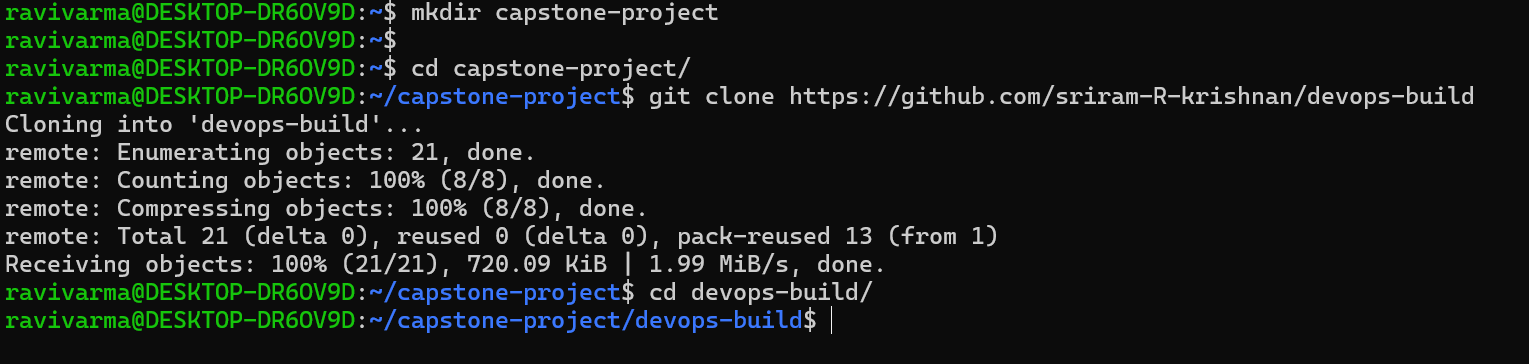
**Application Deployment Document**

**Objective:**  
Deploy the provided React application to a production-ready state, using Docker, Git, Jenkins, and AWS EC2.

**1. Application Setup**

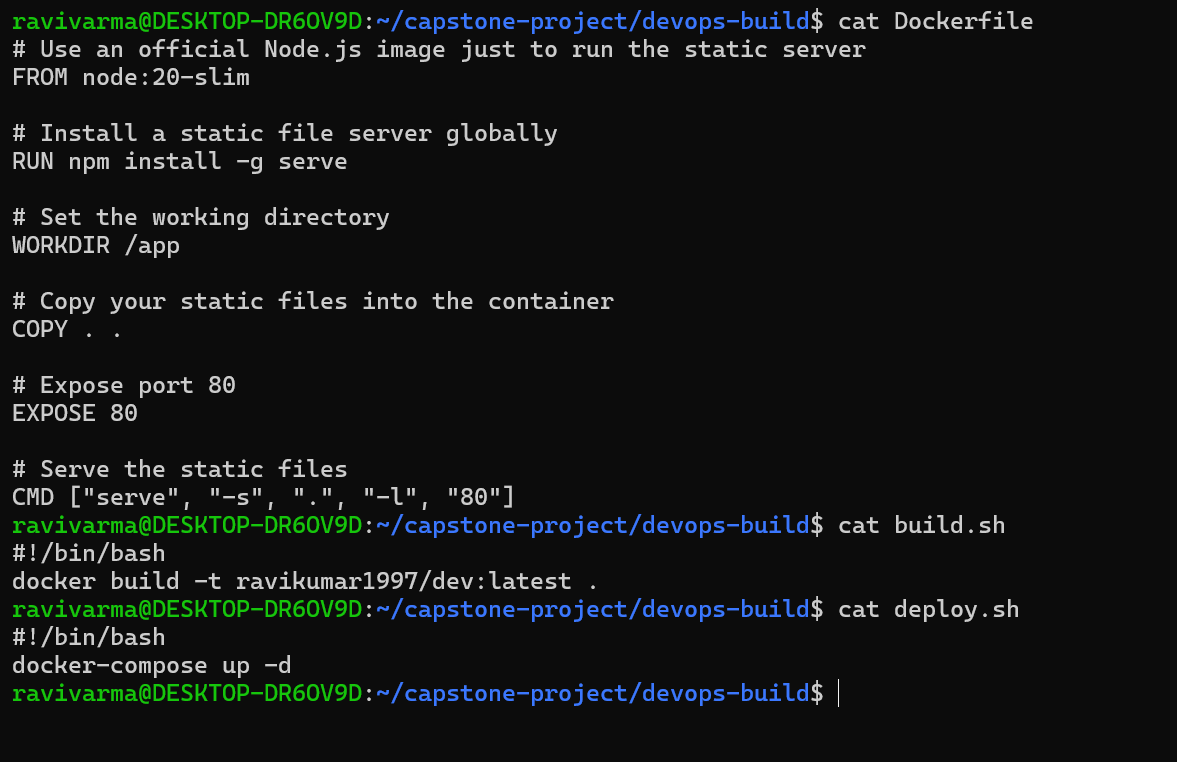
**Repository URL:** <https://github.com/sriram-R-krishnan/devops-build>

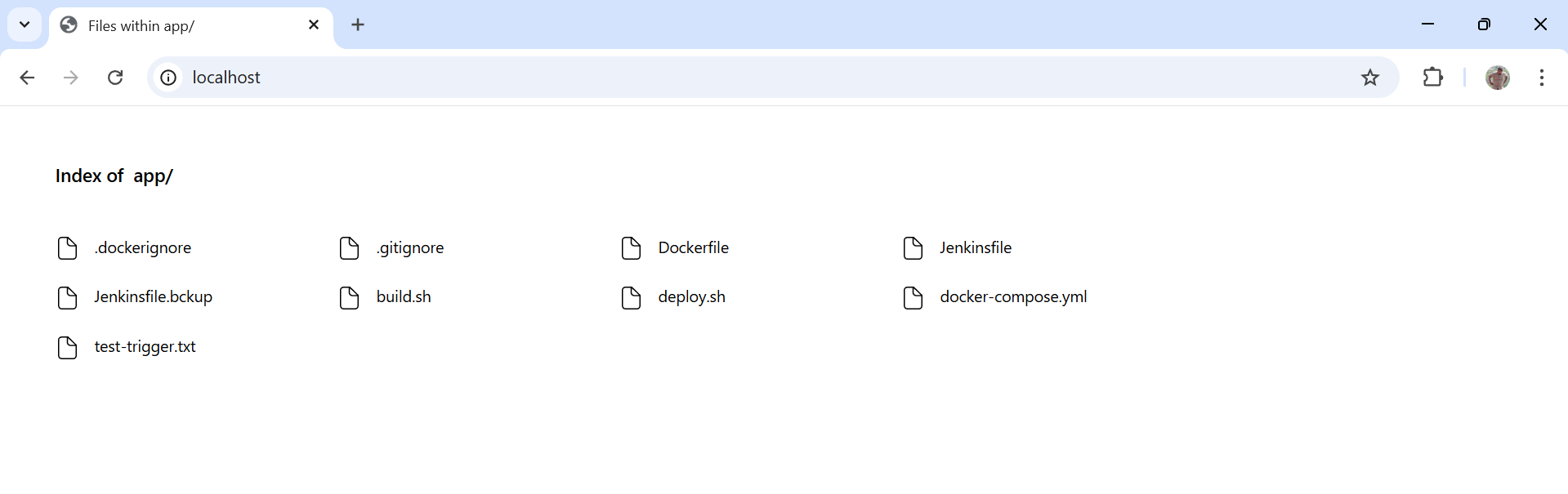
* Clone the repo and deploy the application.
* Run the application on port 80 (HTTP).



**2. Docker Configuration**

* Create a Dockerfile to dockerize the application.
* Create a docker-compose.yml to run the docker container from the image.

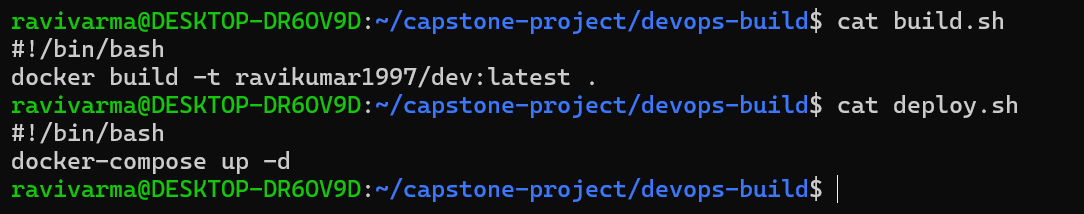


When run locally  


**3. Bash Scripting**

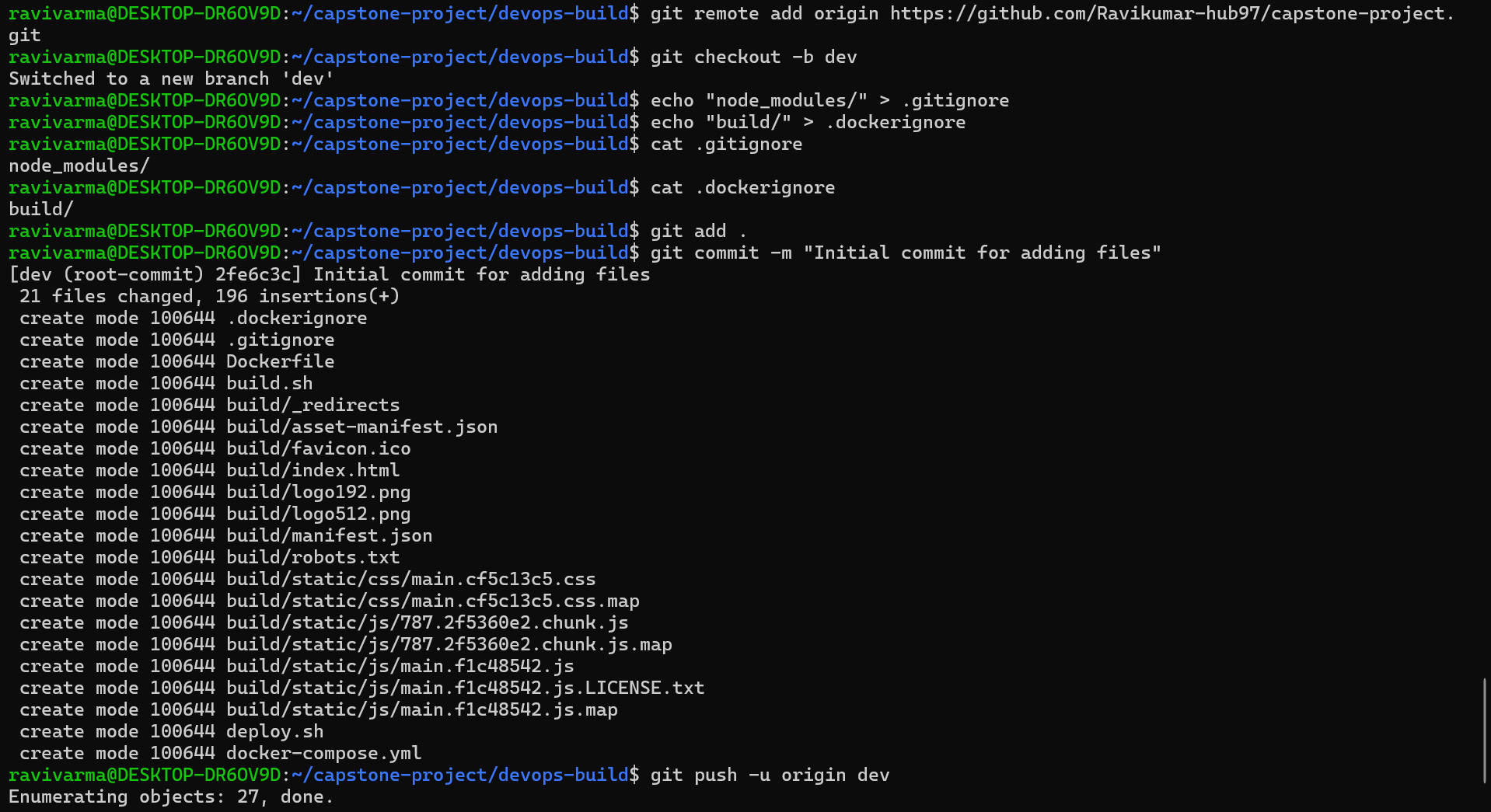
Create two bash scripts:

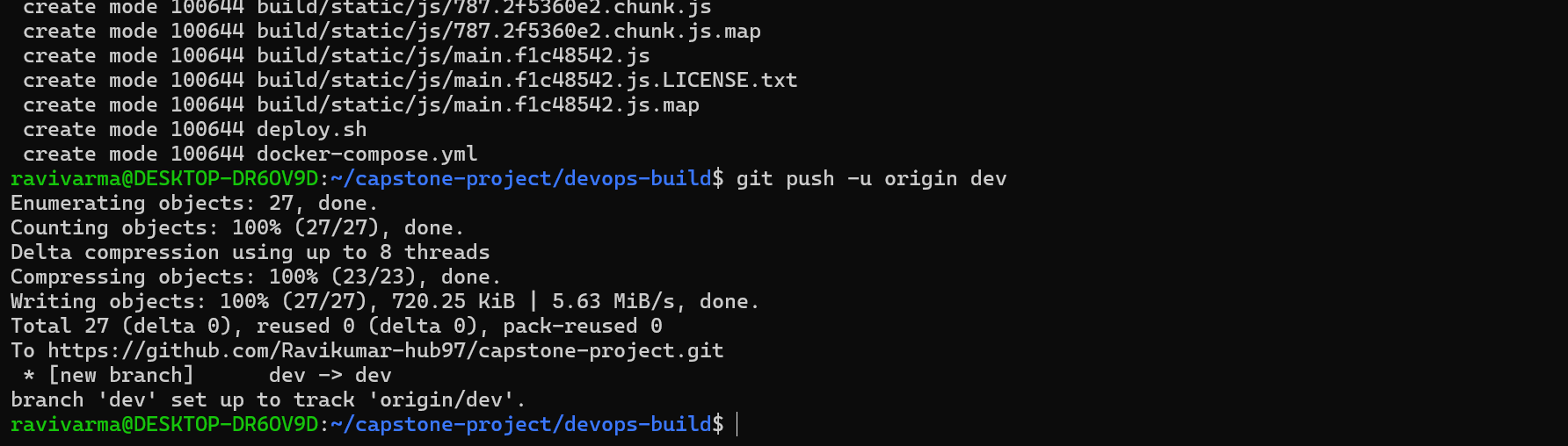
* build.sh - to build the Docker image.
* deploy.sh - to deploy the Docker image to a server.



**4. Version Control (Git)**

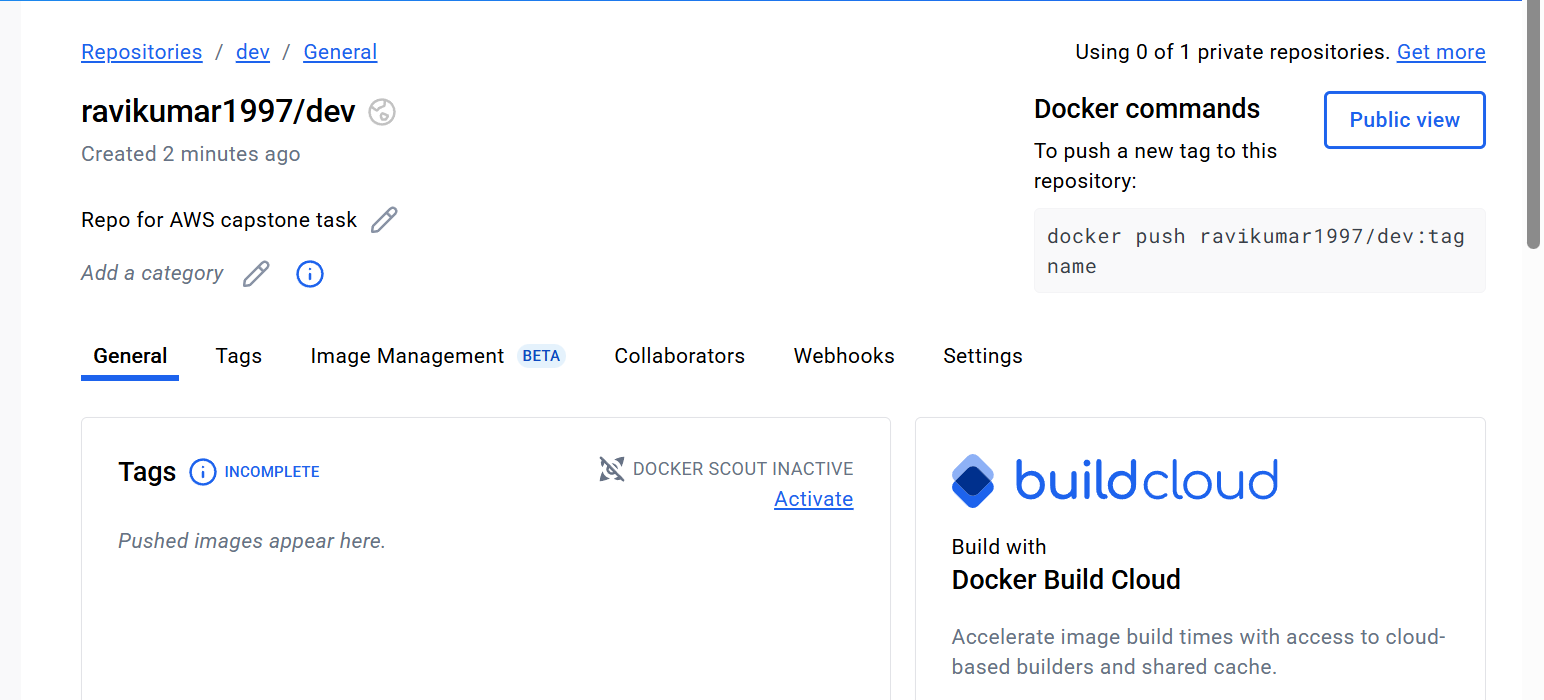
* Use Git CLI to push the code to the dev branch.
* Ensure .dockerignore and .gitignore files are properly set up.



