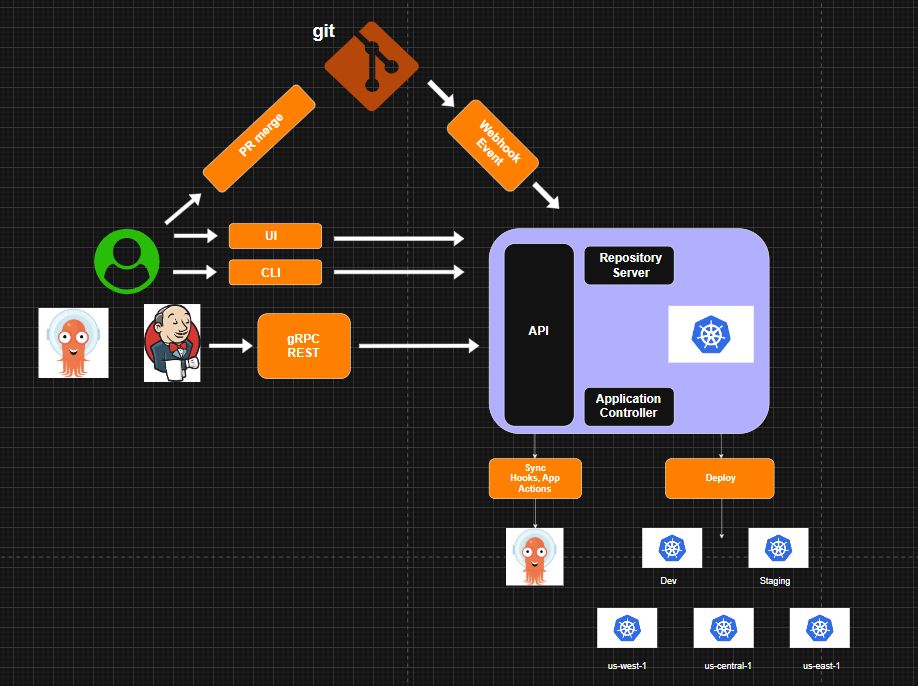
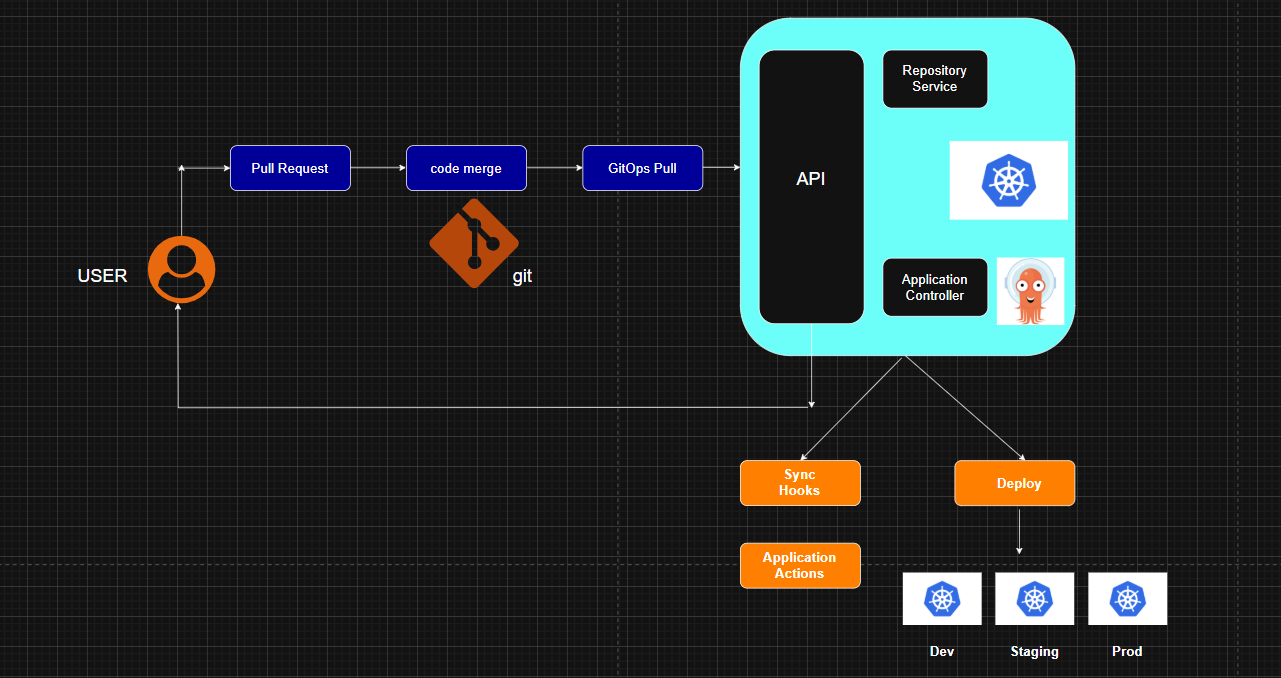
**ARGOCD ARCHITECTURE:**





