CLOUD NATIVE DEVELOPMENT

Charles Moulliard

May 17th - Riviera Dev

https://goo.gl/kgxCRi

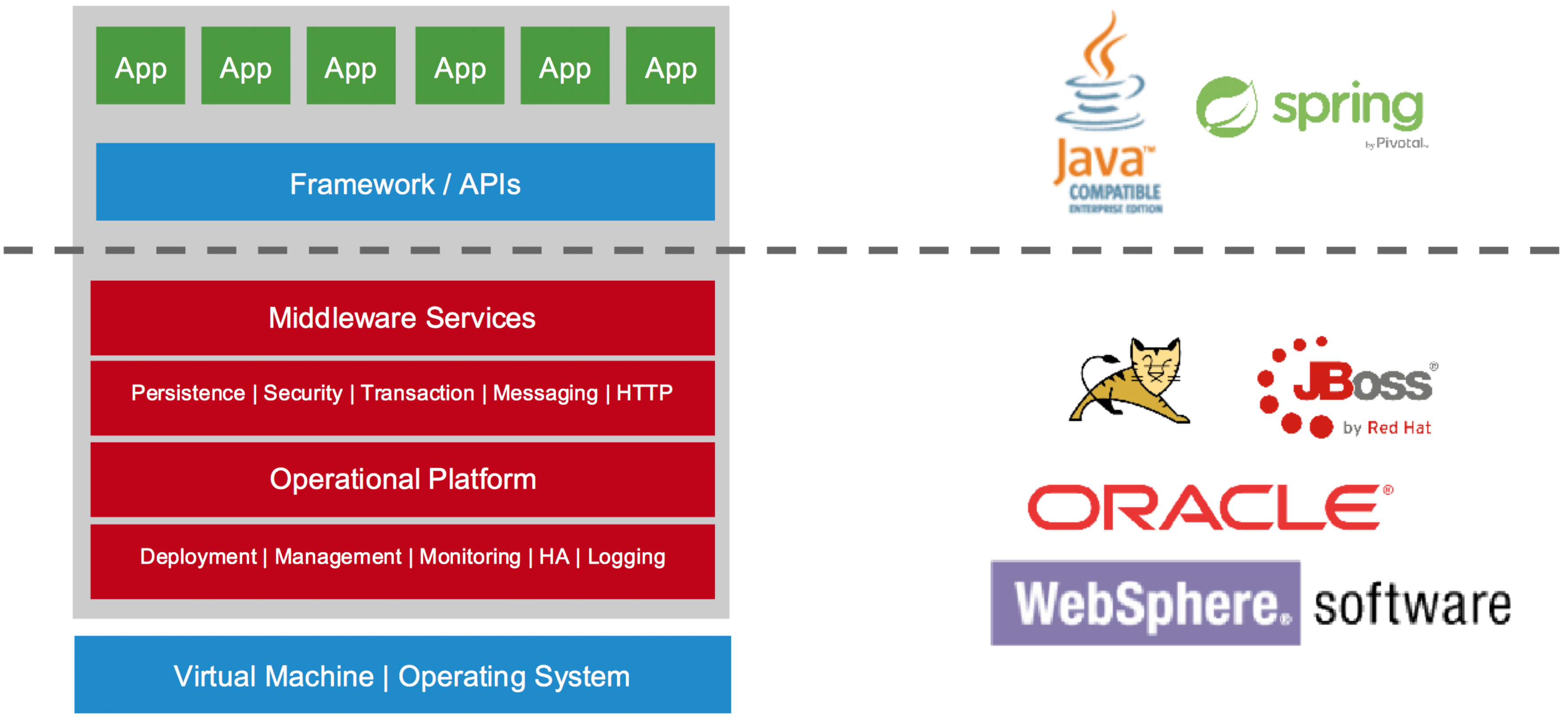
WHO

- Charles Moulliard
- Software Eng. Manager (SpringBoot)
- Technology evangelist
- Twitter: @cmoulliard
- Email: cmoulliard@redhat.com

AGENDA

- Cloud Native Development
- Principles
- What do **I** expect as **C**oder
- Demo time

