



Openshift Trilogy Schedule

PHA: 10/10/2017 **Container Wars: A New Hope**

Openshift - DevOps - Automation Overview

BRNO: 1/11/2017 **Container Wars: API strikes back**

Agile integration & API Workshop

PHA?: 14/11/2017 **Container Wars: Return of the App**

Jedi's App development & DevOps Workshop



@ OCP Trilogy
1/11/2017

OpenShift Trilogy: API strikes back

Jiří Kolář
Solution Architect CZ/SK/CEE
jkolar@redhat.com

Modern App Dev?

Digital Transformation?
It requires an evolution in....



Process



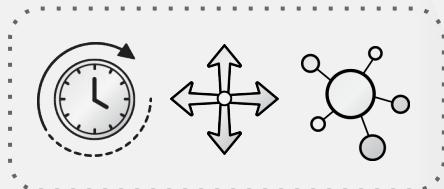
Infrastructure



Architecture

PROCESS ?

PROBLEM:



DEVELOPERS



I.T. OPERATIONS

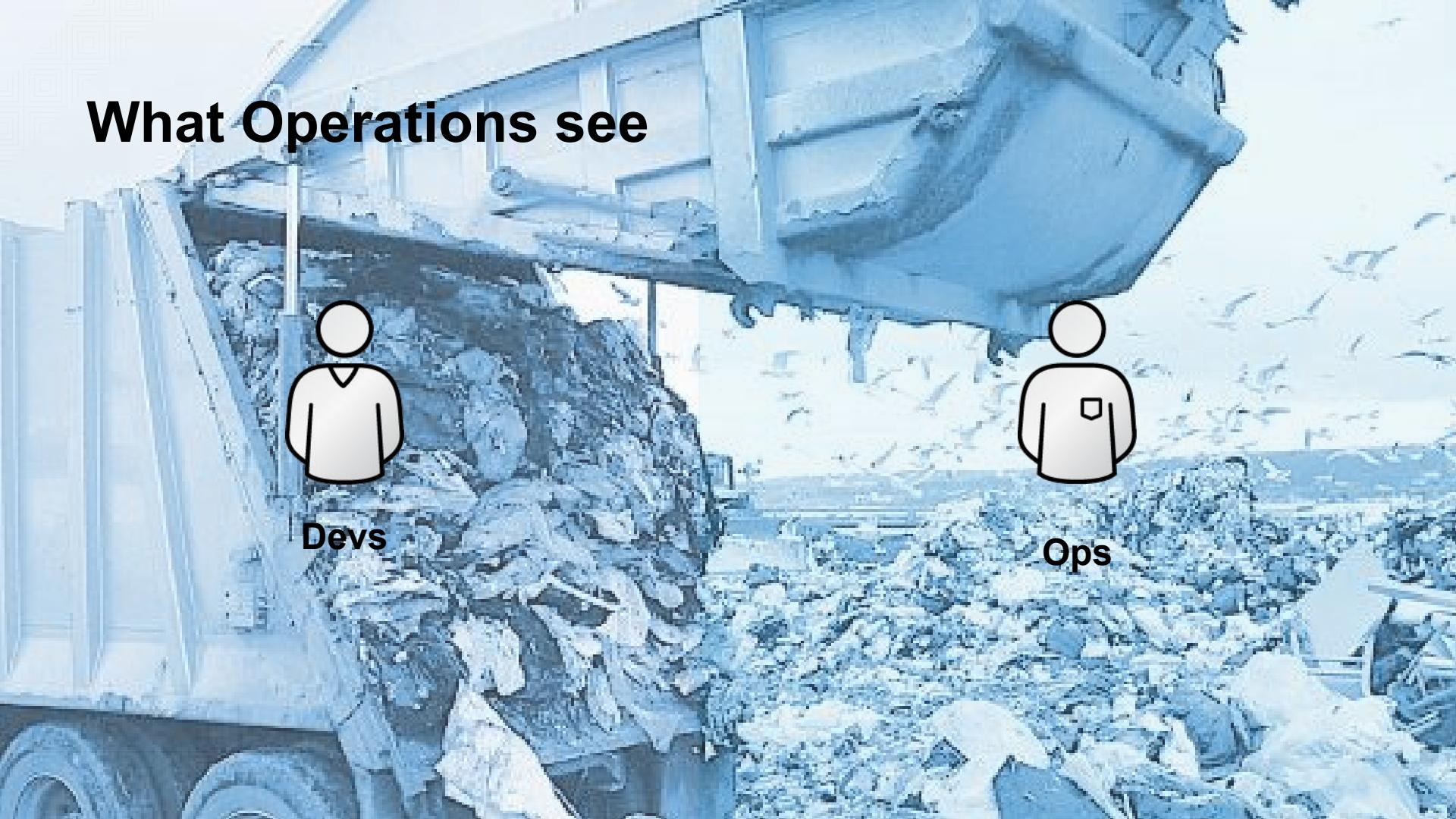
What Devs see



Devs



Ops

A large blue dump truck is shown from a side-on perspective, dumping a massive pile of light-colored rubble or debris onto a larger pile on the ground. The truck's bed is tilted upwards, and the debris is cascading down. The background shows more of the same debris scattered across the ground.

What Operations see

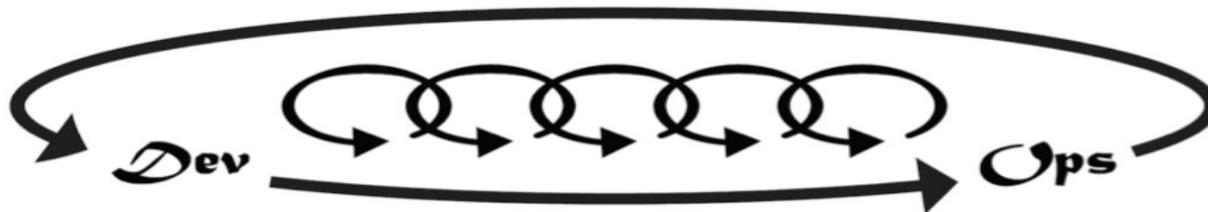


Devs



Ops

SOLUTION:



Key concepts:

- Small changes -> Less Risk
- Delivery pipeline = **Automation!**
- Culture change: Acceptance of failure
- Team takes **ownership and responsibility!**

WHAT IS INFRASTRUCTURE?

PROBLEM:

Craftwork

Physical

How to Build an App:

1. Have Idea
2. Get Budget
3. Submit hardware acquisition request
4. Wait
5. Get Hardware
6. Rack and Stack Hardware
7. Install Operating System
8. Install Operating System Patches
9. Create user Accounts
10. Deploy framework/appserver
11. Deploy testing tools
12. **Code**
13. Test
14. Buy and configure Prod servers
15. Push to Prod
16. Launch
17. Order more servers to meet demand
18. Wait...
19. Deploy new servers
20. Etc.

PROBLEM:

Craftwork

Physical

How to Build an App:

1. Have Idea
2. Get Budget
3. Submit hardware acquisition request
4. Wait
5. Get Hardware
6. Rack and Stack Hardware
7. Install Operating System
8. Install Operating System Patches
9. Create user Accounts
10. Deploy framework/appserver
11. Deploy testing tools
12. **Code**
13. Test
14. Buy and configure Prod servers
15. Push to Prod
16. Launch
17. Order more servers to meet demand
18. Wait...
19. Deploy new servers
20. Etc.

Virtualized

How to Build an App:

1. Have Idea
2. Get Budget
3. Submit VM Request request
4. Wait
5. Deploy framework/appserver
6. Deploy testing tools
7. **Code**
8. Test
9. Configure Prod VMs
10. Push to Prod
11. Launch
12. Request VMs to meet demand
13. Wait
14. Deploy app to new VMs
15. Etc.

PROBLEM:

Craftwork

Physical

How to Build an App:

1. Have Idea
2. Get Budget
3. Submit hardware acquisition request
4. Wait
5. Get Hardware
6. Rack and Stack Hardware
7. Install Operating System
8. Install Operating System Patches
9. Create user Accounts
10. Deploy framework/appserver
11. Deploy testing tools
- Code**
13. Test
14. Buy and configure Prod servers
15. Push to Prod
16. Launch
17. Order more servers to meet demand
18. Wait...
19. Deploy new servers
20. Etc.

Virtualized

How to Build an App:

1. Have Idea
2. Get Budget
3. Submit VM Request request
4. Wait
5. Deploy framework/appserver
6. Deploy testing tools
- Code**
8. Test
9. Configure Prod VMs
10. Push to Prod
11. Launch
12. Request VMs to meet demand
13. Wait
14. Deploy app to new VMs
15. Etc.

Assembly Line

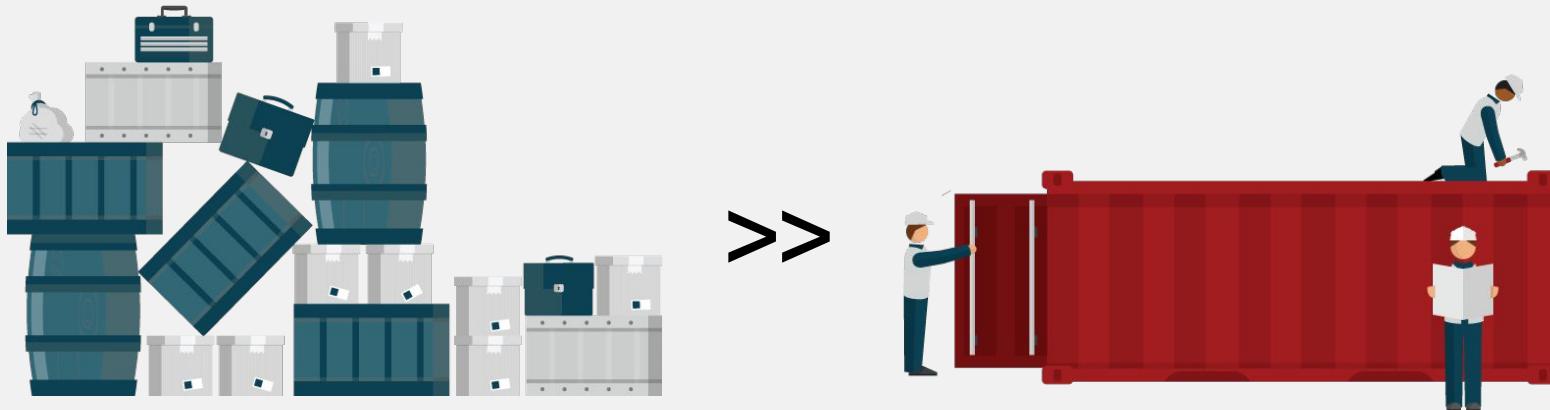
Container PaaS

How to Build an App:

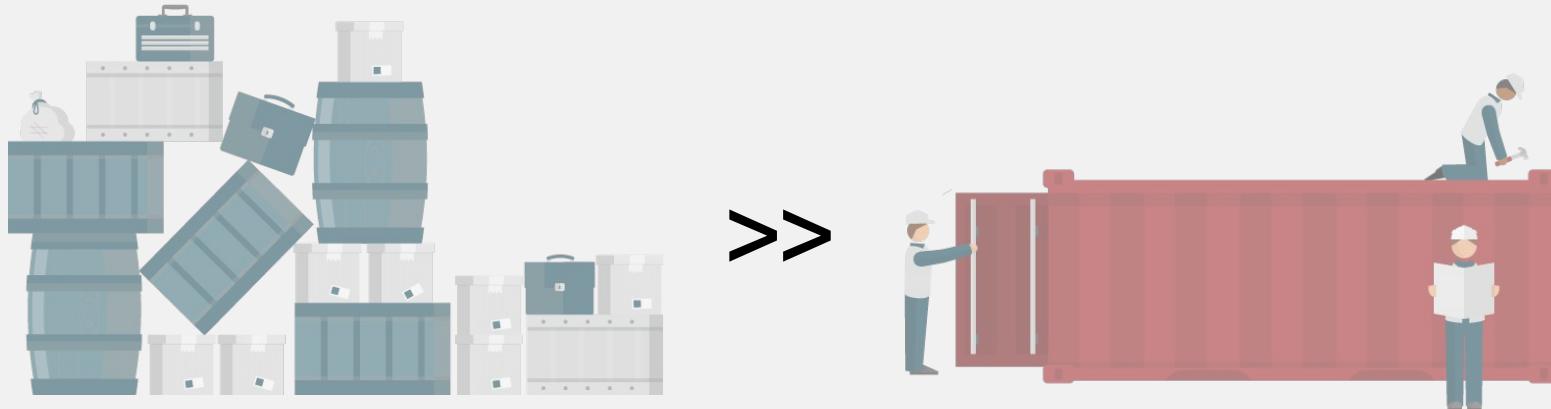
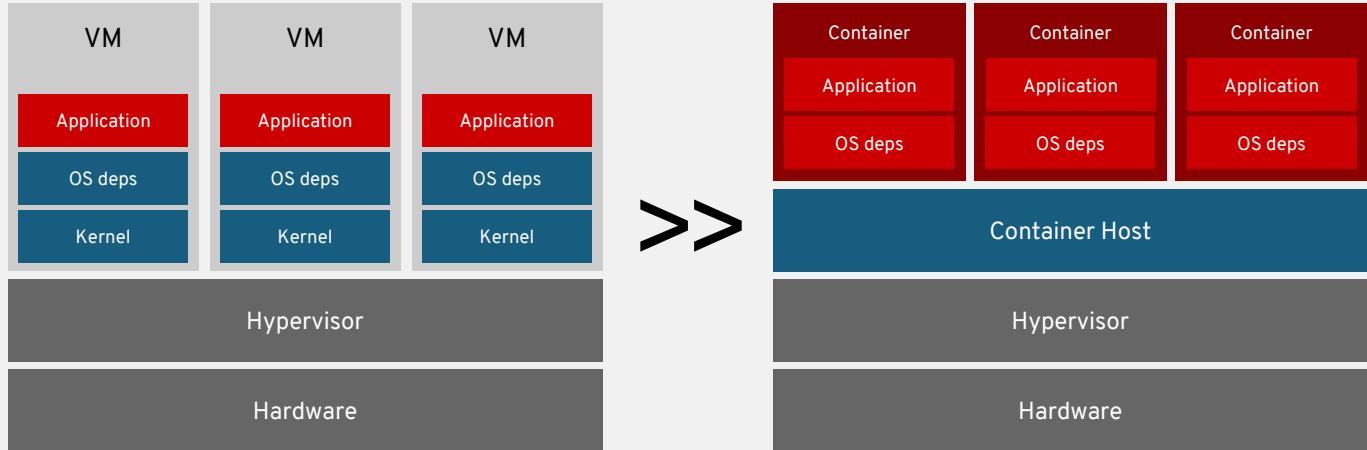
- Have Idea**
- Get Budget**
- Code**
- Test**
- Launch**
- Automatically Scale**



SOLUTION:



Containers!

