Kubernetes
Stateless -> Stateful

HELLO!

I AM smalltown (Tom Tsai)

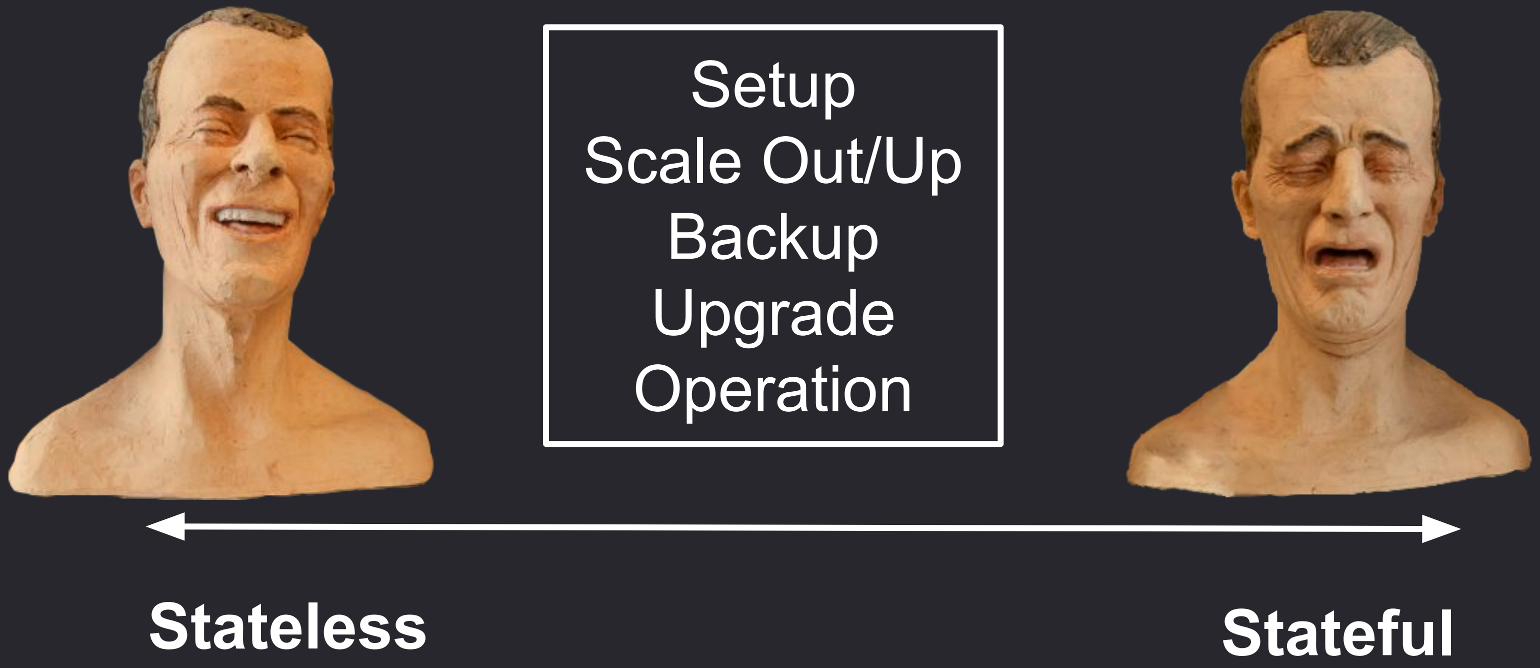
I am here because I want to share something with you

You can find me at smalltown@awsug.tw

Stateful Service

- Replication Setup, e.g. MongoDB, MySQL
- Cluster Management, e.g. Etcd, Kafka, ElasticSearch
- Scale Out, Scale Up
- Upgrade
- Data Backup
- Other Operation Job...

