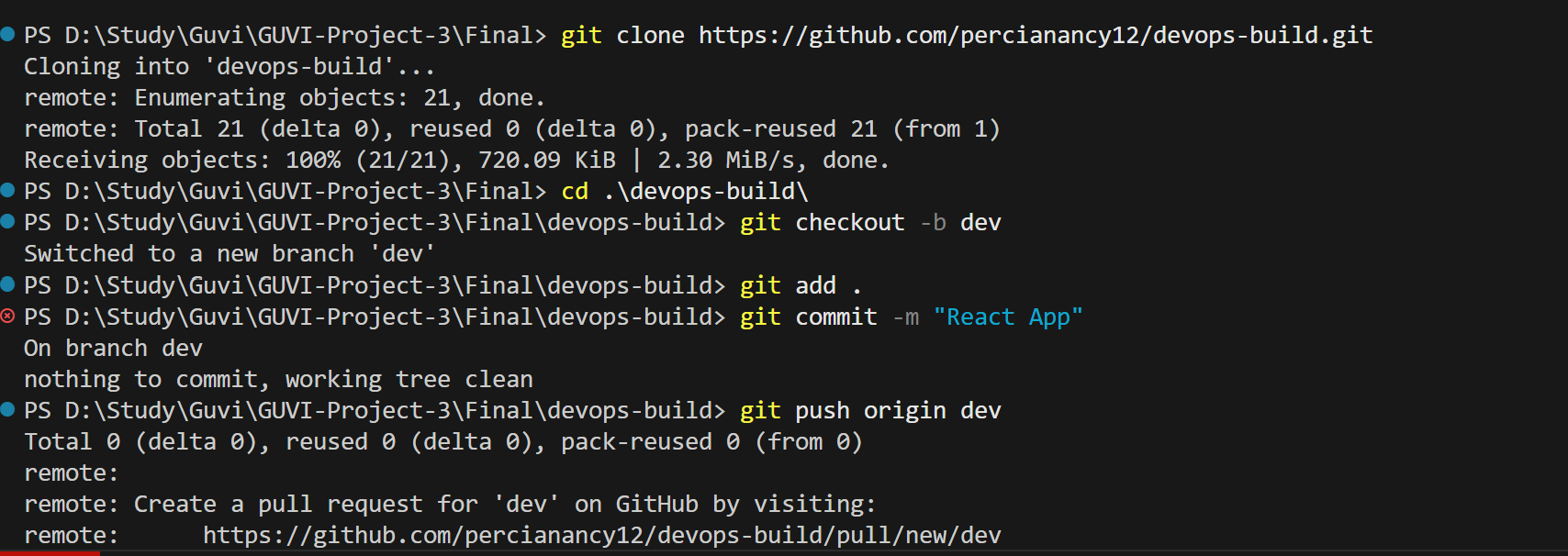
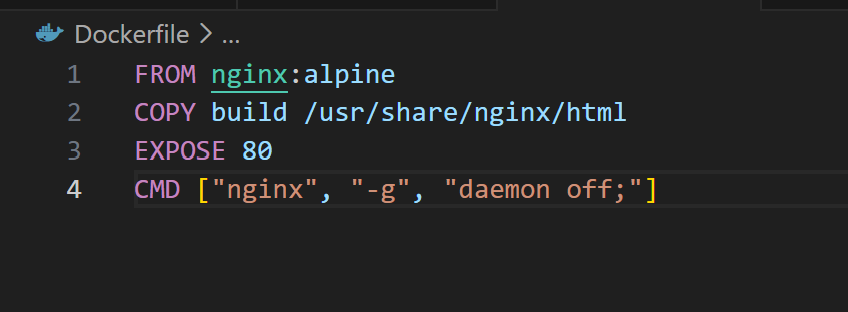
**Project-3**

