

Application and infrastructure continuous delivery

Jim Minter
Principal Solution Architect, UK
jminter@redhat.com

Keith Lynch
Application Development Business Lead, UK
keith@redhat.com

24/06/2015

Agenda

1. Introduction
2. Example monolithic → microservices migration
3. Hosting microservices in OpenShift v3
4. Continuous delivery demos

Introduction

Problems software development functions face

Problem	Solution
Releases are painful and take a lot of time. Monolithic 12month→multi year releases	
Inefficient code promotion	
Architectural issues identified late	
Operational issues discovered late	
Functional regressions	
Replicating environments (it works on my machine)	
Zero downtime	

Problems software development functions face

Problem	Solution
Releases are painful and take a lot of time. Monolithic 12month→multi year releases	Frequent smaller releases, use agile development methodologies, adopt a micro services architectural style
Inefficient code promotion	Build once, deploy many
Architectural issues identified late	Run regular and automated performance tests as part of your build
Operational issues discovered late	Have a unified deployment mechanism
Functional regressions	Run regression testing as part of your build (BDD/ATDD)
Replicating environments (it works on my machine)	Docker containers
Zero downtime	Canary / Blue-Green deployments

Example scenario

- We took a monolithic app (JBoss TicketMonster)
- We refactored it to a microservices architecture
- We used OpenShift v3, JBoss EAP and JBoss Fuse (and OpenStack ;-)
- We embedded the application into a CD pipeline
- We'll show you how we did it...

TicketMonster: monolithic → microservices



TicketMonster.

A JBoss Example.

TicketMonster is an online ticketing demo application that gets you started with JBoss technologies, and helps you learn and evaluate them.

[Buy tickets now](#)

