**ARGOCD SETUP IN KUBEADM CLUSTER**

**KUBEADM CLUSTER:**

Create 3 ubuntu instances of t2.medium and allow all traffic in security group (1 master, 2 workers)

1. Execute the following commands in all three instance

**(Installing docker)**

sudo su

apt update -y

apt install docker.io -y

systemctl start docker

systemctl enable docker

**(Setting up Kubernetes package repository)**

curl -fsSL "https://packages.cloud.google.com/apt/doc/apt-key.gpg" | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/kubernetes-archive-keyring.gpg

echo 'deb https://packages.cloud.google.com/apt kubernetes-xenial main' > /etc/apt/sources.list.d/kubernetes.list

**(Installing kubeadm kubectl and kubelet)**

apt update -y

apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y

1. Execute these commands in master node

sudo su

kubeadm init

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

kubectl apply -f <https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml>

kubeadm token create --print-join-command

1. Execute the following on worker nodes

sudo su

kubeadm reset pre-flight checks

(Paste the join command u got on master node while doing kubeadm init and append `--v=5` at end)

Eg: kubeadm join 172.31.50.112:6443 --token chb0ow.tc72kloffwsjfdr2 --discovery-token-ca-cert-hash sha256:84e581ca7e5442efda6af52bb01ecce1910f74bce5ab5e1b0d1960417363529d --v=5

4) In master node

kubectl get nodes (To check if the worker nodes are

conected to master node)

**ARGOCD SETUP:**

**(Installing Argocd)**

kubectl create namespace argocd (creating dedicated namespace for argocd)

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

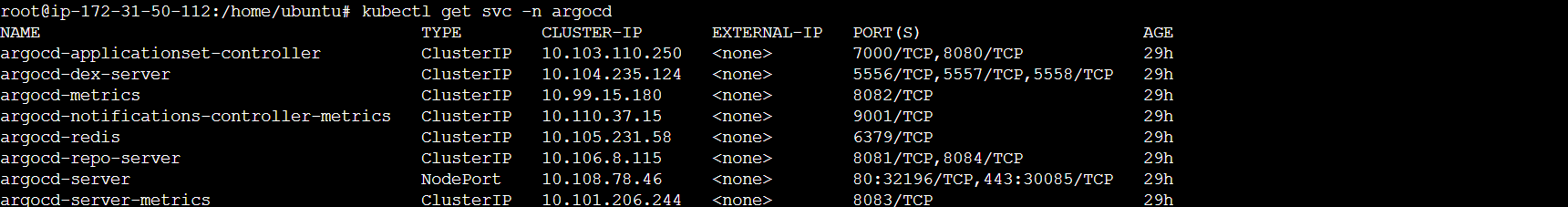
kubectl get pods -n argocd -w (lists the pods that are installed inside that argocd namespace)

