Thinknyx®

You Trust, We Deliver!

Argo CD for the Absolute Beginners | Hands-On

GitOps-based Continuous Delivery Tool

Yogesh Raheja

Yogesh Raheja

Puppet for the Absolute Beginners – Hands-On

SaltStack for the Absolute Beginners – Hands-On

Infrastructure Automation with OpenTofu – Hands-On

AI Ecosystem for the Absolute Beginners - Hands-On

Mastering Prompt Engineering for GenAI

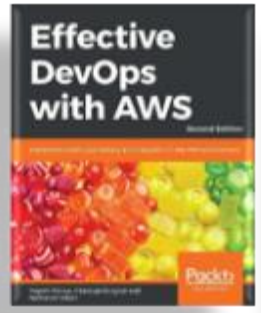
Generative AI Essentials - Practical Use Cases

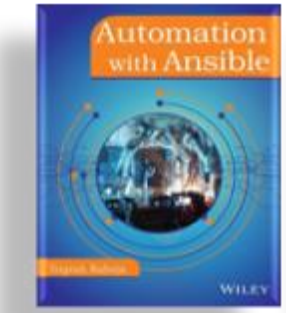
Mastering Docker Essentials - Hands-on

Unlocking Python for the Absolute Beginners

Podman for the Absolute Beginners - Hands-On

Practical Kubernetes – Beyond CKA and CKAD





Yogesh Raheja

Thinknyx Technology Team

Course Workflow

Foundational Concepts

- Argo CD Applications
- Argo CD Projects
- Sync Policies
- Webhooks
- Multi Cluster Environment
- App Of Apps Pattern

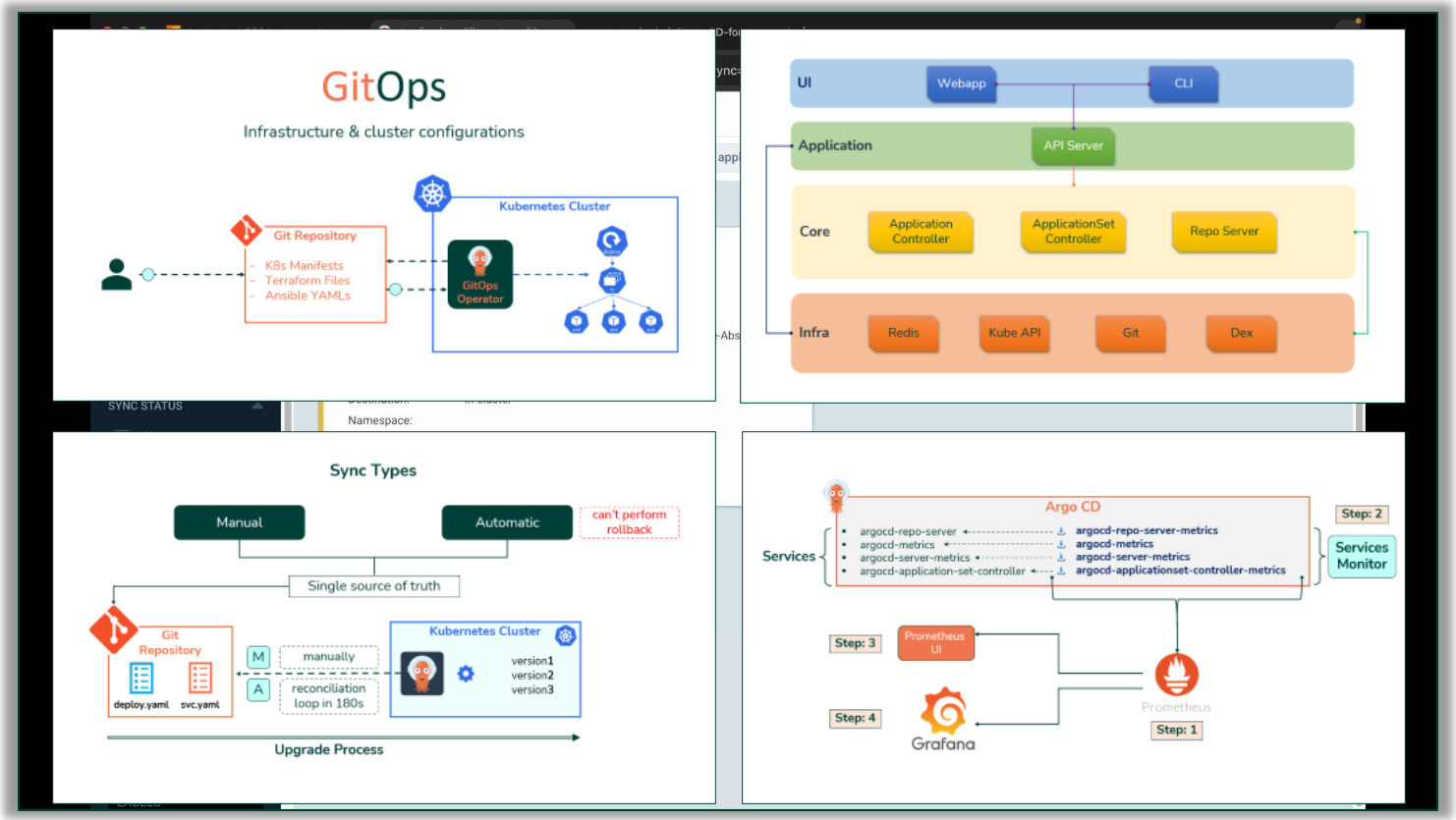
- Manifests
- Helm
- Kustomize

Course Workflow

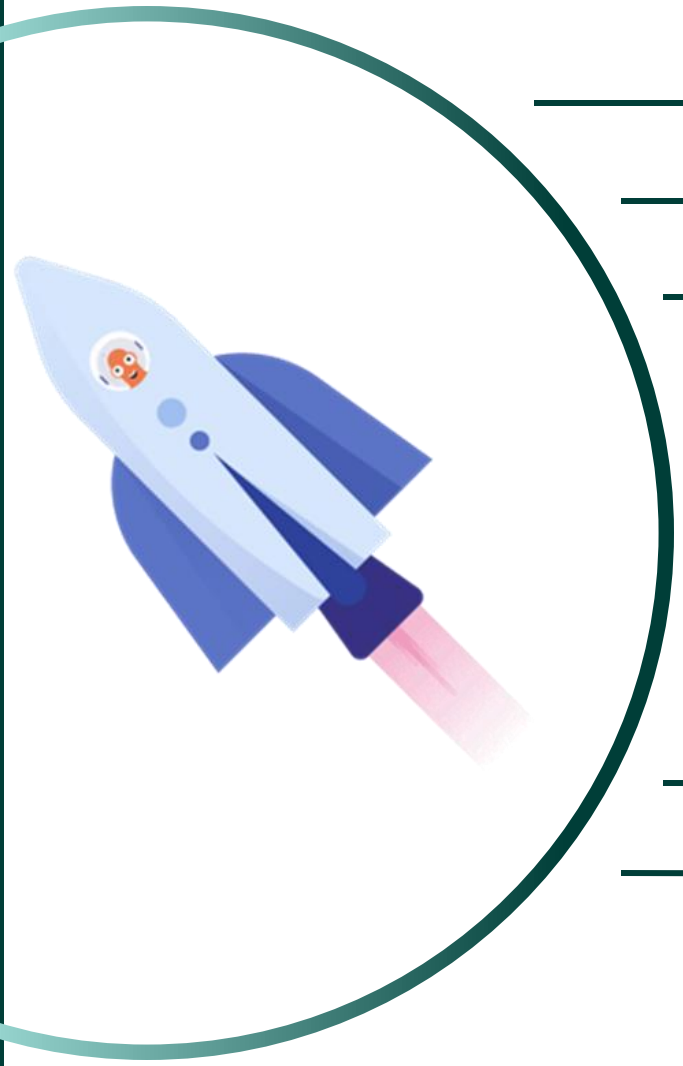
Lectures

Live Demonstrations

Assignments



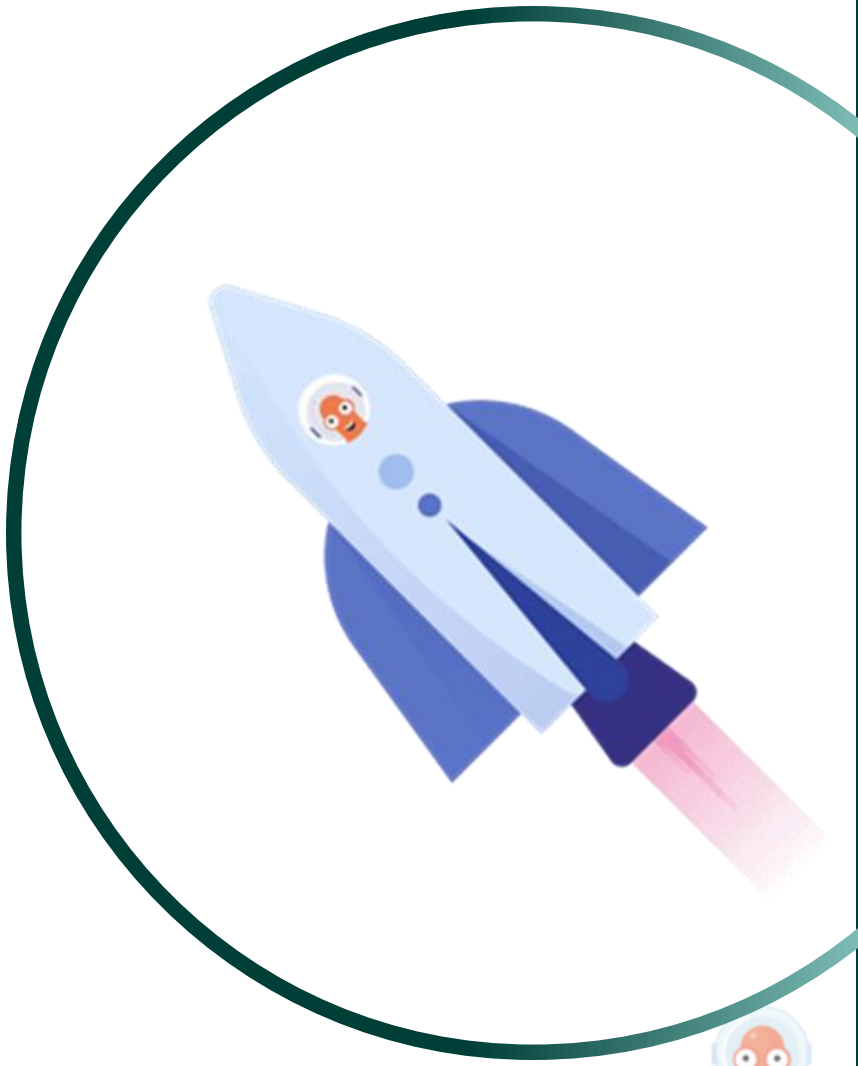
Course Objective



- GitOps Principles
- Necessity & features of Argo CD
- Argo CD Setup & Architecture
- Argo CD Applications & Projects
- Deploying Argo CD objects
- Argo CD CLI & Declarative Approach
- Application Upgrades & Rollbacks
- Integrating Private Git repositories
- Working with Webhooks

Course Objective

- Managing Multiple Kubernetes Clusters
- App of Apps Pattern
- Argo CD Monitoring
- Capstone project



