



@ OCP Trilogy
14/11/2017

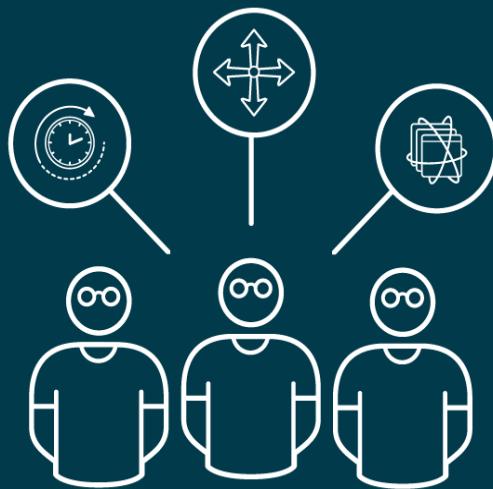
Openshift Trilogy III: Openshift for Devs and Ops

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

WHAT DOES BUSINESS WANT?



CHANGE IMPROVES DEVELOPMENT, CHALLENGES OPERATIONS



DEVELOPERS



OPERATIONS

WHAT DOES BUSINESS WANT?

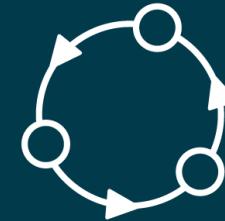
Speed



Agility



Efficiency



WHAT DOES BUSINESS WANT?

Speed



Agility



Efficiency



YOUR DEVOPS CAN MAKE IT,



WHAT DOES BUSINESS WANT?

Speed



Agility



Efficiency

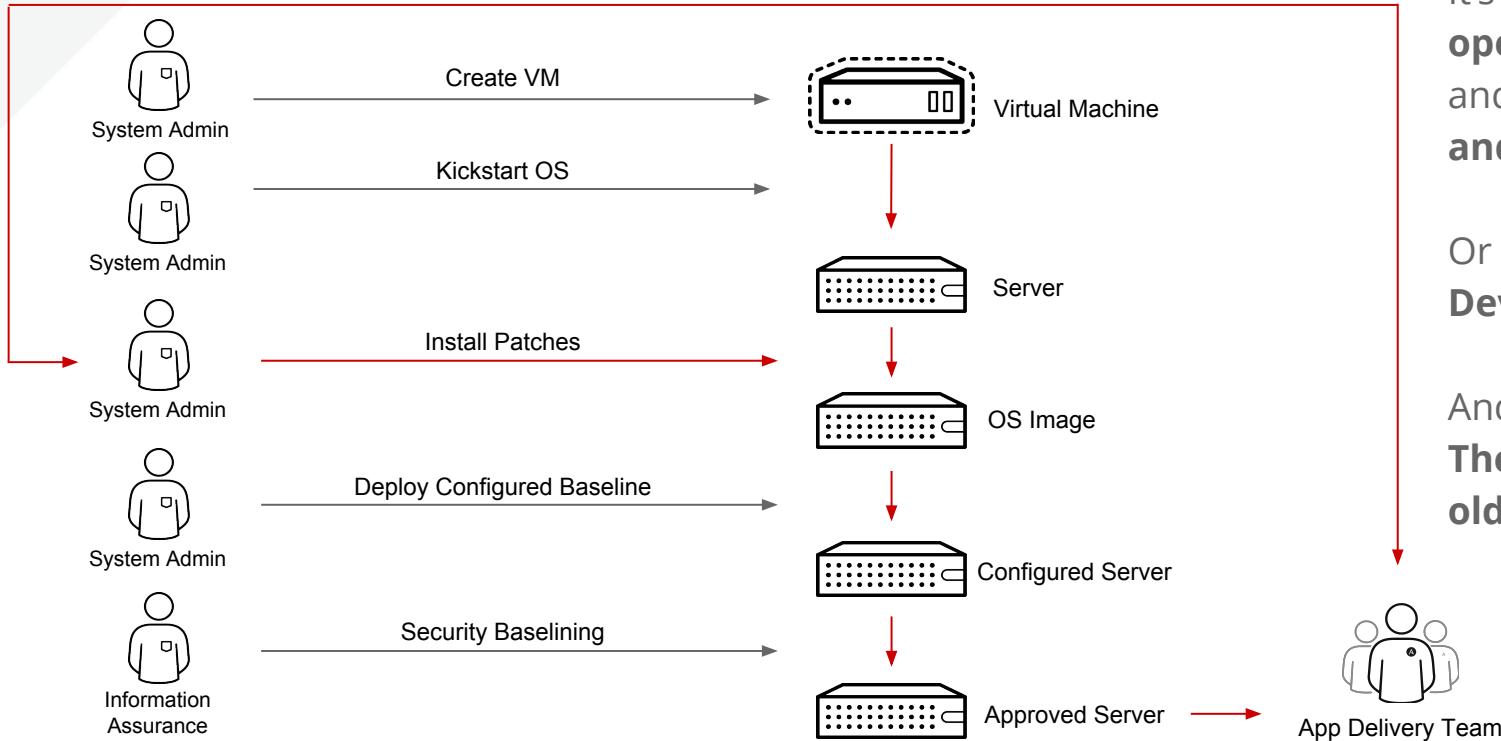


YOUR DEVOPS CAN MAKE IT, IF YOU LET THEM:



~~INSTALL; CONFIGURE - REPEAT~~
AUTOMATE

THE GOOD OLD DAYS...

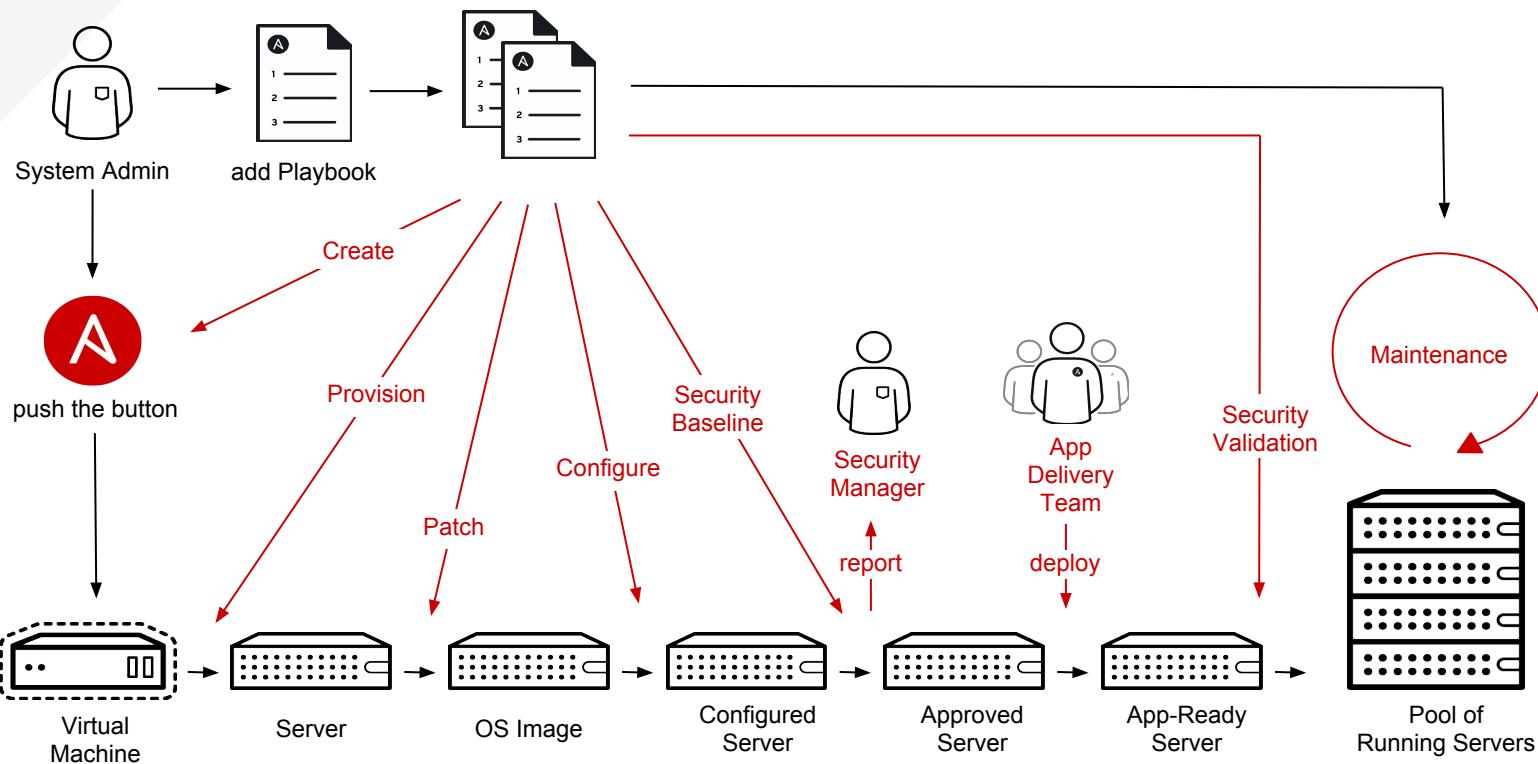


It's hard to **run IT operations** this way and **deliver flexible and agile IT services**.

Or being the **Ops** in **DevOps**.

And yes, you can work **The Cloud** in the **same old way...**

NOTHING ROUTINE SHOULD BE DONE MANUALLY





A N S I B L E
by Red Hat®

AUTOMATE YOUR I.T. PROCESSES & DEPLOYMENTS

- Simple & powerful language
- No agents to install
- Scale with Ansible Tower



WHAT DOES BUSINESS WANT?

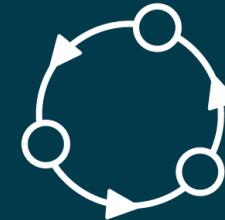
Speed



Agility



Efficiency



YOUR DEVOPS CAN MAKE IT, IF YOU LET THEM:

AUTOMATE

WHAT DOES BUSINESS WANT?

