THAT®

CONFERENCE

Red Hat

Unlocking the Power of Kubernetes: Create Your Own Resources with CRDs

Rose Crisp

Senior Software Engineer @ Red Hat

What we will discuss today

- Kubernetes Operator Pattern
- Custom Resource Definition
- Custom Controllers
- Create/Deploy Operator with Operator-sdk
- Demo

Kubernetes is a container orchestration platform that automates the deployment, scaling, and management of containerized applications.

- Load balancing
- Storage
- Self-healing
- Secrets
- Configuration
- Networking

Managing Stateless and Stateful applications

Stateless Applications

- Does not require data persistence
- No Backup
- Does not require additional knowledge about how these applications operate

Stateful Applications

- Store state of each transaction
- Backup
- Domain-specific knowledge in order to scale, upgrade, and reconfigure applications

Tasks

- Installing database software
- Creating databases
- Performing upgrades of the database and software to new releases
- Starting and shutting down the database instance
- Managing the storage structures of the database
- Managing users and security
- Managing database objects, such as tables, indexes, and views
- Backing up the database and performing recovery operations when necessary
- Monitoring the state of the database and taking preventive or corrective action as required
- Monitoring and tuning database performance
- Diagnosing and reporting critical errors

Stateful Application

Demo

DATASTAX

DS DataStax Kubernetes Operator for Apache Cassandra

Benefits of using Kubernetes Operator

- Custom behavior
- Native Kubernetes experience
- Consistency
- Leverage the Kubernetes ecosystem

Kubernetes Operator

1 Custom Resource Definition

+

2 Custom Controller

+

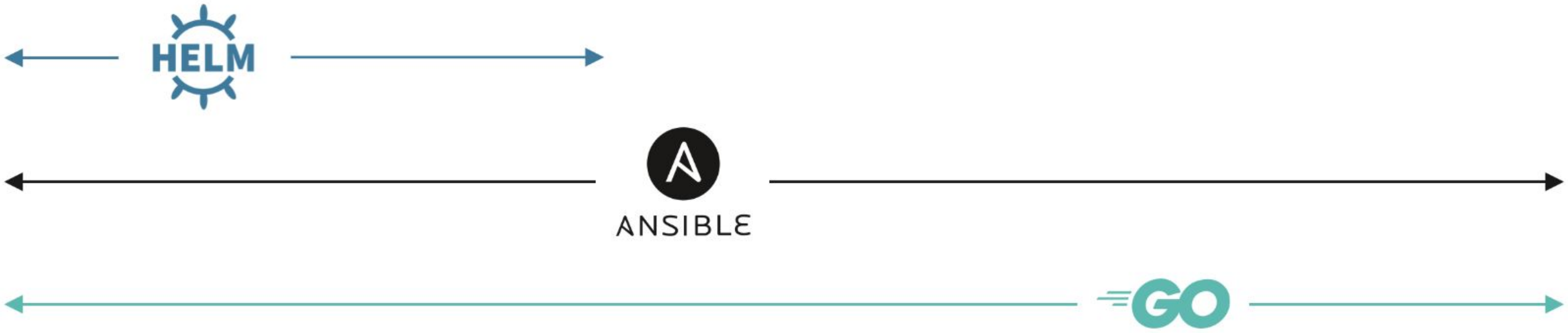
3 Business Logic

= **Operator**

https://github.com/opdev/l5-operator-demo/blob/main/config/crd/bases/pets.bestie.com_besties.yaml

Operator Capability Levels

Level I	Level II	Level III	Level IV	Level V
Basic Install	**Seamless Upgrades**	**Full Lifecycle**	**Deep Insights**	**Auto Pilot**
Automated application provisioning and configuration management	Patch and minor version upgrades supported	App lifecycle, storage lifecycle (backup, failure recovery)	Metrics, alerts, log processing and workload analysis	Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning



Kubernetes Operator

1 Custom Resource Definition + 2 Custom Controller + 3 Business Logic

= **Operator**

Defining: What is a Kubernetes Resource?

A resource is an **endpoint** in the Kubernetes **API** that stores a collection of API objects of a certain kind; for example, the pods resource contains a collection of Pod objects.

