Spring Boot Apps on Kubernetes

Thomas Risberg

Pivotal

#Devoxx #SpringBootK8s

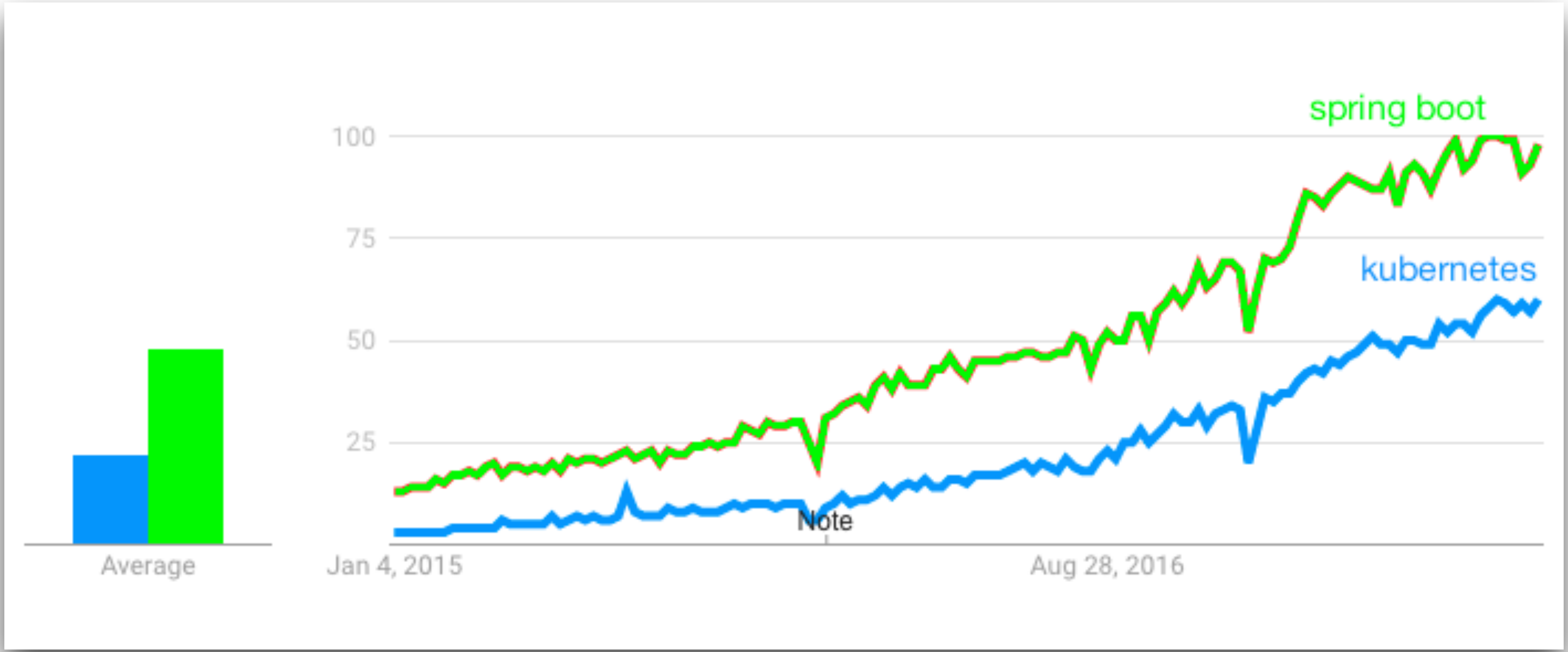
@trisberg

About me

Thomas Risberg (@trisberg)

- Member of the Spring engineering team at Pivotal
- Contributing to Spring Cloud Data Flow, Spring Cloud Deployer for Kubernetes projects
- Joined the Spring Framework open source project in 2003 working on JDBC support

