



RED HAT DEVELOPERS

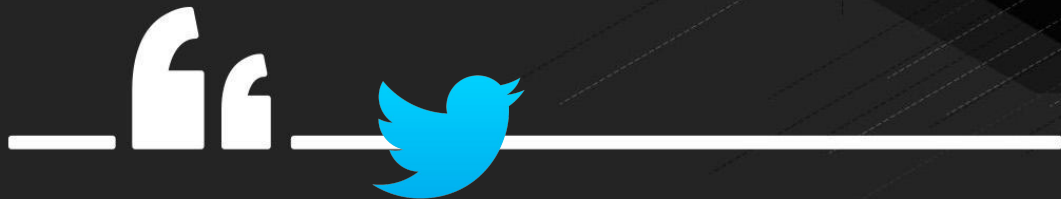
Cloud Native Java with Kubernetes

 @burrsutter

 burr@redhat.com



developers.redhat.com



We cannot solve our problems
with the same thinking we used
when we created them.

Albert Einstein (Theoretical
Physicist)

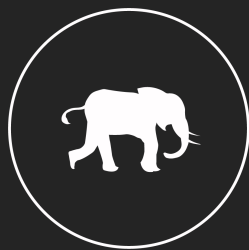


Digital Darwinism

The Developer's Journey



Re-Org to
DevOps



Self-Service,
On-Demand,
Elastic
Infrastructure



Automation
Puppet, Chef,
Ansible,
Kubernetes



CI & CD
Deployment
Pipeline



Advanced
Deployment
Techniques

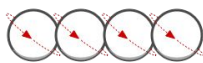


Microservices
(and flying
elephants!)

Our IT World Morphs

Development Process

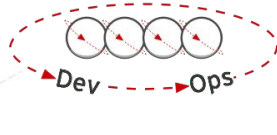
Waterfall



Agile

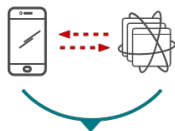


DevOps



Application Architecture

Monolithic



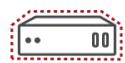
N-Tier

Microservices



Deployment & Packaging

Physical Servers



Virtual Servers

Containers



Application Infrastructure

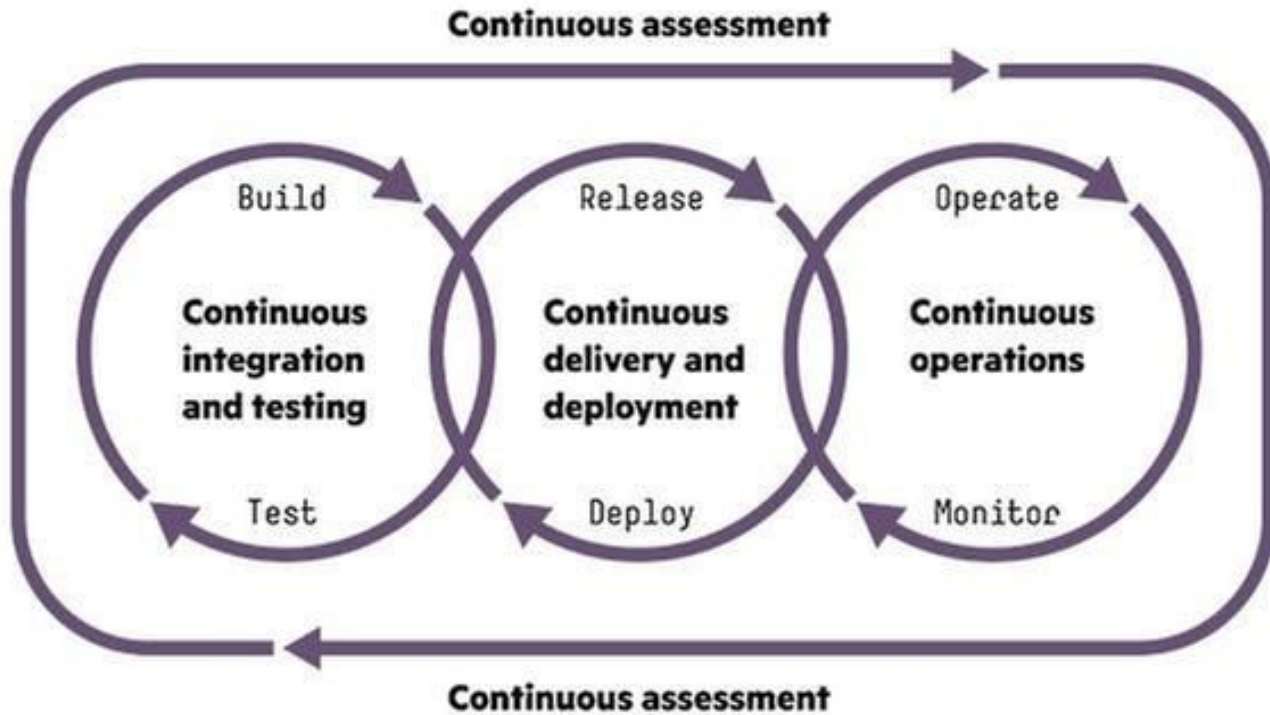
Datacenter



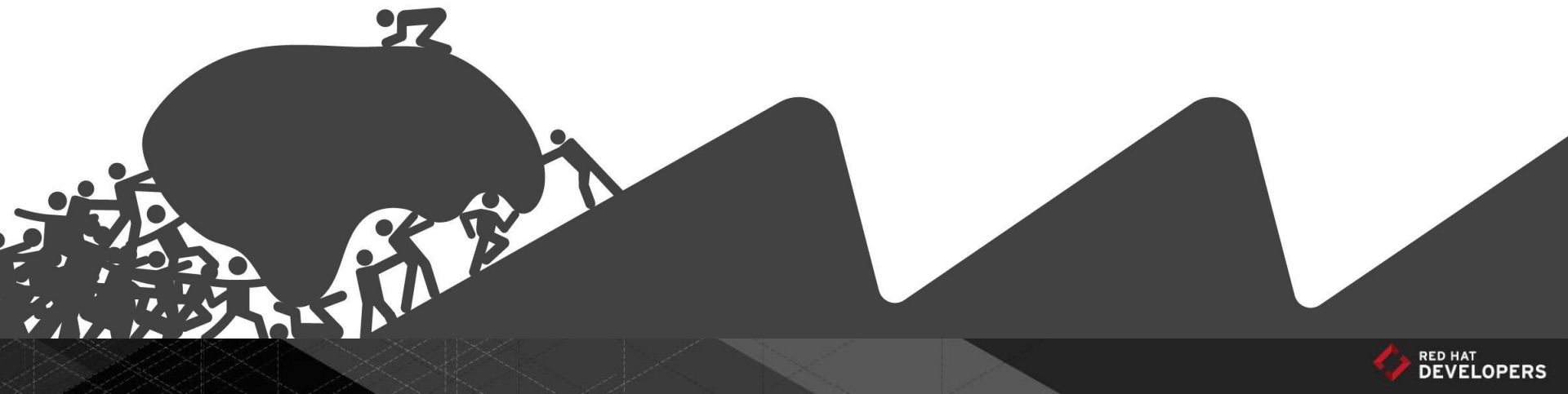
Hosted

Cloud











java -jar myapp.jar

DropWizard

www.dropwizard.io

JAX-RS API

First to market

DropWizard Metrics

Embeddable
servers:
Jetty



Spring Boot

projects.spring.io/spring-boot

Spring API
(@RestController)

