

# RED HAT<sup>®</sup> DEVELOPER PROGRAM



spring

ON



RED HAT<sup>®</sup>  
OPENSHIFT



@kamesh\_sampath



kamesh.sampath@hotmail.com

[bit.ly/springboot-on-openshift](https://bit.ly/springboot-on-openshift)

# [developers.redhat.com](https://developers.redhat.com)

# Who am I ?



## Kamesh Sampath

Director of Developer Experience at Red Hat

- Active Open Source Contributor
- Creator vert.x-maven-plugin → <https://vmp.fabric8.io/>
- Contributor fabric8 platform → <https://fabric8.io/>



ksampath@hotmail.com



kameshsampath



@kamesh\_sampath

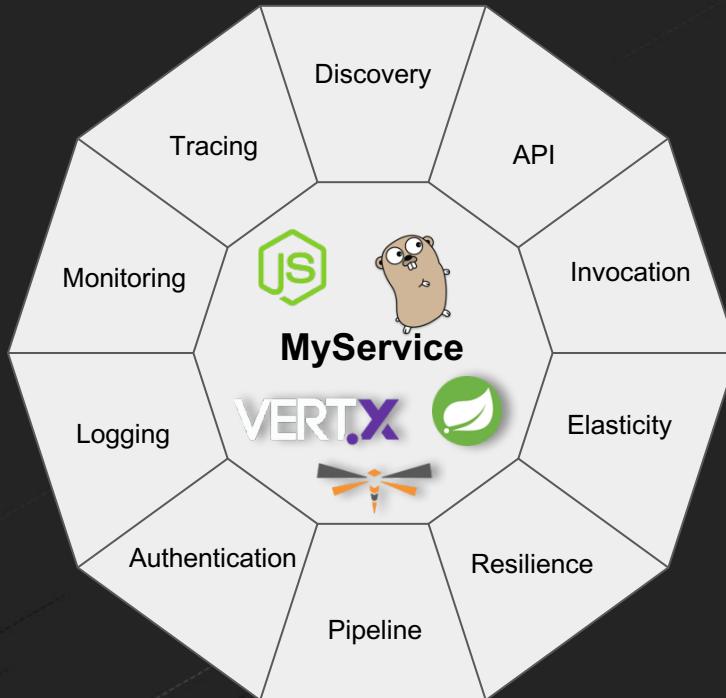
# What is a microservice ?

The microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API.

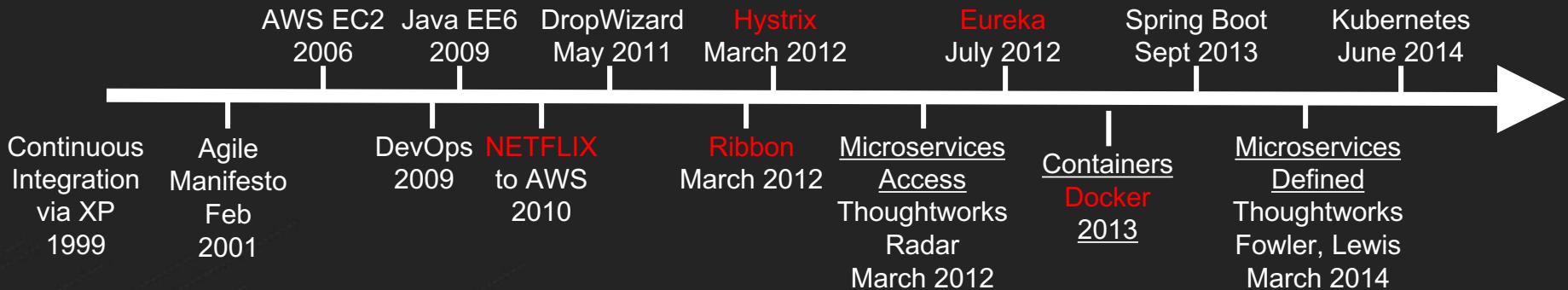
These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

Martin Fowler

# Microservices Properties



# Short History of Microservices



**NETFLIX** | OSS

# java -jar myapp.jar

## DropWizard

[www.dropwizard.io](http://www.dropwizard.io)

JAX-RS API

First to market

DropWizard Metrics

Embeddable  
servers:  
Jetty



## Spring Boot

[projects.spring.io/spring-boot](http://projects.spring.io/spring-boot)

Spring API  
(@RestController)

'Starter' POMs:  
[start.spring.io](http://start.spring.io)

Embeddable servers:  
Tomcat, Jetty, Undertow



## Vert.x

[vertx.io](http://vertx.io)

Reactive  
Async/non-blocking

`vertx run myhttp.java`

HTTP, HTTP/2, TCP,  
UDP, Websockets,  
etc. (out of the box)



## WildFly Swarm

[wildfly-swarm.io](http://wildfly-swarm.io)

Java EE 7 APIs

'Starter' POMs:  
[wildfly-swarm.io/generator](http://wildfly-swarm.io/generator)

Embeddable  
servers:  
WildFly (Undertow)



