Prerequire:

* EC2 VM that is created in terraform infrastructure
* Setup Jenkins, Docker, kubectl in EC2

Repo Jenkins Shared Libraries + CD JenkinsFile at: [hieptran228/sd4871\_devops (github.com)](https://github.com/hieptran228/sd4871_devops)

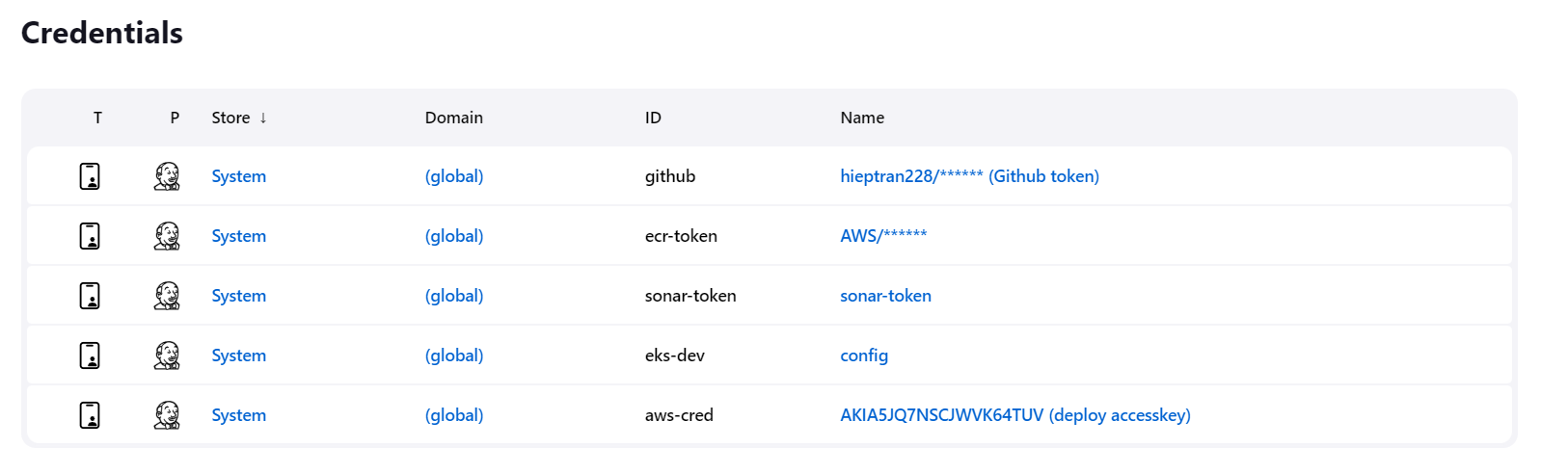
Repo sample MSA app at: [hieptran228/sd4871\_msa (github.com)](https://github.com/hieptran228/sd4871_msa)

1. Finish setup jenkins and logged in.
2. Install Jenkins plugin:

* Jenkins suggested
* Amazon ECR – help get ECR password
* Docker Pipeline
* Pipeline Utility Steps
* Kubernetes

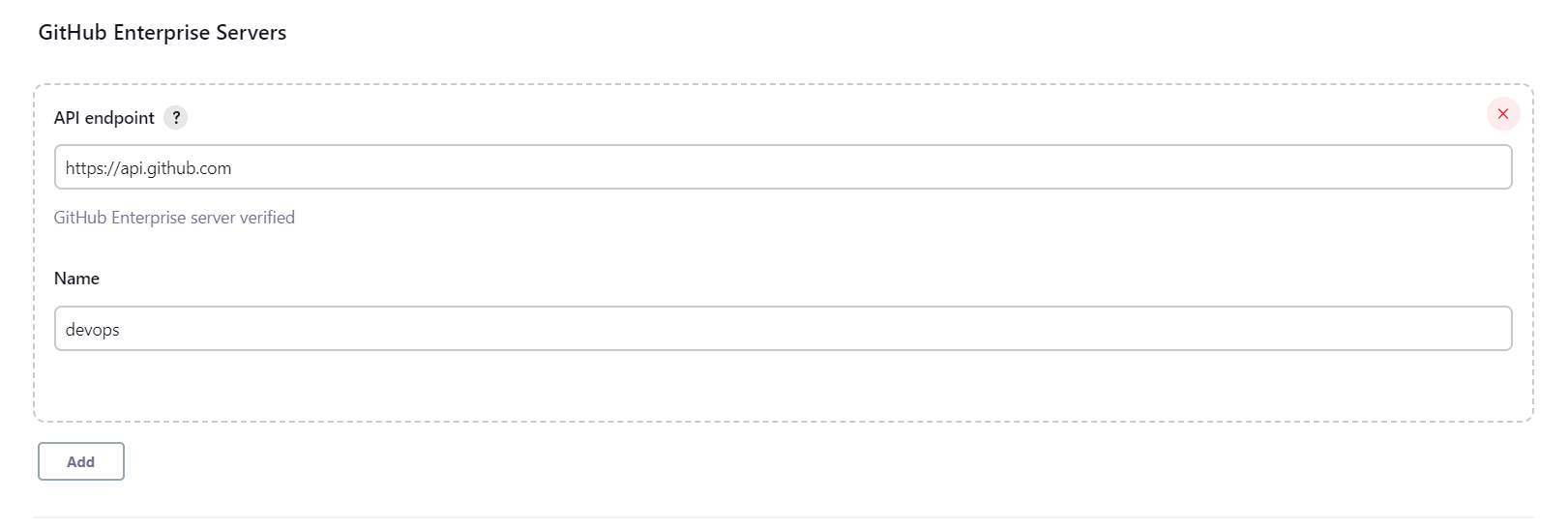
1. Add credentials:

* GitHub with Kind Username with password (ID name: github)
* kubeconig with Kind Secret file (ID name: eks-dev)
* AWS with kind AWS Credential (ID name: aws-cred)

