Giant Swarm

# Declaratively Managing Apps in a Multi-Cluster World

Sept, 17th 2020

Fernando Ripoll

@pipo02mix

# Agenda

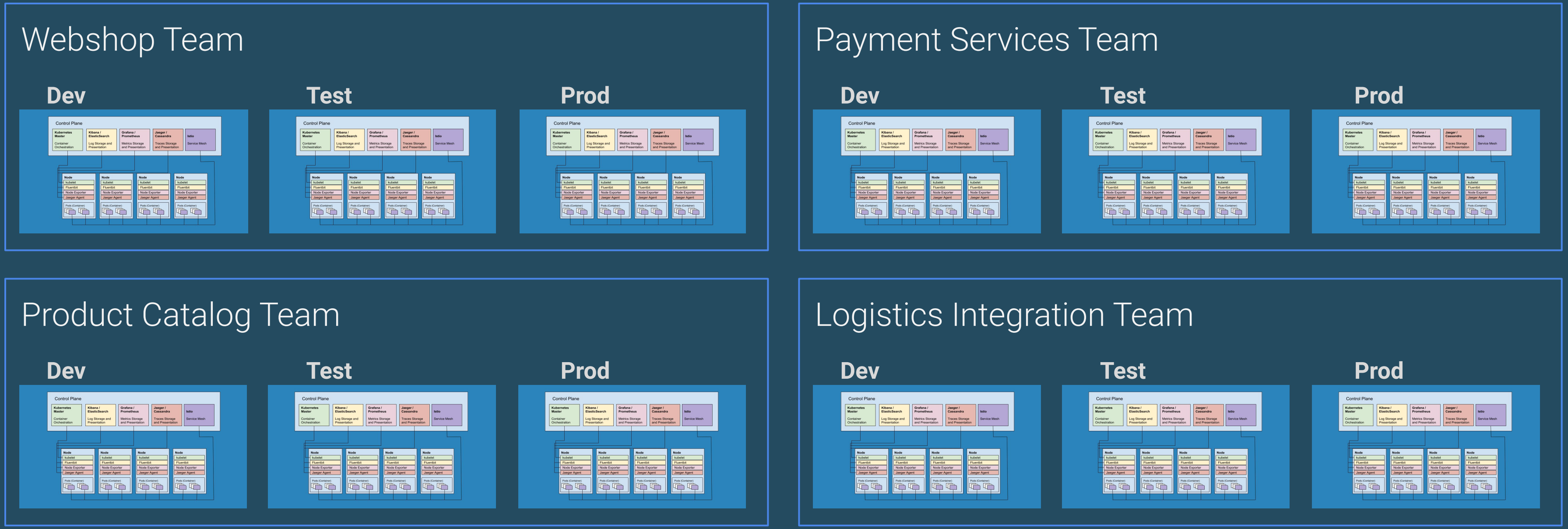
- Status Quo
- Imperative vs Declarative
- Cloud Native Application
- Application Platform
- App Operator
- Chart Operator
- Demo
- Future plans

# Status Quo

- Running a Kubernetes cluster is almost a commodity

# Status Quo

- Running a Kubernetes cluster is almost a commodity
- Companies usually manage multiple clusters



Webshop Team

Dev Test Prod

Payment Services Team

Dev Test Prod

Product Catalog Team

Dev Test Prod

Logistics Integration Team

Dev Test Prod

# Status Quo

- Running a Kubernetes cluster is almost a commodity
- Companies usually manage multiple clusters
- Mature generic abstractions

  (Deployments, Daemonsets, ...)

# Status Quo

- Running a Kubernetes cluster is almost a commodity
- Companies usually manage multiple clusters
- Mature generic abstractions

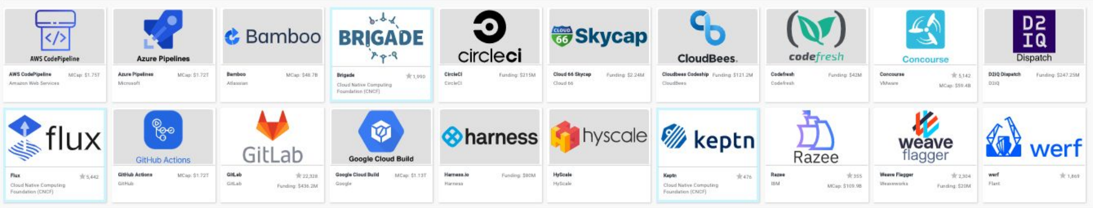
  (Deployments, Daemonsets, ...)
- Multiple options to package your app