Fork me on GitHub

Monolithic TicketMonster

HTML 5 / AngularJS

CDI

EJB

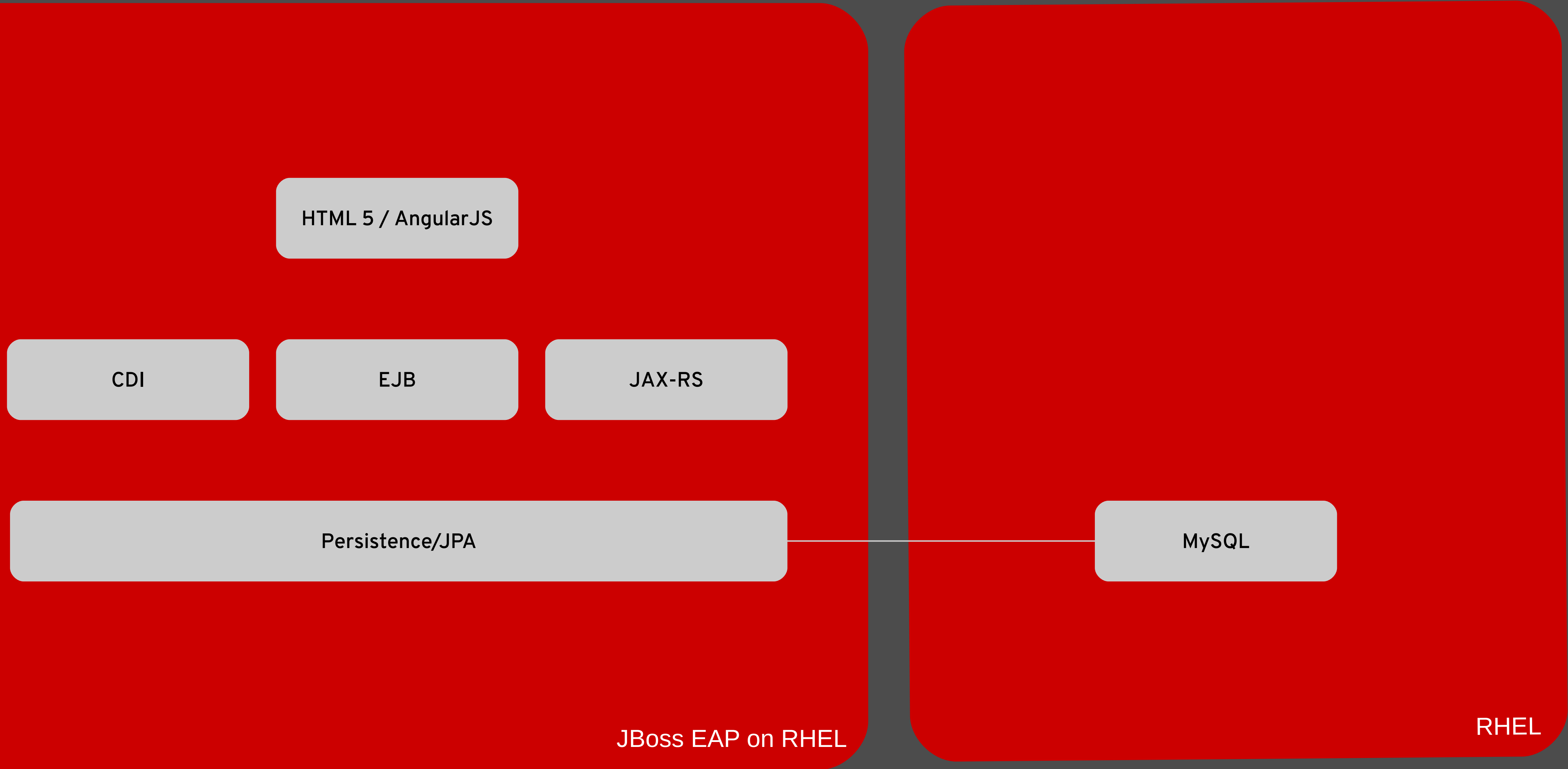
JAX-RS

Persistence/JPA

MySQL

JBoss EAP on RHEL

RHEL



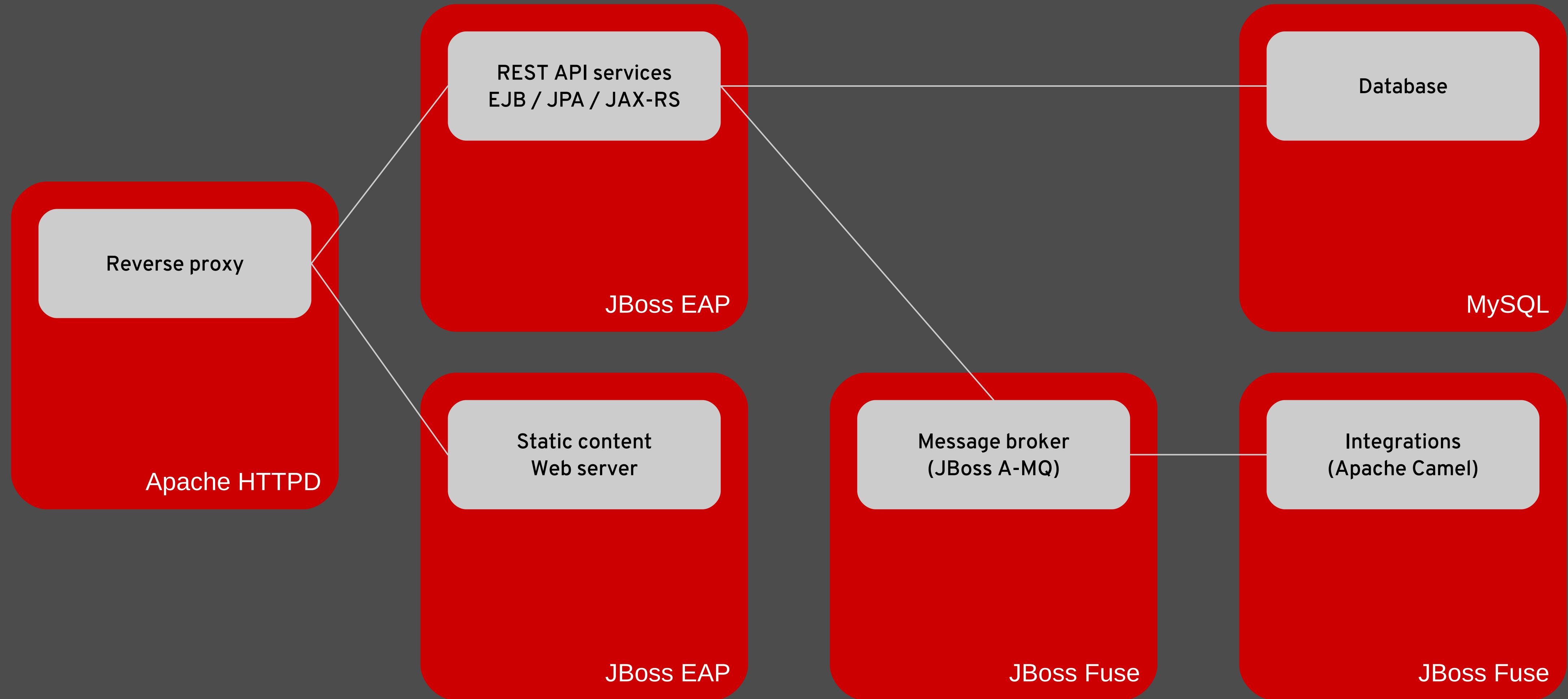
Microservices Ticket Monster



Microservices Ticket Monster



Microservices Ticket Monster



Kubernetes 101

- Pods
- Services
- Replication Controllers
- ...

OpenShift 101

- Pods
- Services
- Replication Controllers
- ...
- Build Configs
- Deployment Configs
- Routes

Implementation in OpenShift 3 (beta4)

- 1 git repo per service
- Separate build & deploy scripts
- Standard EAP and xPaas docker images
- Kubernetes services used to link components
- Source-to-image used to build all components except MySQL and A-MQ
 - (so API services, integration routes, reverse proxy...)

Demo 1

Building towards the CD pipeline

Problems software development functions face

Problem	Solution
Releases are painful and take a lot of time. Monolithic 12month→multi year releases	Frequent smaller releases, use agile development methodologies, adopt a micro services architectural style
Inefficient code promotion	Build once, deploy many
Architectural issues identified late	Run regular and automated performance tests as part of your build
Operational issues discovered late	Have a unified deployment mechanism
Functional regressions	Run regression testing as part of your build (BDD/ATDD)
Replicating environments (it works on my machine)	Docker containers
Zero downtime	Canary / Blue-Green deployments

Defining Continuous Delivery from DevOps

peer review

less management approval required

rigorous automated testing

the ability to create entire environments on demand

one-click deployments

* <https://puppetlabs.com/sites/default/files/2014-state-of-devops-report.pdf> - 9,200 organizations scattered across 110 countries

* Many of the core elements are a reflection of the culture shift required to successfully implement DevOps.- Gene Kim

Defining Continuous Delivery from DevOps

peer review

less management approval required

rigorous automated testing

the ability to create entire environments on demand

one-click deployments

* <https://puppetlabs.com/sites/default/files/2014-state-of-devops-report.pdf> - 9,200 organizations scattered across 110 countries

* Many of the core elements are a reflection of the culture shift required to successfully implement DevOps.- Gene Kim

