Argo CD GitOps for Kubernetes

CONCEPTS

Argo CD compares Git (desired) with the cluster (live). Out of Sync means they differ; Healthy is separate and describes the running resources.

Git repo desired state
↓ render
Application repo + path + target
↓ sync
Cluster live state

With automated sync, a merge to Git deploys by itself; manual sync and rollback are for apps without it.

LOGIN & CONTEXT

`argocd login <server>`
Log in with username and password

`argocd login <server> --sso`
Log in through your SSO provider

`argocd login --core`
Talk straight to the cluster API

`argocd context`
List saved server contexts

`argocd context <name>`
Switch to another server

APPLICATIONS

`argocd app list`
List applications

`argocd app get <app>`
Show status, sources and resources

`argocd app get <app> --refresh`
Re-read Git before showing status

`argocd app manifests <app>`
Print the rendered manifests

`argocd app logs <app>`
Show logs from the app's pods

`argocd app history <app>`
Show deployment history and IDs

`argocd app delete <app>`
Delete an application

CREATE & CONFIGURE

`argocd app create <app> --repo <url> --path <dir> --dest-server https://kubernetes.default.svc --dest-namespace <ns>`
Create an app from a Git path

`argocd app create <app> ... --sync-policy automated`
Deploy automatically on Git change

`argocd app set <app> --revision <rev>`
Point at another branch, tag or commit

SYNC & DIFF

`argocd app diff <app>`
Show live vs target differences

`argocd app sync <app>`
Sync to the target state

`argocd app sync <app> --dry-run`
Preview without changing the cluster

`argocd app sync <app> --prune`
Also delete removed resources

WAIT & ROLLBACK

`argocd app wait <app> --health`
Wait until the app is healthy

`argocd app wait <app> --sync --health`
Wait for synced and healthy

`argocd app rollback <app> <id>`
Roll back to a history ID

REPOSITORIES

`argocd repo list`
List registered repositories

`argocd repo add <url> --username <user> --password <pw>`
Add a private HTTPS Git repo

`argocd repo add <git@host:org/repo> --ssh-private-key-path <key>`
Add a Git repo over SSH

CLUSTERS

`argocd cluster list`
List registered clusters

`argocd cluster add <kube-context>`
Register a cluster from kubeconfig