Section: 1
Introduction to GitOps & Argo CD

Introduction to GitOps & Argo CD

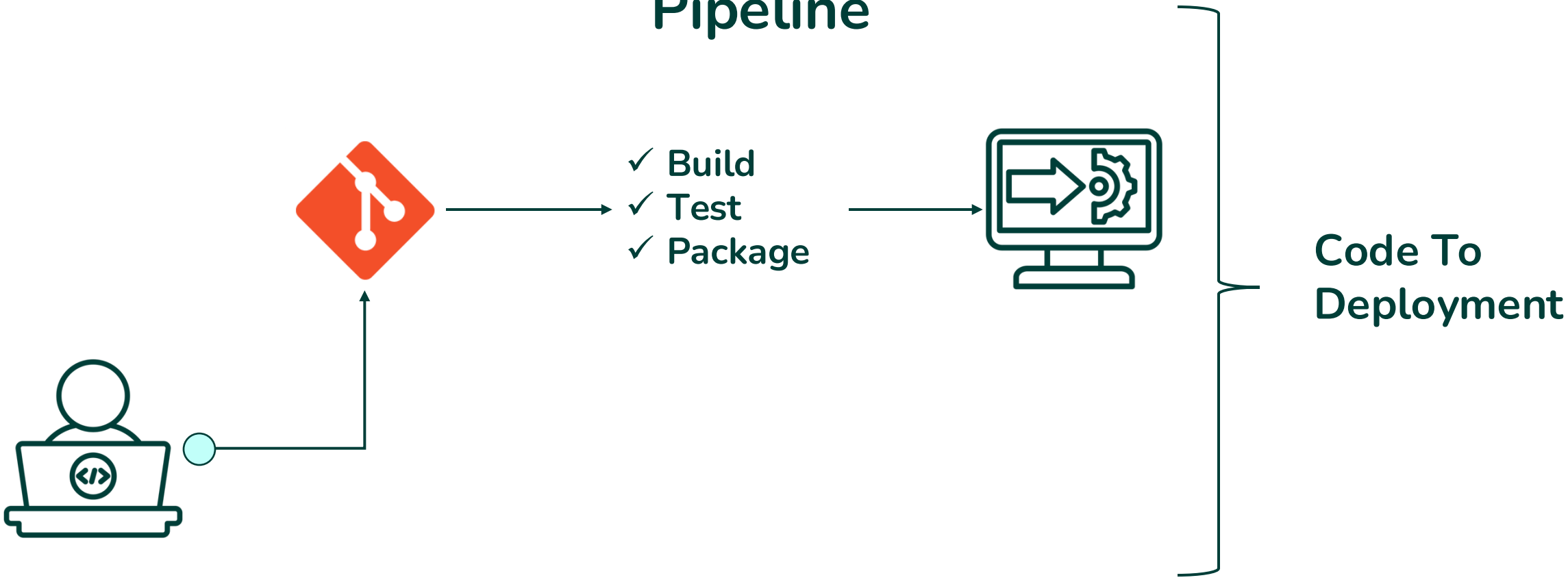
Section Overview

- Introduction to GitOps
- Argo Project Ecosystem
- Introduction to Argo CD
- Argo CD Documentation

Introduction to GitOps

GitOps

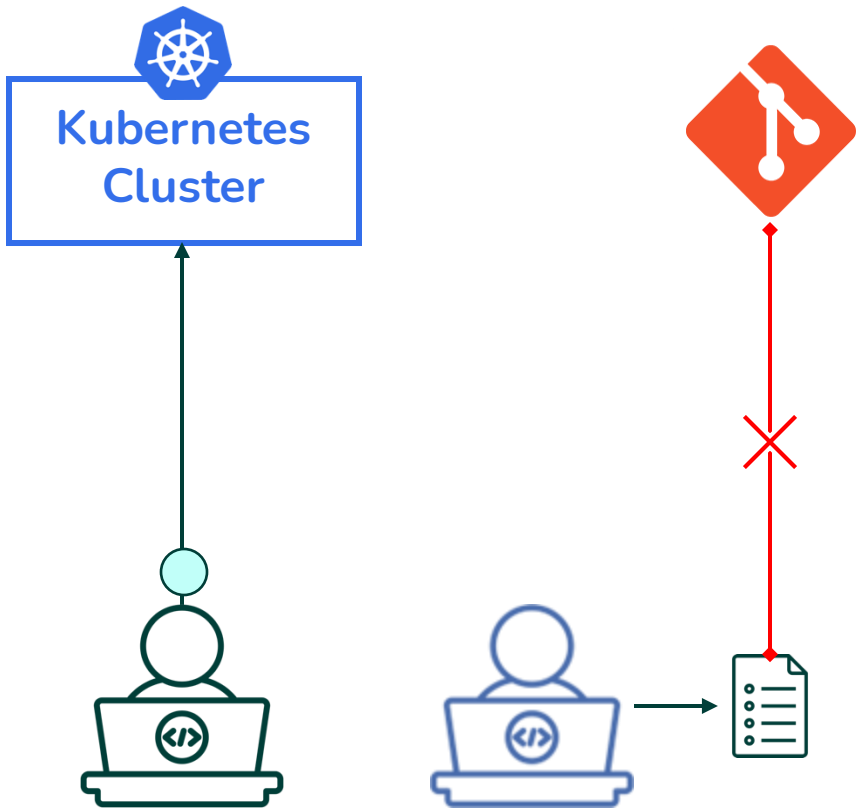
CI/CD Pipeline



Who is managing the infrastructure configurations?

How are changes to Kubernetes manifests or cluster configurations applied?

- Manually running **kubectl** commands
- Manually editing **YAML** files



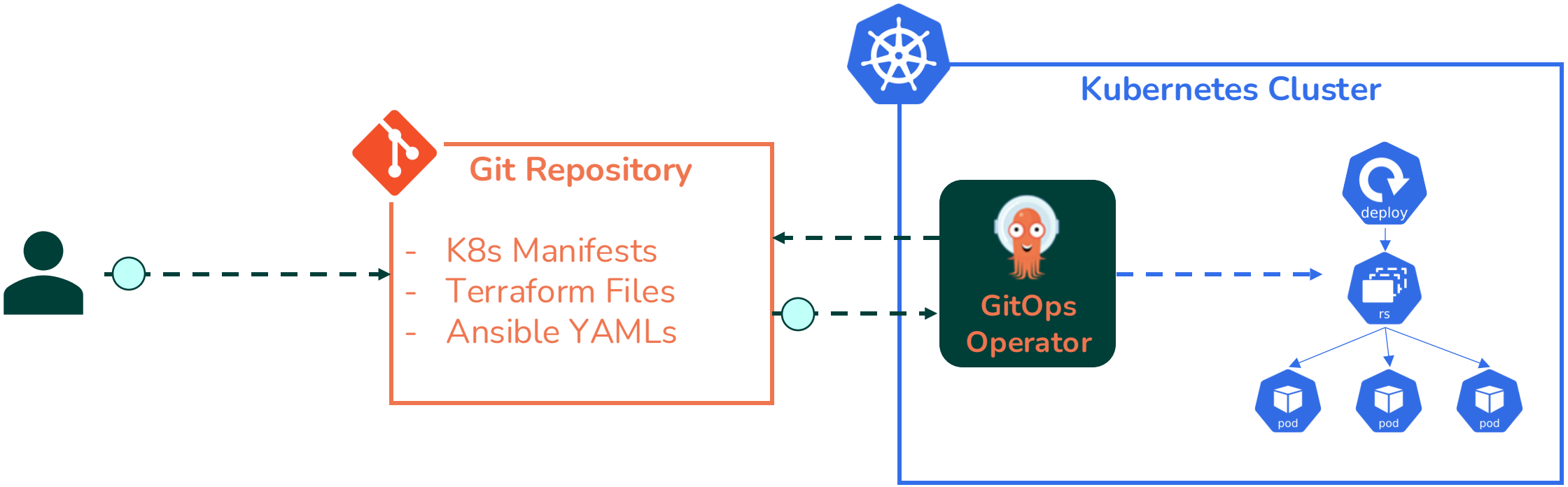
No Audit Trail

Configuration Drift

Disaster Recovery Nightmare

GitOps

Infrastructure & cluster configurations



GitOps

- ✓ Every change to cluster is stored in Git, creating a single source of truth
- ✓ Everything is automated and versioned
- ✓ In case of failure, you can restore the cluster by syncing with Git
- ✓ GitOps operator ensures the cluster's live state matches what's in Git, preventing unwanted changes

GitOps

GitOps is a modern approach to continuous deployment that uses Git as the single source of truth for your application code & infrastructure configurations

Four Principles of GitOps

Declarative Configuration

Version Control

Automated Delivery

Continuous Reconciliation

- **Observe**
- **Diff**
- **Act**

Overview of Argo Projects

Argo Projects

CLOUD NATIVE
COMPUTING FOUNDATION

Argo Projects

Argo Workflows

Workflow engine for orchestrating tasks like CI pipelines, data processing, and batch jobs

Argo Projects

Argo CD

GitOps-based continuous delivery tool, ensures Kubernetes clusters are synchronized with Git repositories, automating deployment and infrastructure management

Argo Projects

Argo Rollouts

Controller for advanced deployment strategies, including **Canary** releases, **Blue-Green** deployments, and **Progressive Rollouts**

Argo Projects

Argo Events

Event-driven automation framework that triggers workflows or actions based on external events

Argo Projects

Argo Workflows

Argo CD

Argo Rollouts

Argo Events

- ✓ Workflows
- ✓ Deployments
- ✓ Event-driven automation

Introduction to Argo CD

What is Argo CD?

Argo CD is a declarative, GitOps-based CD tool for Kubernetes that automates app deployments. It syncs the Git-defined desired state with the actual state of the cluster.

- ✓ Monitors the **current state** of applications running in the cluster
- ✓ Detects any **deviations** from the desired state stored in Git
- ✓ Provides **visualizations** and **automated reconciliation**

Why Argo CD?

Declarative & Version Controlled

Automation & Remediation

Scalability & Flexibility

How Argo CD Works?

Single source of truth

How Argo CD Works?

Syncing Applications

Monitoring State

Reconciliation Loop

Visual Interface

Features of Argo CD

Features of Argo CD

- Automated Application Deployment
- Multi-Cluster Management
- Support for Multiple Configuration Tools
- Single Sign-On (SSO) Integration
- Multi-Tenancy & RBAC
- Drift Detection & Synchronization
- Rollback & Roll-anywhere
- Health Status Analysis
- Web UI and CLI
- Webhook & Automation Integration
- Audit Trails
- Observability & Metrics

Demo

Argo CD Documentation

Documentation Demo

Argo CD - Declarative GitOps

argo-cd.readthedocs.io/en/stable/

Argo CD - Declarative GitOps CD for Kubernetes

Search

GitHub v2.13.3 18.3k 5.6k

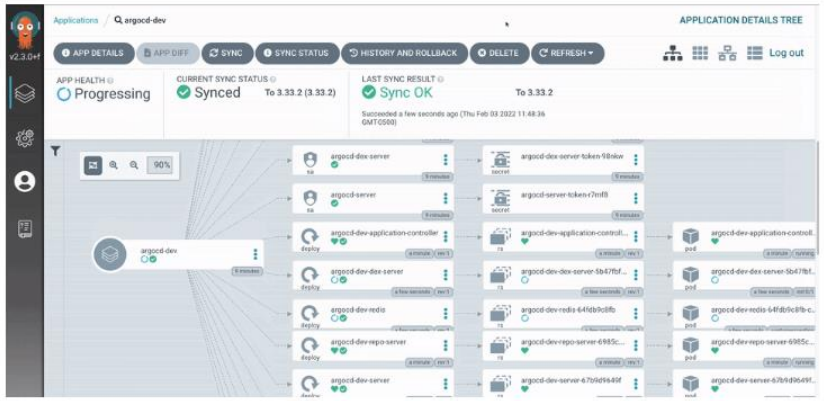
Argo CD - Declarative GitOps CD for Kubernetes

- Overview
- Understand The Basics
- Core Concepts
- Getting Started
- Operator Manual
- User Guide
- Developer Guide
- FAQ
- Security Considerations
- Support
- Roadmap
- Releases
- Blog

Overview

What Is Argo CD?

