GitOps and ArgoCD

Empowering DevOps Teams
for Kubernetes Success

DevOps Engineer @_VOIS

linkedin.com/in/omarmfathy219

twitter.com/omarmfathy219

github.com/omarmfathy219

omarmfathy219@gmail.com

Agenda

01 What is GitOps?

02 GitOps Operation Model

03 Why we need GitOps?

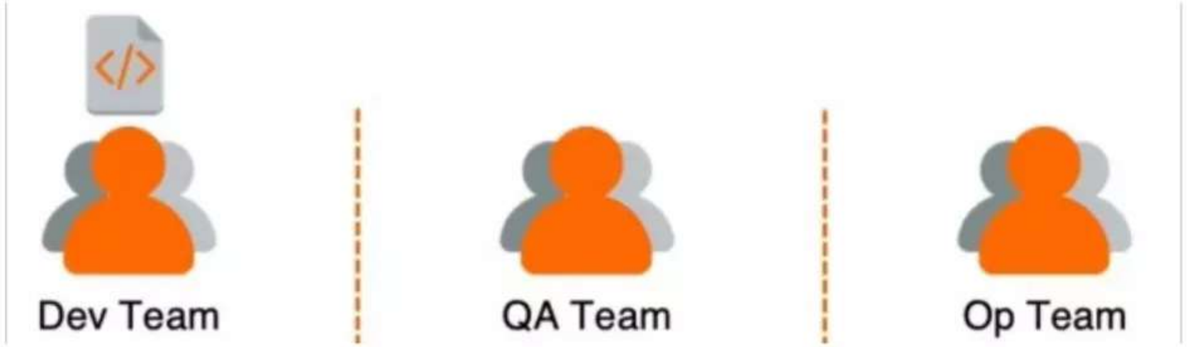
04 GitOps Principles?