kubectl get svc -n argocd

kubectl edit svc argocd-server -n argocd

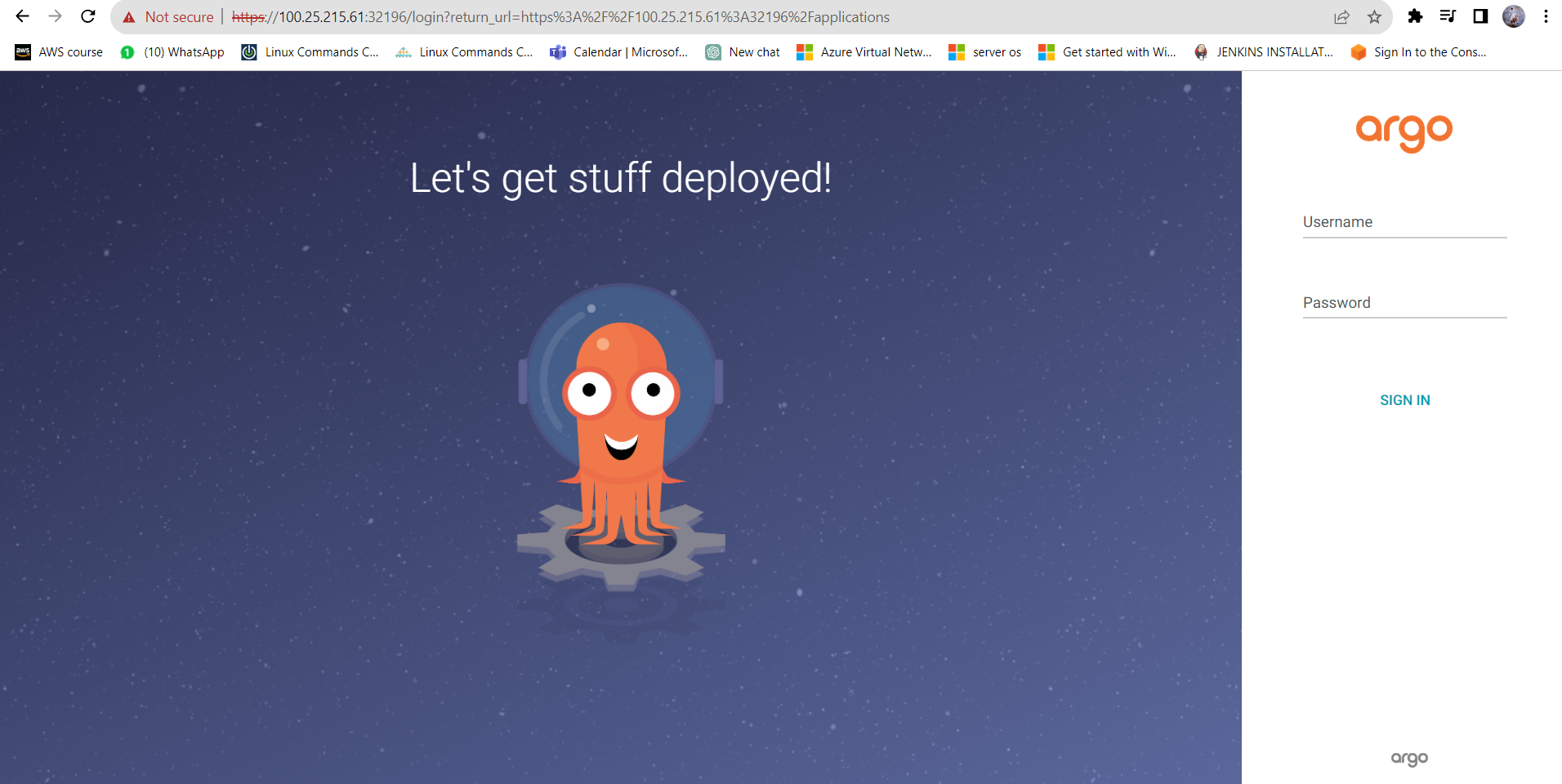
(We should edit the service type as nodeport so that we can access the argocd UI from our browser)

kubectl get svc -n argocd



**(ArgoCD Dashboard)**

Give the public IP along with nodeport provided to access Argocd UI



username : admin

password : run this command (kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d )