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes.



Why Argo CD?

Application definitions, configurations, and environments should be declarative and version controlled. Application deployment and lifecycle management should be automated, auditable, and easy to understand.

Getting Started

Quick Start

Table of contents

stable

Section:2 Understanding Argo CD Framework

Understanding Argo CD Framework

Section Overview

- Key Argo CD Terminologies

Core Argo CD Terminology

Continuous Delivery

GitOps

Core Argo CD Terminology

Application

Application Source Type

Target State

Live State

Sync & Sync Status

Sync Operation Status

Projects

Refresh

Health

Section:3
Setting up Argo CD

Setting up Argo CD

Section Overview

- Argo CD installation options
- Prerequisites to install Argo CD

Argo CD Installation Types

Core Installation

This setup doesn't include UI, API server and minimizes resource usage

Multi-Tenant Installation

Non-H.A. Mode

- `install.yaml`
- `namespace-install.yaml`

H.A. Mode

- `install.yaml`
- `namespace-install.yaml`

`install.yaml`		`namespace-install.yaml`
Requires cluster admin access	Access Requirements	Requires only namespace-level access
Argo CD can deploy applications on the same cluster where it runs	Application Deployment Scope	Argo CD cannot deploy apps in the same cluster where it runs but can manage and deploy apps in the other clusters

Pre-Requisites for Installing Argo CD

Kubernetes cluster

kubectl CLI installed

Demo

Installing Argo CD on Kubernetes

Demo

Argo CD
UI Walkthrough

Setting up Kubernetes Cluster

(Optional)

Section: 4
Argo CD Architecture

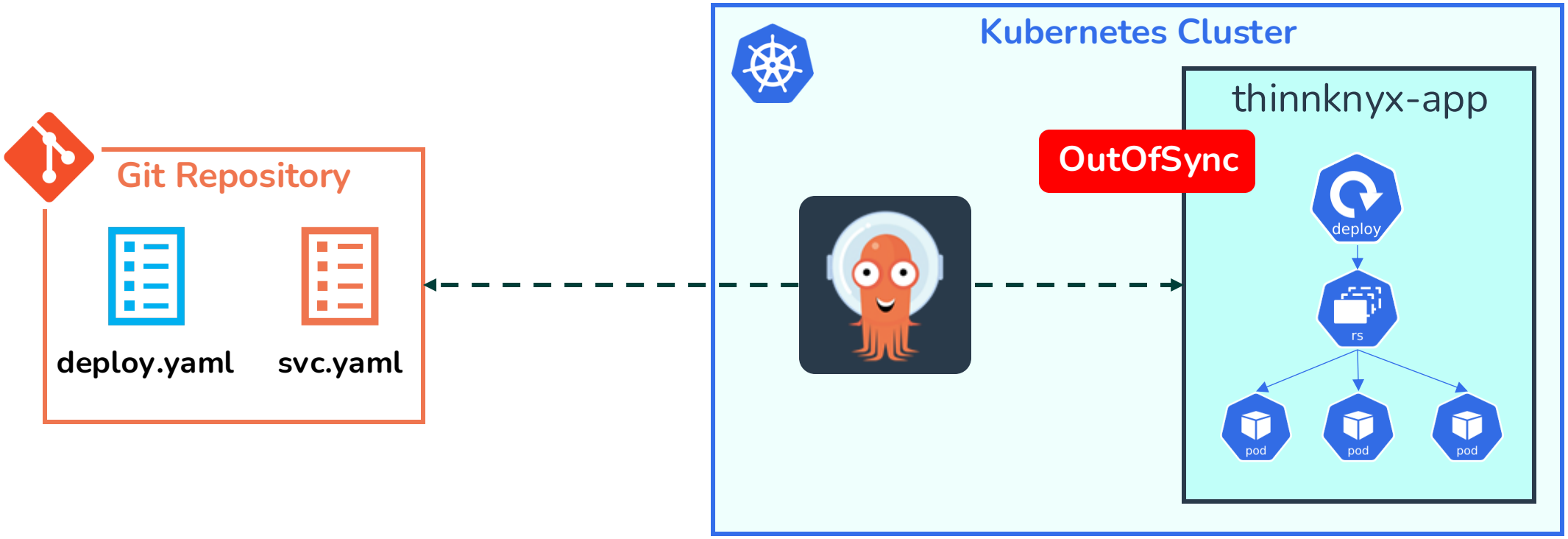
Argo CD Architecture

Section Overview

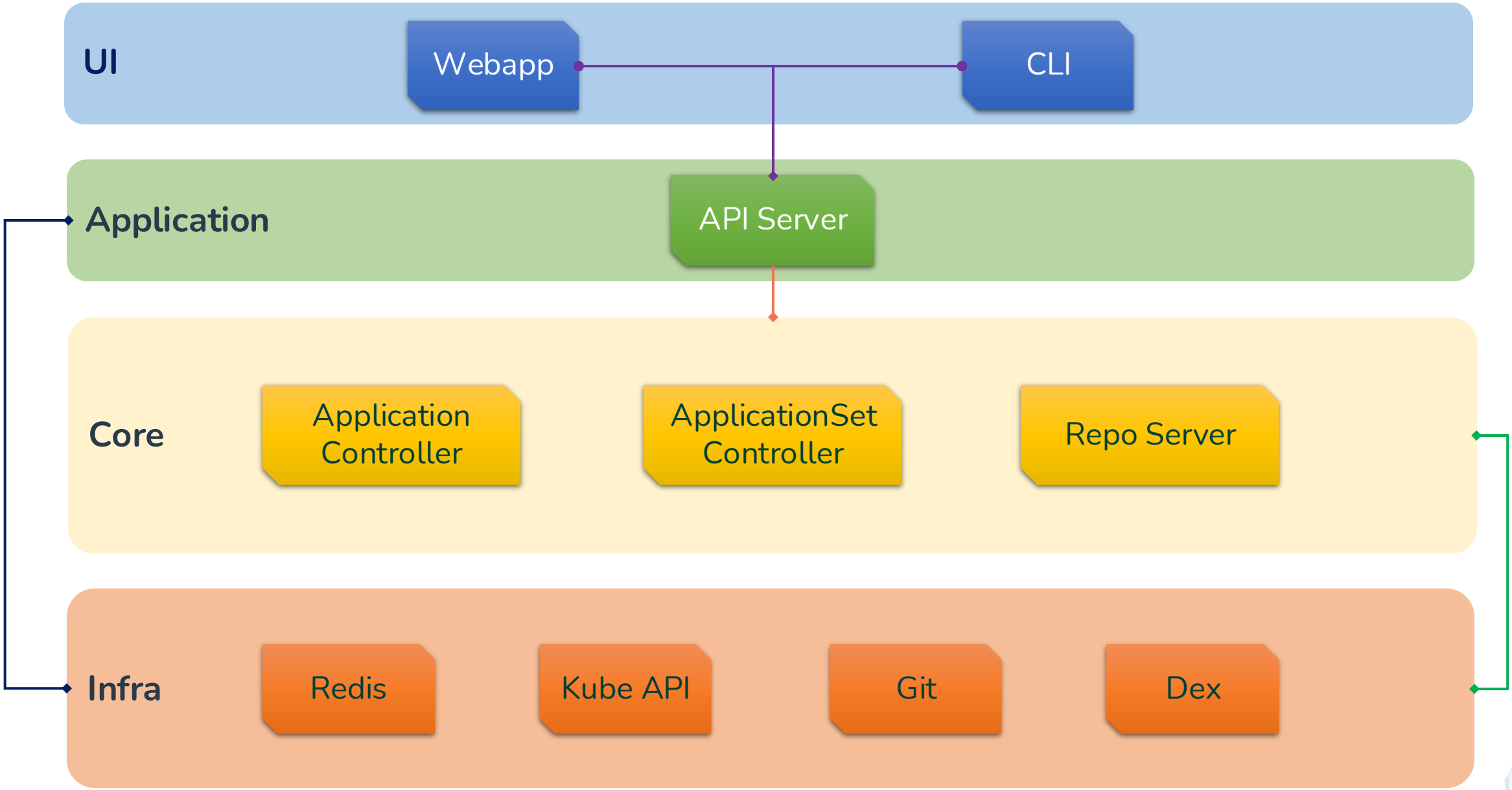
- Understanding Argo CD Architecture

Argo CD Architecture

Argo CD operates as a Kubernetes controller designed to enforce the principles of GitOps



Component-based Architecture



Section: 5
Argo CD Applications

Argo CD Applications

Section Overview

- Introduction to Argo CD Applications
- Argo CD with UI, CLI and Manifests
- Demonstration:
  - Application Deployment
  - Auto Namespace Creation
  - Sync Options
  - Prune Sync
  - Non-cascade Deletion

Introduction to Argo CD Application

An application in Argo CD is a custom resource that represents a deployed instance of Kubernetes resources in a cluster

Key Components

- Source
- Destination
- Sync Process

Application Deployment Methods: UI, CLI, & Manifest

Application Modes

UI (Console)

CLI

Manifests

Application Modes

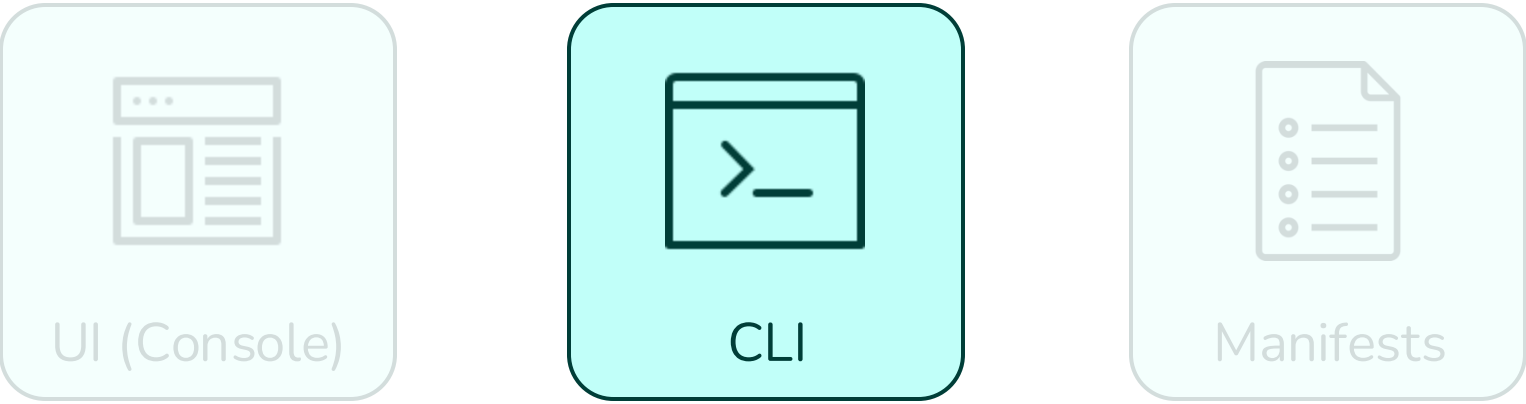
UI (Console)