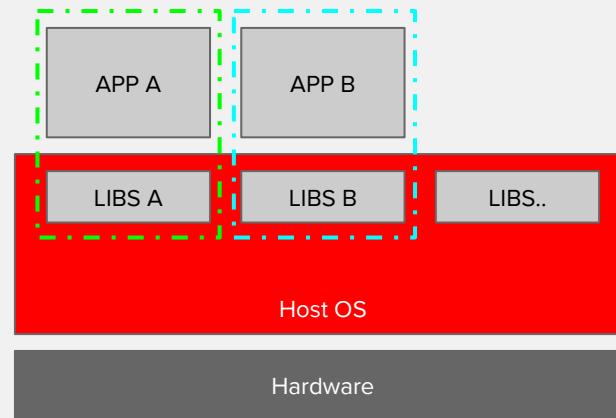




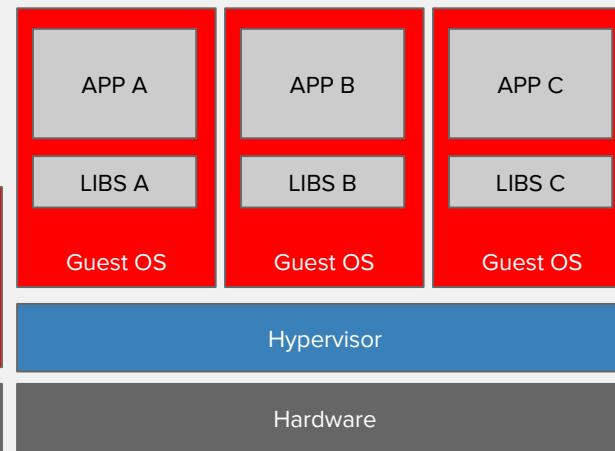
What is container?

Linux containers?

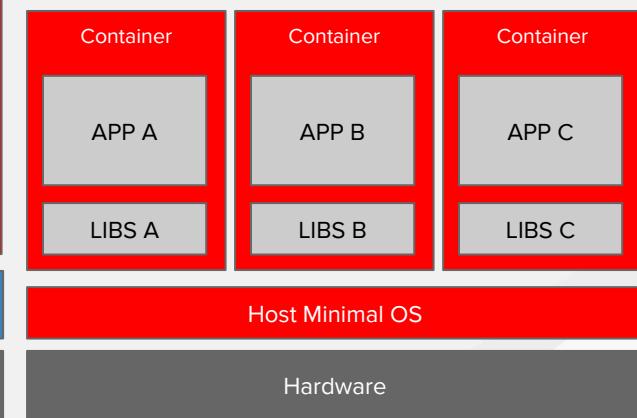
Traditional



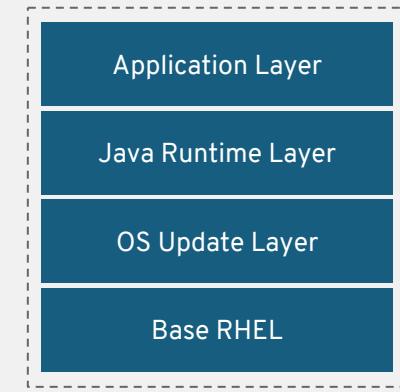
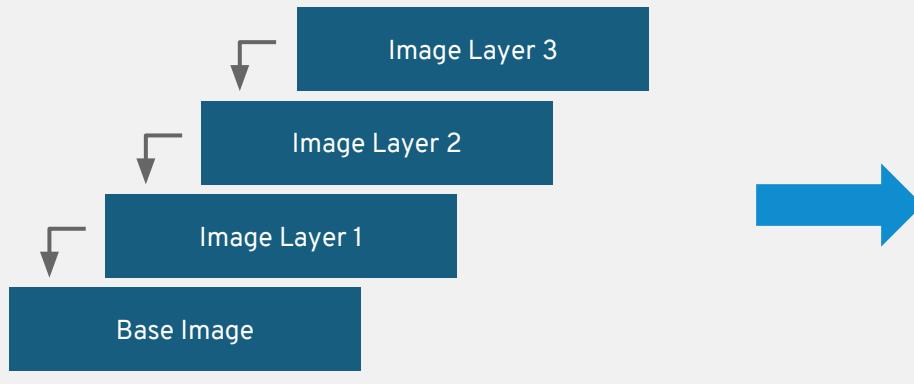
Virtual



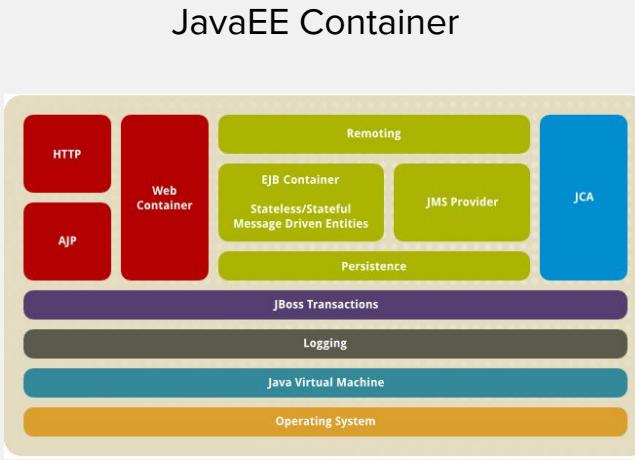
Container



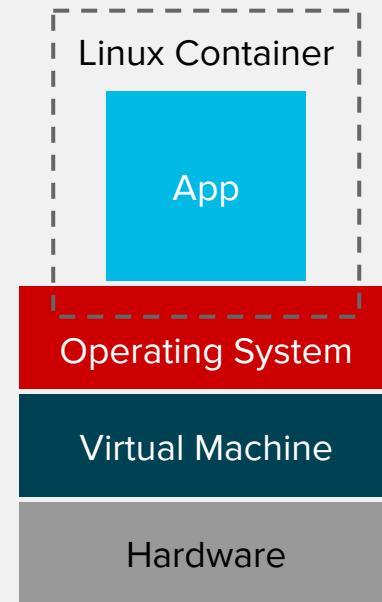
CONTAINER IMAGE LAYERING



Terminology !

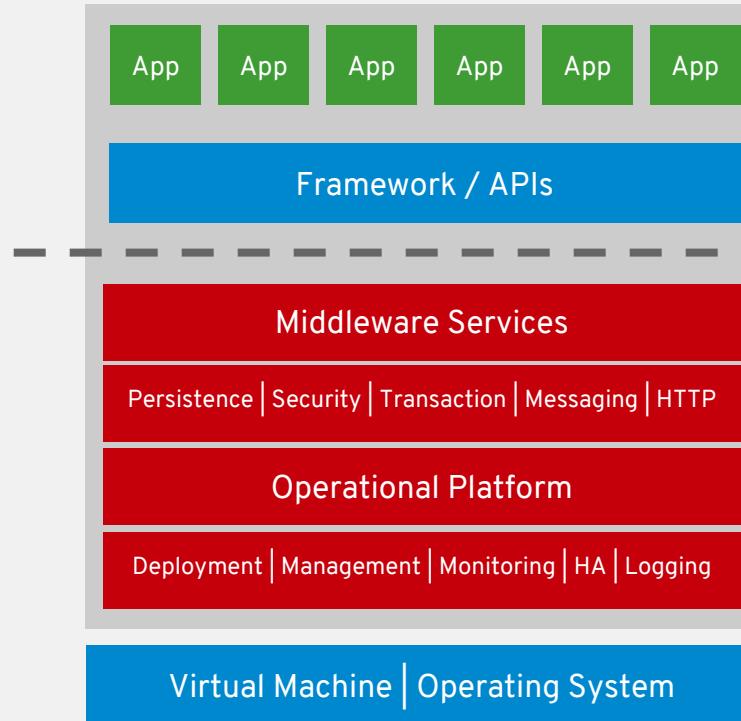


!=

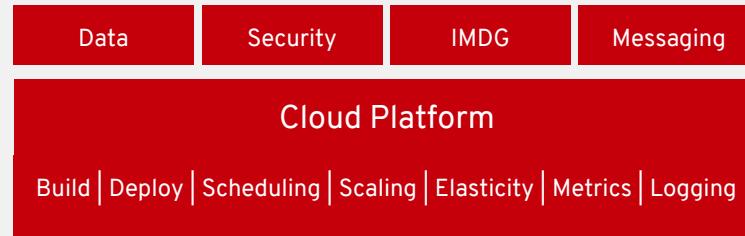


ARCHITECTURE?

THE APPSERVER 2000-2014

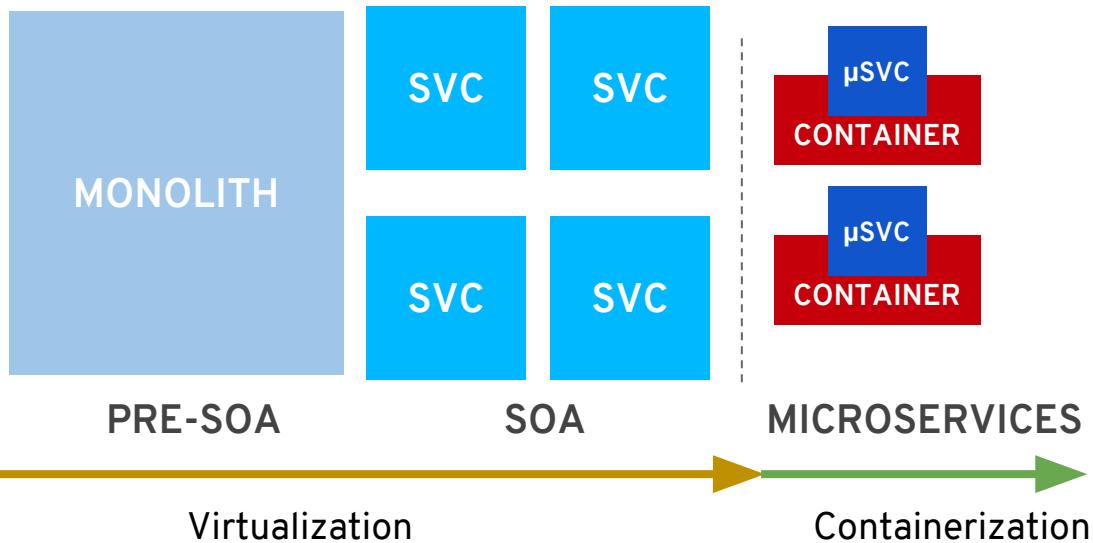


THE APPSERVER 2014 - ...



ARCHITECT FOR SMALLER APPLICATION COMPONENTS

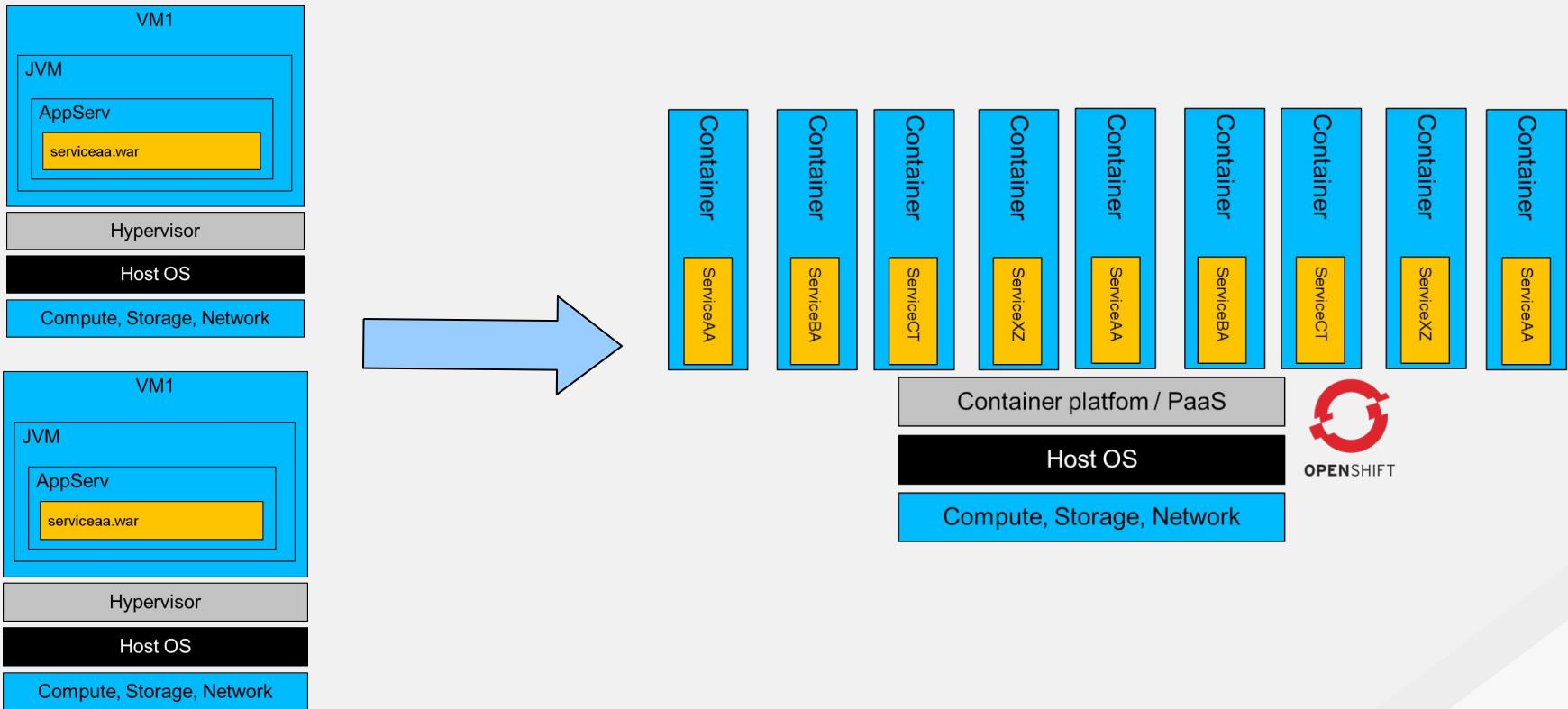
([microservices primer](#))



There is good and bad

- Good
 - Agile, DevOps
 - Polyglot
 - New Architectures
- Bad
 - Complexity
 - Dependencies
 - Consistency
 - Dealing with data

Linux containers ? Modern appservers? Microservices?