**GIT**

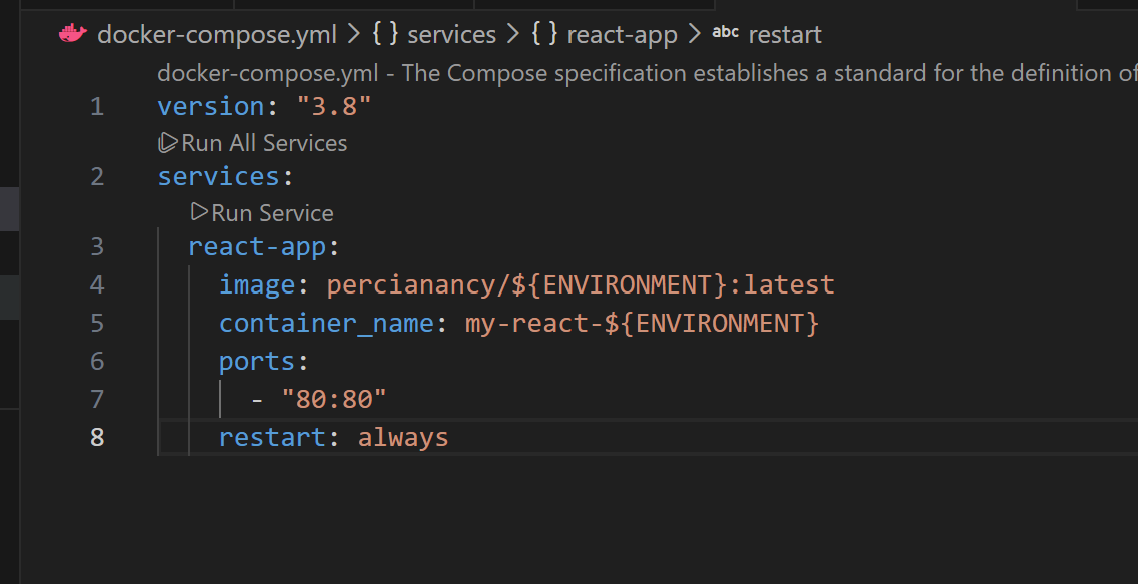


**Docker:**

Dockerize the application using Dockerfile

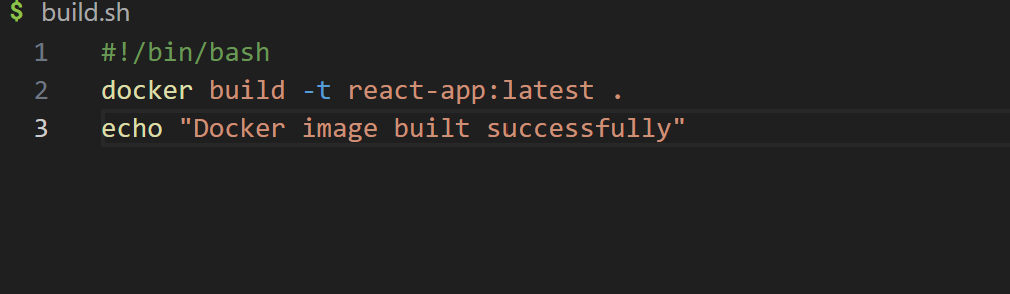


Docker-compose to use the above image

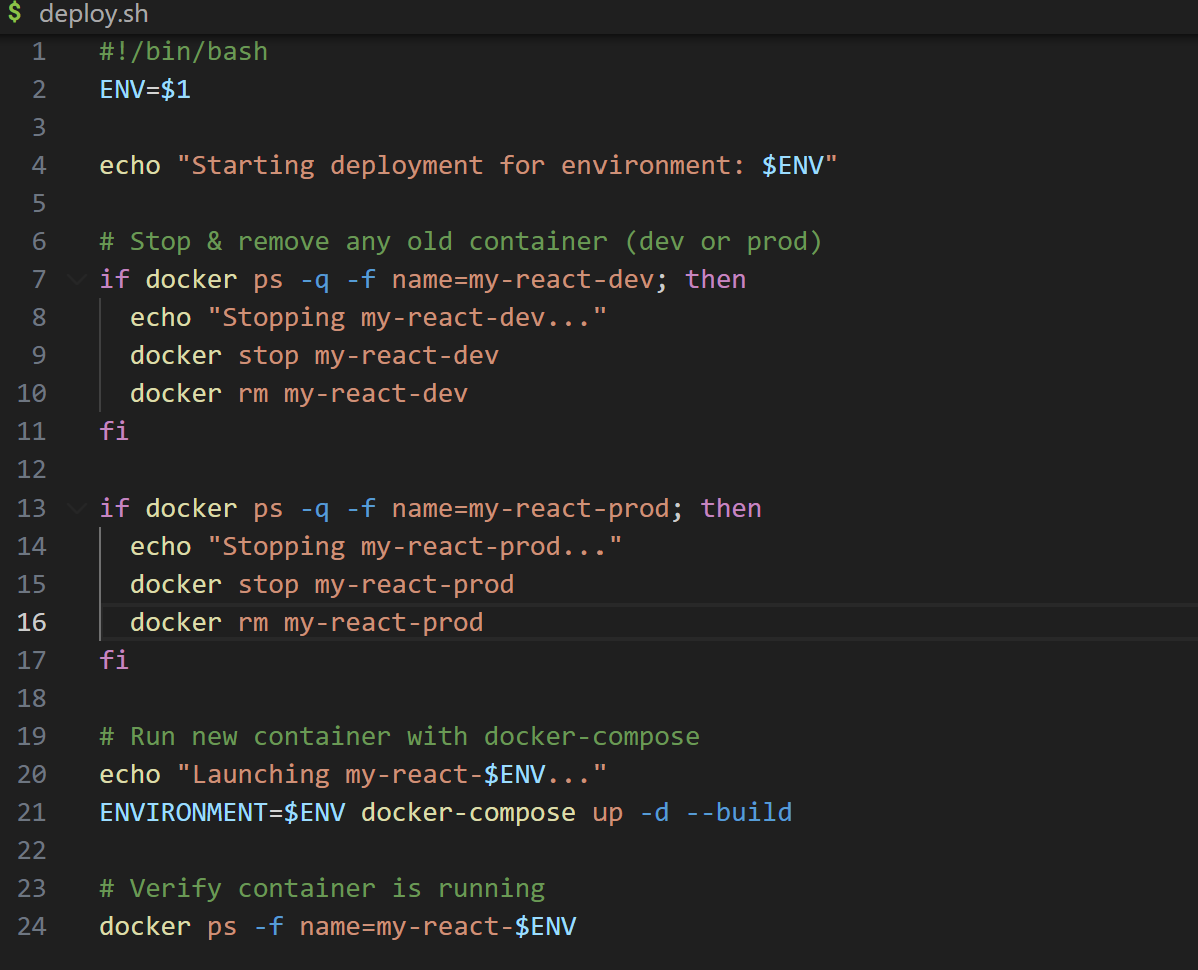


**Bash Scripting**

Build.sh – for building docker images

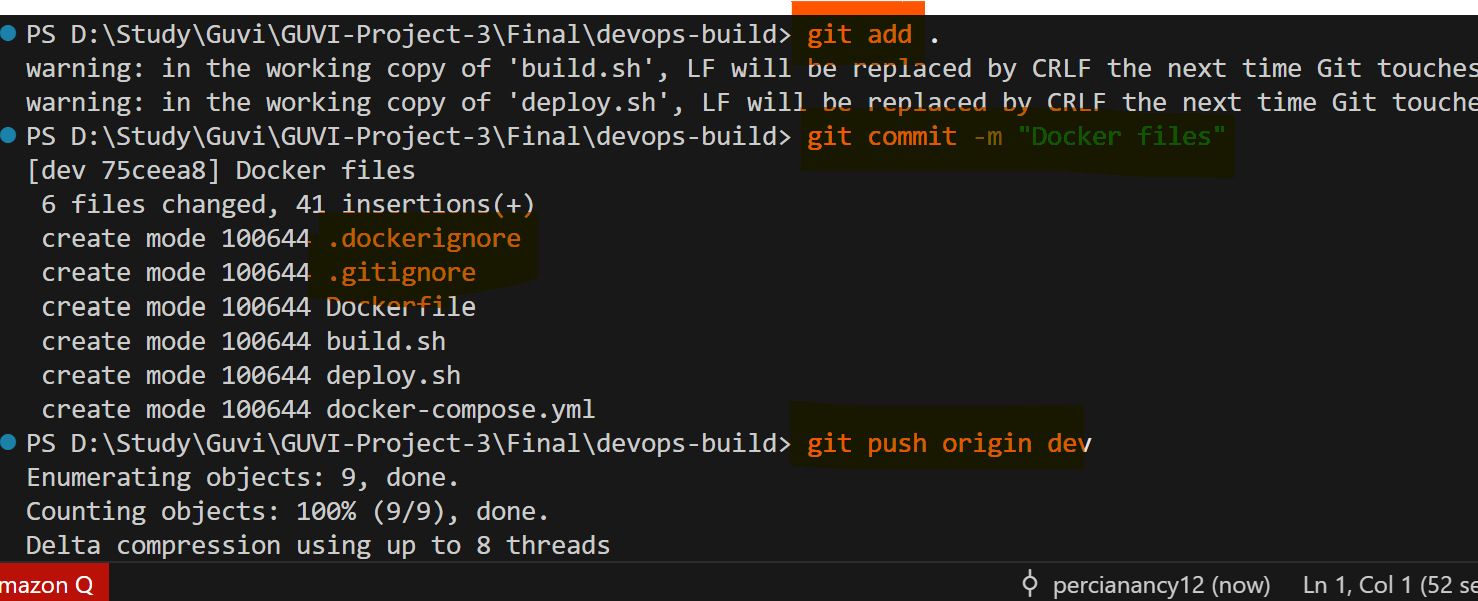


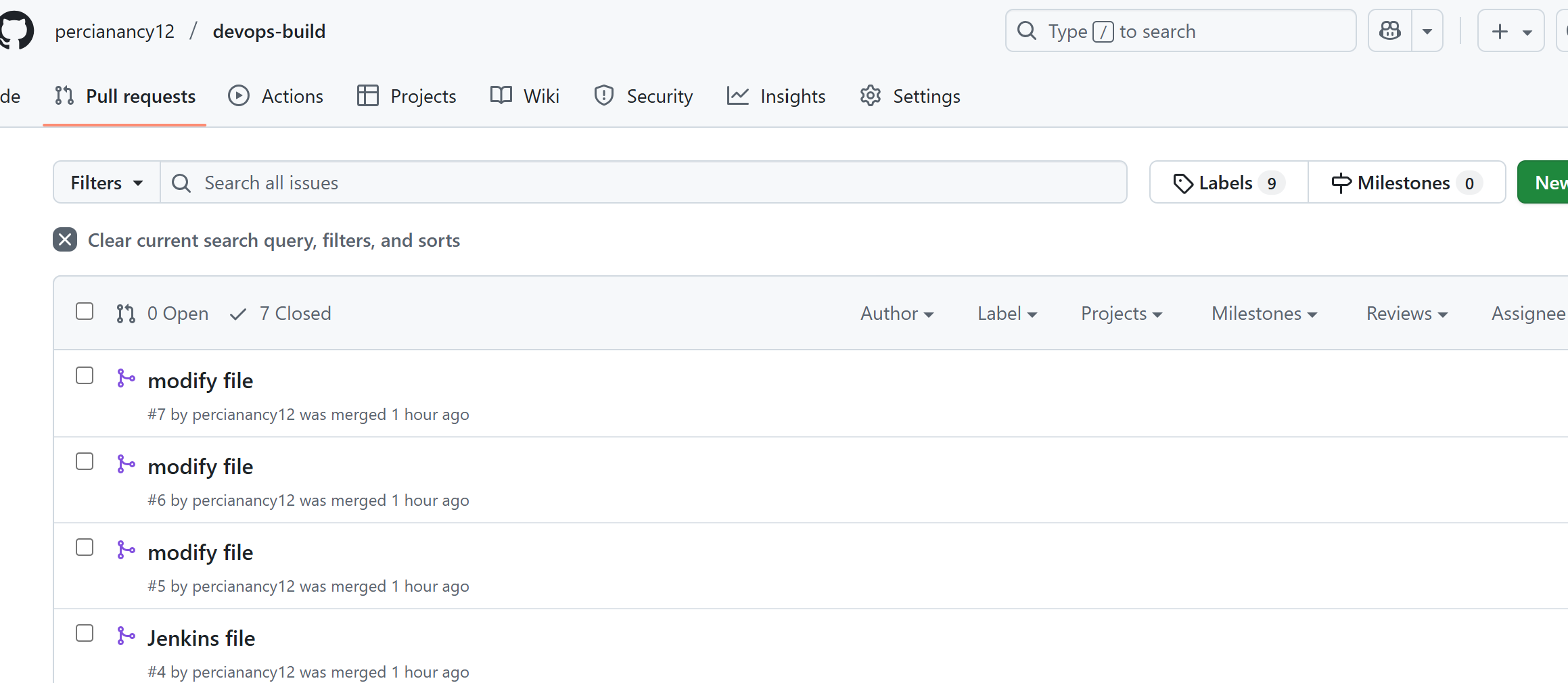
Deploy.sh – for deploying image to server



**GIT**

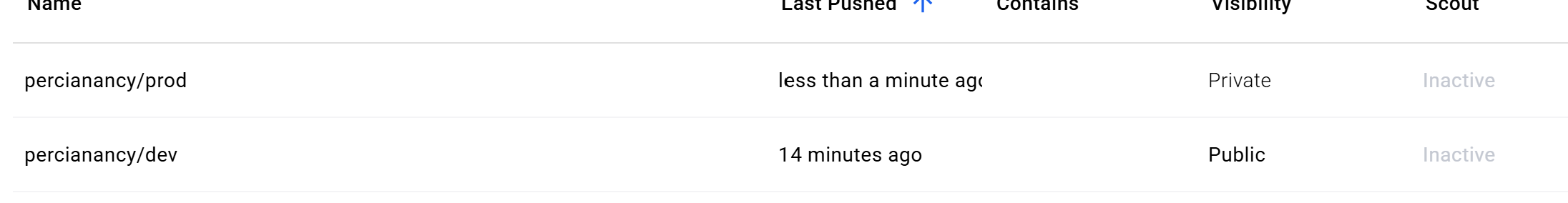
Code pushed to dev branch



Code merged to main branch  


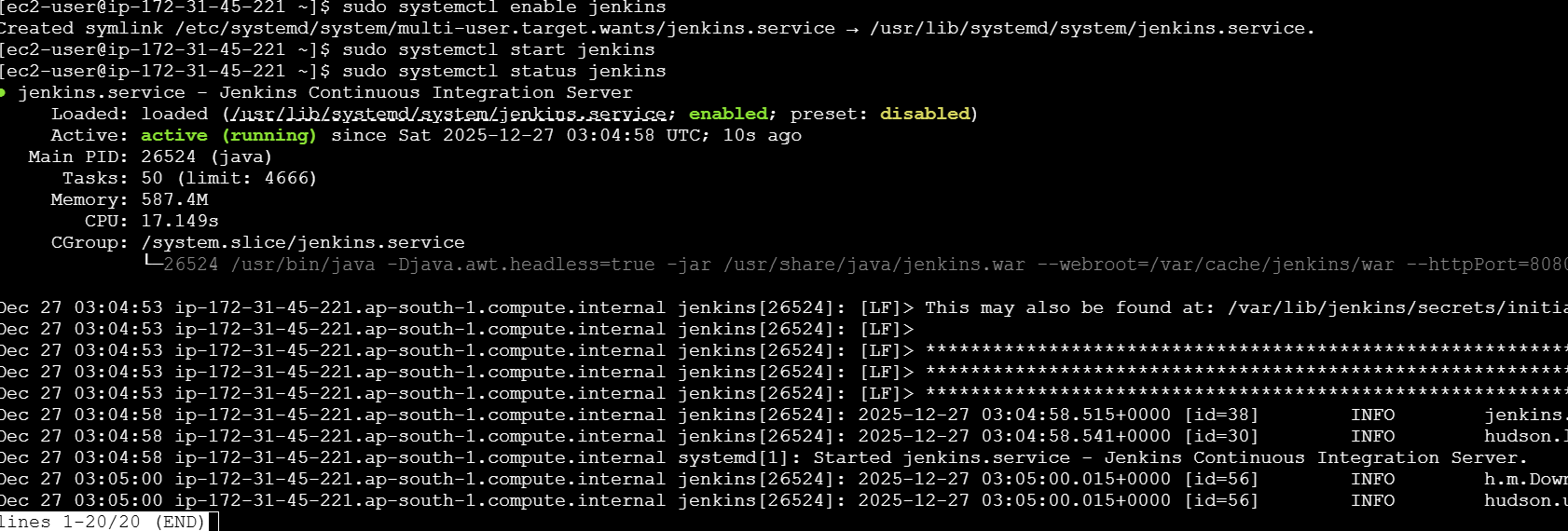
**Docker Hub:**

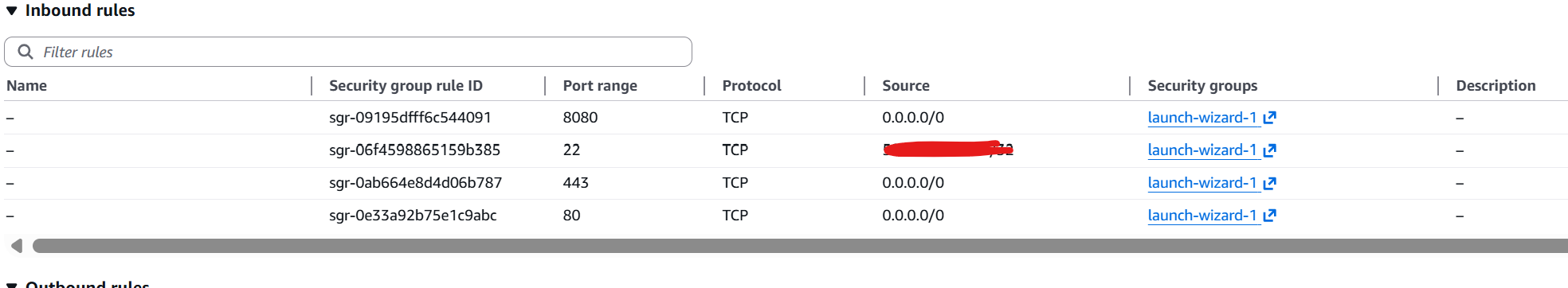
Private and public repo



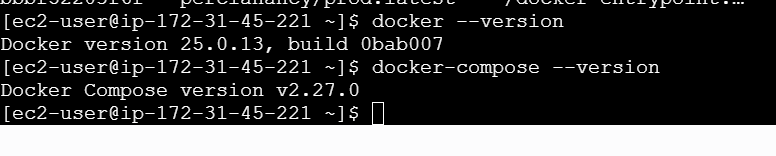
**AWS**

Install Jenkins in EC2

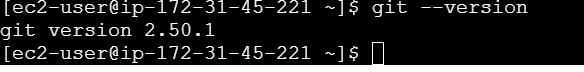


Allow port 8080 and SSH to my IP  
  


Install Docker & docker compose



Install GIT

   
  
Give Jenkins Docker Access.Since Jenkins runs as the jenkins user, add it to the docker group:

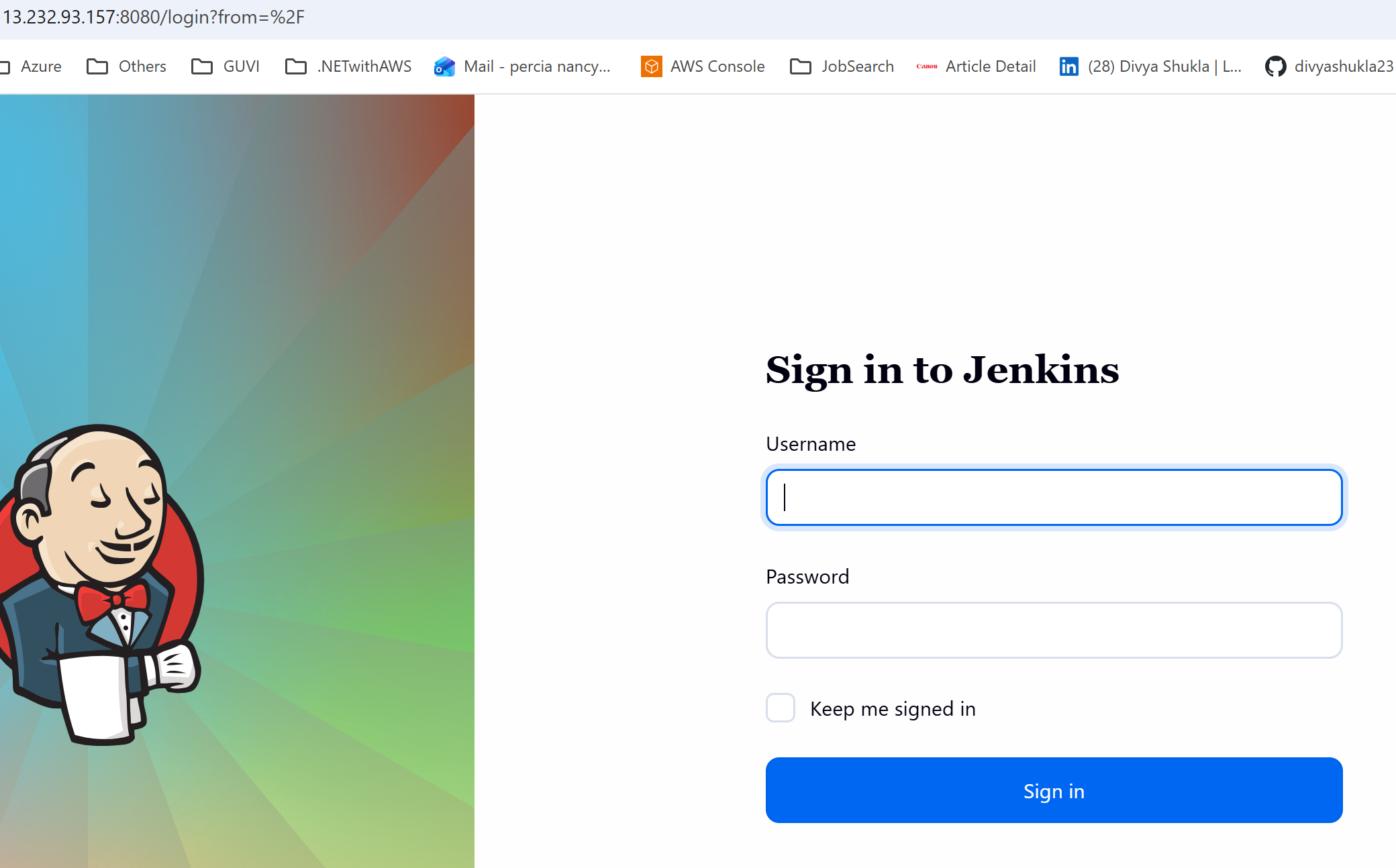
**sudo usermod -aG docker jenkins**

Verify:

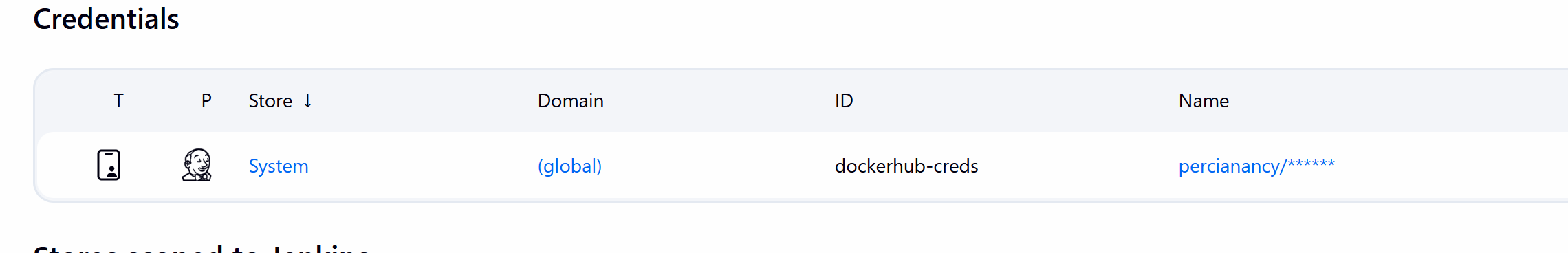


This ensures Jenkins pipelines can run docker build, docker run, etc. without permission errors.

