

Deploying and Managing Applications with Argo CD



Steve Buchanan

CLOUD ARCHITECT

@buchatech | www.buchatech.com



Overview



**Registering a Kubernetes Cluster with
Argo CD**

Setting up Projects in Argo CD

Using Repositories with Argo CD

Deploying an App Using Argo CD

Application Sync & Rollback

Deleting Applications

Application Health & Status Reporting



Registering a Kubernetes Cluster with Argo CD



Kubernetes Clusters in Argo CD

By default with an Argo CD deployment the cluster it is running on is set as “in-cluster” (<https://kubernetes.default.svc>)

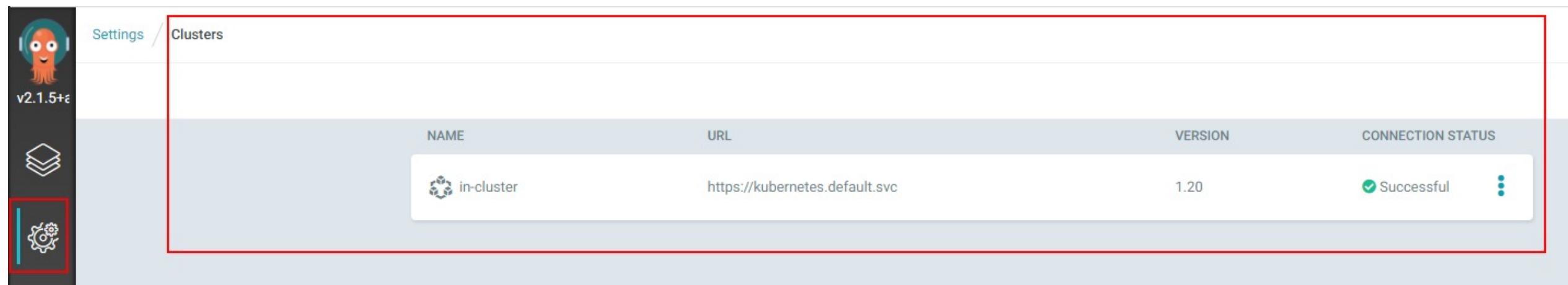
When apps are deployed you can deploy them to the “in-cluster” K8s cluster or an external K8s cluster

You will need to register an external K8s cluster in order to deploy apps outside of the default cluster



Register a Cluster

You are not able to register a new K8s cluster in the Argo CD web UI



The screenshot shows the Argo CD web UI interface. On the left, there's a sidebar with icons for Settings (octopus), Clusters (stacked squares), and Applications (gear). The main area has a header with 'Settings' and 'Clusters'. Below the header is a table with the following data:

NAME	URL	VERSION	CONNECTION STATUS
in-cluster	https://kubernetes.default.svc	1.20	Successful ⋮

However you can modify the in-cluster K8s cluster in the Argo CD web UI



Register a Cluster

You can add and manage K8s clusters using the Argo CD CLI

List of common commands to add & manage K8s clusters in Argo CD

argocd cluster add - Add a target cluster configuration to ArgoCD. The context must exist in your kubectl config

argocd cluster get - Get specific information about a cluster in plain text

argocd cluster list - List all known clusters in JSON format

argocd cluster rm - Removes a target cluster from Argo CD

argocd cluster rotate-auth - Rotates auth token for a cluster



Register a Cluster

Add your target K8s cluster to ArgoCD via the context in your kubectl config



List K8s cluster contexts in your current kubeconfig:

- `kubectl config get-contexts -o name`

Install a Service Account (argocd-manager), into the kube-system namespace of your kubeconfig context:

- `argocd cluster add CONTEXTNAME`



Demo



Demo: Connect Argo CD to an External Kubernetes Cluster

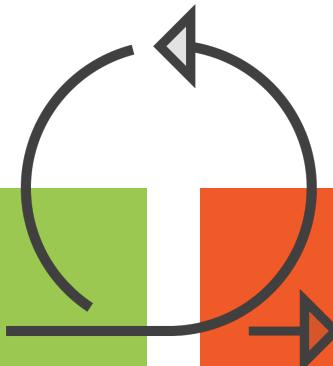


Setting up Projects in Argo CD



Projects in Argo CD

Projects are a logical grouping of apps in Argo CD



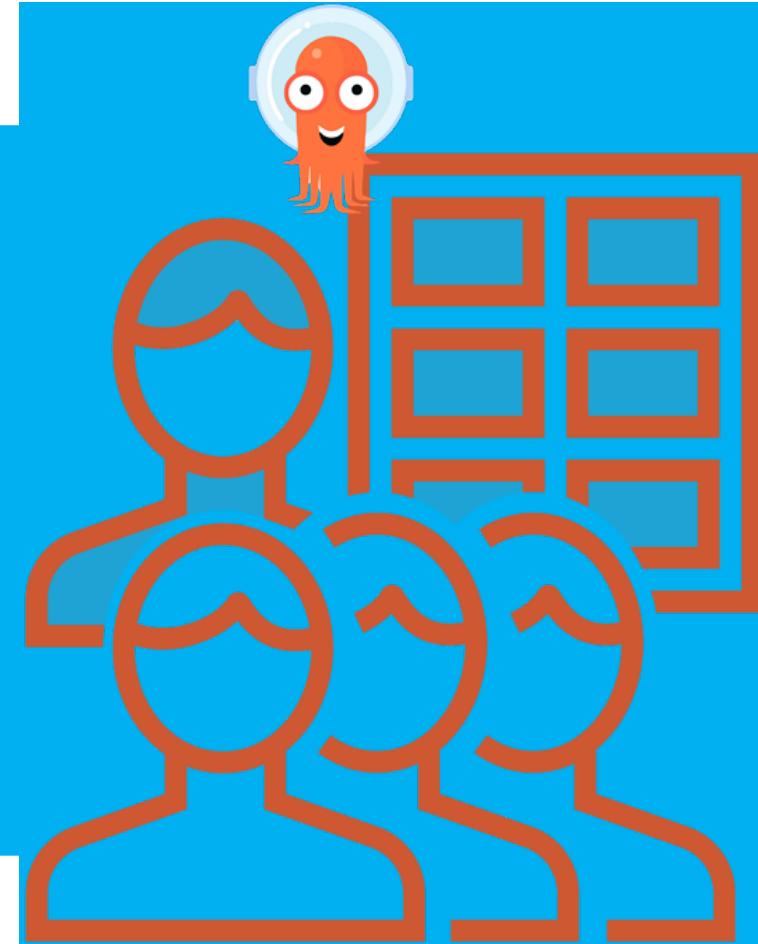
Projects provide:

- Restrict what Git repos can be deployed from
- Restrict what clusters & namespaces can be deployed to
- Restrict kinds of objects can/can't be deployed (i.e. Network Policy, DaemonSets, CRDs etc.)
- Define project roles providing app RBAC



Projects in Argo CD

Projects are especially useful when Argo CD is used by multiple teams to segment permissions such as different levels of access to namespaces, permitted Git repos, resource kinds & more....



Projects in Argo CD

The default project

Every app in Argo CD has to belong to a project

If no other projects exist when creating apps you have to use the default project

Argo CD has “The Default Project”

This is automatically created by default with the deployment of Argo CD, it can be modified but not deleted

The default project allows deployments from any source Git repo, to any cluster, & all resource Kinds



Projects in Argo CD

The screenshot shows the Argo CD Web UI interface. On the left, there's a sidebar with filters for NAME, KINDS, SYNC STATUS (Synced, OutOfSync), and HEALTH STATUS (Healthy, Progressing, Degraded). The main area displays a hierarchical tree of resources under the 'jenkins' project. A large orange box highlights the 'Web UI' text. At the top, there are buttons for APP DETAILS, APP DIFF, SYNC, SYNC STATUS, HISTORY AND ROLLBACK, DELETE, and REFRESH.



The screenshot shows the Argo CD CLI terminal window. It has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with TERMINAL selected. The terminal output shows the following command and its result:

```
$ argo cd version
argo cd: v2.1.6+a346cf9
BuildDate: 2021-10-28T20:03:55Z
GitCommit: a346cf933e10d872eae26bff8e58c5e7ac40db25
GitTreeState: clean
GoVersion: go1.16.5
Compiler: gc
Platform: windows/amd64
```

A blue box highlights the 'CLI' text.

Projects can be created & managed
in the **Web UI**

Projects can be created & managed
in Argo CD **CLI**



Add Project to Argo CD - Web UI

Settings >> Projects >> Click “New Project” or on existing project name

The screenshot shows the Argo CD Web UI interface. On the left, there is a vertical sidebar with icons for Settings, Projects, Clusters, GnuPG keys, Certificates, and Repositories. The 'Projects' icon is highlighted with a red box and has a red arrow pointing to it from the top-left. The main area shows a table with two projects: 'default' and 'dev'. The 'default' project has a checkbox next to its name. A red arrow points from the 'NAME' column of the 'default' row to the 'Name' field in the 'GENERAL' tab of the 'New Project' dialog. The 'New Project' dialog is open on the right, showing fields for 'Name' (set to 'projectname') and 'Description' (set to 'this is my new project'). It also includes 'CANCEL' and 'CREATE' buttons.

