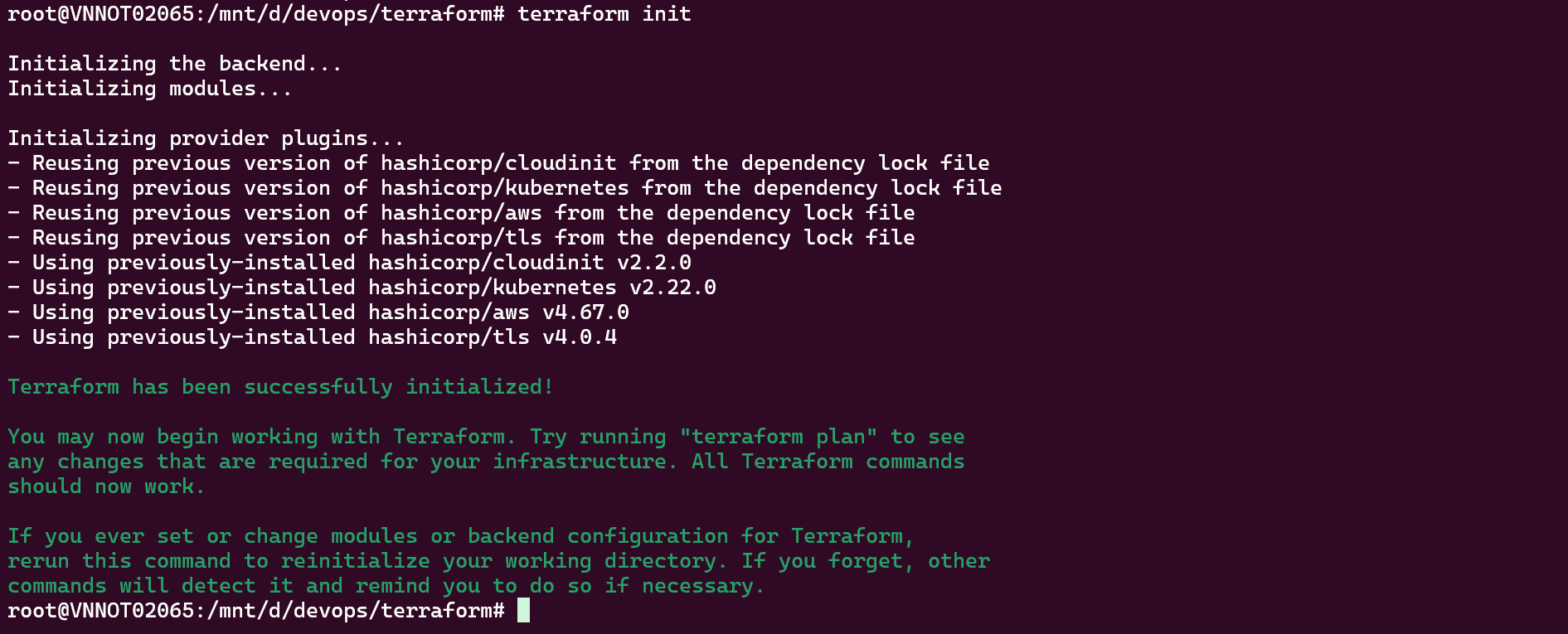
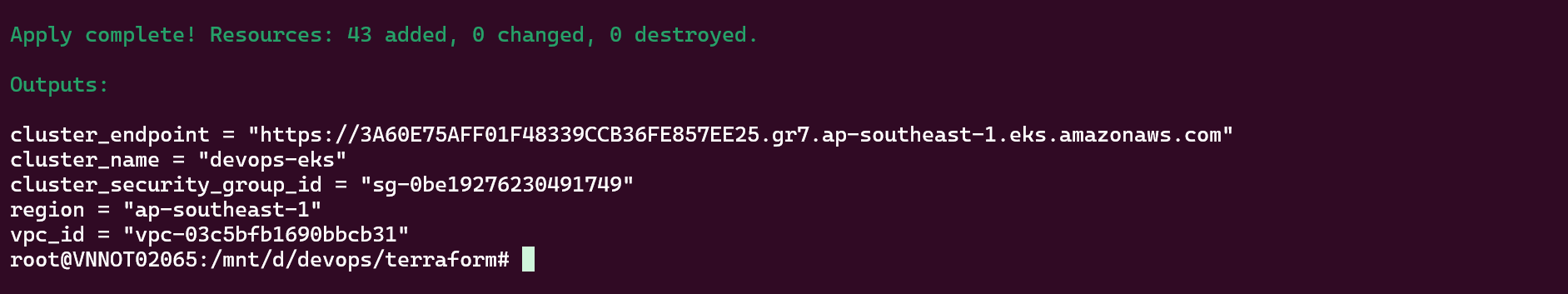
I. Provision resource using terraform

