



KUBERNETES PATTERNS

DevNation Tech Talk, 10/2018

Roland Huß, Red Hat, @ro14nd



"m" for menu, "?" for other shortcuts



KUBERNETES

- Open Source container orchestration system
 - Scheduling
 - Horizontal scaling
 - Self-healing
 - Service discovery
 - Rollout and Rollbacks
- Declarative, resource-centric REST API

Design Patterns

DESIGN PATTERN

A **Design Pattern** describes a
repeatable solution to a
software engineering **problem**.

A Pattern Language

Towns • Buildings • Construction



Christopher Alexander

Sara Ishikawa • Murray Silverstein

WITH

Max Jacobson • Ingrid Fiksdahl-King

Shlomo Angel

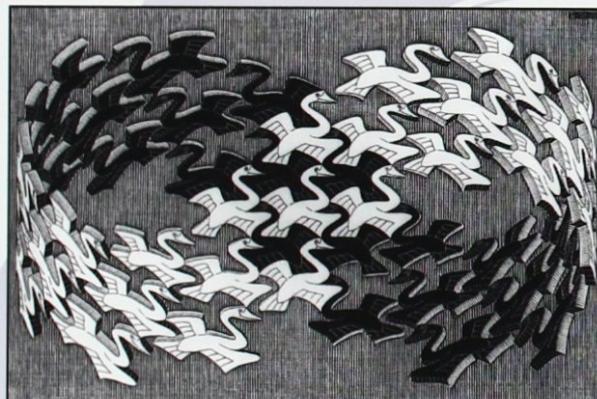


redhat

Design Patterns

Elements of Reusable
Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES



Kubernetes Patterns



Patterns, Principles, and Practices
for Designing Cloud Native Applications

Bilgin Ibryam & Roland Huss

<https://leanpub.com/k8spatterns>



STRUCTURE

- Problem
- Patterns:
 - Name
 - Solution
- <http://www.martinfowler.com/articles/writingPatterns.html>

FOUNDATIONAL PATTERNS

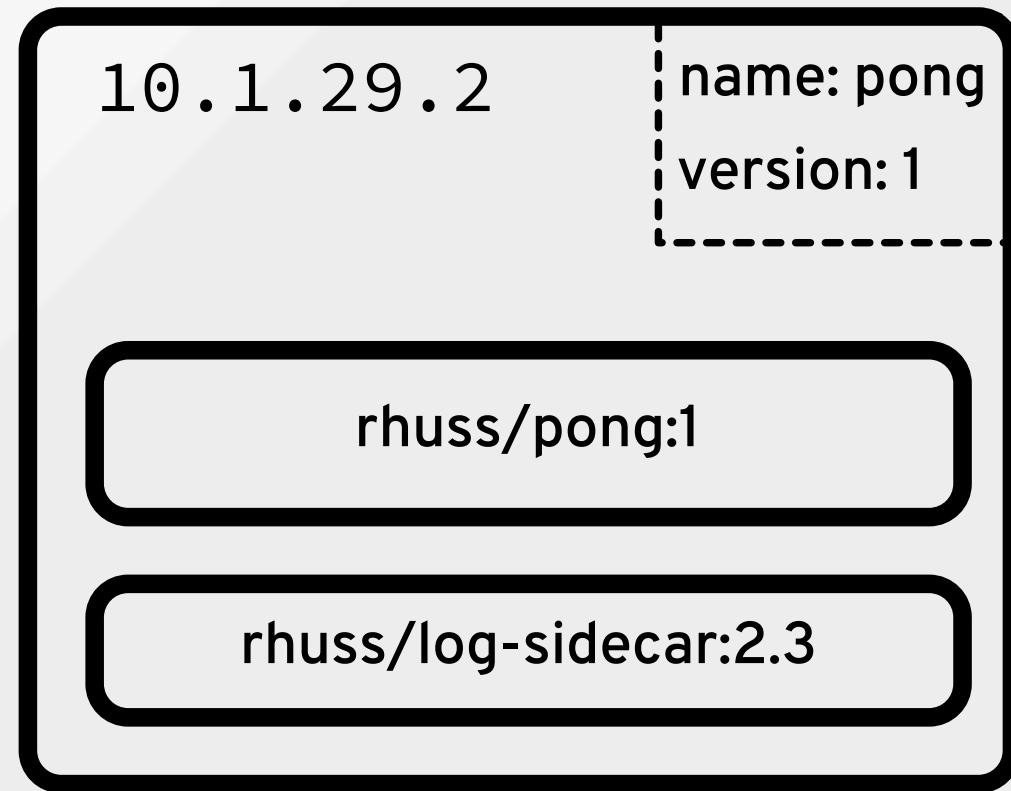
Automatable Unit

How can we create and manage applications with Kubernetes ?

- **Pods:** Atomic unit of containers
- **Services:** Entry point to pods
- Grouping via **Labels,**
Annotations, Namespaces

POD

- Kubernetes Atom
- One or more containers sharing:
 - IP and ports
 - Volumes
- Ephemeral IP address



POD DECLARATION

```
apiVersion: v1
kind: Pod
metadata:
  name: pong
  labels:
    name: pong
    version: "1"
spec:
  containers:
    - image: "rhuss/pong:1"
      name: pong
      ports:
        - containerPort: 8080
    - image: "rhuss/log-sidecar:2.3"
      name: log
```

REPLICA SET

- Responsible for managing **Pods**
- **replicas** : Number of **Pod** copies to keep
- Label selector chooses **Pods**
- Holds a template for creating new **Pods**

ReplicaSet

replicas: **3**

Selector:

name: pong
version: 1

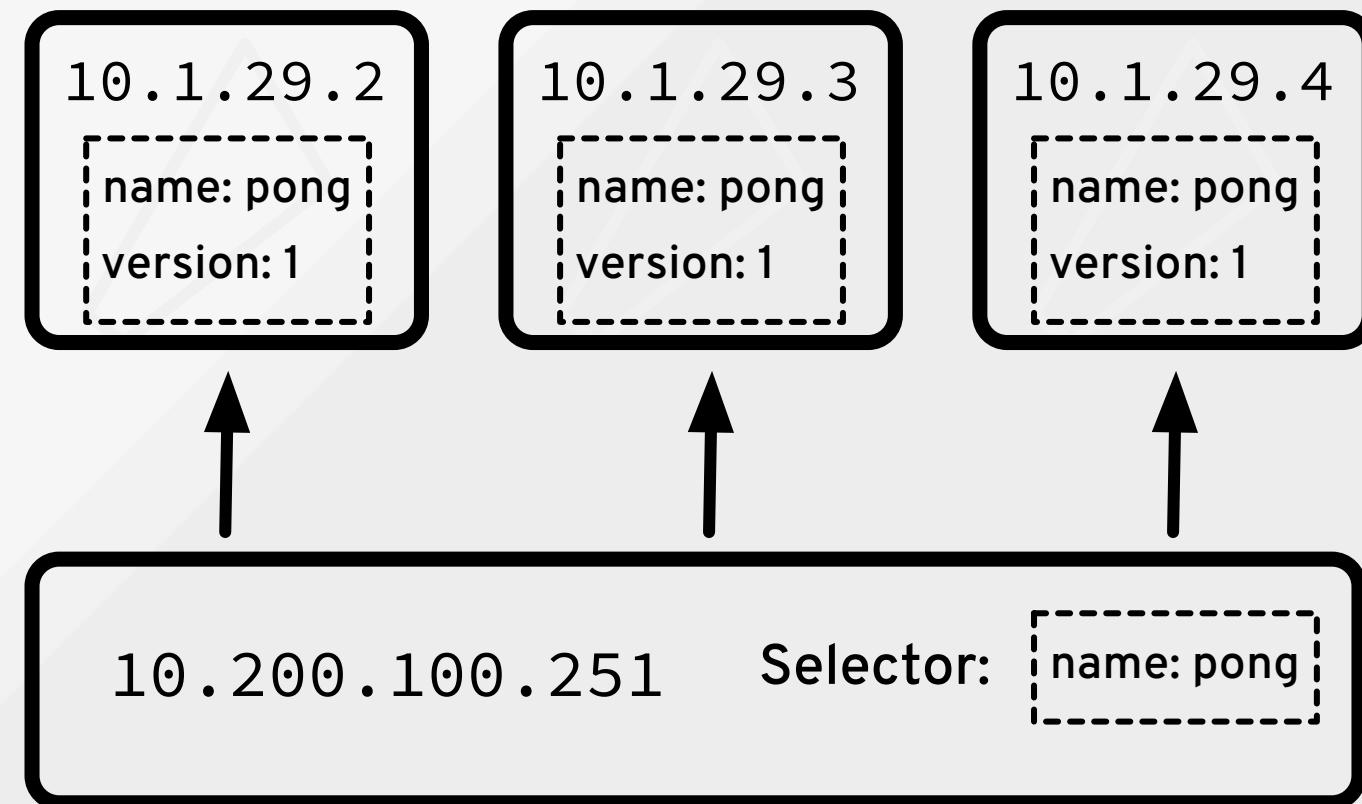
10.1.29.2
name: pong
version: 1

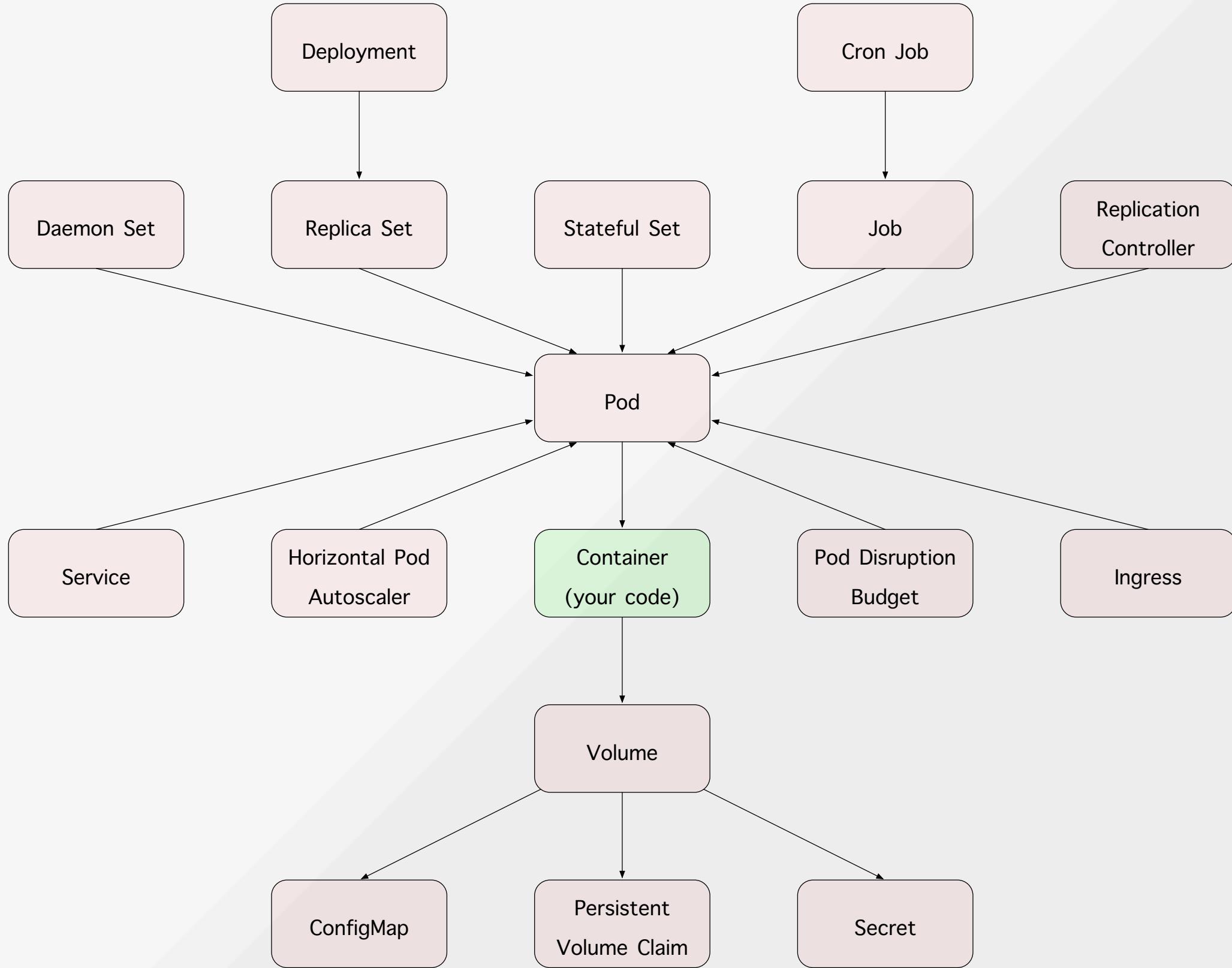
10.1.29.3
name: pong
version: 1

10.1.29.4
name: pong
version: 1

SERVICE

- Entrypoint for a set of **Pods**
- **Pods** chosen by **Label** selector
- Permanent IP address





Declarative Deployment

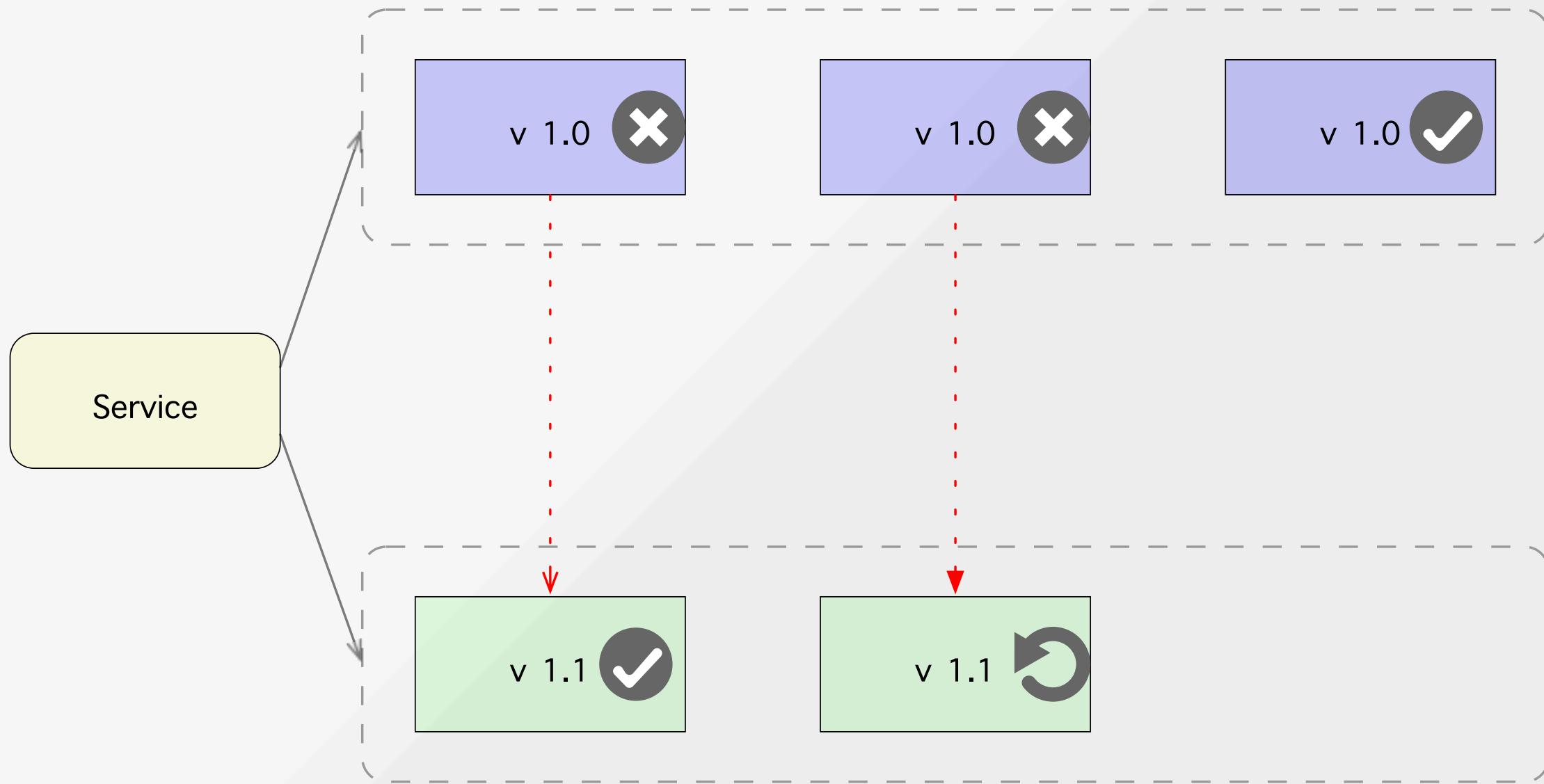
How can applications be deployed
and updated ?