THE APPSERVER 2000-2014



App | App | App | App | App | App

Framework / APIs

Middleware Services

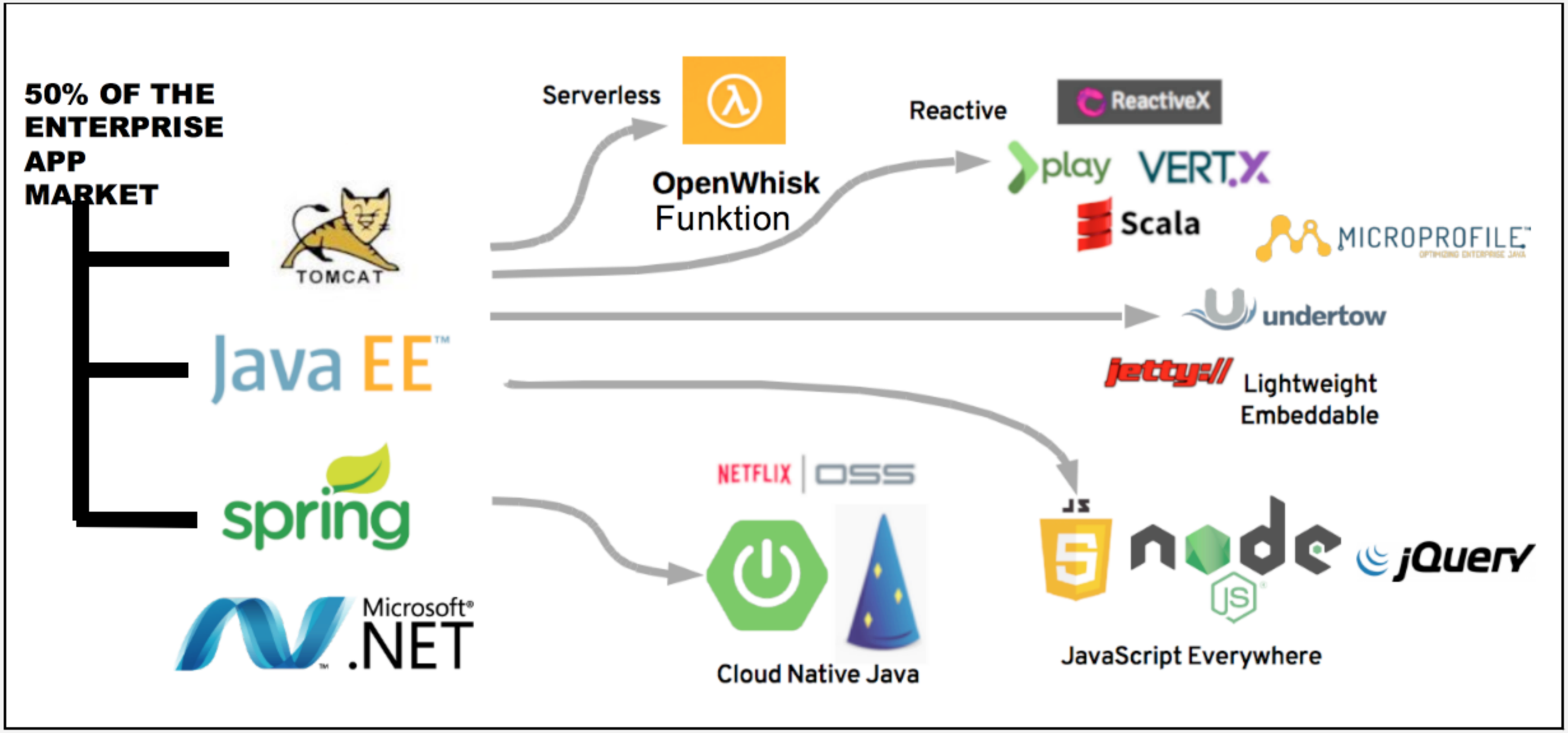
Persistence | Security | Transaction | Messaging | HTTP

Operational Platform

Deployment | Management | Monitoring | HA | Logging

Virtual Machine | Operating System

WHAT CHANGED ...



50% OF THE ENTERPRISE APP MARKET

Serverless

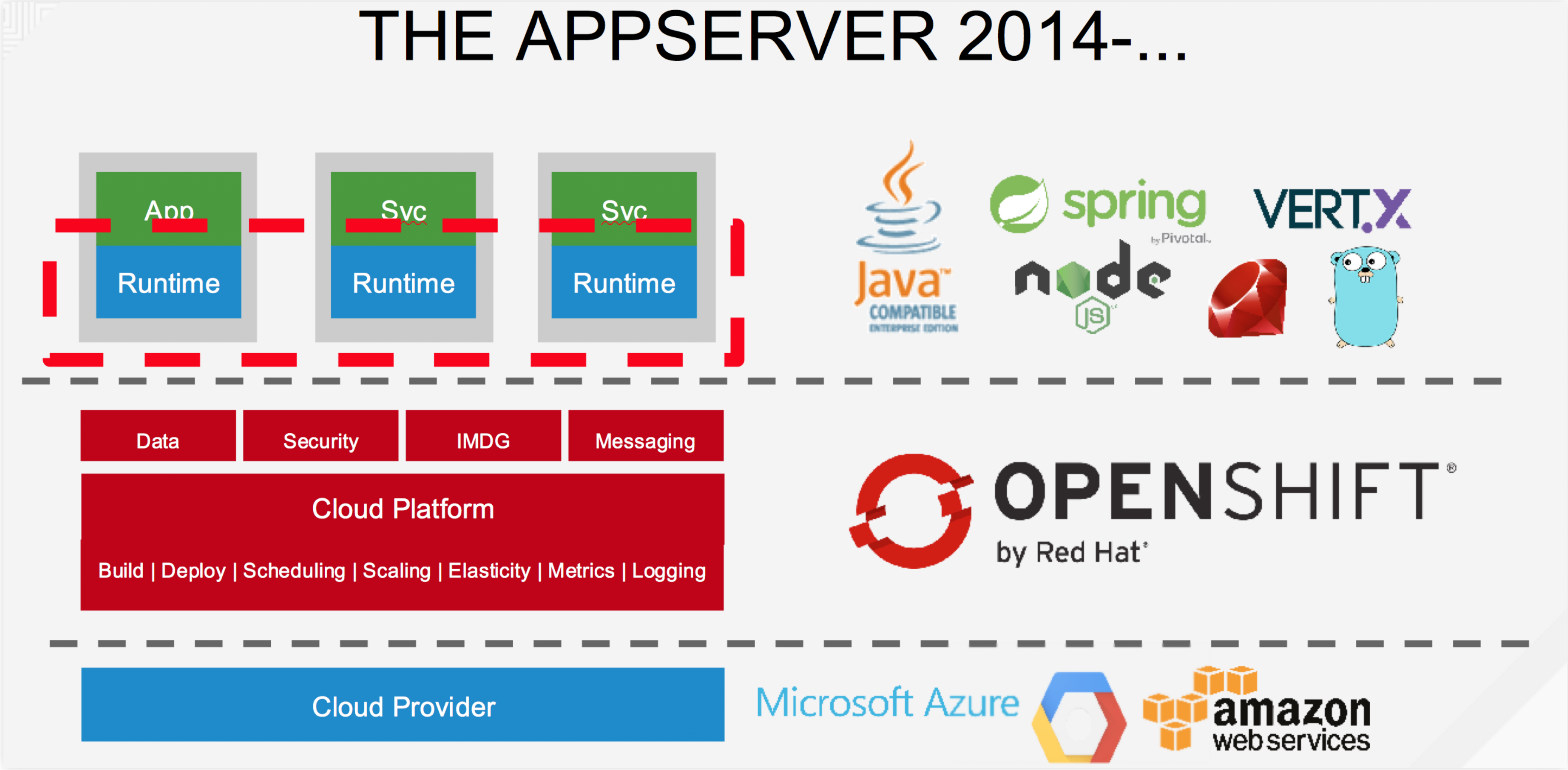
OpenWhisk
Funktion

Reactive

Lightweight
Embeddable

Cloud Native Java

JavaScript Everywhere



THE APPSERVER 2014-...

