**DevOps Application Deployment**

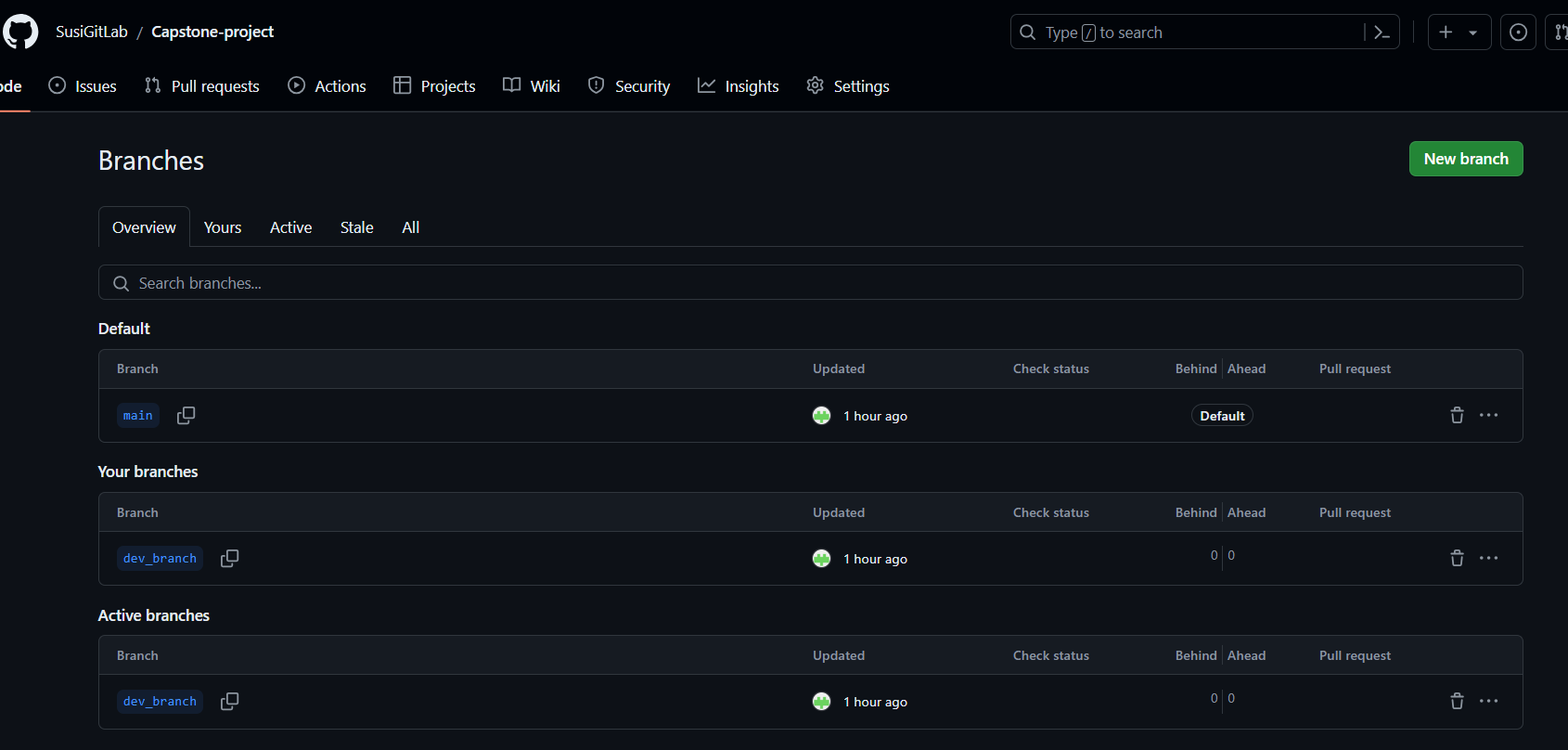
A Docker image is created, containerized, and runs on port 3000. This image serves a React.js application using Nginx as a web server. Whenever there's a change in the application code, it's pushed to GitHub. This triggers a Jenkins job that builds and deploys the application. Finally, the latest Docker image is pushed to Docker Hub.

**Deployed Site URL:** [**http://13.232.162.125:3000/**](http://13.232.162.125:3000/)

**GIT Repo URL:** [**https://github.com/SusiGitLab/Capstone-project.git**](https://github.com/SusiGitLab/Capstone-project.git)

Added .gitignore file to avoid unnecessary files and .dockerignore file to maintain minimal size, focuses on application.

Two branches created, main branch as prod repo and dev\_branch as dev repo.

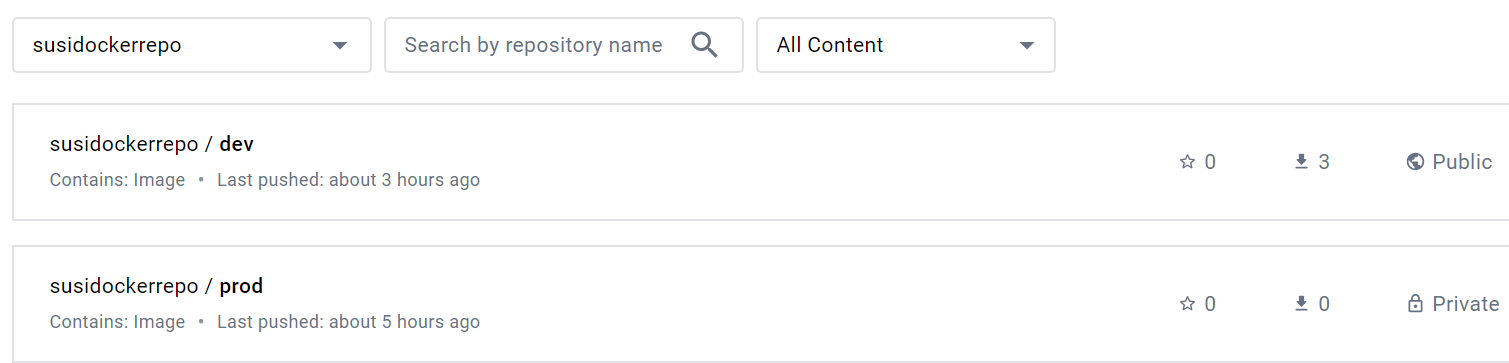


**Docker Hub Repo:**

Created docker hub repo dev and prod to push my changes. Dev repo is a public repo, any one can download and use my docker image and Prod repo is a private repo, no one can download my docker image.

**Docker image:** docker pull susidockerrepo/prod:latest

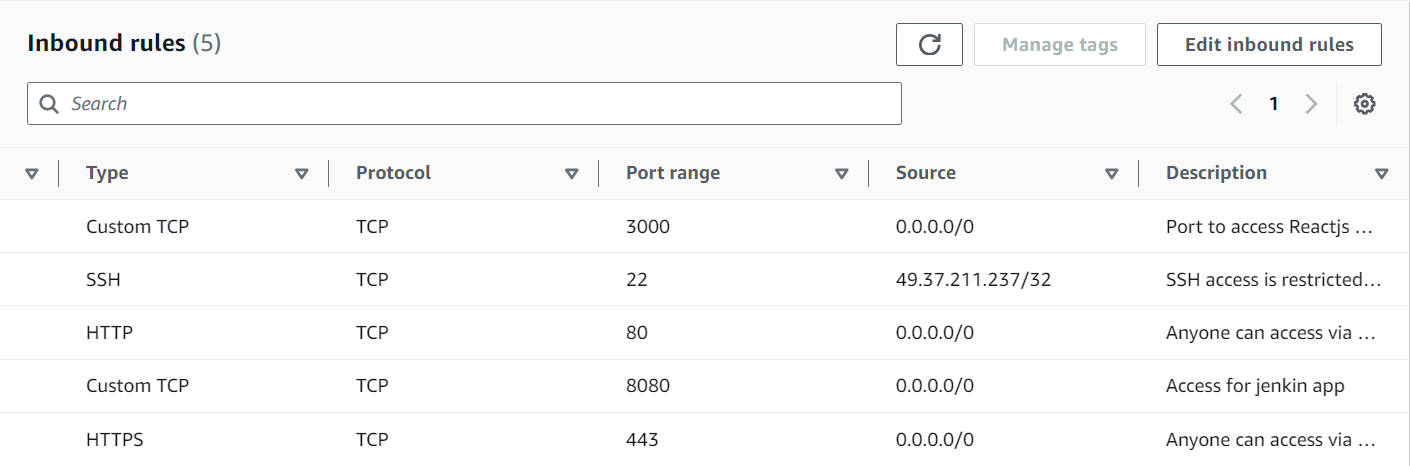
docker pull susidockerrepo/dev:latest

****

**AWS Config:**

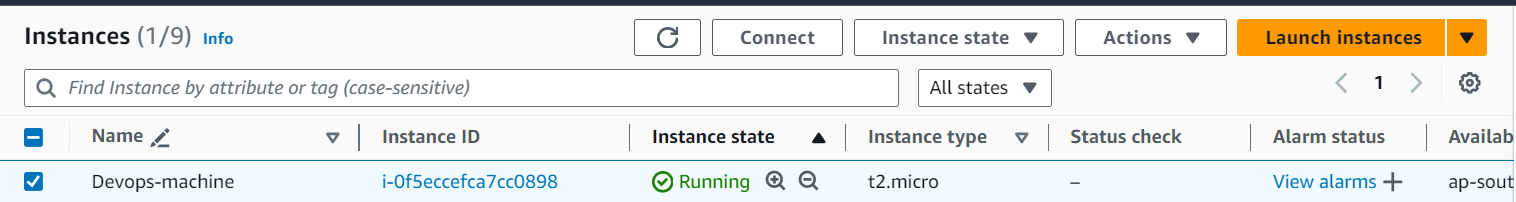
**Create Security Group:**

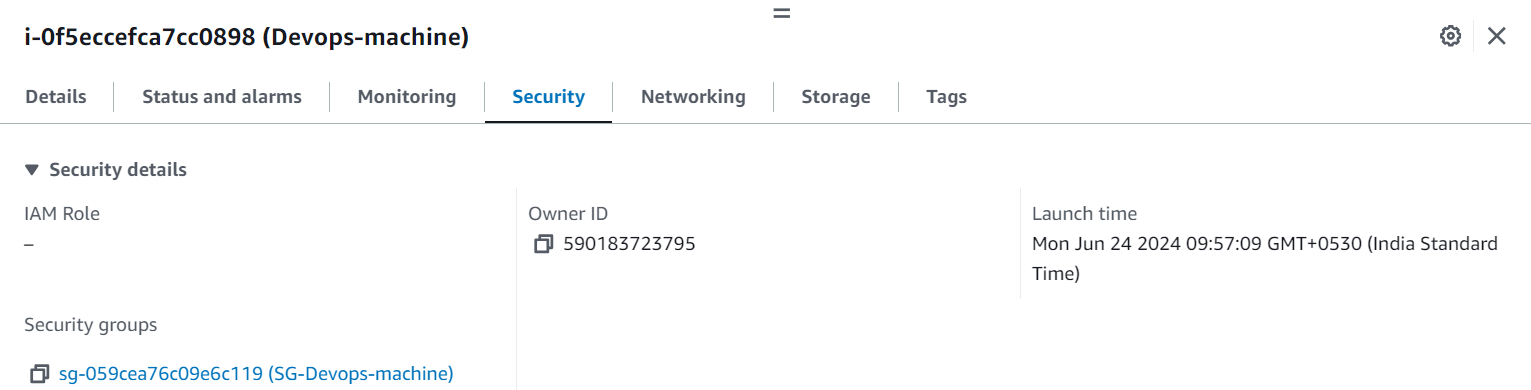
Security group created for the DevOps machine with inbound rules that will allow access to the application for all users. However, only my IP can be used to log in to the server.****

