



KubeCon



CloudNativeCon

Europe 2019

Beyond Operators: Distributed Applications on Kubernetes Reimagined

Vladimir Vivien
Software engineer, VMware
K8s Upstream contributor
@VladimirVivien



Kelsey Hightower 

@kelseyhightower



Kubernetes is a platform for building platforms. It's a better place to start; not the endgame.

10:04 PM · Nov 27, 2017 · [Twitter Web Client](#)

Kubernetes is the platform for building distributed systems!

- Resource management
- Solid scheduler
- Automatic process distribution/scaling
- Built-in fault tolerance
- Available Stream of cluster events
- Distributed datastore for state and synchronization
- Infrastructure primitives
 - Storage, networking, access control, etc

Controllers



KubeCon



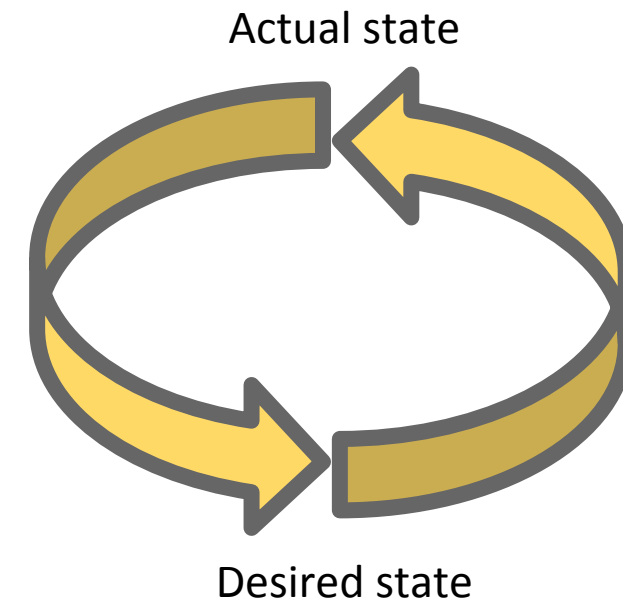
CloudNativeCon

Europe 2019

- State-of-the-art for building distributed apps in K8s
- Interact with the Kubernetes API server
- Control loop to synchronize cluster states
- Proven abstraction used extensively in Kubernetes

But...

- Designed to extend Kubernetes itself
- Exposes large API surfaces (many structural concerns)
- Require knowledge of Kubernetes internals



Operators



KubeCon



CloudNativeCon

Europe 2019

- Hides raw complexities of controllers
- Used to build native K8s applications
- Automates runtime application management
- Powerful abstraction over K8s API



But...

- Still leaves large API surface area to reason about as a developer
- Requires knowledge of K8s API and operator API

What I Want...



KubeCon



CloudNativeCon

Europe 2019

As a developer, who wants to create native distributed systems on top of Kubernetes, I want:

- A smaller API surface easier to reason about
- Complete abstraction of inner workings of Kubernetes
- No assumption about K8s API knowledge

Something as opinionated as `kubectl run` but in a programmatic API form.

My experiment...



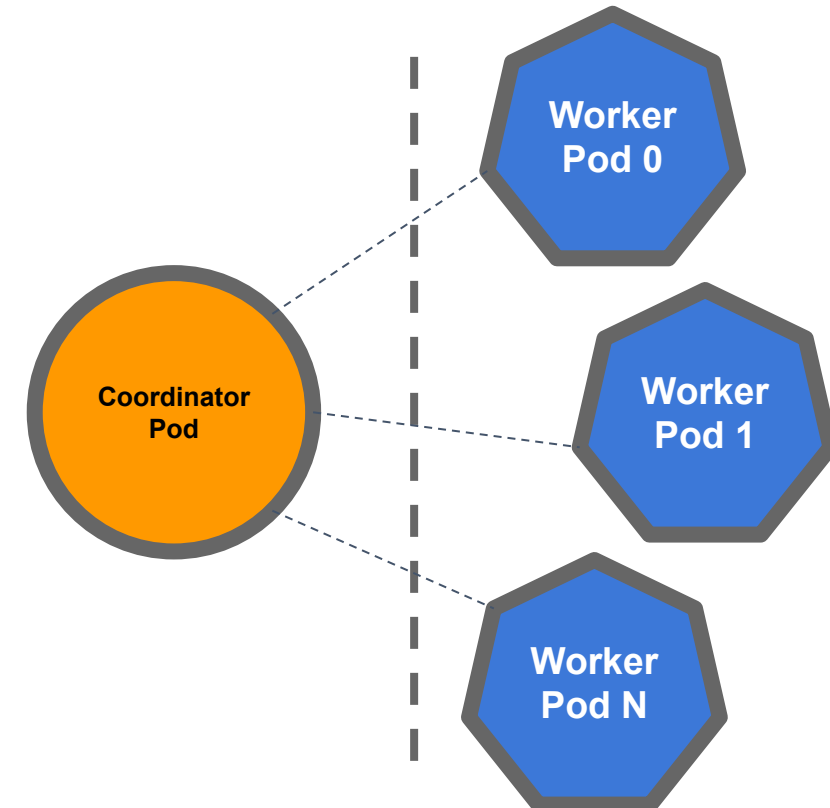
KubeCon



CloudNativeCon

Europe 2019

- A simpler API with event-driven interface
- Leverages existing K8s API objects and resources
- Uses a split design
 - Coordinator - Launches/coordinates/monitors worker processes
 - Worker - implements application logic for distributed app
- Inspired by actor-like distributed systems