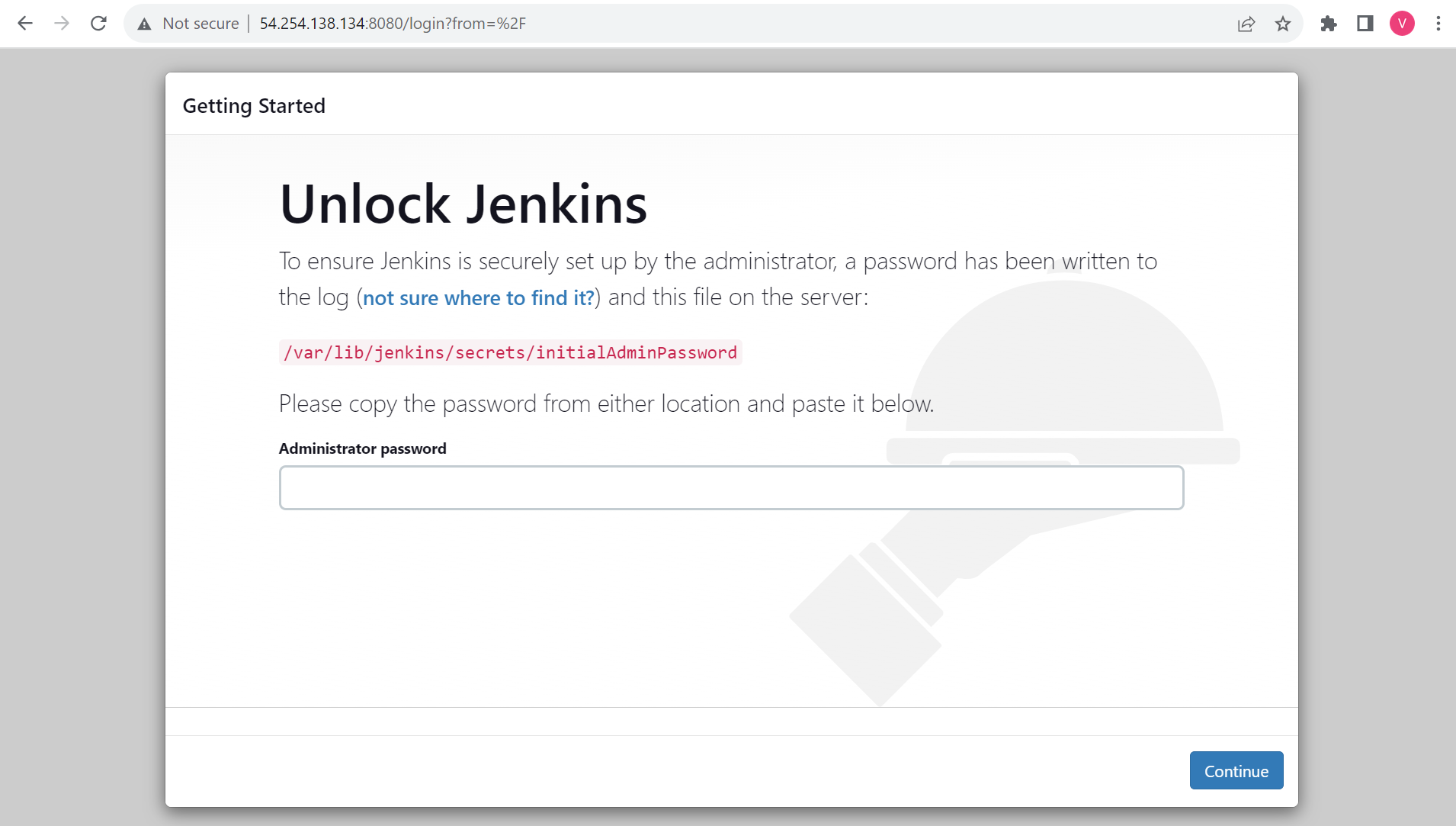
Step 1: terraform init

Step 2: terraform apply

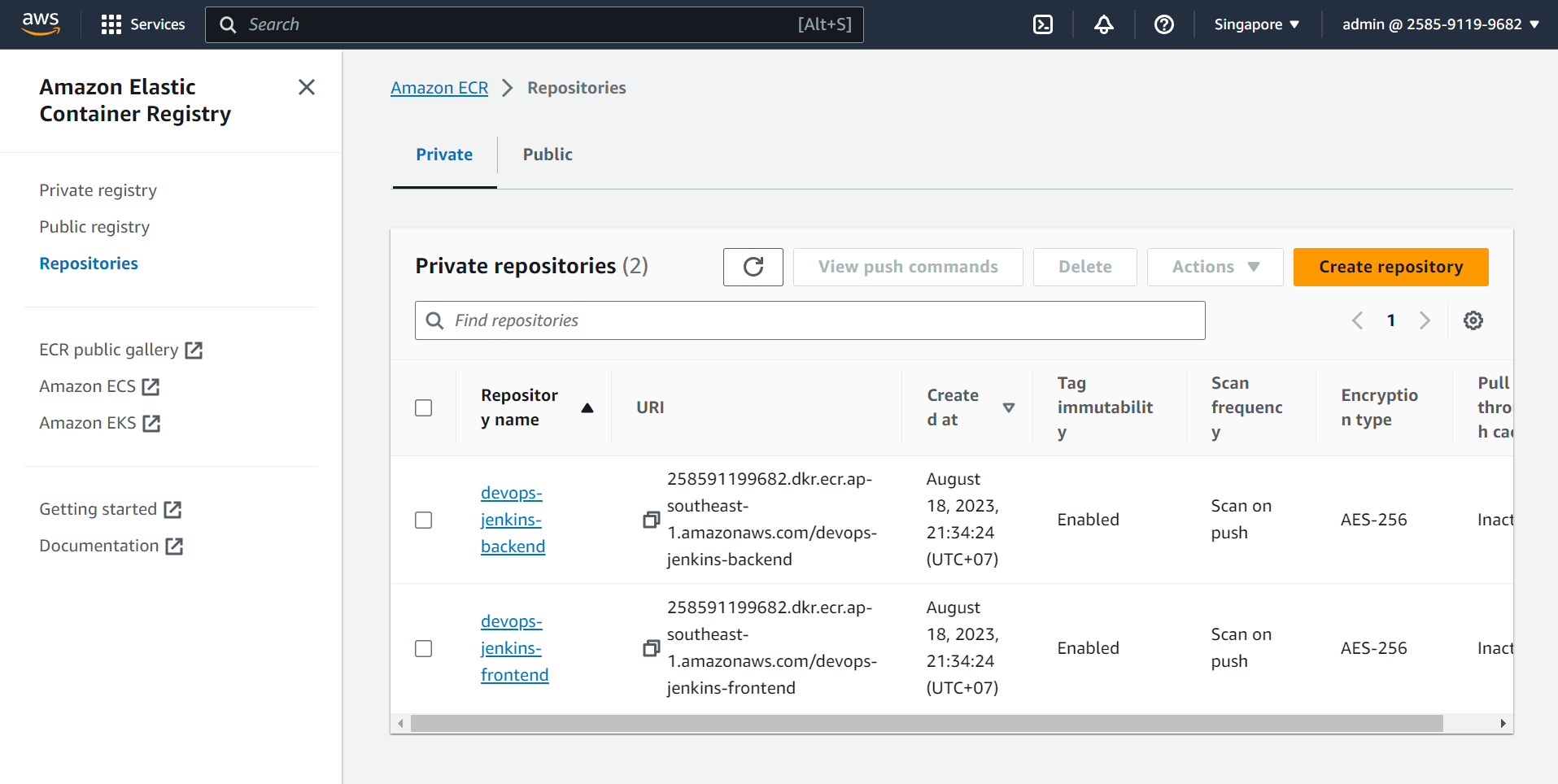


Step 3: Check resources

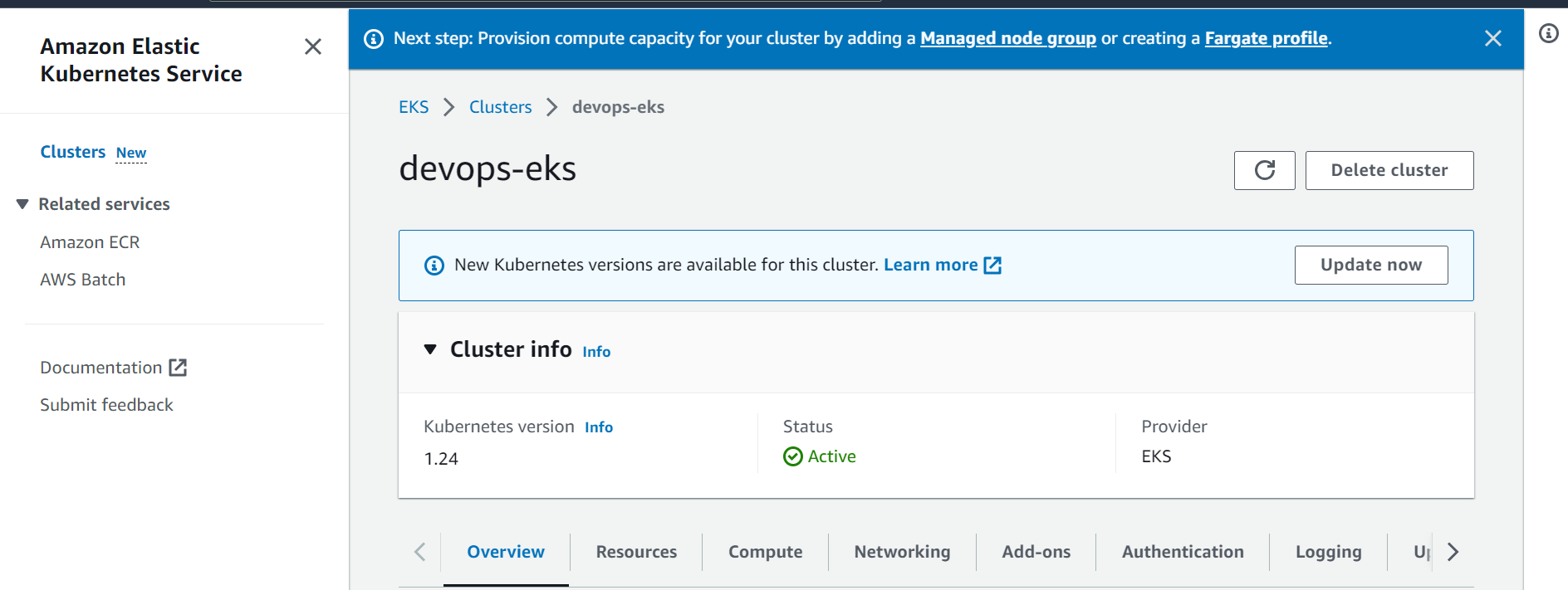
EC2 installed Jenkins



ECR repositories

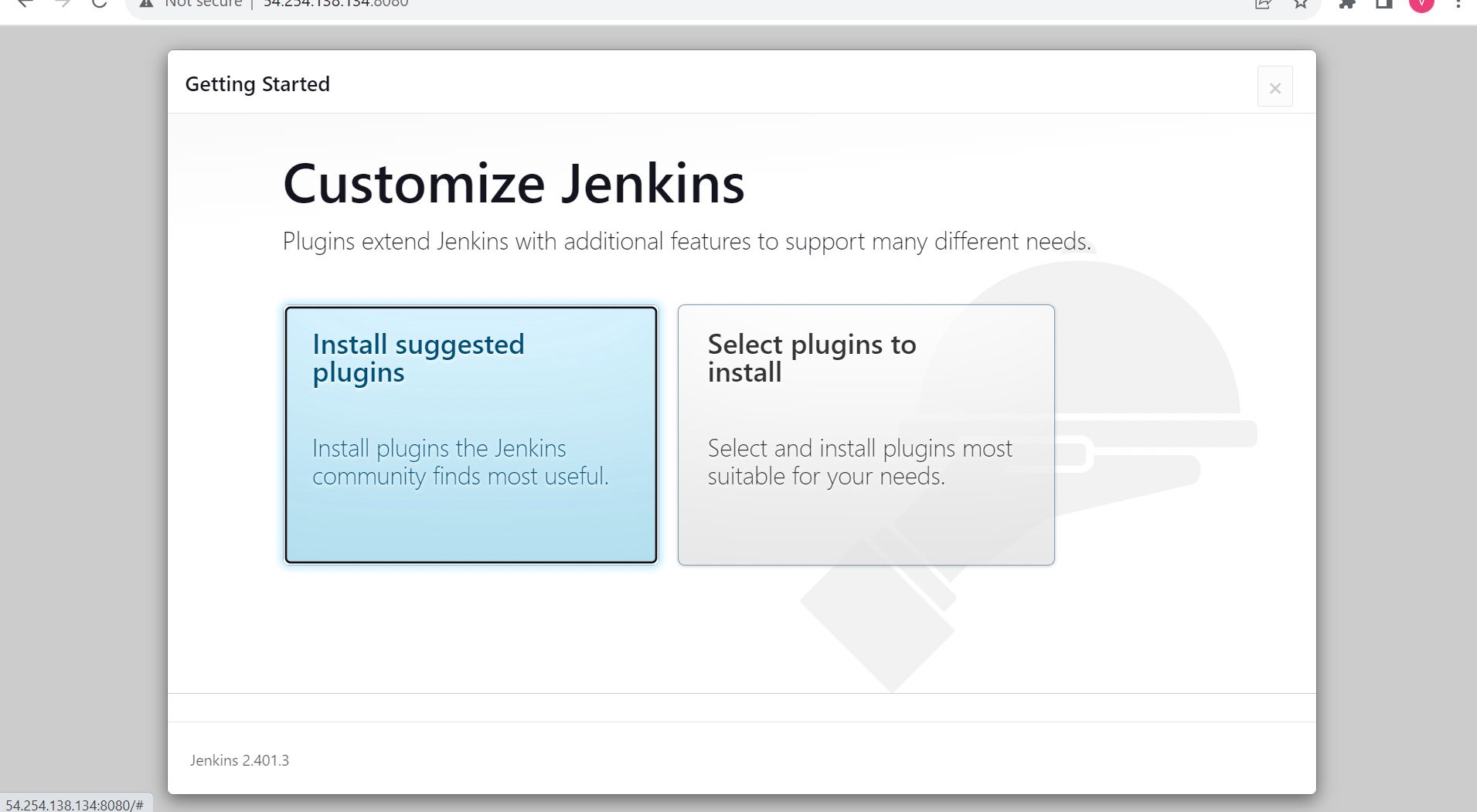


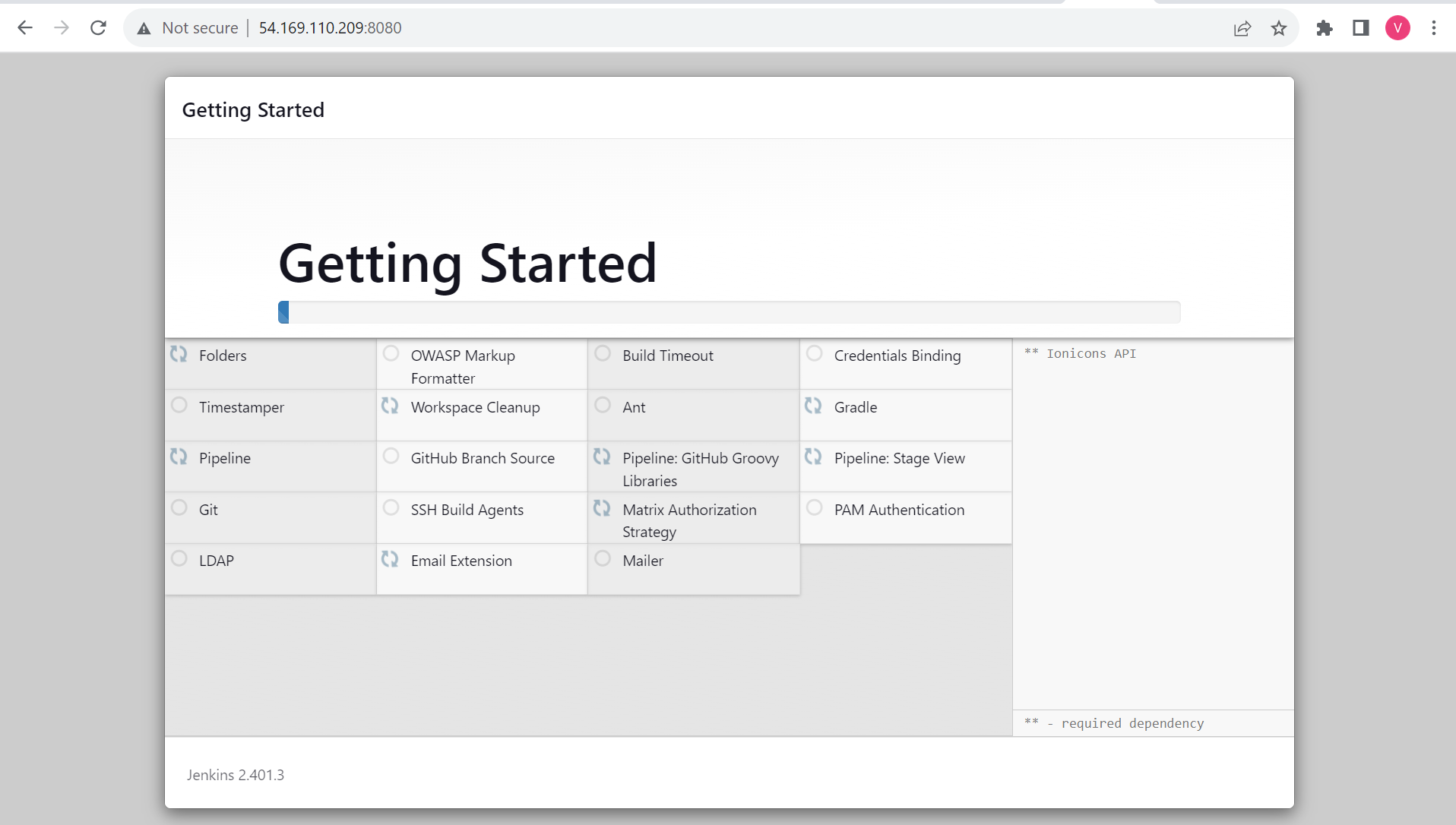