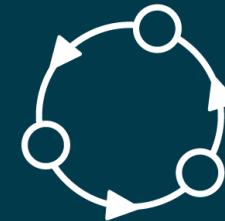
Speed



Agility



Efficiency



YOUR DEVOPS CAN MAKE IT, IF YOU LET THEM:

AUTOMATE

BUILD & RUN

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

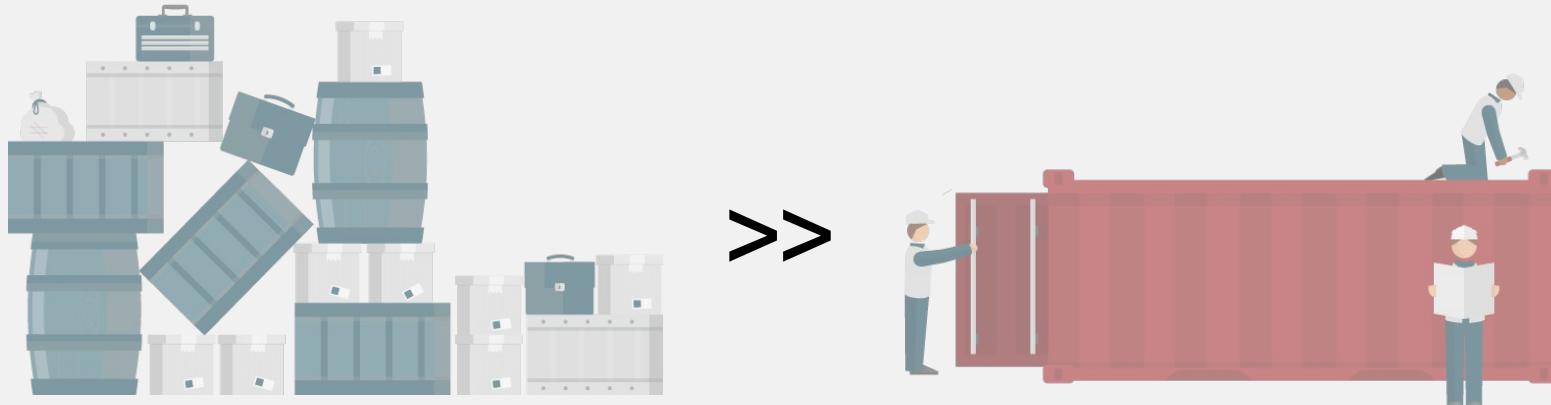
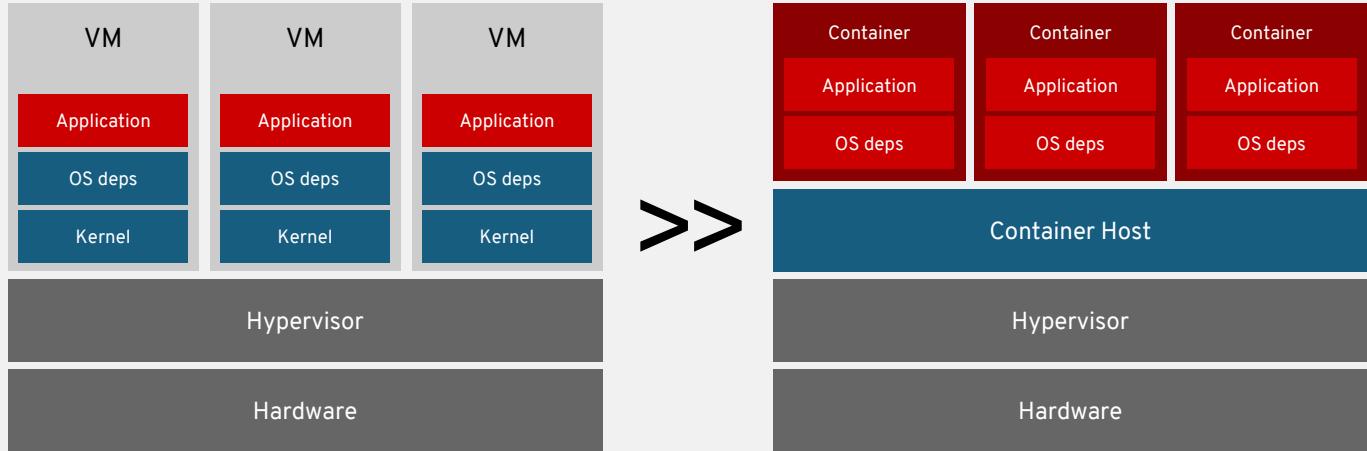
Container PaaS

How to Build an App:

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



Containers!

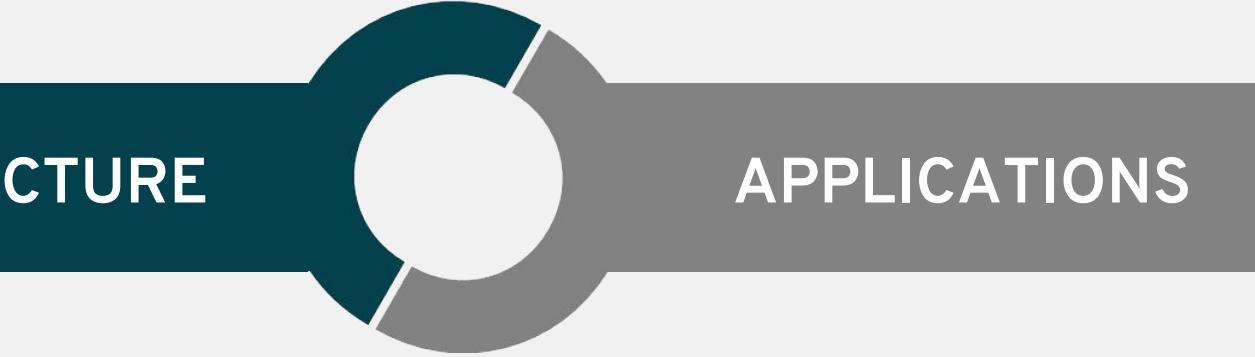


WHAT ARE CONTAINERS?

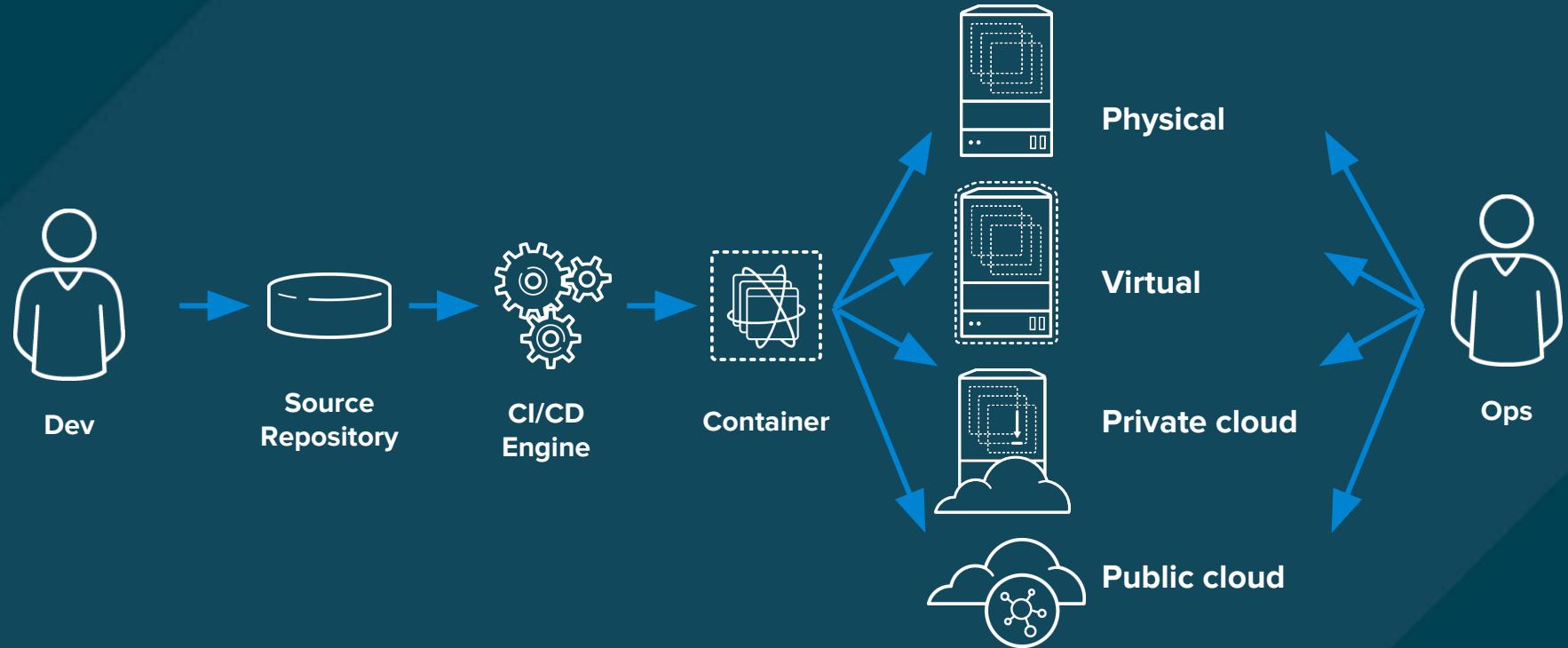
It Depends Who You Ask

INFRASTRUCTURE

APPLICATIONS

- 
- Application processes on a shared kernel
 - Simpler, lighter, and denser than VMs
 - Portable across different environments
 - Package apps with all dependencies
 - Deploy to any environment in seconds
 - Easily accessed and shared

Build & Run Containers !