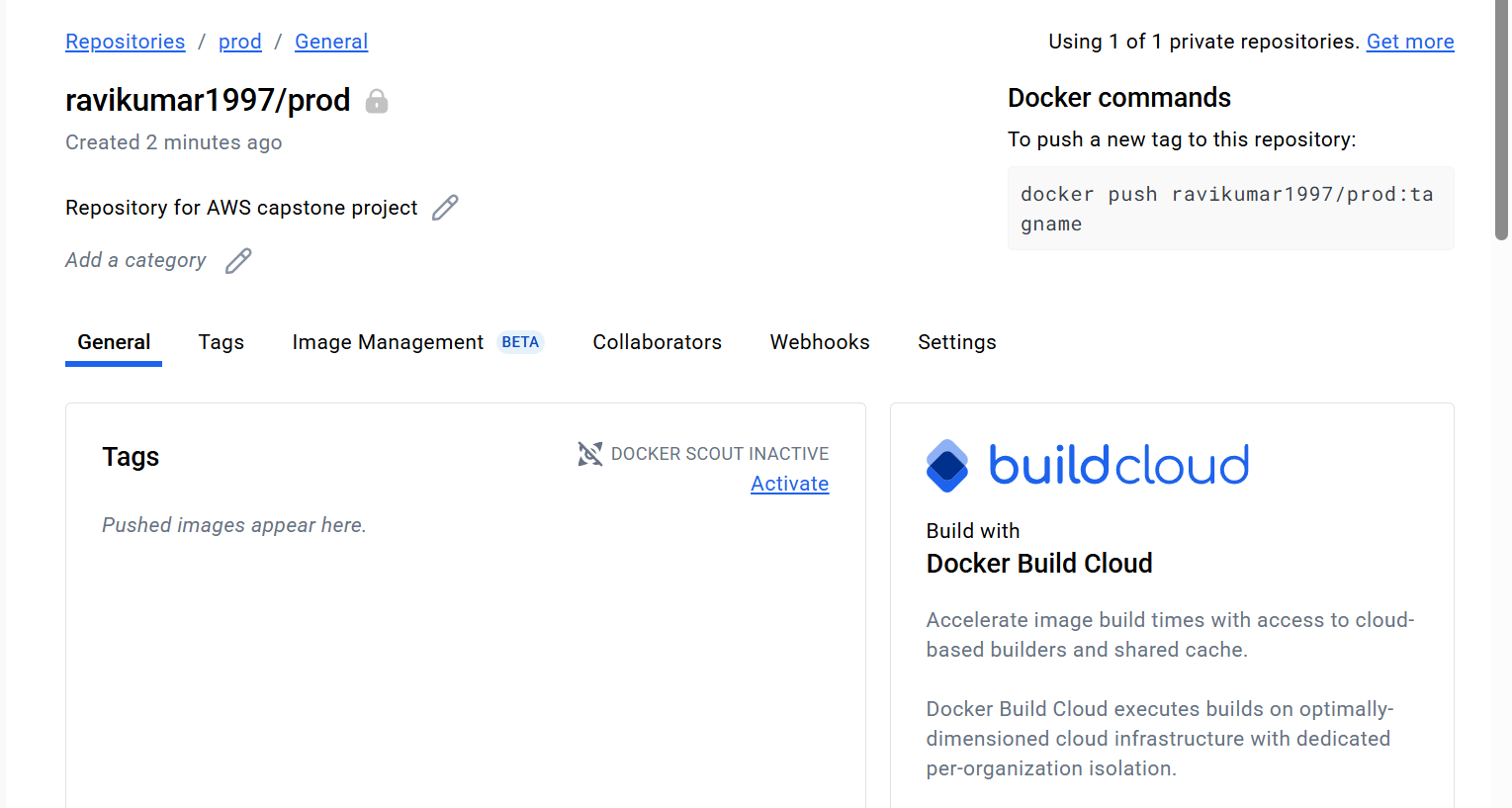


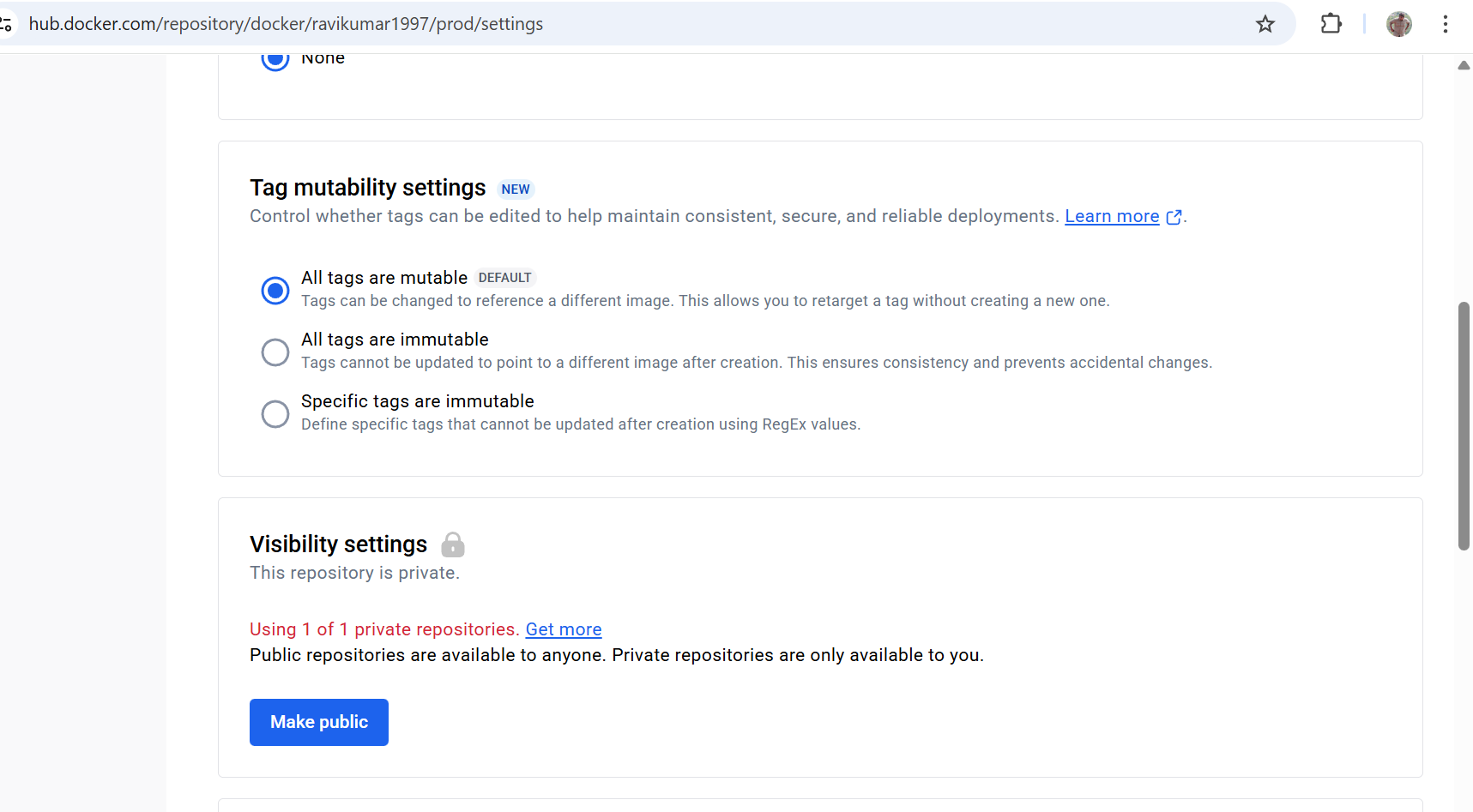
**5. Docker Hub Setup**

* Create two Docker Hub repositories: dev and prod.
* dev can be public.



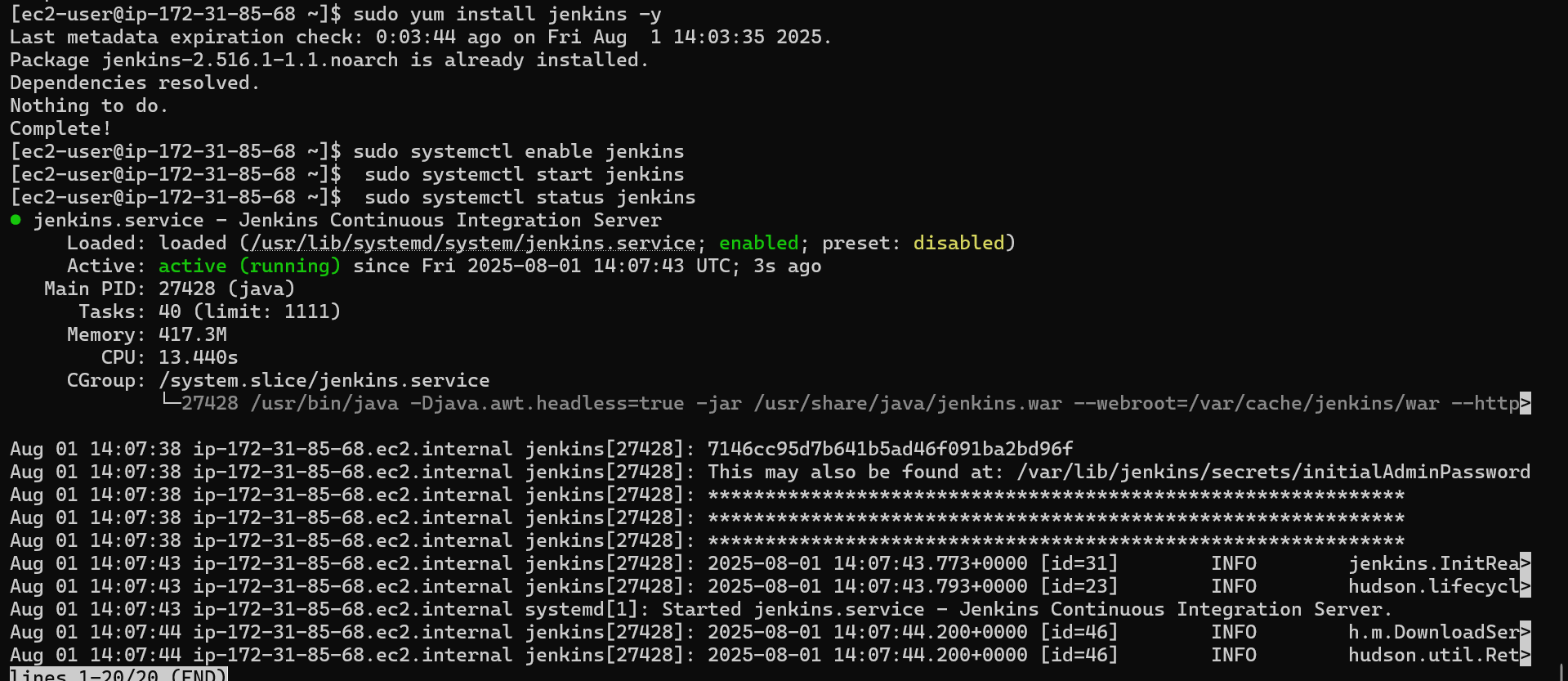
* prod must be private.