Two Hot Technologies



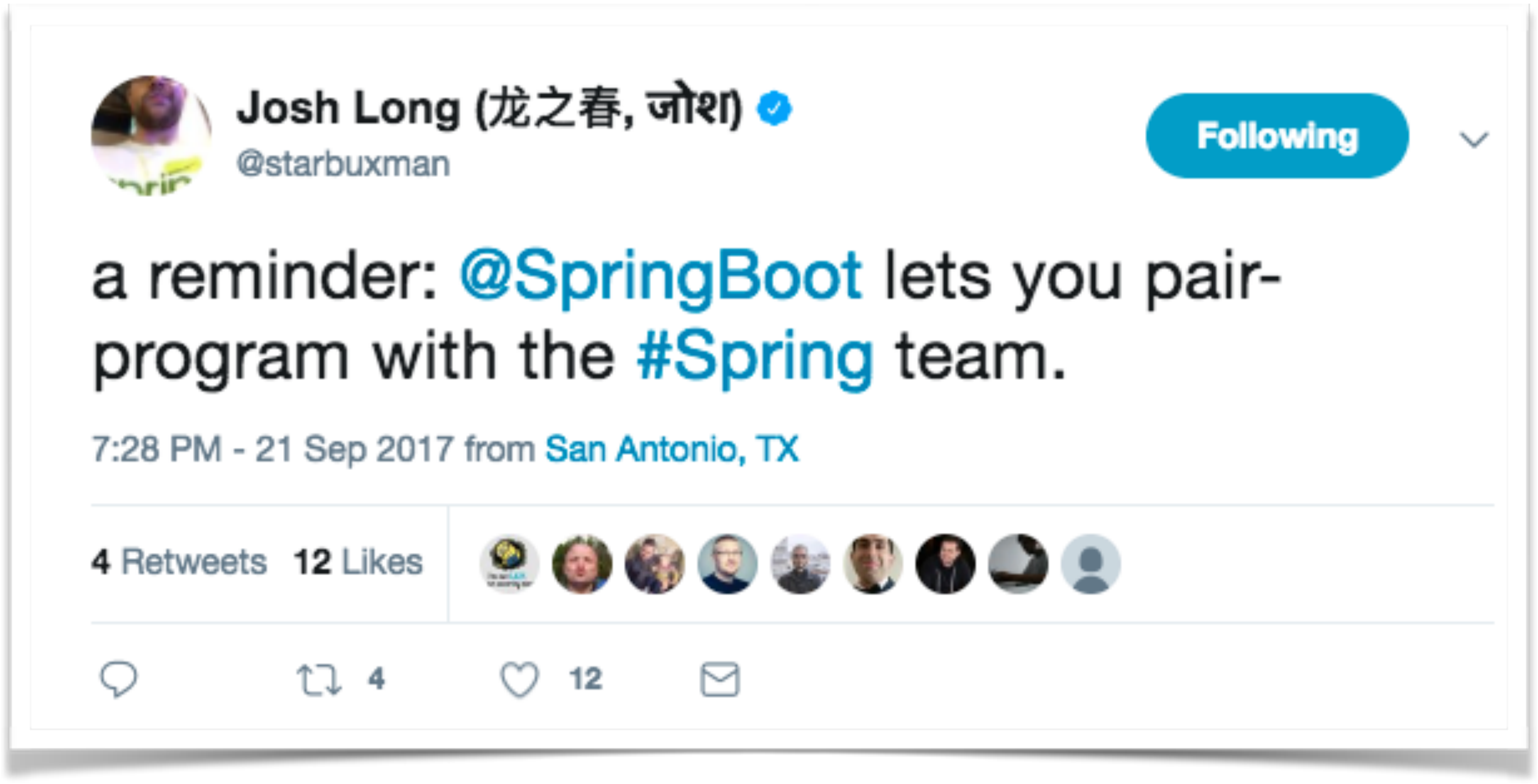
Based on: https://trends.google.com/trends/explore?q=kubernetes,spring%20boot

What is Spring Boot?

> *Spring Boot* takes an opinionated view of building production-ready Spring applications. Spring Boot favors convention over configuration and is designed to get you up and running as quickly as possible.

http://projects.spring.io/spring-boot/

Pair programming with Spring Team



Josh Long (龙之春, जोश) @starbuxman

a reminder: @SpringBoot lets you pair-program with the #Spring team.

7:28 PM - 21 Sep 2017 from San Antonio, TX

4 Retweets 12 Likes

What is Kubernetes?

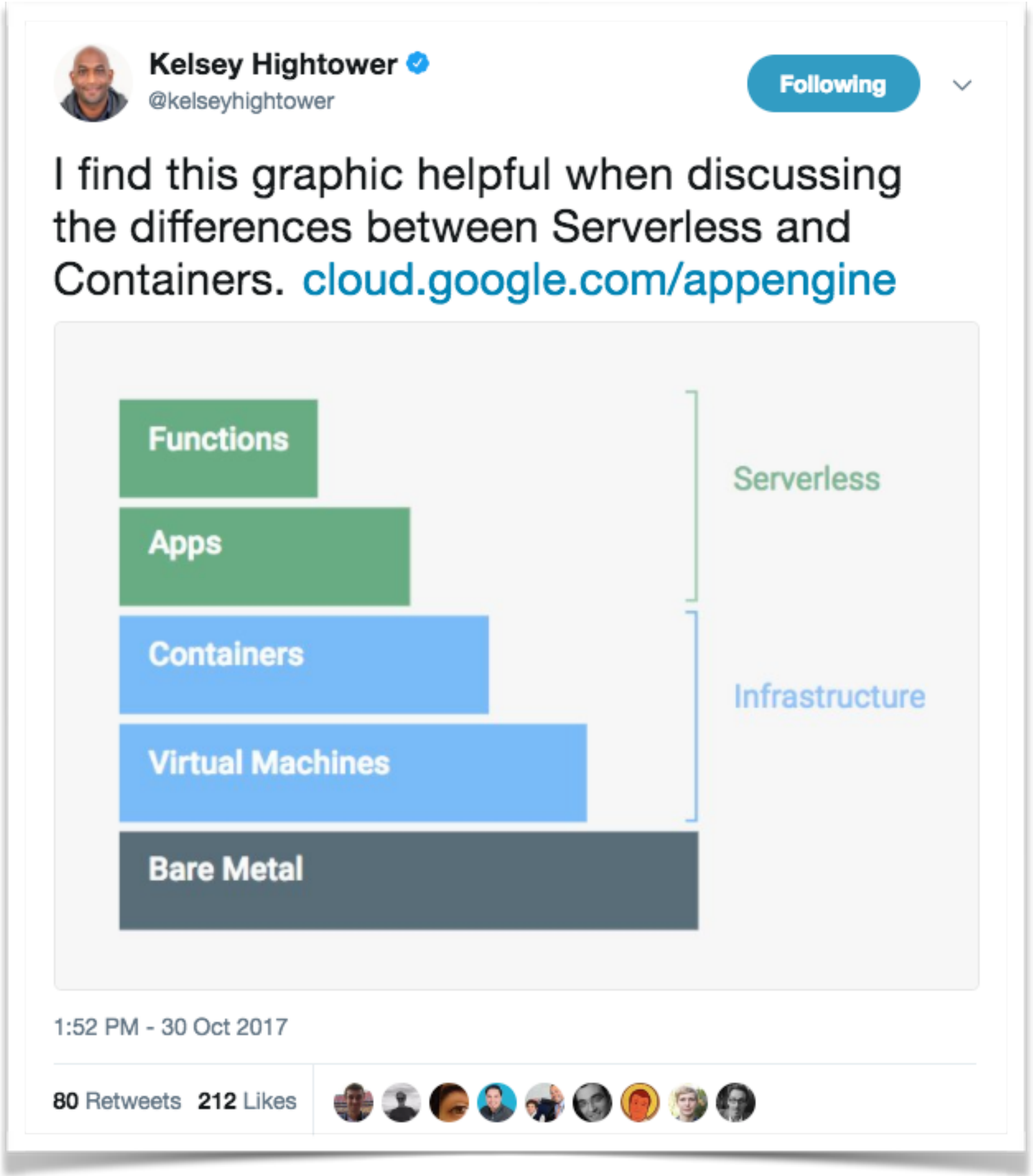
Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

