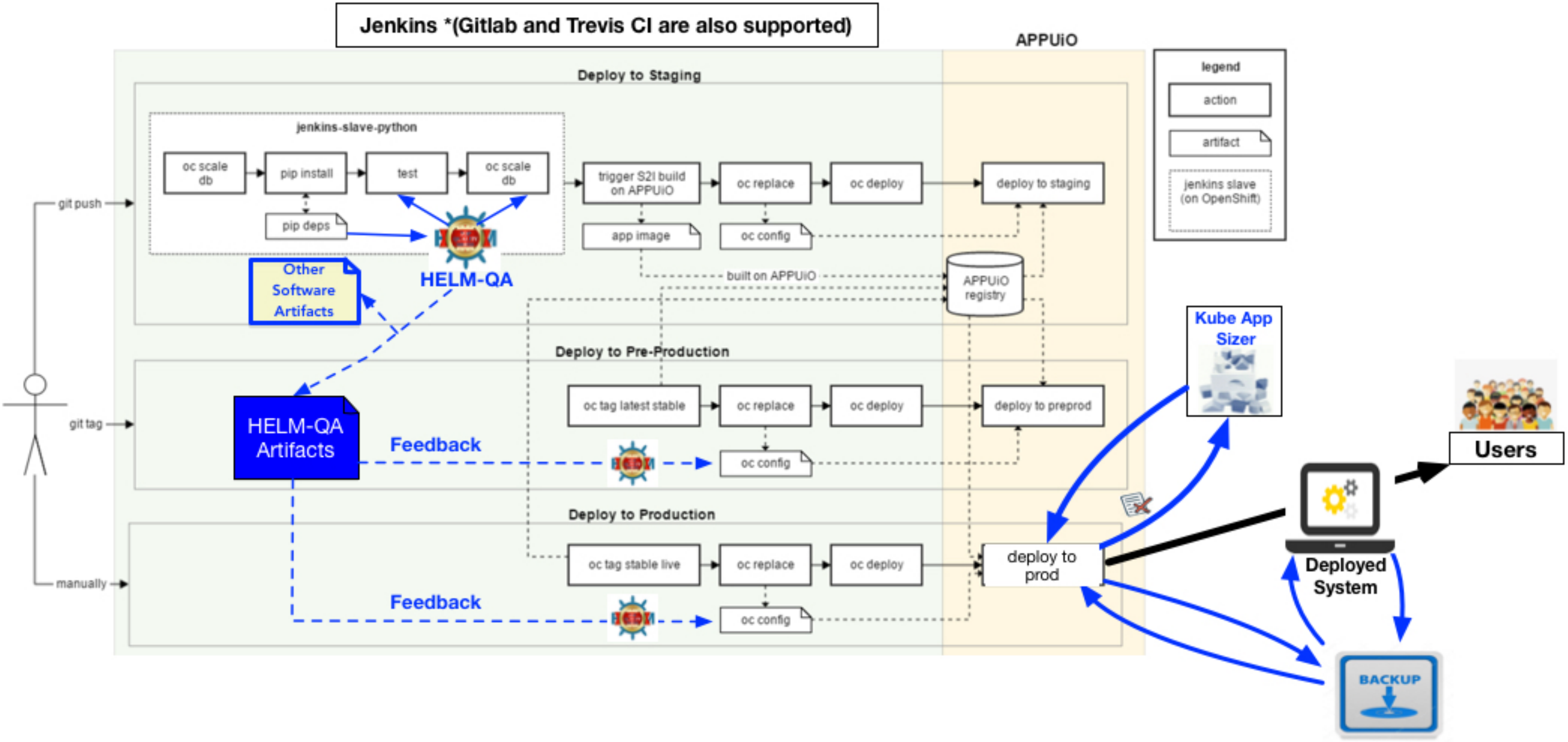
Kubernetes, Operators SDK, And Prometheus

Summer School on Software Evolution: From Monolithic to Cloud-Native

Dr. Sebastiano Panichella
Zurich University of Applied Science (ZHAW)
https://spanichella.github.io/

Dr. Soheila Dehghanzadeh
Denso Automotive Deutschland
https://www.linkedin.com/in/soheiladehghanzadeh/

Our Vision



Service Prototyping Lab (blog.zhaw.ch/splab)

Outline

- **Quick Introduction - Kubernetes and Operators SDK**
    - Current Vision, Challenges, and Opportunities
- **Tutorial on Kubernetes Operators:**
    - **Operators SDK**
        - Example 1: Simple Operator Example
        - Example 2: Advanced Operator Example
- **Prometheus**

Outline

- **Quick Introduction - Kubernetes and Operators SDK**
    - Current Vision, Challenges, and Opportunities
- **Tutorial on Kubernetes Operators:**
    - Operators SDK
        - Example 1: Simple Operator Example
        - Example 2: Advanced Operator Example
- **Prometheus**

Quick Introduction: Kubernetes and Operators SDK

- **Material partially based on previous blogposts & project collaborations**

https://vshn.ch/en/blog/how-to-leverage-kubernetes-operators-using-the-operator-sdk-framework/

https://vshn.ch/en/

- https://vshn.ch/en/blog/introduction-to-kubernetes-operators-operator-framework-and-operators-sdk/
- https://vshn.ch/en/blog/supported-kubernetes-operator-sdk-workflows/
- https://vshn.ch/en/blog/examples-of-supported-kubernetes-operator-sdk-workflows/

Prerequisites

https://github.com/spanichella/operator-sdk-examples-1/blob/master/Documentation/1_Prerequisites.pdf

(Short link: **https://bit.ly/2FusHkf**)

- git clone https://github.com/spanichella/operator-sdk-examples

Other Prerequisites

- dep version v0.5.0+
    - https://golang.github.io/dep/docs/installation.html
- git
- go version v1.12+.
    - https://golang.org/dl/. Or https://nats.io/documentation/tutorials/go-install/
    - Set $GOPATH\
- docker version 17.03+.
    - https://docs.docker.com/install/

- **Minishift installed (or access to a Kubernetes v1.11.3+ cluster):**
    - **https://github.com/minishift/minishift**
    - **kubectl version v1.11.3+.**

Prerequisites

https://github.com/spanichella/operator-sdk-examples-1/blob/master/Documentation/1_Prerequisites.pdf

(Short link: **https://bit.ly/2FusHkf**)

- git clone https://github.com/spanichella/operator-sdk-examples

Other Prerequisites

- dep version v0.5.0+
  - https://golar
- git
- go version v1.12+.
  - https://golan all/
  - Set $GOPAT
- docker version 17.03
  - https://docs.dock
- **Minishift installed (**
  - **https://github.c**
  - **kubectl version v1.11.3+.**

