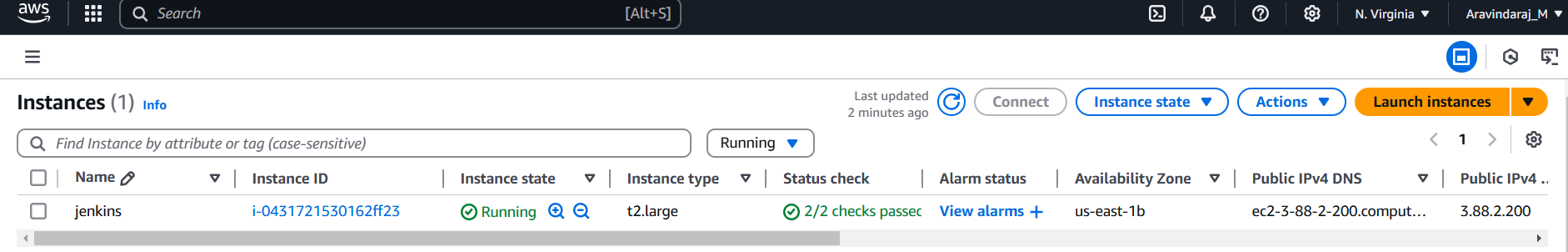
**🡪 Git merge to master**



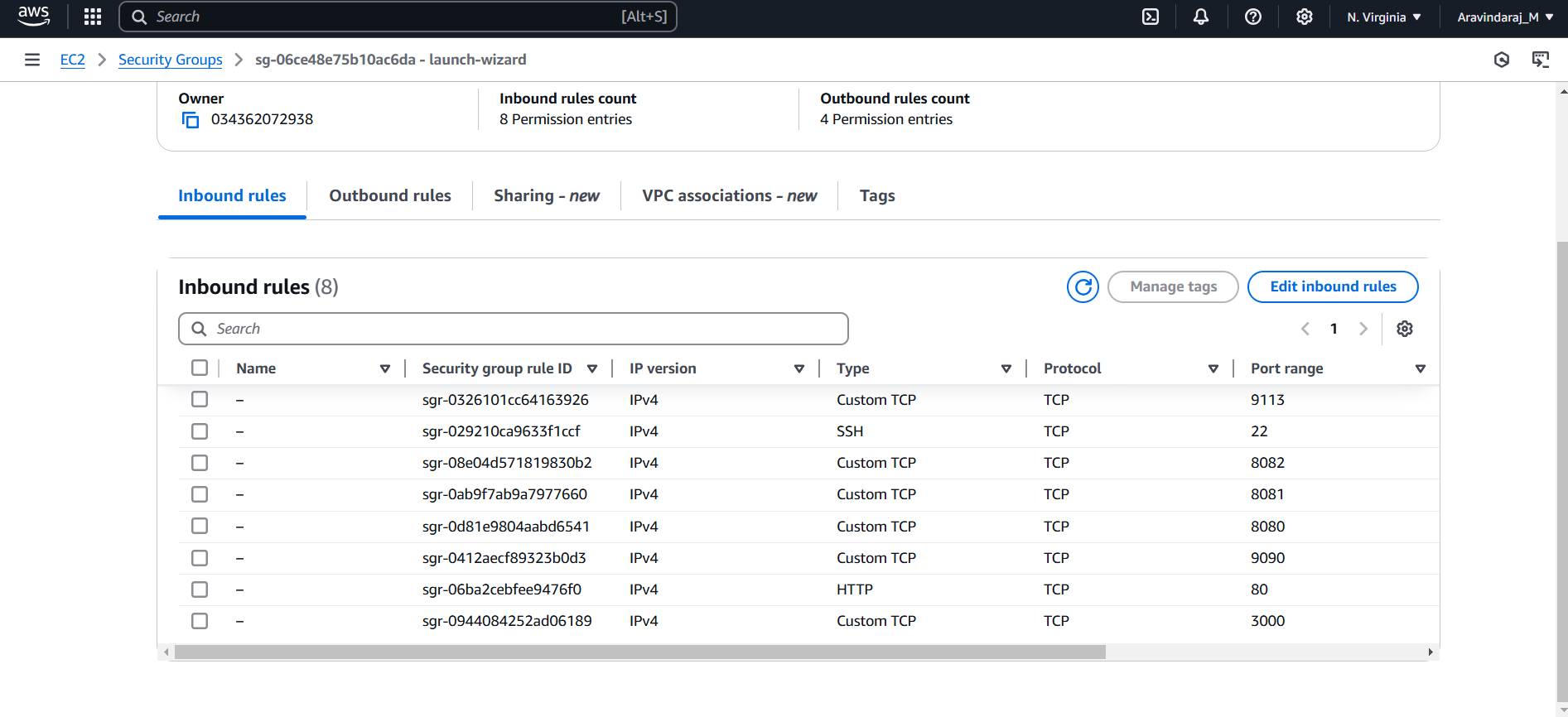
**🡪Docker Hub repositories “dev” and “prod”**