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
    - name: my-container
      image: nginx:latest
      ports:
        - containerPort: 80
```

What are Custom Resource Definitions?

Kubernetes extension mechanism that allows users to **Define custom resources** and their **properties**.

Why does **CRD matter**?

It allows users to Expand the functionally of Kubernetes

Without having to restart it.

https://github.com/opdev/l5-operator-demo/blob/main/config/crd/bases/pets.bestie.com_besties.yaml

Type Metadata

```
apiVersion: apiextensions.k8s.io/v1

kind: CustomResourceDefinition
```

Everything we need to build our URI*

`/apis/apiextensions.k8s.io/v1/customresourcedefinitions`

*Custom Resource Definition is a "cluster-scoped" resource, so no namespace value is required. Namespaced resources need additional information in the metadata.

Object Metadata

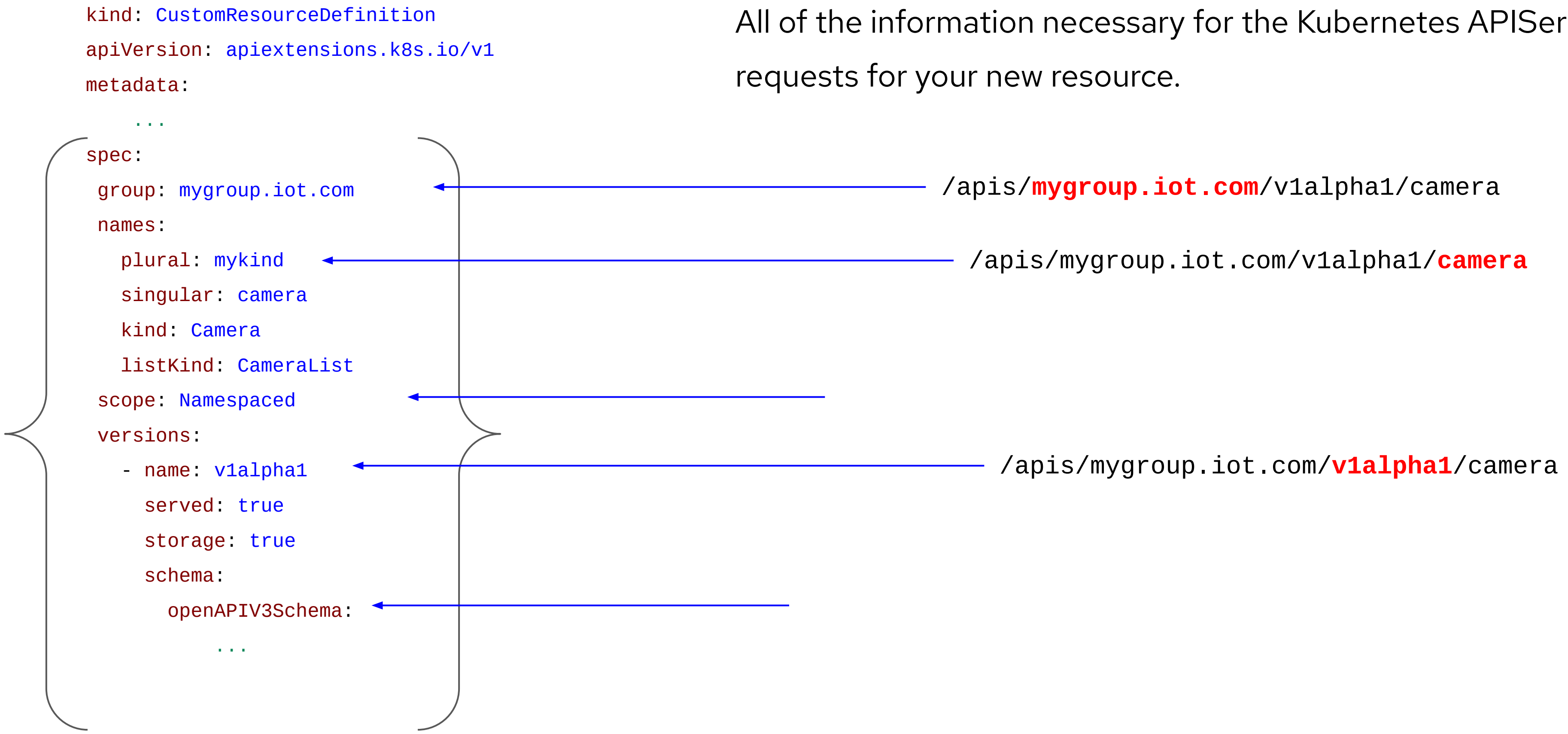
```
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.7.0
  creationTimestamp: null
  name: besties.pets.bestie.com
```

Additional metadata specific to **this** instance of the given resource.

/apis/**apiextensions.k8s.io**/**v1**/customresourcedefinitions/iot.operator.openshift.io

Spec

All of the information necessary for the Kubernetes APIServer to handle requests for your new resource.