The Coordinator



KubeCon



CloudNativeCon

Europe 2019

The Coordinator



KubeCon



CloudNativeCon

Europe 2019

Create coordinator

```
// setup the coordinator
coord, err := coordinator.New("greeter-supervisor", ns, config)
if err != nil {
    log.Fatalf("failed to start greeter-supervisor: %s", err)
}
```

1

The Coordinator



KubeCon



CloudNativeCon

Europe 2019

Define callback function(s)

```
coord.OnPodEvent(func(e api.PodEvent) {
    log.Println("Rcvd pod event")
    if e.Running {
        addr := e.PodIP
        res, err := http.Get(fmt.Sprintf("http://%s:%d/", addr, port))
        if err != nil {
            log.Println("unable to connect to worker process:", err)
            return
        }
        msg, err := ioutil.ReadAll(res.Body)
        defer res.Body.Close()
        if err != nil {
            log.Println("failed to read message from worker:", err)
            return
        }
        log.Println(msg)
    }
})
```

2

The Coordinator



KubeCon



CloudNativeCon

Europe 2019

Apply cluster operation

```
// apply an operation
if err := coord.Run(api.RunParam{
    Replicas:      1,
    Name:          "worker",
    Namespace:     ns,
    Image:         image,
    Port:          8086,
    ImagePullPolicy: "Never",
}); err != nil {
    log.Fatal(err)
}
```

3

The Coordinator



KubeCon



CloudNativeCon

Europe 2019

Run the coordinator

```
// start coordinator
if err := coord.Start(stopCh); err != nil {
    log.Fatal(err)
}
```

4

The Worker



KubeCon



CloudNativeCon

Europe 2019

1

```
worker, err := worker.New("greeter-worker", ns, config)
if err != nil {
    log.Fatalf("failed to start worker: %s", err)
}
```

2

```
worker.OnWorkerEvent(func(e api.WorkerEvent) {
    log.Println("Worker started!")
    go func() {
        http.HandleFunc("/", func(w http.ResponseWriter, req *http.Request) {
            io.WriteString(w, "Hello, world!\n")
        })
        log.Fatal(http.ListenAndServe(":8086", nil))
    }()
})
```

3

```
if err := worker.Start(stopCh); err != nil {
    log.Fatal(err)
}
```

Deployment



KubeCon



CloudNativeCon

Europe 2019

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: supervisor
  namespace: default
---
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: coordinator
rules:
- apiGroups: ["", "extensions", "apps"]
  resources: ["*"]
  verbs: ["get", "watch", "list"]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: coord-role-binding
subjects:
- kind: ServiceAccount
  name: supervisor
  namespace: default
roleRef:
  kind: ClusterRole
  name: coordinator
  apiGroup: rbac.authorization.k8s.io/v1
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    run: super
  name: super
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      run: super
  template:
    metadata:
      labels:
        run: super
    spec:
      containers:
      - image: supervisor:latest
        args:
        - "--worker-image=worker:latest"
        - "--namespace=default"
        imagePullPolicy: Always
        name: super
      serviceAccount: "supervisor"
```

```
> kubectl apply -f K8s-deploy.yaml
serviceaccount/supervisor unchanged
clusterrole.rbac.authorization.k8s.io/coordinator unchanged
clusterrolebinding.rbac.authorization.k8s.io/coord-role-binding unchanged
deployment.apps/super created
```

Running the Application



KubeCon



CloudNativeCon

Europe 2019

```
> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
super-9d489b9b7-fhwp4	1/1	Running	0	7s
worker-c66d6b6f4-ghr9h	1/1	Running	0	

Coordinator pod

Worker pod

```
2019/05/16 23:44:42 Deployment "super" received
2019/05/16 23:44:43 Deployment "worker" received
2019/05/16 23:44:43 Rcvd pod event
2019/05/16 23:44:43 Deployment "worker" ready!
2019/05/16 23:44:43 Rcvd pod event
2019/05/16 23:44:43 Deployment "worker" ready!
2019/05/16 23:44:43 Rcvd pod event
```

More to come



KubeCon



CloudNativeCon

Europe 2019

Currently PoC on my laptop

Open source project by KubeCon NA

Expose more functionalities and primitives

- Services
- Storage
- Scheduling
- Leader election
- Etc

Thank you

Vladimir Vivien

@VladimirVivien