WE NEED MORE THAN JUST CONTAINERS

Scheduling

Decide where to deploy containers

Lifecycle and health

Keep containers running despite failures

Discovery

Find other containers on the network

Monitoring

Visibility into running containers

Security

Control who can do what

Scaling

Scale containers up and down

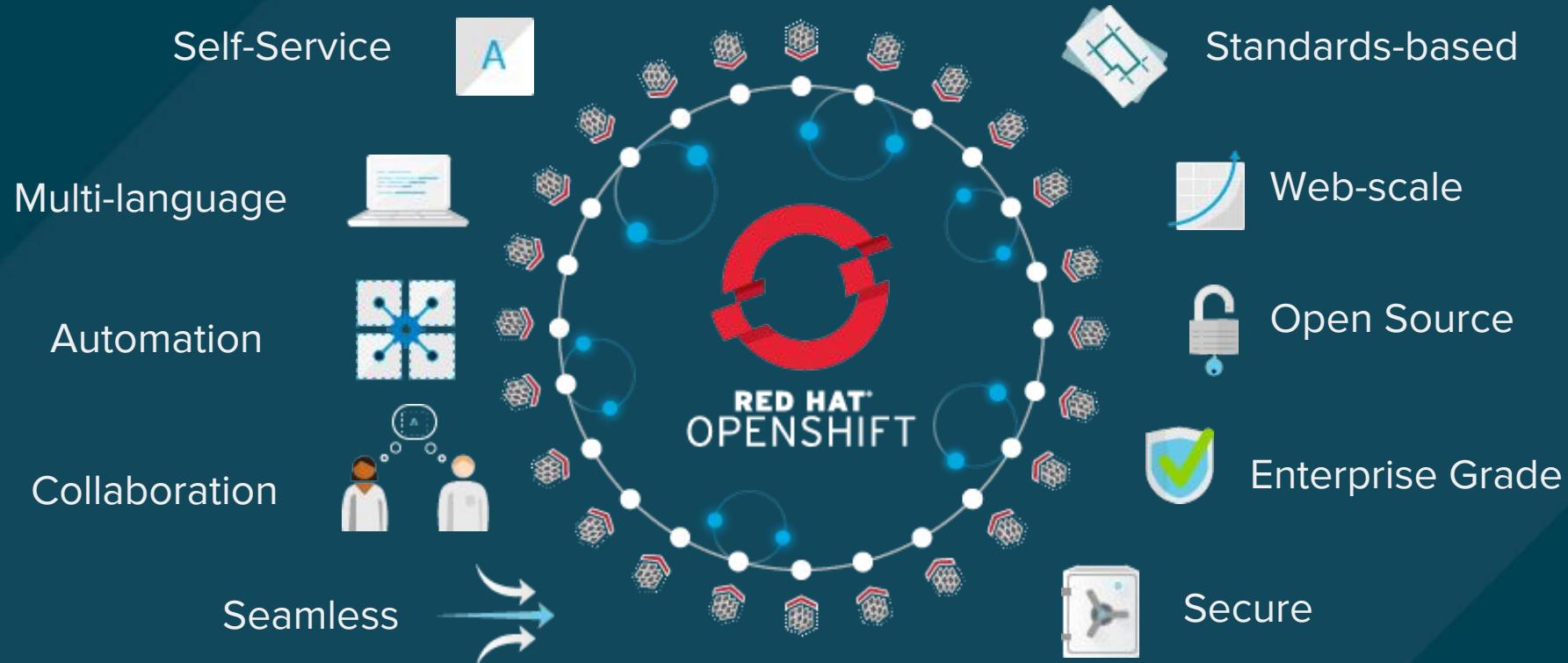
Persistence

Survive data beyond container lifecycle

Aggregation

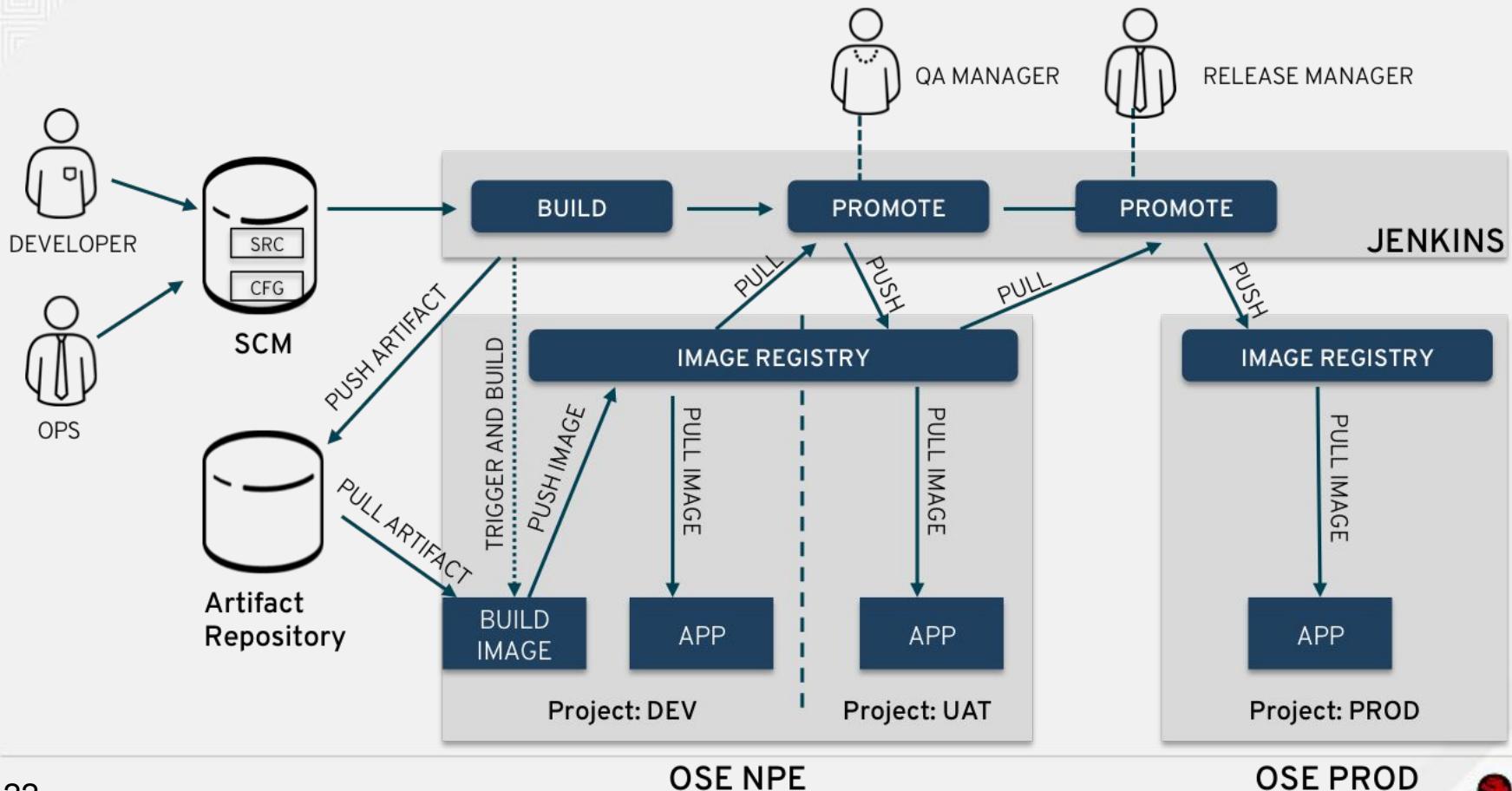
Compose apps from multiple containers

Container Platform !





OpenShift CI/CD flow

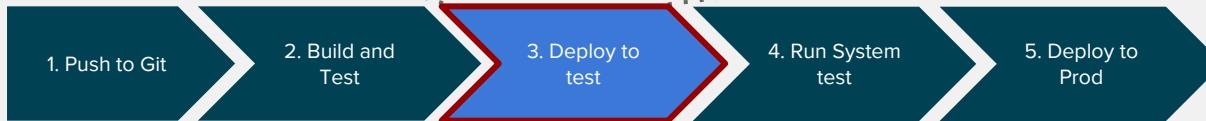


AUTOMATE CI/CD INFRASTRUCTURE

Example of provisioning part of a CD pipeline:



Example of a CD pipeline:



AUTOMATE CI/CD INFRASTRUCTURE WITH OPENSHIFT

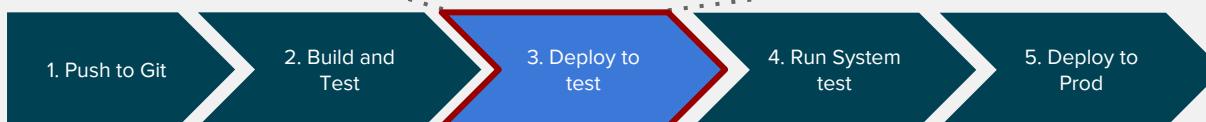
Suggested Tools:



Example of provisioning part of a CD pipeline:



Example of a CD pipeline:



WHAT DOES BUSINESS WANT?

Speed



Agility



Efficiency



YOUR DEVOPS CAN MAKE IT, IF YOU LET THEM:

AUTOMATE

BUILD & RUN



WHAT DOES BUSINESS WANT?

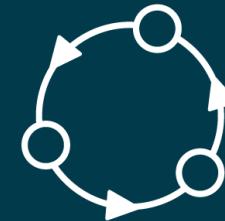
Speed



Agility



Efficiency



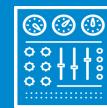
YOUR DEVOPS CAN MAKE IT, IF YOU LET THEM:

AUTOMATE

BUILD & RUN

CONTROL &
MANAGE

RED HAT® CLOUDFORMS



Monitoring



Policy



Inventory &
Chargeback



Unified Hybrid
Management



CONTAINERS

Red Hat® OpenShift Container Platform



VIRTUALIZATION

VMware®
Microsoft® Hyper-V



PRIVATE CLOUD

Red Hat OpenStack® Platform



PUBLIC CLOUD

Amazon® Web
Services
Microsoft Azure



RED HAT®
CLOUDFORMS

DELIVER SERVICES ACROSS HYBRID CLOUDS

- Unified management
- Self-service provisioning
- Policy-driven compliance





Speed



Agility



Efficiency



YOUR DEVOPS CAN MAKE IT, IF YOU LET THEM:

AUTOMATE

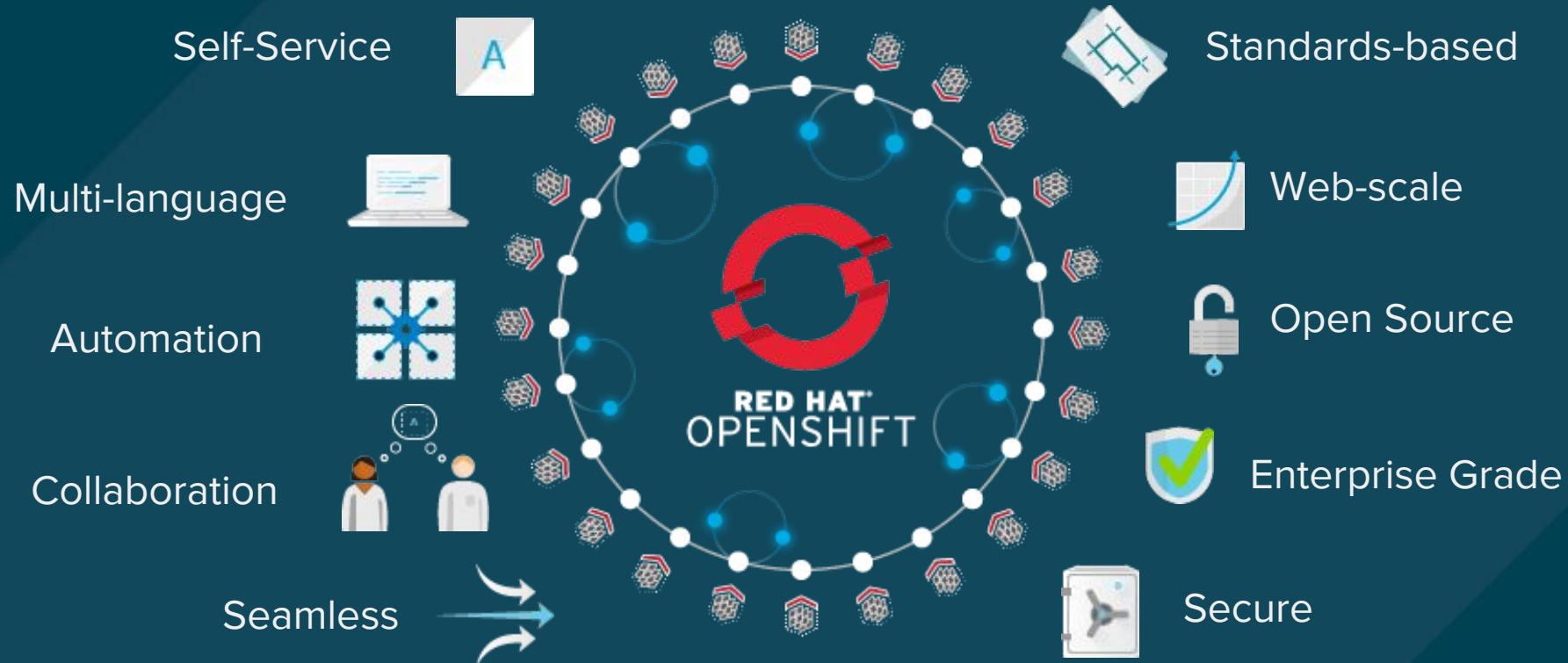
BUILD & RUN

CONTROL &
MANAGE



redhat.[®]

Container Platform !

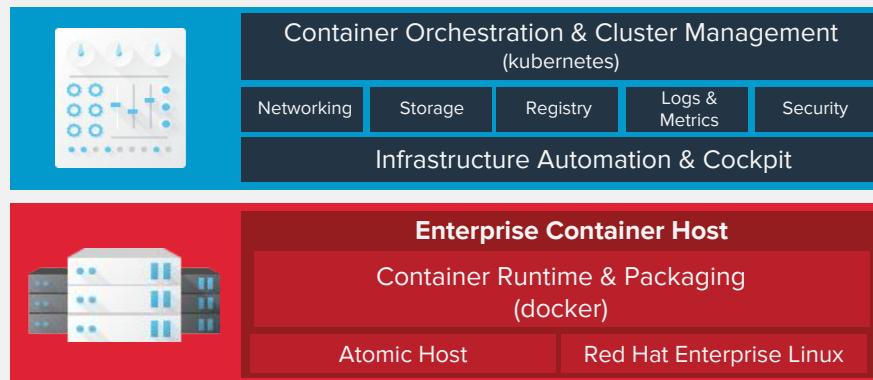


OPENSHIFT CONTAINER PLATFORM



Trusted by Fortune Global 500 companies

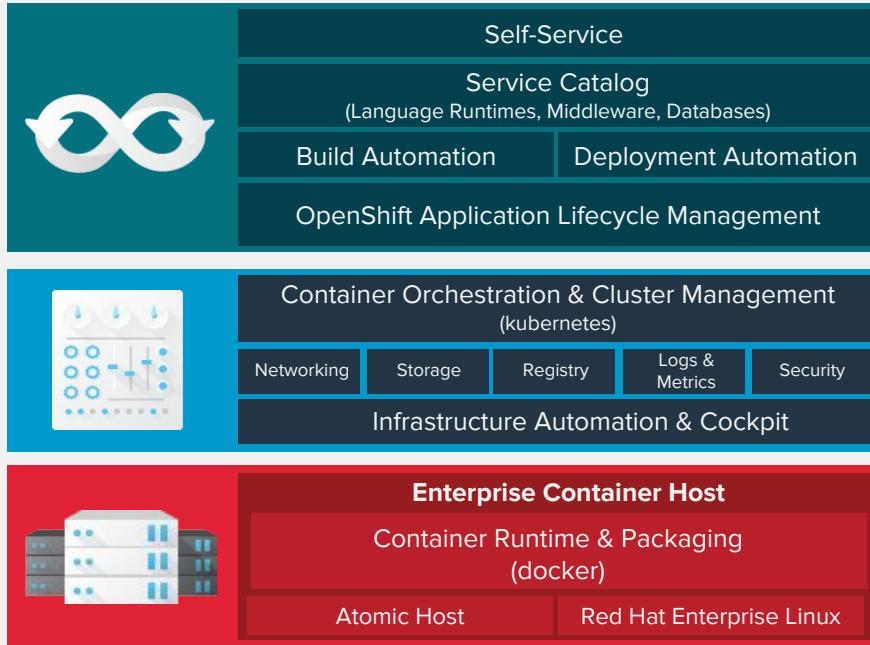
OPENSHIFT CONTAINER PLATFORM



Enterprise Kubernetes++
container orchestration

Trusted by Fortune Global 500
companies

OPENSHIFT CONTAINER PLATFORM

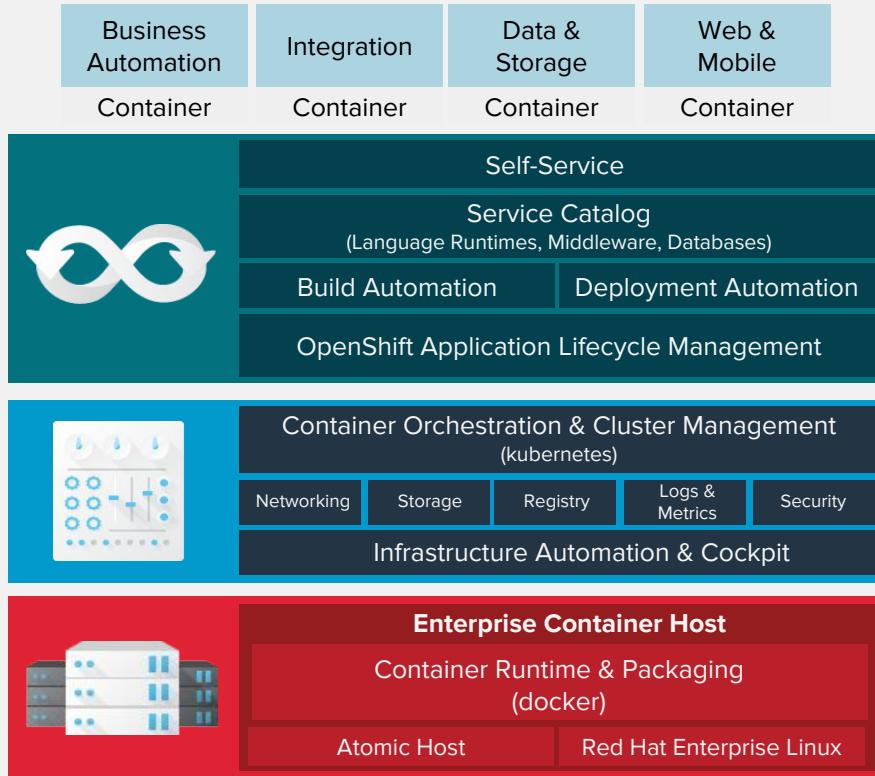


Developer Experience

Enterprise Kubernetes++
container orchestration

Trusted by Fortune Global 500
companies

OPENSHIFT CONTAINER PLATFORM



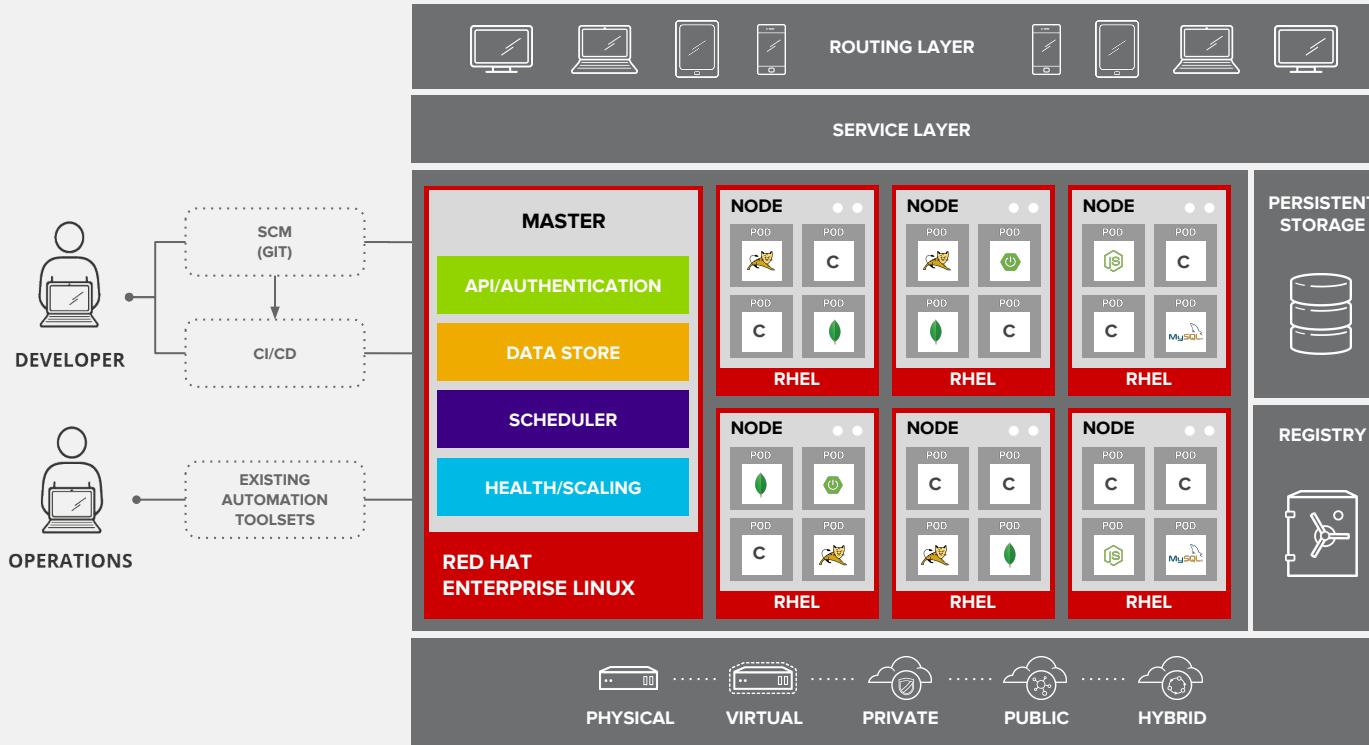
Traditional, stateful, and cloud-native apps

