**GitOps CD using Argo CD for deployment in EKS**

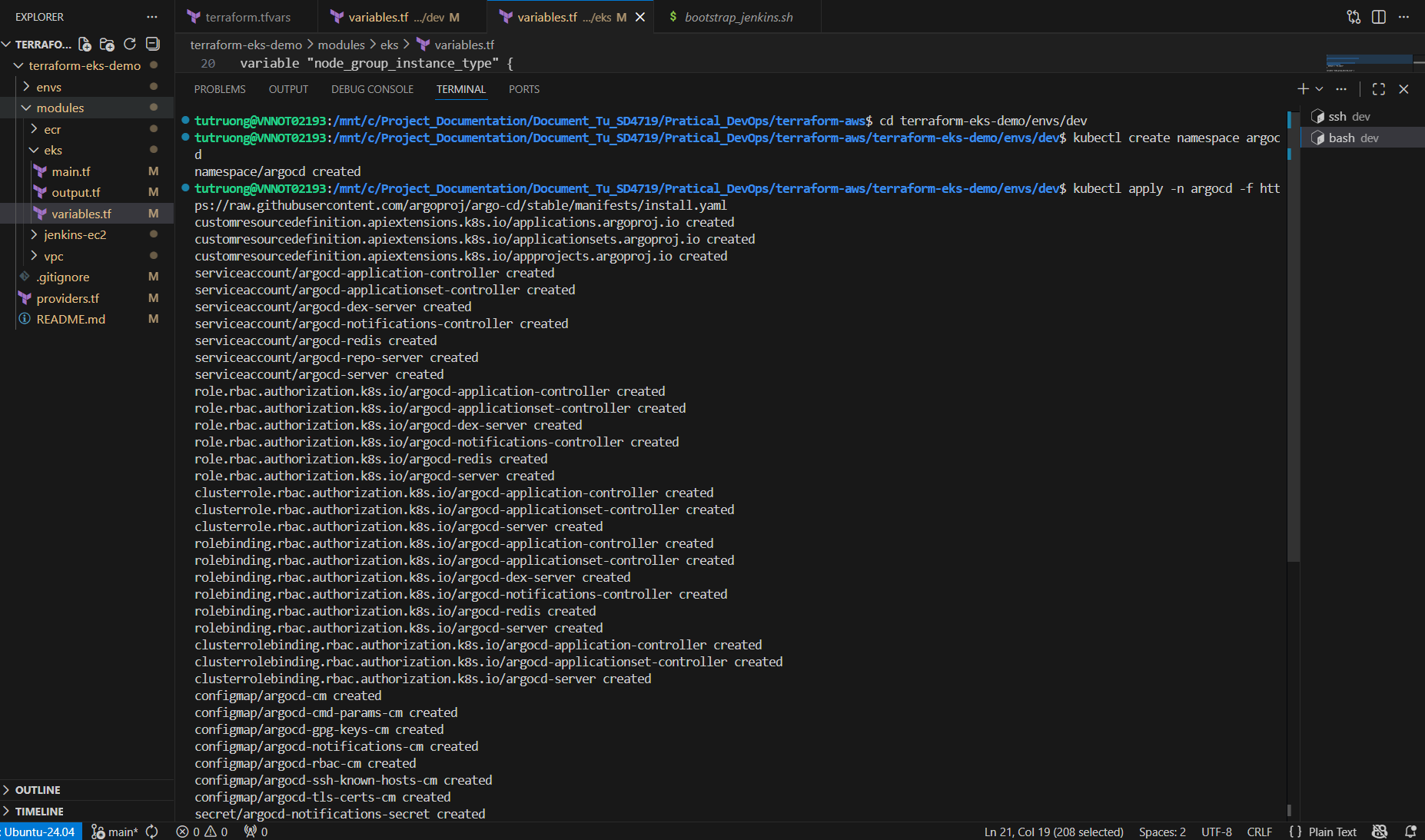
# ****- Install Argo CD on EKS:****

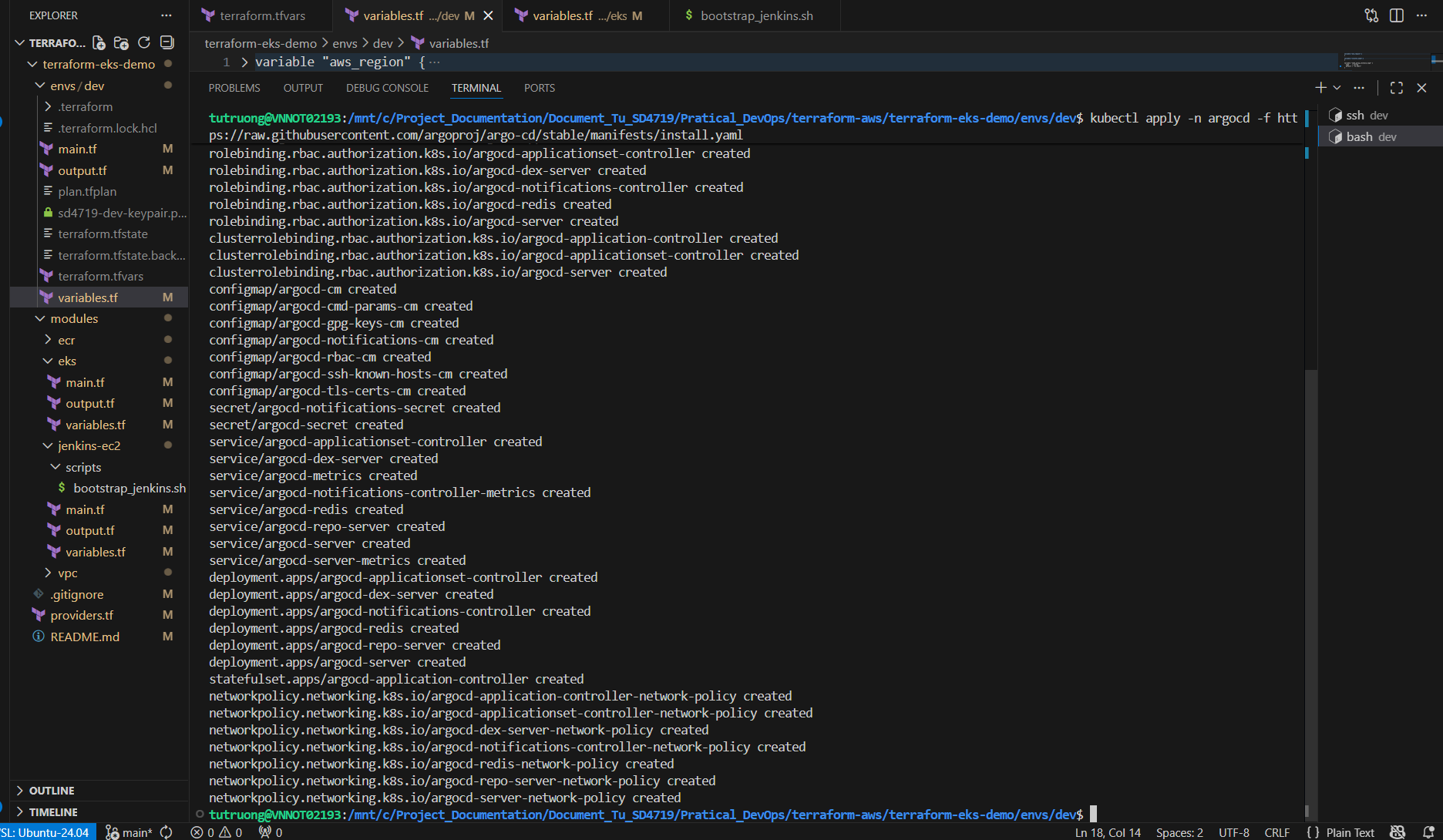
**1. Create namespace for Argo CD, run with:**

*kubectl create namespace argocd*

**2. Install Argo CD, run with:**

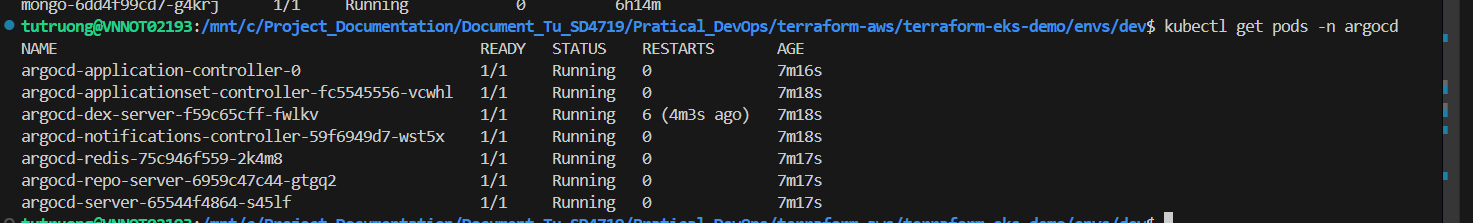
*kubectl apply -n argocd -f* [*https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml*](https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml)