EKS cluster



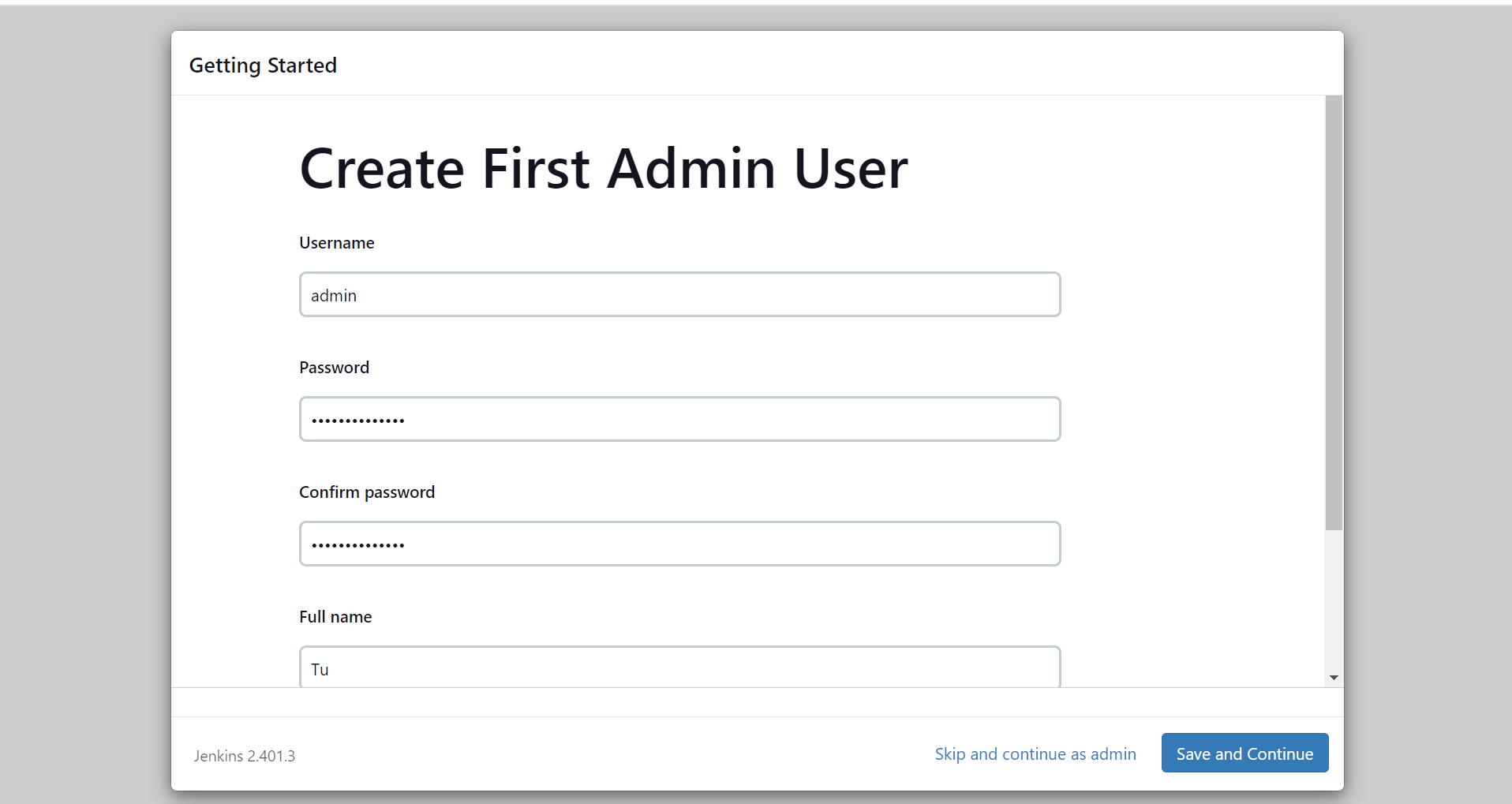
II. Setup Jenkins

Step 1: Install plugin

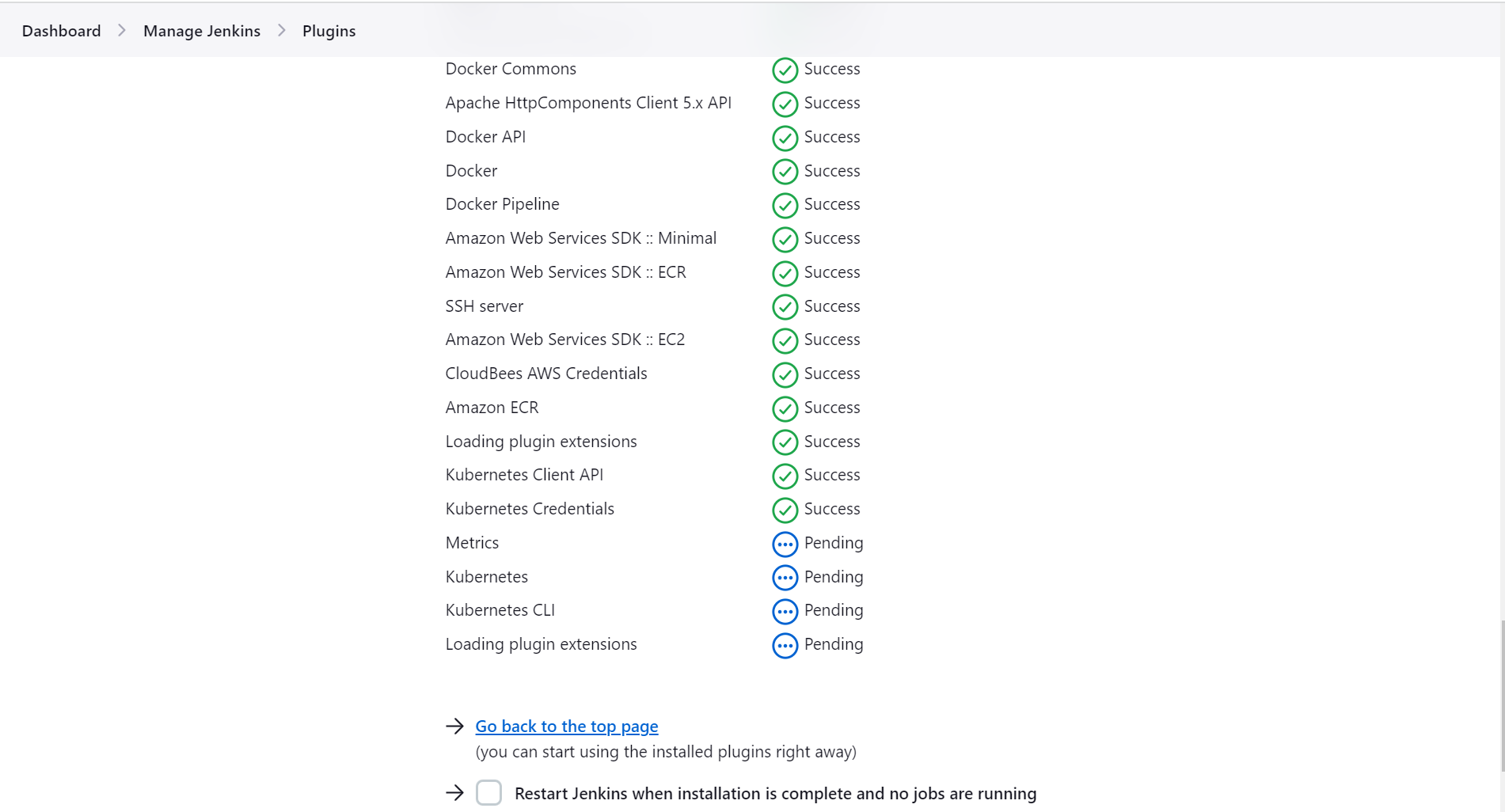




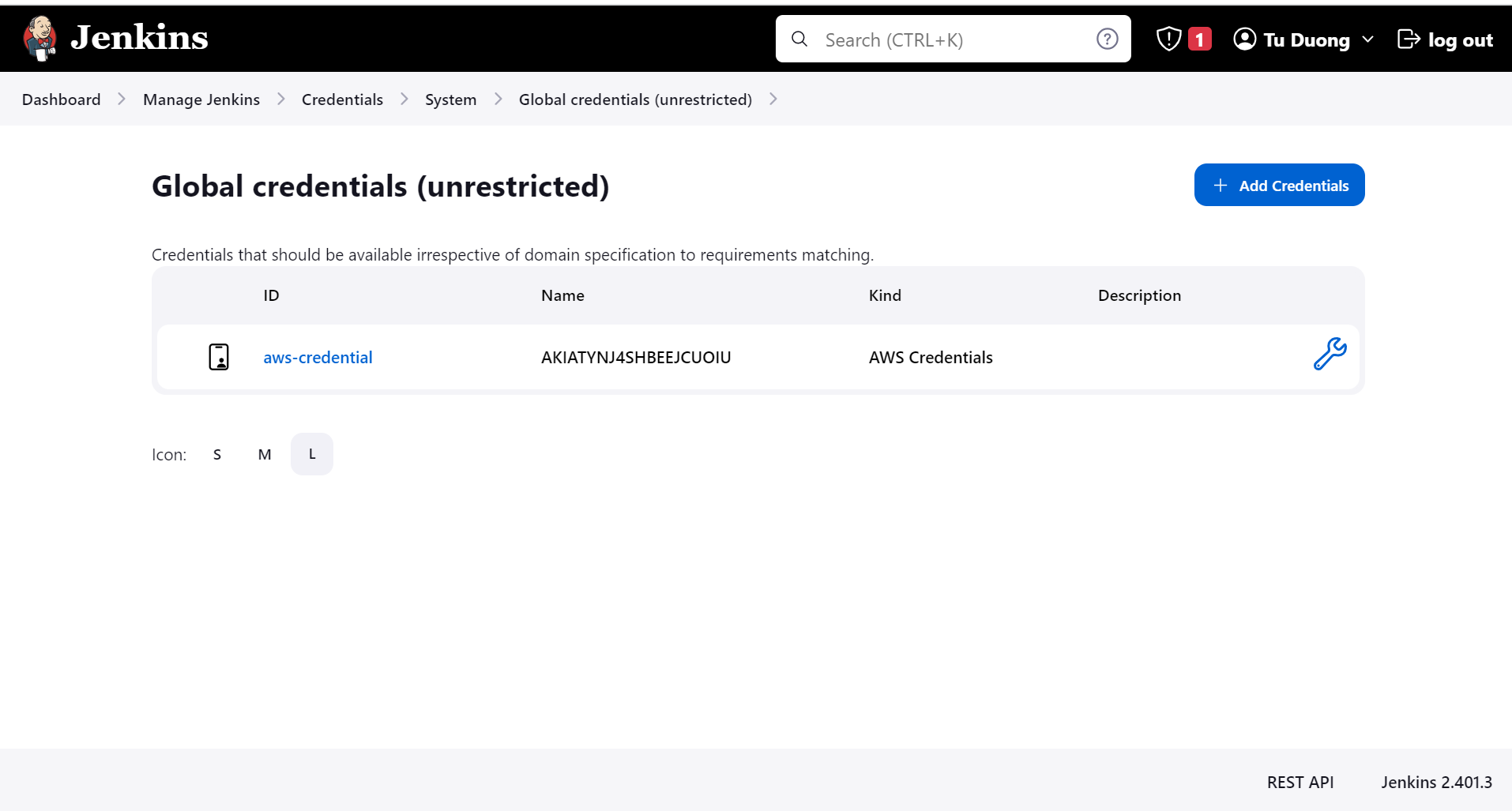
Step 2: Create admin user



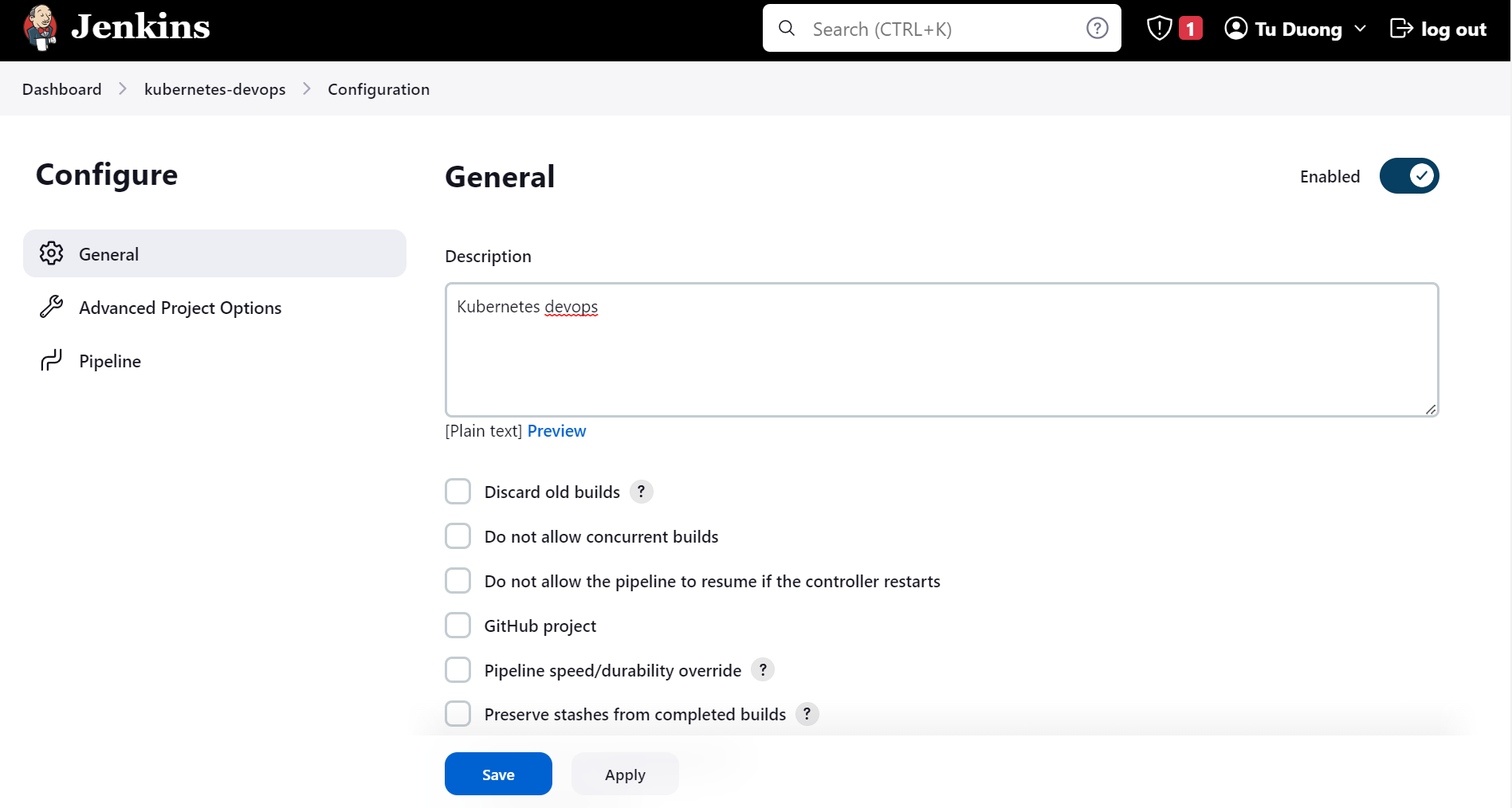
Step 3: Install tool for docker and Kubernetes