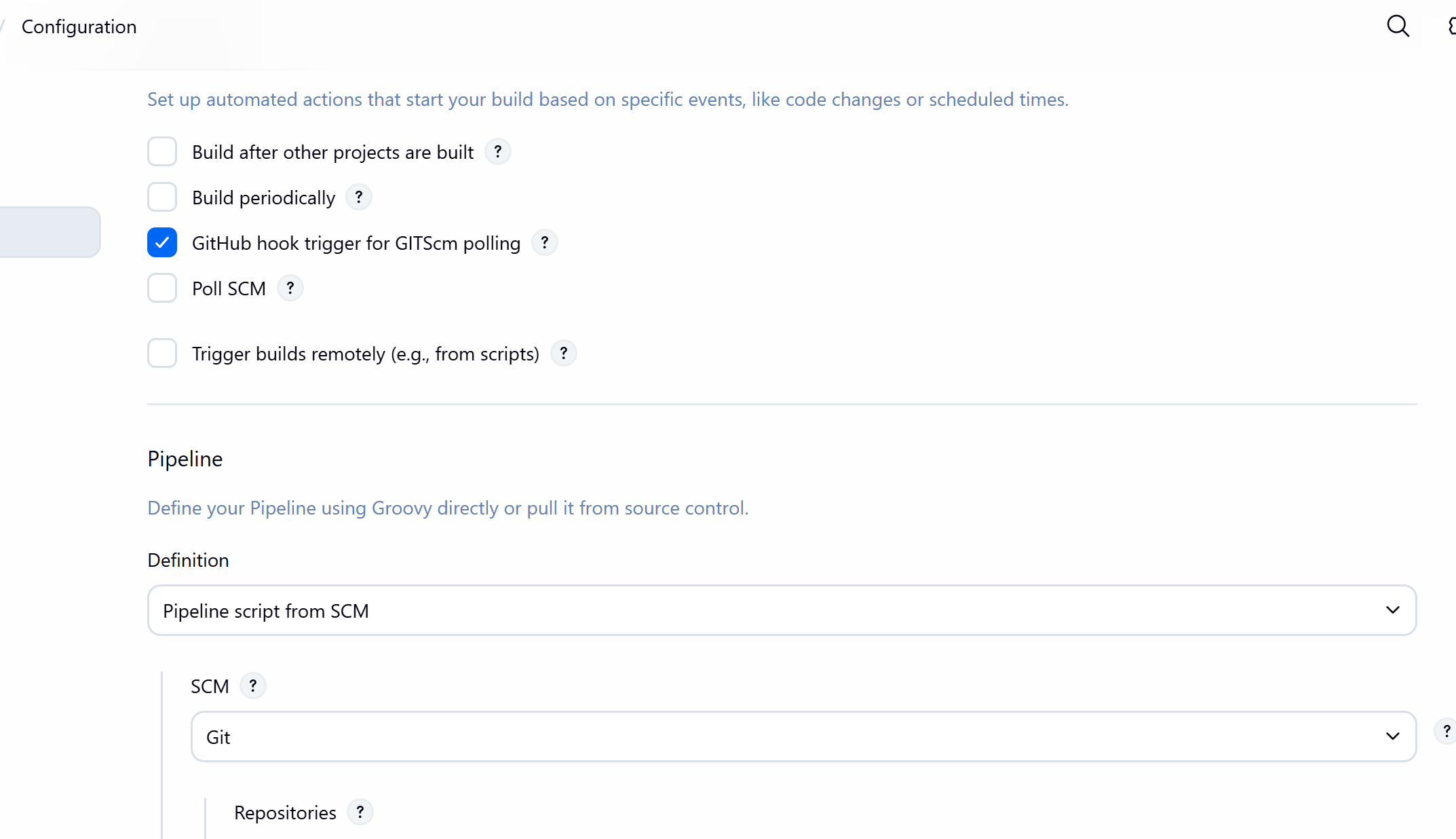
**Jenkins login page**

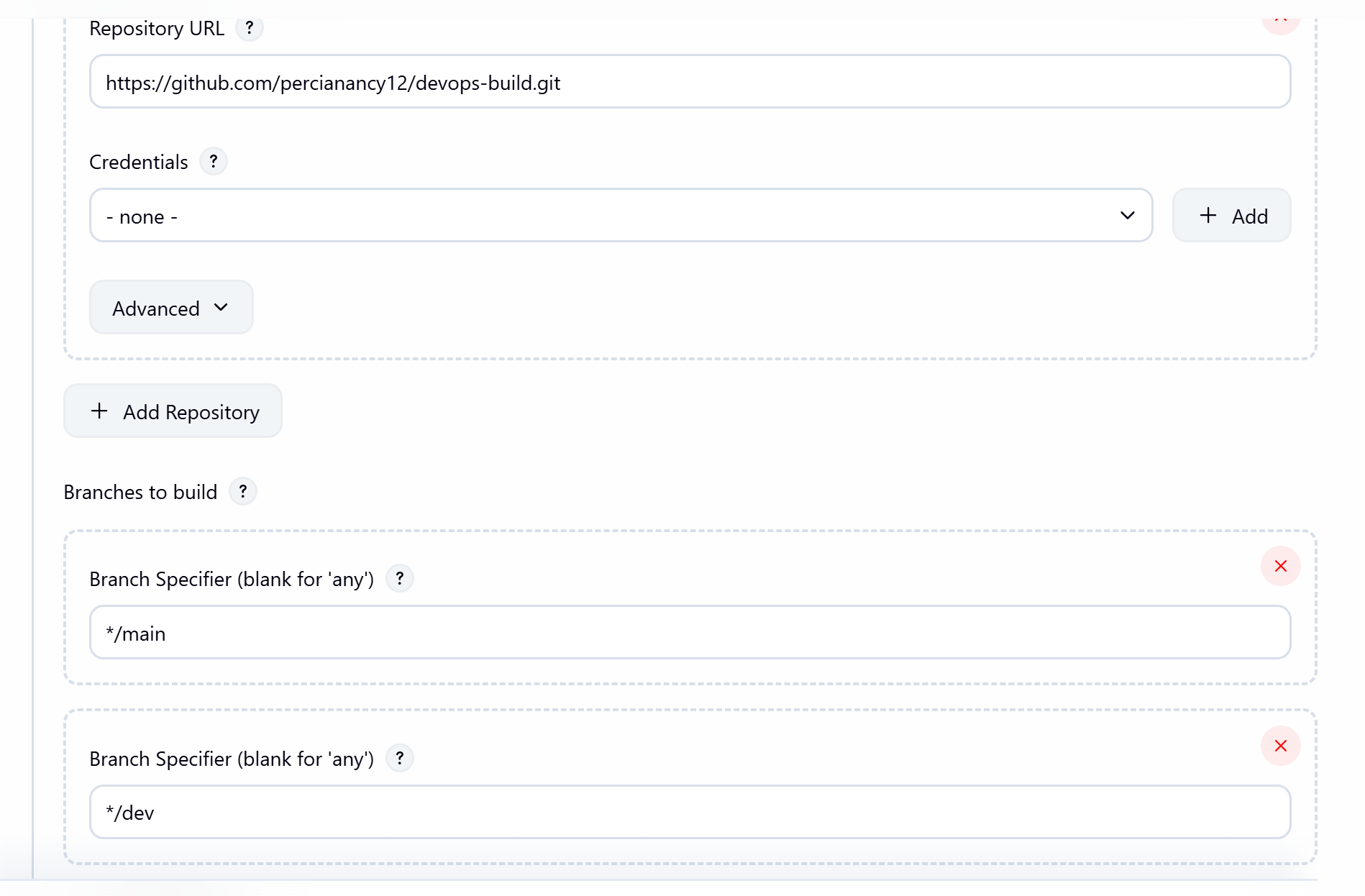


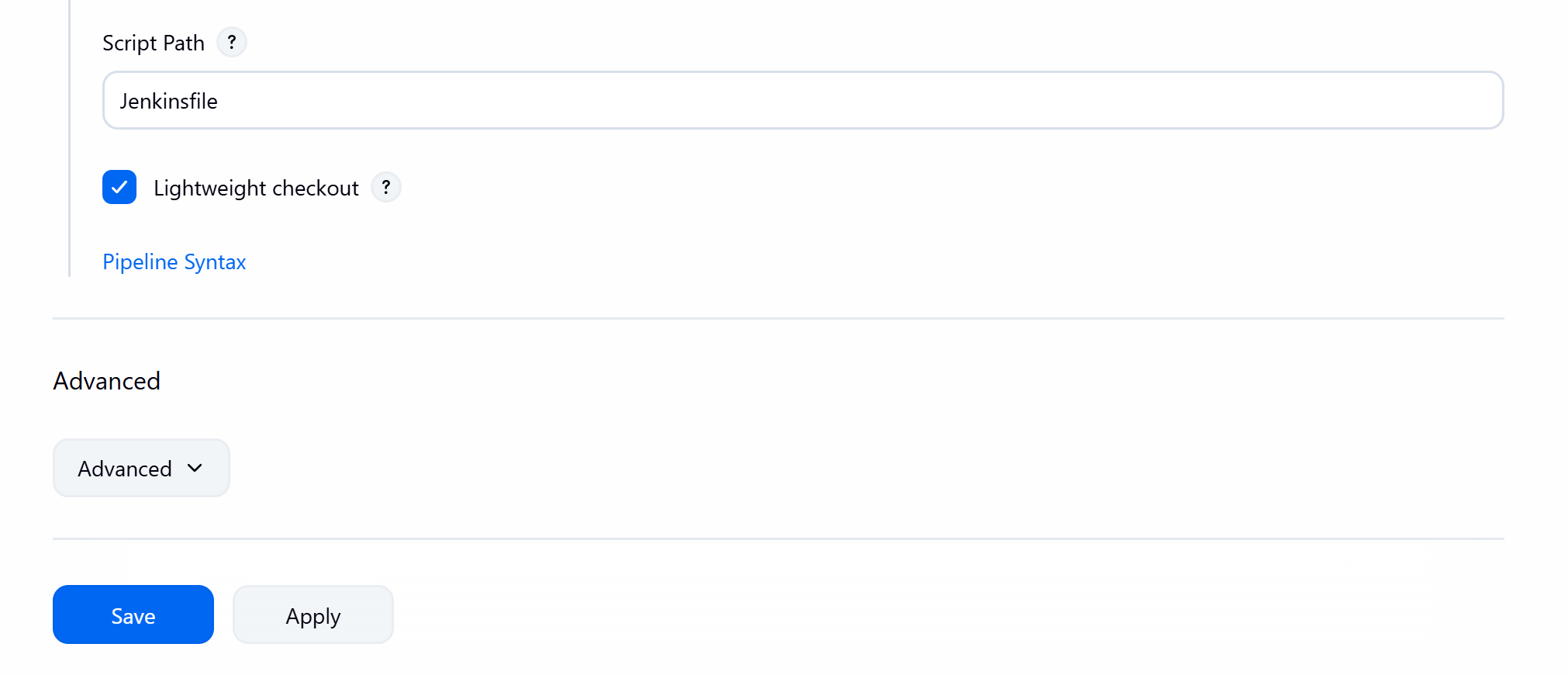
Install Docker, GIT, Pipeline steps plugin in Jenkins and Add Docker hub credentials