# Java Microservices Platform circa 2014



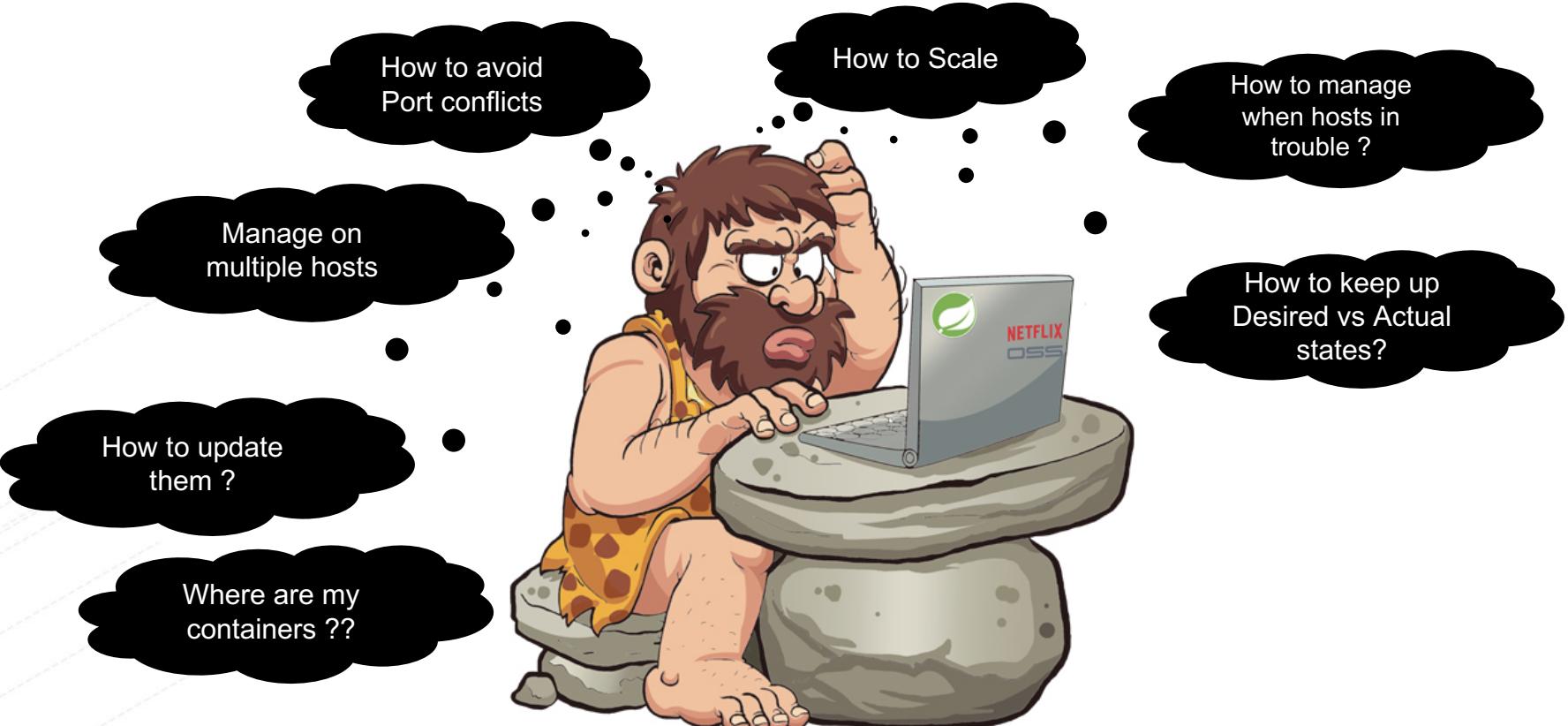
spring



@kamesh\_sampath



# Caveman's Spring Boot



@kamesh\_sampath





@kamesh\_sampath



# Open Container Initiative (OCI)

## Runtime Spec (cri-o)

The Runtime Specification outlines how to run a “filesystem bundle” that is unpacked on disk

e.g. Docker , rkt



## ImageSpec

A container image package with required metadata that will be used by runtimes to run the container images



e.g. Docker formatted Images  
[docker.io/busybox](https://docker.io/busybox)

<https://www.opencontainers.org/>

<https://www.opencontainers.org/image-spec>

<https://cri-o.io>

# Meet Kubernetes

- Greek for “Helmsman,” also the root of the word “Governor” (from latin: gubernator)
- Container orchestrator
- Supports multiple cloud and bare-metal environments
- Inspired by Google’s experience with containers
- Open source, written in Go
- Manage applications, not machines



