vmworld 2020

#CODE4104

Let’s Build a Kubernetes Operator in Go!

Rafael Brito, Application Platform Architect
Michael Gasch, Staff Engineer

vmware

Disclaimer

This presentation may contain product features or functionality that are currently under development.

This overview of new technology represents no commitment from VMware to deliver these features in any generally available product.

Features are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind.

Technical feasibility and market demand will affect final delivery.

Pricing and packaging for any new features/functionality/technology discussed or presented, have not been determined.

The information in this presentation is for informational purposes only and may not be incorporated into any contract. There is no commitment or obligation to deliver any items presented herein.

Our Goal for this Talk

Extending Kubernetes with Custom Resources

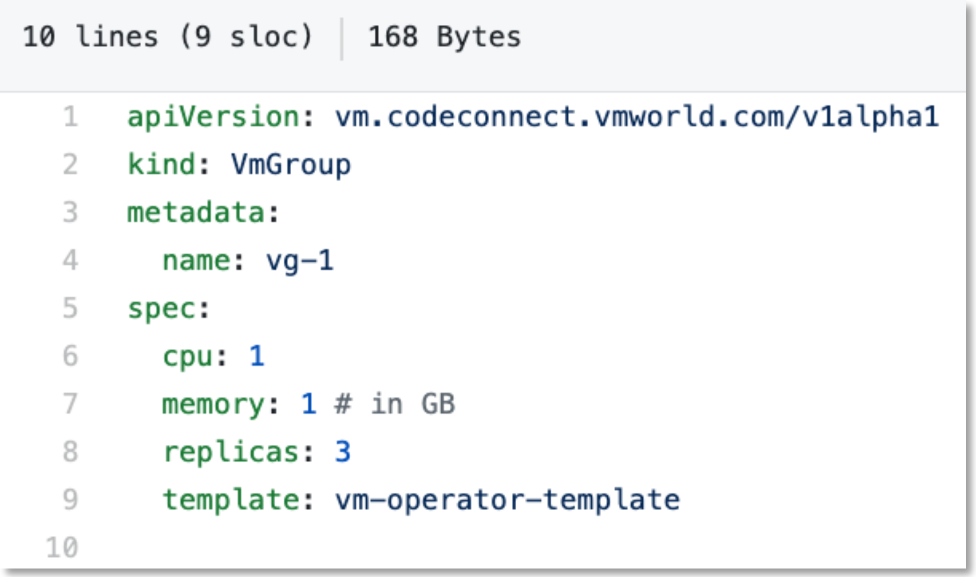
A little bit of (Control) Theory

Demo: Building a Simple VM Operator with kubebuilder

Resources

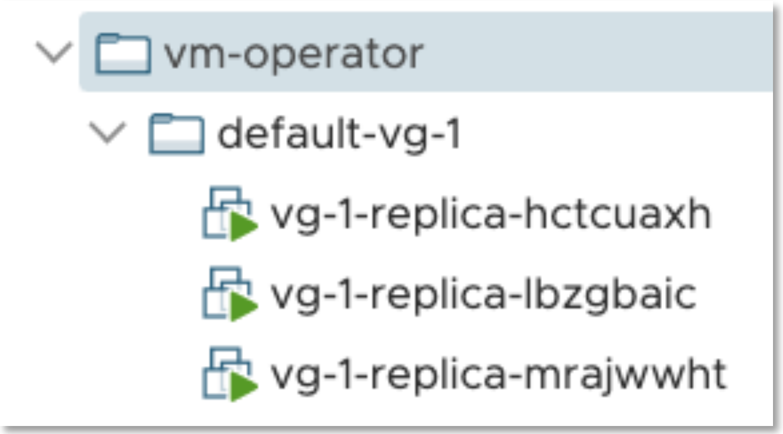
Our Goal for this Talk

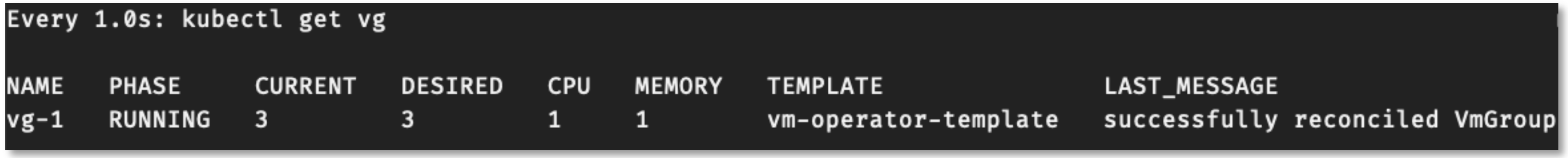
What are we going to build?