Manage Production Service

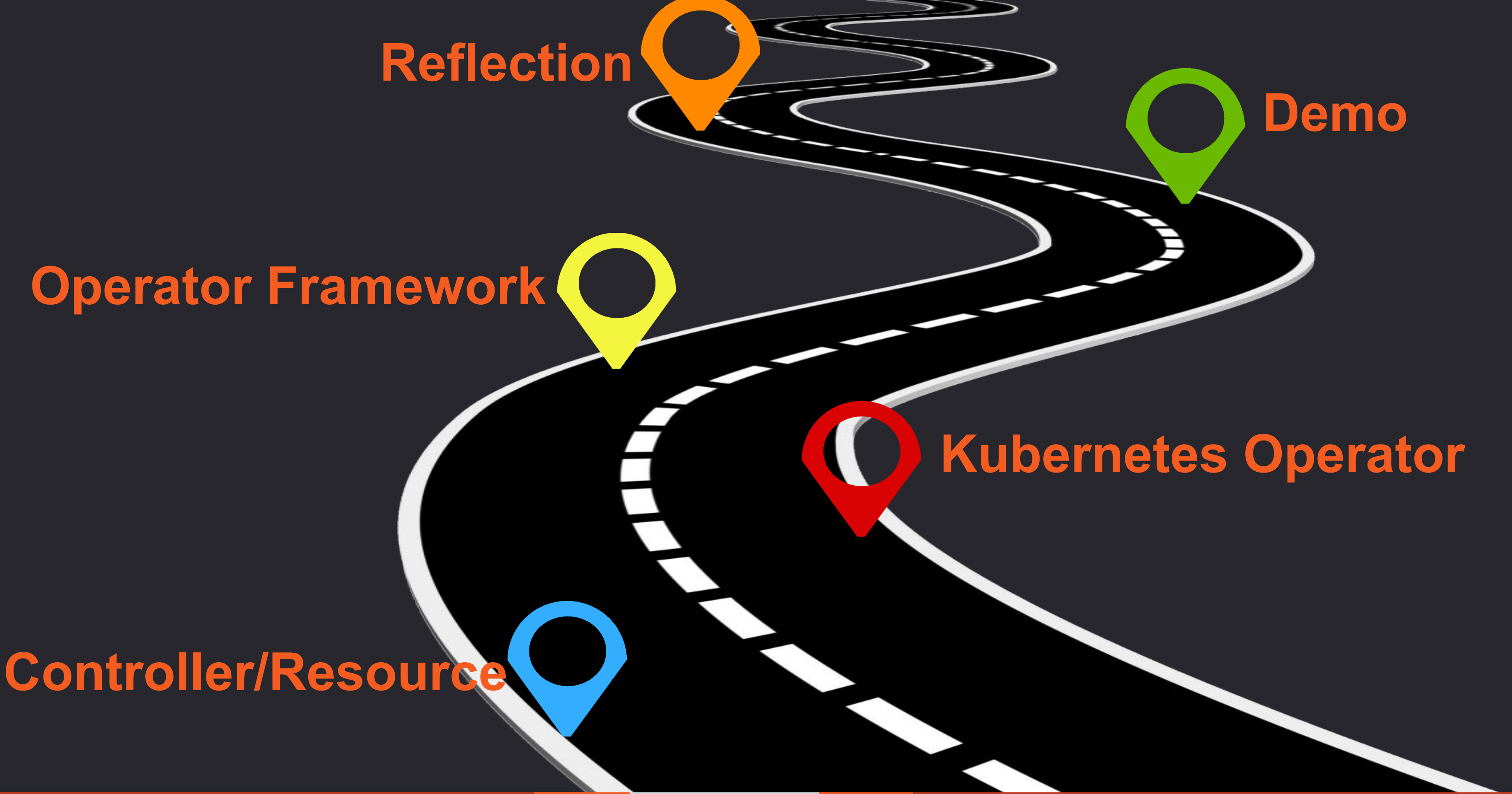


Setup
Scale Out/Up
Backup
Upgrade
Operation

Stateless

Stateful

Kubernetes Operator by CoreOS



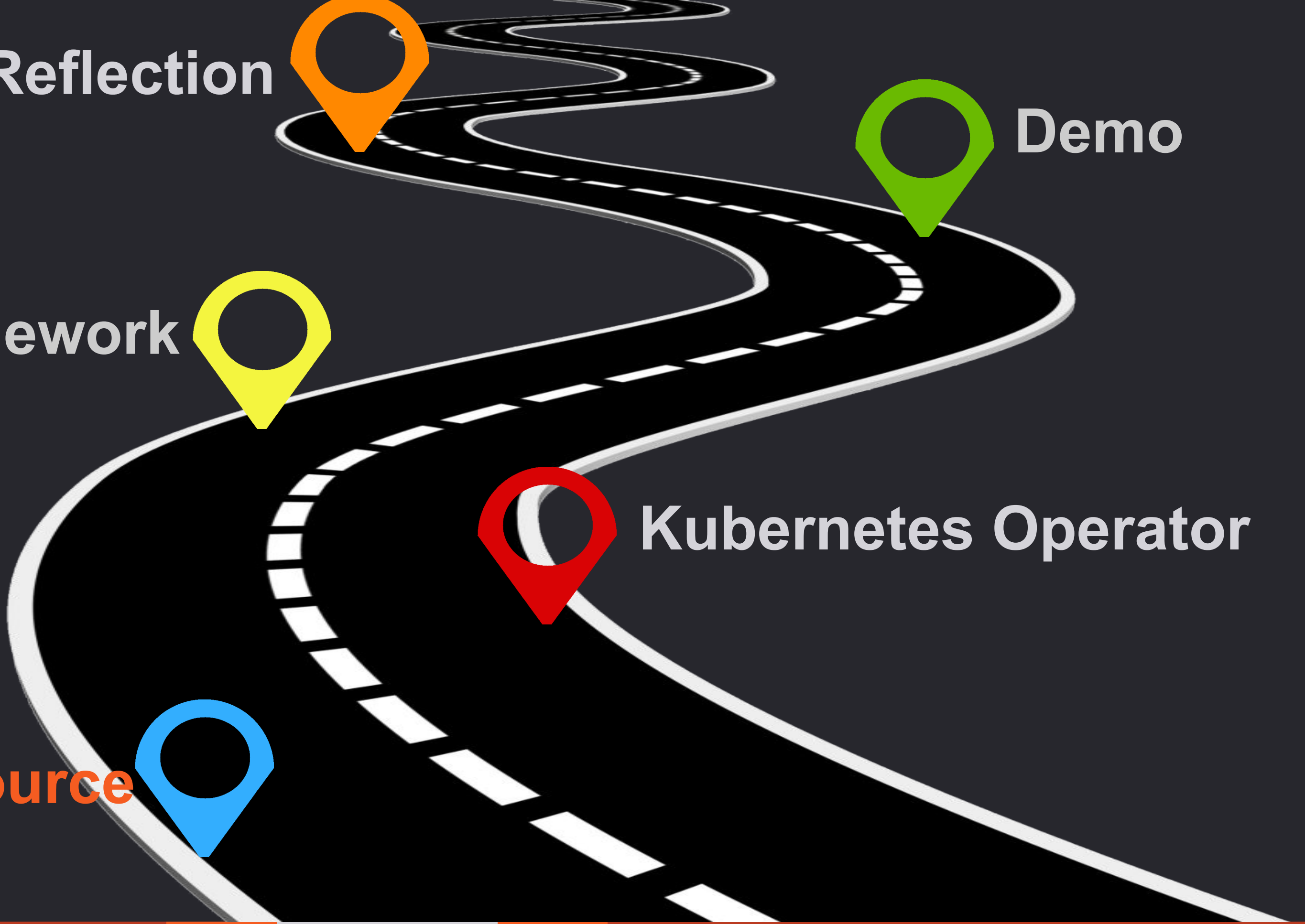
Controller/Resource

Kubernetes Operator

Operator Framework

Demo

Reflection



Controller/Resource

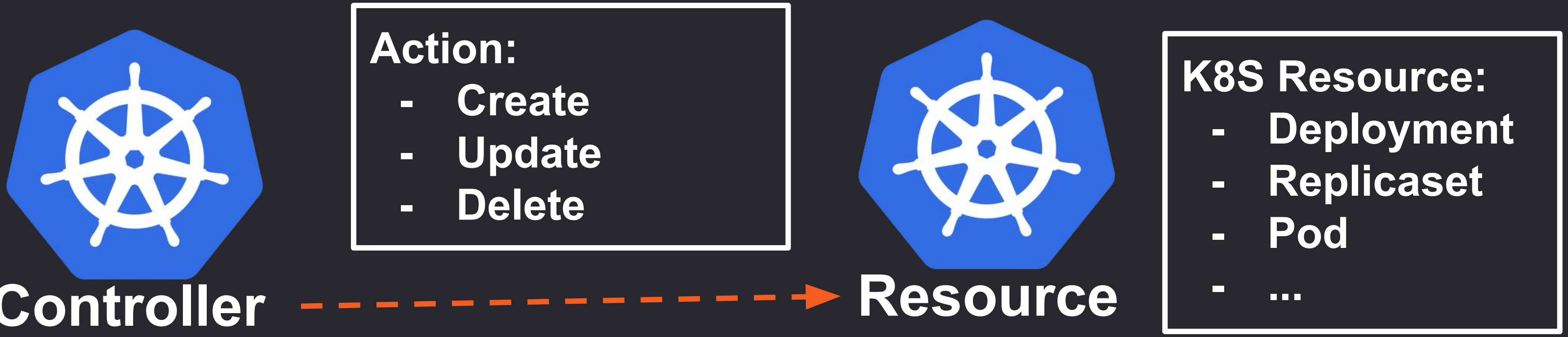
Kubernetes Operator

Operator Framework

Demo

Reflection

Kubernetes Controller & Resource



Action:

- Create
- Update
- Delete

Controller

Resource

K8S Resource:

- Deployment
- Replicaset
- Pod
- ...