https://kubernetes.io/

Pairing with Google Engineering Team?

Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the community.

Serverless, Apps and Containers?



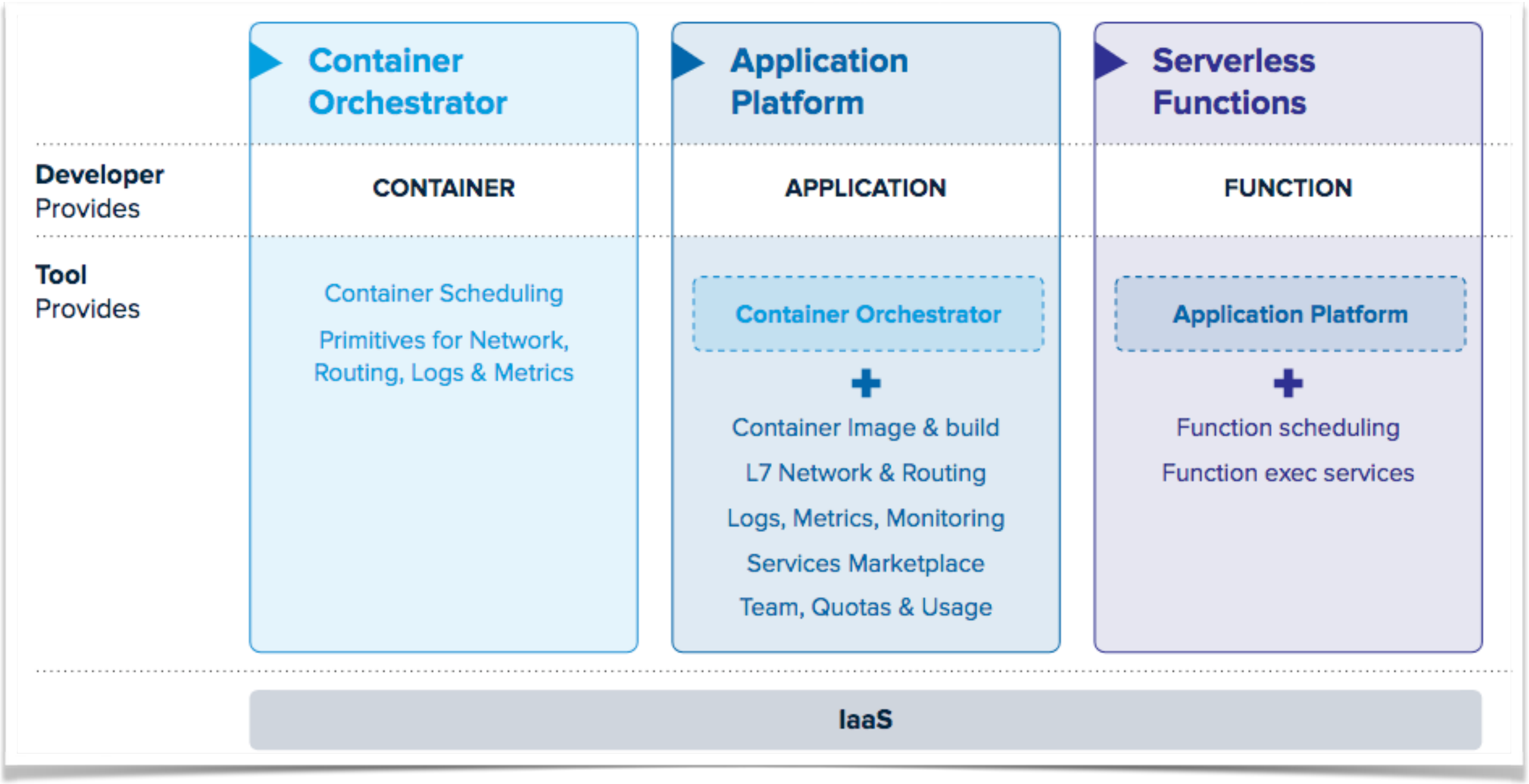
Kelsey Hightower
@kelseyhightower
Following