10 lines (9 sloc) | 168 Bytes

```yaml
apiVersion: vm.codeconnect.vmworld.com/v1alpha1
kind: VmGroup
metadata:
  name: vg-1
spec:
  cpu: 1
  memory: 1 # in GB
  replicas: 3
  template: vm-operator-template

```





```
Every 1.0s: kubectl get vg

NAME   PHASE     CURRENT   DESIRED   CPU   MEMORY   TEMPLATE               LAST_MESSAGE
vg-1   RUNNING   3         3         1     1        vm-operator-template   successfully reconciled VmGroup
```

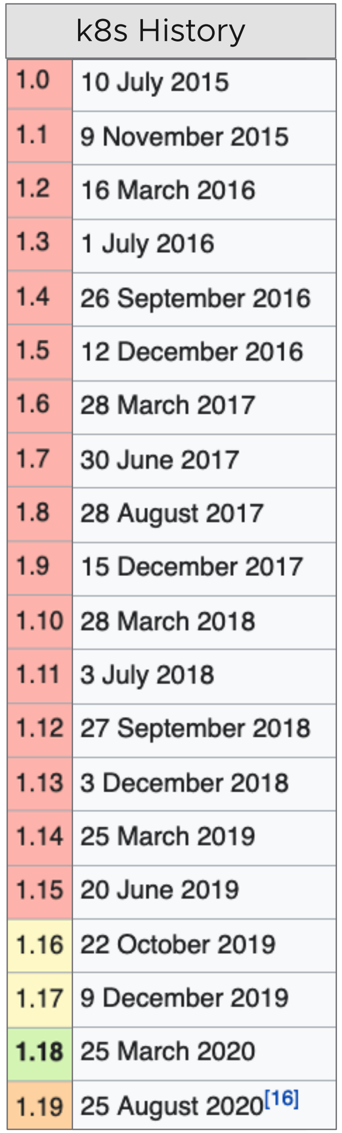
Extending Kubernetes with Custom Resources

Kubernetes Operators

Introduction

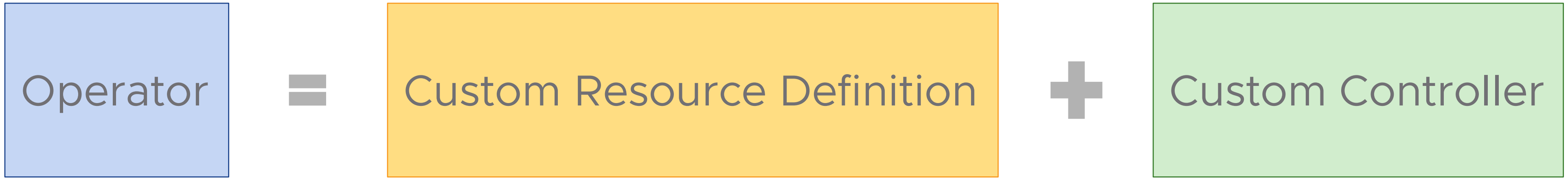
November 03, 2016 - CoreOS Blog Post by Brandon Philips

- Introduced and described the Concept of a Kubernetes Operator
- Published two Examples: etcd and Prometheus Operator



k8s History	
1.0	10 July 2015
1.1	9 November 2015
1.2	16 March 2016
1.3	1 July 2016
1.4	26 September 2016
1.5	12 December 2016
1.6	28 March 2017
1.7	30 June 2017
1.8	28 August 2017
1.9	15 December 2017
1.10	28 March 2018
1.11	3 July 2018
1.12	27 September 2018
1.13	3 December 2018
1.14	25 March 2019
1.15	20 June 2019
1.16	22 October 2019
1.17	9 December 2019
1.18	25 March 2020
1.19	25 August 2020[16]

An Operator is an application-specific controller that extends the Kubernetes API to create, configure, and manage instances of (complex) applications on behalf of a Kubernetes user.



Operator = Custom Resource Definition + Custom Controller

https://kubernetes.io/docs/concepts/extend-kubernetes/operator/
https://en.wikipedia.org/wiki/Kubernetes

Popular Kubernetes Operator Toolkits

Operator Framework ()

- Operator SDK (released Mid 2018 by Red Hat)
- Operator Lifecycle Manager
- Operator Registry
- Supported Languanges: Golang
- You can also use Helm Charts or Ansible Playbooks
- https://github.com/operator-framework

Kubebuilder

- Supported Languages: Golang
- https://github.com/kubernetes-sigs/kubebuilder
- https://book.kubebuilder.io/