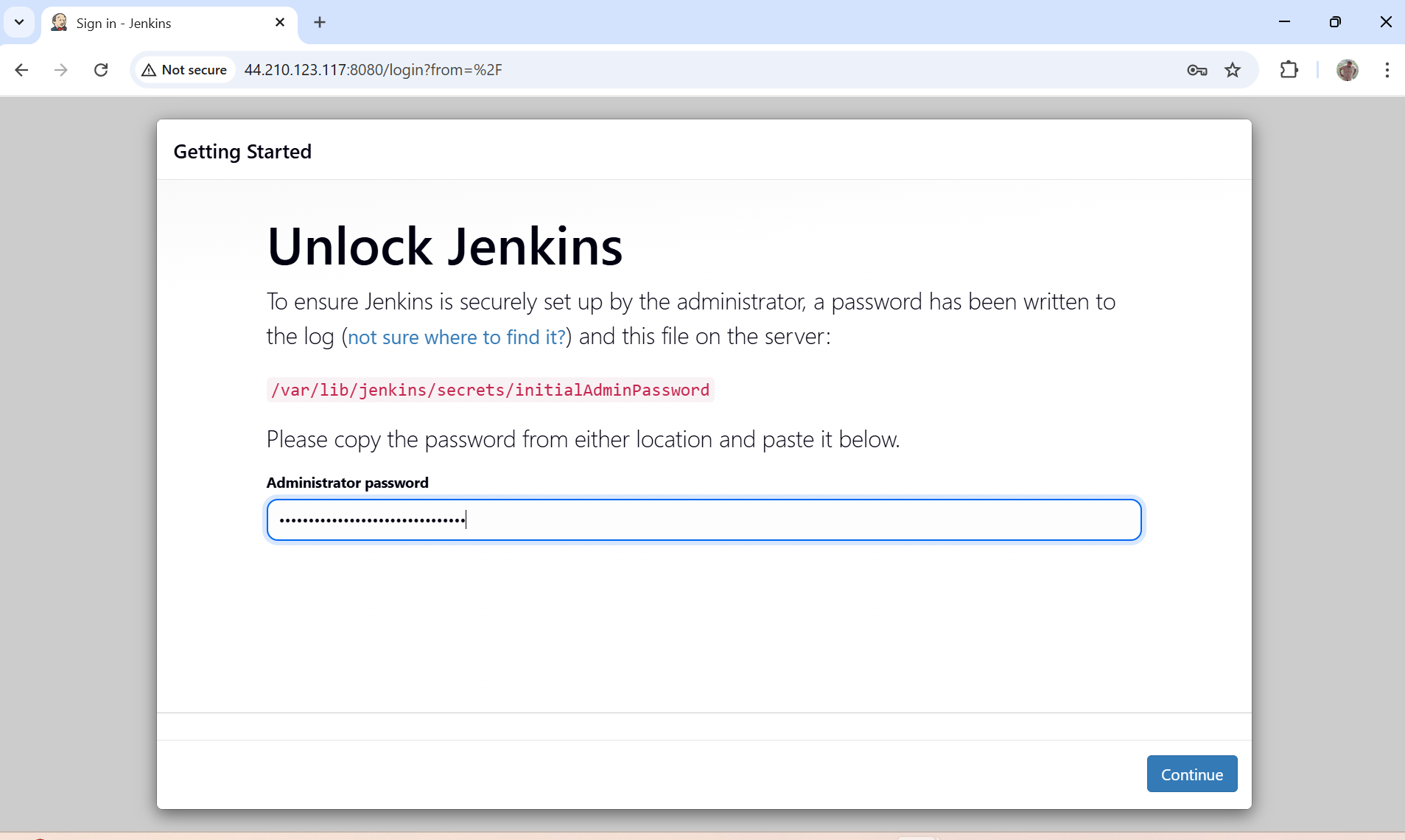


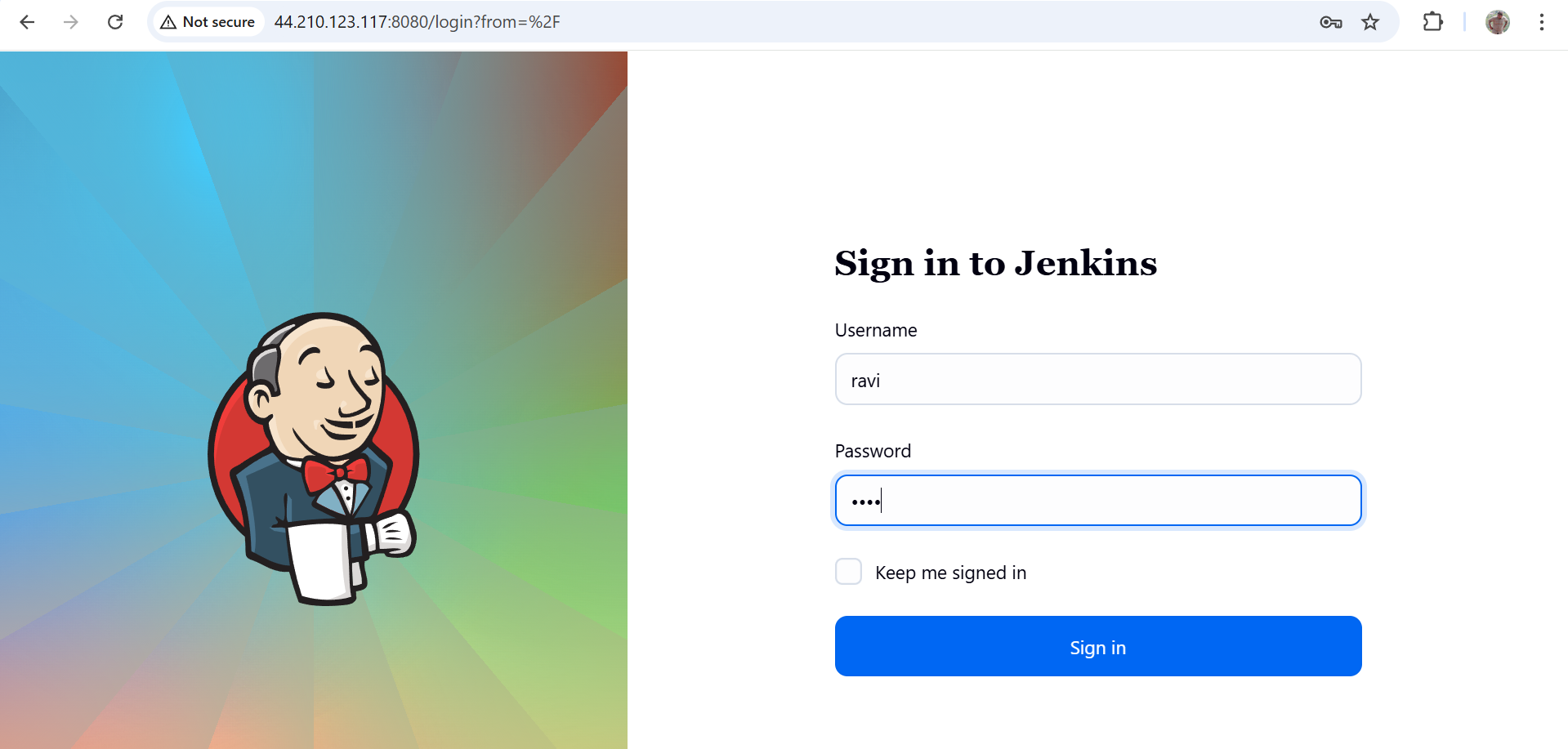


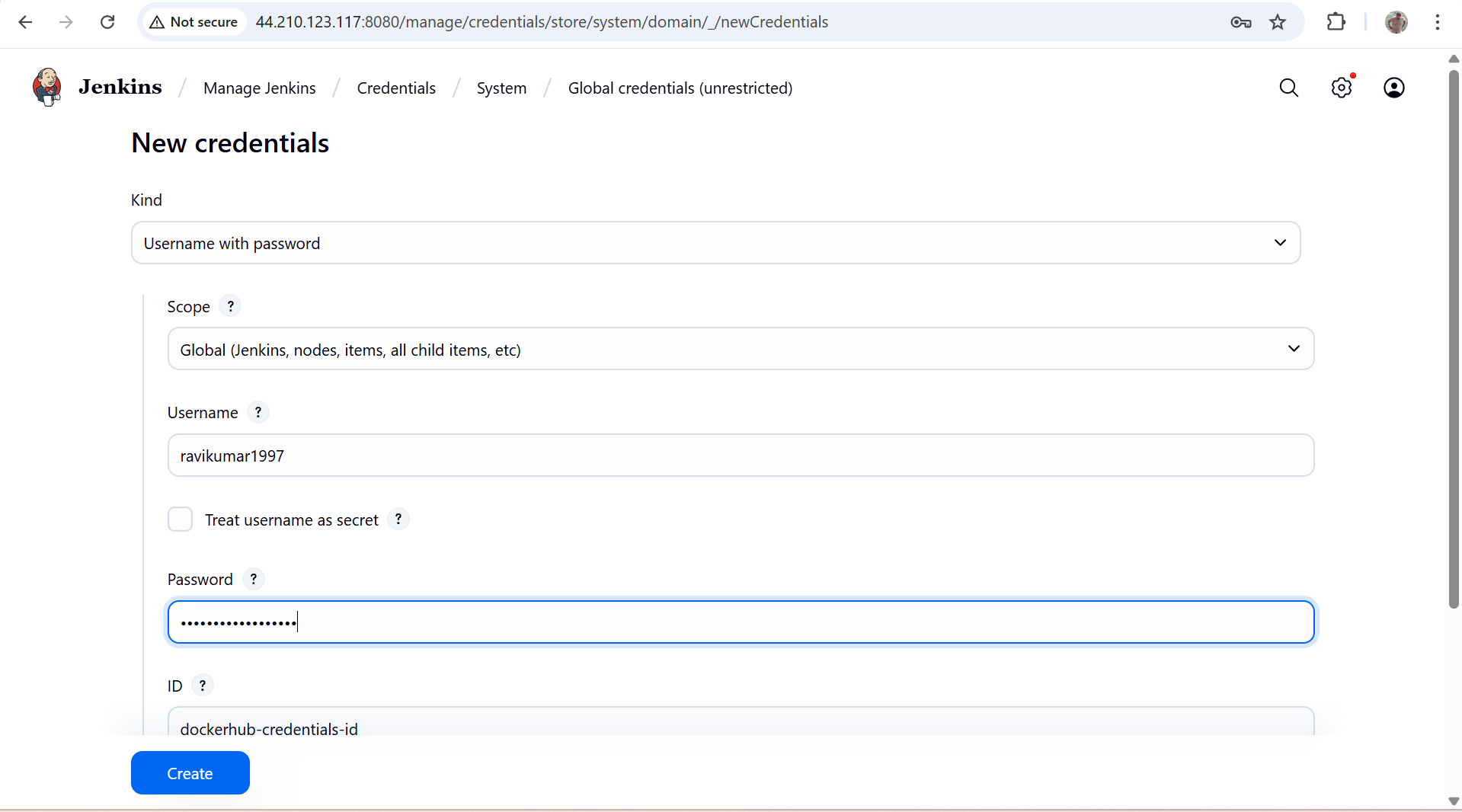
**6. Jenkins CI/CD Pipeline**

* Install and configure Jenkins.

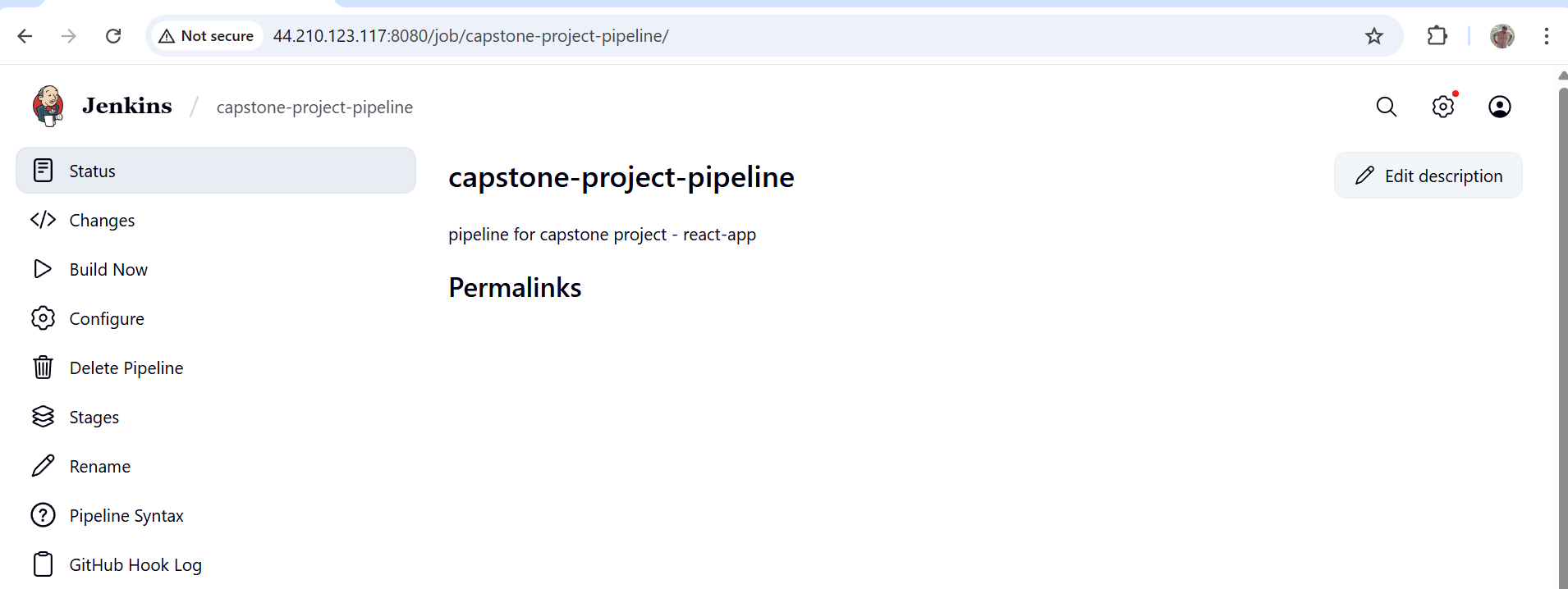


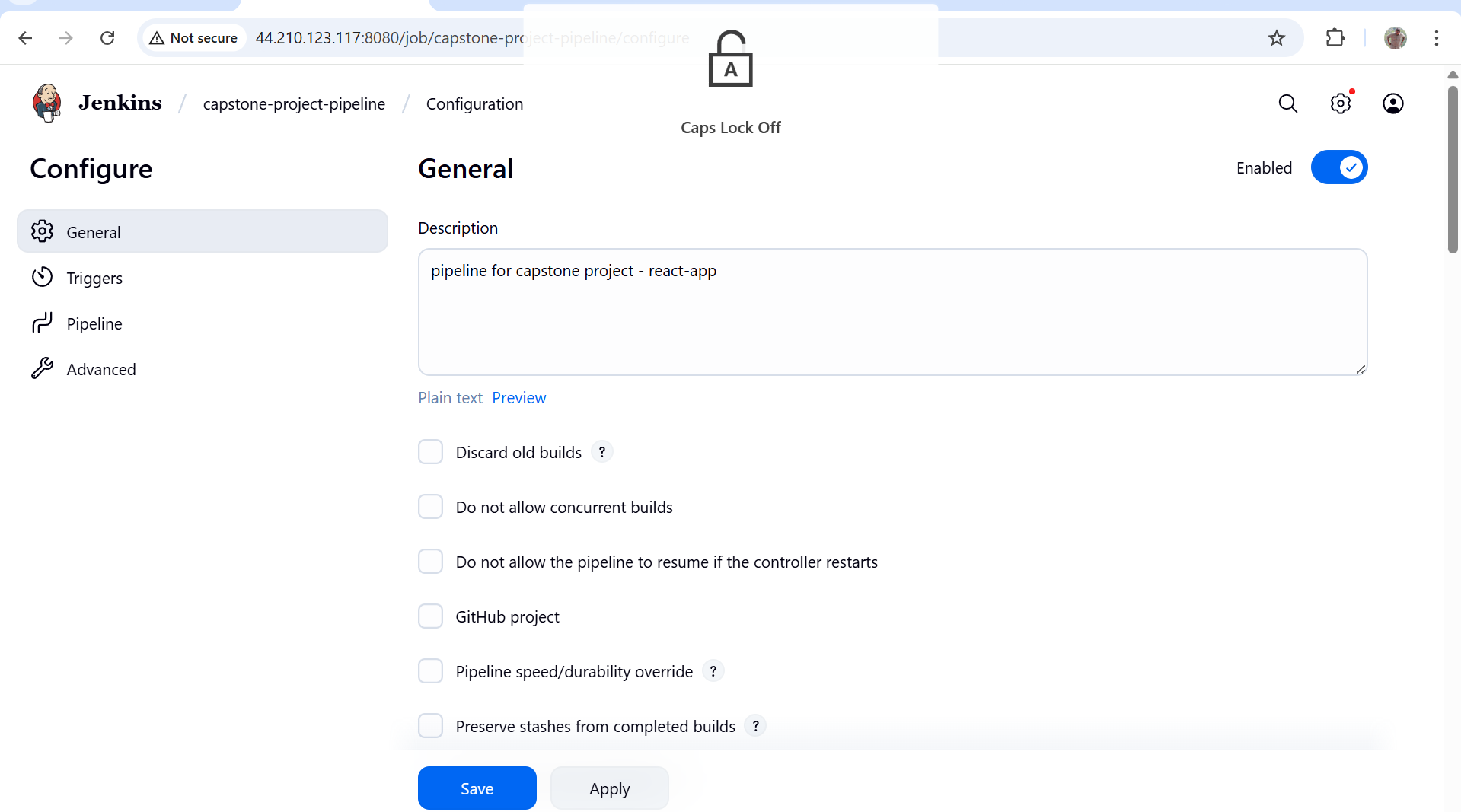




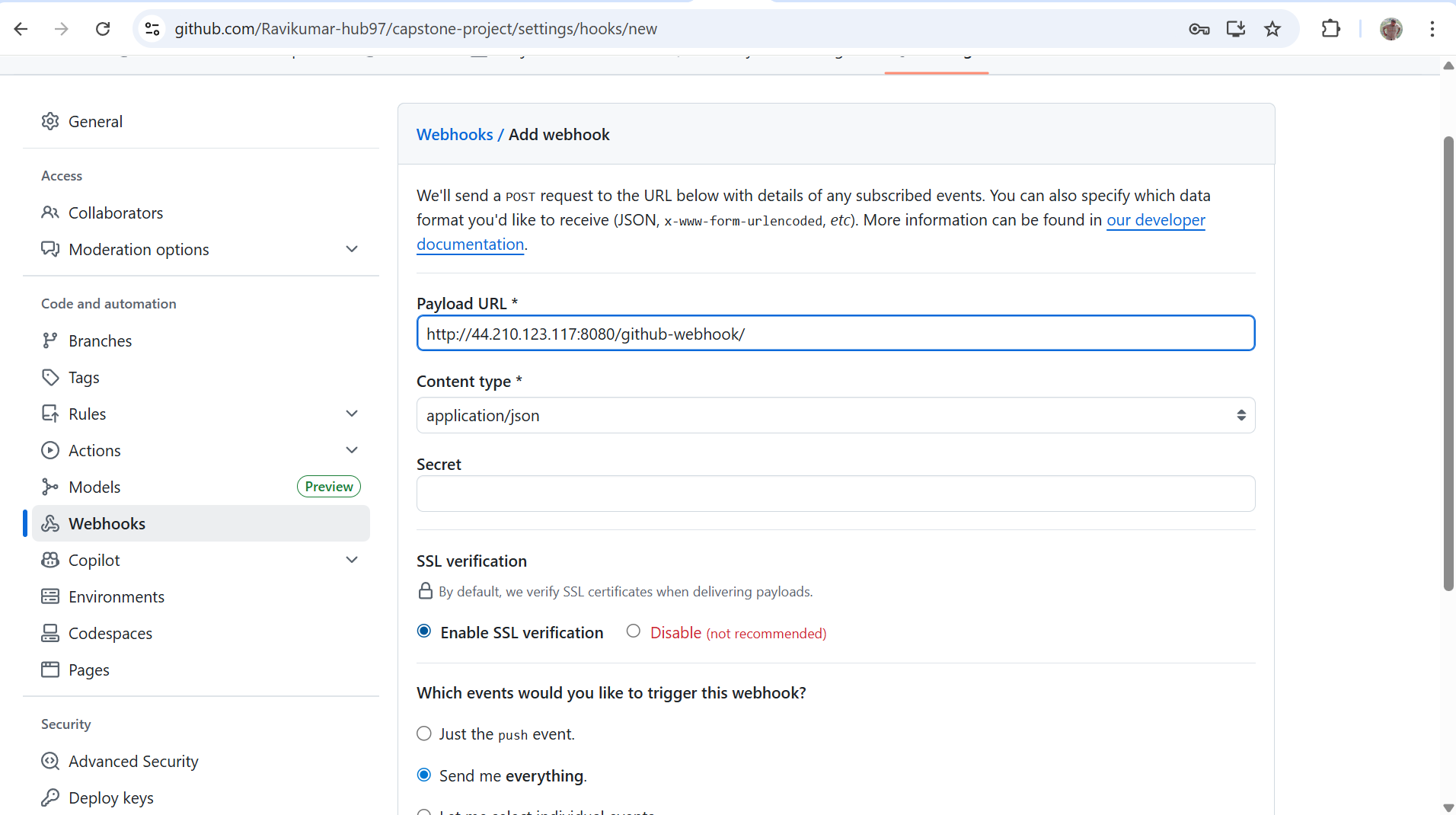


* Create build steps to build, push, and deploy the app.