Settings / Projects

NEW PROJECT

v2.1.5+
default
dev

GENERAL

Name
projectname

Description
this is my new project

CANCEL CREATE

Repositories
Certificates
GnuPG keys
Clusters
Projects



Add Project to Argo CD - CLI



```
argocd proj create projectname -d \  
https://kubernetes.default.svc,namespace -s \  
https://github.com/argoproj/gitreponame.git
```



Projects in Argo CD

Common commands used to manage projects



```
argocd proj list  
argocd proj get  
argocd proj delete  
argocd proj add-destination  
argocd proj add-source  
argocd proj allow-cluster-resource  
argocd proj allow-namespace-resource
```



Using Repositories with Argo CD



Repositories in Argo CD

Argo CD supports connecting to public or private Git repositories

Argo CD can connect to private Git repositories using HTTP, HTTPS, SSH, or a GitHub App



Argo CD Repo - Add Web UI

1 In the Argo CD web UI Navigate to Settings/Repositories

The screenshot shows the Argo CD web UI interface. At the top, there's a navigation bar with 'Settings / Repositories' selected. Below it are three buttons: '+ CONNECT REPO USING SSH', '+ CONNECT REPO USING HTTPS', and '+ CONNECT REPO USING GITHUB APP'. To the right are 'REFRESH LIST', 'Log out', and a 'REPOSITORIES' link. The main area is titled 'REPOSITORIES' and lists one repository: 'https://github.com/Buchatech/ArgoCDandAKS.git' (Type: git). The connection status is 'Successful' with a green checkmark. A vertical sidebar on the left has icons for file storage, settings, users, and documentation, with 'Settings' highlighted by a red box.

2 Click Connect Repo using (SSH/HTTPS/GITHUB APP) button & enter credentials

A modal dialog box is open, titled 'Connect repo using SSH'. It contains fields for 'Name (mandatory for Helm)', 'Repository URL', and 'SSH private key data'. At the bottom are 'CONNECT', 'SAVE AS CREDENTIALS TEMPLATE', and 'CANCEL' buttons. A red box highlights the 'CONNECT' button. The background of the main Argo CD interface is dimmed.

3 Click Connect to test the connection & add the Git repository

Argo CD Repo - Add CLI



```
argocd repo add  
https://github.com/likamrat/hello_arc --username  
USERNAMEHERE --password PASSWORDHERE
```



Deploying an App Using Argo CD



Deploying an App to Argo CD

**Deploying an app in Argo CD
can be done from the web UI
or CLI**

**To deploy an app you point to
a Git repository of either K8s
manifest, Helm, or Kustomize**



Deploying an App to Argo CD - Web UI

1

In the Argo CD web UI ensure you are on the Applications page

2

Click the + NEW APP button

3

Give the app the name **hellok8s**, use the project default, select Automatic for the sync policy, check AUTO-CREATE NAMESPACE

4

On Source for the Repo URL use <https://github.com/paulbouwer/hello-kubernetes.git> & select **deploy/helm/hello-kubernetes** for the path

5

For the DESTINATION select <https://kubernetes.default.svc> for the Cluster URL and use **hellok8s** for the namespace

6

Leave all the defaults under HELM

7

Click the CREATE button at the top of the UI

The screenshot shows the Argo CD Web UI interface. On the left, there's a sidebar with icons for Applications, Repositories, and Metrics, and a version indicator 'v2.1.5+'. The main area is titled 'Applications' with buttons for '+ NEW APP' and 'SYNC APPS'. A search bar and filters are also present. In the center, two applications are listed: 'dev-app1' (Synced, Healthy) and 'gb' (Missing, OutOfSync). A modal window is open for creating a new application:

- Application Name:** hellok8s
- Project:** dev
- Sync Policy:** Automatic
- SOURCE:** Repository URL: <https://github.com/paulbouwer/hello-kubernetes.git>, Path: **deploy/helm/hello-kubernetes**
- DESTINATION:** Cluster URL: <https://kubernetes.default.svc>, Namespace: **hellok8s**
- Labels:** (empty)
- Sync Options:** Includes checkboxes for PRUNE RESOURCES, SELF HEAL, SKIP SCHEMA VALIDATION, PRUNE LAST, and REPLACE.
- Prune Propagation Policy:** foreground
- Advanced Options:** Includes checkboxes for AUTO-CREATE NAMESPACE (checked), APPLY OUT OF SYNC ONLY, and a dropdown for GIT.

