

Cloud Native Apps with Cloud Foundry & Spring

Kenny Bastani
Spring Developer Advocate

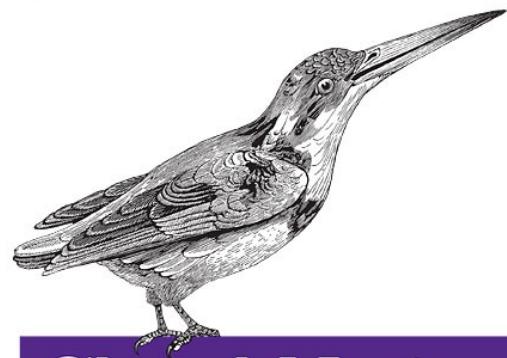


Kenny Bastani

Spring Developer Advocate



O'REILLY®



Cloud Native Java

DESIGNING RESILIENT SYSTEMS WITH SPRING BOOT,
SPRING CLOUD, AND CLOUD FOUNDRY

Josh Long & Kenny Bastani

Agenda

Agenda

1 Microservices & Cloud Native

2 Spring Boot

3 Spring Cloud

4 Cloud Native Operations

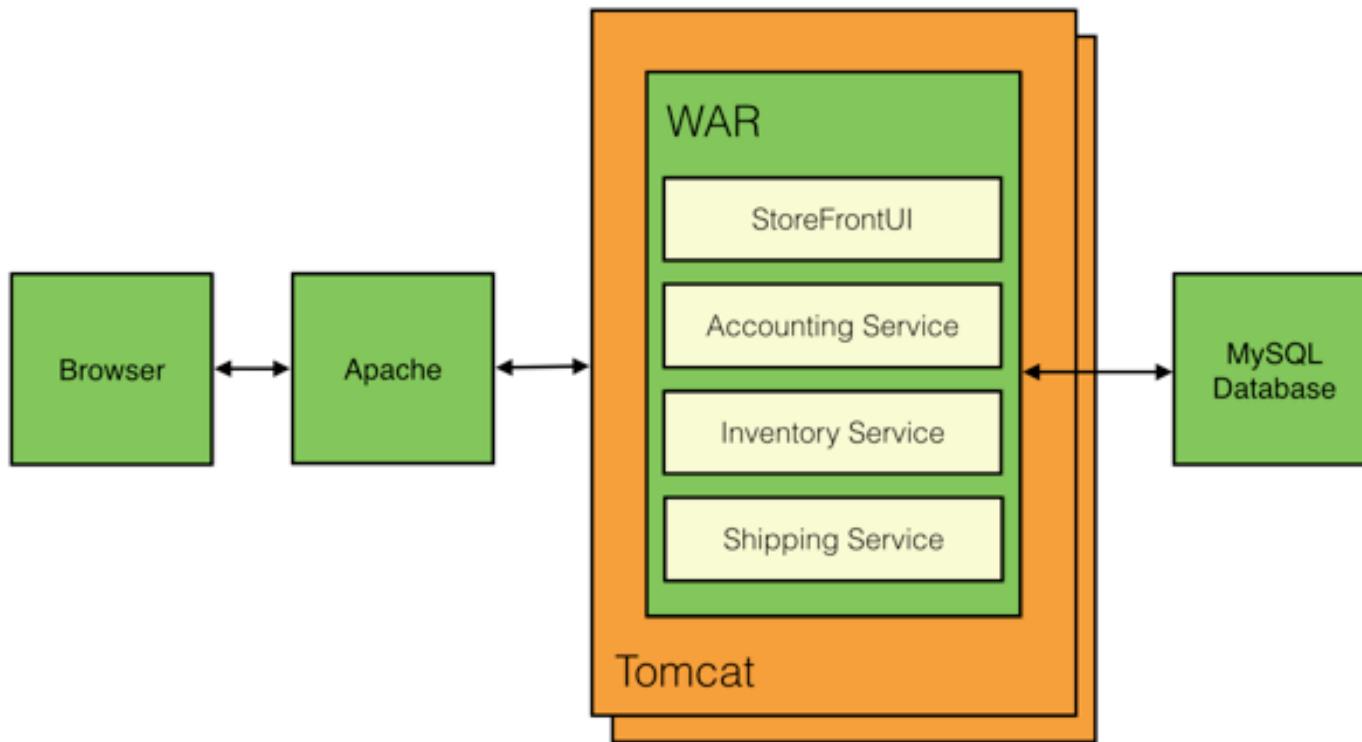
5 Microservice Reference App

Microservices

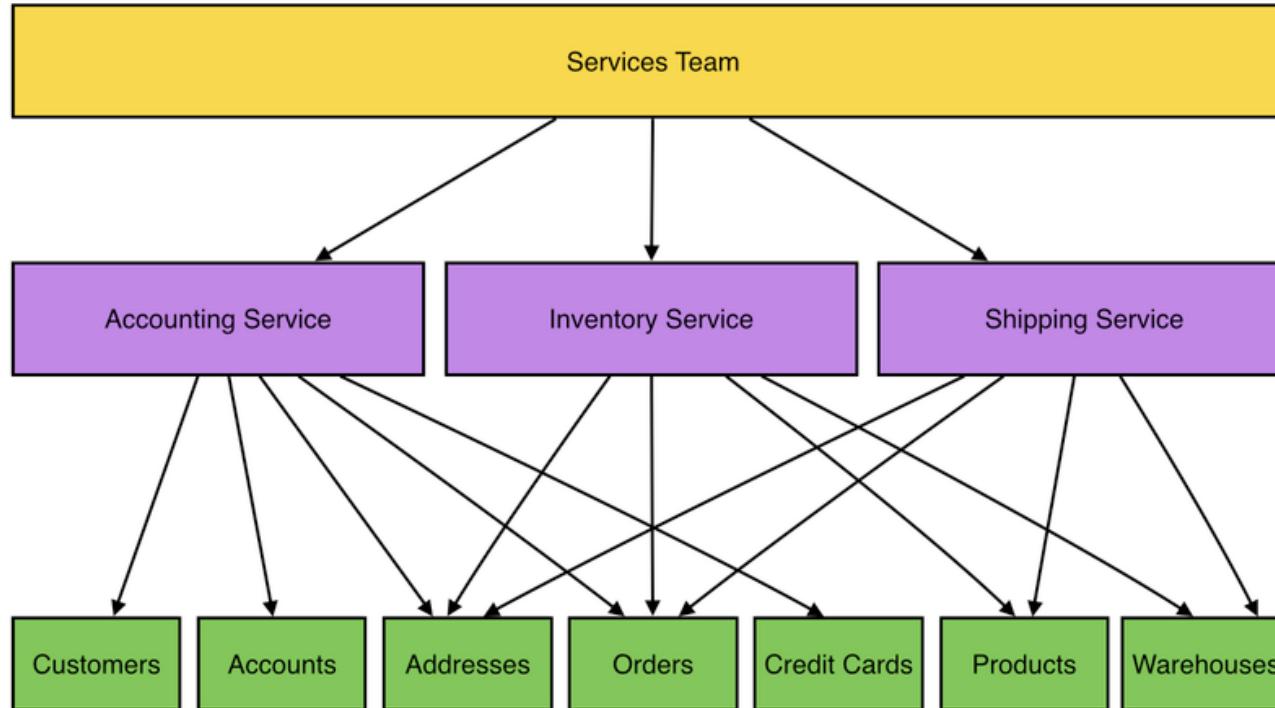
WHAT ARE MICROSERVICES?

How did we get there?

We started with the monolith....

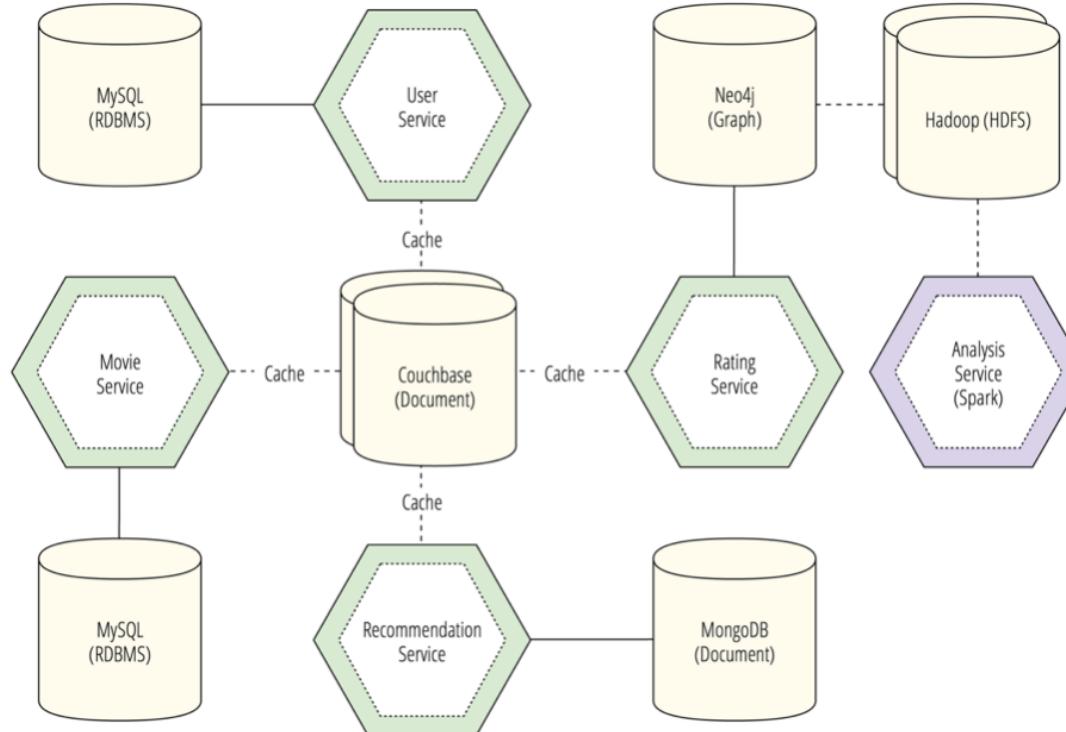


We moved towards SOA (Service Oriented Architecture)...