****

**Launch t2.micro instance:**

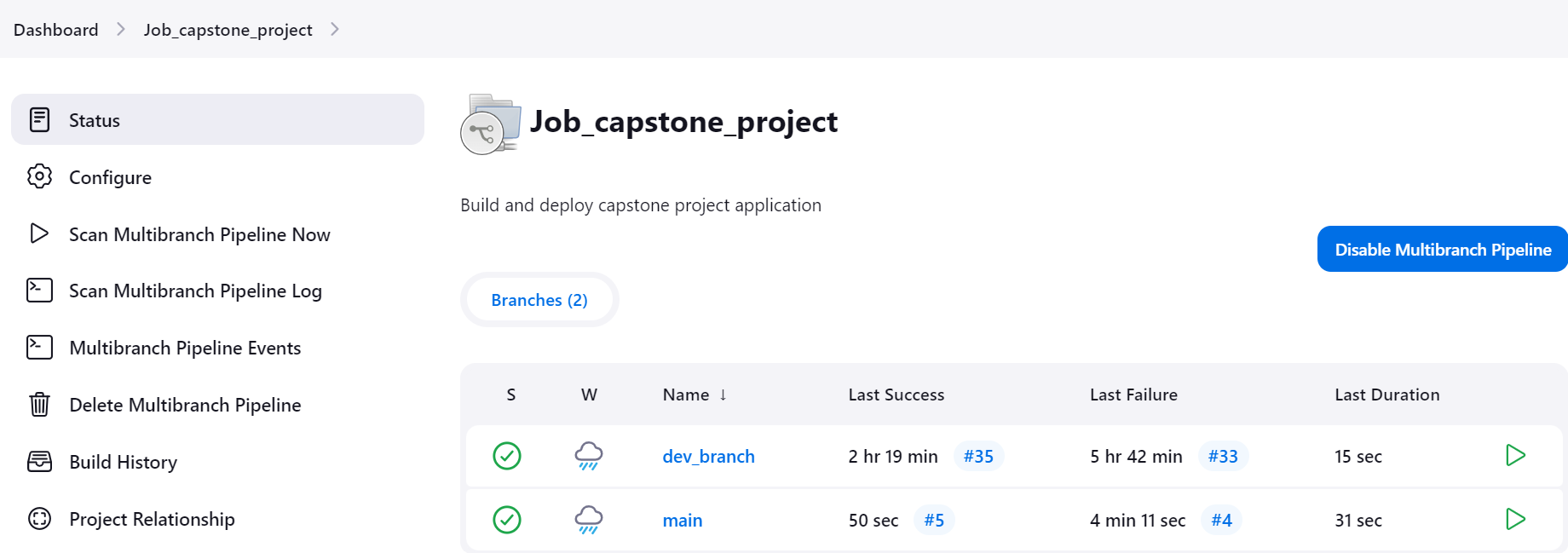
EC2 machine launched using the DevOps security group.

****

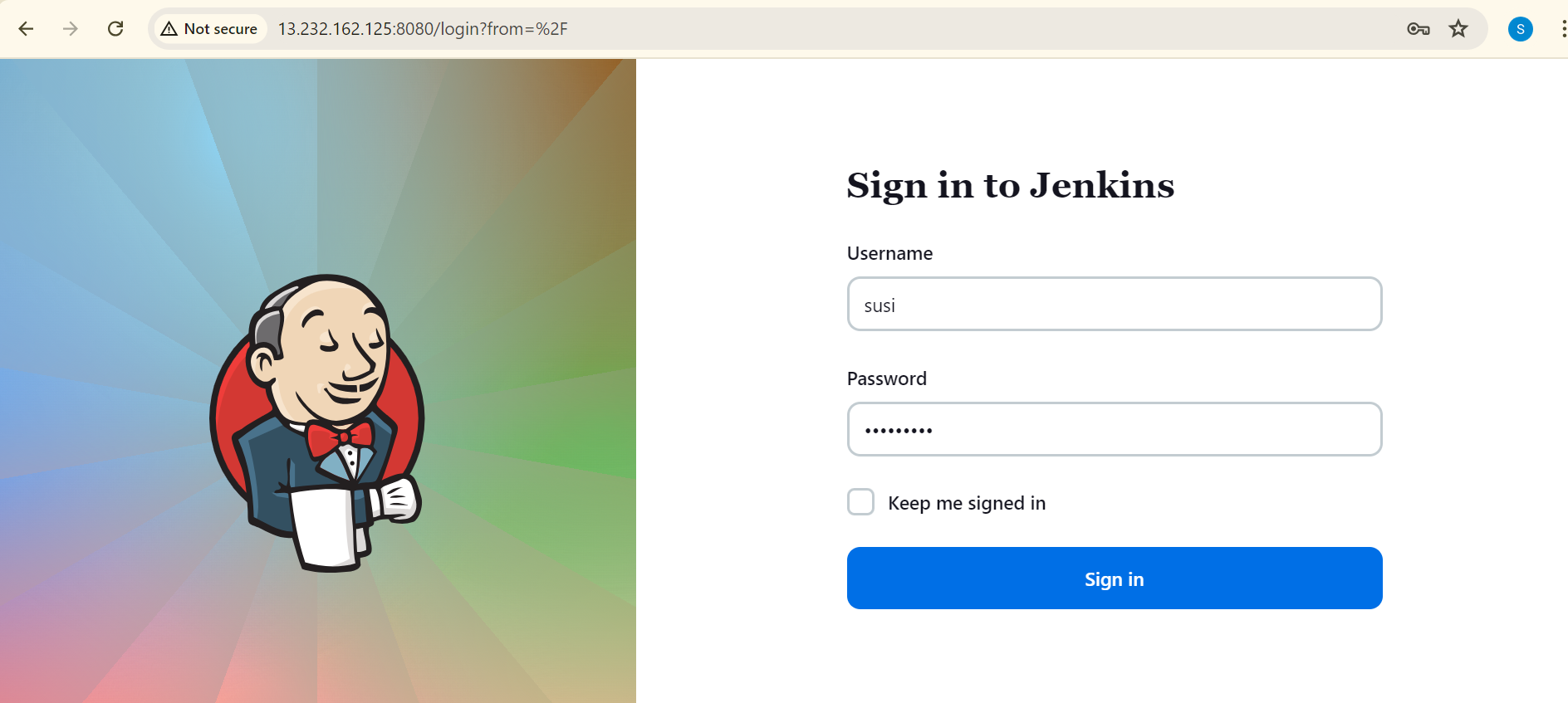


**Jenkins:**

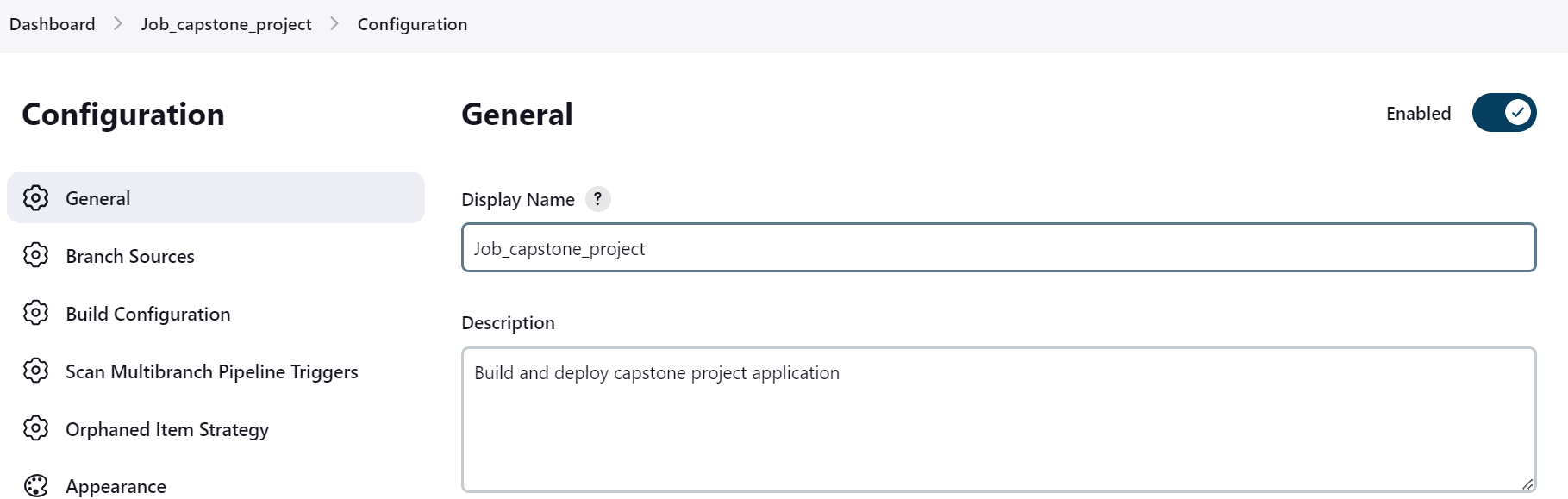
Created a multibranch project in Jenkins and configured the Git URL. Whenever there is a new push to the dev\_branch in GitHub, the dev\_branch job will be triggered and push the Docker image to the dev repository in Docker Hub. Similarly, whenever there is a new push to the main branch in GitHub, the main job will be triggered and push the Docker image to the prod repository.