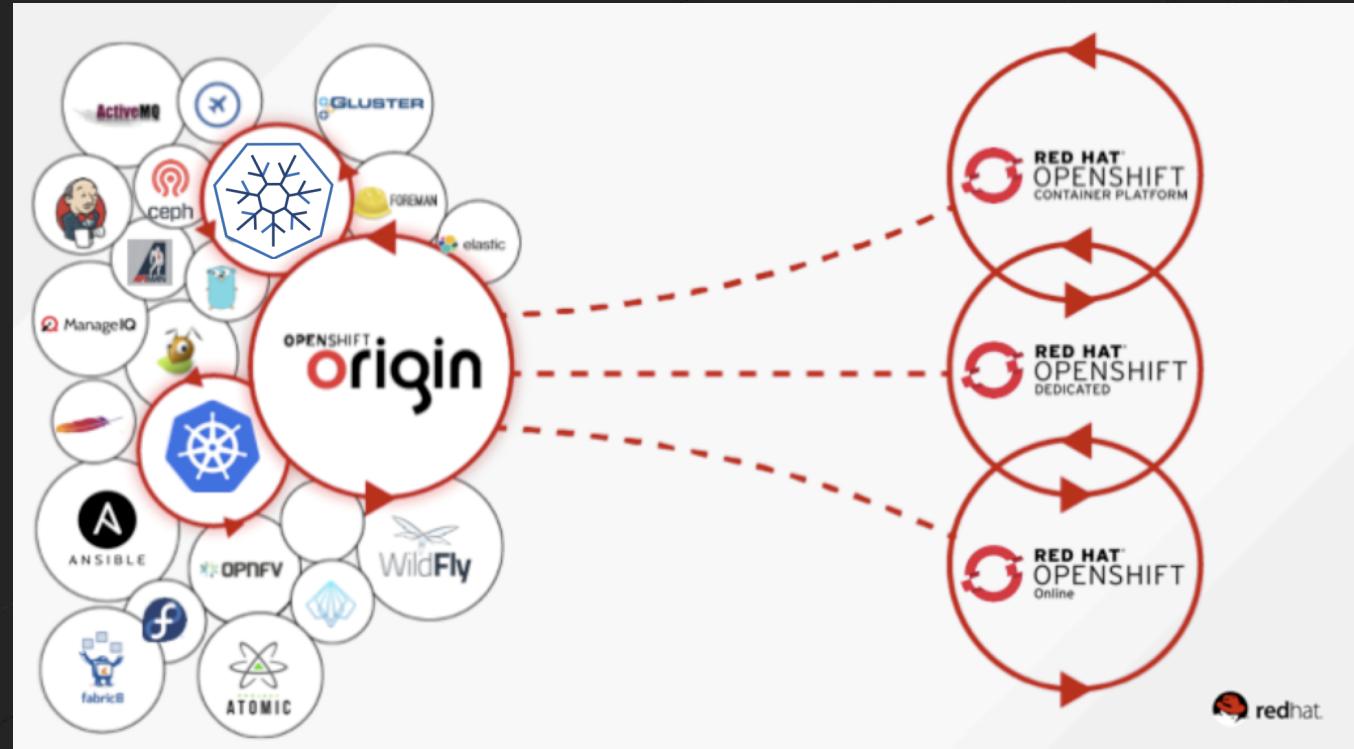
# Pods

A group of whales is commonly referred to as a pod and a pod usually consists a group of whales that have bonded together either because of biological reasons or through friendships developed between two or more whales.

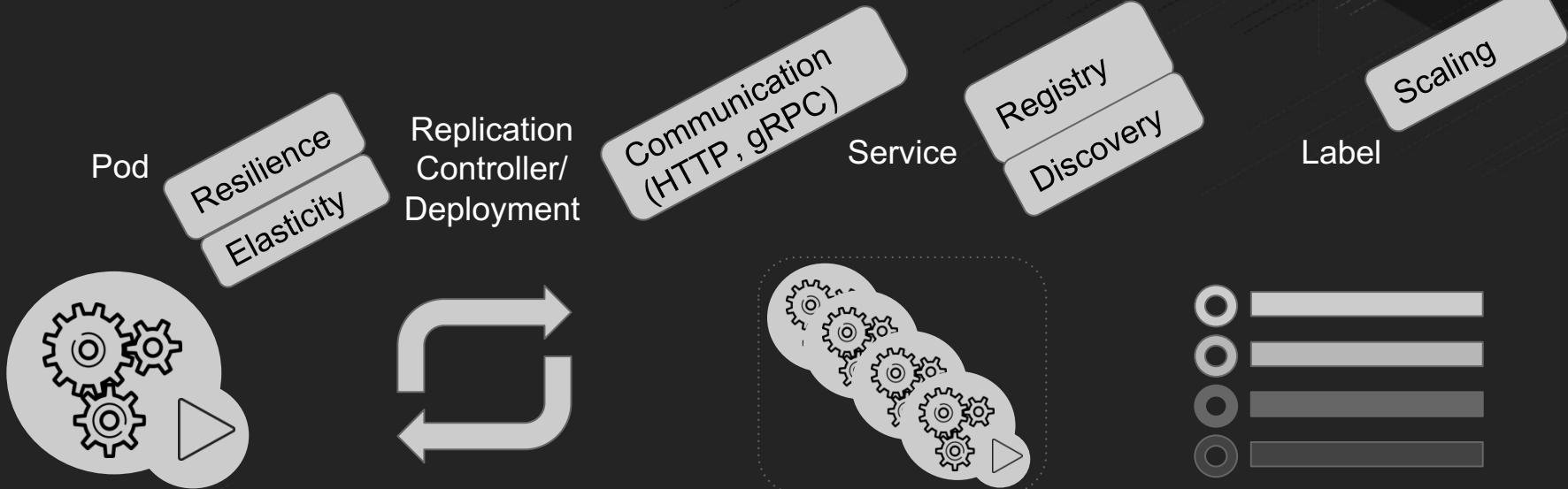
In many cases a typical whale pod consists of anywhere from 2 to 30 whales or more.\*