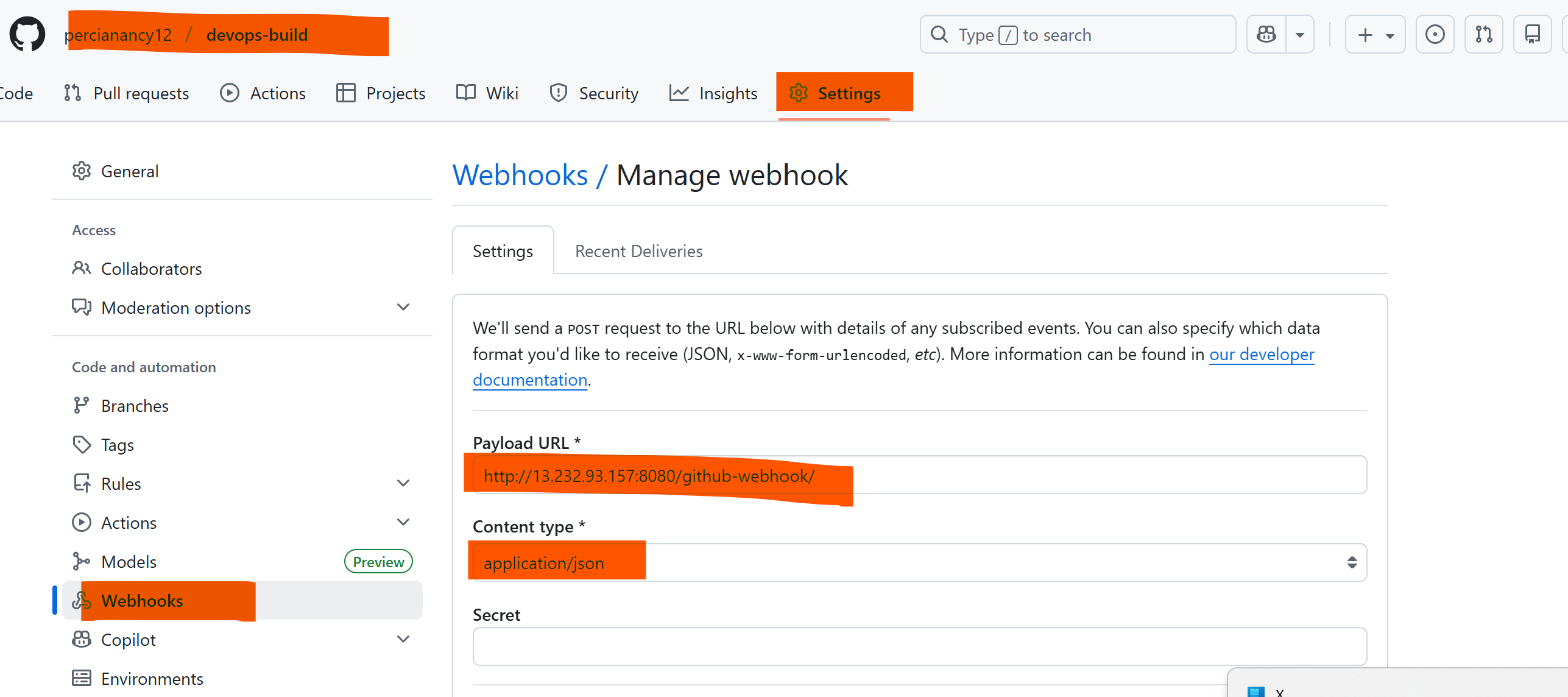
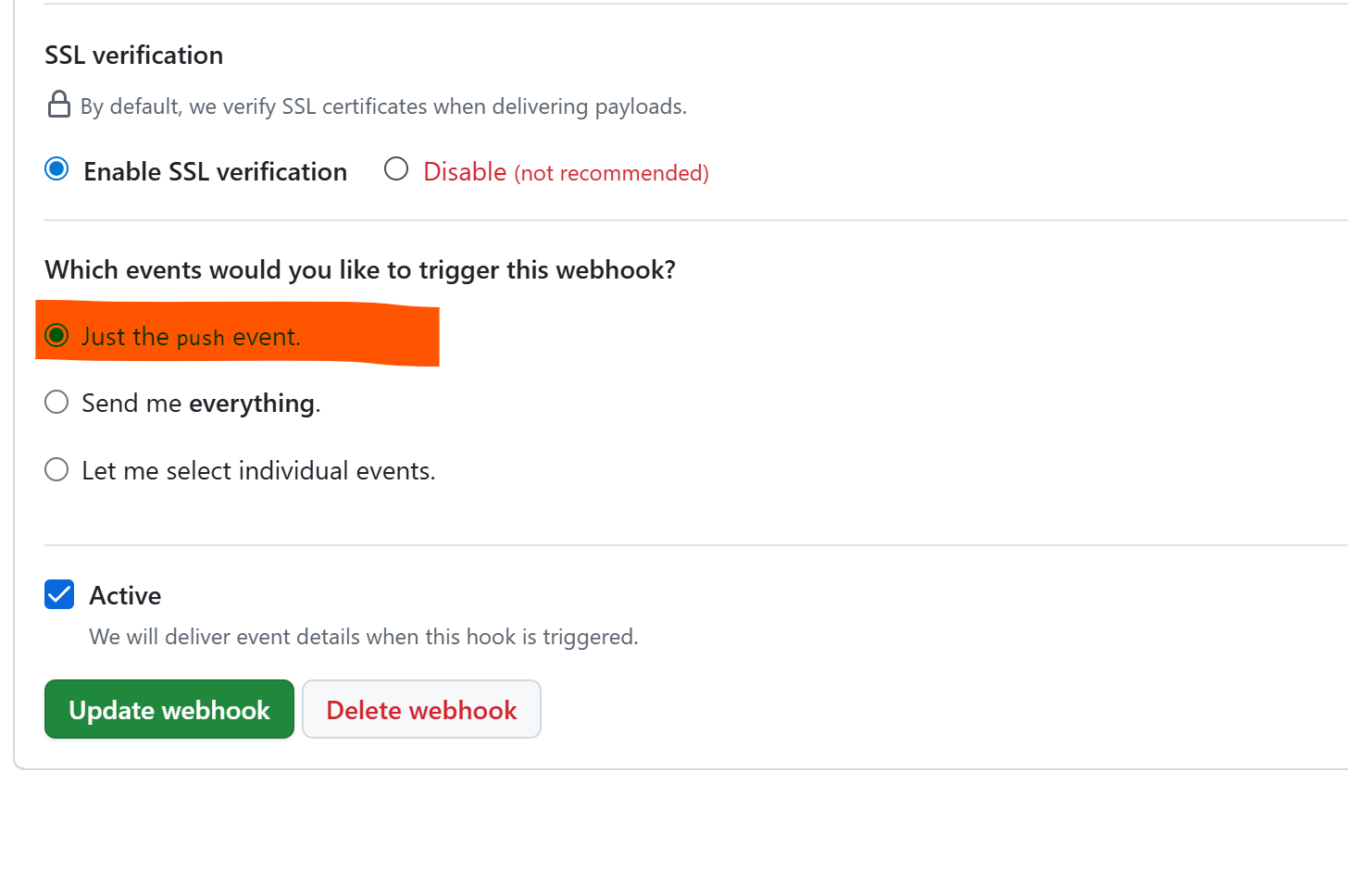
Create pipeline