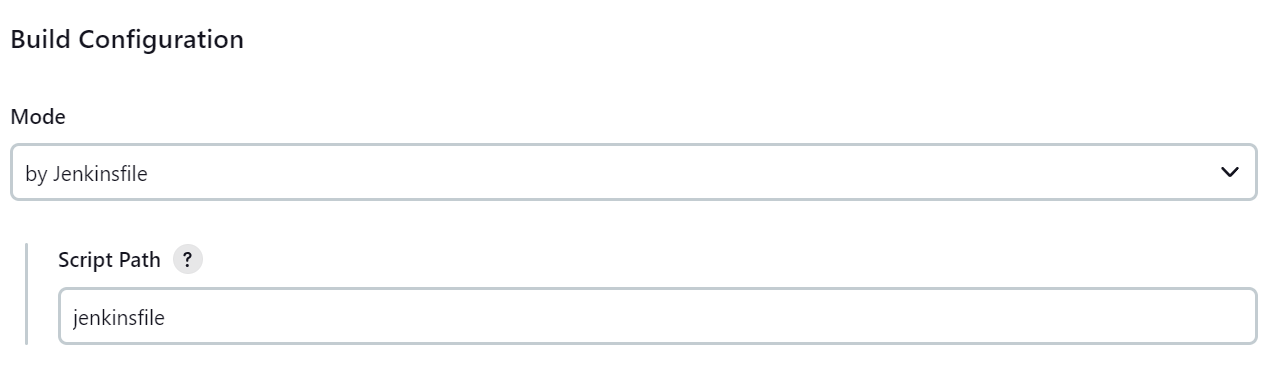
**Login Page:**

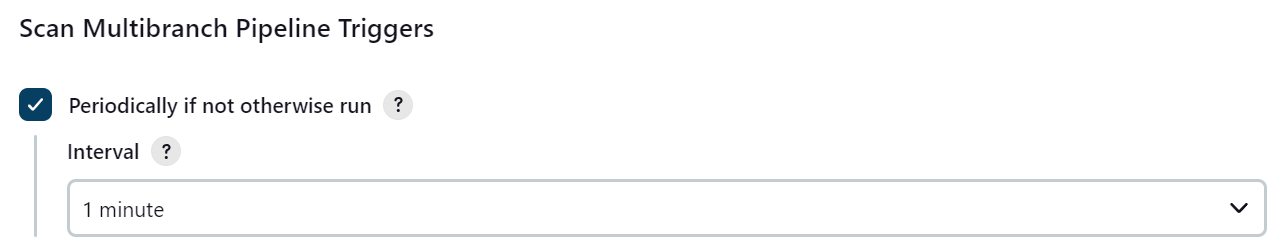
****

**Job Configuration:**



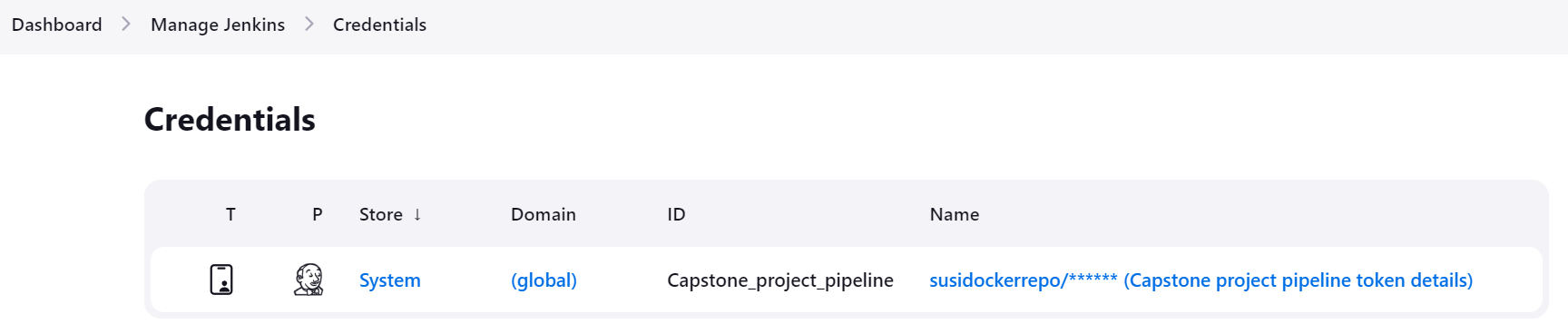






When we created multibranch project , it will automatically scan for multi branch and create multibranch pipeline for all available branches in GitHub

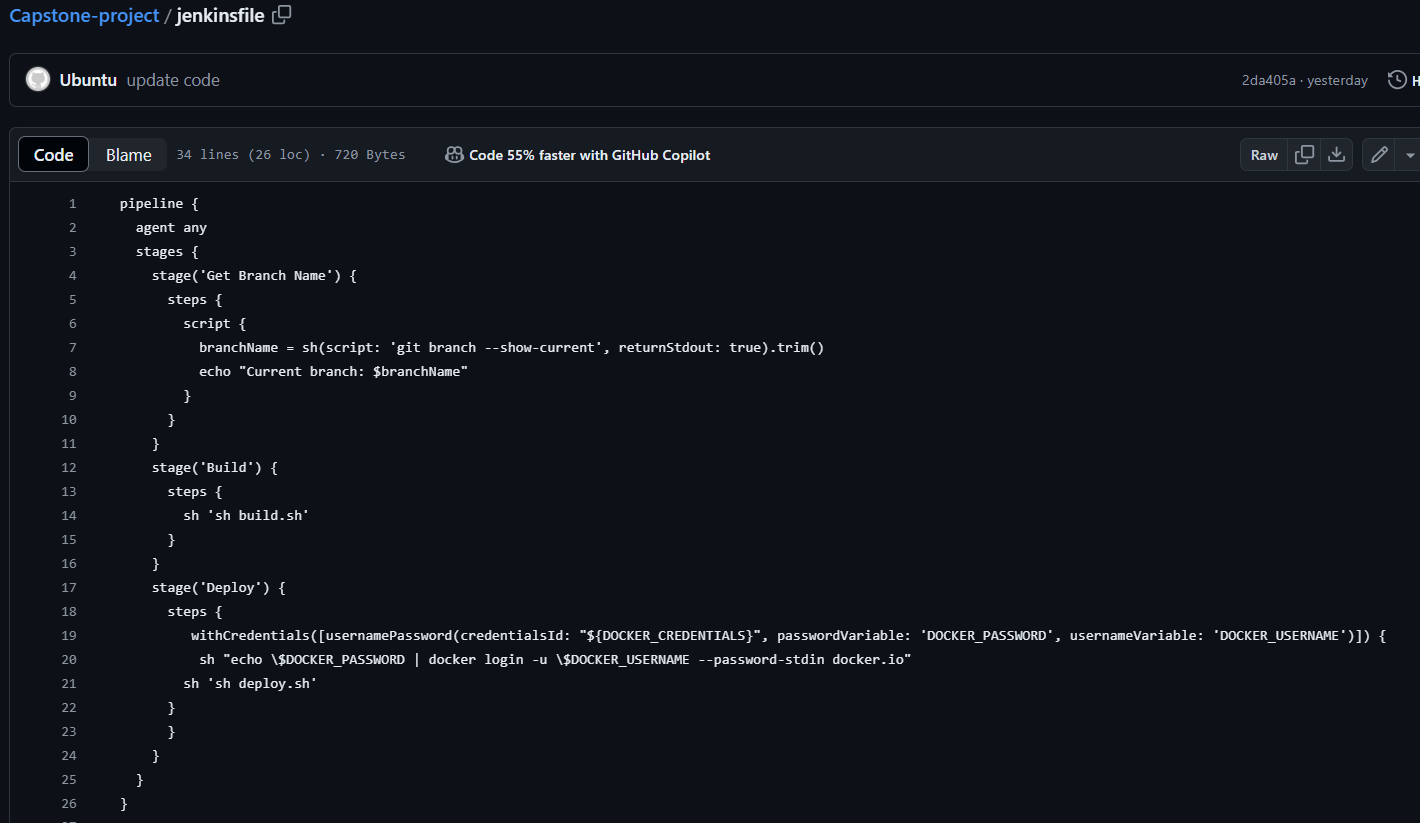
**Docker Hub Registry Credentials configuration in jenkins:**

****

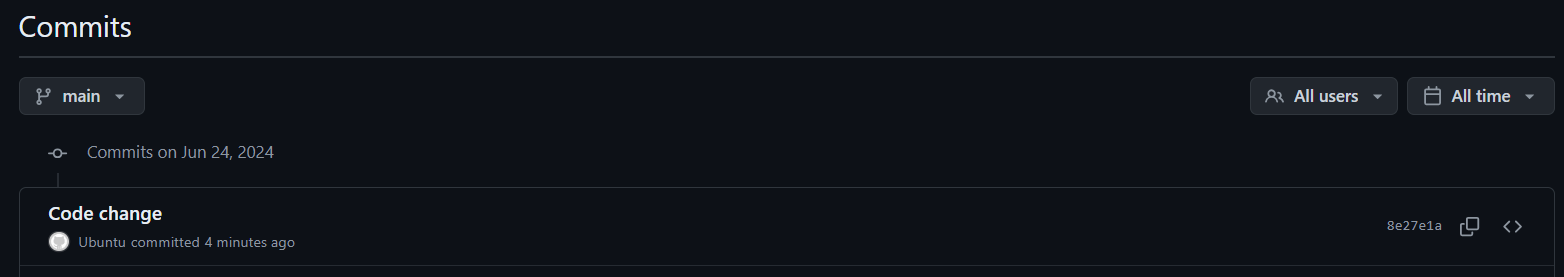
**Execute shell command:**

Used those credentials for Docker Hub login in jenkinsfile.

Jenkinsfile hold the execution steps.