ANSIBLE
HELM
Kustomize.io

# Status Quo

- Running a Kubernetes cluster is almost a commodity
- Companies usually manage multiple clusters
- Mature generic abstractions

  (Deployments, Daemonsets, ...)
- Multiple options to package your app
- Full-fledged CI/CD landscape



AWS CodePipeline MCap: $1.75T
Amazon Web Services

Azure Pipelines MCap: $1.72T
Microsoft

Bamboo MCap: $48.7B
Atlassian

Brigade ★ 1,990
Cloud Native Computing Foundation (CNCF)

CircleCI Funding: $215M
CircleCI

Cloud 66 Skycap Funding: $2.24M
Cloud 66

Cloudbees Codeship Funding: $121.2M
CloudBees

Codefresh Funding: $42M
Codefresh

Concourse ★ 5,142
VMware MCap: $59.4B

D2IQ Dispatch Funding: $247.25M
D2IQ

Flux ★ 5,442
Cloud Native Computing Foundation (CNCF)

GitHub Actions MCap: $1.72T
GitHub

GitLab ★ 22,328
GitLab Funding: $436.2M

Google Cloud Build MCap: $1.13T
Google

Harness.io Funding: $80M
Harness

HyScale
HyScale

Keptn ★ 476
Cloud Native Computing Foundation (CNCF)

Razee ★ 355
IBM MCap: $109.9B

Weave Flagger ★ 2,304
Weaveworks Funding: $20M

werf ★ 1,869
Flant

https://landscape.cncf.io/category=continuous-integration-delivery

# Imperative vs Declarative

Script the steps to deploy your application in a cluster

- No idempotency
- Have to maintain state of cluster in a secondary place
- Better maintenance of dependencies
- Steps are scripted in the workflow definition (limitations)

# Imperative vs **Declarative**

Defines the desired state of the system or application

- Benefit from idempotency
- The state is in the own system
- Hard to define the dependencies
- The logic of the deployment is embedded in the controllers

SQL

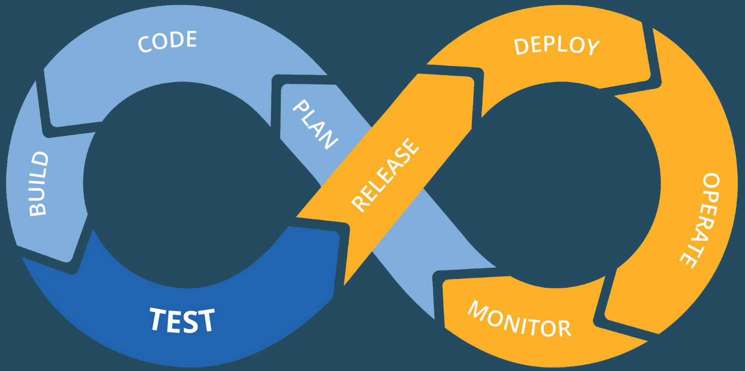
CloudFormation

Pulumi

# Cloud Native App Lifecycle

1. A developer codes an application in given language
2. They containerized the application
3. They create the configuration and package the application
4. They create a CI workflow for building and testing the application
5. They define a CD pipeline to describe how the application is deployed in different environments
6. They instrument and monitor the Application performance

N. New change in the application code retriggers the cycle



# Container layer

The Cloud Native applications are wrapped as containers

- Less overhead
- Lightweight artifacts
- Great portability
- Fair isolation
- Good consistency
- Awesome adoption (tooling)

# Application Configuration

Kubernetes provides different ways to define configuration and infrastructure of our applications

- Configmaps
- Secrets
- Environment variables

---

- Service
- Ingress
- Autoscaling
- Network Policies
- RBAC

# Application package

Cloud Native application package of configuration and infrastructure.