Developer Experience

Enterprise Kubernetes++ container orchestration

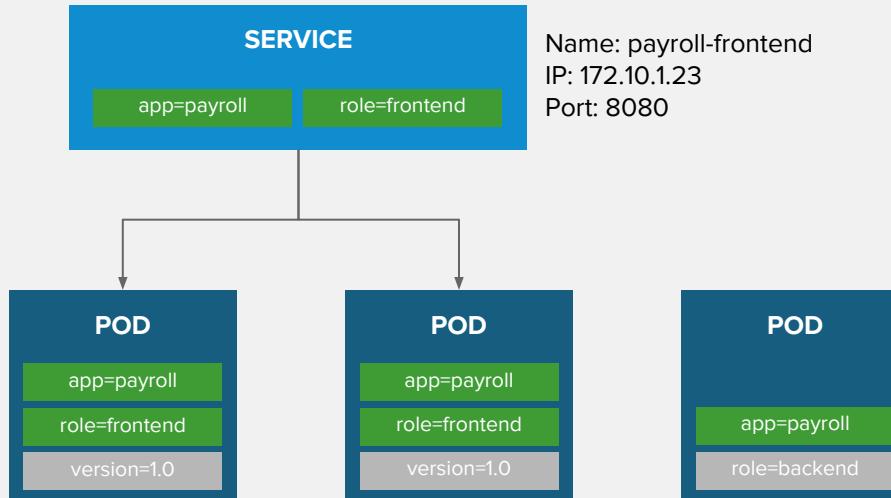
Trusted by Fortune Global 500 companies

OPENShift ARCHITECTURE

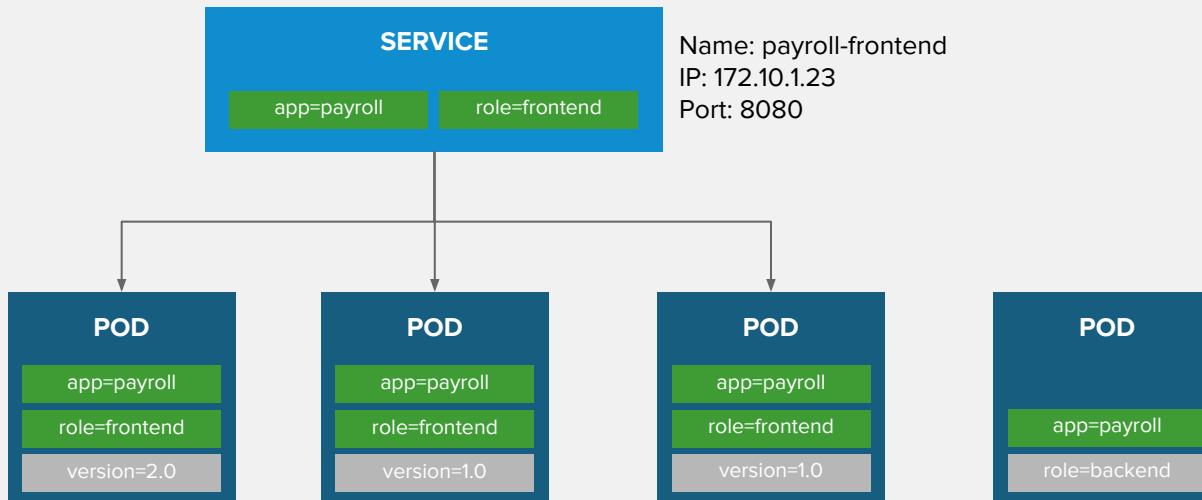


NETWORKING

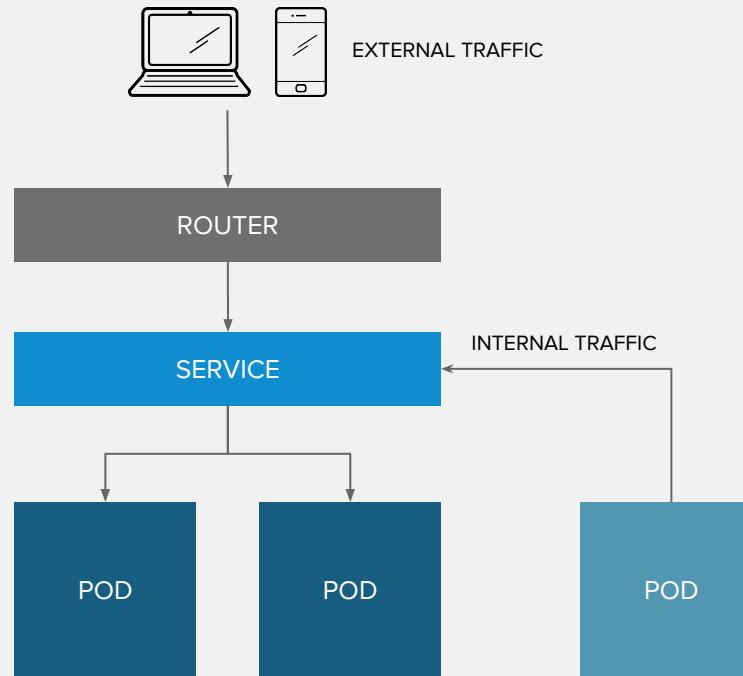
BUILT-IN SERVICE DISCOVERY INTERNAL LOAD-BALANCING



BUILT-IN SERVICE DISCOVERY INTERNAL LOAD-BALANCING

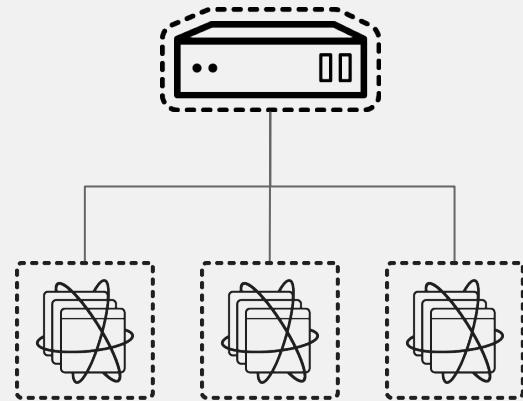


ROUTE EXPOSES SERVICES EXTERNALLY



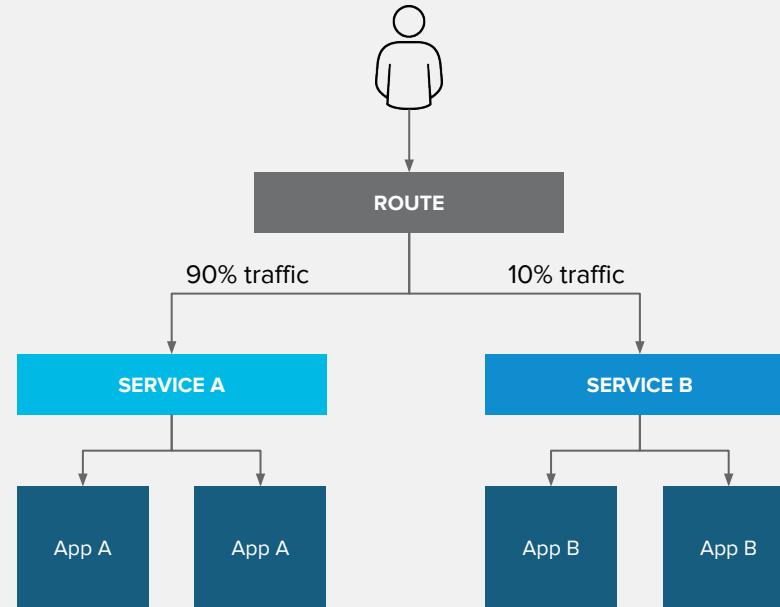
ROUTING AND EXTERNAL LOAD-BALANCING

- Pluggable routing architecture
 - HAProxy Router
 - F5 Router
- Multiple-routers with traffic sharding
- Router supported protocols
 - HTTP/HTTPS
 - WebSockets
 - TLS with SNI
- Non-standard ports via cloud load-balancers, external IP, and NodePort



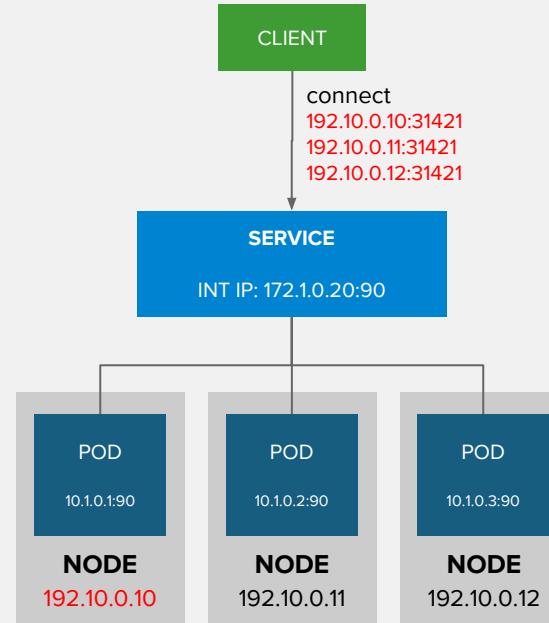
ROUTE SPLIT TRAFFIC

Split Traffic Between
Multiple Services For A/B
Testing, Blue/Green and
Canary Deployments