The background of the slide features a large stack of shipping containers in a port terminal. The containers are stacked high, filling most of the frame. They are primarily white with some blue and green ones interspersed. The perspective is from a low angle looking up at the top of the stack, with industrial structures like cranes and metal walkways visible above.

OpenShift

Container Platform



Self-Service



Multi-language



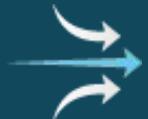
Automation



Collaboration



Seamless



RED HAT[®]
OPENSHIFT



Standards-based



Web-scale



Open Source



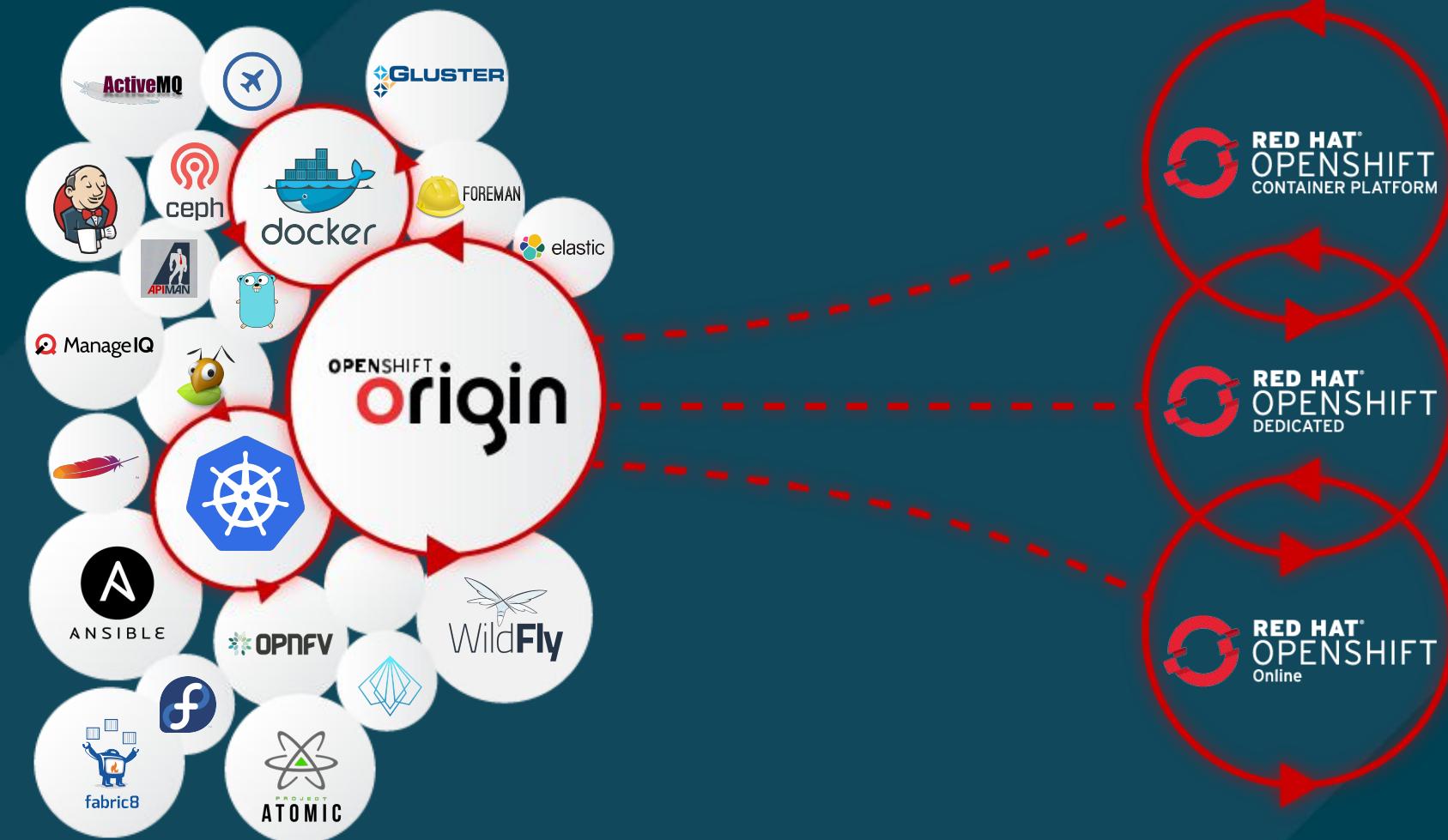
Enterprise Grade



Secure



Community Powered Innovation



TRUE POLYGLOT PLATFORM

LANGUAGES	Java	NodeJS	Python	PHP	Perl	Ruby	.NET Core	Third-party Language Runtimes	
DATABASES	MySQL	PostgreSQL	MongoDB	Redis	<p>...and virtually any docker image out there!</p>				CrunchyData GitLab Iron.io Couchbase Sonatype EnterpriseDB NuoDB Fujitsu and many more
WEB SERVERS	Apache HTTP Server	nginx	Varnish	Phusion Passenger	Tomcat	Third-party App Runtimes			
MIDDLEWARE	Spring Boot	Wildfly Swarm	Vert.x	JBoss Web Server	JBoss EAP	JBoss A-MQ	JBoss Fuse	Third-party Middleware	
	3SCALE API mgmt	JBoss BRMS	JBoss BPMS	JBoss Data Virt	JBoss Data Grid	RH Mobile	RH SSO	Third-party Middleware	



OpenShift

What is inside?



Trusted Container OS



Enterprise Container Host

Container Runtime & Packaging
(Docker)

Atomic Host

Red Hat Enterprise Linux

Trusted by Fortune Global
500 companies



Enterprise Kubernetes



Container Orchestration & Cluster Management
(kubernetes)

Networking Storage Registry Logs & Metrics Security

Infrastructure Automation & Mg



Enterprise Container Host

Container Runtime & Packaging
(Docker)

Atomic Host

Red Hat Enterprise Linux



kubernetes
Cloudforms
Red Hat Storage



Enterprise Container Platform



Self-Service

Service Catalog
(Language Runtimes, Middleware, Databases)

Build Automation Deployment Automation

OpenShift Application Lifecycle Management
(CI/CD)



Container Orchestration & Cluster Management
(kubernetes)

Networking Storage Registry Logs & Metrics Security

Infrastructure Automation & Cockpit



Enterprise Container Host

Container Runtime & Packaging
(Docker)

Atomic Host Red Hat Enterprise Linux

**Source-2-Image
Application Pipelines
Dev Tools**

Traditional, Stateful, and Microservices-based Apps

Business Automation

Integration

Data & Storage

Web & Mobile

Container

Container

Container

Container



Self-Service

Service Catalog
(Language Runtimes, Middleware, Databases)

Build Automation Deployment Automation

OpenShift Application Lifecycle Management
(CI/CD)



Container Orchestration & Cluster Management
(kubernetes)

Networking Storage Registry Logs & Metrics Security

Infrastructure Automation & Cockpit



Enterprise Container Host

Container Runtime & Packaging
(Docker)

Atomic Host

Red Hat Enterprise Linux

JBOSS EAP
JBOSS DATA GRID
JBOSS DATA VIRTUALIZATION
JBOSS AM-Q
JBOSS BRMS
JBOSS BPM
JBOSS FUSE
RED HAT MOBILE
3 Scale

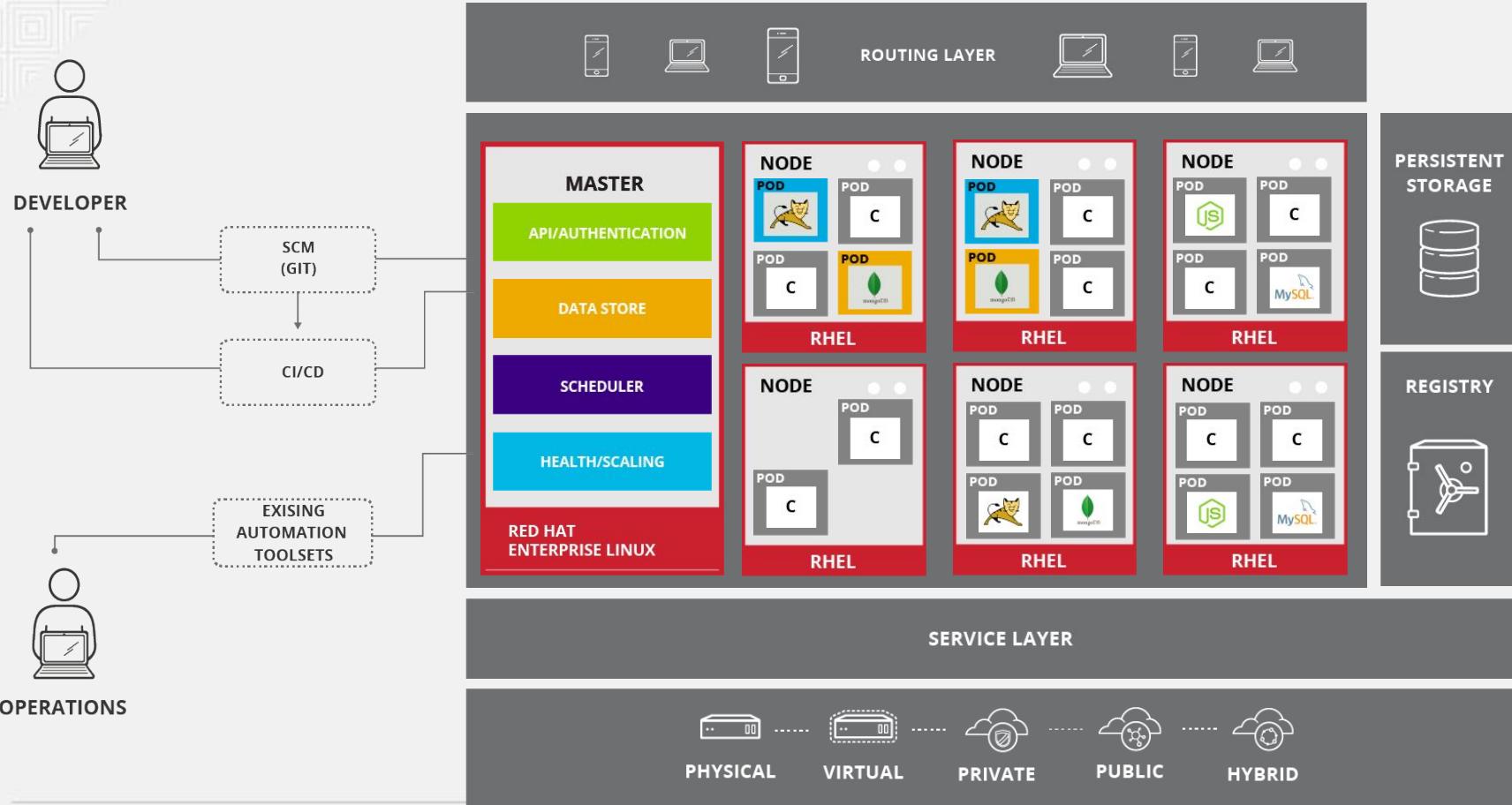
A photograph of a large stack of shipping containers in a port terminal. The containers are stacked high, filling the frame. The perspective is from a low angle looking up at the top of the stack. The sky is clear and blue.

