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?

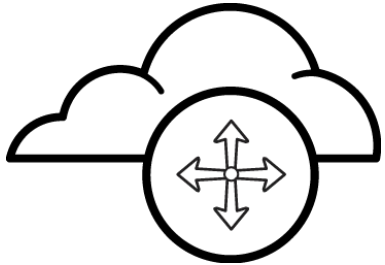
Fast, efficient, high-quality, and secure digital delivery across multiple sites & providers.

Basically, enterprise quality delivery.

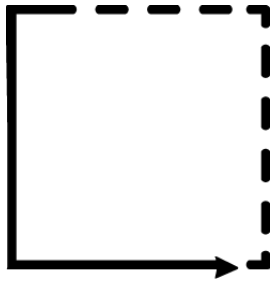
What does this require?

Multi-Site Hybrid Clouds
Service Owner Point of View

Global Content Delivery



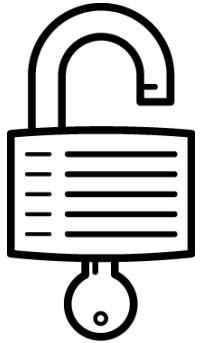
Cloud Burst



Business Continuity

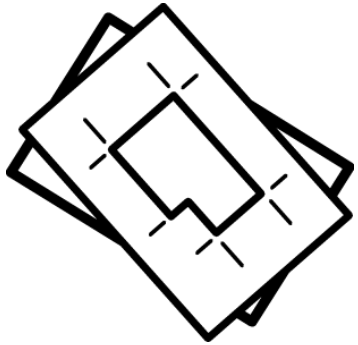
Multi-Site Hybrid Clouds
Your CFO's Point of View

Cost management across cloud providers

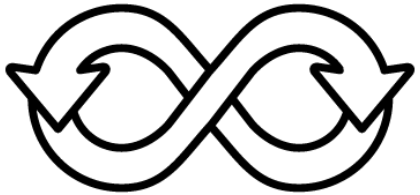


The ability to avoid vendor lock-in

Automated Delivery Pipelines



Development should use standard, shared patterns.



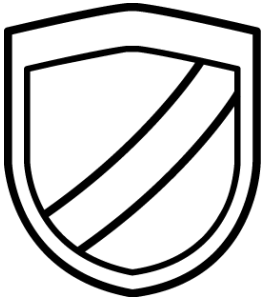
These processes should be automated, from end to end.

Compliance and Security

Legal

Contractual

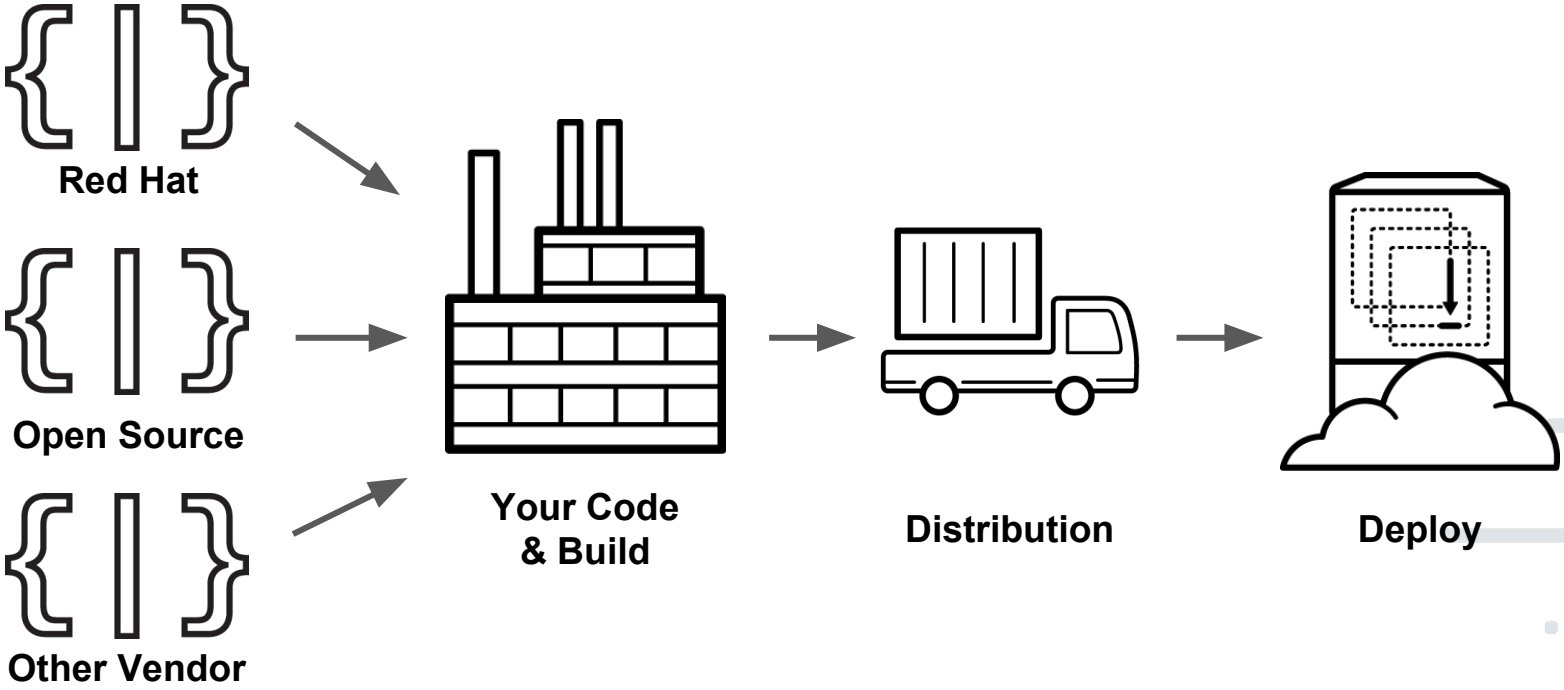
Internal policy
(InfoSec, etc.)