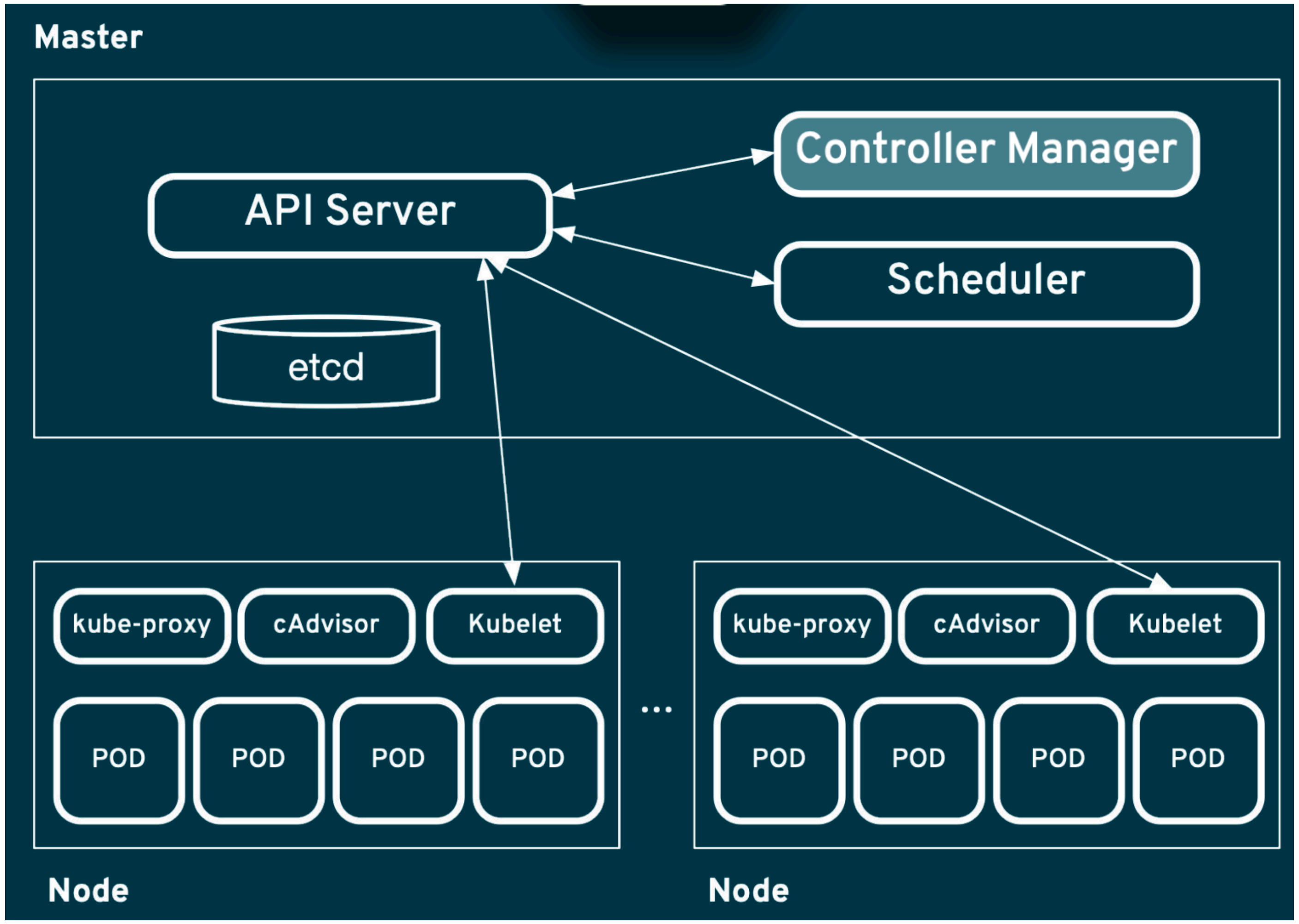
Outline

- **Quick Introduction - Kubernetes and Operators SDK**
    - Current Vision, Challenges, and Opportunities
- **Tutorial on Kubernetes Operators:**
    - Operators SDK
        - Example 1: Simple Operator Example
        - Example 2: Advanced Operator Example
- **Prometheus**

Kubernetes

What is Kubernetes?

- Open source platform for managing containerized workloads and services
- Containers, containers, containers
- Name originates from Greek - helmsman or pilot
- Google open-sourced in 2014
- Based on Borg - Google' internal project
- K_ _ _ _ _ _ _ _ _S -> **k8s**



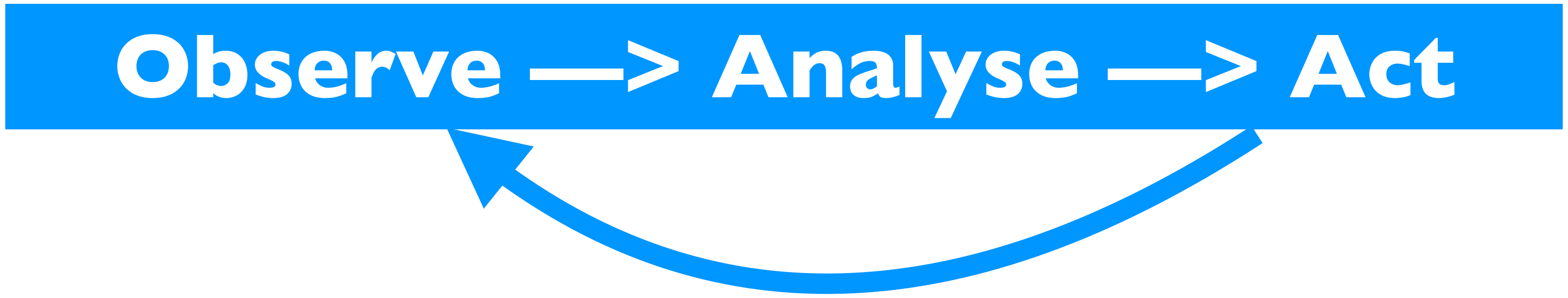
A Kubernetes application is an application that is both deployed on Kubernetes and managed using the Kubernetes **APIs** and **kubectl** tooling.

What is great about Kubernetes?

- Scalability of workloads
- Separation of workloads
- Native stable resources
  - (Deployments, Pods)
- API / Custom Resources

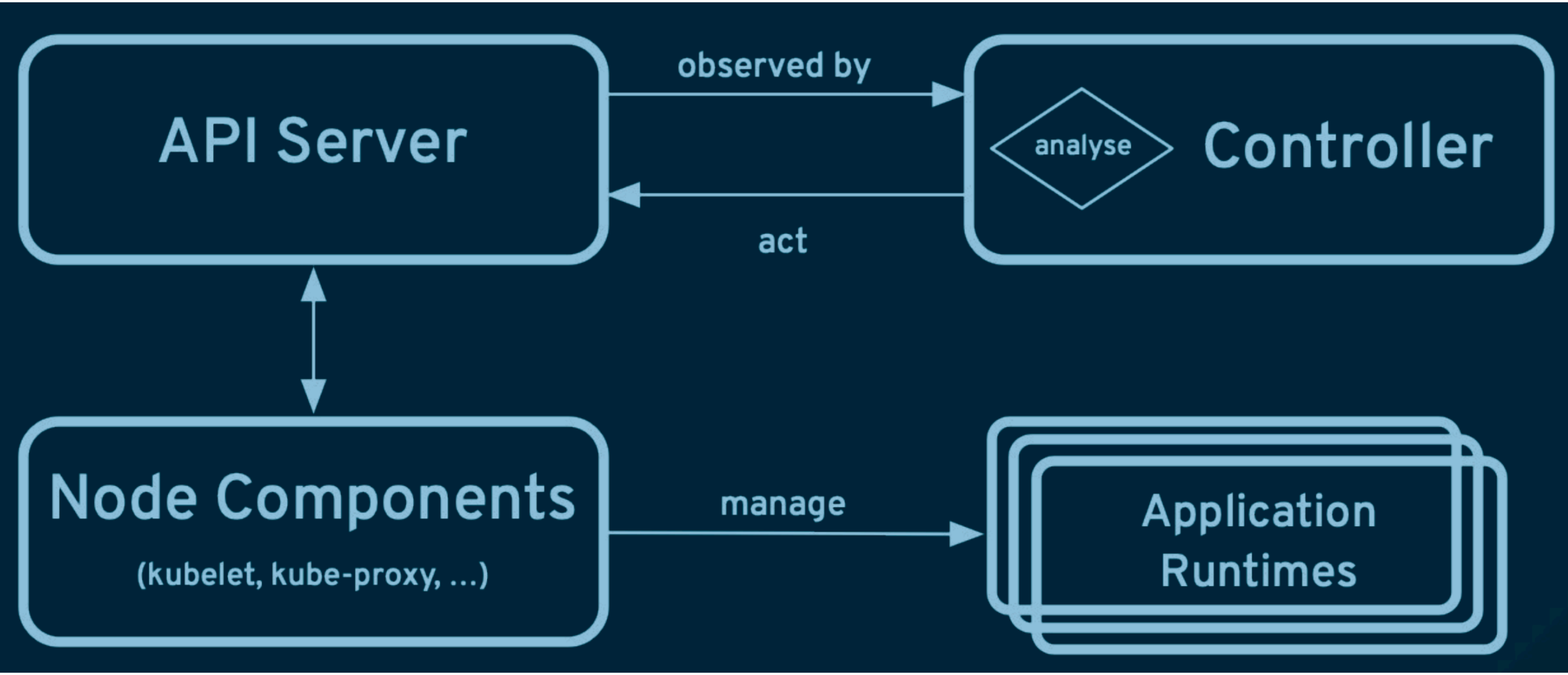
Kubernetes Controllers

- Kubernetes: Declaratige orchestration platform
- Based on resource objects for declaring target state
- Reconciliation:
  - Observe **current** state
  - Analyse and compare against **declared** state
  - **Bring** current state **closer** to declared state



Observe —> Analyse —> Act

Kubernetes Controllers



Observe —> Analyse —> Act

Kubernetes Controllers

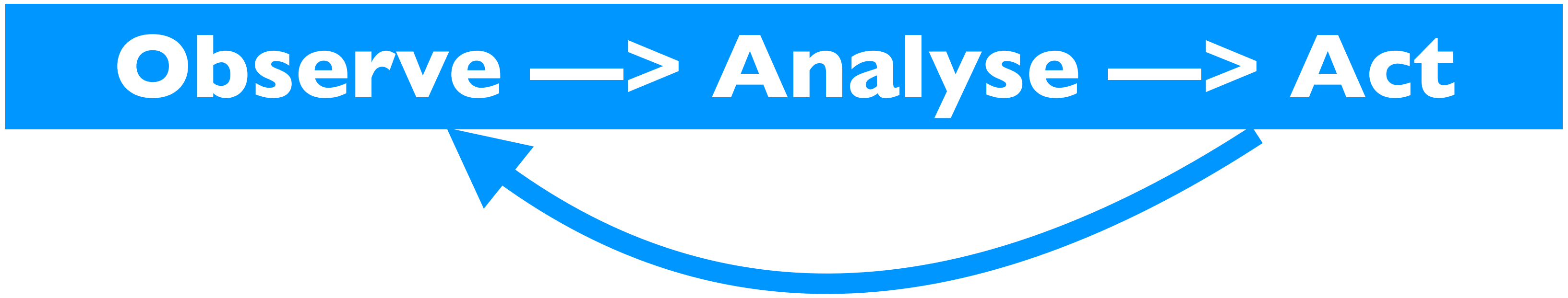
- **Watches Kubernetes Resources**

- **Enhances platform behaviour or introduces new feautures**
  - **Non terminating** loop that regulates the state of the system
  - A control loop that **watches** the **shared state of the cluster** via the **API server** and **makes changes** to move from current **to desired state**