**Argocd:-**

**What is Argo CD?**

1) It is a declarative, GitOps-based continuous delivery tool for K8s.

2) It will synchronize the desired state (stored in git) with actual state (in K8s).

3) In short:- Git is the source of the truth

4) Argocd ensures the cluster matches it

**Why Argo CD?**

1) Traditional CD tools push deployments manually or via pipeline

2) Argocd automates this by pulling desired manifests from Git.

**Benefits:**

1) GitOps model: single source of the truth

2) Declarative: infra > app as code

3) Visibility: Web UI & CLI to see real-time app health

4) Multi-cluster support: manage multiple clusters from one control

5) Rollbacks:- just revert git commit

**Argocd Architecture:-**

1. API server: It will handle CLI/UI requests and expose REST

2. Repository server: It will connect to the Git/Helm repo, and it will pull manifest files

3. Application controller: It will reconcile the desired state (git) vs the live state (k8s cluster)

4. Redis: caching for performance

5. Web UI:- visualization of app state

**Flow:-**

Dev > push code/manifest > git repo > argod > k8s cluster (sync)

**Core concepts:**

1. Application: a CRD (custom resource definition) representing what to deploy and where

2. Project:- logical grouping of applications

3. Sync—it will bring the live cluster state inline with git

4. Hooks:- presyns, sync, postsync hooks for automation

5. Sync status: synced and out of sync

**Deployment strategies:**

1. Manual sync: Admin triggers sync manually

2. Auto sync: Argo CD automatically applies changes from git

3. Self-heal: if someone changes in the clusters, Argo CD reverts it to match Git.

**Some importance commands of argocd :-**

**Start**

**Install ArgoCd :**

**kubectl create namespace argocd**

**kubectl apply -n argocd -f** <https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml>

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

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

**rm argocd-linux-amd64**

**Change the argocd-server service type to LoadBalancer:**

**kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'**

**After a short wait, your cloud provider will assign an external IP address to the service. You can retrieve this IP with:**