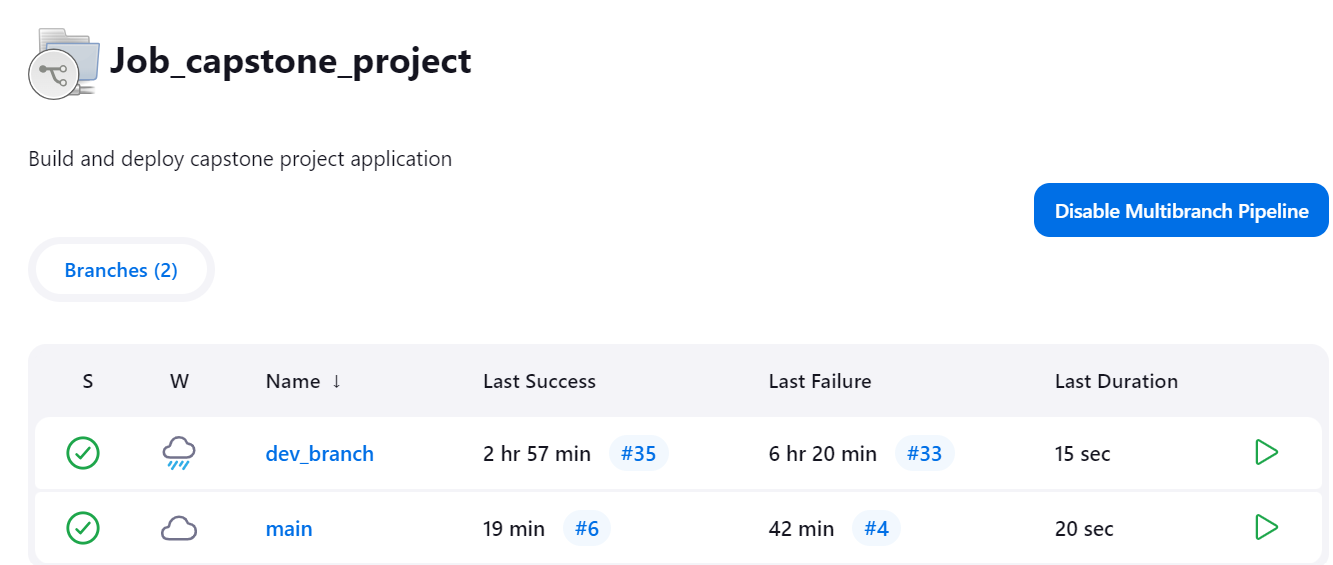
****

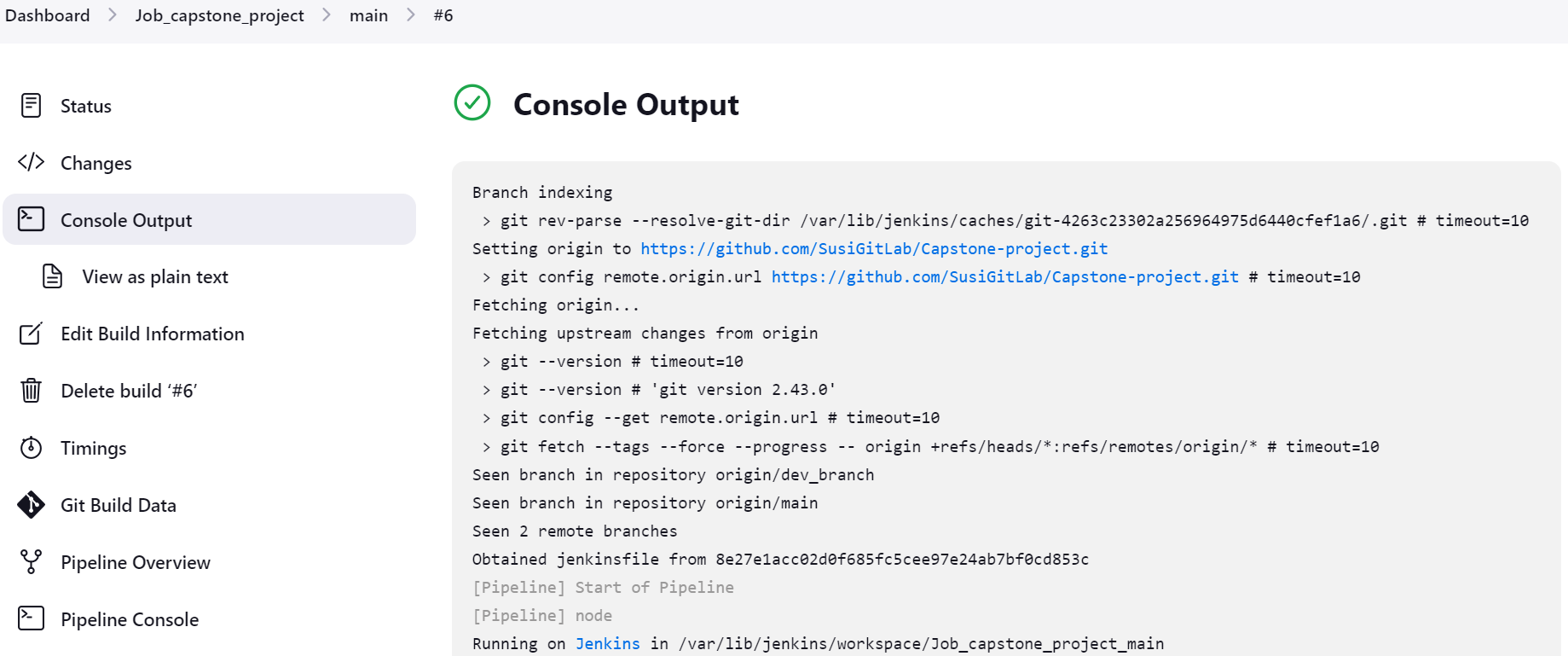
**Code change pushed to main branch:**

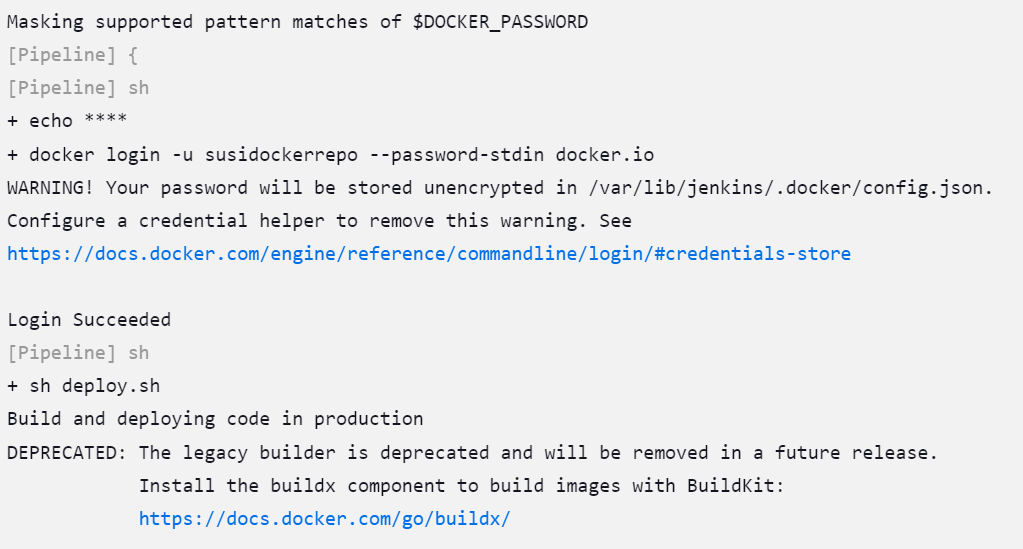


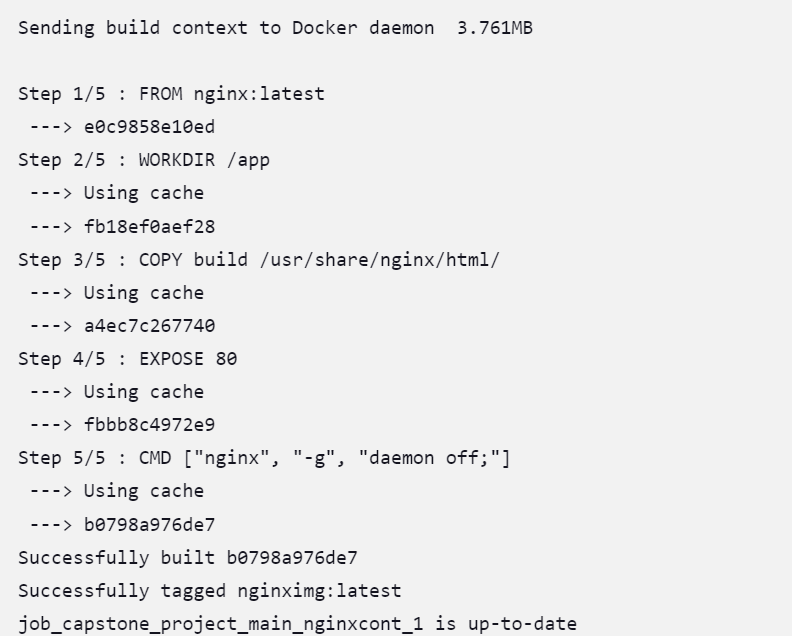
**Main branch pipeline will get automatically triggered and docker images will be pushed to Docker Hub Registry:**