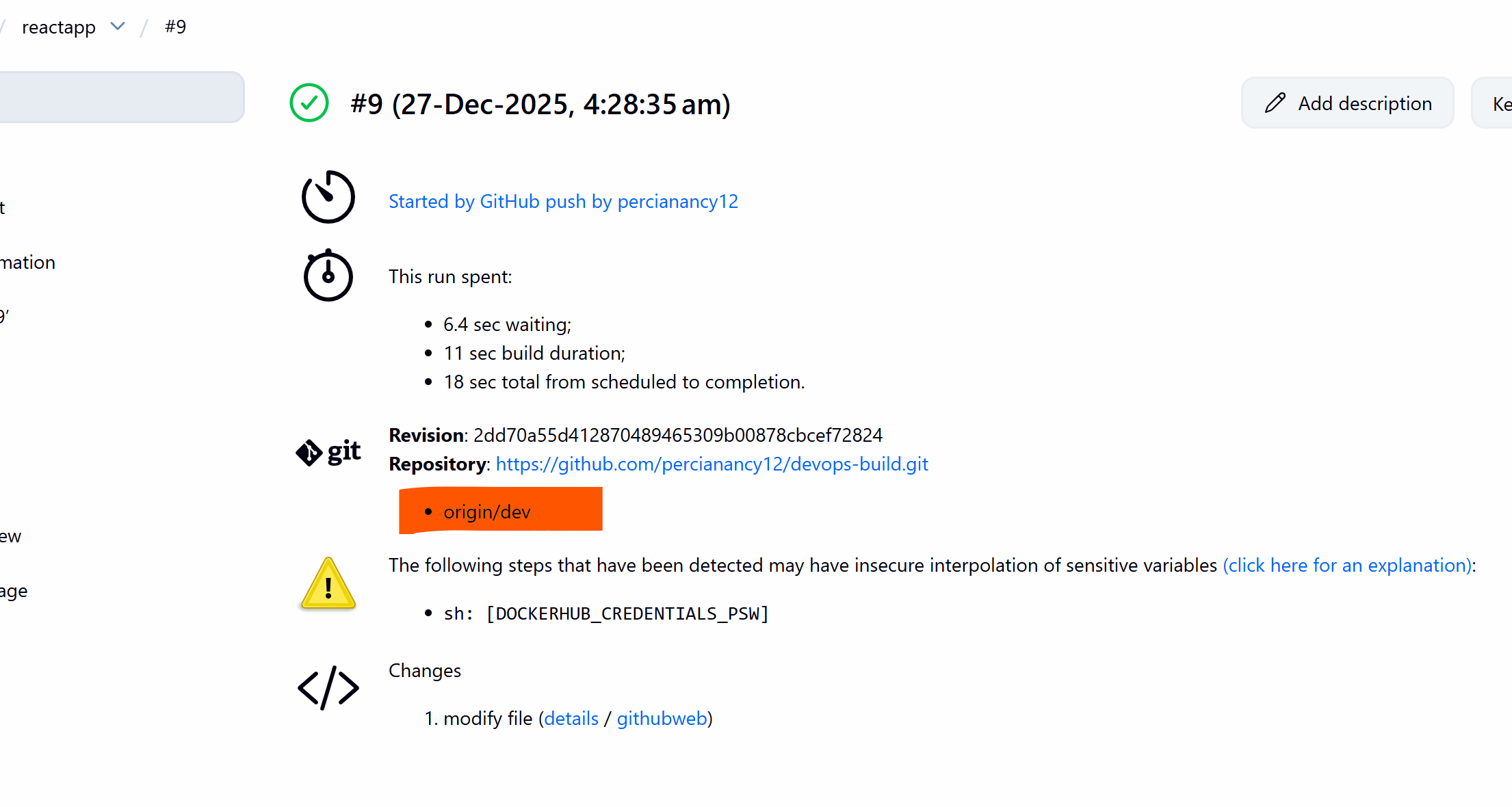


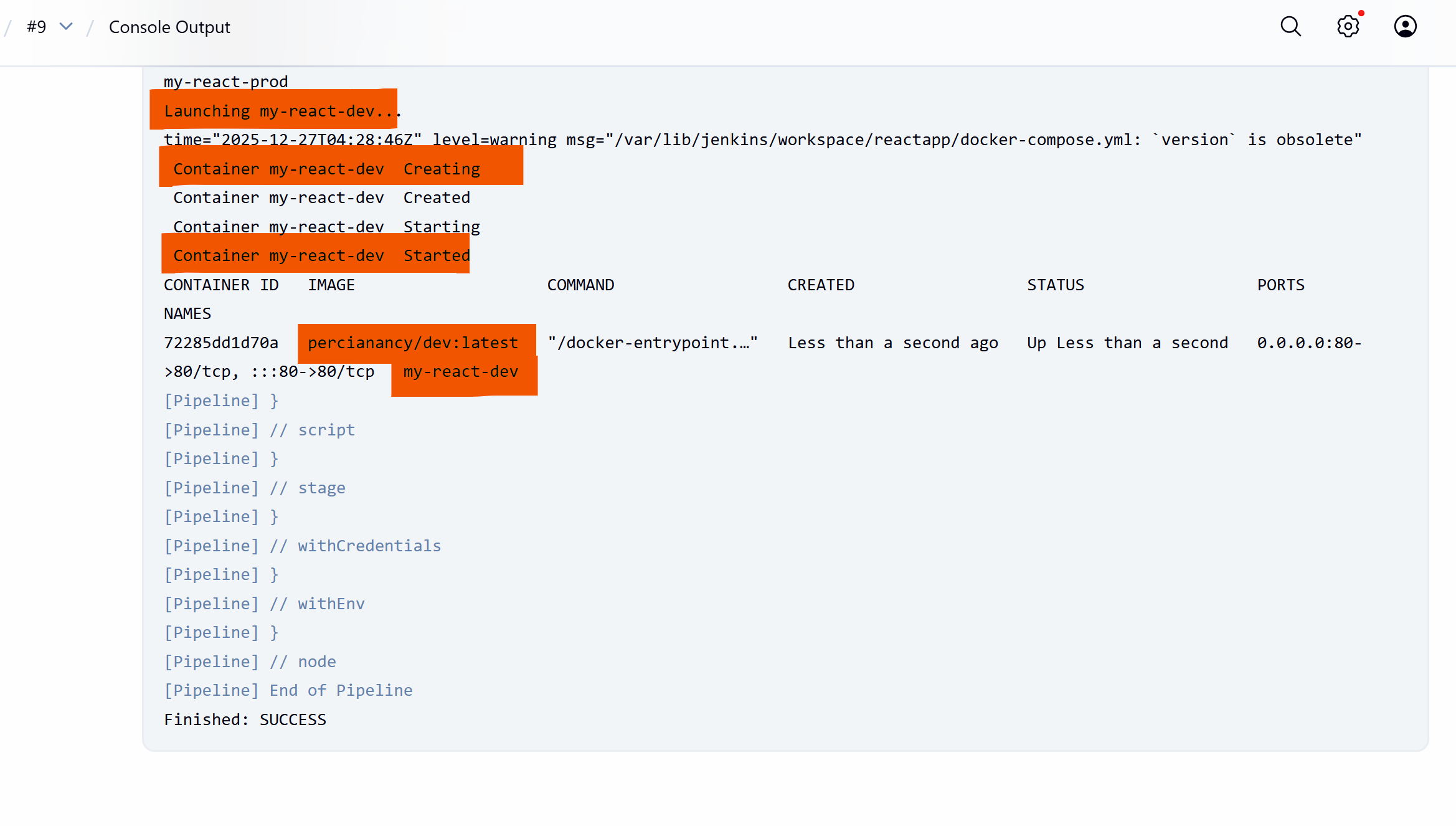
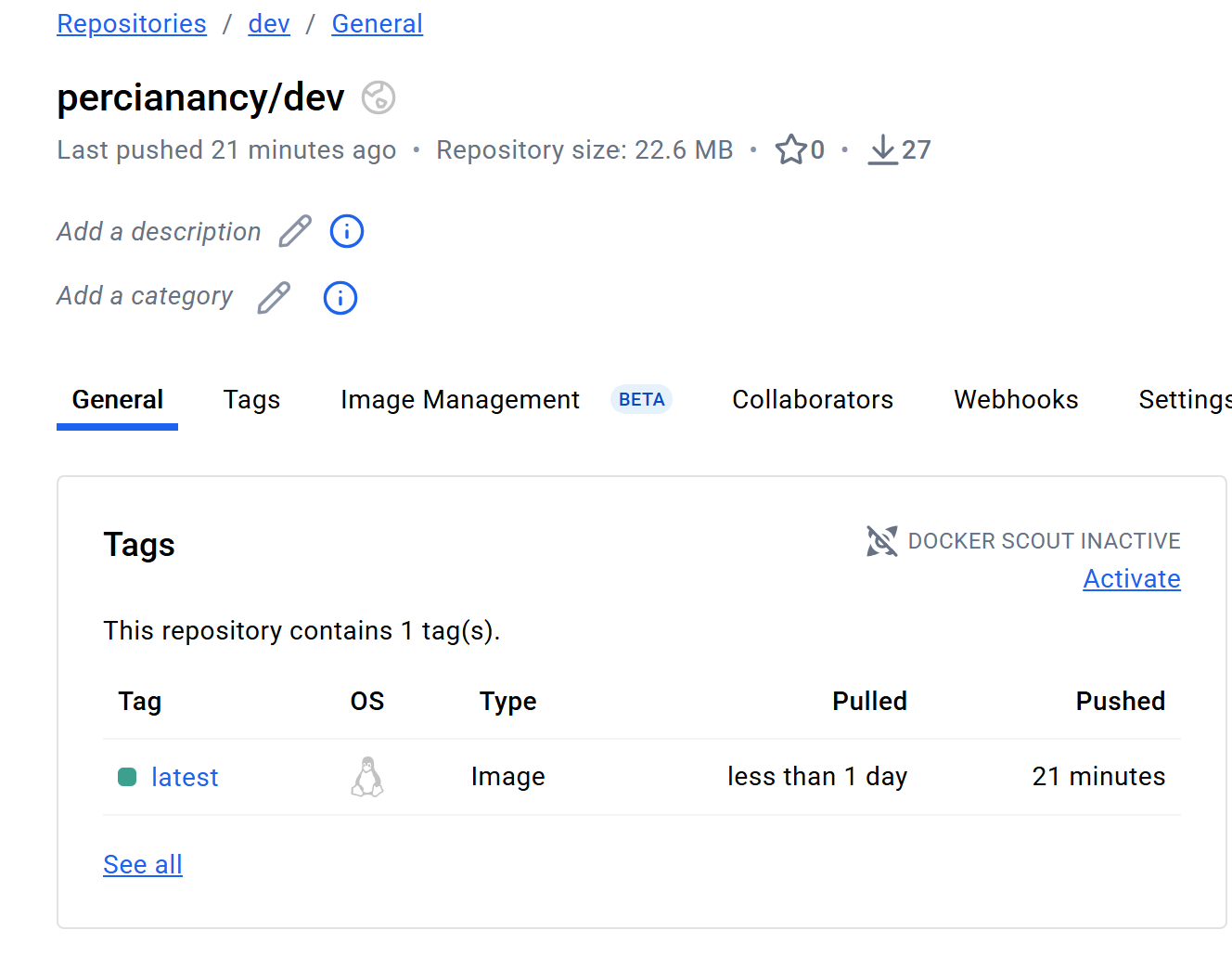


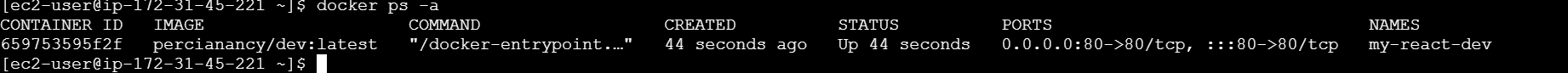
To enable auto build enable webhook in GIT .Payload url is EC2 public IP

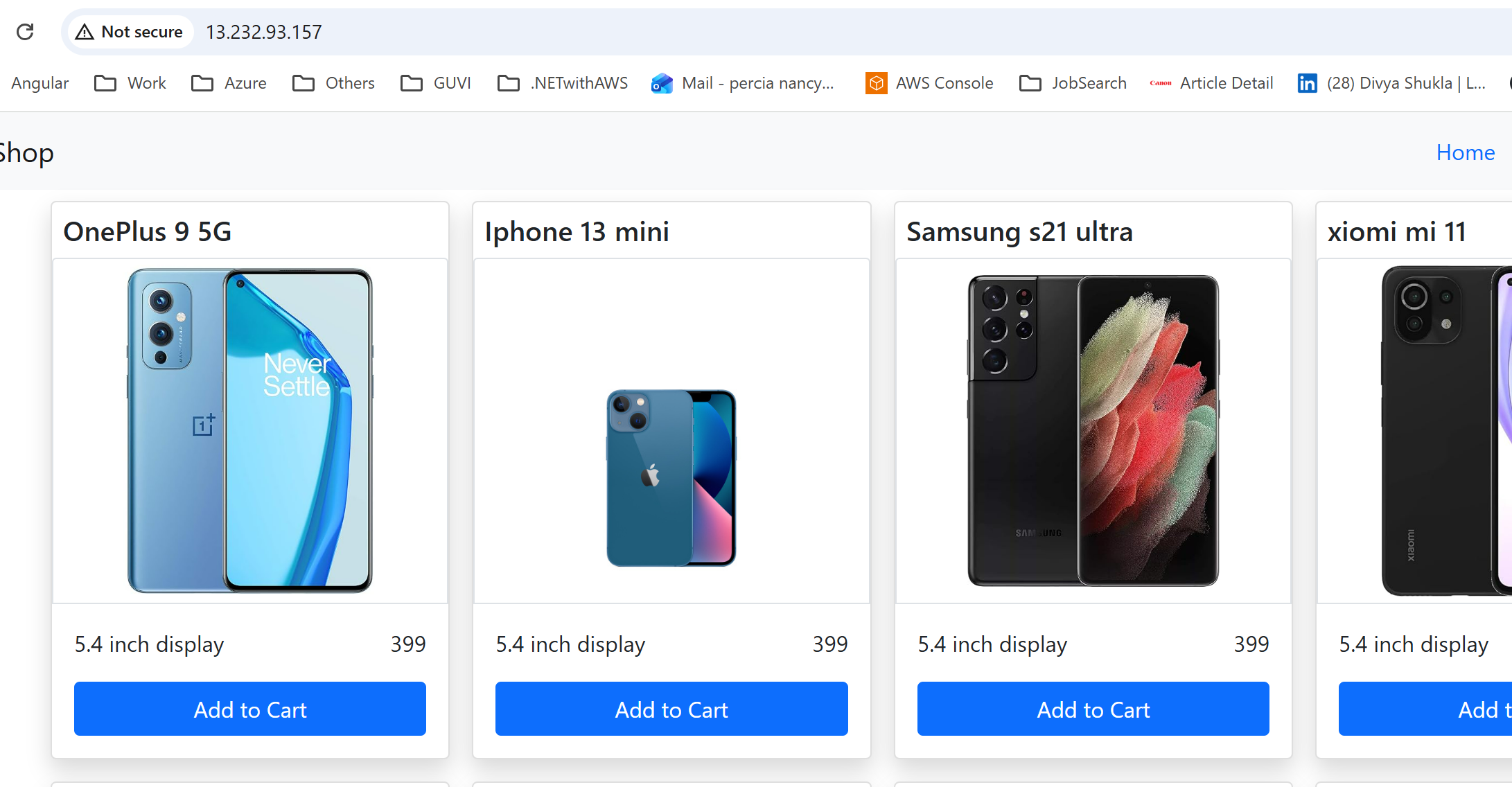
  
  


Once the code pushed to dev, deployment will start for dev

Push to **dev branch** → Jenkins detects origin/dev → tags/pushes percianancy/dev:latest → runs deploy.sh dev → container my-react-dev runs.  
  


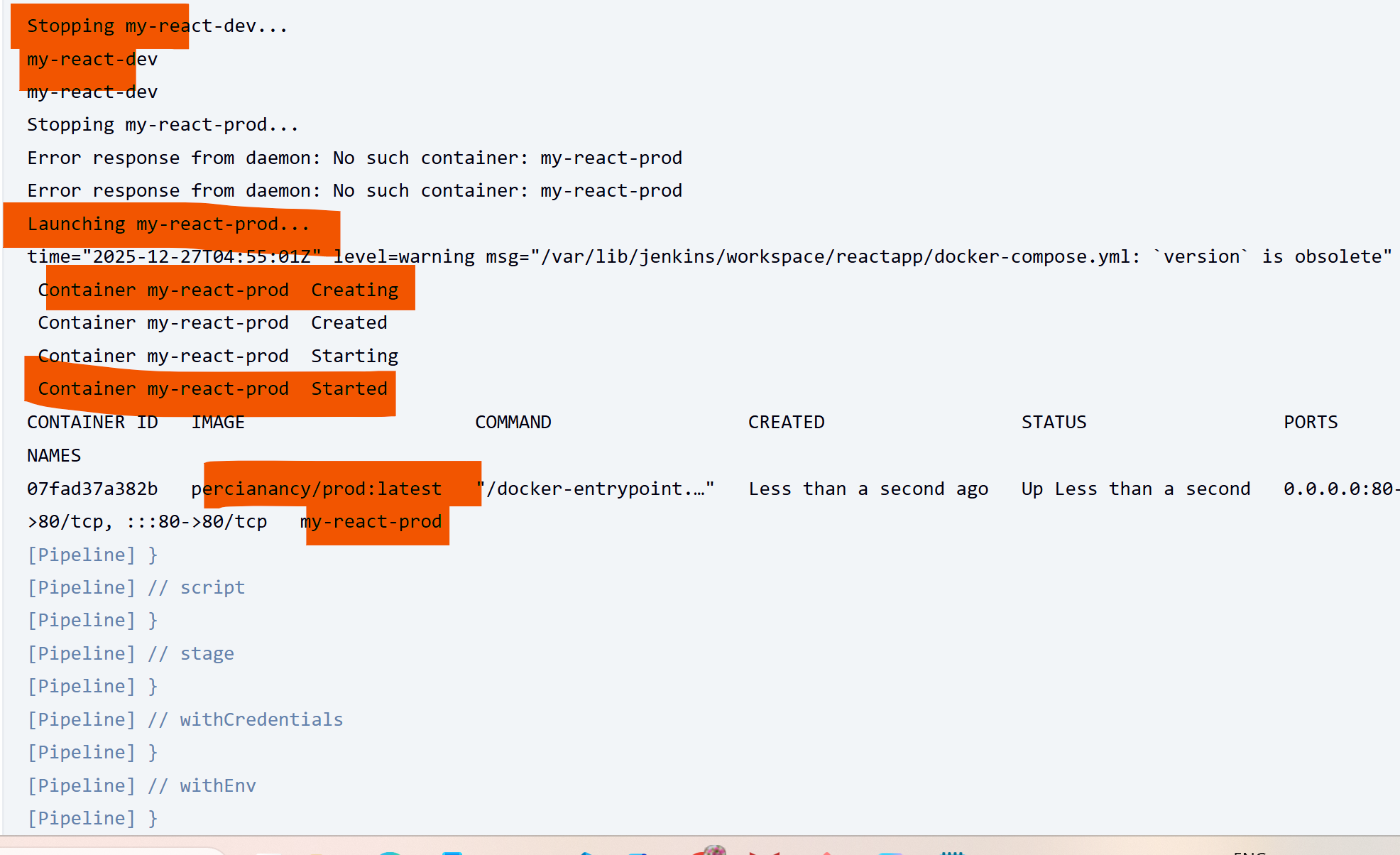
  
  