**3. Verify Pods:**

*kubectl get pods -n argocd*



**- Expose Argo CD UI:**

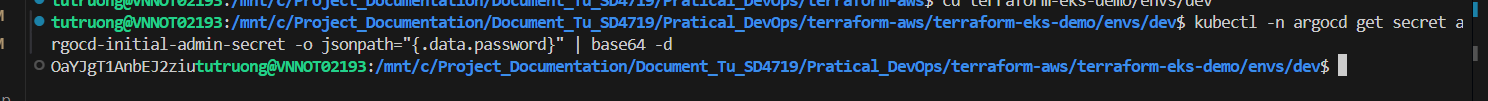
**Use port-forward, run with:**

*kubectl port-forward svc/argocd-server -n argocd 8080:443 &*

Then visit: [https://localhost:8080](https://localhost:8080/)

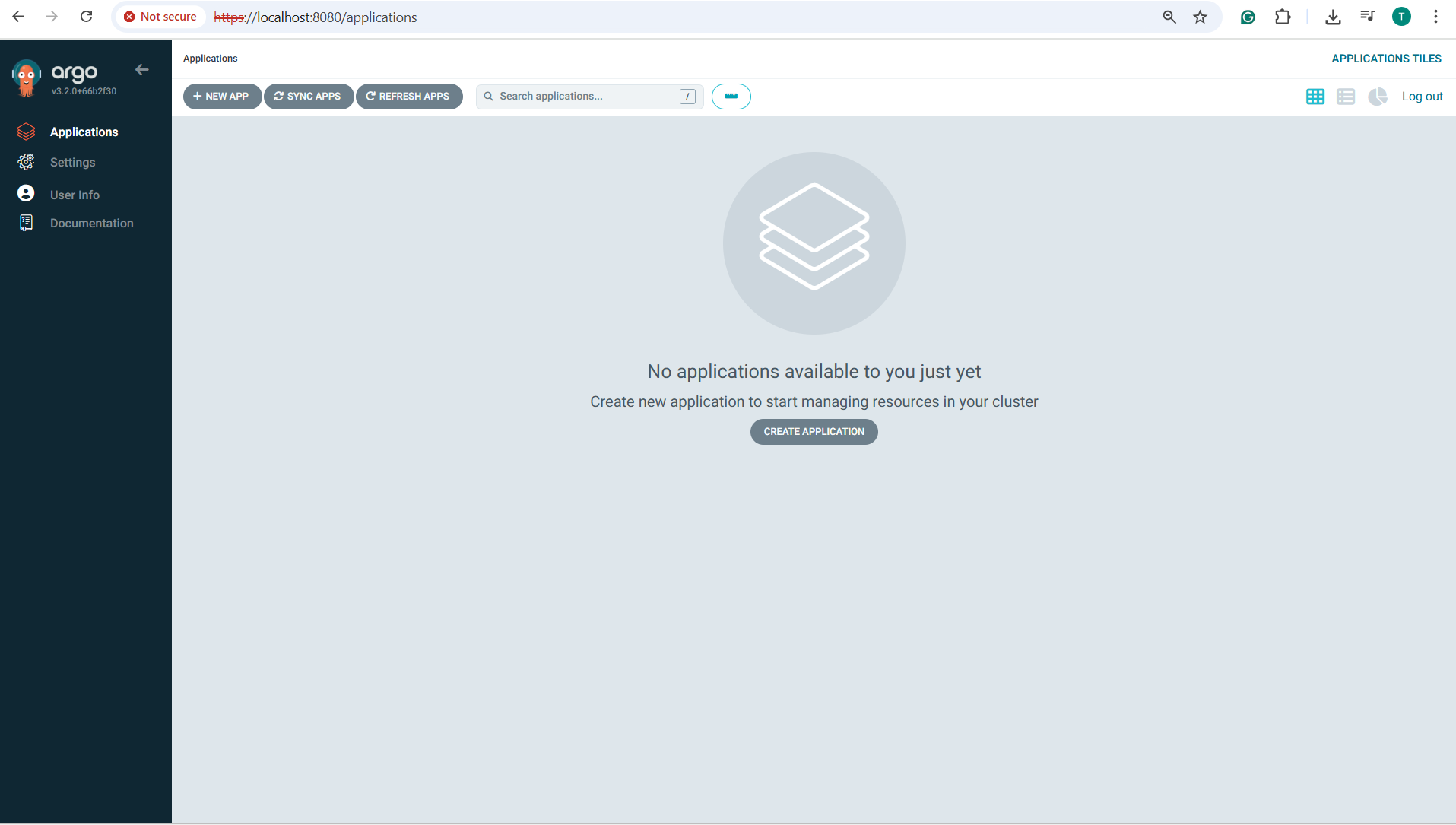
Get admin password:

*kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d*



Username: admin

password: OaYJgT1AnbEJ2ziu



**- Create an Application in Argo CD:**

****1. Install** argocd (using CLI in WSL terminal)**:****

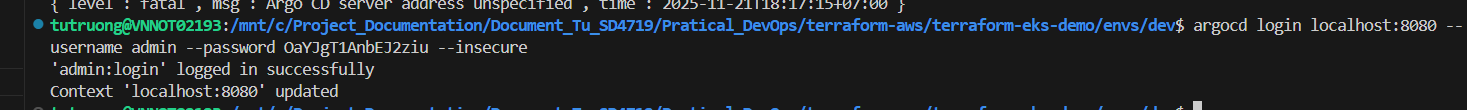
***VERSION=$(curl -L -s https://raw.githubusercontent.com/argoproj/argo-cd/stable/VERSION)*  *curl -sSL -o argocd-linux-amd64 https://github.com/argoproj/argo-cd/releases/download/v$VERSION/argocd-linux-amd64***

***sudo install -m 555 argocd-linux-amd64 /usr/local/bin/argocd***

***rm argocd-linux-amd64***

****2. Login to Argo CD server****

***argocd login localhost:8080 --username admin --password OaYJgT1AnbEJ2ziu –insecure***

**

**3. Create Application:**

*argocd app create my-app-argocd \*

*--repo https://github.com/TuTruonng/sd4719\_msa.git \*

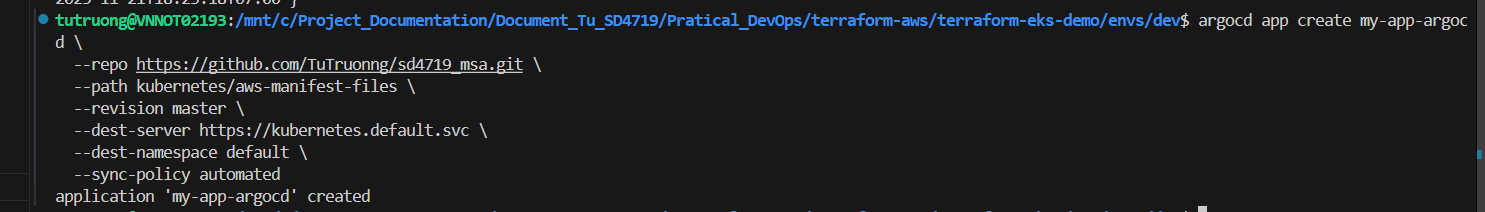
*--path kubernetes/aws-manifest-files \*