'Starter' POMs:
start.spring.io

Embeddable servers:
Tomcat, Jetty, Undertow



WildFly Swarm

wildfly-swarm.io

Java EE 7 APIs

'Starter' POMs:
wildfly-swarm.io/generator

Embeddable
servers:
WildFly (Undertow)



Vert.x

vertx.io

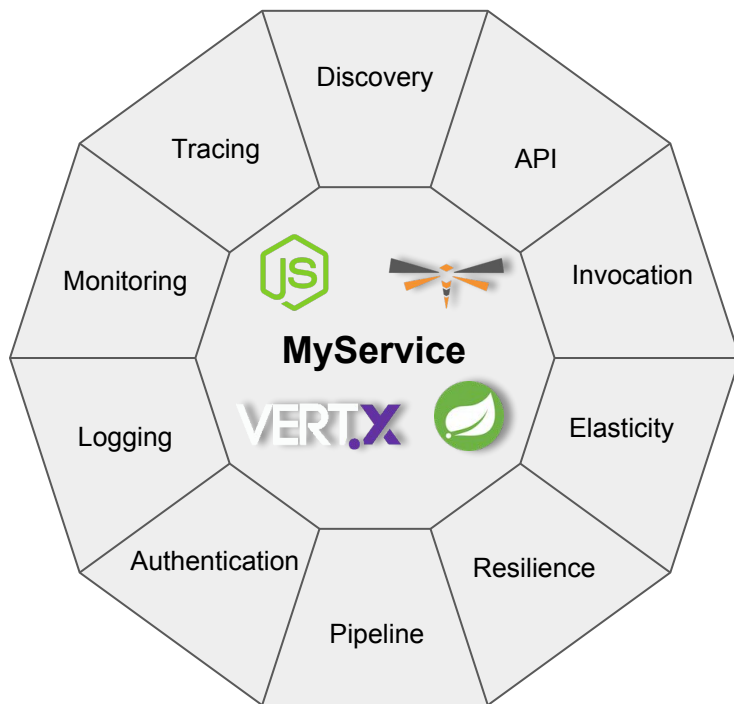
Reactive
Async/non-blocking

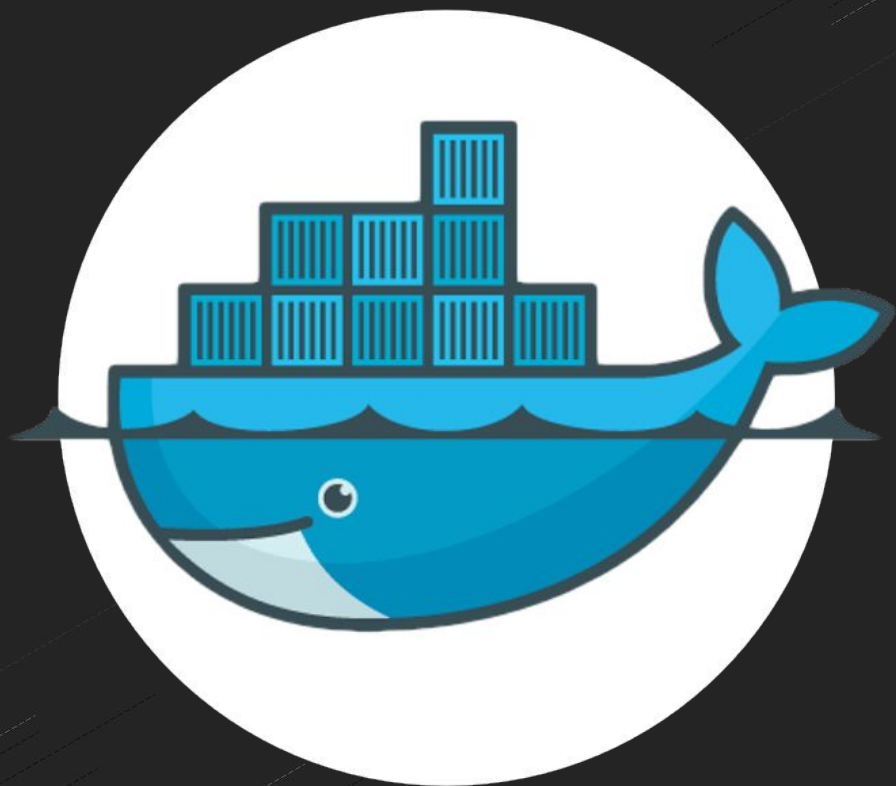
vertx run myhttp.java

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



Microservices Properties





Your Stack Matters

your.war
Custom Configuration
Application Server
Java Virtual Machine
Operating System
Server Hardware

JDBC driver, datasource, JMS queue, users

Weblogic 10.x.y, Tomcat 6.x.y, JBoss EAP 6.x.y

Java 1.6.6_45 or Java 1.7.0_67

Linux Kernel Version & Distribution

DevOps Challenges for Multiple Containers

- How to scale?
- How to avoid port conflicts?
- How to manage them on multiple hosts?
- What happens if a host has trouble?
- How to keep them running?
- How to update them?
- Where are my containers?