CLI

Manifests

The Argo CD UI offers an easy-to-use graphical interface for managing applications, allowing users to create, monitor, and manage apps with minimal effort

Application Modes



The Argo CD CLI allows users to create and manage applications efficiently using commands, ideal for automation and terminal-based workflows

Application Modes

Applications can be defined using YAML manifests, allowing users to manage configurations as code for version control and collaboration

Application Modes

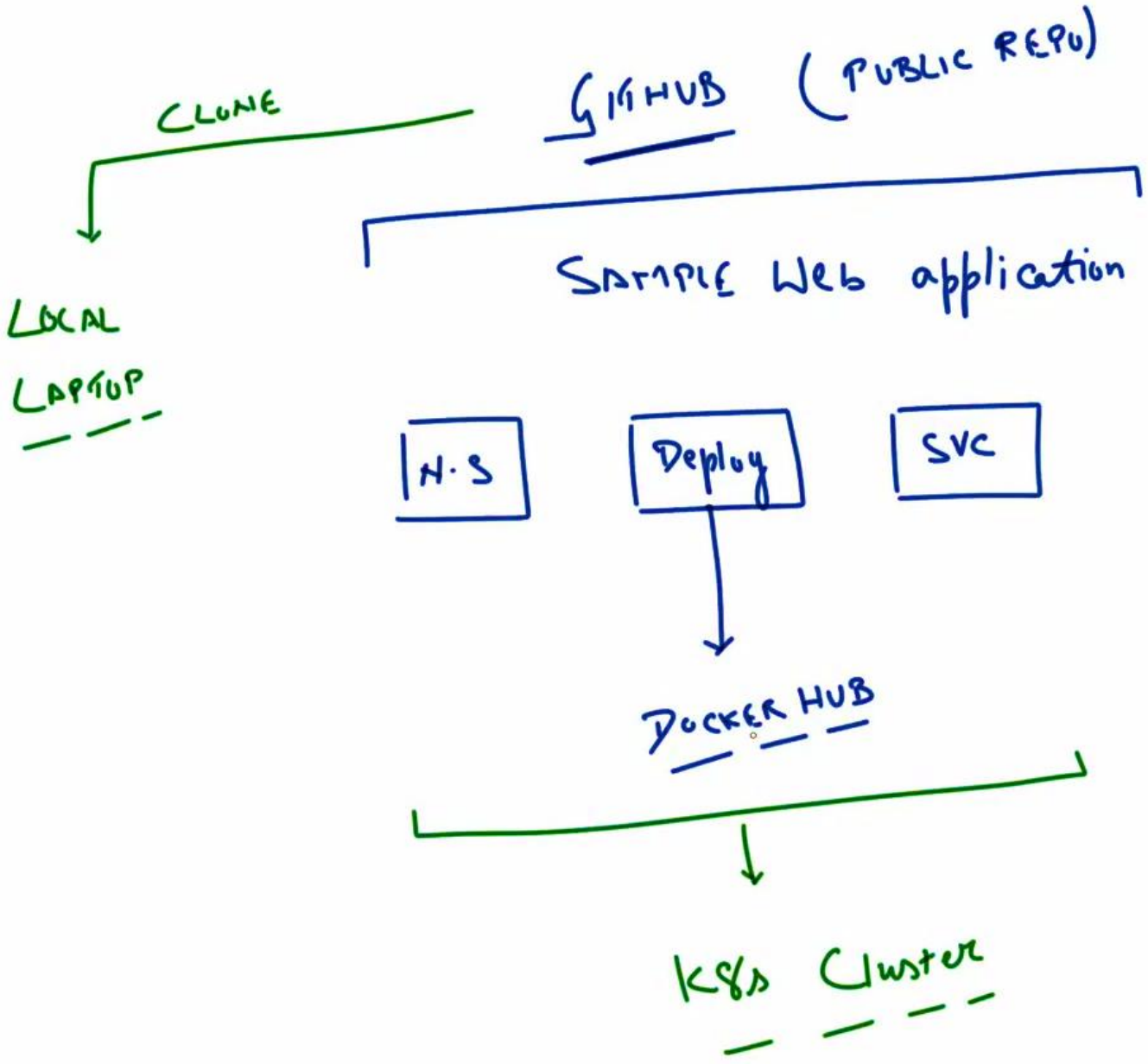
UI (Console)

CLI

Manifests

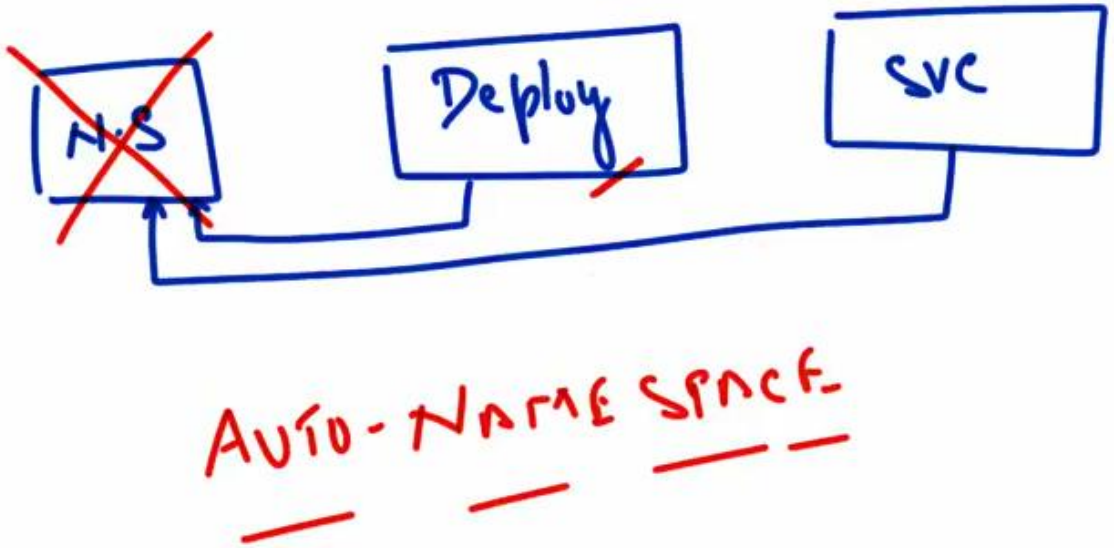
Demo

First Application Deployment Using Argo CD



Demo

Auto-namespace With Application Deployment

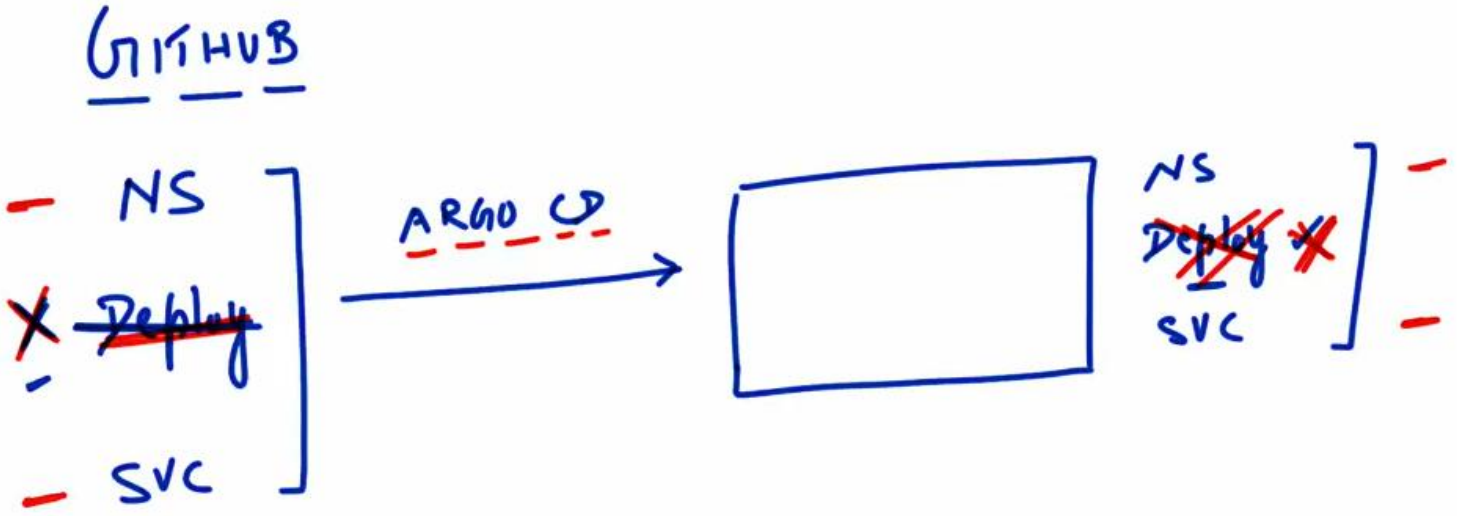


Demo

Sync Options For Argo CD Applications

Demo

Prune Sync Option With Argo CD Applications



CASE1 NS, Deploy, SVC → Remain (OUT OF SYNC)

CASE2 PRUNE (SYNC)

Demo

Argo CD Application Deletion In Non-cascade Mode

Section: 6
Argo CD Projects

Argo CD Projects

Section Overview

- Introduction to Argo CD Projects
- Demonstration:
  - Project Creation in Argo CD
  - Deploying & Troubleshooting Argo CD Applications in Custom Project

Argo CD Projects

Provides a way to logically group applications, especially in multi-team environments

- Source Restrictions
- Destination Restrictions
- Resource Restrictions
- Access Control