I find this graphic helpful when discussing the differences between Serverless and Containers. cloud.google.com/appengine

1:52 PM - 30 Oct 2017

80 Retweets 212 Likes

Choose the right tool for the job



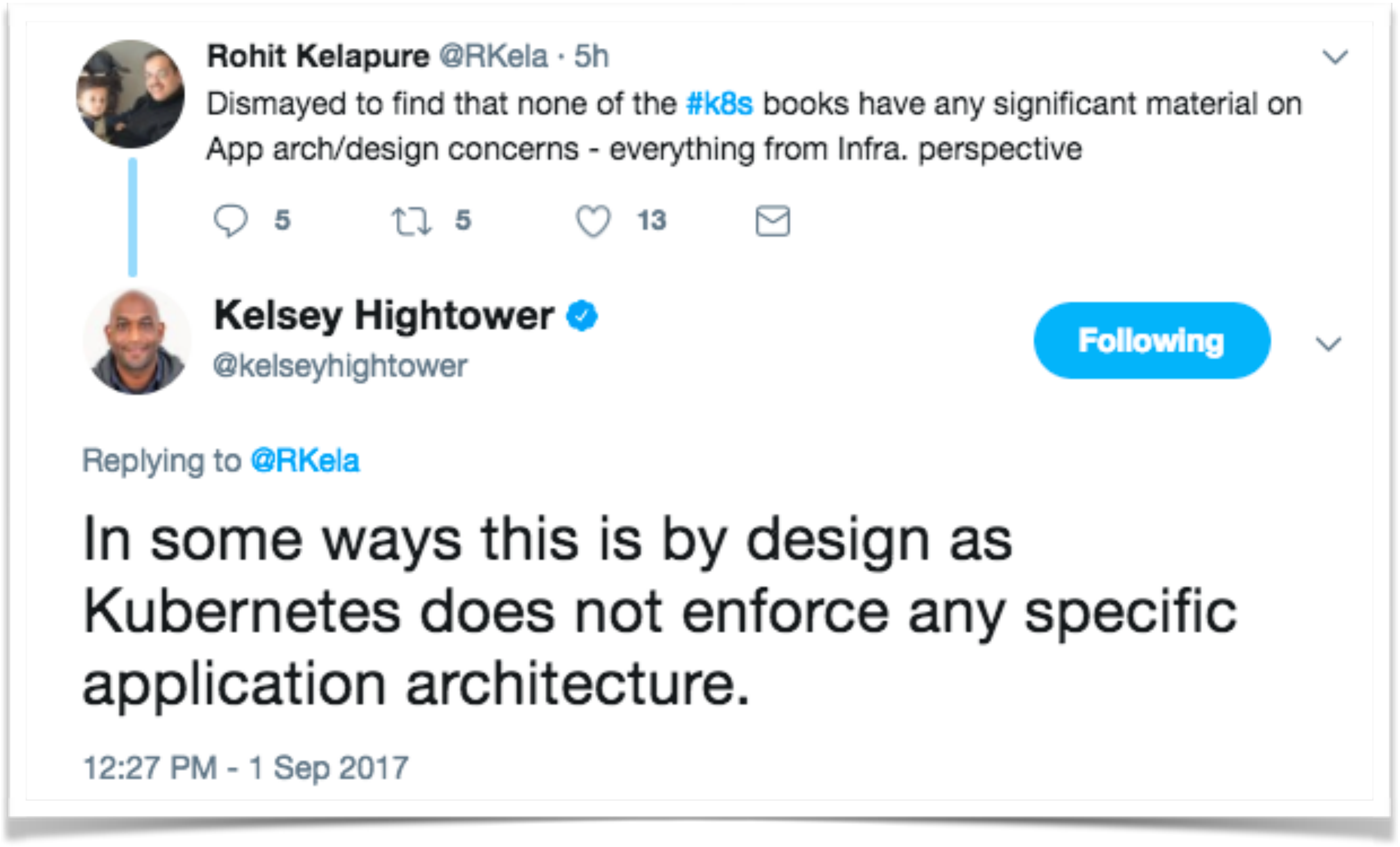
	Container Orchestrator	Application Platform	Serverless Functions
Developer Provides	CONTAINER	APPLICATION	FUNCTION
Tool Provides	Container Scheduling Primitives for Network, Routing, Logs & Metrics	Container Orchestrator + Container Image & build L7 Network & Routing Logs, Metrics, Monitoring Services Marketplace Team, Quotas & Usage	Application Platform + Function scheduling Function exec services
	IaaS		

Demo

Simple Hello
Spring Boot/Kubernetes
app deployment

https://github.com/trisberg/devoxx-spring-boot-k8s/blob/master/demo-hello.adoc

Building Apps for Kubernetes



Rohit Kelapure @RKela · 5h

Dismayed to find that none of the #k8s books have any significant material on App arch/design concerns - everything from Infra. perspective