- **Regular Kubernetes application** (Deployments, Pod)

- **Running permanently in the background**

- **Common Custom Controller triggers:**
  - Labels
  - Annotations
  - ConfigMap



Observe —> Analyse —> Act

Kubernetes Operators

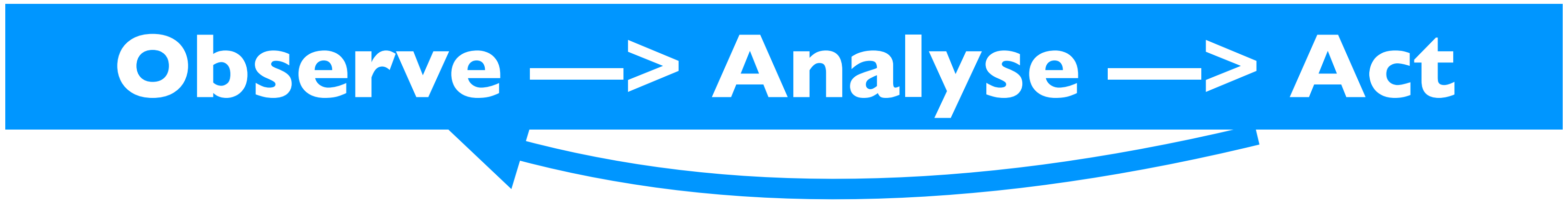
> *“ Operator is a K8s controller - specific to operating an application. It can automate tasks that usually require a human… ”*

An Operator is a *Kubernetes native application* that deals with the Kubernetes API and Custom Resources to create/operate new Resources, *to facilitate the management of complex stateful applications*.

Operator **is a method of packaging, deploying and managing a Kubernetes application.**

It’s an intelligent piece of software that embed the templating to deploy your resources.

The Operator **watch events on the K8s API** and **react** (ex : re-create a pod, change Labels, update a Secret, Remove a Service…)



Kubernetes Operators: Terminology

> *“ Operator is a K8s controller - specific to operating an application. It can automate tasks that usually require a human… ”*

- Operator **IS-A** Controller
- Makes use of CRDs - **Custom Resource Definition**
- **Semantically:** Operator automates operational tasks, by *holding the **knowledge** of **how an application needs** to be **deployed, managed** and **packaged***
- **Technically:** Operator = Controller + CRD = **Extends** the Kubernetes API to make the application a custom resource - part of the Kubernetes ecosystem
- **Reconciles given state..**
- **CoreOS** pioneered the Operator movement

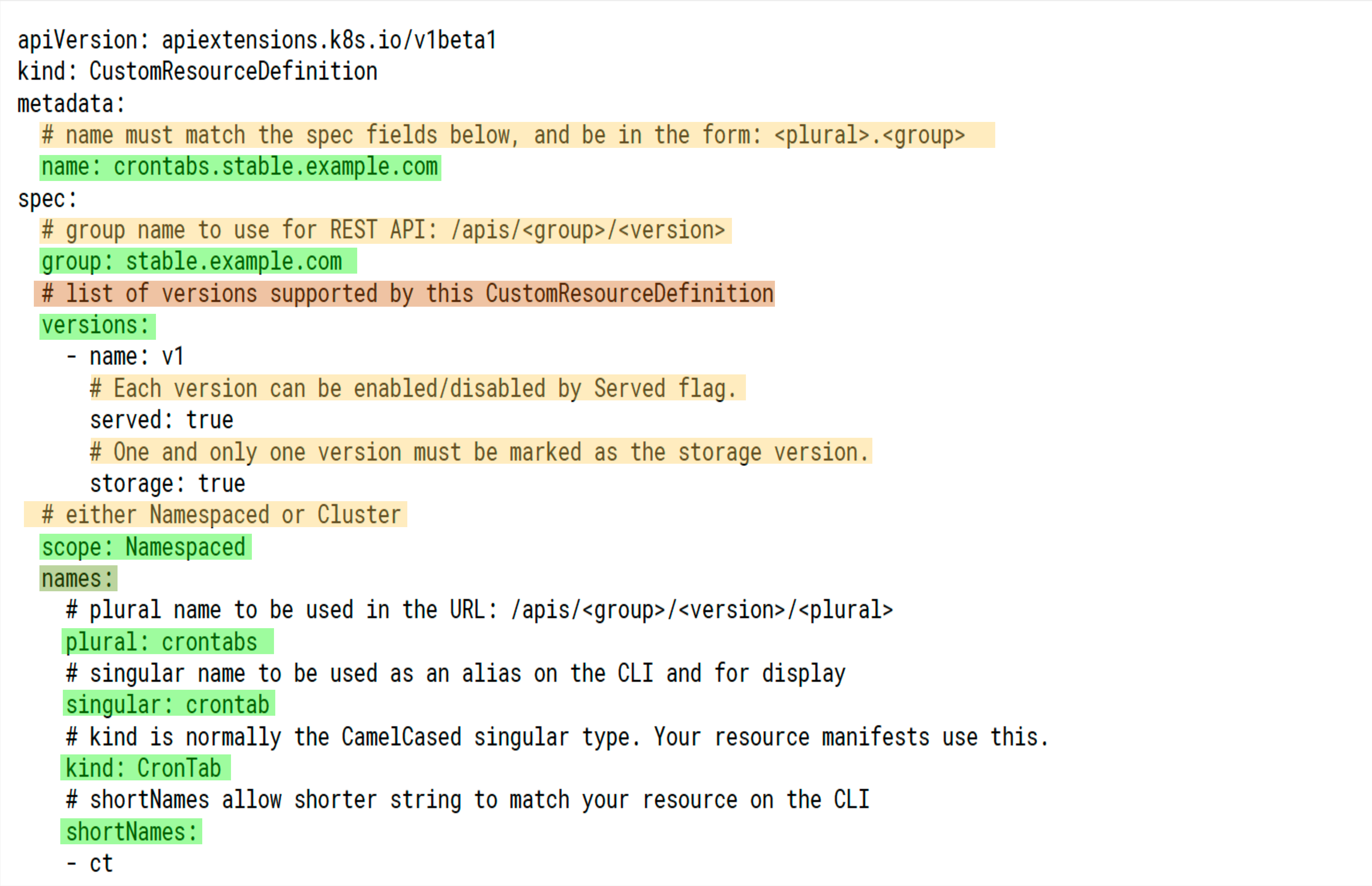
Custom Resource Definition

" Operator is a K8s controller - specific to operating an application. It can automate tasks that usually require a human… "

"CRD are new Resources, like Pods, Deployments, Secrets that you can create. They are managed through the K8s API the same way as official resources"

```
$ kubectl get crd
NAME                                                              AGE
appservices.app.example.com                                                 2d
kafkaconnects.kafka.strimzi.io                                            35d
kafkaconnects2is.kafka.strimzi.io                                          35d
kafkamirrormakers.kafka.strimzi.io                                          35d
kafkas.kafka.strimzi.io                                               35d
kafkausers.kafka.strimzi.io                                             35d
openshiftwebconsoleconfigs.webconsole.operator.openshift.io          40d
servicecertsigneroperatorconfigs.servicecertsigner.config.openshift.io   40d
...
```