05 What is ArgoCD?

06 Demo...

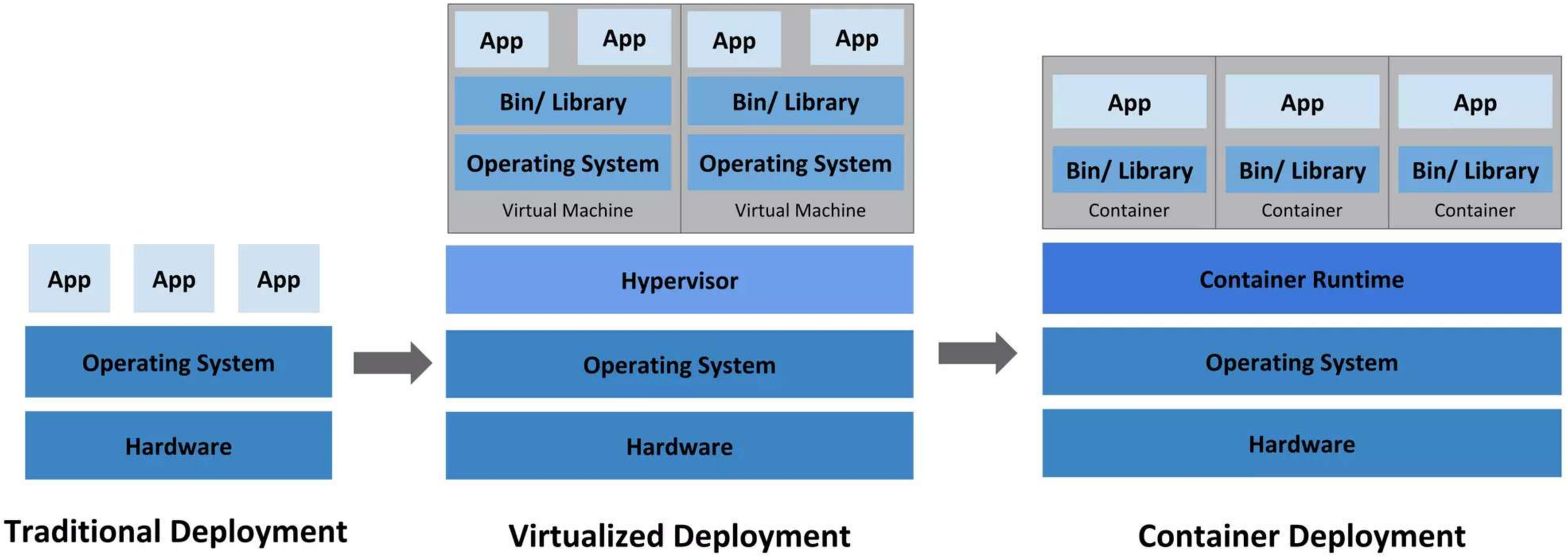
Traditional Ops Process

- Devs write code
- "throw it over the wall" to QA
- Code bounces back and forth between Dev and QA as QA discovers problems and Devs fix them
- Finally, it is ready for production



DevOps Process

- Devs write code
- Code commit triggers automated build, integration, and tests
- QA can get their hands on it almost immediately
- Once it is ready, kick off an automated deployment to production


App
App
App
Operating System
Hardware
Traditional Deployment
App
App
App
App
Bin/ Library
Bin/ Library
Operating System
Operating System
Virtual Machine
Virtual Machine
Hypervisor
Operating System
Hardware
Virtualized Deployment
App
App
App
Bin/ Library
Bin/ Library
Bin/ Library
Container
Container
Container
Container Runtime
Operating System
Hardware
Container Deployment

imperative Way

```
# create namespace
kubectl create namespace sandbox

# Create deployment with name simple-deployment
kubectl create deployment simple-deployment --image=docker.io/kostiscodefresh/gitops-simple-app:v1.0 --namespace=sandbox

# Scale deployment to 1 replica
kubectl scale deployment simple-deployment --replicas=1 --namespace=sandbox

# Expose deployment as a service
kubectl expose deployment simple-deployment --name=simple-service --type=NodePort --port=3100 --target-port=8080 --namespace=sandbox

# Port-forward service to localhost
kubectl port-forward service/simple-service 3100:3100 --namespace=sandbox
```