Default Project

✓ Can be modified

x Can't be deleted

Demo

Project Creation in Argo CD

Demo

Deploying & Troubleshooting Argo CD Applications in Custom Project

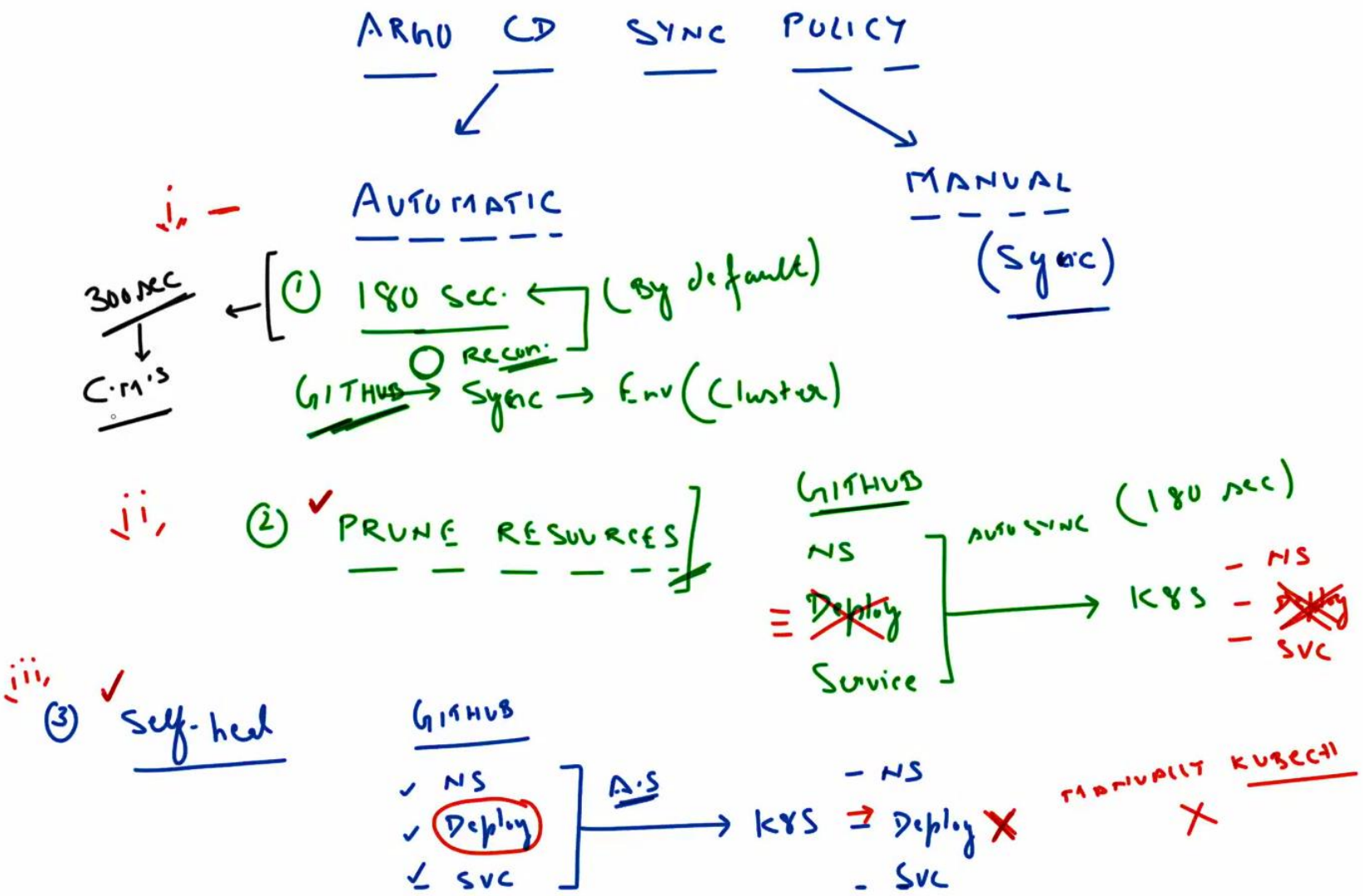
Section: 7
Automatic Sync Policy in Argo CD

Automatic Sync Policy in Argo CD

Section Overview

- Automatic sync policy in Argo CD
- Demonstrations:
  - Application deployment
  - Resource pruning
  - Self-healing
  - Configuring sync reconciliation time

Argo CD Sync Policy



Demo

Application Deployment With Automatic Sync

Demo

Prune Resources Option With Automatic Sync

Demo

Self-healing Option With Automatic Sync

Demo

Configure Automatic Sync Reconciliation Time

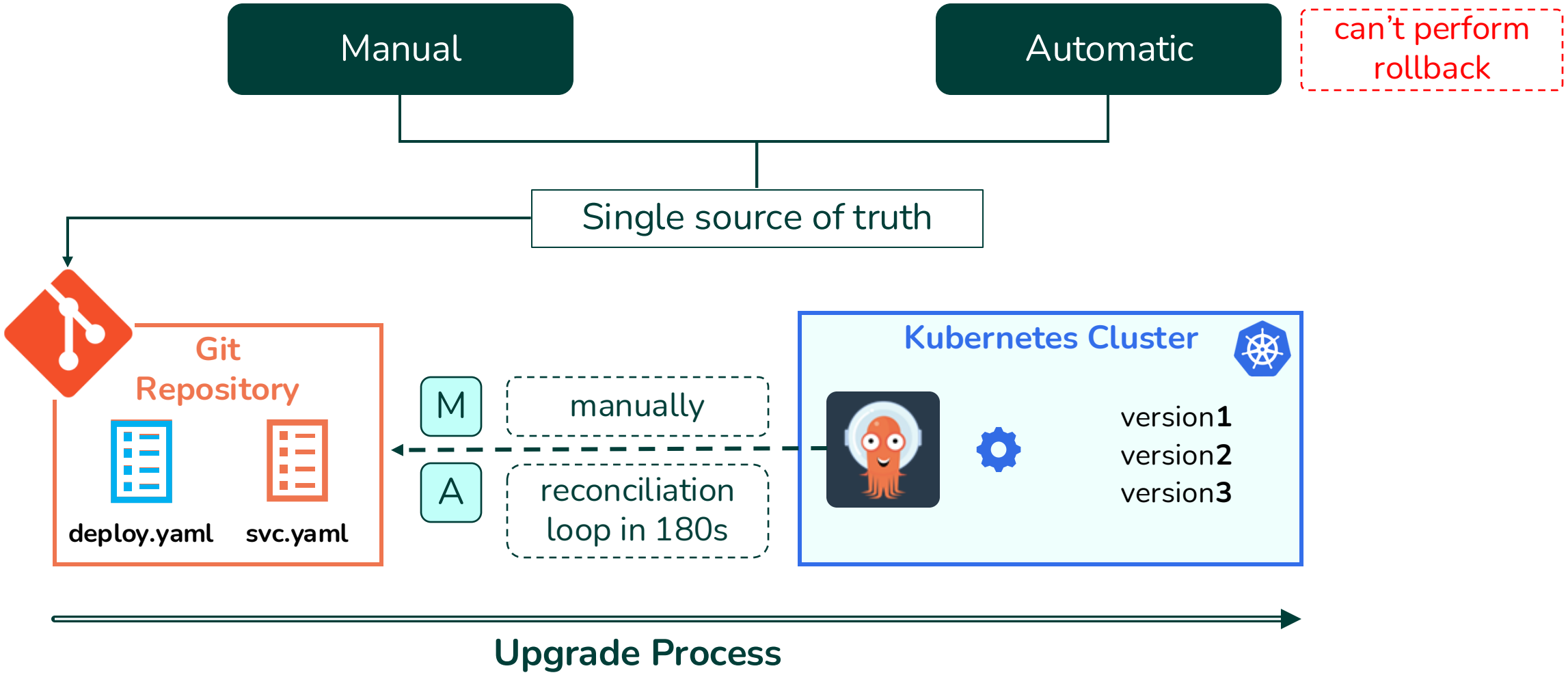
Section: 8 Upgrades and Rollbacks using Argo CD

Upgrades and Rollbacks using Argo CD

Section Overview

- Upgrades and Rollbacks
- Demonstration:
  - Upgrades and Rollbacks

Sync Types



Manual

Automatic

Argo CD Application

HISTORY AND ROLLBACK

version**4**	rolled back version	**u7dd8sl**
version**3**	current version	**9de71b2**
version**2**	second last version	**3sdj67s**
version**1**	third last version	**u7dd8sl**

all version **commit id**

Manual

Automatic

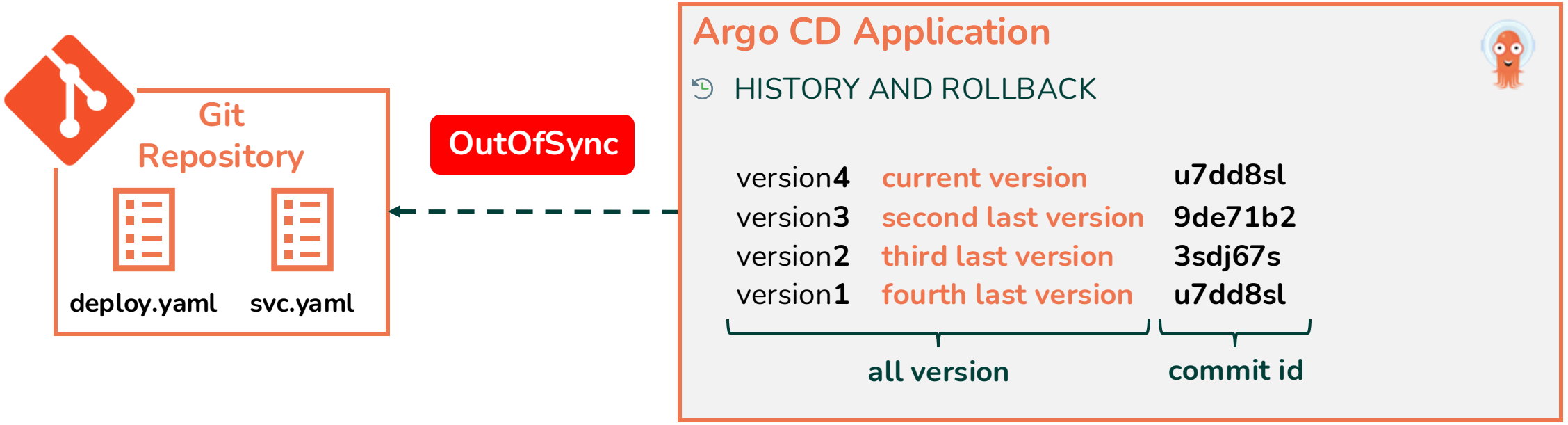
Argo CD Application

HISTORY AND ROLLBACK

version**4**	current version	**u7dd8sl**
version**3**	second last version	**9de71b2**
version**2**	third last version	**3sdj67s**
version**1**	fourth last version	**u7dd8sl**

