Argo CD 101 Workshop

Understanding GitOps and Continuous Delivery using Argo CD and Helm

$ whoami

Nicholas Morey (@morey_tech)

a Platform Engineer with a passion for DevOps practices.


AKUITY

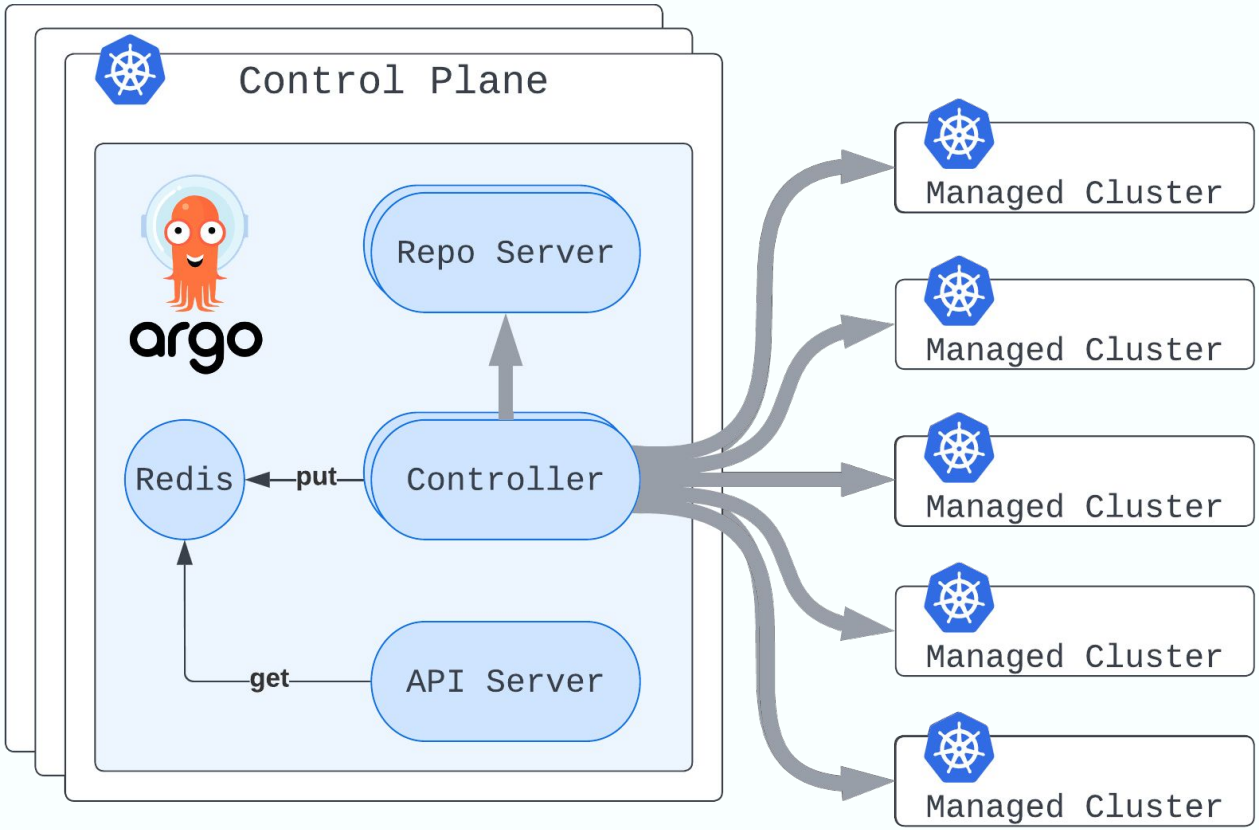
Argo CD

OSS Architecture

- $ Expensive traffic
- $ Maintenance cost

- 📣 Requires VPN or public ip

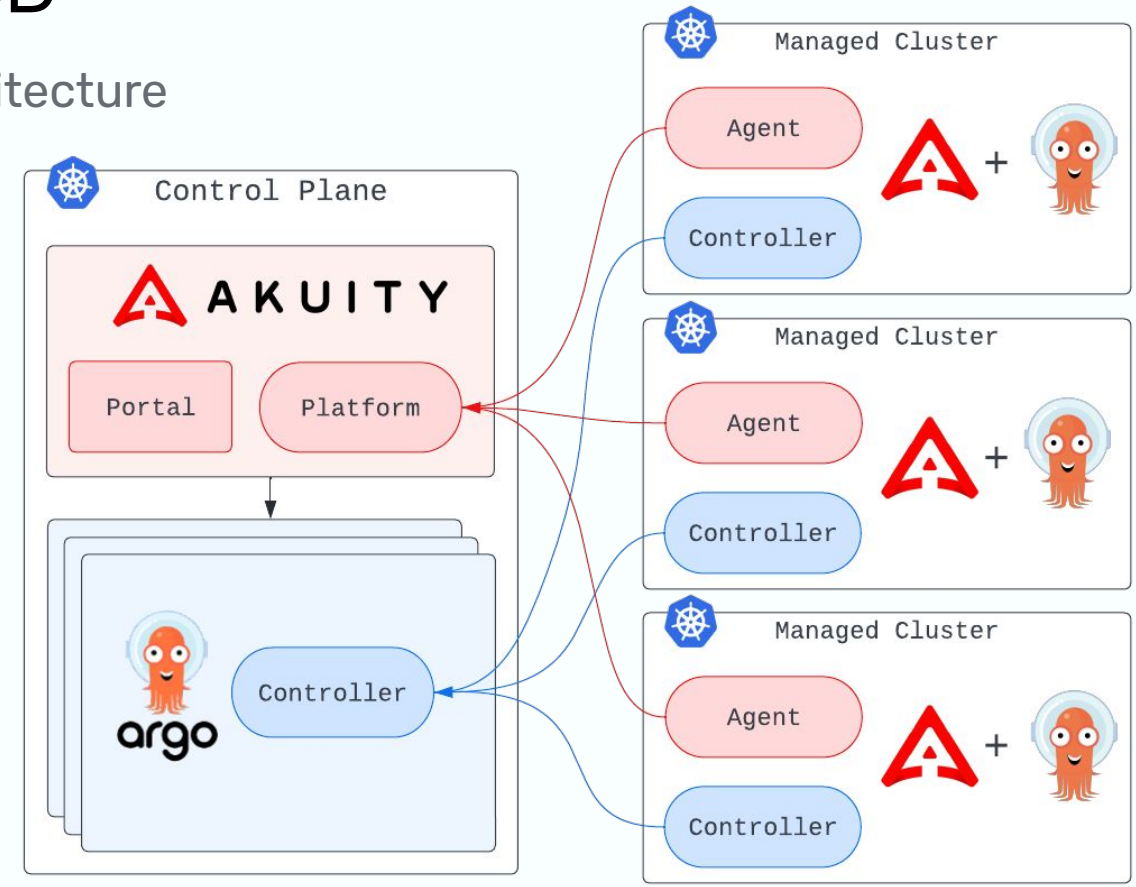
- 👓 Scaling might be challenging
- 👓 Multiple control planes



Managed Argo CD

Agent based hybrid architecture

- $ Minimal traffic reduces cost
- $ Maintenance is included
- 🔒 No external cluster access
- 🔒 No cluster credentials
- 👓 Greatly improved scalability
- 👓 Single control plane



Workshop Topics

- Introduction to Continuous Delivery and GitOps practices.
- Argo CD core concepts.
- Deploying Helm charts with Argo CD Applications.
- Deploying Applications declaratively.
- Discussion on best practices.

After the workshop, you will receive a certificate of completion and a digital badge.

What is Continuous Delivery?

Depends on who you ask.

What is Continuous Delivery?

“Continuous delivery (CD) is the process of automating build, test, configuration, and deployment from a build to a production environment.”

What is Continuous Delivery?

“Continuous delivery is a software development practice where code changes are automatically prepared for a release to production.”

What is Continuous Delivery?

“Continuous delivery lets development teams automate the process that moves software through the software development lifecycle”

What is Continuous Delivery?

“Continuous delivery is a software development practice that uses automation to speed the release of new code.”

What is Continuous Delivery?

“Continuous Delivery is the ability to get changes of all types — including new features, configuration changes, bug fixes and experiments — into production, or into the hands of users, safely and quickly in a sustainable way.”

~ Jez Humble,
continuousdelivery.com

What is Continuous Delivery?

“Continuous delivery (CD) is a software development practice that aims to release software changes frequently and reliably.”

What is Continuous Delivery?

“Continuous Delivery (CD) is the software development practice of automating the release process, from once the build is complete to running in production.”

What CD is not, is CI.

- Continuous Integration (CI)
- Regularly merging code.
- Automated testing and packaging.
- “Integrating” the build is CD.

Blurred Lines

- How the source is packaged.
- Container Image versus NPM module.

