Kubernetes Exercise: Deploy a Sample Application Using Argo CD (GitOps)

Objectives

By the end of this exercise, you will:

1. Install **Argo CD** on a Kubernetes cluster.
2. Deploy a sample application from a **GitHub repository** using **Argo CD**.
3. Access and verify the deployed application.

Prerequisites

- A running Kubernetes cluster (Minikube, kind, or cloud-based)
- `kubectl` installed and configured
- `argocd` CLI installed (`brew install argocd` or via https://argo-cd.readthedocs.io/en/stable/cli_installation/)
- Access to GitHub (optional: your own repo)

Step 1: Install Argo CD

```
kubectl create namespace argocd

kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
```

Wait for the pods to become ready:

```
kubectl get pods -n argocd
```

Step 2: Access the Argo CD UI

Option 1: Port-forward the Argo CD API server

```
kubectl port-forward svc/argocd-server -n argocd 8080:443
```

Now access it via: https://localhost:8080

Step 3: Login to Argo CD

Get the initial admin password:

```
kubectl get secret argocd-initial-admin-secret -n argocd -o jsonpath="
{.data.password}" | base64 -d && echo
```

Then log in via CLI:

```
argocd login localhost:8080 --username admin --password <copied-password> --
insecure
```

Step 4: Deploy a Sample Application

We'll use a public GitHub repo with a simple nginx deployment:

Repo: `https://github.com/argoproj/argocd-example-apps.git` **Path:** `guestbook`

Create a new namespace for the app

```
kubectl create namespace guestbook
```

Create an Argo CD Application manifest (`guestbook-app.yaml`)

```
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: guestbook
  namespace: argocd
spec:
  destination:
    namespace: guestbook
    server: https://kubernetes.default.svc
  project: default
  source:
    repoURL: https://github.com/argoproj/argocd-example-apps.git
    targetRevision: HEAD
    path: guestbook
  syncPolicy:
    automated:
      prune: true
      selfHeal: true
```

Apply it:

```
kubectl apply -f guestbook-app.yaml
```

Step 5: Verify the Deployment

Check the application in Argo CD UI or via CLI:

```
argocd app list
argocd app get guestbook
```

Make sure the sync status is `Synced` and health is `Healthy`.

Step 6: Access the Application

Port-forward the frontend service:

```
kubectl port-forward svc/guestbook-ui -n guestbook 8081:80
```

Visit: http://localhost:8081

You should see the **Guestbook** UI running!

August 22, 2025