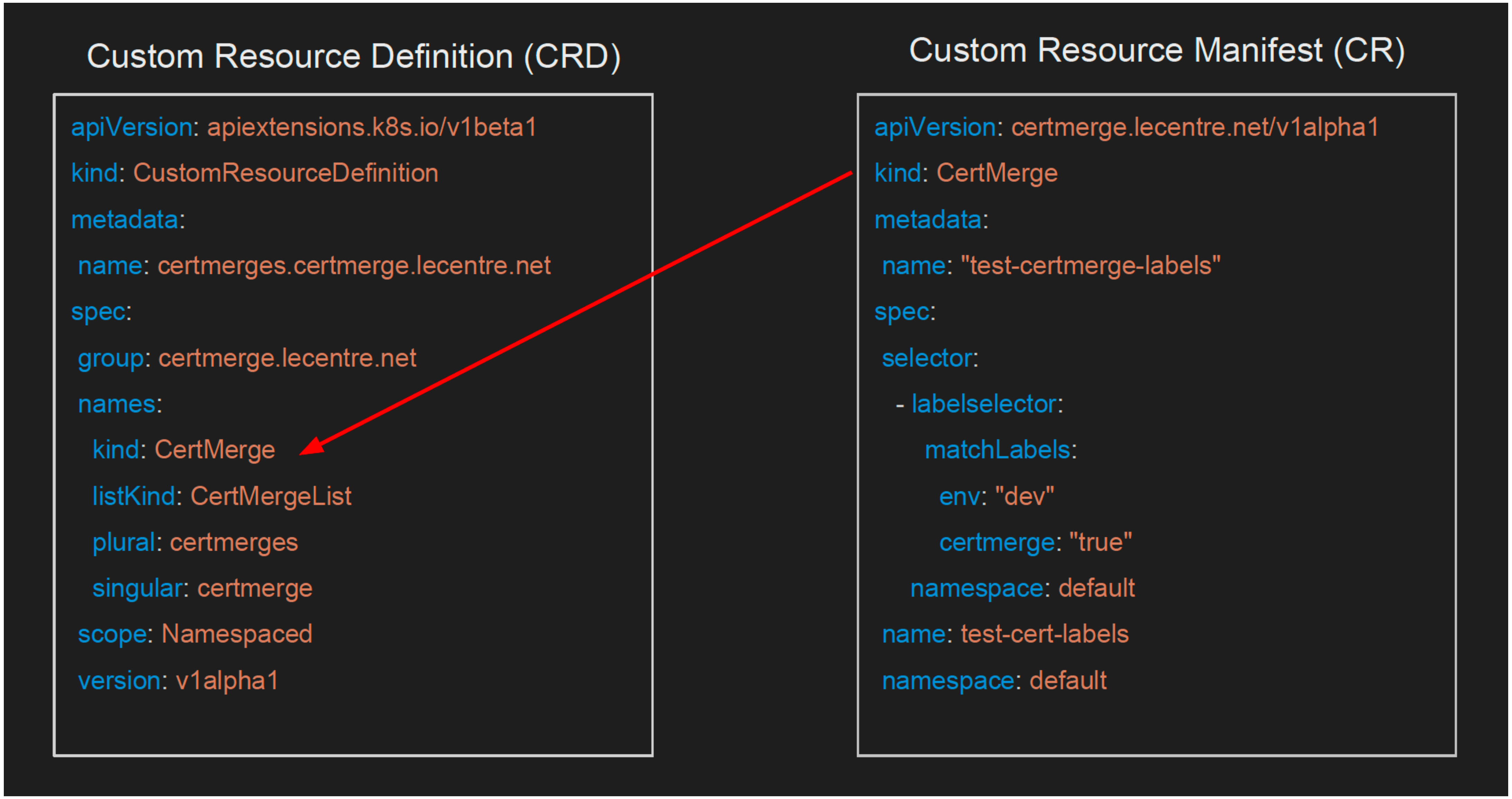
CRD - custom resource definition: **Roles**



```yaml
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  # name must match the spec fields below, and be in the form: <plural>.<group>
  name: crontabs.stable.example.com
spec:
  # group name to use for REST API: /apis/<group>/<version>
  group: stable.example.com
  # list of versions supported by this CustomResourceDefinition
  versions:
    - name: v1
      # Each version can be enabled/disabled by Served flag.
      served: true
      # One and only one version must be marked as the storage version.
      storage: true
  # either Namespaced or Cluster
  scope: Namespaced
  names:
    # plural name to be used in the URL: /apis/<group>/<version>/<plural>
    plural: crontabs
    # singular name to be used as an alias on the CLI and for display
    singular: crontab
    # kind is normally the CamelCased singular type. Your resource manifests use this.
    kind: CronTab
    # shortNames allow shorter string to match your resource on the CLI
    shortNames:
    - ct
```

Custom Resource Definition: Example

> *" Operator is a K8s controller - specific to operating an application. It can automate tasks that usually require a human… "*



Custom Resource Definition (CRD)

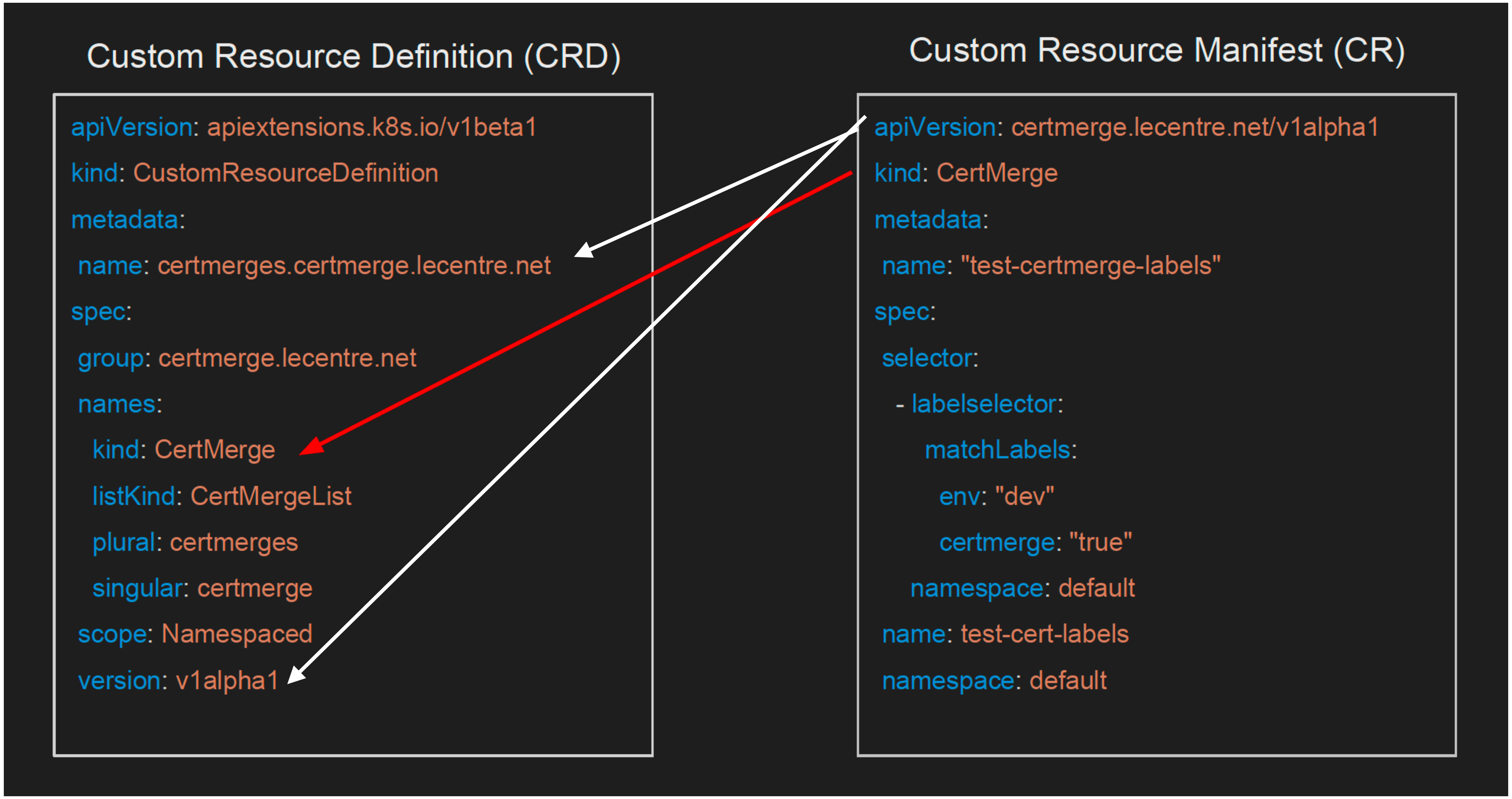
```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
 name: certmerges.certmerge.lecentre.net
spec:
 group: certmerge.lecentre.net
 names:
   kind: CertMerge
   listKind: CertMergeList
   plural: certmerges
   singular: certmerge
 scope: Namespaced
 version: v1alpha1
```

Custom Resource Manifest (CR)

```
apiVersion: certmerge.lecentre.net/v1alpha1
kind: CertMerge
metadata:
 name: "test-certmerge-labels"
spec:
 selector:
  - labelselector:
      matchLabels:
        env: "dev"
        certmerge: "true"
     namespace: default
 name: test-cert-labels
 namespace: default
```

Custom Resource Definition: Example

> *“ Operator is a K8s controller - specific to operating an application. It can automate tasks that usually require a human… ”*



Custom Resource Definition (CRD)