Stateless Service

Origin Controller and Resource **Enough** to Handle

- Blue/Green Deployment
- Auto-Healing
- Scale Out, Scale Up
- …

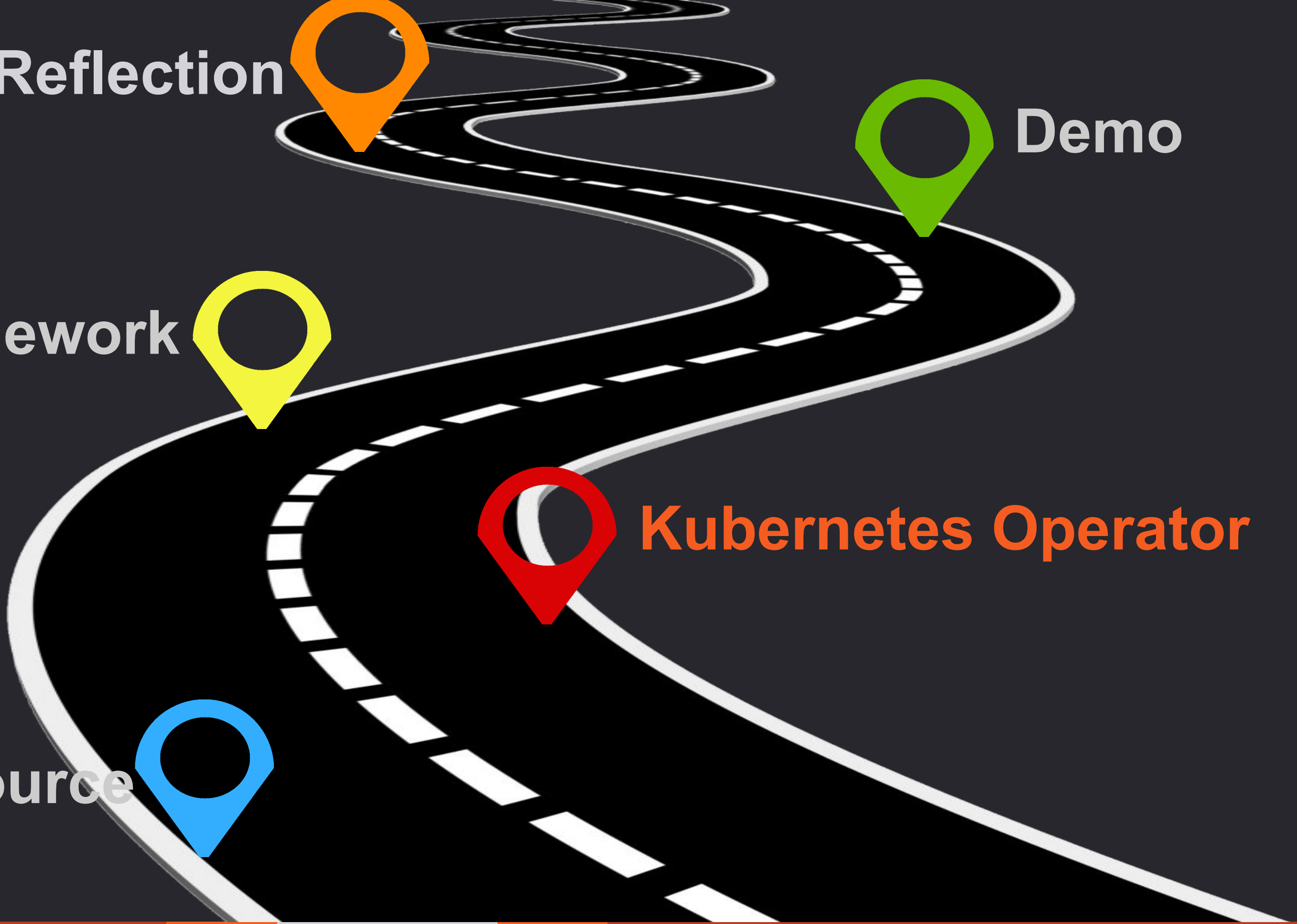
Stateful Service

Example: **Add** One New Etcd Member into Cluster

- **Create** a DNS Name for the New Etcd Member
- **Launch** the New Etcd Instance
- **Use** the Etcd Administrative Tools to Tell the Existing Cluster About this New Member

Custom Controller & Resource

- Apparently, Origin Controller & Resource **Cannot Fulfill**.
- Hence, Kubernetes Support **Custom** Controller & Resource
- It Allows Us to **Extend** Kubernetes