Kopf

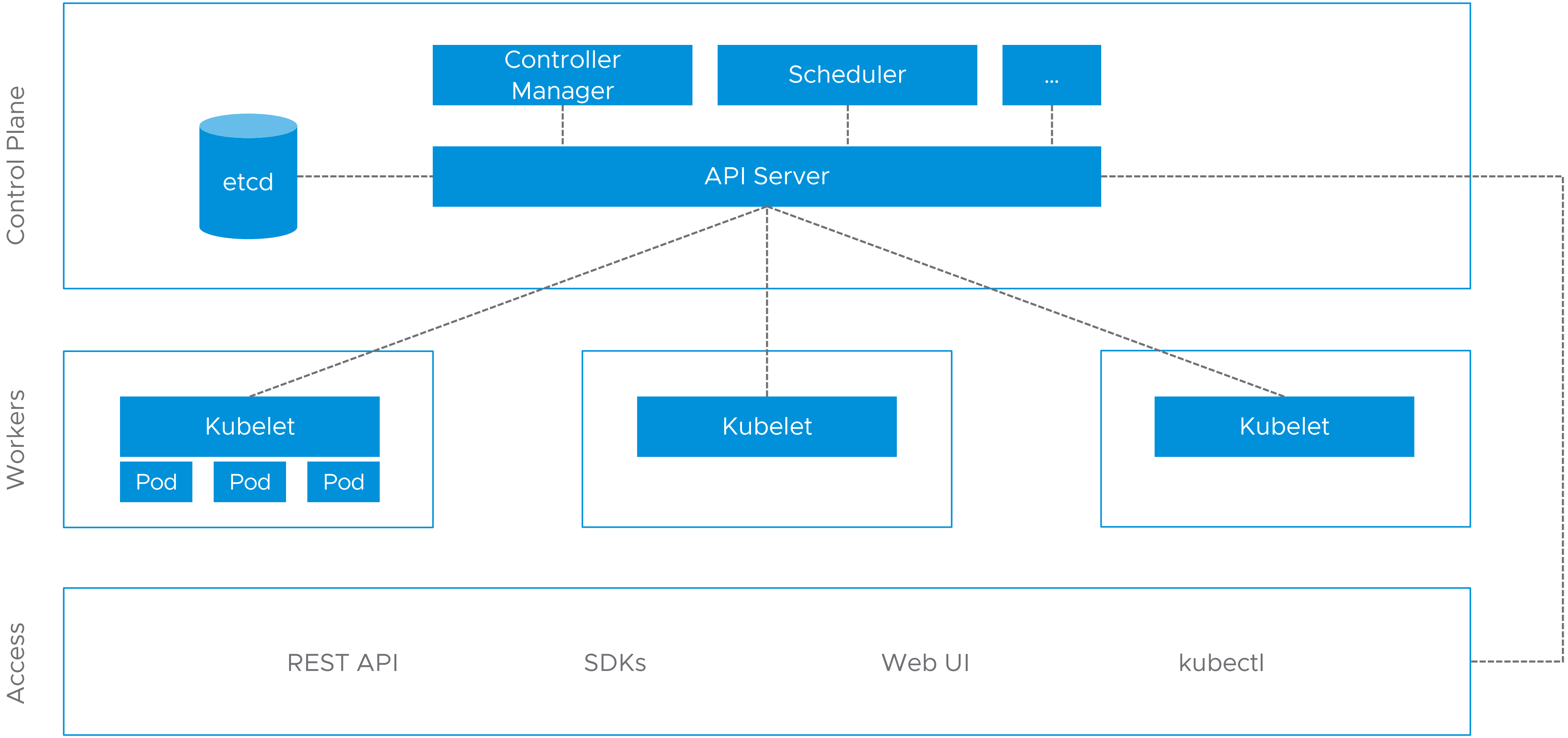
- Zalando Open Source Project
- Supported Languages: Python
- https://github.com/zalando-incubator/kopf