Secure Service Architecture

Digital Supply Chain Security

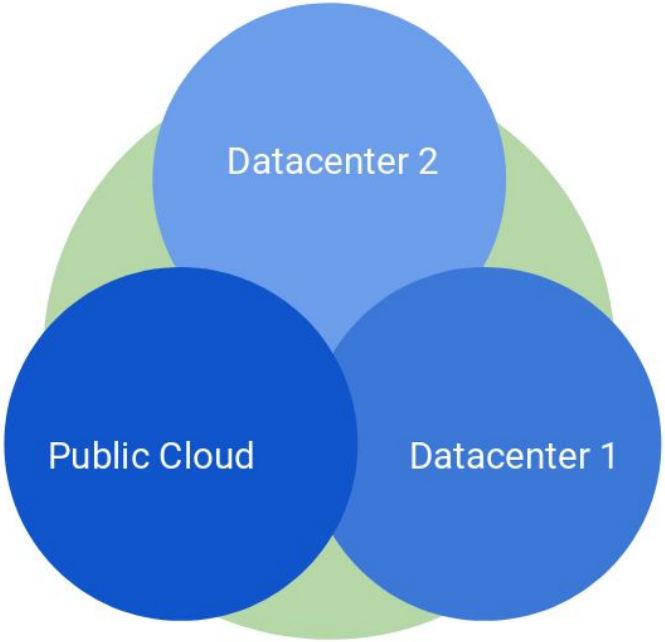
Digital Supply Chain Security



How

Our Platform and Environments

Our Multi-Site Hybrid Cloud



Why Three Sites?

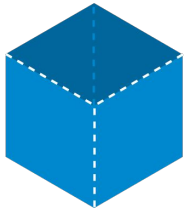
- It provides solutions for “N” sites
  - Solutions for one or two sites typically don’t scale to many sites.
  - Solving for 3 sites makes you tackle scaling problems up front.
- Service Quorums
  - Distributed systems typically require at least 3 nodes to ensure service quorums are available when a 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