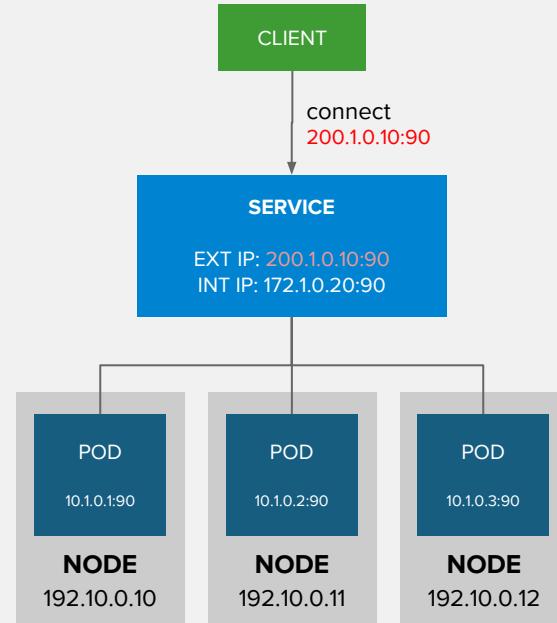
EXTERNAL TRAFFIC TO A SERVICE ON A RANDOM PORT WITH NODEPORT

- NodePort binds a service to a unique port on all the nodes
- Traffic received on any node redirects to a node with the running service
- Ports in 30K-60K range which usually differs from the service
- Firewall rules must allow traffic to all nodes on the specific port

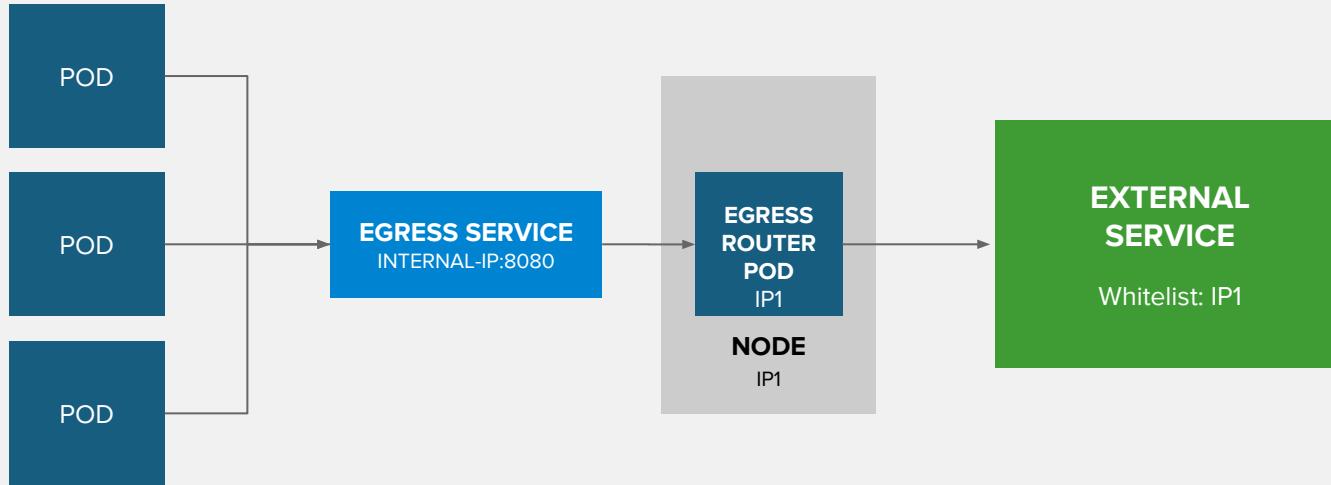


EXTERNAL TRAFFIC TO A SERVICE ON ANY PORT WITH INGRESS

- Access a service with an external IP on any TCP/UDP port, such as
 - Databases
 - Message Brokers
- Automatic IP allocation from a predefined pool using Ingress IP Self-Service
- IP failover pods provide high availability for the IP pool

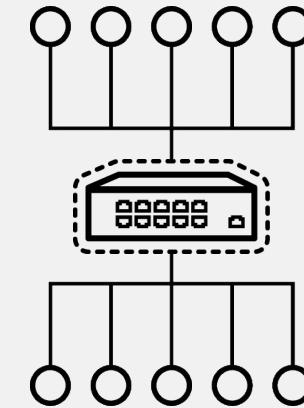


CONTROL OUTGOING TRAFFIC SOURCE IP WITH EGRESS ROUTER

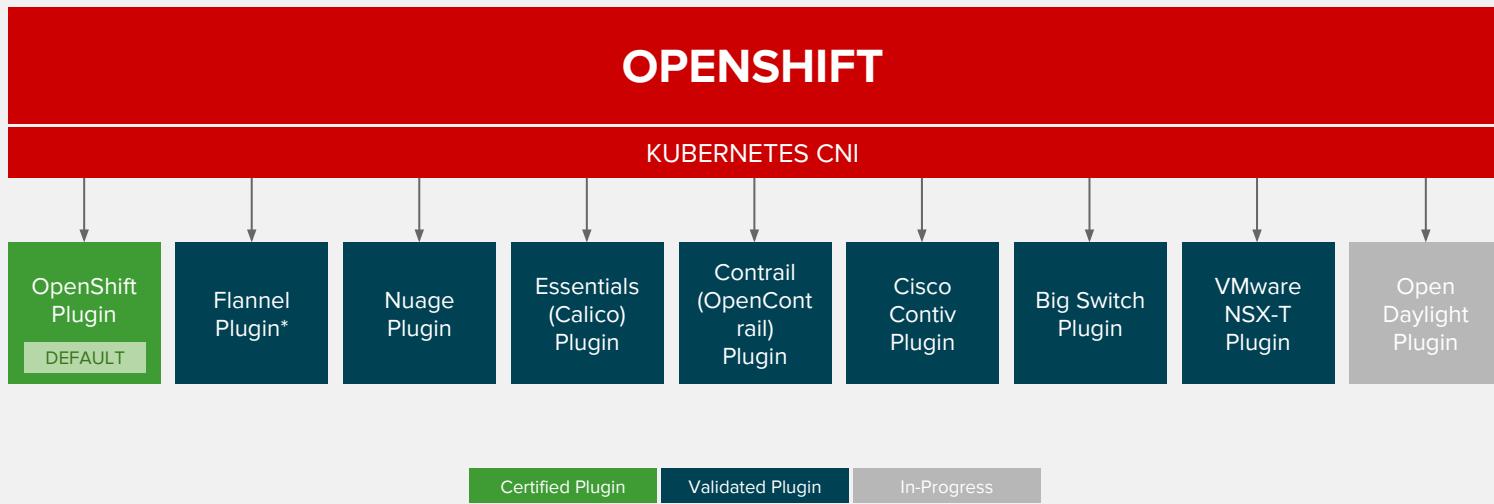


OPENShift NETWORKING

- Built-in internal DNS to reach services by name
- Split DNS is supported via SkyDNS
 - Master answers DNS queries for internal services
 - Other nameservers serve the rest of the queries
- Software Defined Networking (SDN) for a unified cluster network to enable pod-to-pod communication
- OpenShift follows the Kubernetes Container Networking Interface (CNI) plug-in model

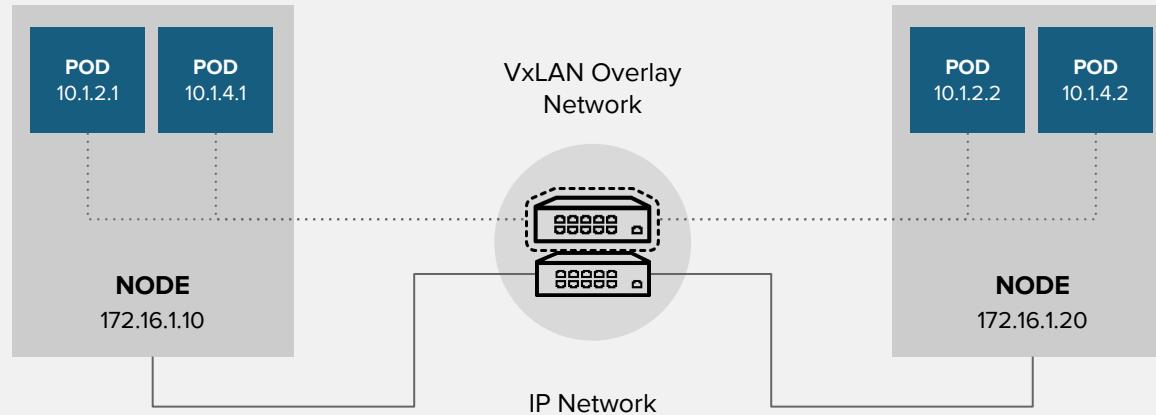


OPENShift NETWORK PLUGINS



* Flannel is minimally verified and is supported only and exactly as deployed in the OpenShift on OpenStack reference architecture

OPENShift NETWORKING



OPENShift SDN

FLAT NETWORK (Default)

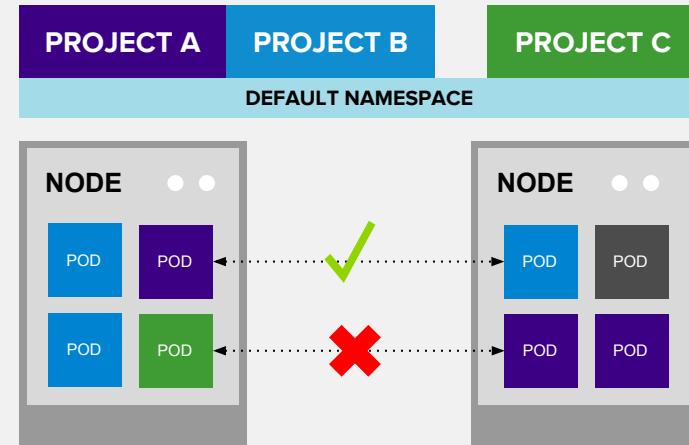
- All pods can communicate with each other across projects