**🡪AWS EC2 Instance**



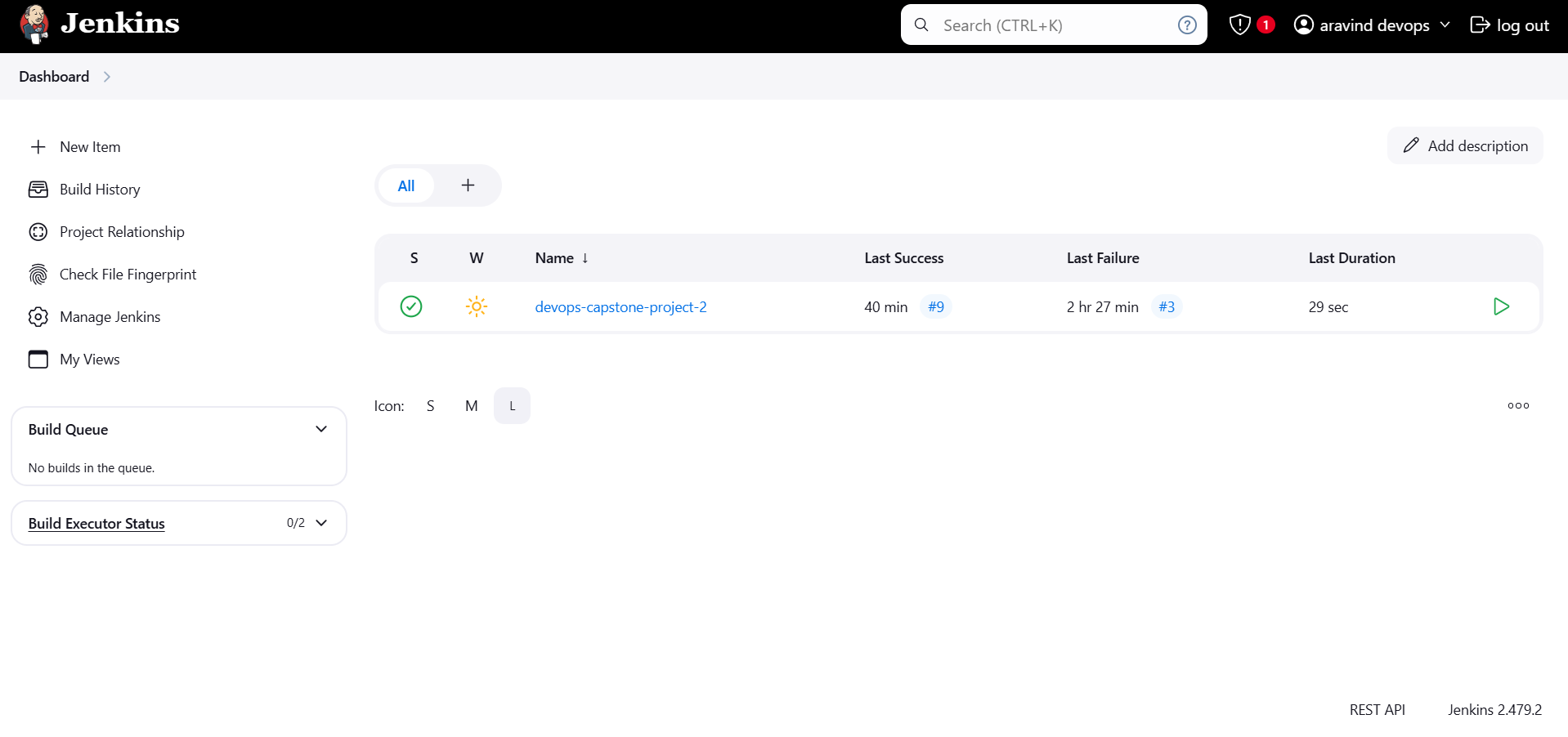
**🡪AWS Security Group**



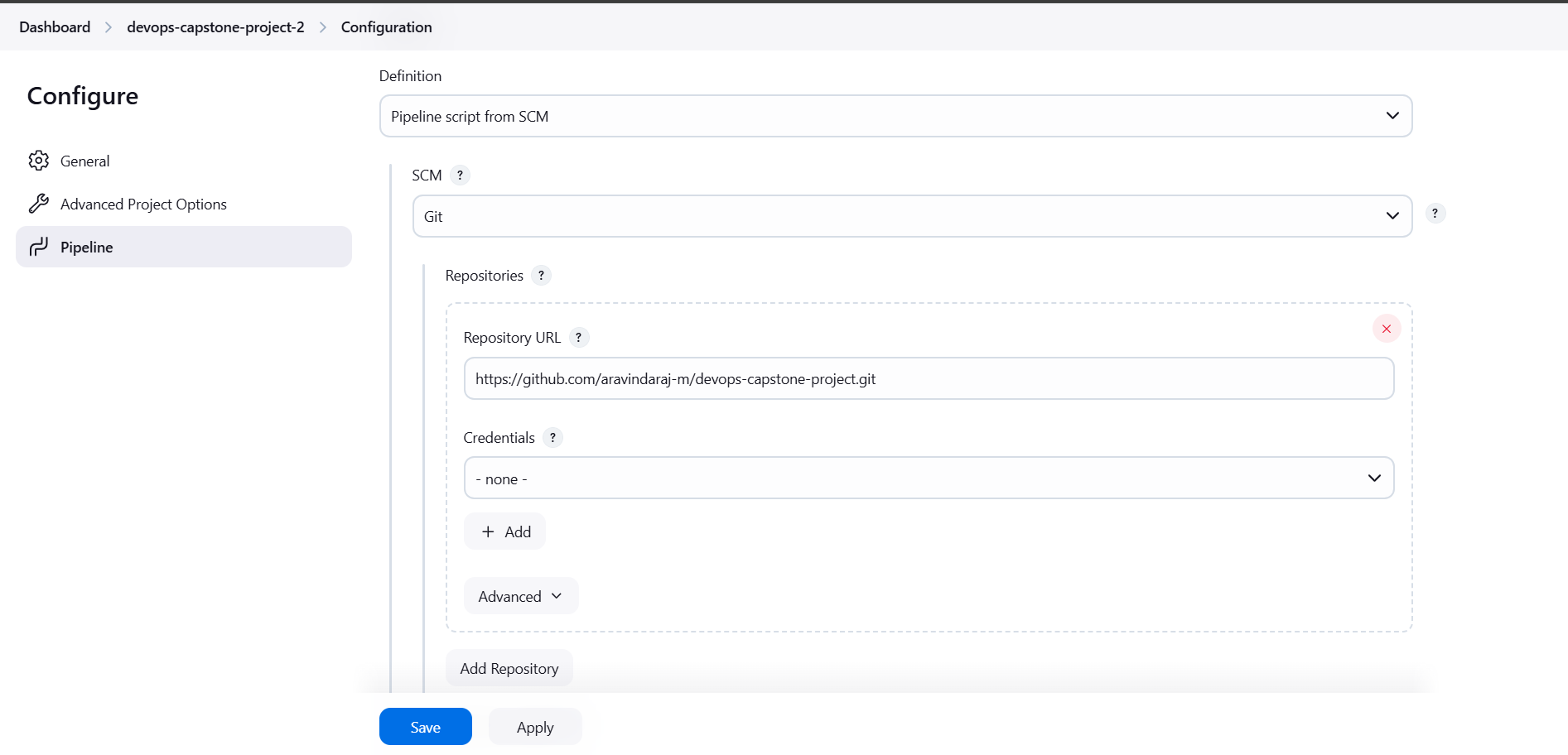
**🡪Jenkins Installation and configurations**