Reflection

Demo

Operator Framework

Kubernetes Operator

Controller/Resource

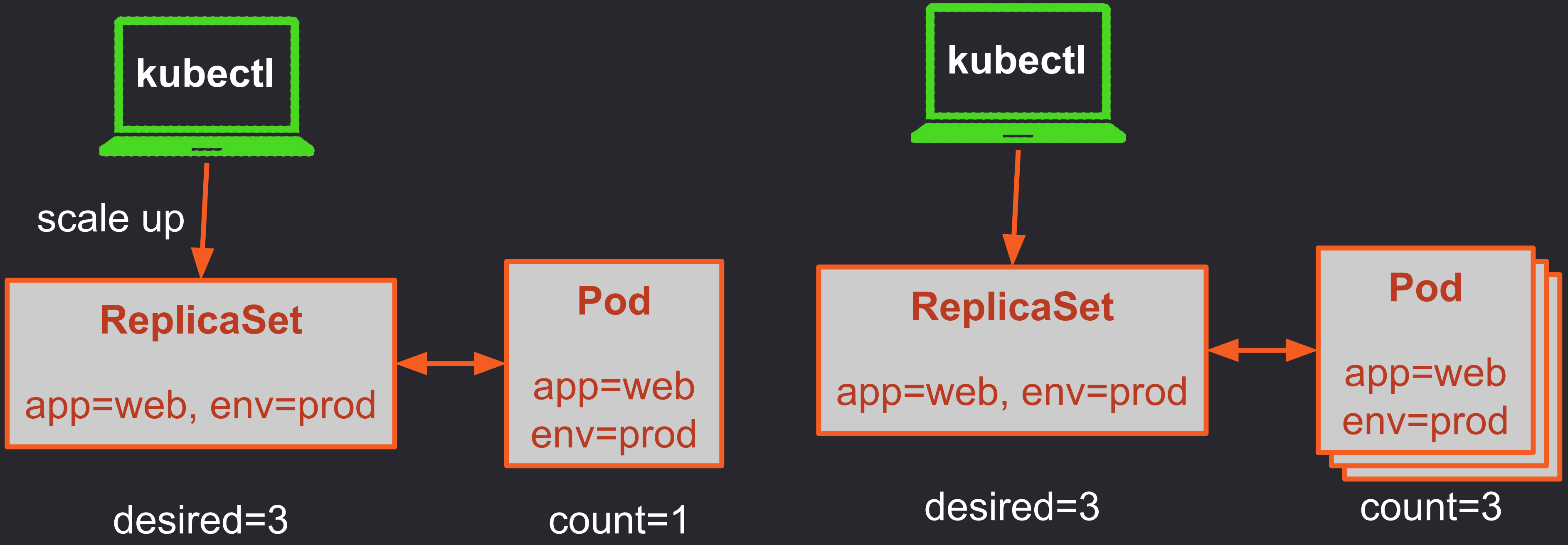
What is Kubernetes Operator?

- An Operator is a **Software** with **Specific Application Knowledge**
- **Extends** the Kubernetes through **Custom Controller/Resource**
- **Enable** Users to **Create**, **Configure**, and **Manage** Applications

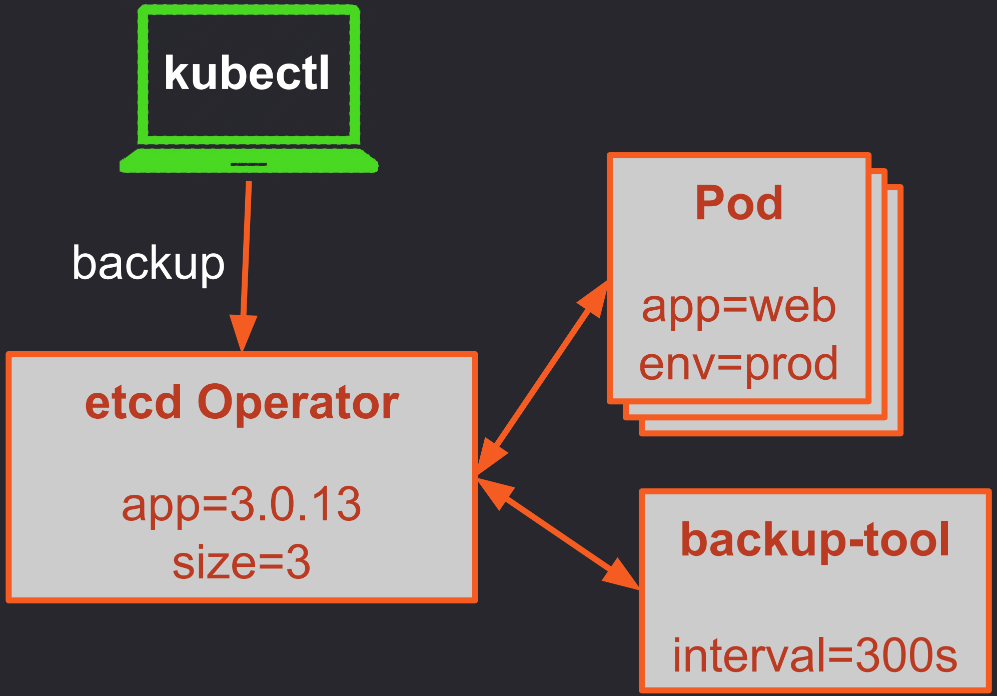
What is Kubernetes Operator?

An Operator is a Domain Specific Controller

Operation of Stateless Service



Operation of Stateful Service (Operator)