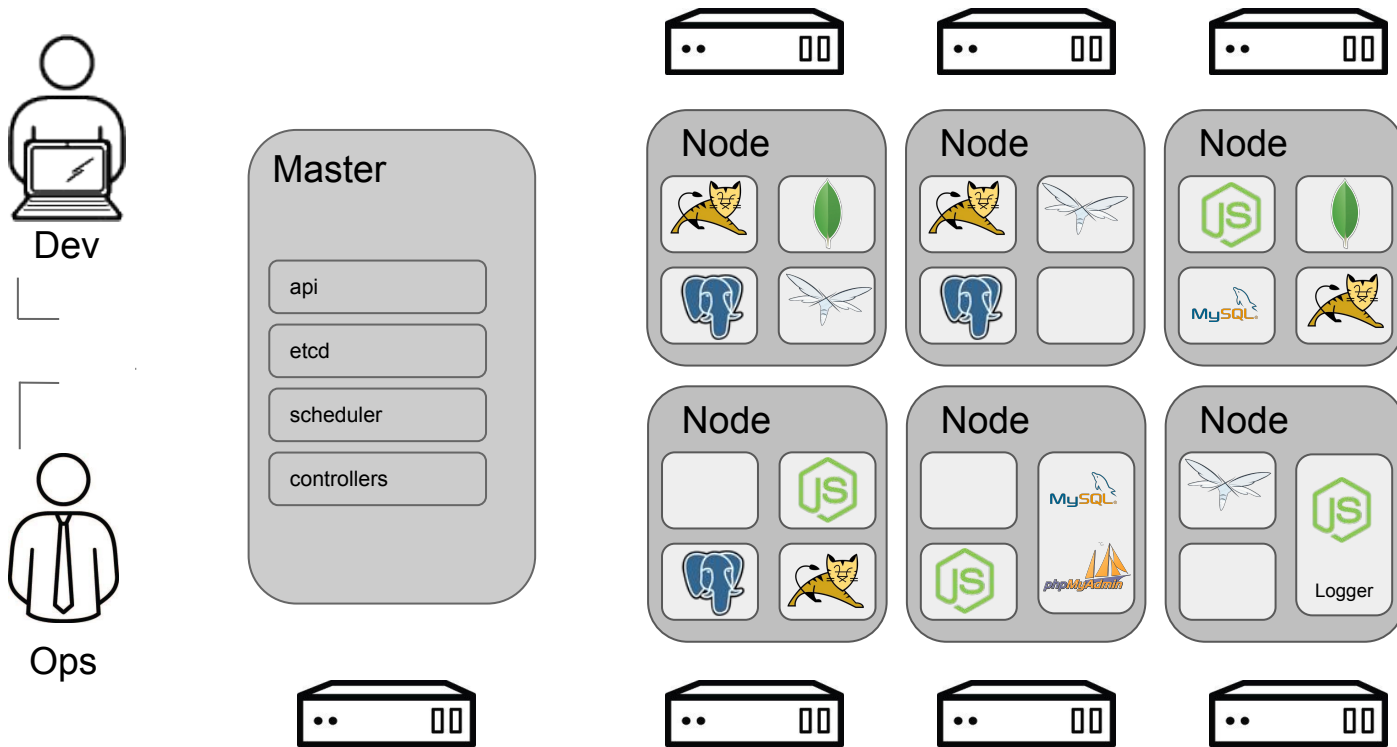


Meet Kubernetes

- 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

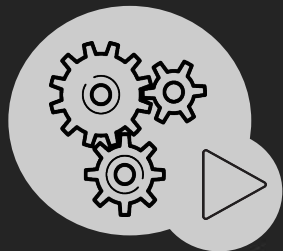


Kubernetes Cluster



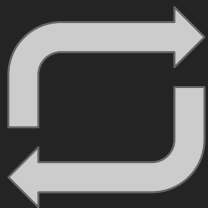
Kubernetes Cluster

Pod



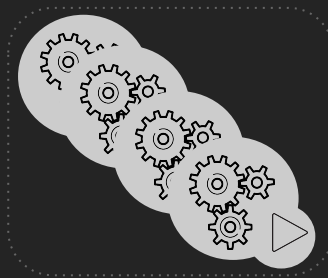
- ✓ 1+ containers
- ✓ Shared IP
- ✓ Shared storage volume
- ✓ Shared resources
- ✓ Shared lifecycle

Replication
Controller/
Deployment



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

Service



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

Label



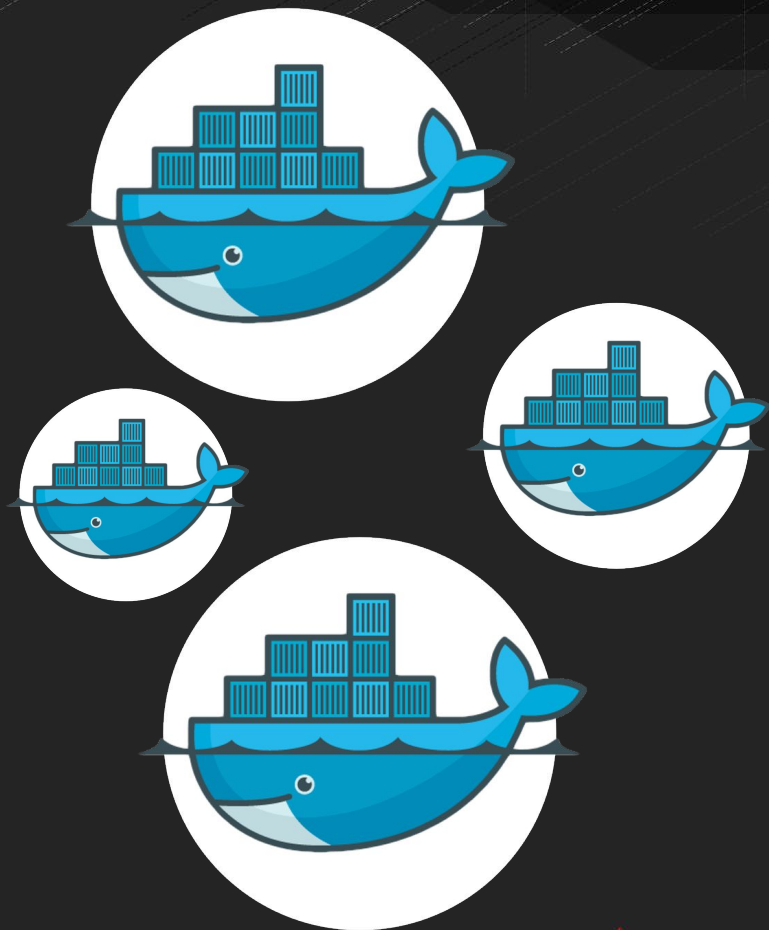
- ✓ Key/Value pairs associated with Kubernetes objects (env=production)

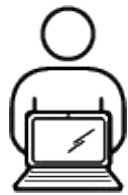
Pods

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/>



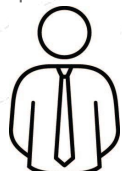


Dev

SCM
(Git/Svn)

CI/CD

Automation



Ops

Registry

Master

API Server

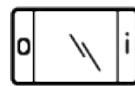
Kubernetes

OpenShift

- Builds
- ImageStreams

Controllers

- Scheduler
- Deployments
- Services
- Builds
- Routes
- DeploymentConfigs



Routing Layer

Node



Node



Node



SDN Overlay Network

Node



Node



Node



Service Layer



Physical



Virtual



Private



Public



Persistent
Storage





OPENSHIFT

openshift.org



OPENSIFT

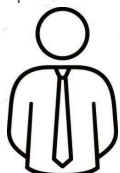


Dev

SCM
(Git/Svn)

CI/CD

Automation



Ops

Registry

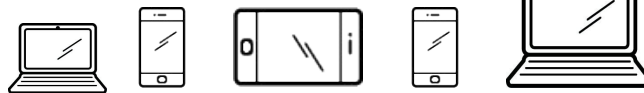
Master

API Server

Kubernetes

OpenShift
- Builds
- ImageStreams

Controllers
- Scheduler
- Deployments
- Services
- Builds
- Routes
- DeploymentConfigs



Routing Layer

Node



Node



Node



Node



Node



Node



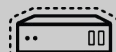
Persistent
Storage



Service Layer



Physical



Virtual



Private



Public



OPENSIFT

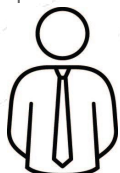


Dev

SCM
(Git/Svn)

CI/CD

Automation



Ops

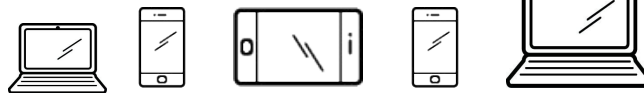
Registry

Master

API Server
Kubernetes

OpenShift
- Builds
- ImageStreams

Controllers
- Scheduler
- Deployments
- Services
- Builds
- Routes
- DeploymentConfigs



Routing Layer

Node



Node



Node



SDN Overlay Network

Node



Node



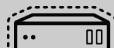
Node



Service Layer



Physical



Virtual



Private



Public



Persistent
Storage





OPENSIFT

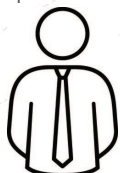


Dev

SCM
(Git/Svn)