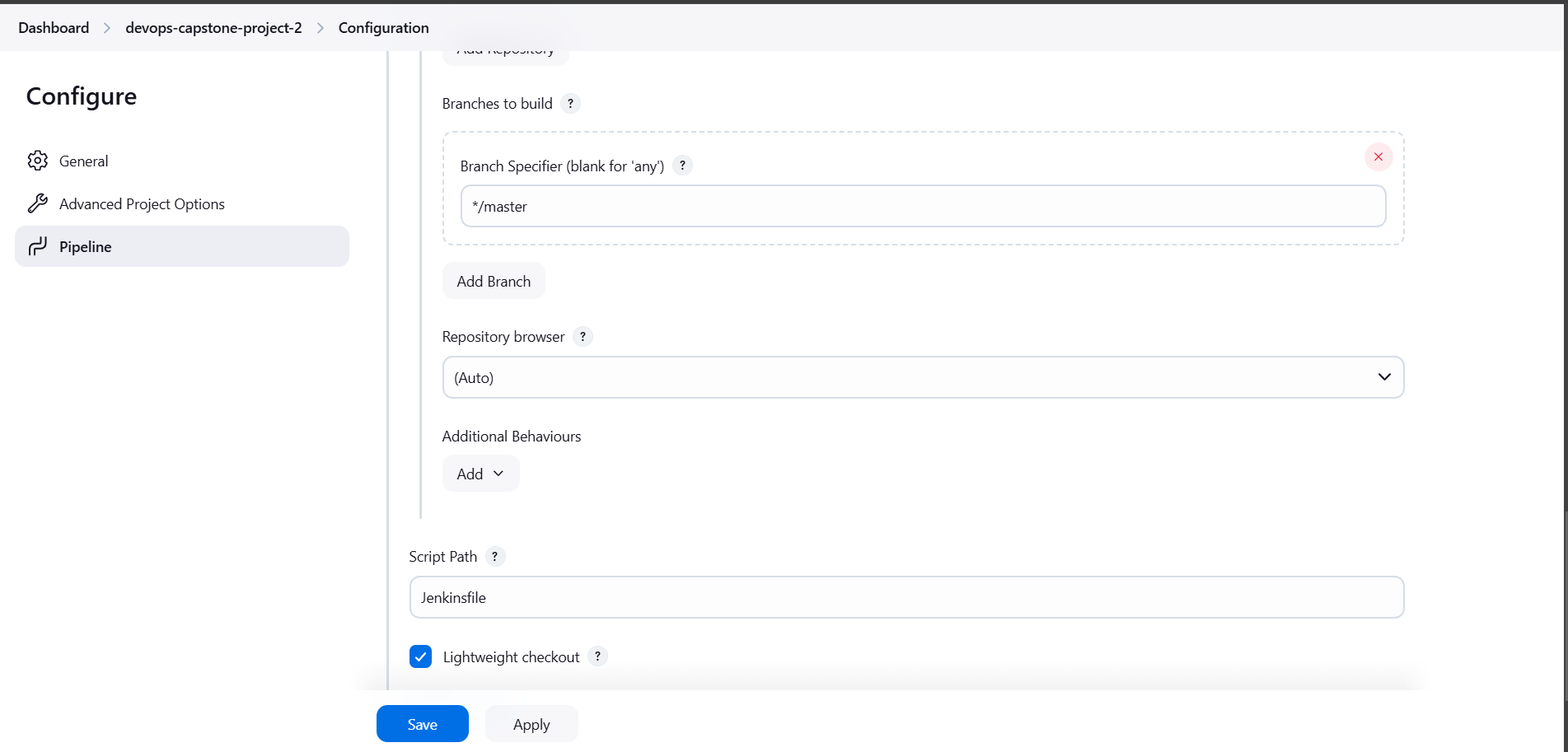
Login into Jenkins page and start setting up the Jenkins for your pipeline project by enabling GitHub Integration, SSH Agent