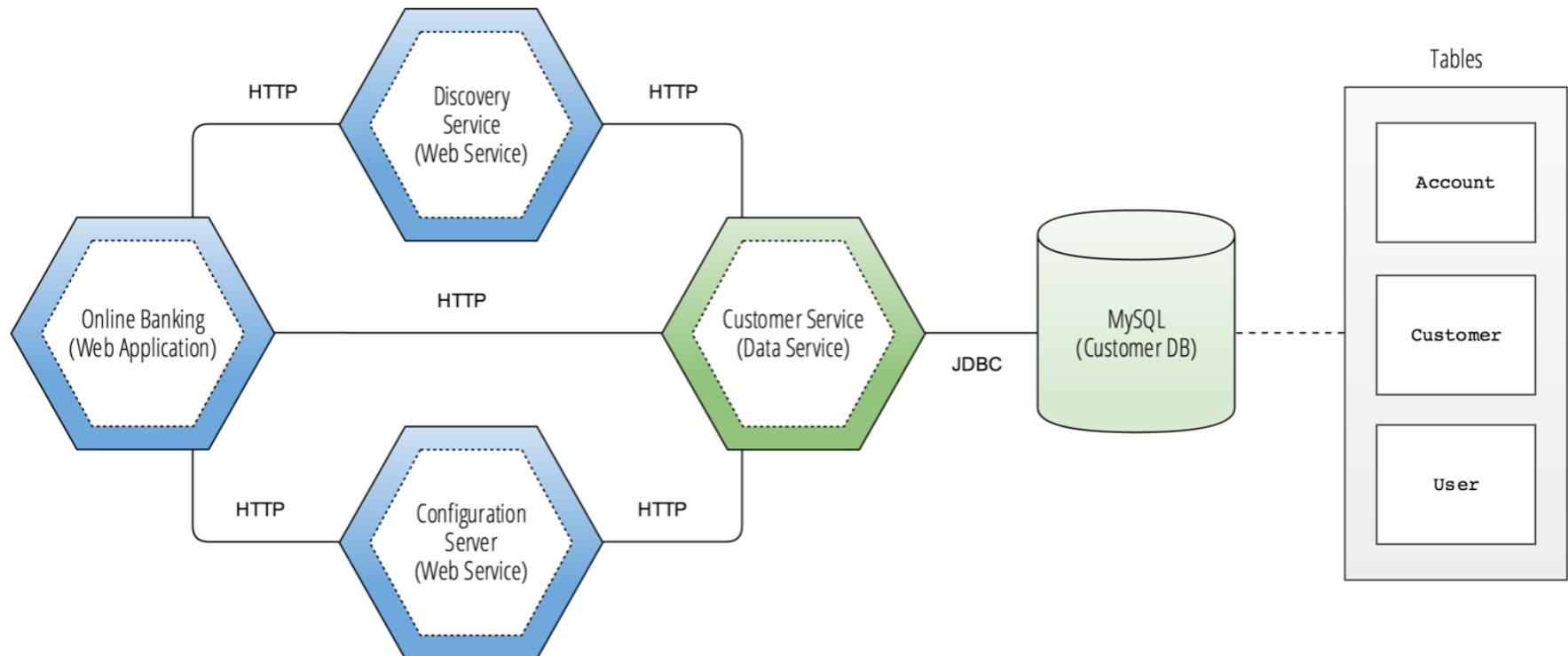


We arrived at microservices...

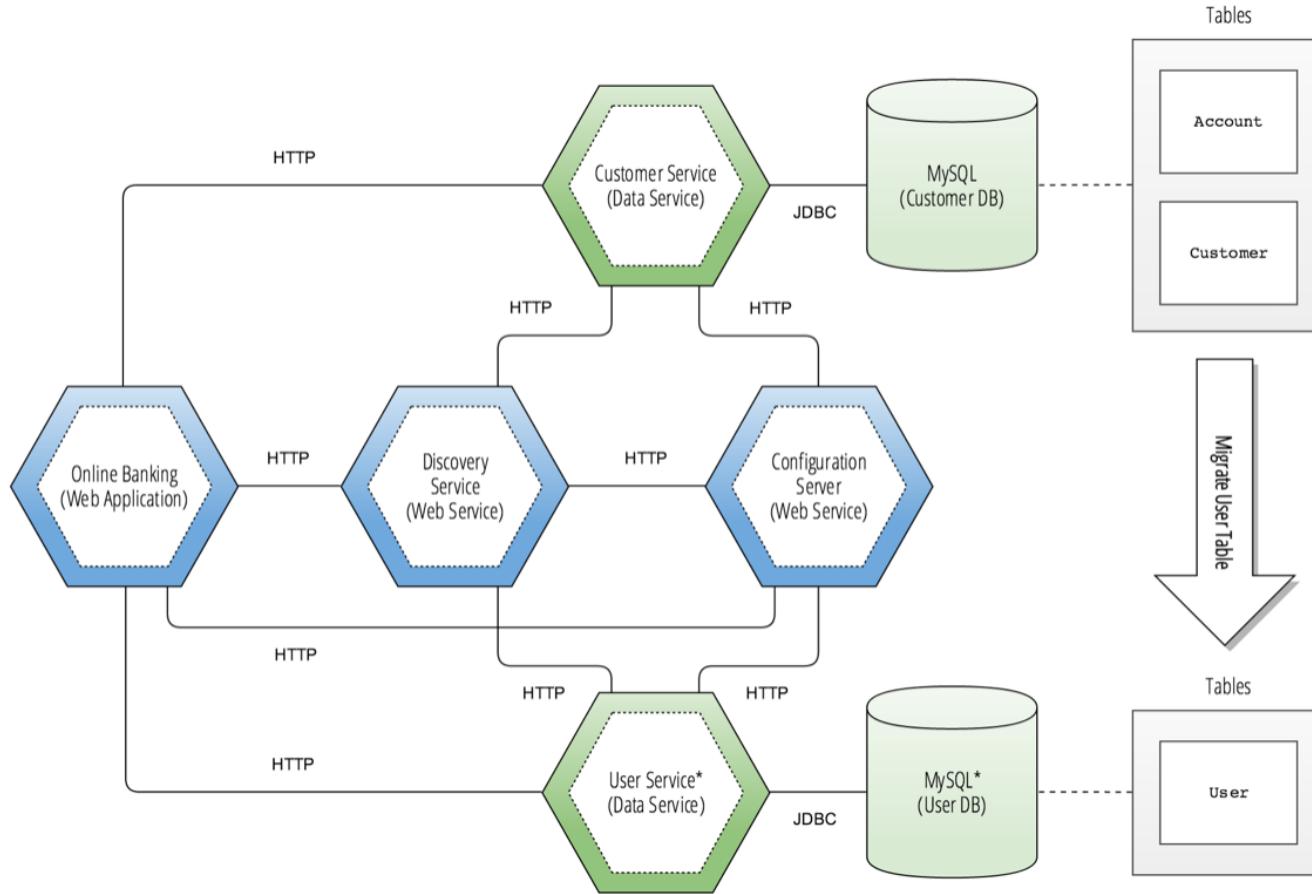
- Each team gets one database and one service
- Shared caches are platform provided services that are shared for consistency



Monolith to Microservice



Monolith to Microservice



Online Store Example

Cloud native application as microservices

Cloud Native Outfitters

online-store-web.cfapps.io/index.html#/products/SKU-34563

Search

Welcome, John Doe! Orders Settings Cart Logout

Home / Products / Like a BOSH (T-Shirt, Women's Medium)



Product Details

The BOSH Outer Shell (**BOSH**) is an elegant release engineering tool for a more virtualized cloud-native age. The feeling of spinning up a highly available distributed system of VMs is second only to the feeling of frequently pushing code to production. Show the cloud who's *BOSH* with this stylish cloud native ops tee.

Cloud Native Tee Collection
99% YAML, 11% CLI
BOSH CCK *recommended*
4 nines of *re-washability*

Price: \$14.99

Product number: SKU-34563

0 Add to Cart

Summary

Product name: Like a BOSH (T-Shirt, Women's Medium)

Price: 14.99

Related



Best. Cloud. Ever.
(T-Shirt, Men's)



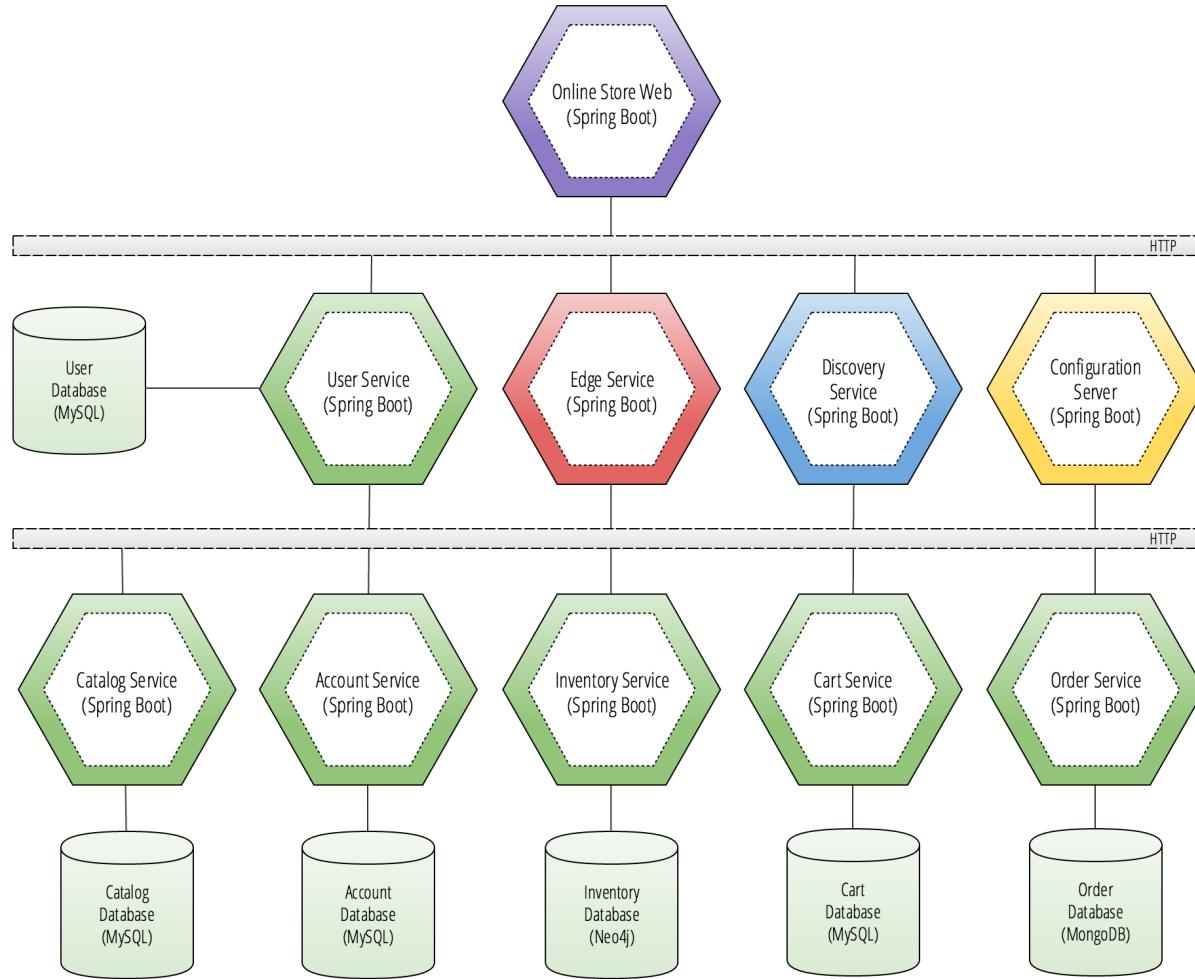
We're gonna
need a bigger
VM
(T-Shirt, Women's)