CI/CD

Automation



Ops

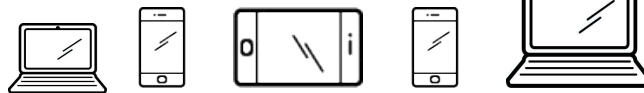
Registry

Master

API Server
Kubernetes

OpenShift
- Builds
- ImageStreams

Controllers
- Scheduler
- Deployments
- Services
- Builds
- Routes
- DeploymentConfigs



Routing Layer

Node



Node



Node



SDN Overlay Network

Node



Node



Node

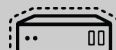


Logger

Service Layer



Physical



Virtual



Private



Public



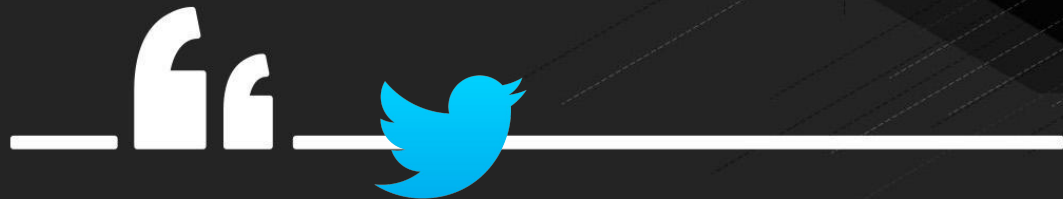
Persistent
Storage



Kubernetes Demo

<https://github.com/burrsutter/kube4docker>

Recorded Demo: <https://youtu.be/AoDhQt8PtUQ>



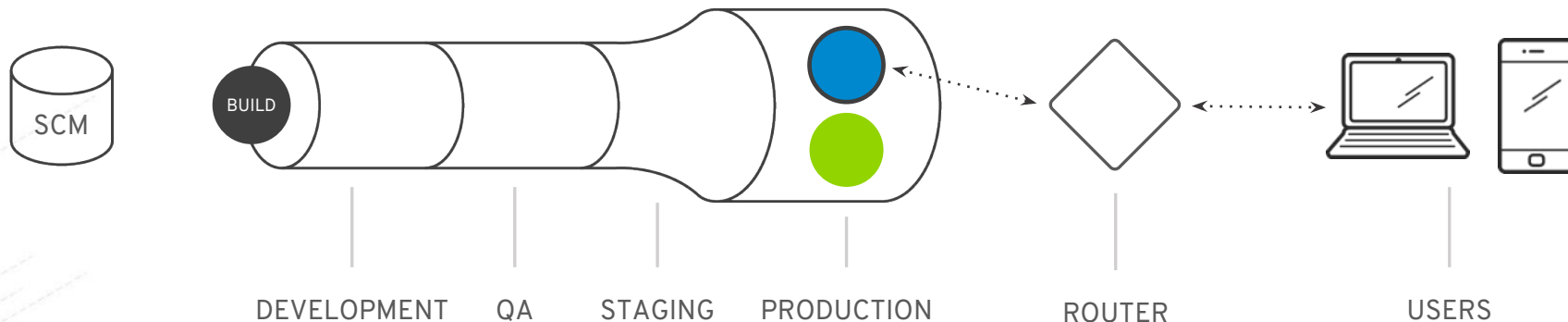
The job of a deployment pipeline
is to prove that the release
candidate is unreleasable.

Jez Humble





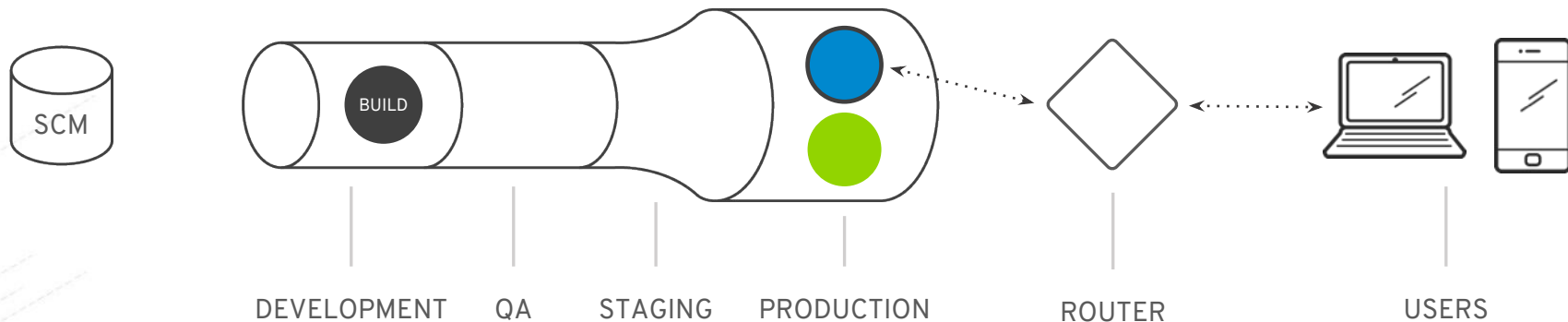
Blue/Green Deployment



Starts with a “git commit and git push”