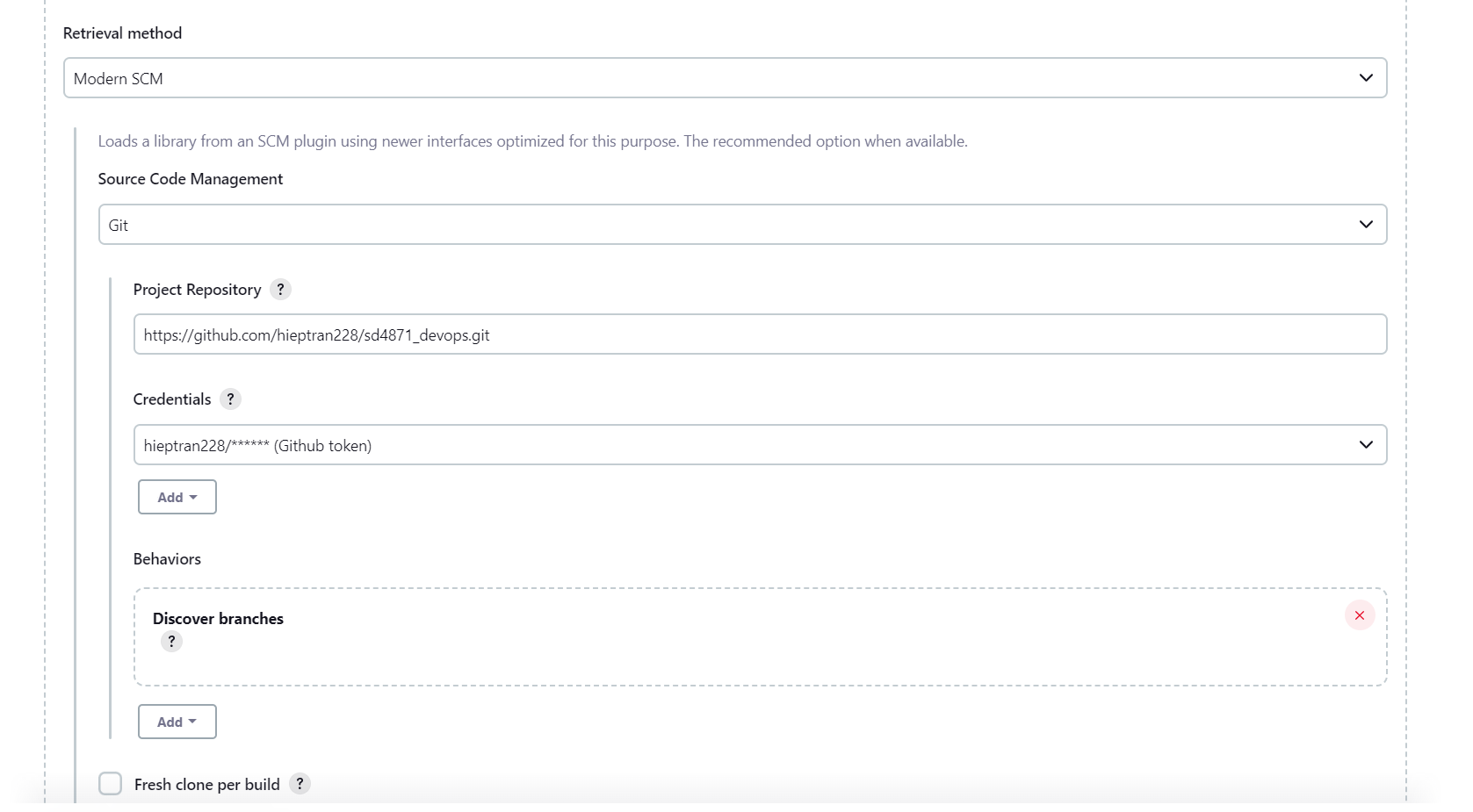
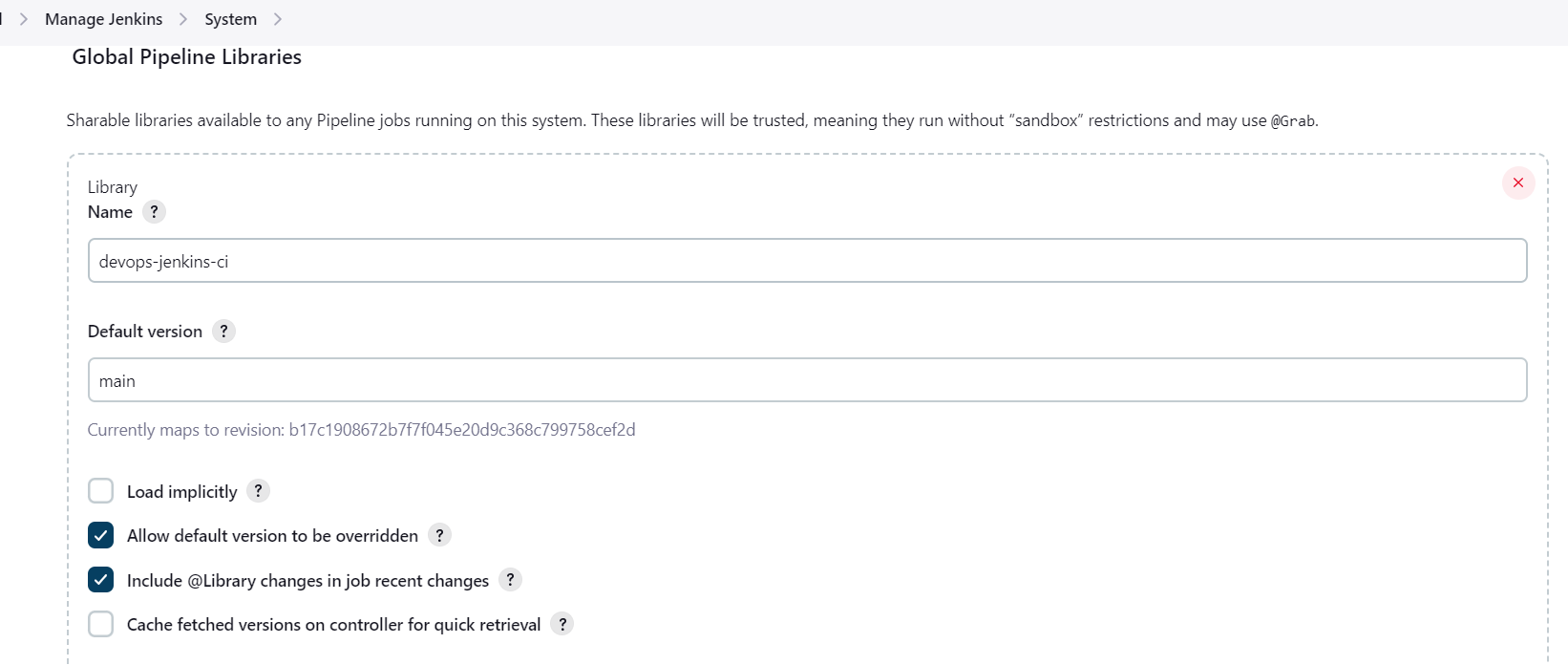


1. Manage Jenkins -> System

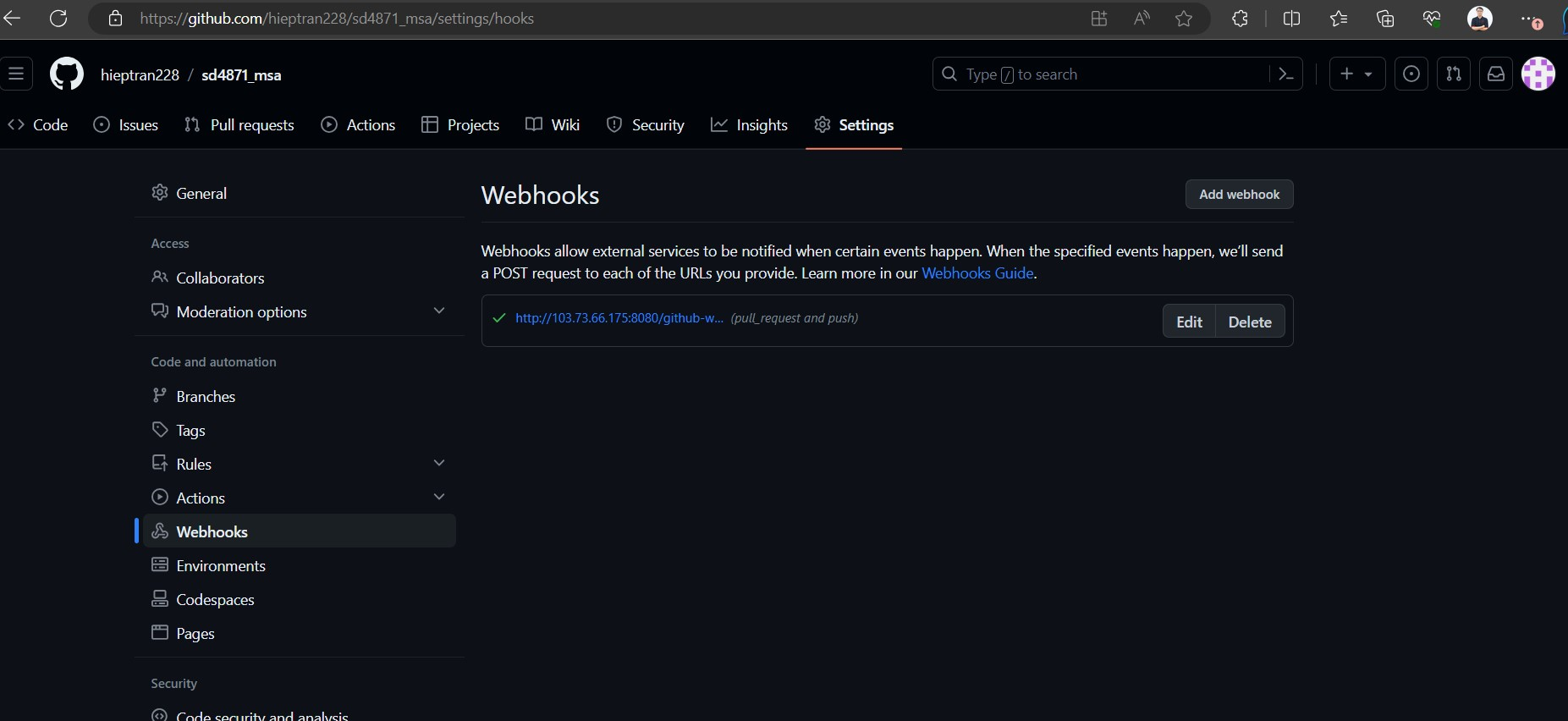
* GitHub Enterprise Servers
  + API endpoint: [https://api.github.com](https://api.github.com/)
  + Name: devops