kubectl

backup

etcd Operator

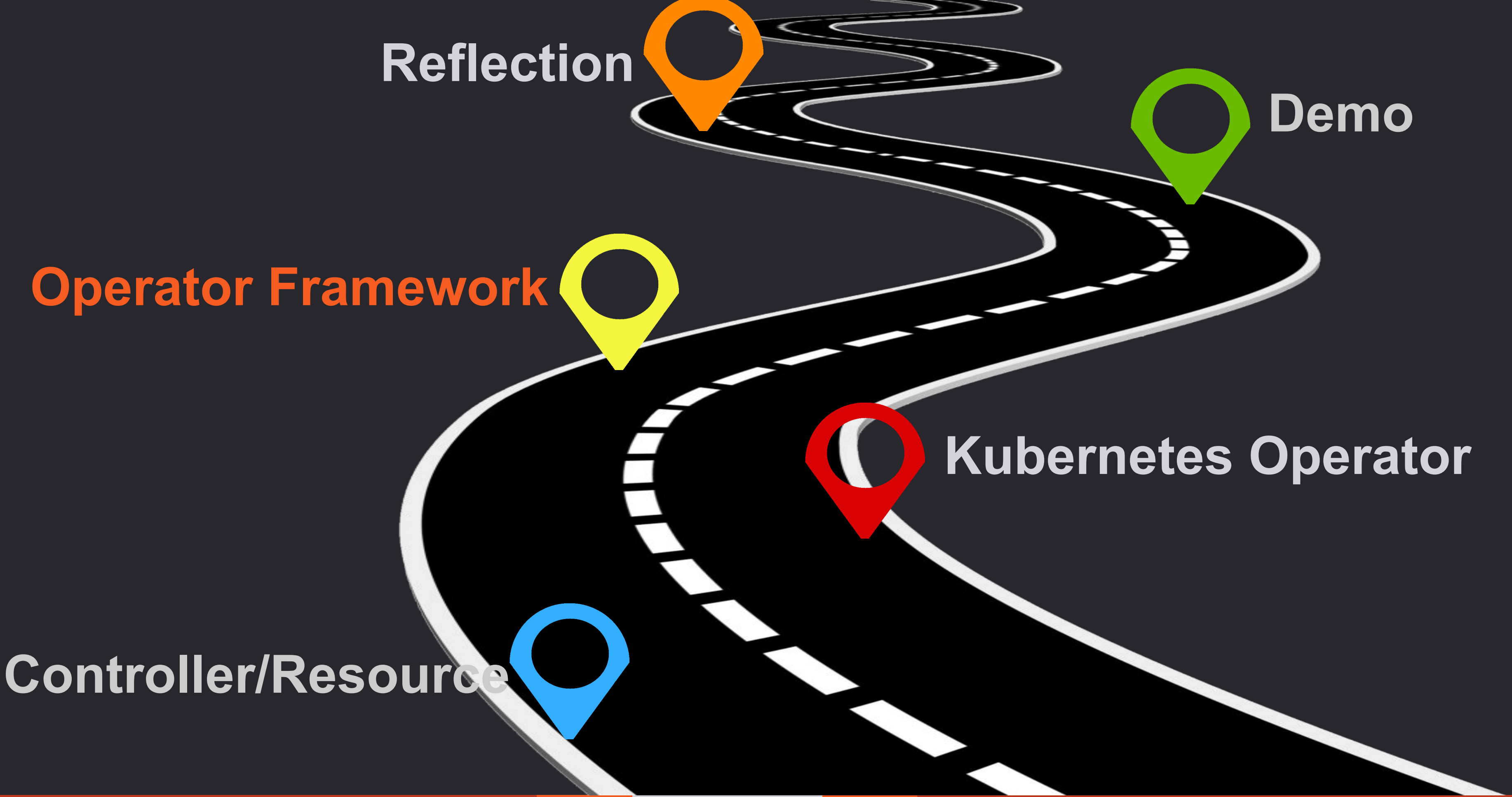
app=3.0.13
size=3

Pod

app=web
env=prod

backup-tool

interval=300s



Controller/Resource

Kubernetes Operator

Operator Framework

Demo

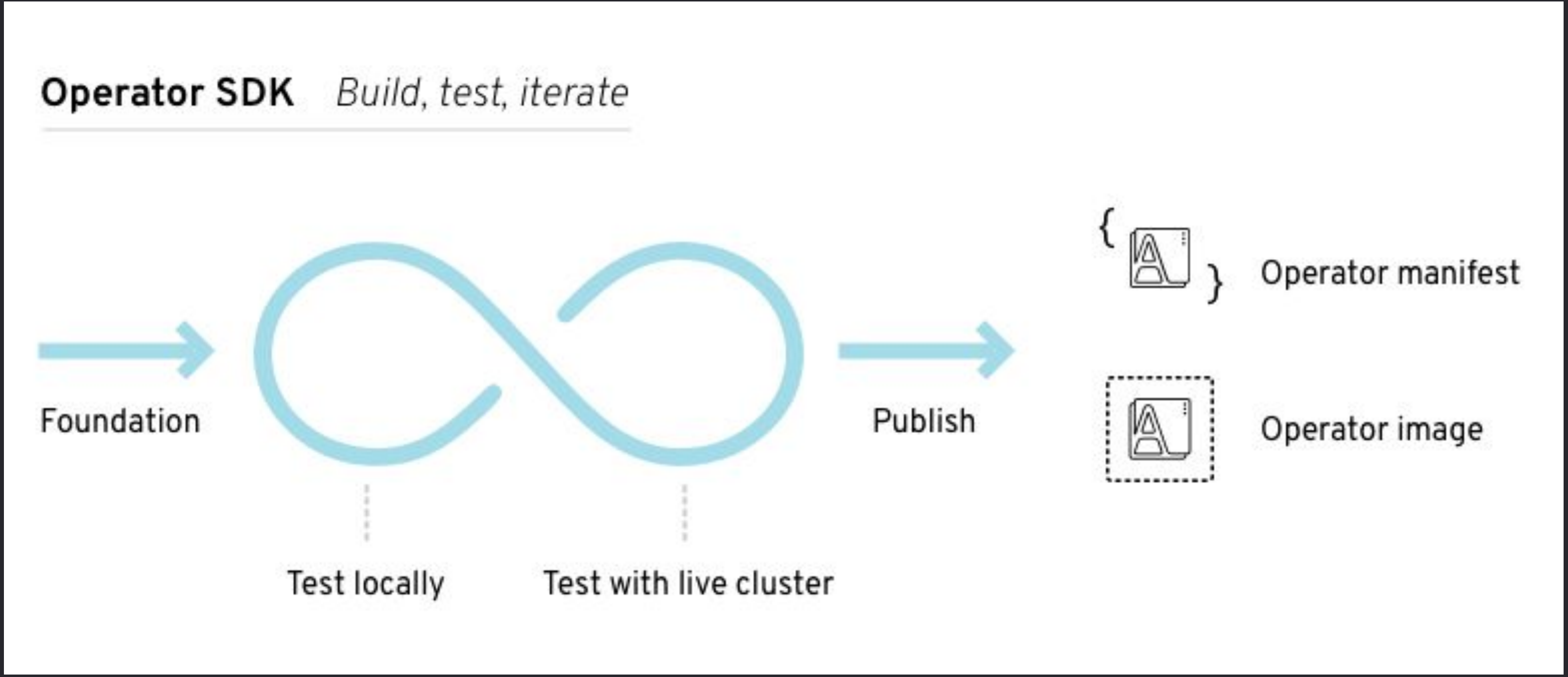
Reflection

Operator Framework

The Concept of **Kubernetes Operator** is Introduced by CoreOS at Nov. 2016, then releasing **Operator Framework** at May 2018