Once you login,

**i) In case your manifest files are present in public repository, follow these steps:**

1) Click on New App

2) Give any name for your application

3) Give any name for your project or default

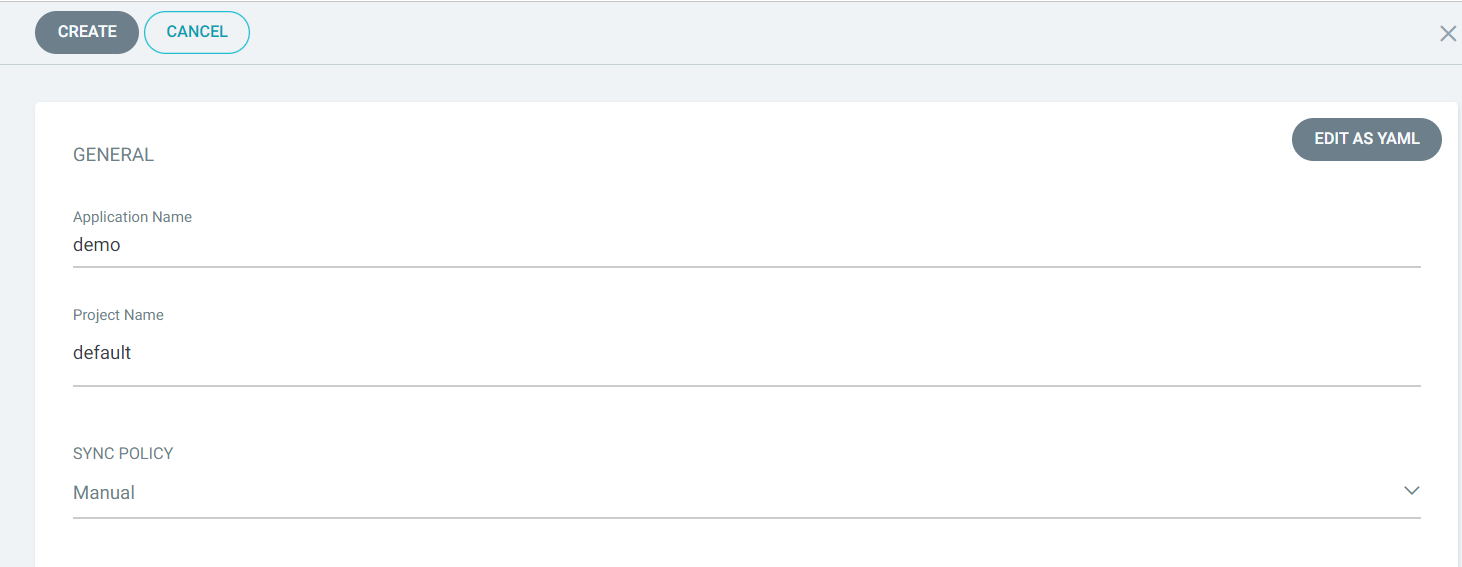
4) Give your Github Repository url

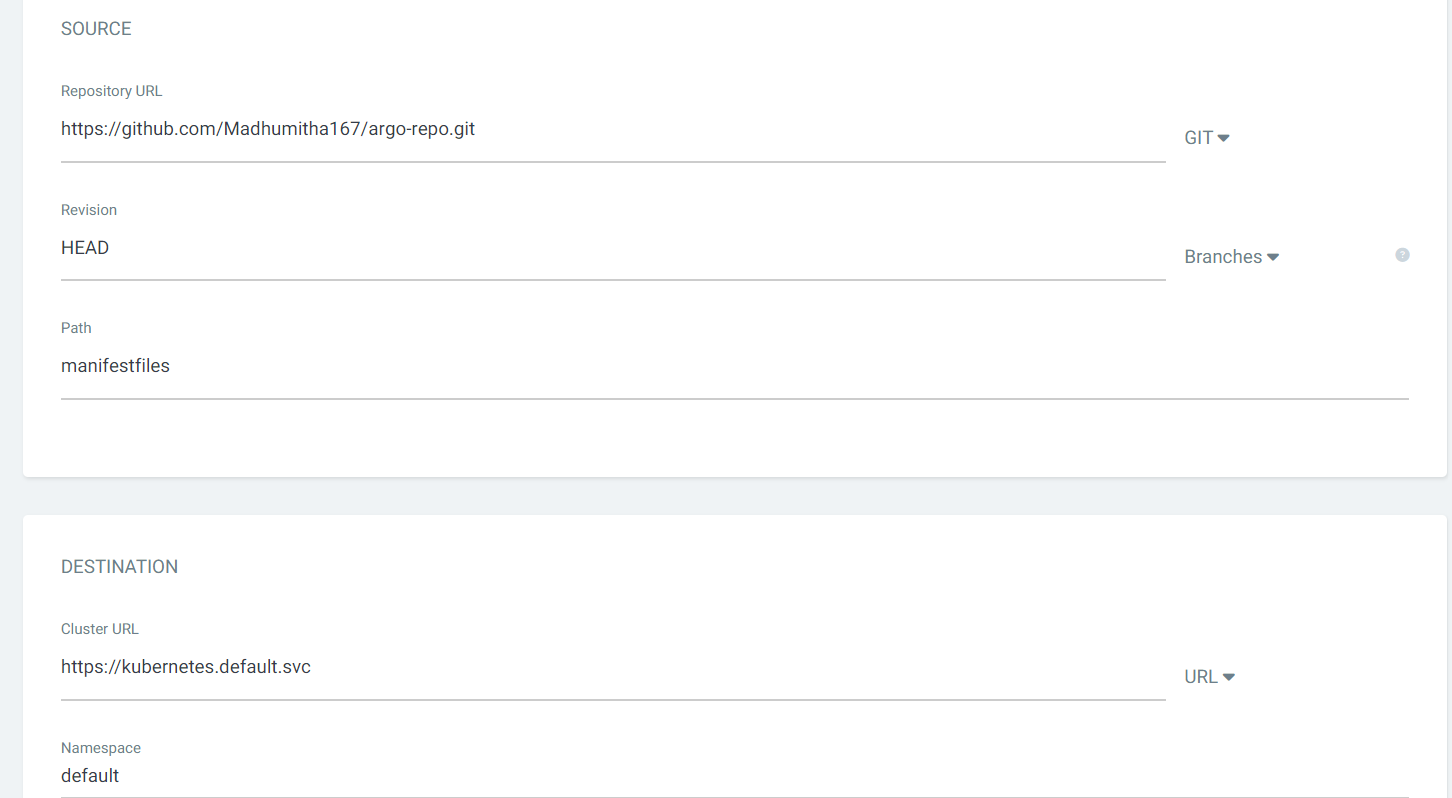