MULTI-TENANT NETWORK

- Project-level network isolation
- Multicast support
- Egress network policies

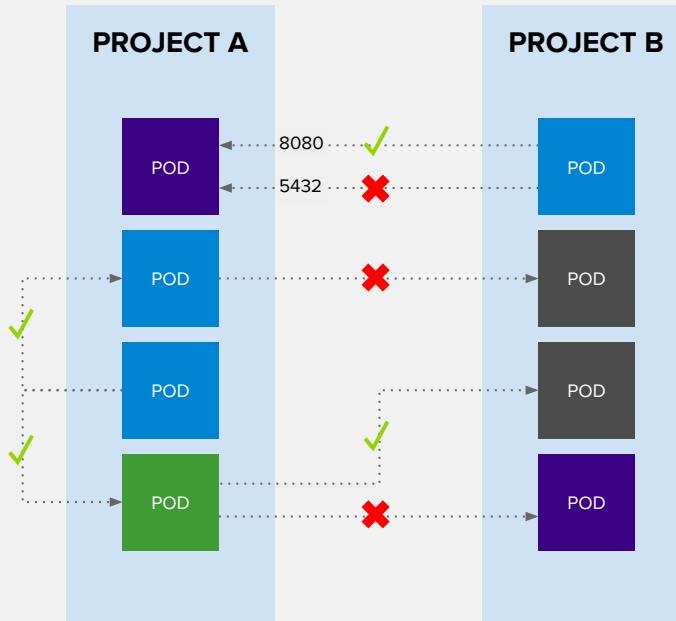
NETWORK POLICY (Tech Preview)

- Granular policy-based isolation



Multi-Tenant Network

OPENShift SDN - NETWORK POLICY



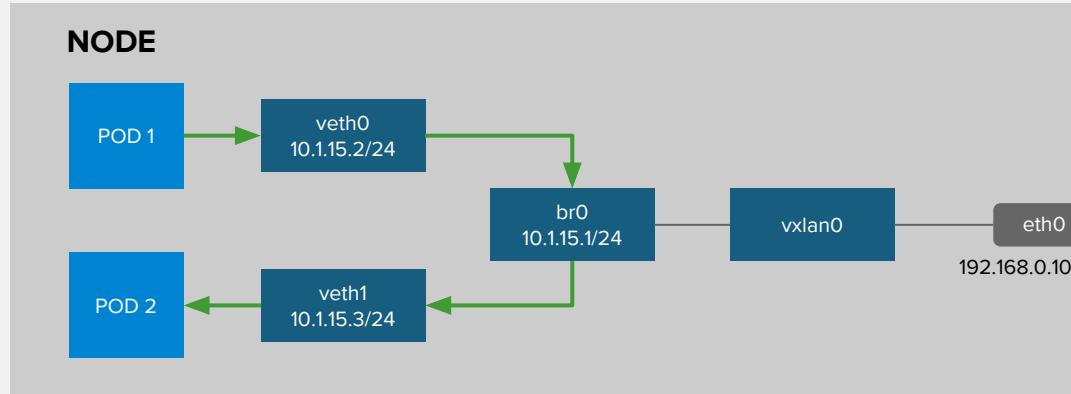
Example Policies

- Allow all traffic inside the project
- Allow traffic from green to gray
- Allow traffic to purple on 8080

```
apiVersion: extensions/v1beta1
kind: NetworkPolicy
metadata:
  name: allow-to-purple-on-8080
spec:
  podSelector:
    matchLabels:
      color: purple
  ingress:
    - ports:
        - protocol: tcp
          port: 8080
```

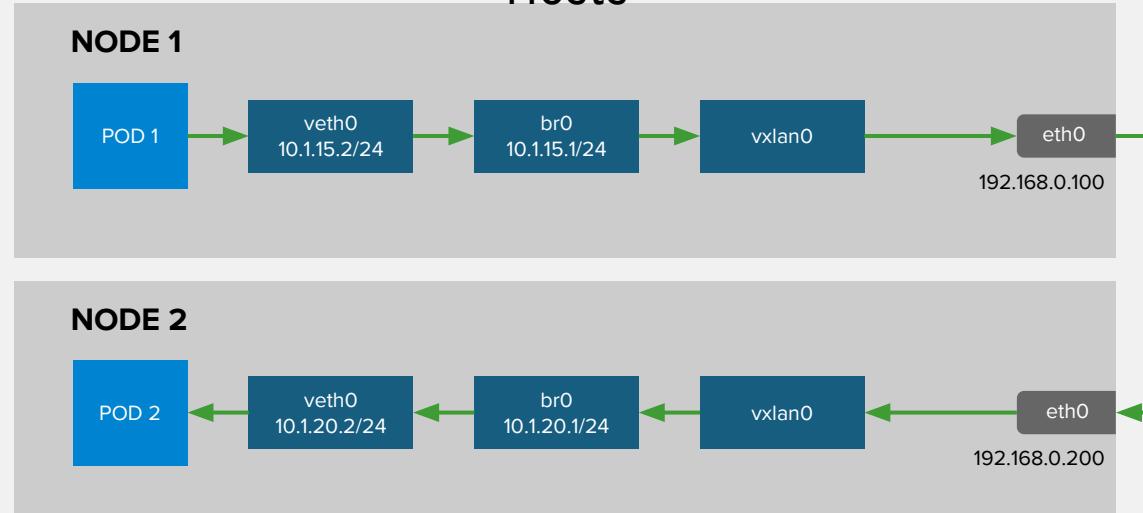
OPENShift SDN - OVS PACKET FLOW

Container to Container on the Same Host



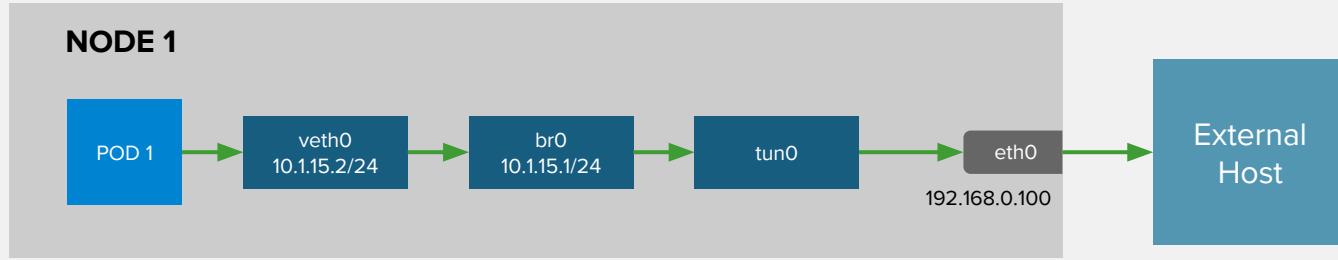
OPENShift SDN - OVS PACKET FLOW

Container to Container on the Different Hosts

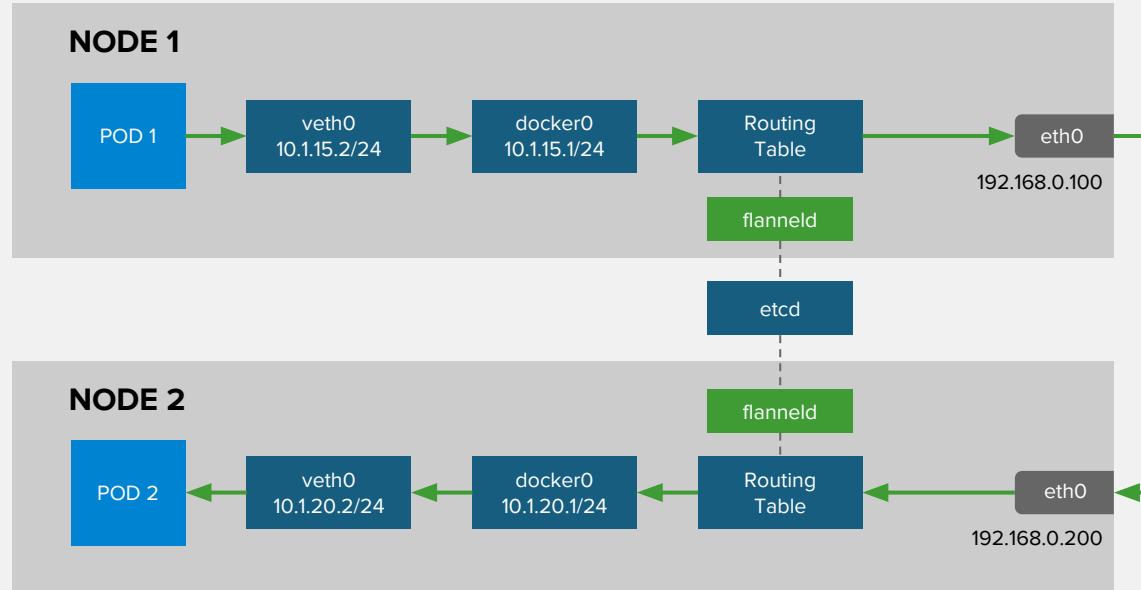


OPENShift SDN - OVS PACKET FLOW

Container Connects to External Host



OPENShift SDN WITH FLANNEL FOR OPENSTACK



Flannel is minimally verified and is supported only and exactly as deployed in the OpenShift on OpenStack reference architecture <https://access.redhat.com/articles/2743631>

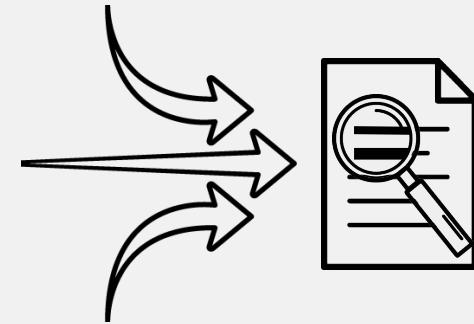


redhat.[®]