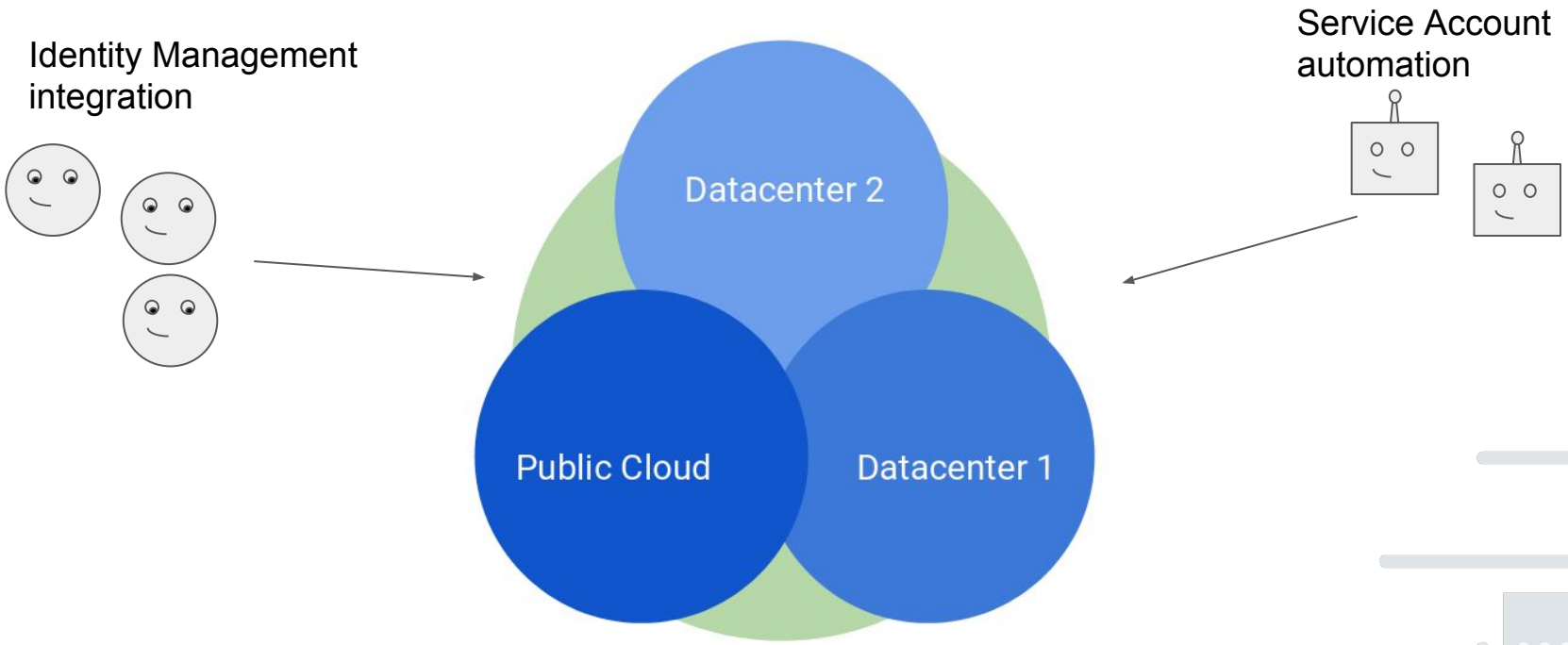
OpenShift Container Platform provides our standard platform, running in all 3 sites. It is now the default choice for Red Hat IT's applications.