* Connect Jenkins to the GitHub repo with auto-trigger on changes to dev and master branches.

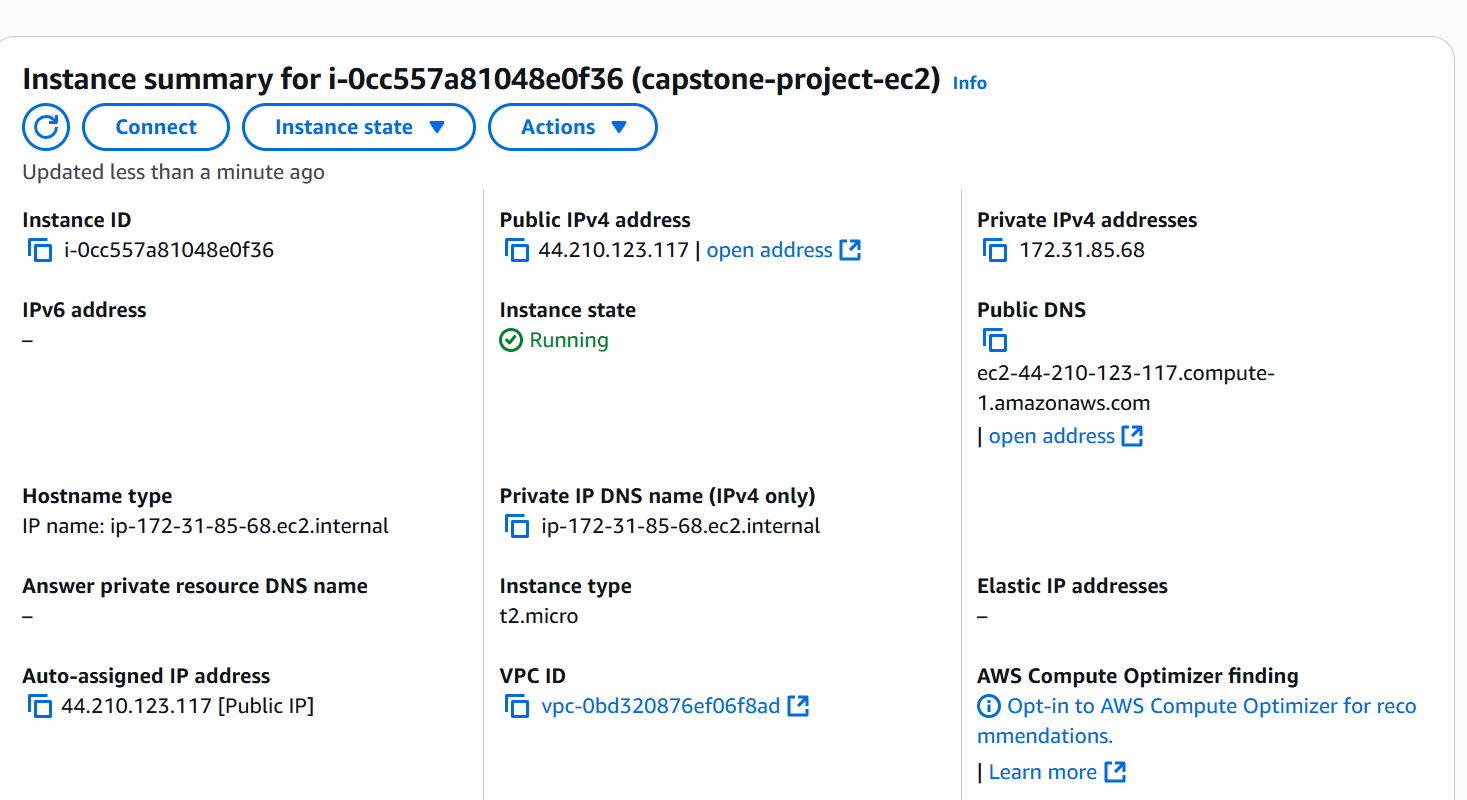


**Build Logic:**

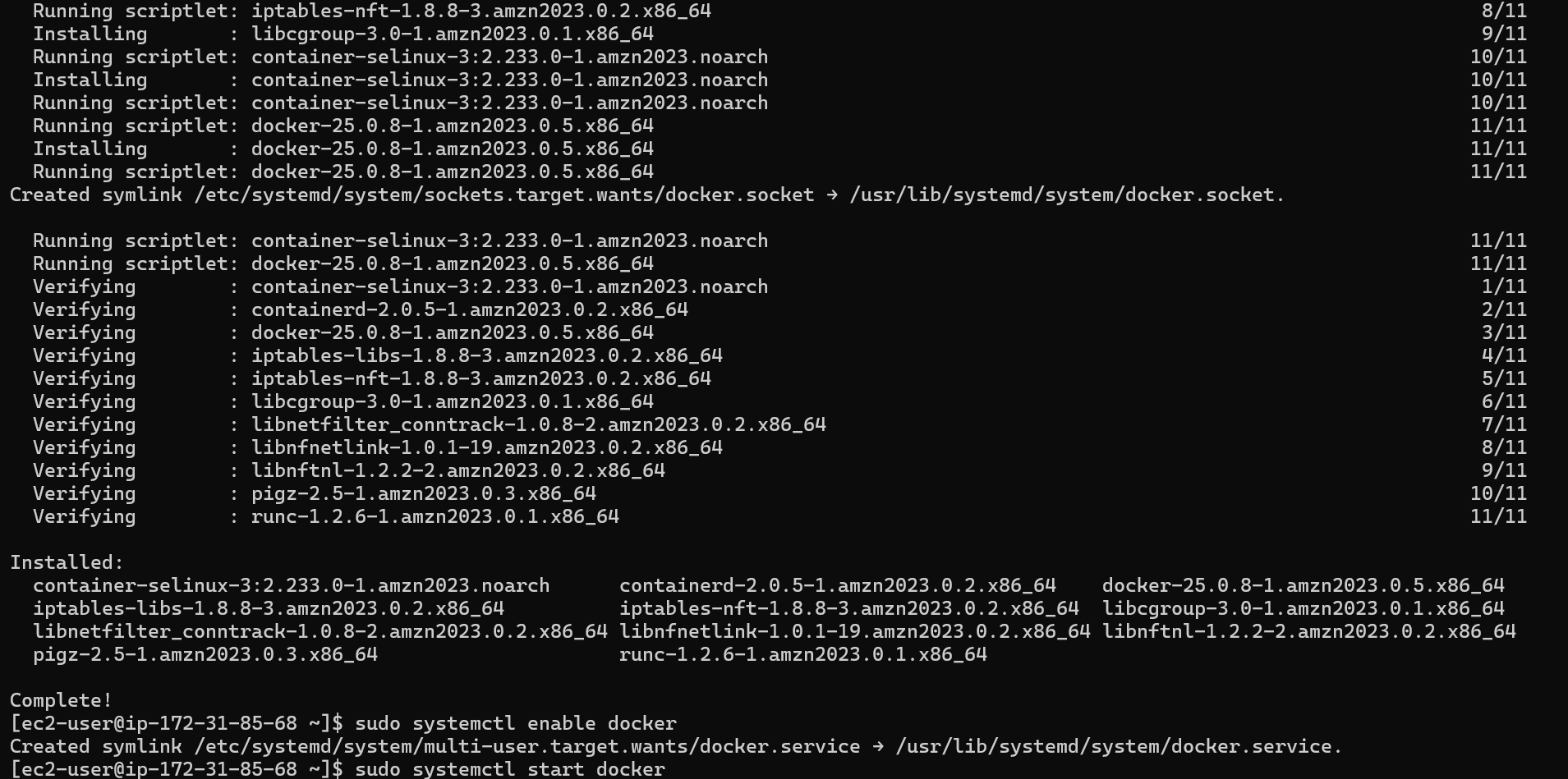
* If code is pushed to dev, build image and push to dev repo.
* If code is merged to master, push image to prod repo.

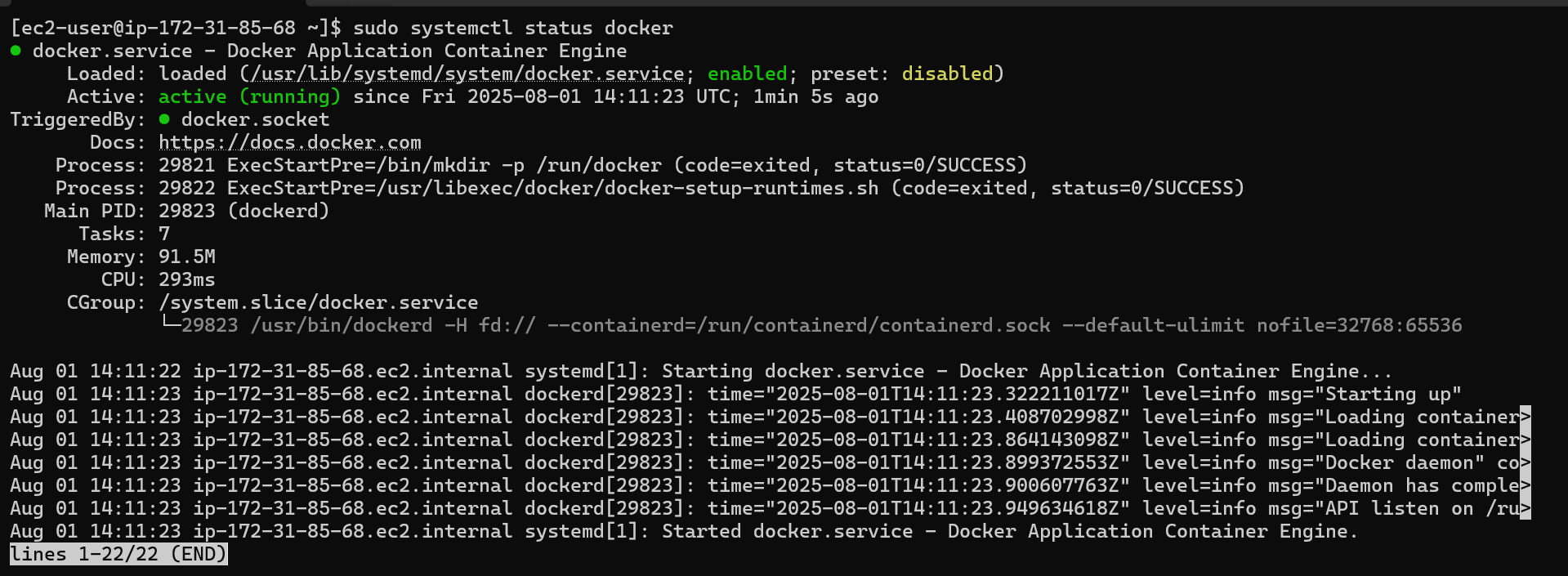
**7. AWS EC2 Deployment**

* Launch a t2.micro EC2 instance.



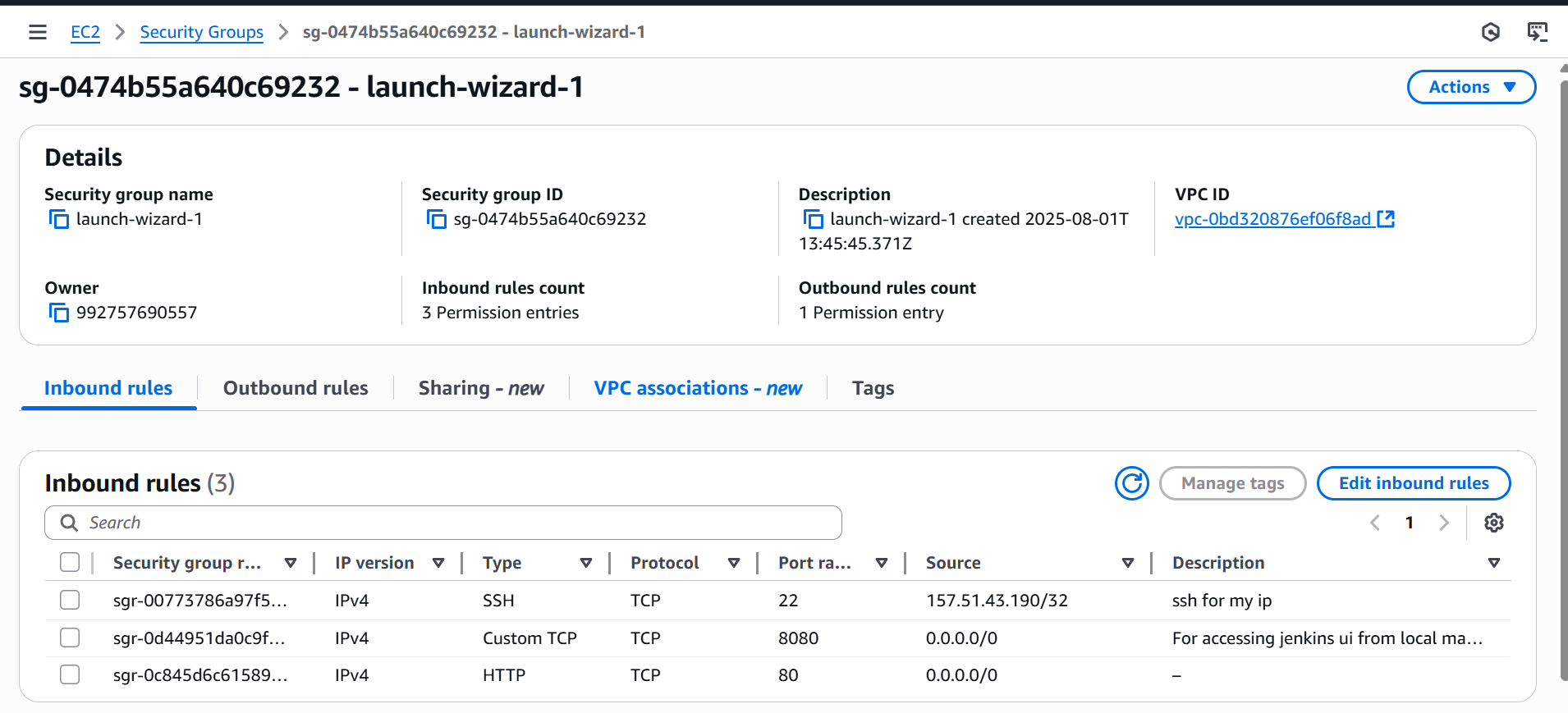
* Deploy the built application.