The other CD

- Continuous Deployment (CD?)
- Hands-off deployments from merge to production

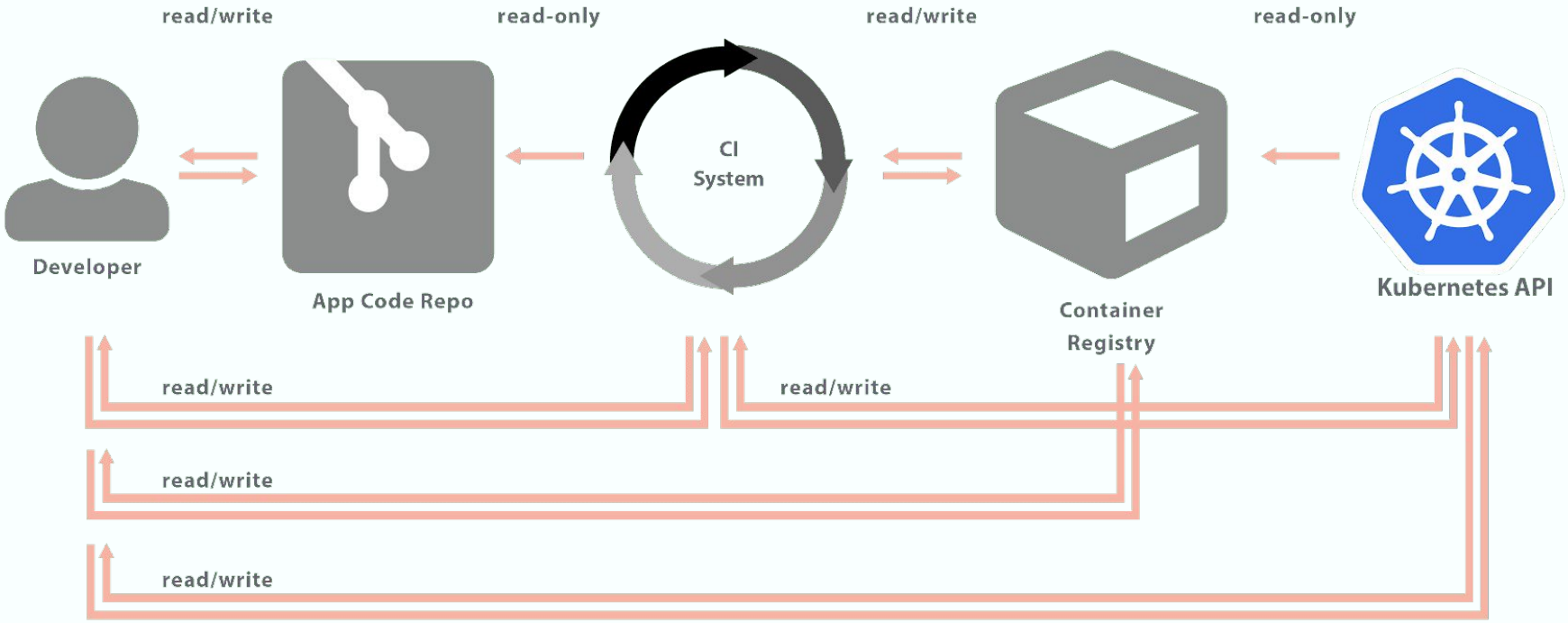
What is GitOps?

- ~~Depends on who you ask.~~
- Ask **opengitops.dev**

The Four GitOps Principles (v1.0.0)