Kelsey Hightower
@kelseyhightower

Replying to @RKela

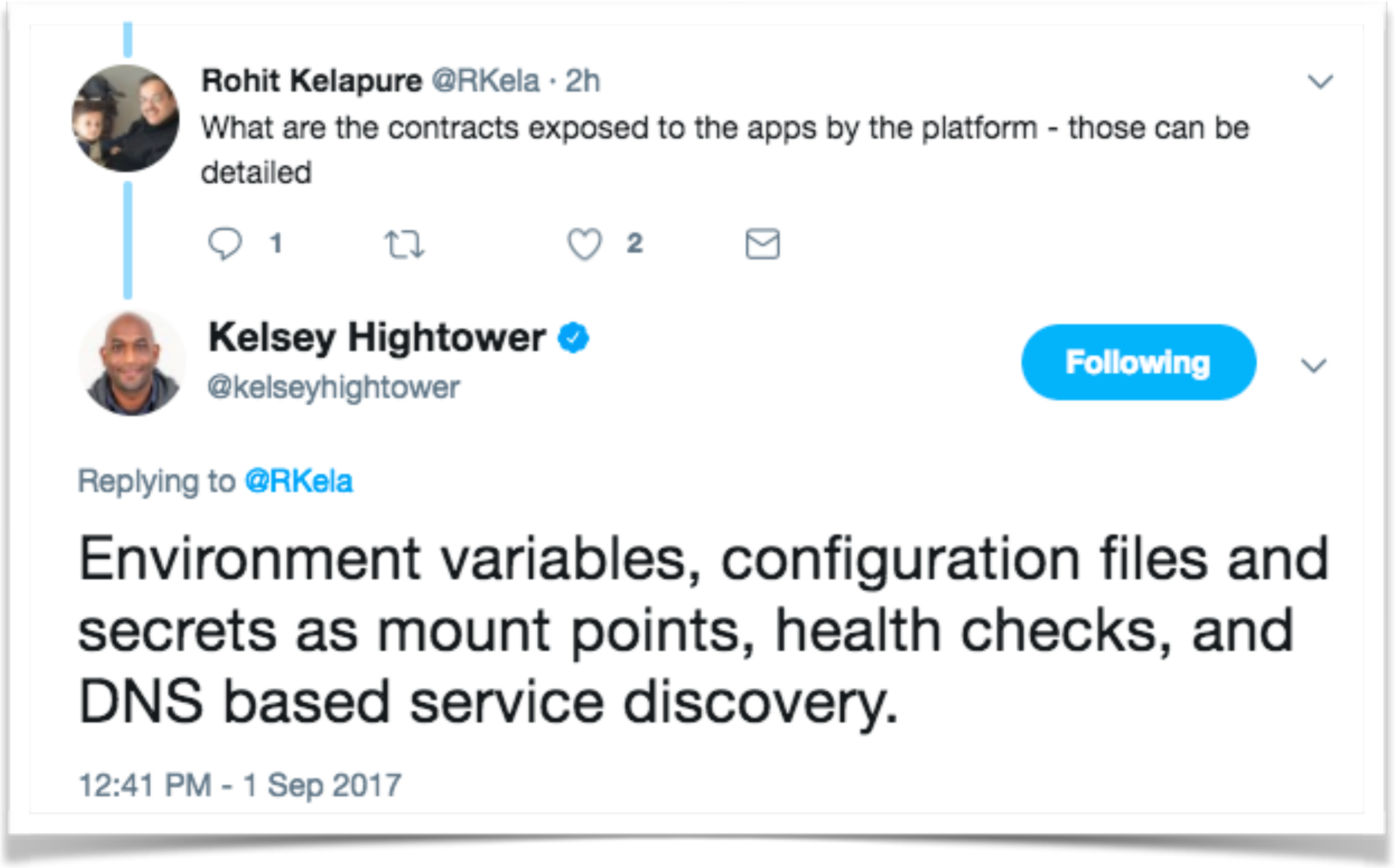
In some ways this is by design as Kubernetes does not enforce any specific application architecture.

12:27 PM - 1 Sep 2017

https://twitter.com/kelseyhightower/status/903640408613306369

Contracts exposed to the apps by the platform

- Environment variables
- Configuration files
- Secrets as mount points
- Health checks
- DNS based service discovery



Rohit Kelapure @RKela · 2h

What are the contracts exposed to the apps by the platform - those can be detailed