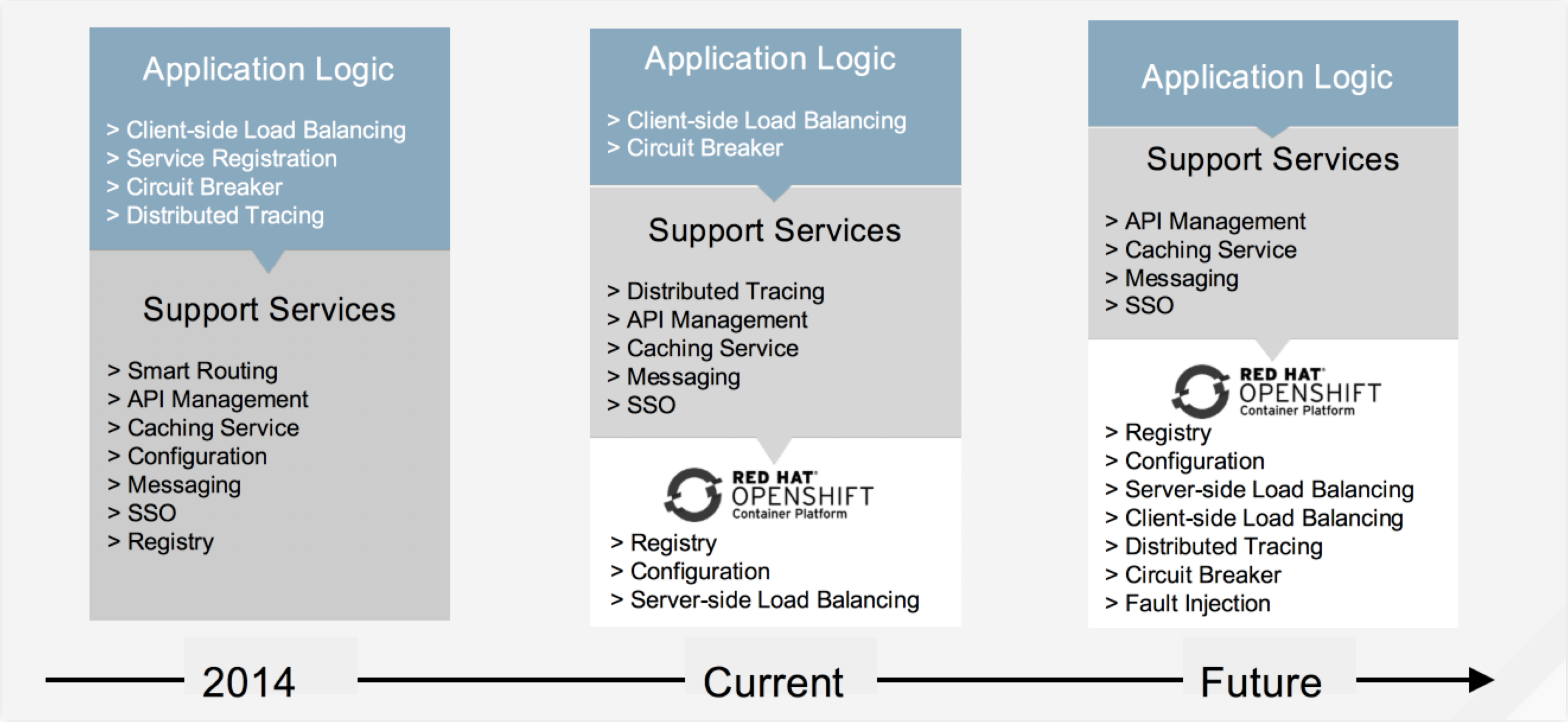
App | Svc | Svc
Runtime | Runtime | Runtime

Data | Security | IMDG | Messaging

Cloud Platform

Build | Deploy | Scheduling | Scaling | Elasticity | Metrics | Logging

Cloud Provider



Application Logic

> Client-side Load Balancing
> Service Registration
> Circuit Breaker
> Distributed Tracing

Support Services

> Smart Routing
> API Management
> Caching Service
> Configuration
> Messaging
> SSO
> Registry

Application Logic

> Client-side Load Balancing
> Circuit Breaker

Support Services

> Distributed Tracing
> API Management
> Caching Service
> Messaging
> SSO

RED HAT OPENSHIFT Container Platform

> Registry
> Configuration
> Server-side Load Balancing

Application Logic

Support Services

> API Management
> Caching Service
> Messaging
> SSO

RED HAT OPENSHIFT Container Platform

> Registry
> Configuration
> Server-side Load Balancing
> Client-side Load Balancing
> Distributed Tracing
> Circuit Breaker
> Fault Injection

IT GOALS

- Speed to develop
- Agility to deliver new features
- Increase margin (cost)
- Maximise infrastructure/tools

DEFINITION

“Cloud-native is an approach to **build** and **run** applications that can **leverage** the capabilities of the cloud platform”