Deploying an App to Argo CD - CLI



```
argocd app create hellok8s --repo  
https://github.com/paulbouwer/hello-kubernetes.git --  
path deploy/helm/hello-kubernetes --dest-server  
https://kubernetes.default.svc --dest-namespace default
```



App of Apps in Argo CD

You can create an app that creates other apps,
this is known as the “App of Apps pattern”

You are declaratively deploying an app that
consists of other apps deploying them at the
same time



App of Apps in Argo CD

Git repository Structure:

```
└── Chart.yaml
└── templates
    ├── guestbook.yaml
    ├── helm-dependency.yaml
    ├── helm-guestbook.yaml
    └── kustomize-
        └── guestbook.yaml
└── values.yaml
```

<https://github.com/argoproj/argocd-example-apps>



App of Apps in Argo CD

A boiler plate chart.yaml file

```
apiVersion: v2
name: applications
description: Applications

# A chart can be either an 'application' or a 'library' chart.
#
# Application charts are a collection of templates that can be packaged into
versioned archives
# to be deployed.
#
# Library charts provide useful utilities or functions for the chart developer. They're
included as
# a dependency of application charts to inject those utilities and functions into the
rendering
# pipeline. Library charts do not define any templates and therefore cannot be
deployed.
type: application

# This is the chart version. This version number should be incremented each time
you make changes
# to the chart and its templates, including the app version.
# Versions are expected to follow Semantic Versioning (https://semver.org/)
version: 0.1.0

# This is the version number of the application being deployed. This version number
should be
# incremented each time you make changes to the application. Versions are not
expected to
# follow Semantic Versioning. They should reflect the version the application is
using.
appVersion: "1.0"
```

App of Apps in Argo CD

Each child app in the Git repository>Templates folder should have a yaml file

```
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: helm-guestbook
  namespace: argocd
  finalizers:
    - resources-finalizer.argocd.argoproj.io
spec:
  destination:
    namespace: helm-guestbook
    server: {{ .Values.spec.destination.server }}
  project: default
  source:
    path: helm-guestbook
    repoURL: {{ .Values.spec.source.repoURL }}
    targetRevision: {{ .Values.spec.source.targetRevision }}
```



App of Apps in Argo CD

The `values.yaml` should contain similar

```
spec:  
  destination:  
    server: https://kubernetes.default.svc  
  source:  
    repoURL:  
      https://github.com/argoproj/argocd-  
      example-apps  
    targetRevision: HEAD
```



Demo



**Demo: Deploy an Application to a
Kubernetes Cluster via Argo CD**



Application Sync & Rollback



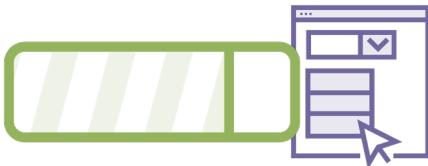
Sync with Argo CD

Automatic Sync



- With auto sync Argo CD applies every object in the application
- Argo CD will sync when it detects differences between the Git repo, & the live state in the K8s cluster

Selective Sync



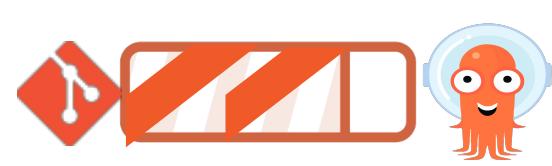
These sync out-of-sync resources only

Sync Windows



- These are configurable windows of time when syncs happen
- They can be set to allow or deny
- They can apply to manual & automated syncs
- Schedules are in cron format & can be targeted to applications, namespaces & clusters

Sync Phases & Waves



- Argo CD executes a sync operation in three phases: pre-sync, sync and post-sync
- Each phase can have one or more waves, to ensure certain resources are healthy before subsequent resources are synced in the next phase



Rollback with Argo CD

With Argo CD you have an option to rollback an application

A rollback is typically used when an application is not healthy

Argo CD tracks an applications history and you can rollback to a previous version