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



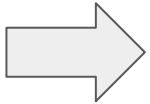
Managed Platform Enablement

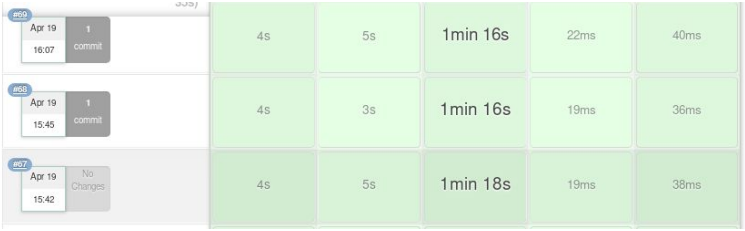
Pipeline as Code

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

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

    def config = readYaml file: 'project_templates/infrastructure/pipeline/config.yaml'
```

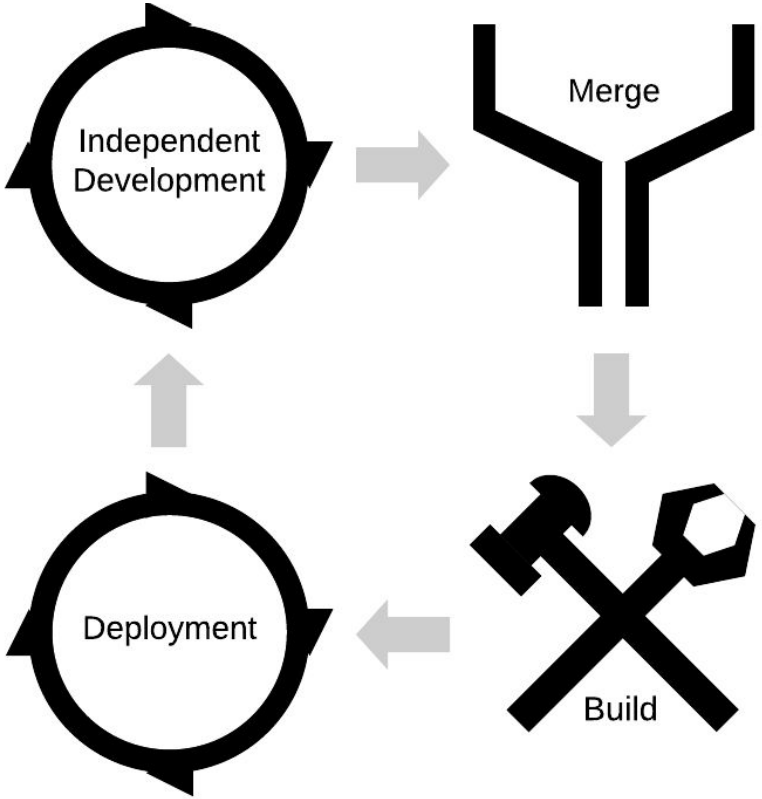


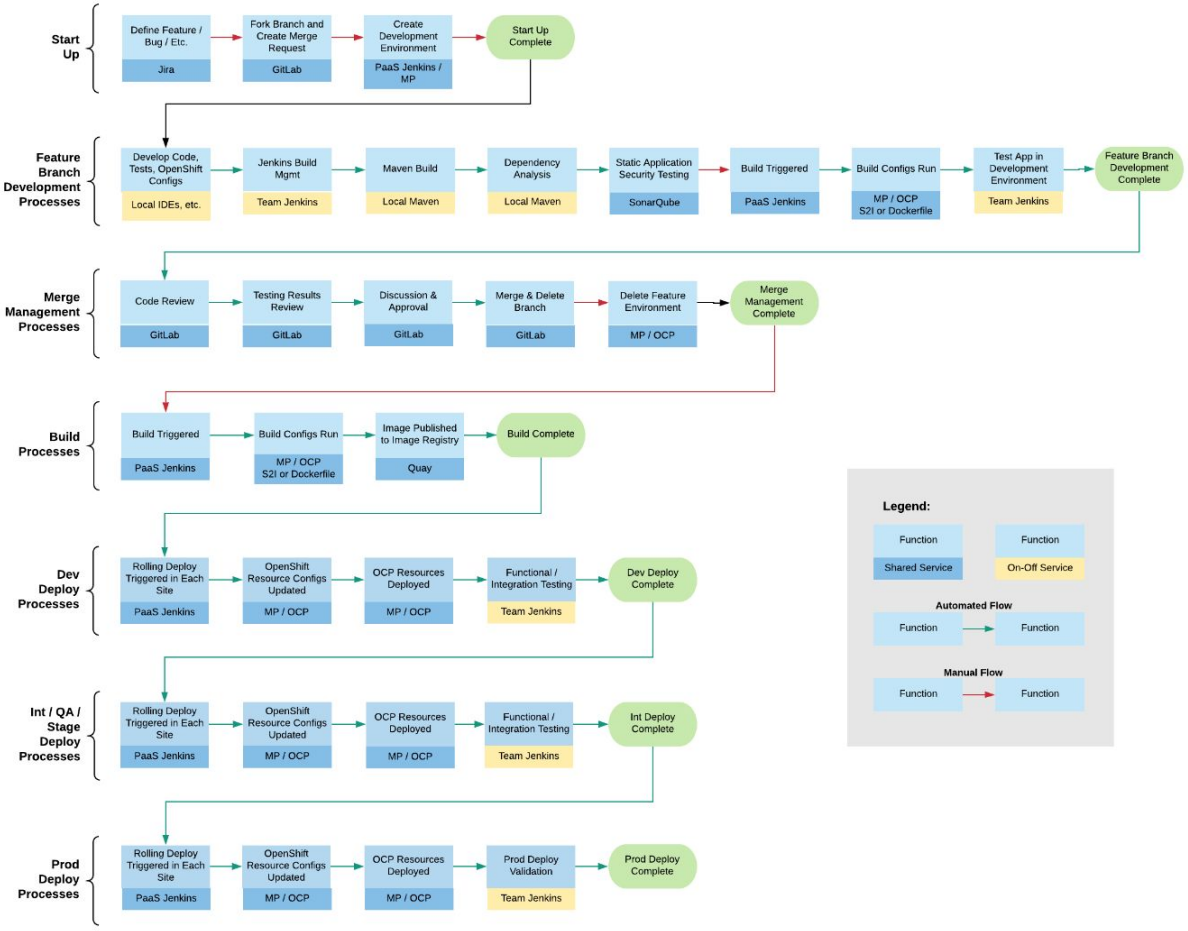


Our Delivery Solution

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)

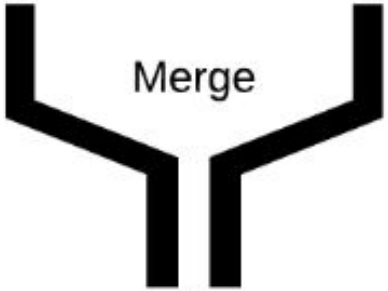

Independent
Development
Merge
Build
Deployment


Start Up
Define Feature / Bug / Etc.
Jira
Fork Branch and Create Merge Request
GitLab
Create Development Environment
PaaS Jenkins / MP
Start Up Complete
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
Merge Management Processes
Code Review
GitLab
Testing Results Review