Kelsey Hightower @kelseyhightower

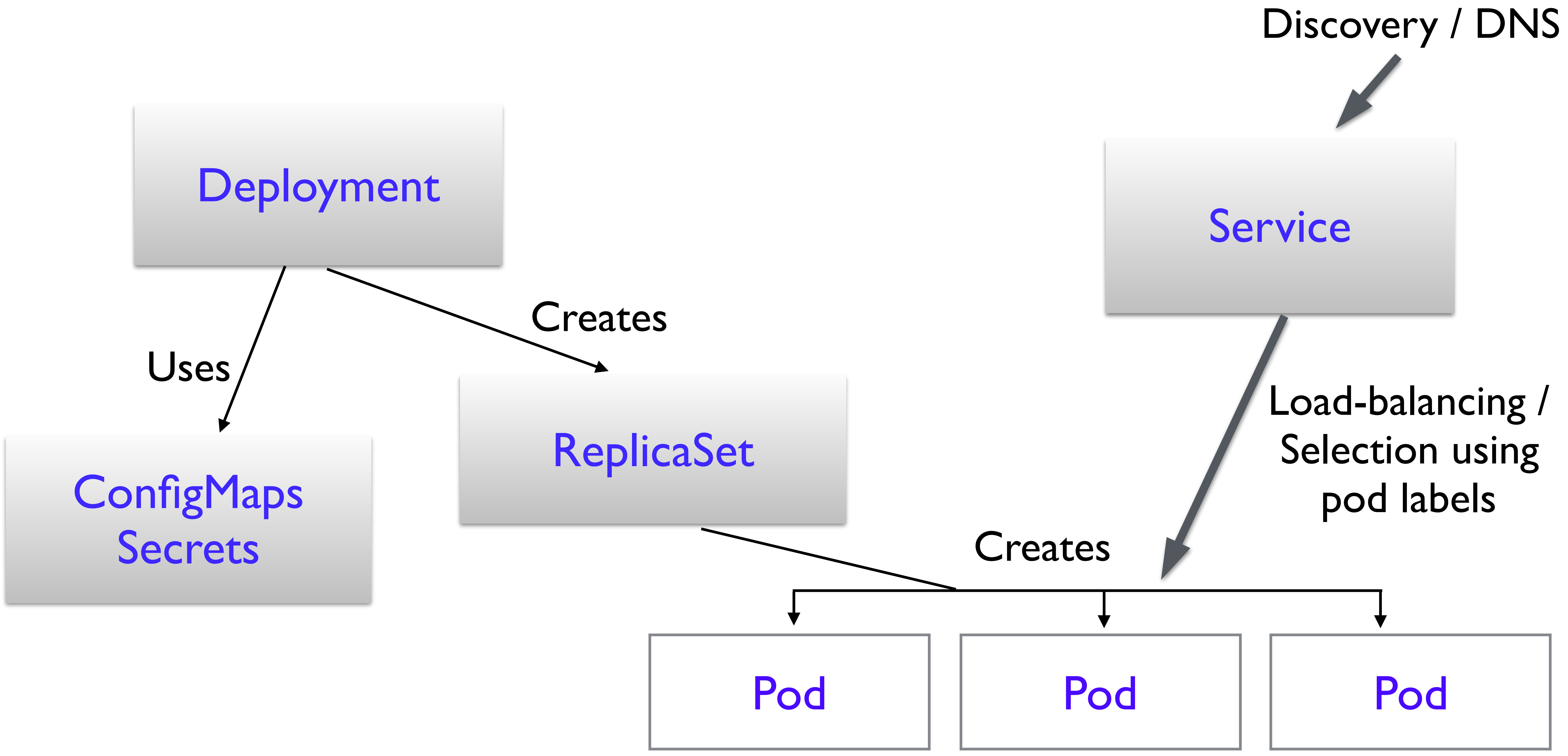
Replying to @RKela

Environment variables, configuration files and secrets as mount points, health checks, and DNS based service discovery.

12:41 PM - 1 Sep 2017

https://twitter.com/kelseyhightower/status/903643916599046145

Kubernetes Resources



Externalized Configuration

- Environment variables
  - Easy to set in `deployment.yaml`
  - Might need to use `SPRING_APPLICATION_JSON` for map based properties
- ConfigMaps and Secrets
  - Can be set using environment or mounted as config files
- Use Spring Cloud Config Server
- Init container can write properties file to shared volume

Demo

**Simple REST Repository App
as part of a
Microservice Architecture**

https://github.com/trisberg/devoxx-spring-boot-k8s/blob/master/demo-actors.adoc

https://github.com/trisberg/devoxx-spring-boot-k8s/blob/master/demo-microservices.adoc

https://github.com/trisberg/boot-k8s-microservices/tree/devoxx-2017

Mount ConfigMaps

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: actors
  labels:
    app: actors
data:
  application.yaml: |-
    security:
      basic:
        enabled: false
    spring:
      datasource:
        url: jdbc:mysql://${MYSQL_SERVICE_HOST}:${MYSQL_SERVICE_PORT}/mysql
        username: root
        password: ${MYSQL_ROOT_PASSWORD}
        driverClassName: com.mysql.jdbc.Driver
        testOnBorrow: true
        validationQuery: "SELECT 1"
```

```
spec:
  containers:
  - name: actors
    image: trisberg/actors:0.0.1-SNAPSHOT
…
     volumeMounts:
      - name: application-config
        mountPath: "/config"
        readOnly: true
  volumes:
  - name: application-config
    configMap:
      name: actors
      items:
      - key: application.yaml
        path: application-kubernetes.yaml
```

Access Secrets in Env Var

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mysql
  labels:
    app: mysql
data:
  mysql-root-password: eW91cnBhc3N3b3Jk
```

```yaml
env:
- name: SERVER_PORT
  value: '80'
- name: SPRING_PROFILES_ACTIVE
  value: kubernetes
- name: MYSQL_ROOT_PASSWORD
  valueFrom:
    secretKeyRef:
      name: mysql
      key: mysql-root-password
```

Spring Cloud / Netflix OSS

https://projects.spring.io/spring-cloud/

- Spring Cloud Config
- Service Discovery
  - Netflix Eureka
  - Consul
- Load balancing / routing
  - Netflix Ribbon & Zuul
- Circuit Breakers
  - Netflix Hystrix