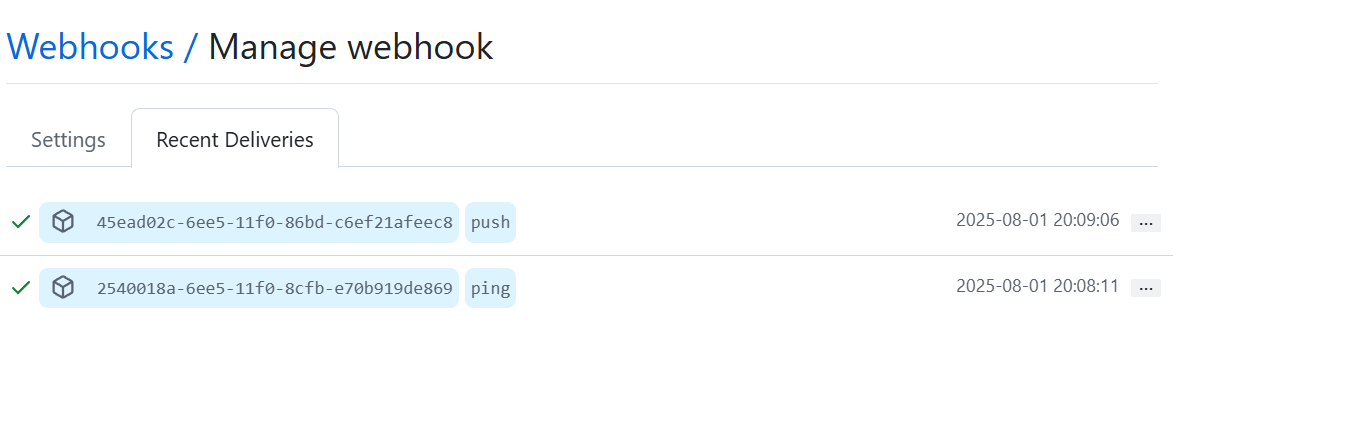


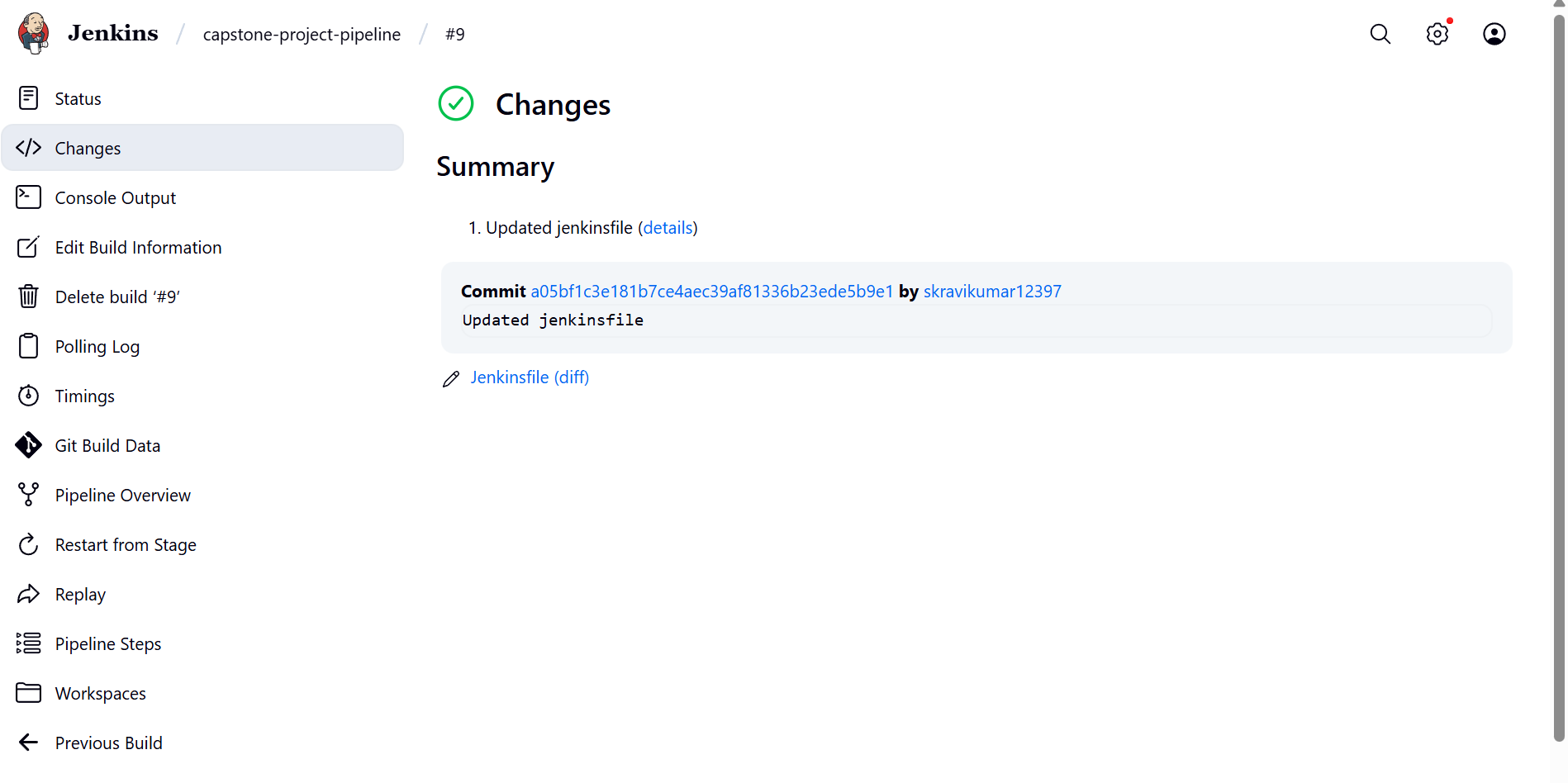
**Security Group Configuration:**

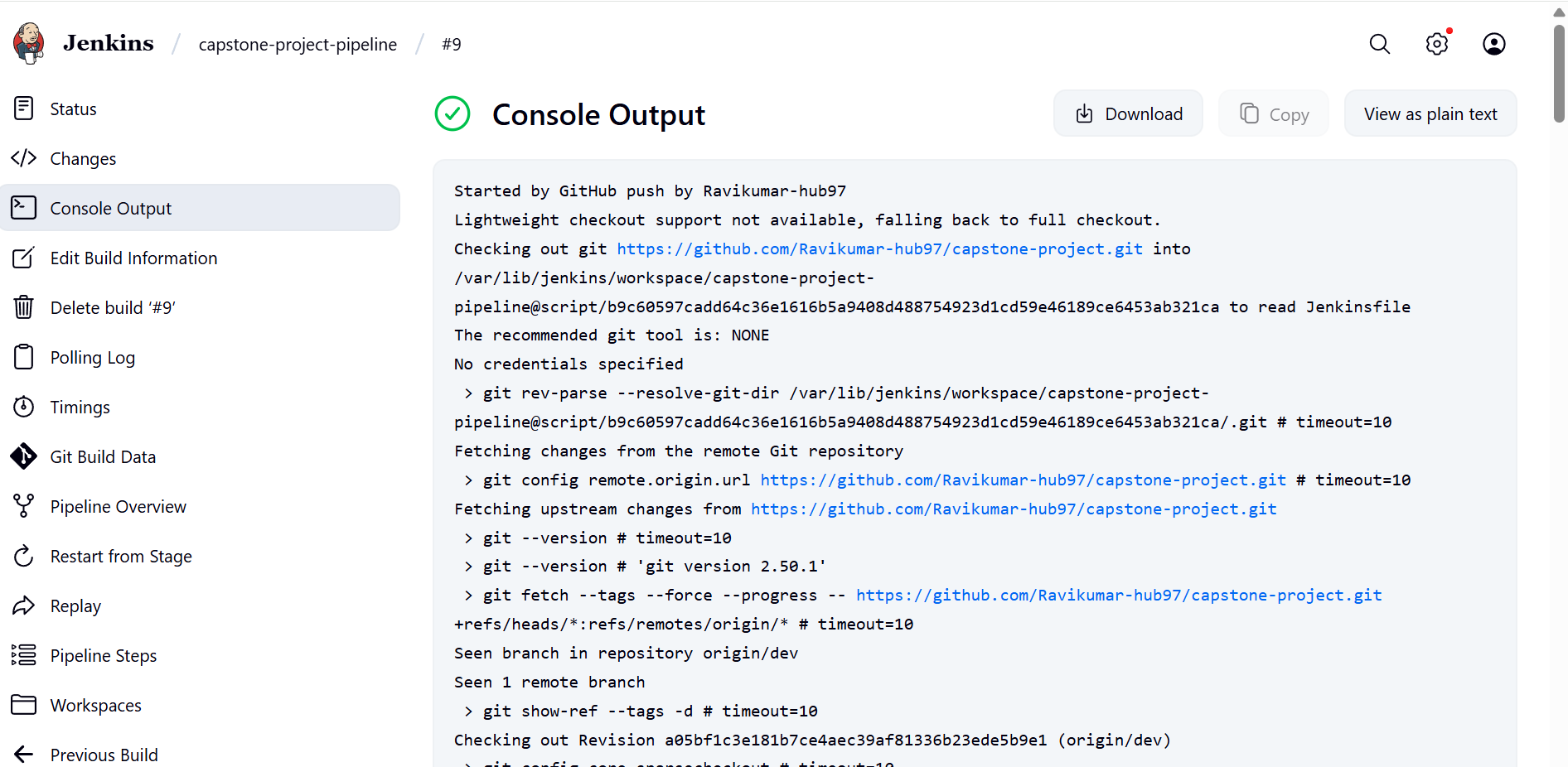
* Allow HTTP (port 80) access from any IP.
* Allow SSH (port 22) access only from your IP.