Continuous Delivery

Git

Jenkins

Container registry

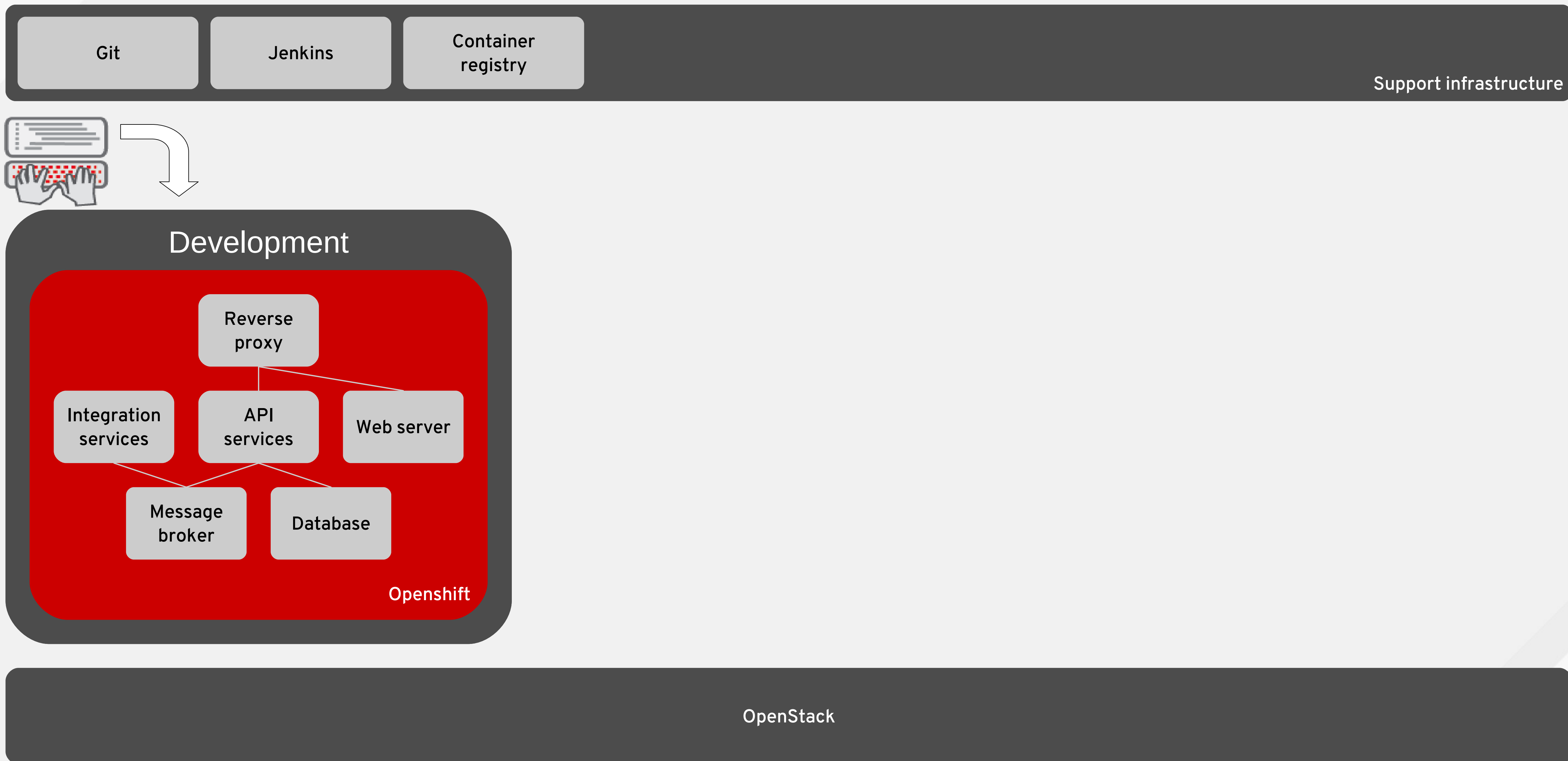
Support infrastructure

Development

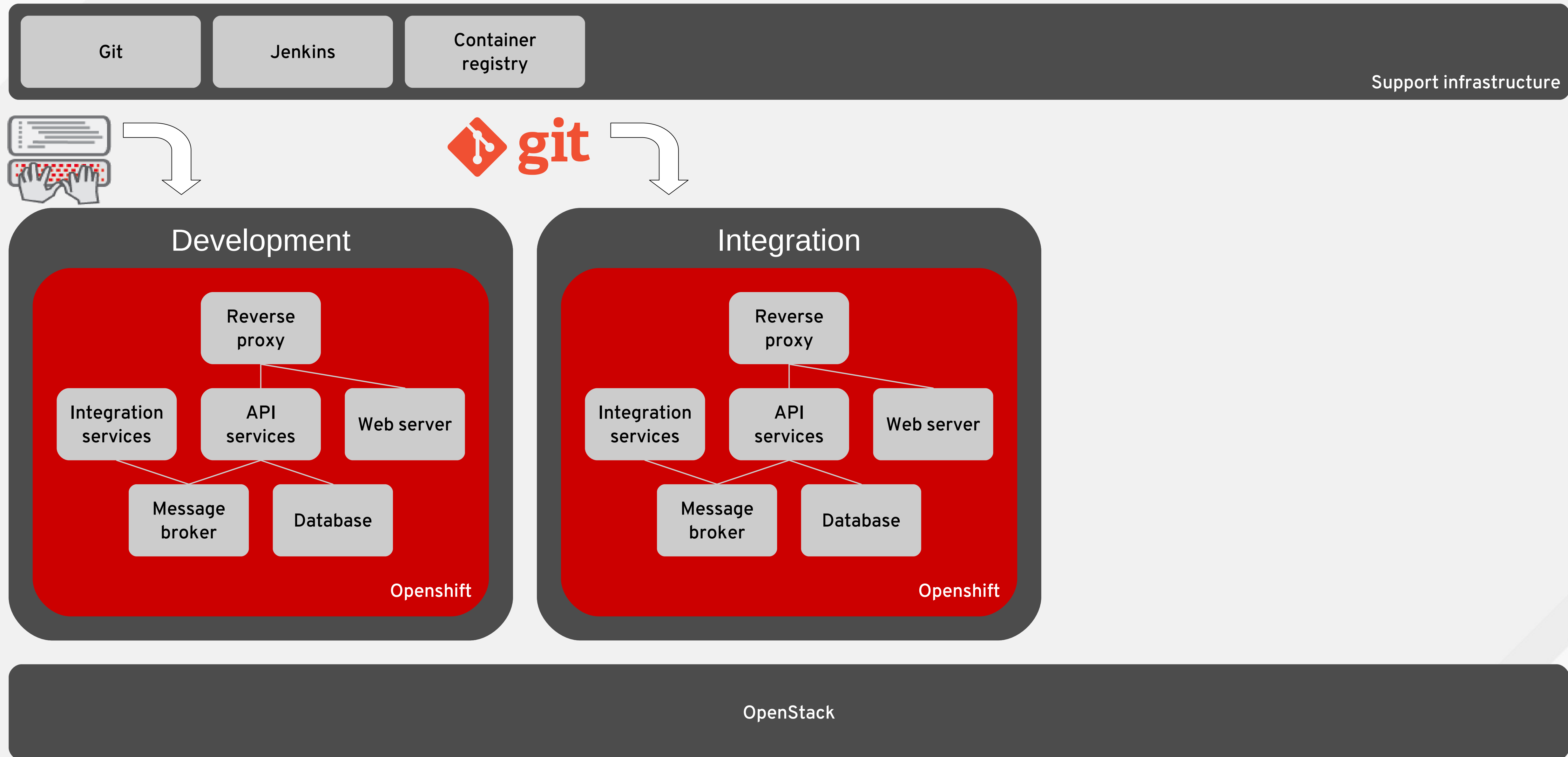
Openshift

OpenStack

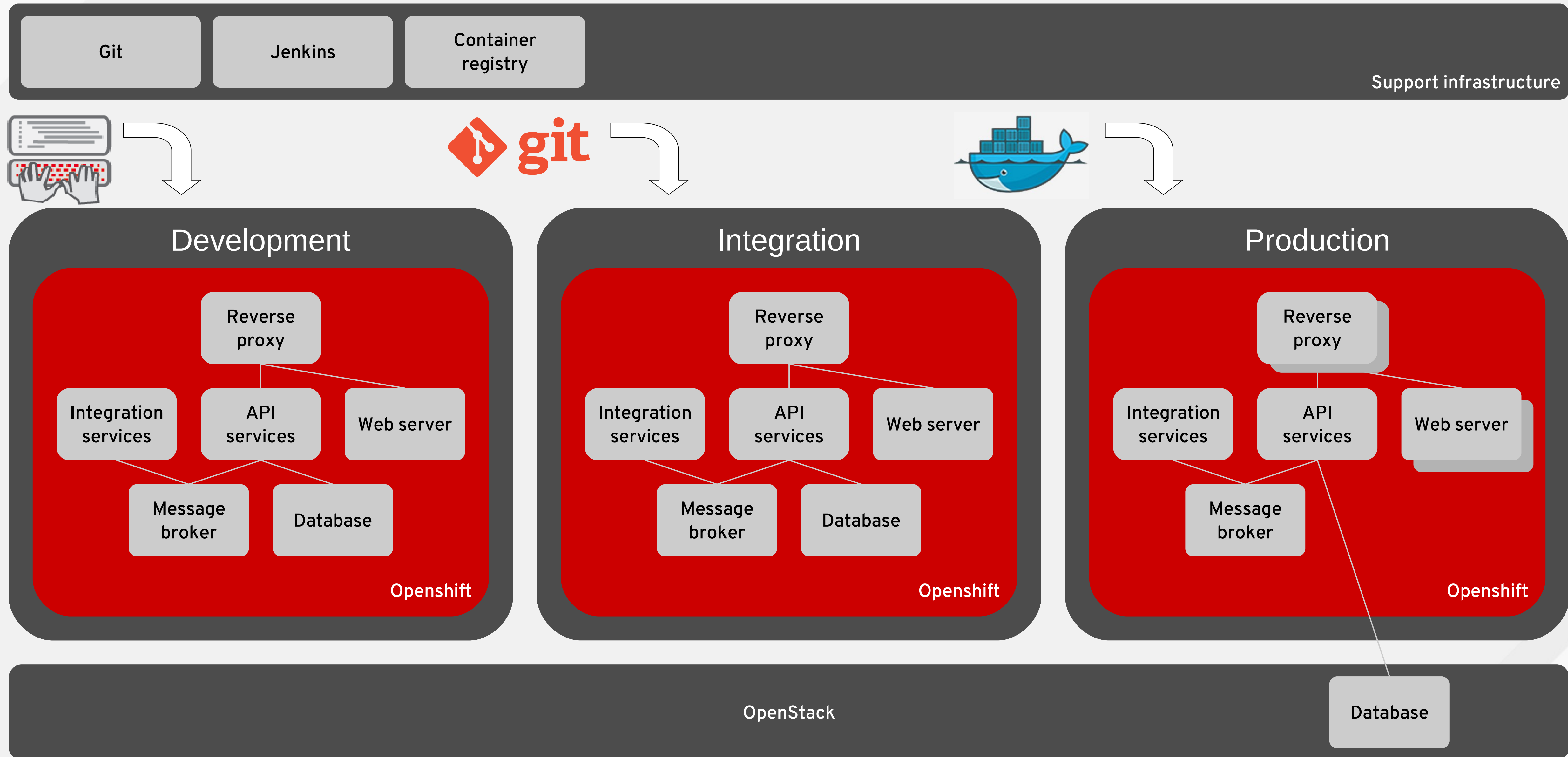
Continuous Delivery



Continuous Delivery



Continuous Delivery



Workflow stories #1: development

- **As a** developer **I want** to clone the code of an application **so that** I can deliver a new feature.
- **As a** developer **I want** an environment where I can deploy changes to **so that** I can develop and test the feature I'm working on.
 1. Fork code in git and checkout
 2. Make change(s) and commit to private repo
 3. OpenShift builds your code and runs unit tests
 4. OpenShift deploys your code in your private environment

Workflow stories #2: route to production

- **As a** developer **I want** to release my tested changes back into a shared integration environment **so that** I can set my feature as done
- **As an** operator **I want** to update a production environment **so that** I can deploy a tested feature set
 1. Merge change(s) upstream
 2. OpenShift builds upstream code and runs unit tests
 3. OpenShift deploys upstream code to integration environment
 4. Jenkins runs integration tests
 5. Jenkins provides gated process to promote to production
 6. Jenkins copies docker images to promote released code