PRINCIPLE - 1

- **Adopt Linux Container**
- unit of packaging
- executable
- portable

PRINCIPLE - 2

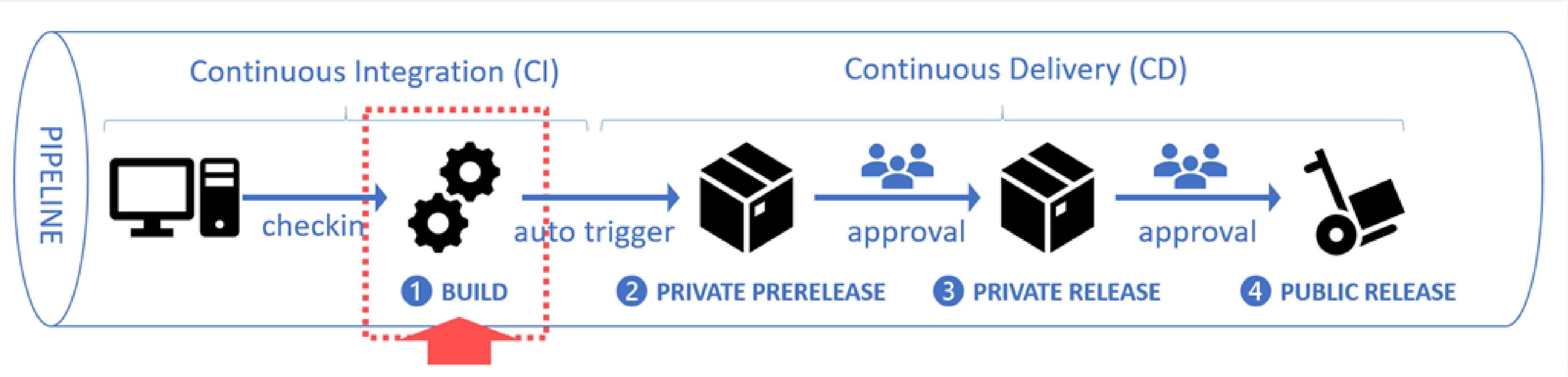
- Design **MicroService** based Architecture
- Isolation
- Health Check
- Circuit Breaker
- Scalability

PRINCIPLE - 3

- Use Cloud **Native features**
- Access provisioned **MicroServices**
- RBAC & Security
- Consume Services from **Catalog**
- Routing, ACL, A/B testing

