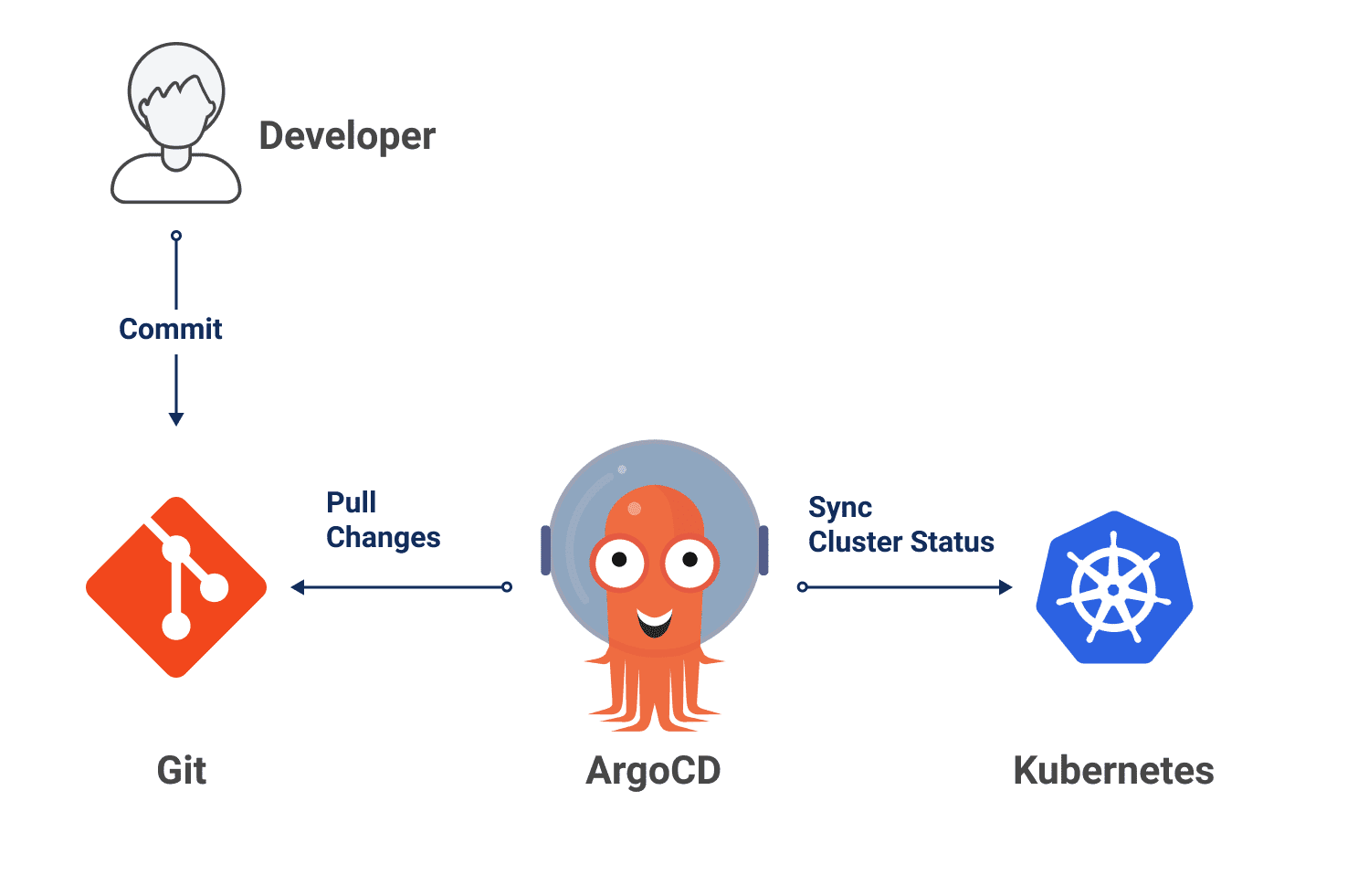
Argo CD Project

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes. It helps developers and DevOps teams to automate the deployment of applications and services to a Kubernetes cluster. With Argo CD, users can define the desired state of their applications and services in a declarative way using YAML manifests, and then Argo CD takes care of deploying and managing the actual resources on the Kubernetes cluster.

By integrating Argo CD with Kubernetes, you can streamline your application deployment process and reduce the risk of errors and downtime.

Argo CD also provides a web-based user interface and a CLI tool for managing and monitoring deployments. It can automatically detect changes in Git repositories and trigger new deployments. Additionally, it supports rollbacks and can automatically roll back to the previous version if a deployment fails.

