INTRODUCTION TO OPENSHIFT

Martin Sauvé
Architecte de solutions
8 décembre 2014
BUSINESS DEMANDS DRIVE I.T. TRANSFORMATION

- Business wants agility, lower cost, new capabilities
- IT struggling with existing legacy infrastructure architecture and cost model
- Cloud providers are using next-generation IT built on open source technologies
- IT needs to adopt cloud architectures and technologies to close innovation gap
I.T. TRANSFORMATION FOR EVERYONE

**BUSINESS CHALLENGES**
- Provide ubiquitous access to data and services
- Achieve better quality of service
- Rapid innovation and faster time to market

**DEVELOPER CHALLENGES**
- Reduce time to provision and develop, improve productivity
- Test new features and update applications faster
- Improve availability of platforms and resources

**I.T. OPERATIONS CHALLENGES**
- Increase operation efficiency
- Maximize resource utilization
- Reliable, secure, compliant
THREE PILLARS OF AN I.T. ORGANIZATION

- People
- Process
- Technology
THE PEOPLE

- People
- Process
- Technology
THE I.T. BALANCING ACT

To meet the growing demands of the business, developers and I.T. operations must find balance...
BRINGING DEV AND OPS TOGETHER

A methodology to deliver software more efficiently by emphasizing on collaboration, communication and integration between development and I.T. operations.
THE PROCESS AND THE TECHNOLOGY
### Typical Development Lifecycle

**Physical**
- 1. Have Idea
- 2. Get Budget
- 3. Submit Hardware 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 Application Server
- 11. Deploy Framework/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.

**Virtual**
- 1. Have Idea
- 2. Get Budget
- 3. Submit VM Request
- 4. Wait...
- 5. Deploy Application Server
- 6. Deploy Framework/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 New VMs
TYPICAL ENVIRONMENT PROVISIONING
TYPICAL DEPLOYMENT PIPELINE

LOCAL DEV → DEV → TEST → PROD

MANUAL, ERROR-PRONE PROCESS → MANUAL, ERROR-PRONE PROCESS → MANUAL, ERROR-PRONE PROCESS
WHAT IF...

We could **automate** environment provisioning?
We could **standardize** technology stacks and platforms?
We could **consolidate** our resources and pool usage?
REALIZING I.T. EFFICIENCY

COLLABORATION

AUTOMATION

STANDARDIZATION

CONSOLIDATION
APPLYING THE METHODOLOGIES
CLOUD CLOSES THE INNOVATION GAP

BUSINESS DEMAND FOR INNOVATION

I.T. INNOVATION GAP

CAPABILITY FOR INNOVATION ON TRADITIONAL INFRASTRUCTURE

CLOUD COMPUTING
CLOUD SERVERS ARE...

EPHEMERAL

ANONYMOUS

MULTI-TENANT
CLOUD COMPUTING CHARACTERISTICS

SELF-SERVICE PROVISIONING

RAPID ELASTICITY

RESOURCE POOLING

NETWORK ACCESS

MEASURED SERVICE
CLOUD SERVICE MODELS

IAAS

APPLICATION

DEVELOPER TOOLING
(Source Control, Build Tools, CI)

APPLICATION PLATFORM
(App Server, Middleware, Languages, Frameworks)

OPERATING SYSTEM

VIRTUAL GUEST IMAGES

COMPUTE RESOURCES
(CPU, RAM, Network, Disk)

Provided and Controlled by the Cloud Consumer

Automated and Managed by the Cloud Provider

INCREASED AUTOMATION

INCREASED CONTROL

PAAS

SAAS
The use of Platform-as-a-Service technologies will enable IT organizations to become more agile and more responsive to the business needs. —GARTNER
OPENSCHIFT IS PAAS BY RED HAT

- Multi-Language
- Auto-Scaling
- Self-Service
- Open Source
- Enterprise-Grade
- Secure
- Built on Red Hat

Powering Your Apps
RED HAT'S PAAS STRATEGY

Open Source PaaS Project

Public PaaS Service

On-premise or Private PaaS Software
OpenShift DEMO
OPENS SHIFT PAAS ON YOUR CHOICE OF CLOUD OR INFRASTRUCTURE...

Public - Hybrid - Private - Virtualization - Bare Metal
AN OPENSHIFT BROKER MANAGES MULTIPLE OPENSHIFT NODES

OpenShift Broker
Management and Orchestration Engine

OpenShift Nodes
Application Hosting Infrastructure

Public - Hybrid - Private - Virtualization - Bare Metal
A NODE IS AN INSTANCE OF RHEL

Broker (RHEL)

Node (RHEL)

Node (RHEL)

Red Hat Enterprise Linux

Public - Hybrid - Private - Virtualization - Bare Metal
OPENSHIFT USER APPLICATIONS RUNS IN CONTAINERS CALLED GEARS
GEARS USE SELINUX FOR PRE-CONFIGURED, NSA-GRADE SECURITY