GitLab
Discussion & Approval
GitLab
Merge & Delete Branch
GitLab
Delete Feature Environment
MP / OCP
Merge Management Complete
Build Processes
Build Triggered
PaaS Jenkins
Build Configs Run
MP / OCP S2I or Dockerfile
Image Published to Image Registry
Quay
Build Complete
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
Int / QA / Stage 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
Int Deploy Complete
Prod Deploy Processes
Rolling Deploy Triggered in Each Site
PaaS Jenkins
OpenShift Resource Configs Updated
MP / OCP
OCP Resources Deployed
MP / OCP
Prod Deploy Validation
Team Jenkins
Prod Deploy Complete
Legend:
Function
Shared Service
Function
On-Off Service
Automated Flow
Function
Function
Manual Flow
Function
Function

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

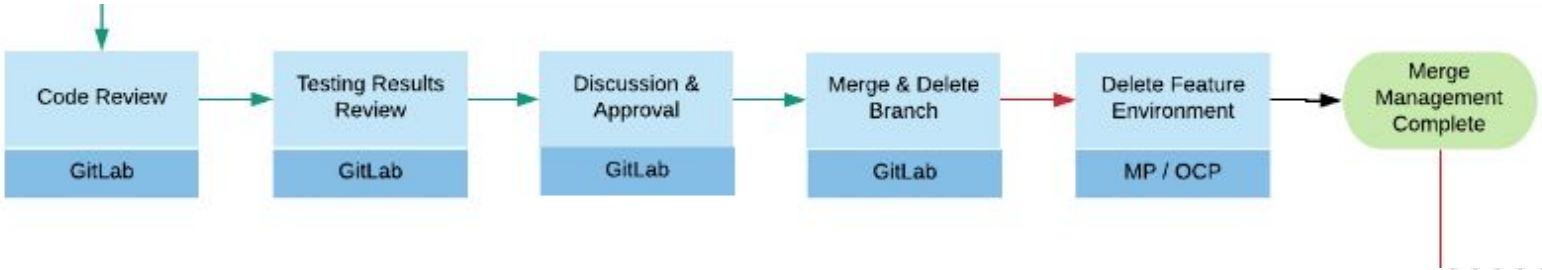