Kubernetes Architecture

From the View of a Controller Developer

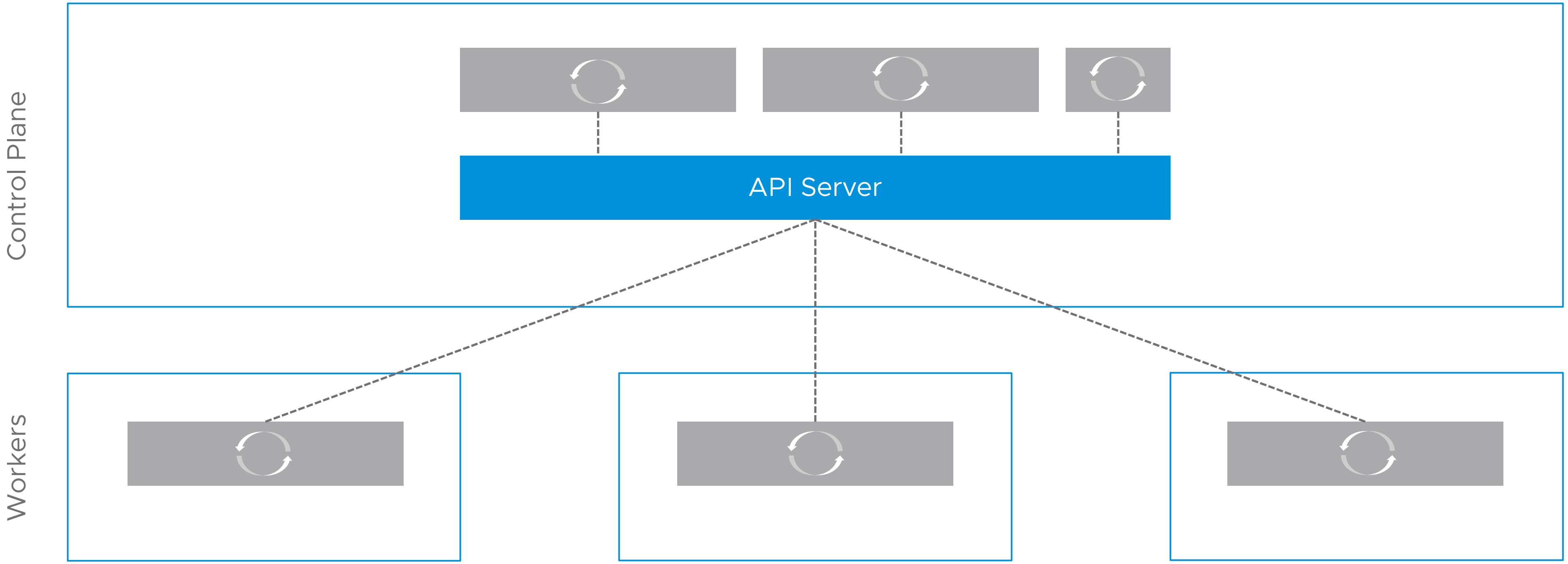
Kubernetes Architecture

Overview



Kubernetes Architecture

In Simple Terms…

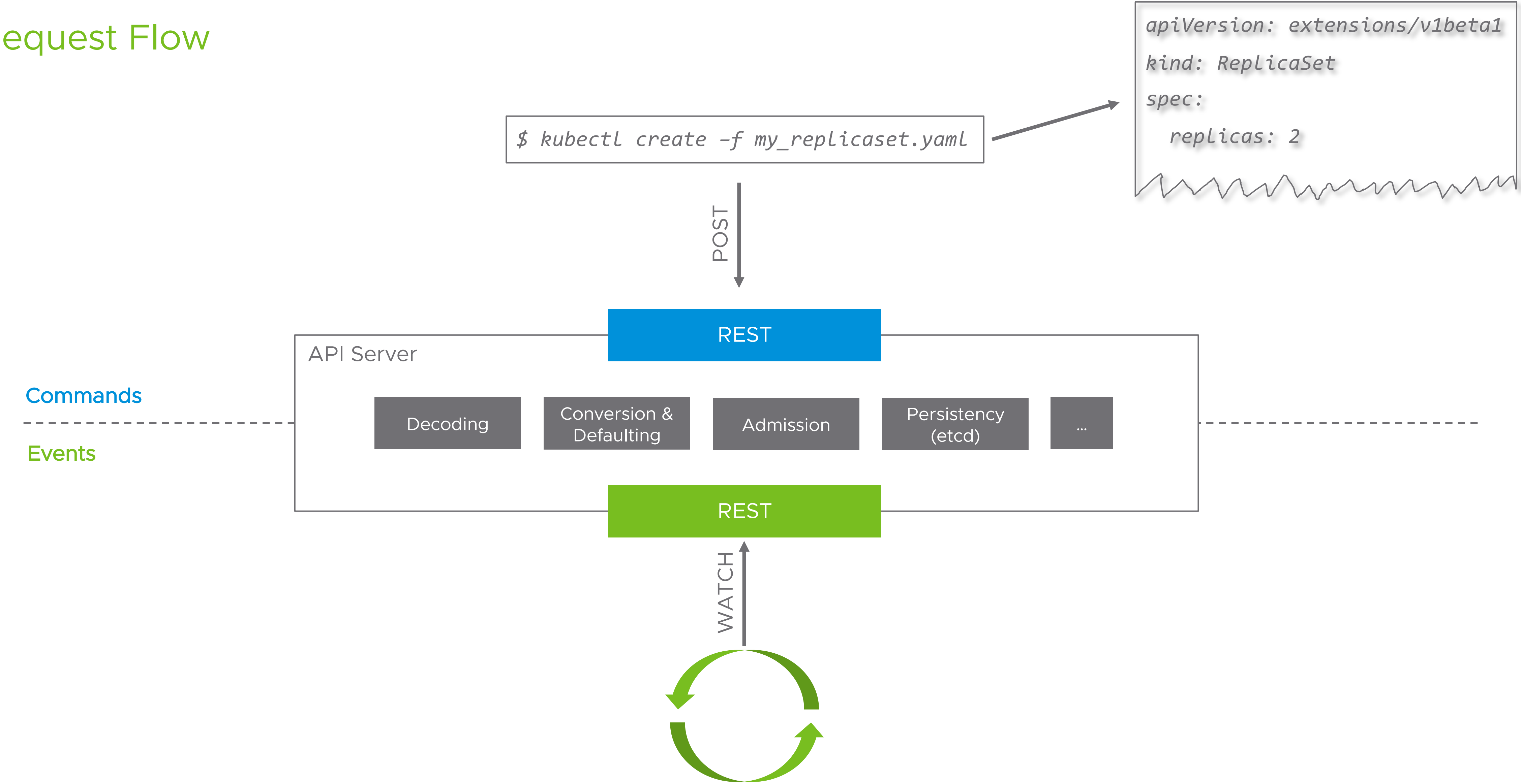


⟳ = Control Loop

©2020 VMware, Inc.

Kubernetes Architecture

Request Flow



```
$ kubectl create -f my_replicaset.yaml
```

```
apiVersion: extensions/v1beta1
kind: ReplicaSet
spec:
  replicas: 2
```

