How DevOps and microservices impact your application architecture and development

Arun Gupta - Director, Developer Advocacy and Technical Marketing, Red Hat
Christina Wong - Principal Product Marketing Manager, Red Hat
Arun – avid marathoner, STEM to kids
Arun Gupta, helping solve problems

Director, JBoss Middleware
- Technical Marketing
- Developer Advocacy

Hot topics
- JBoss Middleware
- Microservices
- DevOps
- Containers
- Developer Tooling
Christina Wong, crazy race car driver
Christina Wong, responsible product marketing manager

JBoss Middleware application runtimes and services
- Red Hat JBoss Enterprise Application Platform
- Red Hat JBoss Data Grid

Hot topics
- Devops
- Microservices
- Application platforms
- Fast data
What’s the problem?
Business back in the day...
Monolithic architecture

- Logically modular, deployed as monolith
- Great for small apps, small teams
- Simple systems
  - Dev, test, deploy
- Optimized for efficiency and latency
Business today...
What’s the problem?

● More, more, more
  ○ Customers demand more
  ○ Business demands more
  ○ FAST!

● Large complex monolithic apps
● Difficult maintenance
● Unwieldy
● Hard to test, trial new tech, fix
● Stuck with the original app

● But wait!!!
  ○ Enterprise stability, reliability
  ○ Predictable processes
  ○ Manage risk, technical debt
The goal?

● Meet the needs of modern business
  o interconnected
  o immediate
  o high variety of customer touch points
  o data from many sources
  o engaging
  o Personalized
● Experiment, fail fast
● Mobility
The solution

- Scales out (on x or z axis?)
- Fast to update, refresh
- Hybrid cloud
- Developers can innovate
- Operations can manage, maintain
- Easy to upgrade
- Easy to isolate problems
- Potentially polyglot
- Enables cross team communication

...the “modern application”
A new architecture for a modern business

![Diagram comparing traditional and modern web architectures](source)

Source: Gartner (October 2014)
More than just new approaches to applications...

- Processes
- Architectures
- Platforms
- Technologies
- Languages
Processes
Better deployment quality: 63%
Faster release frequency: 63%
Improved process visibility: 61%

DevOps Value in Action: Velocity at Amazon AWS

Max Deployments/Hour: 10,000
Mean Time Between Deployments (Seconds): 11.6
Software Deployments Causing an Outage: 0.001%
WORKED FINE IN DEV

OPS PROBLEM NOW
It’s DevOps!

It’s DevOps!

It’s DevOps!

It’s DevOps!

It’s DevOps!

It’s DevOps!
What is DevOps?

DevOps is an approach to process, culture, and tools for delivering increased business value and responsiveness through rapid, iterative, and high-quality IT service delivery.
“Dev”  “Ops”
Five “C”s of DevOps