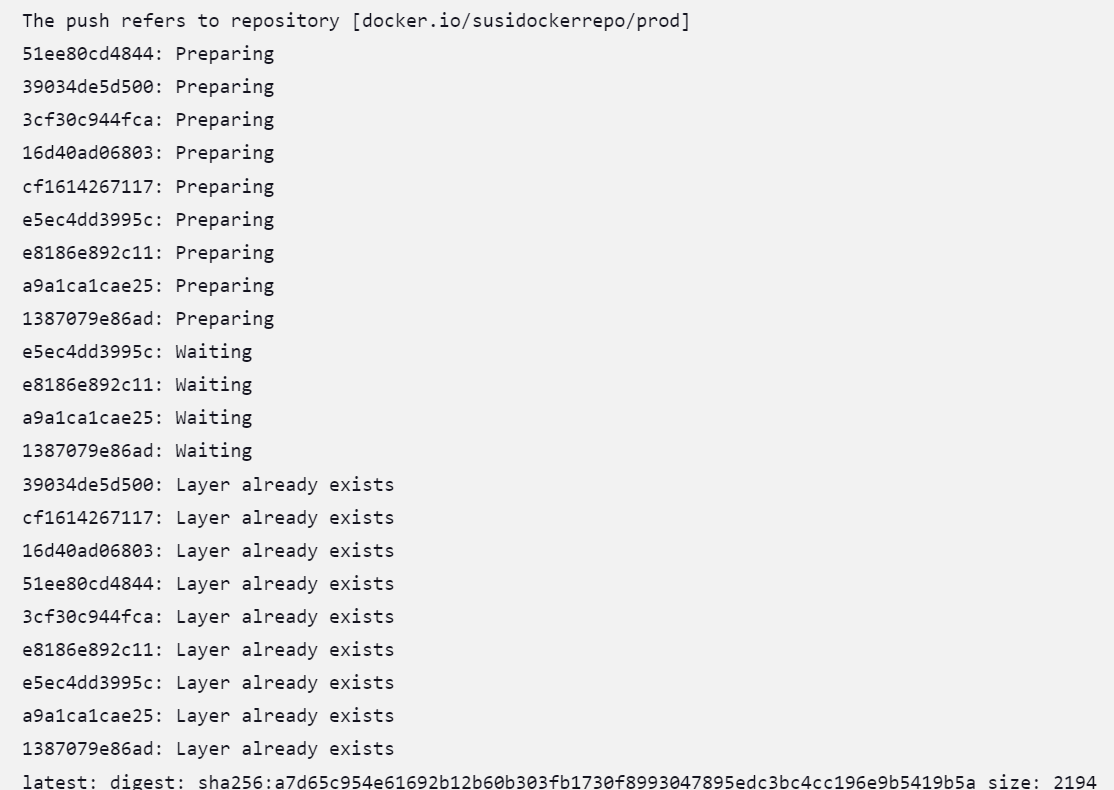
Only main branch job got triggered

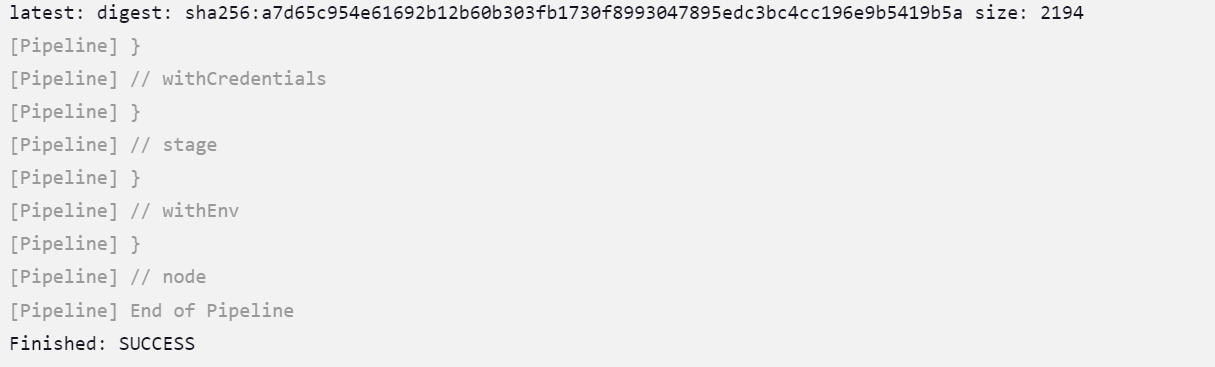
****



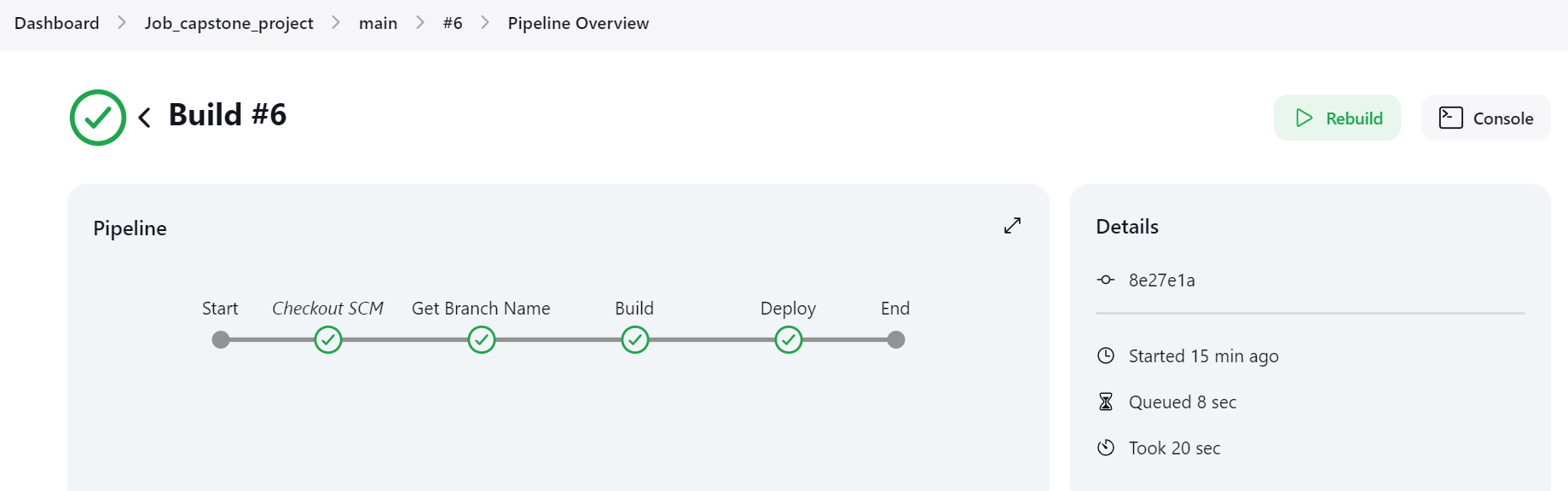




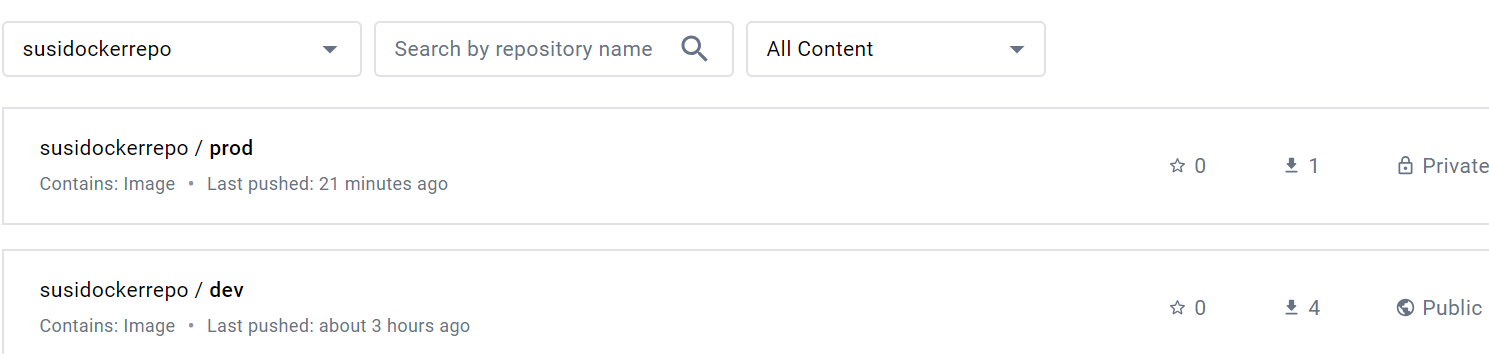




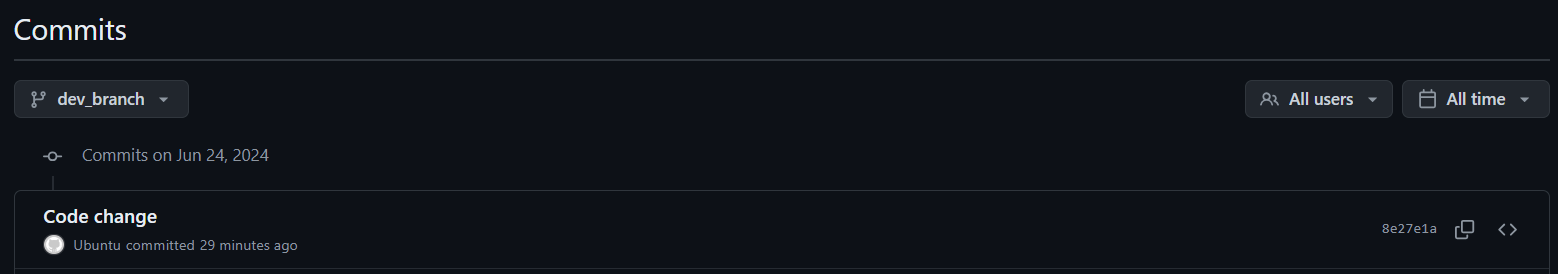
**Pipeline output overview:**



**Changes pushed only to prod repo:**

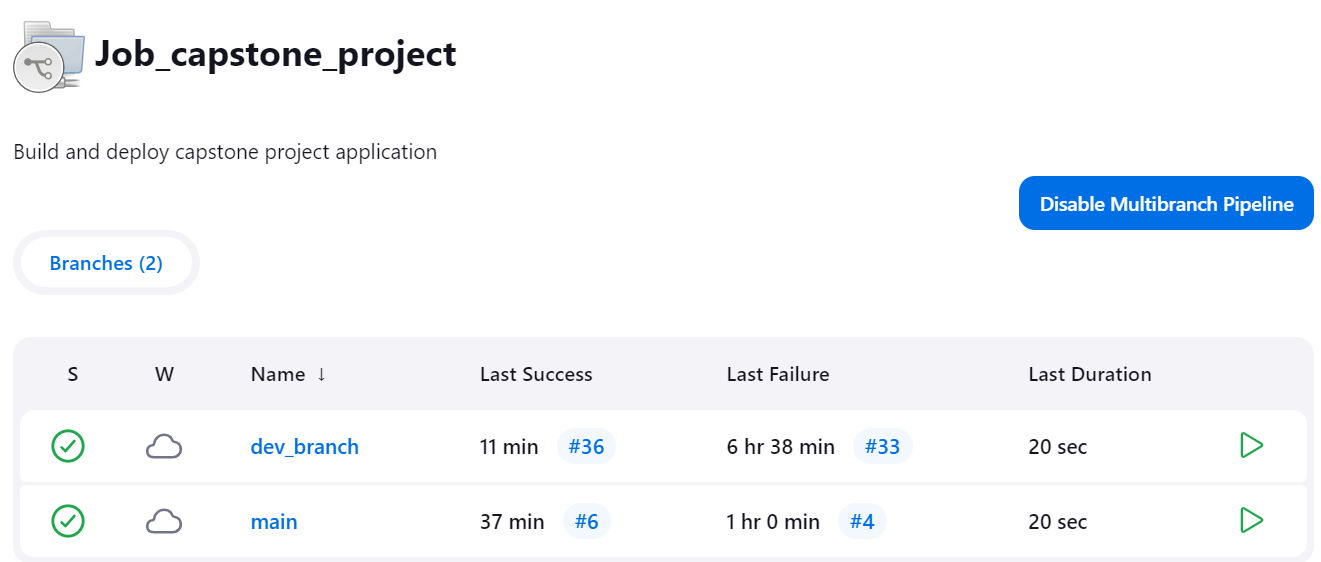


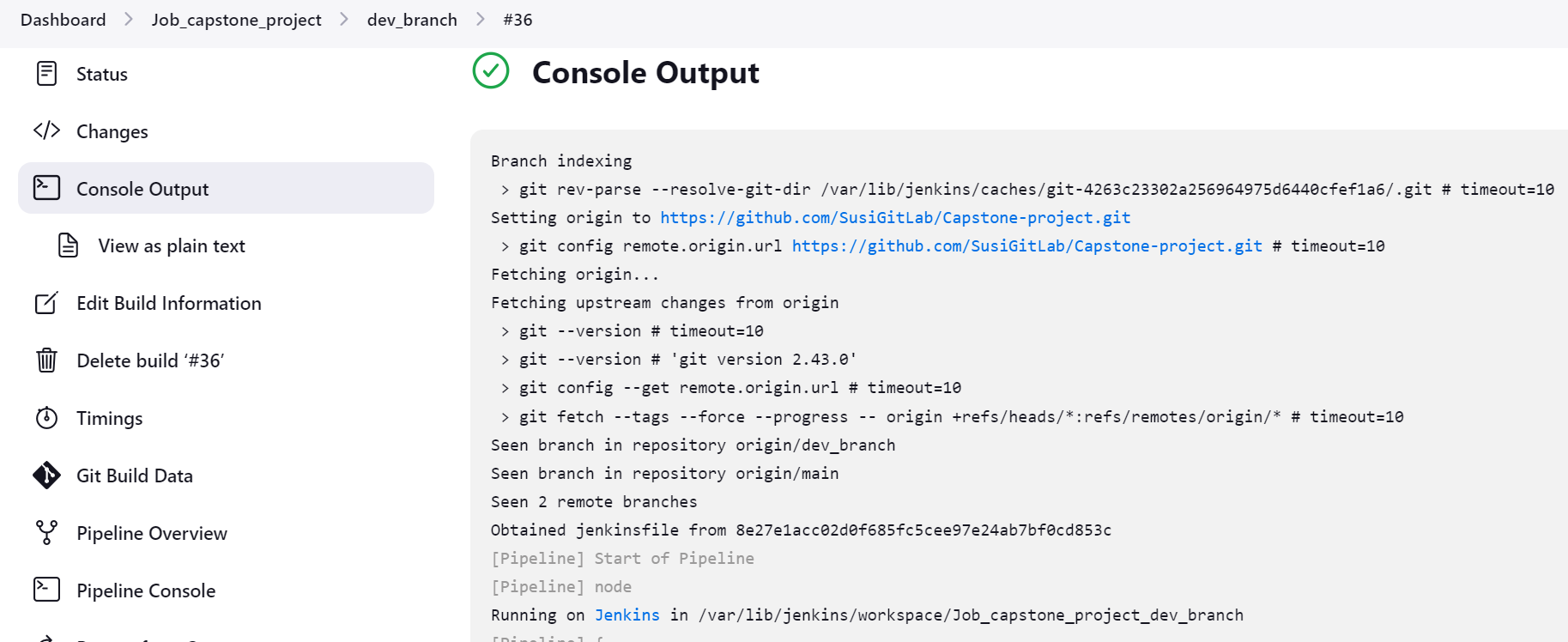
**Code change pushed to dev\_branch:**

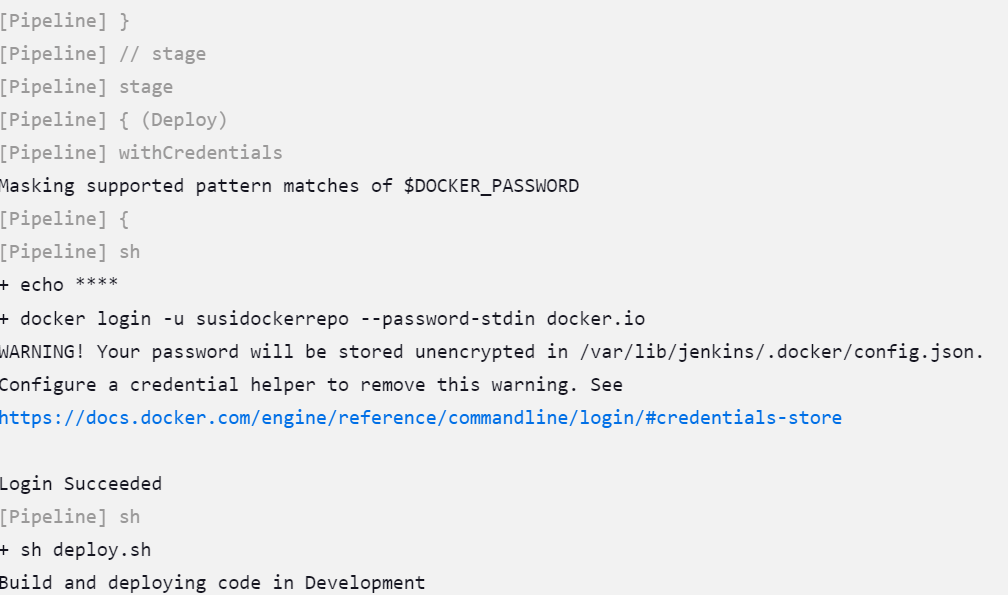


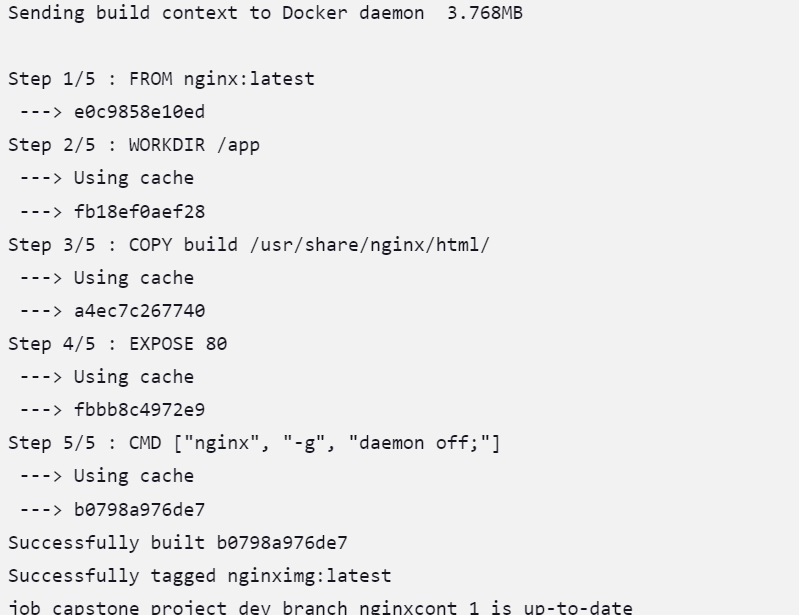
**Dev branch pipeline will get automatically triggered and docker images will be pushed to Docker Hub Registry:**