- Templating system
- Easy distribution
- Release management
- Configurability

# Application Platform

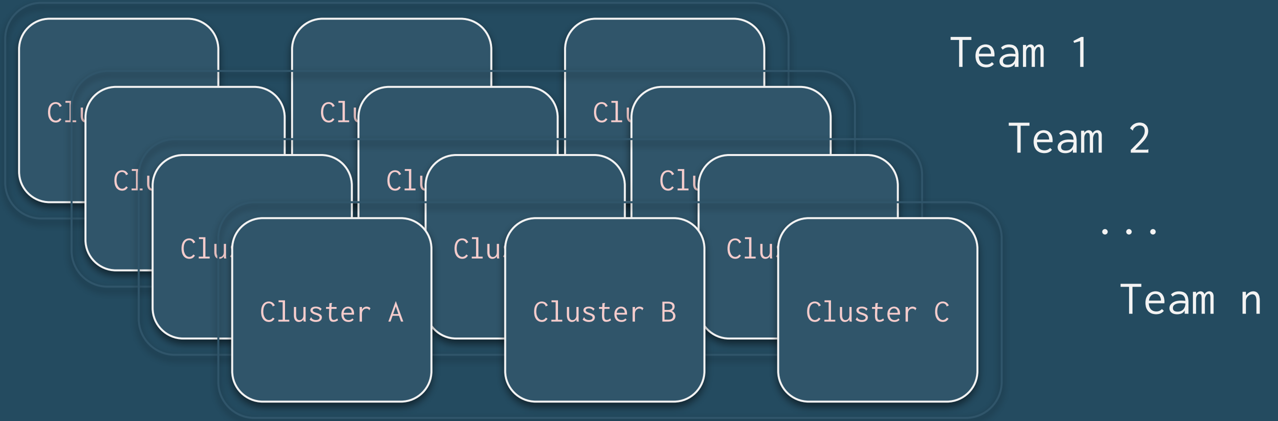
Following the evolution, we created a system that help us to solve some of necessities

Goals of the system:

- Manage App Deployments over several targets
- Different levels of configuration
- Declarative way to define your applications
- Add observability to the App Lifecycle

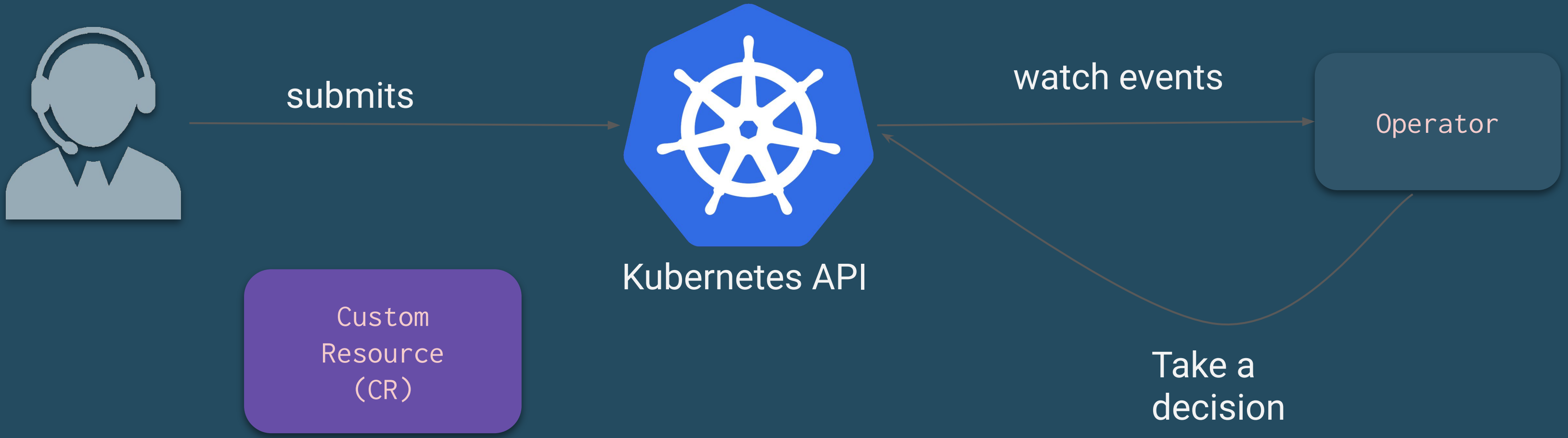
# Kubernetes principles

- Maintain state
- Observability
- Automation
- Versioning



# Extending Kubernetes

*Operator pattern*



# App Platform Concepts

- App Catalog CR

  - Place to store the App packages
  - Default Configuration
  - Metadata (title, description, logo, …)

# App Platform Concepts

- App Catalog CR
- App CR
  - Target cluster
  - Cluster scope Configuration
  - User scope Configuration
  - Metadata (name, namespace, version, …)

# App Platform Concepts

- App Catalog CR

- App CR

- App Operator

  - Validation and management of configuration
  - Manages the creation and update of chart CR in each cluster
  - Expose the status of the deployment