- **Collaboration** between “dev” and “ops”
- **Culture**
- **Code** everything - application and configuration
- **Consistency** - automation over documentation
- **Continuous delivery**
<table>
<thead>
<tr>
<th></th>
<th>Initial</th>
<th>Managed</th>
<th>Defined</th>
<th>Quantitatively Managed</th>
<th>Optimizing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Culture &amp; Organization</strong></td>
<td>• Teams organized based on platform/technology</td>
<td>• One backlog per team</td>
<td>• Extended team collaboration</td>
<td>• Cross-team continuous improvement</td>
<td>• Cross functional teams</td>
</tr>
<tr>
<td></td>
<td>• Defined and documented processes</td>
<td>• Adopt agile methodologies</td>
<td>• Remove boundary dev/ops</td>
<td>• Teams responsible all the way to production</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Policing CI builds</td>
<td>• Polling CI builds</td>
<td>• Common process for all changes</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Build &amp; Deploy</strong></td>
<td>• Centralized version control</td>
<td>• Any build can be re-created from source control</td>
<td>• Commit hook CI builds</td>
<td>• Team priorities keeping codebase deployable over doing new work</td>
<td>• Zero touch Continuous Deployments</td>
</tr>
<tr>
<td></td>
<td>• Automated build scripts</td>
<td>• Management of build artifacts</td>
<td>• Build fails if quality is not met (code analysis, performance, etc.)</td>
<td>• Builds are not left broken</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• No management of artifacts</td>
<td>• Automated deployment scripts</td>
<td>• Push button deployment and release of any releasable artifact to any</td>
<td>• Orchestrated deployments</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Manual deployment</td>
<td>• Automated provisioning of environments</td>
<td>• Standard deployment process for all environments</td>
<td>• Blue Green Deployments</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Environments are manually provisioned</td>
<td>• Automated provisioning of environments</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Release</strong></td>
<td>• Infrequent and unreliable releases</td>
<td>• Painful infrequent but reliable releases</td>
<td>• Frequent fully automated releases</td>
<td>• No rollbacks, always roll forward</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Manual process</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data Management</strong></td>
<td>• Data migrations are performed manually, no scripts</td>
<td>• Data migrations using versioned scripts, performed manually</td>
<td>• Automated and versioned changes to datastores</td>
<td>• Automatic datastore changes and rollbacks tested with every</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>deployment</td>
<td></td>
</tr>
<tr>
<td><strong>Test &amp; Verification</strong></td>
<td>• Automated unit tests</td>
<td>• Automatic Integration Tests</td>
<td>• Fully automatic acceptance tests</td>
<td>• Verify expected business value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Separate test environment</td>
<td>• Static code analysis</td>
<td>• Automatic performance/security tests</td>
<td>• Defects found and fixed immediately (roll forward)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Test coverage analysis</td>
<td>• Manual exploratory testing based on risk based testing analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Information &amp; Reporting</strong></td>
<td>• Baseline process metrics</td>
<td>• Measure the process</td>
<td>• Automatic generation of release notes</td>
<td>• Report trend analysis</td>
<td>• Dynamic self-service of information</td>
</tr>
<tr>
<td></td>
<td>• Manual reporting</td>
<td>• Automatic reporting</td>
<td>• Pipeline traceability</td>
<td>• Real time graphs on deployment pipeline metrics</td>
<td>• Customizable dashboards</td>
</tr>
<tr>
<td></td>
<td>• Visible to report runner</td>
<td>• Visible to team</td>
<td>• Reporting history</td>
<td></td>
<td>• Cross-reference across organizational boundaries</td>
</tr>
</tbody>
</table>
Architectures
Client tier
- Mobile clients
- Wearables
- Internet of things
- Responsible for experience delivery

Delivery tier
- Optimizes content for proper display on device
- Caches content for performant delivery
- Drives personalization by using analytics to monitor user behavior

Aggregation tier
- Aggregates and federates services tier data
- Provides discovery for the underlying service library
- Performs data protocol translation (e.g., SOAP to JSON)

Services tier
- Existing on-premises systems of record, services, and data
- External third-party services (e.g., Box, Twilio, Urban Airship)

http://blogs.forrester.com/ted_schadler/13-11-20-mobile_needs_a_fourTier_engagement_platform
“America is all about speed. Hot, nasty badass speed.”
An architectural approach, that emphasizes the decomposition of applications into single-purpose, loosely coupled services managed by cross-functional teams, for delivering and maintaining complex software systems with the velocity and quality required by today’s digital business.
Business-driven microservices
Platforms

Technologies and tools

Languages
Platforms

**PaaS**

**ONLINE**
Public PaaS

Host your applications in the public cloud. OpenShift Online automates the provisioning, management and scaling of applications so that you can focus on development and creativity.

Learn more ▶

**ENTERPRISE**
Private PaaS

Accelerate your IT service delivery and streamline application development by leveraging PaaS in your own datacenters or private cloud.

Learn more ▶

**ORIGIN**
Community PaaS

Explore the community-driven open source upstream of OpenShift. Download the bits, join the growing community, and help extend the functionality of OpenShift.

Learn more ▶
Technologies and tools
Moving forward...
The effect on development teams

- What changes to implement?
- What scope?
- Timing?
- Who?
- Expectations?
- Tradeoffs?
Some guidance needed?

- Prioritize devops
- Evaluate needs according to the maturity matrix
- Determine tools, tech, etc
- Microservices – proceed with care!
Bringing it all together

DevOps → Microservices → Containers → Public & Private Clouds

Red Hat, JBoss, OpenShift

#redhat #rhsummit
LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.