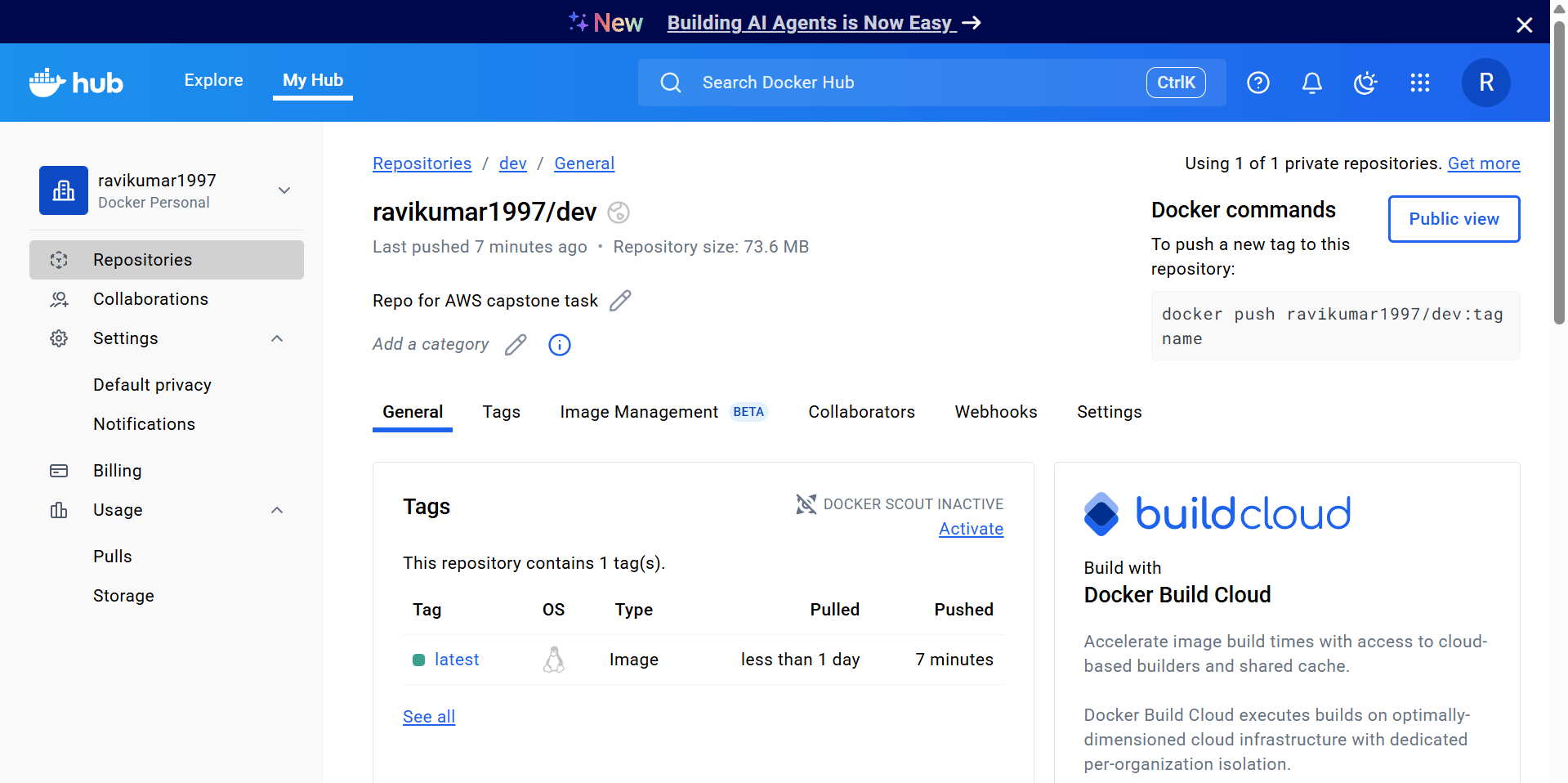
Push gets delivered through webhook

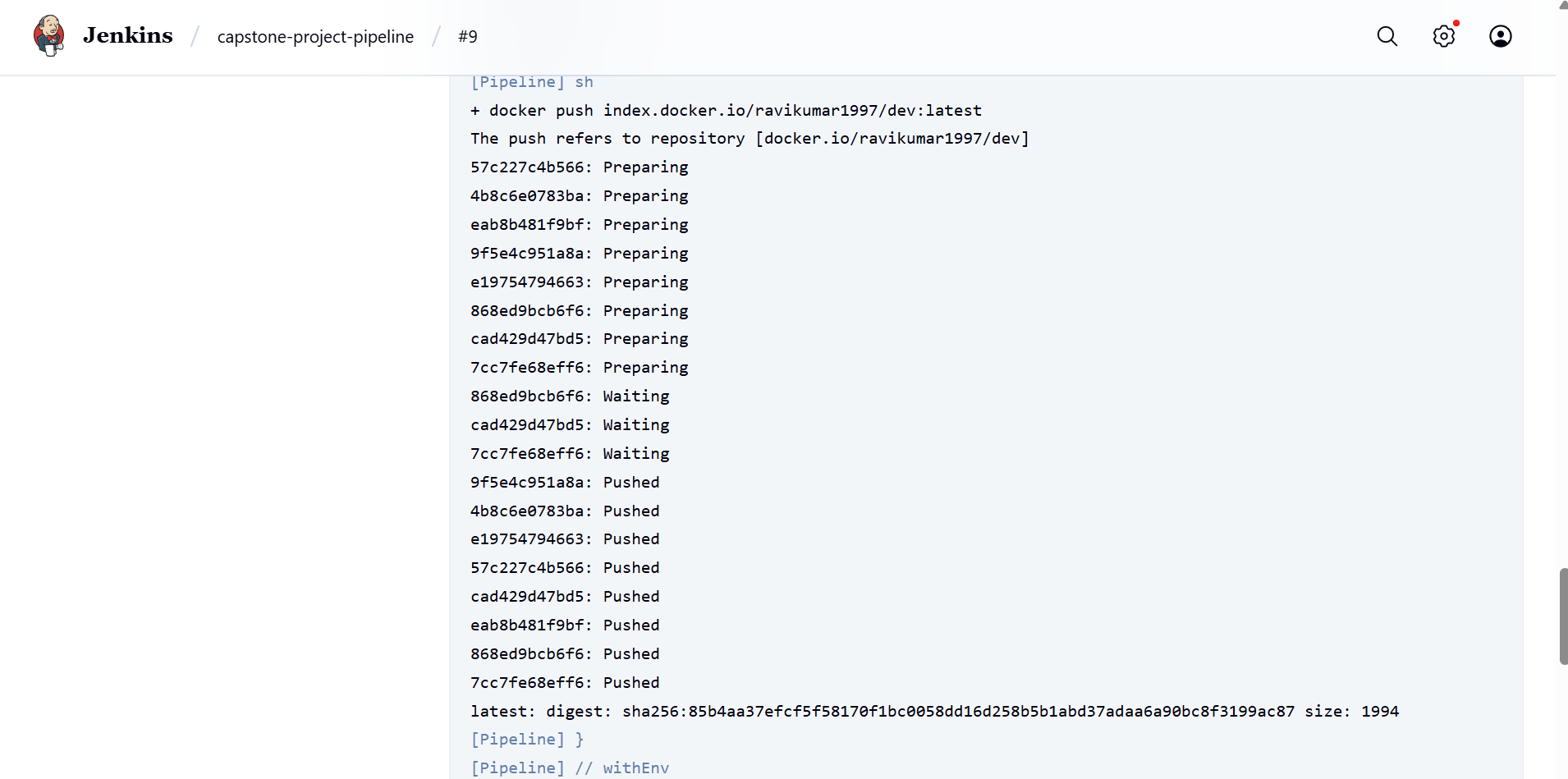




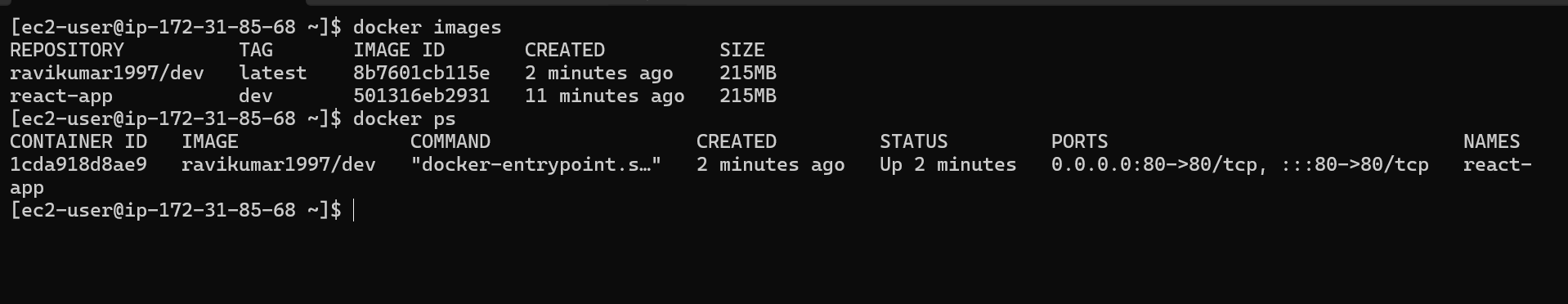


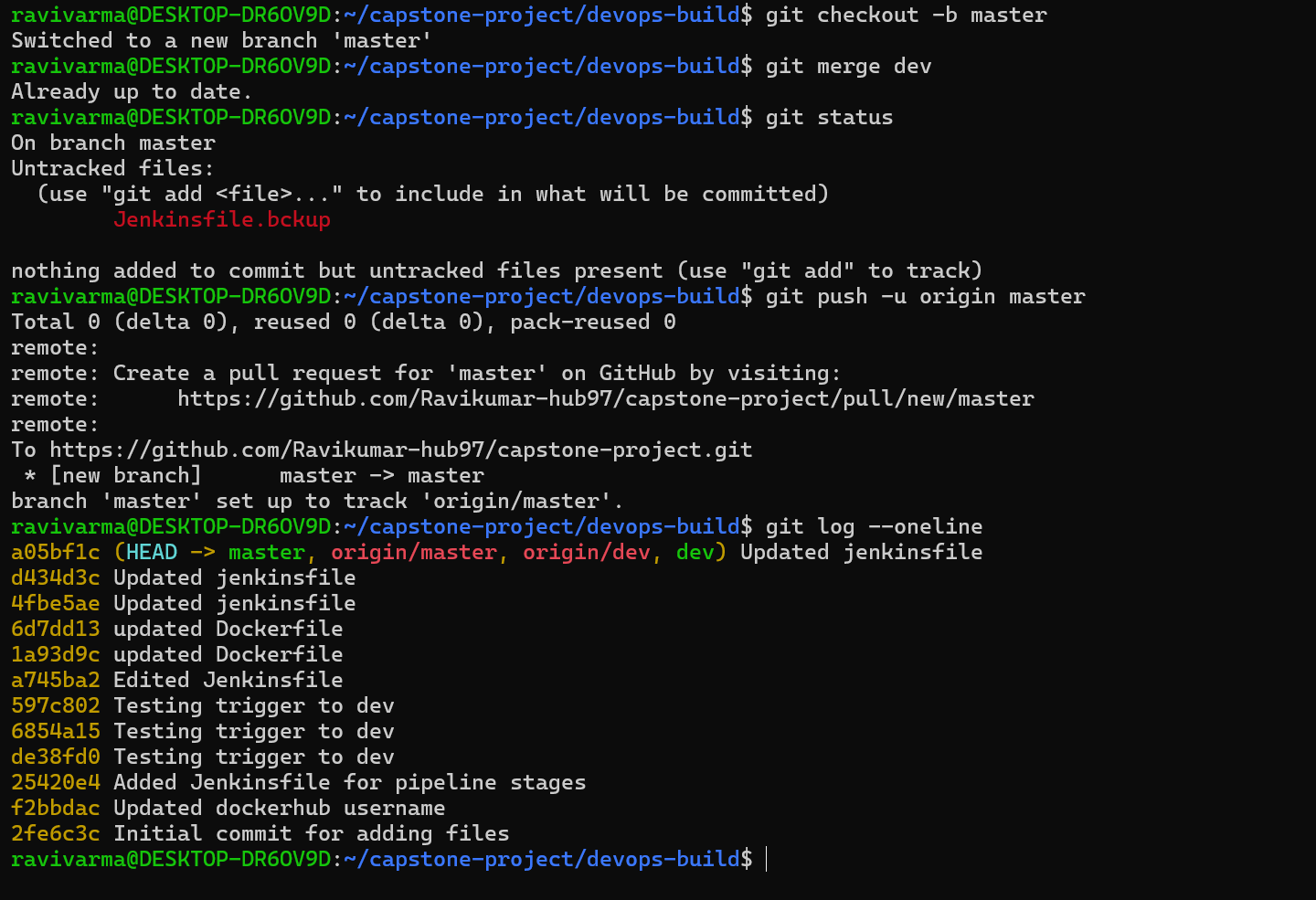
The push to dev is configured to push the image to dev repo in dockerhub

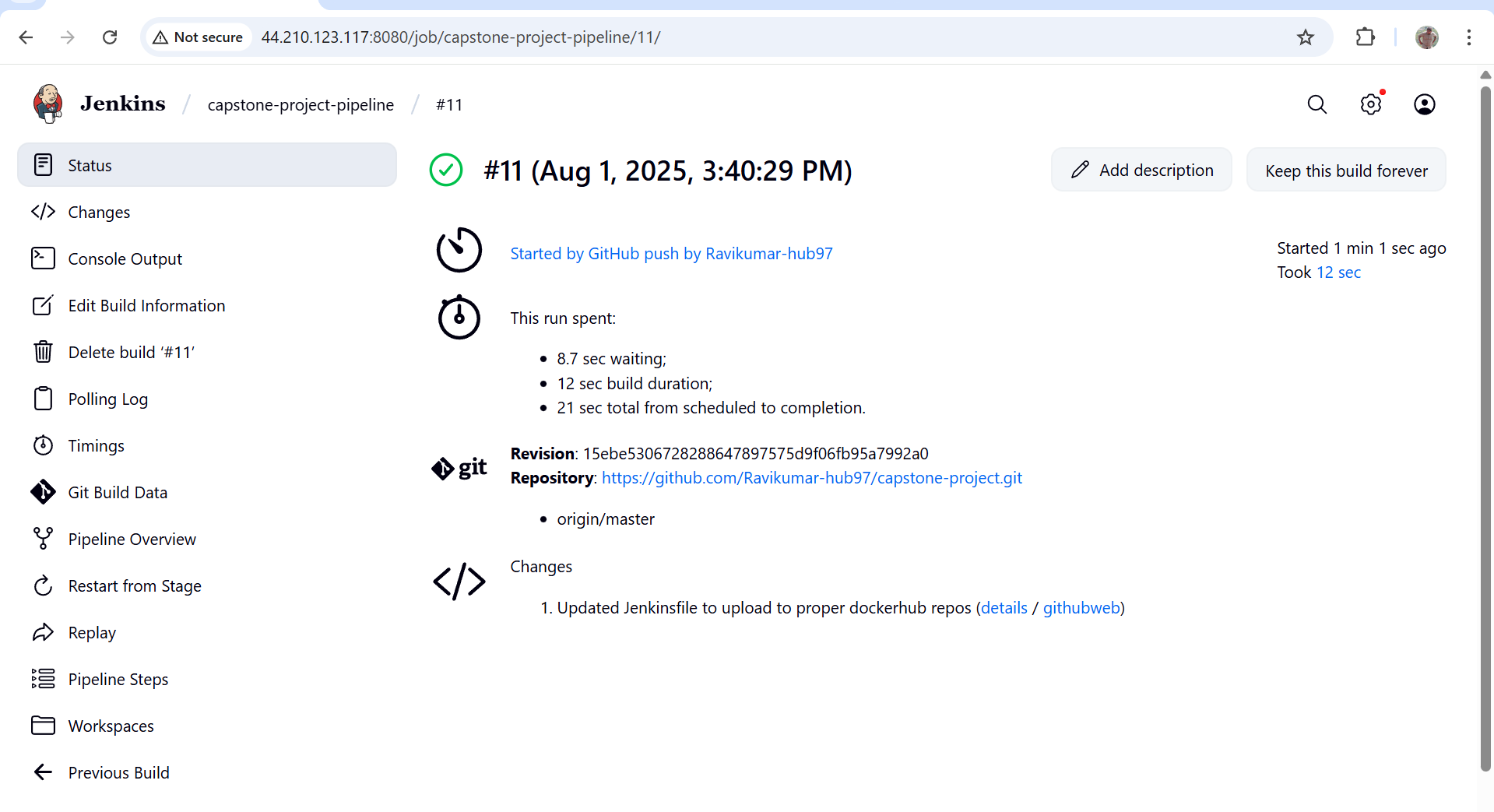


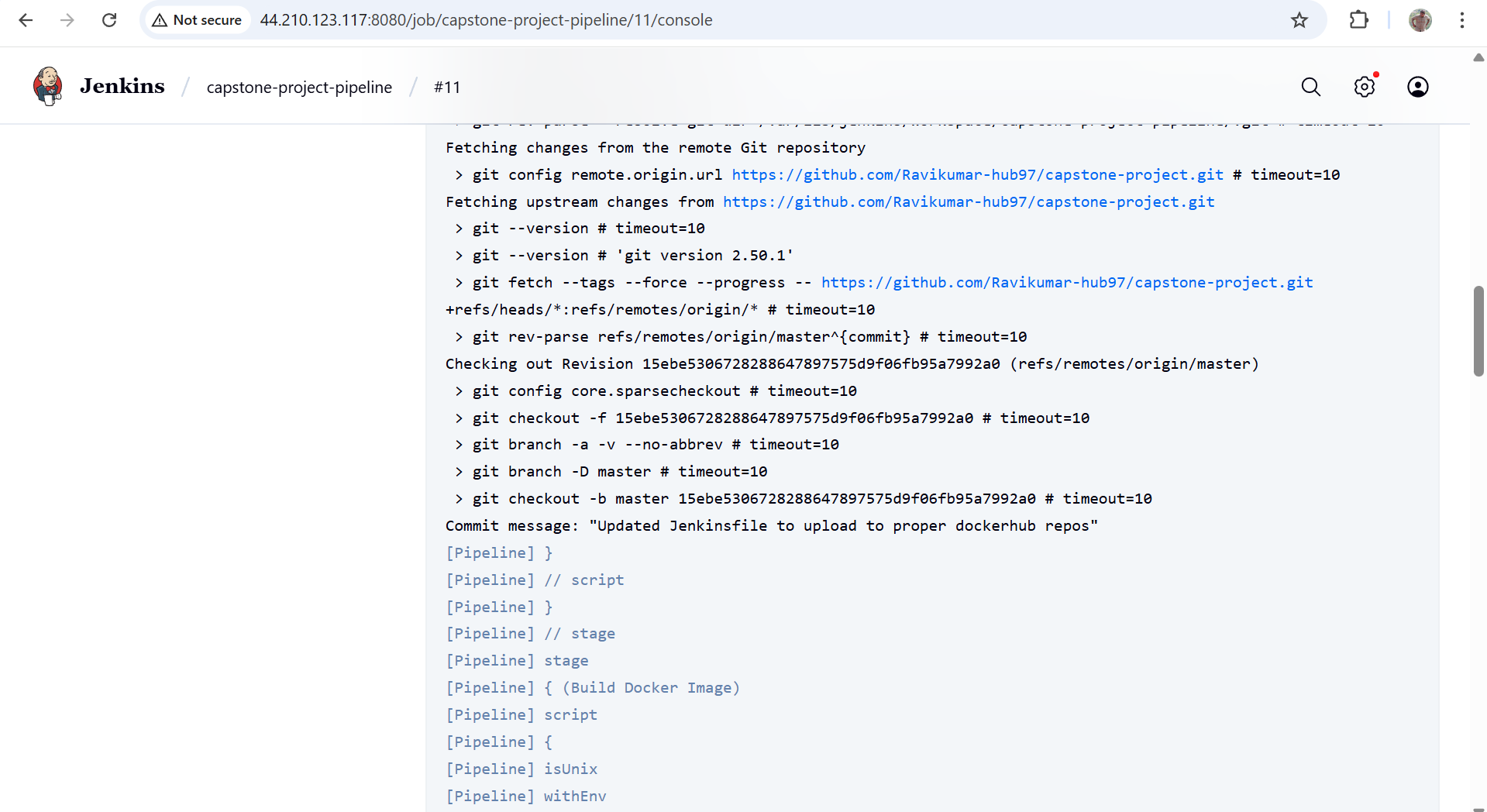


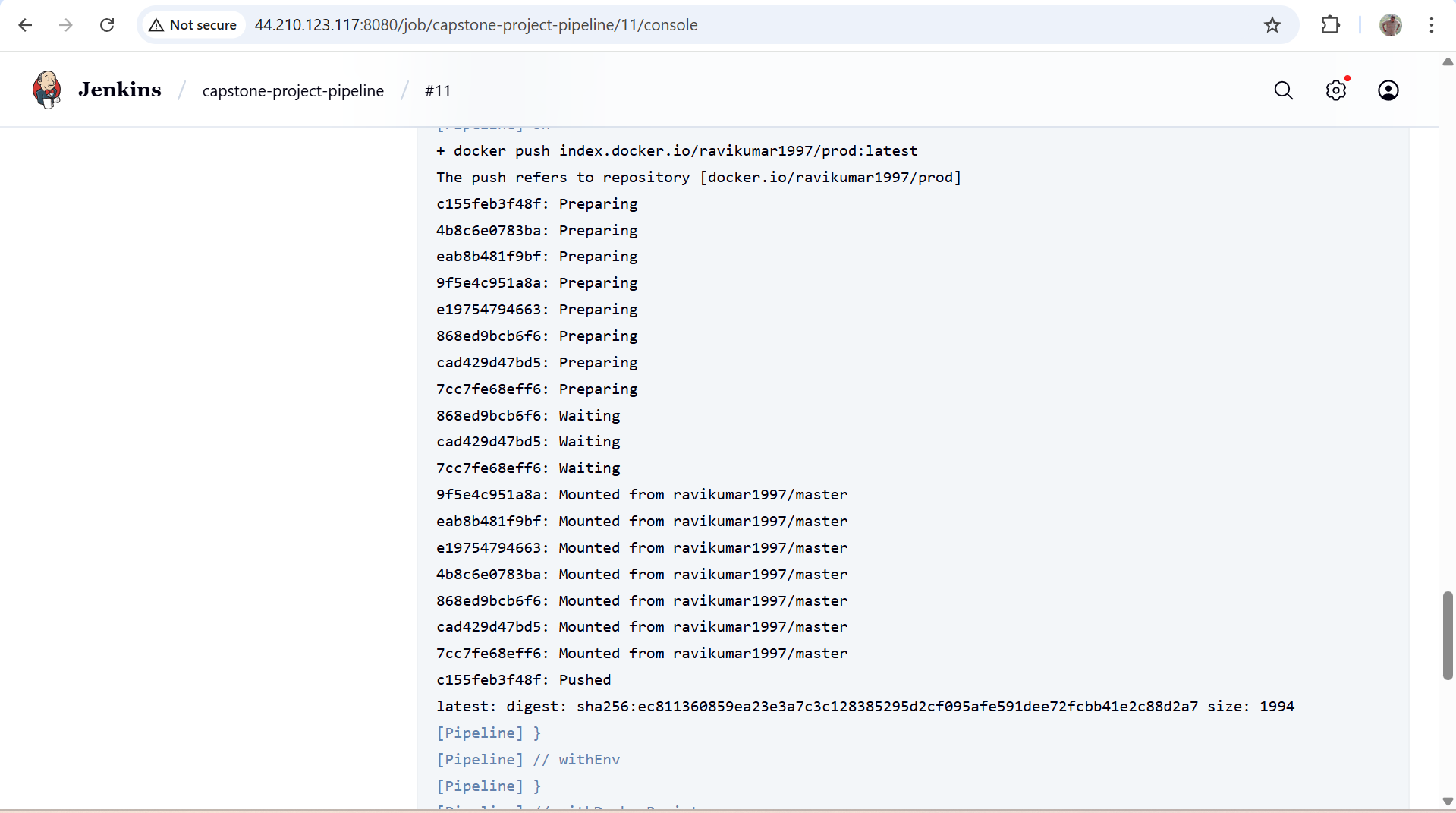
I added the deployment step too whenever code is pushed.