5) Give the path of your source files(folder in which the manifest files are located inside repo)

6) Give any name for your namespace / default

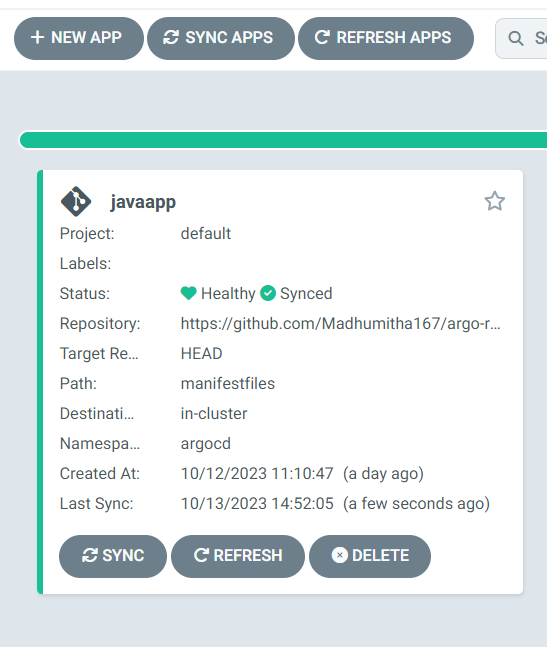
7) Give the url of your cluster (for current cluster select the url popup)

8) Click on create (Argocd will refer the github repository and deploy the application in the cluster)