Only dev\_branch job got triggered

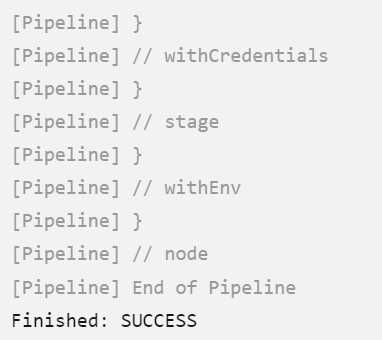




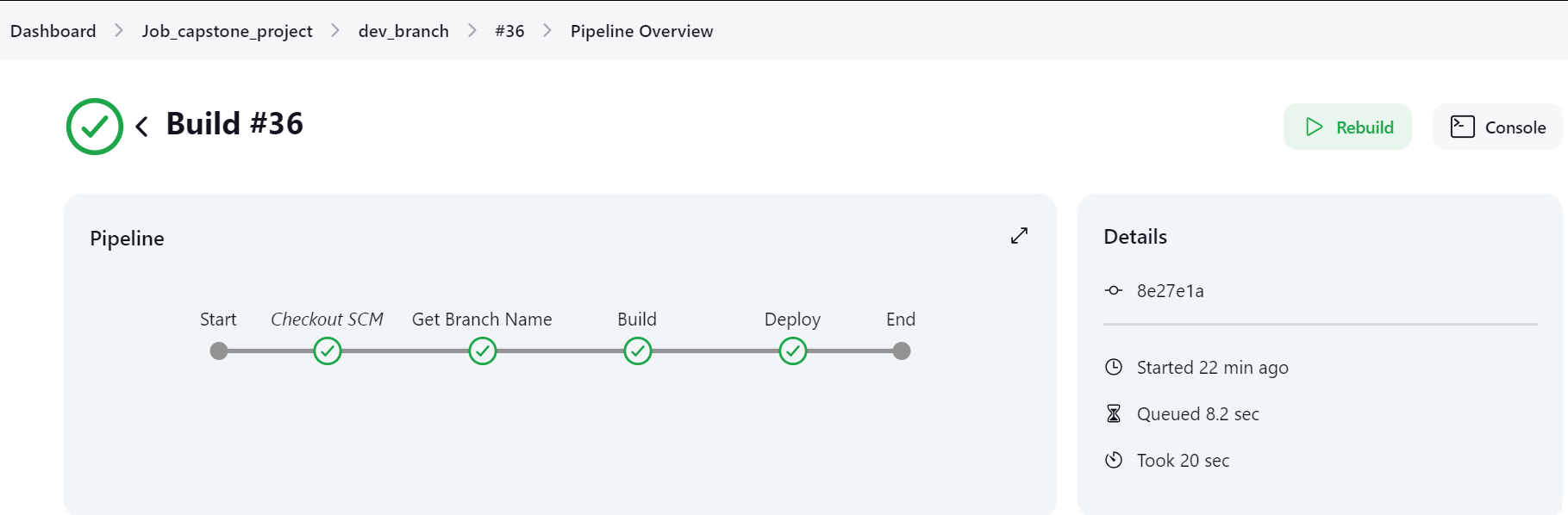




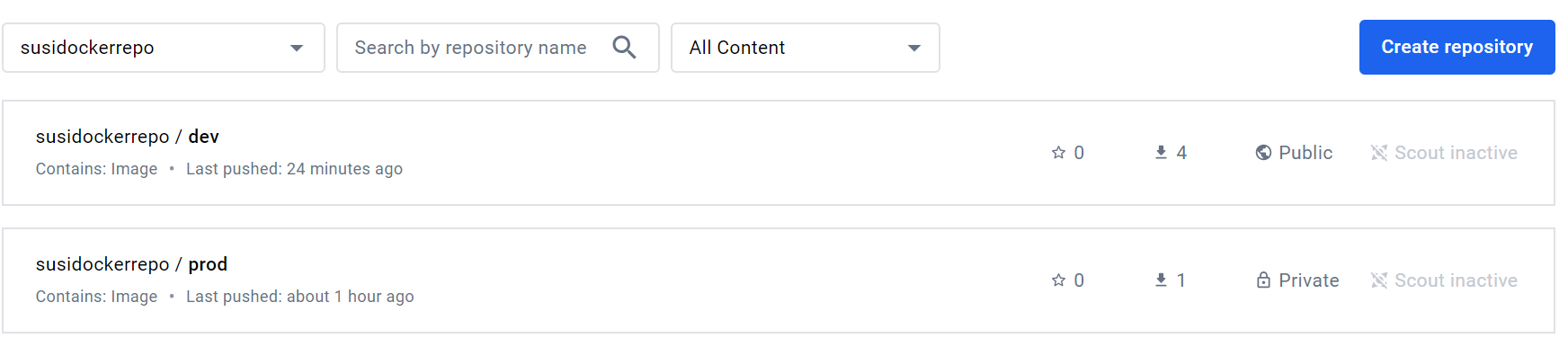




**Pipeline output overview:**

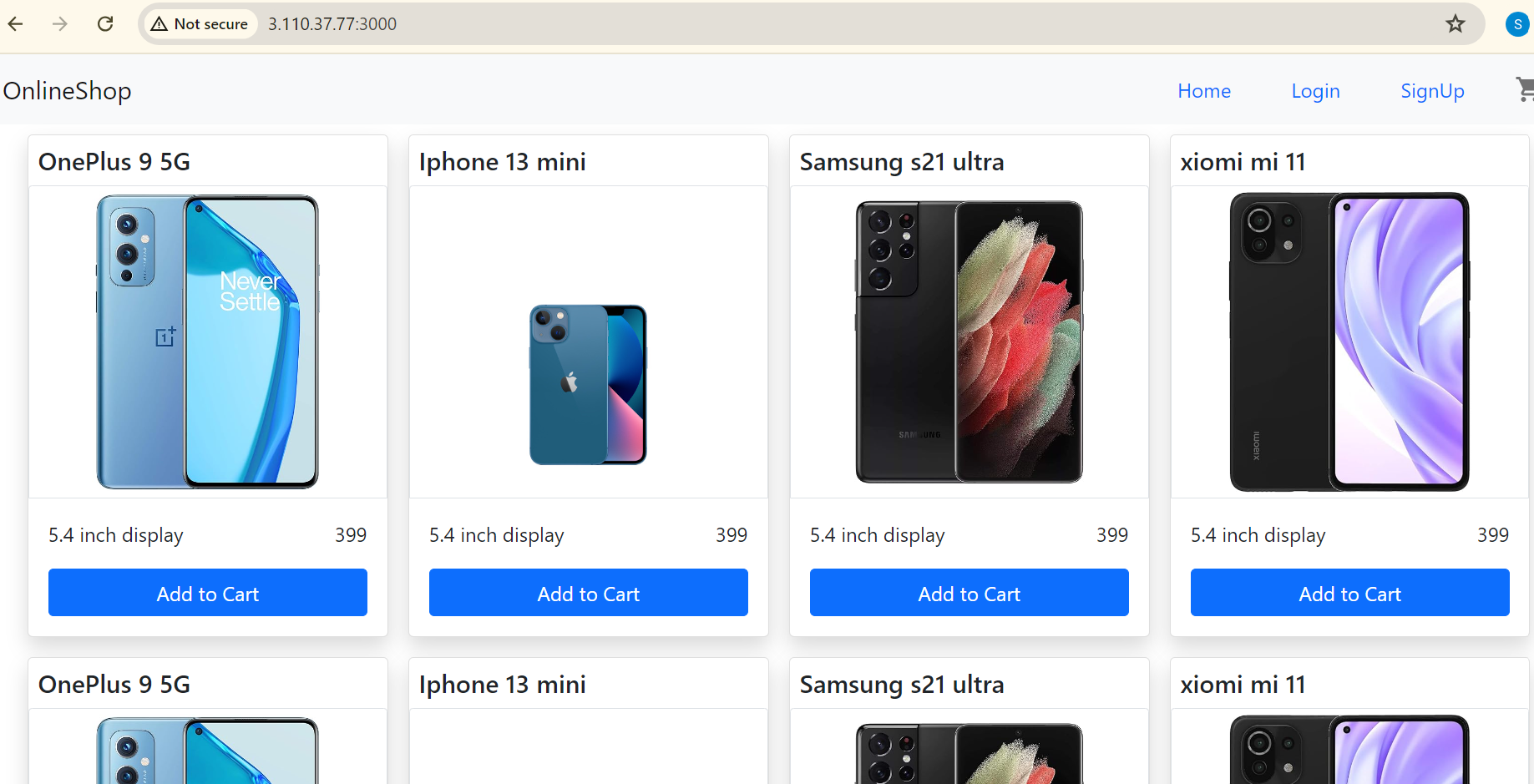


**Changes pushed only to dev repo:**

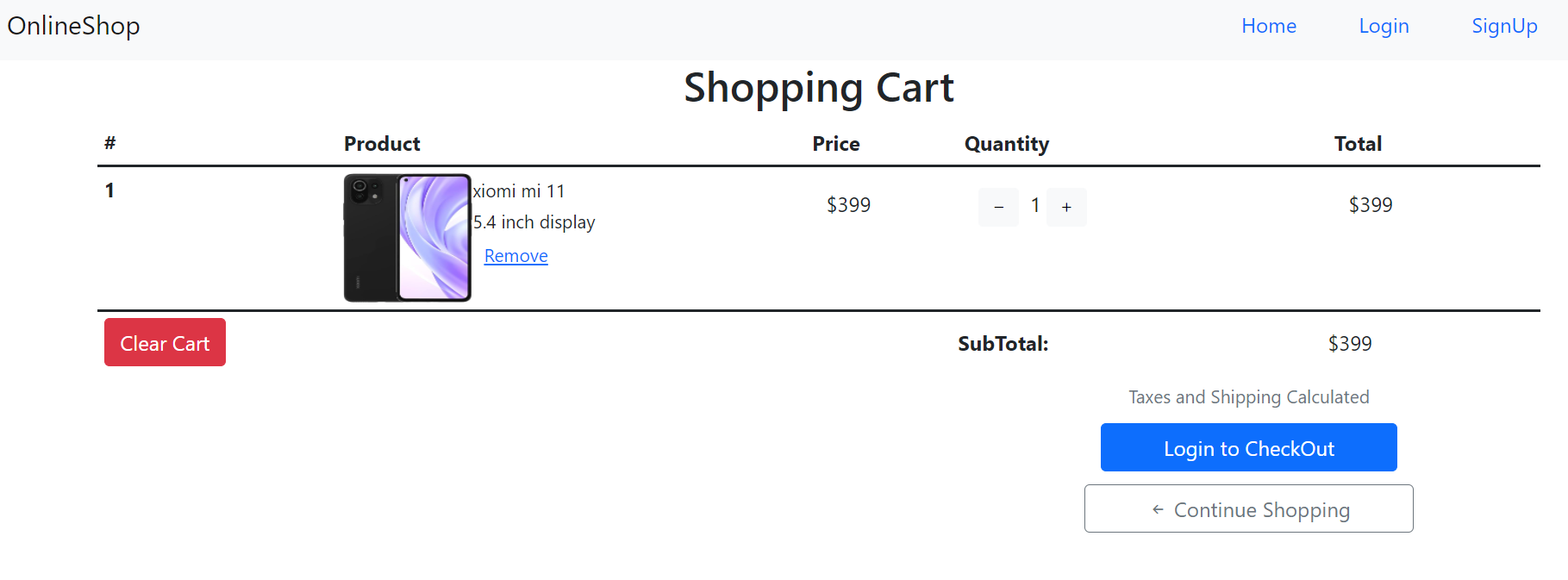
