

ArgoCD Deployment on Minikube - Report

1. Overview

In this project, we have successfully deployed ArgoCD on a local Minikube cluster to manage Kubernetes resources through GitOps. The deployment and configuration of ArgoCD was done step-by-step, with a focus on ensuring smooth installation and integration with Git repositories to automate the application deployment process.

2. Tools and Technologies Used

- Minikube: A tool to create a local Kubernetes cluster on a personal machine.
- ArgoCD: A GitOps continuous delivery tool for Kubernetes that synchronizes applications with their corresponding Git repositories.
- Kubernetes: A container orchestration platform that was used for managing the resources.
- Docker: A containerization platform used for ArgoCD components and Minikube.

3. Steps Followed

Step 1: Installing Minikube and Starting the Cluster

- Minikube was installed using the official instructions.
- A local Kubernetes cluster was created using the Docker driver to ensure that resources can be managed on the local machine.

Step 2: Installing ArgoCD

- ArgoCD was installed via a Kubernetes manifest file, and the ArgoCD components were deployed

in the 'argocd' namespace.

Step 3: Verifying Installation

- The status of the deployed pods and services was checked to ensure that all components of ArgoCD were running properly.

Step 4: Accessing the ArgoCD UI

- A port-forwarding command was used to access the ArgoCD UI from a local browser.

Step 5: Logging into ArgoCD

- The ArgoCD server was accessed via the browser at localhost:8080.
- The initial username was 'admin', and the password was fetched using a Kubernetes command.

Step 6: Setting Up the Application

- The Git repository containing YAML files for deployment was connected to ArgoCD.
- A new ArgoCD application was created using the Git repository as a source, and the destination cluster was set to the Minikube cluster.

Step 7: Synchronizing the Application

- After the application was created, ArgoCD synchronized the application with the Kubernetes cluster, ensuring that the resources in the Git repository were deployed.

Step 8: Monitoring and Troubleshooting

- Logs were viewed for debugging when issues occurred with pod startup or synchronization.
- Events and logs were used to identify the root cause of issues during the deployment and synchronization process.

4. Challenges Faced

- Several issues were faced during the process, such as port-forwarding failures, pod initialization issues, and misconfigured application specs.
- The ArgoCD application name had to be corrected to comply with the DNS subdomain naming conventions.
- Initially, the correct paths for the application manifests were required to be specified for ArgoCD to recognize and deploy the resources correctly.

5. Final Results

- ArgoCD was successfully deployed on Minikube.
- The application was successfully synchronized from a Git repository to the Minikube Kubernetes cluster.
- The ArgoCD UI was accessible for further monitoring and management of the deployment.