- Declarative versus Imperative deployment
- Various update strategies

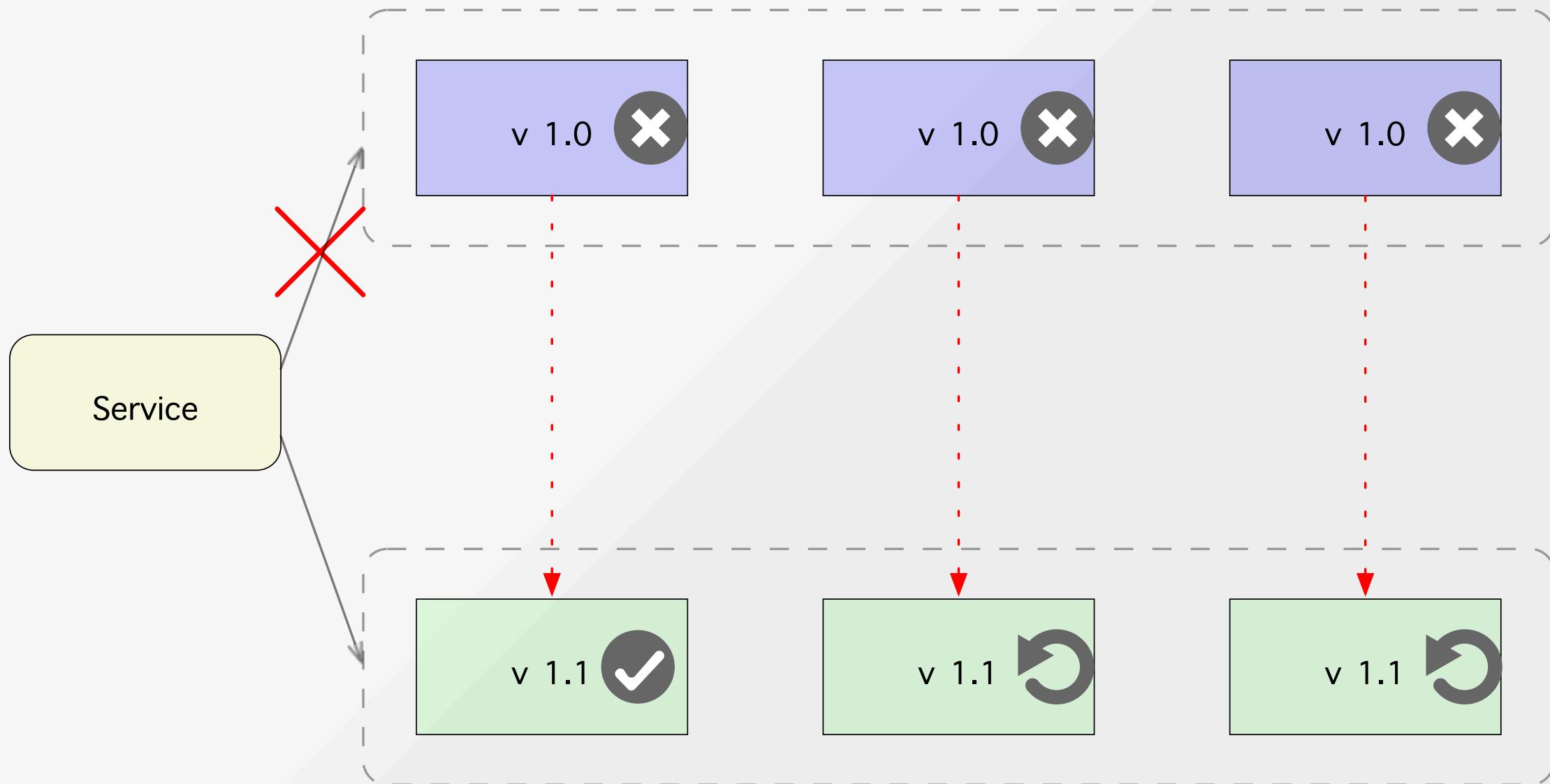
DEPLOYMENT

- Holds template for Pod
- Creates ReplicaSet on the fly
- Allows rollback
- Update strategies declarable
- Inspired by DeploymentConfig from OpenShift