Summary

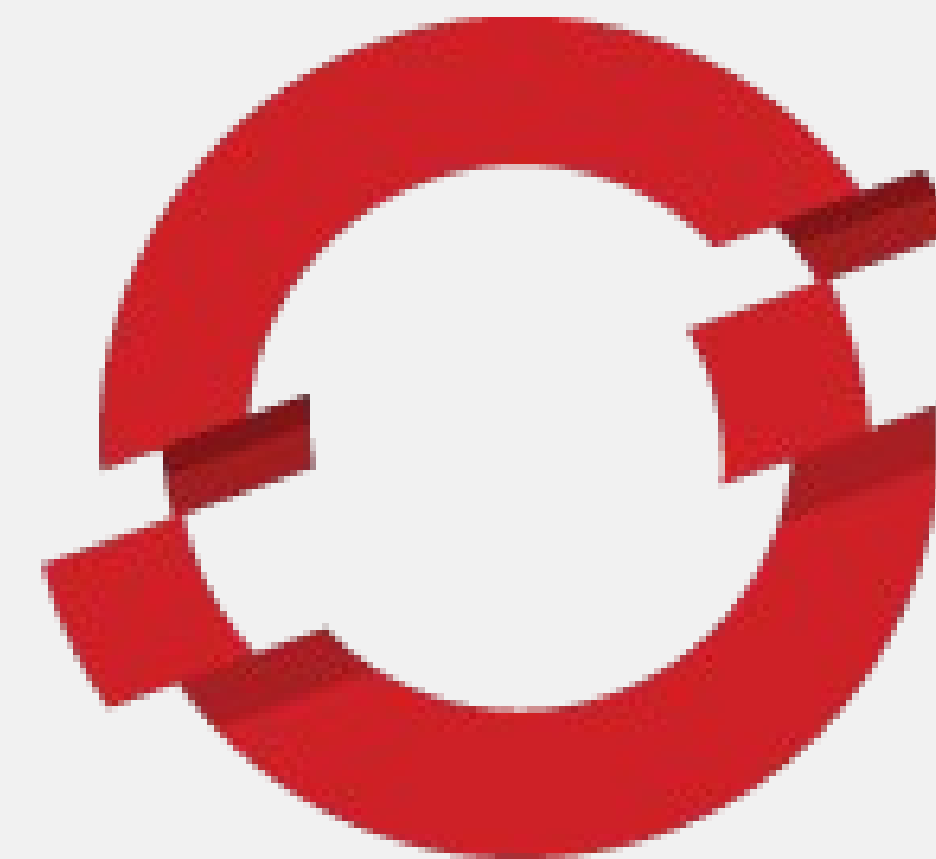
Agenda

1. Introduction
2. Example monolithic → microservices migration
3. Hosting microservices in OpenShift v3
4. Continuous delivery demos

RED HAT® JBOSS®
FUSE

RED HAT® JBOSS®
ENTERPRISE
APPLICATION PLATFORM

RED HAT® JBOSS®
A-MQ



OPENSHIFT®
by Red Hat®



Some resources / next steps

- Today's demo – sources and videos
 - <https://github.com/CICD-Demo/CICD-Demo>
- Things to watch!
 - OpenShift 3 - <https://blog.openshift.com/>
 - Fabric8 - <http://fabric8.io/>
- Our e-mail addresses
 - jminter@redhat.com
 - keith@redhat.com



RED HAT
SUMMIT



SUMMIT BY DAY

PARTY BY NIGHT

JOIN OUR **JBOSS,**
OPENSHIFT,
AND **MOBILE** TEAMS
FOR A NIGHT OF GAMES, DANCING,
AND OPEN CONTAINERS

Visit the Red Hat booth
in Hall D for location
and invitation.

RED HAT **SUMMIT**

**LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.**