“One of the key benefits of Argo CD is that it provides a single source of truth for the desired state of applications and services”. This makes it easier to manage and maintain complex, distributed systems on Kubernetes. Overall, Argo CD is a powerful tool for managing Kubernetes deployments with GitOps principles.

**Tools Involved**

**============**

Github , Git , Jenkins , Maven , Dockerhub , Docker , Tomcat , Kops , Kubectl , ArgoCD

S1:EC2 ubuntu 22.04 -(t2.medium)-> Git,Java Jenkins , Docker , kubectl , Kops

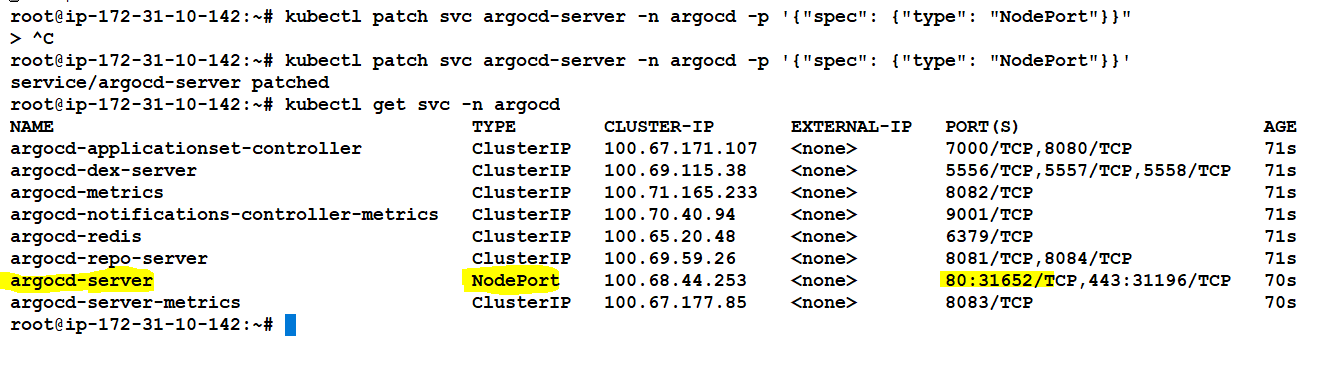
* **STEP -1 (Kubernetes Cluster Setup)**

Refer our kops installation document and create kubernetes clusters with 2 nodes

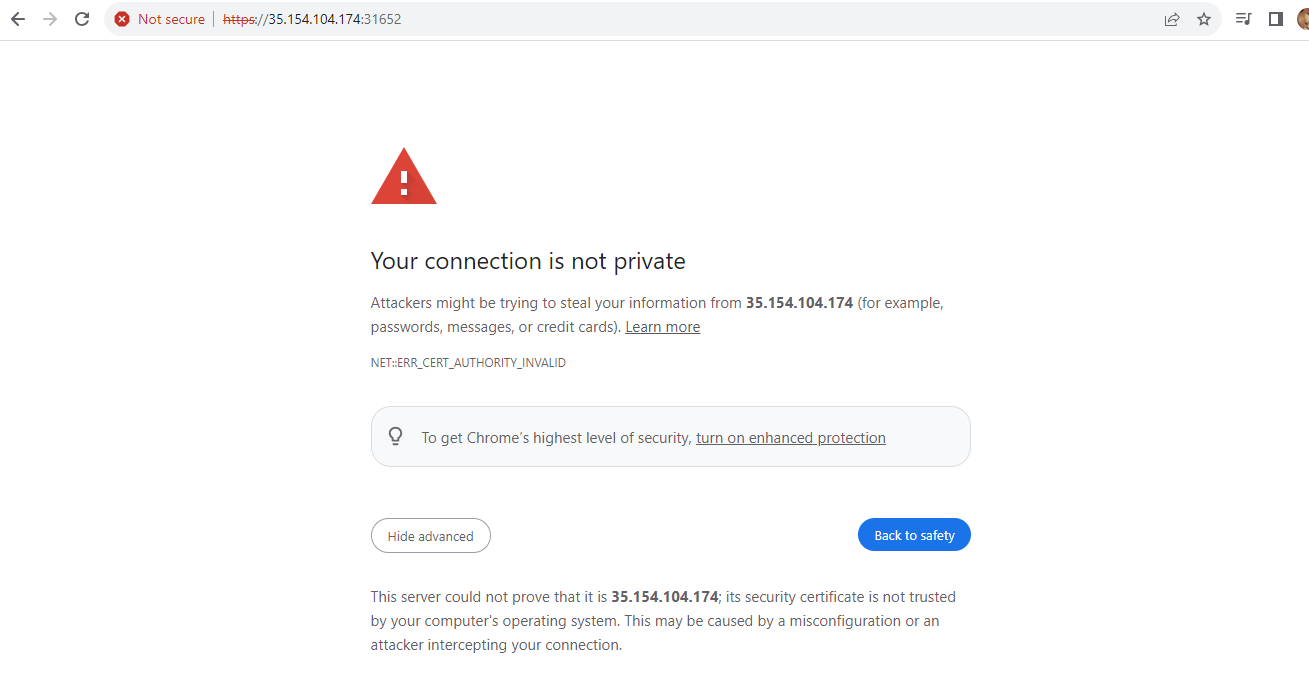
* **STEP -2 (Argo CD Cluster Set up in K8 Cluster)**