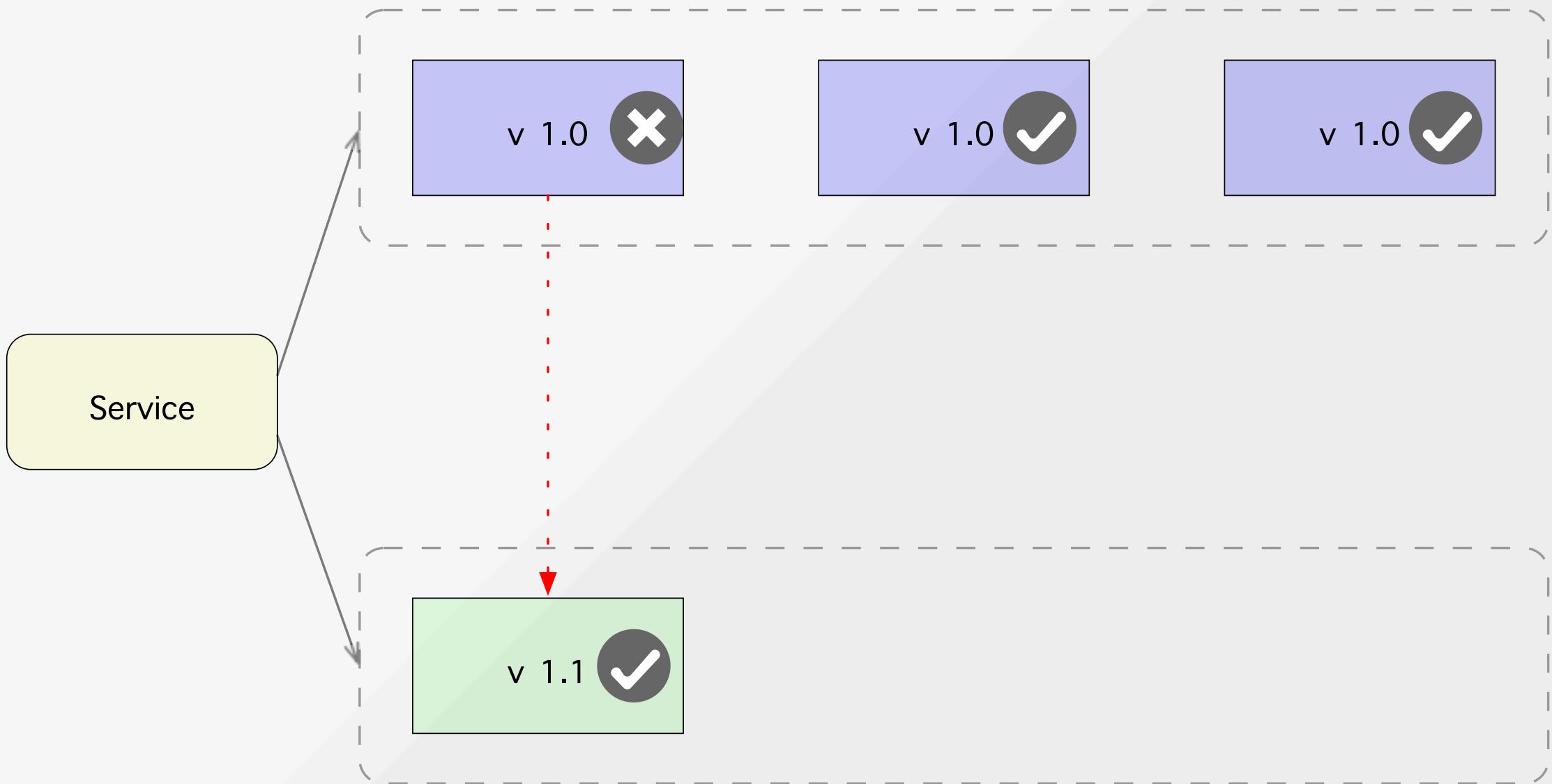
ROLLING



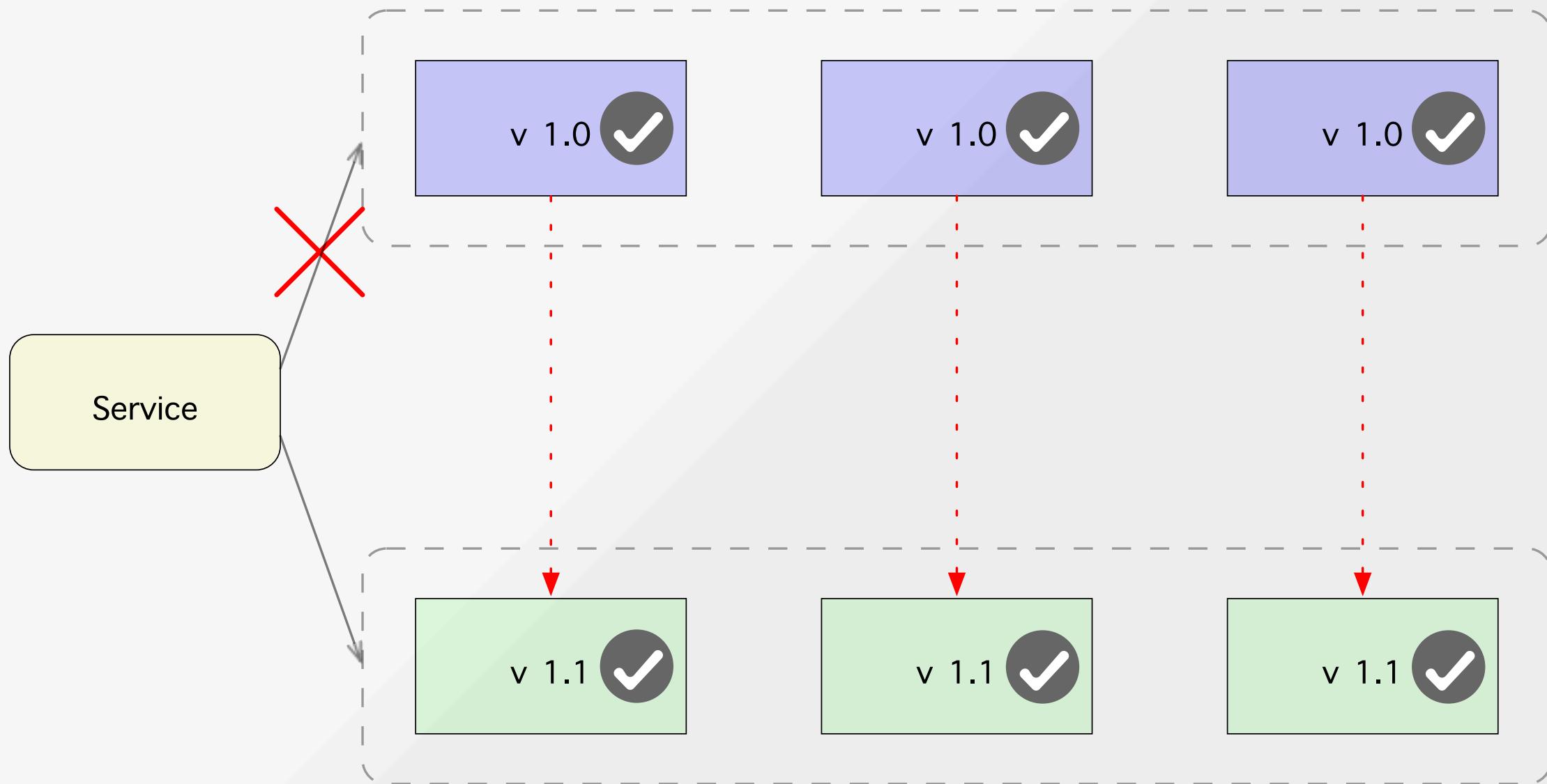
FIXED



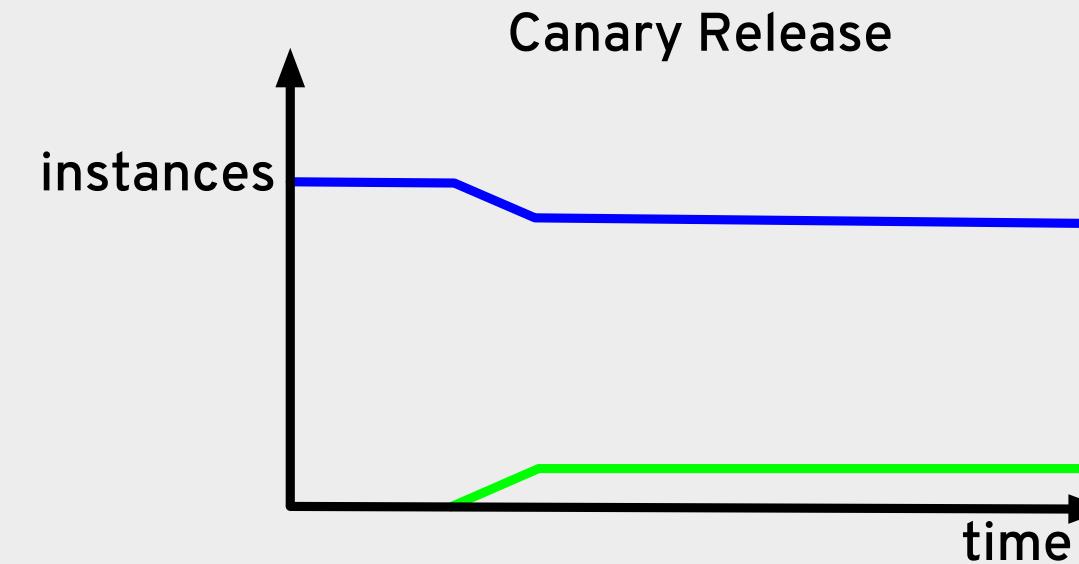