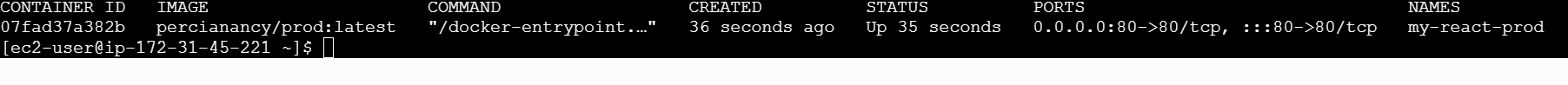
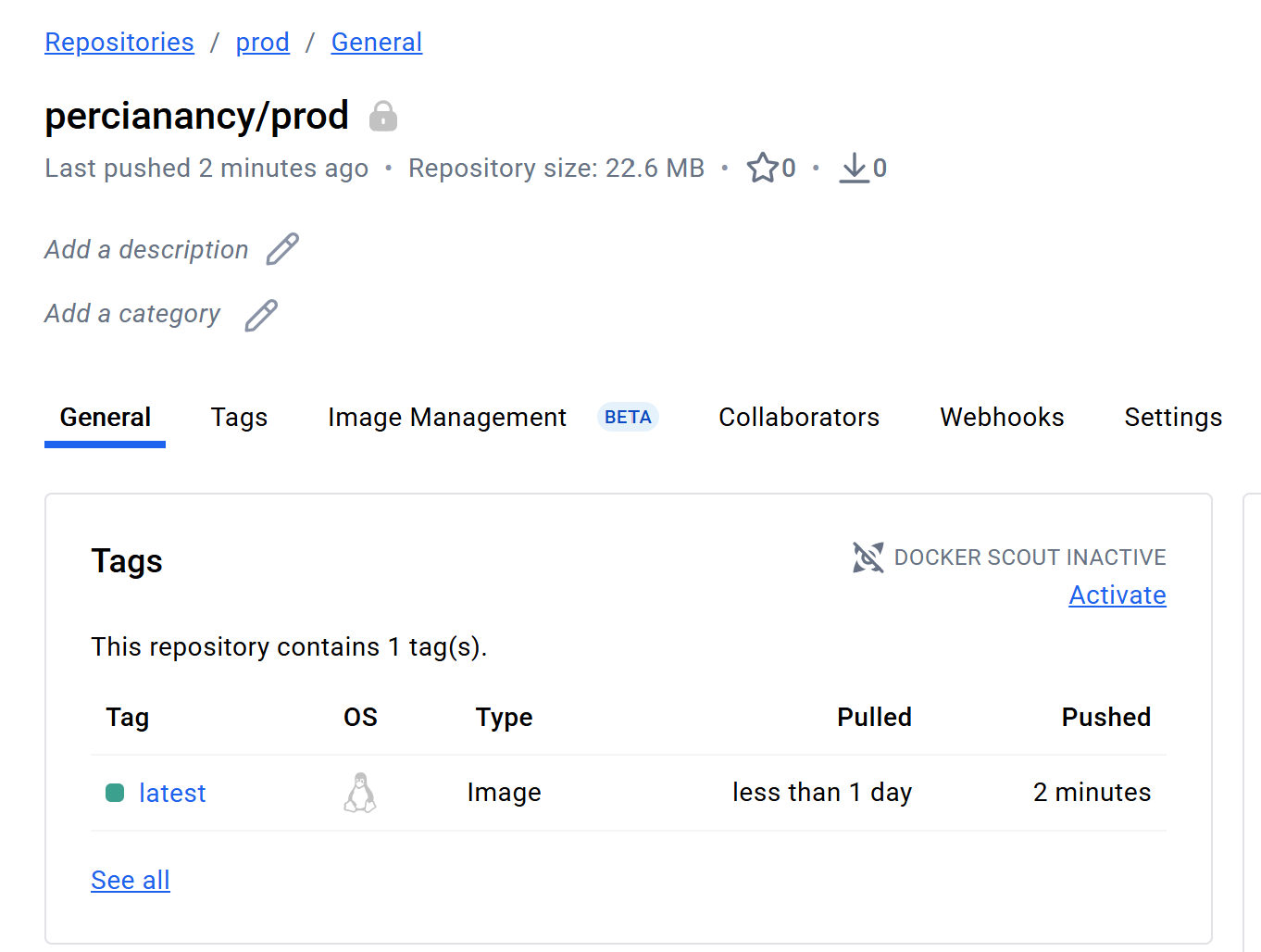


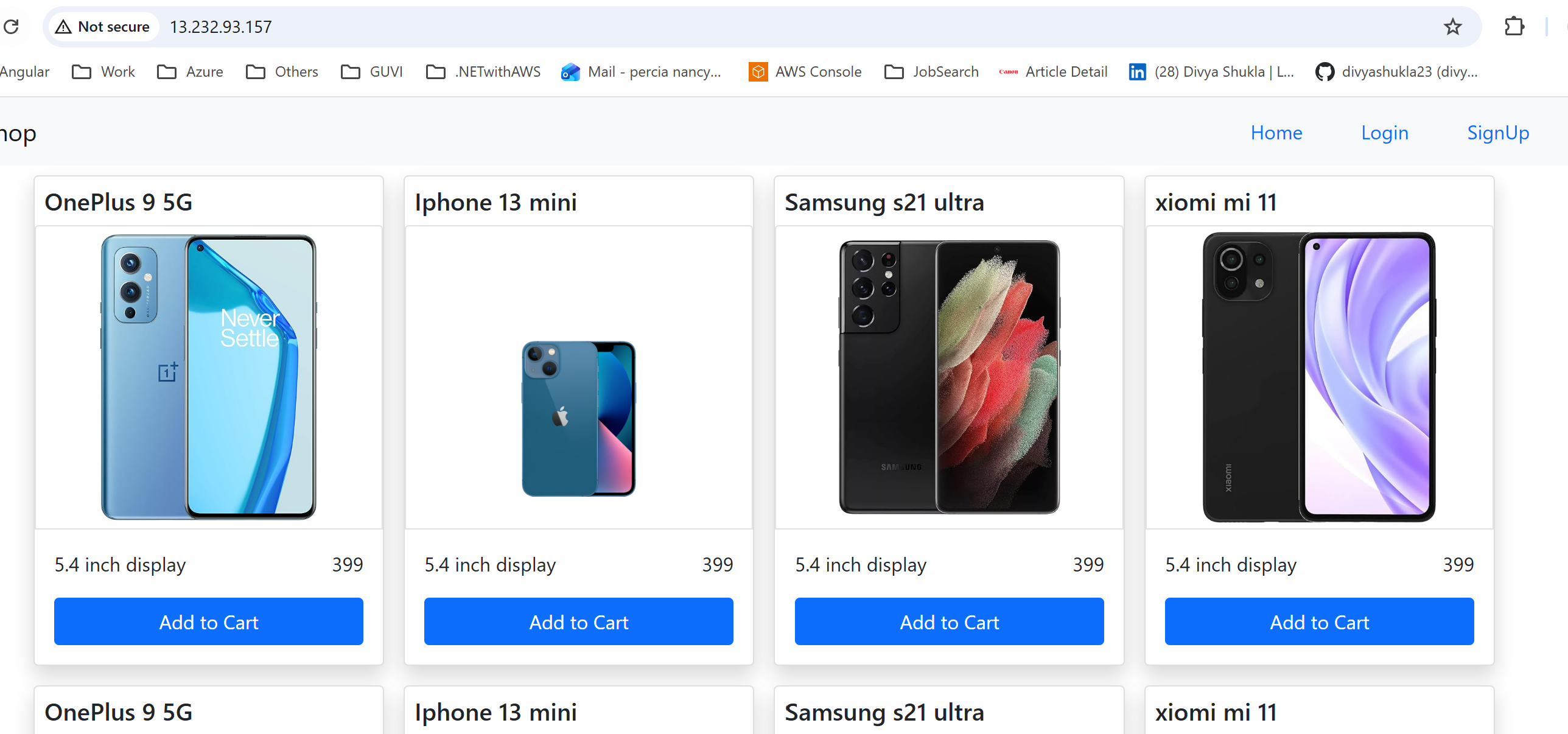


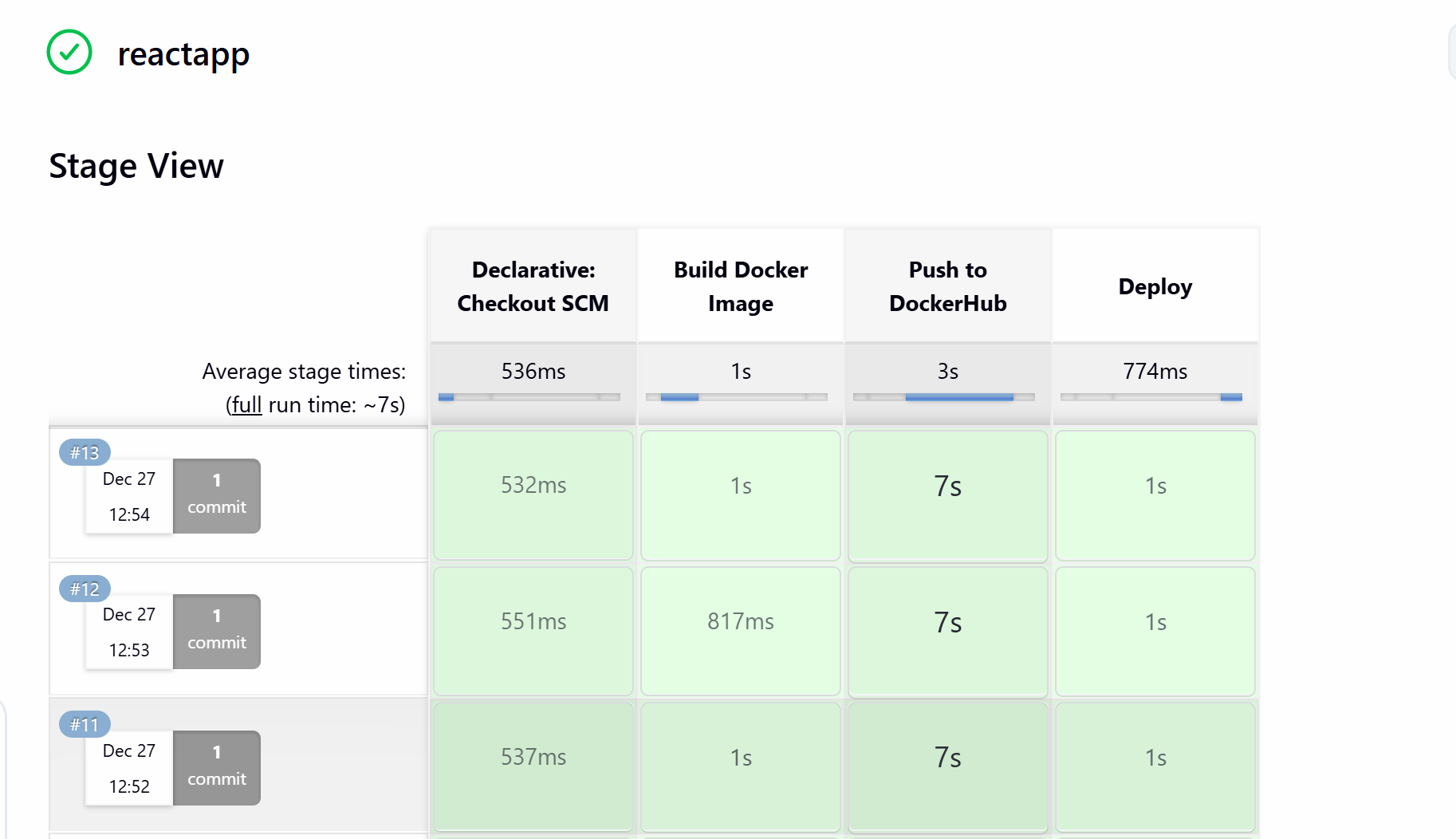
Push/merge to **main branch** → Jenkins detects origin/main → tags/pushes percianancy/prod:latest → runs deploy.sh prod → container my-react-prod replaces dev on port 80.









**Monitoring**

**full installation process for Monit**

*sudo dnf update -y*

**Install Build Tools and Dependencies**

Monit requires compilers and libraries to build from source:

*sudo dnf groupinstall "Development Tools" -y*

*sudo dnf install gcc make wget tar openssl-devel pam-devel -y*

**Download Monit Source**

*wget https://mmonit.com/monit/dist/monit-5.33.0.tar.gz*

*tar xvf monit-5.33.0.tar.gz*

*cd monit-5.33.0*

**Configure Build**

Run configure to generate the Makefile. If PAM causes issues, disable it:

*./configure --without-pam*

**Compile and Install**

*make*

*sudo make install*

This installs Monit into /usr/local/bin/monit.