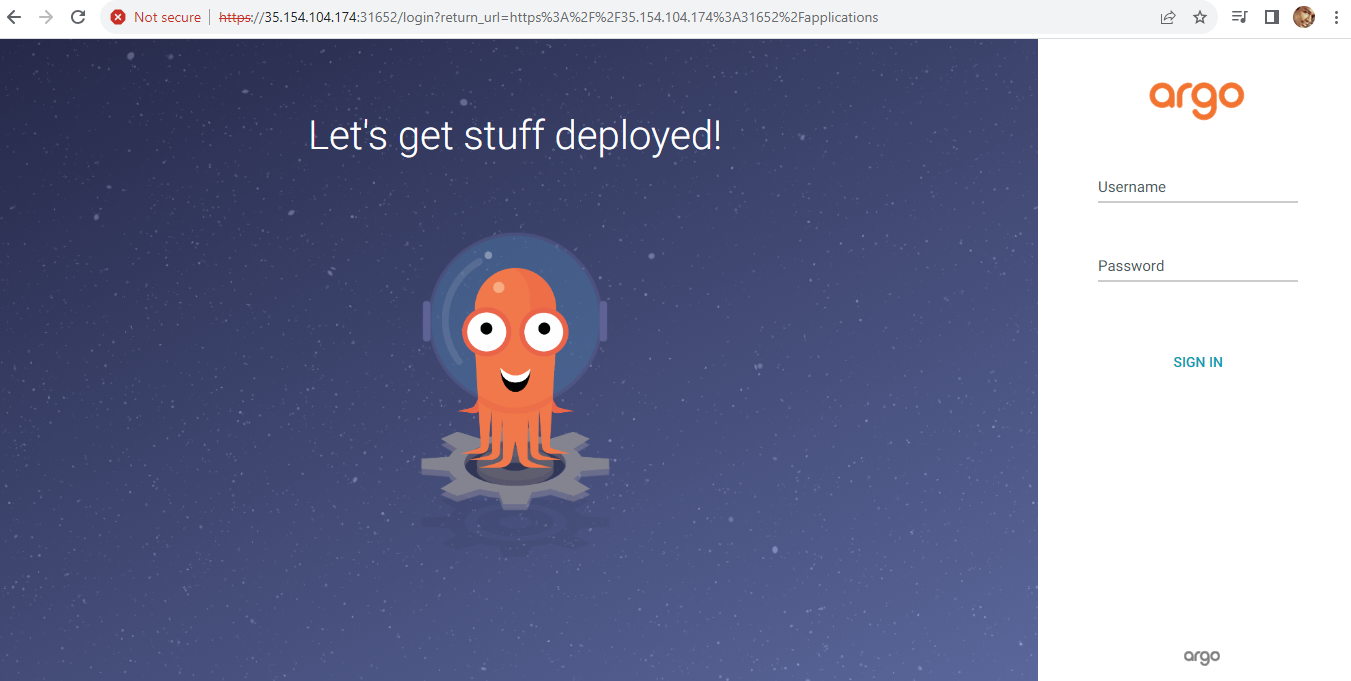
**# kubectl create namespace argocd**

**# 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)**# kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "NodePort"}}"# Kubectl get svc -n argocd**