**🡪Jenkins Dashboard**

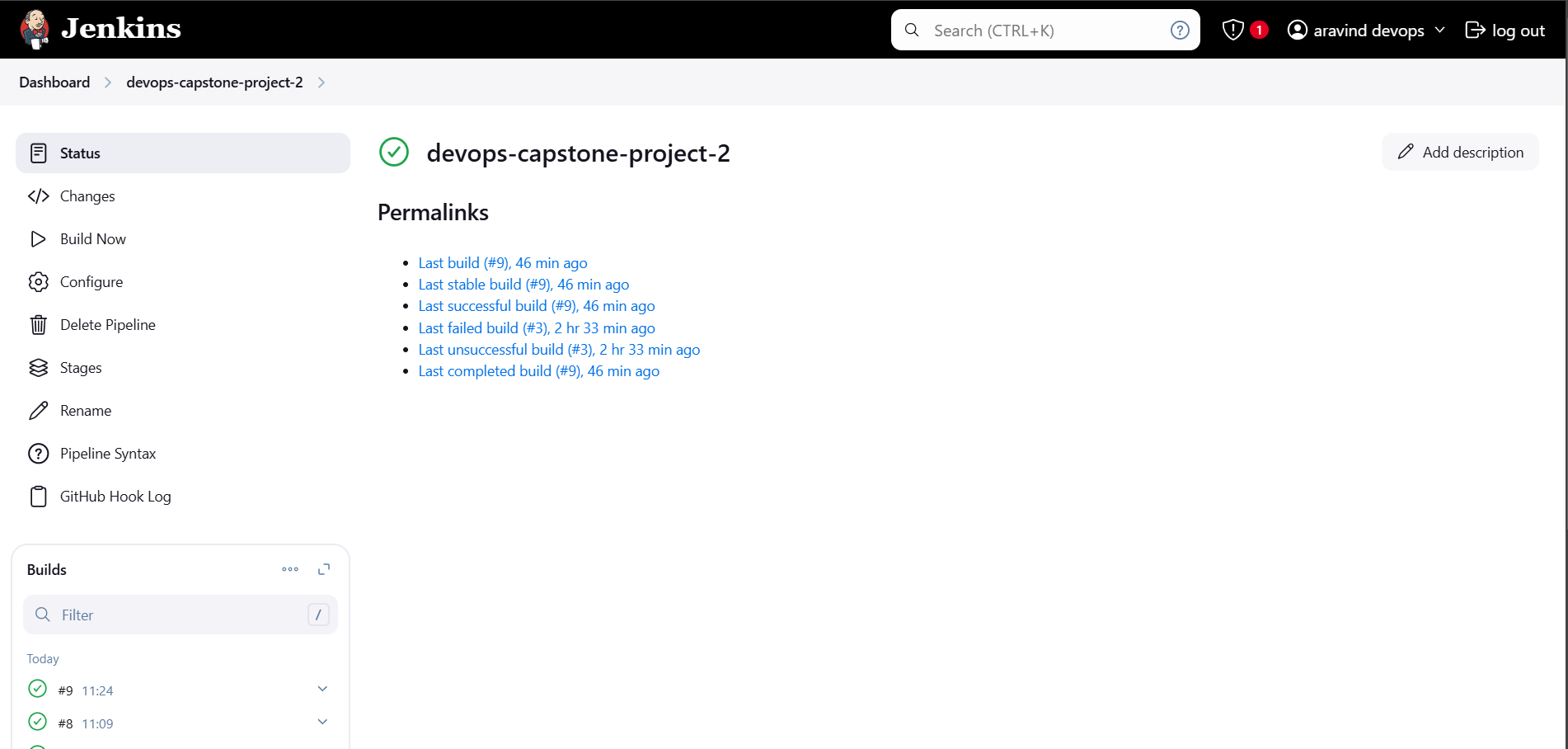


**🡪Jenkins Pipeline Job Configuration**





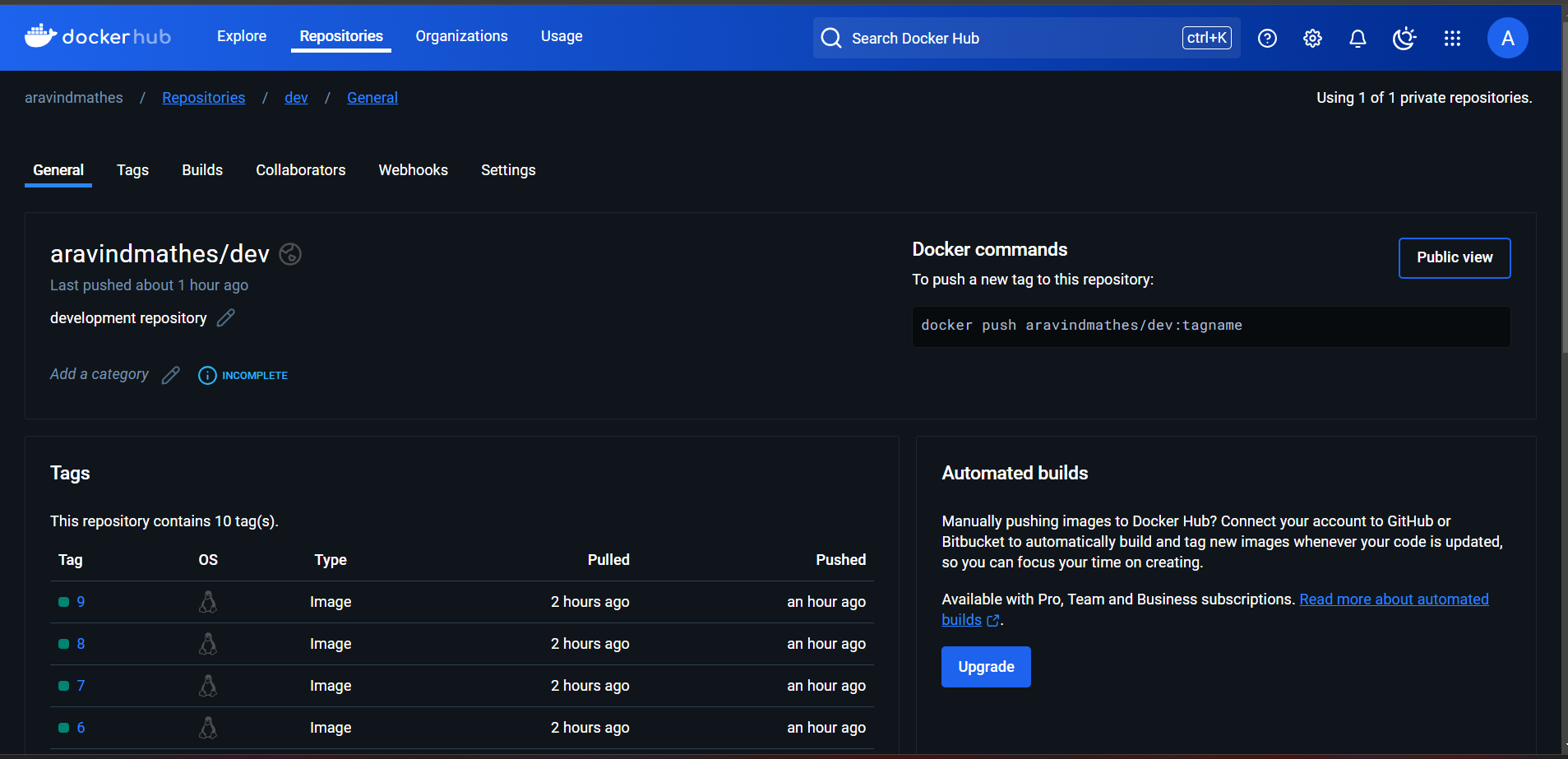
**🡪Jenkins Pipeline project**



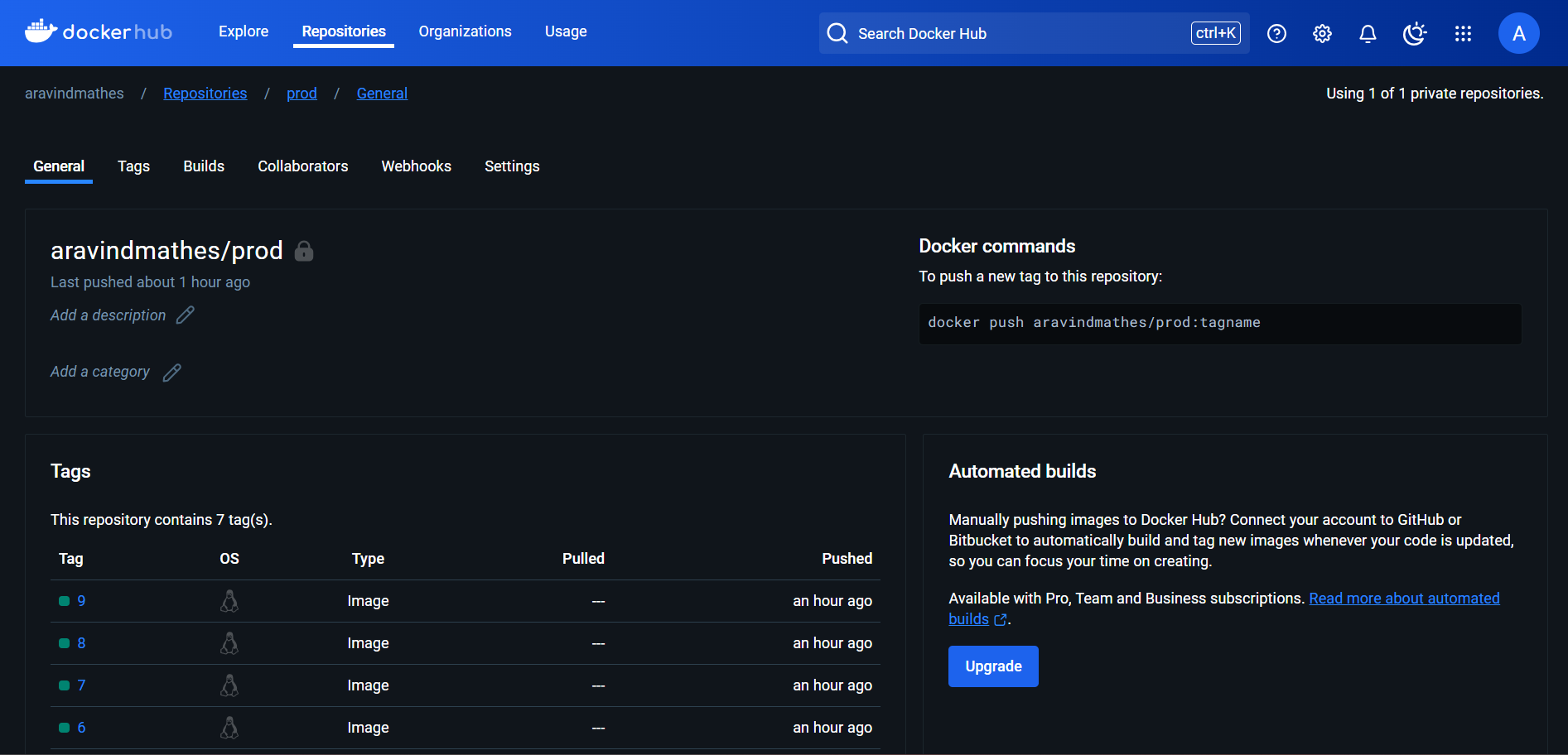
**🡪Container Running**



**🡪Docker Hub dev repo**

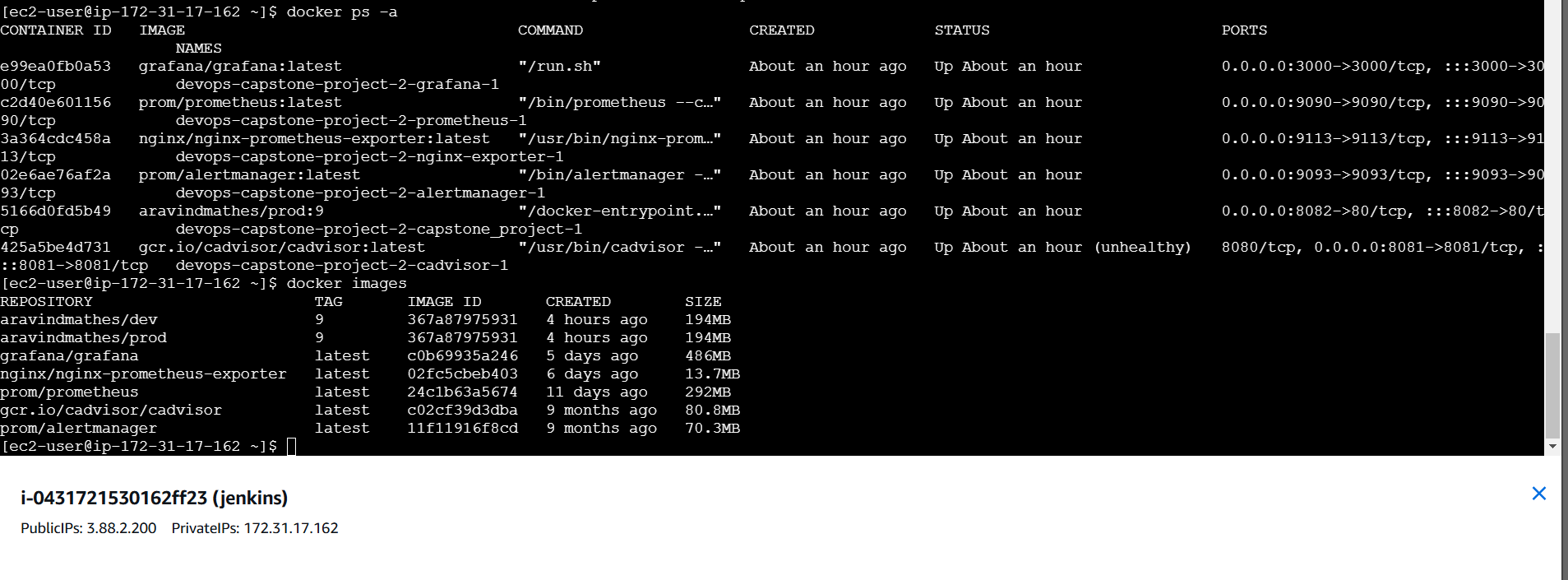