```
kind: CustomResourceDefinition
apiVersion: apiextensions.k8s.io/v1
metadata:
    ...
spec:
 group: mygroup.iot.com
 names:
   plural: mykind
   singular: camera
   kind: Camera
   listKind: CameraList
 scope: Namespaced
 versions:
   - name: v1alpha1
     served: true
     storage: true
     schema:
       openAPIV3Schema:
           ...
```

/apis/**mygroup.iot.com**/v1alpha1/camera

/apis/mygroup.iot.com/v1alpha1/**camera**

/apis/mygroup.iot.com/**v1alpha1**/camera

Status

```
...
status:
 conditions:
   - type: NamesAccepted
     status: 'True'
     lastTransitionTime: '2023-04-07T18:34:40Z'
     reason: NoConflicts
     message: no conflicts found
   - type: Established
     status: 'True'
     lastTransitionTime: '2020-12-07T18:34:40Z'
     reason: InitialNamesAccepted
     message: the initial names have been accepted
 acceptedNames:
   plural: cameras
   singular: camera
   kind: Camera
   listKind: CameraList
 storedVersions:
   - v1
```

The current status of this resource as determined by the Kubernetes controller responsible for this particular Kind (Custom Resource Definition)

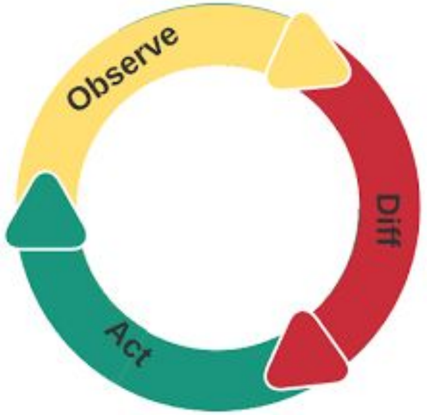
Kubernetes Operator

1 Custom Resource Definition + 2 Custom Controller + 3 Business Logic

= **Operator**

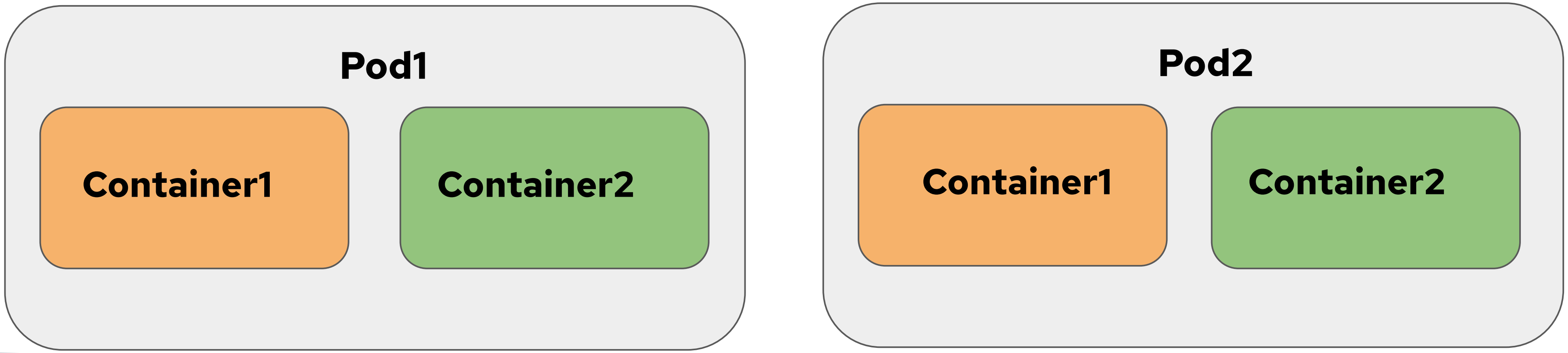
Kubernetes

Controller



Current state always match the Desired state

“I want 2 pods running in my cluster”



Pod1

Container1 | Container2

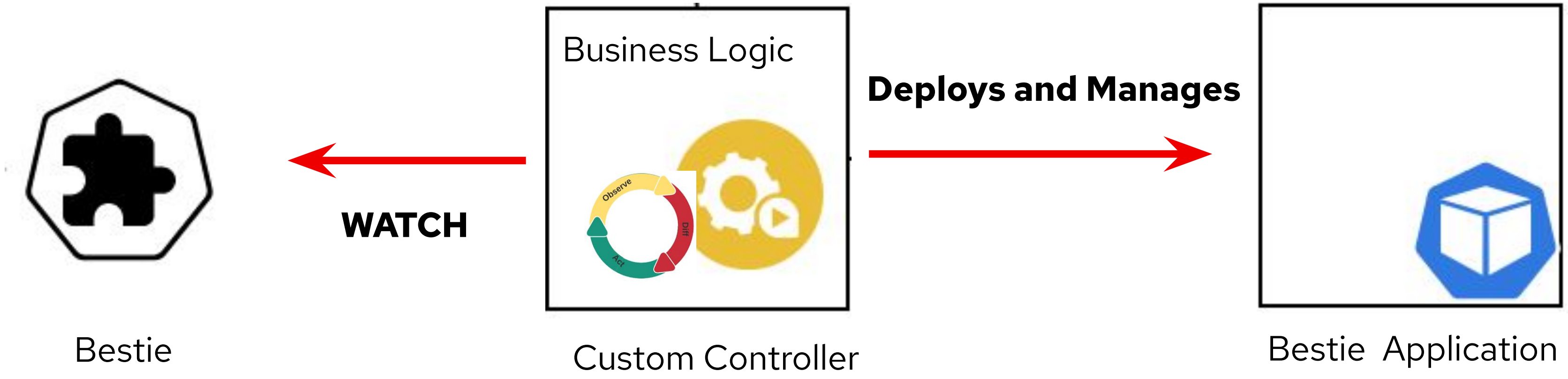
Pod2

Container1 | Container2

Kubernetes Controllers

- ReplicaSet
- Deployment
- DaemonSet
- Job

Custom Controller



Business Logic

WATCH

Deploys and Manages

Bestie

Custom Controller

Bestie Application

An Operator is?

1 + 2 + 3

Custom Resource Definition

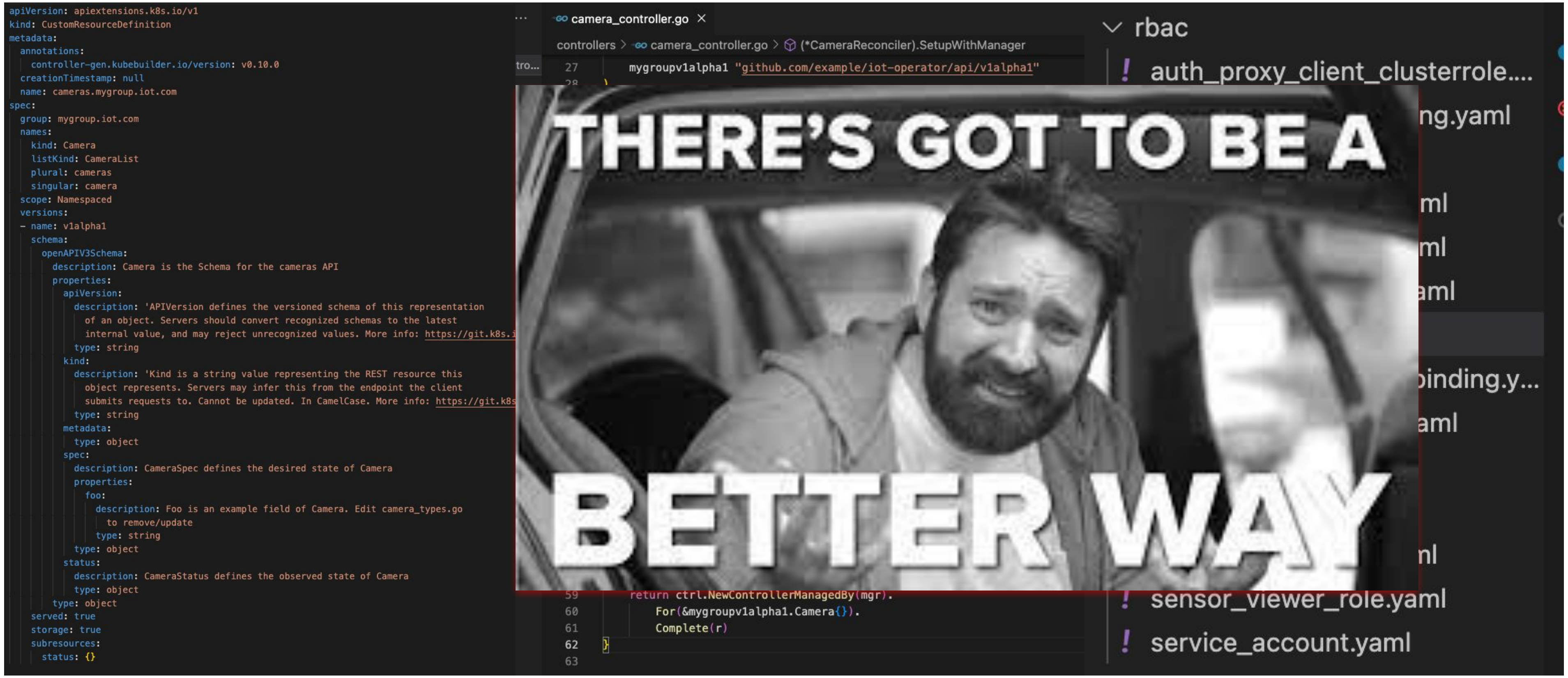
Custom Controller

Business Logic

3 2 1

A Kubernetes operator is an application-specific controller that extends the functionality of the Kubernetes API to create, configure, and manage instances of complex applications.

Writing CRDs, Custom Controllers by hand is tedius