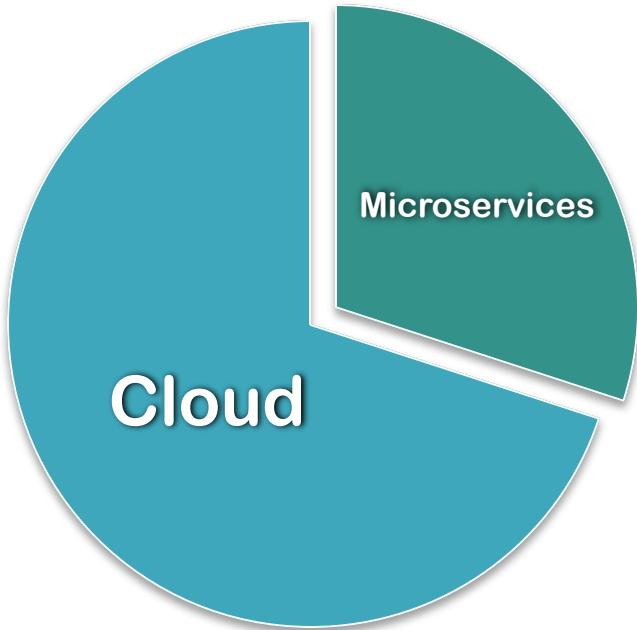
Like a BOSH
(T-Shirt, Women's)



cf push awesome
(Hoodie, Men's)

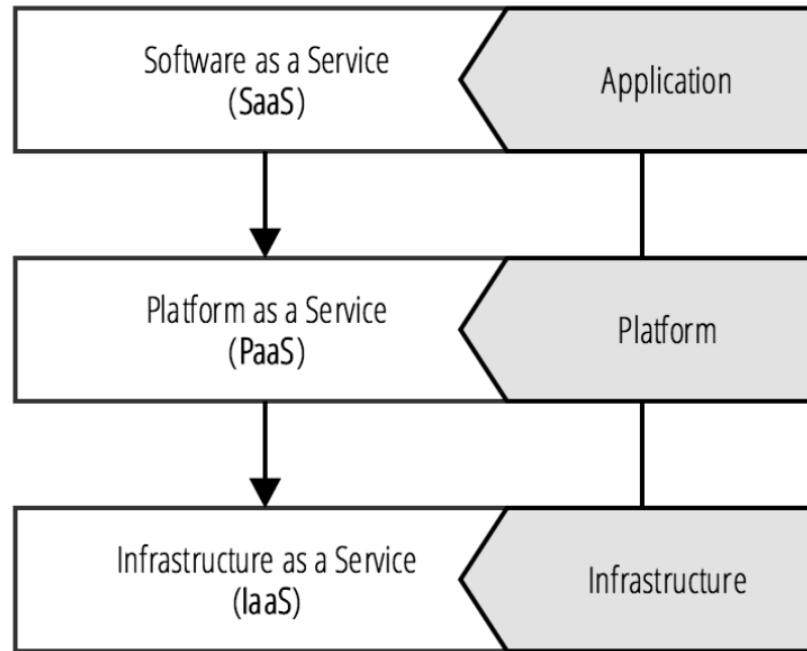


Cloud Native & Microservices



- Microservices are only a small part of a larger picture

Cloud Platform



THE TWELVE FACTORS

I. Codebase

One codebase tracked in revision control, many deploys

II. Dependencies

Explicitly declare and isolate dependencies

III. Config

Store config in the environment

IV. Backing services

Treat backing services as attached resources

V. Build, release, run

Strictly separate build and run stages

VI. Processes

Execute the app as one or more stateless processes

VII. Port binding

Export services via port binding

VIII. Concurrency

Scale out via the process model

IX. Disposability

Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity

Keep development, staging, and production as similar as possible

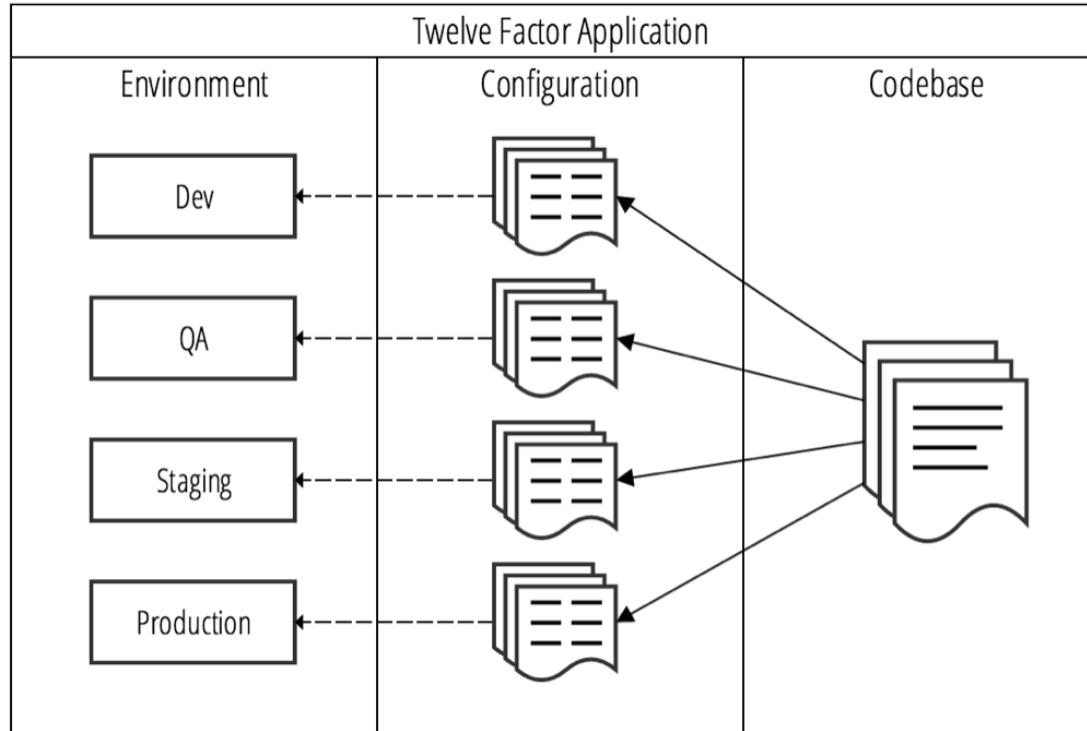
XI. Logs

Treat logs as event streams

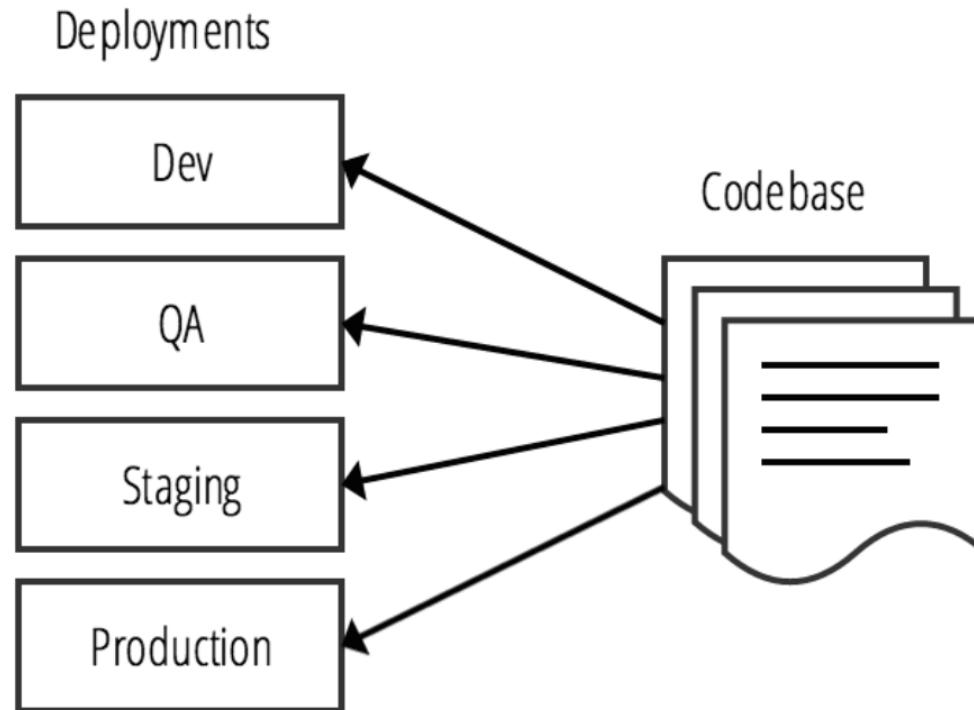
XII. Admin processes

Run admin/management tasks as one-off processes

Twelve-factor Application Configuration



Twelve-factor Application Deployment



Cloud Native Java

Spring Boot, Spring Cloud, and Cloud Foundry

Spring Boot

A JVM micro-framework for building microservices

What is Spring Boot?



Phil Webb
@phillip_webb



Following

For those on [reddit.com/r/java/](https://www.reddit.com/r/java/) saying it's Spring Boot is "the framework for a framework" here's a diagram:

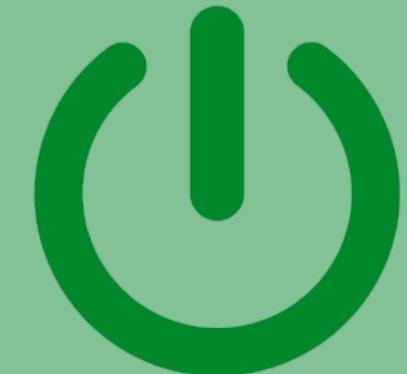


RETWEETS
111

FAVORITES
68



7:54 PM - 8 Sep 2015



spring boot

supports rapid development of production-ready applications and services

Spring Initializr for bootstrapping your applications

The screenshot shows the Spring Initializr interface at start.spring.io. The page title is "SPRING INITIALIZR bootstrap your application now". A header bar says "Generate a [Maven Project] with Spring Boot 1.3.2".

Project Metadata

- Artifact coordinates:
 - Group: com.example
 - Artifact: demo
- Name: demo
- Description: Demo project for Spring Boot
- Package Name: com.example
- Packaging: Jar
- Java Version: 1.8
- Language: Java

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies: Web, Security, JPA, Actuator, Devtools...