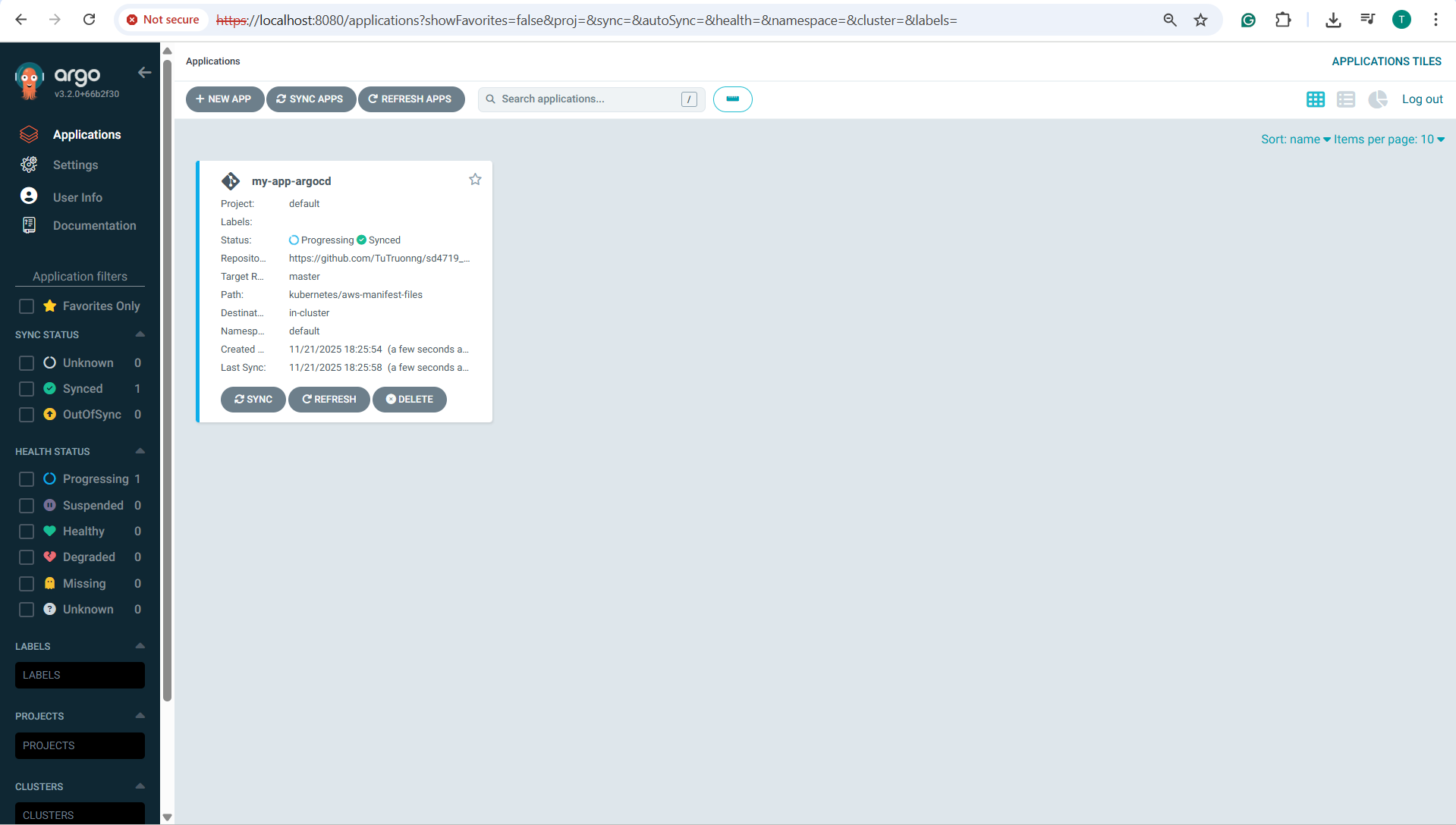
*--revision master \*

*--dest-server https://kubernetes.default.svc \*

*--dest-namespace default \*

*--sync-policy automated*

**

**

**- Install Argo CD Image Updater:**

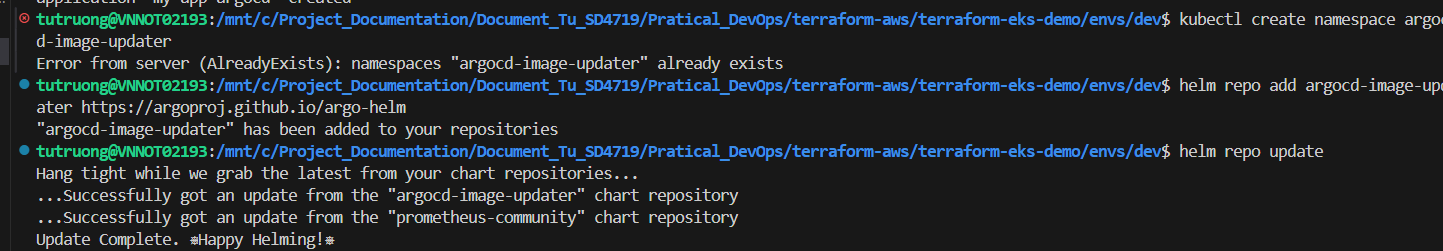
**1. Create namespace:**

***kubectl create namespace argocd-image-updater***

**2. Install using Helm:**

***helm repo add argocd-image-updater https://argoproj.github.io/argo-helm***

***helm repo update***

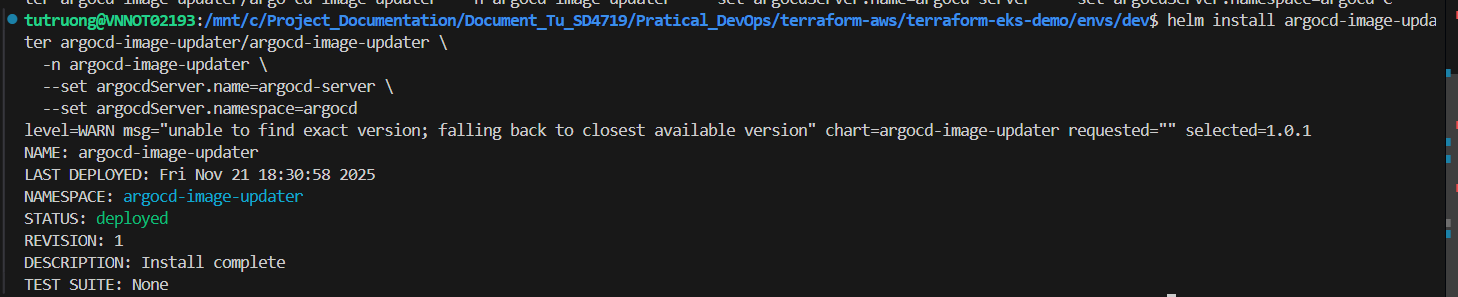


***helm install argocd-image-updater argocd-image-updater/argocd-image-updater \***

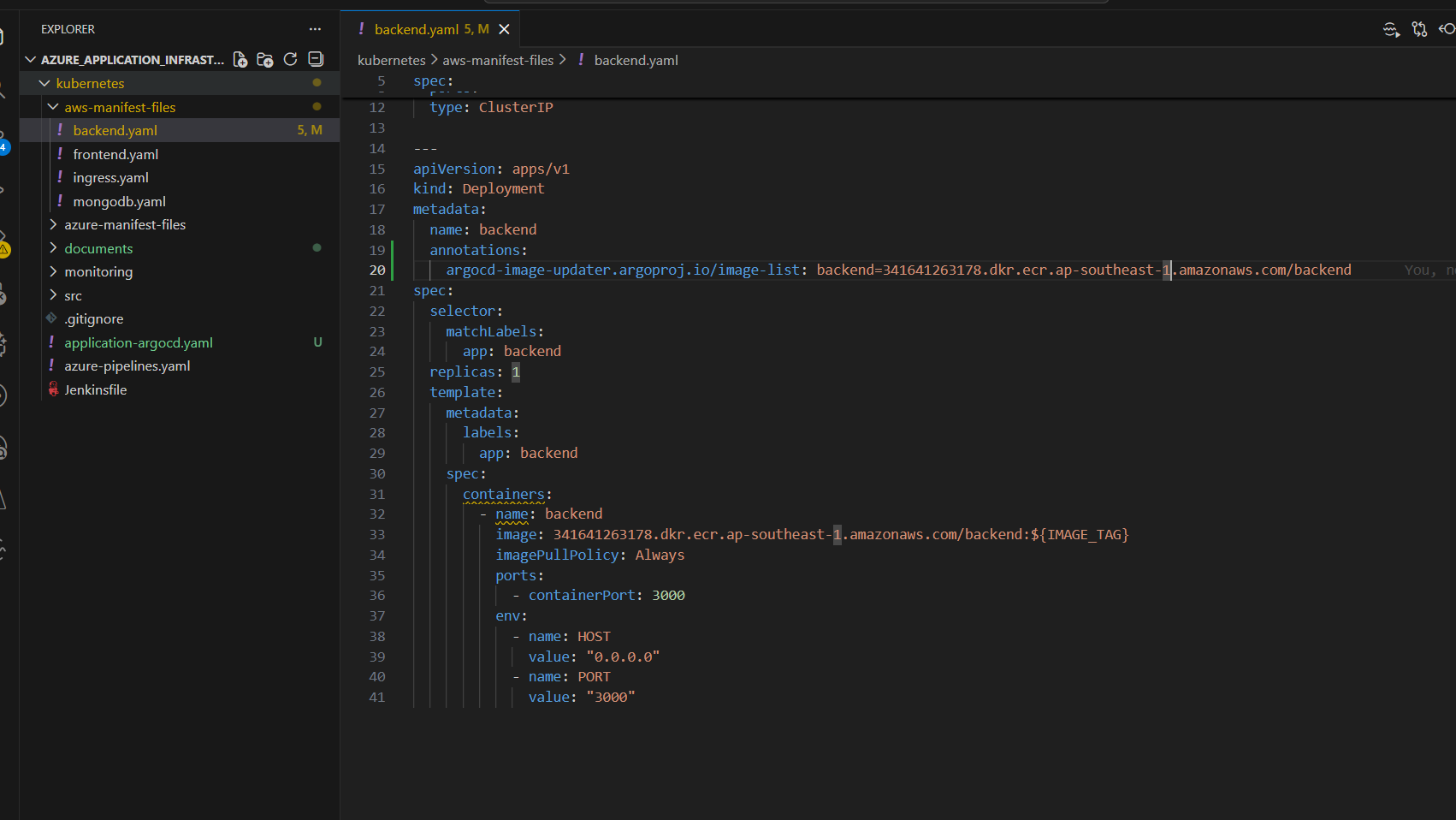
***-n argocd-image-updater \***

***--set argocdServer.name=argocd-server \***

***--set argocdServer.namespace=argocd***

**

**- Annotate the backend.yaml file for Image Updater:**



**- Add Argo CD Application YAML (https://github.com/TuTruonng/sd4719\_msa/blob/master/kubernetes/application-argocd.yaml)**