```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.10.0
  creationTimestamp: null
  name: cameras.mygroup.iot.com
spec:
  group: mygroup.iot.com
  names:
    kind: Camera
    listKind: CameraList
    plural: cameras
    singular: camera
  scope: Namespaced
  versions:
  - name: v1alpha1
    schema:
      openAPIV3Schema:
        description: Camera is the Schema for the cameras API
        properties:
          apiVersion:
            description: 'APIVersion defines the versioned schema of this representation
              of an object. Servers should convert recognized schemas to the latest
              internal value, and may reject unrecognized values. More info: https://git.k8s.i
            type: string
          kind:
            description: 'Kind is a string value representing the REST resource this
              object represents. Servers may infer this from the endpoint the client
              submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s
            type: string
          metadata:
            type: object
          spec:
            description: CameraSpec defines the desired state of Camera
            properties:
              foo:
                description: Foo is an example field of Camera. Edit camera_types.go
                  to remove/update
                type: string
            type: object
          status:
            description: CameraStatus defines the observed state of Camera
            type: object
        type: object
    served: true
    storage: true
    subresources:
      status: {}
```

camera_controller.go

controllers > camera_controller.go > (*CameraReconciler).SetupWithManager

```go
	mygroupv1alpha1 "github.com/example/iot-operator/api/v1alpha1"
)
	return ctrl.NewControllerManagedBy(mgr).
		For(&mygroupv1alpha1.Camera{}).
		Complete(r)
}
```

Tools and libraries to help you build your Operator

- Charmed Operator Framework
- Java Operator SDK
- Kopf (Kubernetes Operator Pythonic Framework)
- kube-rs (Rust)
- kubebuilder
- KubeOps (.NET operator SDK)
- KUDO (Kubernetes Universal Declarative Operator)
- Mast
- Metacontroller along with WebHooks that you implement yourself
- Operator-SDK
- shell-operator

Introducing Operator-sdk

Creating Operator with Operator-sdk

```
$ operator-sdk init --domain example.com --repo github.com/example/my-operator
```

```
$ operator-sdk create api --group mygroup --version v1alpha1 --kind Mykind
--resource --controller
```

Deploying Operator with Operator-sdk

```
$ make docker-build docker-push

$ make deploy
```

Demo

Key Takeaways

- Kubernetes is a container orchestration platform that automates the deployment, scaling, and management of containerized applications.
- CRD allows users to extend Kubernetes API infinitely without the need to rebuild Kubernetes
- CRD + custom controller + Business Logic = Operator
- Operator-sdk is tool to build and manage operators

Find us on GitHub

Our project is on GitHub! Visit us at https://github.com/opdev/l5-operator-demo to view our code, contribute to the project, or contact us. Thank you for your support!
Operator-sdk uses the controller-runtime library

Session Feedback

SPECIAL THANKS TO ALL OUR AWESOME CAMP SPONSORS!

TruStage™

HEARTLAND BUSINESS SYSTEMS

Intellibus

aws

Unspecified SOFTWARE CO

Architect.io

ionic

CAMUNDA

alexa

Google Cloud

CORE BTS

GrapeCity®