Selected Starters: Web, JPA, Security, Actuator, Rest Repositories

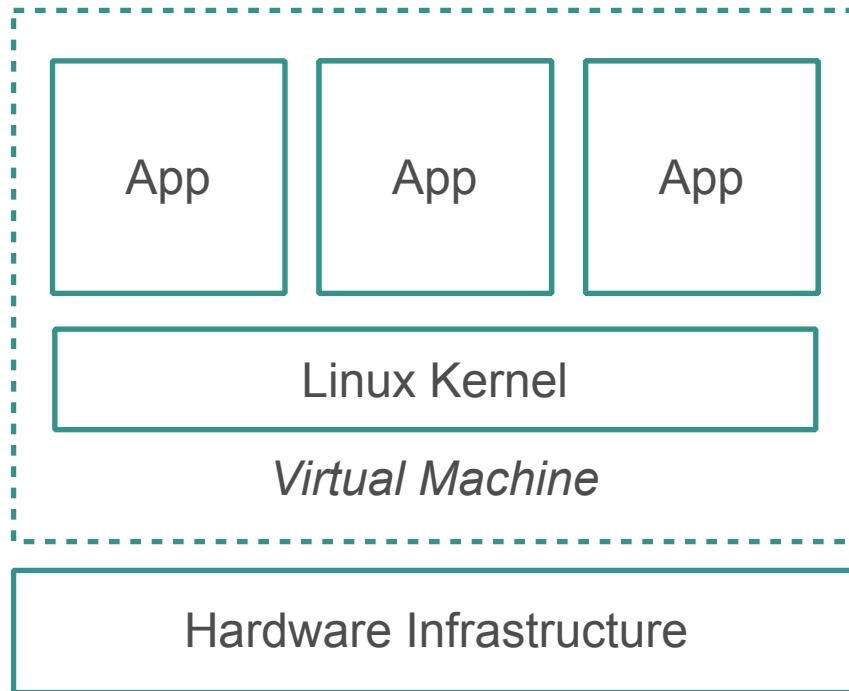
At the bottom, a note says "Too many options? [Switch back to the simple version.](#)" and a "Generate Project" button is visible.

Cloud Native & DevOps



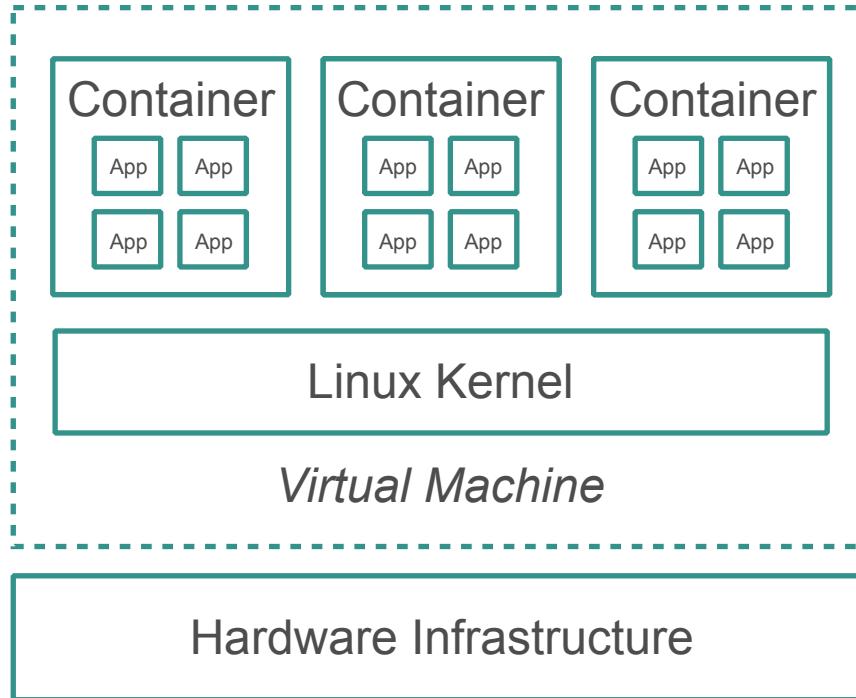
Ditching the application server

Application Server Deployment - Monolith



- Load balancing requires provisioning of new VMs and app server installations
- Poor resource isolation; memory leaks can cause other applications to become unavailable
- Runtime environment is driven by the operator

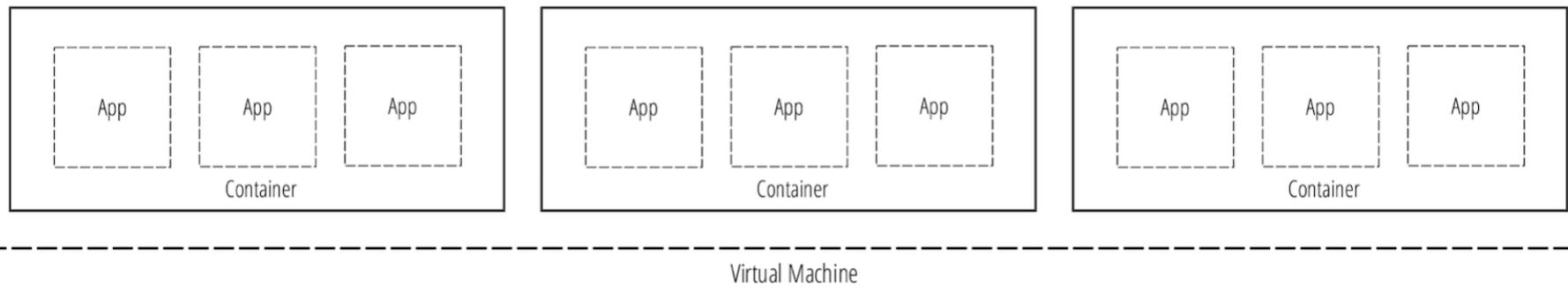
Linux Container Deployment - Microservice



- Development team drives the application runtime of a container
- Containers are resource isolated, allowing efficient scheduling onto a grid of VMs
- Containers take seconds to start, VMs take minutes
- It's not a rule that there is one microservice per container

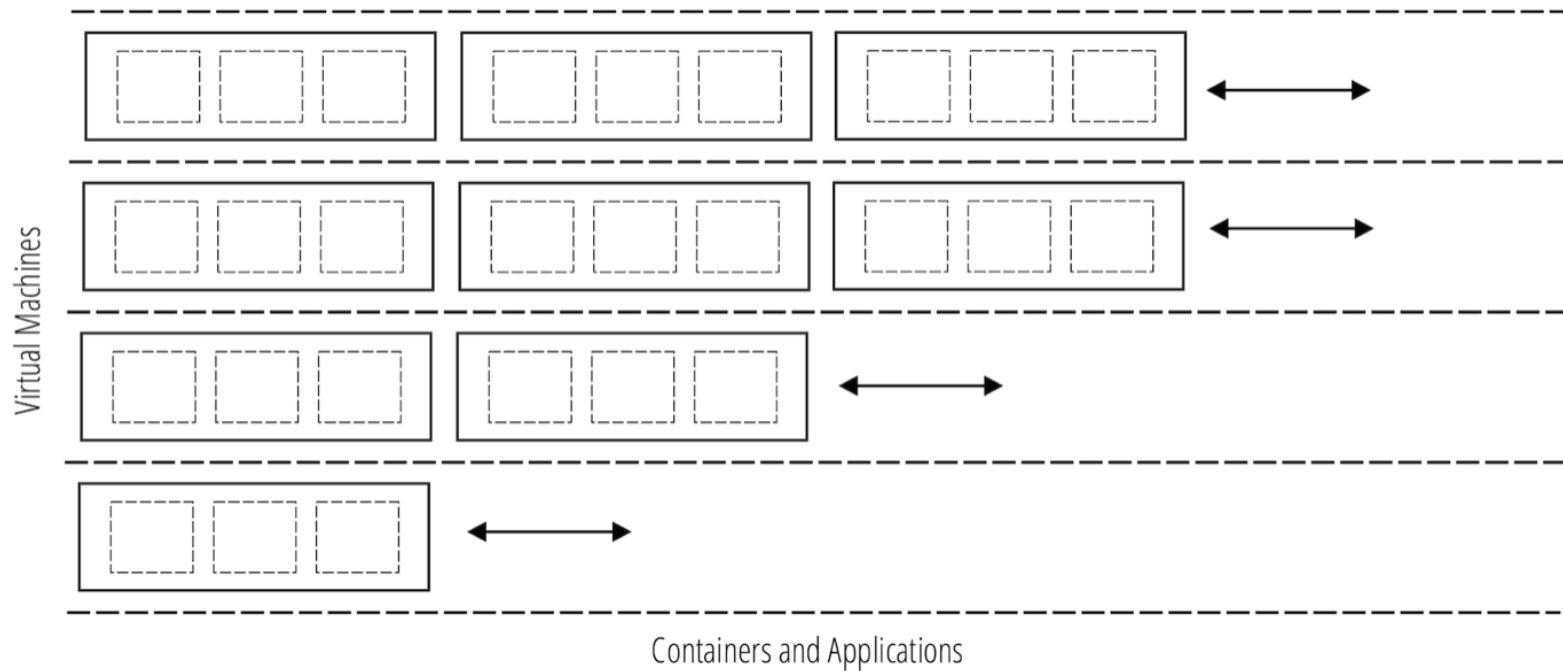
Microservices - Container Deployment

- Each microservice can be containerized with their application dependencies
- Containers get scheduled on virtual machines with an allotted resource policy