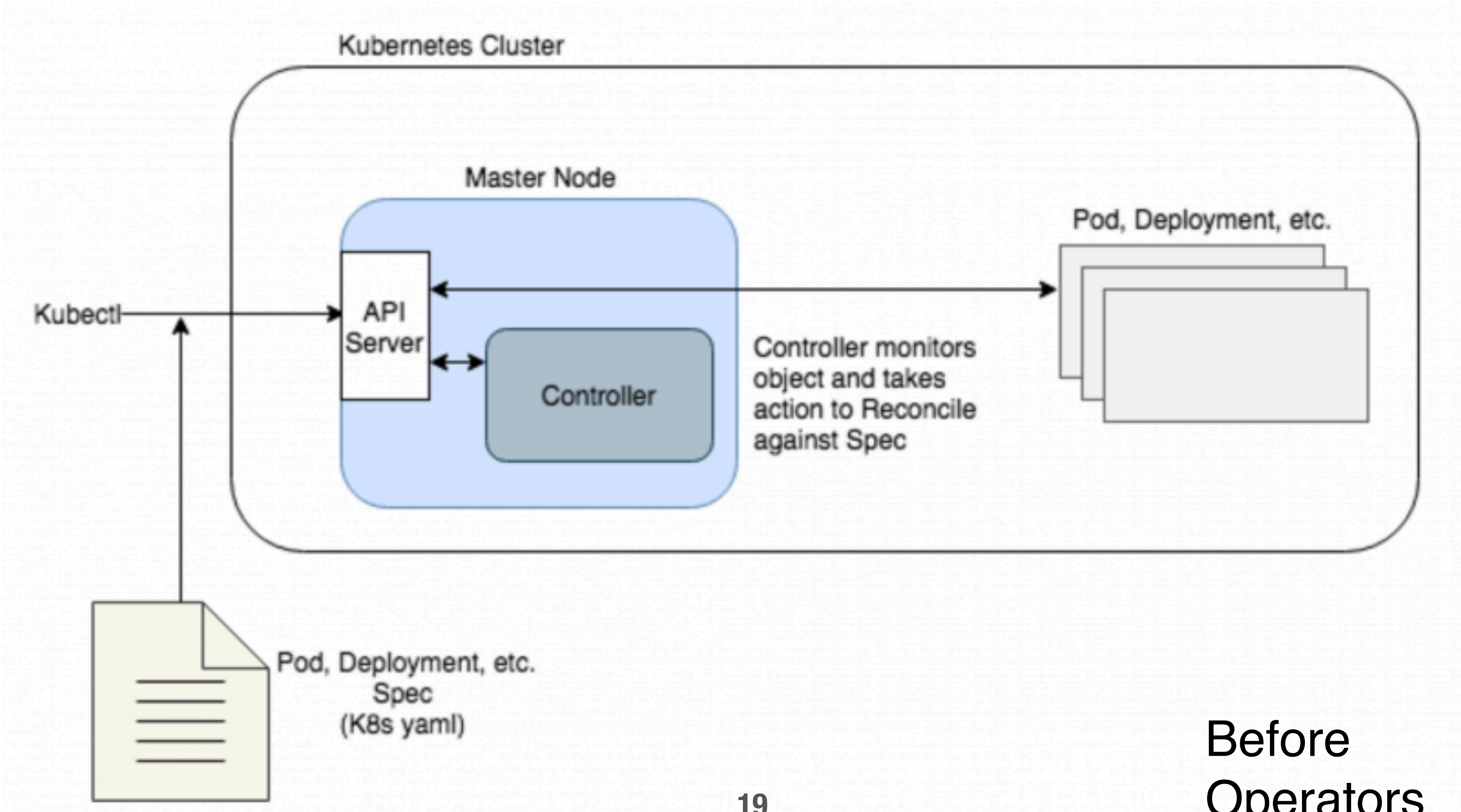
```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
 name: certmerges.certmerge.lecentre.net
spec:
 group: certmerge.lecentre.net
 names:
   kind: CertMerge
   listKind: CertMergeList
   plural: certmerges
   singular: certmerge
 scope: Namespaced
 version: v1alpha1
```

Custom Resource Manifest (CR)

```
apiVersion: certmerge.lecentre.net/v1alpha1
kind: CertMerge
metadata:
 name: "test-certmerge-labels"
spec:
 selector:
  - labelselector:
      matchLabels:
        env: "dev"
        certmerge: "true"
     namespace: default
 name: test-cert-labels
 namespace: default
```

Kubernetes Operators

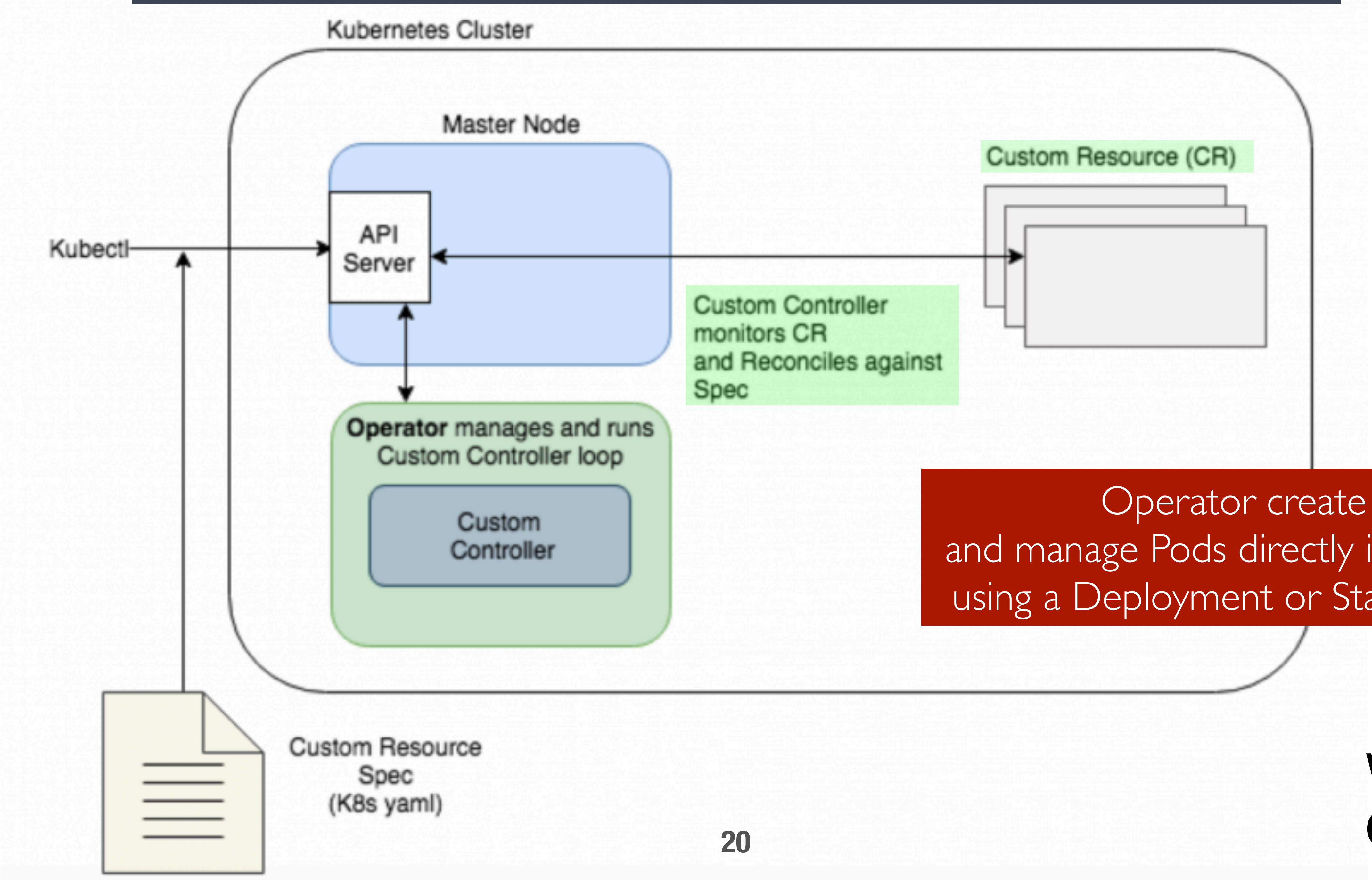
> *“ Operator is a K8s controller - specific to operating an application. It can automate tasks that usually require a human… ”*


Kubernetes Cluster
Master Node
Pod, Deployment, etc.
Kubectl
API Server
Controller
Controller monitors object and takes action to Reconcile against Spec
Pod, Deployment, etc. Spec (K8s yaml)

Before Operators

Kubernetes Operators

> *“ Operator is a K8s controller - specific to operating an application. It can automate tasks that usually require a human… ”*


Kubernetes Cluster
Master Node
Kubectl
API Server
Custom Resource (CR)
Custom Controller monitors CR and Reconciles against Spec
Operator manages and runs Custom Controller loop
Custom Controller
Custom Resource Spec (K8s yaml)

Operator create and manage Pods directly instead of using a Deployment or StatefulSets

With Operators

Controller Operator Spectrum

> *“ Operator is a K8s controller - specific to operating an application. It can automate tasks that usually require a human… ”*

When to choose creating an operator?