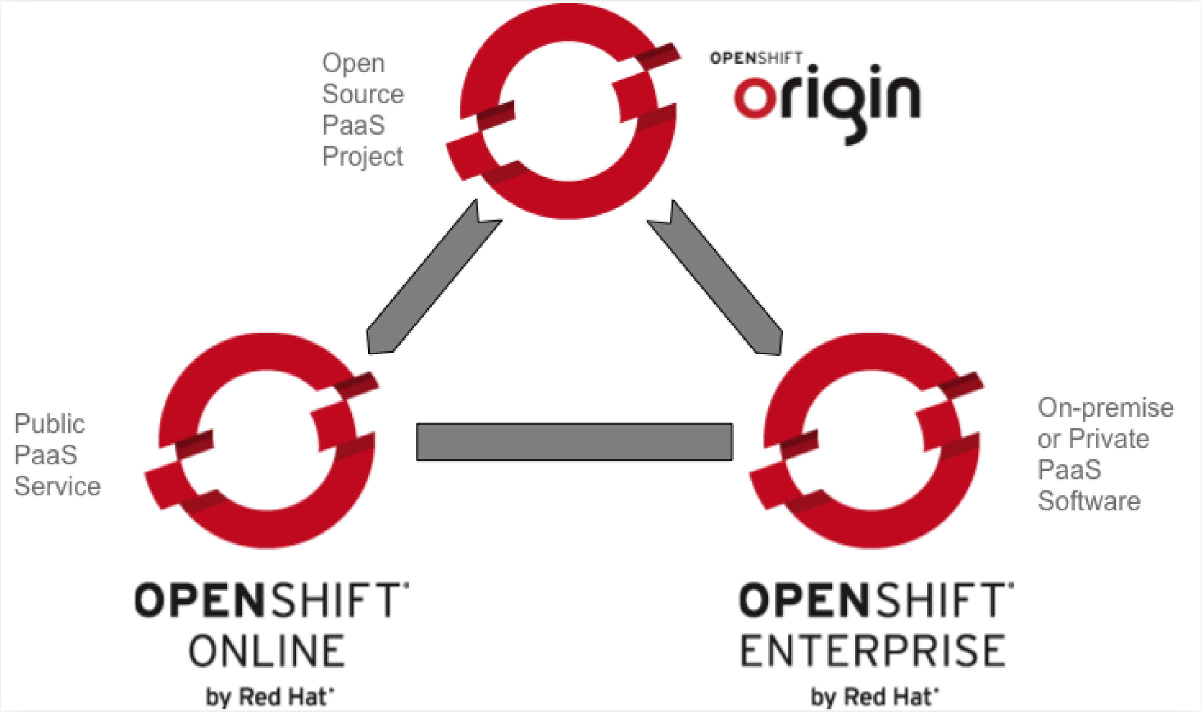
PRINCIPLE - 4

- **DevOps** : CI/CD pipelines to automate the build/deployment process

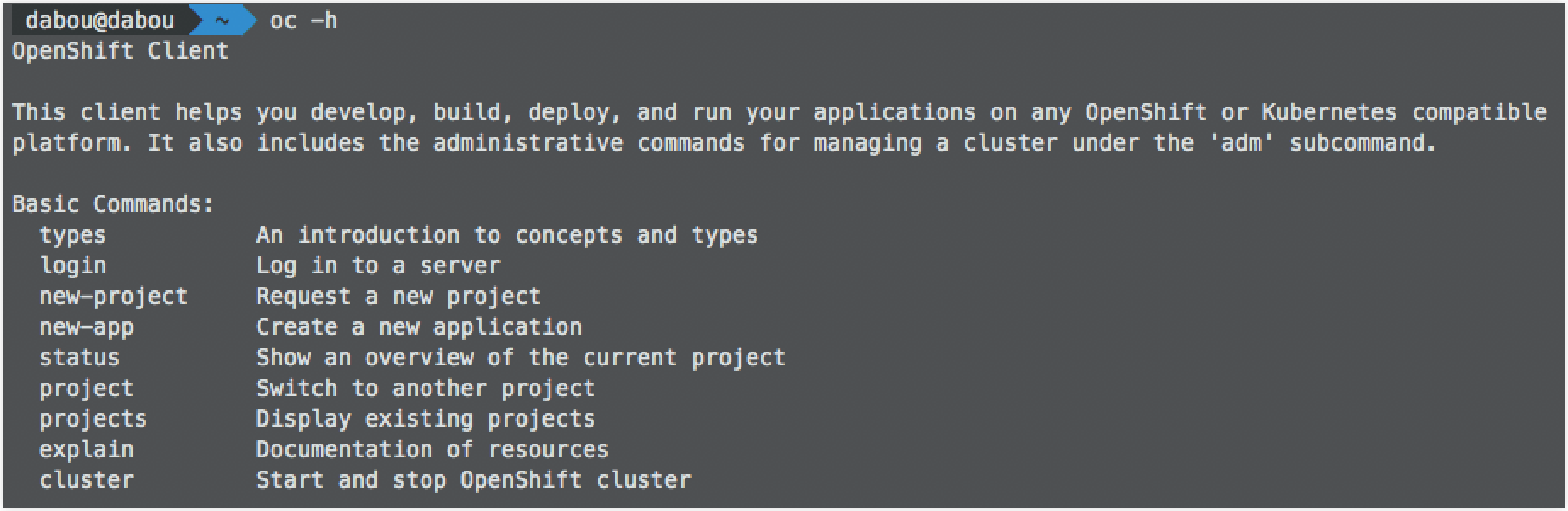


WHAT DO I EXPECT AS CODER

CLOUD MACHINE

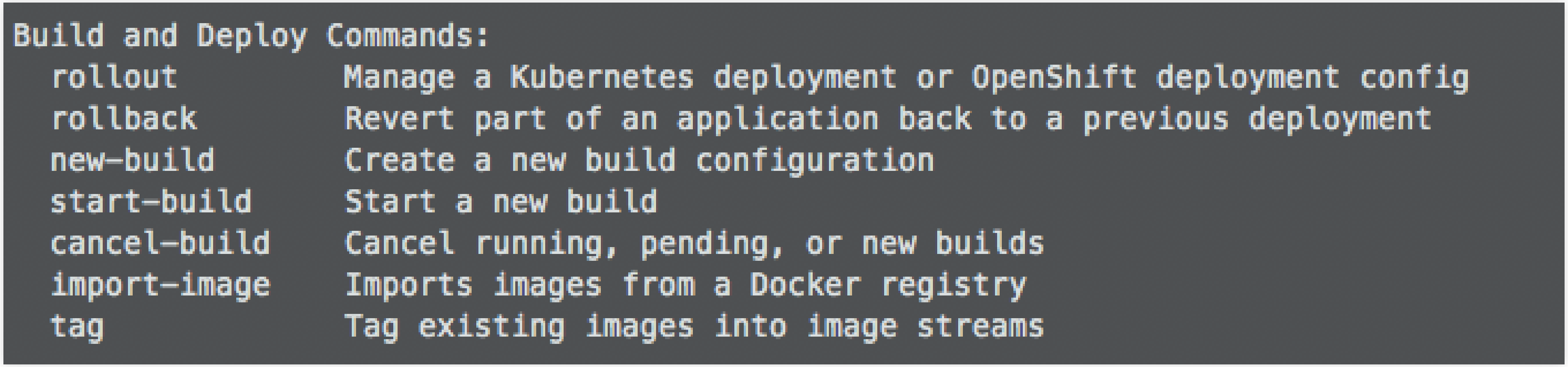


TOOL - MANAGE



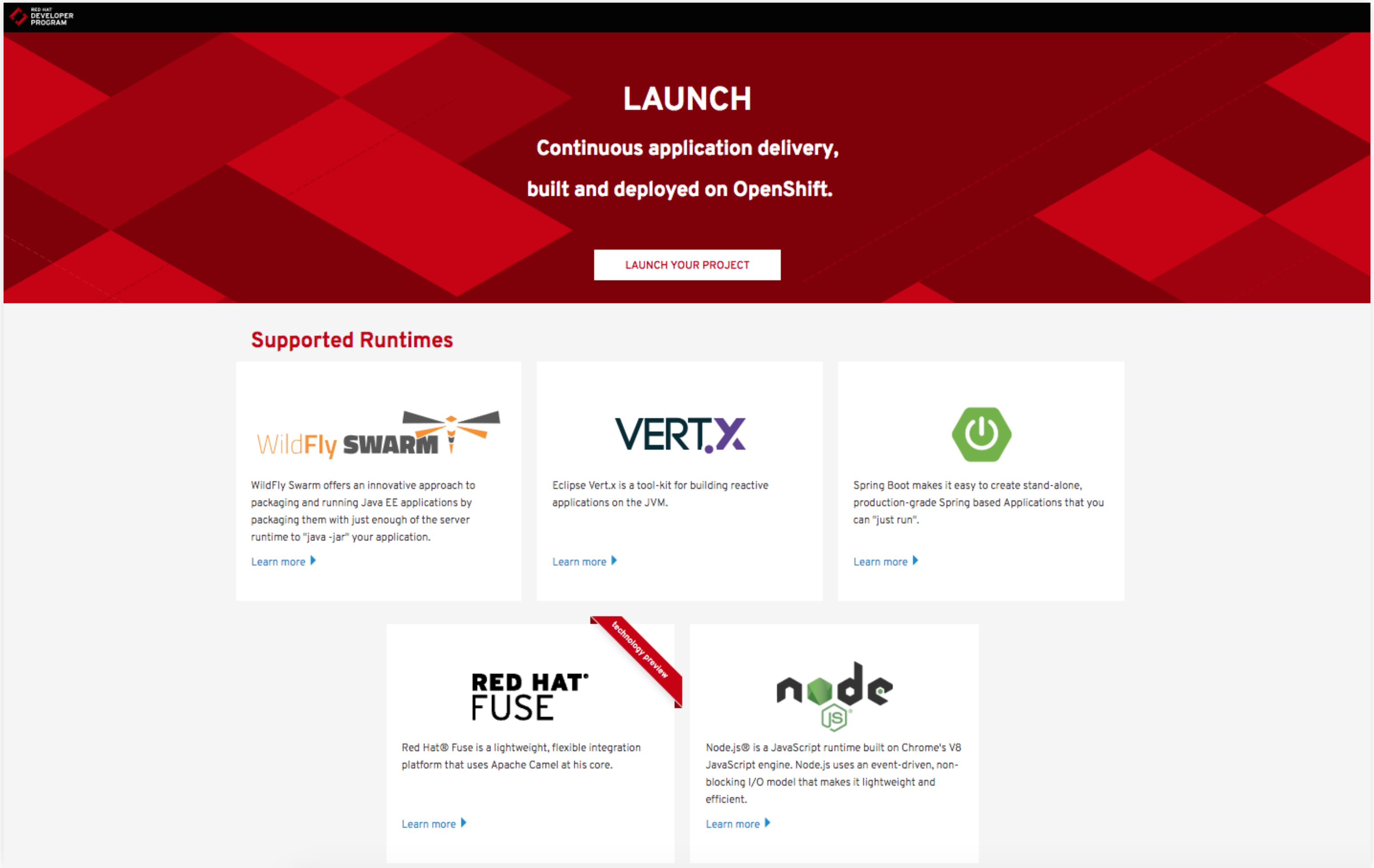
```
dabou@dabou  ~  oc -h
OpenShift Client

This client helps you develop, build, deploy, and run your applications on any OpenShift or Kubernetes compatible
platform. It also includes the administrative commands for managing a cluster under the 'adm' subcommand.

Basic Commands:
  types          An introduction to concepts and types
  login          Log in to a server
  new-project    Request a new project
  new-app        Create a new application
  status         Show an overview of the current project
  project        Switch to another project
  projects       Display existing projects
  explain        Documentation of resources
  cluster        Start and stop OpenShift cluster
```

BUILD - DEPLOY



```
Build and Deploy Commands:
  rollout         Manage a Kubernetes deployment or OpenShift deployment config
  rollback        Revert part of an application back to a previous deployment
  new-build       Create a new build configuration
  start-build     Start a new build
  cancel-build    Cancel running, pending, or new builds
  import-image    Imports images from a Docker registry
  tag             Tag existing images into image streams
```

TOOLBOX



LAUNCH

Continuous application delivery, built and deployed on OpenShift.

LAUNCH YOUR PROJECT

Supported Runtimes

WildFly Swarm offers an innovative approach to packaging and running Java EE applications by packaging them with just enough of the server runtime to "java -jar" your application.

Learn more ▸

Eclipse Vert.x is a tool-kit for building reactive applications on the JVM.

Learn more ▸

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

Learn more ▸

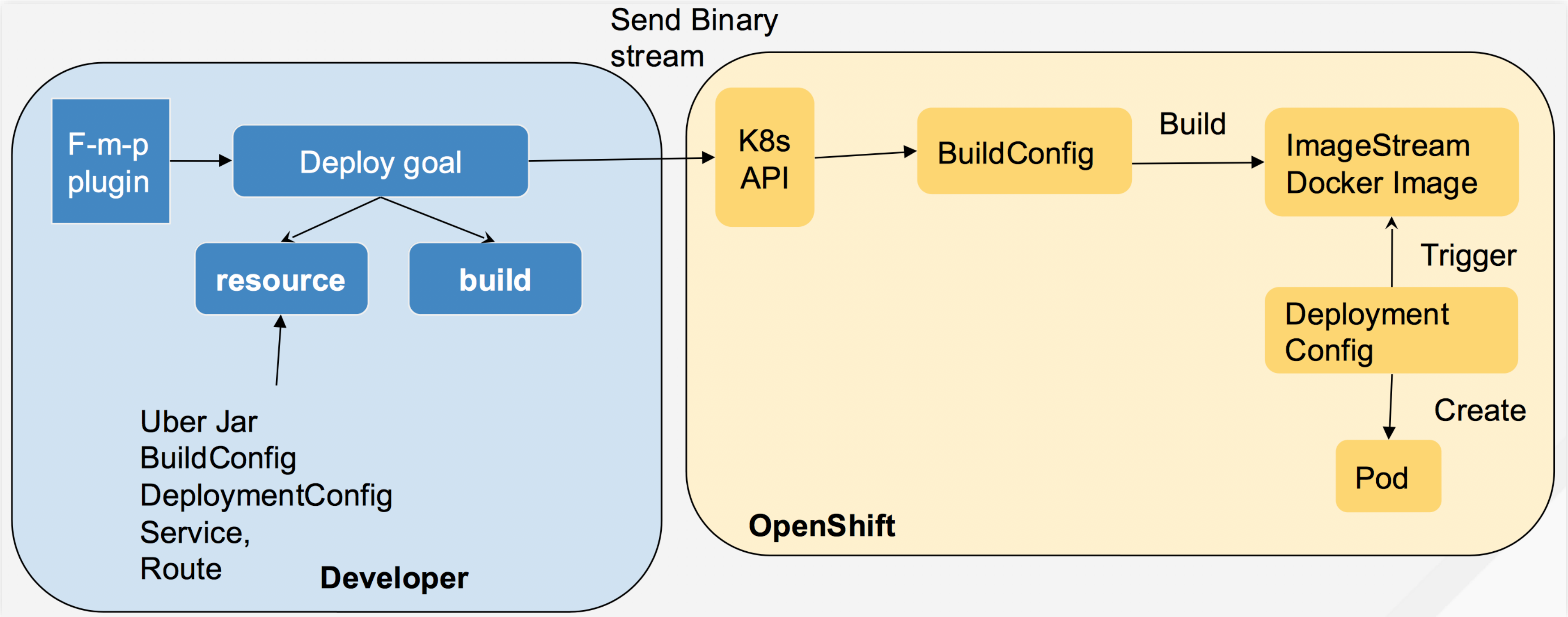
Red Hat® Fuse is a lightweight, flexible integration platform that uses Apache Camel at his core.