**🡪Docker Hub prod repo**

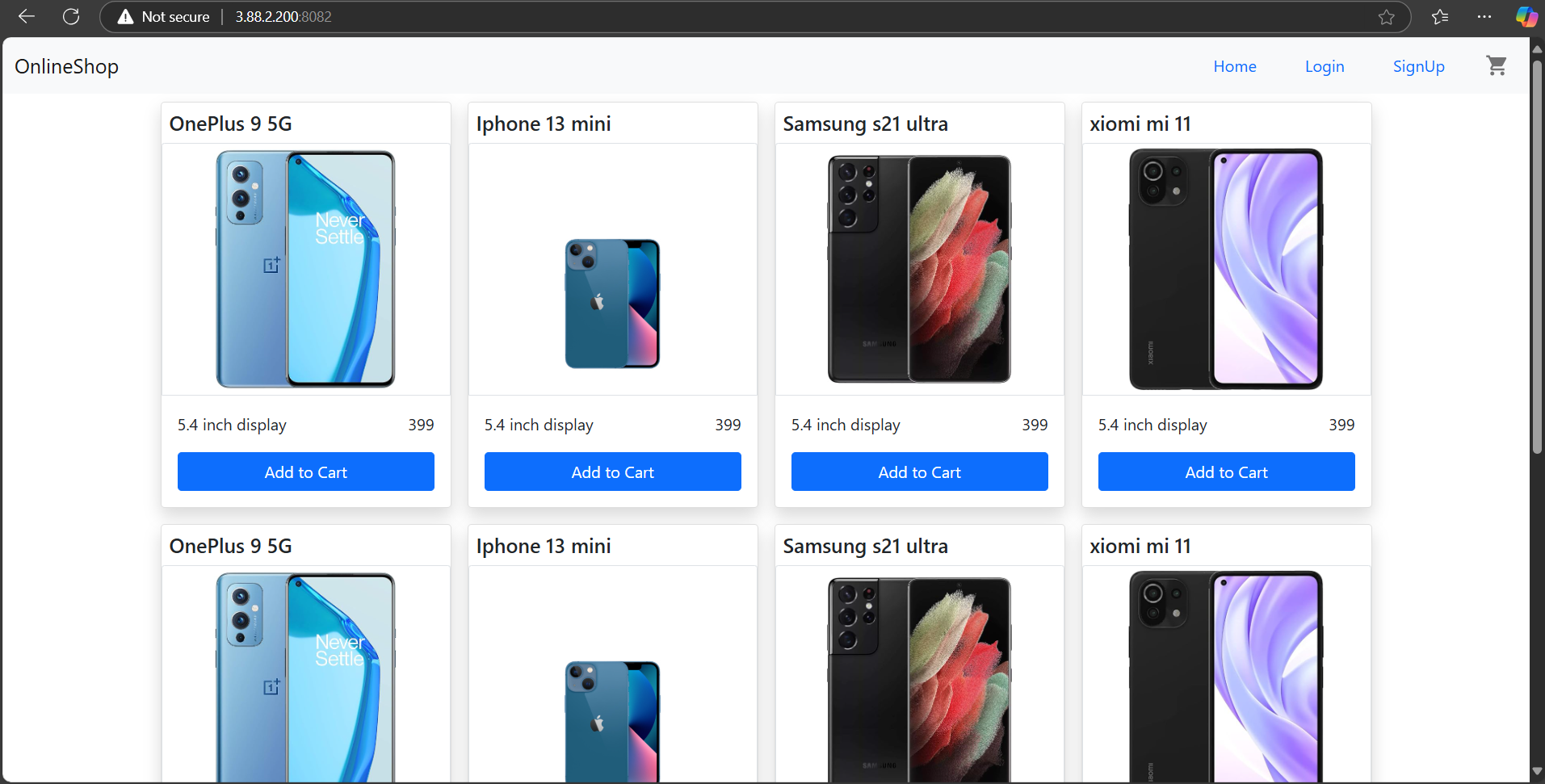


* Application is deployed and running in container.
* Cadvisor is used to check container health status.
* Prometheus is used to get data from cadvisor
* Grafana is used for Graphical display for the metrics
* Alert manager is set to receive email notification on application goes down