****

**Copy the worker node ip with the nodeport hit in browser you will get this output**

**Click proceed to unsafe**

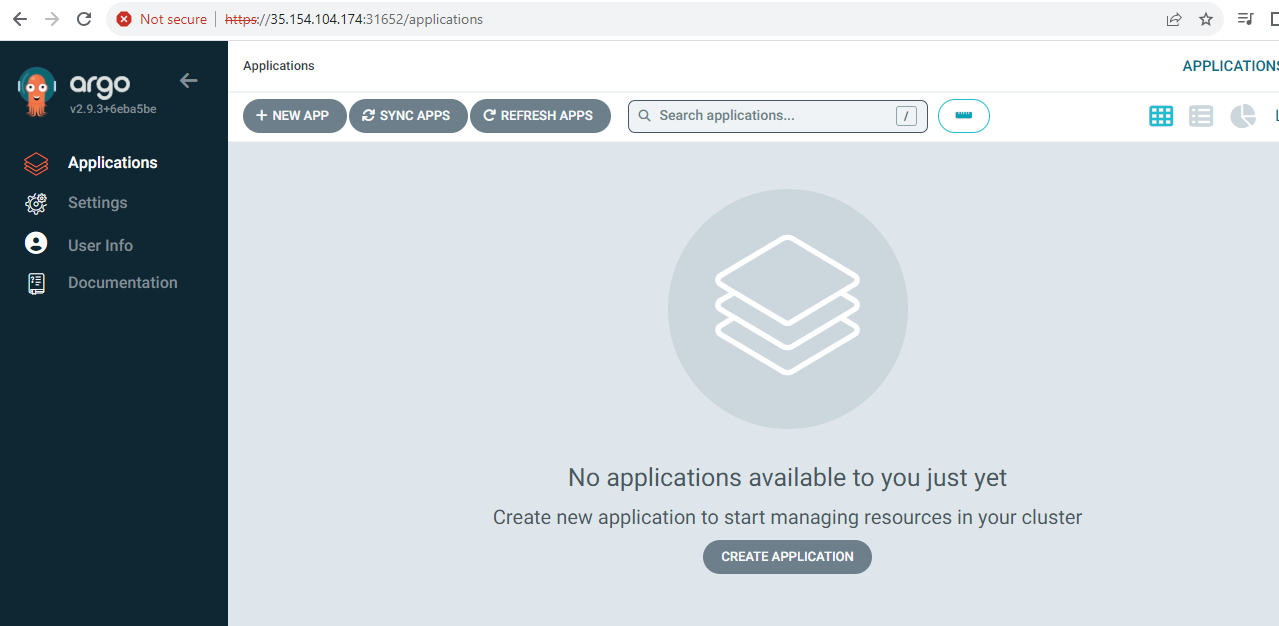
****

**User name is admin**

**Password :**

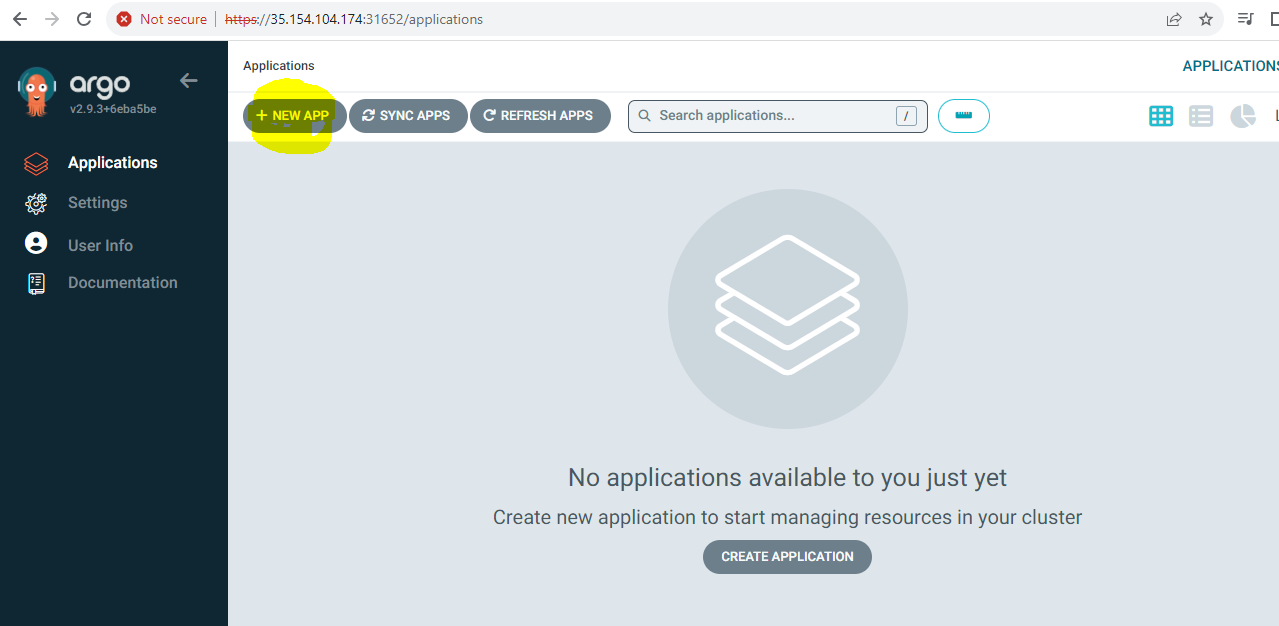
**# kubectl -n argocd get secret argocd-inital-admin-secret -o jsonpath="{.data.password}" | base64 -d**