Rollback with Argo CD

Auto-Sync needs
to be disabled in
order for rollback
to occur

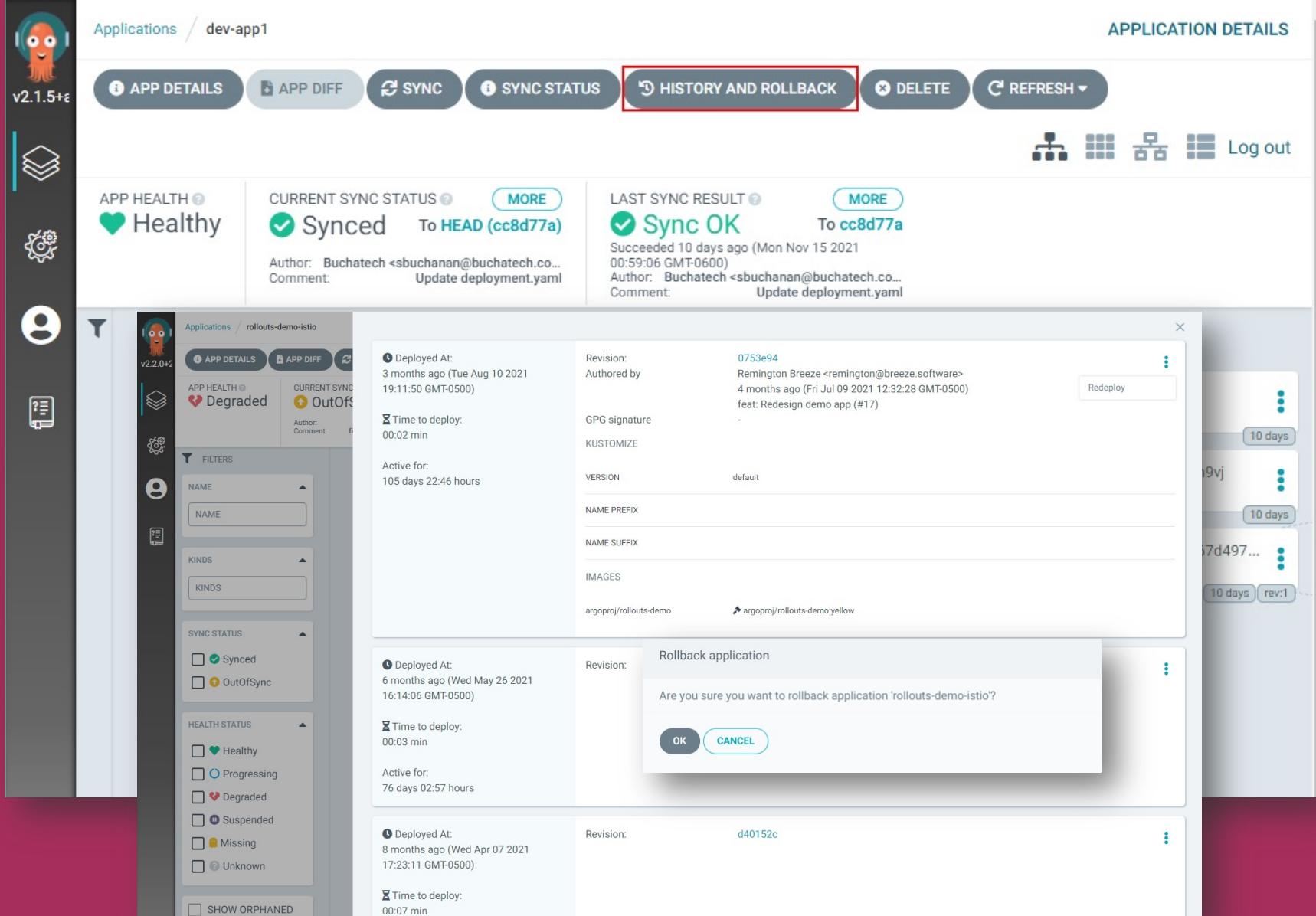
Argo CD keeps a
cache of your Git
repository

Rollback can be
used as a
safeguard to
quickly backout of
an application
upgrade



Rollback with Argo CD - Web UI

1 In the Argo CD web UI ensure you are on the Applications page



The screenshot shows the Argo CD Web UI interface. On the left, there is a sidebar with icons for Applications, App Diff, Sync, Sync Status, History and Rollback (which is highlighted with a red border), Delete, Refresh, and Logout. The main content area is titled "Applications / dev-app1". It displays the following information:

- APP HEALTH:** Healthy
- CURRENT SYNC STATUS:** Synced To HEAD (cc8d77a)
- LAST SYNC RESULT:** Sync OK To cc8d77a (Succeeded 10 days ago)