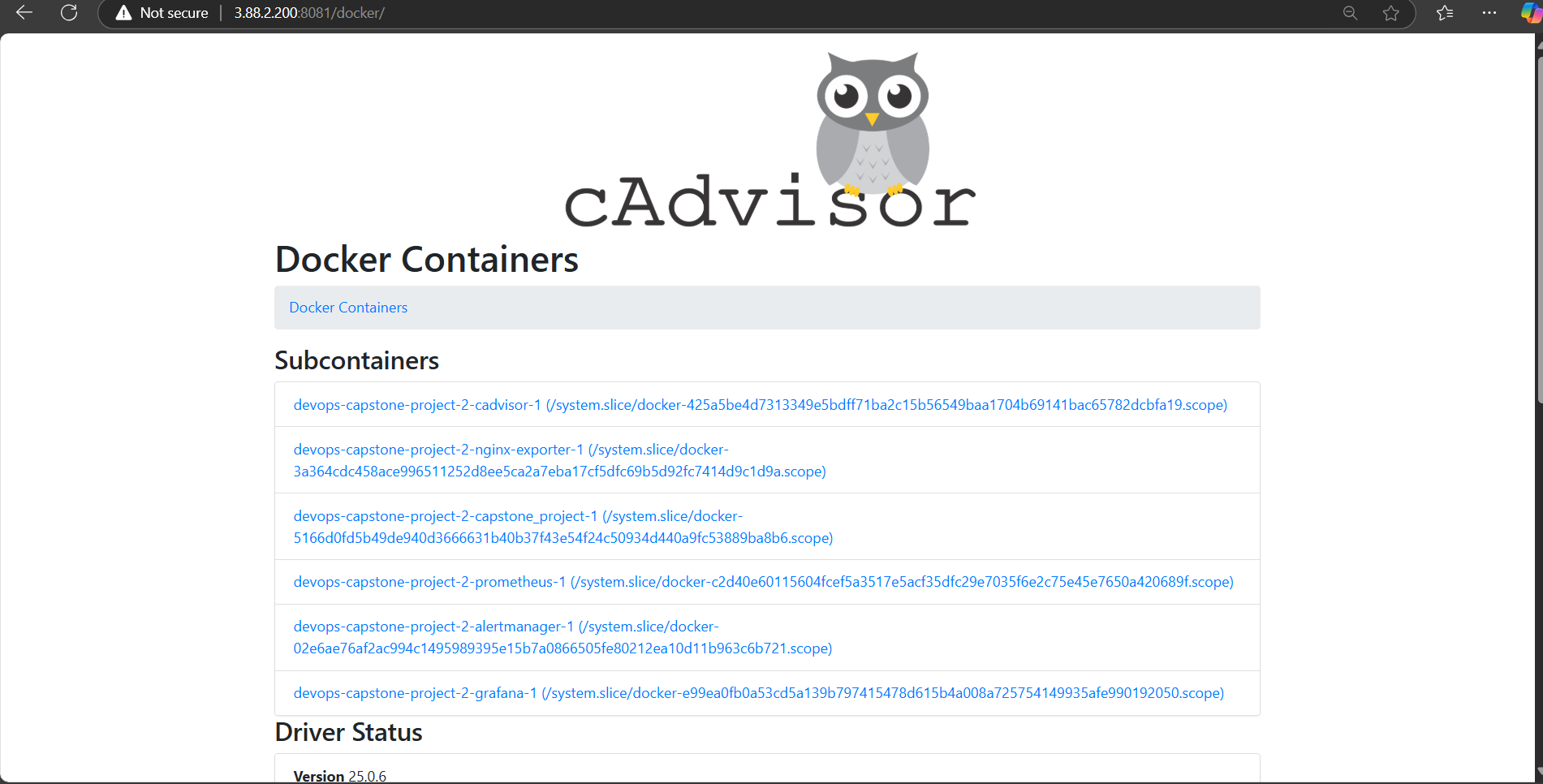
**🡪EC2 Console**



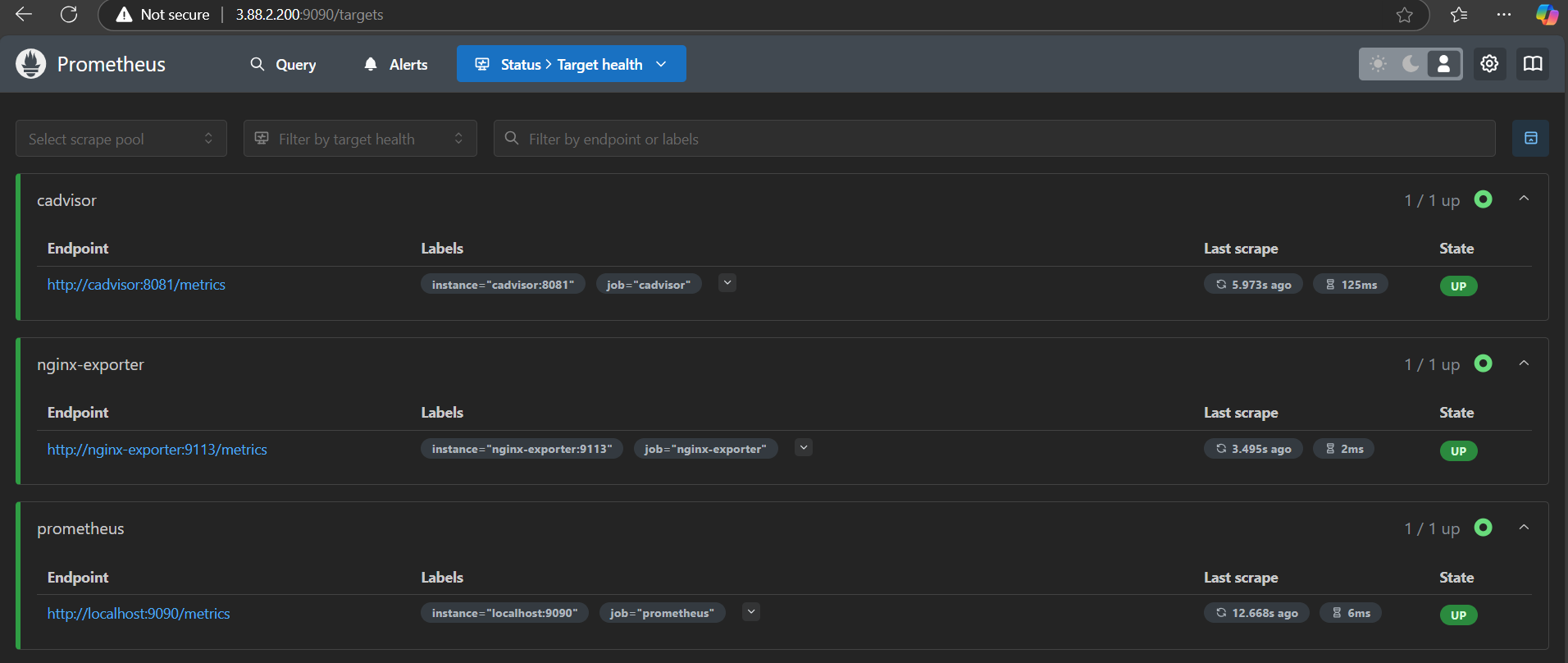
**🡪Application Deployed**



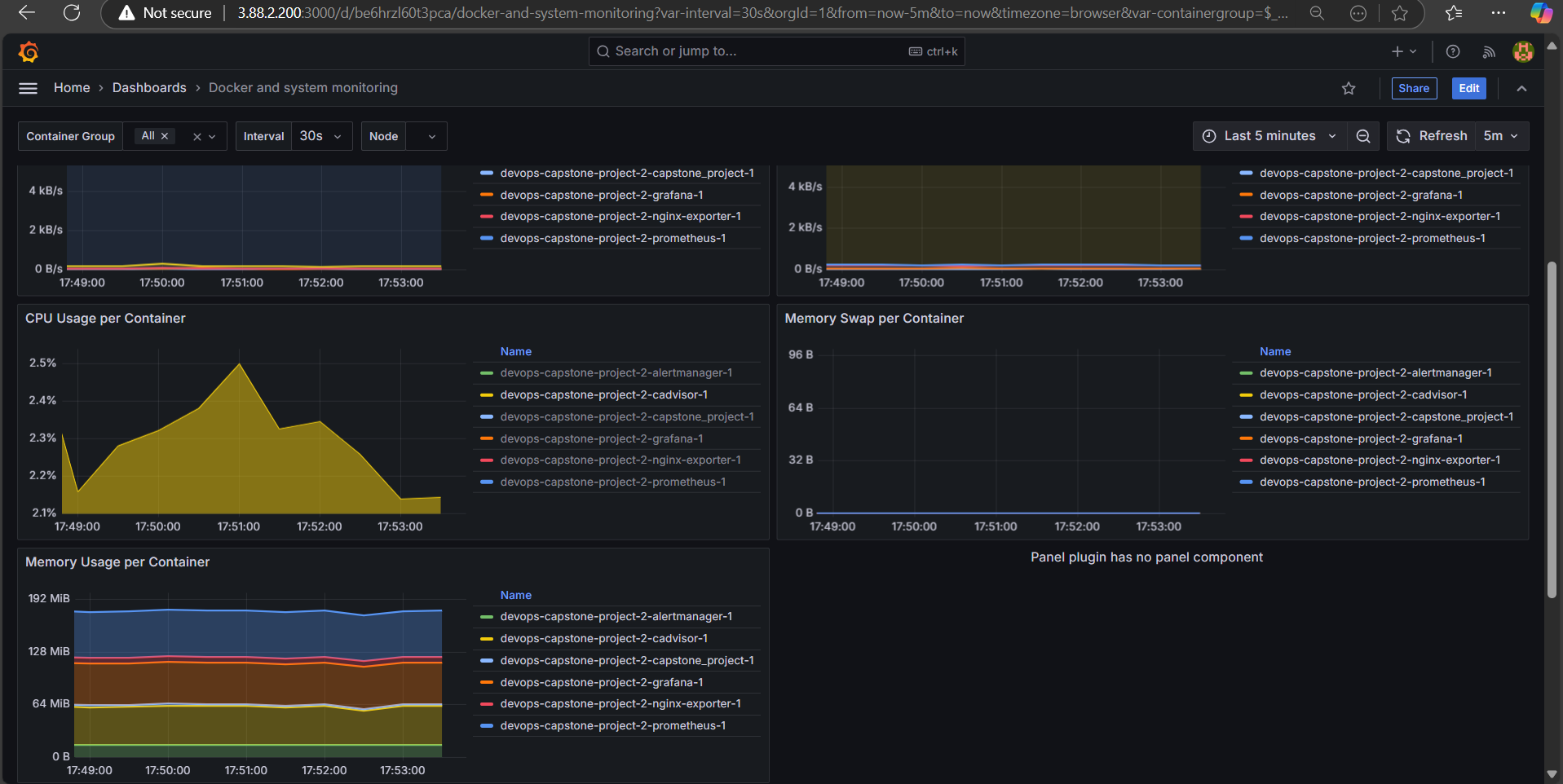
**🡪Cadvisor**



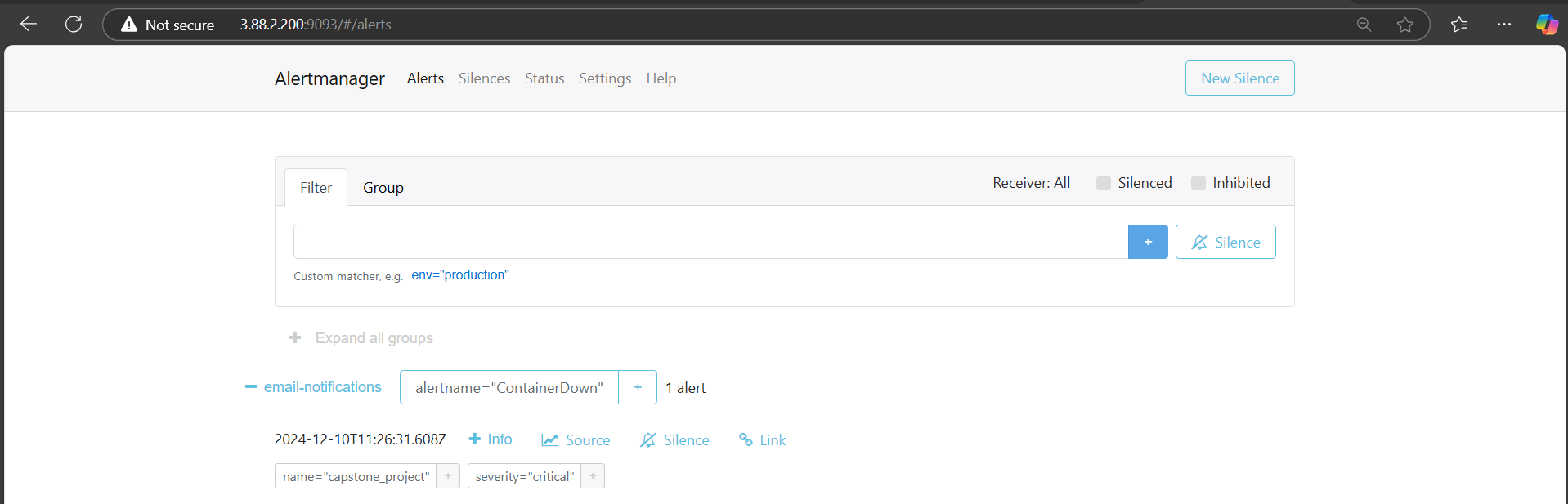
**🡪Prometheus**



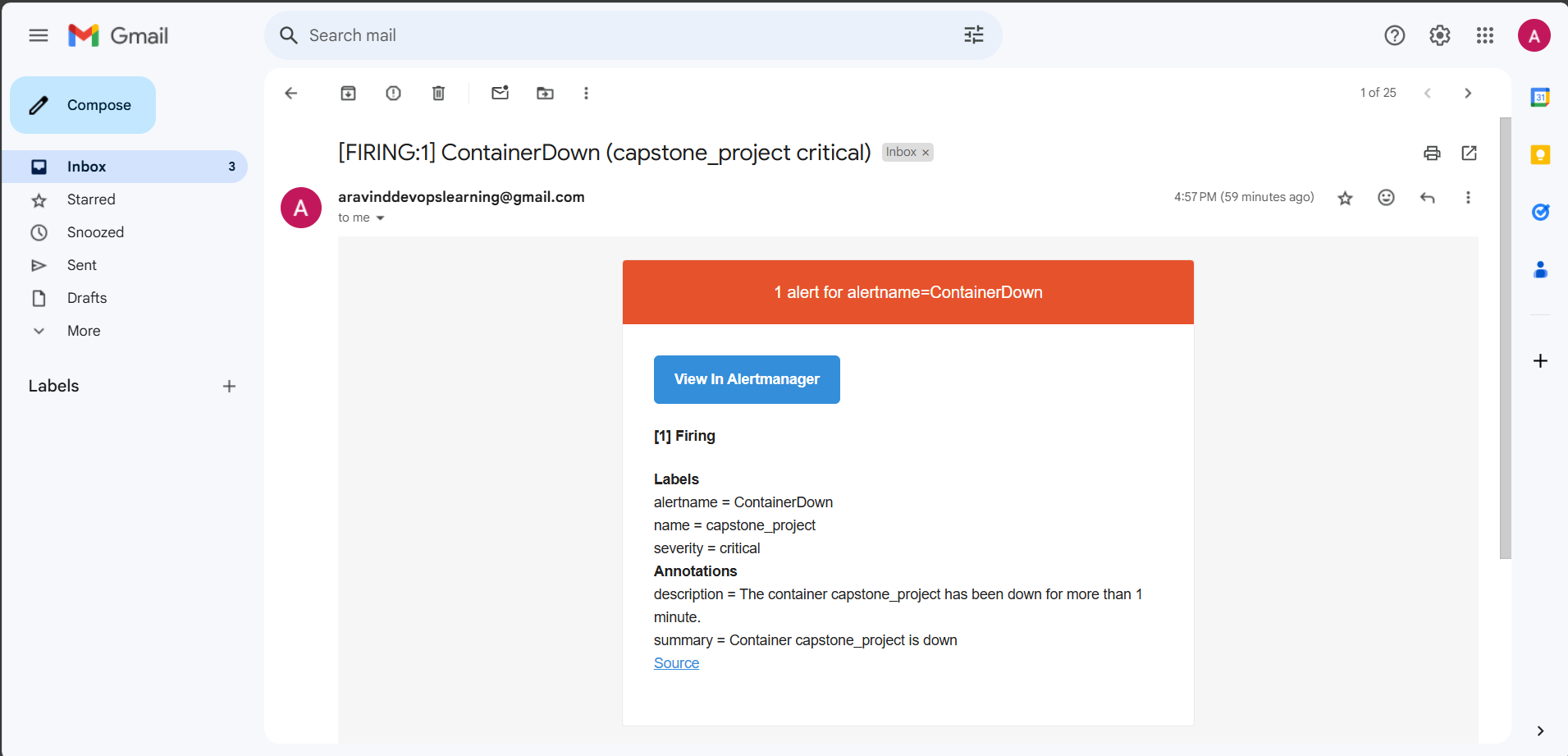
**🡪Grafana**



**🡪Alert Manager**



**🡪e-mail Notification**



**🡪All Docker Images Name**

REPOSITORY TAG IMAGE ID CREATED SIZE

aravindmathes/dev 9 367a87975931 3 hours ago 194MB

aravindmathes/prod 9 367a87975931 3 hours ago 194MB

grafana/grafana latest c0b69935a246 5 days ago 486MB

nginx/nginx-prometheus-exporter latest 02fc5cbeb403 5 days ago 13.7MB