OpenShift

How it works



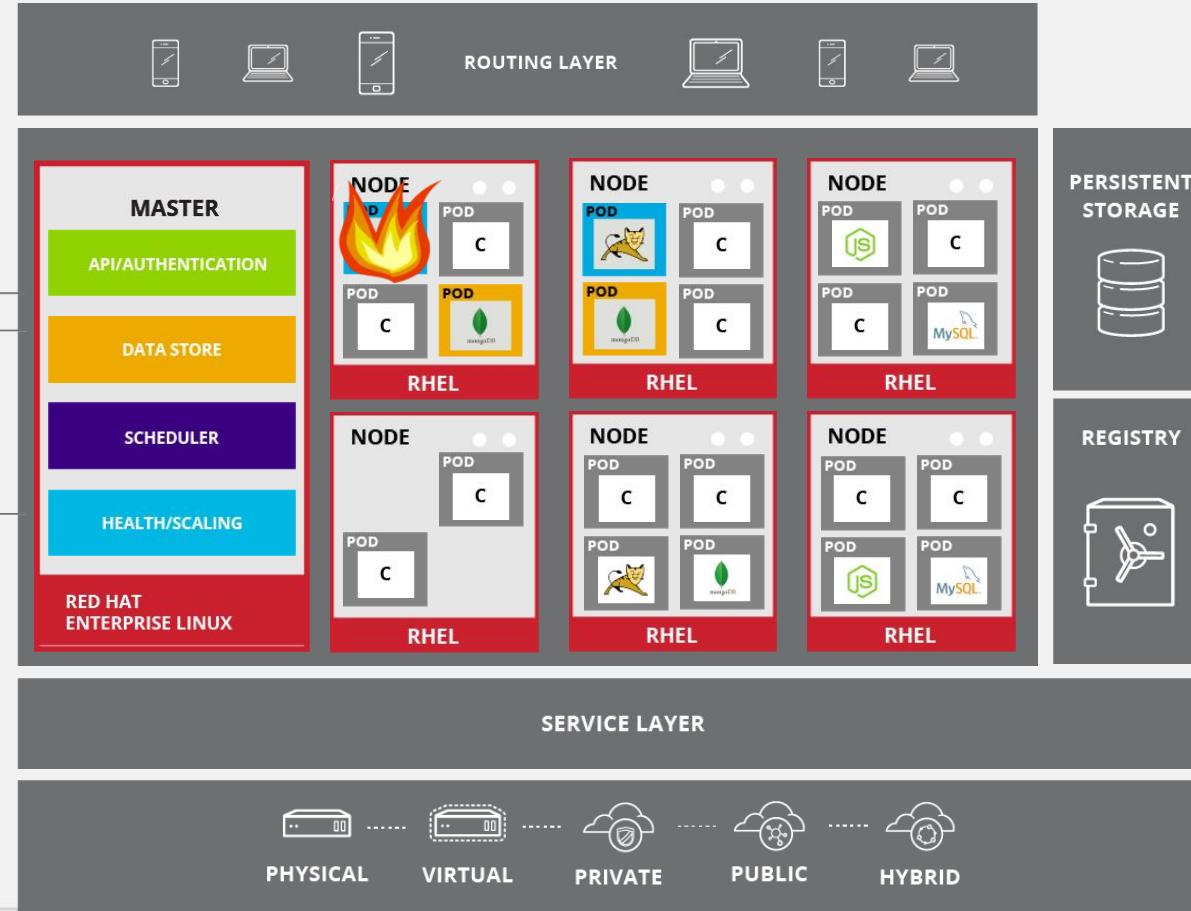
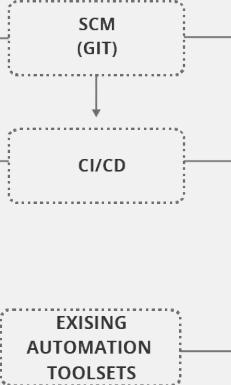
Openshift architecture



What if ..



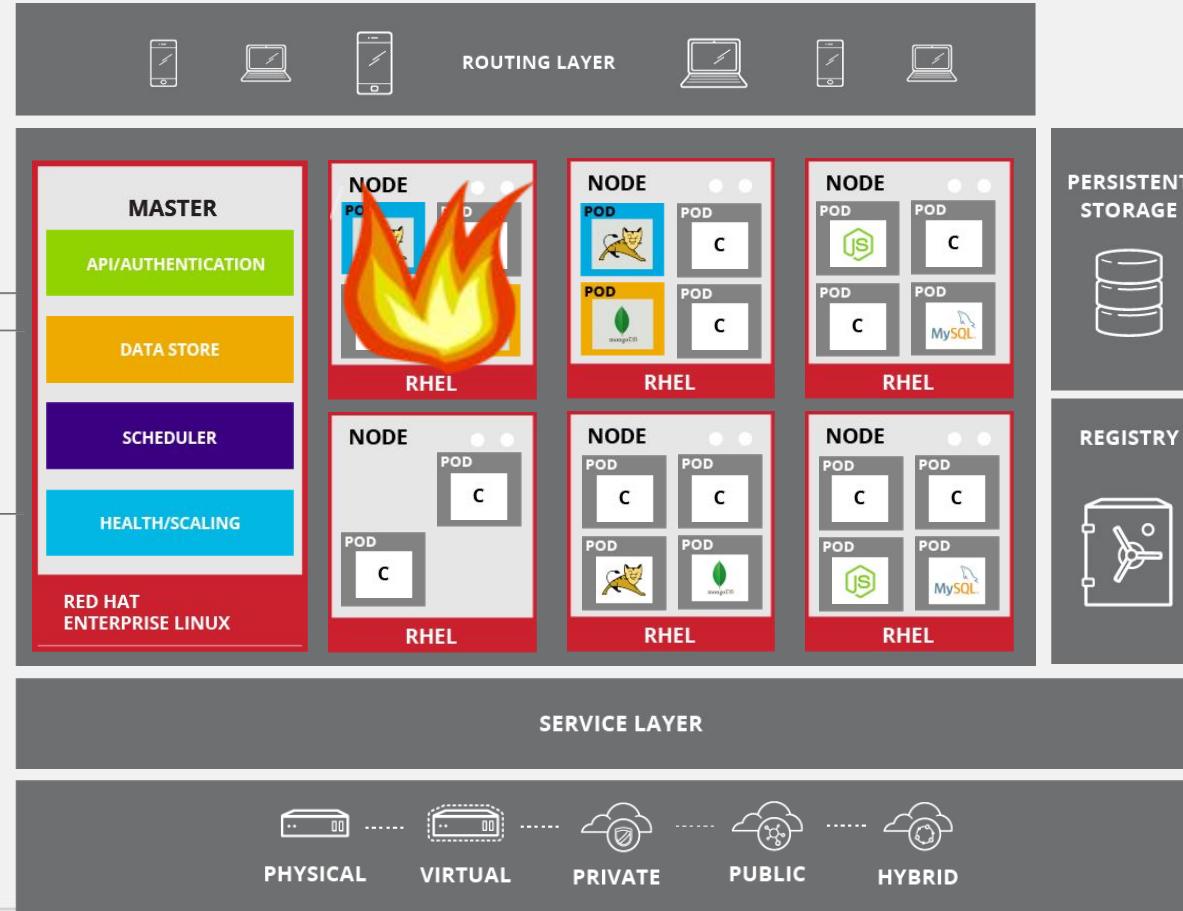
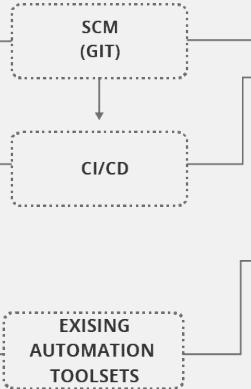
DEVELOPER



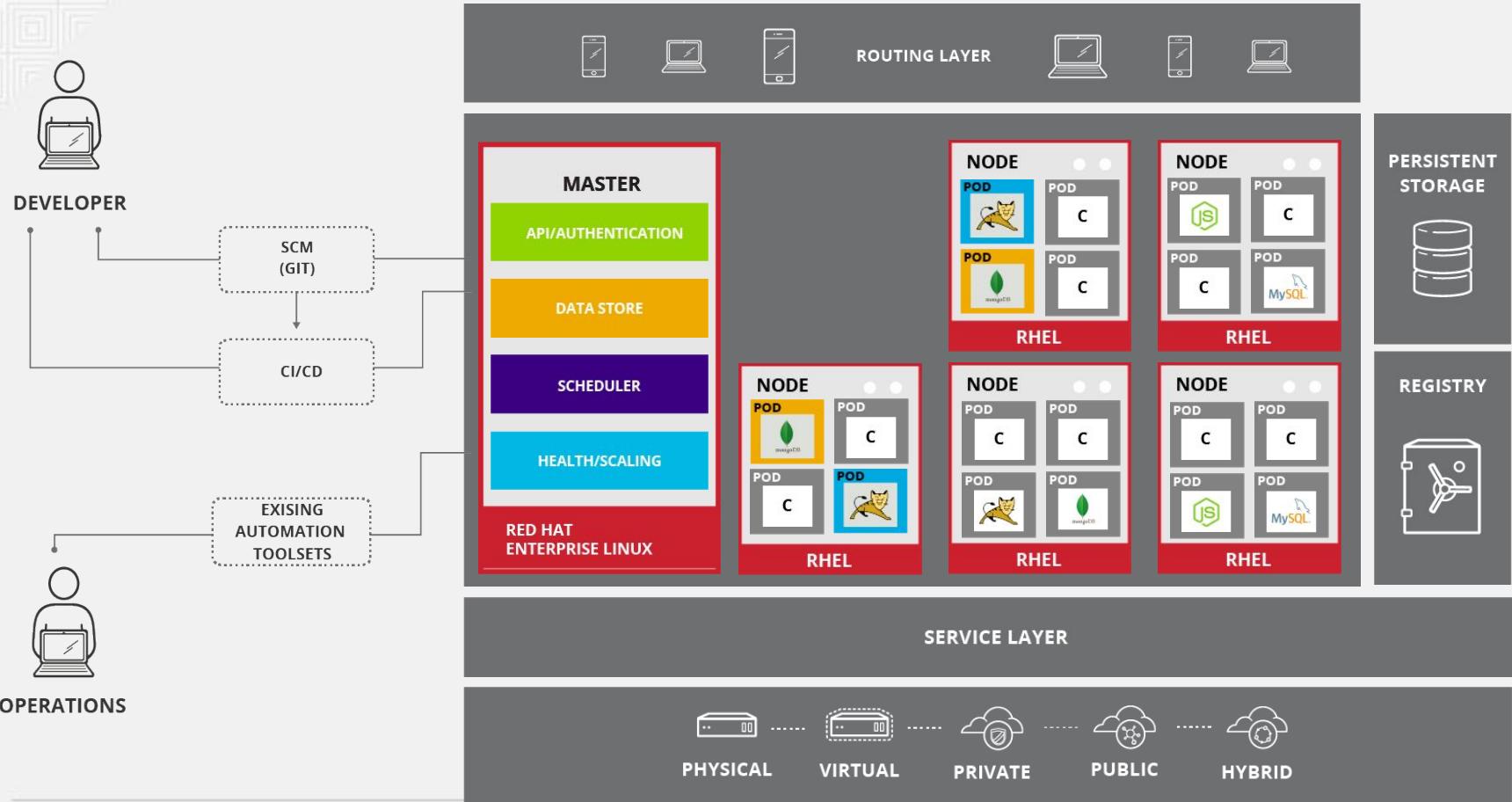
What if ...



DEVELOPER

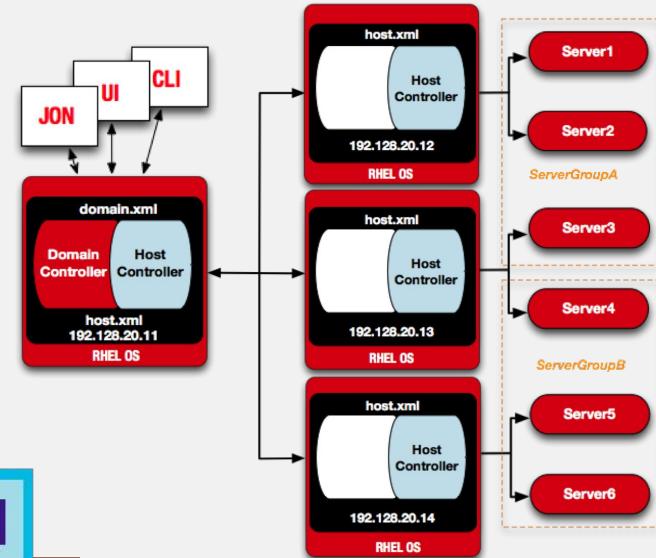
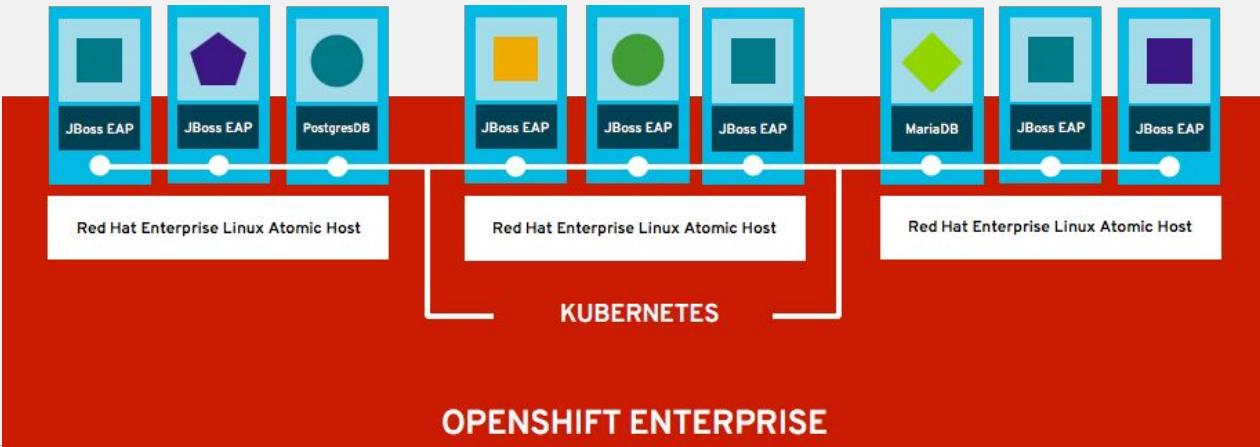