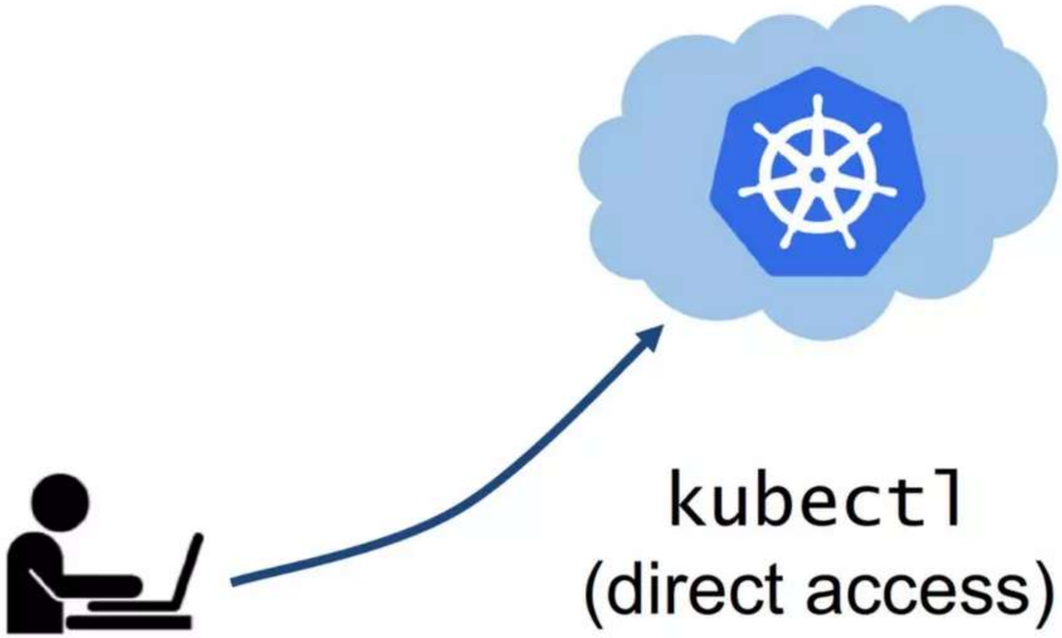
Declarative Way

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: simple-deployment
  namespace: sandbox
spec:
  replicas: 1
  selector:
    matchLabels:
      app: simple-app
  template:
    metadata:
      labels:
        app: simple-app
    spec:
      containers:
      - name: simple-app
        image: docker.io/kostiscodefresh/gitops-simple-app:v1.0
        ports:
        - containerPort: 8080
```

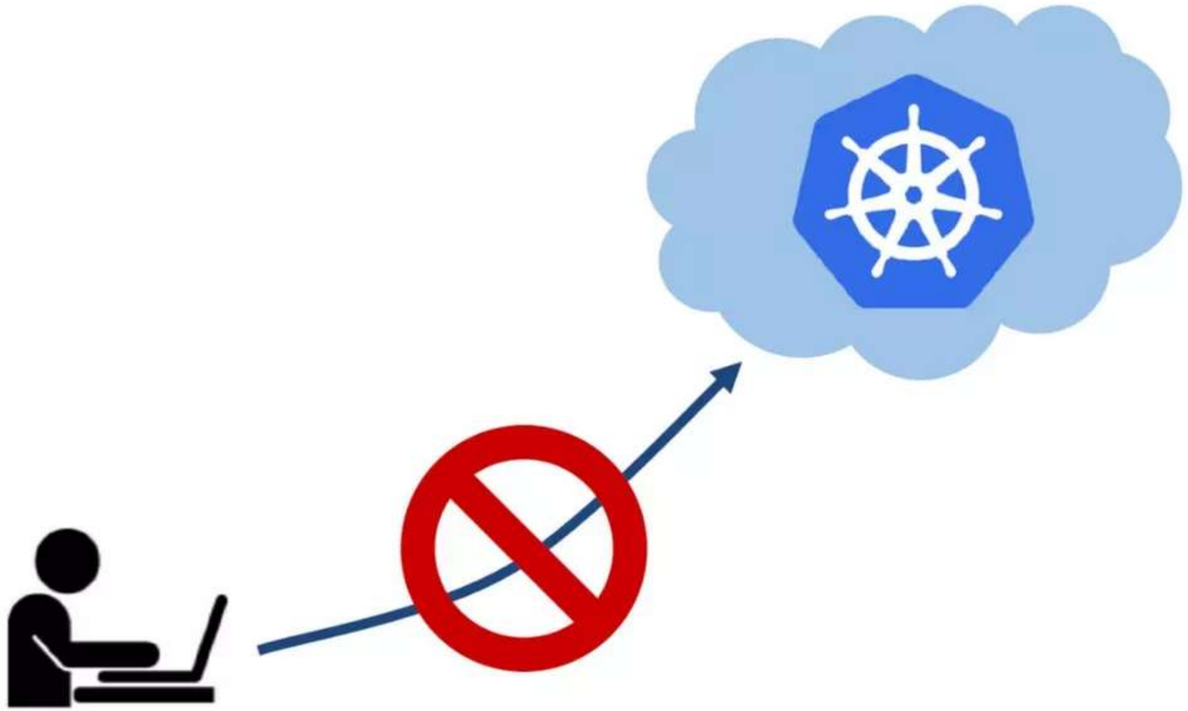
What is GitOps?