prom/prometheus latest 24c1b63a5674 11 days ago 292MB

gcr.io/cadvisor/cadvisor latest c02cf39d3dba 9 months ago 80.8MB

prom/alertmanager latest 11f11916f8cd 9 months ago 70.3MB

**Git Hub URL** <https://github.com/aravindaraj-m/devops-capstone-project.git>

**Application Deployed on** <http://3.88.2.200:8082>

**Steps for setting up Jenkins server for Docker Containerization**

**🡪EC2 Setup for Jenkins:**

Jenkins, Java 17, Docker and Git are required for the setup to run

🡪Install Docker

sudo yum update -y

sudo yum install -y docker

sudo service docker start

🡪Install Docker Compose (standalone binary)

sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

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

sudo docker-compose –version

🡪Add the EC2 user to the docker group

sudo usermod -aG docker ec2-user

🡪Install Git for Jenkins to use git repository and pull the changes

sudo yum install git

🡪Install Jenkins

Refer the link to install Jenkin in EC2 Instance

<https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/>

🡪Add the EC2 user to the docker group

sudo usermod -aG docker ec2-user

🡪Add user to the docker group

sudo usermod -aG docker Jenkins

**Ports enabled in AWS Security Group:**

#Inbound Rules:

SSH-22

HTTP-80

CUSTOM TCP-8080

CUSTOM TCP-9113

CUSTOM TCP-9090

CUSTOM TCP-3000

CUSTOM TCP-8082

CUSTOM TCP-8081

CUSTOM TCP-9093