Self-healing AKA. Contant HA

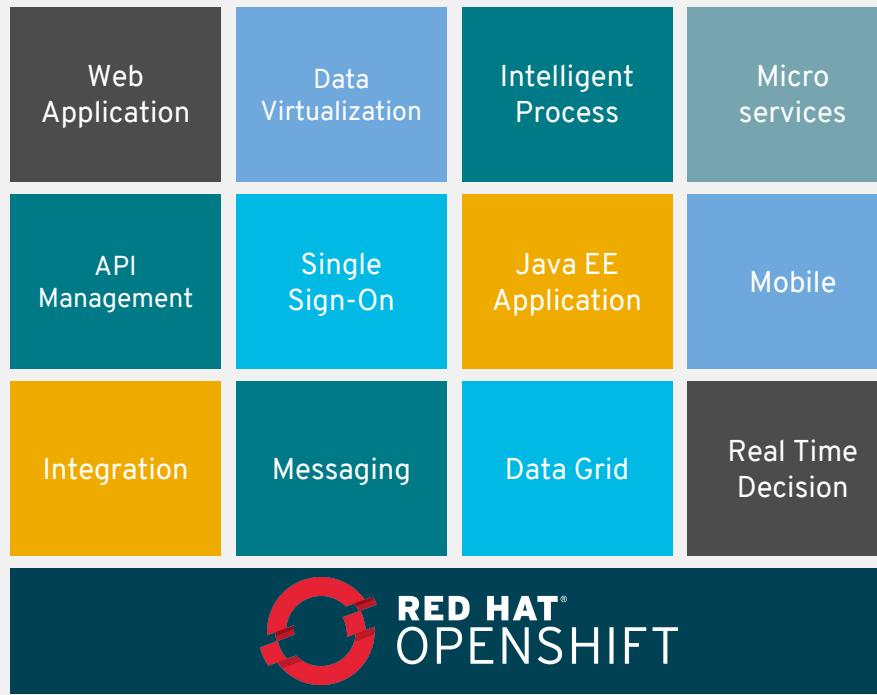


EAP in Cloud (PaaS)

- xPaaS subscription – OpenShift v3.*
 - Standalone only – cluster managed by Openshift
 - Configuration baked into container image
 - Kill and build a new one
- Microservices out of the box

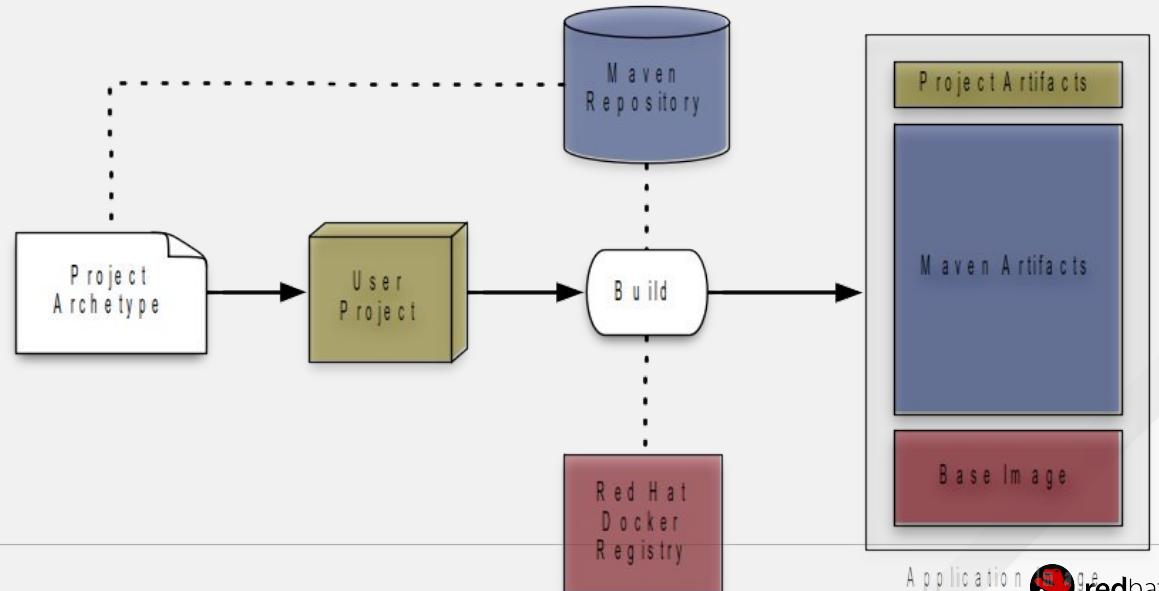


A PLATFORM THAT GROWS WITH YOUR BUSINESS



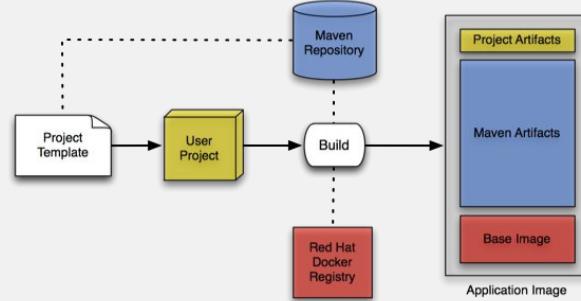
Building Java app

- Cattle approach is used for deployment of nodes
- 2 different strategies
 - Build It, Push It (outside OpenShift).
 - Uses Maven plugins to drive build and deploy workflow
 - Push It, Build It (inside OpenShift).
 - Uses traditional OpenShift Application Templates (S2I)

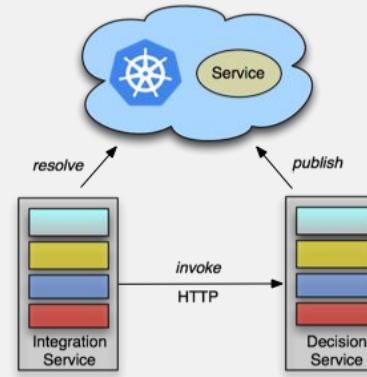


Fuse Integration Services for OpenShift

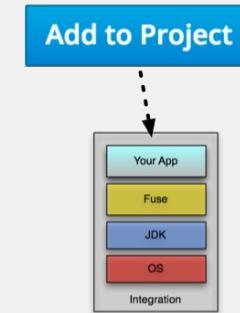
Tooling creates a complete runtime customized to an application's requirements and delivered as a Docker image

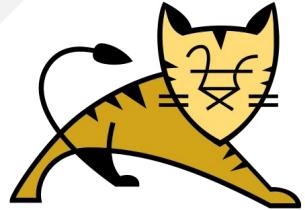


Integrate, cluster, and scale applications using Kubernetes.



Bridge development and operations with build, deployment, and management integration within OpenShift.





VERT.X



WildFly SWARM

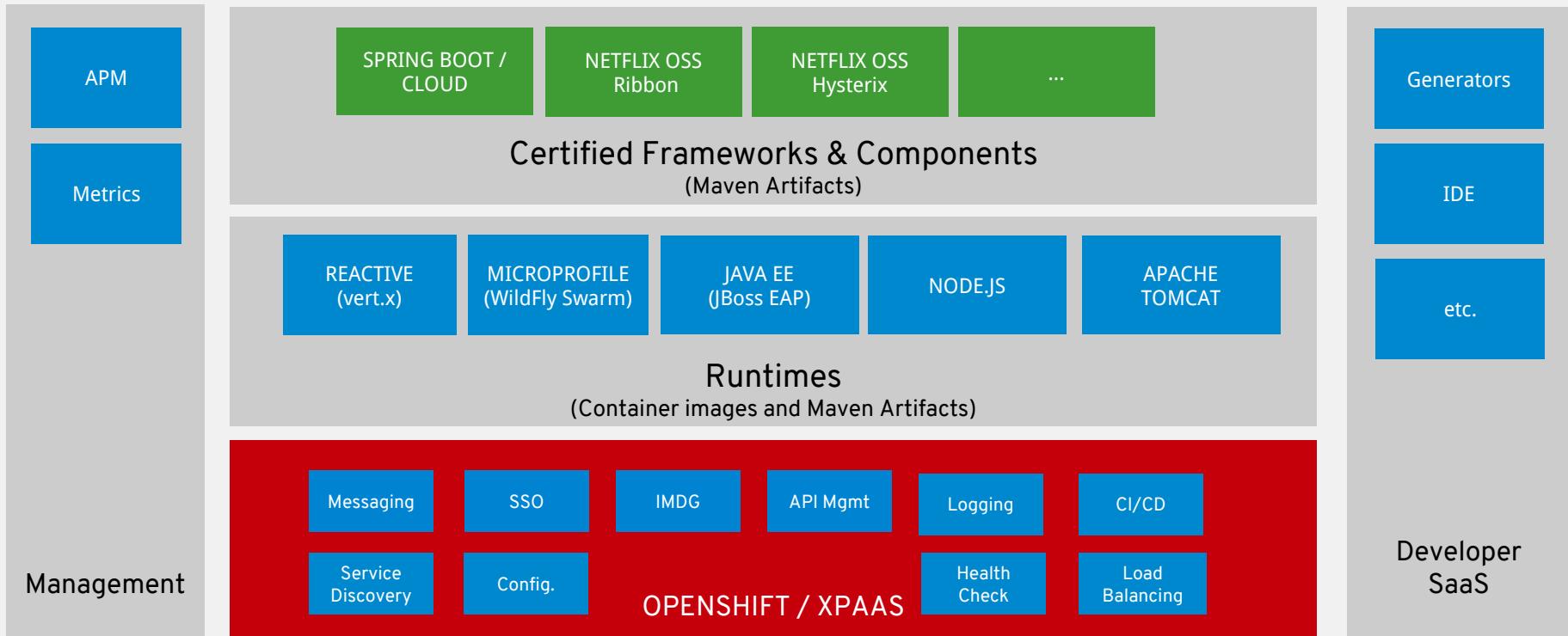
NETFLIX | OSS



HYSTRIX
DEFEND YOUR APP



OPENShift APPLICATION SERVICES





OpenShift

Build automation



Source 2 Image Walk Through

Code



git



DEV

Build

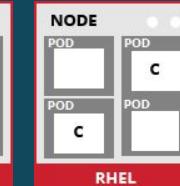
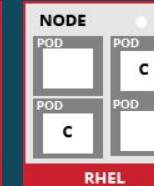
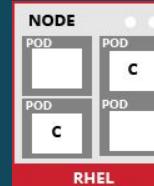


Container
Image



Registry

Deploy



OPS

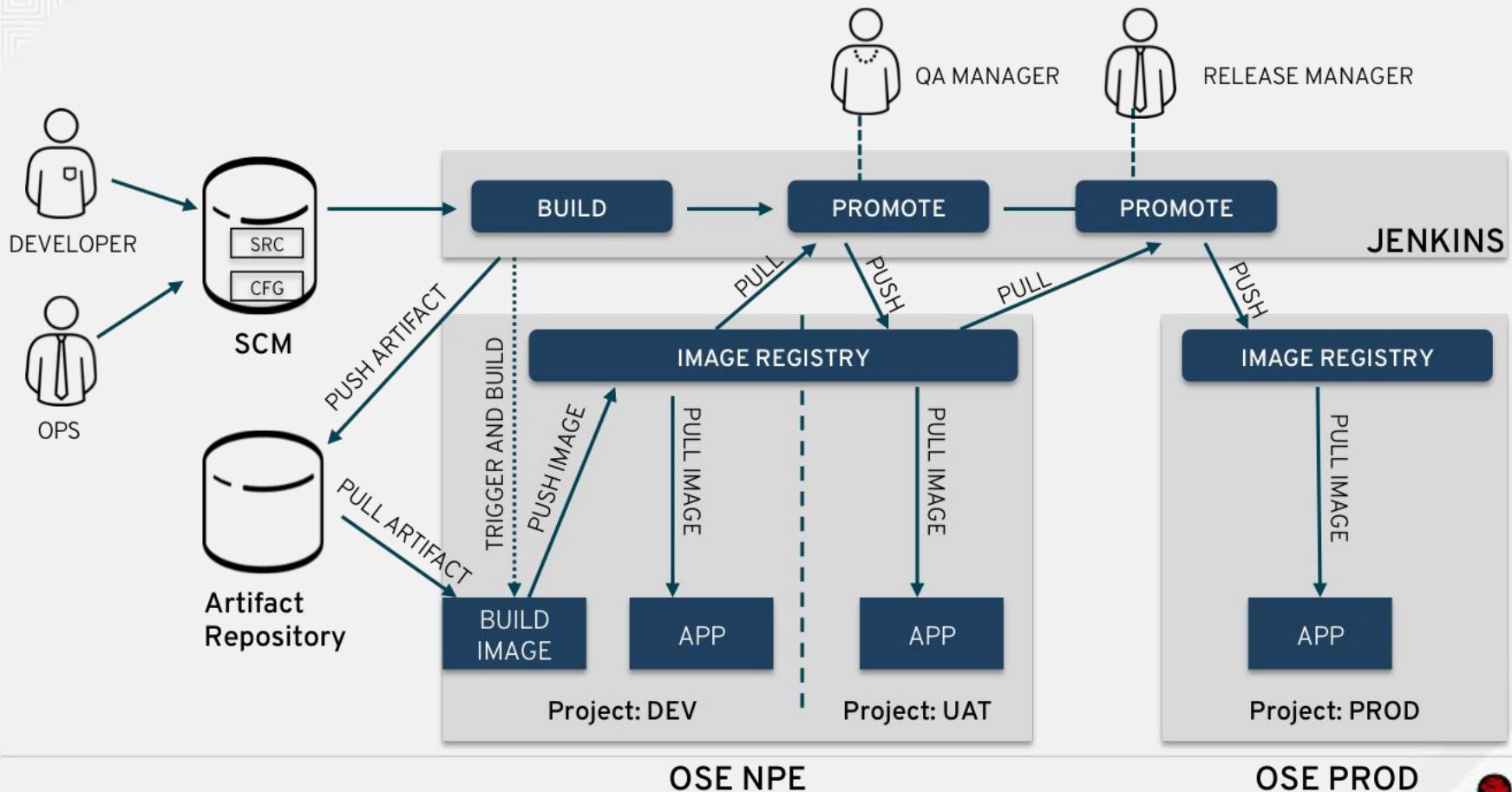
A photograph of a large stack of shipping containers in a port terminal. The containers are stacked high, filling the frame. The colors are somewhat muted, with a yellowish tint. In the center, there is a white overlay containing text.