Kubernetes Architecture

Commands and Events

Commands

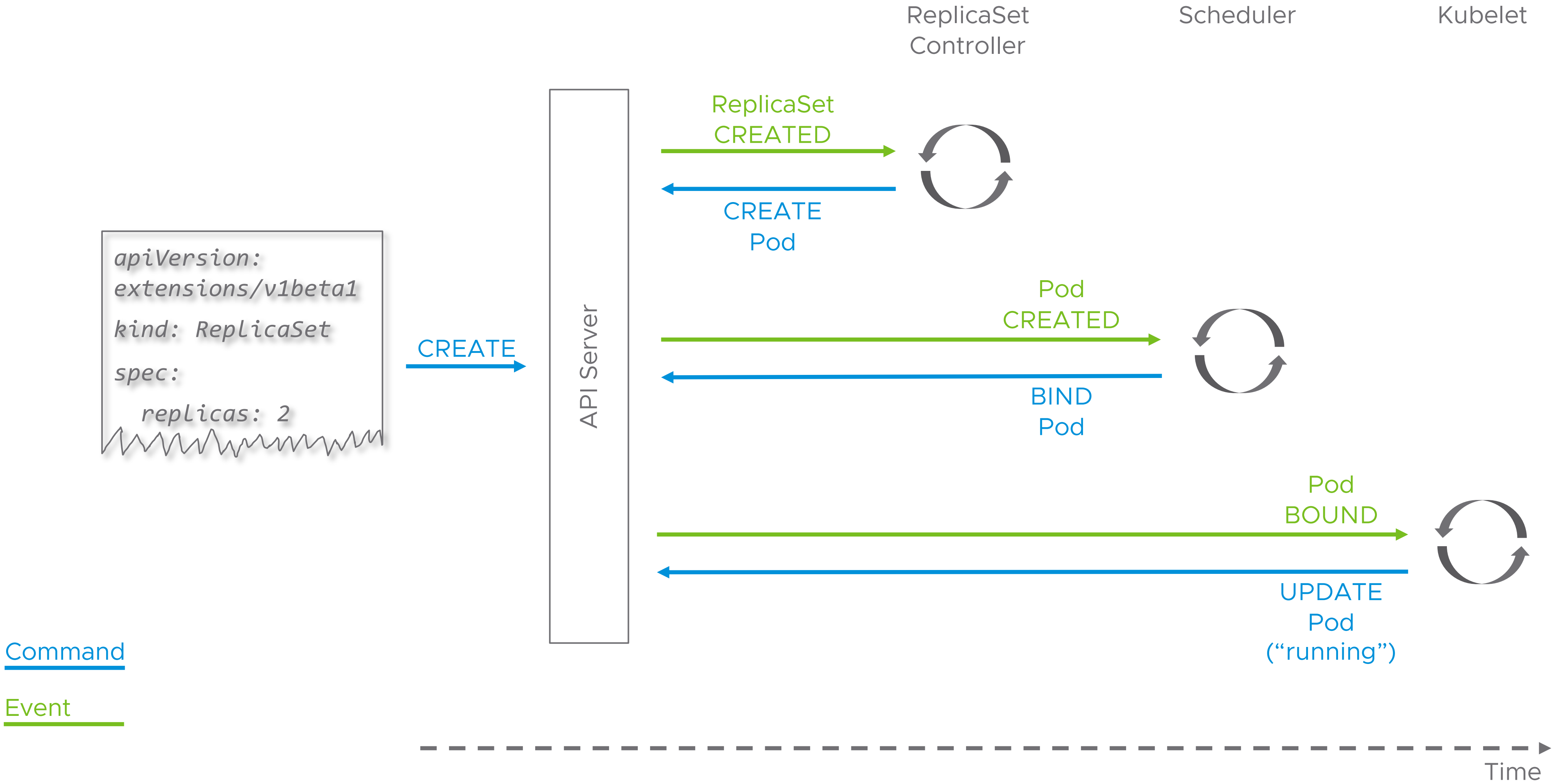
- Requests (**intent**) to do something
- Named in the **imperative**, e.g. “CREATE”
- **Can be rejected**
- **Higher coupling** between sender and owner
- Typically used in **synchronous** 1-to-1 request/response communication

Events

- Something that has happened (a **fact**)
- Named in **past tense**, e.g. “CREATED”
- Cannot (semantically) be rejected by receiver
- **Lowest coupling** between sender and owner
- **Asynchronous** 1-to-many communication, e.g. publish/subscribe

