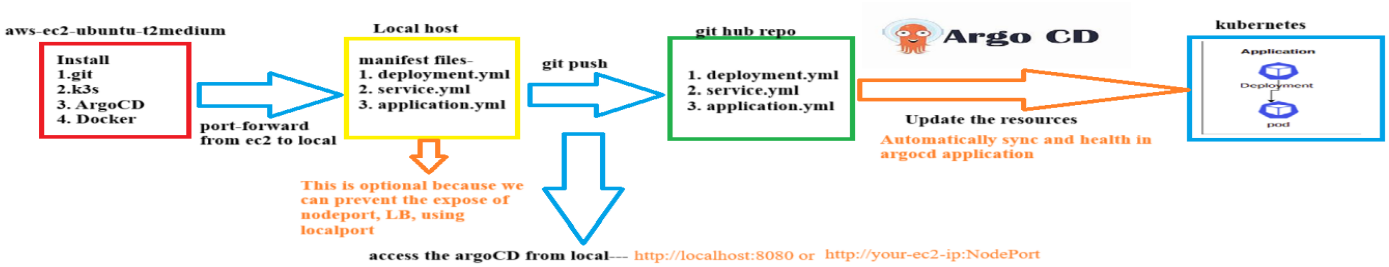
PROJECT-GitOps Workflow using ArgoCD on Kubernetes

Objective: Implement GitOps to automate Kubernetes deployments by syncing with a GitHub repository using ArgoCD.

Tools:

- EC2 (Ubuntu) ---- pick t2.medium instance
- K3s (Lightweight Kubernetes)
- ArgoCD
- GitHub
- Docker (for building and pushing images)
- MetalLB for K3s or Ingress controller (optional)

Architecture



Step-by-Step Guide

- **Set Up K3s (Kubernetes) on EC2 ----->** curl -sfL https://get.k3s.io | sh –
- **Check node status:** sudo kubectl get nodes
- **Set up kubectl for current user:**
  mkdir -p $HOME/.kube
  sudo cp /etc/rancher/k3s/k3s.yaml $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
- **Install ArgoCD in K3s**
  - kubectl create namespace argocd
  - kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
- **Expose the ArgoCD UI:**
  - nohup kubectl port-forward svc/argocd-server -n argocd 8080:443 > portforward.log 2>&1 &
  - ssh -i "C:\Path\To\YourKey.pem" -L 8080:localhost:8080 ubuntu@your-remote-ip # We don't want to expose of server NodePort address of server (optional)
  - kubectl get nodes

```
No resources found in default namespace.
root@ip-172-31-47-12:~# kubectl get nodes
NAME               STATUS   ROLES                  AGE     VERSION
ip-172-31-47-12    Ready    control-plane,master   3m51s   v1.32.3+k3s1
```

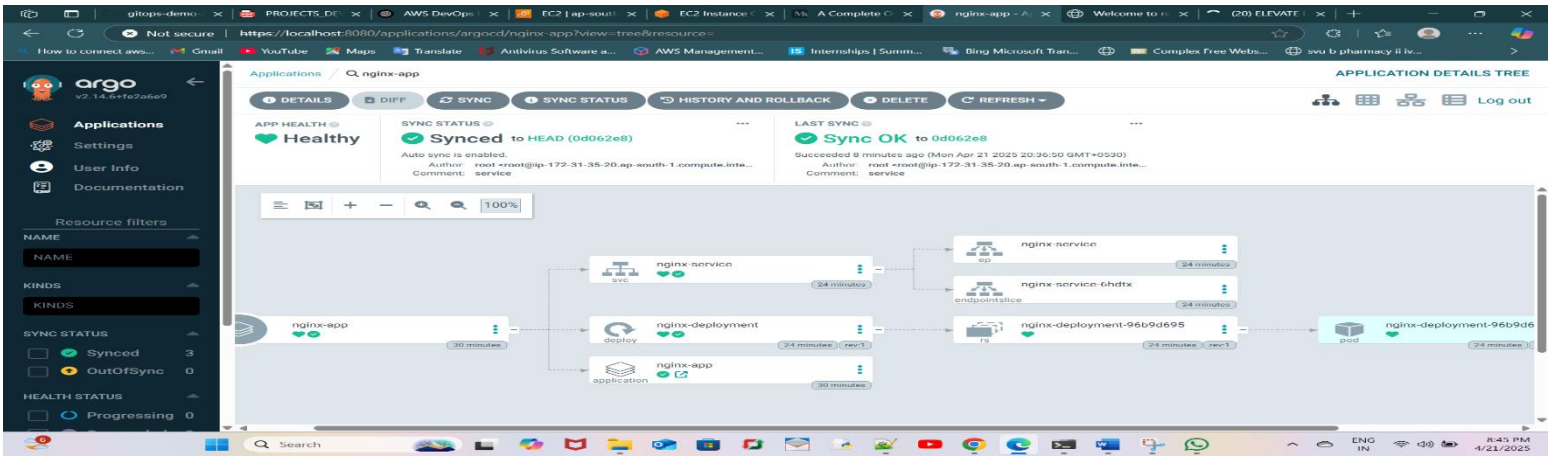
  - kubectl edit svc argocd-server -n argocd #change the clusterIP to nodeport
- **Access vi :** http://<ec2-user-ip>:8080
- **Get ArgoCD Admin Password:** kubectl get secret argocd-initial-admin-secret -n argocd -o jsonpath="{.data.password}" | base64 -d