**kubectl get svc argocd-server -n argocd -o=jsonpath='{.status.loadBalancer.ingress[0].ip}'**

**For External ip address**

**kubectl get all -n argocd**

**Login Using The CLI**

**argocd admin initial-password -n argocd**

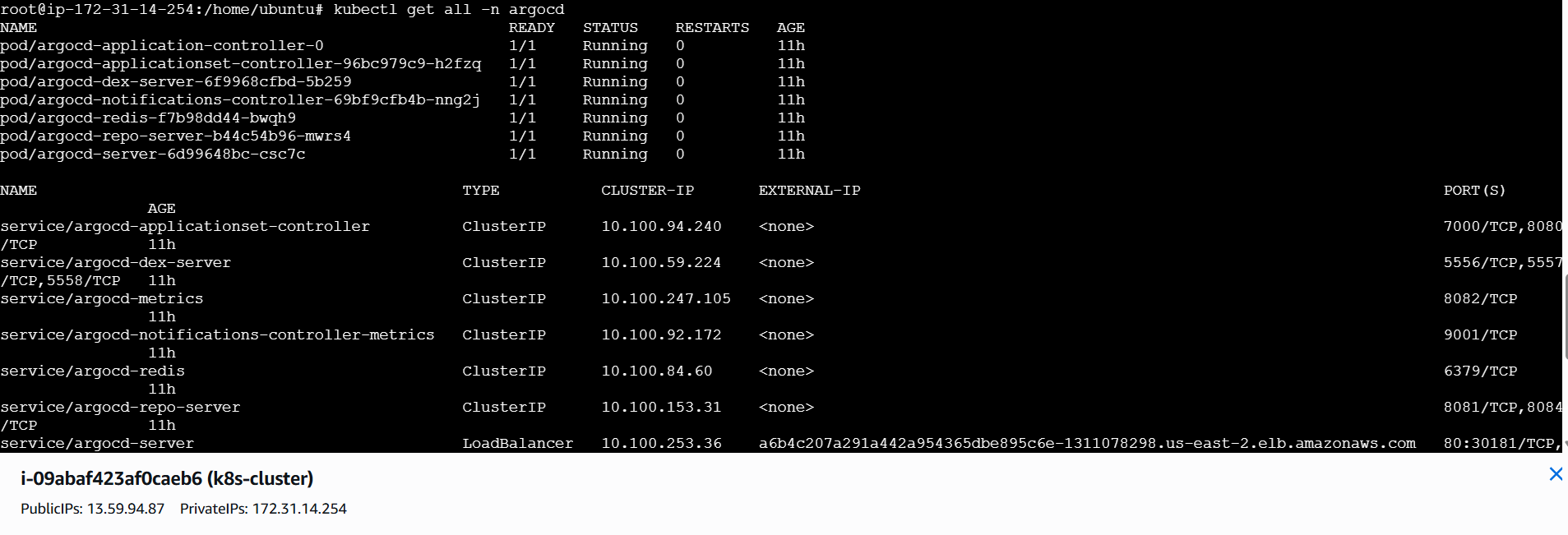
**Using the username admin and the password from above, login to Argo CD's IP or hostname:**

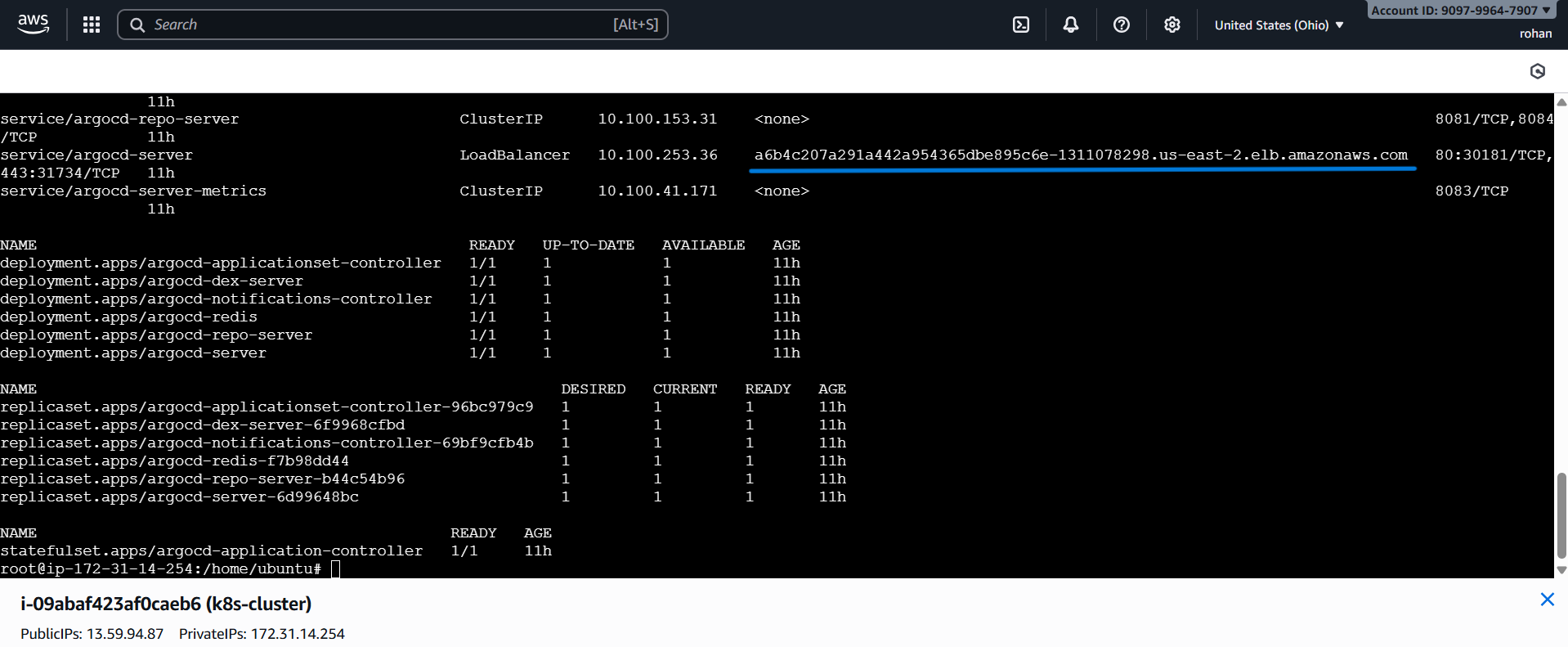
**argocd login** [**a6b4c207a291a442a954365dbe895c6e-1311078298.us-east-2.elb.amazonaws.com**](http://a6b4c207a291a442a954365dbe895c6e-1311078298.us-east-2.elb.amazonaws.com)

