GitOps Workflow Automation using ArgoCD and Kubernetes on AWS EC2

Introduction

This project is about using **GitOps**, a modern way to manage application deployments. With GitOps, we use **Git** as the main place to store all deployment files. Any changes we make in Git automatically update the application running on **Kubernetes**. In this project, we used **ArgoCD** to make this automation possible. Everything was done on a cloud server using an **AWS EC2 instance with Ubuntu**.

Abstract

The goal of this project was to set up an automatic deployment system where applications are managed through a GitHub repository. We used **K3s** (a lightweight version of Kubernetes) to create a Kubernetes cluster. **ArgoCD** was installed on this cluster to monitor the GitHub repo. Whenever we updated the files in GitHub (like changing the app version), ArgoCD automatically updated the app running on the cluster. This process reduced manual work and made deployments fast and error-free.

Tools Used

- **AWS EC2 (Ubuntu)** – A virtual server to run everything
- **K3s** – A simple and lightweight Kubernetes platform
- **Docker** – To build and run containerized apps
- **ArgoCD** – Tool that watches Git and applies changes to Kubernetes
- **GitHub** – Stores the app's deployment files
- **kubectl & ArgoCD CLI** – Command-line tools for managing Kubernetes and ArgoCD
-

Steps Involved in Building the Project

Set Up the Server

Created an EC2 instance on AWS using Ubuntu. Installed Docker, Git, and Kubernetes tools.

Install Kubernetes (K3s)

Set up a small Kubernetes cluster using K3s on the EC2 instance.

Install ArgoCD

Deployed ArgoCD in the Kubernetes cluster. This gave us a dashboard and tools to control apps.

Create GitHub Repo

Made a new GitHub repo with Kubernetes files like deployment.yaml, service.yaml, kustomization.yaml.

Connect ArgoCD to GitHub

Told ArgoCD to monitor the GitHub repo. Created an app inside ArgoCD linked to that repo.

Enable Auto-Sync

Turned on auto-sync in ArgoCD. This made sure that any updates in the Git repo would automatically be applied to the app in Kubernetes.

Test the Setup

Changed the app version in the GitHub file. ArgoCD noticed the change and updated the app without us doing anything manually.

Conclusion

This project showed how powerful GitOps can be. Using ArgoCD with Kubernetes on an EC2 instance, we created a system where deployments happen automatically whenever we push changes to GitHub. This saves time, reduces mistakes, and makes app management easier. The same method can be used for larger production systems.