all version **commit id**

Manual



Demo

Upgrade & Rollback In Argo CD

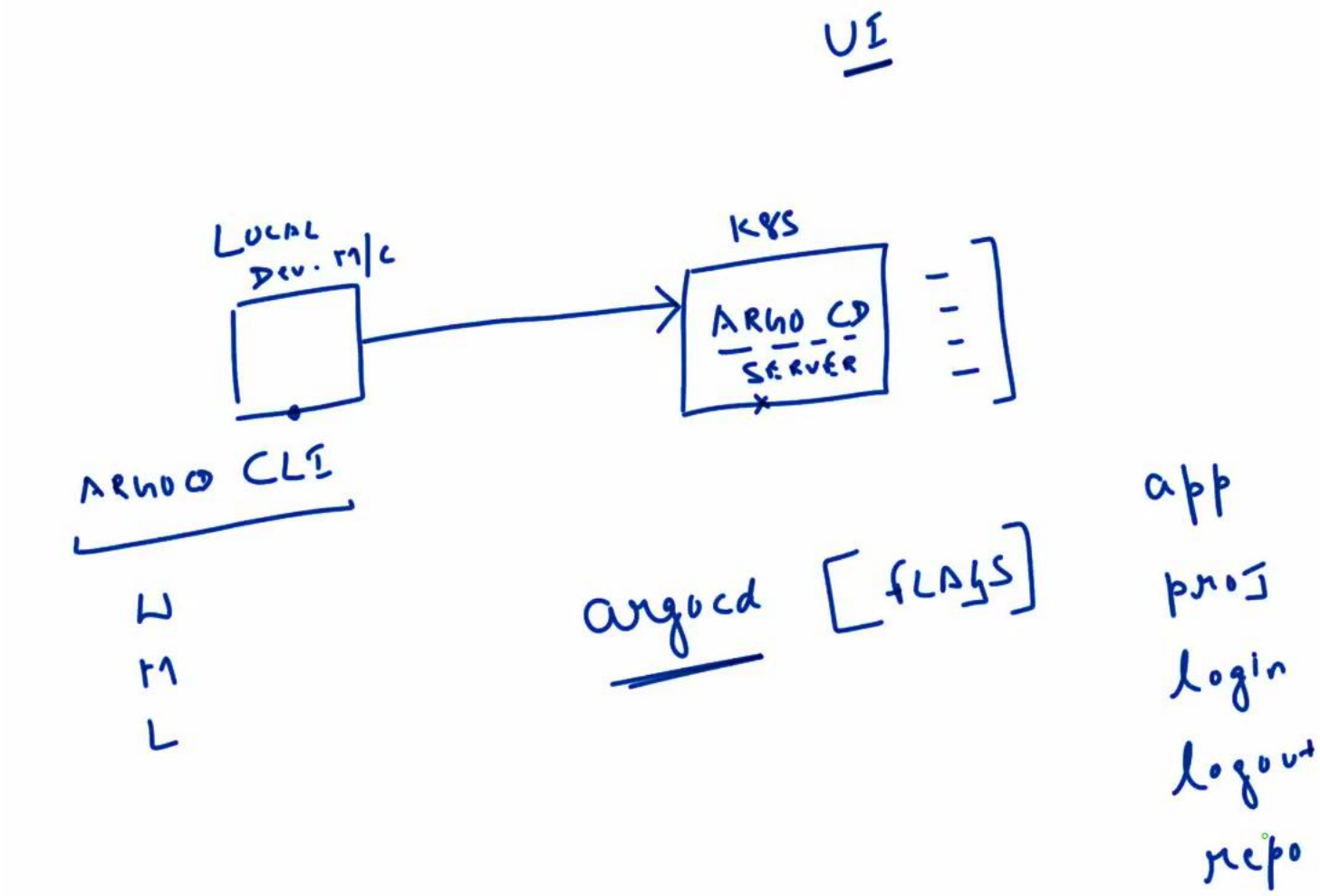
Section: 9
Command Line Interface (CLI)

Command Line Interface (CLI)

Section Overview

- Understanding Argo CD CLI
- Demonstration:
  - Argo CD CLI Installation
  - Application Lifecycle using Argo CD CLI
  - Project Lifecycle Operations using Argo CD CLI
  - Auto Sync Policy using Argo CD CLI

Argo CD CLI



Demo

Argo CD CLI Installation

Demo

Application Lifecycle Using Argo CD CLI

Demo

Project Lifecycle Operations Using Argo CD CLI

Demo

Auto Sync Policy Using Argo CD CLI

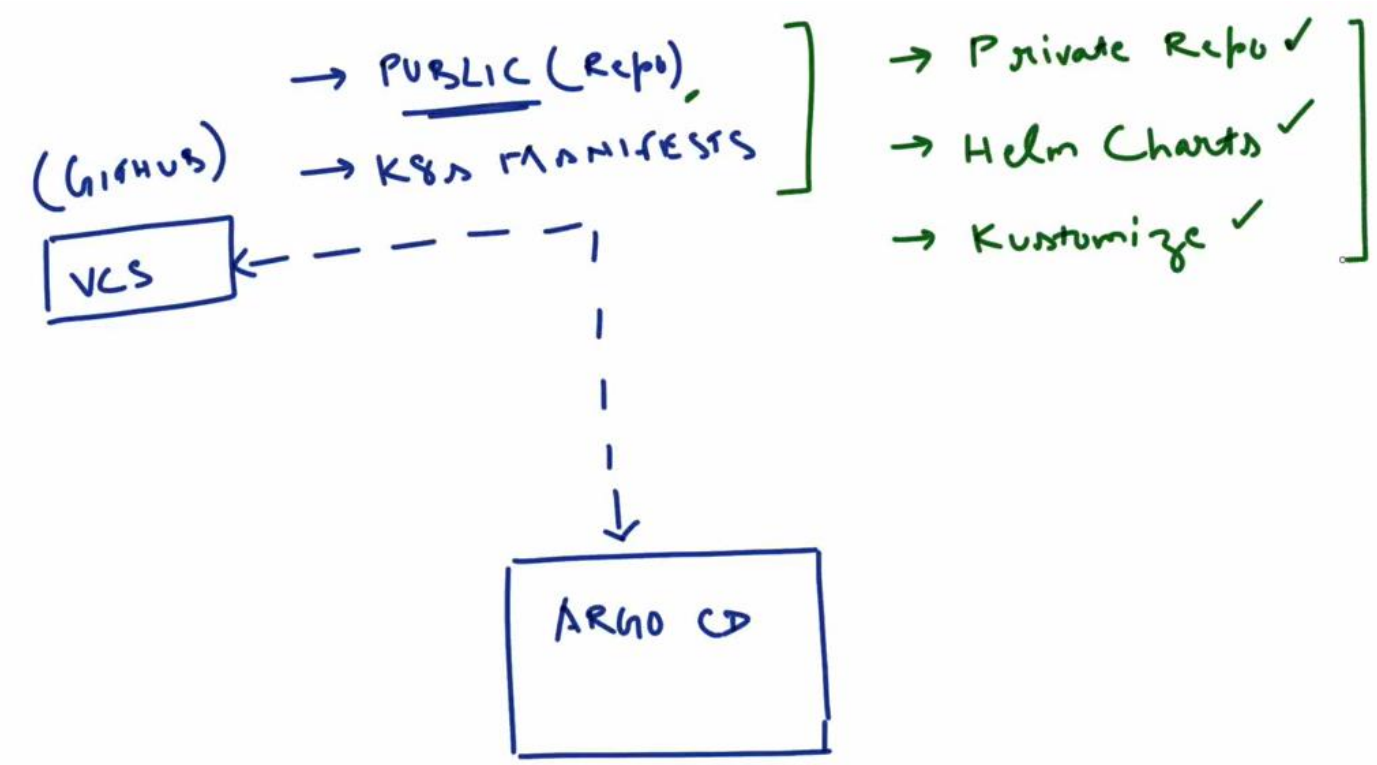
Section: 10 Application Deployment using Helm, Kustomize & Private Repositories

Application Deployment using Helm, Kustomize & Private Repositories

Section Overview

- Application Deployment using Helm
- Kustomize
- Private Repositories
- Demonstration:
  - Application Deployment using Helm (Argo CD CLI & UI)
  - Application Deployment using Kustomize
  - Git Private Repository Integration (https & ssh)

Application Deployment using various methods



Demo

Application Deployment Using Helm (Argo CD CLI)

Demo

Application Deployment Using Helm (Argo CD UI)

Demo

Application Deployment Using Kustomize

Demo

Git Private Repository Integration (https & ssh)