- Operator SDK
- Operator Lifecycle Manager
- Operator Metering

Operator SDK



Operator SDK *Build, test, iterate*

Foundation

Test locally

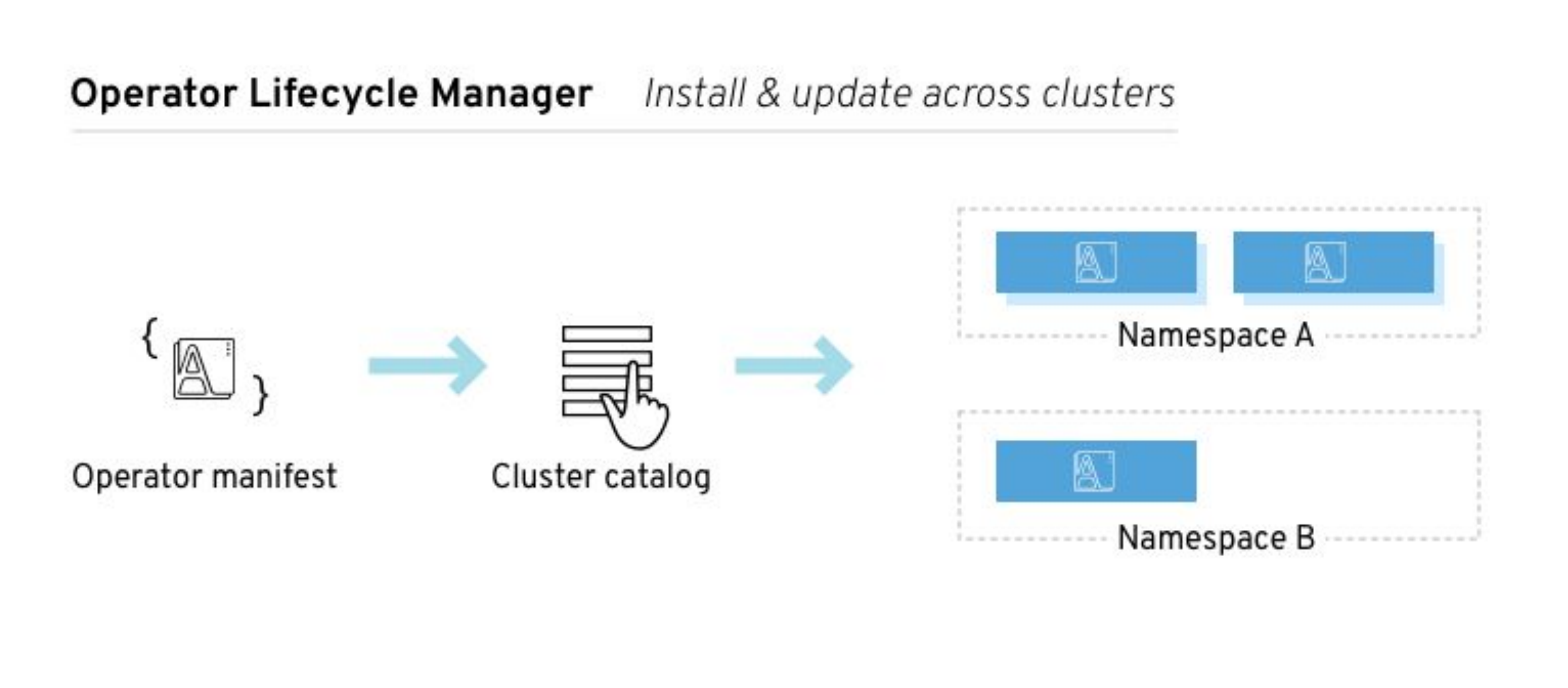
Test with live cluster

Publish

Operator manifest

Operator image

Operator Lifecycle Manager



Operator Lifecycle Manager *Install & update across clusters*

Operator manifest

Cluster catalog

Namespace A

Namespace B

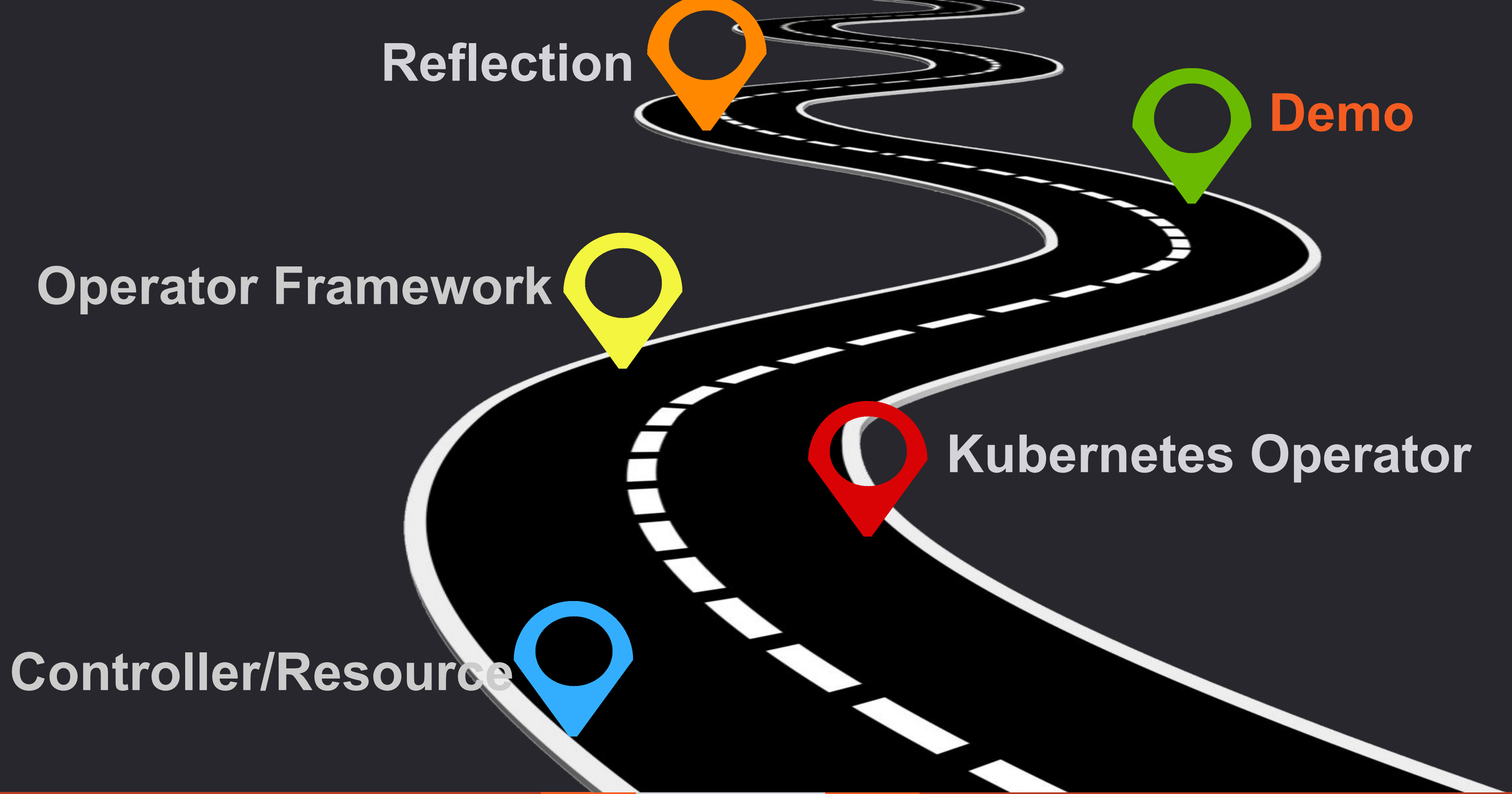
Operator Metering

Just Release Alpha Several Days Ago…

- Cluster’s CPU and Memory Reporting
- Infrastructure-as-a-Service (IaaS) Costs and Customized Metrics