**Change the password using the command:**

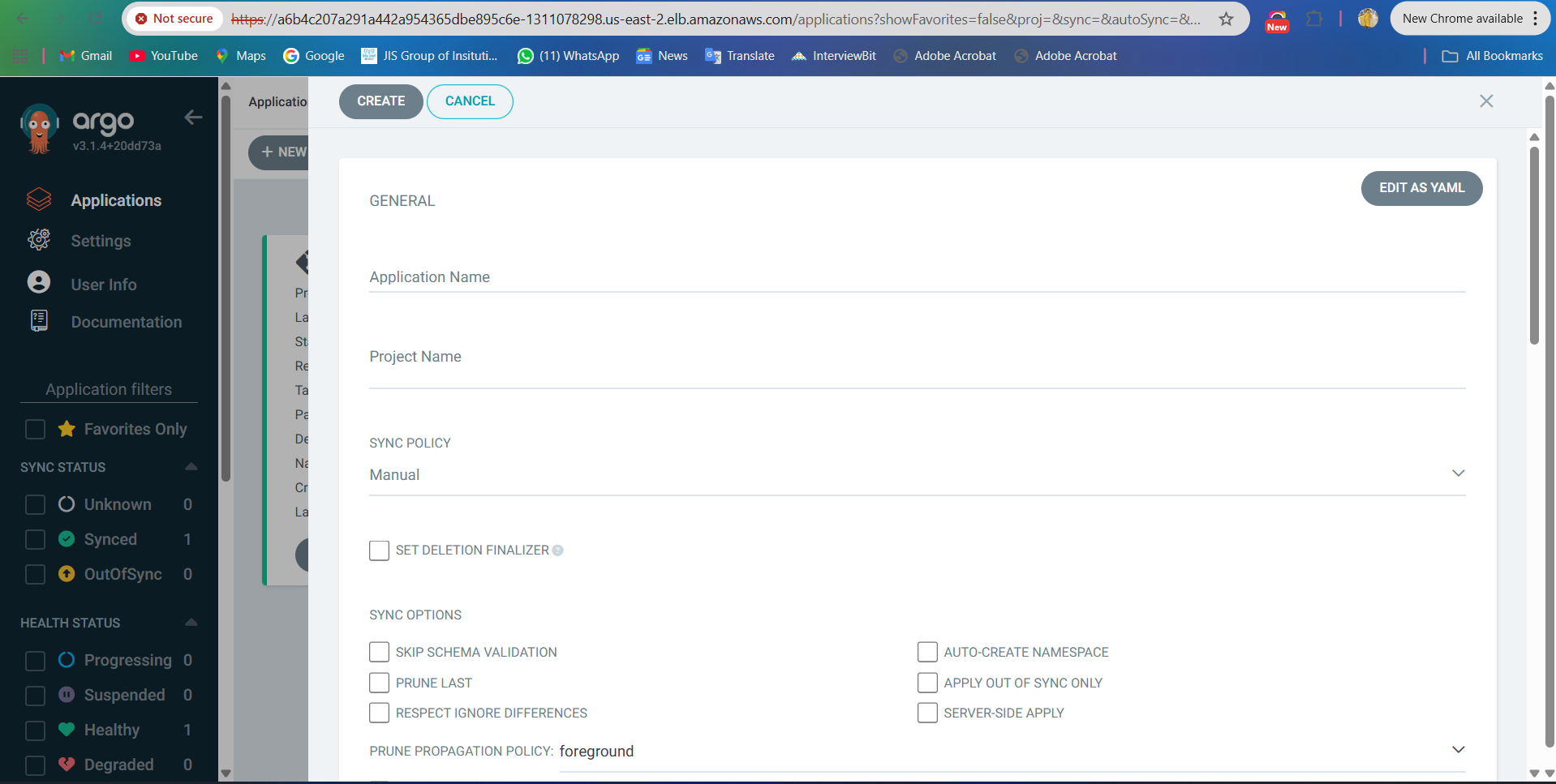
**argocd account update-password**

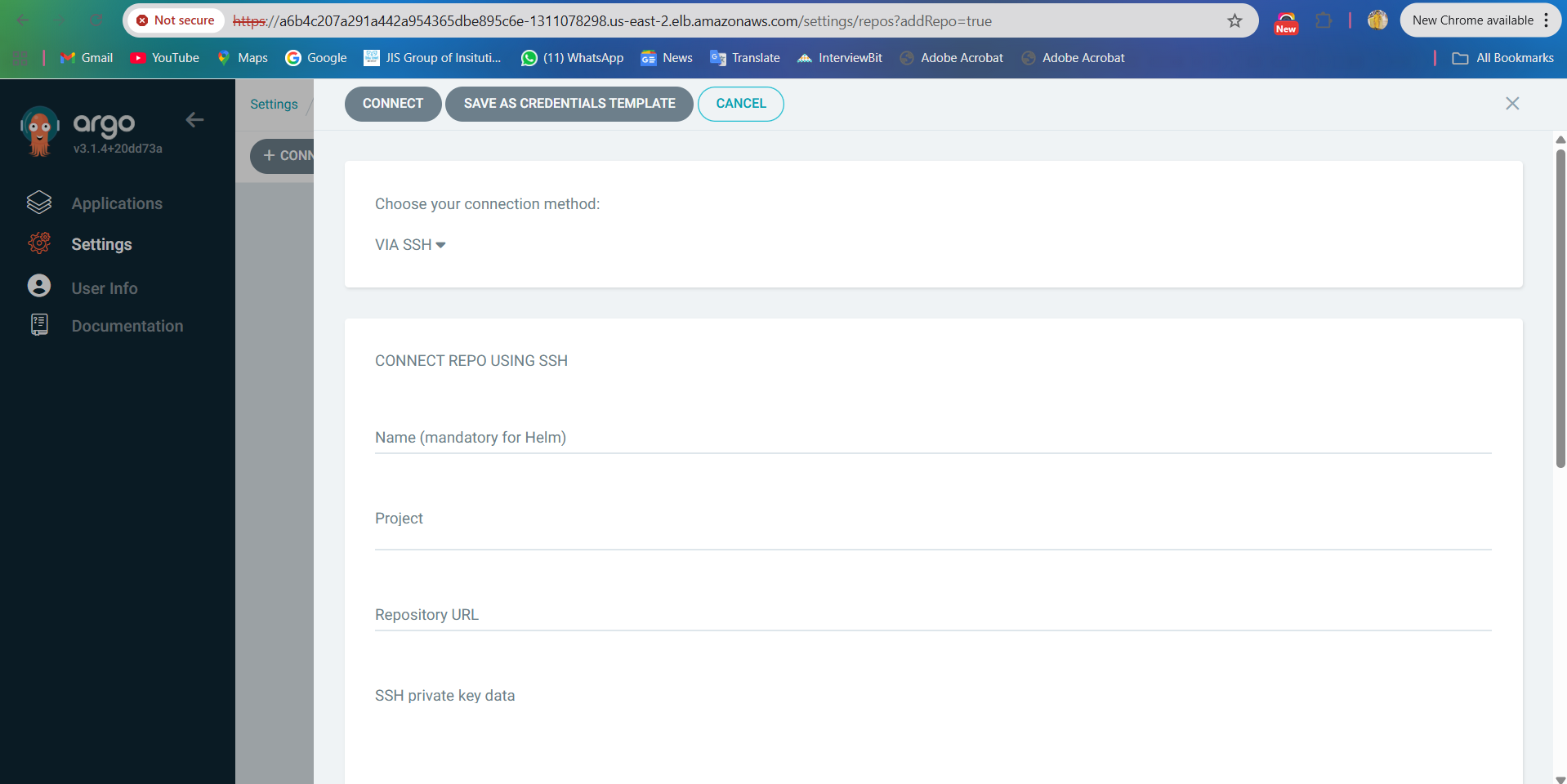