\*<http://www.whalefacts.org/what-is-a-group-of-whales-called/>

# OpenShift is Enterprise Kubernetes



# Kubernetes Cluster



- ✓ 1+ containers
- ✓ Shared IP
- ✓ Shared storage volume
- ✓ Shared resources
- ✓ Shared lifecycle

- ✓ Ensures that a specified number of pod replicas are running at any one time

- ✓ Grouping of pods (acting as one) has stable virtual IP and DNS name

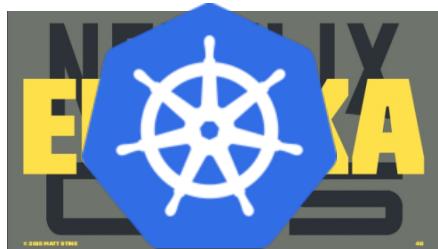
- ✓ Key/Value pairs associated with Kubernetes objects (env=production)

# My Boot is Kube Awesome

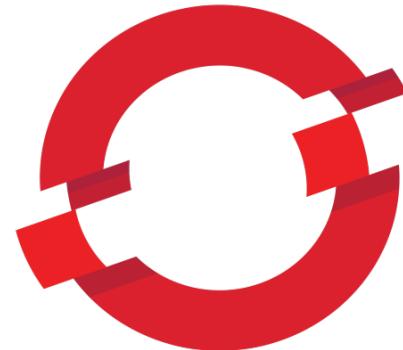
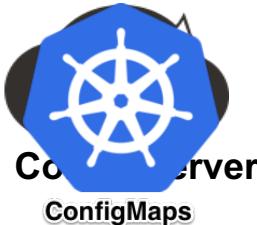


# DEMO (Hello Boot k8s deployment)

# Better Spring Boot Microservices Platform circa 2016



Services



## OPENSIFT



@kamesh\_sampath



# Spring Cloud Kubernetes

- Service Discovery

Spring Discovery Client using Kubernetes Service Discovery

- ConfigMap Property Source

How to use Kubernetes ConfigMap as Spring Property source

- Zipkin Service Discovery

Using Zipkin with Kubernetes for distributed tracing

# How to add to My Project?

## Spring Cloud Kubernetes Discovery

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-kubernetes-discovery</artifactId>
  <version>0.2.0.RELEASE</version>
</dependency>
```

## Spring Cloud Kubernetes Netflix

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-kubernetes-netflix</artifactId>
  <version>0.2.0.RELEASE</version>
</dependency>
```

## Spring Cloud Kubernetes Zipkin

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-kubernetes-config</artifactId>
  <version>0.2.0.RELEASE</version>
</dependency>
```

## Spring Cloud Kubernetes Config

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-kubernetes-config</artifactId>
  <version>0.2.0.RELEASE</version>
</dependency>
```