**- Update the Jenkinsfile file in application (https://github.com/TuTruonng/sd4719\_msa/blob/master/kubernetes/Jenkinsfile)**

**- Install Argo CD CLI on Jenkins agent:**

#### In Jenkins agent, run with:

*curl -sSL -o /tmp/argocd https://github.com/argoproj/argo-cd/releases/latest/download/argocd-linux-amd64*

*chmod +x /tmp/argocd*

*sudo mv /tmp/argocd /usr/local/bin/argocd*

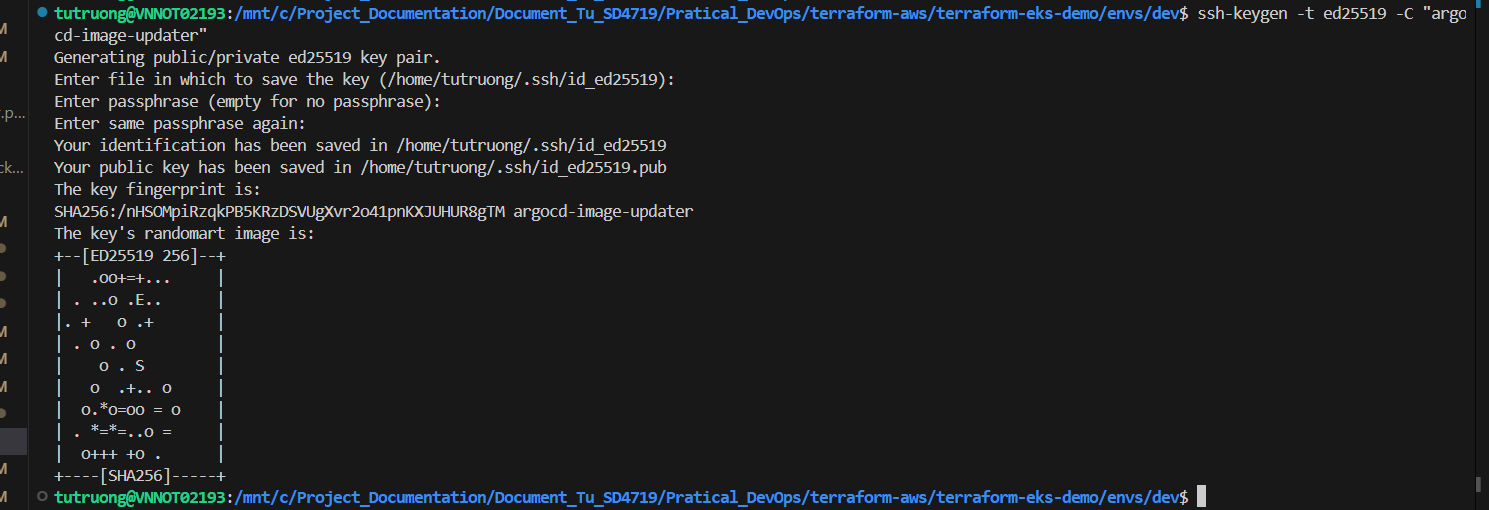
*argocd version*



**- Register GitHub credentials for Argo CD Image Updater Gets GitHub Access: Using GitHub SSH Key**

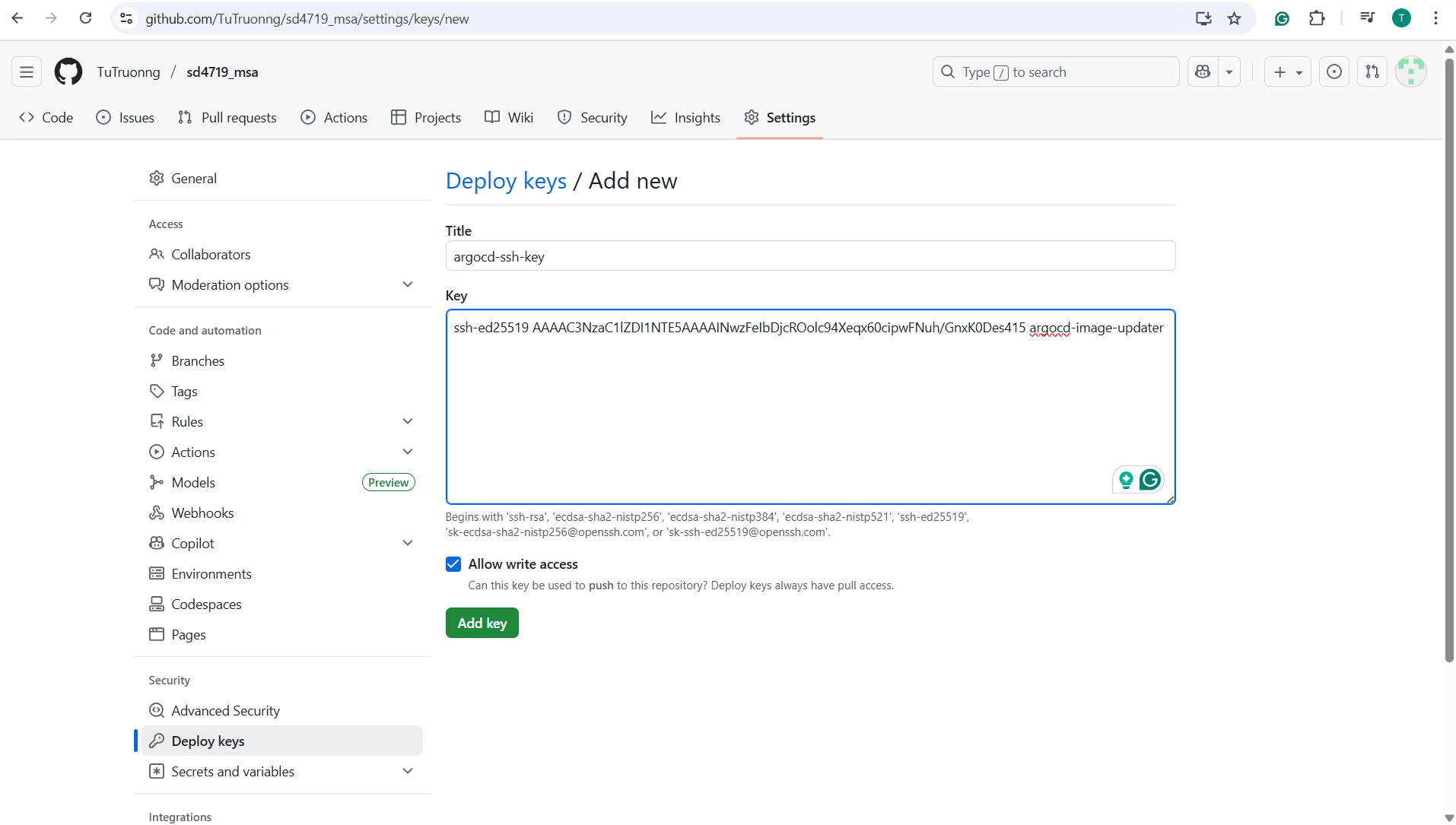
**1. Generate SSH key:**

*ssh-keygen -t ed25519 -C "argocd-image-updater"*



**2. Add public key to GitHub:**

Go to **GitHub => Your GitOps Repo => Settings => Deploy Keys => Add Deploy Key**



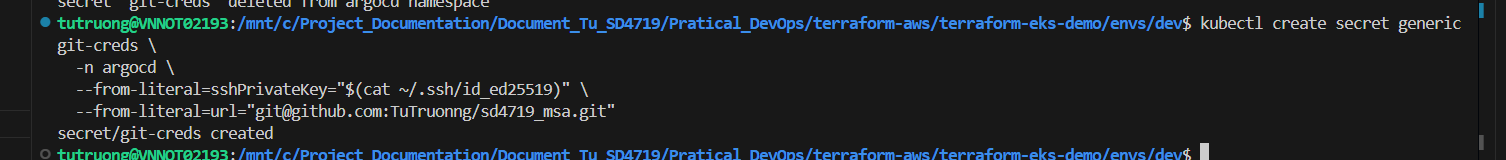
## 3. ****Store private key in Argo CD:****

