**PROJECT REPORT**

**Objective: Implemented GitOps Workflow with ArgoCD on Kubernetes**

Implemented a production-ready GitOps Pipeline using ArgoCD and Kubernetes, demonstrating modern day DevOps practices for continuous deployment

Project Repository: <https://github.com/3lton007/GitOps-Workflow-using-ArgoCD-on-Kubernetes>

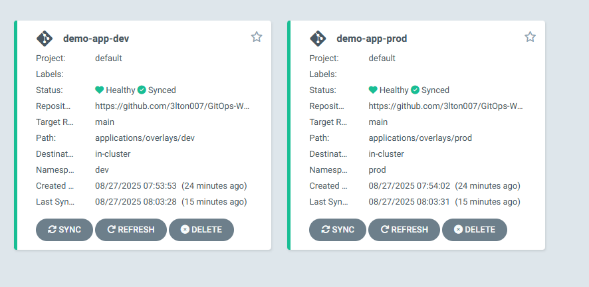
**Tools Used**

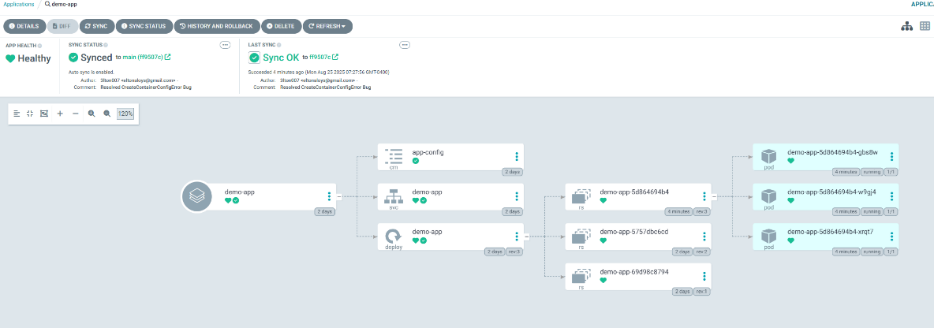
* **Container Orchestration:** Kubernetes (Minikube)
* **GitOps Controller:** ArgoCD
* **Version Control:** GitHub repository
* **Application Runtime:** Docker Engine
* **Configuration Management:** Kustomize

**Implementation:**

Source code is defined in git with complete tracking and rollback commit history. ArgoCD continuously monitors repository and infrastructure cluster state. Automatically self-heals by checking drift detection. Developers commit code to git repository, ArgoCD detects commit changes and automatically validated the deployment changes in the Kubernetes cluster. Realtime monitoring and health status reporting through Argocd dashboard. Rollback capabilities through git commit tracking sync. Created Demo App for ArgoCD Deployment on Kubernetes. Pushed The app code deployment changes to Git to Auto Sync with argoCD. Synced with the latest commit. Monitored multi-enviroment application setup with health checks using Prometheus.

**Screenshots**

****

****

**Conclusion:** Successfully implemented GitOps by syncing Kubernetes deployment states directly from a git Repository