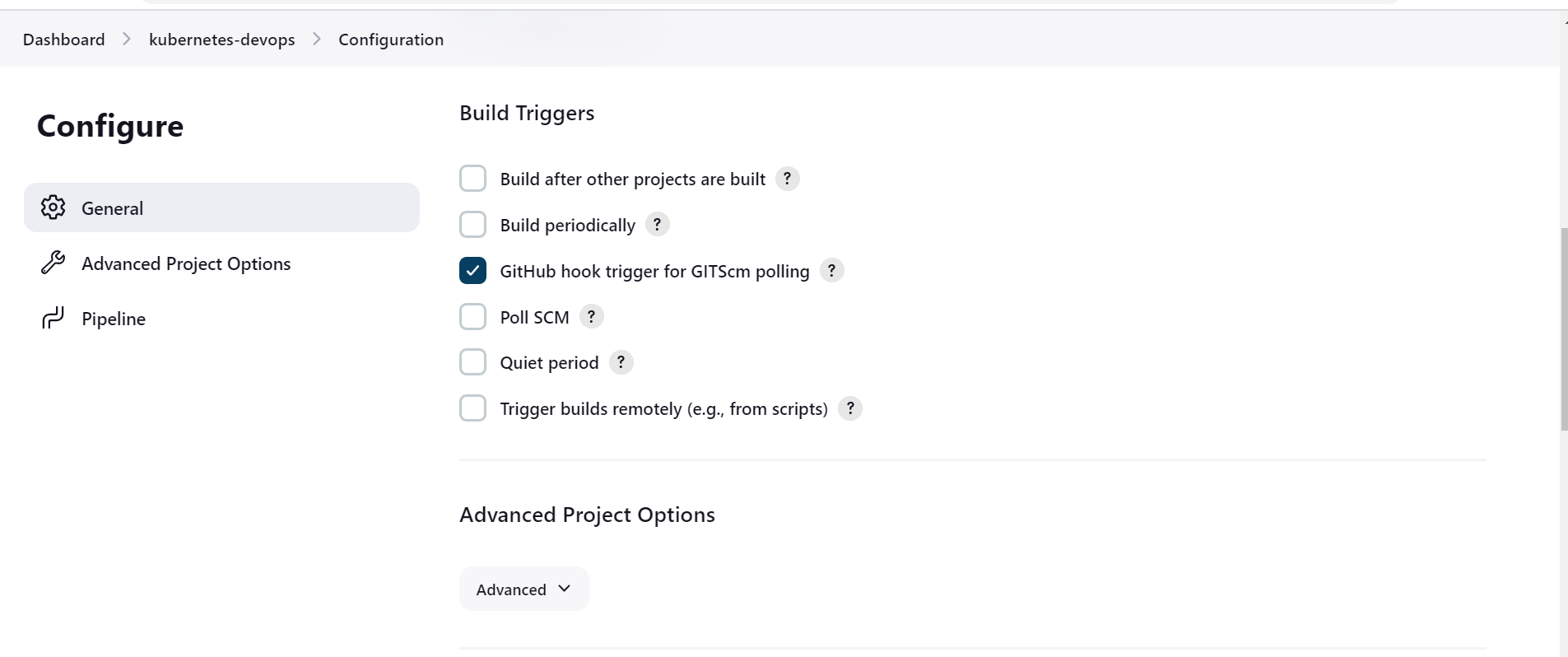


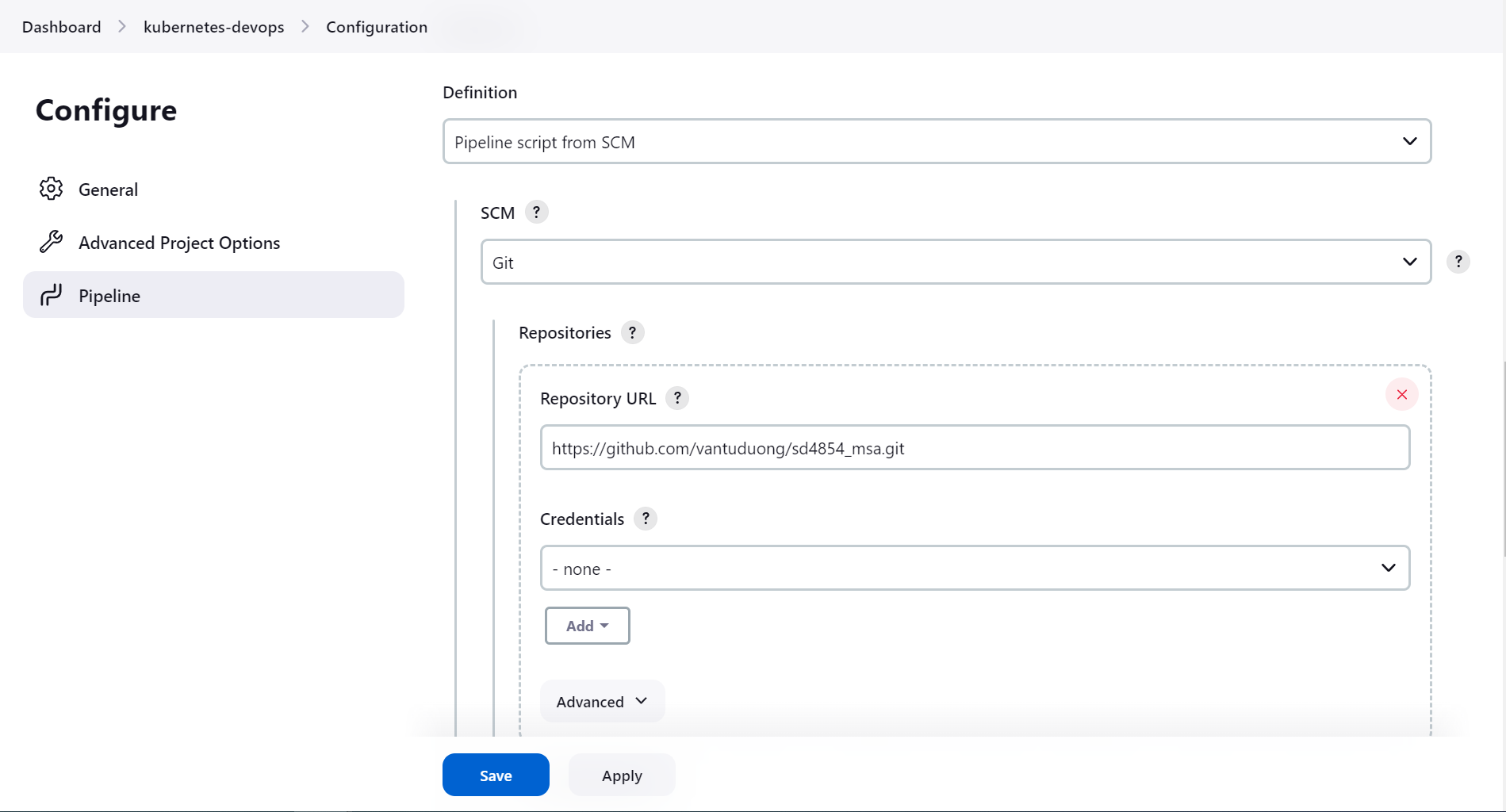
Step 4: Add credentials

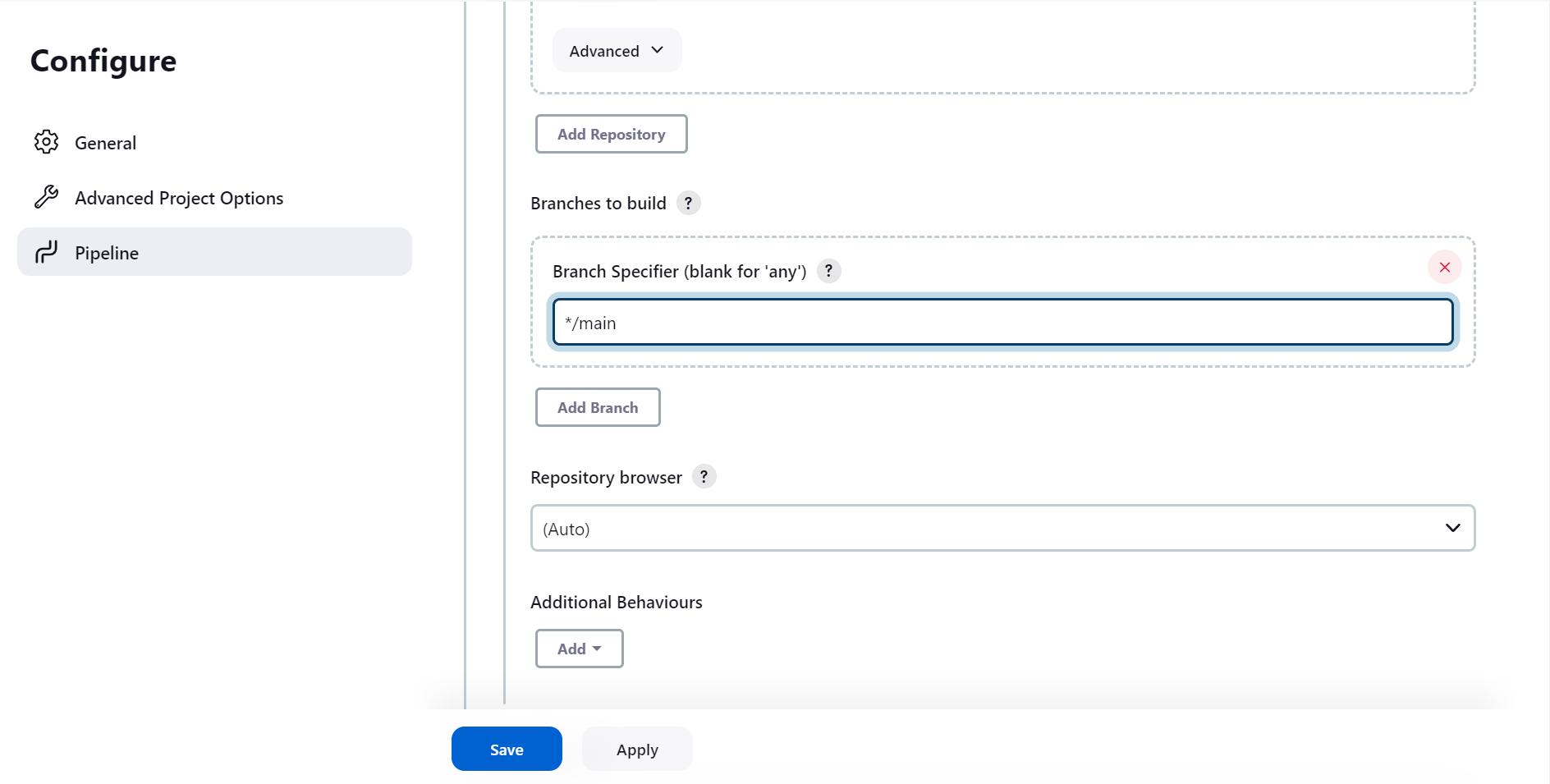


Step 5: Create a pineline

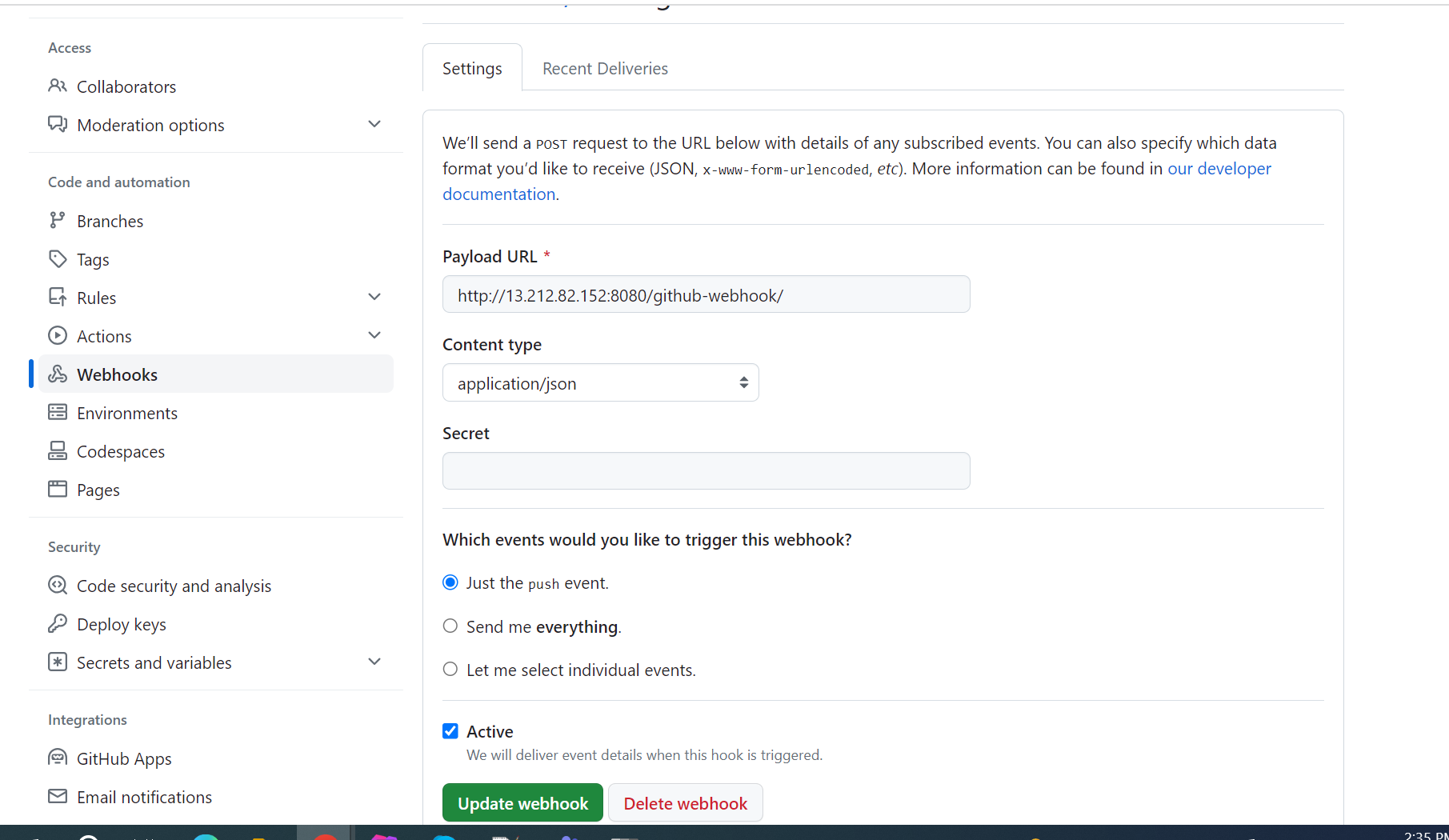








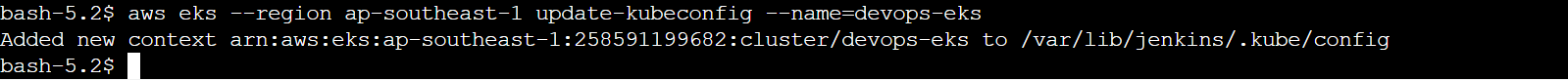
Step 5: Add github webhook



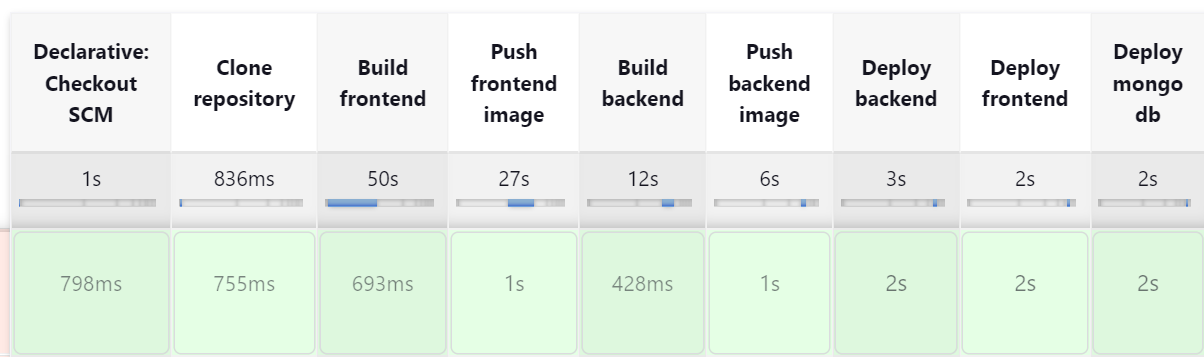
Step 6: Write Jenkin Pipeline

Check in repo: <https://github.com/vantuduong/sd4854_msa>

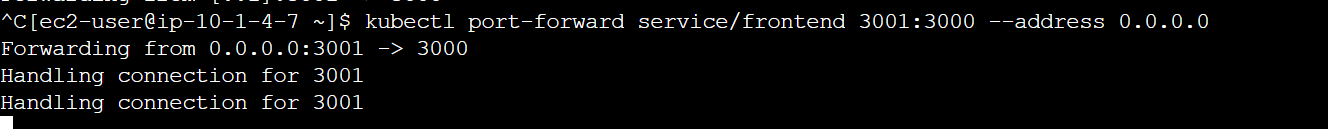
Step 7: Config aws credential and update kube config for jenkins user



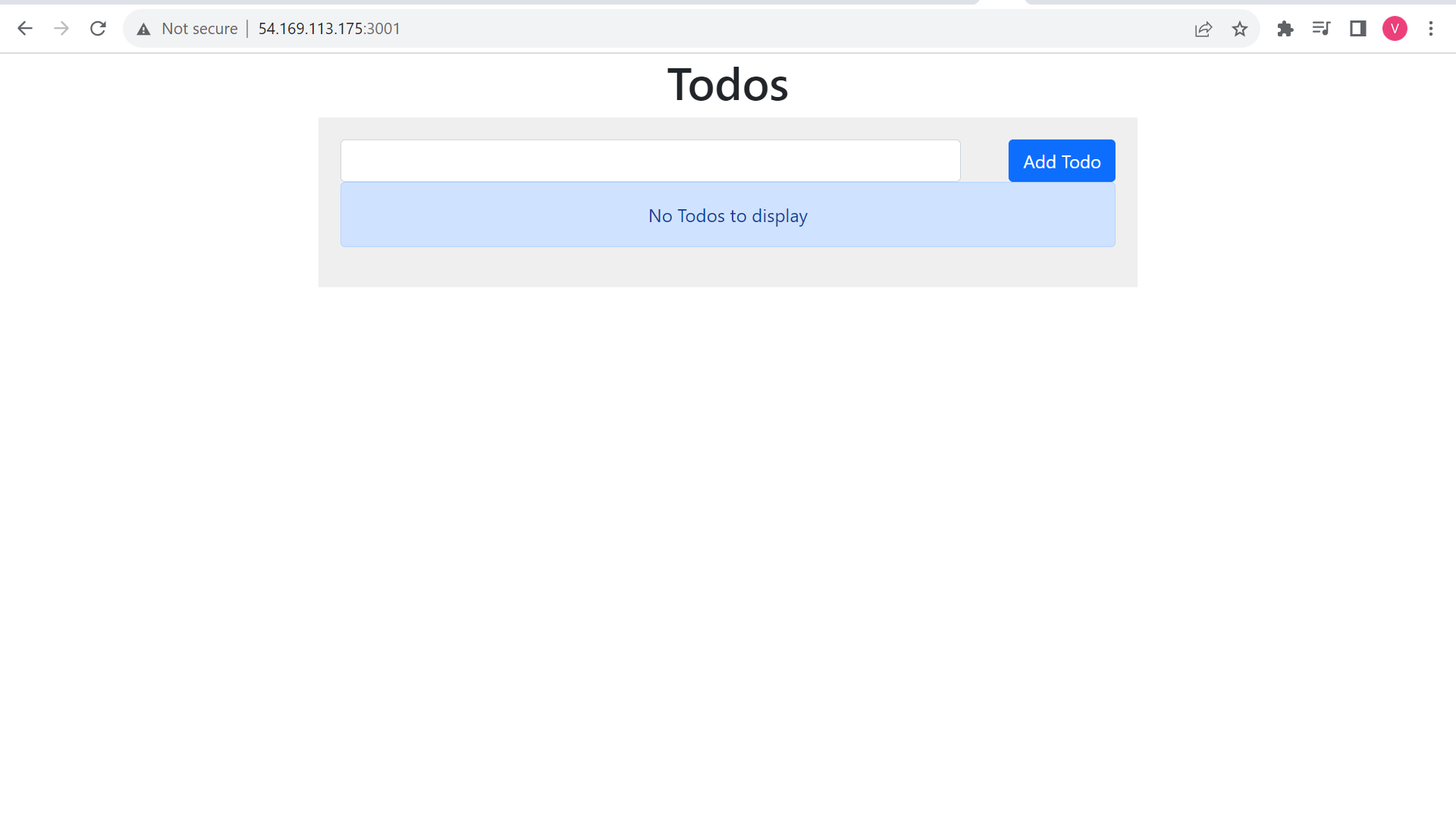
Step 8: Commit code to **main** branch & wait for pipeline run complete