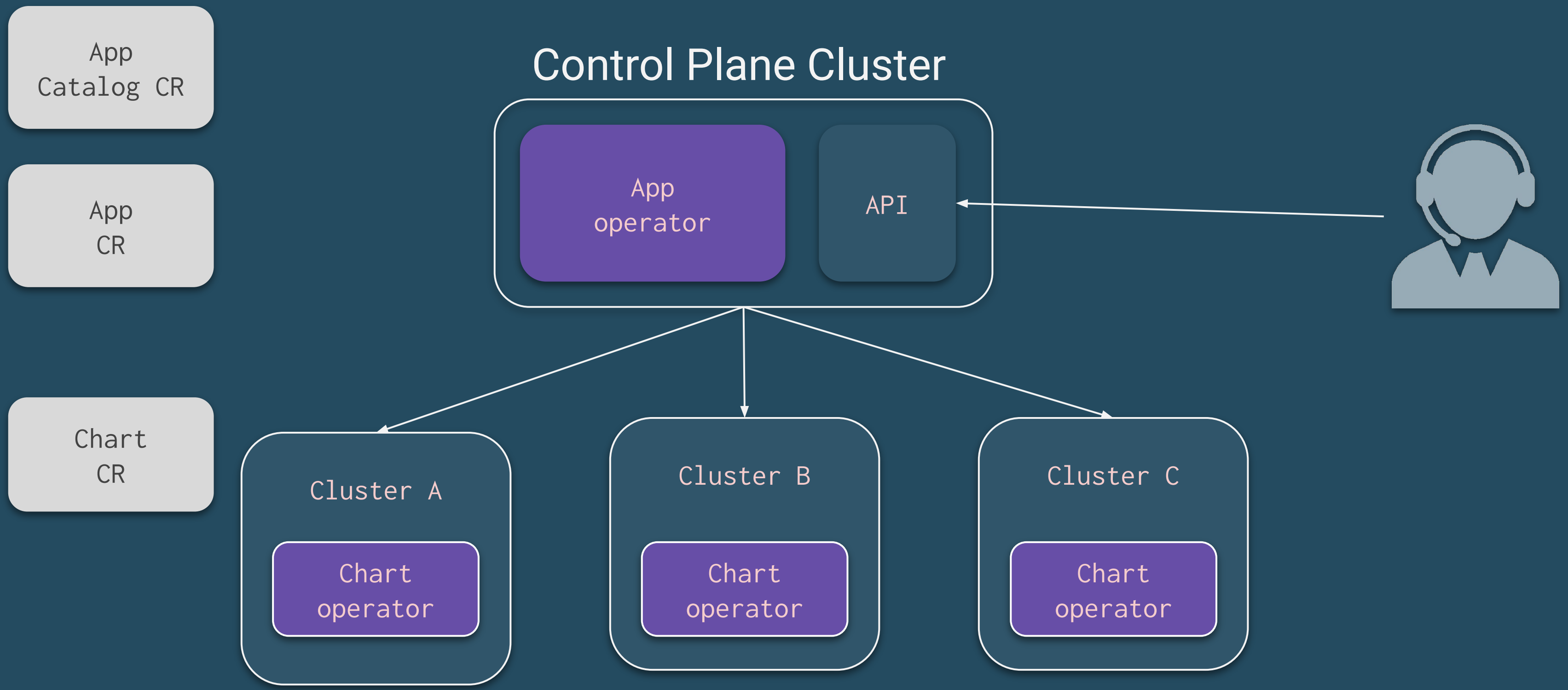
# App Platform Concepts

- App Catalog CR
- App CR
- App Operator
- Chart CR
  - Package URL
  - Three level Configuration merged
  - Metadata (name, namespace, version, …)

# App Platform Concepts

- App Catalog CR
- App CR
- App Operator
- Chart CR
- Chart Operator
    - Abstract the logic of the deployment logic
    - React to changes in configuration
    - Expose the status of the application deployment

# App Platform Architecture



# App Catalog Custom Resource

```
apiVersion: application.giantswarm.io/v1alpha1
kind: AppCatalog
metadata:
  name: my-playground-catalog
spec:
  config:
    configMap:
      name: my-playground-catalog
      namespace: default
    secret:
      name: my-playground-catalog
      namespace: default
  description: A catalog to store all new application packages.
  logoURL: https://my-org.github.com/logo.png
  title: My Playground Catalog
  storage:
    URL: https://my-org.github.com/my-playground-catalog/
    type: helm
```

Default Catalog Configuration Values

# App Catalog Custom Resource

```
apiVersion: application.giantswarm.io/v1alpha1
kind: AppCatalog
metadata:
  name: my-playground-catalog
spec:
  config:
    configMap:
      name: my-playground-catalog
      namespace: default
    secret:
      name: my-playground-catalog
      namespace: default
  description: A catalog to store all new application packages.
  logoURL: https://my-org.github.com/logo.png
  title: My Playground Catalog
  storage:
    URL: https://my-org.github.com/my-playground-catalog/
    type: helm
```