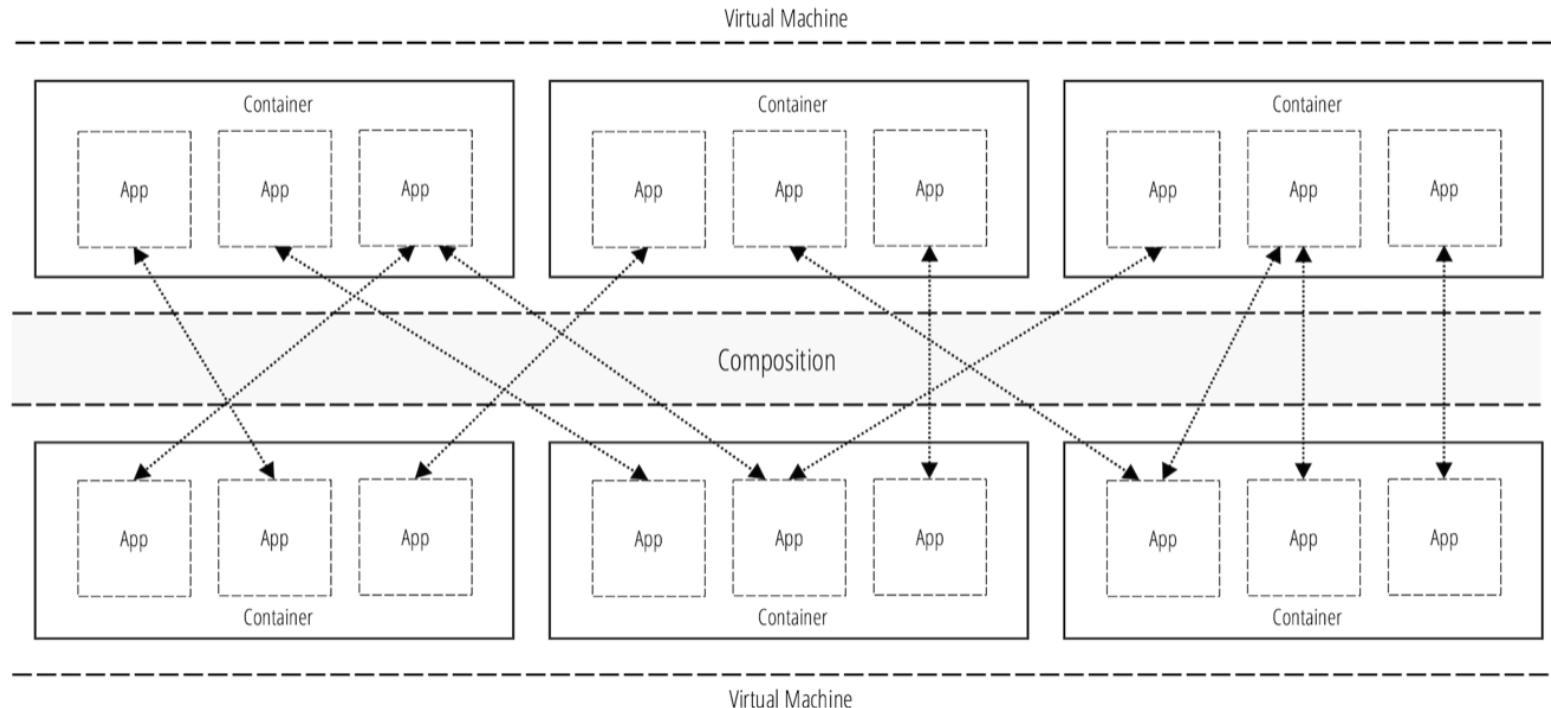
Auto-scaling

- Minutes to start a VM, but seconds to start a container
- An elastic runtime handles auto-scaling of VMs with cloud providers



Orchestration

- Each microservice needs to communicate outside containers
- Service discovery provides an automatic method for finding other service dependencies



Spring Cloud



A toolset designed for building distributed systems

What is Spring Cloud?

- Spring Cloud provides a way to turn Spring Boot microservices into distributed applications

Phil Webb
@phillip_webb

Following

For those on [reddit.com/r/java/](https://www.reddit.com/r/java/) saying it's Spring Boot is "the framework for a framework" here's a diagram:

SPRING FRAMEWORK

SPRING BOOT

111 RETWEETS 68 FAVORITES

Imgflip.com



@kennybastani



spring cloud



Apache Zookeeper



* these logos are all **trademark/copyright** their respective owners (T-B, L-R):

Netflix, amazon.com, Apache Software Foundation, Cloud Foundry, Hashicorp
they are ALL great organizations and we love their open-source and their APIs!!



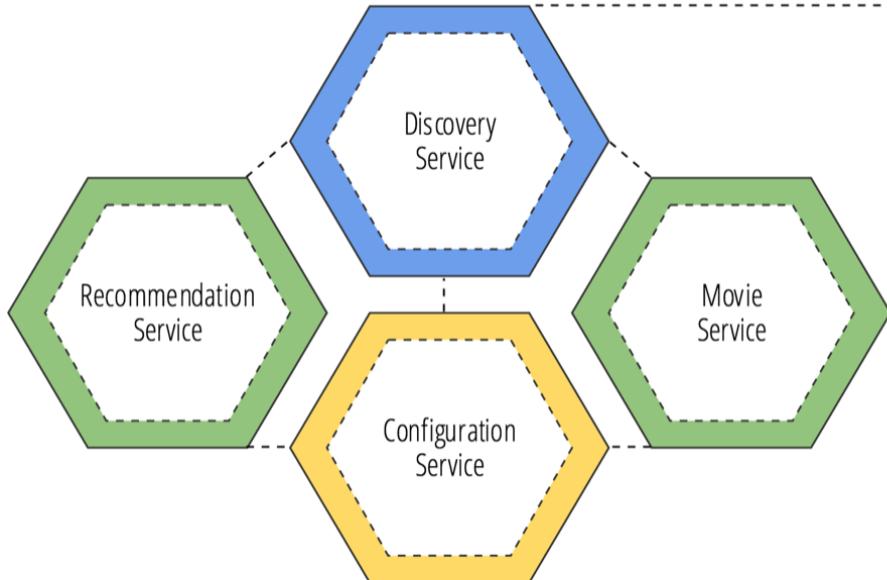
What is Spring Cloud?

- * Service Discovery
- * API Gateway
- * Config Server
- * Circuit Breakers
- * Distributed Tracing

Service Discovery



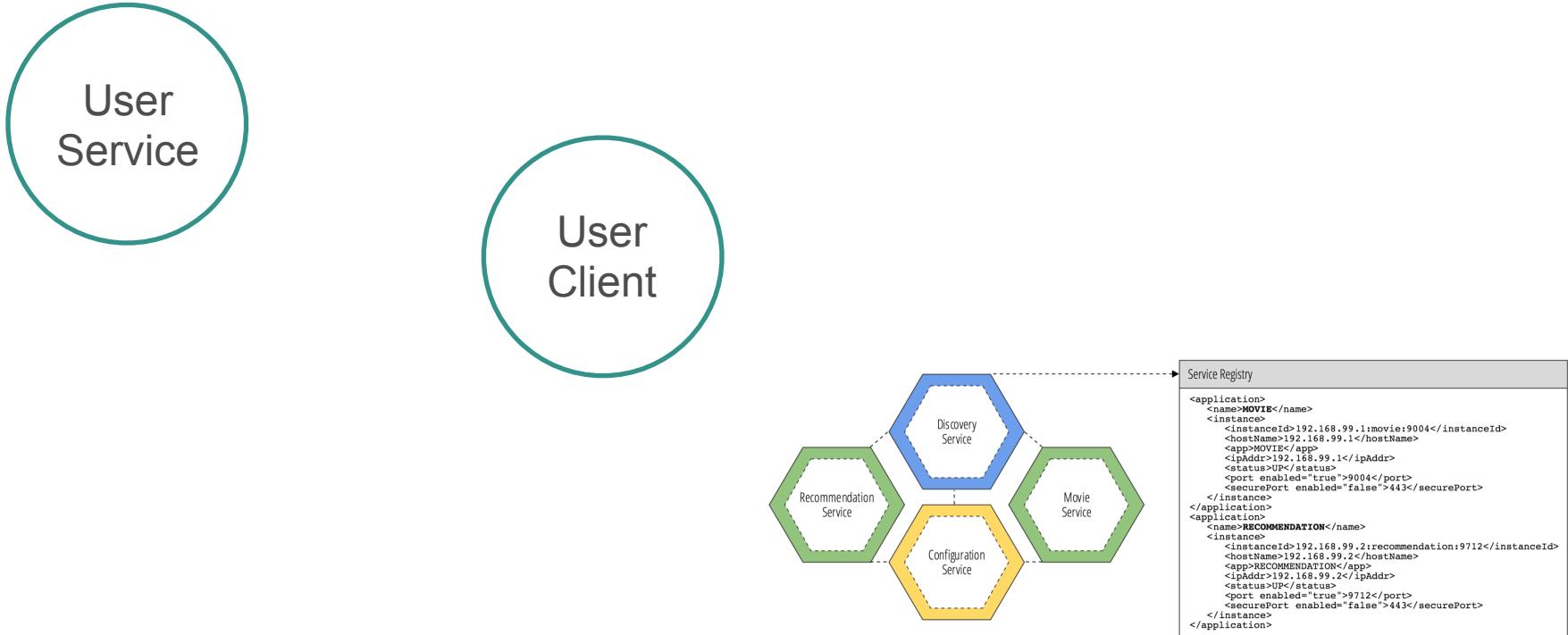
Service Registry



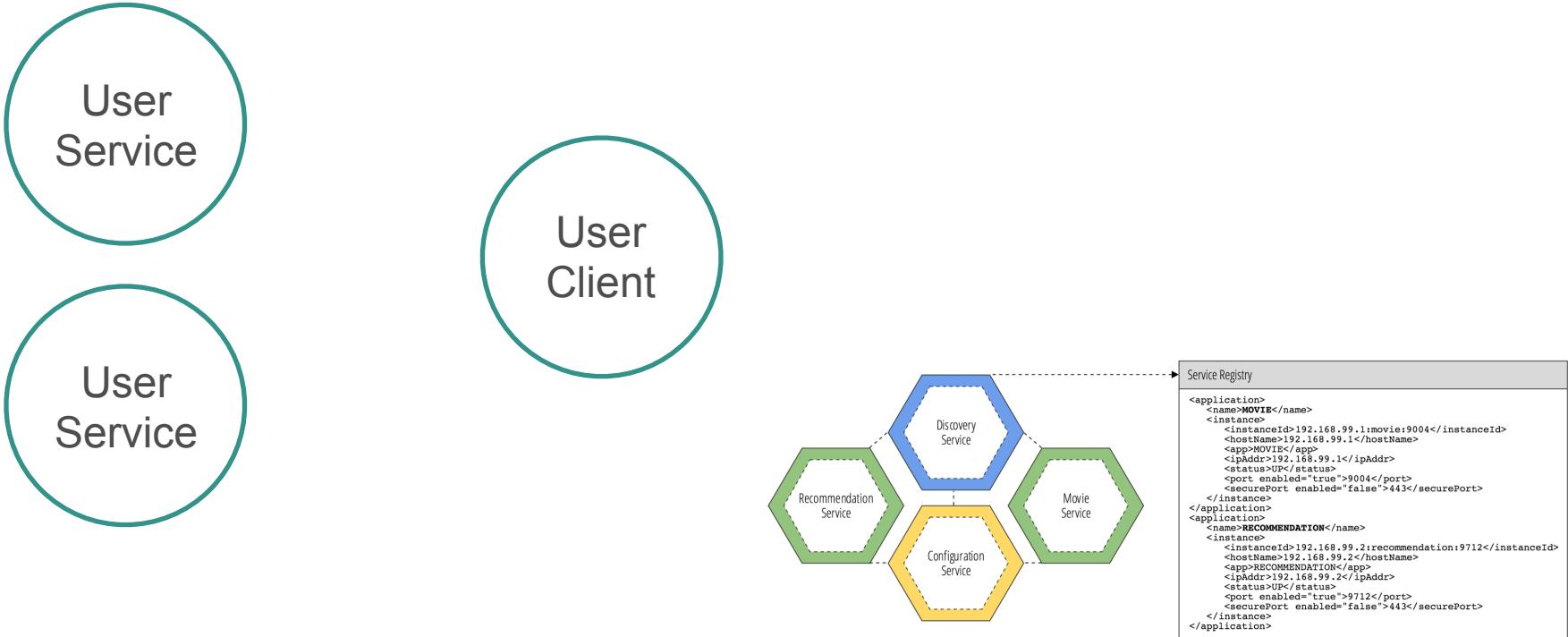
```
Service Registry

<application>
  <name>MOVIE</name>
  <instance>
    <instanceId>192.168.99.1:movie:9004</instanceId>
    <hostName>192.168.99.1</hostName>
    <app>MOVIE</app>
    <ipAddr>192.168.99.1</ipAddr>
    <status>UP</status>
    <port enabled="true">9004</port>
    <securePort enabled="false">443</securePort>
  </instance>
</application>
<application>
  <name>RECOMMENDATION</name>
  <instance>
    <instanceId>192.168.99.2:recommendation:9712</instanceId>
    <hostName>192.168.99.2</hostName>
    <app>RECOMMENDATION</app>
    <ipAddr>192.168.99.2</ipAddr>
    <status>UP</status>
    <port enabled="true">9712</port>
    <securePort enabled="false">443</securePort>
  </instance>
</application>
```

