Developing and Deploying Applications in a Multi-Site Hybrid Cloud

Tom Benninger - Solutions Architect, IT Application Lifecycle Management
Ben Pritchett - Principal Systems Engineer, IT PaaS
Ivan Atanasov - Principal Software Engineer, IT Application Delivery Services
May 2019 - Red Hat Summit - Boston
Agenda

● Introduction
● What do we want?
● Our platforms and environments
● Our solutions
● Demo: Pipeline in Action
● The Future
● Q&A
Who we are

Tom Benninger
Solutions Architect
Red Hat IT: Application Lifecycle Management Team

Ben Pritchett
PaaS Tech Lead / Principal Systems Engineer
Red Hat IT: PaaS Team

Ivan Atanasov
Principal Software Engineer
Red Hat IT: Application Delivery Services
A few notes on the content...

- The content of this presentation represents the actual patterns and processes used by Red Hat IT to manage business critical applications.
- Our live demo will run in our real pre-production and productions environment, but will use a dummy application.
- This is an overview, any section of the document could provide hours of content.
- Our examples and demos assume some familiarity with common modern development tools and technologies.
What do we want?
Fast, efficient, high-quality, and secure digital delivery across multiple sites & providers.
Basically, enterprise quality delivery.

What does this require?
Multi-Site Hybrid Clouds
Service Owner Point of View

Global Content Delivery
Cloud Burst
Business Continuity
Multi-Site Hybrid Clouds
Your CFO’s Point of View

Cost management across cloud providers
The ability to avoid vendor lock-in
Automated Delivery Pipelines

Development should use standard, shared patterns.

These processes should be automated, from end to end.
Compliance and Security

Legal
Contractual
Internal policy (InfoSec, etc.)

Secure Service Architecture
Digital Supply Chain Security
Digital Supply Chain Security

- Raw Materials: Code, binaries, and packages pulled from...
- Suppliers: Open Source projects, commercial vendors, etc.
- Manufacturing: Your code and how you build and package your applications.
- Distribution: How you distribute and deploy your applications.
Our Platform and Environments
Our Multi-Site Hybrid Cloud

Datacenter 2

Public Cloud

Datacenter 1
Why Three Sites?

- It provides solutions for “N” sites
  - Solutions for one or two sites typically don’t scale to many sites.
  - Solving for 3 sites makes you tackle scaling problems up front.

- Service Quorums
  - Distributed systems typically require at least 3 nodes to ensure service quorums are available when a single site goes down.

- Reduced spare capacity requirements for each site
  - If one site goes down, its workload can be spread out across two sites
Our Core Platform of Choice

OpenShift Container Platform provides our standard platform, running in all 3 sites. It is now the default choice for Red Hat IT’s applications.
Red Hat IT’s Managed Platform

- Built on top of OpenShift
- Cross-site image replication
- Cross-site / cross-cluster automation
- Standard Templates & Application Building blocks
- Application Security Testing
  - Static Application Security Testing
  - OWASP Dependency Check
Managed Platform Enablement

Identity Management integration

Service Account automation

Public Cloud

Datacenter 1

Datacenter 2
Managed Platform Enablement

Pipeline as Code

```python
node(cluster.jenkinsNodeSelector()) { 
  def scmVars
  def gitBranch = AdsPipelineUtils.gitBranch(params.branch, "params.branch")

  stage('Checkout Templates') {
    dir('project_templates') {
      scmVars = checkout scm
    }
  }

  def config = readYaml file: 'project_templates/infrastructure/pipeline/config.yaml'
```
Our Delivery Solution
Key Decisions We Made

- Design for end to end automation
- Design for modularity / componentization
- Provide ‘external orchestration’
- Build once, deploy globally
- No “secrets” in images
- No environmental configs or images (dev image is prod image)
Independent Development

Start Up
- Initialize development resources and environments.

Feature Branch Development
- Developers independently code, build, deploy, and test as needed to delivery high quality code.