Below this, a detailed view of a deployment is shown:

Deployment Details	Revision
Deployed At: 3 months ago (Tue Aug 10 2021 19:11:50 GMT-0500)	0753e94
Authored by: Buchatech <sbuchanan@buchatech.co...	Remington Breeze <remington@breeze.software>
Comment: Update deployment.yaml	4 months ago (Fri Jul 09 2021 12:32:28 GMT-0500)
GPG signature: -	feat: Redesign demo app (#17)
KUSTOMIZE: default	
VERSION: default	
NAME PREFIX:	
NAME SUFFIX:	
IMAGES: argoproj/rollouts-demo:yellow	

A modal window titled "Rollback application" is open, asking "Are you sure you want to rollback application 'rollouts-demo-istio'?". It has "OK" and "CANCEL" buttons.

2 Click on the application you want to rollback

3 Click the History and Rollback button

4 Click on the ellipsis of the previous deployed version you want to redeploy

5 Click OK



Rollback with Argo CD - CLI



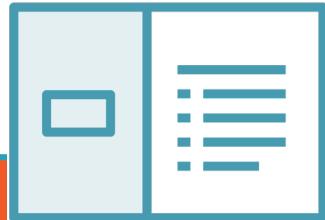
```
argocd app rollback YOURAPPNAME [HISTORYID]  
[flags]
```



Deleting Applications

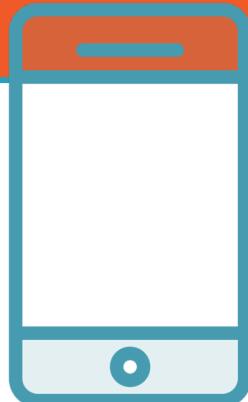


Deleting an App in Argo CD



Apps in Argo CD can be deleted via the web UI or from the CLI

You can delete an app with or without cascading



Deleting an App in Argo CD

A non-cascade delete, deletes
only the app

A cascade delete, deletes
both the app and its resources



Deleting an App in Argo CD - Web UI

The screenshot illustrates the process of deleting an application in Argo CD, specifically focusing on the 'dev-app1' application.

Top Left Panel: Delete Application Confirmation

A modal window titled "Delete application" asks if you're sure you want to delete the application 'dev-app1'. It contains a yellow warning icon, a text input field with placeholder "Please type 'dev-app1' to confirm the deletion of the resource", and three propagation policy options: "Foreground" (selected), "Background", and "Non-cascading". Buttons for "OK" and "CANCEL" are at the bottom.

Top Right Panel: Applications Overview

The main interface shows the "Applications" screen with a search bar and a "SYNC APPS" button. A sidebar on the left includes icons for "v2.1.5+", "Deployments", "Services", "ReplicaSets", and "Pods". A "FILTERS" section allows filtering by "SYNC STATUS": Unknown (0), Synced (2), and OutOfSync (1). Two applications are listed:

- dev-app1**: Project: default, Labels: none, Status: Healthy Synced, Repository: https://github.com/Buchatech/ArgoC..., Target R...: HEAD, Path: hellok8s/overlay/dev, Destination: in-cluster, Names...: dev-app1. The "DELETE" button is highlighted with a red box.
- eotd-app**: Project: default, Labels: none, Status: Healthy Synced, Repository: https://github.com/Buchatech/EOTD-..., Target R...: HEAD, Path: k8s, Destination: in-cluster, Names...: eotd. The "DELETE" button is highlighted with a red box.

Bottom Left Panel: Application Details

The "Applications / dev-app1" details page shows the application status as "Synced" (Healthy) and the last sync result as "Sync OK". It includes tabs for "APP DETAILS", "APP DIFF", "SYNC", "SYNC STATUS", "HISTORY AND ROLLBACK", and "DELETE". The "DELETE" button is highlighted with a red box. Below this, a deployment diagram shows the application's components: dev-app1 (Deployment) → dev-hello-kubernetes (Service) → dev-hello-kubernetes (Endpoint) → dev-hello-kubernetes-jm9vj (ReplicaSet) → dev-hello-kubernetes-767d497... (Pod) → dev-hello-kubernetes-767d497... (Pod).