****

**React app running in browser: (**[**http://3.110.37.77:3000/**](http://3.110.37.77:3000/) **)**

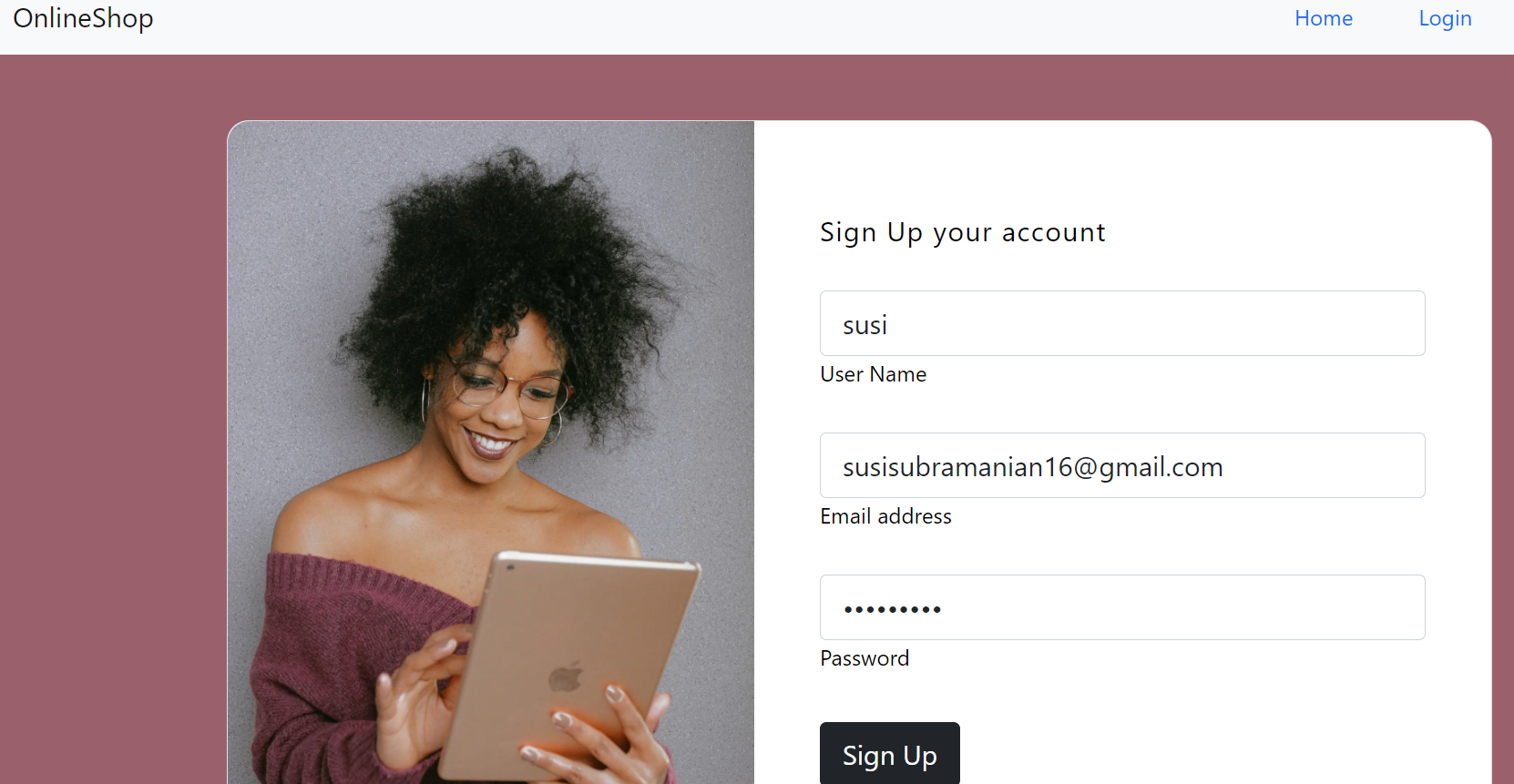
First page of an application.

****

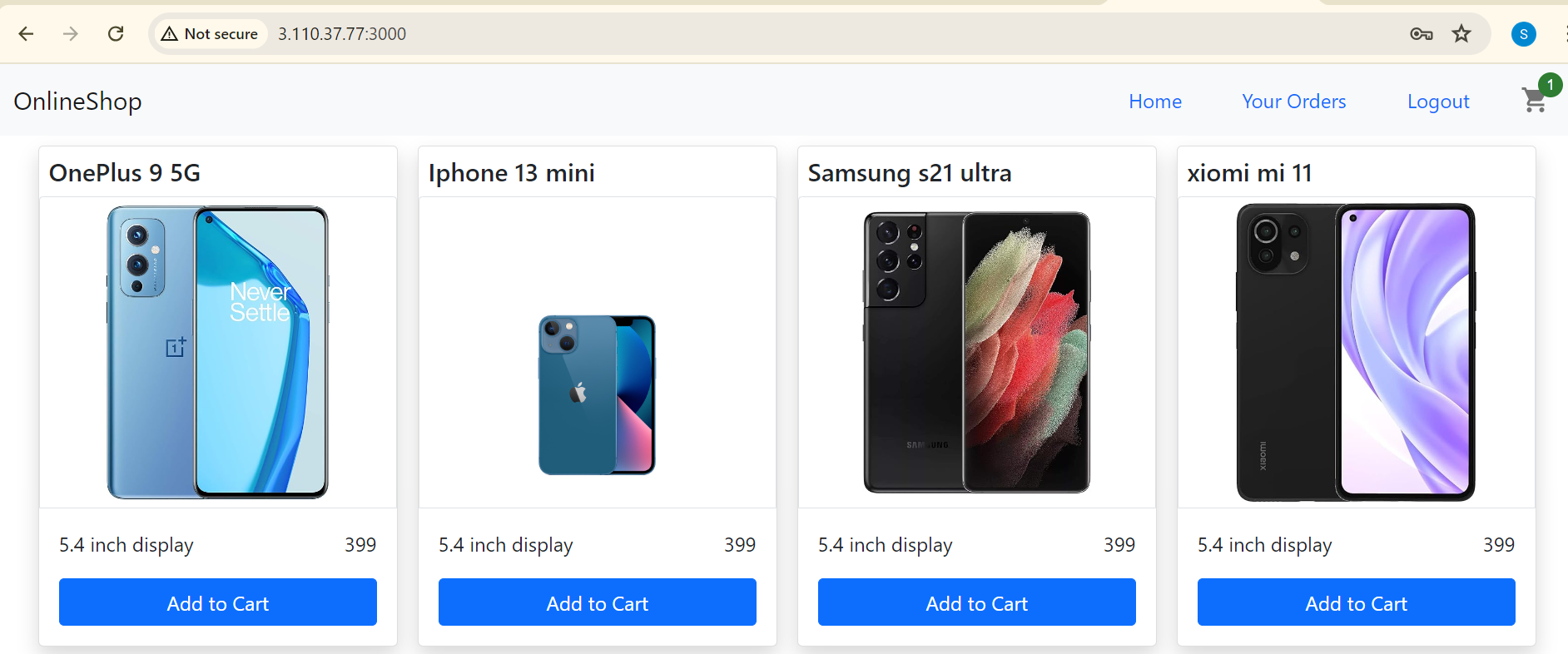
Added items to cart.

****

Sign Up to application:

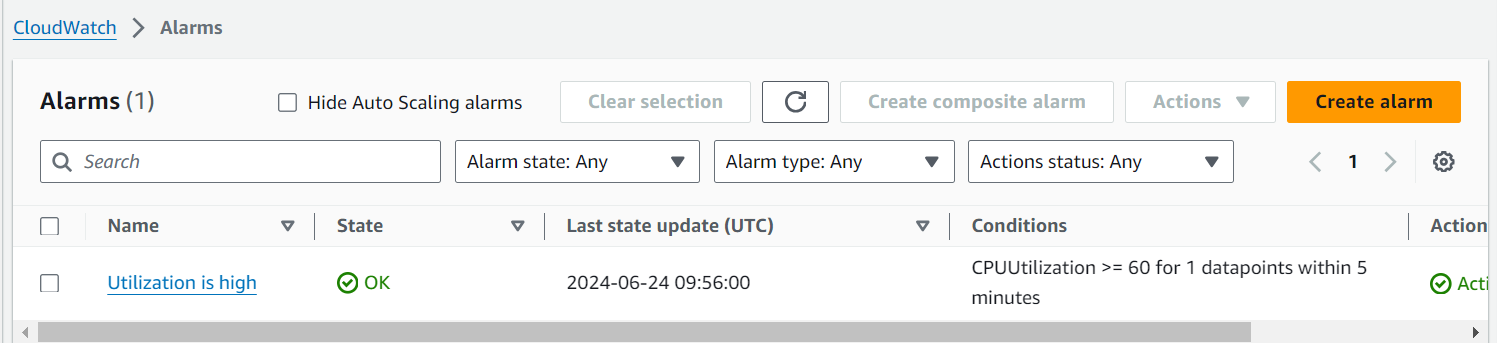
****

Logged in to an application.