GitOps is a set of best practices where the entire code delivery process is controlled via **Git**, including infrastructure and application definition as code and automation to complete updates and rollbacks.

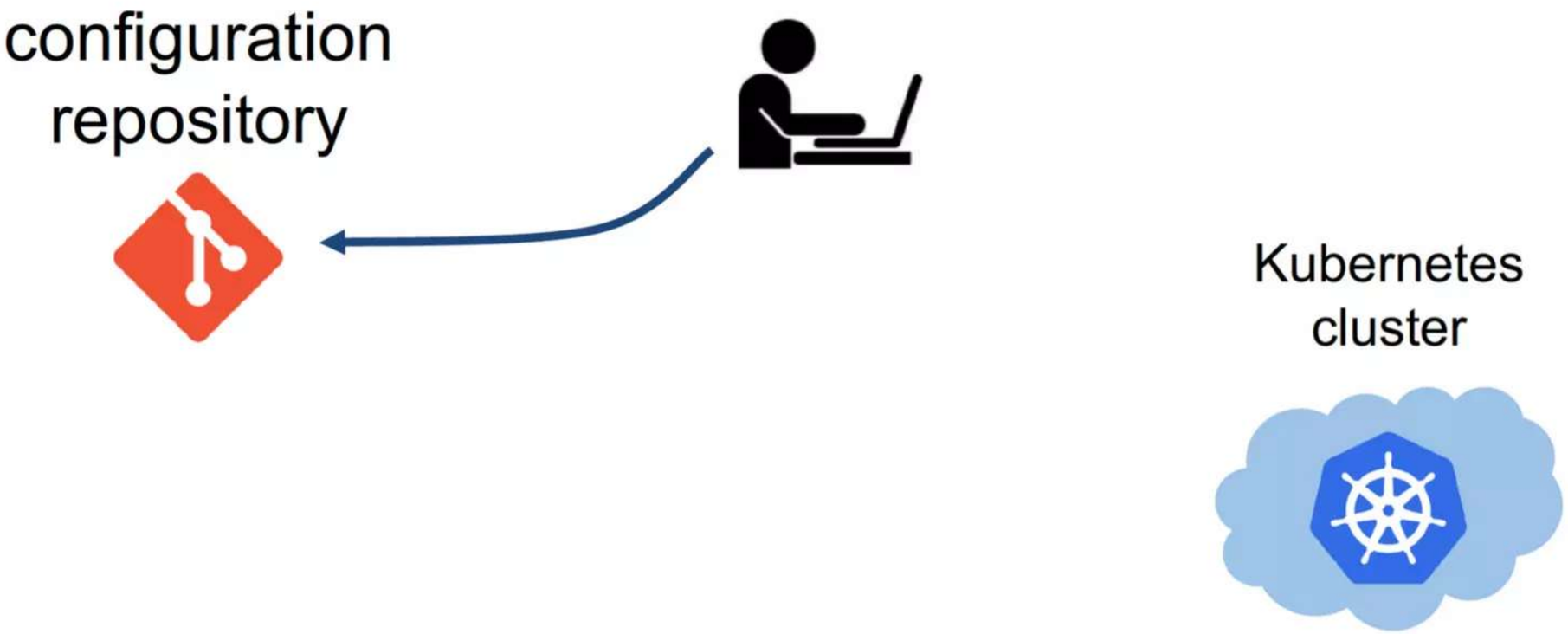
Traditional Operation Model



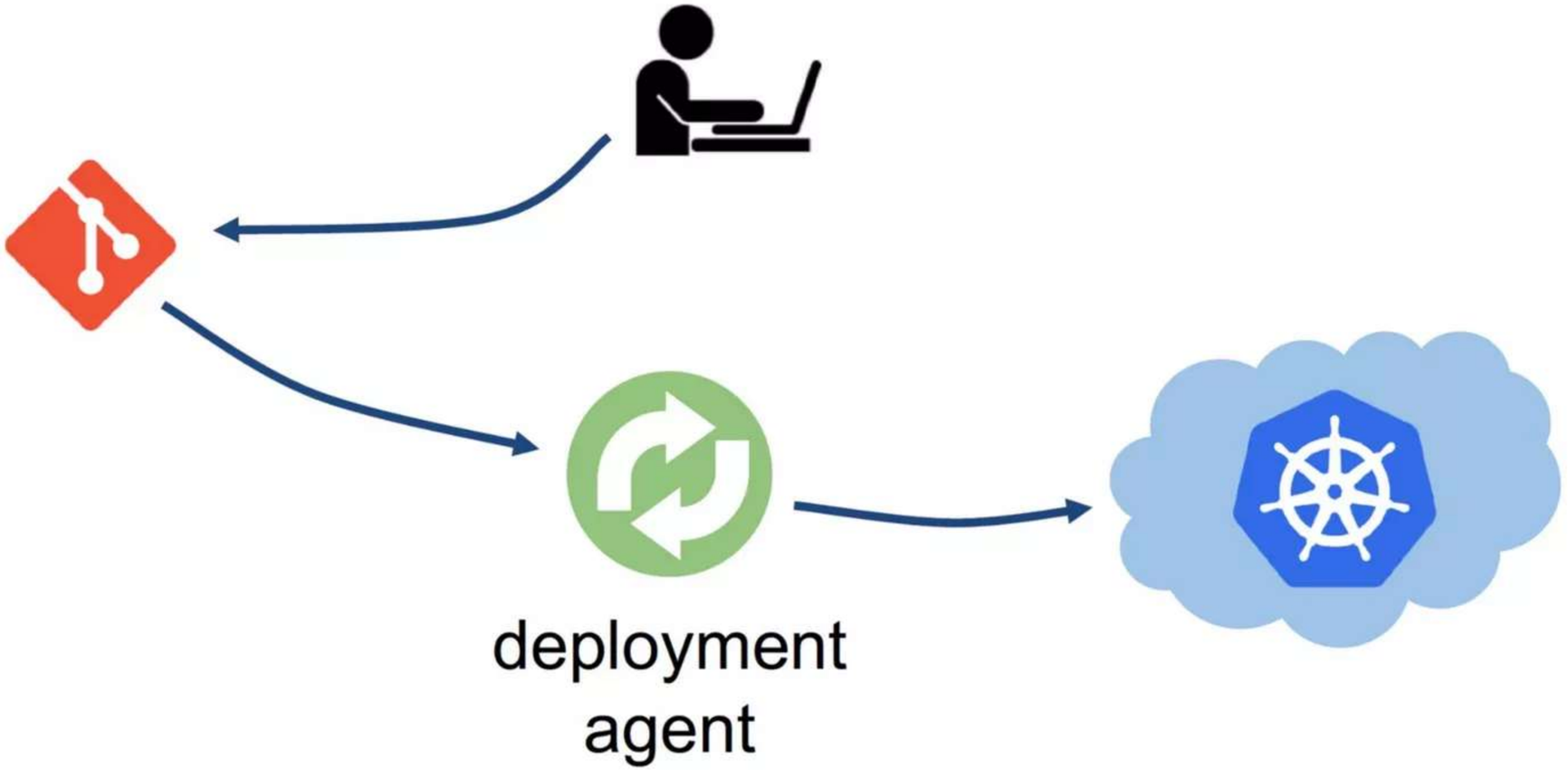
GitOps Operation Model



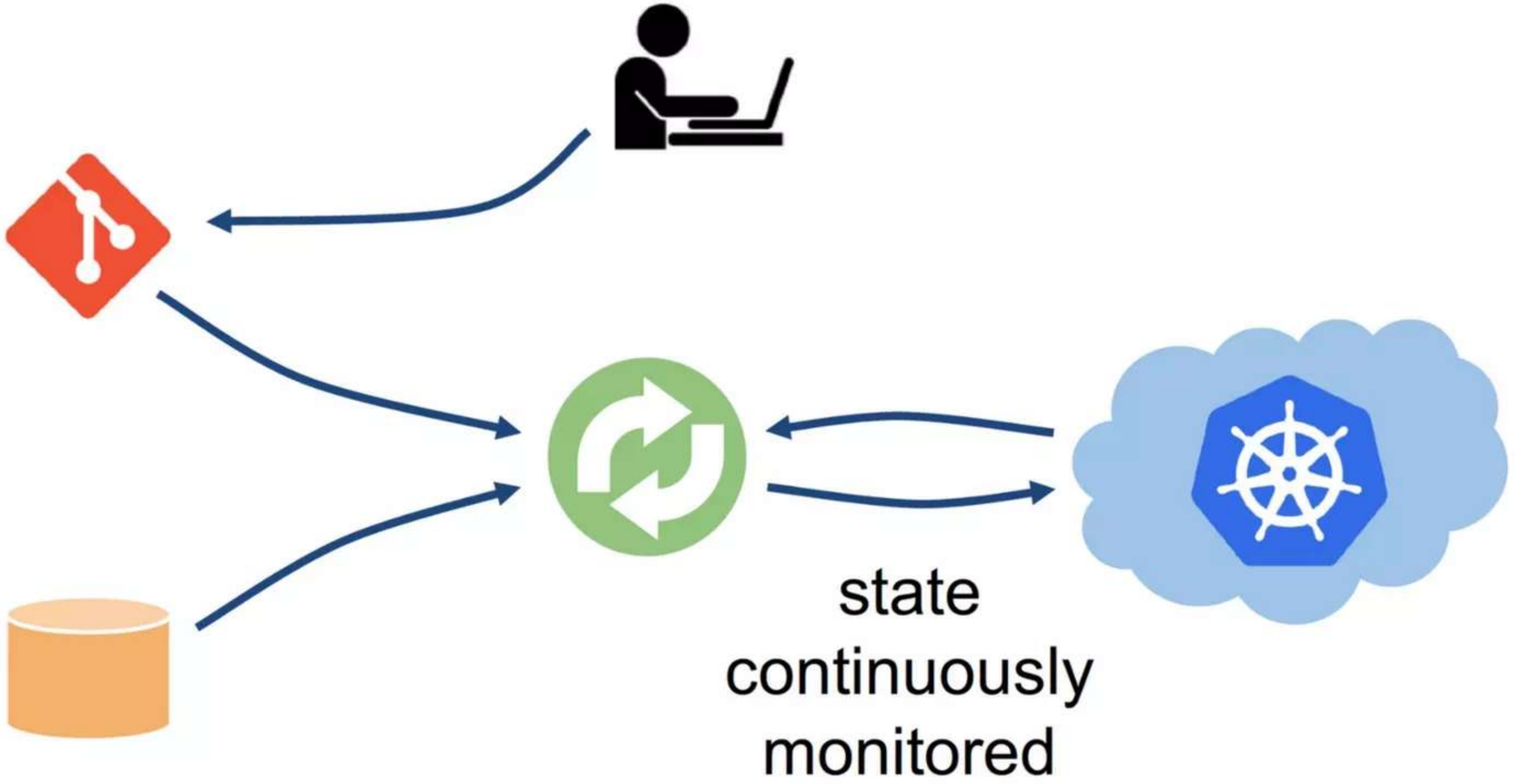
GitOps Operation Model



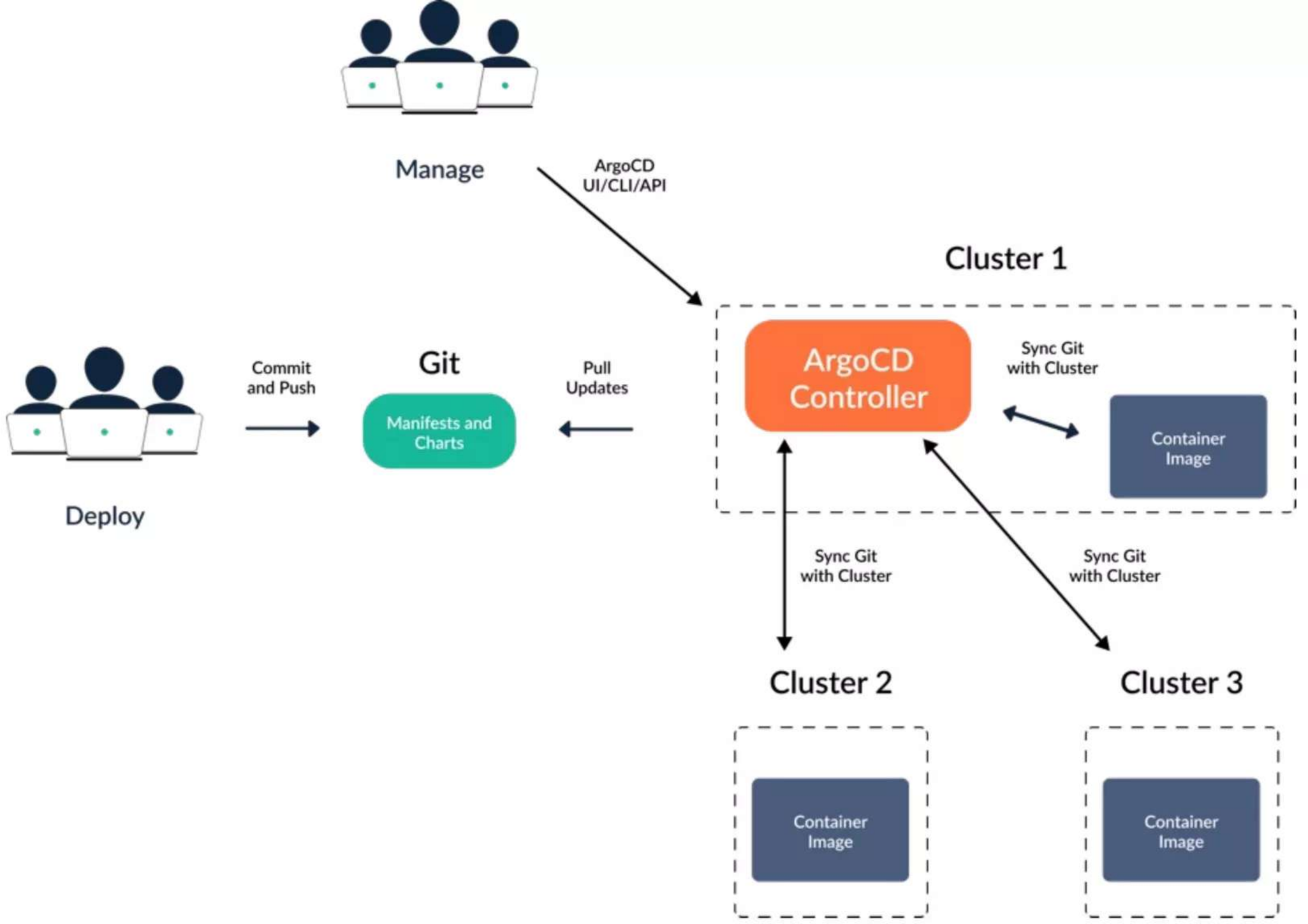
GitOps Operation Model



GitOps Operation Model