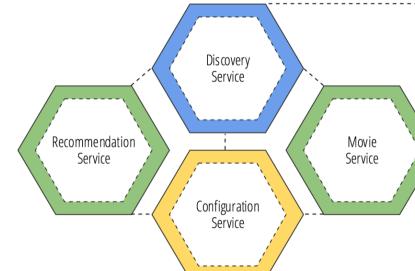
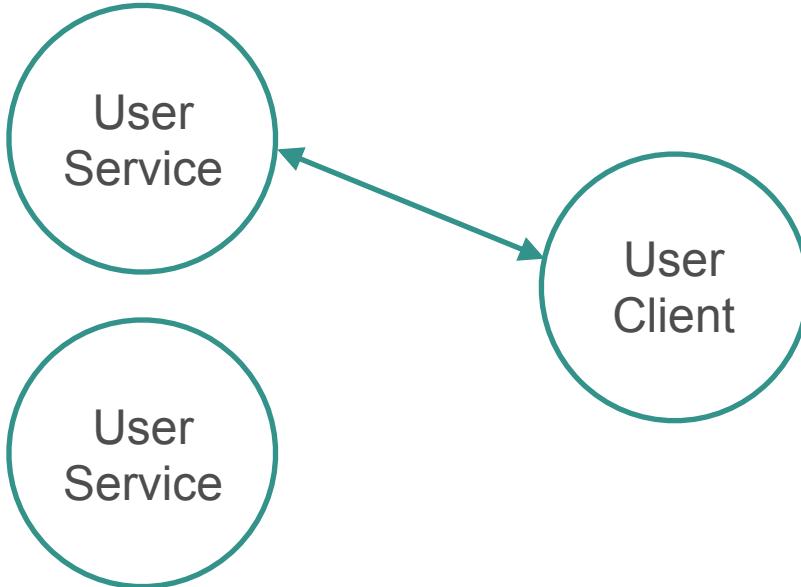
Client-side Load Balancing



Client-side Load Balancing



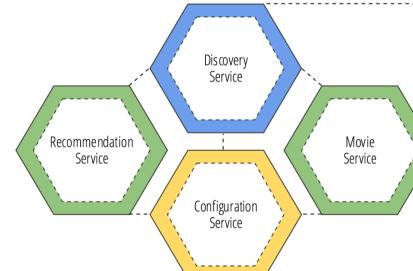
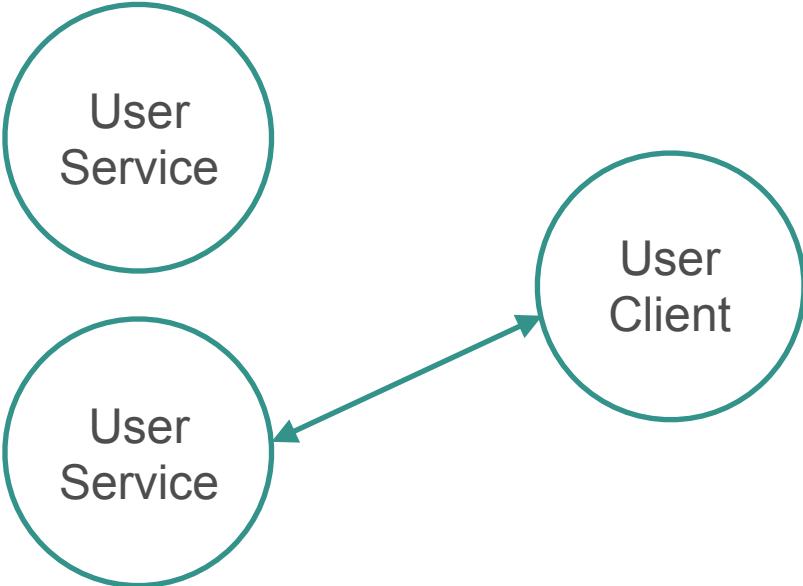
Client-side Load Balancing



```
Service Registry
```

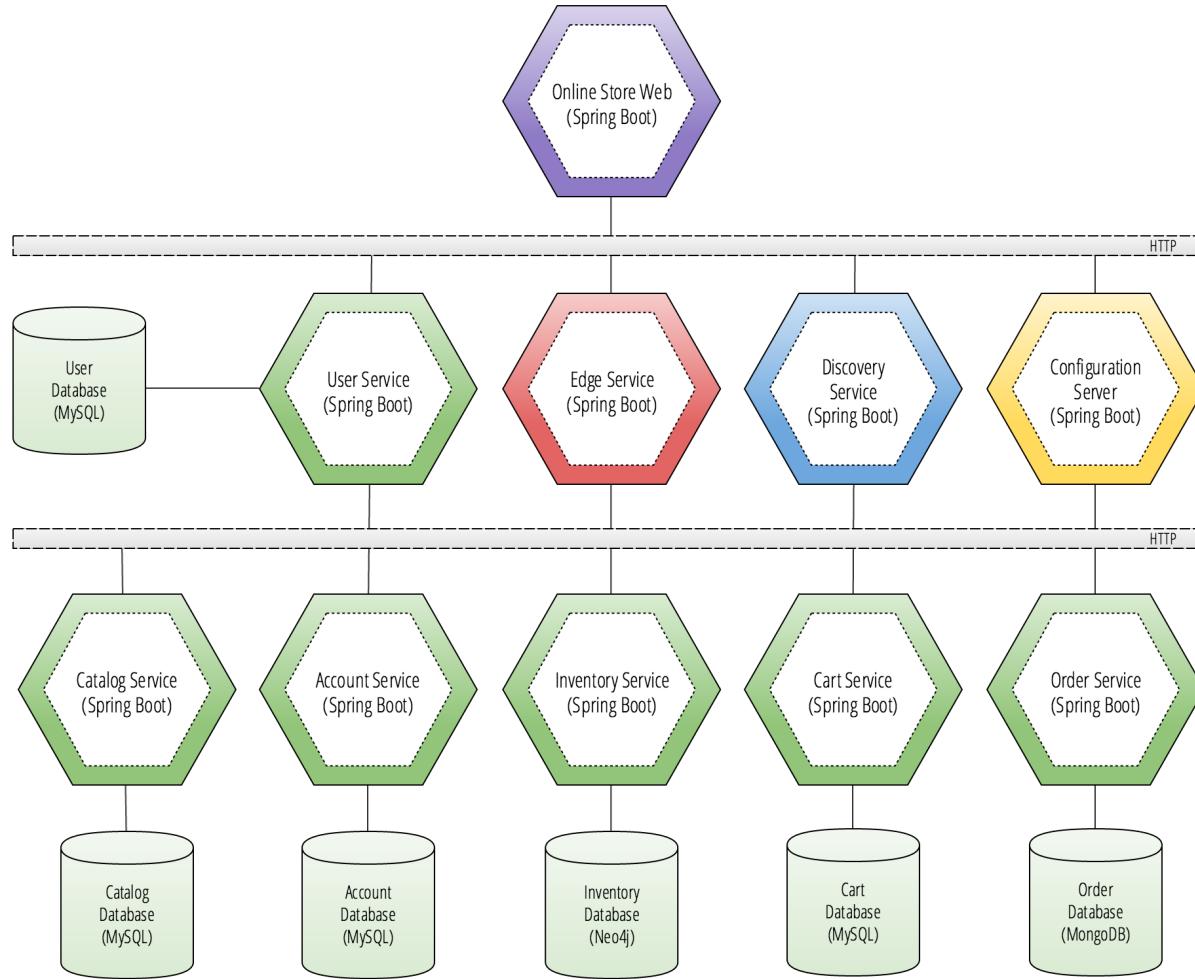
```
<application>
  <name>MOVIE</name>
  <instance>
    <instanceId>192.168.99.1:movie:9004</instanceId>
    <hostName>192.168.99.1</hostName>
    <app>MOVIE</app>
    <ipAddr>192.168.99.1</ipAddr>
    <status>UP</status>
    <port enabled="true">9004</port>
    <securePort enabled="false">443</securePort>
  </instance>
</application>
<application>
  <name>RECOMMENDATION</name>
  <instance>
    <instanceId>192.168.99.2:recommendation:9712</instanceId>
    <hostName>192.168.99.2</hostName>
    <app>RECOMMENDATION</app>
    <ipAddr>192.168.99.2</ipAddr>
    <status>UP</status>
    <port enabled="true">9712</port>
    <securePort enabled="false">443</securePort>
  </instance>
</application>
```