Kubernetes Architecture

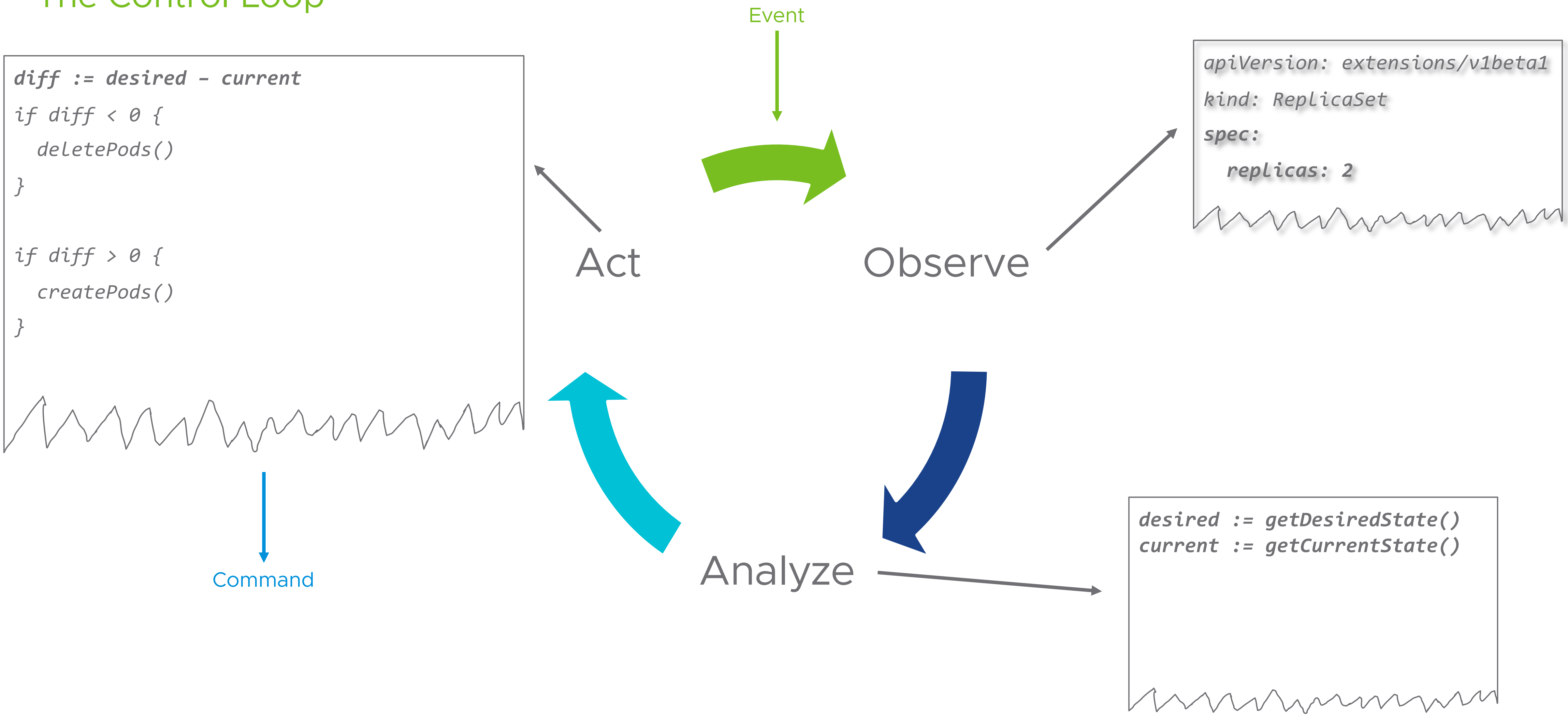
Choreography, not Orchestration



```
apiVersion:
extensions/v1beta1
kind: ReplicaSet
spec:
  replicas: 2
```

Kubernetes Architecture

The Control Loop



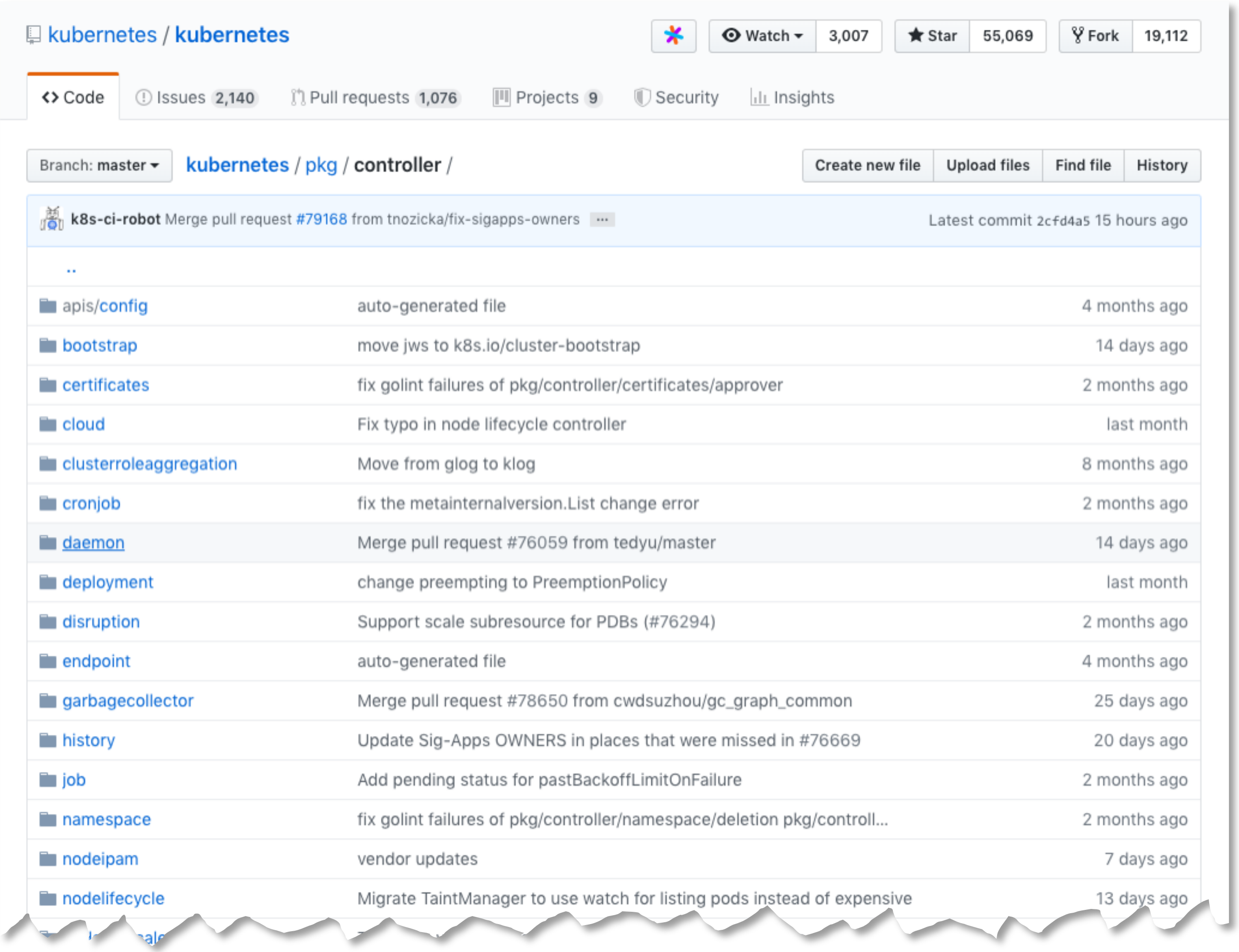
```
diff := desired - current
if diff < 0 {
  deletePods()
}

if diff > 0 {
  createPods()
}
```

```
apiVersion: extensions/v1beta1
kind: ReplicaSet
spec:
  replicas: 2
```

```
desired := getDesiredState()
current := getCurrentState()
```