**Kubectl get all -n argocd**



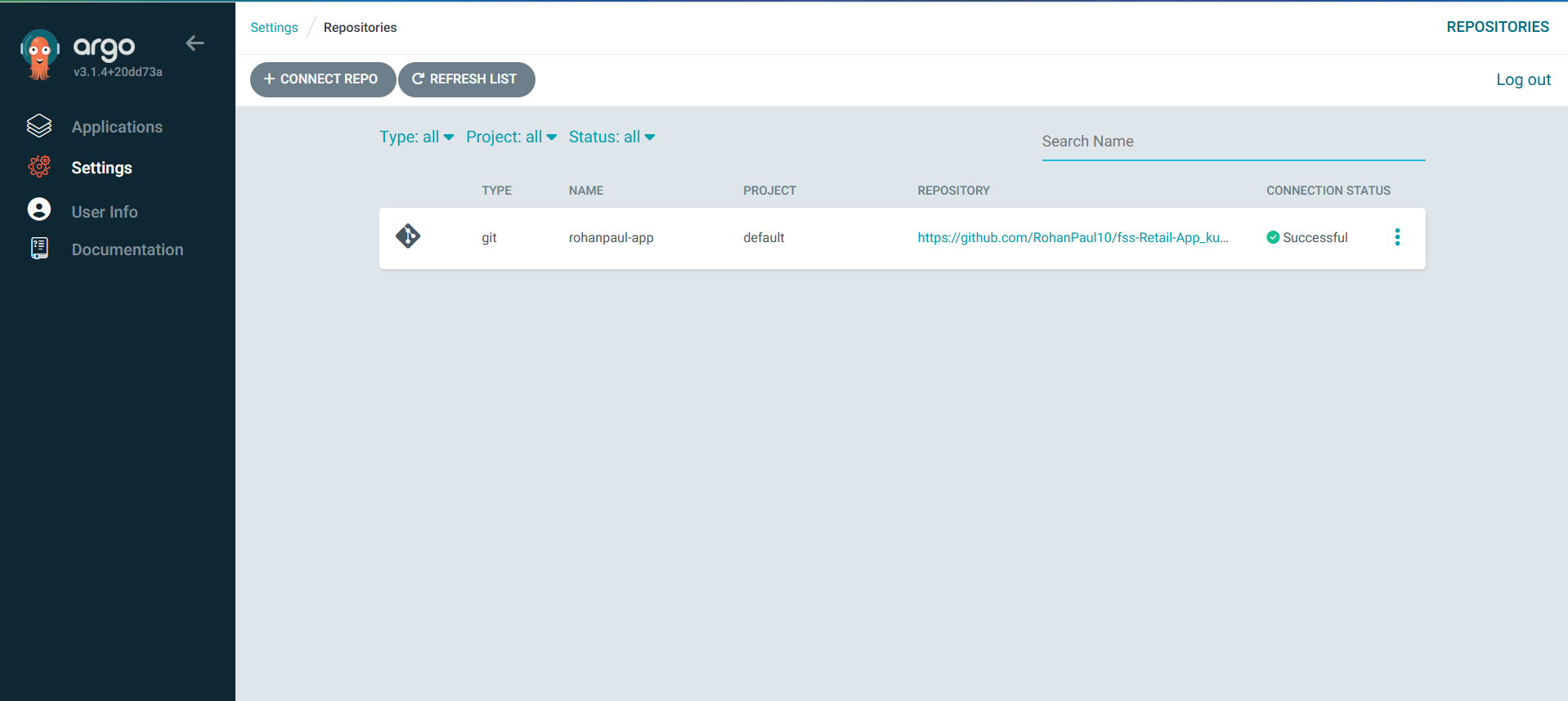


**Give your app the name rohan-app, use the project default, and leave the sync