Client-side Load Balancing

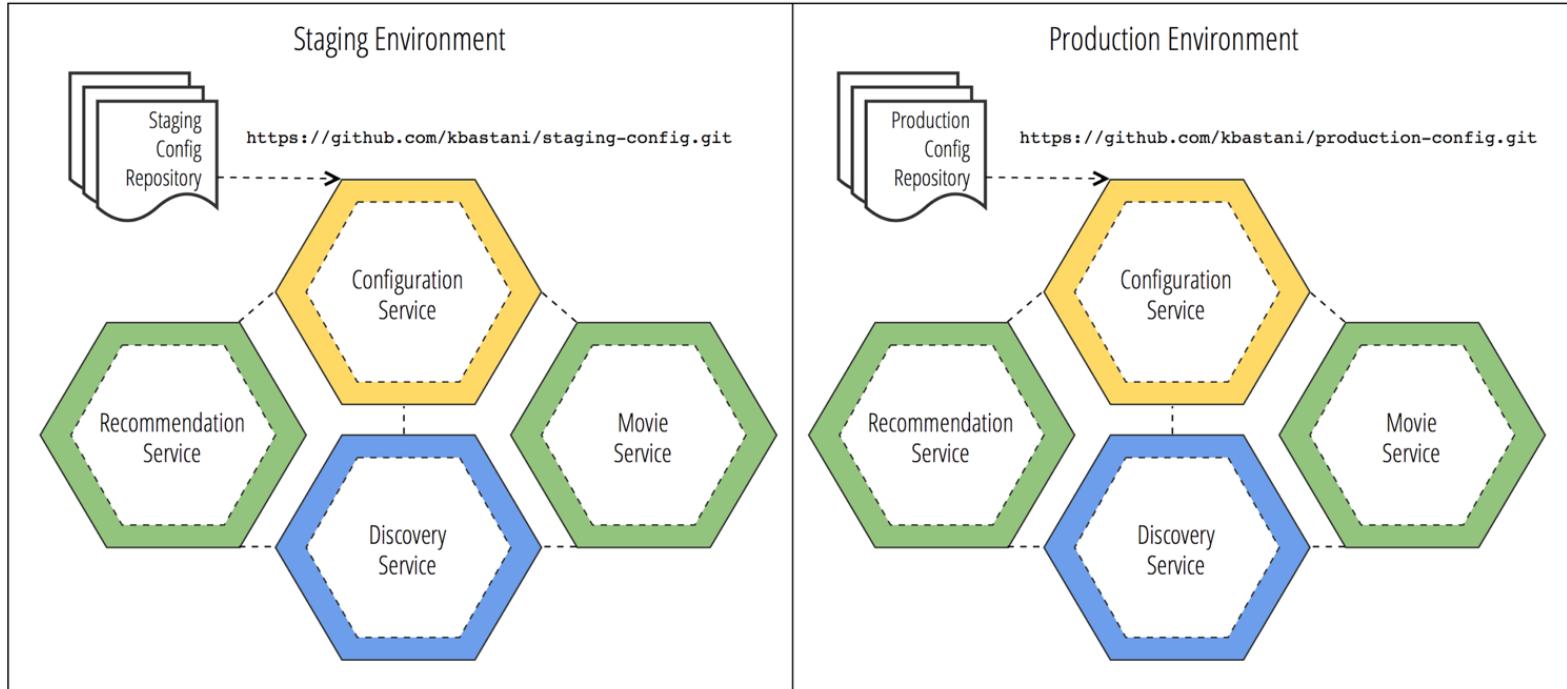


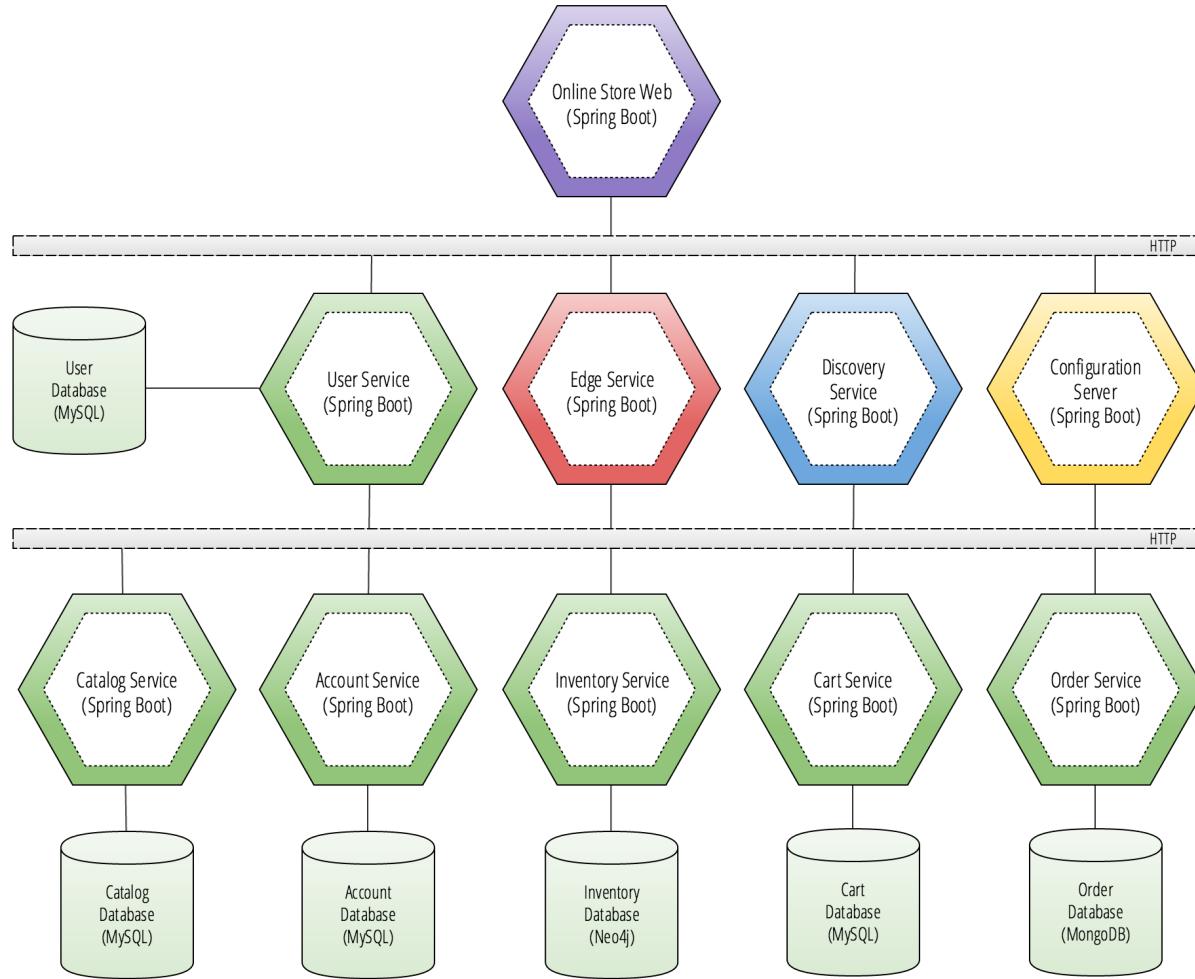
```
Service Registry
```

```
<application>
  <name>MOVIE</name>
  <instance>
    <instanceId>192.168.99.1:movie:9004</instanceId>
    <hostName>192.168.99.1</hostName>
    <app>MOVIE</app>
    <ipAddr>192.168.99.1</ipAddr>
    <status>UP</status>
    <port enabled="true">9004</port>
    <securePort enabled="false">443</securePort>
  </instance>
</application>
<application>
  <name>RECOMMENDATION</name>
  <instance>
    <instanceId>192.168.99.2:recommendation:9712</instanceId>
    <hostName>192.168.99.2</hostName>
    <app>RECOMMENDATION</app>
    <ipAddr>192.168.99.2</ipAddr>
    <status>UP</status>
    <port enabled="true">9712</port>
    <securePort enabled="false">443</securePort>
  </instance>
</application>
```