Kubernetes Architecture

Controllers, oh my…

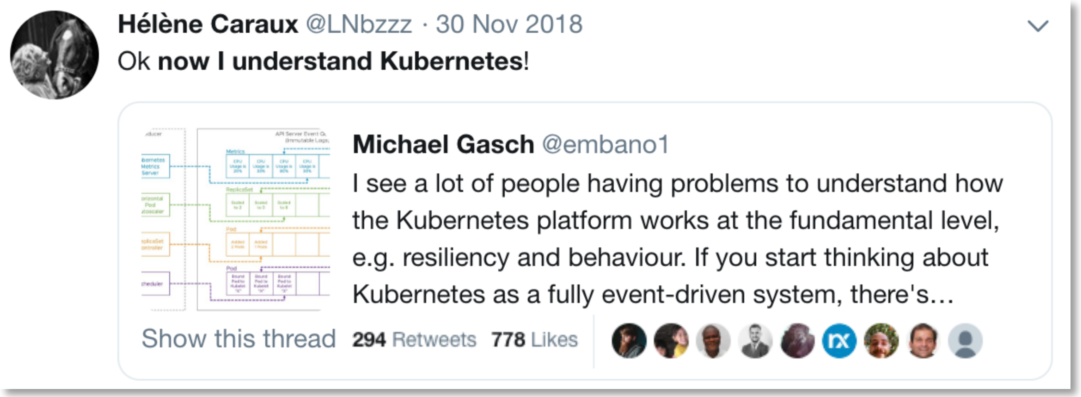


k8s-ci-robot Merge pull request #79168 from tnozicka/fix-sigapps-owners … Latest commit 2cfd4a5 15 hours ago

..		
apis/config	auto-generated file	4 months ago
bootstrap	move jws to k8s.io/cluster-bootstrap	14 days ago
certificates	fix golint failures of pkg/controller/certificates/approver	2 months ago
cloud	Fix typo in node lifecycle controller	last month
clusterroleaggregation	Move from glog to klog	8 months ago
cronjob	fix the metainternalversion.List change error	2 months ago
daemon	Merge pull request #76059 from tedyu/master	14 days ago
deployment	change preempting to PreemptionPolicy	last month
disruption	Support scale subresource for PDBs (#76294)	2 months ago
endpoint	auto-generated file	4 months ago
garbagecollector	Merge pull request #78650 from cwdsuzhou/gc_graph_common	25 days ago
history	Update Sig-Apps OWNERS in places that were missed in #76669	20 days ago
job	Add pending status for pastBackoffLimitOnFailure	2 months ago
namespace	fix golint failures of pkg/controller/namespace/deletion pkg/controll…	2 months ago
nodeipam	vendor updates	7 days ago
nodelifecycle	Migrate TaintManager to use watch for listing pods instead of expensive	13 days ago

Kubernetes Architecture

Further Reading



Hélène Caraux @LNbzzz · 30 Nov 2018

Ok **now I understand Kubernetes**!

Michael Gasch @embano1

I see a lot of people having problems to understand how the Kubernetes platform works at the fundamental level, e.g. resiliency and behaviour. If you start thinking about Kubernetes as a fully event-driven system, there's…

Show this thread **294** Retweets **778** Likes

Core Kubernetes: Jazz Improv over Orchestration

Joe Beda Follow

May 30, 2017 · 7 min read

Writing Controllers

Guidelines and Principles

Writing Controllers

Net Benefits Working With Kubernetes

Kubernetes comes with several features to make the life of a (controller) developer easier

- Scheduling and Supervision (self-healing*)
- Configuration and Secret Management
- Service Discovery and Networking
- Storage Management
- (Cloud) Portability
- Declarative API Stability and Extensibility (CRDs)
- AuthN and AuthZ (RBAC)
- SDKs
- (Probably a lot more than we could fit on this slide)

* Self-healing (eventual recovery) != HA

Writing Controllers

BUT!

