The Operator Pattern

Managing Stateful Services in Kubernetes

Jakob Karalus, @krallistic

codecentric

codecentric AG
10x in Germany

HAMBURG

BERLIN

MÜNSTER

DORTMUND

SOLINGEN

AMSTERDAM

BREDA

FRANKFURT AM MAIN

NÜRNBERG

KARLSRUHE

STUTTGART

MÜNCHEN

Guuuude!
Guuuude!

Greetings from
Solingen

You never go Solo
in Karlsruhe

Normal Kubernetes Deployment

- Write some Deployment, Services, Configmaps etc
- Deploy them to K8s
- Maybe create a Helm Chart

3

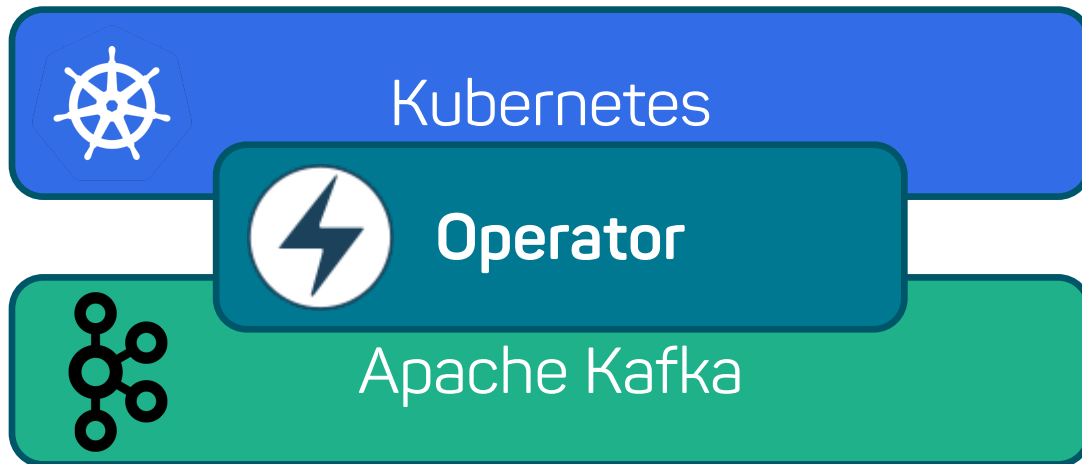# Success?!

But Day 2 Operations?

- Backups?
- Upscaling? Reshuffle Data?
- Downscaling? Without Dataloss?
- Healing? Restore Backups?
- Configuration? Tedious Templating?

If only we could automate this!

In a Kubernetes native way!

Operators

Kubernetes

Operator

Apache Kafka

Operators

- Human Operational Software
  - Custom Software
- Kubernetes Native:
  - CustomResourceDefinition
- So lets write one:

CustomResourceDefinition

- Defines a new API
- Seamless integration with existing API
- Kubectl support

```yaml
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
name: crontabs.stable.example.com
spec:
group: stable.example.com
version: v1
scope: Namespaced
names:
plural: crontabs
singular: crontab
kind: CronTab
shortNames:
- ct
validation:
# openAPIV3Schema is the schema for validating
custom objects.
```
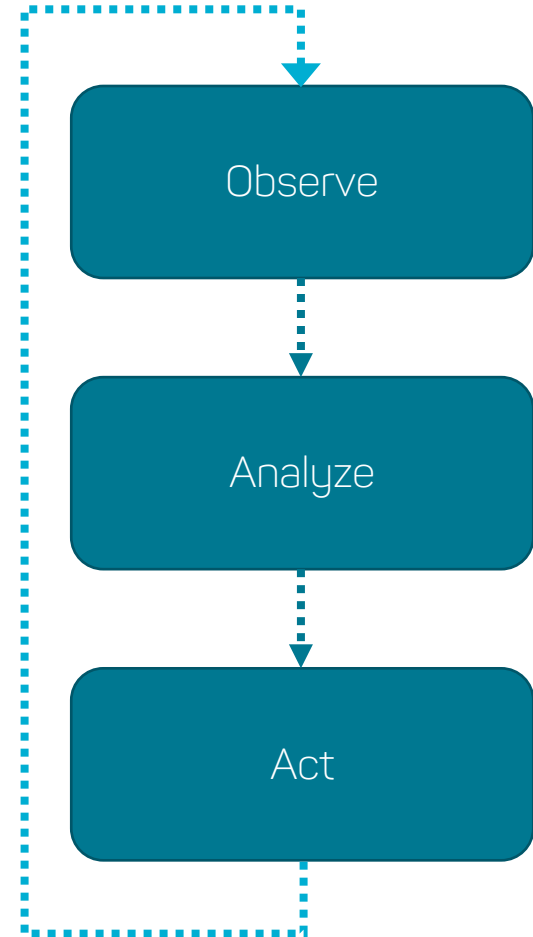
CustomResourceDefinition
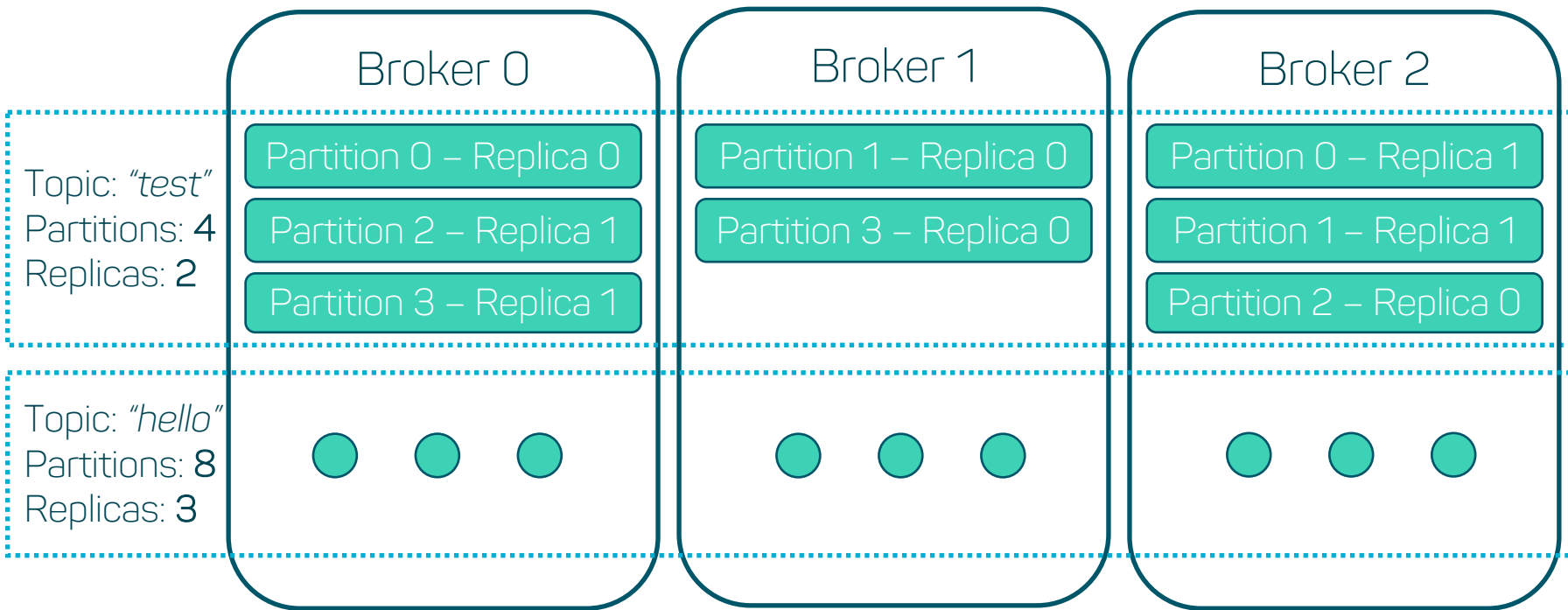
- Actual Object in new API
- No functionality.

```yaml
apiVersion: "stable.example.com/v1"
kind: CronTab
metadata:
name: my-new-cron-object
spec:
cronSpec: "* * * * */5"
image: my-awesome-cron-image
replicas: 5
```
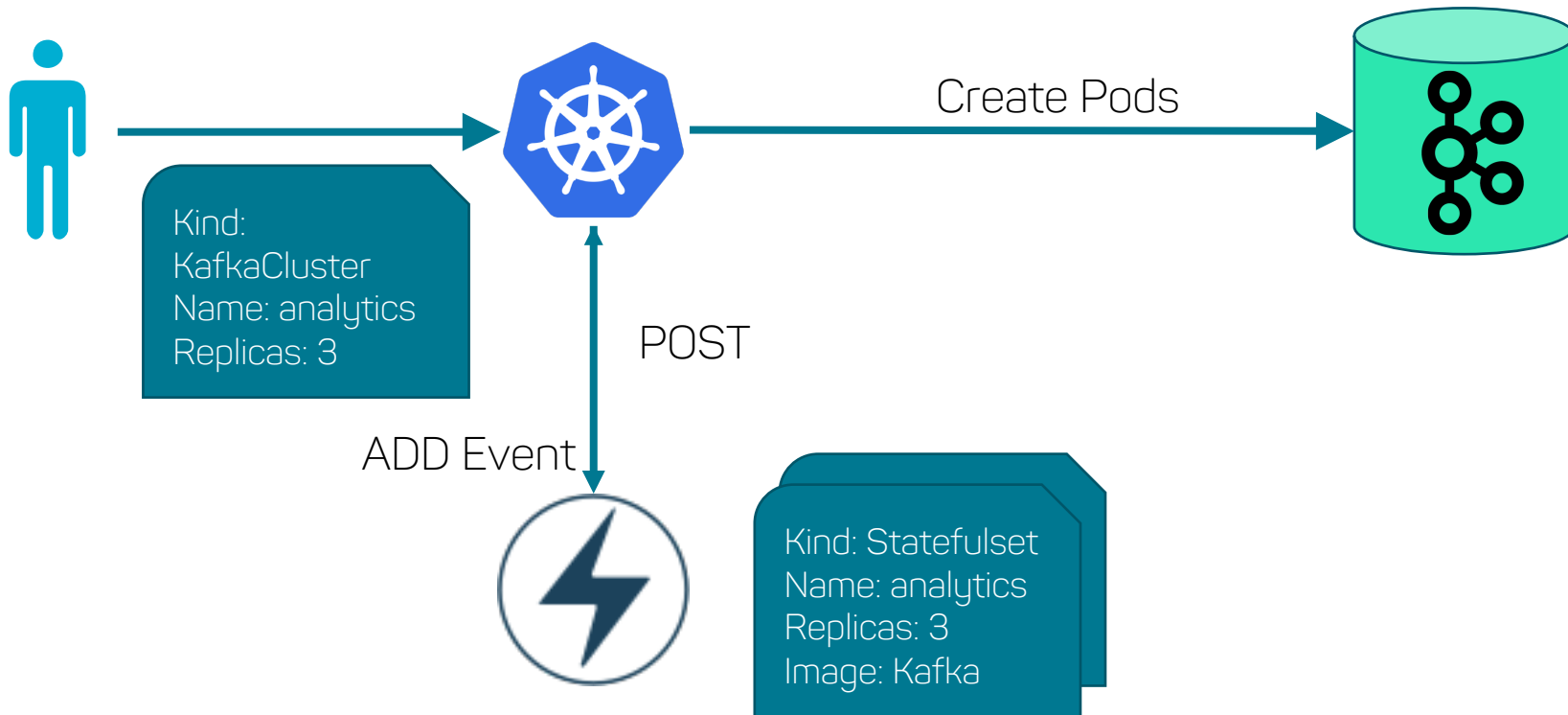
Control Loop