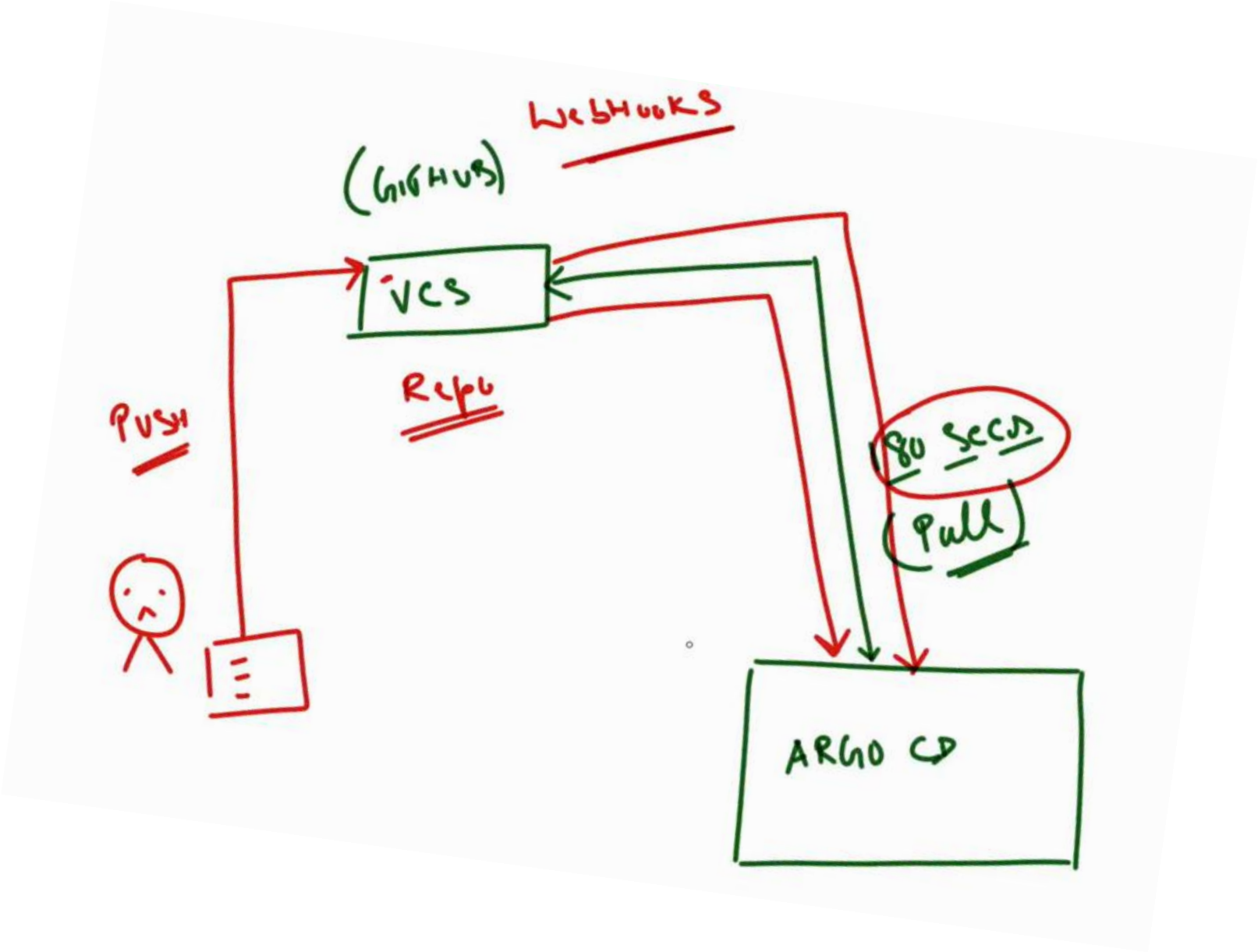
Section: 11 Webhooks with Argo CD

Webhooks with Argo CD

Section Overview

- Configuring Webhooks with Argo CD
- Demonstration:
  - Webhooks with Argo CD

Introduction to webhooks



Demo

Webhooks With Argo CD

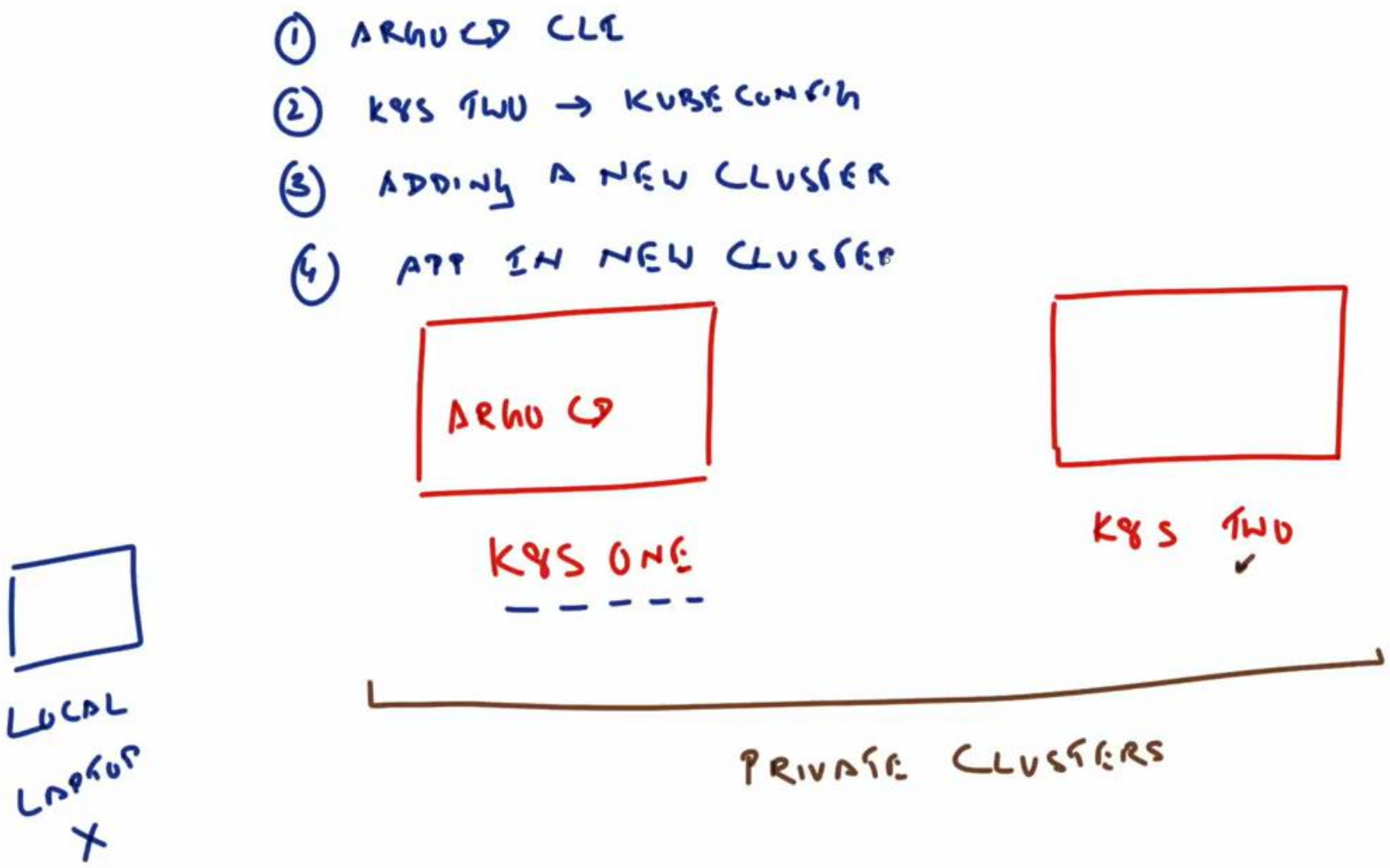
Section: 12
Managing Multiple Clusters with Argo CD

Managing Multiple Clusters with Argo CD

Section Overview

- Adding Multiple Clusters
- Demonstration:
  - Managing Multiple Clusters with Argo CD

Introduction to webhooks



Demo

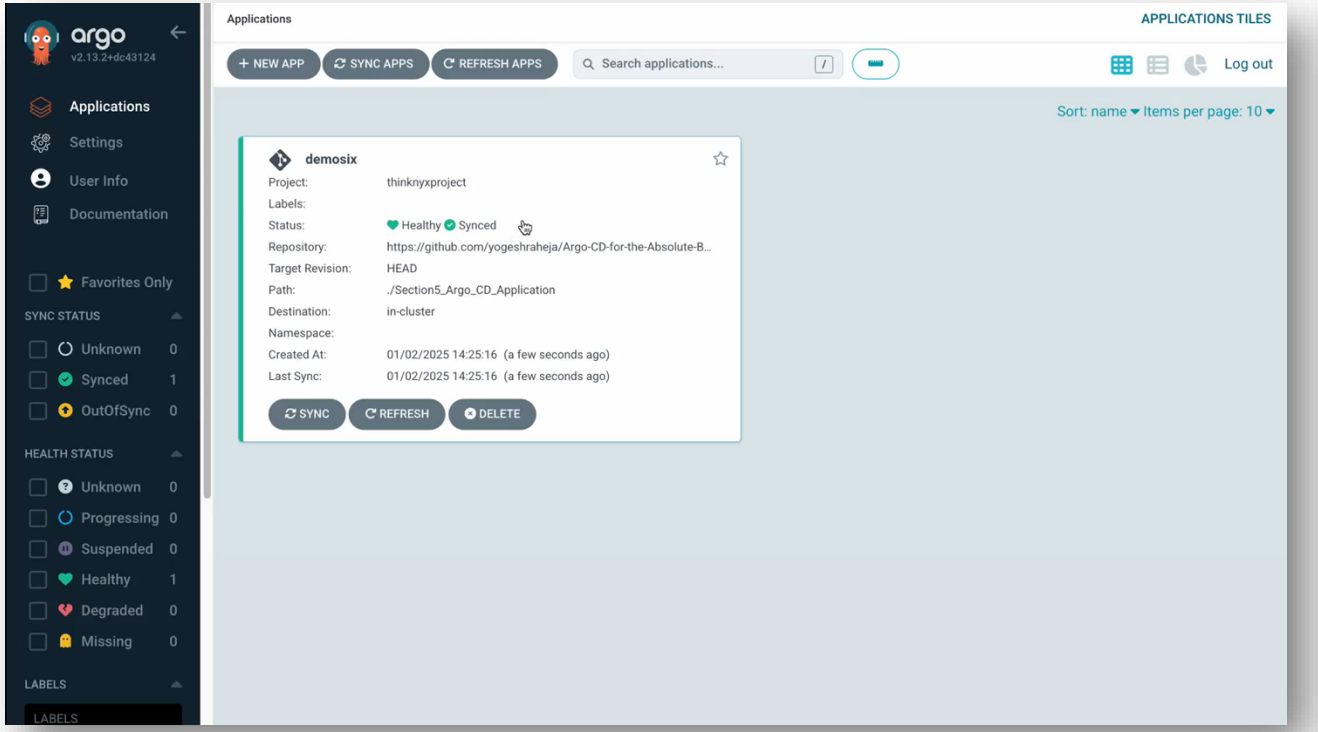
Managing Multiple Clusters With Argo CD

Section: 13 Declarative Approach for Argo CD

Declarative Approach for Argo CD

Section Overview

- Argo CD applications with Declarative Approach

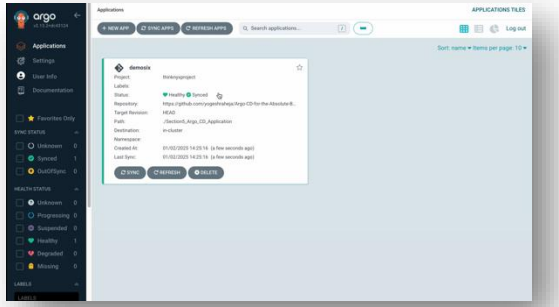


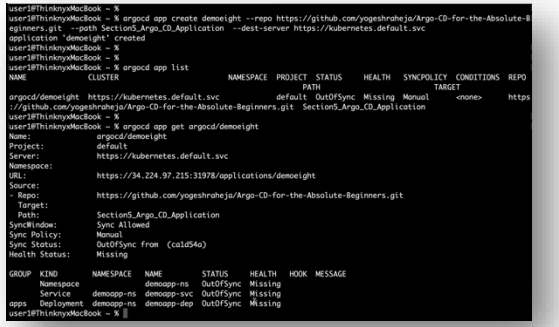
Argo CD UI

```
user1@ThinknyxMacBook ~ %
user1@ThinknyxMacBook ~ % argocd app create demoeight --repo https://github.com/yogeshraheja/Argo-CD-for-the-Absolute-B
eginners.git  --path Section5_Argo_CD_Application  --dest-server https://kubernetes.default.svc
application 'demoeight' created
user1@ThinknyxMacBook ~ %
user1@ThinknyxMacBook ~ %
user1@ThinknyxMacBook ~ % argocd app list
NAME              CLUSTER                         NAMESPACE  PROJECT  STATUS     HEALTH   SYNCPOLICY  CONDITIONS  REPO
                                                                      PATH                          TARGET
argocd/demoeight  https://kubernetes.default.svc  default    OutOfSync  Missing  Manual      <none>      https
://github.com/yogeshraheja/Argo-CD-for-the-Absolute-Beginners.git  Section5_Argo_CD_Application
user1@ThinknyxMacBook ~ %
user1@ThinknyxMacBook ~ % argocd app get argocd/demoeight
Name:               argocd/demoeight
Project:            default
Server:             https://kubernetes.default.svc
Namespace:
URL:                https://34.224.97.215:31978/applications/demoeight
Source:
- Repo:             https://github.com/yogeshraheja/Argo-CD-for-the-Absolute-Beginners.git
  Target:
  Path:             Section5_Argo_CD_Application
SyncWindow:         Sync Allowed
Sync Policy:        Manual
Sync Status:        OutOfSync from  (ca1d54a)
Health Status:      Missing

GROUP  KIND        NAMESPACE   NAME         STATUS     HEALTH   HOOK  MESSAGE
       Namespace               demoapp-ns   OutOfSync  Missing
       Service     demoapp-ns  demoapp-svc  OutOfSync  Missing
apps   Deployment  demoapp-ns  demoapp-dep  OutOfSync  Missing
user1@ThinknyxMacBook ~ %
```