policy as Auto:**



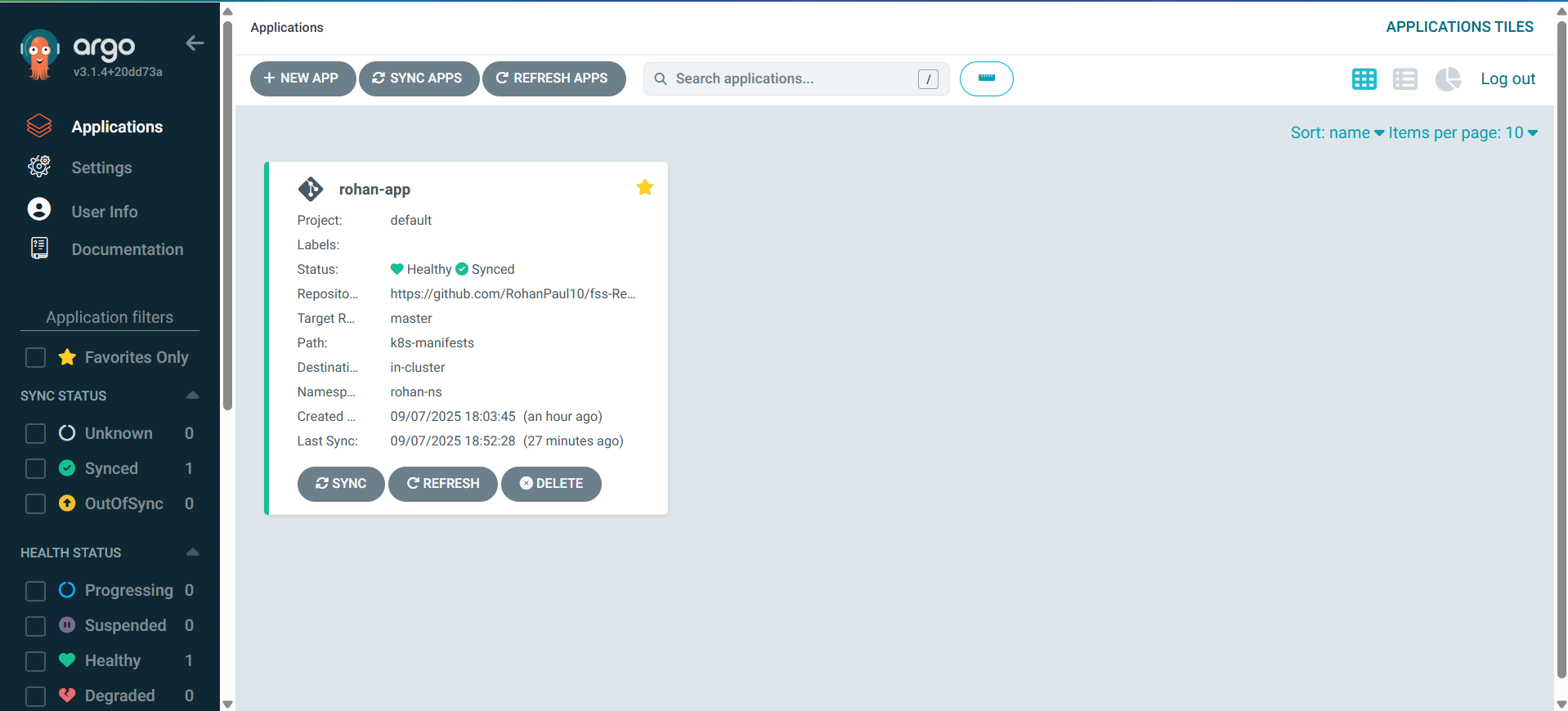


**Repository:-**



**Syncing via UI**

**On the Applications page, click on Sync button of the rohan-app application:**



**A panel will be opened and then, click on Synchronize button**

**You can see more details by clicking at the rohan-app application :**