Step 9: Forward port

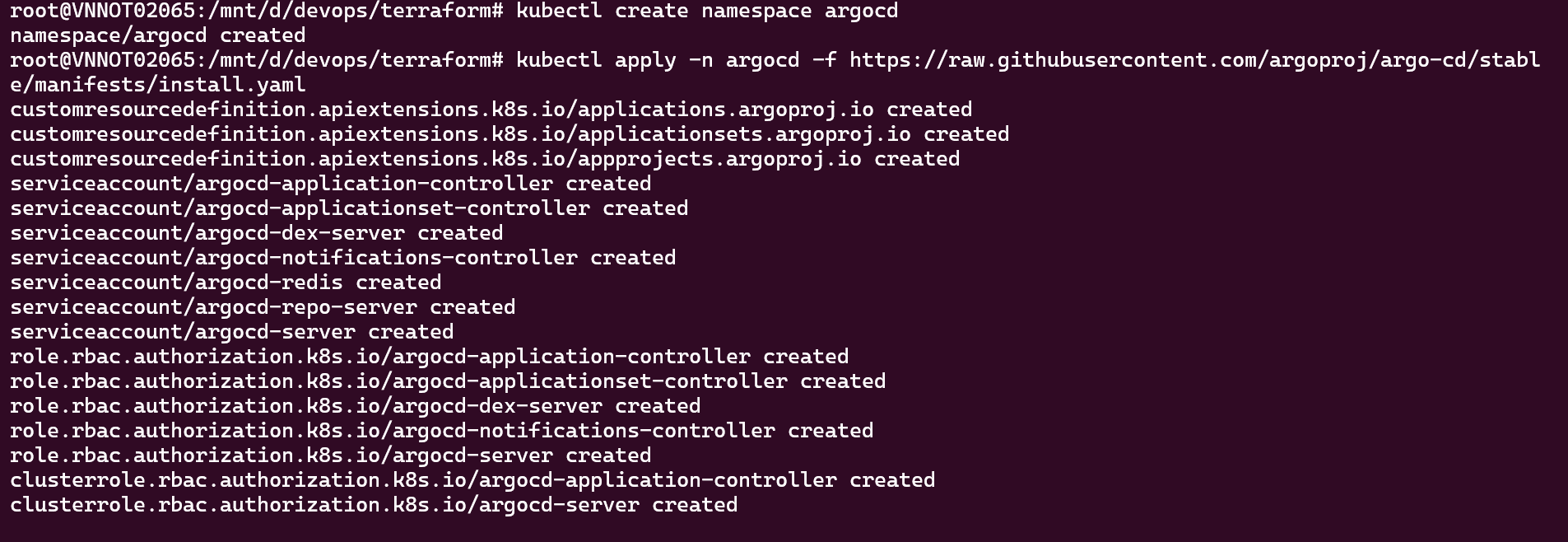


Step 10: Check result

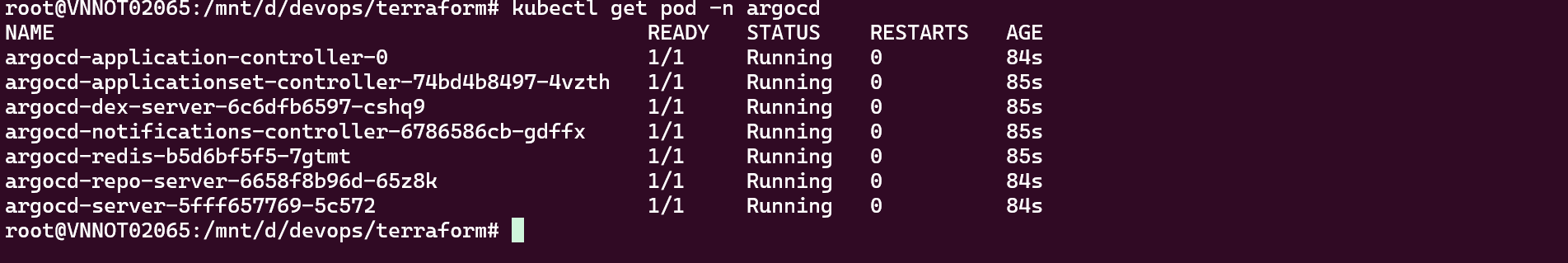


III. Setting ArgoCD

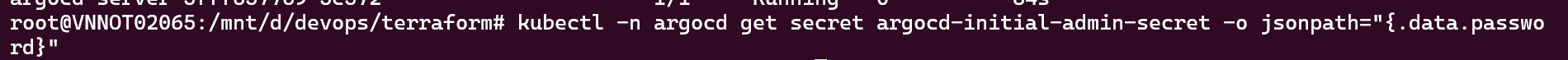
Step 1: Create namespace and install argocd into cluster