Develop Code, Tests, OpenShift Configs	Jenkins Build Mgmt	Maven Build	Dependency Analysis	Static Application Security Testing	Build Triggered	Build Configs Run	Test App in Development Environment	Feature Branch Development Complete
Local IDEs, etc.	Team Jenkins	Local Maven	Local Maven	SonarQube	PaaS Jenkins	MP / OCP S2I or Dockerfile	Team Jenkins	



Merge Management

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

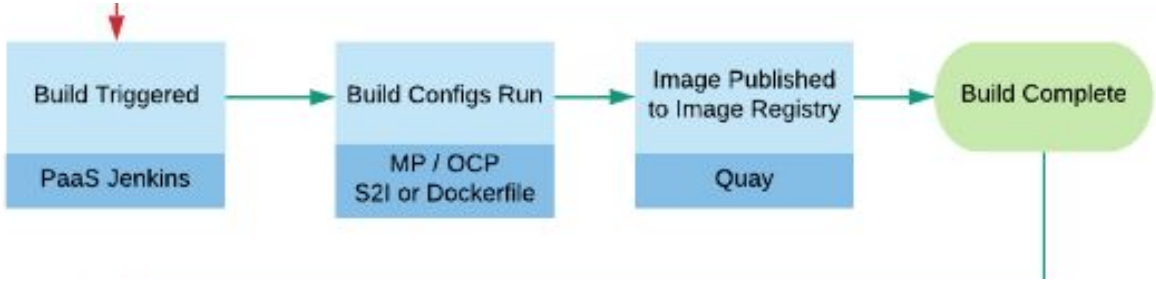
Merge Management Processes

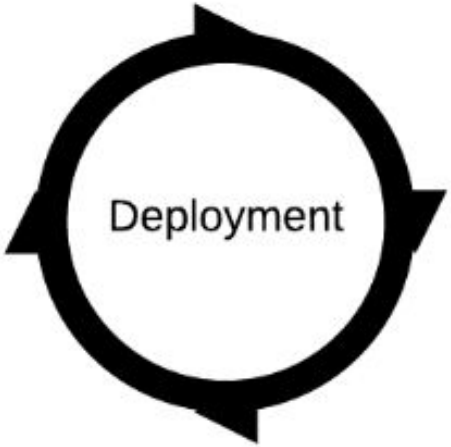


Build

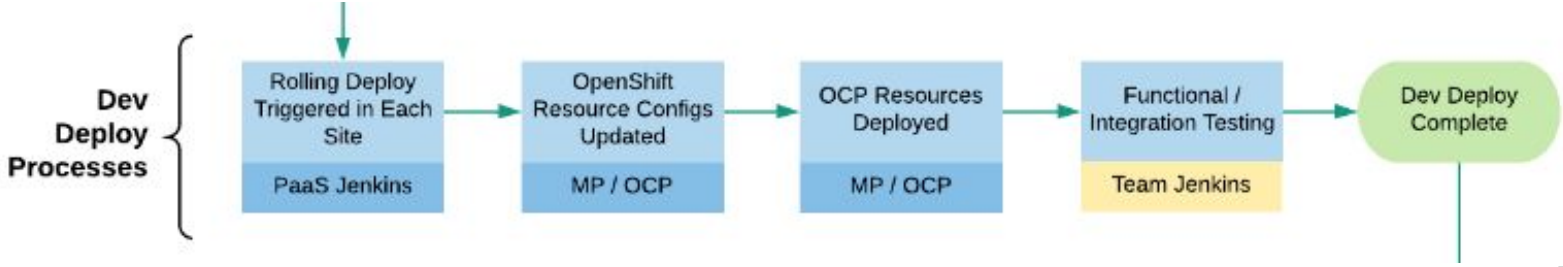
A final build is done on the merged code, and the resulting artifacts are distributed to all sites.

Build Processes

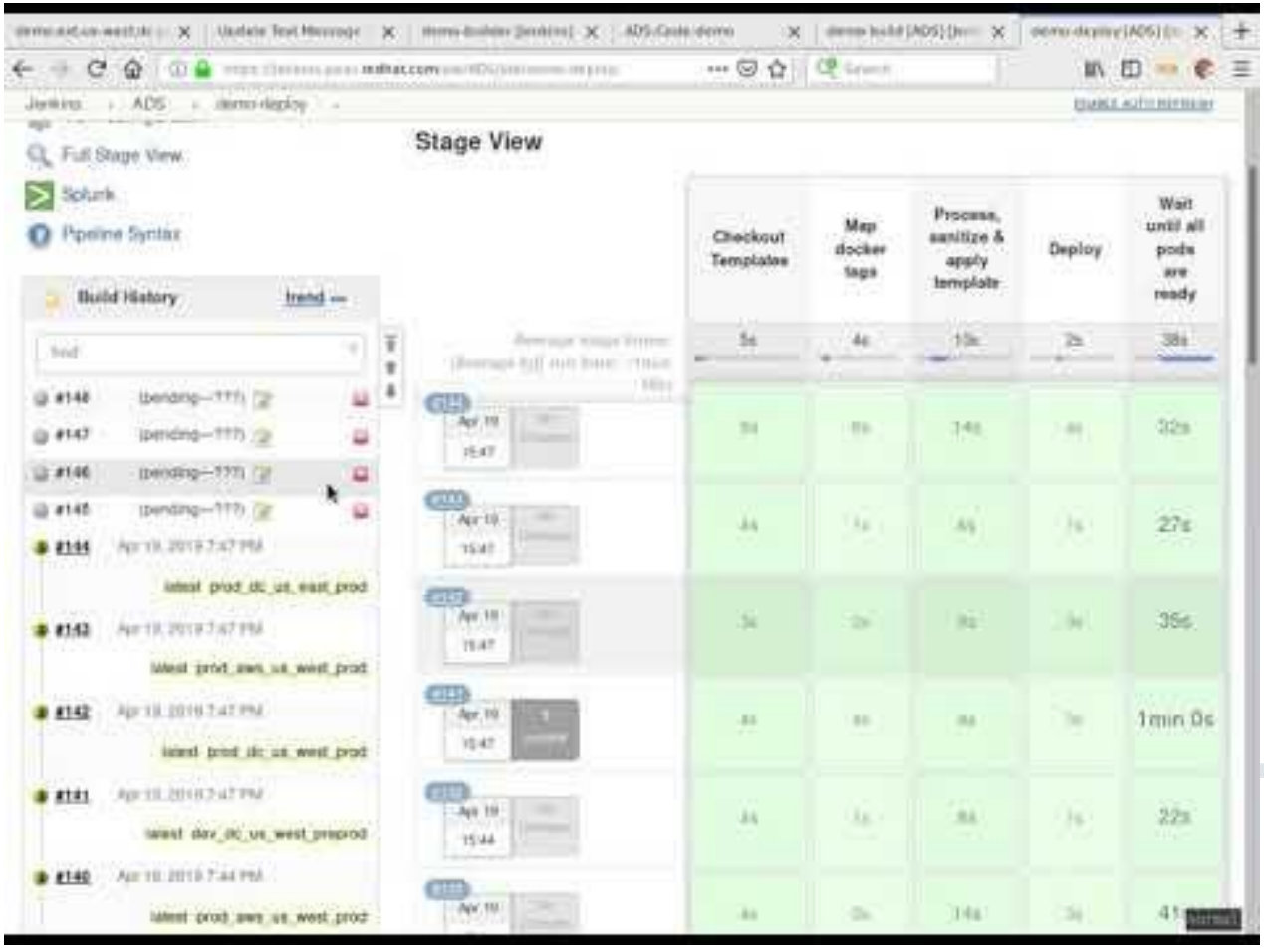




Dev Environment Deploy	The build is deployed and tested in central DEV environments.
Int Environment Deploy	The build is deployed and tested in central INT environments.
Prod Environment Deploy	The build is deployed and validated in central PROD environments.



The Pipeline In Action


Jenkins
ADS
demo-deploy
Stage View
Full Stage View
Splunk
Pipeline Syntax
Build History
trend
Checkout Templates
Map docker tags
Process, sanitize & apply template
Deploy
Wait until all pods are ready
5s
4s
19s
2s
38s
#148 (pending—???)
#147 (pending—???)
#146 (pending—???)
#145 (pending—???)
#144
#143
#142
#141
#140
latest_prod_dc_us_east_prod
latest_prod_aws_us_west_prod
latest_prod_dc_us_west_prod
latest_dev_dc_us_west_preprod
latest_prod_aws_us_west_prod
27s
35s
1min 0s
22s

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.
- https://blog.openshift.com/introducing-the-operator-framework/

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

RED HAT ON RED

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

RED HAT SUMMIT

THANK YOU


plus.google.com/+RedHat


facebook.com/redhatinc


linkedin.com/company/red-hat


twitter.com/redhat


youtube.com/user/RedHatVideos