OpenShift

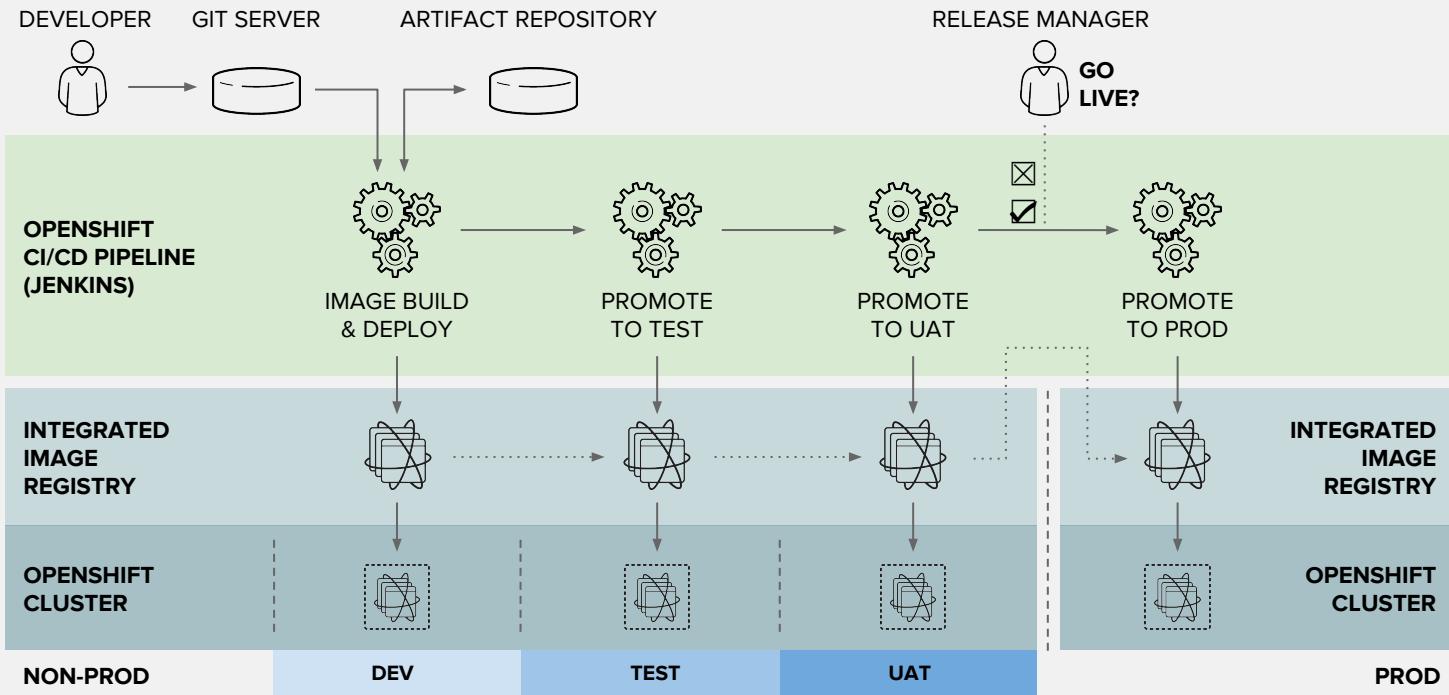
Full CI/CD platform



OpenShift CI/CD flow

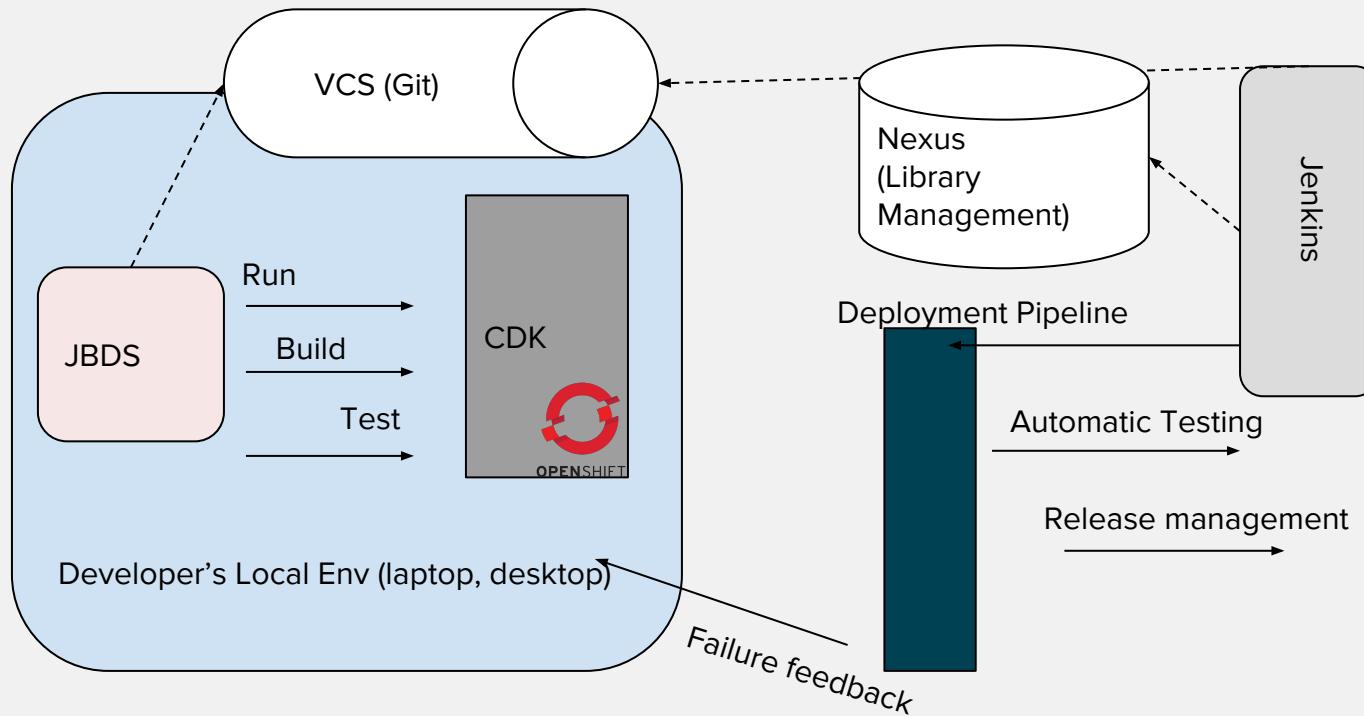


CONTINUOUS DELIVERY PIPELINE

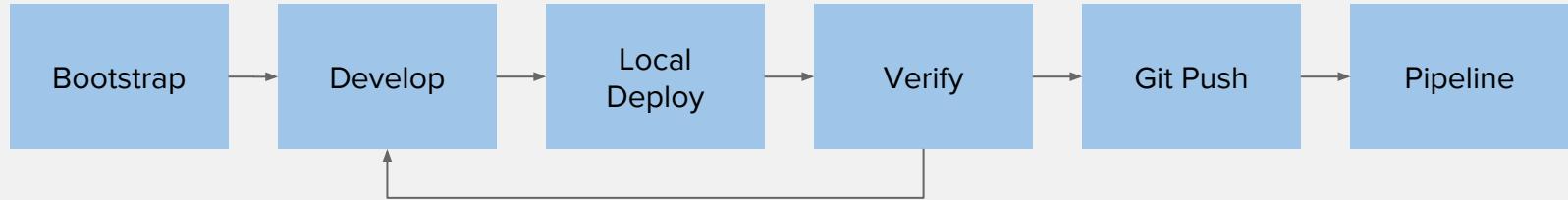


**PIPELINE AFTER EVERY COMMIT??
DEBUGGING??**

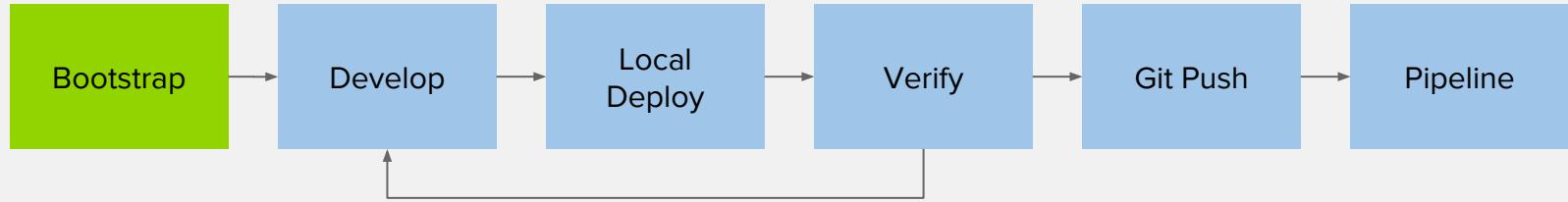
LOCAL DEVELOPMENT WITH CONTAINERS



LOCAL DEVELOPMENT WORKFLOW



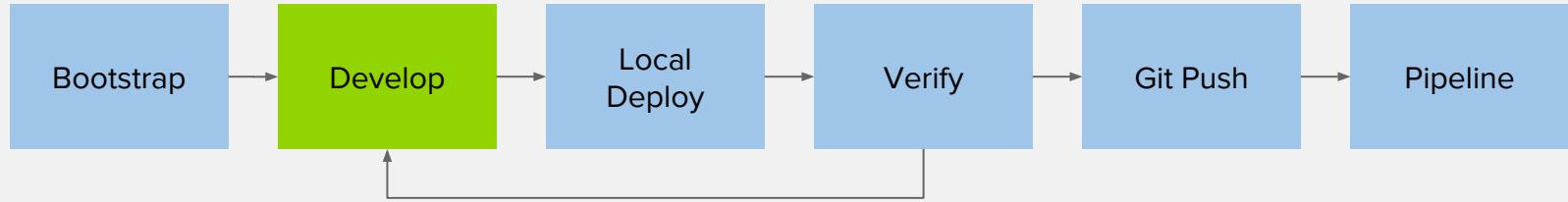
LOCAL DEVELOPMENT WORKFLOW



BOOTSTRAP

- Pick your programming language and application runtime of choice
- Create the project skeleton from scratch or use a generator such as
 - Maven archetypes
 - Quickstarts and Templates
 - OpenShift Generator
 - Spring Initializr

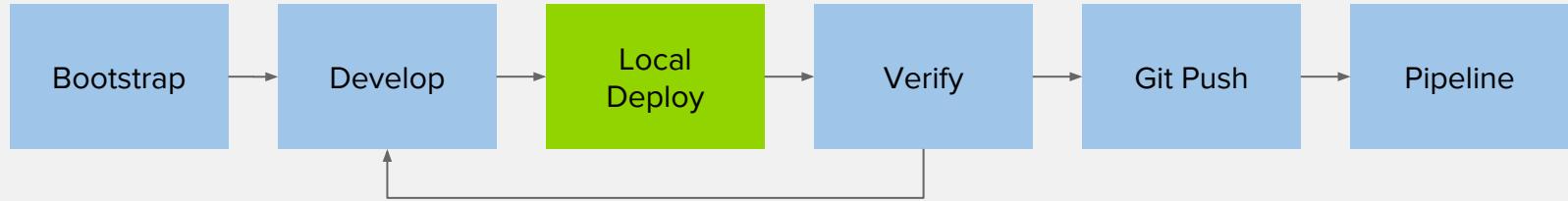
LOCAL DEVELOPMENT WORKFLOW



DEVELOP

- Pick your framework of choice such as Java EE, Spring, Ruby on Rails, Django, Express, ...
- Develop your application code using your editor or IDE of choice
- Build and test your application code locally using your build tools
- Create or generate OpenShift templates or Kubernetes objects

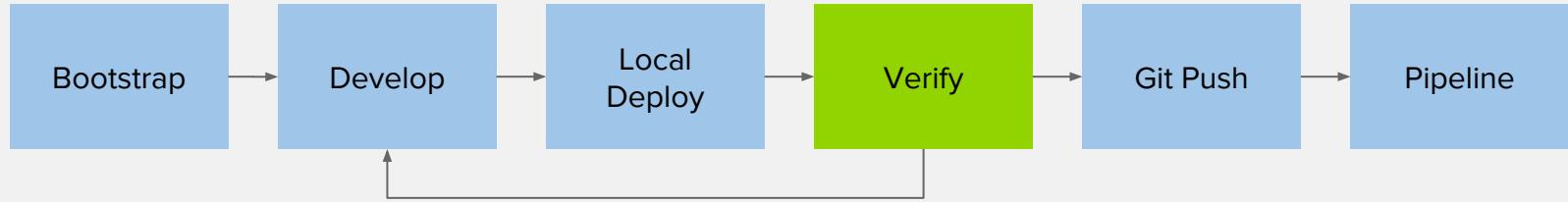
LOCAL DEVELOPMENT WORKFLOW



LOCAL DEPLOY

- Deploy your code on a local OpenShift cluster
 - Red Hat Container Development Kit (CDK), minishift and oc cluster
- Red Hat CDK provides a standard RHEL-based development environment
- Use binary deploy, maven or CLI rsync to push code or app binary directly into containers

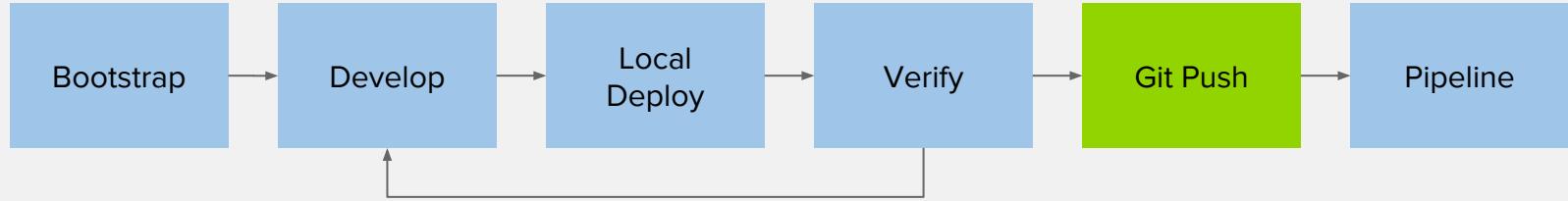
LOCAL DEVELOPMENT WORKFLOW



VERIFY

- Verify your code is working as expected
- Run any type of tests that are required with or without other components (database, etc)
- Based on the test results, change code, deploy, verify and repeat

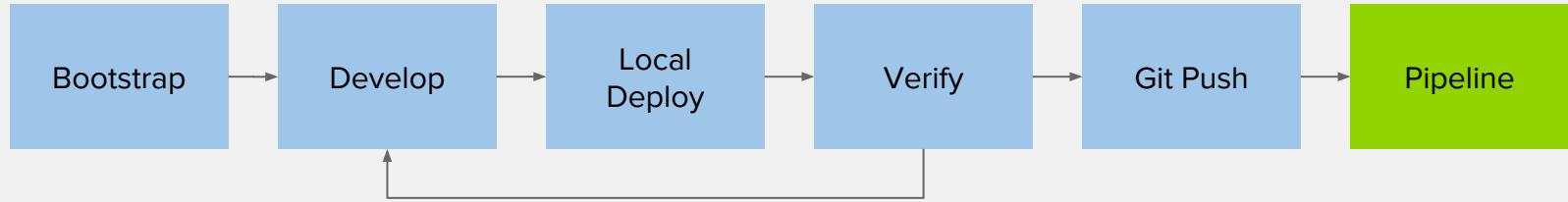
LOCAL DEVELOPMENT WORKFLOW



GIT PUSH

- Push the code and configuration to the Git repository
- If using Fork & Pull Request workflow, create a Pull Request
- If using code review workflow, participate in code review discussions

LOCAL DEVELOPMENT WORKFLOW



PIPELINE

- Pushing code to the Git repository triggers one or multiple deployment pipelines
- Design your pipelines based on your development workflow e.g. test the pull request
- Failure in the pipeline? Go back to the code and start again

DEPLOYMENT OPTIONS?