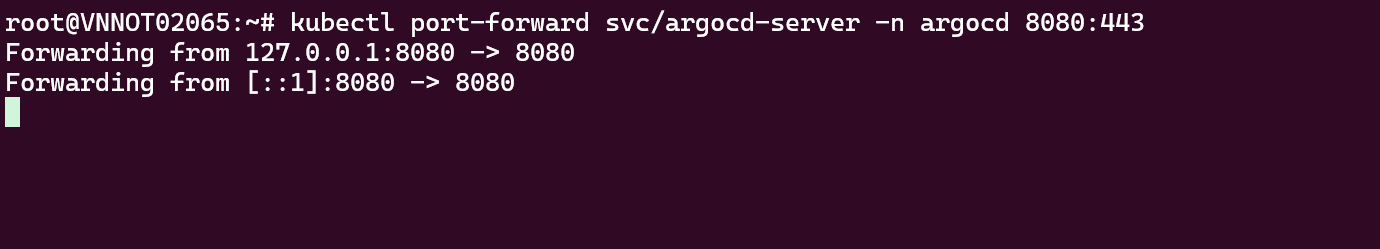
Step 2: Check pods running

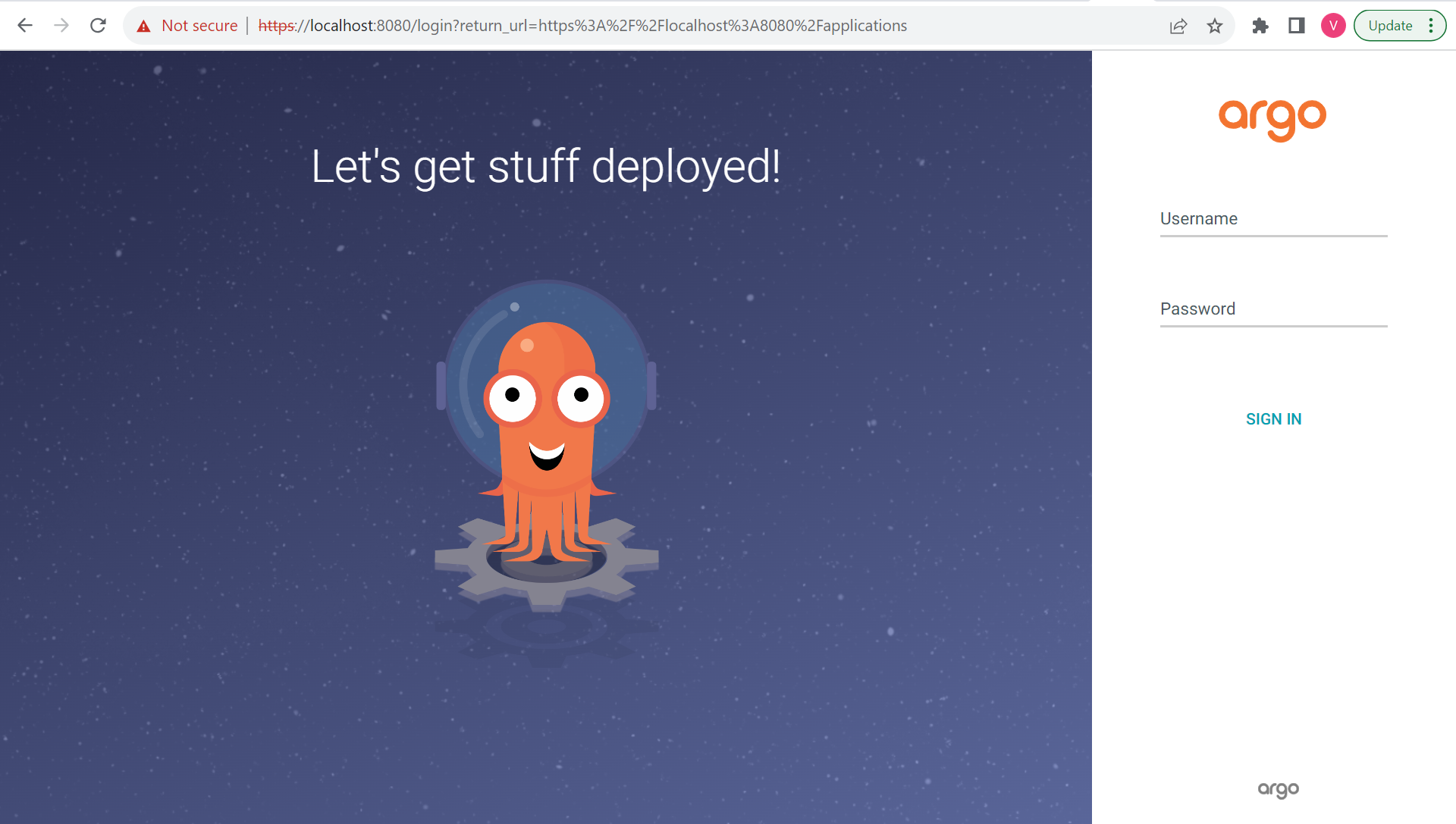


Step 3: Get initial admin password

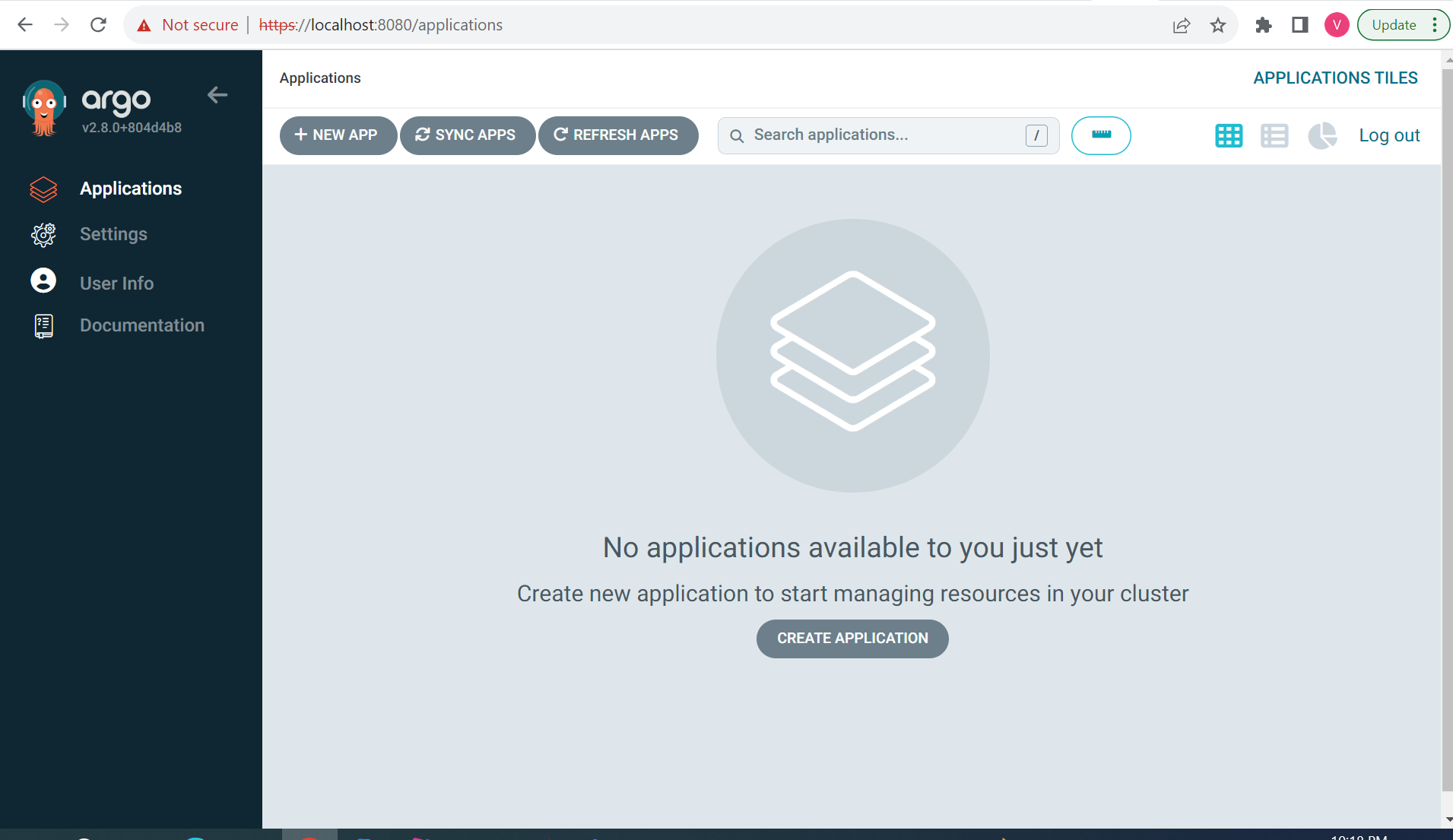


Step 4: Forward port to run on local

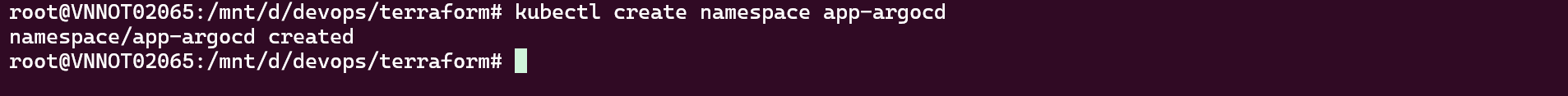




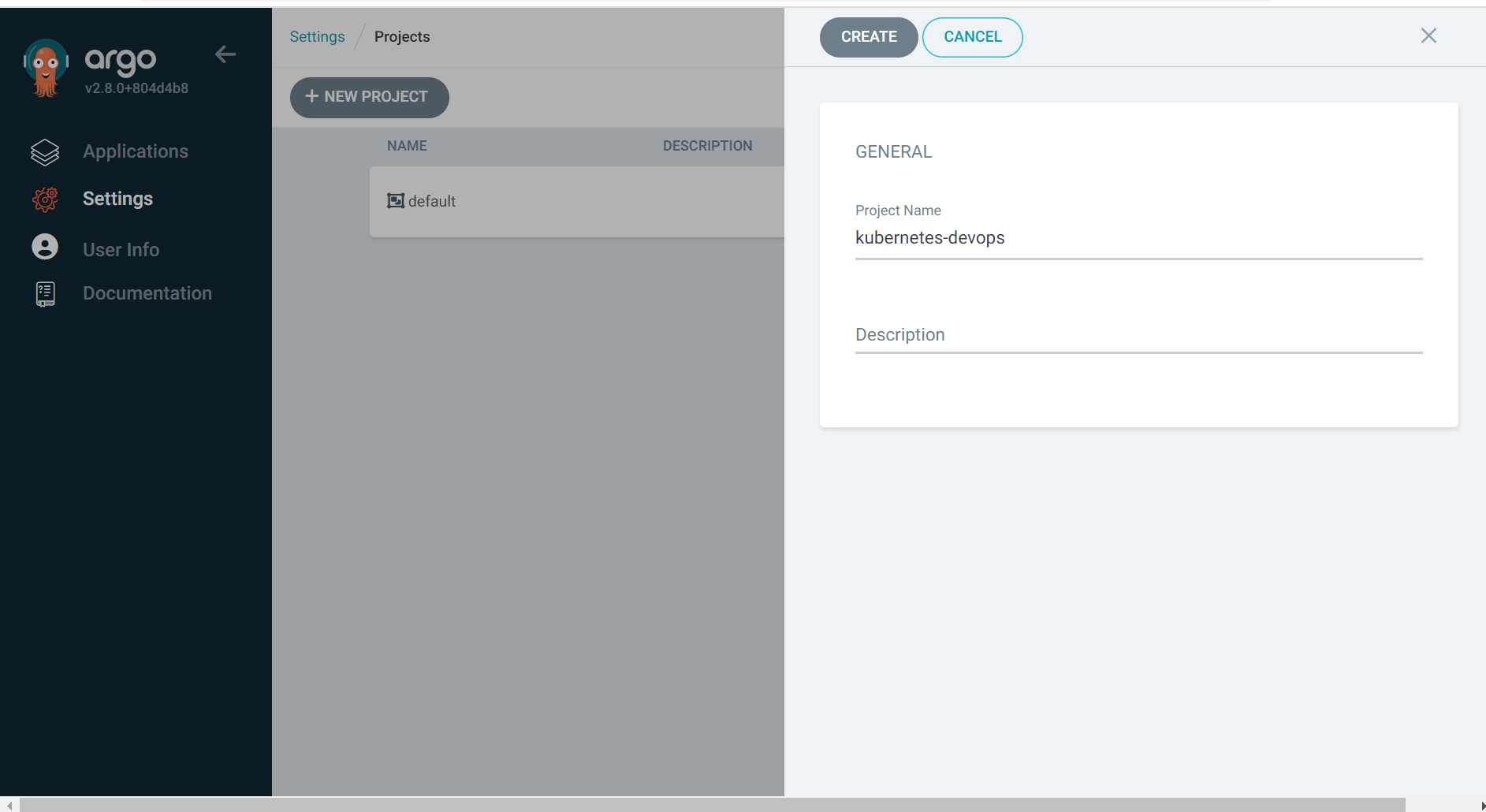