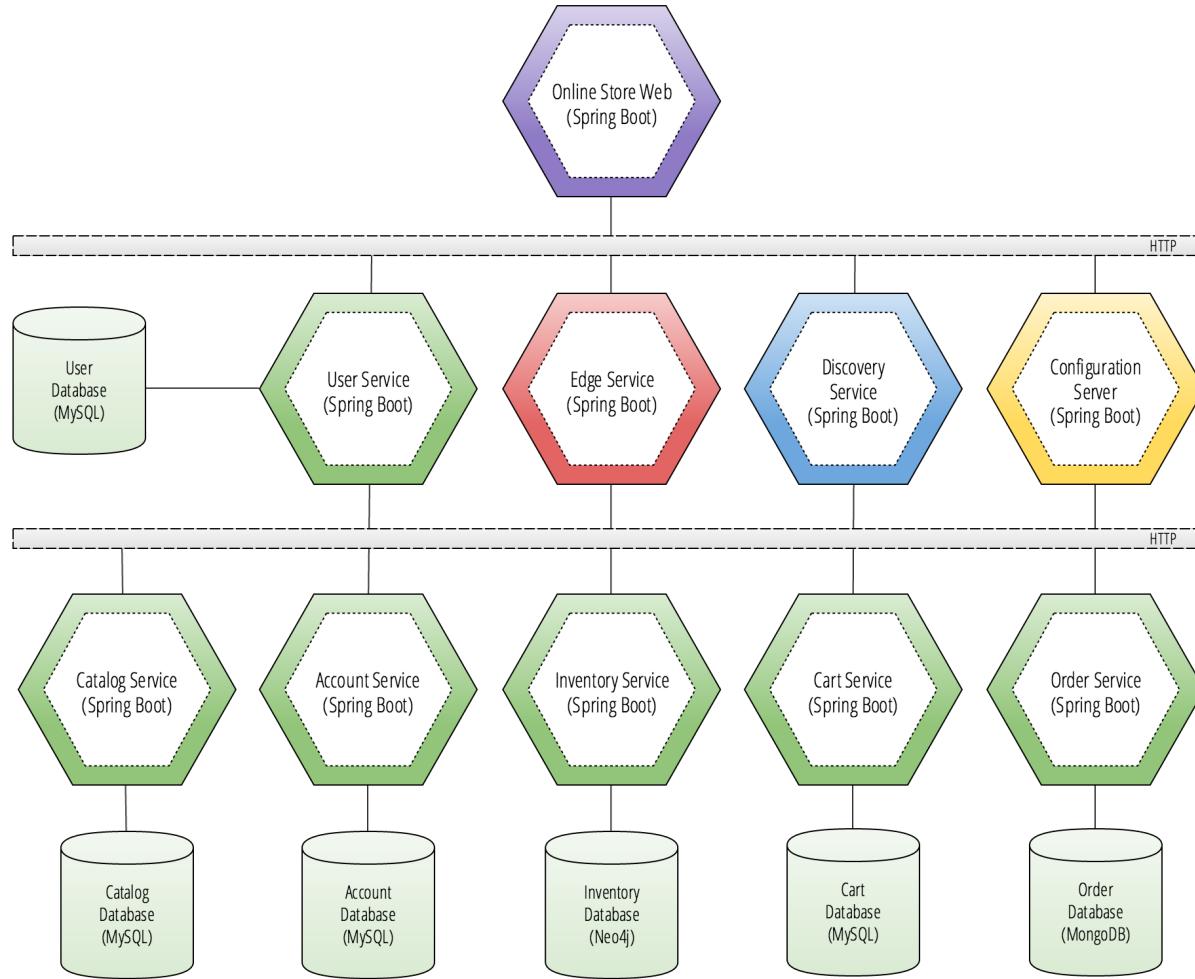
Configuration Service



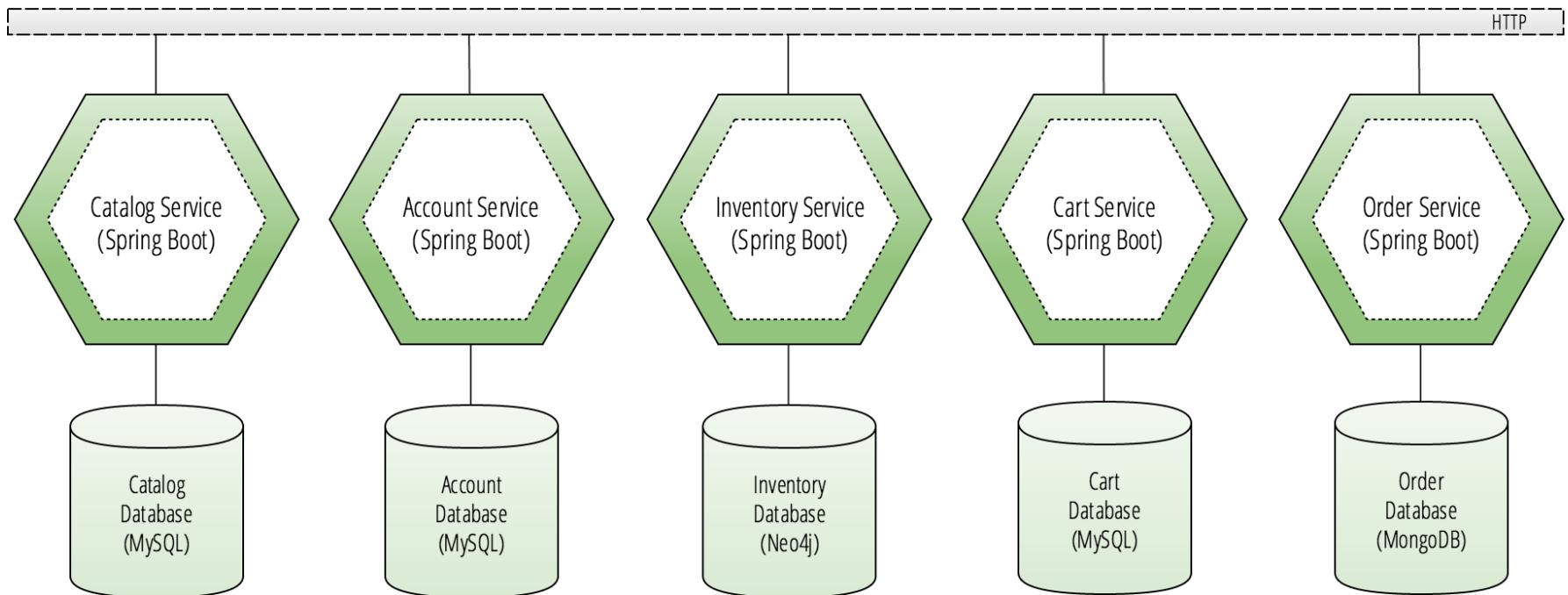


API Gateway

Building edge services that route calls to backend microservices



Backend Microservices



REST API Gateway / Edge Service



- API gateway downloads routes from services exposing REST APIs
- Each route of other microservices will be hosted on the API gateway
- API gateway will automatically reverse proxy to backend services
- Hypermedia allows traversing entire REST API description of microservices

HATEOAS

Hypermedia as the Engine of Application State

(REST APIs that self-describe)

Richardson Maturity Model

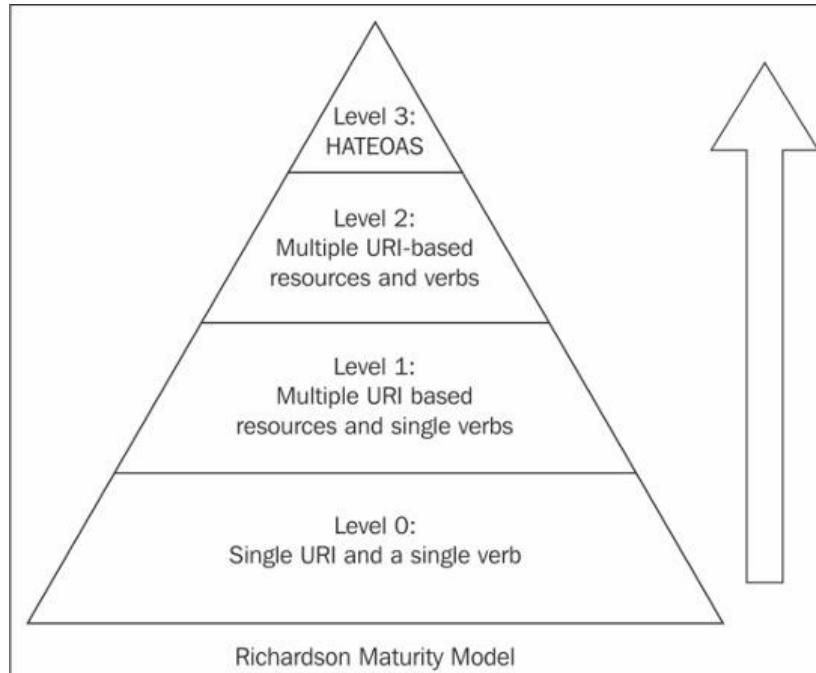
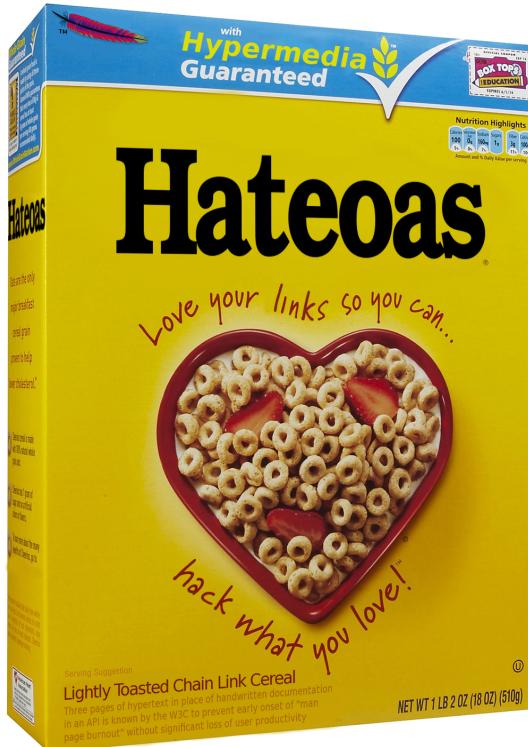


Photo credit: Packt Publishing

It's pronounced hawt-ee-oh-as



Browsing Hypermedia REST APIs

```
{  
  "_links": {  
    "ratings": {  
      "href": "http://rating-77.cfapps.io/ratings{?page,size,sort}",  
      "templated": true  
    },  
    "products": {  
      "href": "http://rating-77.cfapps.io/products{?page,size,sort}",  
      "templated": true  
    },  
    "users": {  
      "href": "http://rating-77.cfapps.io/users{?page,size,sort}",  
      "templated": true  
    },  
    "profile": {  
      "href": "http://rating-77.cfapps.io/alps",  
      "templated": false  
    }  
  }  
}
```

Each API call provides links

```
{  
  "_links": {  
    "first": {  
      "href": "http://rating-77.cfapps.io/users?page=0&size=20",  
      "templated": false  
    },  
    "self": {  
      "href": "http://rating-77.cfapps.io/users",  
      "templated": false  
    },  
    "next": {  
      "href": "http://rating-77.cfapps.io/users?page=1&size=20",  
      "templated": false  
    },  
    "last": {  
      "href": "http://rating-77.cfapps.io/users?page=47&size=20",  
      "templated": false  
    },  
    "search": {  
      "href": "http://rating-77.cfapps.io/users/search",  
      "templated": false  
    }  
  },  
  "_embedded": {  
    "users": [  
      {  
        "id": 0,  
        "knownId": "196",  
        "_links": {  
          "self": {  
            "href": "http://rating-77.cfapps.io/users/0",  
            "templated": false  
          },  
          "user": {  
            "href": "http://rating-77.cfapps.io/users/0",  
            "templated": false  
          }  
        }  
      }  
    ]  
  }  
}
```

Distributed Tracing

Disrupting the Cloud

Tracing the Online Store

Start time

05-13-2016

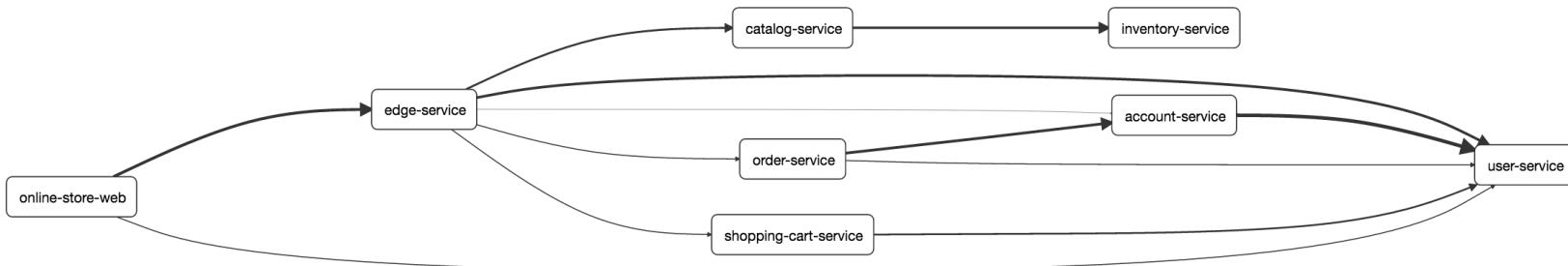
11:22

End time

05-20-2016

11:22

Analyze Dependencies



Bringing it all together

What is a cloud native application?

Cloud native applications

- If you have to implement the same functionality in each application, it should instead be provided as a service using the platform
- The only thing you'll be left with is the valuable business logic

The two road blocks of cloud native

- Legacy applications
- Legacy culture

Legacy culture looks
like this



Legacy applications look like this



Next steps

- Microservices can be used for more than just web applications
- You can use these same principles and frameworks for creating data processing applications that are easier to manage and deploy

Questions?

dangerous

Thanks!

Twitter:

@kennybastani

Email:

kbastani@pivotal.io