*kubectl create secret generic git-creds \*

*-n argocd \*

*--from-literal=sshPrivateKey="$(cat ~/.ssh/id\_ed25519)" \*

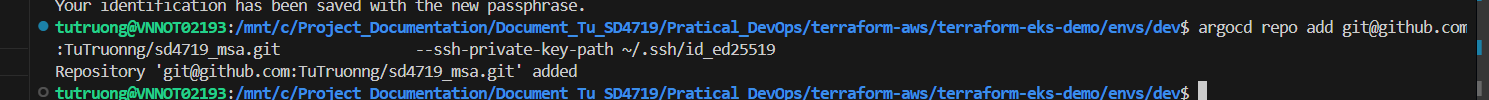
*--from-literal=url="*[*git@github.com*](mailto:git@github.com)*:TuTruonng/sd4719\_msa.git"*



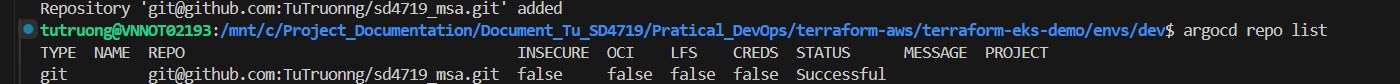
4.**Register repo in ArgoCD:**

*argocd repo add git@github.com:TuTruonng/sd4719\_msa.git \*

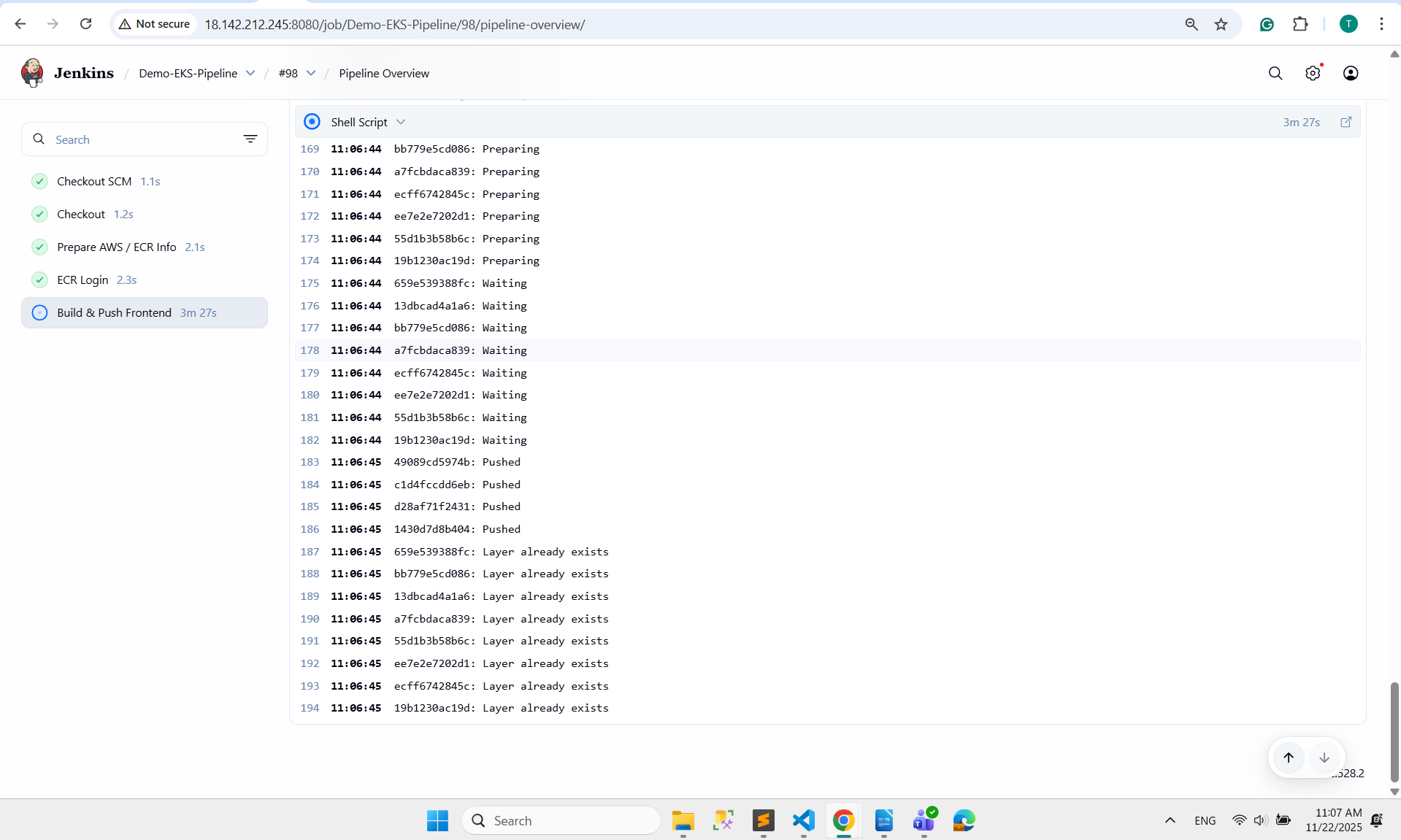
*--ssh-private-key-path ~/.ssh/id\_ed25519*