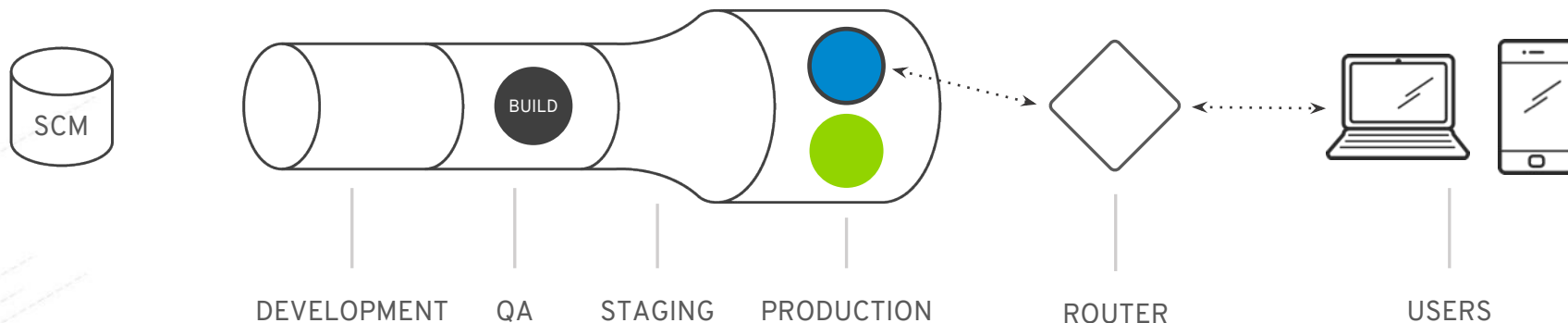


Blue/Green Deployment



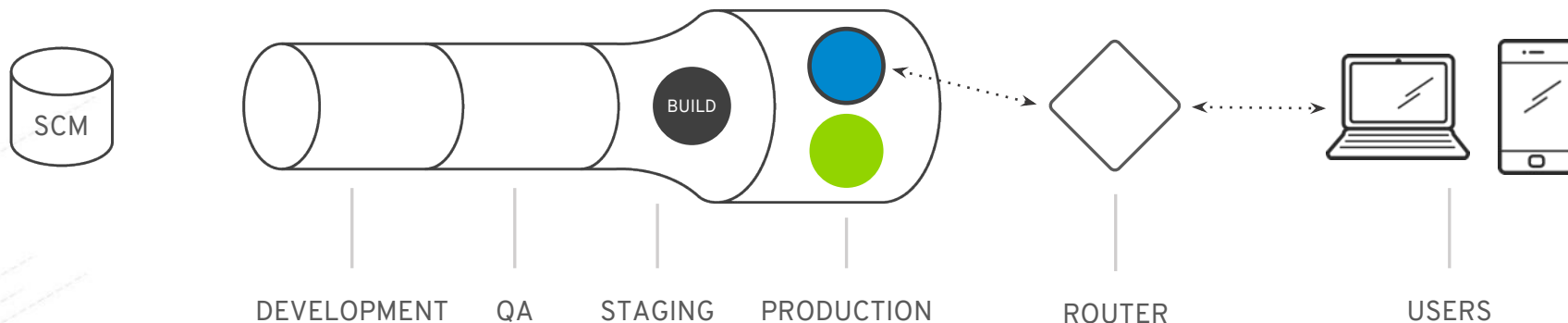


Blue/Green Deployment



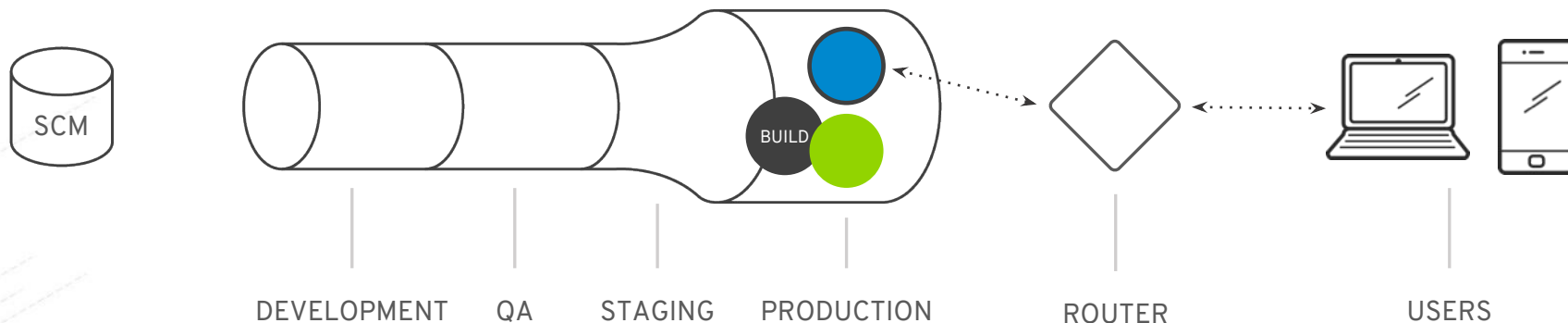


Blue/Green Deployment



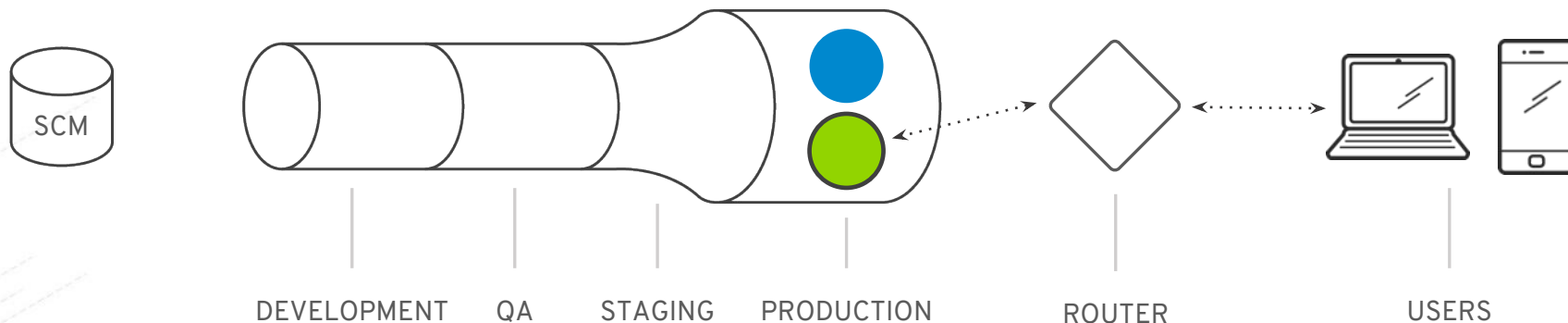


Blue/Green Deployment



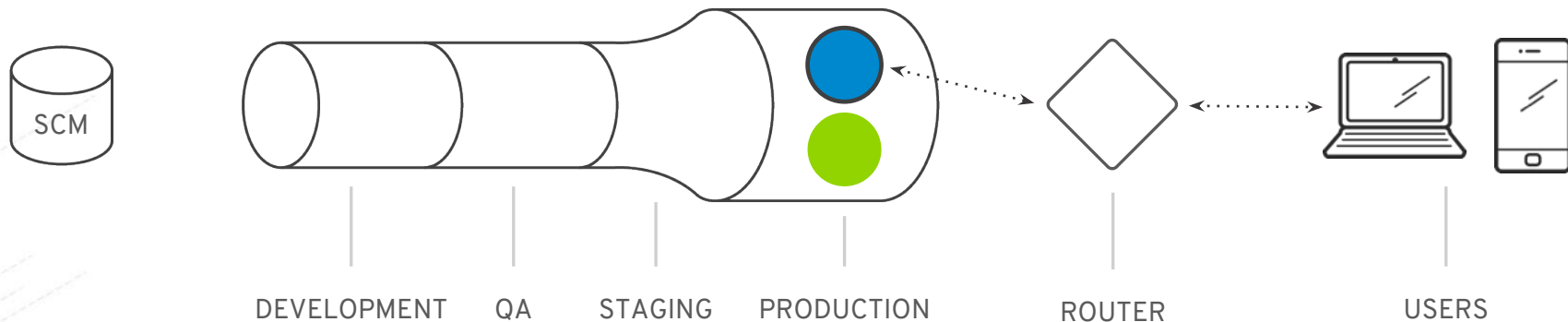


Blue/Green Deployment



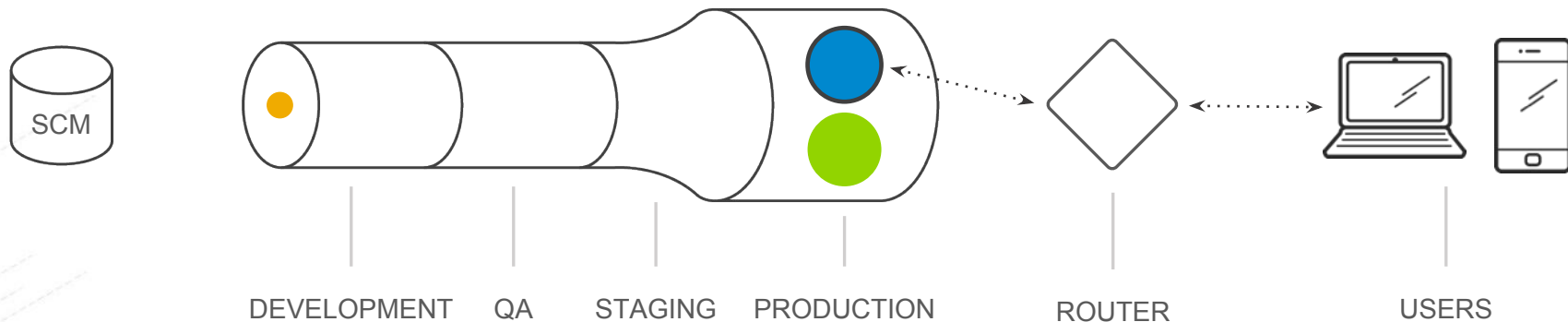


Blue/Green Deployment

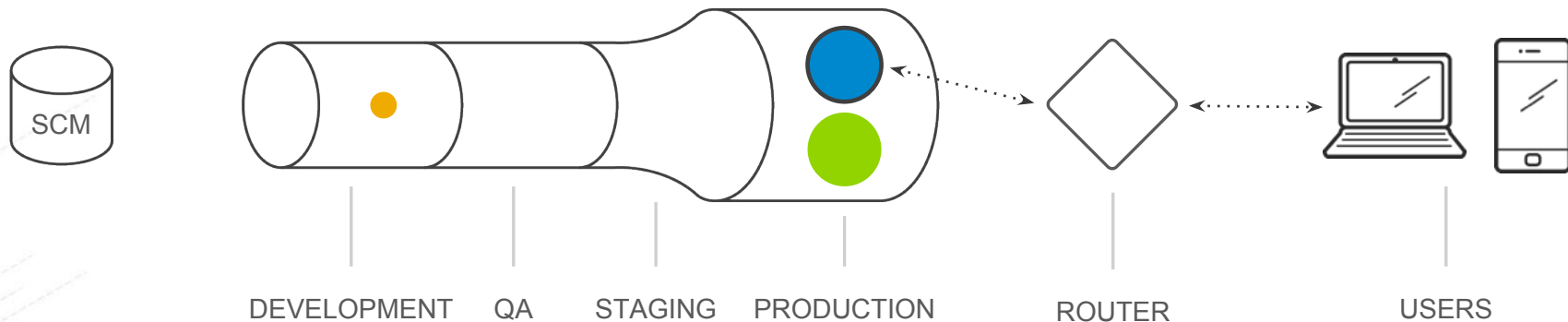




