GitOps Workflow using ArgoCD on Kubernetes (EC2)

Introduction

This project demonstrates the implementation of a GitOps workflow using ArgoCD on a Kubernetes cluster deployed on an AWS EC2 instance. The objective was to enable automated, version-controlled application deployments directly from a GitHub repository.

Abstract

GitOps is a modern approach to Continuous Deployment where the desired state of infrastructure and applications is stored in Git. ArgoCD ensures the Kubernetes cluster is always in sync with the Git repository. This project set up ArgoCD on a Kubernetes cluster running on EC2 (using K3s) and connected it with a GitHub repository containing Kubernetes manifests for an NGINX application. Any change pushed to Git automatically triggered updates in the cluster, showcasing the core concept of GitOps.

Tools Used

- AWS EC2 (Ubuntu 20.04) - K3s (Lightweight Kubernetes) - ArgoCD (GitOps Continuous Delivery tool) - GitHub (Source code repository) - NGINX (Sample application) - kubectl (Kubernetes CLI)

Steps Involved in Building the Project

1. Launched an EC2 instance (Ubuntu) and installed K3s to set up a lightweight Kubernetes cluster. 2. Installed ArgoCD in the cluster and exposed its UI via NodePort. 3. Created a GitHub repository with Kubernetes manifests (deployment and service for NGINX). 4. Configured an ArgoCD Application resource pointing to the GitHub repo. 5. Verified that ArgoCD automatically deployed the NGINX application from GitHub. 6. Demonstrated GitOps by updating the image version in GitHub and observing auto-sync in ArgoCD.

Conclusion

The project successfully implemented a GitOps pipeline using ArgoCD on Kubernetes running on EC2. It showcased the advantages of GitOps such as automation, version control, and easy rollbacks. This setup highlights industry-standard practices for managing modern cloud-native applications.