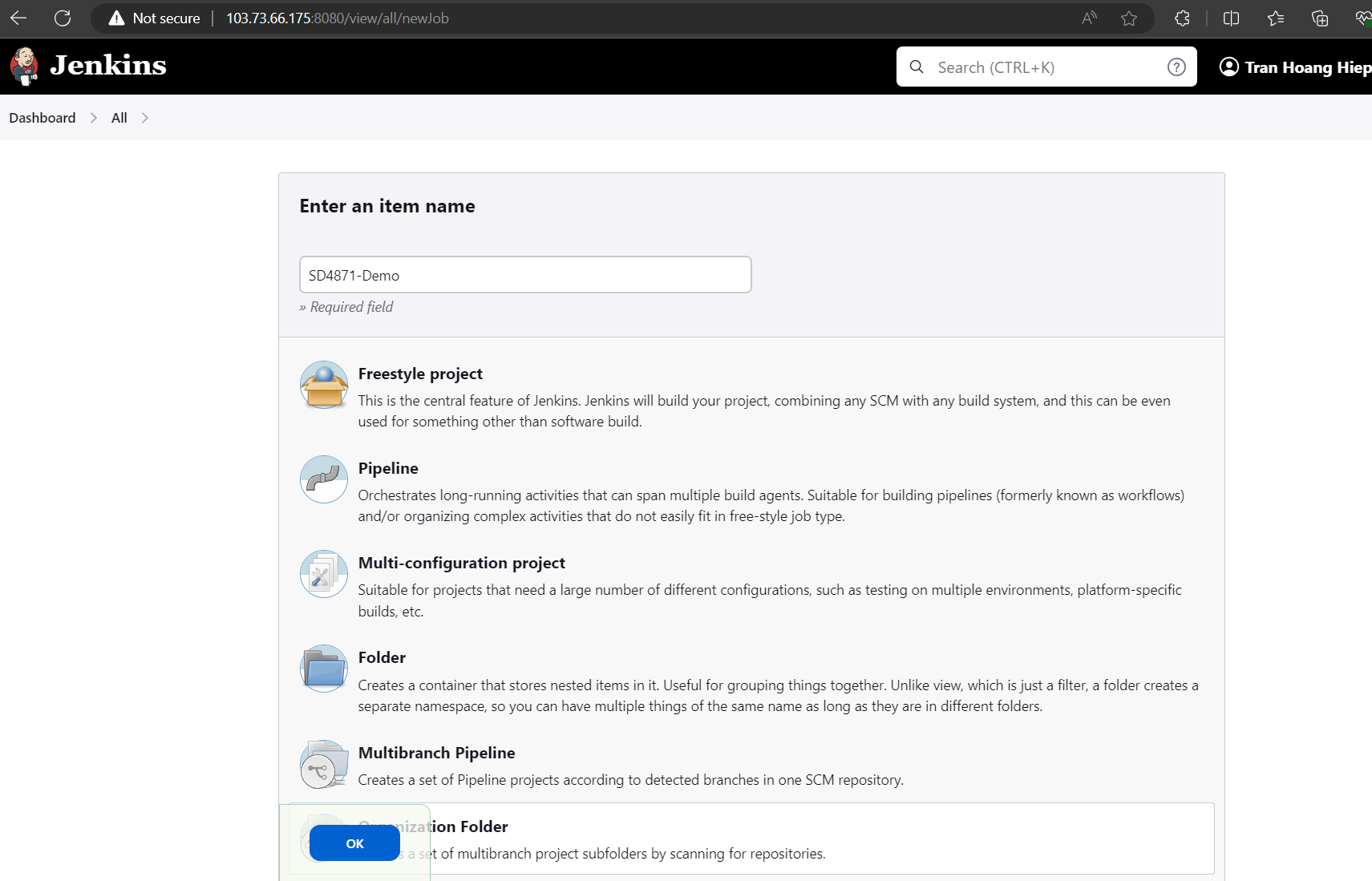
* Global Pipeline Libraries
  + Name: devops-jenkins-ci
  + Default version: jenkins
  + Allow default version to be overridden
  + Include @Library changes in job recent changes
  + Retrieval method: Modern SCM
    - Source Code Management: Git
    - Project Repository: https://github.com/hieptran228/sd4871\_devops.git
    - Credentials: github



1. Setup github webhook for MSA and Devops repo point to jenkins server (jenkins-endpoint/github-webhook) for pull request and push



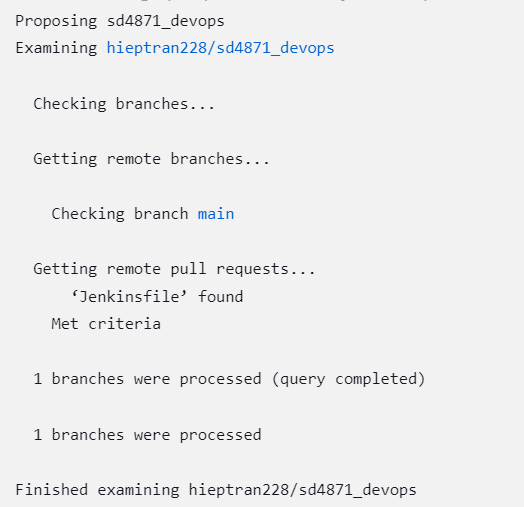
1. Create jenkins job (Organization Folder) -> OK

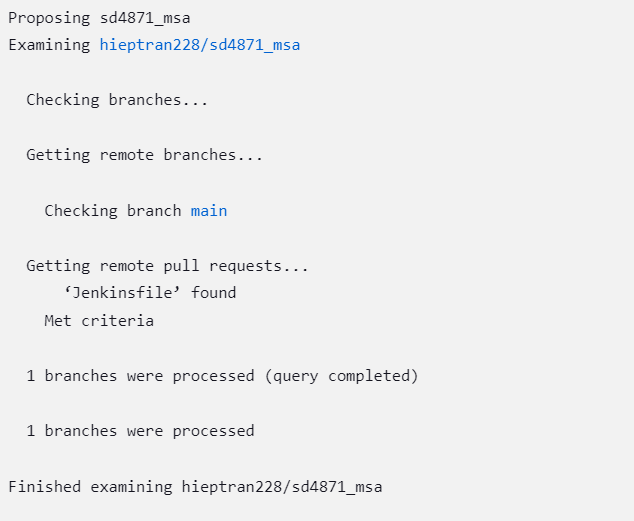


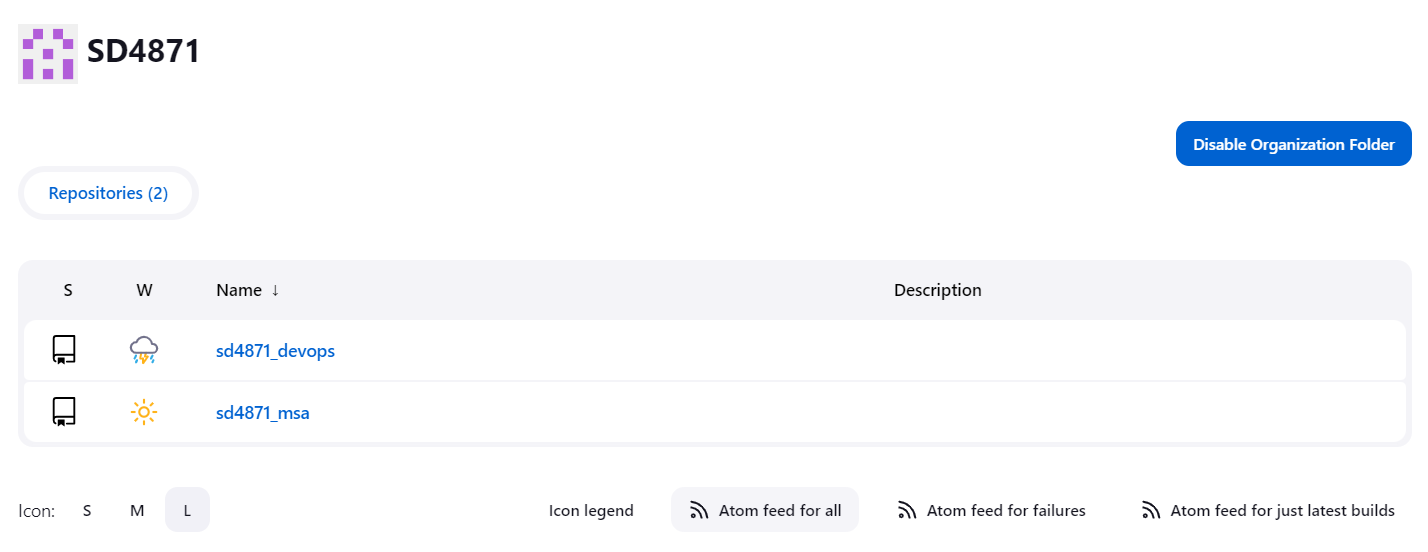
1. Projects section. Add:

* Credentials: hieptran228/\*\*\*\*\*\* (Github token)
* Owner: hieptran228

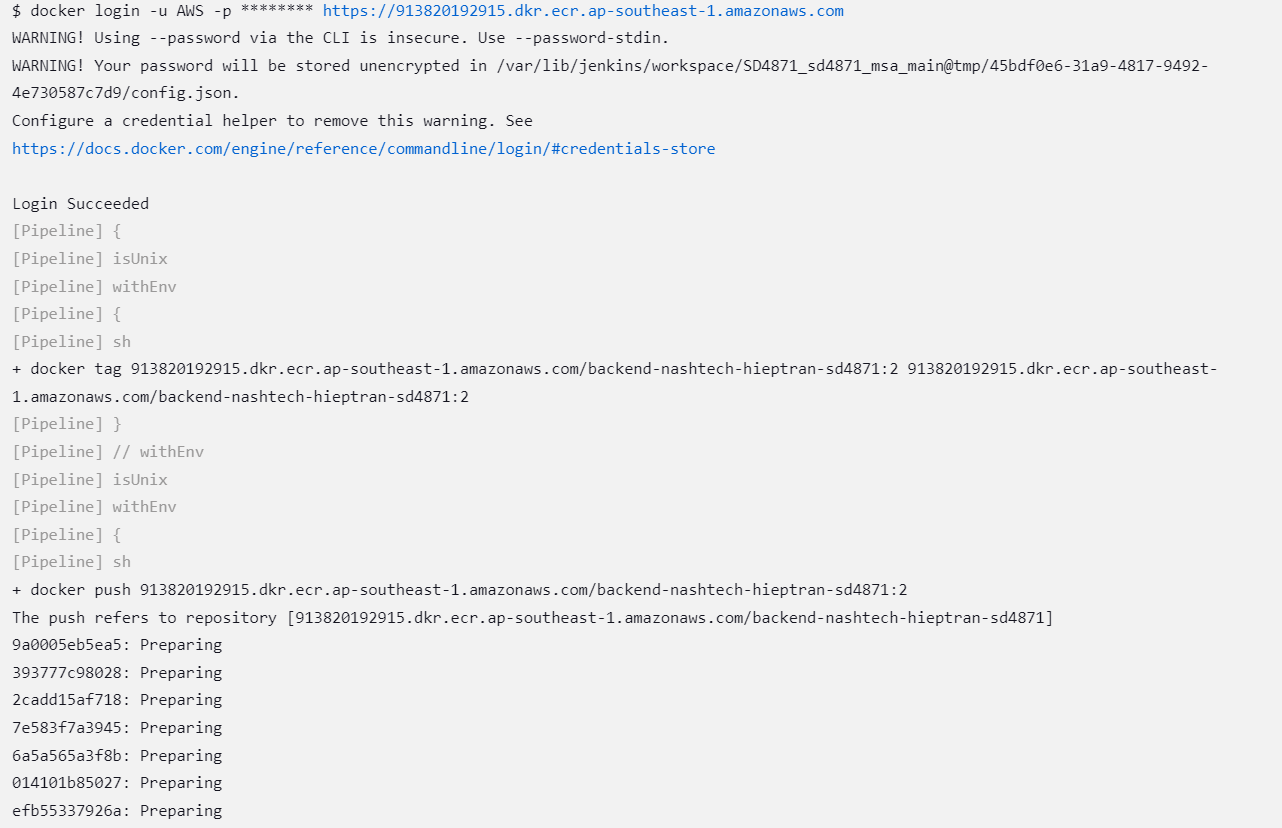
Then Save. Jenkins will scan. Result:

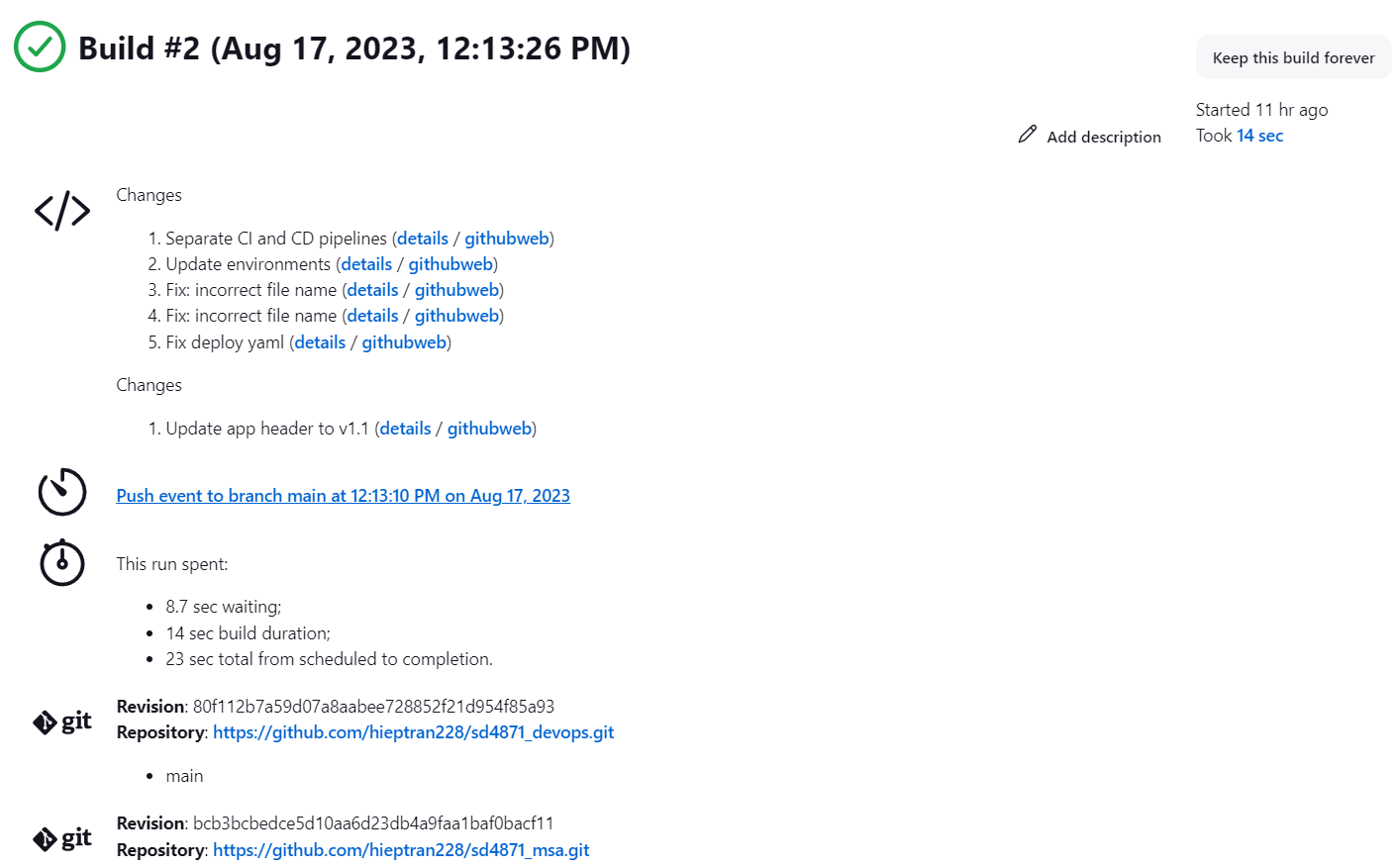




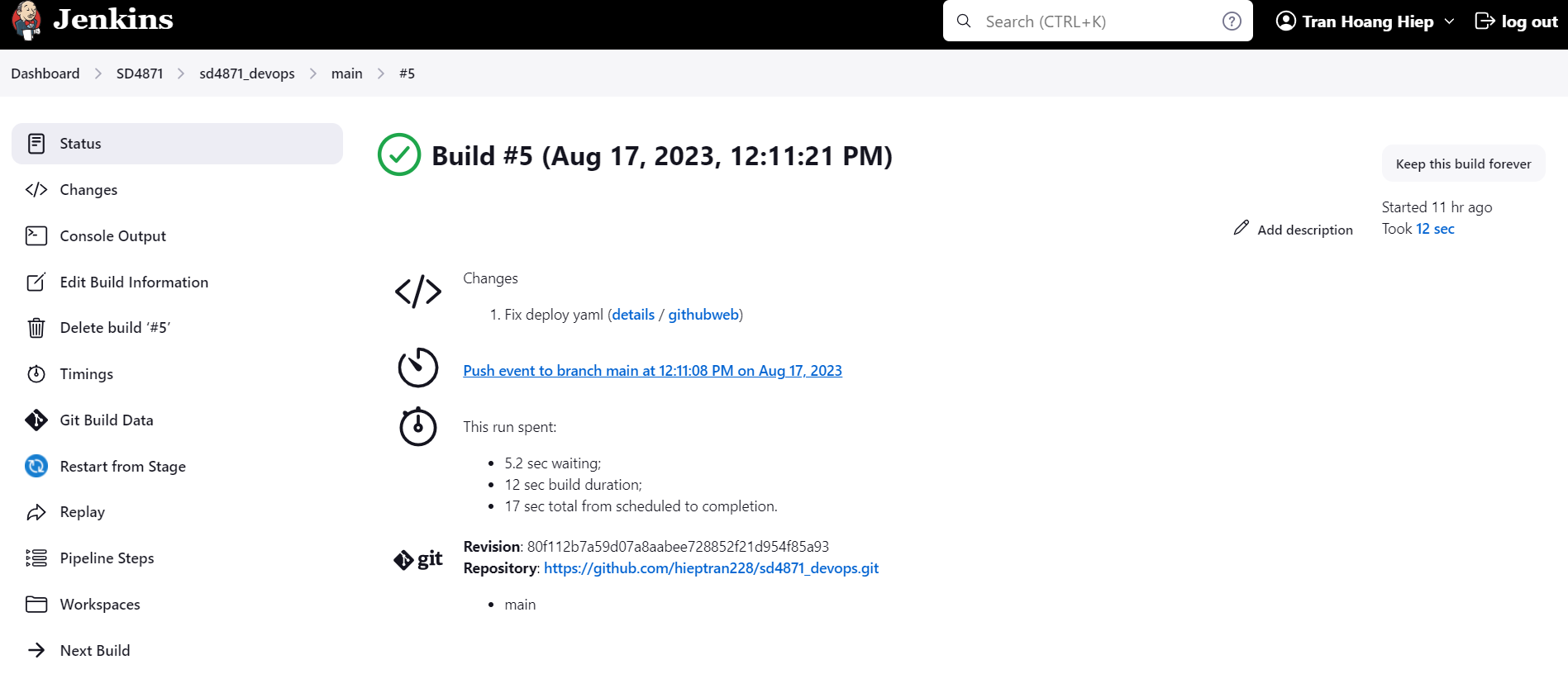


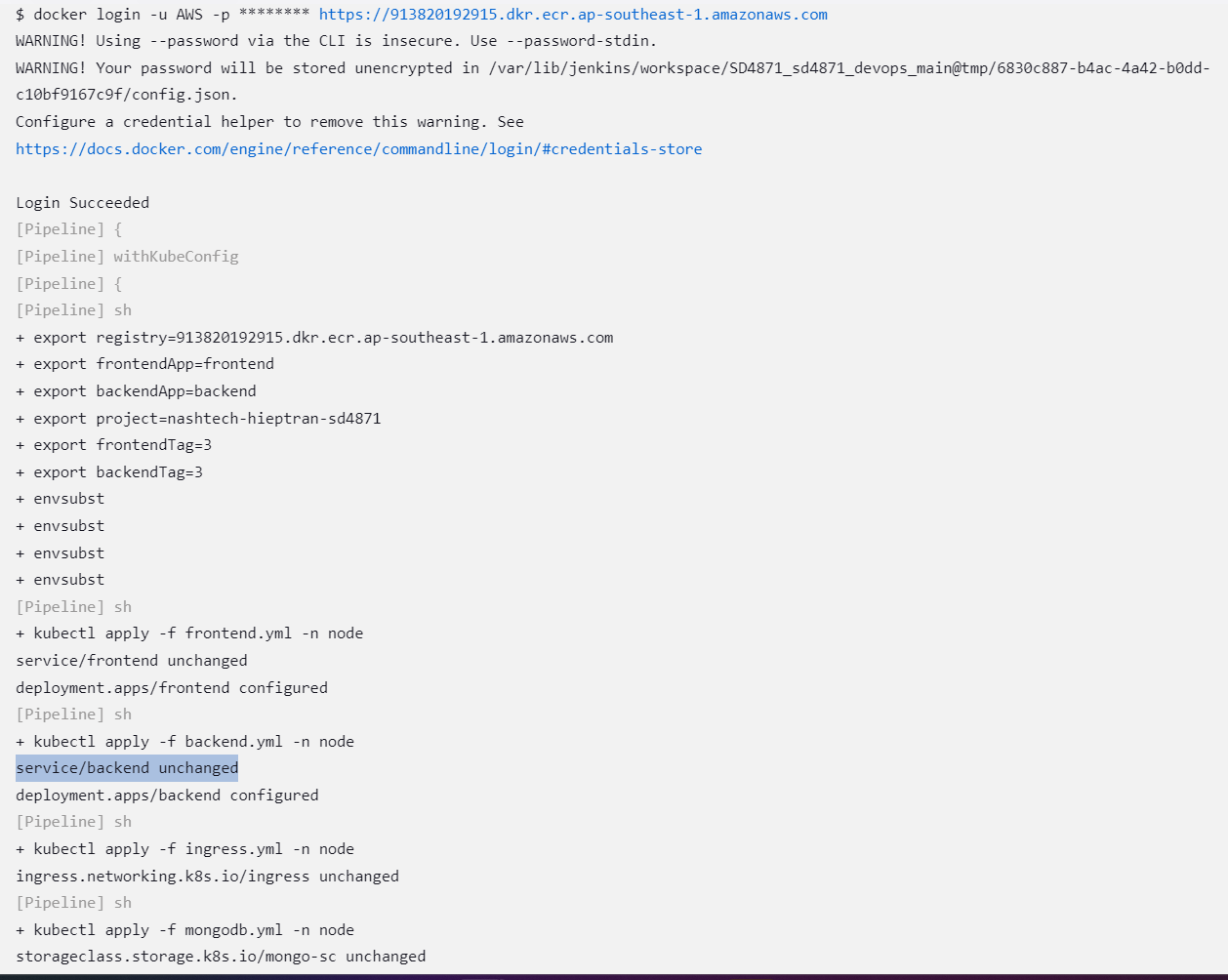
1. When a change to the MSA app is pushed to github. Jenkins pipeline triggered. The MSA app is pushed successful to repo.