**

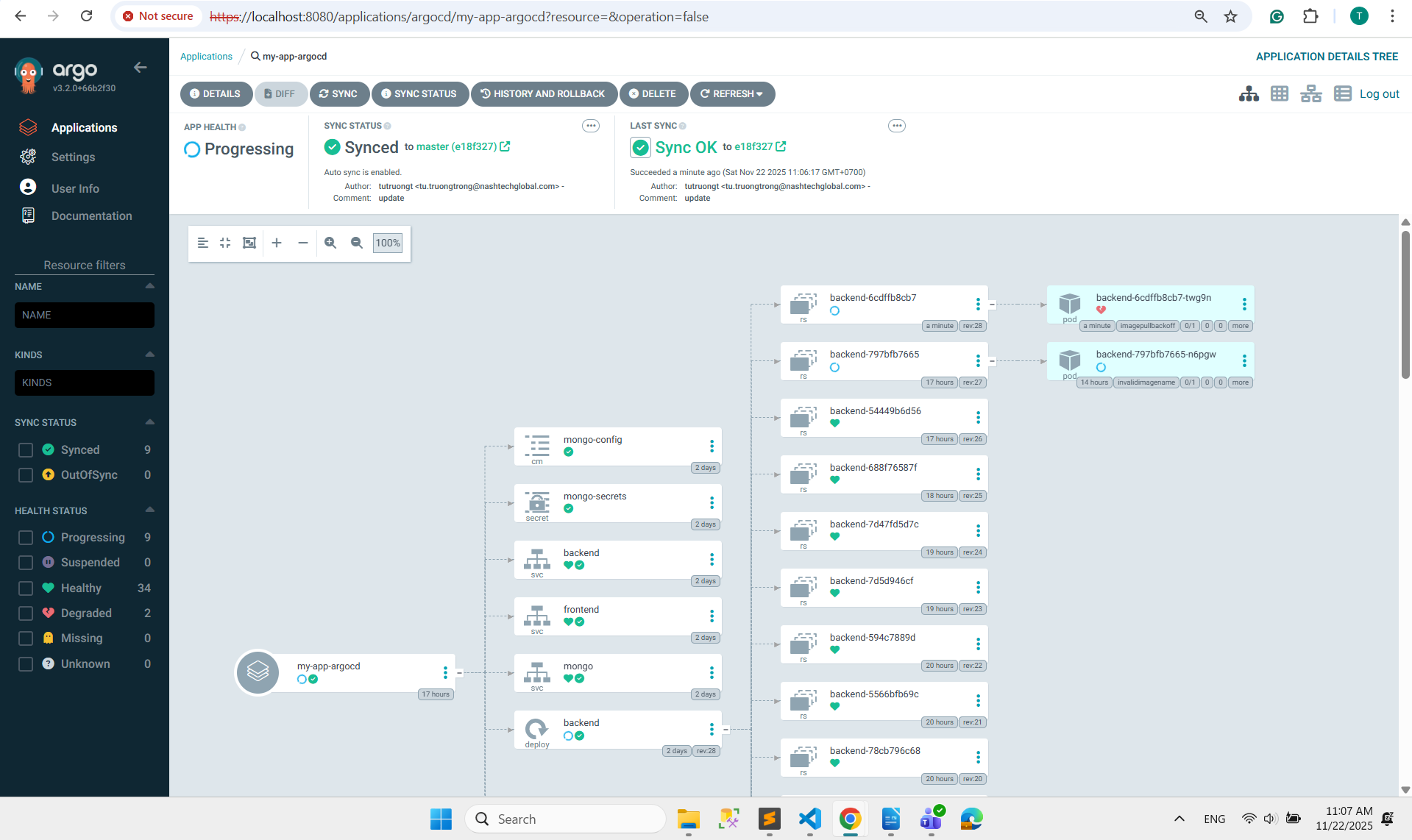
***5.* Confirm ArgoCD sees the repo:**



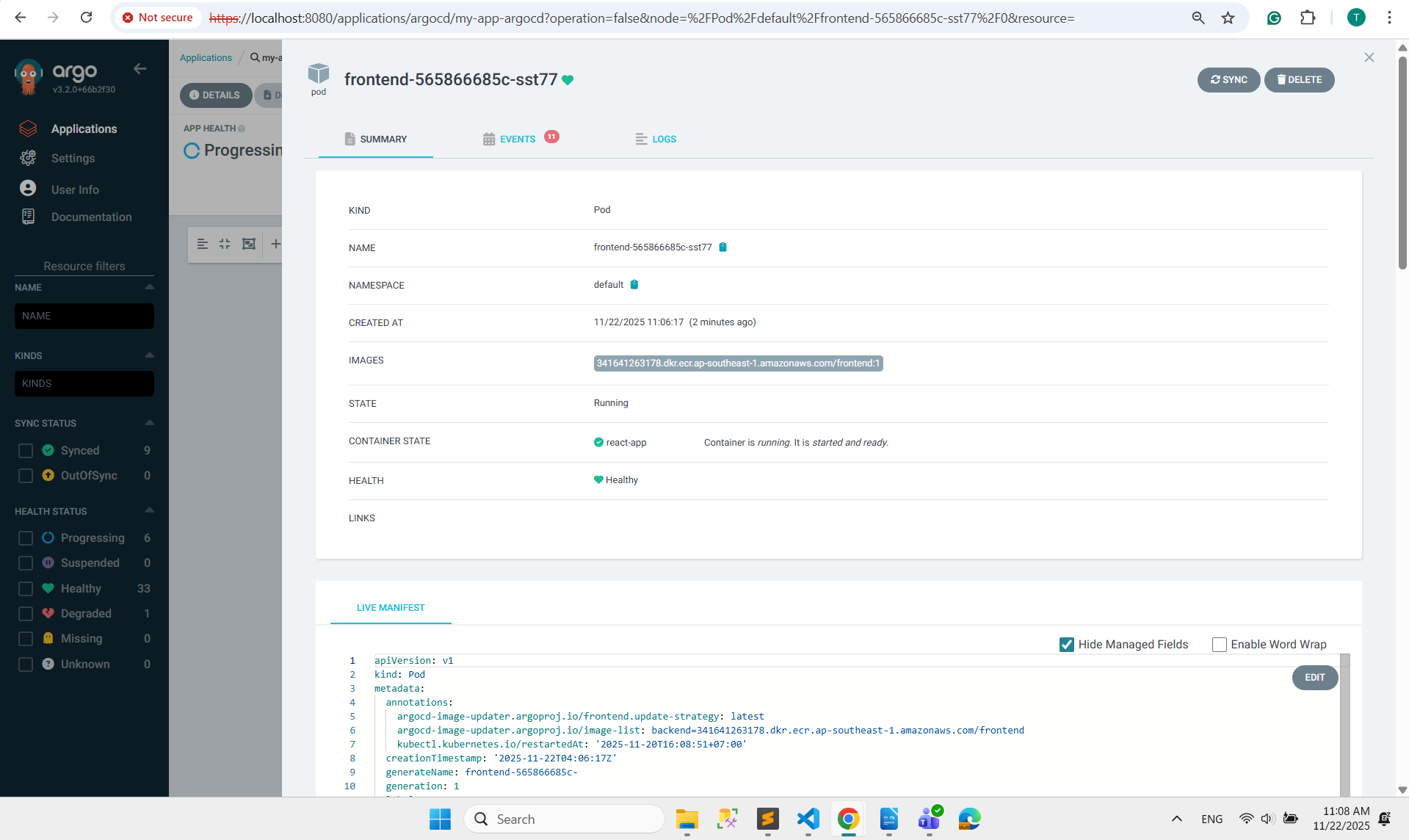
**- Jenkins CI is building and pushing the frontend image into ECR**



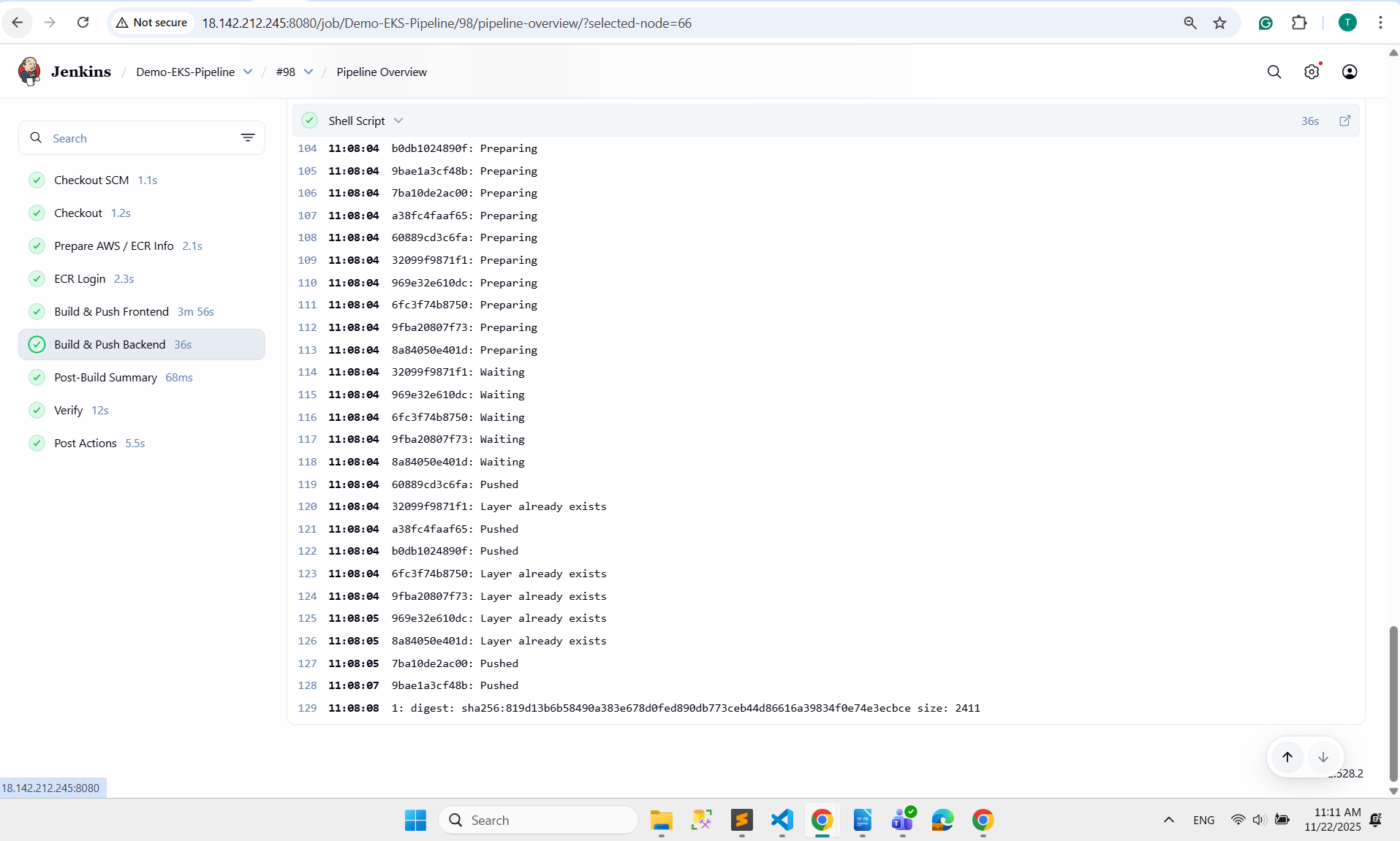
**- In ArgoCD UI, check the SYNC STATUS => Synced**