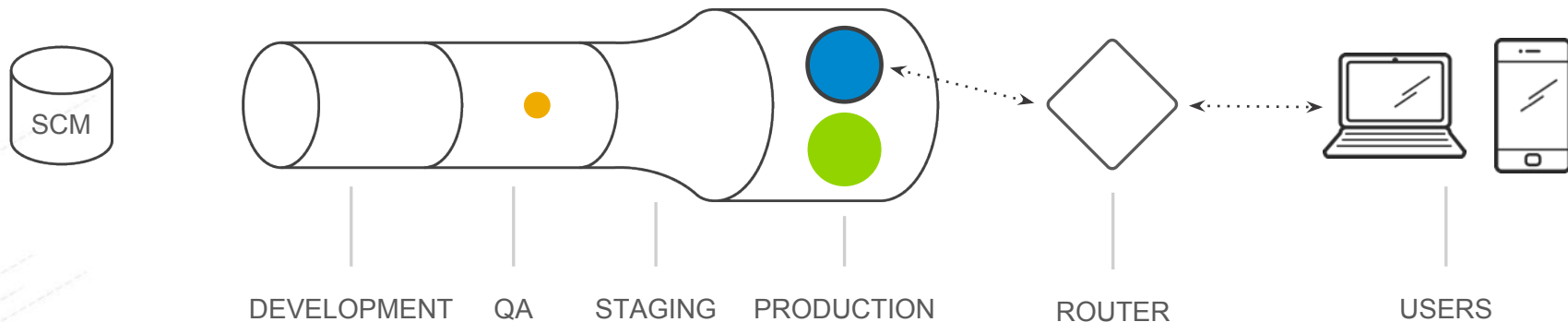
Canary Deployment



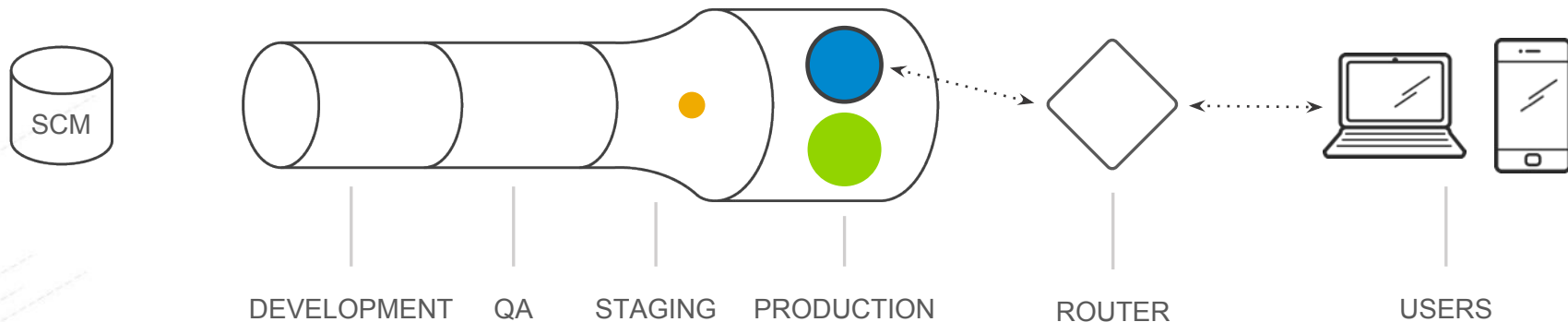
Canary Deployment



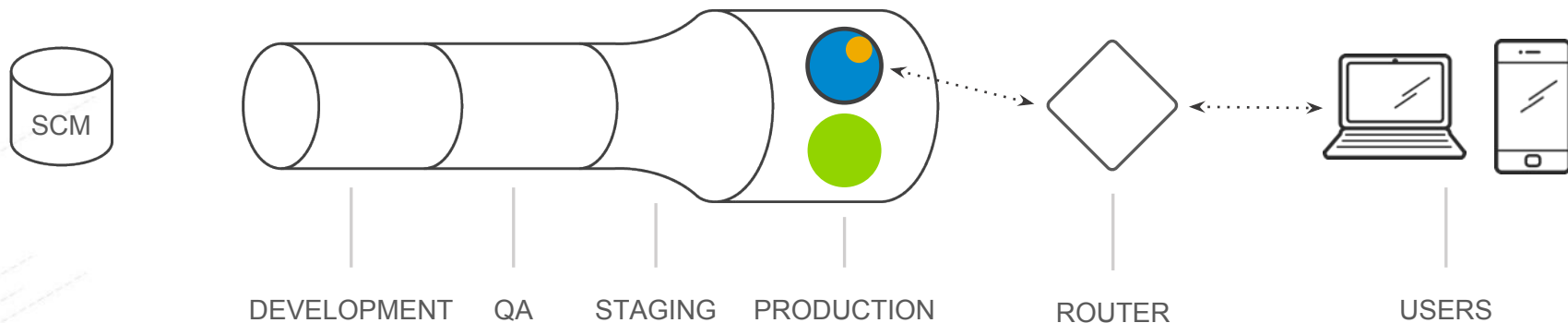
Canary Deployment



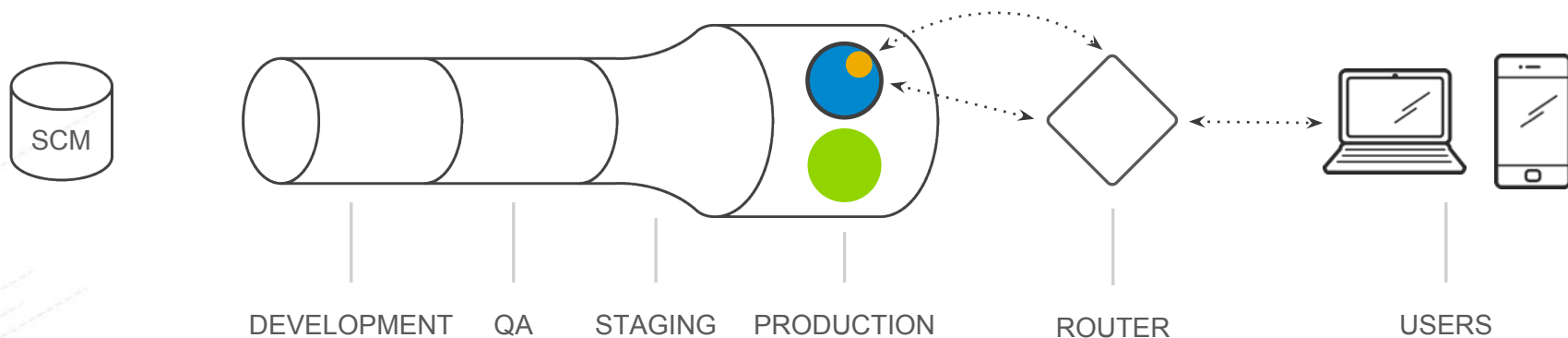
Canary Deployment



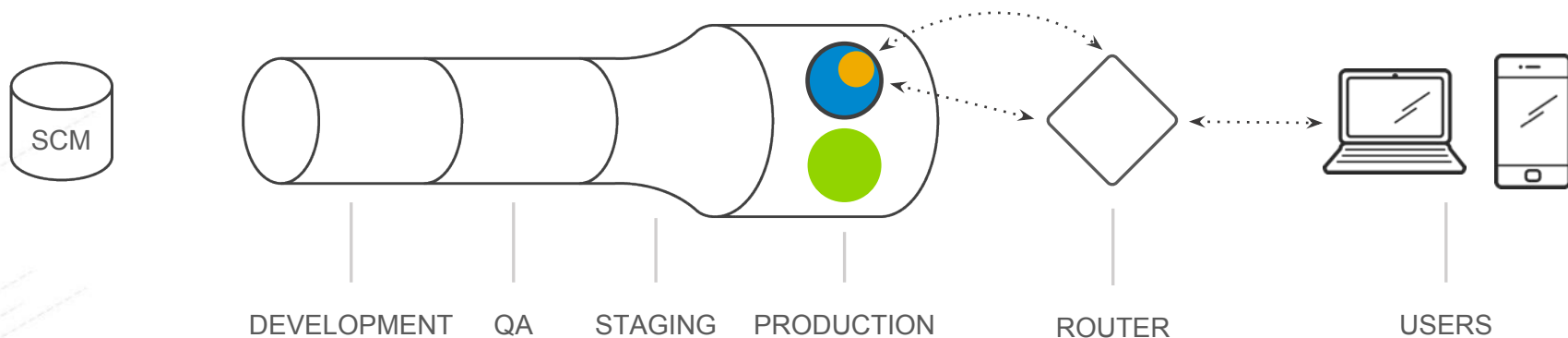
Canary Deployment



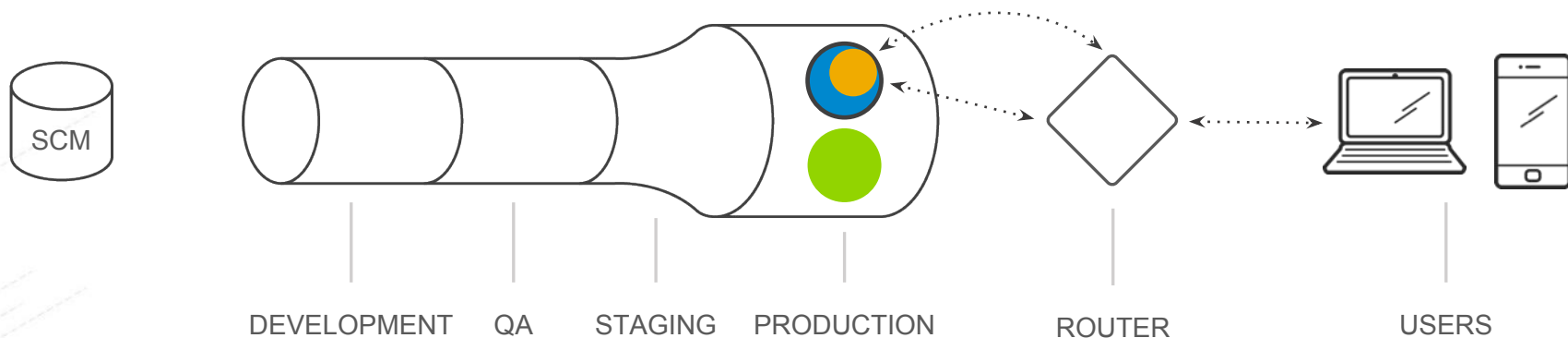
Canary Deployment



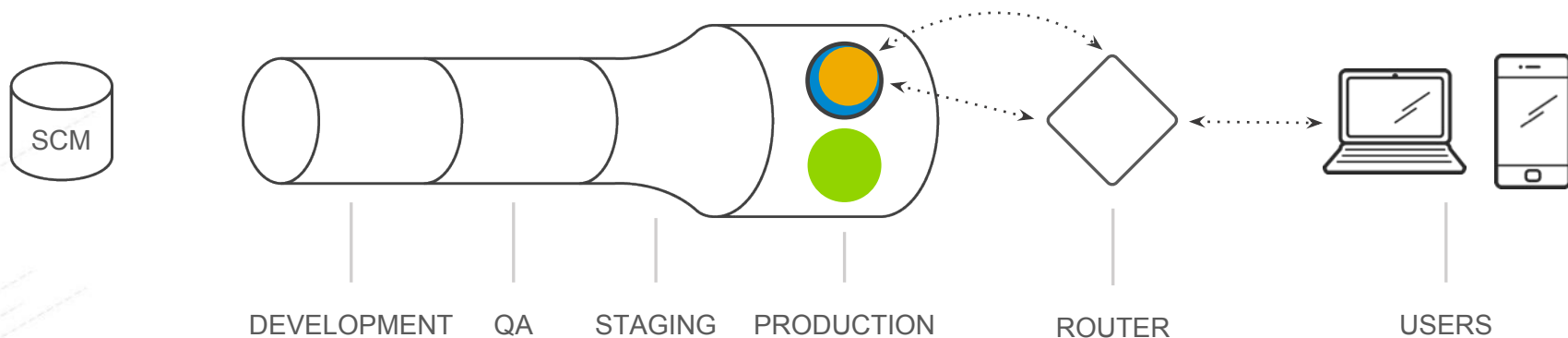