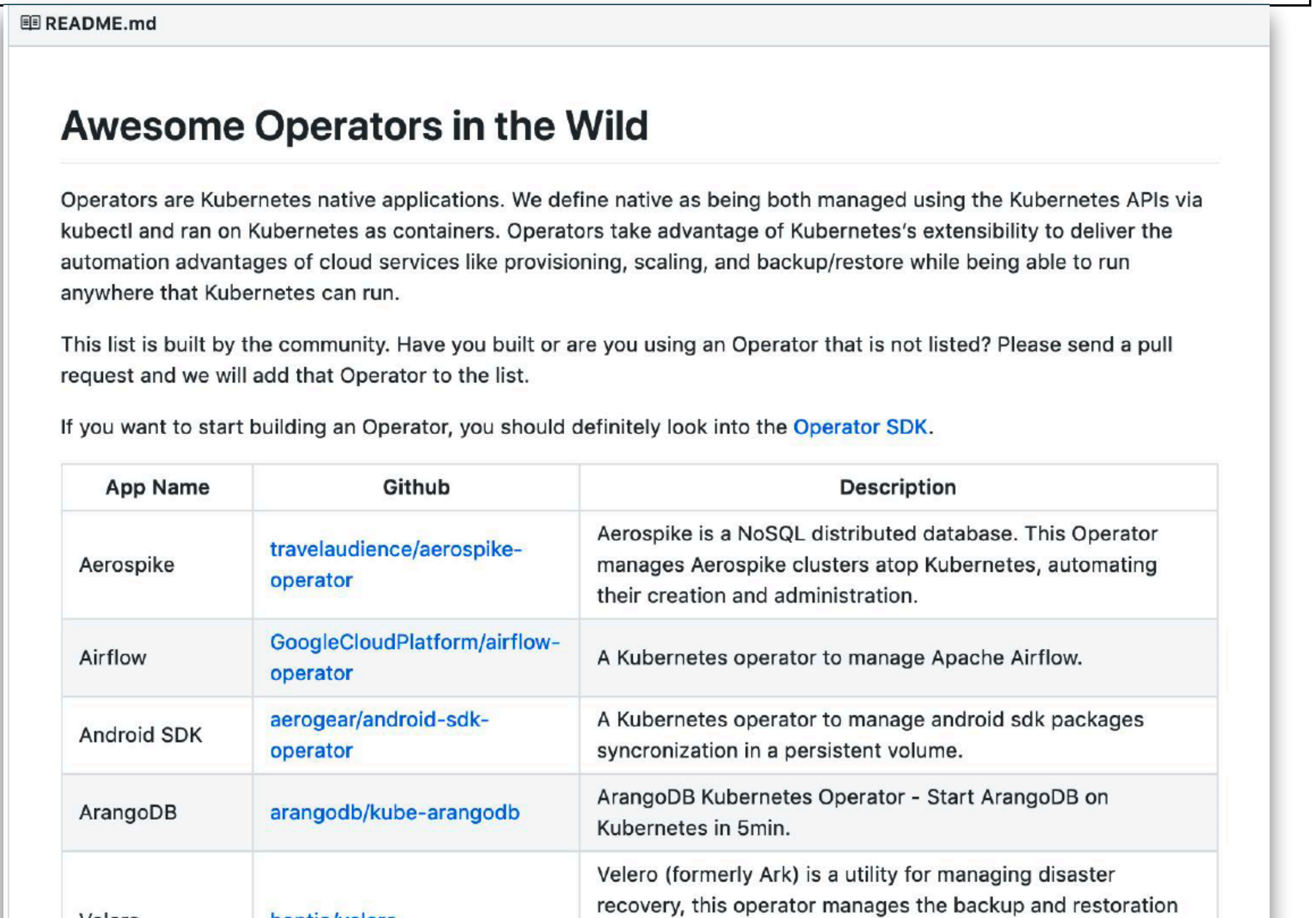
- **Resources** are **scoped** to a namespace or a cluster
- **Encapsulate** business logic
- Build **automation** that **watches** for **updates** of Kubernetes objects
- **Create** or **update resources** via the Kubernetes native API
- Top level support from **kubectl:**
  - *You need to automate some Resource creation*.
  - *You need more intelligence in the management.* ex : the Etcd-Operator create and manage Pods directly instead of using a Deployment or StatefulSets

Kubernetes Operators

> *" Operator is a K8s controller - specific to operating an application. It can automate tasks that usually require a human… "*

Existing Operators ?

(check https://github.com/operator-framework/awesome-operators)



README.md

Awesome Operators in the Wild

Operators are Kubernetes native applications. We define native as being both managed using the Kubernetes APIs via kubectl and ran on Kubernetes as containers. Operators take advantage of Kubernetes's extensibility to deliver the automation advantages of cloud services like provisioning, scaling, and backup/restore while being able to run anywhere that Kubernetes can run.

This list is built by the community. Have you built or are you using an Operator that is not listed? Please send a pull request and we will add that Operator to the list.

If you want to start building an Operator, you should definitely look into the Operator SDK.

App Name	Github	Description
Aerospike	travelaudience/aerospike-operator	Aerospike is a NoSQL distributed database. This Operator manages Aerospike clusters atop Kubernetes, automating their creation and administration.
Airflow	GoogleCloudPlatform/airflow-operator	A Kubernetes operator to manage Apache Airflow.
Android SDK	aerogear/android-sdk-operator	A Kubernetes operator to manage android sdk packages syncronization in a persistent volume.
ArangoDB	arangodb/kube-arangodb	ArangoDB Kubernetes Operator - Start ArangoDB on Kubernetes in 5min.
Velero	heptio/velero	Velero (formerly Ark) is a utility for managing disaster recovery, this operator manages the backup and restoration

Kubernetes Operators

> *" Operator is a K8s controller - specific to operating an application. It can automate tasks that usually require a human… "*

Writing Operators ?

(check https://github.com/operator-framework/awesome-operators)

"…**writing** such **operators** can be **very difficult** because of **challenges** such as using (i) low level APIs and (ii) a lack of modularity which leads to duplication, inconsistencies, and **unexpected behaviors**".

Operator SDK

github.com/operator-framework/operator-sdk

In ***GitHub***, the Operator SDK is a very **active project, with:**

- over 200 developers that forked its contributions on GitHub, most of them playing/extending some simple examples of operators;
- around 1000 commits done since the first release (5 Apr 2018).
- over 10 releases produced in less than a year.
- an increasing number of **GitHub issues opened** by other contributors in recent months
  - [some of them were opened also by us, see **https://github.com/operator-framework/operator-sdk/issues/651 and https://github.com/operator-framework/operator-sdk/issues/927**]
- with over 1200 stars it is one of the most *popular/trending Go projects on Github* https://github.com/trending/go?since=monthly

Its ***project Status is still “pre-alpha“***, which means that “***are expected breaking changes to the API in the upcoming releases***“.

https://vshn.ch/en/blog/introduction-to-kubernetes-operators-operator-framework-and-operators-sdk/

Operator SDK

github.com/operator-framework/operator-sdk

- Framework and Tookit for **creating** Operators
  - Part of the Operator Framework ————>

OPERATOR
FRAMEWORK

- Tools for **scaffolding** and **code generation** to bootstrap a new project fast

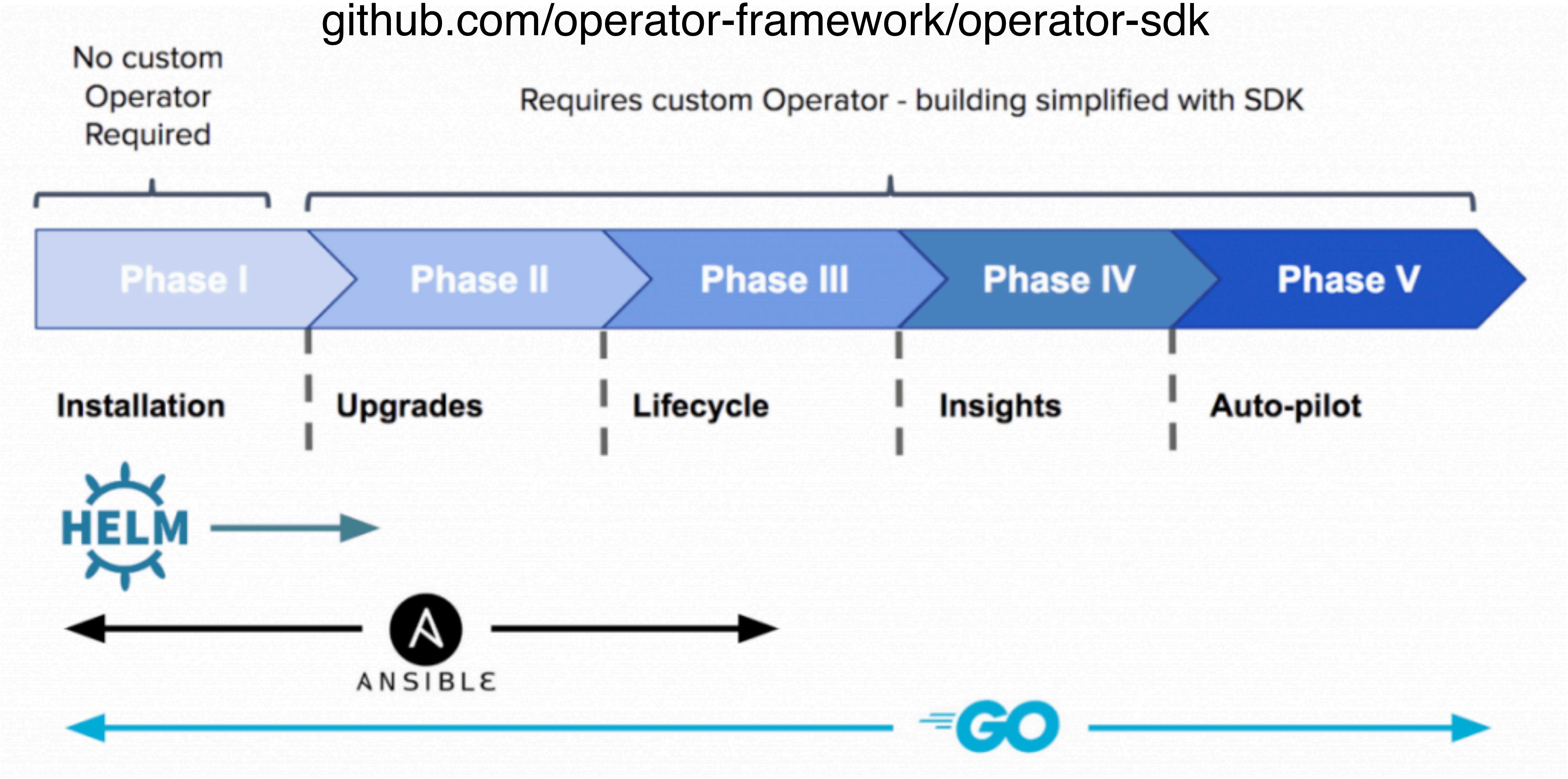
- Extensions to cover **common** operator **use cases**