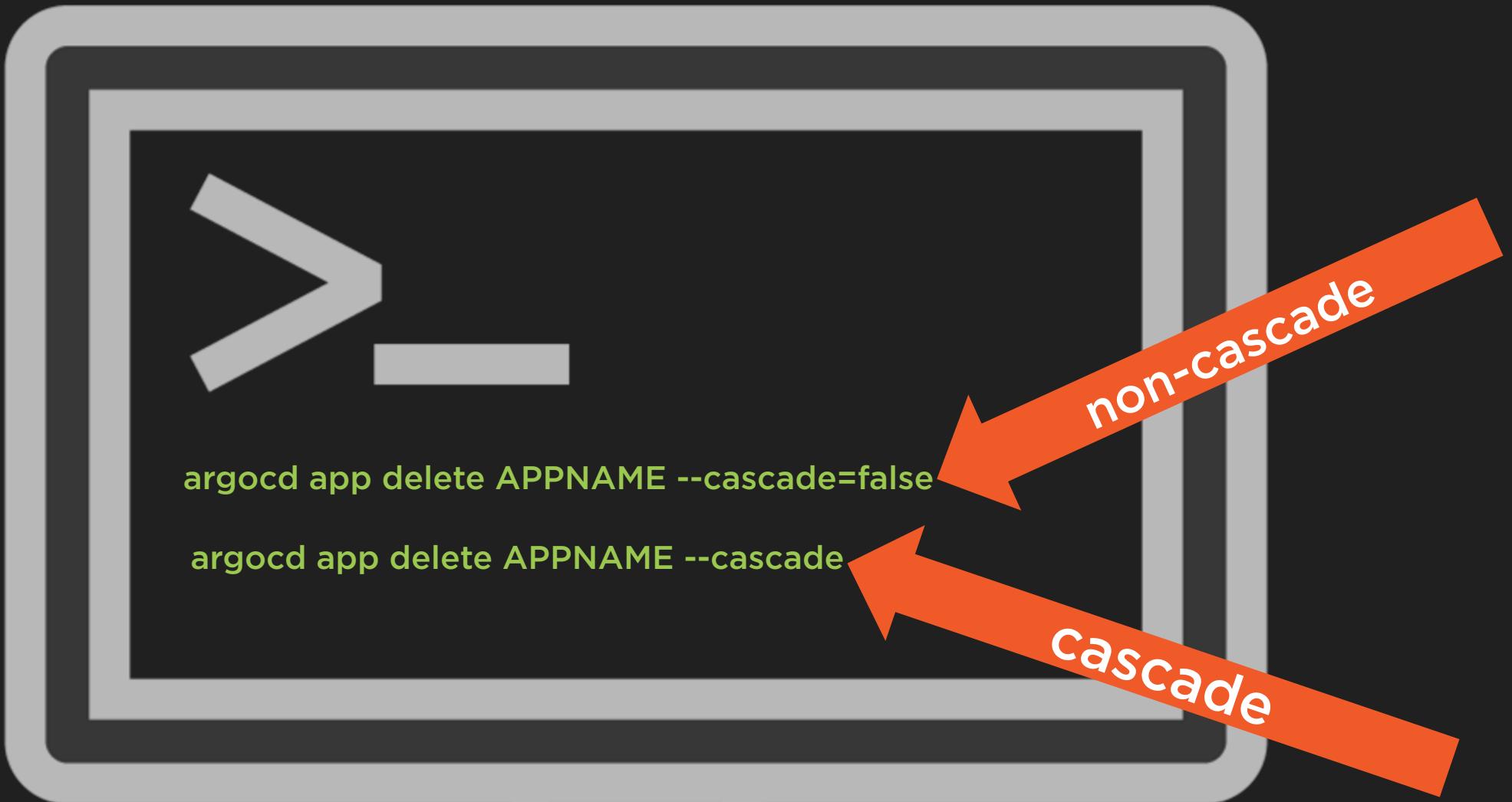
Bottom Right Panel: Application Details for eotd-app

The "Applications / eotd-app" details page shows the application status as "Synced" (Healthy) and the last sync result as "Sync OK". It includes tabs for "SYNC", "REFRESH", and "DELETE". The "DELETE" button is highlighted with a red box.

Bottom Right Corner: Navigation Arrow

A circular navigation arrow icon is located in the bottom right corner of the interface.

