# Argo CD Documentation

## Description

Comprehensive documentation covering ArgoCD implementation.  
  
Key concepts, setup instructions, and usage examples are included.  
  
Documentation is well-structured, clear, and easy to follow.  
  
Reviewed and validated for accuracy and completeness.

## Overview

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes. It automates the deployment of applications by tracking changes in Git repositories and ensuring that the live state of applications matches the desired state defined in manifests.

## Features

- Declarative and Git-centric application deployment  
- Automated synchronization with Git repositories  
- Role-based access control (RBAC)  
- Multi-cluster management  
- Web UI and CLI for managing deployments  
- Integration with Kubernetes-native CRDs  
- Automated rollback and self-healing

## Installation

Prerequisites  
- Kubernetes cluster (v1.16 or later)  
- kubectl installed and configured  
- Helm (optional for Helm-based installation)  
  
Install using kubectl  
  
kubectl create namespace argocd  
kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml  
  
  
Install using Helm  
  
helm repo add argo https://argoproj.github.io/argo-helm  
helm repo update  
helm install argocd argo/argo-cd --namespace argocd --create-namespace

## Accessing Argo CD UI

To access the Argo CD UI, expose the API server:  
  
kubectl port-forward svc/argocd-server -n argocd 8080:443  
  
Then navigate to `https://localhost:8080` in your browser.

Retrieve the Admin Password  
  
kubectl get secret -n argocd argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 --decode

## Managing Applications

Create an Application via CLI  
  
argocd app create my-app \  
 --repo https://github.com/example/repo.git \  
 --path k8s \  
 --dest-server https://kubernetes.default.svc \  
 --dest-namespace default  
  
 Sync Application

argocd app sync my-app  
  
  
Check Application Status  
  
argocd app get my-app

## Role-Based Access Control (RBAC)

RBAC is managed via `argocd-rbac-cm` ConfigMap. Example configuration:  
yaml  
apiVersion: v1  
kind: ConfigMap  
metadata:  
 name: argocd-rbac-cm  
 namespace: argocd  
data:  
 policy.default: role:readonly

## Automatic Sync and Self-Healing

Enable auto-sync to ensure the application state is continuously updated:  
  
argocd app set my-app --sync-policy automated

## High Availability Deployment

For HA setup, deploy using Helm with HA configurations:  
  
helm install argocd argo/argo-cd --namespace argocd --set server.replicas=2

## Logging and Troubleshooting

Check logs for Argo CD components:  
  
kubectl logs -n argocd -l app.kubernetes.io/name=argocd-server  
  
  
Restart the Argo CD server if needed:  
  
kubectl rollout restart deployment argocd-server -n argocd