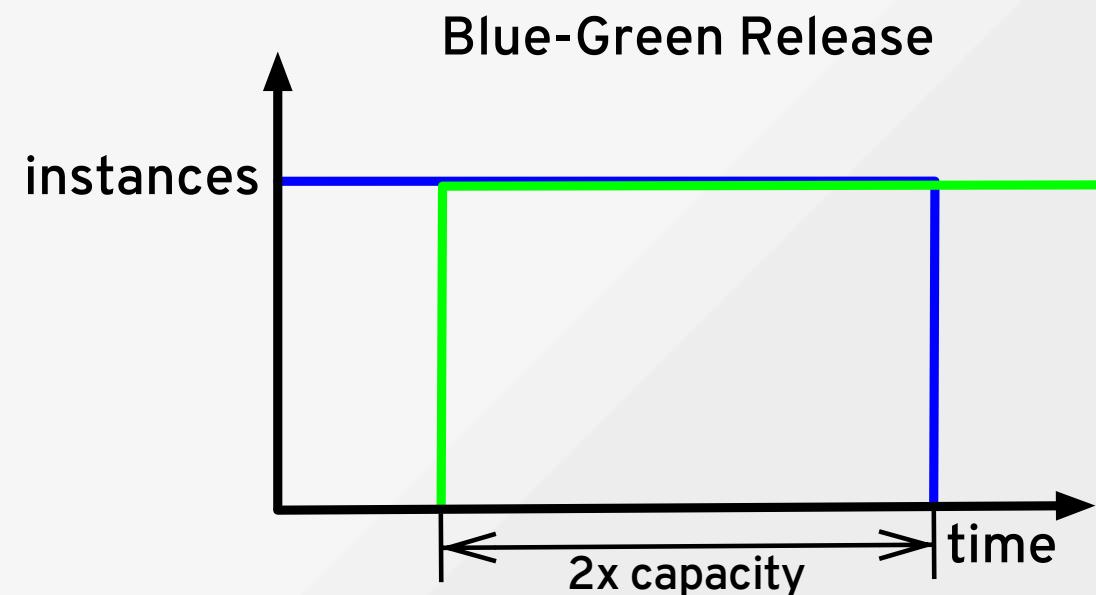
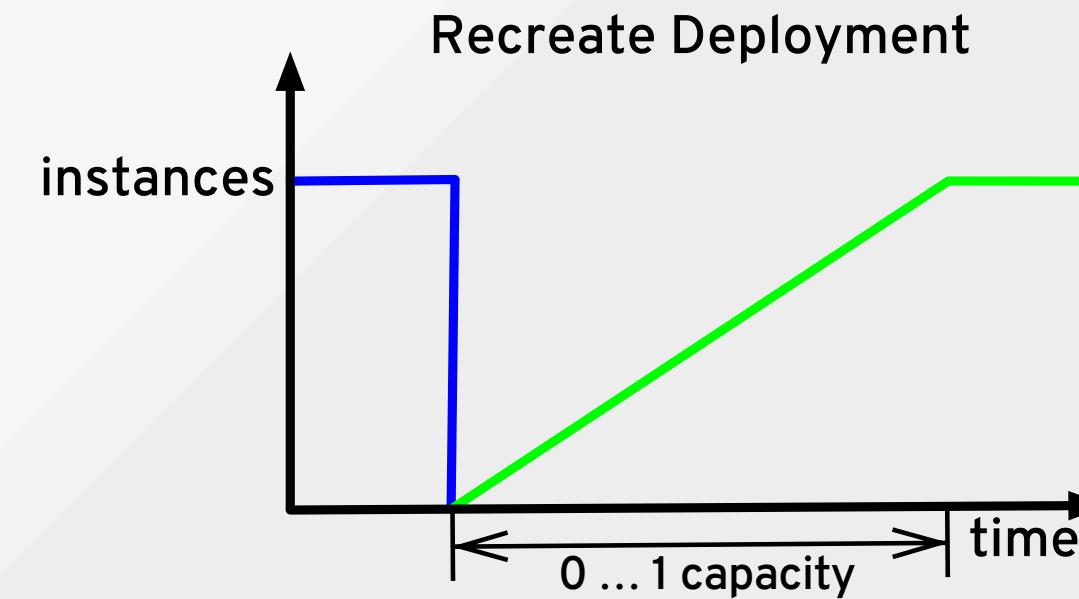
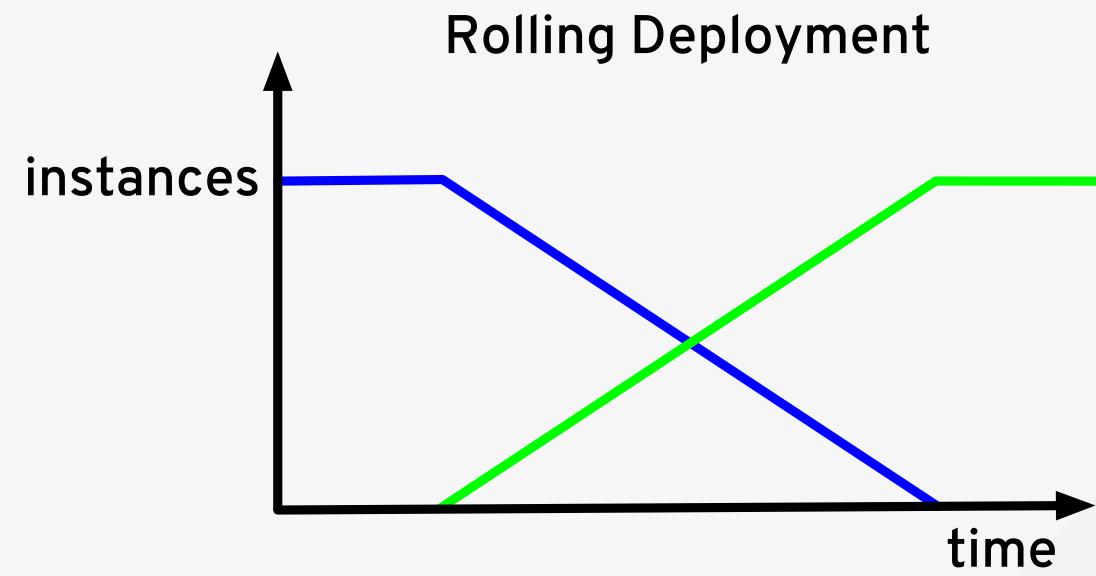
CANARY