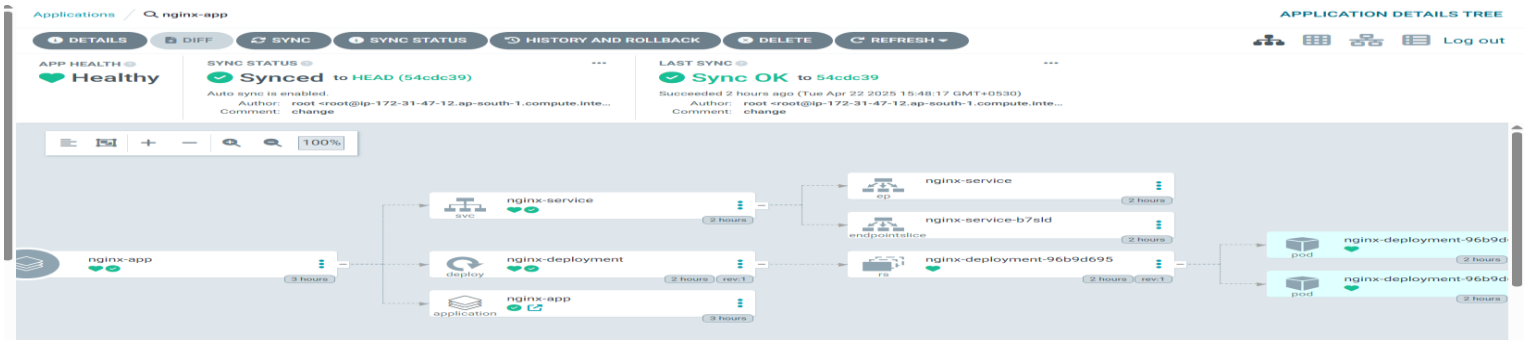
Create Your GitHub Repo

- Push the following sample YAML files to a public GitHub repo:
  1. deployment.yaml
  2. service.yaml

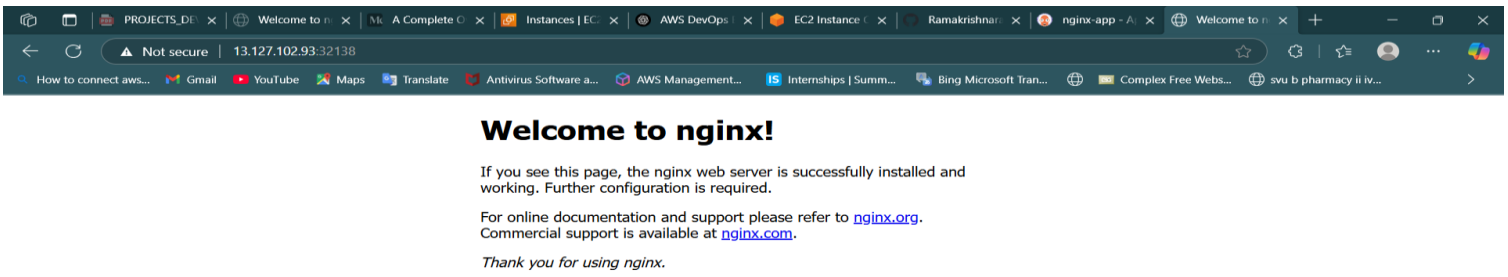
- **Configure ArgoCD to Sync from Git:** application.yml # Autosync file
- **Apply it:** kubectl apply -f .
- **Verify Deployment:** kubectl get all
- Then push the files into the GitHub repo, check the argocd UI

- Click refresh





- Check the svc of ngnix server NodePort ----> kubectl get svc -n argocd
- Then access it from your local browser: http://your-ec2-ip:NodePort or http://localhost:NodePort



- Check the EC2 port in security Group ----> if application is not accessing means check the server sg

GitHub Link: https://github.com/Ramakrishnaragi/project-GitOps-Workflow-using-ArgoCD-on-Kubernetes.git