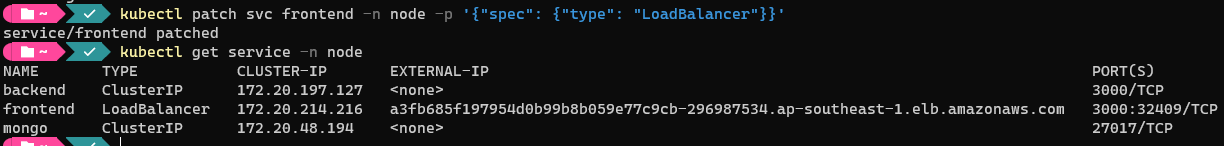


1. Same for devops repo. Changes are pushed to github. Then deployment using deploy pipeline is triggered successful.

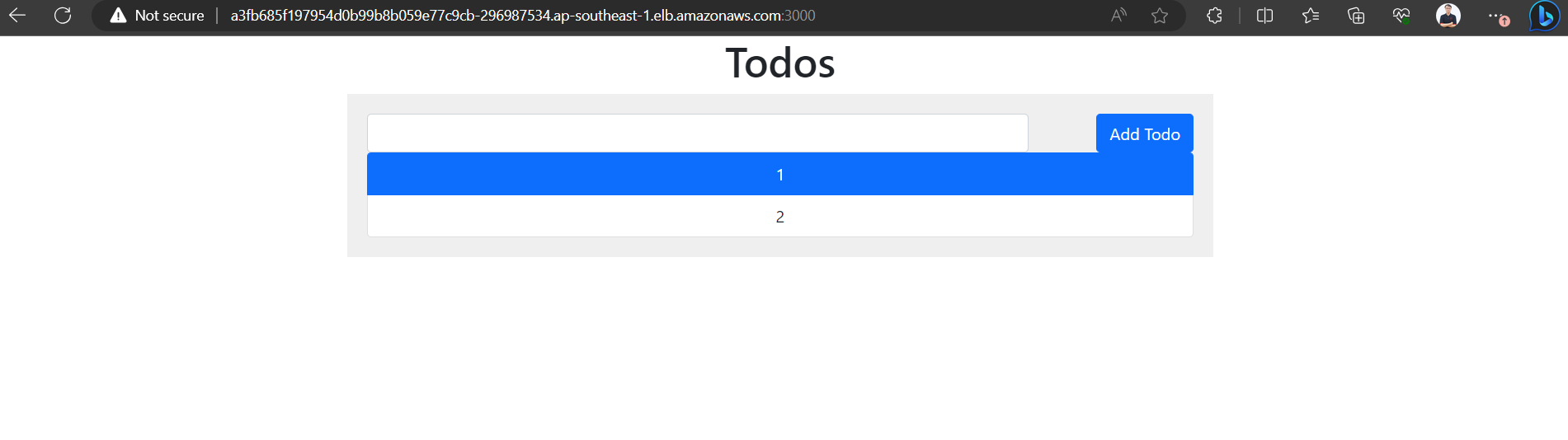




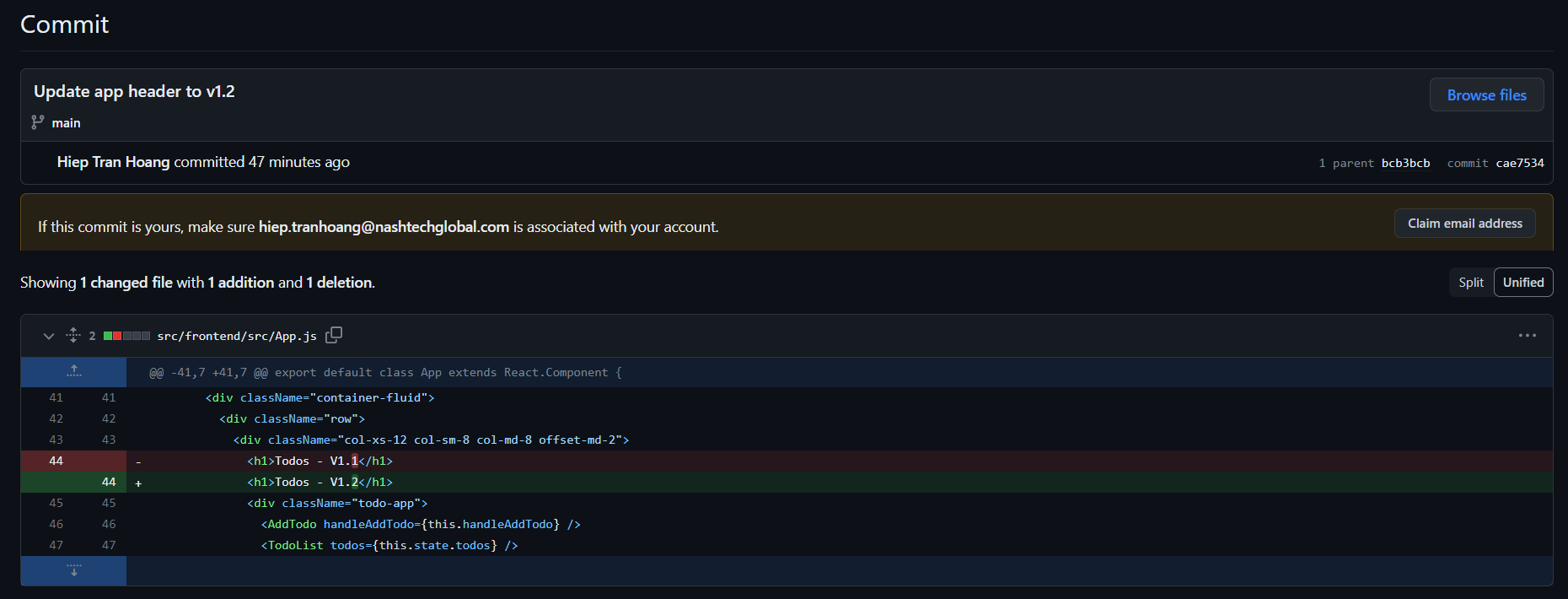
1. Run “kubectl patch svc frontend -n node -p '{"spec": {"type": "LoadBalancer"}}'” then get service external url. Wait for a few minutes for DNS get updated.