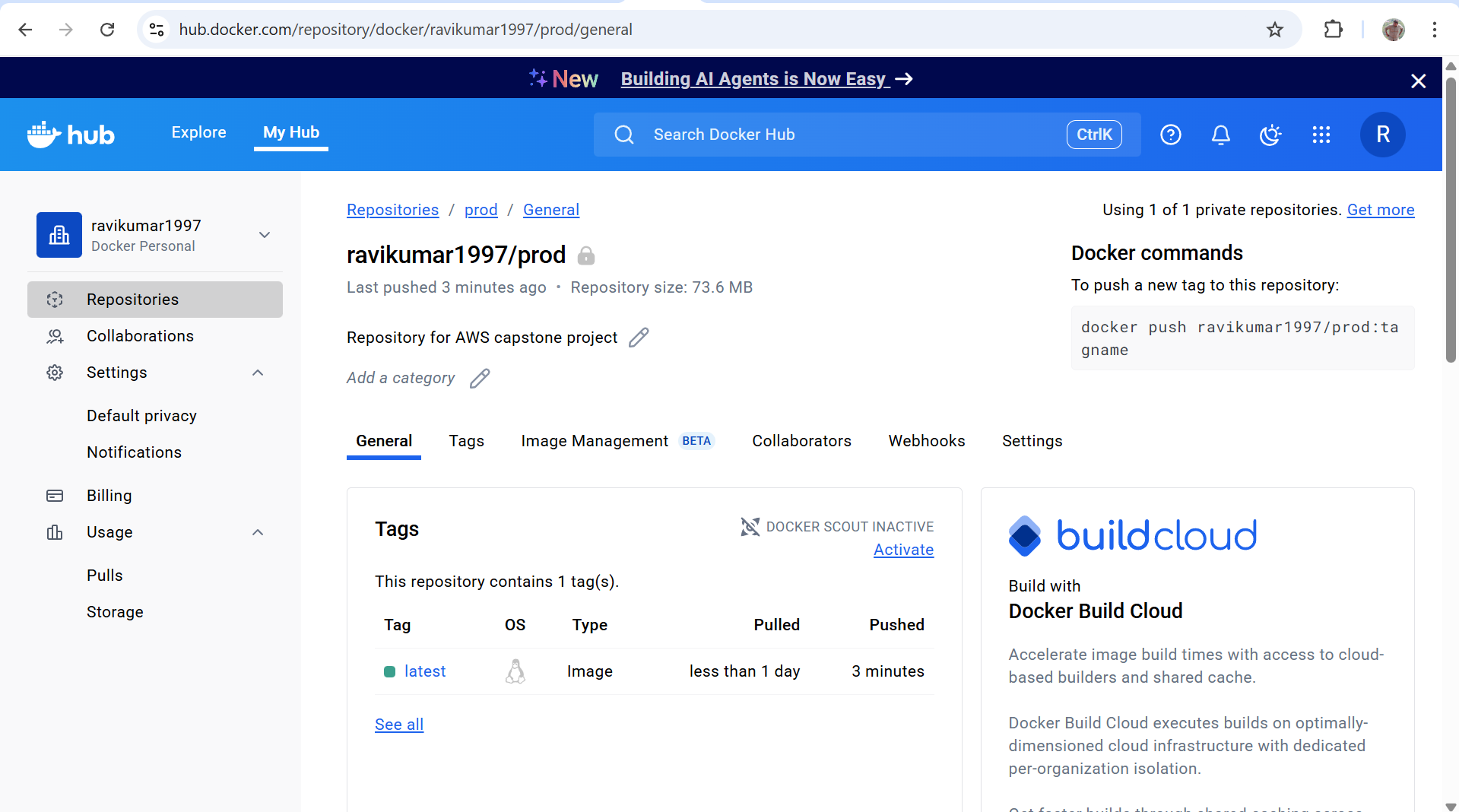


Now creating master branch   


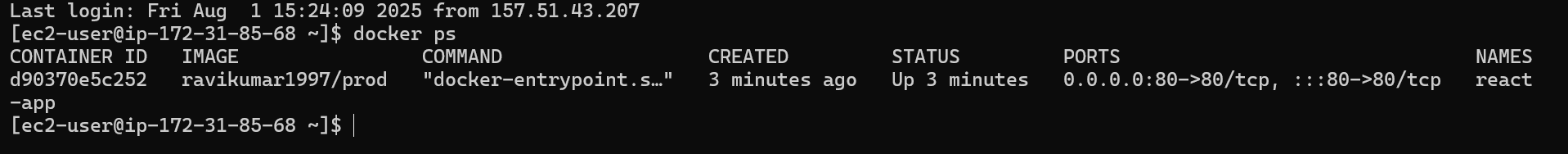
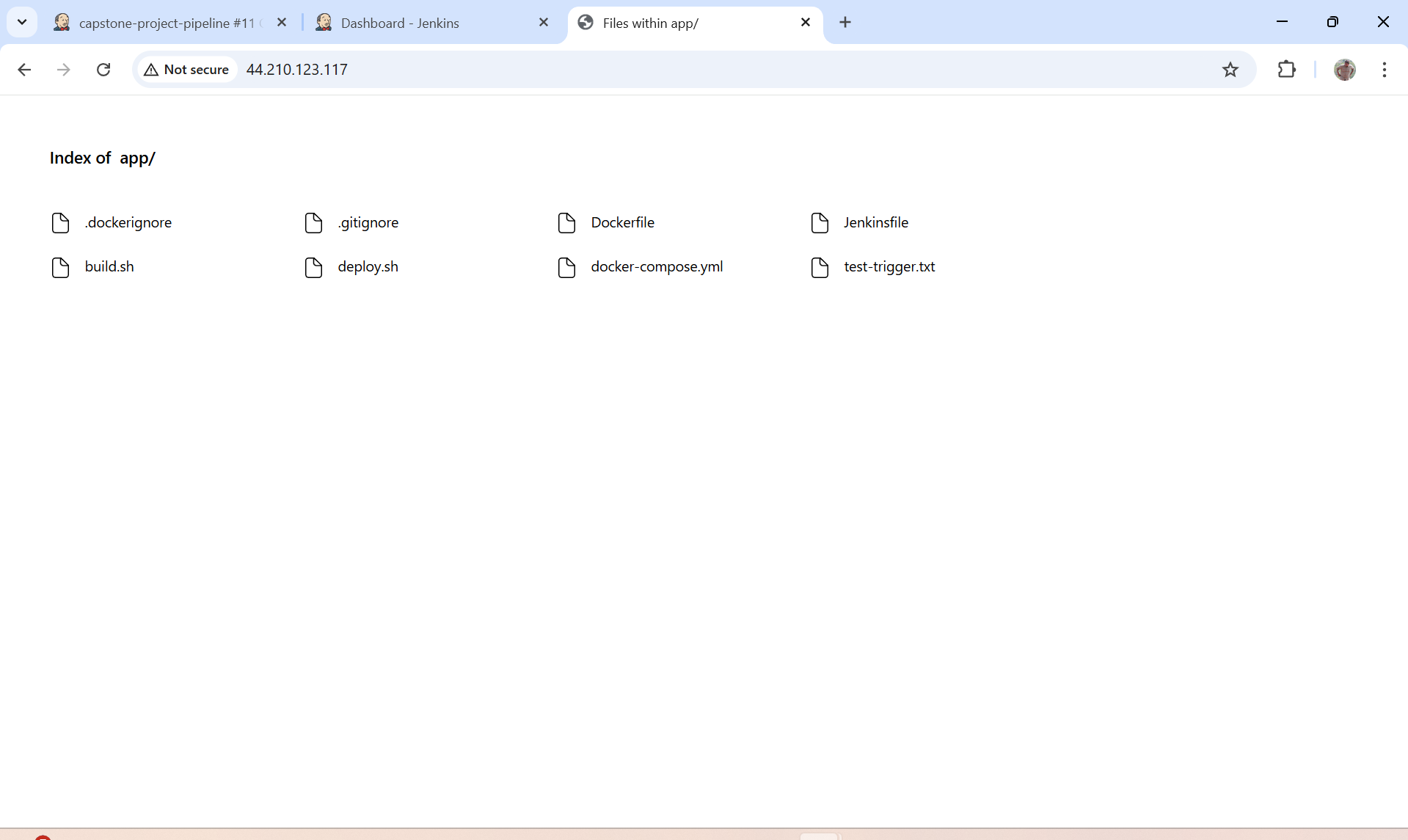








Deployed after master push too

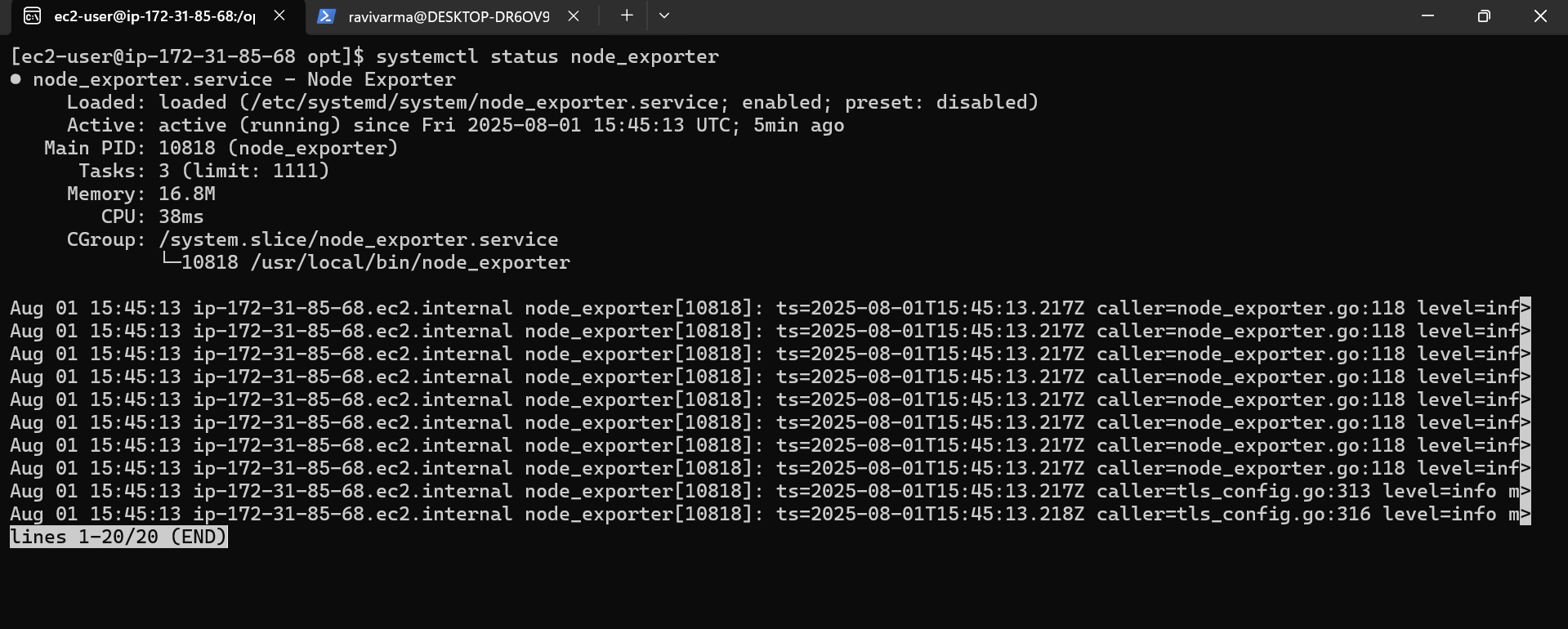
  


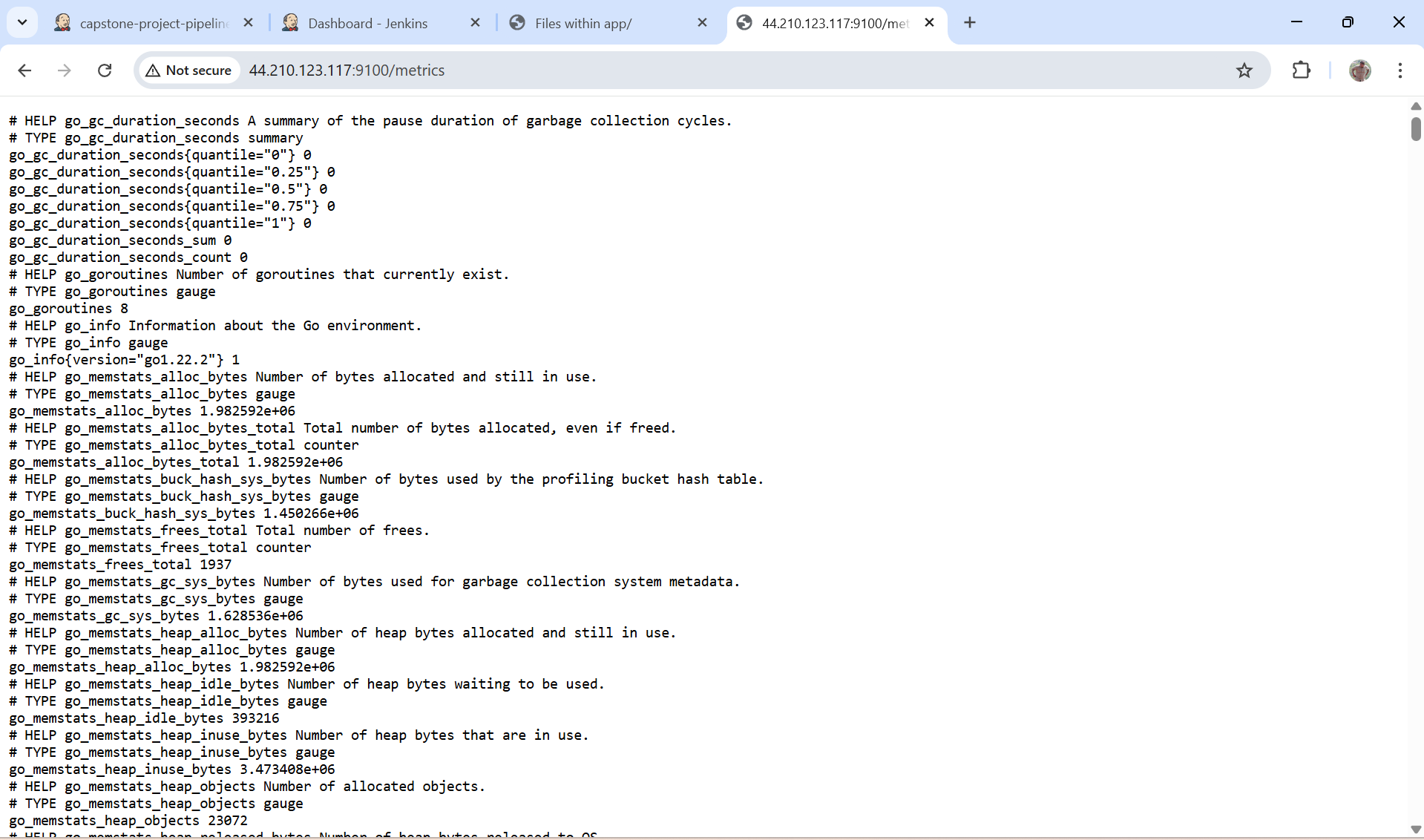
**8. Monitoring**

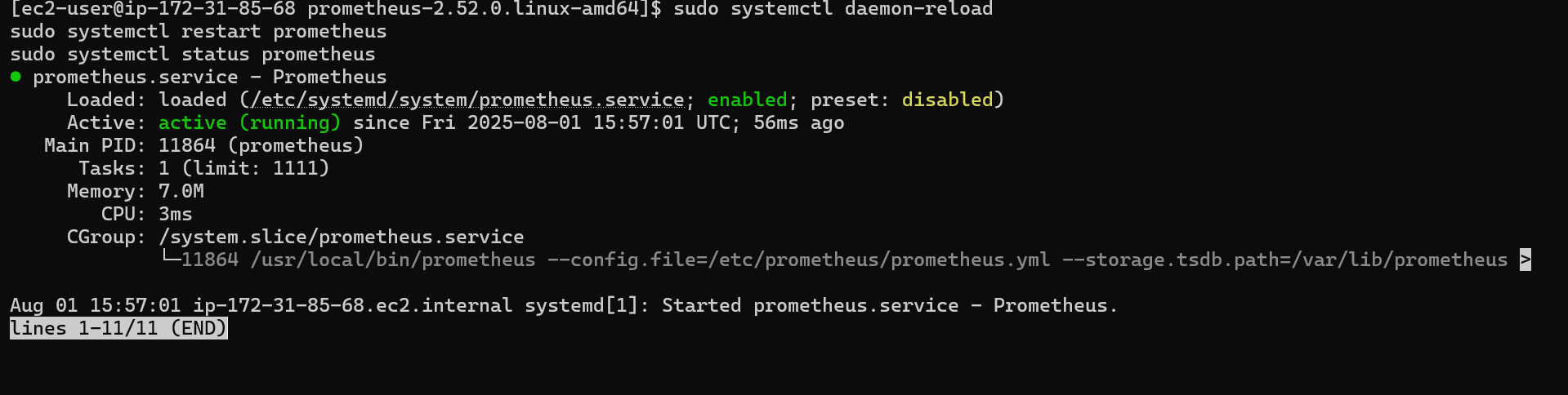
* Use an open-source tool to monitor application health.
* Setup notification if the application goes down (optional but preferred).