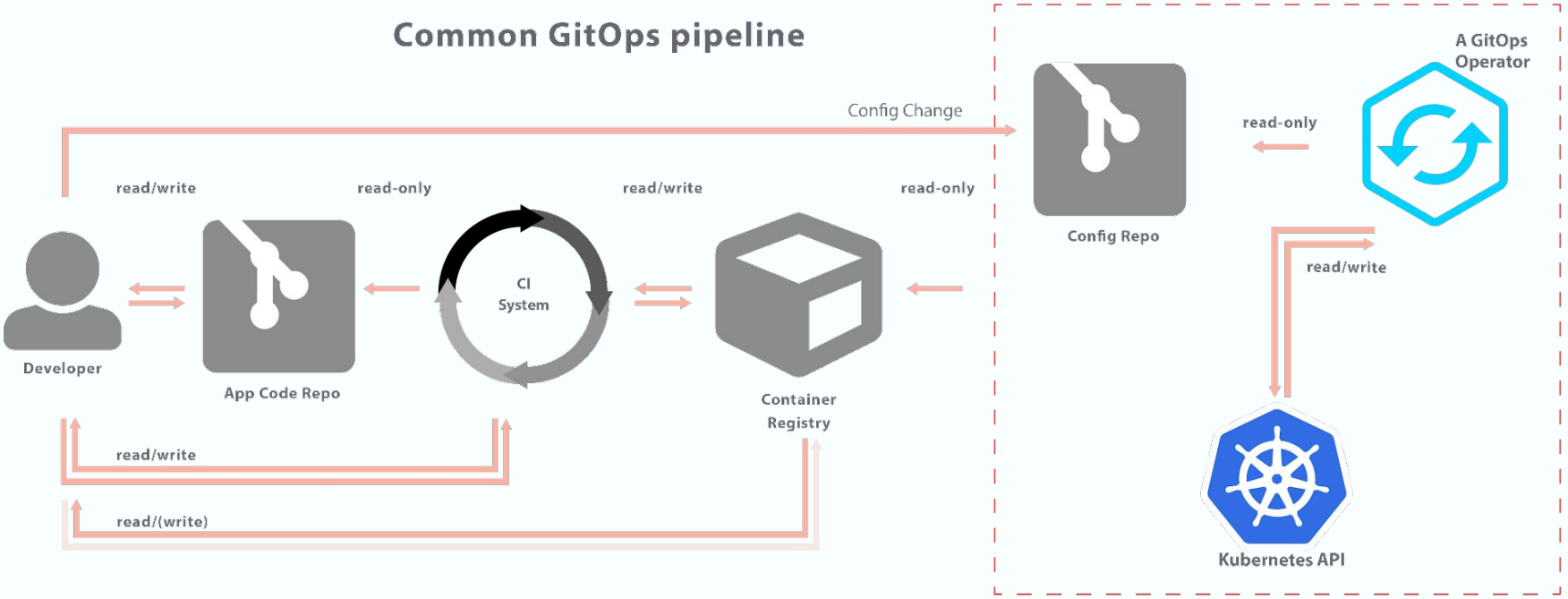
- Declarative desired state.
- Immutable desired state versions.
- Continuous state reconciliation.
- Operations through declaration.

What Happens Without GitOps?



https://www.weave.works/blog/kubernetes-anti-patterns-let-s-do-gitops-not-ciops

What Happens With GitOps?



https://www.weave.works/blog/kubernetes-anti-patterns-let-s-do-gitops-not-ciops

What Problems does GitOps Solve?

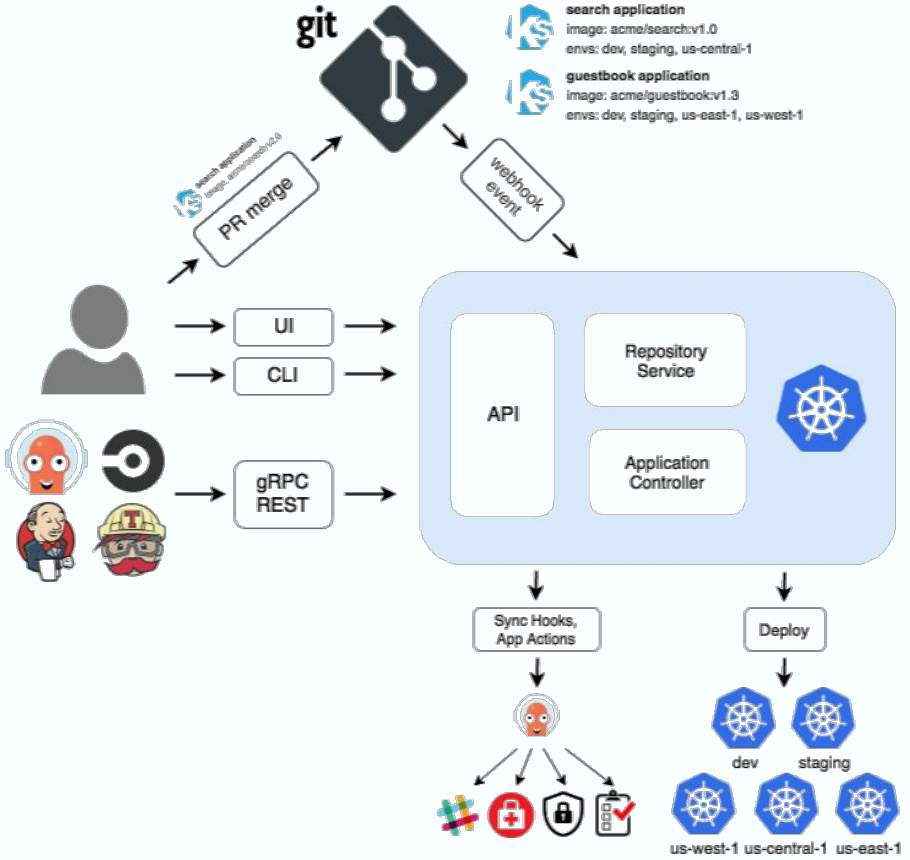
- **Control** - All interactions through Git.
- **Collaboration** - Proposed changes in PRs.
- **Compliance** - Clear audit log of state changes.
- **Rollbacks** - Forget backing up manifests before making changes.
- **Disaster Recovery** - Bootstrap cluster with GitOps agent.