- **Uses controller-runtime** under the hood
  - https://github.com/kubernetes-sigs/controller-runtime

- **Operator types** you can create:
  - *Golang*
  - *Ansible*
  - *Helm*

Operator SDK

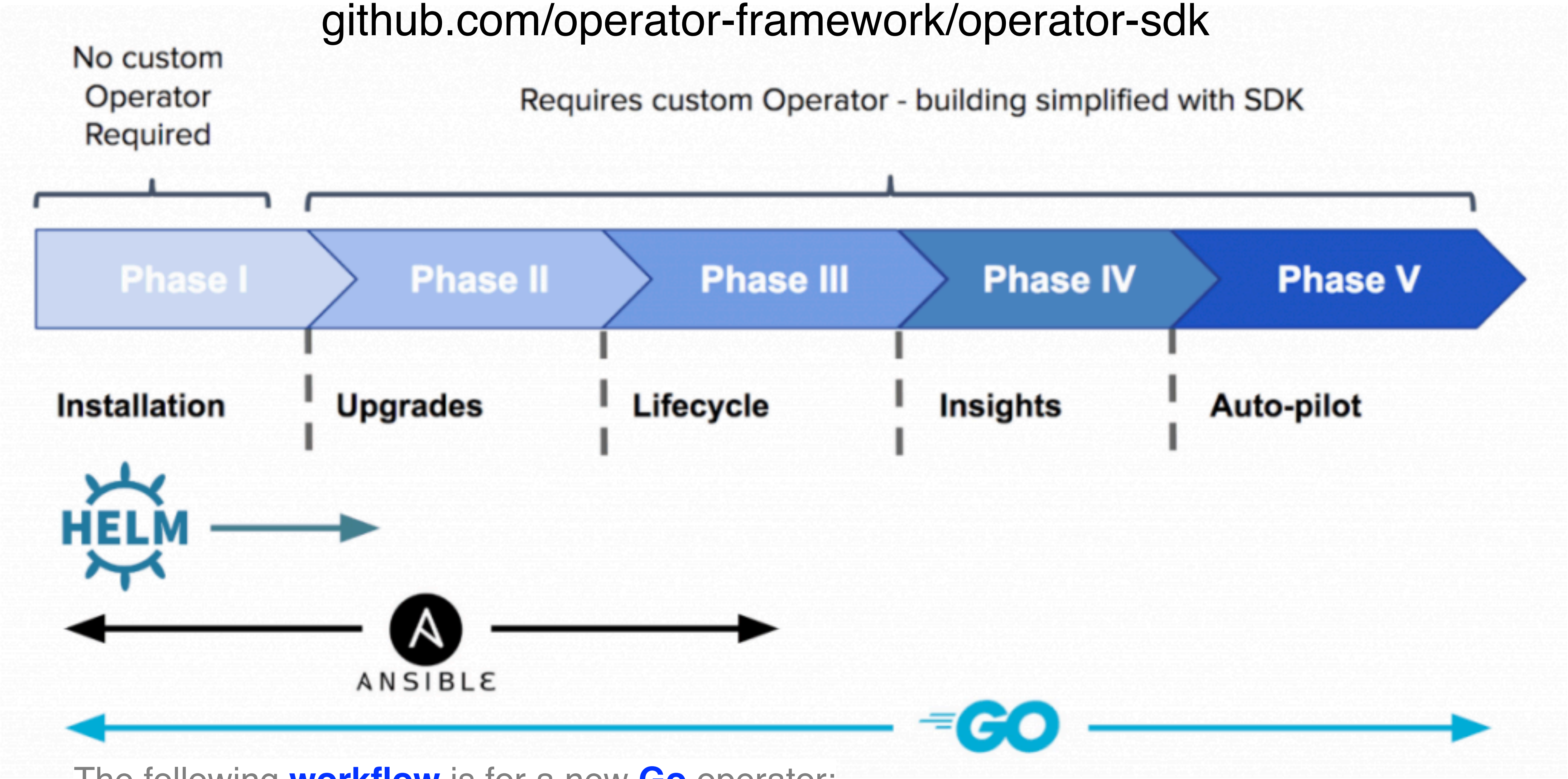
github.com/operator-framework/operator-sdk



- **Operator types** you can create:
  - *Golang*
  - *Ansible*
  - *Helm*

Operator SDK

github.com/operator-framework/operator-sdk



The following **workflow** is for a new **Go** operator:

1. **Create** a new **operator project** using the SDK Command Line Interface (CLI)
2. **Define** new **resource** APIs by adding Custom Resource Definitions (**CRD**)
3. **Define Controllers** to watch and reconcile resources
4. **Write** the **reconciling logic** for your Controller using the SDK and controller-runtime APIs
5. Use the SDK CLI to **build** and **generate** the **operator deployment manifests**

Outline

- **Quick Introduction - Kubernetes and Operators SDK**
    - Current Vision, Challenges, and Opportunities
- **Tutorial on Kubernetes Operators:**
    - Operators SDK
        - Example 1: Simple Operator Example
        - Example 2: Advanced Operator Example
- **Prometheus**

Prerequisites

https://github.com/spanichella/operator-sdk-examples-1/blob/master/Documentation/1_Prerequisites.pdf

(Short link: **https://bit.ly/2FusHkf**)

- git clone https://github.com/spanichella/operator-sdk-examples

Other Prerequisites

- dep version v0.5.0+
    - https://golang.github.io/dep/docs/installation.html
- git
- go version v1.12+.
    - https://golang.org/dl/. Or https://nats.io/documentation/tutorials/go-install/
    - Set $GOPATH\
- docker version 17.03+.
    - https://docs.docker.com/install/

- **Minishift installed (or access to a Kubernetes v1.11.3+ cluster):**
    - **https://github.com/minishift/minishift**
    - **kubectl version v1.11.3+.**

1) Simple Operator Example:

BEFORE STARTING:

1) Install required tools: (Short link: **https://bit.ly/2FusHkf**)

2) Install Operator SDK (Short link: **https://bit.ly/2IDOZSI**)

For this first example of operator we provide a short and fast way to deploy it:

[LONG VERSION]: full steps to deploy the operator

[SHORT VERSION]: few lines and you will deploy the operator. **Minishift needed**.

(Short link: **https://bit.ly/2FtiHb5**)

Install Operator SDK

```
$ mkdir -p $GOPATH/src/github.com/operator-framework

$ cd $GOPATH/src/github.com/operator-framework

$ git clone https://github.com/operator-framework/operator-sdk # This will
download the git repository and not install it
# alternative command to the aforementioned one $ go get -d github.com/
operator-framework/operator-sdk

$ cd $GOPATH/src/github.com/operator-framework/operator-sdk
$ git checkout  33b3bfe
$ make dep
$ make install # in case of problem “brew uninstall --ignore-dependencies go”

$ operator-sdk version
 operator-sdk version: v0.8.0-28-ged16656,
```

(Short link: **https://bit.ly/2IDOZSI**)

1) Simple Operator Example:

BEFORE STARTING:

1) Install required tools: (Short link: **https://bit.ly/2FusHkf**)

2) Install Operator SDK (Short link: **https://bit.ly/2IDOZSI**)

For this first example of operator we provide a short and fast way to deploy it:

[LONG VERSION]: full steps to deploy the operator

[SHORT VERSION]: few lines and you will deploy the operator. **Minishift needed**.

(Short link: **https://bit.ly/2FtiHb5**)

The resulting automatically generated **GO** operator will present the following reference **Structure**:

File/Folders	**Purpose**
cmd	Contains **manager/main.go** which is the **main program of the operator**. This instantiates a new **manager** which **registers** all **custom resource definitions** under **pkg/apis/...** and starts all **controllers** under **pkg/controllers/...** .
pkg/apis	**Contains** the directory tree that defines the APIs of the **Custom Resource Definitions**(**CRD**). **Users are expected to edit the** **pkg/apis/<group>/<version>/<kind>_types.go** files to define the API **for each resource** type and import these packages in their controllers **to watch for these resource types**.
pkg/controller	This pkg contains the controller implementations. **Users are expected to edit the** **pkg/controller/<kind>/<kind>_controller.go** to define the controller's reconcile logic for handling a resource type of the specified **kind**.
build	Contains the **Dockerfile** and build scripts used to **build** the **operator**.
deploy	Contains various **YAML manifests for registering CRDs**, setting up **RBAC**, and deploying the operator as a Deployment.
(Gopkg.toml Gopkg.lock) or (go.mod go.sum)	The **Go mod** or **Go Dep** manifests that **describe** the external **dependencies of this operator**, depending on the dependency manager chosen when initializing or migrating a project.
vendor	The golang **vendor** directory that **contains** local **copies** of external **dependencies** that satisfy Go imports in this project. **Go Dep/Go modules** manages the vendor directly. If using modules, this directory will not exist unless the project is initialized with the --vendor flag, or go mod vendor is run in the project root.