Although There is Operator Framework…

- It’s Easy to Complete the First Operator by Following the Get Started, But Before Starting…
- Recommend to Realize Kube Controller Flow, And Custom Controller/Resource Implement Method
  - Ref 1
  - Ref 2



Controller/Resource

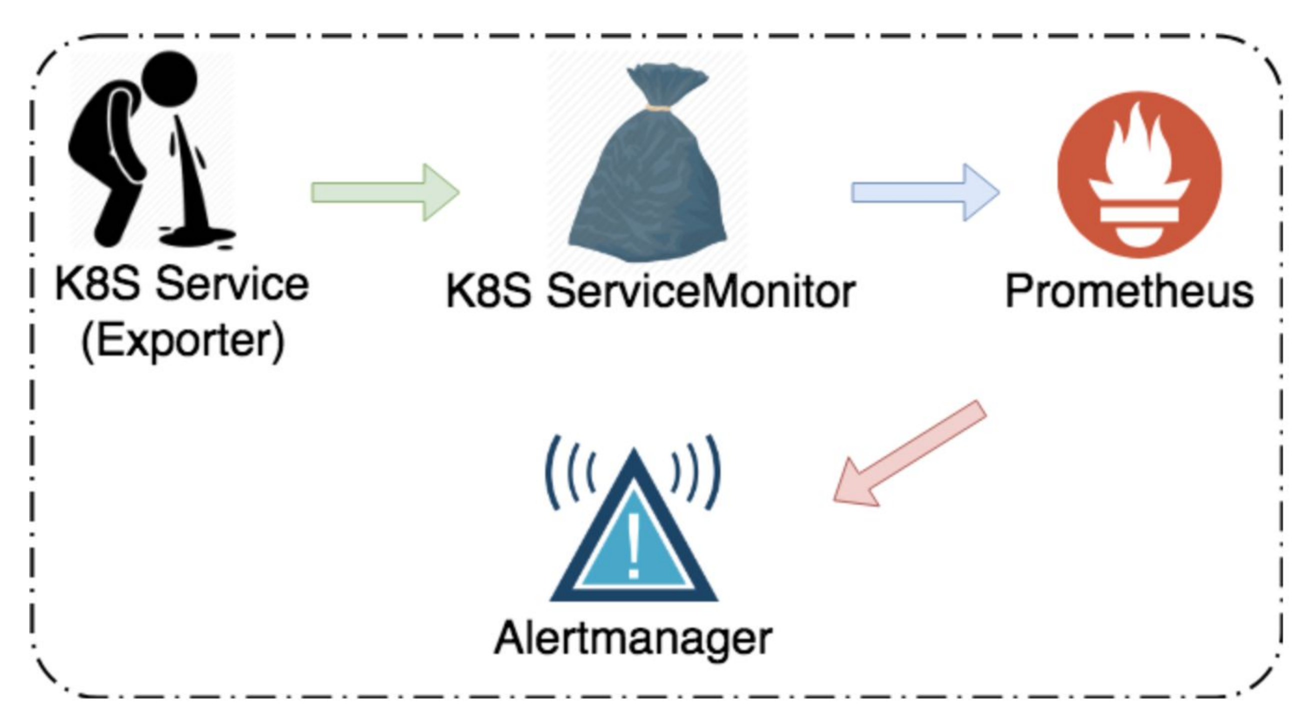
Kubernetes Operator

Operator Framework

Demo

Reflection

Prometheus Operator



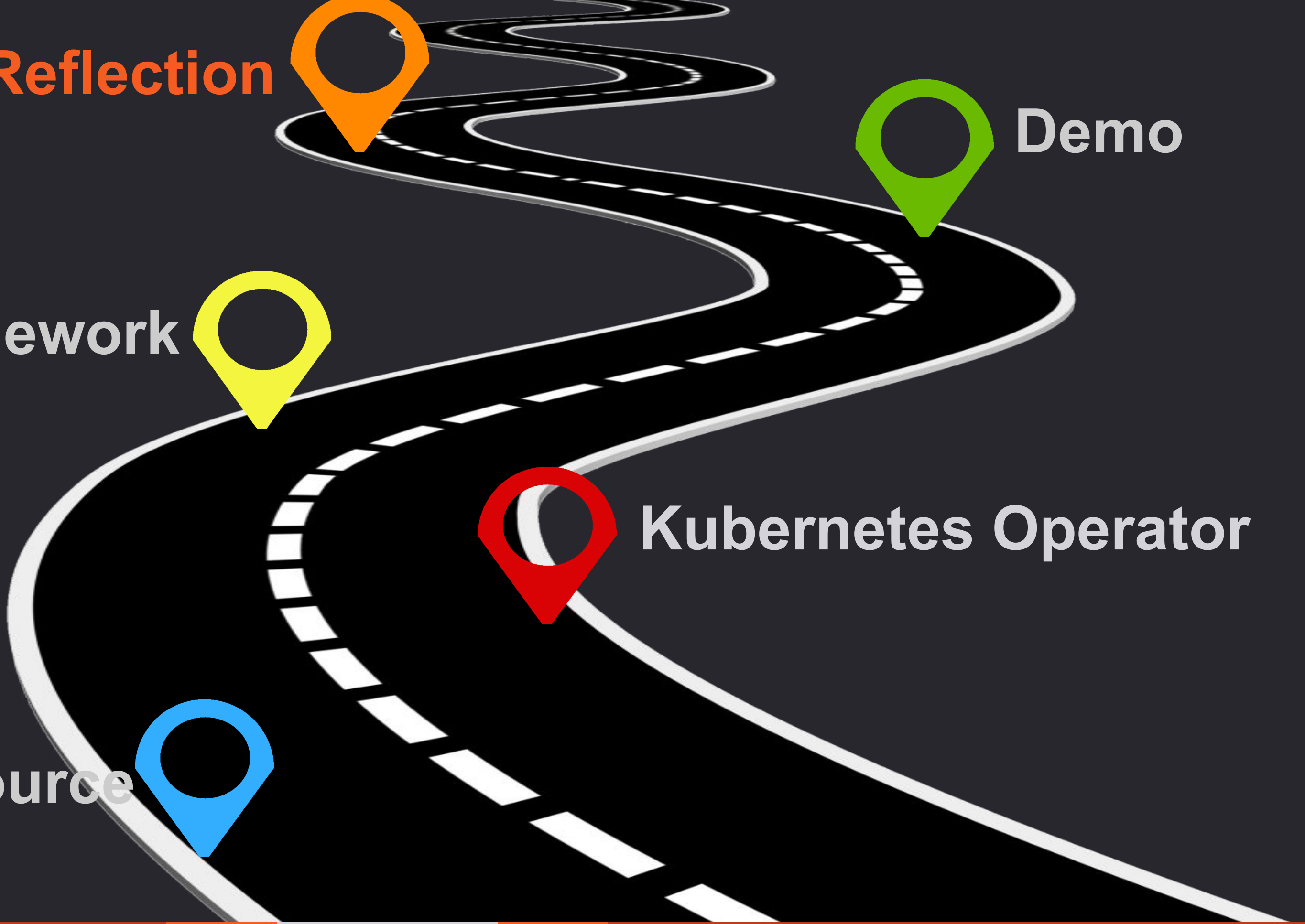
Ref

ElasticSearch Operator

Centralized Log **ASAP**

Demo Repository
https://goo.gl/zk5SB1

- Install Filebeat in each server
- Prepare Logstash indexer
- Setup Elasticsearch
  - Kibana
  - Master, Data, Ingest Node
  - Replicas
  - Backup
  - ...



Controller/Resource

Kubernetes Operator

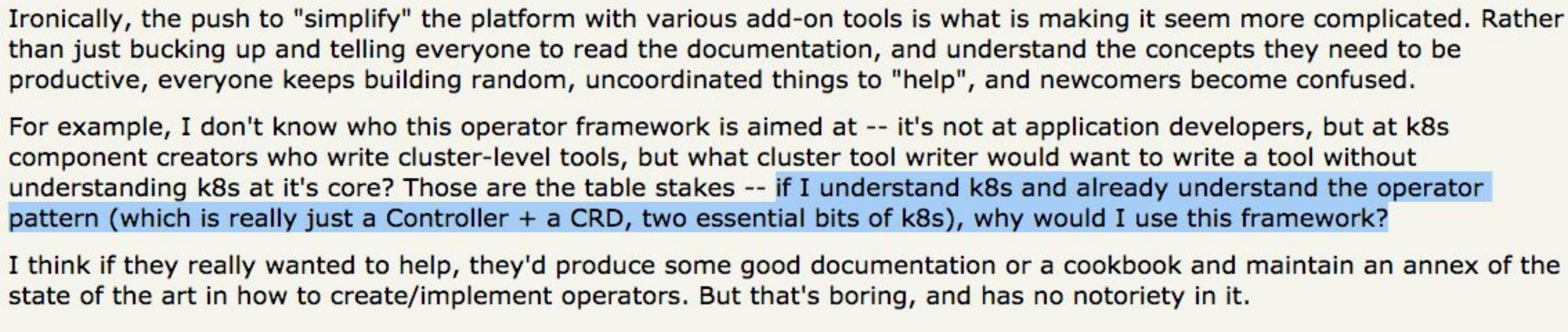
Operator Framework

Demo

Reflection

Operator is just a Controller + a CRD ?!

- https://news.ycombinator.com/item?id=16968873



Ironically, the push to "simplify" the platform with various add-on tools is what is making it seem more complicated. Rather than just bucking up and telling everyone to read the documentation, and understand the concepts they need to be productive, everyone keeps building random, uncoordinated things to "help", and newcomers become confused.