After it gets created, click on sync

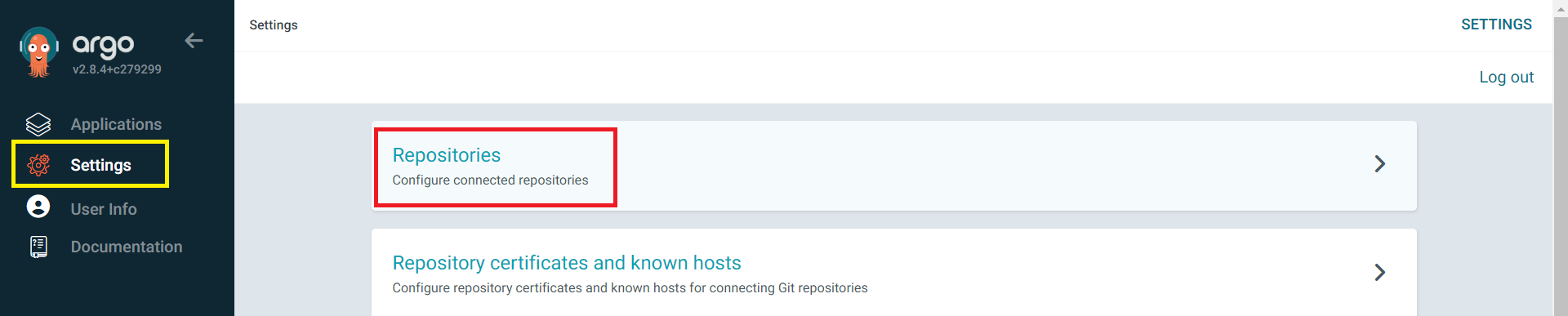


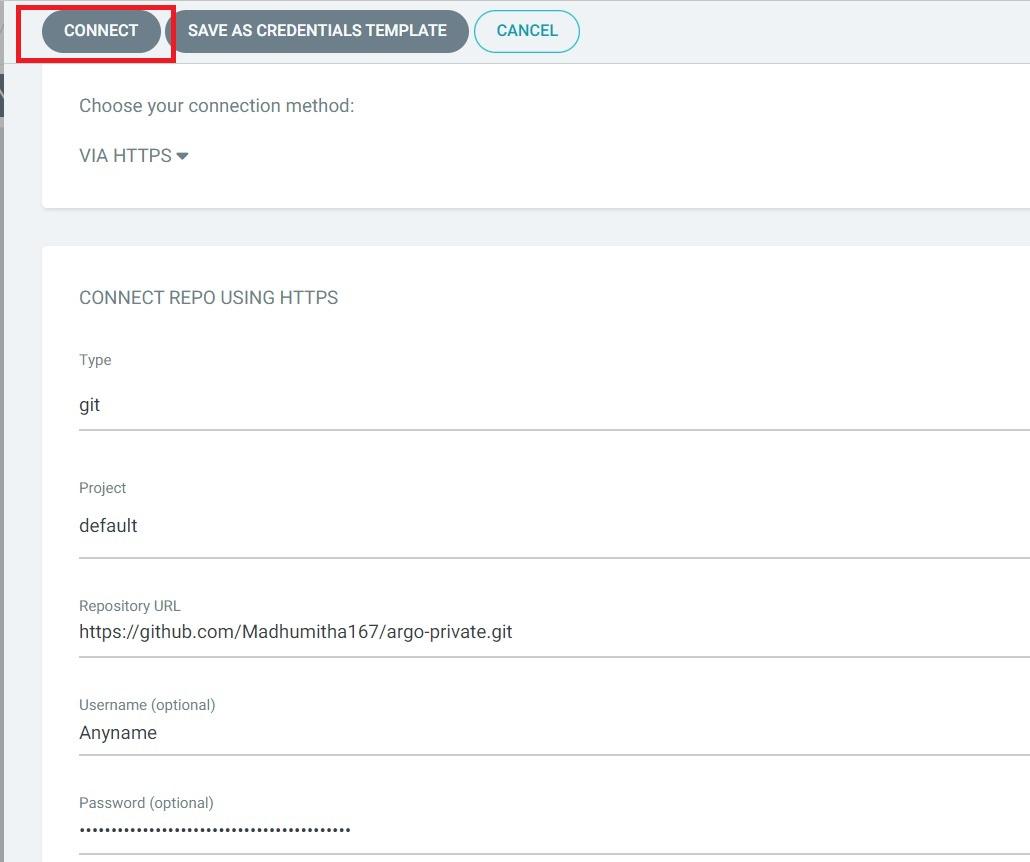
**ii) In case your manifest files are present in private repository, follow these steps:**