## Spring Cloud Kubernetes Archaius

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-kubernetes-archaius</artifactId>
  <version>0.2.0.RELEASE</version>
</dependency>
```



@kamesh\_sampath

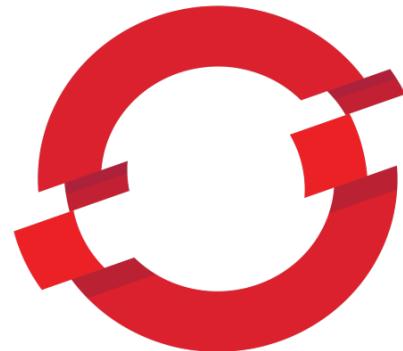
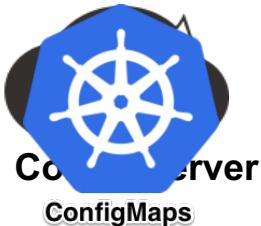


# DEMO (Discovery)

# Better Spring Boot Microservices Platform circa 2017



Services



## OPENSIFT



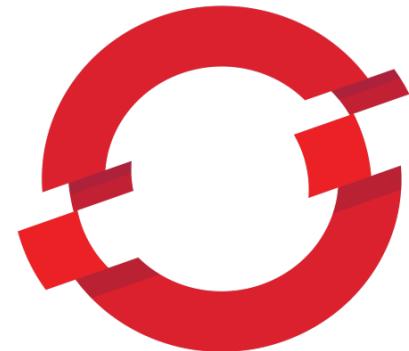
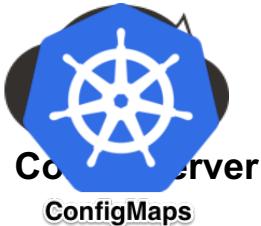
@kamesh\_sampath



# Better Spring Boot Microservices Platform circa 2018



Services



## OPENSIFT



Istio