Canary Deployment



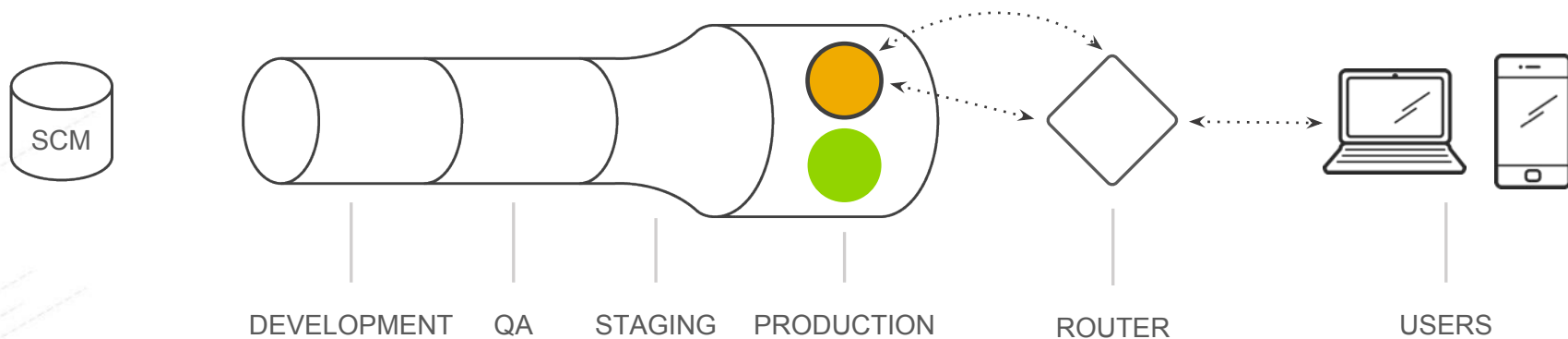
Canary Deployment



Canary Deployment



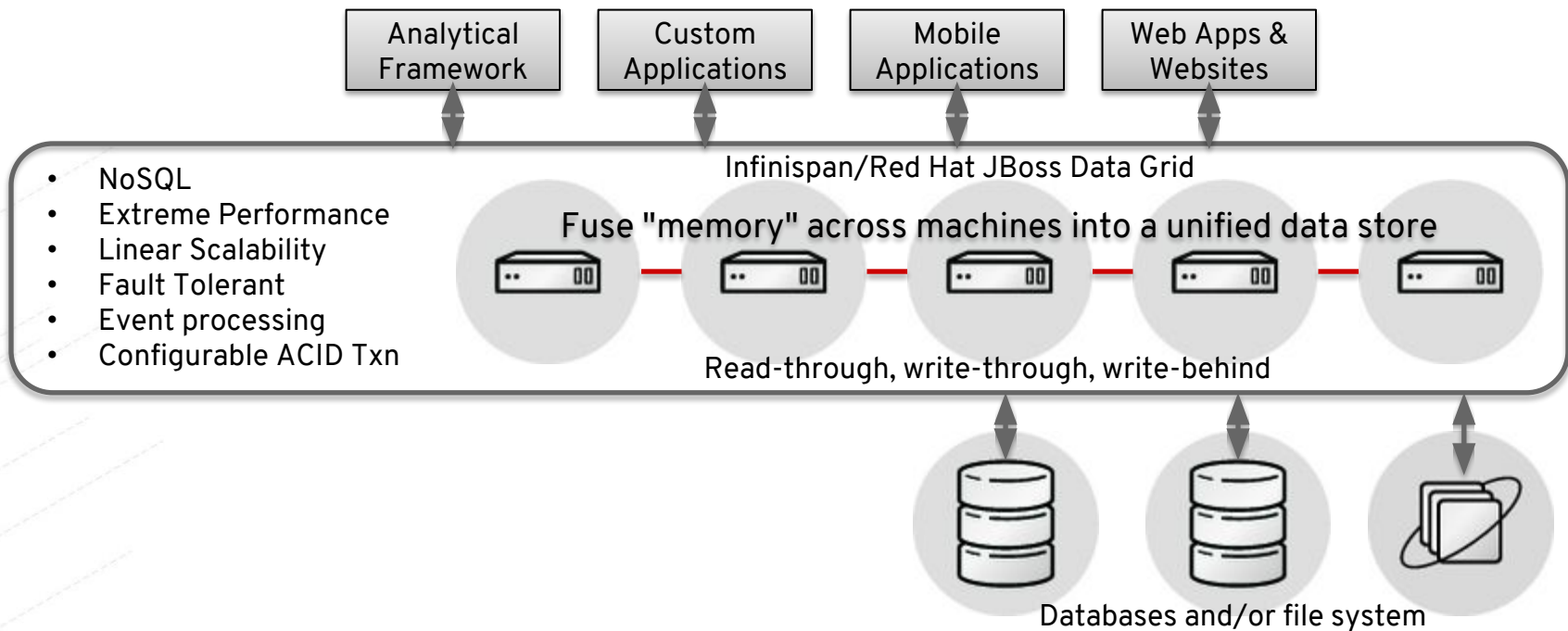
Canary Deployment



Infinispan.org

IN-MEMORY DATA = All data needed is kept in memory

GRID = Too big for one node, data is scaled-out in cluster

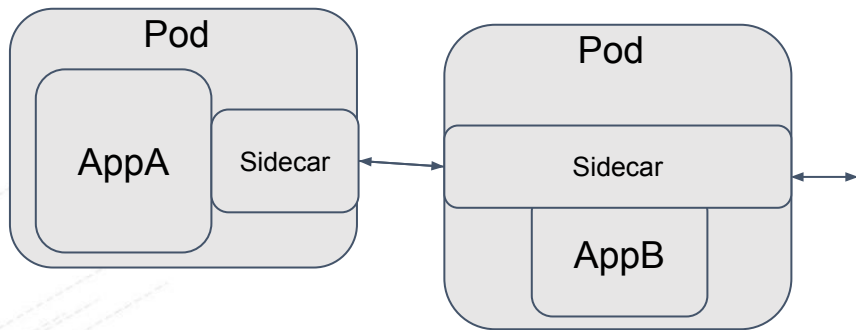


Demo

Sidecars



Sidecars



C++



Scala
CNCF



Go



Go
Consul

<https://news.ycombinator.com/item?id=14194026>

Summary



- Minishift and Minikube
- Fabric8 Maven Plugin
- Eureka & Ribbon not required with Kubernetes
- Sidecars: Envoy, Linkerd, Traefik.io, etc.
- Microservices, Blue/Green, CI/CD & Canary bit.ly/msa-instructions
- Liveness and Readiness Probes
- Stateful Canary leveraging Infinispan
- https://github.com/burrsutter/devoxxUK17_kubernetes



RED HAT DEVELOPERS THANK YOU!

<http://bit.ly/msa-instructions>

 @burrsutter