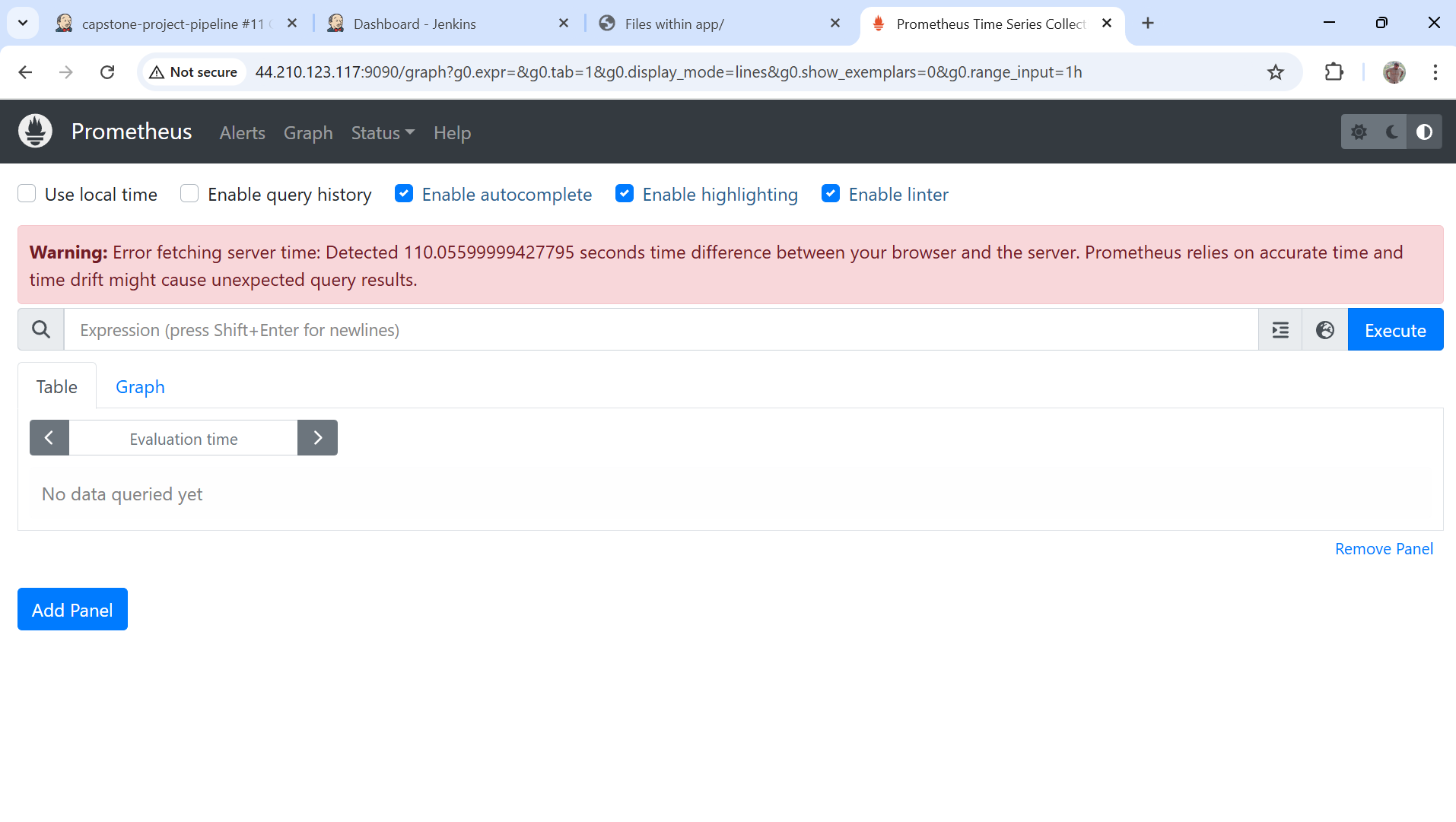
Updated SG to allow incoming traffic on ports 9100, 9090, 3000 for node exporter, Prometheus and grafana

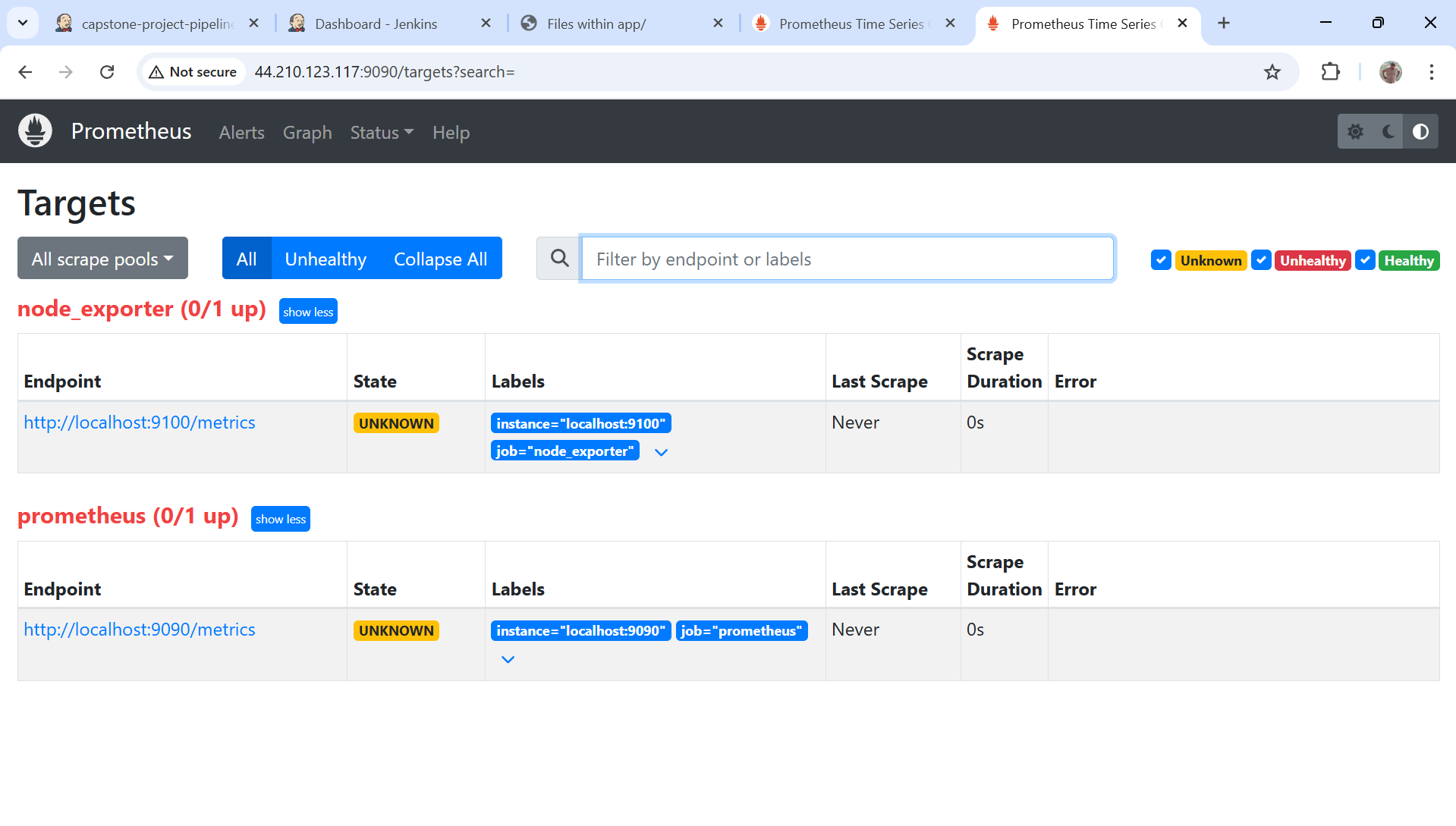
Installed node exporter

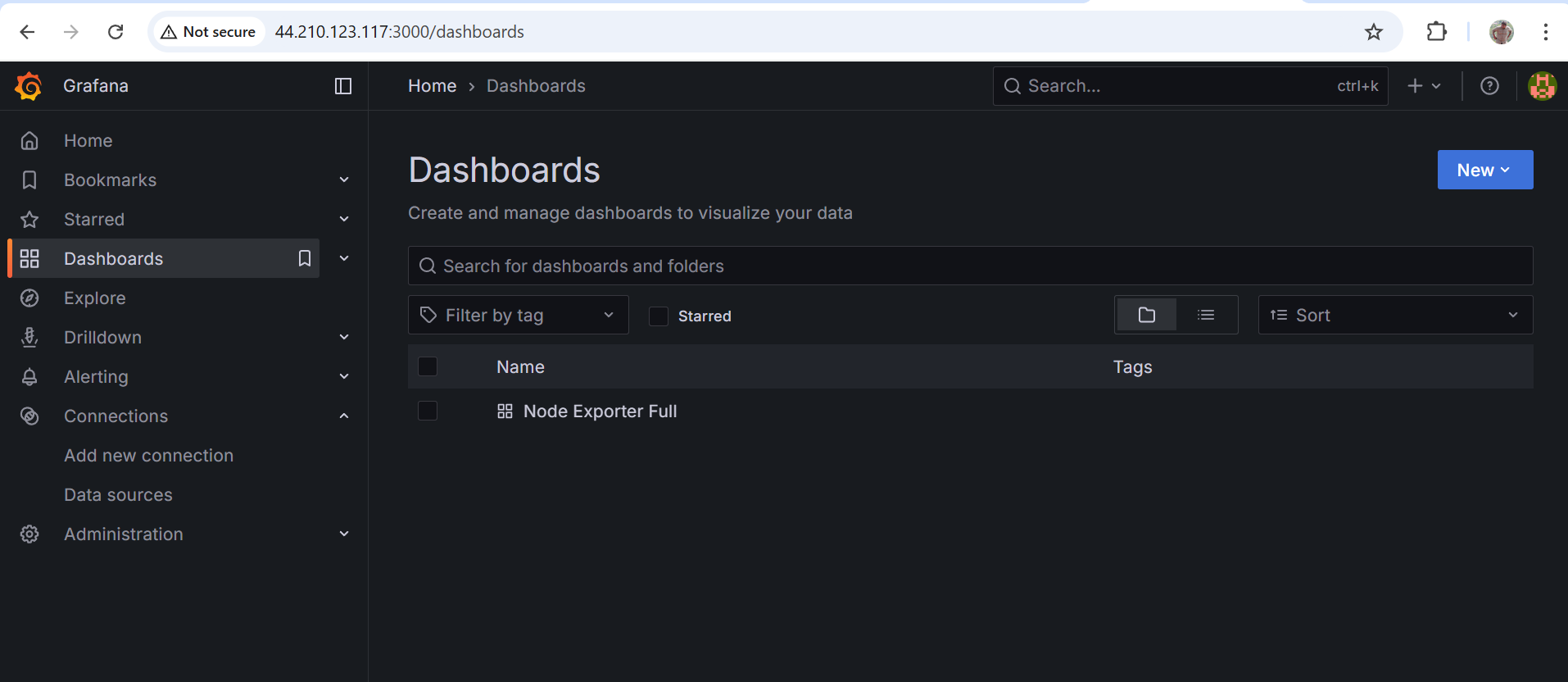


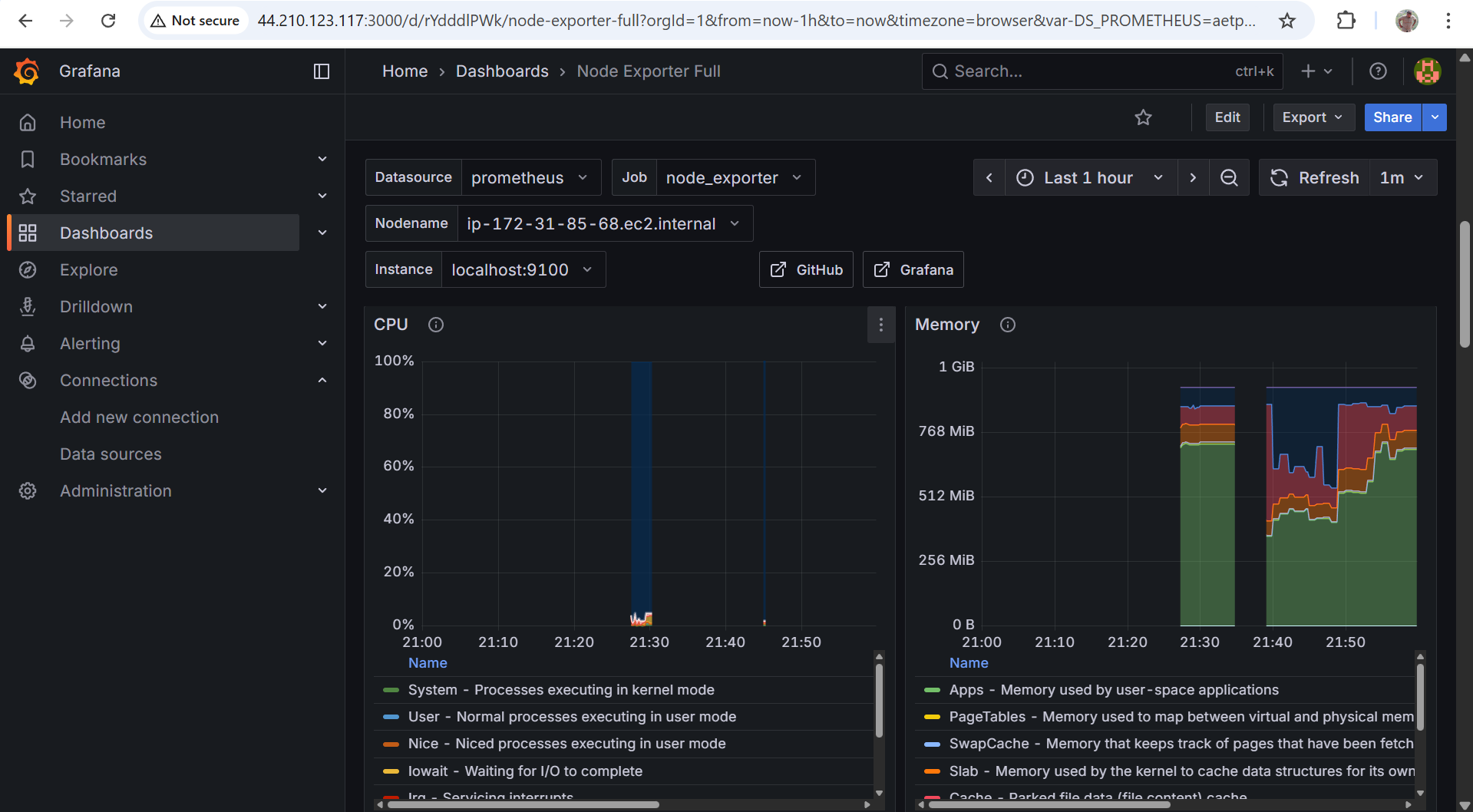












All steps are completed.  
Stopping all ec2 resources to avoid additional billing cost.

