GitOps Workflow using ArgoCD and Kubernetes

Introduction:

This project demonstrates the implementation of a GitOps workflow using ArgoCD and Kubernetes. The goal is to automate the deployment of Kubernetes applications by using a Git repository as the single source of truth. GitOps enhances reliability, repeatability, and traceability in CI/CD processes.

Abstract:

GitOps is a modern approach to continuous delivery where Git repositories serve as the canonical source for declarative infrastructure and applications. This project utilizes ArgoCD to watch a GitHub repository containing Kubernetes manifest files and automatically sync changes to a Minikube cluster. The system ensures that the Kubernetes cluster state reflects the desired state defined in Git, offering seamless and automated deployment.

Tools Used:

- Minikube: Local Kubernetes environment
- ArgoCD: GitOps CD tool for Kubernetes
- Docker: Container platform used by Minikube
- GitHub: Source code and manifest hosting
- kubectl: CLI for Kubernetes cluster interaction
- Visual Studio Code: Code editor for YAML and configuration files

Steps Involved in Building the Project:

1. Setup Environment: Install Docker, Minikube, kubectl, and VS Code on the local system.
2. Start Minikube: Launch local Kubernetes cluster using minikube start.
3. Install ArgoCD: Deploy ArgoCD on Minikube in a separate namespace using official manifests.
4. Expose ArgoCD UI: Port-forward ArgoCD server to access the web UI at https://localhost:8080.

5. Login to ArgoCD: Retrieve the initial admin password and log in via browser.
6. Prepare Git Repository: Create a GitHub repository and push Kubernetes manifests including deployment, service, namespace, and ArgoCD application definitions.
7. Deploy Application via ArgoCD: Apply the ArgoCD Application manifest (argocd-app.yaml) to the cluster.
8. Monitor Sync and Health: Verify application status through ArgoCD Dashboard; ensure it shows "Healthy" and "Synced".
9. Update via Git: Make changes in Git and watch ArgoCD auto-sync the updated state to Kubernetes.

Conclusion:

This project successfully demonstrates the GitOps methodology by integrating ArgoCD with Kubernetes running on Minikube. It highlights how application lifecycle management can be simplified, version-controlled, and automated using Git repositories. The approach improves operational efficiency, reduces human error, and ensures traceable deployments.