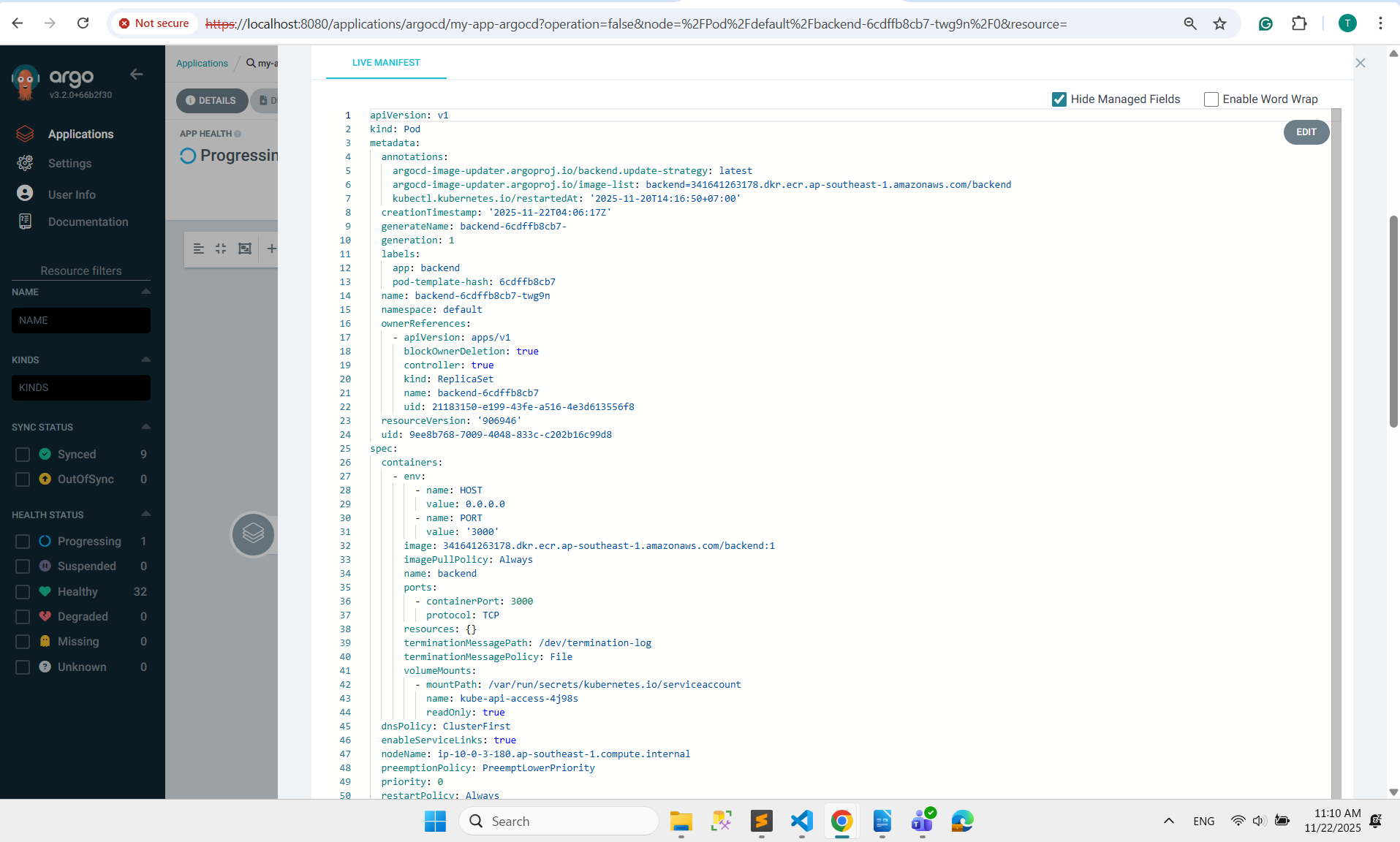
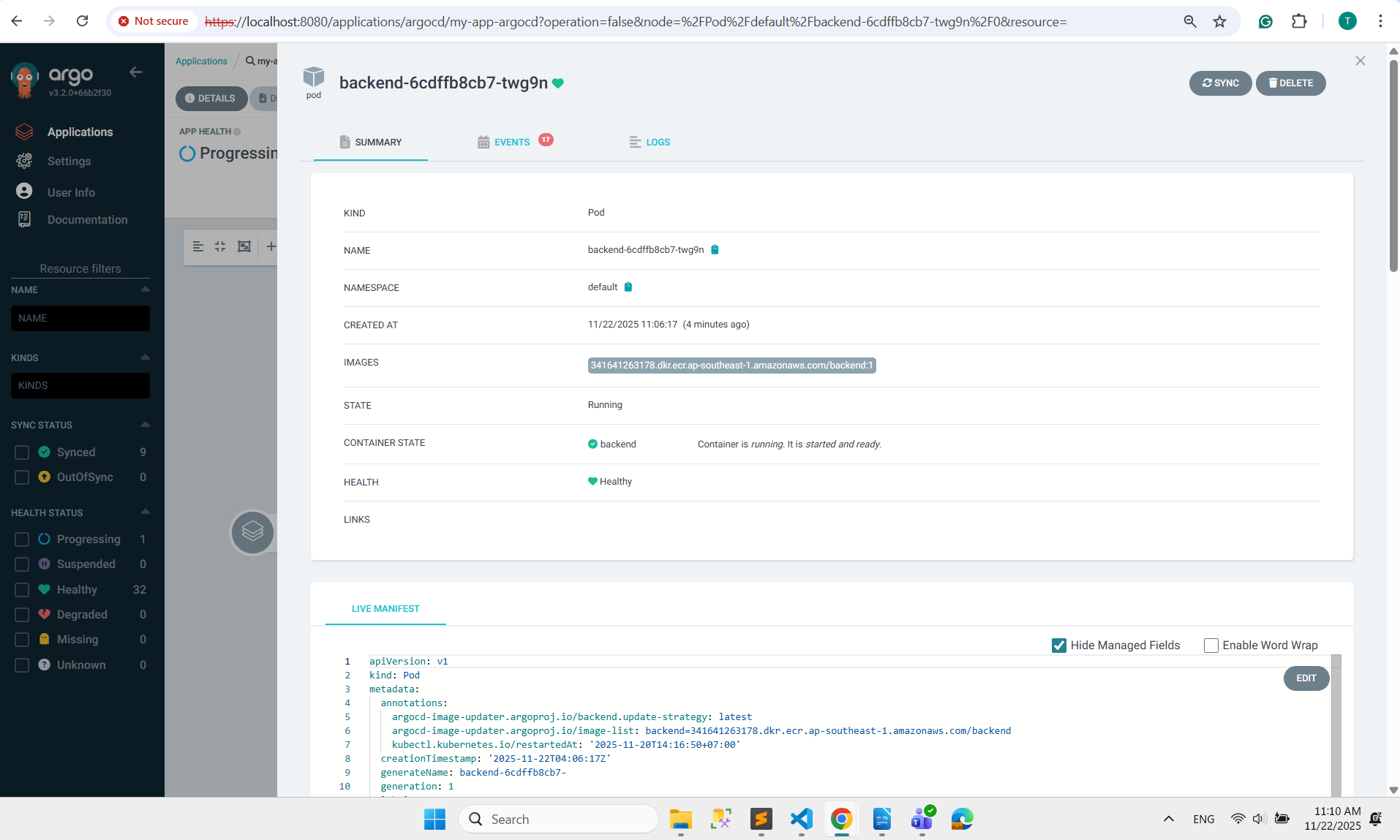
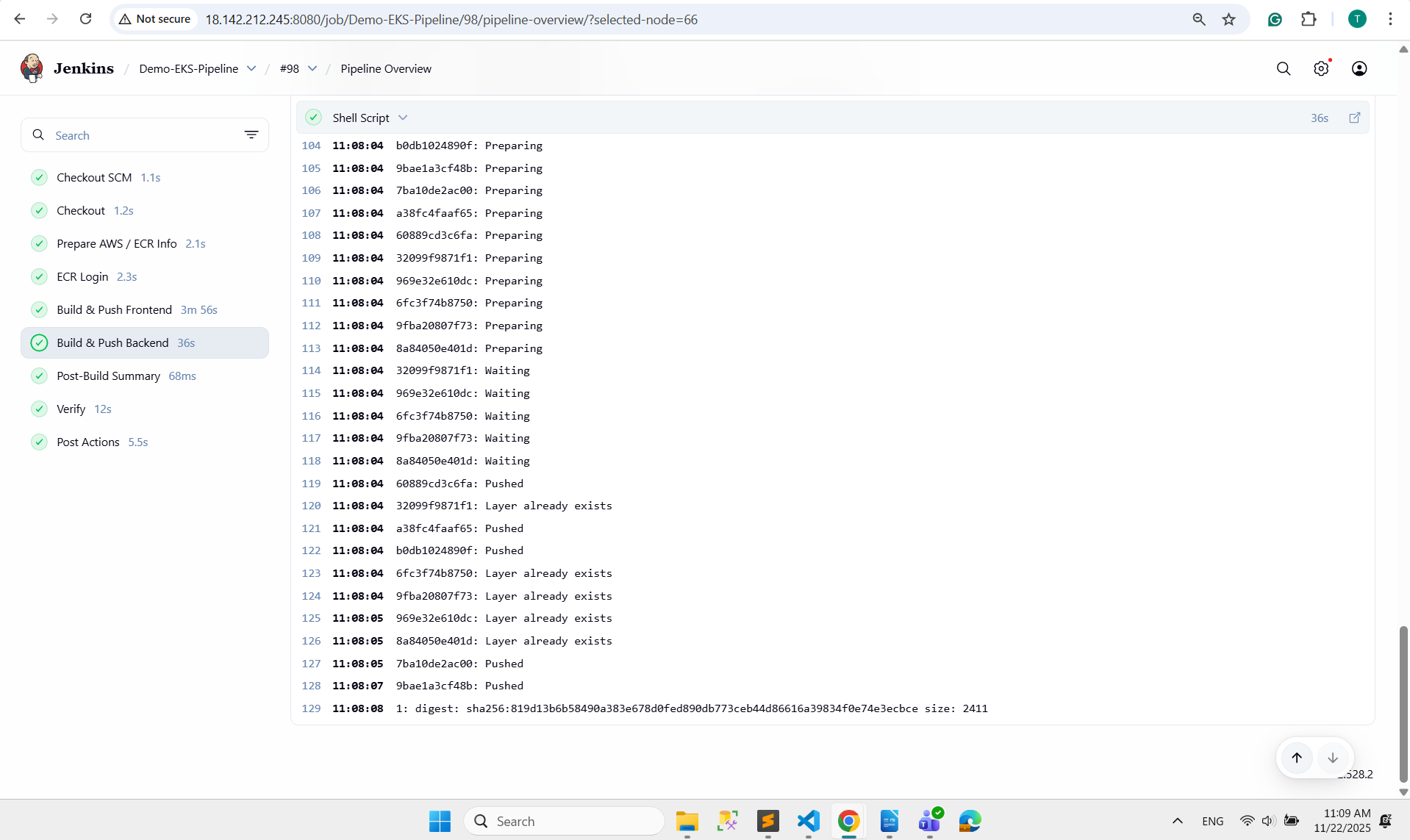
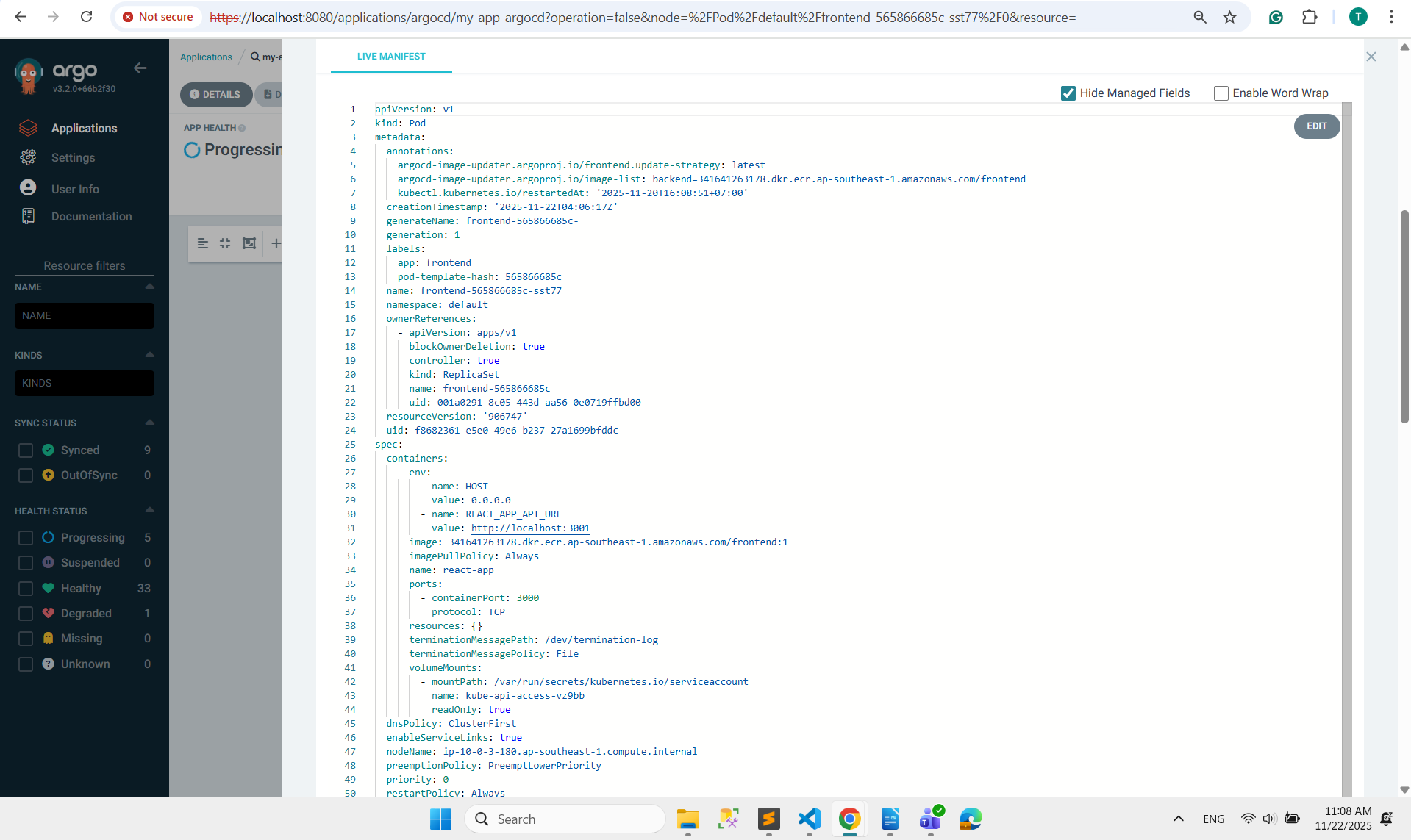
**- In ArgoCD UI, check the frontend pod => Updated the image tag to 1**