Deleting an App in Argo CD - CLI



Application Health & Status Reporting



Application Health in Argo CD

The best place to get your application health and status is in the Argo CD web UI



Application Health in Argo CD

Applications APPLICATIONS

+ NEW APP **SYNC APPS** / [grid]

Log in

FILTERS

SYNC STATUS

- Unknown 0
- Synced 16
- OutOfSync 6

HEALTH STATUS

- Unknown 0
- Progressing 1
- Suspended 1
- Healthy 17
- Degraded 1
- Missing 2

SUMMARY

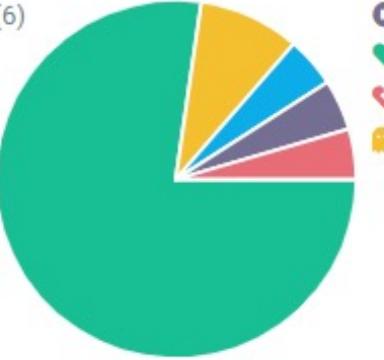
APPLICATIONS:	22
SYNCED:	16
HEALTHY:	17
CLUSTERS:	1
NAMESPACES:	13

Sync



- Unknown (0)
- Synced (16)
- OutOfSync (6)

Health



- Unknown (0)
- Progressing (1)
- Suspended (1)
- Healthy (17)
- Degraded (1)
- Missing (2)

v2.2.0+2

[grid] [grid] [grid] [grid]

Application Health in Argo CD

The screenshot shows the Argo CD web interface with a sidebar on the left and a main content area on the right.

Left Sidebar:

- Version: v2.2.0+2
- Icons for Applications, Projects, Workspaces, and Logs.
- Sync Status Filter: Includes Unknown (0), Synced (0), and OutOfSync (3).
- Health Status Filter: Includes Unknown (0), Progressing (1), Suspended (1), and Healthy (17). The Degraded and Missing filters are highlighted with a red border.

Top Bar:

- Applications
- APPLIANCES
- + NEW APP
- SYNC APPS
- Search applications...
- /
- Log in

Content Area:

Items per page: all

Application Name	Project	Status	Last Synced	Actions
guestbook	default	Degraded	2023-06-01 10:00:00	<button>SYNC</button> <button>REFRESH</button> <button>DELETE</button>
rollouts-demo-istio	default	Degraded	2023-06-01 10:00:00	<button>SYNC</button> <button>REFRESH</button> <button>DELETE</button>
test	default	Degraded	2023-06-01 10:00:00	<button>SYNC</button> <button>REFRESH</button> <button>DELETE</button>



Application Health in Argo CD

Applications / rollouts-demo-istio APPLICATION DETAILS

APP DETAILS APP DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH ▾

APP HEALTH Degraded

CURRENT SYNC STATUS OutOfSync From HEAD (6179837) MORE

Author: cskh <huikang209@gmail.com> - Comment: fix: analysis args for istio-subset example (#21)

LAST SYNC RESULT Sync OK To 9647f4a MORE

Succeeded 2 months ago (Thu Oct 07 2021 15:50:17 GMT-0500)
Author: Alexander Matyushentsev <AMatyushentsev@gmail.com> - Comment: Remove oauth annotations from demo ingress

FILTERS

NAME NAME

KINDS KINDS

SYNC STATUS Synced Synced
OutOfSync OutOfSync

HEALTH STATUS Healthy Healthy
Progressing Progressing
Degraded Degraded
Suspended Suspended

The diagram illustrates the deployment of an Istio subset example across various Kubernetes resources. The root application 'rollouts-demo-istio' contains one parameter override. It branches into five main components: 'istio-rollout-stable', 'istio-success-rate', 'istio-rollout', 'istio-rollout-gateway', and 'istio-rollout-vsvc'. Each component is shown with its current status (e.g., 'Sync OK' or 'Synced') and revision history. The 'istio-rollout' component is currently 'OutOfSync' and has three active rollout instances: 'istio-rollout-5b65565cd7', 'istio-rollout-67c9d9c96', and 'istio-rollout-67c9d9c96-17'. The 'istio-rollout-gateway' component is also 'OutOfSync' and has three active analysis runs: 'istio-rollout-bbbc78968-tqlpk', 'istio-rollout-bbbc78968-v55vd', and 'istio-rollout-bbbc78968-16'. The 'istio-rollout-vsvc' component is 'Synced' and has one active analysis run: 'istio-rollout-bbbc78968-22ssf'. The 'istio-rollout-stable' component is 'Synced' and has one active endpointslice: 'istio-rollout-stable-fphb'.

Summary



In this module we covered:

- How projects & repos work with Argo CD
- How to connect Argo CD to external Kubernetes clusters
- We learned about deploying apps, syncing them, rolling them back, deleting them, & viewing their health status in Argo CD

Why this is important?:

- Apps are at the center of Argo CD & GitOps. When it comes to Argo CD it is important to know how to deploy apps in Argo CD & additional features to help you manage them