For example, I don't know who this operator framework is aimed at -- it's not at application developers, but at k8s component creators who write cluster-level tools, but what cluster tool writer would want to write a tool without understanding k8s at it's core? Those are the table stakes -- if I understand k8s and already understand the operator pattern (which is really just a Controller + a CRD, two essential bits of k8s), why would I use this framework?

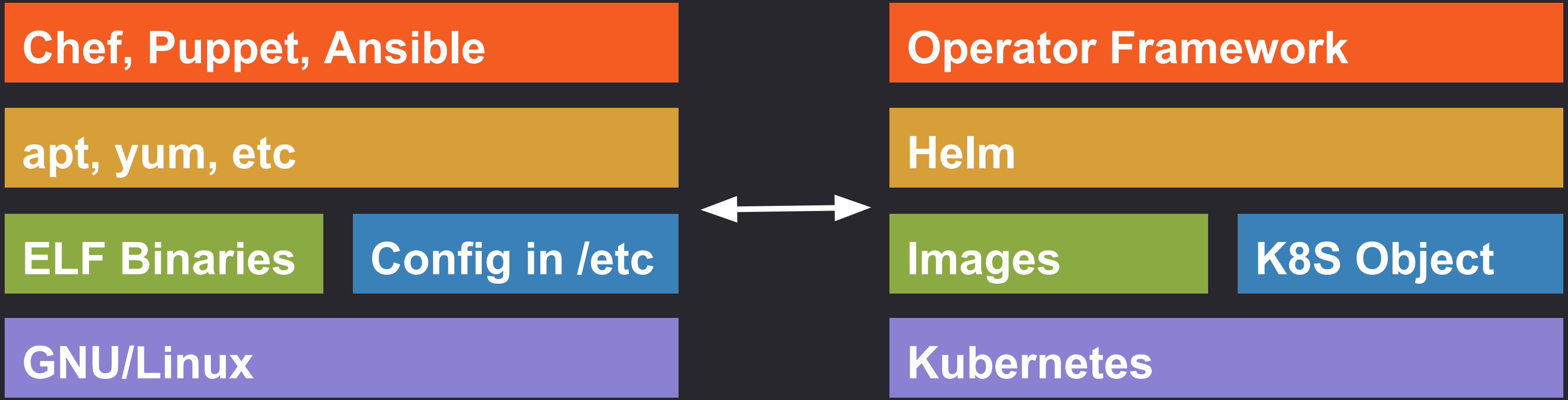
I think if they really wanted to help, they'd produce some good documentation or a cookbook and maintain an annex of the state of the art in how to create/implement operators. But that's boring, and has no notoriety in it.

helm-app-operator-kit

Installing a Custom Helm-Based Application

- https://github.com/operator-framework/helm-app-operator-kit
- Leverage a pre-existing Helm Chart to Deploy Kubernetes as a Operator

Good Old Days V.S. Kubernetes



Chef, Puppet, Ansible

apt, yum, etc

ELF Binaries

Config in /etc

GNU/Linux

Operator Framework

Helm

Images

K8S Object

Kubernetes

THANKS!

ANY QUESTIONS?