BLUE-GREEN



SUMMARY

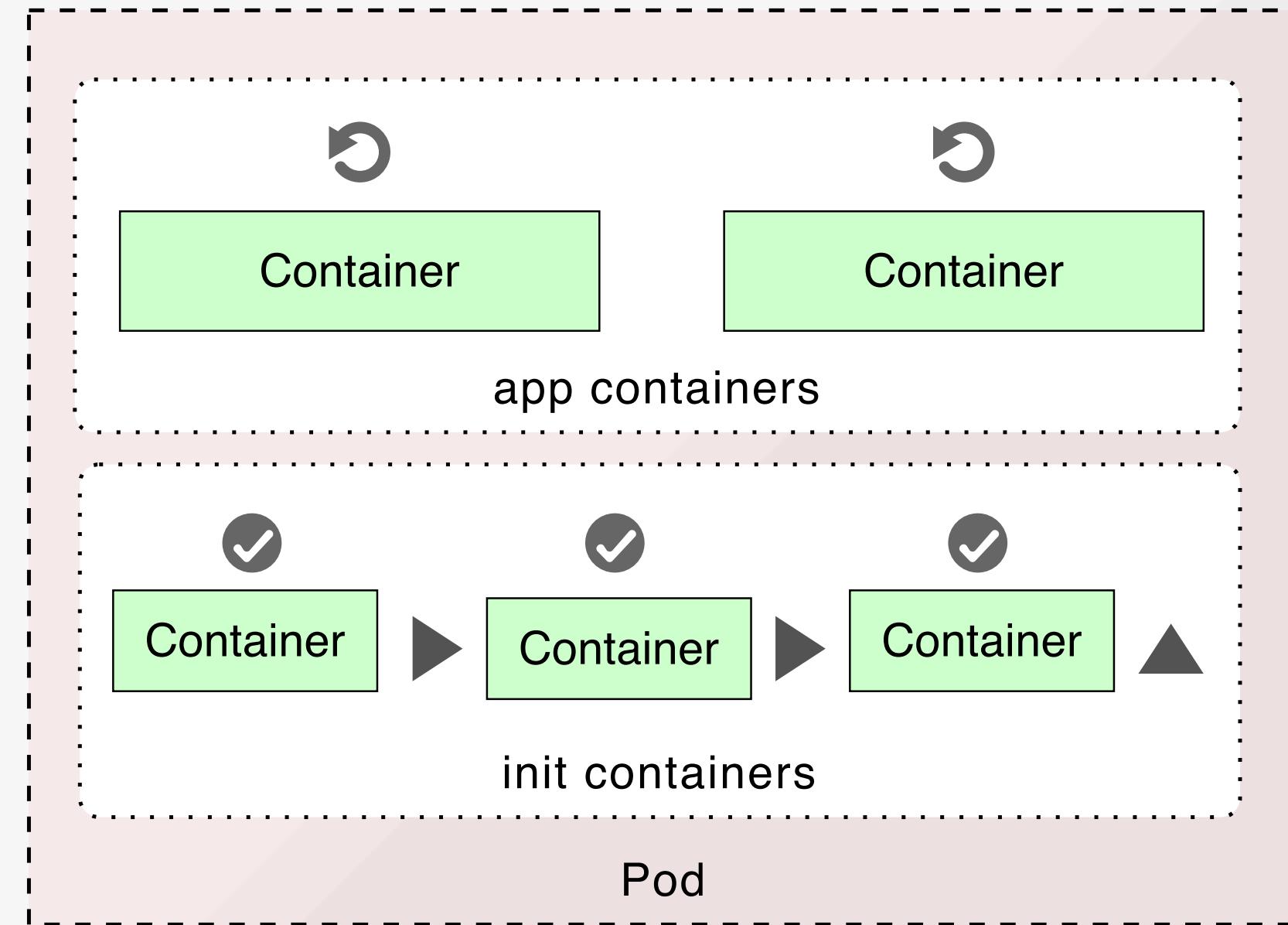


STRUCTURAL PATTERNS

Initializer

How can I initialize my containerized applications ?

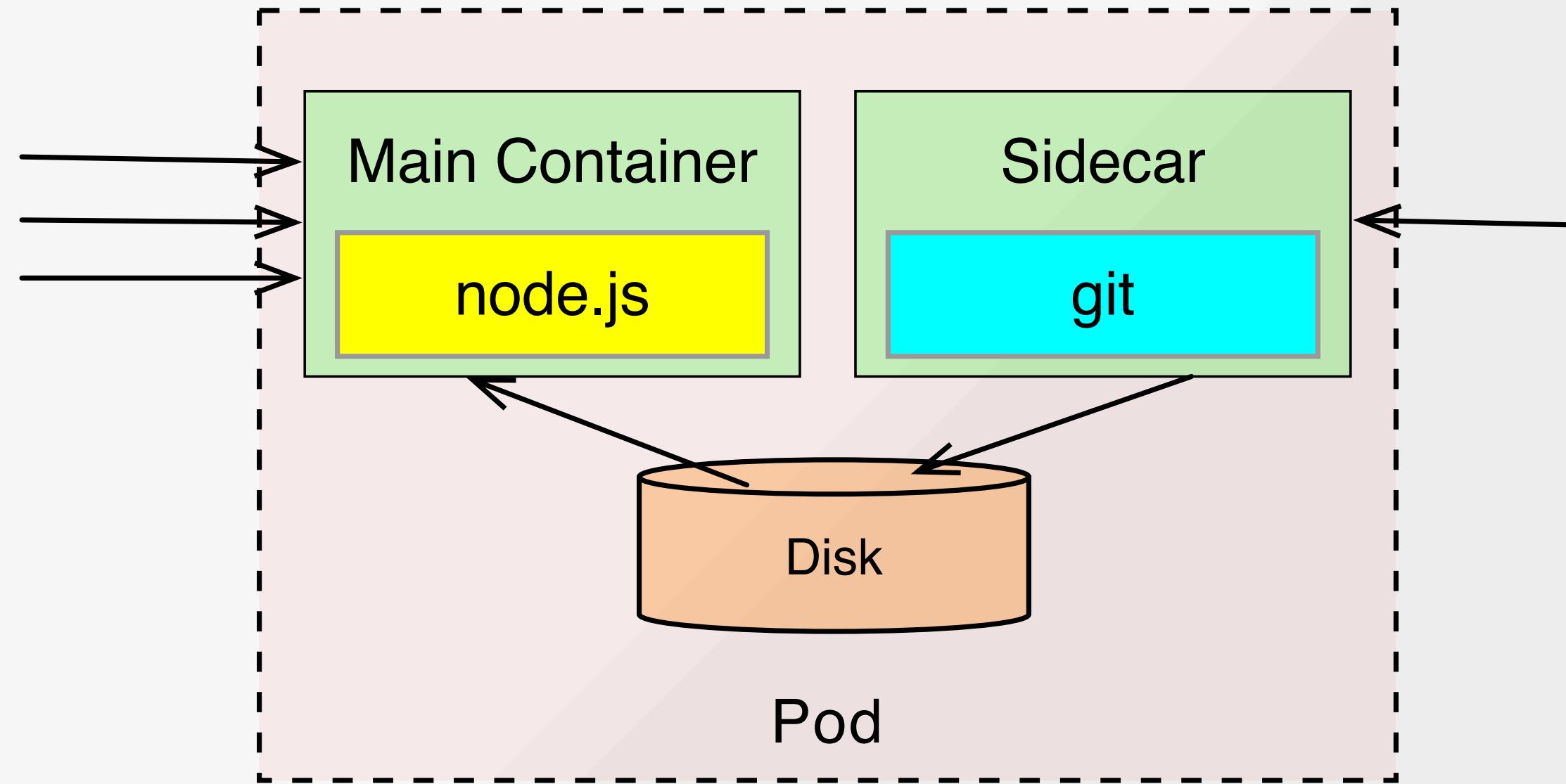
- Init container :
 - Part of a Pod
 - One shot action before Pod starts
 - Needs to be idempotent
 - Has own resource requirements



Sidecar

How can I extend the functionality of
an existing container ?