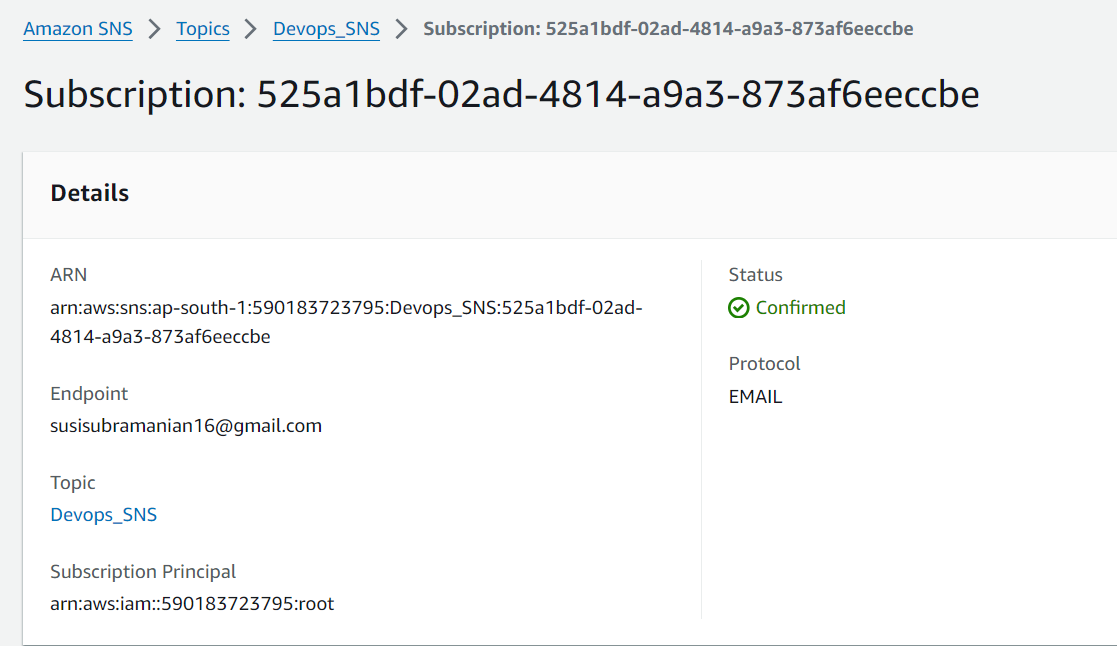
****

**CloudWatch Configuration:**

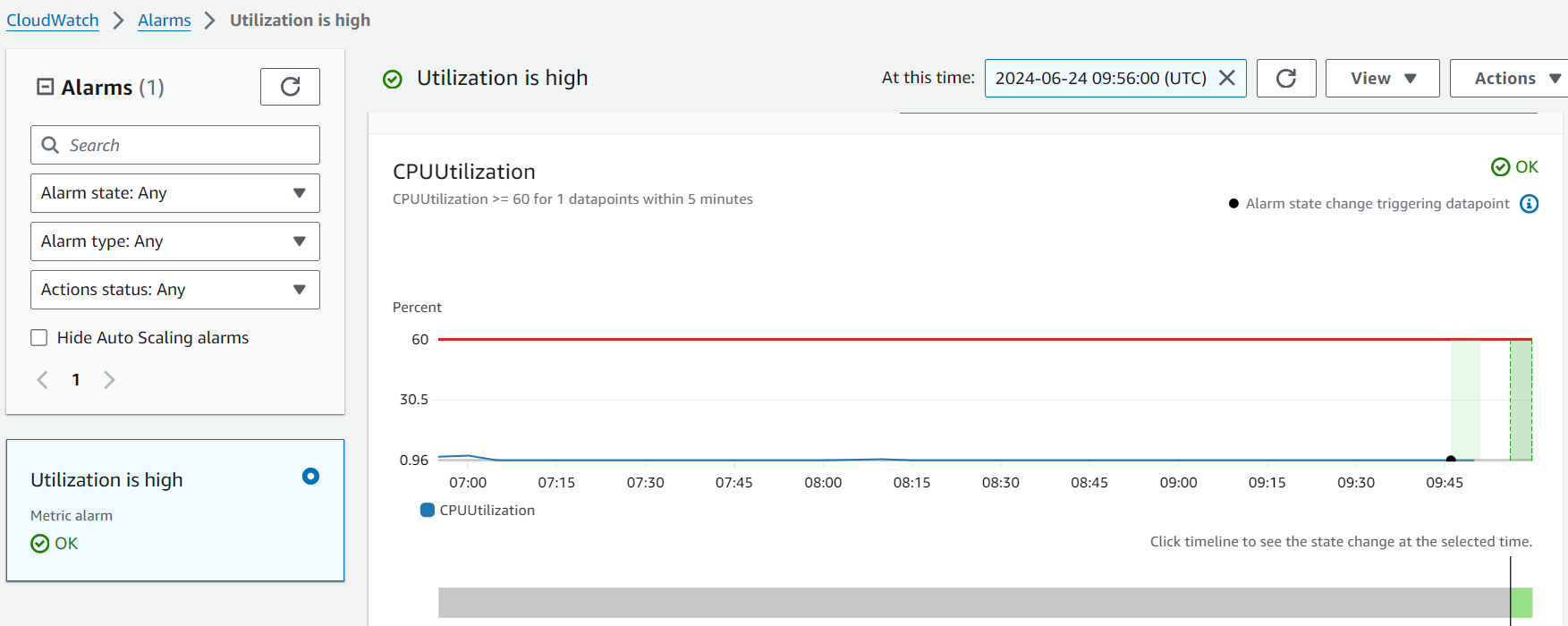
Alarm created for Cloudwatch.



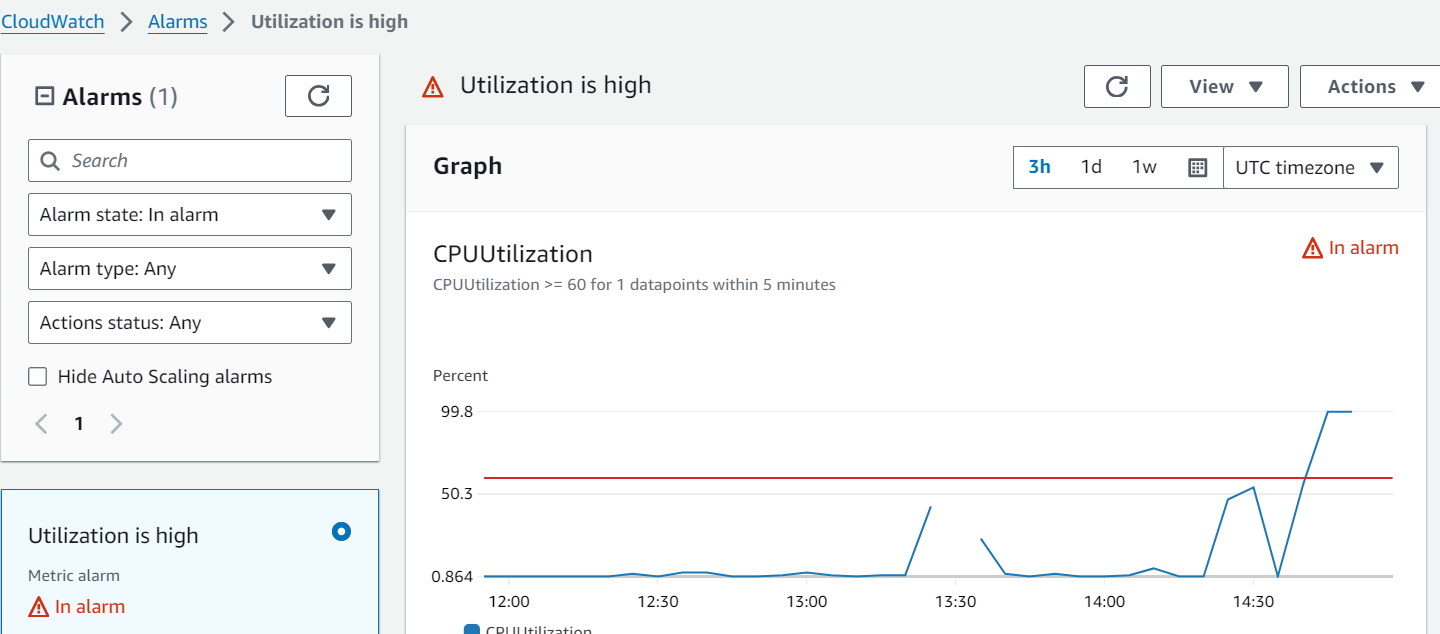
SNS topic created.

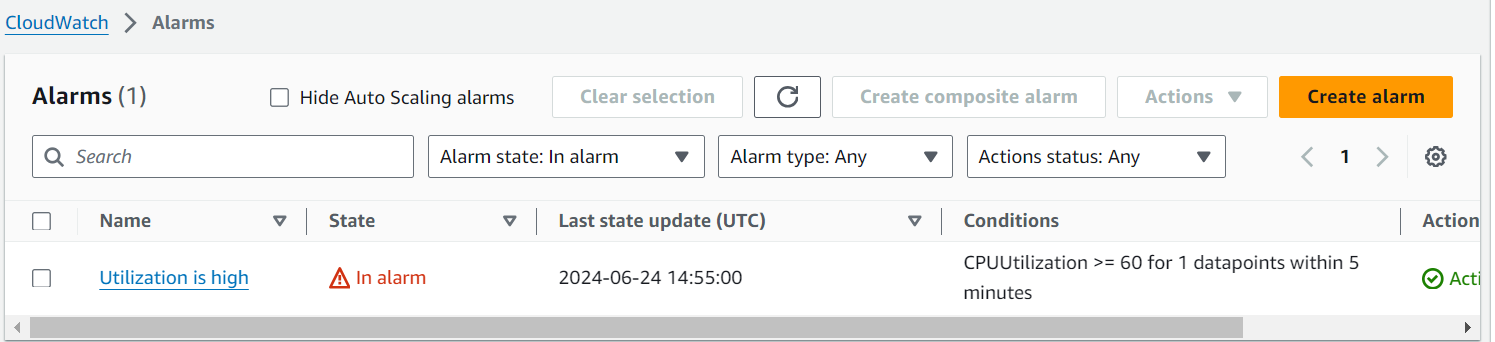


**Metrics output:**

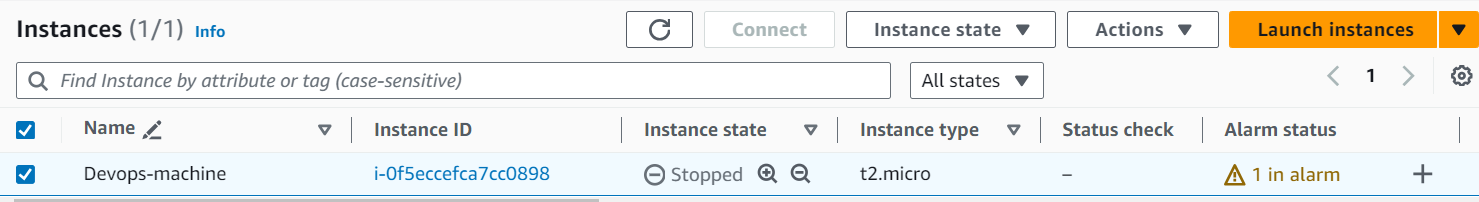


After giving stress to EC2 the alarm will get triggered.

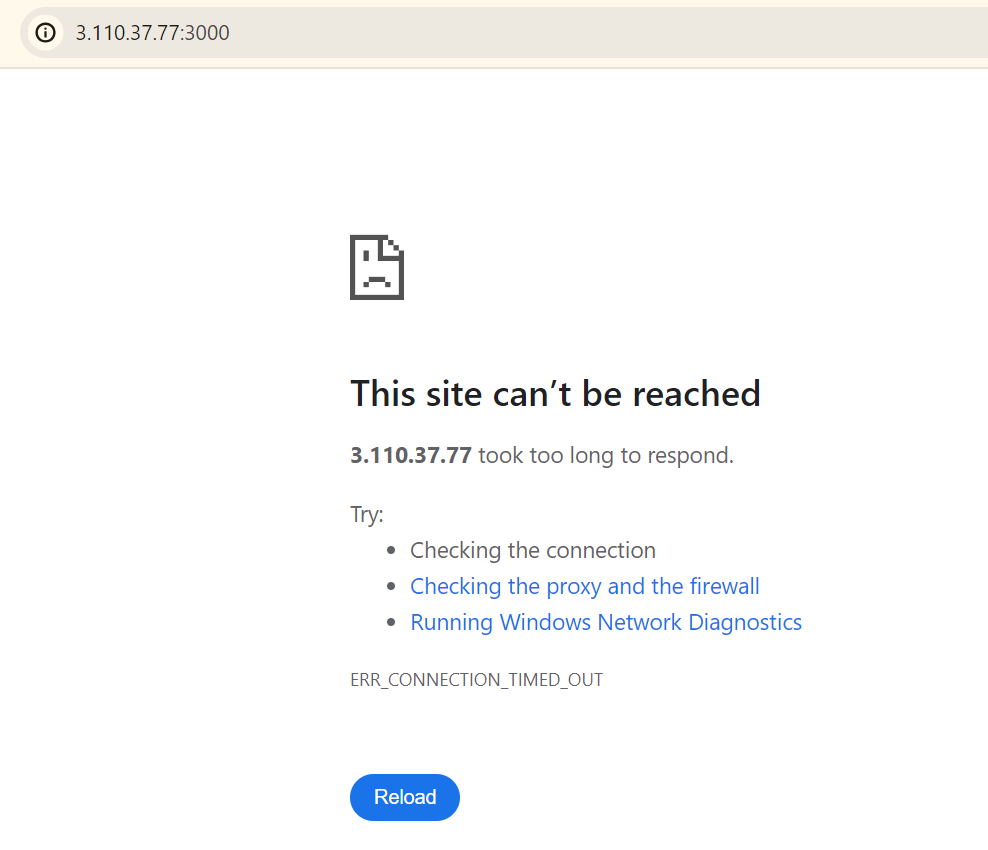




EC2 machine will be stopped.



Application will be down.



Email notification sent to topic.