@kamesh\_sampath

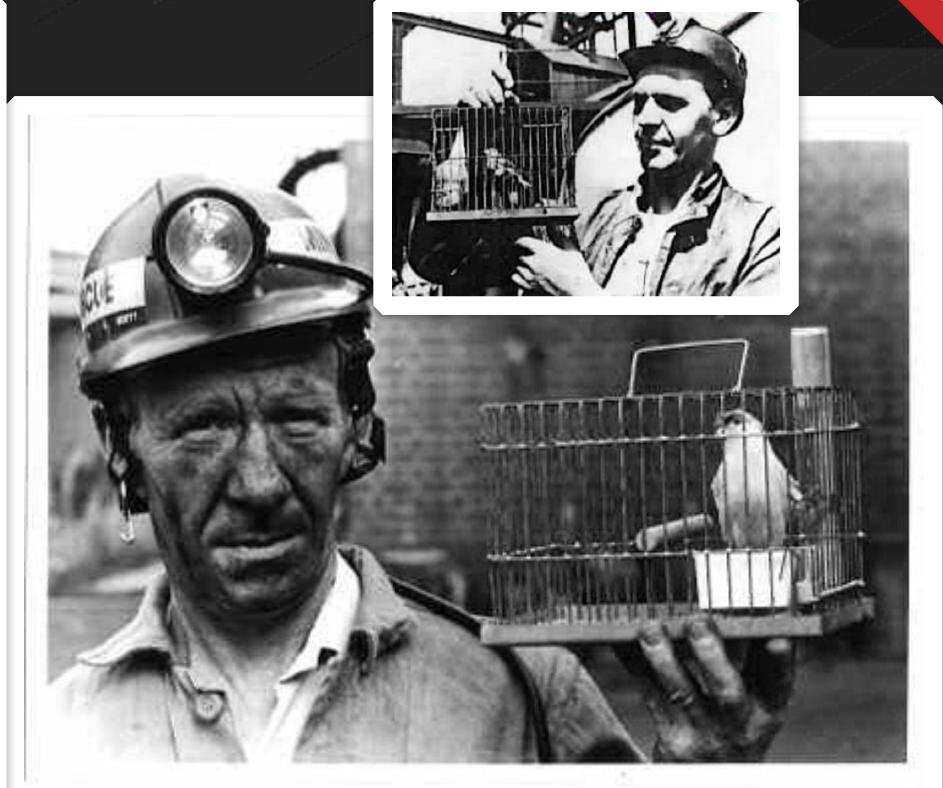


# Microservice Architecture - Principles

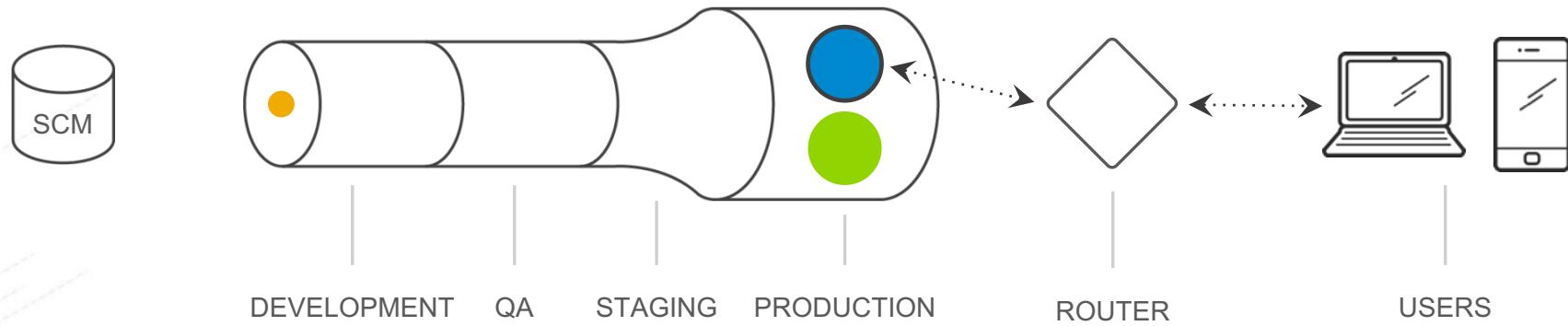
## Canary Release

Canary release is a technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users before rolling it out to the entire infrastructure and making it available to everybody.

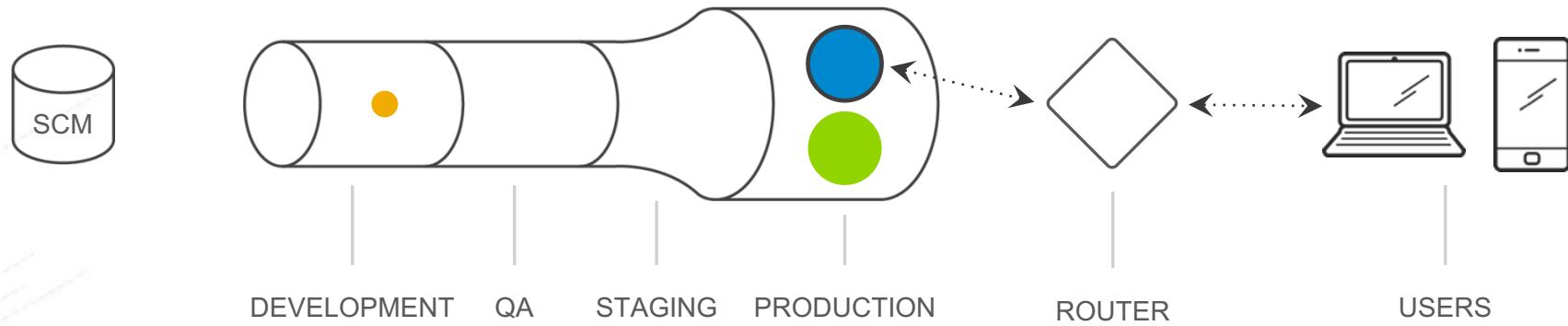
Source: <https://martinfowler.com/bliki/CanaryRelease.html>



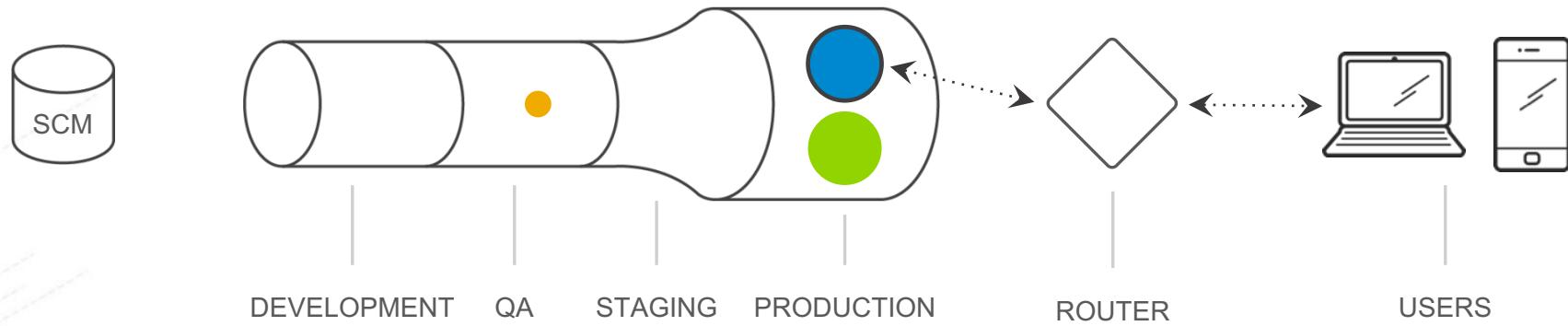
# Canary Deployment



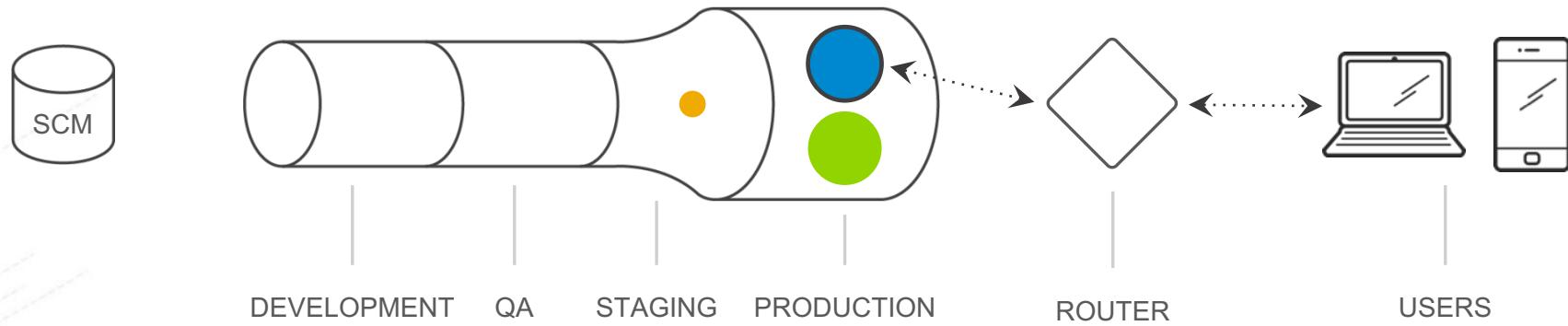
# Canary Deployment



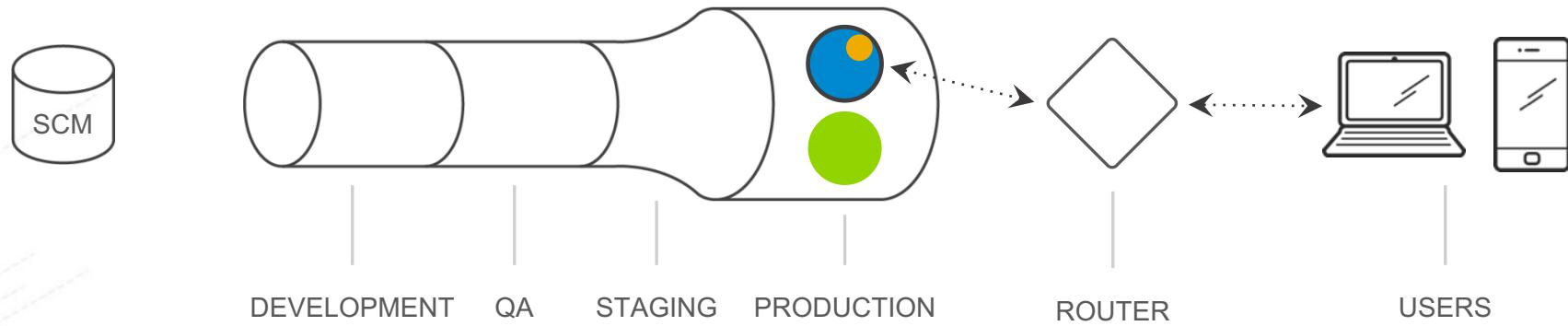