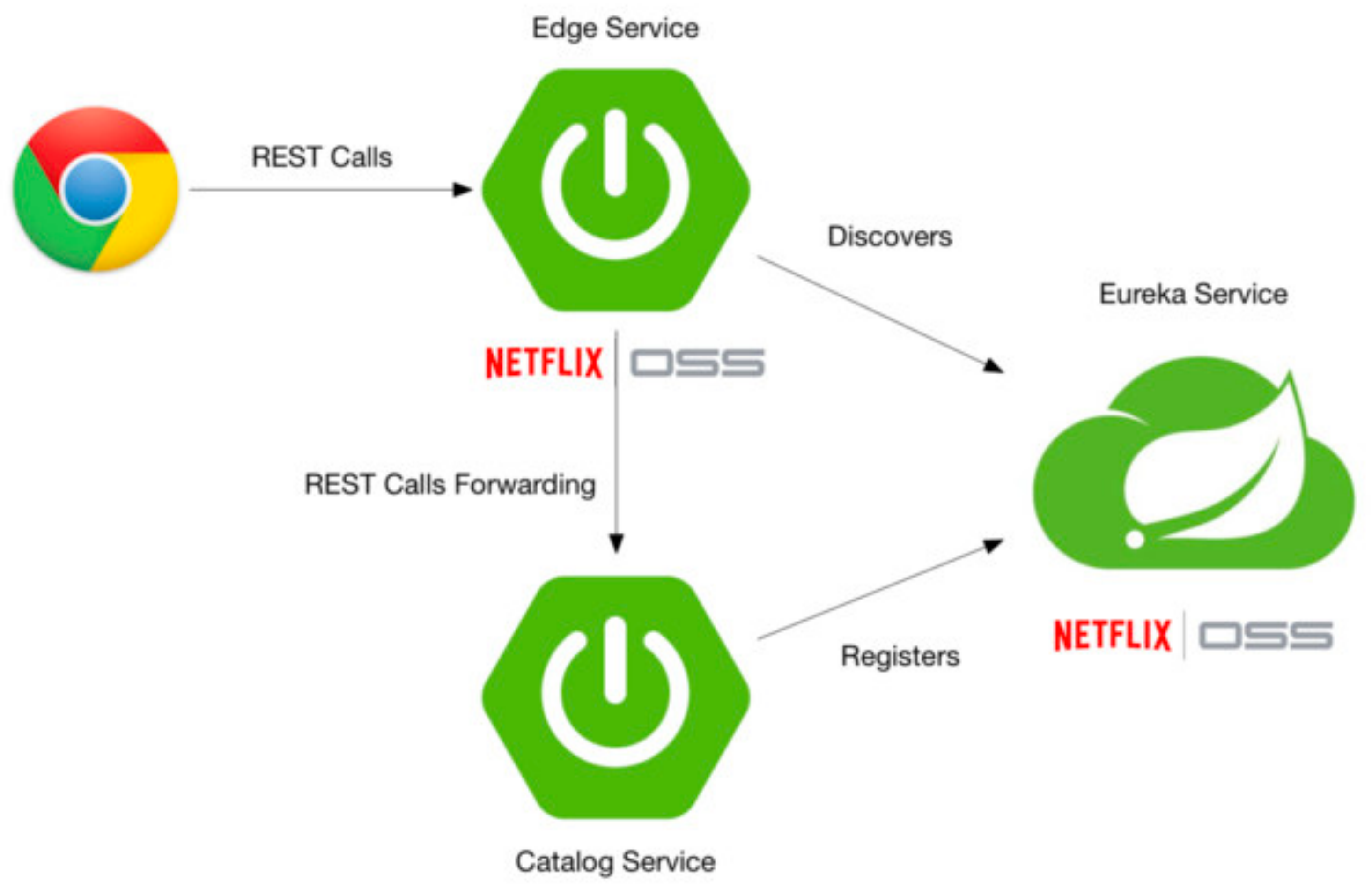
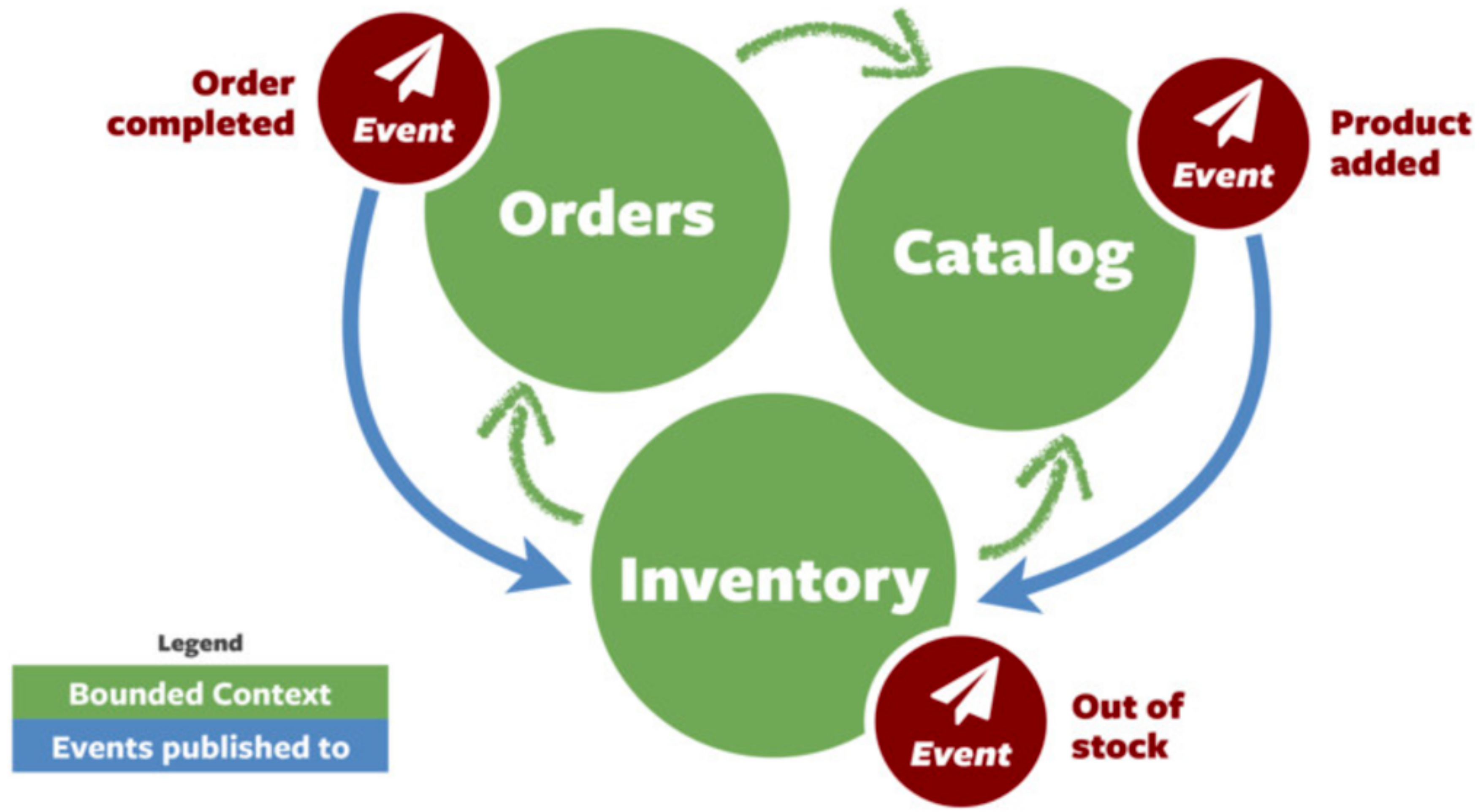