- Operator create WATCH on CR Objects

- Analyze difference Actual vs Desired State

- Act on changes

```
Observe
   ↓
Analyze
   ↓
Act
```

Kafka Basics

Broker 0

Broker 1

Broker 2

Topic: *"test"*
Partitions: 4
Replicas: 2

Partition 0 – Replica 0	Partition 1 – Replica 0	Partition 0 – Replica 1
Partition 2 – Replica 1	Partition 3 – Replica 0	Partition 1 – Replica 1
Partition 3 – Replica 1		Partition 2 – Replica 0

Topic: *"hello"*
Partitions: 8
Replicas: 3

Create Cluster

Kind:
KafkaCluster
Name: analytics
Replicas: 3

Create Pods

POST

ADD Event

Kind: Statefulset
Name: analytics
Replicas: 3
Image: Kafka

13

Downsize Cluster

Kind:
KafkaCluster
Name: analytics
Replicas: 2

Delete Pod

UPDATE

Rebalance Topic

Update Event

Cluster Rebalanced

Desired:
Kind: Statefulset
Name: analytics
Replicas: 2
Image: Kafka

Current:
Kind: Statefulset
Name: analytics
Replicas: 3
Image: Kafka

Rebalance Topics with Hot Partitions

Not enough?

- Extra Scheduling needs? -> Scheduler Extenders (Webhooks)

- More Advanced Scheduling Features -> Custom Schedulers

- Complex APIs beyond CRD -> Custom API Server with API Aggregation

- Control over what object can be created -> Admission Webhooks

- Control how objects should be initialized? -> Initializer

- Different Runtime needs? -> CRI, CNI
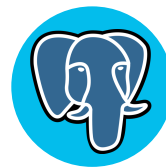
- Special Storage -> Flex Volumes, (CSI)

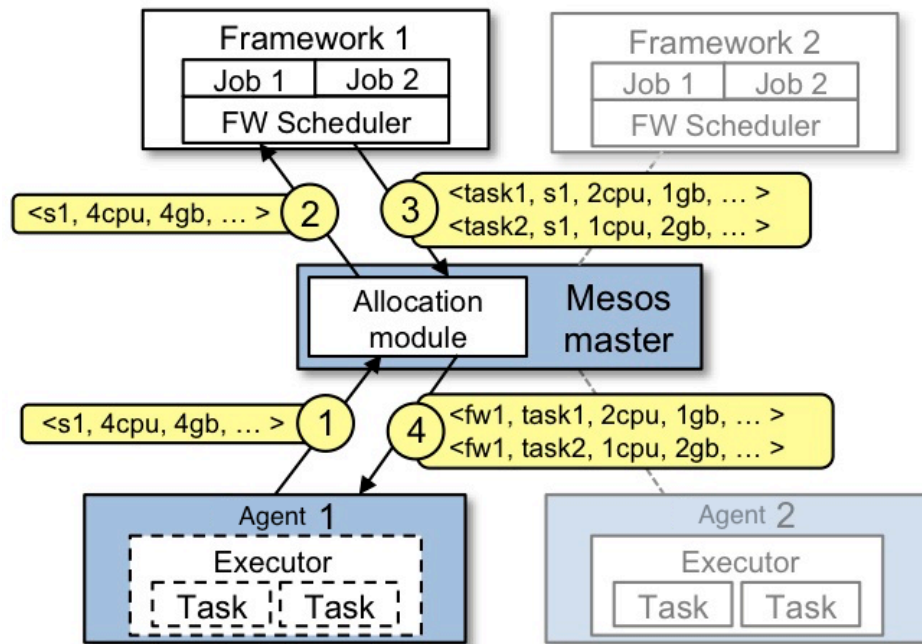# Other Operators

Elasticsearch

Prometheus

Postgres

Take a step back

- Are we reinventing the Wheel?
- Mesos Frameworks?
- Nomad Custom Scheduler?
- Docker Swarm Plugins?

Custom Logic in Mesos

Comparison

Task	Mesos	Kuberentes
Custom Resource Placement	Write a framework	Write a custom scheduler
Special resource init	Write a framework	Initializer
API access	Every Framework has its own API	Unified API
Special lifecyle	Write a framework	Kubernetes Operator
Custom execution	Write a framework + executioner	CRI Interface + Scheduler

Questions? Discuss!

codecentric