Argo CD Core Concepts

- The Application resource
- Config Management Tools
- Application Sync
- Application Health
- App Projects

Argo CD Components

- API Server
- Repository Server
- Application Controller



Argo CD Application

Argo CD Application

- source
- destination
- syncPolicy

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: guestbook
  namespace: argocd
spec:
  project: default
  source:
    repoURL: 'https://github.com/<username>/intro-argo-cd-tutorial'
    path: guestbook
    targetRevision: HEAD
  destination:
    namespace: guestbook
    name: <environment> # Update this value.
  syncPolicy:
    syncOptions:
      - CreateNamespace=true
```

Config Management Tools

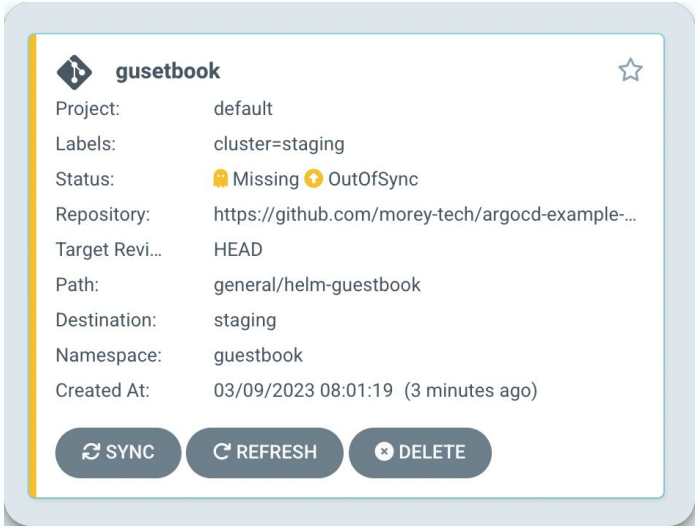
- Kustomize
- Helm
- Directory of YAML (or Jsonnet)
- Config Management Plugin (CMP)
- Tool Detection

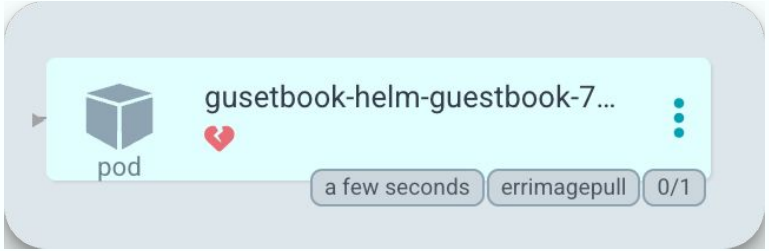
Application Sync

- Automated sync policy.
- Self-healing.
- Automatic Namespace creation.

Application Health

- Top-level health of Application.
- Based on tracked resources.
- Custom health checks.





Deploying Helm Charts with Argo CD

https://docs.akuity.io/tutorials/introduction-to-argo-cd/

Best Practices

- Separating Source Code from GitOps.
- Repository Structure
- Helm + Kustomize
- GitOps anti-patterns.

Separating Source Code from GitOps

- Clear separation.
- Simplify CI workflows.
- Separation of concern.

Repository Structure

- base/ - contains the manifests to deploy the application
  - deployment.yaml - defines the deployment
  - service.yaml - defines the service
- env/ - contains the environment specific overlays
  - dev/
  - stage/
  - prod/

Helm + Kustomize

- Helm is best for templating.
- Kustomize is best for last-mile patches.
- Requires setting `--enable-helm`.

kustomization.yaml

```yaml
helmCharts:
- name: minecraft
  version: 3.1.3
  repo: https://itzg.github.io/minecraft-server-charts
```

GitOps Anti-Patterns

- Rollbacks in Argo CD.
- Overriding parameters in Argo CD.
- Leaving room for imperativeness.

Questions?