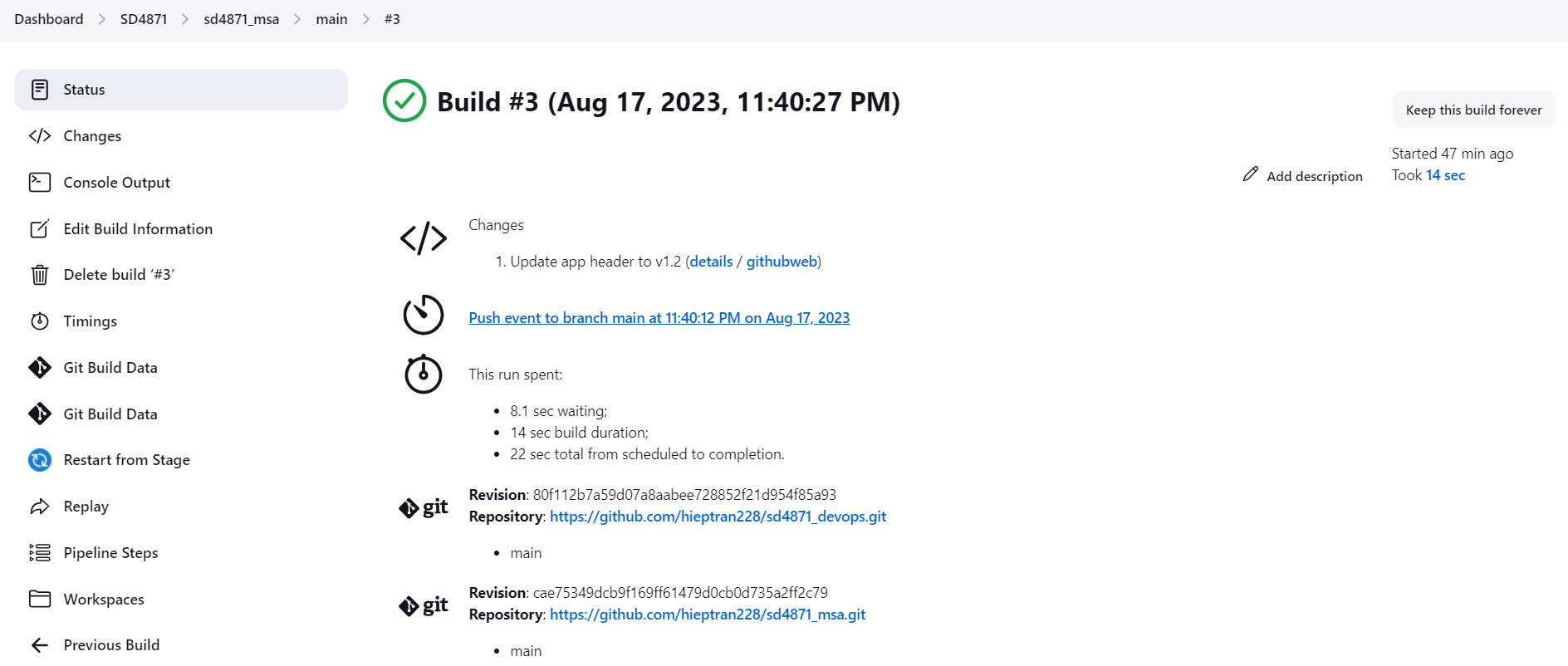


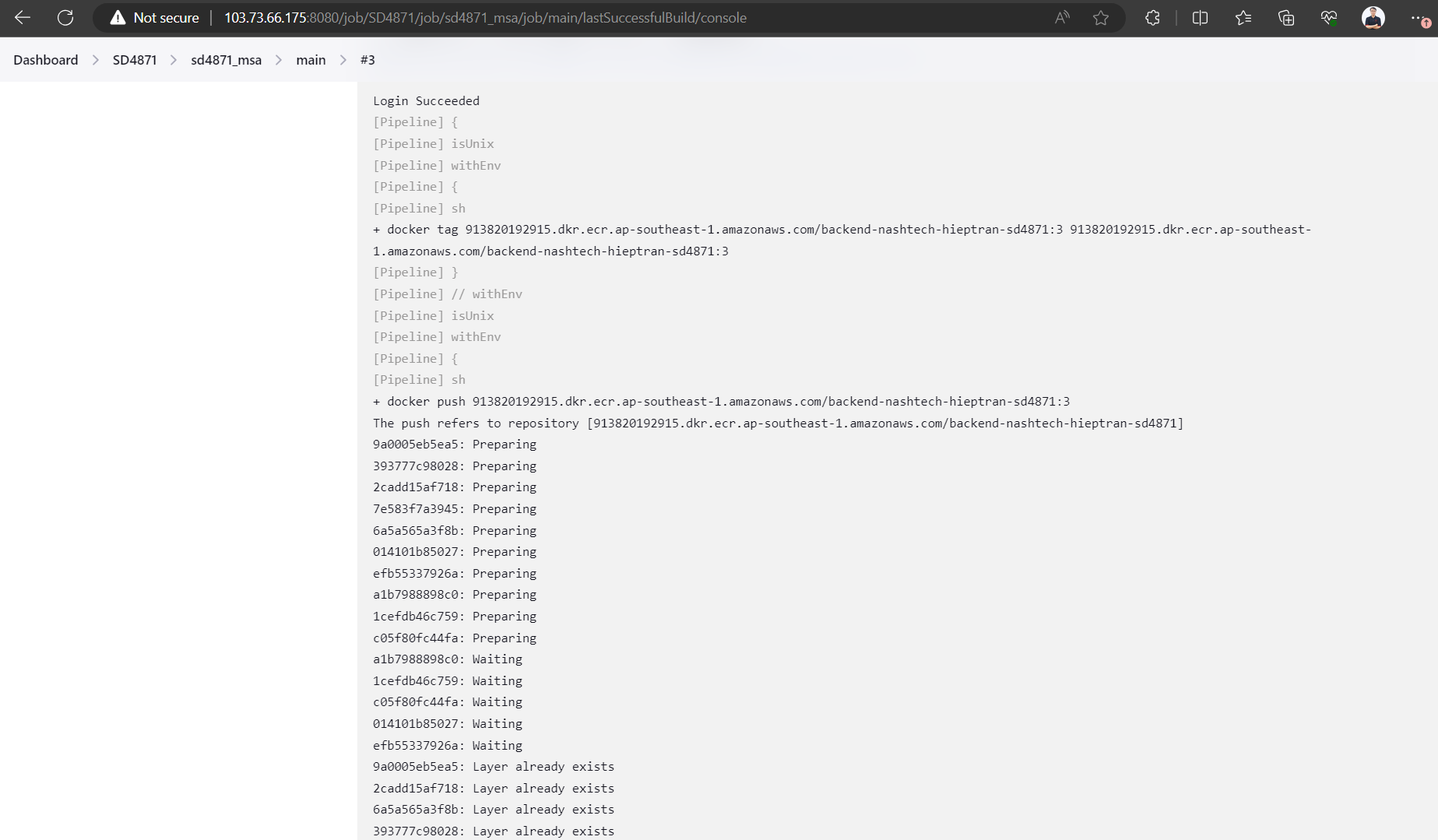
1. Deploy was successful!