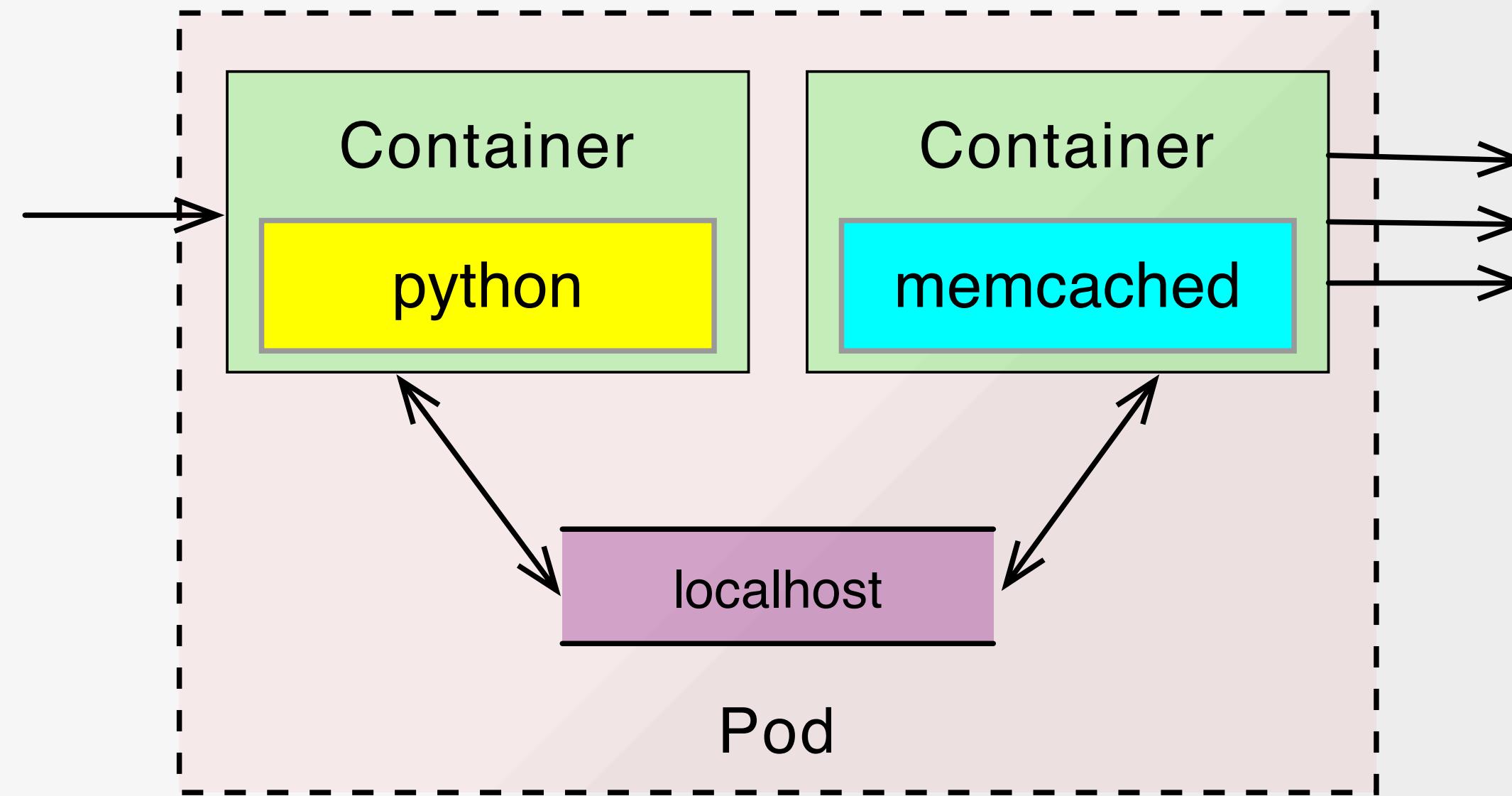
- Runtime collaboration of containers
- Connected via shared resources:
 - Network
 - Volumes



Ambassador

How to decouple a container's
access **to** the outside world ?

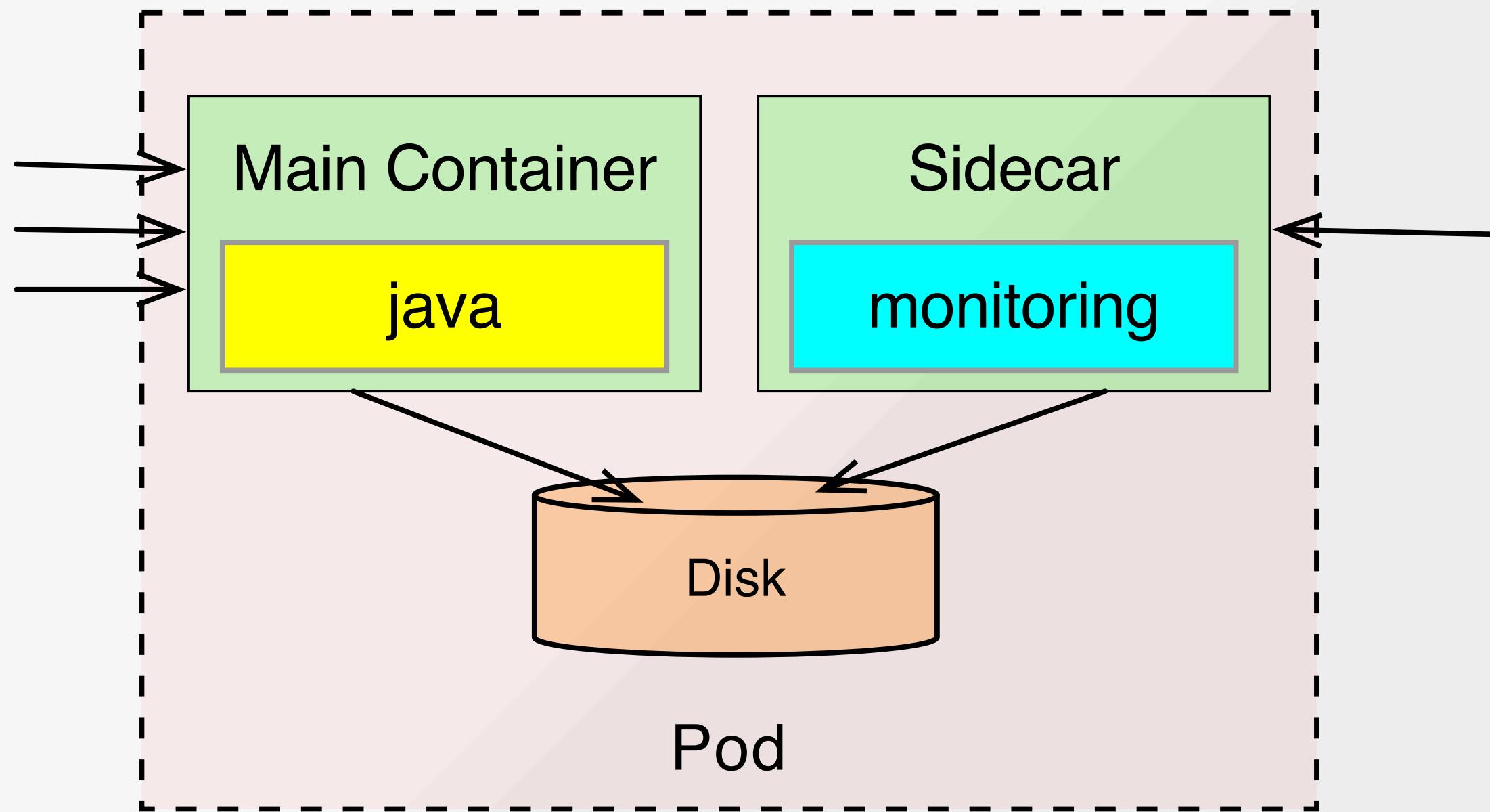
- Also known as **Proxy**
- Specialization of a Sidecar
- E.g. infrastructure services
 - Circuit breaker
 - Tracing



Adapter

How to decouple access to a container **from** the outside world ?

- Opposite of Ambassador
- Uniform access to application
- Examples:
 - Monitoring
 - Logging



ADVANCED PATTERNS

Custom Controller

How can I extend the platform itself
without changing it ?

- Watching resources by registering for Kubernetes events
- Reacting on changes in resource declarations

CONTROLLER

- Managed pod listening for Kubernetes API events
- **State Reconciliation** : Make the current state like the declared desired state
- Often used in combination with **CustomResources**

CATEGORIES

- **Extension Controller**: Extend the Kubernetes platform itself
- **Application Controller**: Combine Kubernetes with an application specific domain

Custom Resource

How can I manage custom domain specific resources ?