**Verify Installation**

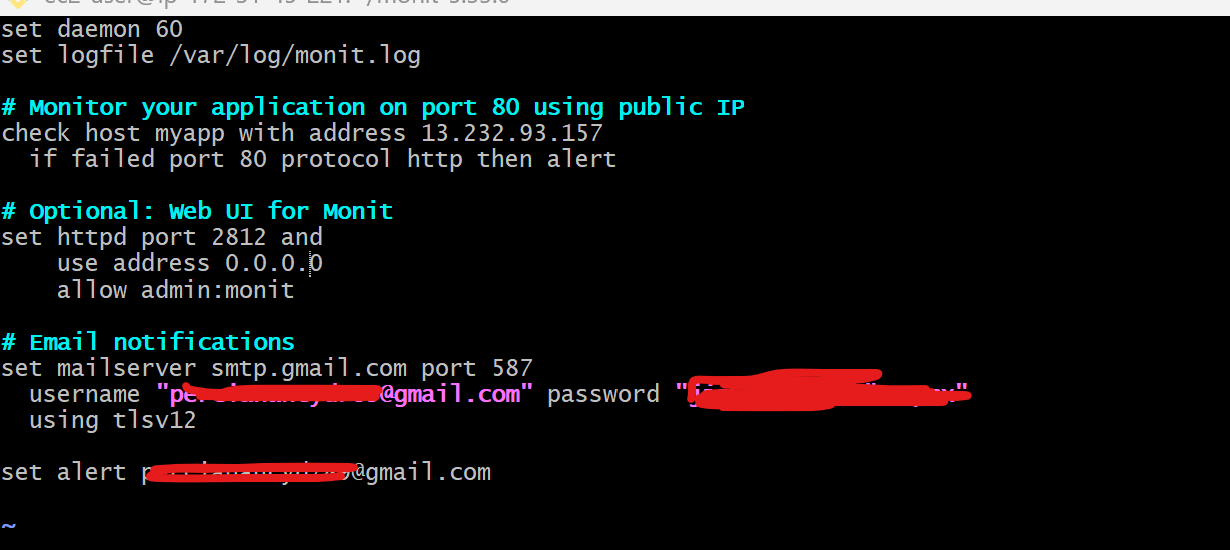
*/usr/local/bin/monit -V*



**Create Config File**

Monit looks for /etc/monitrc. Create and edit it:

*sudo vi /etc/monitrc*



**Secure Config File**

Monit requires strict permissions:

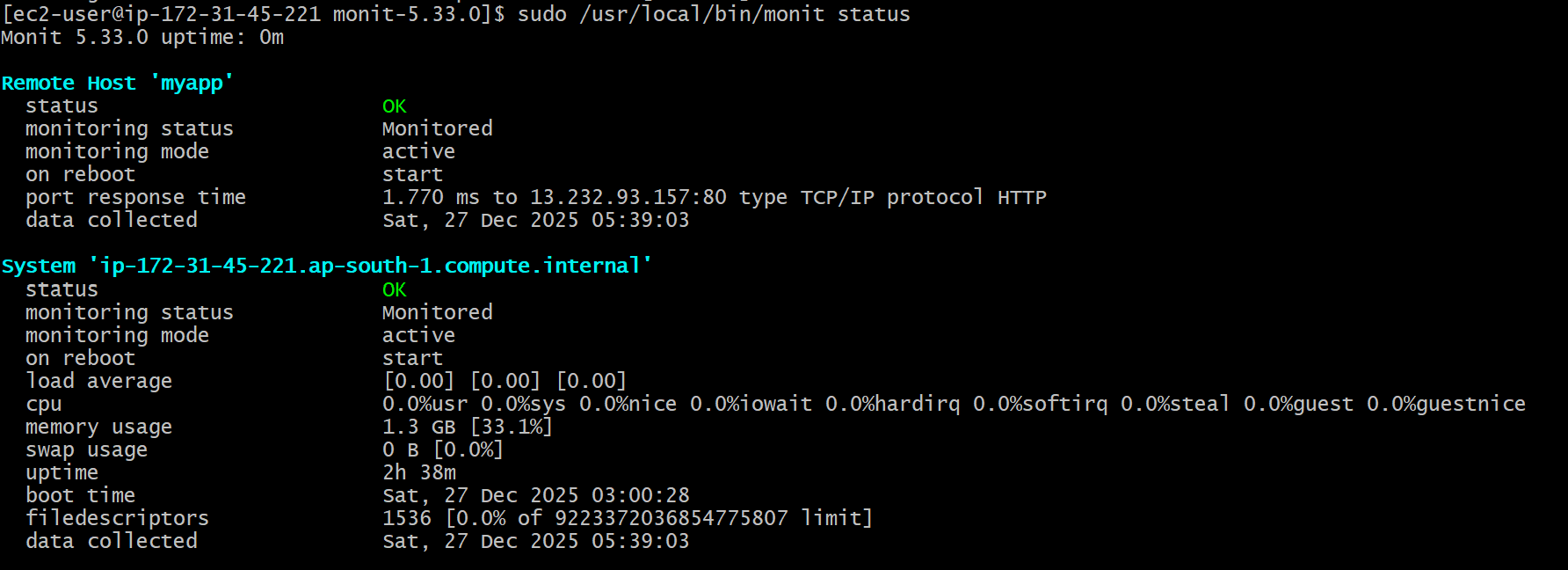
*sudo chmod 600 /etc/monitrc*

**Start Monit**

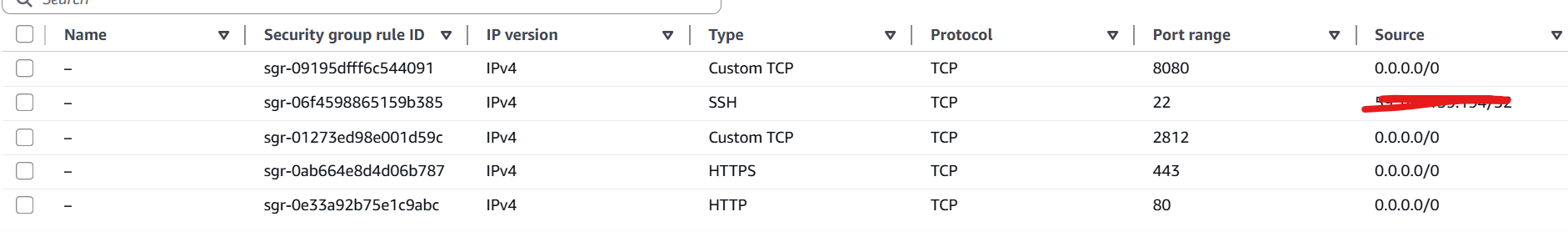
*sudo /usr/local/bin/monit -d 60*

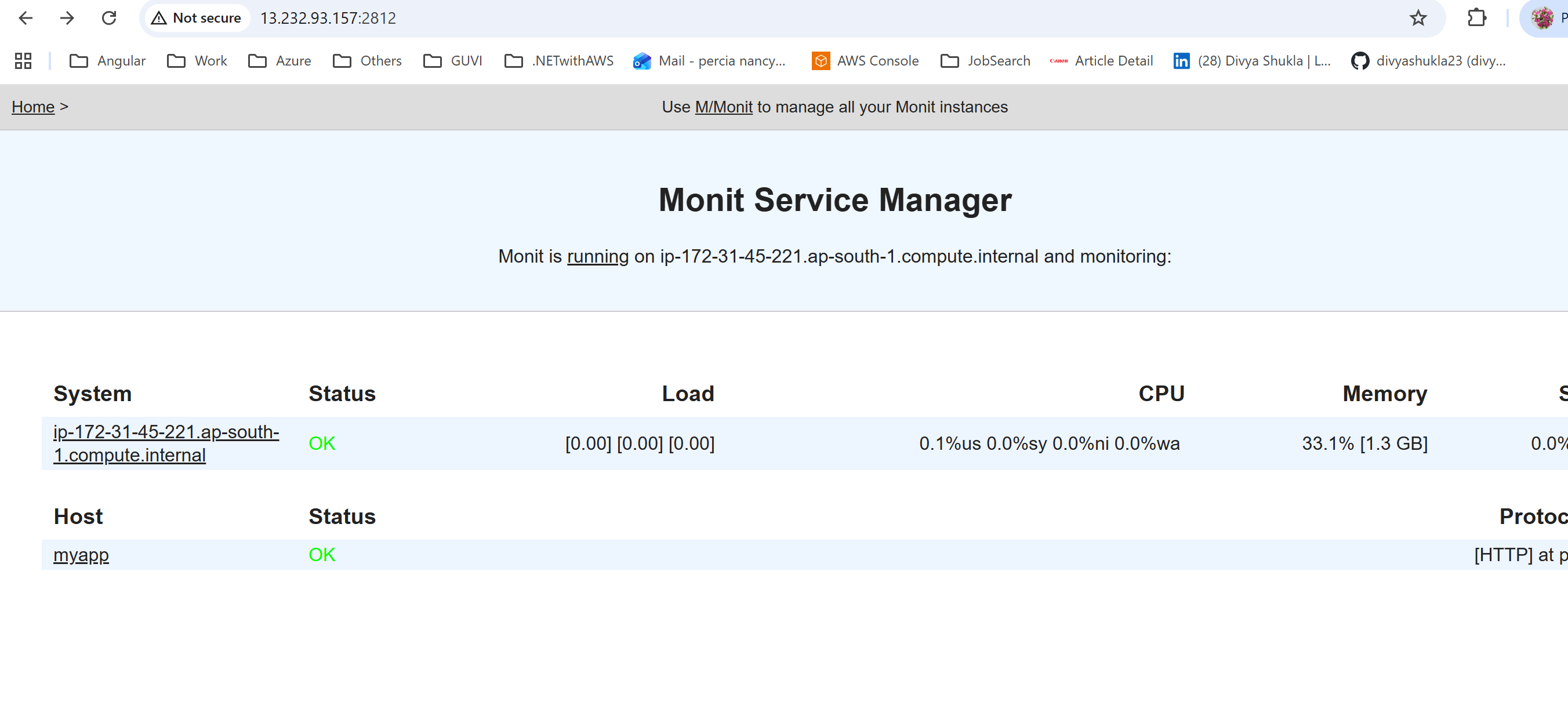
Check status:

*/usr/local/bin/monit status*

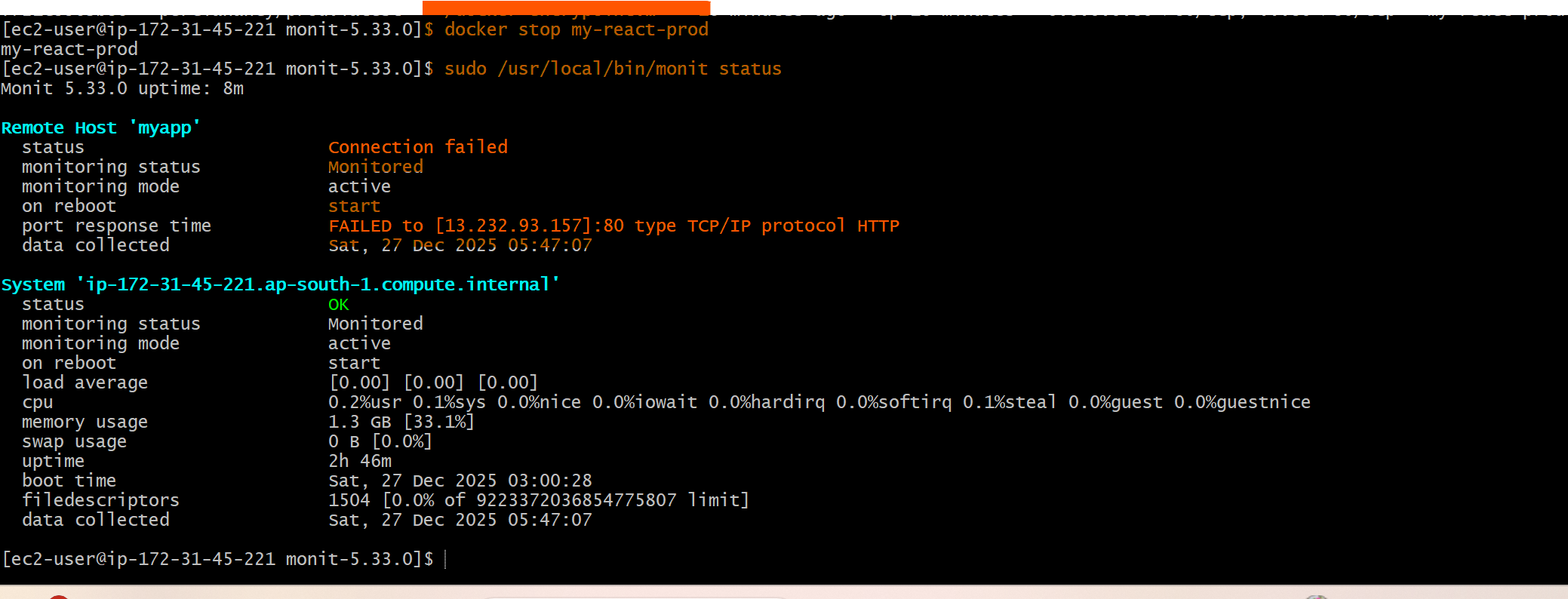


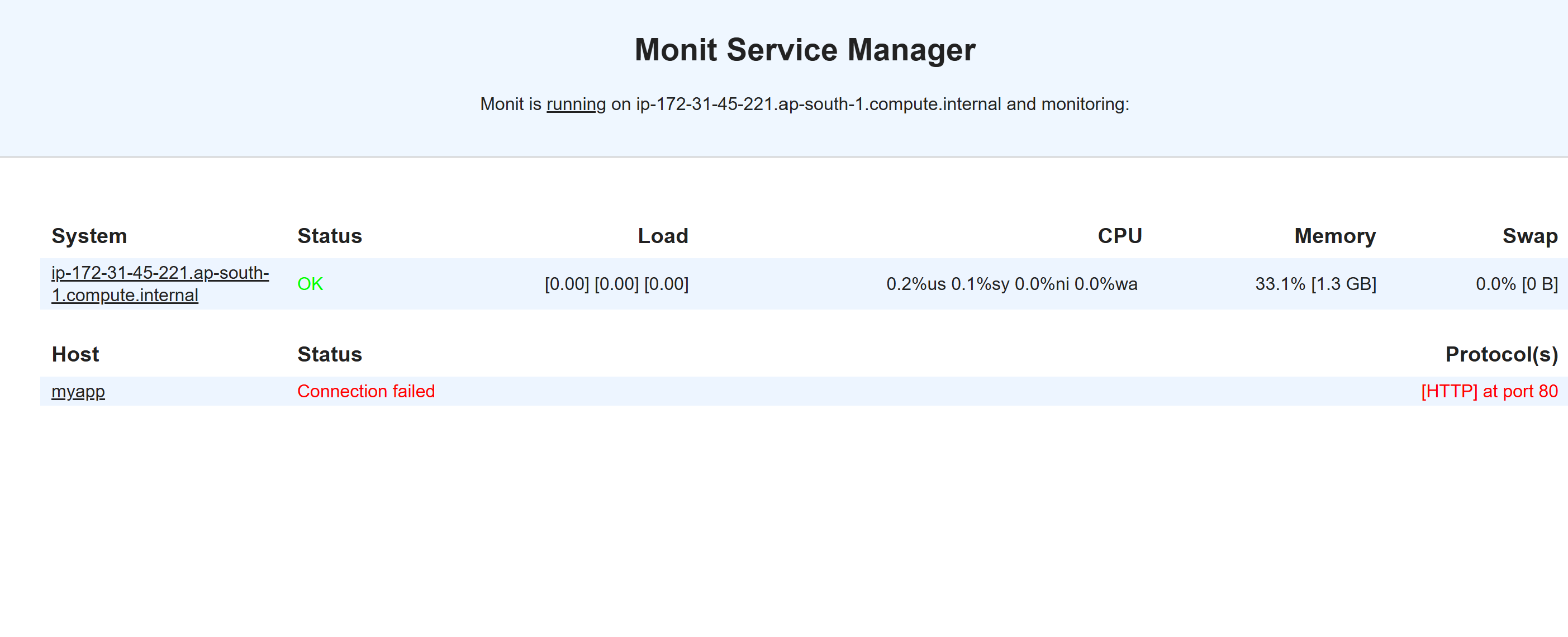
Allow port 2812 in SG

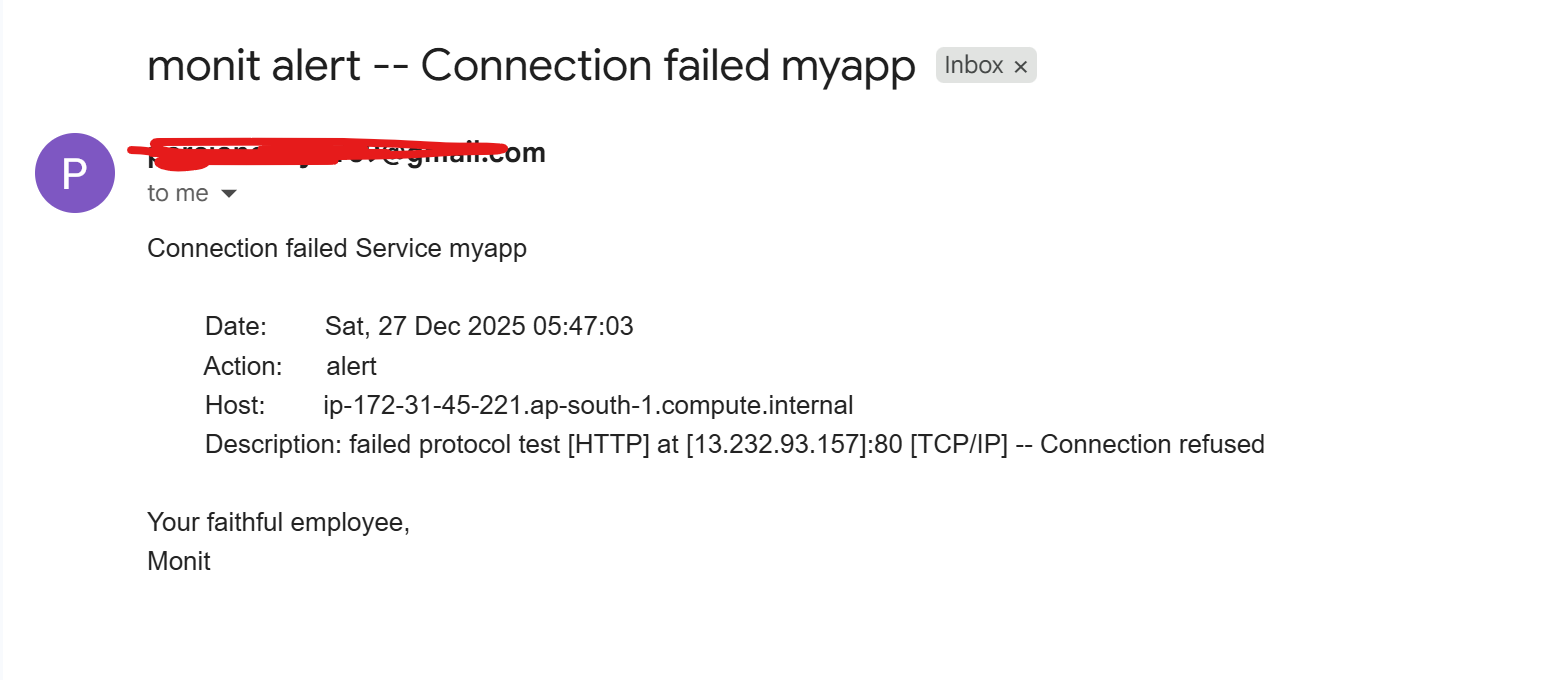




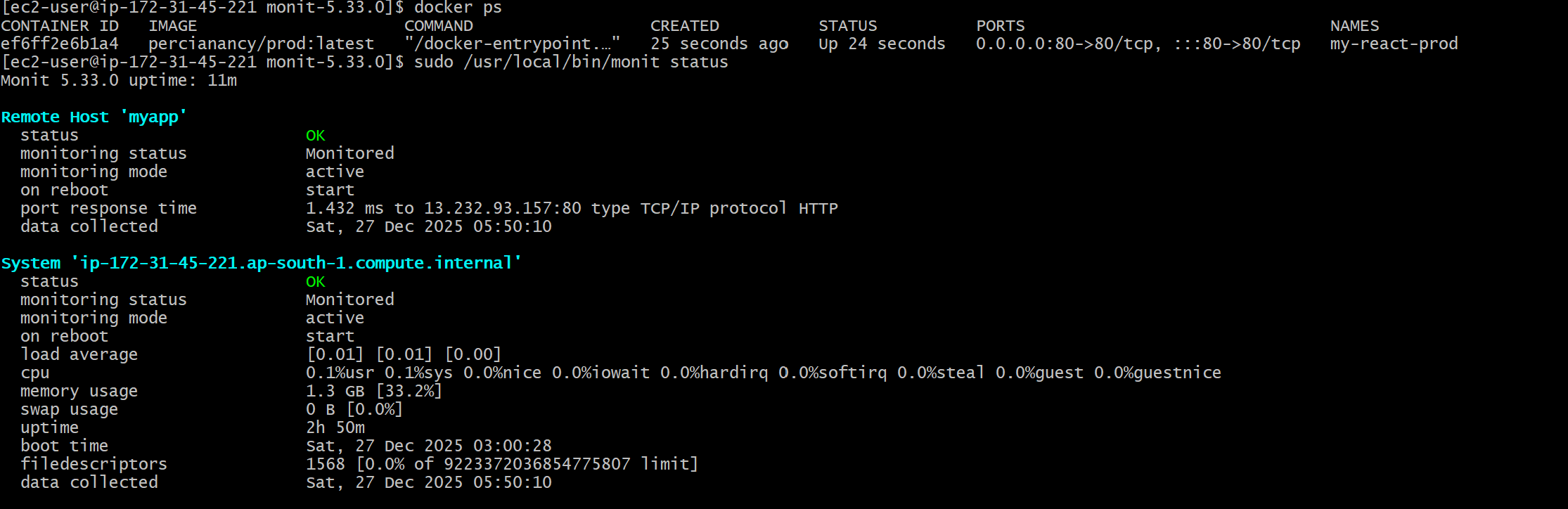
Once I stopped the container, connection failed and mailwill be triggered

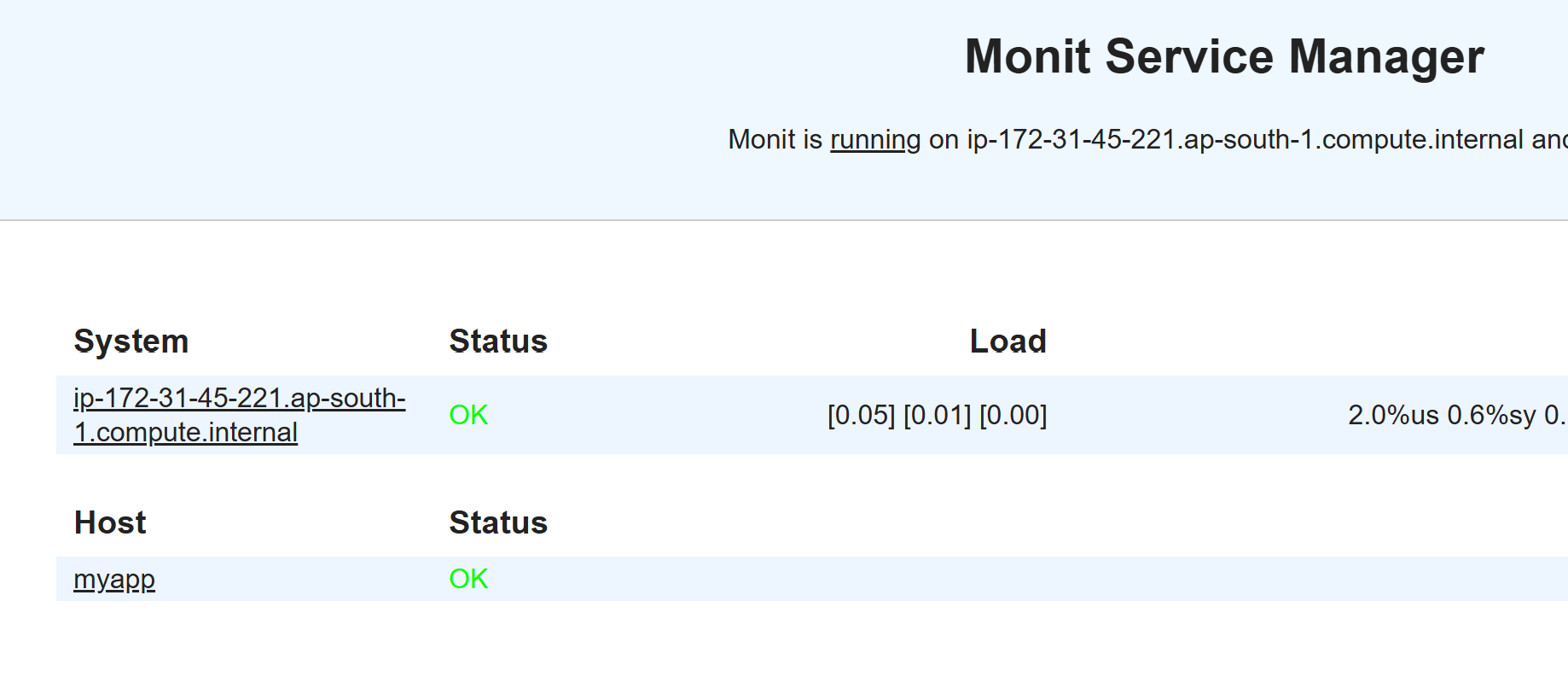
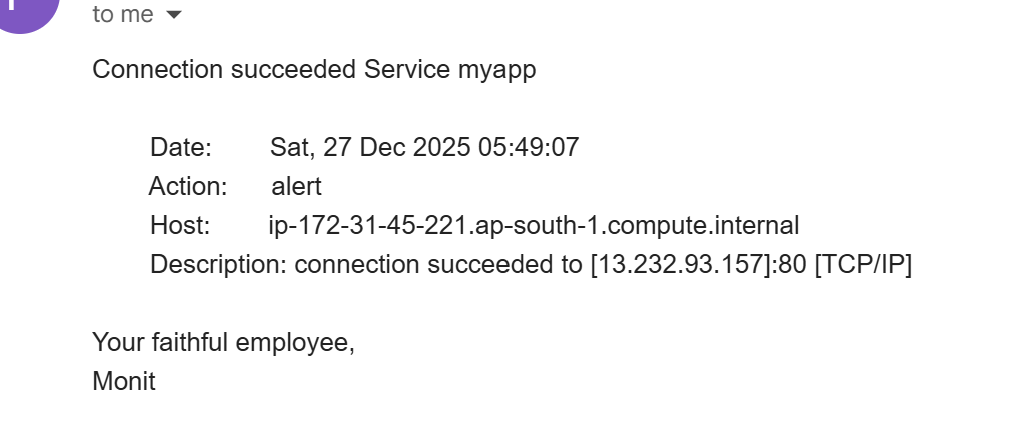






Once container started



Notes: I have changed the instance type post preparing the document so public IP has been changed