# Canary Deployment



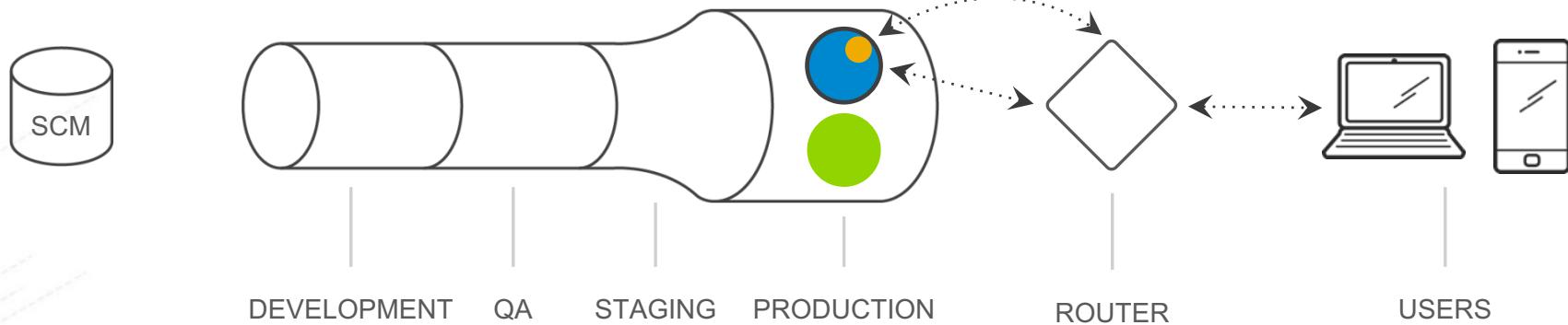
# Canary Deployment



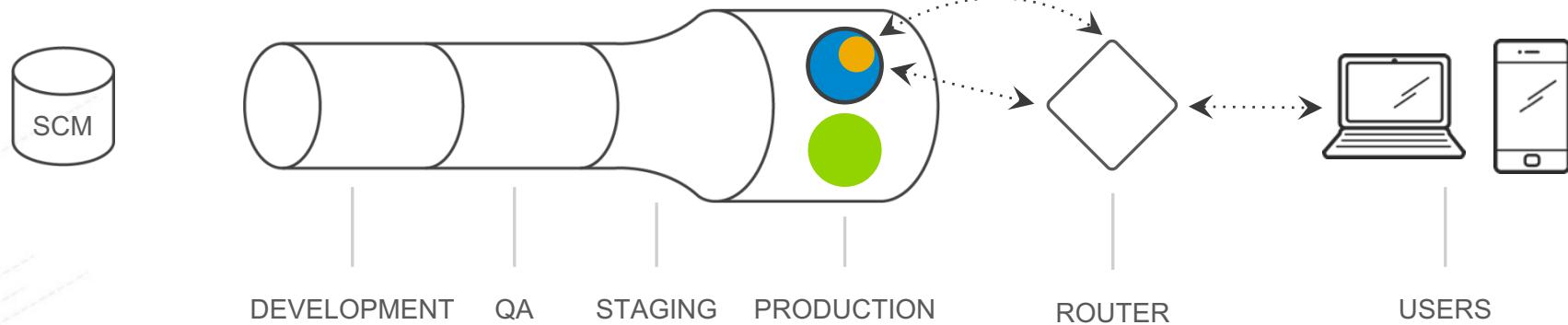
# Canary Deployment



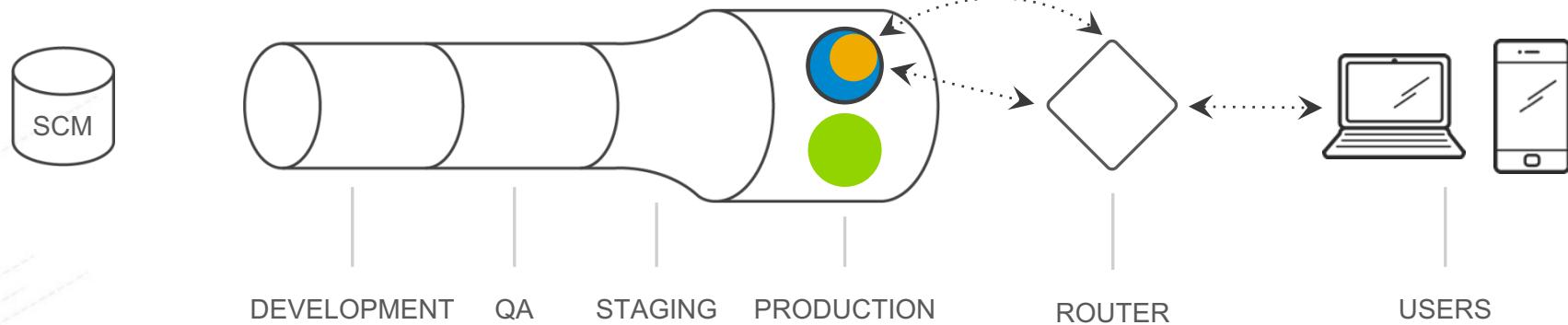
# Canary Deployment



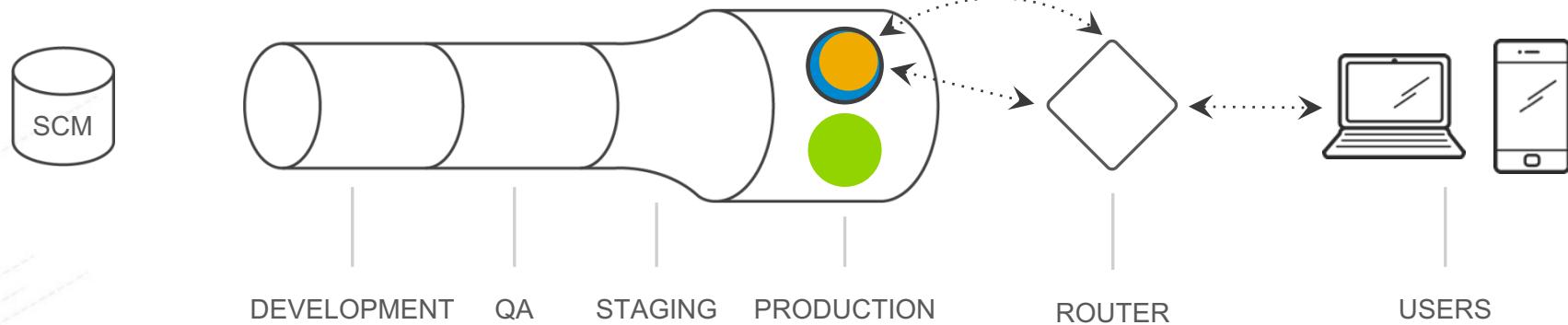
# Canary Deployment



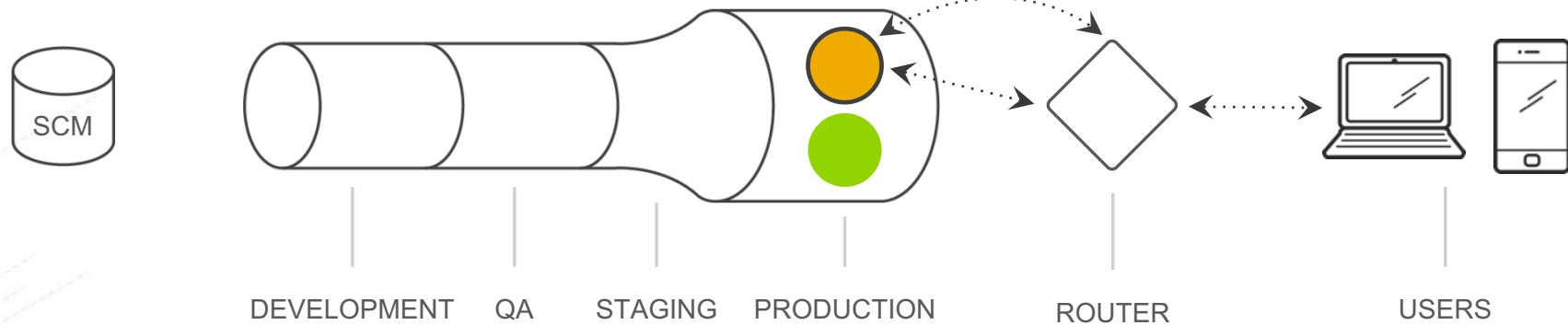
# Canary Deployment



# Canary Deployment



# Canary Deployment



# Demo

# Summary



@kamesh\_sampath

- Minishift and Minikube
- Fabric8 Maven Plugin
- Eureka & Ribbon not required with Kubernetes
- Demos
  - [bit.ly/springboot-on-openshift](http://bit.ly/springboot-on-openshift)
  - Spring Cloud Brewery Application Migration
    - [bit.ly/green-cloud-demo](http://bit.ly/green-cloud-demo)
    - [bit.ly/green-cloud-demo-sources](http://bit.ly/green-cloud-demo-sources)
  - Stateful Canary leveraging Infinispan
    - [bit.ly/popdemo](http://bit.ly/popdemo)
- Open Containers Initiative
  - Runtime Spec - <http://cri-o.io/>
  - Image Spec – <https://github.com/opencontainers/image-spec>



# RED HAT DEVELOPERS



@kamesh\_sampath



kamesh.sampath@hotmail.com

[bit.ly/springboot-on-openshift](https://bit.ly/springboot-on-openshift)