**Like this you will get password → 2ZZLHe-B8AbzLpEp**

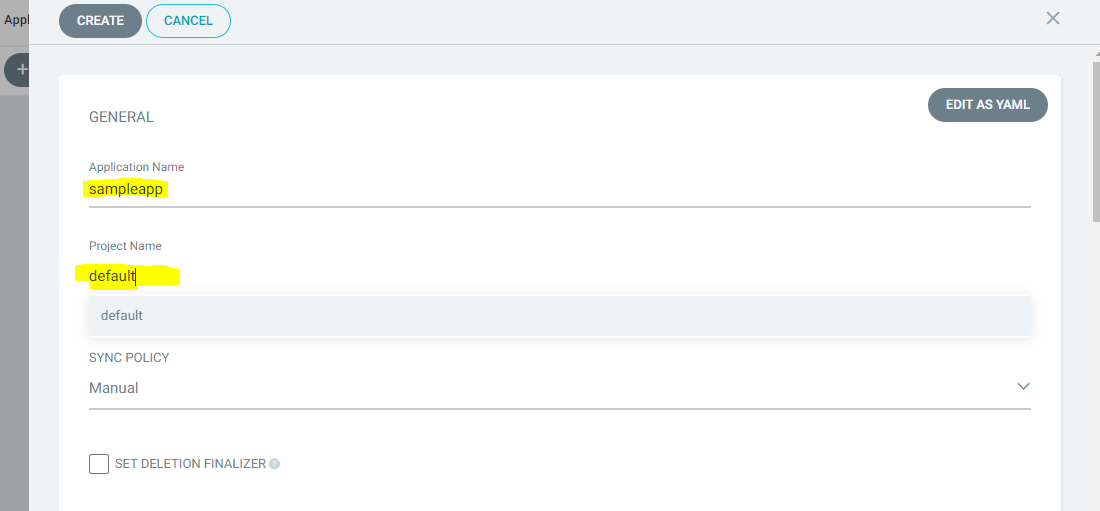
****

**Source : OFFICIAL DOCS →** [**https://argo-cd.readthedocs.io/en/stable/**](https://argo-cd.readthedocs.io/en/stable/)

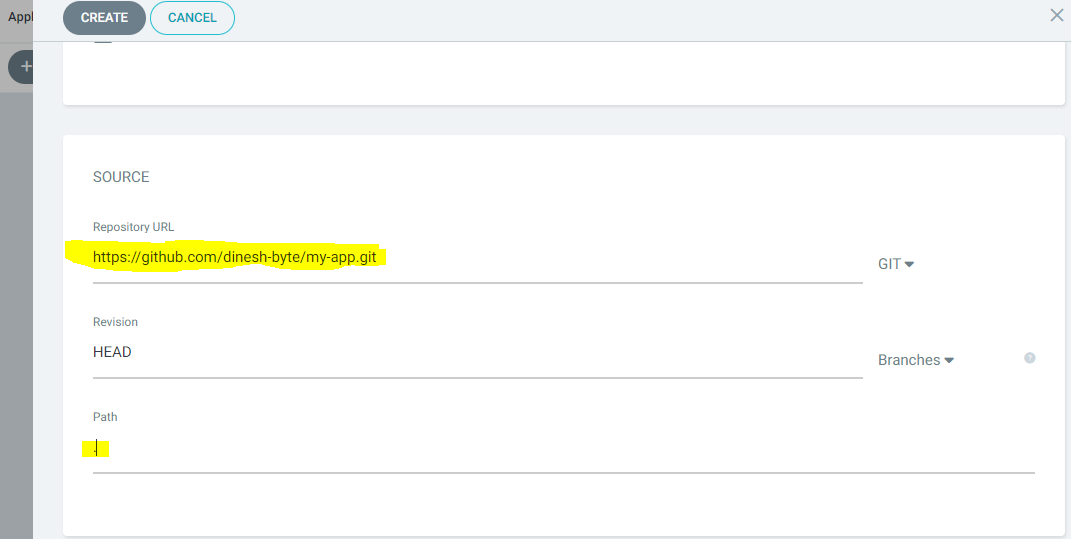
* **STEP -3 (Connecting Github Repo with Argo Cd)**