- **Custom Resource Definition** (CRD) managed by Kubernetes
- Accessible via the Kubernetes API
- Watched by Custom Controllers

EXAMPLE CRD

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: prometheuses.monitoring.coreos.com
spec:
  group: monitoring.coreos.com
  names:
    kind: Prometheus
    plural: prometheuses
  scope: Namespaced
  version: v1
  validation: ....
```

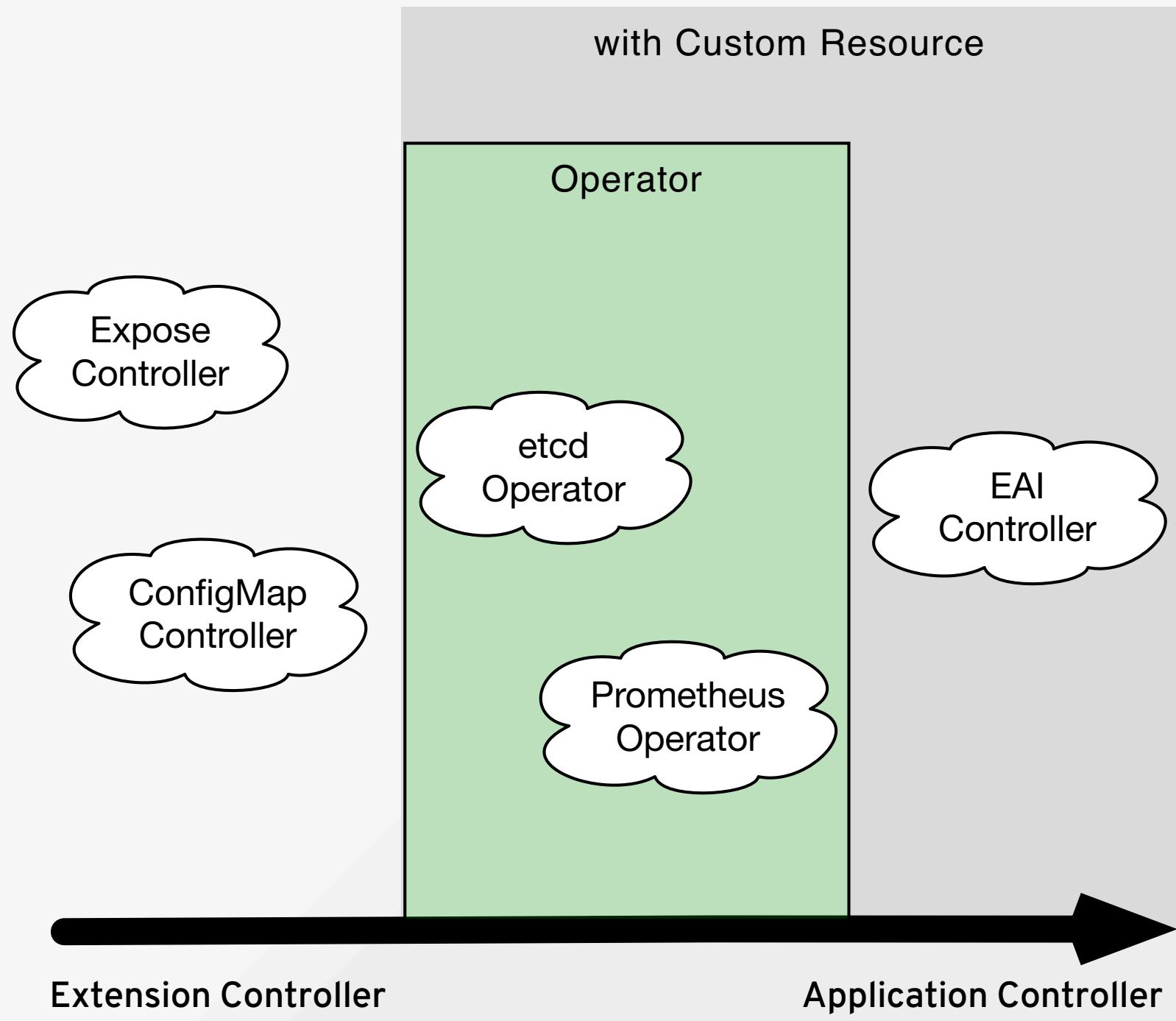
EXAMPLE CRD

```
apiVersion: monitoring.coreos.com/v1
kind: Prometheus
metadata:
  name: prometheus
spec:
  serviceMonitorSelector:
    matchLabels:
      team: frontend
  resources:
    requests:
      memory: 400Mi
```

OPERATORS

- Combine Custom Controller and Custom Resource
- Manages and deploys custom Kubernetes application
- Operator Framework by CoreOS:
 - Operator SDK
 - Operator Lifecycle Manager
 - Operator Metering

SPECTRUM



Kubernetes Patterns



**Patterns, Principles, and Practices
for Designing Cloud Native Applications**

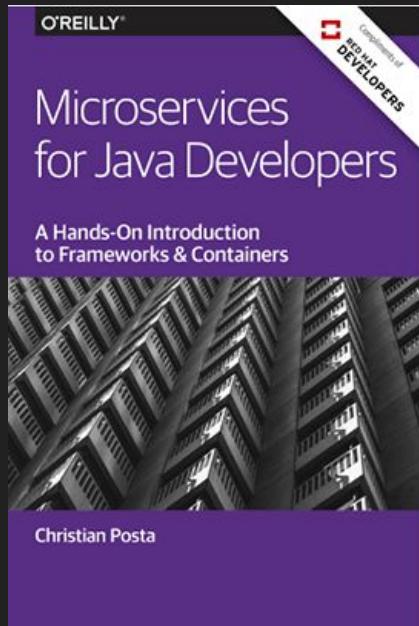
Bilgin Ibryam & Roland Huss

<https://leanpub.com/k8spatterns>

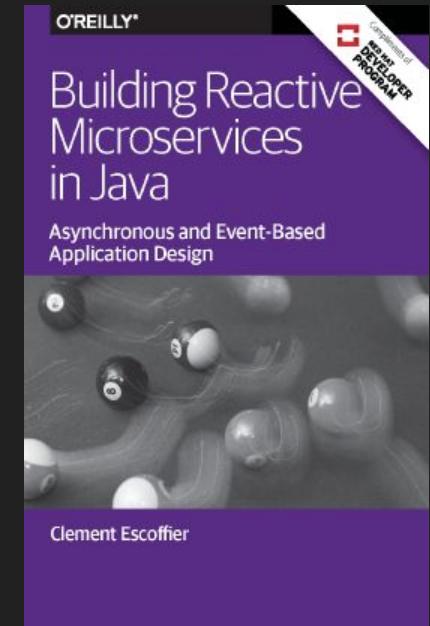
Free Whitepaper: <http://bit.ly/cloud-native-design-whitepaper>



bit.ly/javamicroservicesbook



bit.ly/reactivemicroservicesbook



Free eBooks from developers.redhat.com

Microservices Introductory Materials

Demo: bit.ly/msa-tutorial

Slides: bit.ly/microservicesdeepdive

Video Training: bit.ly/microservicesvideo

[Kubernetes for Java Developers](#)

Advanced Materials

bit.ly/istio-tutorial

learn.openshift.com/servicemesh

bit.ly/faas-tutorial

learn.openshift.com/serverless

O'REILLY®

Migrating to Microservice Databases

From Relational Monolith
to Distributed Data



Edson Yanaga

Compliments of
**RED HAT
DEVELOPERS**

bit.ly/mono2microdb

O'REILLY®

Introducing Istio Service Mesh for Microservices

Build and Deploy Resilient, Fault-Tolerant
Cloud-Native Applications



Christian Posta & Burr Sutter

Compliments of
RED HAT
DEVELOPER
PROGRAM

bit.ly/istio-book



QUESTIONS ?

Twitter ro14nd

Book <https://leanpub.com/k8spatterns>

Slides <https://github.com/ro14nd-talks/kubernetes-patterns>