Step 5: Login



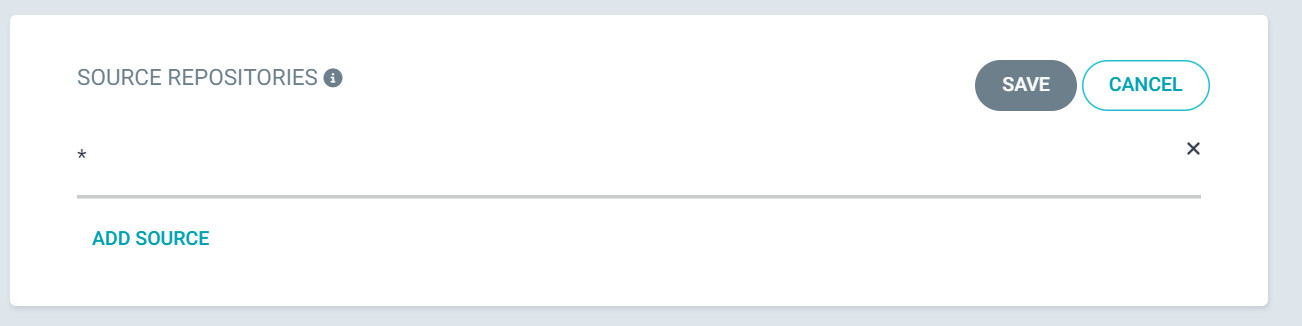
Step 6: Create app-argocd namespace

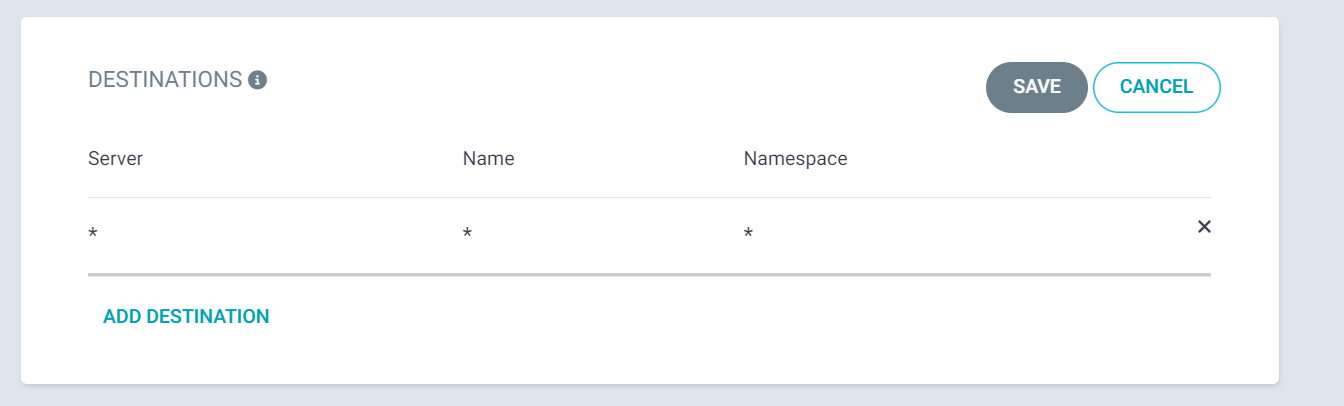


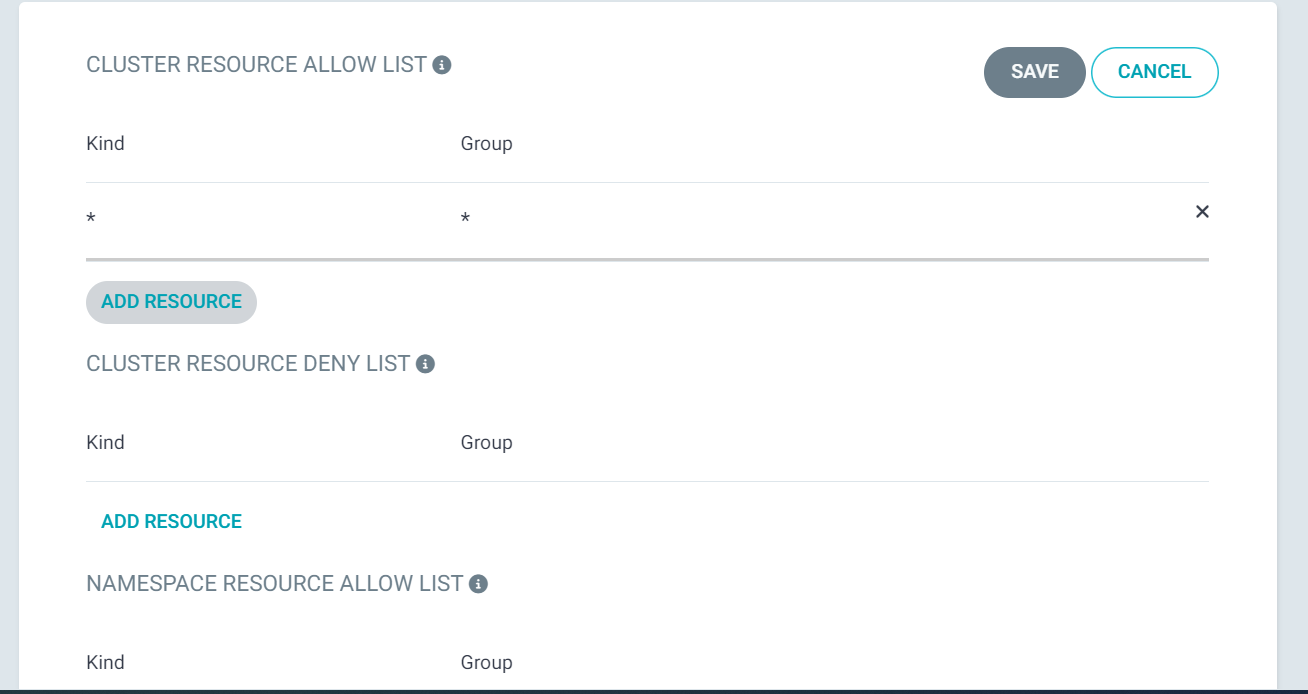
Step 7: Create a project



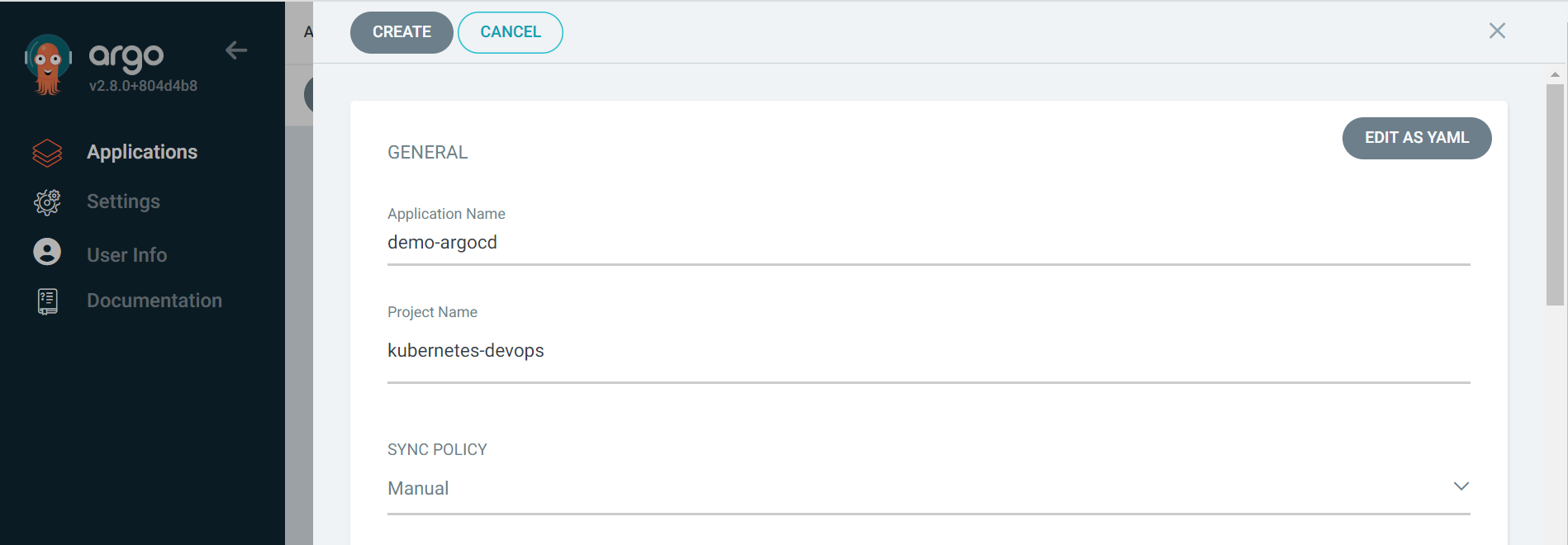
Step 8: Add source repo, destination, cluster resource allow list