****

**Select the NEW APP OPTION**

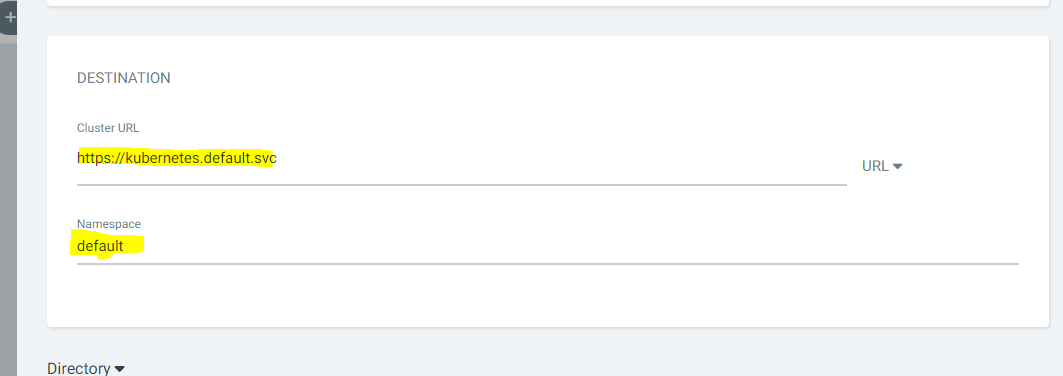
****

**Set application name and Project name as you like**

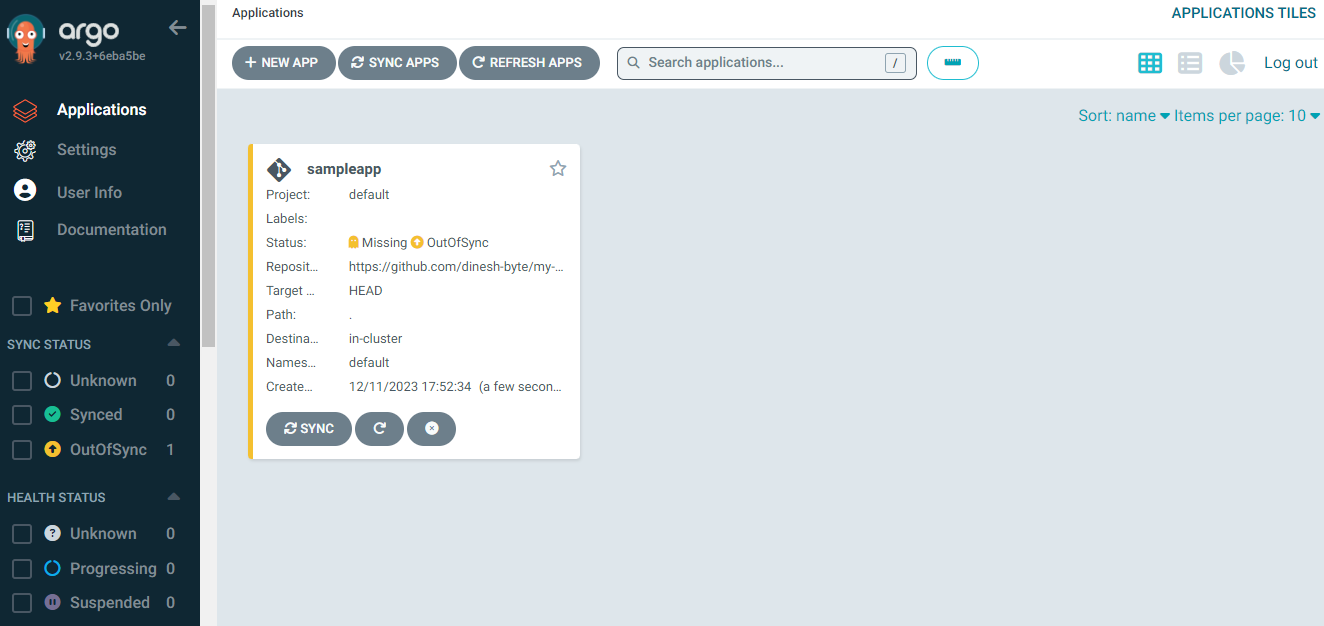
****

**Repository Link → Add your Repo url**

**Add Path of your Kubernetes manifest files in Your GitHub .**

****

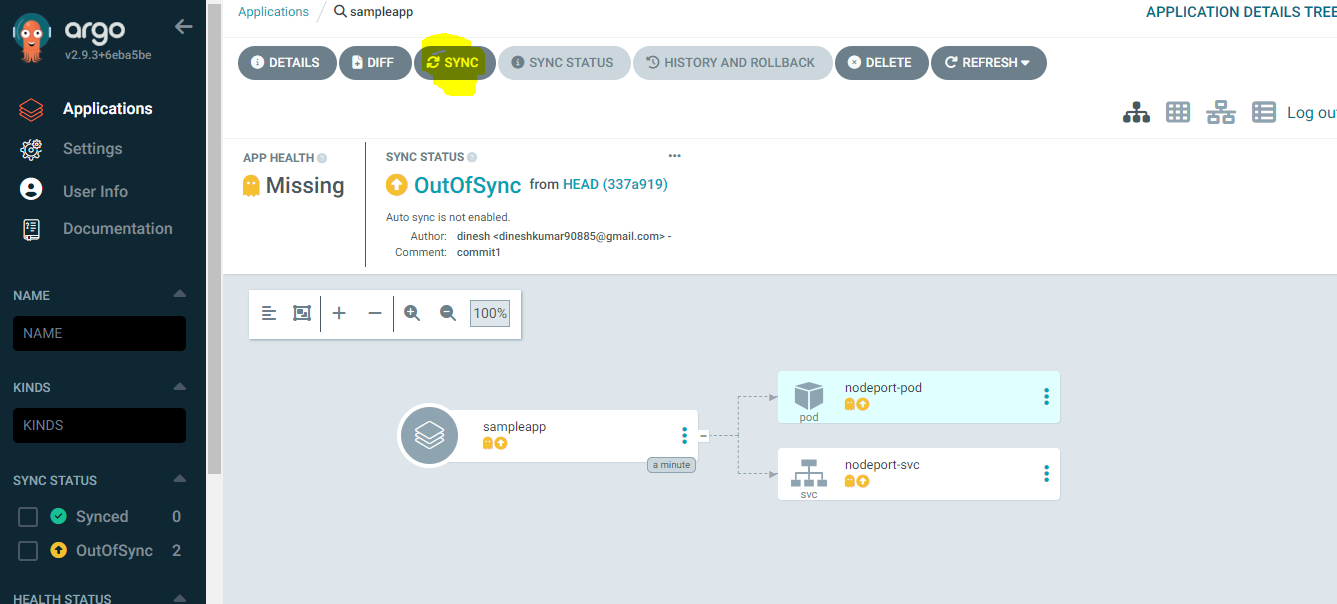
**Add the cluster information and namespace information**