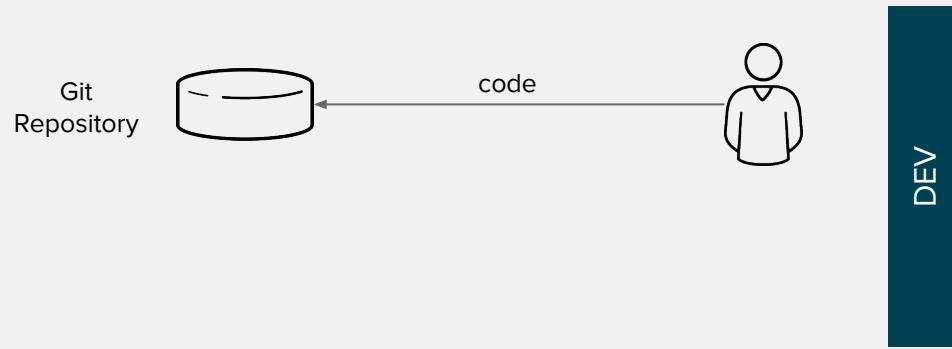


DEPLOY YOUR SOURCE CODE

DEPLOY SOURCE CODE WITH SOURCE-TO-IMAGE (S2I)

CODE

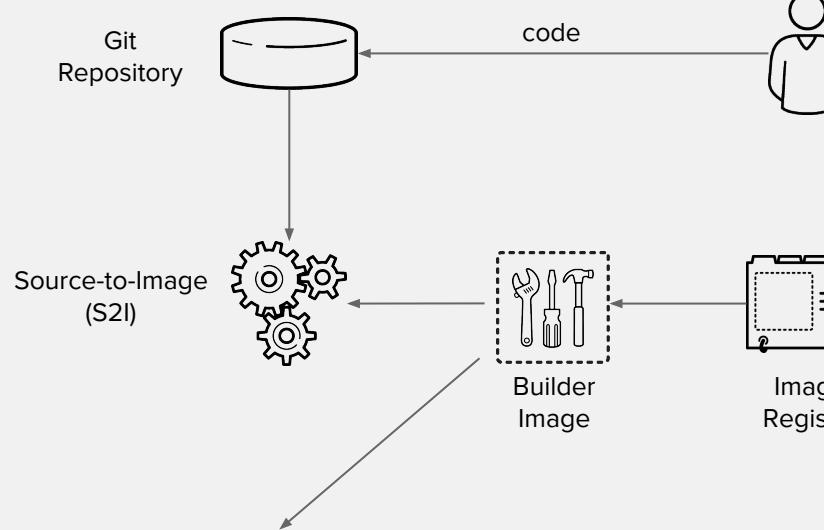
Developers write code using existing development tools such as Maven, NPM, Bower, PIP, Dockerfile and Git and then access the OpenShift Web, CLI or IDE to create an app from the code



DEPLOY SOURCE CODE WITH SOURCE-TO-IMAGE (S2I)

BUILD

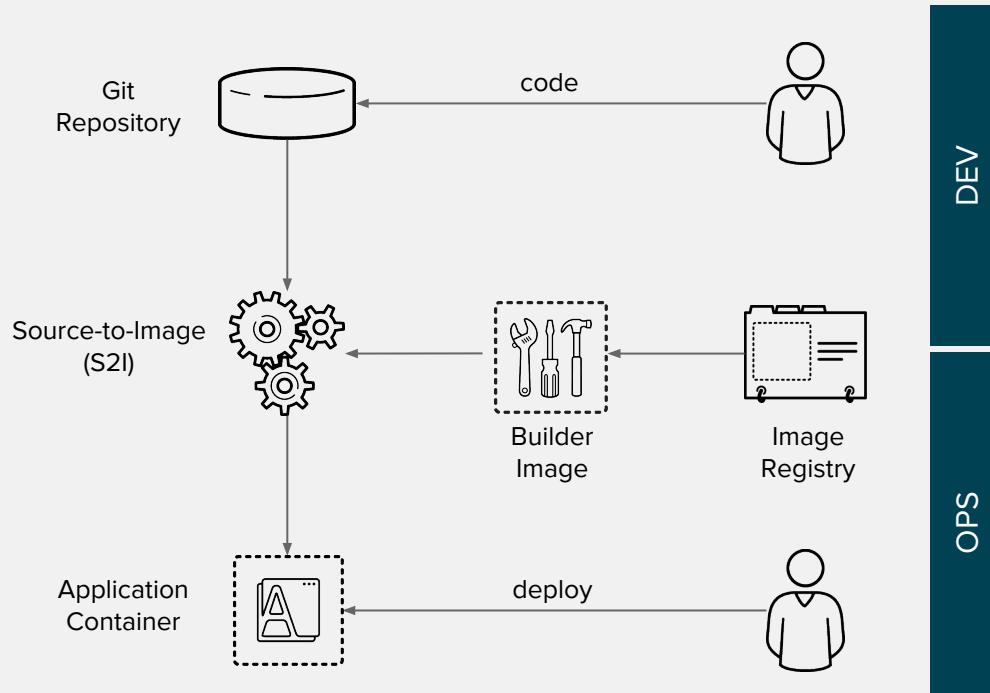
S2I combines source code with a builder image (language and application runtimes) and stores the resulting application image in the image registry



DEPLOY SOURCE CODE WITH SOURCE-TO-IMAGE (S2I)

DEPLOY

OpenShift automates the deployment of application containers across multiple hosts via the Kubernetes. Users can trigger deployments, rollback, configure A/B or other custom deployments



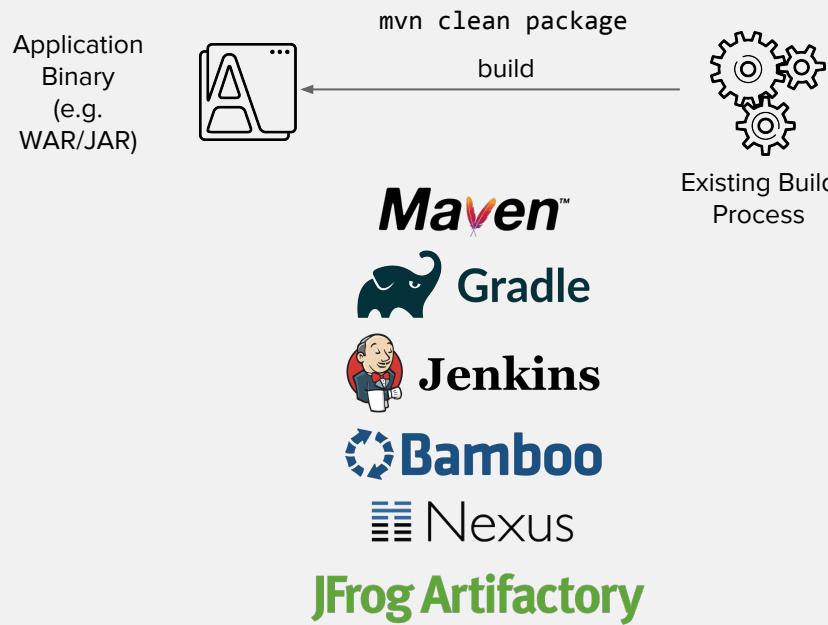


DEPLOY YOUR APP BINARY

DEPLOY APP BINARY WITH SOURCE-TO-IMAGE (S2I)

BUILD APP

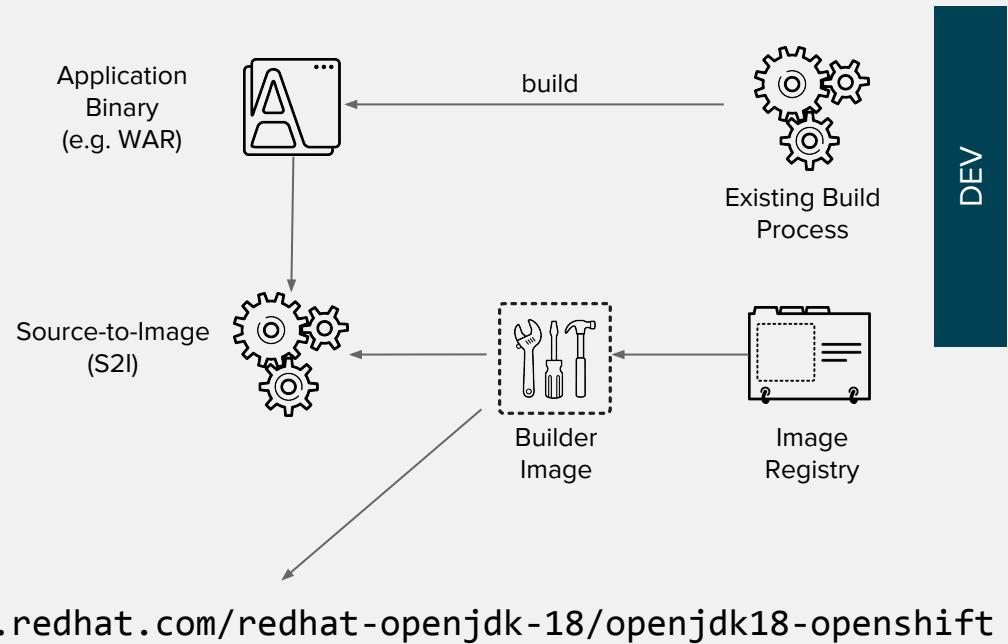
Developers use the existing build process and tools (e.g. Maven, Gradle, Jenkins, Nexus) to build the app binaries (e.g. JAR, WAR, EAR) and use OpenShift CLI to create an app from the app binaries



DEPLOY APP BINARY WITH SOURCE-TO-IMAGE (S2I)

BUILD IMAGE

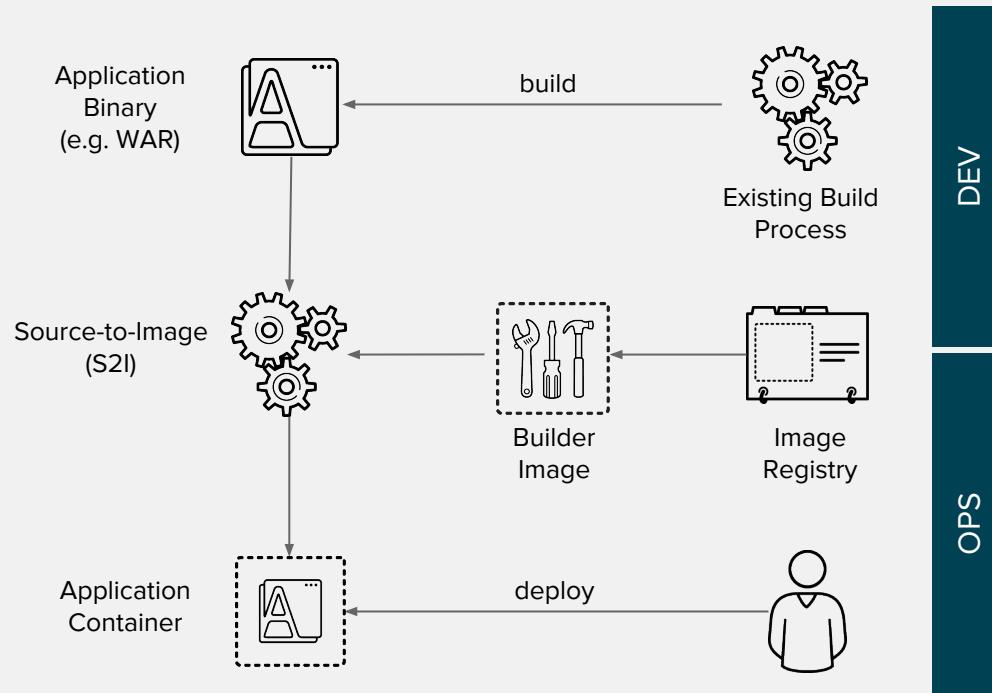
S2I combines app binaries (e.g. JAR, WAR, EAR) with a builder image (language and application runtimes) and stores the resulting application image in the image registry



DEPLOY APP BINARY WITH SOURCE-TO-IMAGE (S2I)

DEPLOY

OpenShift automates the deployment of application containers across multiple hosts via the Kubernetes. Users can trigger deployments, rollback, configure A/B or other custom deployments