1)Generate token for your Github account

(Profile → Settings → Developer Settings → Personal Access Token → Token(classic) → Generate new Token)

2) Go to ArgoCD UI and click on settings then click on Repositories



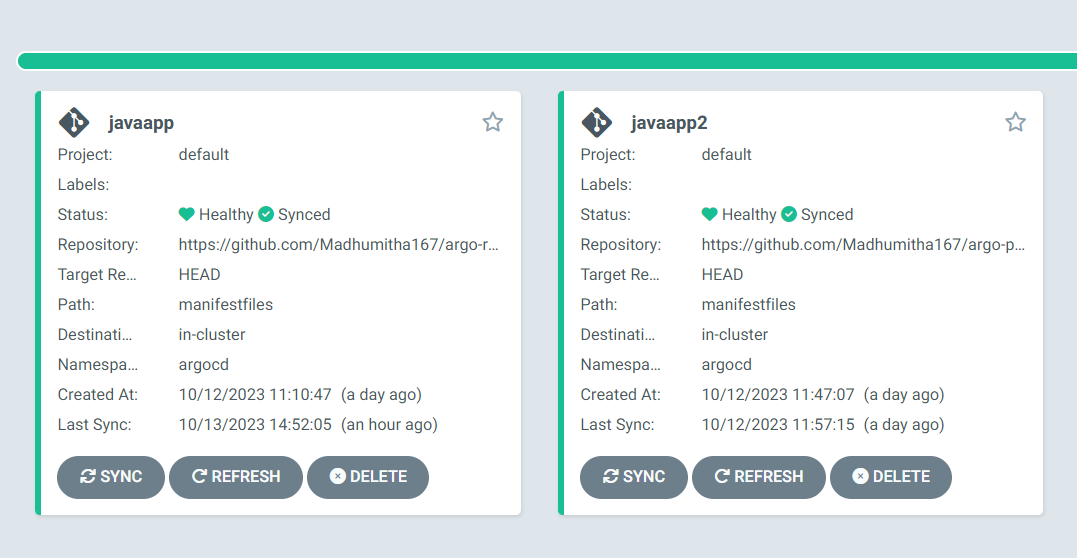
3) Click on Connect Repo →Choose your connection method ( We have used Https) → Give your private Repo url → Username: Can give any name → Password: Github token →Then click on connect 

Once your repository is connected , you can see the status as Successful



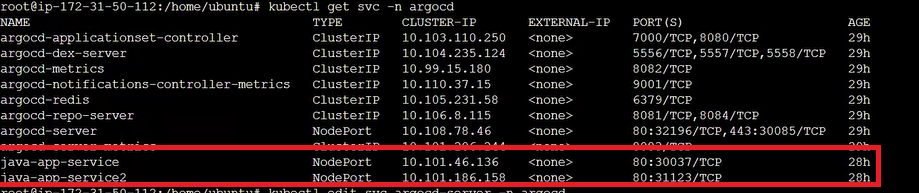
Now, you can click on the three dots in right → Create Application and **follow the same steps you followed for public repository .**

These are the two applications deployed by argocd



Now, go to master node and give

kubectl get svc -n argocd



Now, we can access our applications using

Public\_IP:NodePort