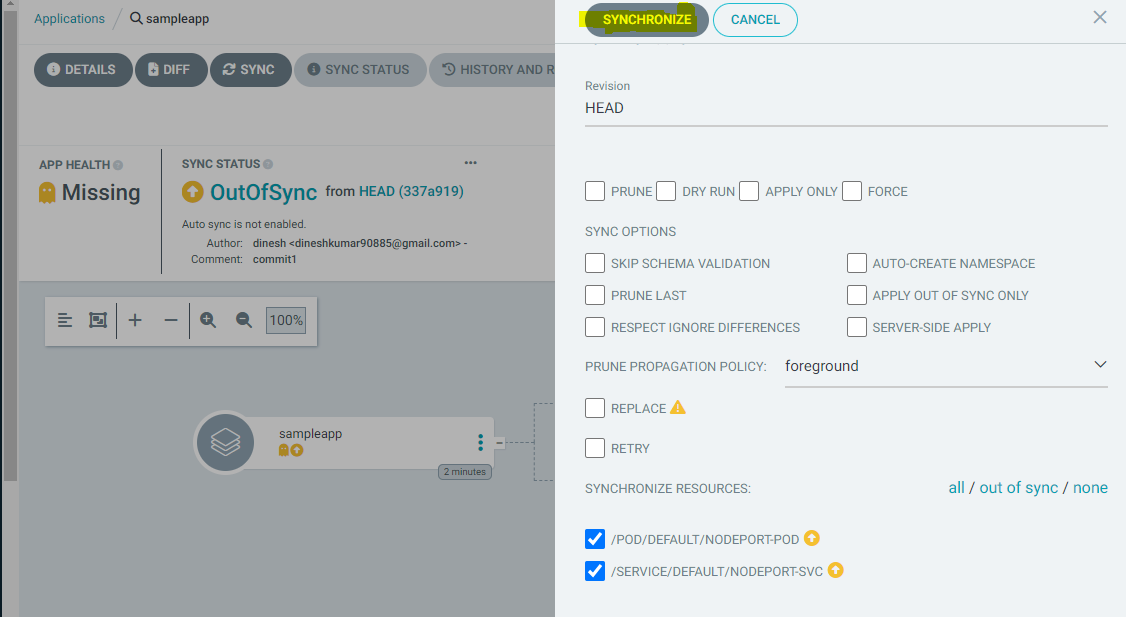
****

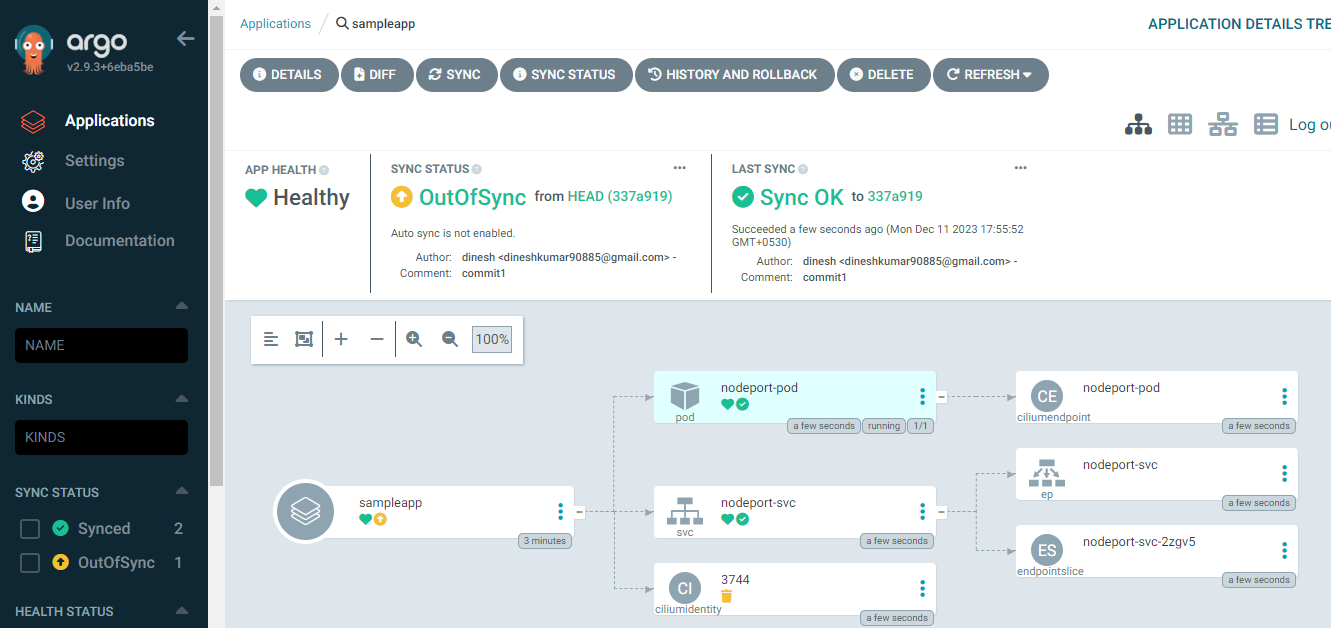
**Now We have Integrated the Github with Argo CD**

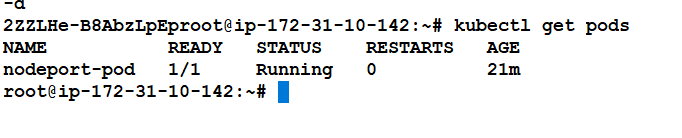
* **STEP -4 (Argo CD Deployment)**

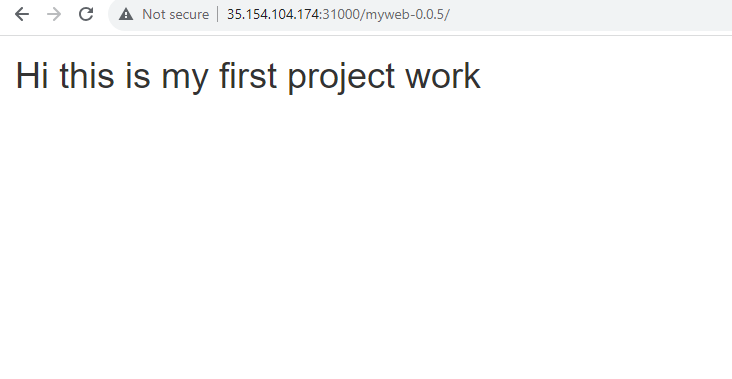
**Now Press Sync tab**



****

**Click Synchronize option**

****

**Output**

**Now We have Successfully done the Deployment**

**Features**

* **Automated deployment of applications to specified target environments**
* **Support for multiple config management/templating tools (Kustomize, Helm, Jsonnet, plain-YAML)**
* **Ability to manage and deploy to multiple clusters**
* **Rollback/Roll-anywhere to any application configuration committed in Git repository**
* **Health status analysis of application resources**
* **Automated configuration drift detection and visualization**
* **Automated or manual syncing of applications to its desired state**
* **Web UI which provides real-time view of application activity**
* **CLI for automation and CI integration**
* **Webhook integration (GitHub, BitBucket, GitLab)**