Learn more ▸

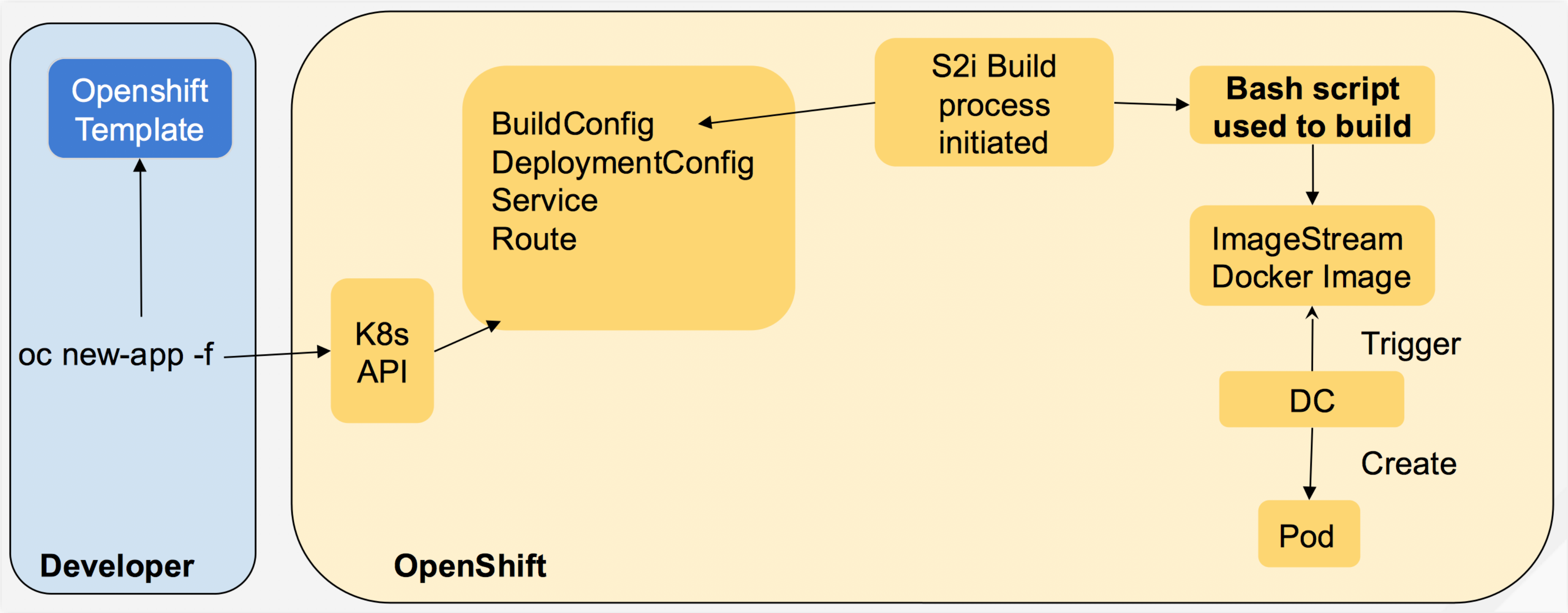
Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