Image from: https://www.slideshare.net/mraible/develop-hip-apis-and-apps-with-spring-boot-and-angular-connecttech-2017/16?src=clipshare

Microservice Architecture Concerns

- Externalized Configuration ✓
  - ConfigMap and Secrets
- Service Discovery ✓
  - DNS, DiscoveryClient
- Circuit-breaker
- Distributed Tracing
- Metrics
- Log aggregation



https://speakerdeck.com/olivergierke/refactoring-to-a-system-of-systems?slide=29

Microservice Runtime Management

- Circuit-breaker - Netflix Hystrix
- Distributed Tracing - Spring Cloud Sleuth / Zipkin
- Metrics - Spring Boot Actuator / Micrometer

- Service Mesh - Istio
  - load balancing / routing
  - policy enforcement
  - telemetry and reporting

Log Aggregation

- Spring Cloud Sleuth
  - https://cloud.spring.io/spring-cloud-sleuth/
- Stackdriver
  - https://kubernetes.io/docs/tasks/debug-application-cluster/logging-stackdriver/
- Elasticsearch and Kibana
  - https://kubernetes.io/docs/tasks/debug-application-cluster/logging-elasticsearch-kibana/
- Loggly
  - https://www.weave.works/blog/log-aggregation-kubernetes-loggly/

Packaging

- **Helm**
  - The package manager for Kubernetes
    - https://docs.helm.sh/using_helm/#quickstart-guide
- **KubeApps**
  - Discover & launch great Kubernetes-ready apps
    - https://kubeapps.com/
- **Example**
    - https://github.com/trisberg/boot-k8s-microservices/tree/devoxx-2017

Helm Repos

- Stable/Incubator official published charts
  - https://github.com/kubernetes/charts
- Your own repo
  - Use `helm init` to create and then publish the repo with any HTTP server
  - Use `helm repo add` to add it to helm CLI
- Local repo
  - Use `helm serve`

Spring Cloud for Kubernetes

- Fabric8 team created `spring-cloud-kubernetes`
  - `DiscoveryClient for Kubernetes`
  - `ConfigMap and Secrets PropertySource`
  - `Ribbon discovery in Kubernetes`
  - `Zipkin discovery in Kubernetes`
  - `and more …`
- Now available in `spring-cloud-incubator` on GitHub

Summary / Recommendations

- Mount ConfigMaps as `application-kubernetes.yaml`
- Access Secrets in Environment Variables
- Use Spring Cloud Sleuth for Tracing with Zipkin
- Use Micrometer for Metrics with Prometheus/Grafana
- Keep an eye on Istio for Service Mesh features
- Use Helm for Packaging to simplify installation of your app

Questions?

Useful Links

- https://github.com/trisberg/devoxx-spring-boot-k8s
- https://projects.spring.io/spring-boot/
- https://kubernetes.io/
- https://projects.spring.io/spring-cloud/
- http://www.oreilly.com/programming/free/kubernetes-for-java-developers.csp
- https://github.com/spring-cloud-incubator/spring-cloud-kubernetes
- https://developers.redhat.com/blog/2017/10/03/configuring-spring-boot-kubernetes-configmap/
- https://developers.redhat.com/blog/2017/10/04/configuring-spring-boot-kubernetes-secrets/