SELinux Security
- “Jail” around Gear
- MAC instead of DAC
- Process/memory protection
- Pre-configured for you
GEARS USE LINUX CGROUPS FOR RESOURCE MANAGEMENT

CGroups Resource Mgmt
- Sets upper bounds
- CPU
- Memory
- Disk

Red Hat Enterprise Linux

Broker (RHEL)
Node (RHEL)
Node (RHEL)

Public - Hybrid - Private - Virtualization - Bare Metal
IDLE GEARS CAN BE “DE-HYDRATED” BY THE OPENSSHIFT BROKER
OPENSSHIFT MULTI-TENANCY PROVIDES DENSITY, EFFICIENCY, AND SECURITY

Broker (RHEL)

Node (RHEL)

Node (RHEL)

Red Hat Enterprise Linux

Node

Public - Hybrid - Private - Virtualization - Bare Metal
The use of Platform-as-a-Service technologies will enable IT organizations to become more agile and more responsive to the business needs. —GARTNER
TYPICAL DEVELOPMENT LIFECYCLE

**PHYSICAL**

1. Have Idea
2. Get Budget
3. Submit Hardware 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 Application Server
11. Deploy Framework/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.

**VIRTUAL**

1. Have Idea
2. Get Budget
3. Submit VM Request
4. Wait...
5. Deploy Application Server
6. Deploy Framework/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 New VMs
15. Etc.
A developer has a new idea for an application. First, they need to create a new gear in OpenShift...
GEAR CREATION (WEB, CLI, ECLIPSE)

- REST
- Web Console
- Eclipse IDE
- Cmd Line

DEVELOPER

Broker (RHEL)

Node (RHEL)

Node (RHEL)

Red Hat Enterprise Linux

Public - Hybrid - Private - Virtualization - Bare Metal
OPENSSHIFT AUTOMATES GEAR CONFIGURATION VIA CARTRIDGES

Cartridges are how OpenShift installs languages and middleware.

Red Hat Enterprise Linux

Broker (RHEL)

Node (RHEL)

Node (RHEL)

Public - Hybrid - Private - Virtualization - Bare Metal
CARTRIDGE TYPES

Broker (RHEL)
Node (RHEL)
Node (RHEL)

Red Hat Enterprise Linux

Public - Hybrid - Private - Virtualization - Bare Metal

REDEVELOPER

Web Console Eclipse IDE Cmd Line

REST

JBoss EWS
JBoss EAP
Java
MySQL
Postgres
tools
Ruby
Python
Etc.

CUSTOM

Etc.

Ruby
Python
Postgres
MySQL
Java

JBoss EWS
JBoss EAP
Java
MySQL
Postgres

JBoss EWS
JBoss EAP
Java
MySQL
Postgres

RED HAT CONFIDENTIAL
NOW, CODE AND PUSH

Developer pushes application code via GIT source code management system

DEVELOPER

Red Hat Enterprise Linux

Node

Public - Hybrid - Private - Virtualization - Bare Metal
OPENSHEET CAN AUTOMATED BUILD AND TEST WITH MAVEN AND JENKINS FOR CI

Broker (RHEL)  Node (RHEL)  Node (RHEL)

Red Hat Enterprise Linux Node

Public - Hybrid - Private - Virtualization - Bare Metal

MY GEAR

Code

Git Repo

jenkins

cron

logs

sshd

JBoss

MySQL

Maven

Jenkins

GPIO/SSH

Web Console
Eclipse IDE
Cmd Line

REST

DEVELOPER
HTTP(S) SERVED FROM GEARS

- RED HAT Enterprise Linux
- Node
- Public - Hybrid - Private - Virtualization - Bare Metal

MY GEAR
- JBoss
- Code
- Git Repo
- jenkins
- cron
- logs
- sshd

MY GEAR
- Maven
- MySQL

DEVELOPER
- REST
- Web Console
  Eclipse IDE
  Cmd Line

USER
- HTTP/S
- USER

GIT/SSH

Broker (RHEL)
- Node (RHEL)
- Node (RHEL)
STREAMLINING DEVELOPMENT WITH PAAS

**PHYSICAL**

1. Have Idea
2. Get Budget
3. Submit Hardware 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 Application Server
11. Deploy Framework/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.

**VIRTUAL**

1. Have Idea
2. Get Budget
3. Submit VM Request
4. Wait...
5. Deploy Application Server
6. Deploy Framework/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 New VMs
15. Etc.

**WITH PAAS**

1. Have Idea
2. Get Budget
3. Code
4. Test
5. Launch
6. Automatically Scale

CRAFTWORK

ASSEMBLY LINE
DEPLOYMENT PIPELINE WITH OPENSHEET
"Our motto is enable and get out of the way"

"With OpenShift we've built a push-button developer stack"

"In minutes we have you up and running in a fully connected container and you are developing"
THANK YOU.

Red Hat Video on YouTube
youtube.com/RedHatVideo

Want to keep in touch?

Red Hat on Facebook
facebook.com/RedHatInc

@RedHatNews on Twitter
twitter.com/RedHatNews

Red Hat on LinkedIn
linkedin.com/company/3545