2) More Advanced Operator Example:

BEFORE STARTING:

1) Install required tools: (Short link: **https://bit.ly/2FusHkf**)

2) Install Operator SDK (Short link: **https://bit.ly/2IDOZSI**)

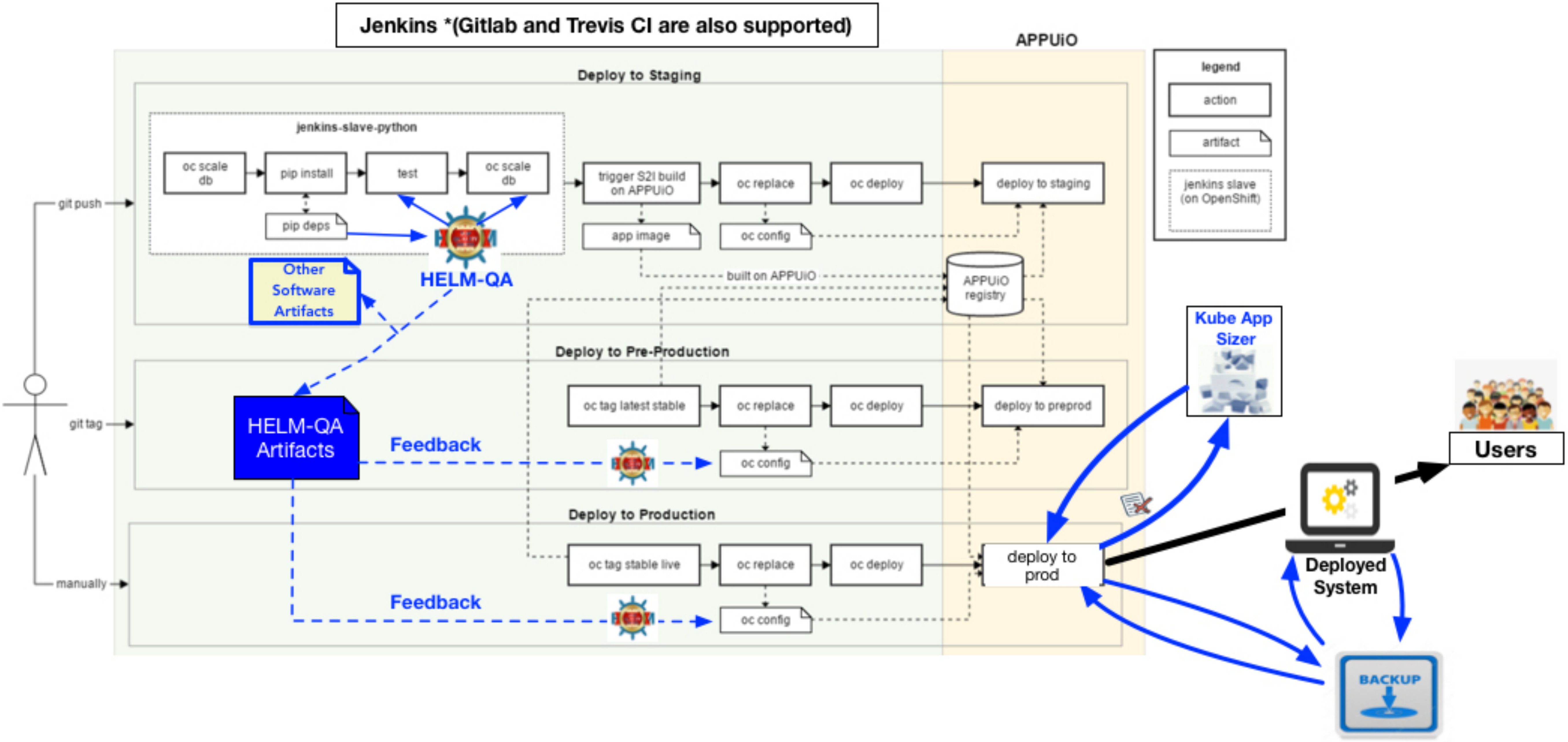
For this first example of operator we provide a short and fast way to deploy it:

[LONG VERSION]: full steps to deploy the operator

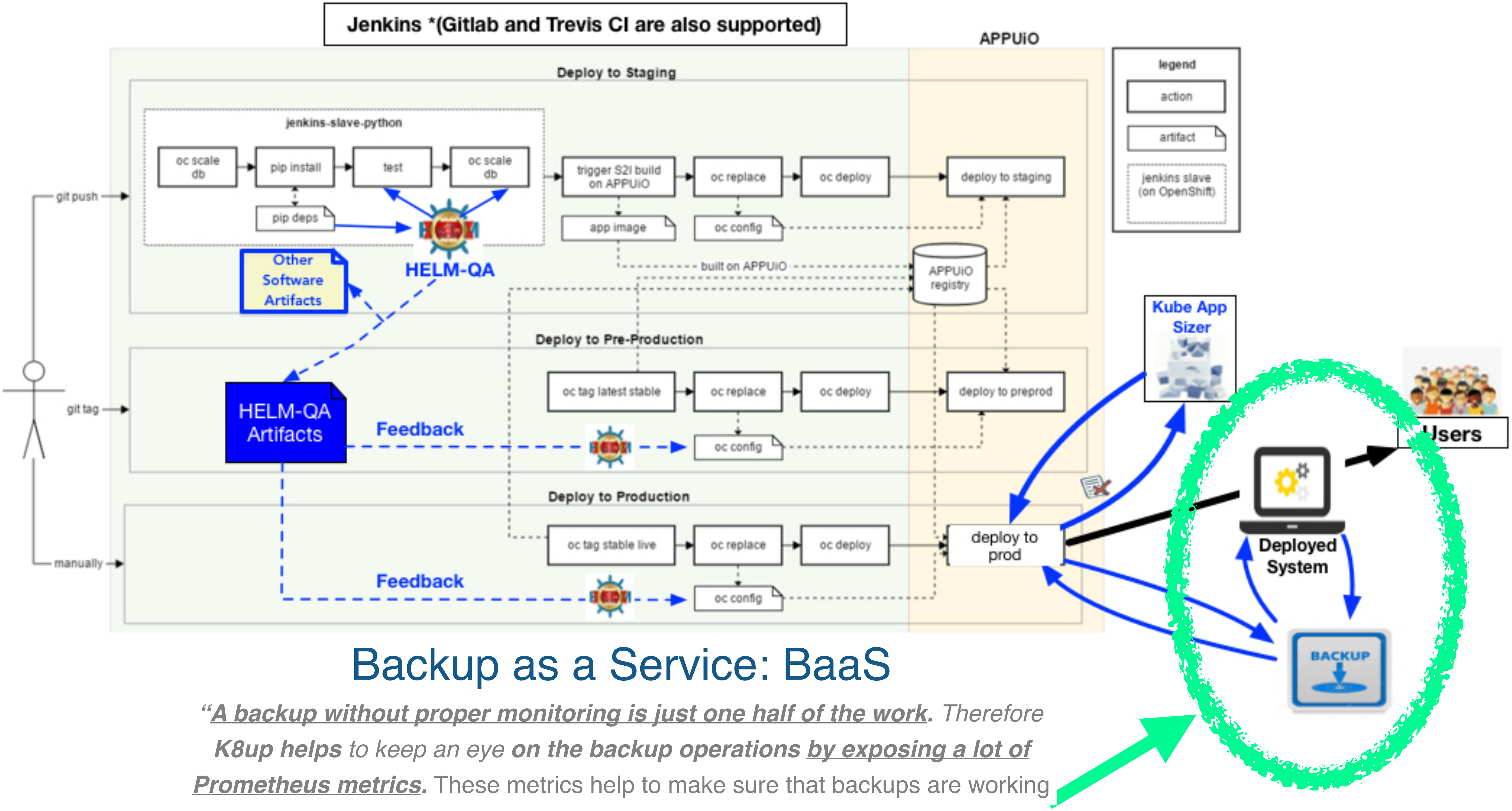
[SHORT VERSION]: few lines and you will deploy the operator. **Minishift needed**.

(Short link: **https://bit.ly/2xc8jjk**)

Our Vision: Summary



Our Vision: Summary



Backup as a Service: BaaS

*“**A backup without proper monitoring is just one half of the work**. Therefore **K8up helps** to keep an eye **on the backup operations by exposing a lot of Prometheus metrics**.* These metrics help to make sure that backups are working and are executed on the right time without errors”.

https://vshn.ch/en/k8up-backup-operator-for-kubernetes/

Outline

- **Quick Introduction - Kubernetes and Operators SDK**
    - Current Vision, Challenges, and Opportunities
- **Tutorial on Kubernetes Operators:**
    - Operators SDK
        - Example 1: Simple Operator Example
        - Example 2: Advanced Operator Example
- **Prometheus**

Thanks for the Attention!

“Kubernetes, Operators SDK, And Prometheus”

- Any Questions?

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering

Dr. Sebastiano Panichella
Zurich University of
Applied Science (ZHAW)
https://spanichella.github.io/

DENSO
Crafting the Core

Dr. Soheila Dehghanzadeh
Denso Automotive Deutschland
https://www.linkedin.com/in/soheiladehghanzadeh/