Learn more ▸

LOCAL BUILD → JAR PUSH



PUSH SOURCE → BUILD ON OPENSHIFT



Openshift Template

oc new-app -f

Developer

K8s API

BuildConfig
DeploymentConfig
Service
Route

S2i Build process initiated

Bash script used to build

ImageStream
Docker Image

Trigger

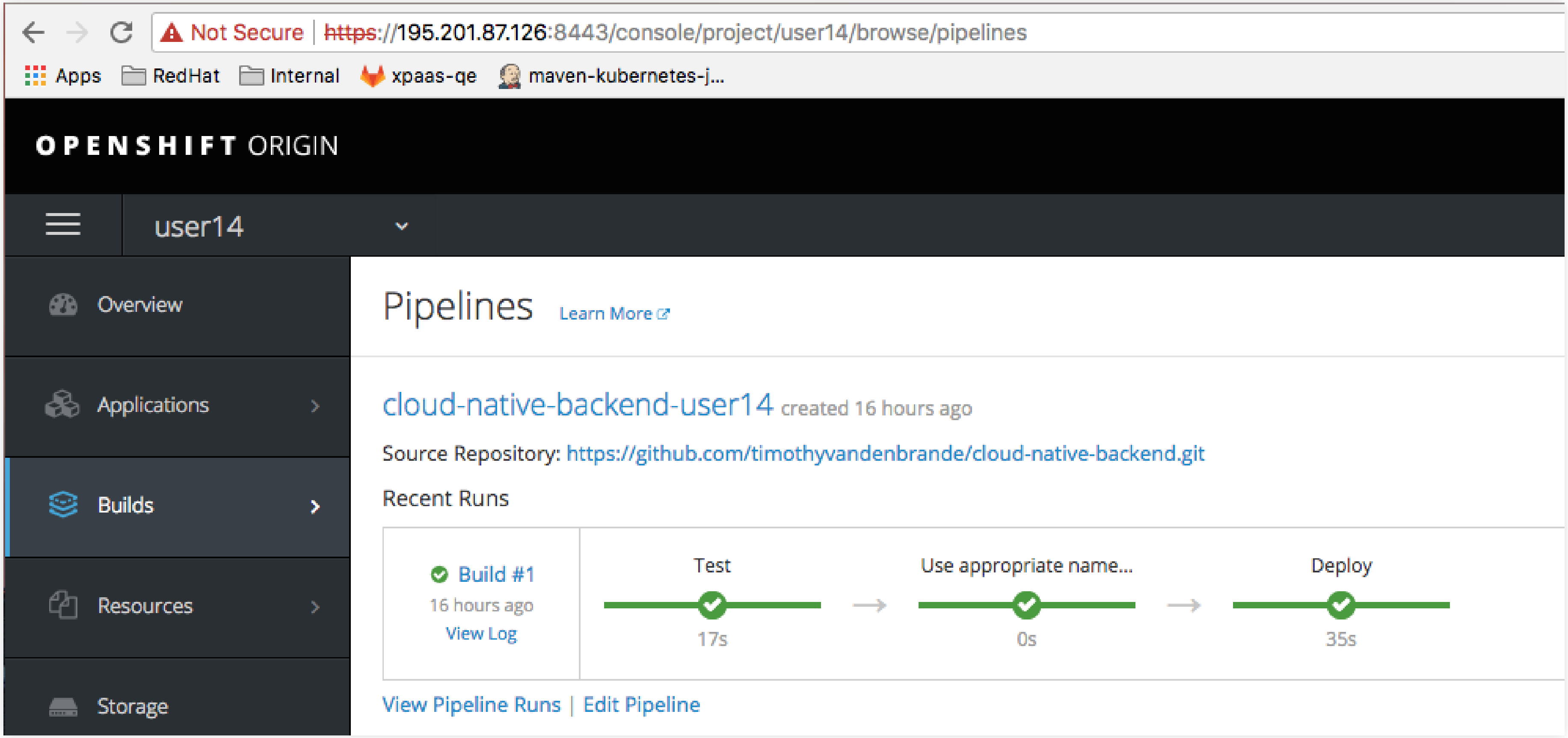
DC

Create

Pod

OpenShift

AUTOMATED → JENKINS



Pipelines

Learn More

cloud-native-backend-user14 created 16 hours ago

Source Repository: https://github.com/timothyvandenbrande/cloud-native-backend.git

Recent Runs

✓ Build #1
16 hours ago
View Log

Test → Use appropriate name... → Deploy

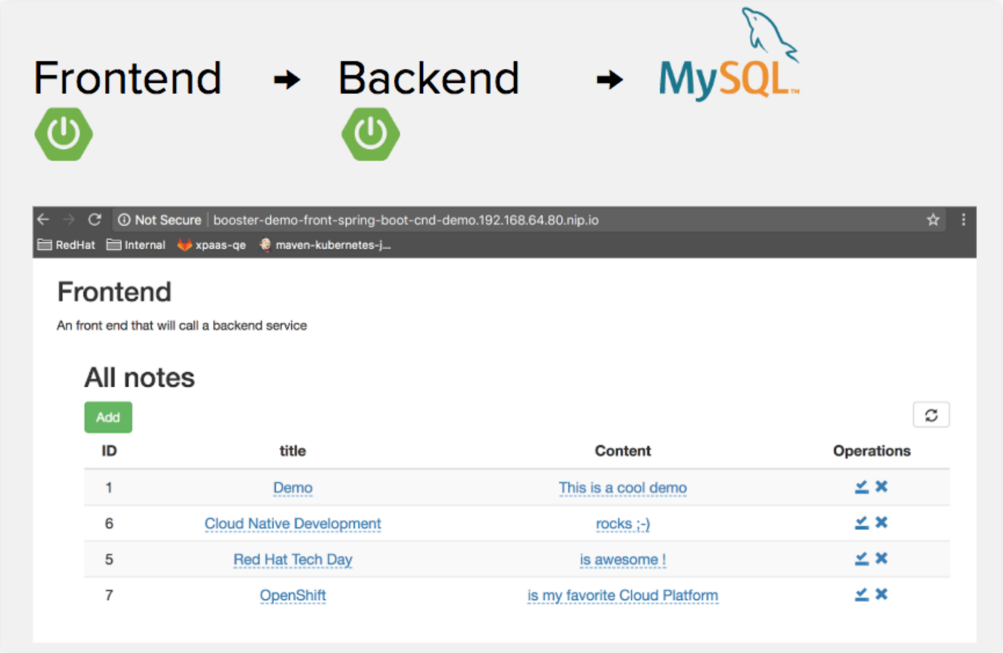
17s → 0s → 35s

View Pipeline Runs | Edit Pipeline

WHAT'S ELSE

- Service Catalog
- Security (RBAC,Keycloak - OAuth2,…)
- Metrics (Prometheus, Actuator)
- Remote Debugging
- Integration testing (Arquillian)
- Logging (Jaeger)
- Routing/ACL/CircuitBreaker … (Google Istio)

DEMO



Not Secure | booster-demo-front-spring-boot-cnd-demo.192.168.64.80.nip.io

Frontend

An front end that will call a backend service

All notes

ID	title	Content	Operations
1	Demo	This is a cool demo	✓ ✖
6	Cloud Native Development	rocks ;-)	✓ ✖
5	Red Hat Tech Day	is awesome !	✓ ✖
7	OpenShift	is my favorite Cloud Platform	✓ ✖

https://github.com/snowdrop/cloud-native-lab