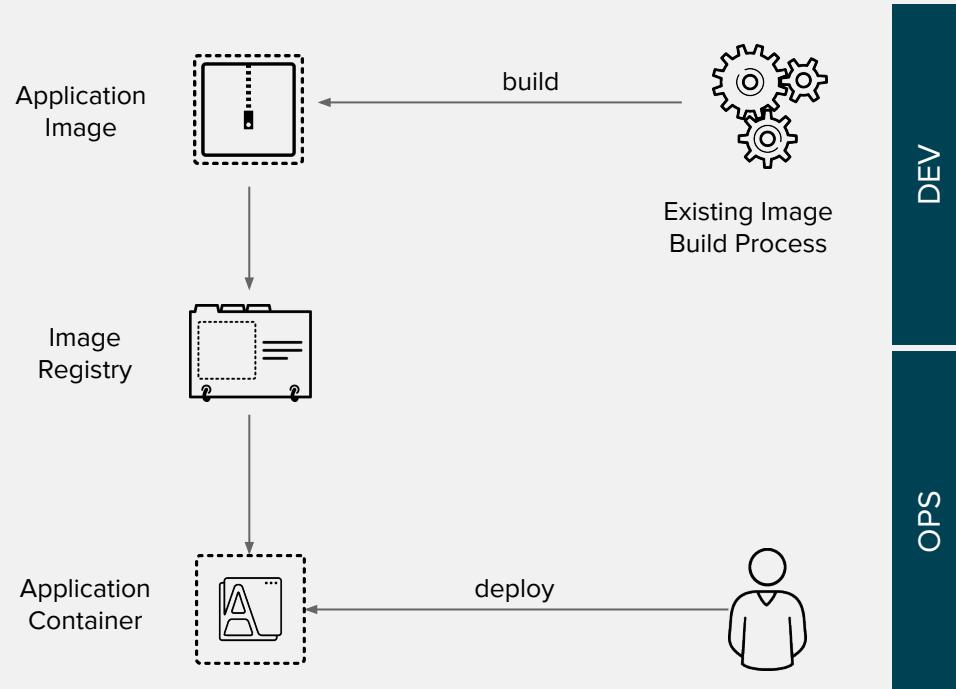
DEPLOY YOUR CONTAINER IMAGE

DEPLOY DOCKER IMAGE

BUILD

App images are built using an existing image build process. OpenShift automates the deployment of app containers across multiple hosts via the Kubernetes. Users can trigger deployments, rollback, configure A/B, etc

DEPLOY



Self-Service



Multi-language



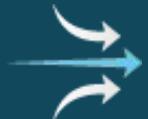
Automation



Collaboration



Seamless



RED HAT[®]
OPENSHIFT



Standards-based



Web-scale



Open Source



Enterprise Grade

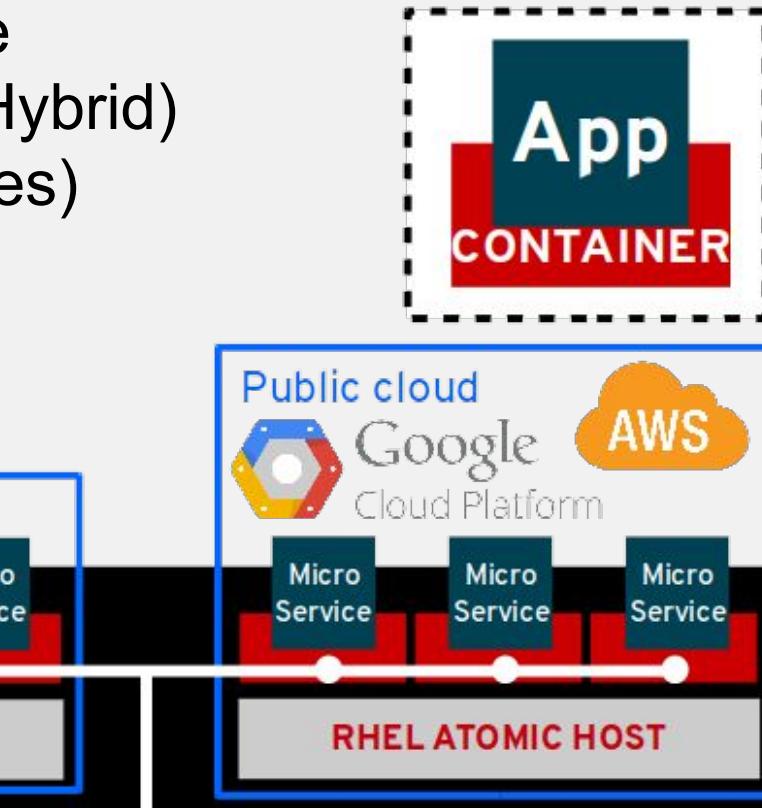
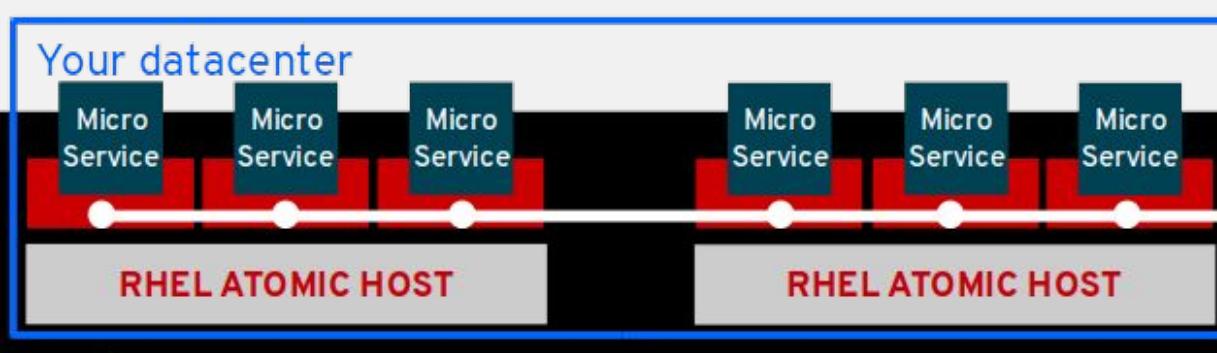


Secure



Openshift summary

- Automates whole container lifecycle
- On Premise / In cloud / Both (aka. Hybrid)
- Containers orchestration (Kubernetes)
- Microservices O-o-the-box
- CI/CD automation, Dev Ops
- Scalability & HA O-o-the-box



KUBERNETES
Container Orchestration



redhat.[®]



OpenShift Trilogy Schedule

PHA: 10/10/2017 **Container Wars: A New Hope**

OpenShift - DevOps - Automation Overview

BRNO: 1/11/2017 **Container Wars: API strikes back**

Agile integration & API Workshop

PHA?: 14/11/2017 **Container Wars: Return of the App**

Jedi's App development & DevOps Workshop

Container LAB @ Opensource Summit Prague !!!

23.10. - 27.10 2017

**Capacity limited
Register today !**

Reoccurring hands-on Workshops :

Every day, 10:00 , 12:00, 14:00

- Spin-up your first container in seconds
- Develop simple Multi-language apps in Openshift
- Build your first fully automated CI/CD flow in Openshift
- Integrate services in Openshift
- Attach persistent storage to your containerized apps

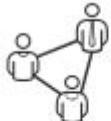




redhat.[®]



Classroom



Virtual
training



Online
training



On-site
training



redhat.

**ALLIANCE
TRAINING
PARTNER**

- DO080** Deploying Containerized Applications Technical Overview
- DO180** Introduction to Containers, Kubernetes, and Red Hat OpenShift
- DO280** Red Hat OpenShift Administration I
- DO290** Developing and Deploying Applications on OpenShift

Knowledge is the power. Training is the key!



redhat.[®]

The background image shows the interior of a large industrial ship's cargo hold. Numerous shipping containers are stacked in several layers. The containers are primarily white and green. The ship's metal structure, including beams, ladders, and walkways, is visible throughout the frame.

OpenShift

Customer References



Evolving Application Architecture at Volvo

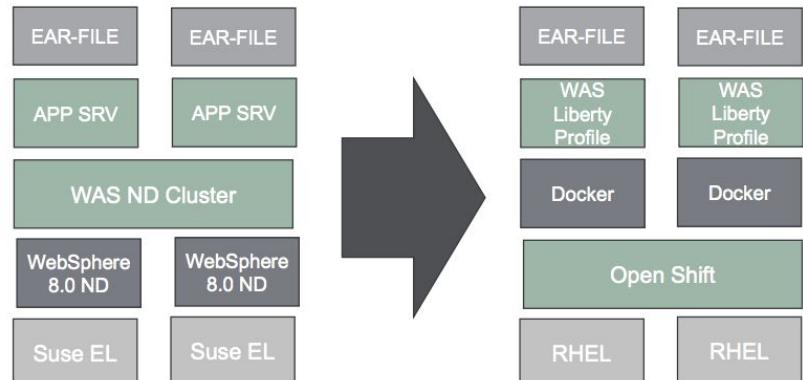
785 apps across 560 app servers

OpenShift provides build, distribution & runtime environment

Platform for DevOps and Microservices

Running OpenShift on Azure,
automatically provisioned with
Ansible

OUR NEW ENVIRONMENT



DEUTSCHE BANK - Technology Transformation



“Delivering Everything as a Service. From 20% adoption to 40% in 1yr; planning to move 85% of all applications to OpenShift platform. We won the hearts and minds of developers.” -- Pat Healy, CTO, Deutsche Bank

- Macro trends are radically reshaping the banking industry
- Need to regain software expertise that was previously outsourced
- OpenShift replaced an internal, homegrown PaaS platform
- Over 300 internal projects moved to OpenShift
- 6x better efficiency of computing resources using containers and OpenShift. Driving overall utilization up via multi-tenancy.
- Leverage OpenShift across multiple public clouds.

Ideas to Production, safely in a day.

View the [Deutsche Bank keynote](#)

DEUTSCHE BANK - Technology Transformation

“Delivering Everything as a Service. From 20% adoption to 40% in 1yr; planning to move 85% of all applications to OpenShift platform. We won the hearts and minds of developers.”

EVERYTHING AS A SERVICE

Provider abstraction, best execution venue

Drive utilisation up through multi-tenancy – we have history

Standardised application building blocks, API focus

Frictionless, safe & rapid path to production

The “AO” – ‘Ideas to production, safely in a day’





Reduce complexity of our IT Infrastructure

Key performance indicators	2015	2020 Plan	Change
Operating systems	45	4	~90%
End-of-life hardware / software	166	0	100%
% virtualisation	46%	95%	49ppts
Private cloud adoption	20%	80%	60ppts
Intersystem reconciliations	~1,000	~300	70%

“Run the bank” costs targeted to decline by EUR ~800 m

BARCLAYS BANK - Digital Transformation



"OpenShift is the primary platform to deploy Barclays apps across any clouds. We couldn't be happier with our OpenShift progress to date."
-- Kieran Broadfoot, CTO of Hosting, Barclays

- 300 year old bank
- Focused on delivering new services to banking customers
- Moving from proprietary middleware to open source
- Moved 3000 applications to their cloud in last 2yrs; 500 in production.
- Implemented CI/CD and Infrastructure-as-Code pipelines to allow Devs and Ops to work better together.

View the [Barclays keynote](#).

BARCLAYS BANK - Digital Transformation

“OpenShift is the primary platform to deploy Barclays apps across any clouds. We couldn’t be happier with our OpenShift progress to date.”



Enable DevOps at scale
and deliver on our agile agenda

BMW GROUP - Evolving the Connected Car



Digital customer experience, connected and automated driving and digitalized business processes lead to a transformation of the BMW Group towards software and services (Tech).



- Global manufacturer of luxury automobiles, motorcycles and engines. “The Ultimate Driving Machine”.
- Evolving in-vehicle communications and telematics for 15yrs.
- OpenShift platform enables BMW ConnectedDrive service.
- Enables Electric Cars, Service Calls, Real-Time Traffic, Driving Assistance, Anonymous Cars.
- Moving from Monolithic applications to Microservices, Containers and Kubernetes



Read the [press release](#) and view Red Hat Summit [presentation](#).

SCHIPHOL AIRPORT - Rethinking Traveler Experience



Red Hat OpenShift Container Platform, truly, has stolen my heart, because the platform is innovative, I can deploy quickly, and I am in control of the containers. - Michael Aalbers, Amsterdam Airport Schiphol

- International Airport, 3rd busiest in Europe, 64M passengers per year.
- Goal to become the world's best digital airport by 2018.
- Needed to accelerate application development time through cloud-agnostic platform.
- Deployed OpenShift across Private Cloud, AWS and Azure.
- Leverage JBoss FUSE and 3Scale for API Management, delivering API-based services to passengers and partners.

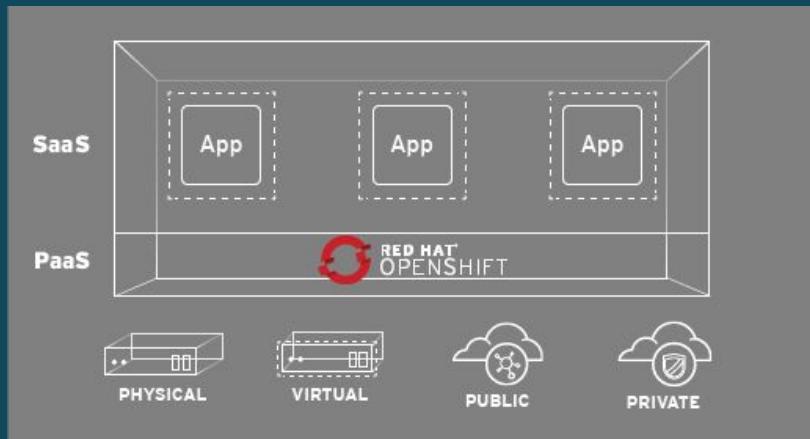
Read the [press release](#)

AMADEUS



Platform Infrastructure

- Shift from virtualization to scale-out cloud infrastructure
- Rapid growth in public cloud usage for enterprises
- Hybrid cloud deployments span private & multiple public clouds



IT'S A DIFFERENT JOURNEY FOR EVERYONE

BUT HERE'S WHAT IT LOOKS LIKE FOR ONE FINANCIAL SERVICES CUSTOMER