**- Jenkins CI is building and pushing the frontend image into ECR**

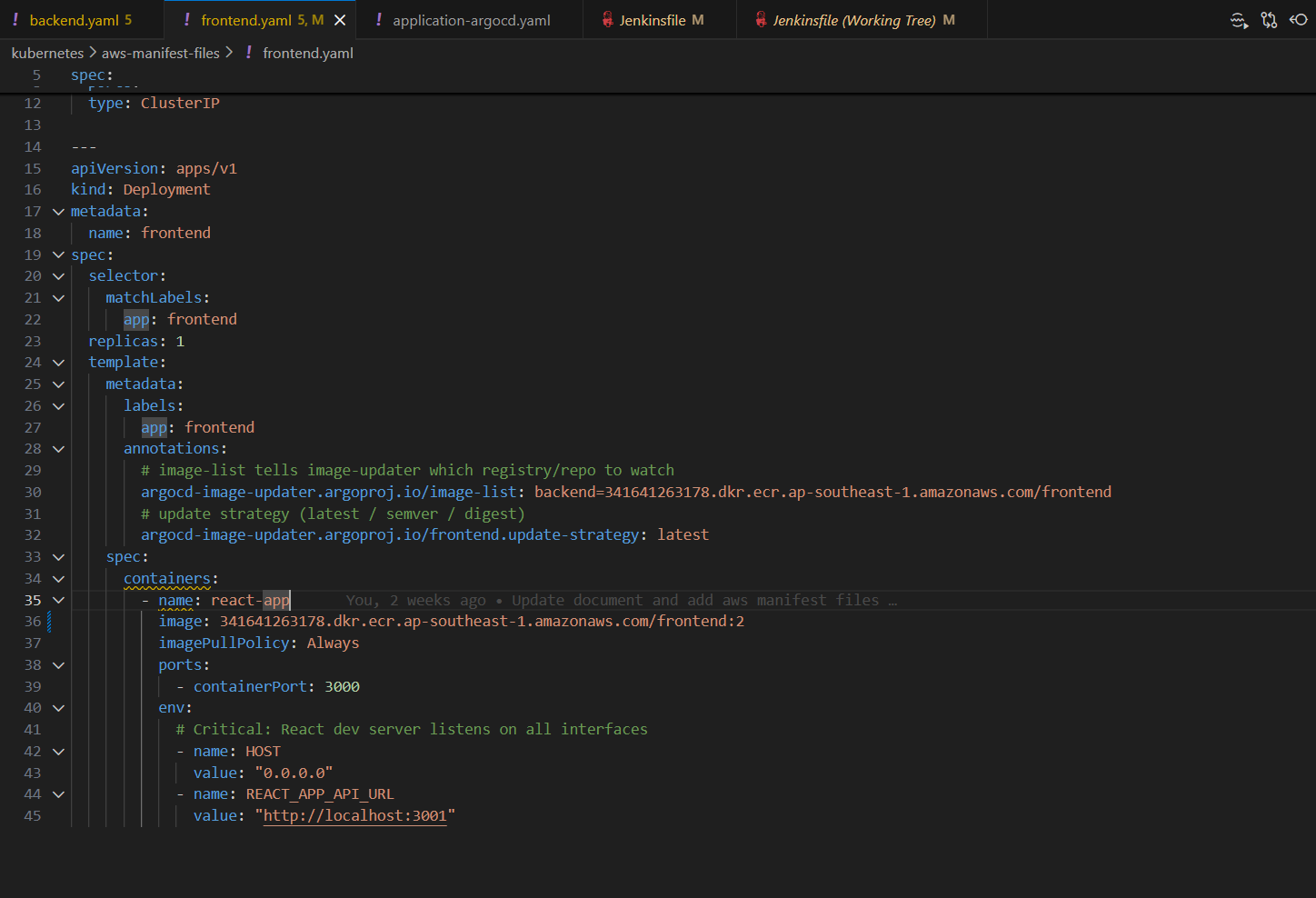


**- In ArgoCD UI, check the backendW pod => Updated the image tag to 1**

****

**- Update the image tags in the frontend and backend manifest:**

**Update image tag of frontend: 1 => 2**



**Update image tag of backend: 1 => 2**