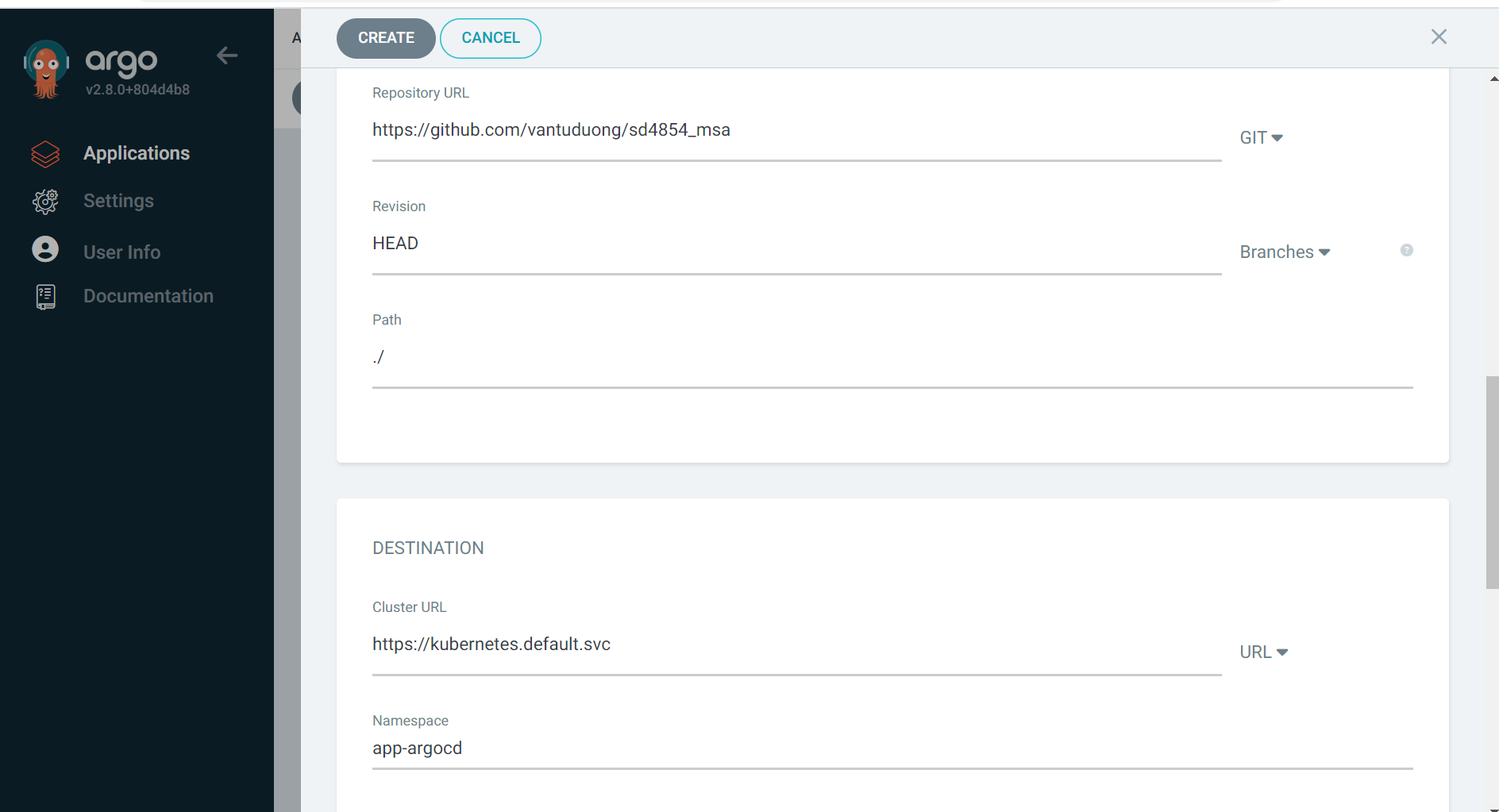


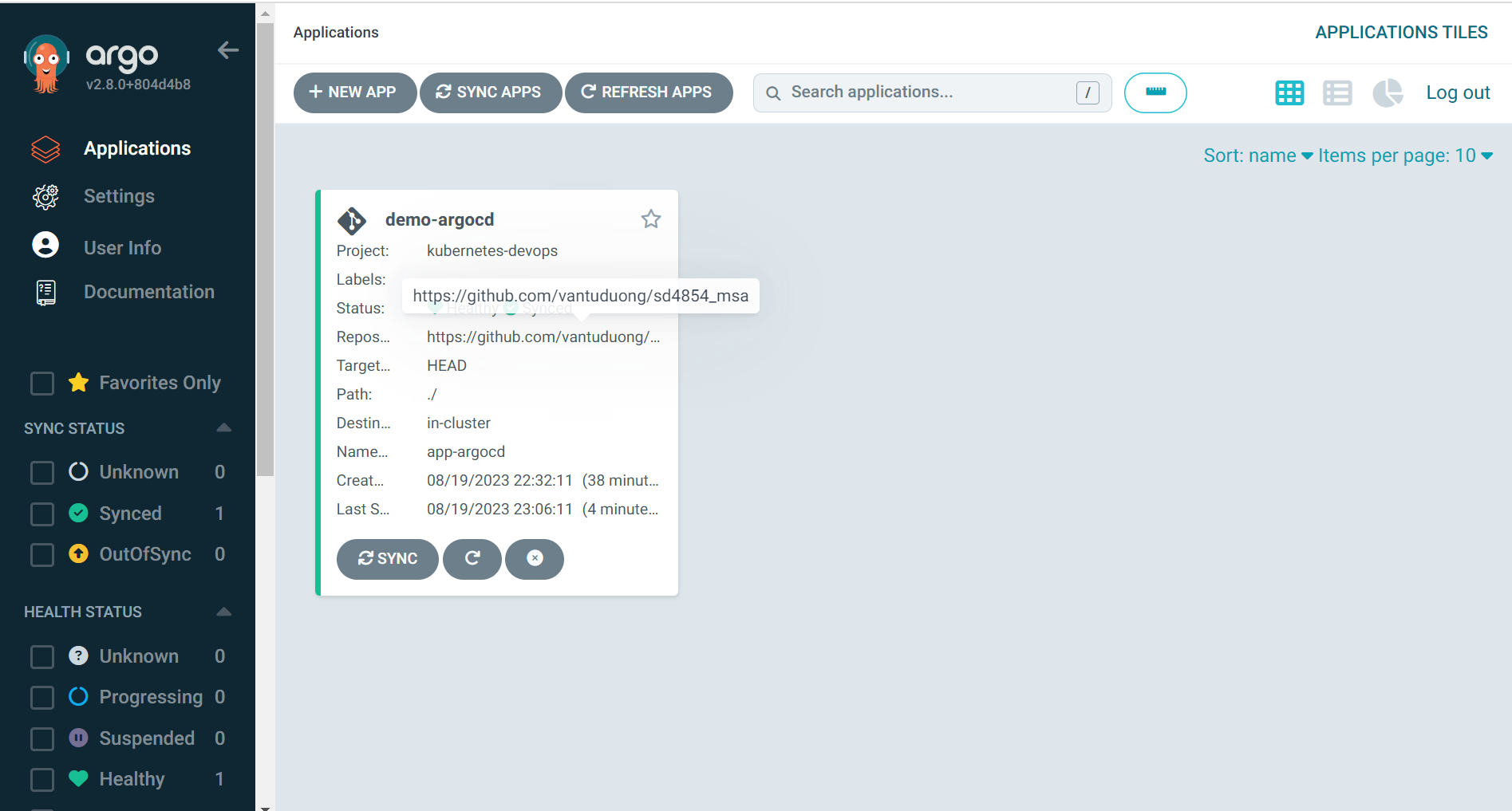




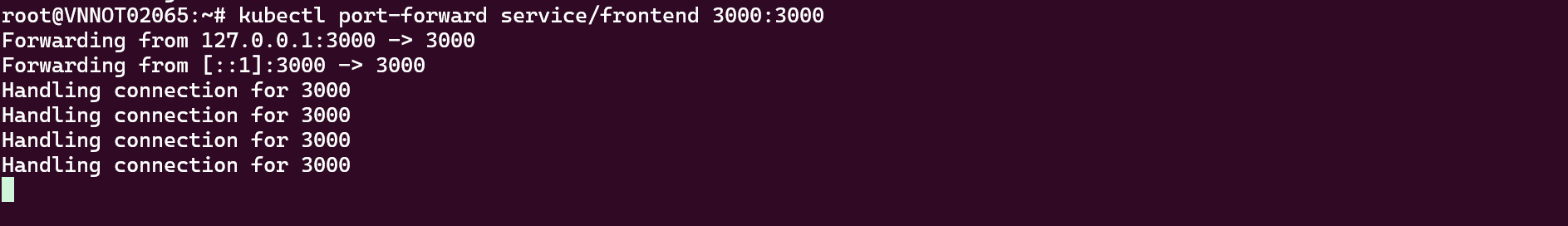
Step 9: Create app

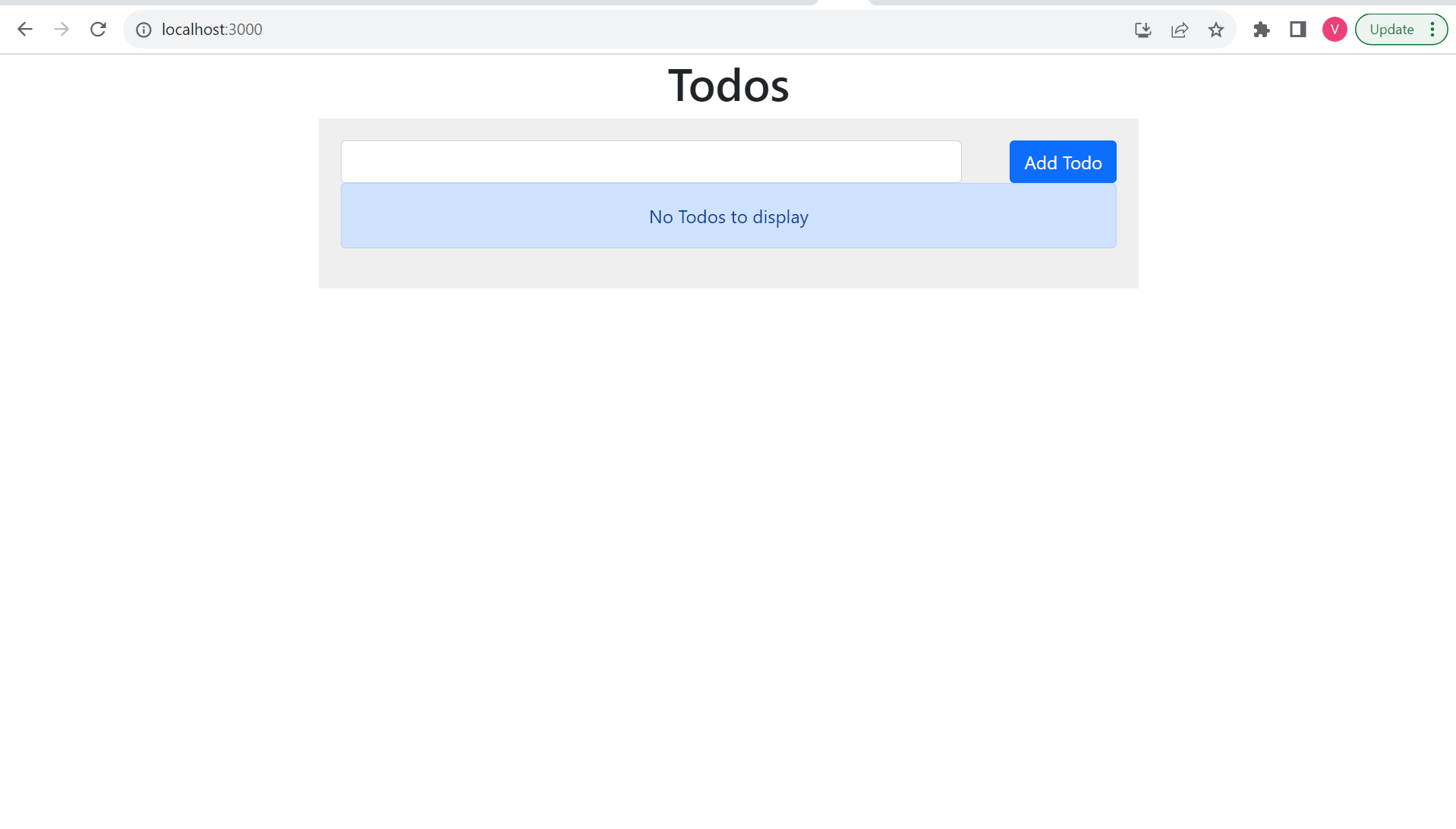






Step 11: Run application





Step 11: Update Jenkins file and frontend.yaml

* Remove CD scripts
* Push frontend image with tag **v2**
* Change image tag from **v1** to **v2**

Step 12: Resync argocd