Depending on the complexity of your controller, there might be a steep learning curve

- Lots of primitives and objects to learn
- client-go (de facto SDK) *“is not for mere mortals”* (Bryan Liles)
- Optimistic concurrency in an asynchronous eventual consistent system
  - There is No Now
  - The (global) state is always behind you (distributed, delayed and unknown to the local observer)
- Fast moving project

Writing Controllers

Required Mindset

Autonomous Processes

Single Responsible Principle

Decoupling via event-driven messaging

No central coordinator

Concurrency & Asynchrony

Eventual consistent by design

Don’t rely on (assume) order

Stateless over Stateful

API server (etcd) is the source of truth*

In-memory cache via reconciliation

Defensive Programming

Things will go wrong (crash)

No shared (wall) clock

Anticipate effects on the rest of the system

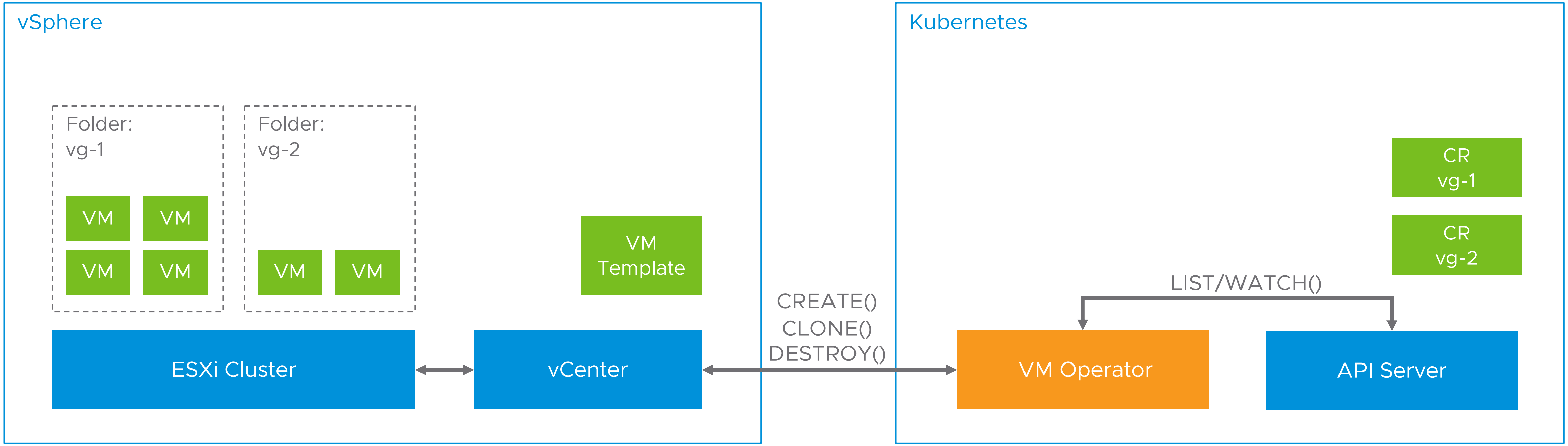
Side Effects

Delivery and processing guarantees only within Kubernetes

Show me the {code}!

VM Operator

Conceptual Overview



DEMO

https://github.com/embano1/codeconnect-vm-operator

SCAN ME

Resources

Links and Resources (1)

VMware {code} Connect VM Operator Github Repository

- https://github.com/embano1/codeconnect-vm-operator

Kubebuilder Quick Start

- https://book.kubebuilder.io/quick-start.html

Tutorial: Zero to Operator in 90 Minutes! - Solly Ross:

- https://www.youtube.com/watch?v=KBTXBUVNF2I

KubeBuilder Workshops

- https://github.com/directxman12/kubebuilder-workshops

Programming Kubernetes (Book)

- https://learning.oreilly.com/library/view/programming-kubernetes/9781492047094/

Links and Resources (2)

Kubernetes API Reference (v1.18)

- https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.18/

Kubernetes Sample Controller (CRD)

- https://github.com/kubernetes/sample-controller

Writing Controllers

- https://github.com/kubernetes/community/blob/master/contributors/devel/sig-api-machinery/controllers.md

Internals of how Controllers are implemented under the Covers

- https://medium.com/@cloudark/kubernetes-custom-controllers-b6c7d0668fdf

Kubernetes Operator Best Practices

- https://blog.openshift.com/kubernetes-operators-best-practices/

Links and Resources (3)

KubeCon Sessions (Controllers and Operators)

- Writing Kube Controllers for Everyone
  - https://www.youtube.com/watch?v=AUNPLQVxvmw
- Writing Kubernetes Controllers for CRDs: Challenges, Approaches and Solutions
  - https://www.youtube.com/watch?v=7wdUa4Ulwxg
- How We Built Contour, and What You Can Learn From Our Experience
  - https://www.youtube.com/watch?v=4usXJE0EwHo
- Keep the Space Shuttle Flying: Writing Robust Operators
  - https://www.youtube.com/watch?v=uf97lOApOv8
- Controllers: Lambda Functions for Extending your Infrastructure
  - https://www.youtube.com/watch?v=TM-2GgQ6Q2A

Links and Resources (4)

KubeCon Sessions (Architecture)

- The Kubernetes Control Plane for Busy People Who Like Pictures
  - https://www.youtube.com/watch?v=zCXiXKMqnuE
- The Magic of Kubernetes Self-Healing Capabilities
  - https://www.youtube.com/watch?v=91dgNqma7-Q&t=2s
- Kubernetes Design Principles: Understand the Why
  - https://www.youtube.com/watch?v=ZuIQurh_kDk&t=1s

Thank you!

vmworld® 2020

possible together

vmware®