Argo CD CLI

Declarative Approach





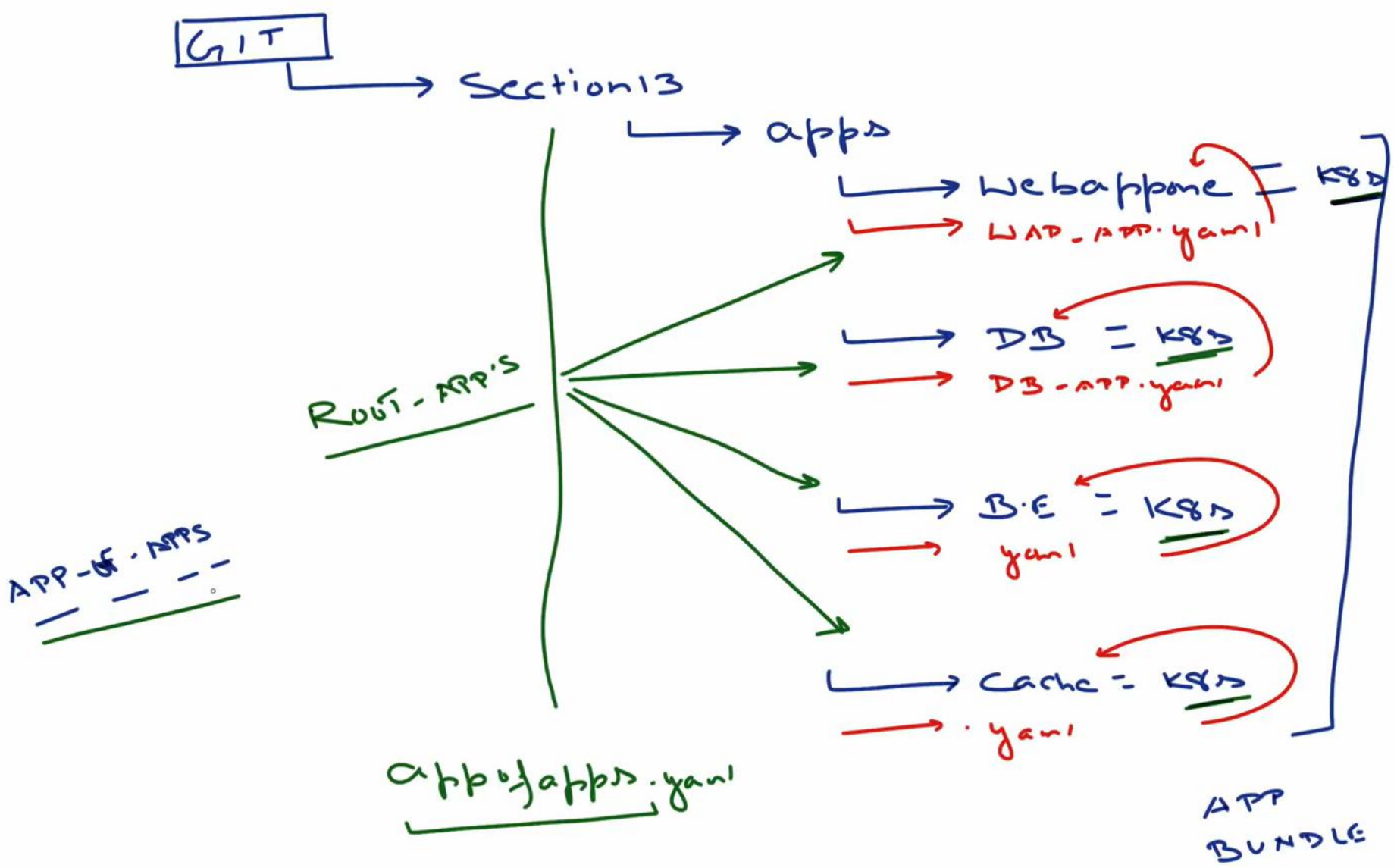

```
user1@Thinknyx-MacBook ~ % cat argocd_declarative_examples/apps/webappone_application.yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: webappone
  namespace: argocd
spec:
  destination:
    namespace: webappone
    name: in-cluster
  syncPolicy:
    automated: {}
    syncOptions:
      - CreateNamespace=true

  project: default

  source:
    repoURL: https://github.com/yogeshraheja/argocd_declarative_examples.git
    path: apps/webappone
    targetRevision: HEAD
user1@Thinknyx-MacBook ~ %
```

Demo

Application Deployment Using Declarative Way

What are App of Apps?

What are App of Apps?



Demo

App of Apps in Argo CD

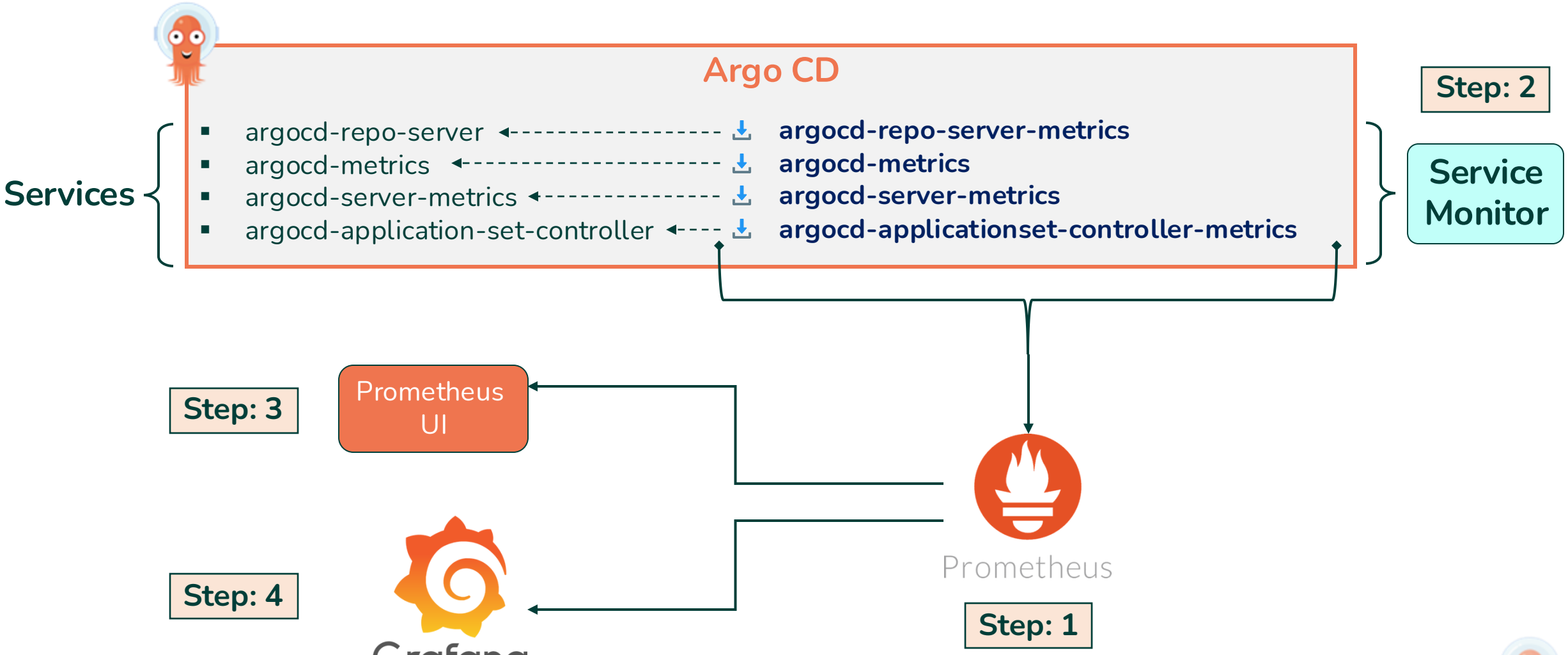
Section: 14
Argo CD Monitoring

Argo CD Monitoring

Section Overview

- Argo CD Metrics & Monitoring with Prometheus
- Demonstration:
  - Argo CD Monitoring using Prometheus and Grafana

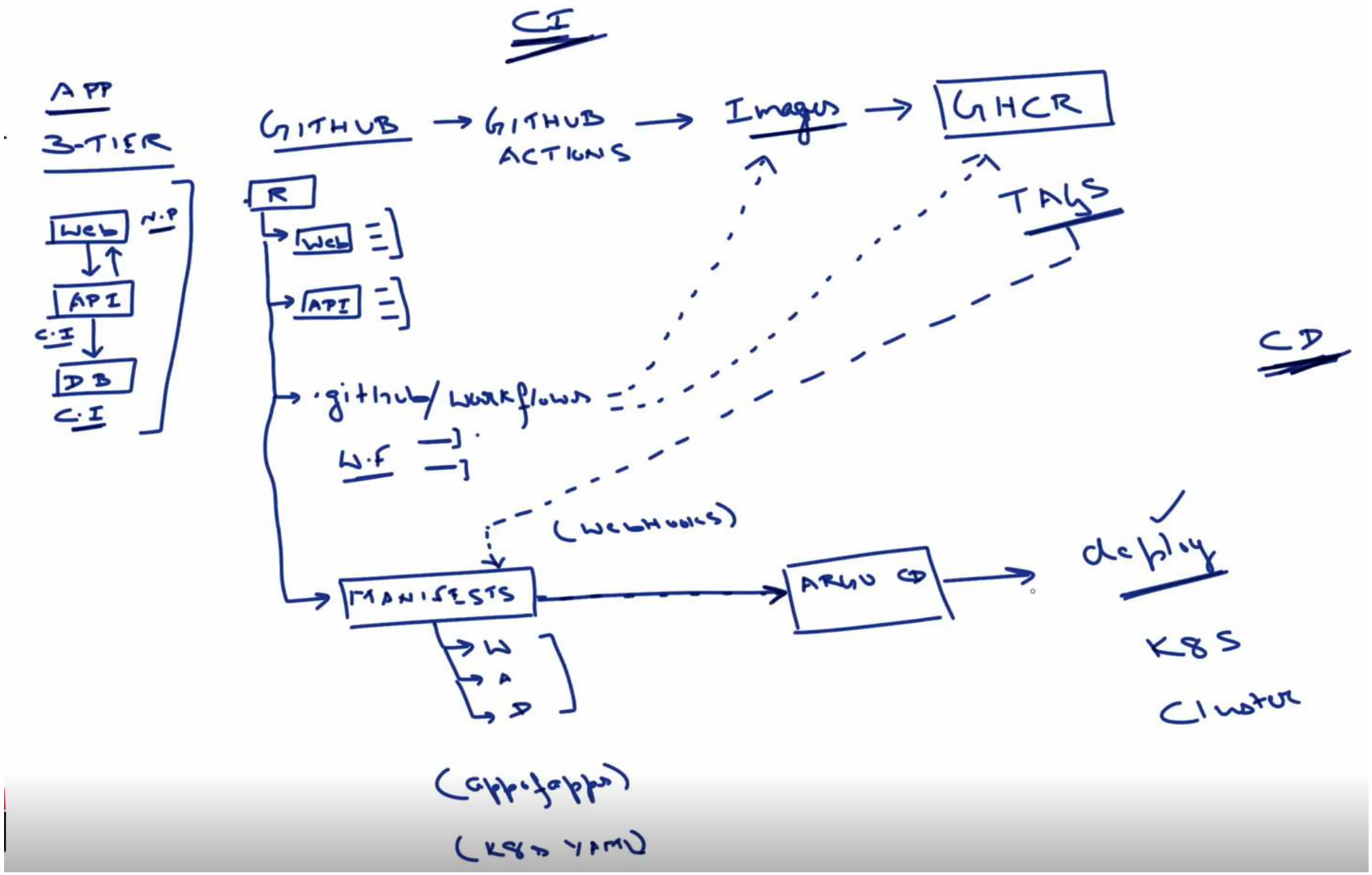
Argo CD Monitoring



Demo

Argo CD Monitoring Using Prometheus & Grafana

Section: 15
Overview of Capstone Project



Demo

Capstone Project

Conclusion

Thinknyx®

You Trust, We Deliver!

Argo CD for the Absolute Beginners Hands-On

Deploy

Manage

Troubleshoot

Advanced Sync Configurations

HELM

Kustomize.io

Follow us on:

@thinknyx

@thinknyx

@thinknyx-technologies

@thinknyx

@thinknyx-technologies