Feature Branch Development Processes

- Develop Code, Tests, OpenShift Configs
  - Local IDEs, etc.

- Jenkins Build Mgmt
  - Team Jenkins

- Maven Build
  - Local Maven

- Dependency Analysis
  - Local Maven

- Static Application Security Testing
  - SonarQube

- Build Triggered
  - PaaS Jenkins

- Build Configs Run
  - MP / OCP S2I or Dockerfile

- Test App in Development Environment
  - Team Jenkins

Feature Branch Development Complete

#redhat #rhsummit
Merge Management

The team reviews and merges or declines the delivered code, test results, and other resources.

Merge Management Processes

- Code Review
- Testing Results Review
- Discussion & Approval
- Merge & Delete Branch
- Delete Feature Environment

Final Step: Merge Management Complete
A final build is done on the merged code, and the resulting artifacts are distributed to all sites.
Deployment

- Dev Environment Deploy: The build is deployed and tested in central DEV environments.
- Int Environment Deploy: The build is deployed and tested in central INT environments.
- Prod Environment Deploy: The build is deployed and validated in central PROD environments.

Dev Deploy Processes:
- Rolling Deploy Triggered in Each Site: PaaS Jenkins
- OpenShift Resource Configs Updated: MP / OCP
- OCP Resources Deployed: MP / OCP
- Functional / Integration Testing: Team Jenkins
- Dev Deploy Complete
The Pipeline In Action
The Future
The Operator Framework

Operators are a new way to manage not just the definition of resources in OpenShift, but their entire lifecycle.

- We anticipate this allowing us to simplify and standardize our development and deployment processes further, using operator services rather than custom Jenkins jobs for many things.

Kubernetes Federation v2

Federation allows OpenShift resources to be spun up across multiple OpenShift clusters.

- This potentially removes the need for external orchestration to deploy to many sites.
- May not solve for advanced release processes (blue/green, a/b, etc.)
- Currently in developer preview in OCP v3.11
- https://blog.openshift.com/kubernetes-federation-v2-on-openshift-3-11/
For Red Hat IT

Using these patterns we’ve developed to extend our multi-site hybrid cloud to:

1. New geographies
2. New cloud providers
Q & A
RED HAT ON RED HAT

RED HAT I.T.
Powered by Red Hat Products

Ask us how it’s done

Visit us at the “Ask Me Anything” booth and ask us about how we implement and deploy Red Hat products!

redhat.com/redhat-on-redhat
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Room</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tues, May 7</td>
<td>1:30 - 2:15pm</td>
<td>Room 104AB</td>
<td>OpenShift, Hybrid Clouds, and the Enterprise</td>
</tr>
<tr>
<td>Tues, May 7</td>
<td>1:30 - 2:15pm</td>
<td>Room 153</td>
<td>The whole is greater than the sum of its parts: Technology and collaboration at Red Hat</td>
</tr>
<tr>
<td>Tues, May 7</td>
<td>2:30 - 2:50pm</td>
<td>Room 157C</td>
<td>Pulling the puppet strings with Ansible</td>
</tr>
<tr>
<td>Wed, May 8</td>
<td>1:00 - 3:00pm</td>
<td>Room 253C</td>
<td>Red Hat IT features: Sidecar that authentication</td>
</tr>
<tr>
<td>Thurs, May 9</td>
<td>11:00 - 11:45am</td>
<td>Room 104AB</td>
<td>Red Hat on Red Hat: Transitioning Red Hat IT to hybrid cloud infrastructure using OpenStack and Ceph Storage</td>
</tr>
<tr>
<td>Thurs, May 9</td>
<td>3:15 - 4:00pm</td>
<td>Room 155</td>
<td>Developing and running cloud-native apps on OpenShift in Red Hat's IT organization</td>
</tr>
<tr>
<td>Thurs, May 9</td>
<td>3:15 - 4:00pm</td>
<td>Room 154</td>
<td>Developing and deploying applications in a multisite hybrid cloud</td>
</tr>
</tbody>
</table>
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/redhat