12. Update frontend app to V1.2 in header.

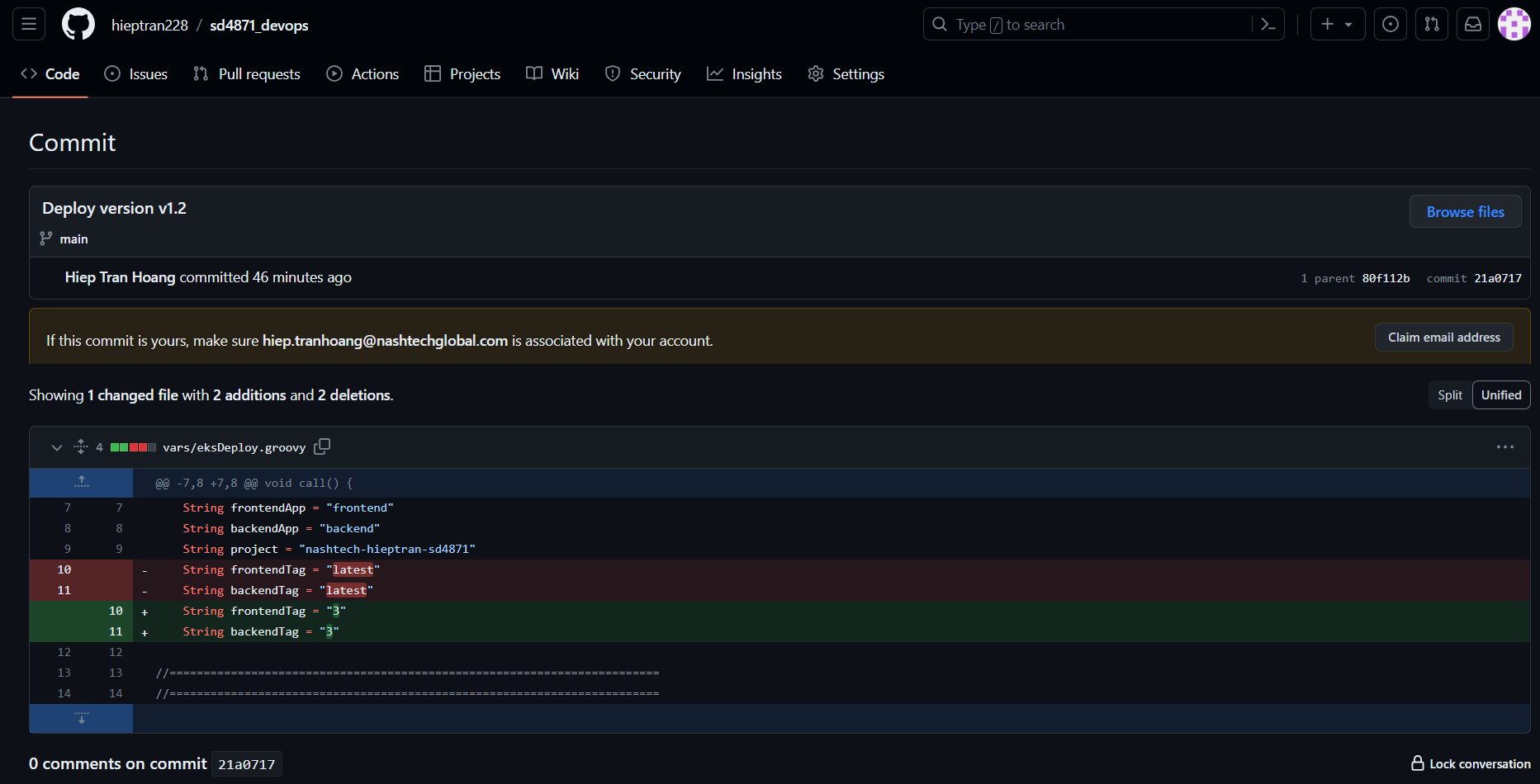


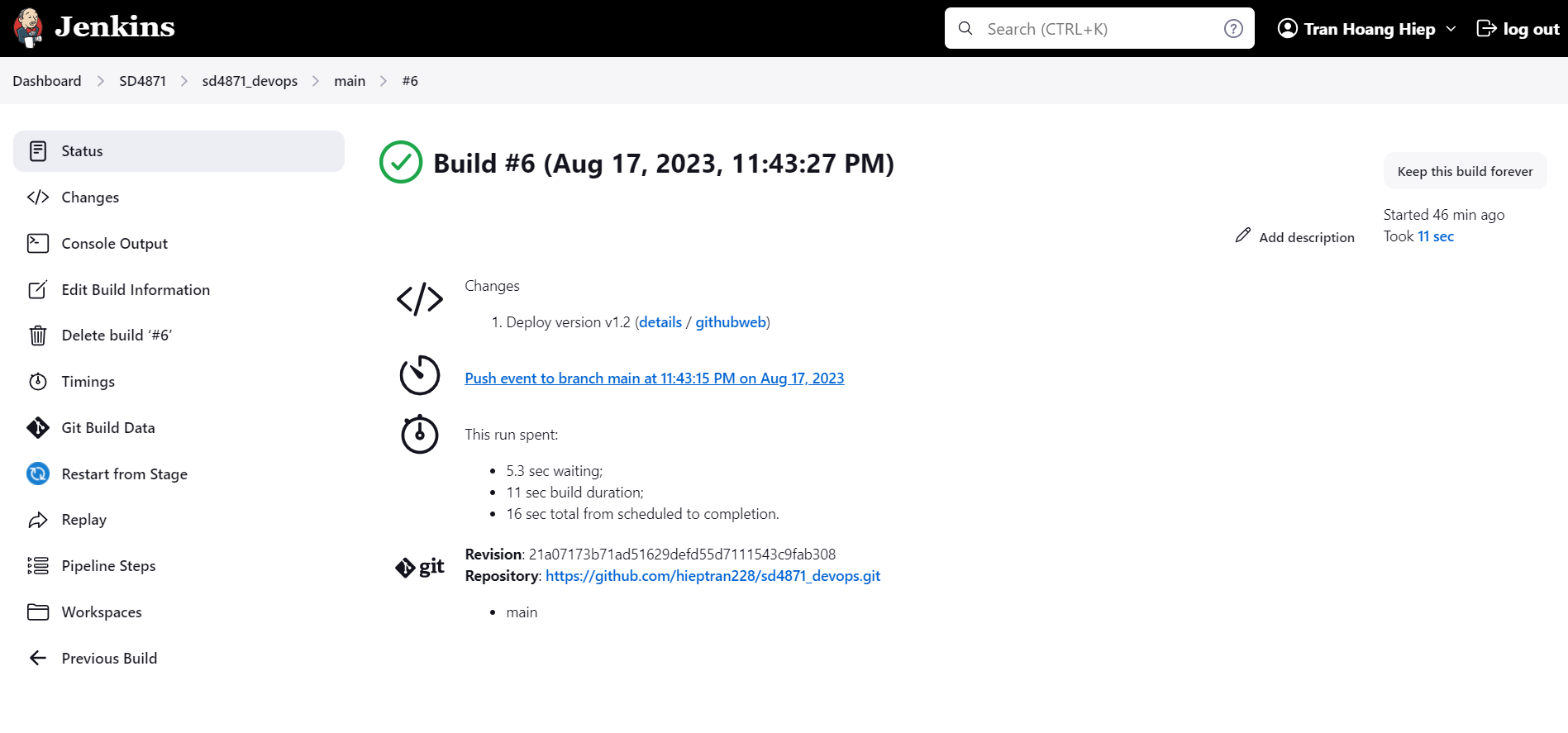


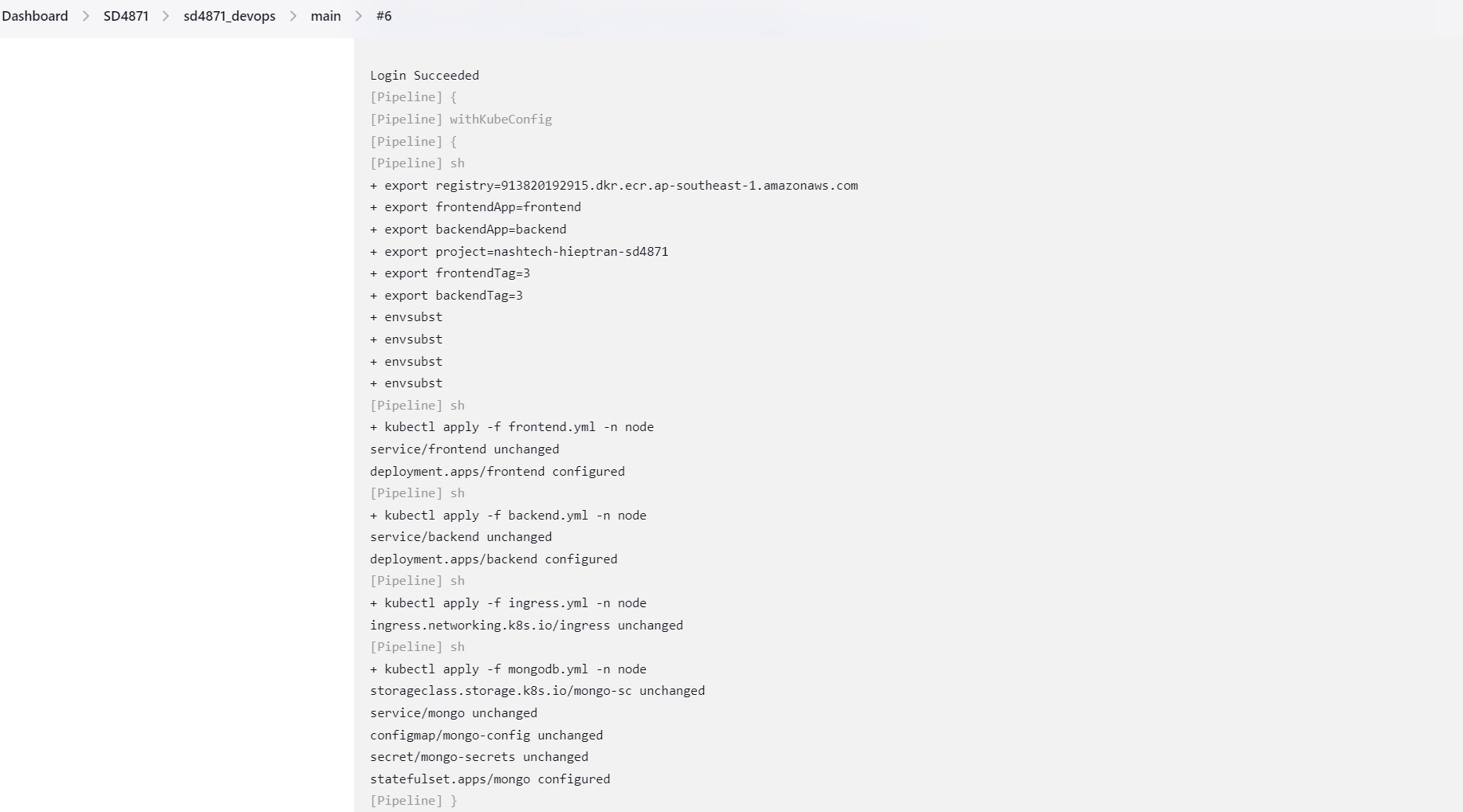


12. App pushed to ECR with tag = 3.

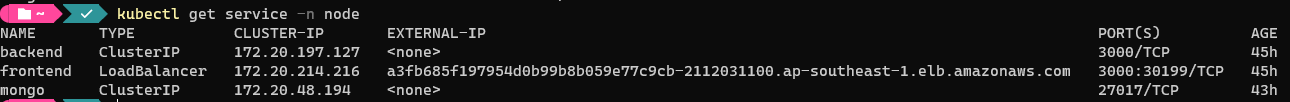
13. Change devops repo to deploy image tag = 3.







Get another externet ip.



Wait a little bit. And the frontend is deployed successful