Catalog metadata

# App Catalog Custom Resource

```yaml
apiVersion: application.giantswarm.io/v1alpha1
kind: AppCatalog
metadata:
  name: my-playground-catalog
spec:
  config:
    configMap:
      name: my-playground-catalog
      namespace: default
    secret:
      name: my-playground-catalog
      namespace: default
  description: A catalog to store all new application packages.
  logoURL: https://my-org.github.com/logo.png
  title: My Playground Catalog
  storage:
    URL: https://my-org.github.com/my-playground-catalog/
    type: helm
```

Storage reference

# App Custom Resource

```
apiVersion: application.giantswarm.io/v1alpha1
kind: App
metadata:
  name: prometheus
  namespace: f2def
spec:
  kubeConfig:
    context:
      name: f2def
    inCluster: false
    secret:
      name: f2def-kubeconfig
      namespace: f2def
  catalog: my-playground-catalog
  name: prometheus
  namespace: monitoring
  config: ...
  userConfig: …
  version: 1.0.1
```

Target cluster

# App Custom Resource

```
apiVersion: application.giantswarm.io/v1alpha1
kind: App
metadata:
  name: prometheus
  namespace: f2def
spec:
  kubeConfig:...
  catalog: my-playground-catalog
  name: prometheus
  namespace: monitoring
  config: ...
  userConfig: …
  version: 1.0.1
```

App metadata

# App Custom Resource

```yaml
apiVersion: application.giantswarm.io/v1alpha1
kind: App
metadata:
  name: prometheus
  namespace: f2def
spec:
  kubeConfig: ...
  catalog: my-playground-catalog
  name: prometheus
  namespace: monitoring
  config:
    configMap:
      name: f2def-cluster-values
      namespace: f2def
    secret:
      name: f2def-cluster-values
      namespace: f2def
  userConfig: …
  version: 1.0.1
```

# App Custom Resource

```yaml
apiVersion: application.giantswarm.io/v1alpha1
kind: App
metadata:
  name: prometheus
  namespace: f2def
spec:
  kubeConfig: ...
  catalog: my-playground-catalog
  name: prometheus
  namespace: monitoring
  config: ...
  userConfig:
    configMap:
      name: f2def-user-values
      namespace: f2def
    secret:
      name: f2def-user-values
      namespace: f2def
  version: 1.0.1
```

Cluster scope configuration

# Chart Custom Resource

```
apiVersion: application.giantswarm.io/v1alpha1
kind: Chart
metadata:
  name: prometheus
  namespace: monitoring
spec:
  config:
    configMap:
      name: prometheus-chart-values
      namespace: monitoring
    secret:
      name: prometheus-chart-values
      namespace: monitoring
  tarballURL: https://github.io/my-playground-catalog/prometheus-1.0.1.tgz
  version: 1.0.1
status:
  appVersion: 1.0.1
  release:
    lastDeployed: 2020-09-01T13:25:53Z
    status: deployed
```

Three levels merged configuration

# Chart Custom Resource

```yaml
apiVersion: application.giantswarm.io/v1alpha1
kind: Chart
metadata:
  name: prometheus
  namespace: monitoring
spec:
  config:
    configMap:
      name: prometheus-chart-values
      namespace: monitoring
    secret:
      name: prometheus-chart-values
      namespace: monitoring
  tarballURL: https://github.io/my-playground-catalog/prometheus-1.0.1.tgz
  version: 1.0.1
status:
  appVersion: 1.0.1
  release:
    lastDeployed: 2020-09-01T13:25:53Z
    status: deployed
```

Location of the application package

# Chart Custom Resource

```yaml
apiVersion: application.giantswarm.io/v1alpha1
kind: Chart
metadata:
  name: prometheus
  namespace: monitoring
spec:
  config:
    configMap:
      name: prometheus-chart-values
      namespace: monitoring
    secret:
      name: prometheus-chart-values
      namespace: monitoring
  tarballURL: https://github.io/my-playground-catalog/prometheus-1.0.1.tgz
  version: 1.0.1
status:
  appVersion: 1.0.1
  release:
    lastDeployed: 2020-09-01T13:25:53Z
    status: deployed
```

State of the application

# DEMO TIME !

# Future plans

- Improved User Experience
  - App Catalog Entry CR
  - Better validation and defaulting
  - Kubectl plugin
- Automatic Updates
- App Stacks (templates)
- Application CRD

https://github.com/giantswarm/app-operator
https://github.com/giantswarm/chart-operator
https://docs.giantswarm.io/basics/app-platform

@pipo02mix
fernando@giantswarm.io