Traditional Model


Manage
ArgoCD
UI/CLI/API
Cluster 1
Deploy
Commit
and Push
Git
Manifests and
Charts
Pull
Updates
ArgoCD
Controller
Sync Git
with Cluster
Container
Image
Sync Git
with Cluster
Sync Git
with Cluster
Cluster 2
Cluster 3
Container
Image
Container
Image

Why we need GitOps?

- Faster deployments
- Safer deployments
- Easier rollbacks
- Straightforward auditing
- Better traceability
- Eliminating configuration drift

What is ArgoCD?

ArgoCD is a declarative, GitOps continuous delivery tool for Kubernetes.

ArgoCD follows the GitOps pattern of using Git repositories as the source of truth for defining the desired application state.


Pull
Deploy
Git Repo
(Source of Manifests)
ArgoCD
(Destination) Cluster

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

Log out

APP HEALTH
Healthy

CURRENT SYNC STATUS
Synced
MORE
To HEAD (77b98b6)
Author: Kostis (Codefresh) <39800303+kostis-codefresh@users.n...
Comment: Added cleanup instructions

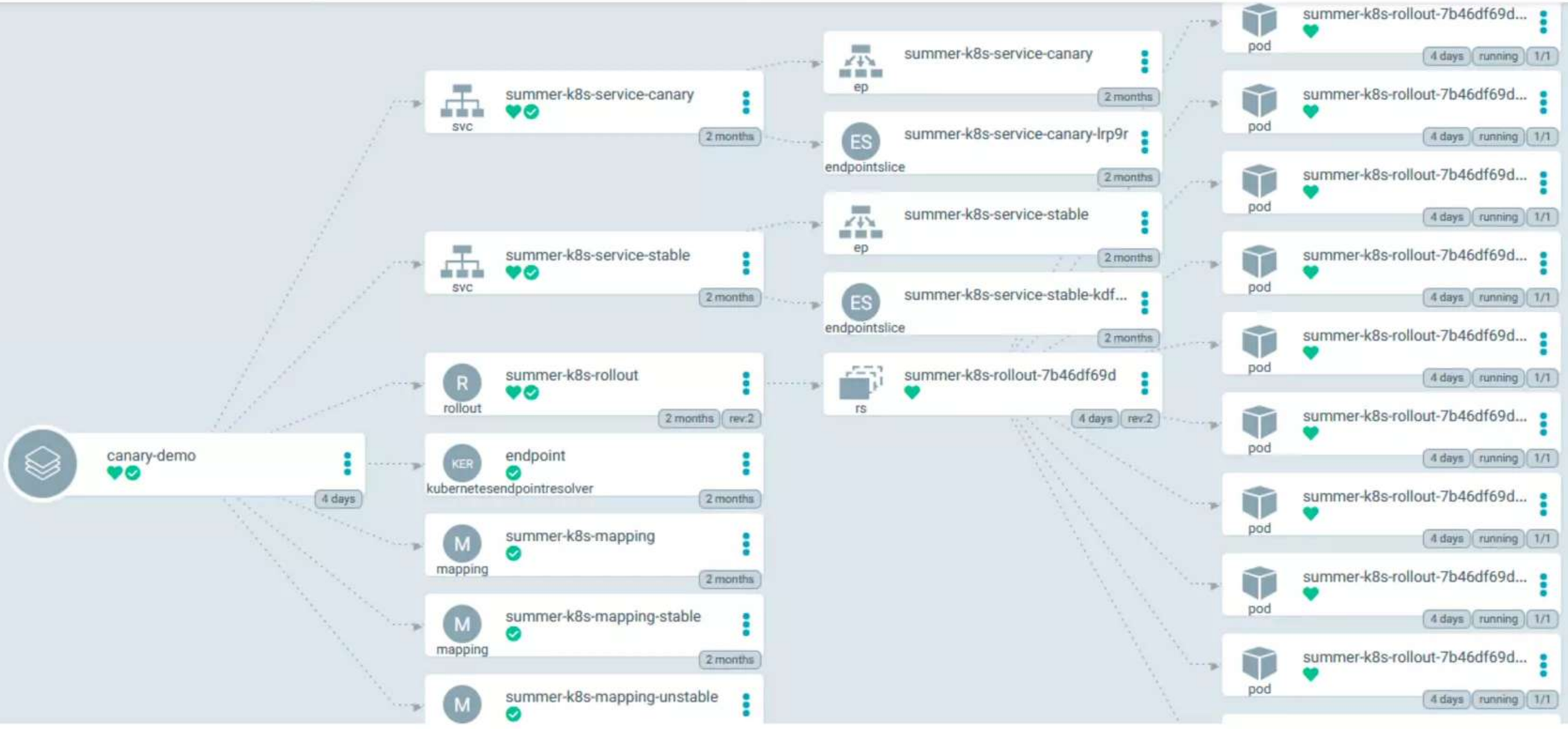
LAST SYNC RESULT
Sync OK
MORE
To 77b98b6
Succeeded 4 days ago (Thu Oct 21 2021 19:00:28 GMT+0300)
Author: Kostis (Codefresh) <39800303+kostis-codefresh@users.n...
Comment: Added cleanup instructions

FILTERS

KINDS
KINDS

SYNC STATUS
- Synced
- OutOfSync

HEALTH STATUS
- Healthy
- Progressing
- Degraded
- Suspended
- Missing
- Unknown



ArgoCD Health Status

- **Sync:** The process by which an argoCD application reaches the desired target state defined in the repository from its current state.
- **Healthy:** The application's live and desired state is in sync.
- **Out Of Sync:** The desired state and the live state are not the same.
- **Progressing:** The sync operation is ongoing and the application will be healthy soon
- **Degraded:** The app health is degrading

GitOps Principles?

- The entire system (infrastructure and applications) is described **declaratively**
- The desired system state is versioned in **Git**
- Changes approved are automated and applied to the system

Let's make our hands dirty...

Thank You!