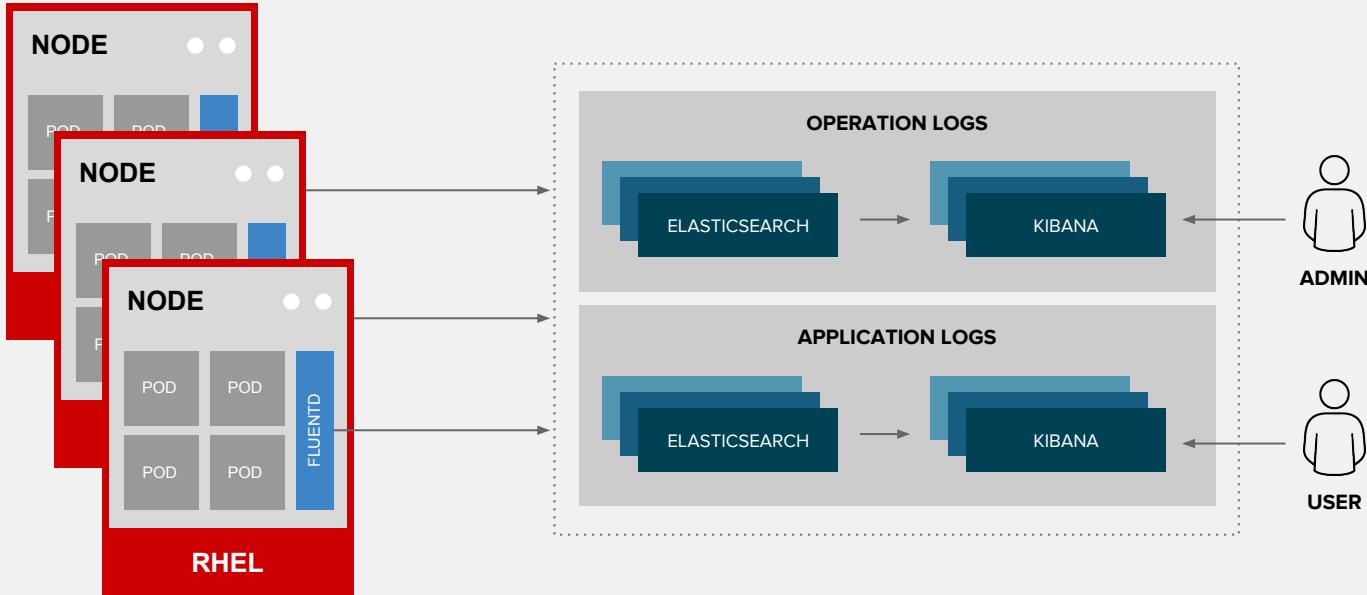
LOGGING & METRICS

CENTRAL LOG MANAGEMENT WITH EFK

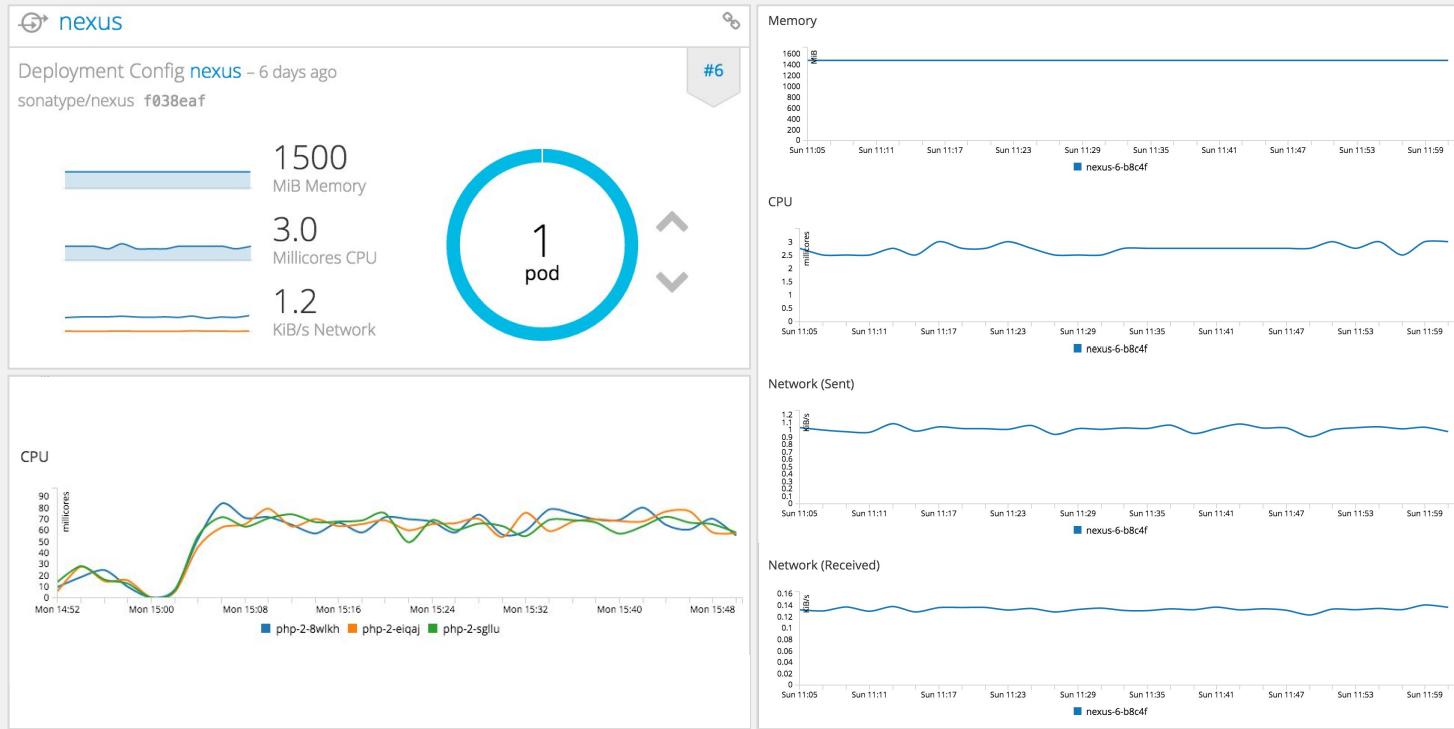
- EFK stack to aggregate logs for hosts and applications
 - **Elasticsearch:** an object store to store all logs
 - **Fluentd:** gathers logs and sends to Elasticsearch.
 - **Kibana:** A web UI for Elasticsearch.
- Access control
 - Cluster administrators can view all logs
 - Users can only view logs for their projects
- Ability to send logs elsewhere
 - External Elasticsearch, Splunk, etc



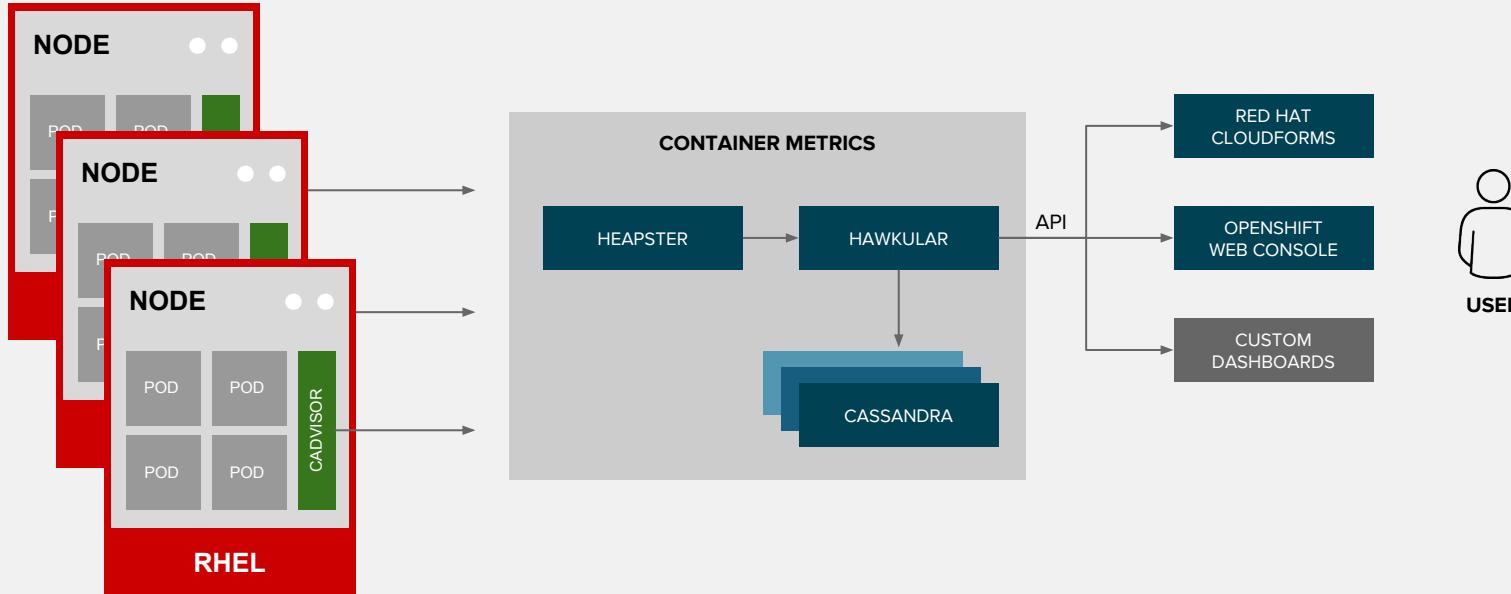
CENTRAL LOG MANAGEMENT WITH EFK



CONTAINER METRICS



CONTAINER METRICS



OPERATIONAL MANAGEMENT

TOP CHALLENGES OF RUNNING CONTAINERS AT SCALE



OPERATIONAL
EFFICIENCY



SERVICE
HEALTH



SECURITY
& COMPLIANCE



FINANCIAL
MANAGEMENT



RED HAT[®] CLOUDFORMS

Operational Management Across the Stack

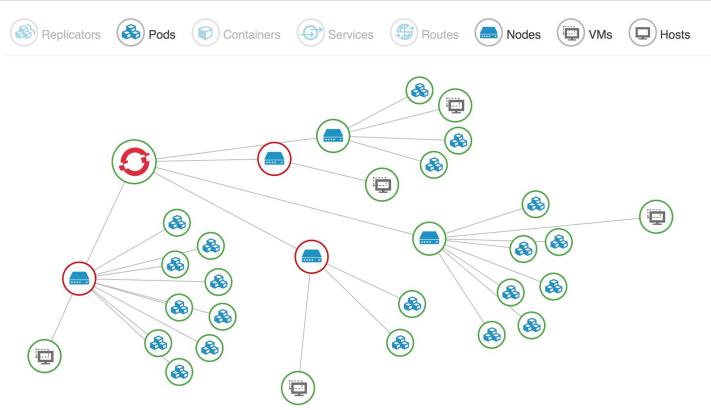
- Real-time discovery
- Visualize relationships
- Monitoring and alerts
- Vulnerability scanning
- Security compliance
- Workflow and policy
- Automation
- Chargeback

OPERATIONAL EFFICIENCY

- CloudForms continuously discovers your infrastructure in near real time.
- CloudForms discovers and visualizes relationships between infra components
- CloudForms cross references inventory across technologies.
- CloudForms offers custom automation via control policy or UI extensions



OPERATIONAL EFFICIENCY



A network diagram illustrating the relationship between pods and nodes. Nodes are represented by blue icons with a server tower, and pods are represented by green icons with a circular arrow. Lines connect individual pods to their respective nodes, showing a distributed architecture.

Projects by Number of Pods Widget	
Project Name	Number of Pods
demo-project	12
demo-project	7
default	3
openshift-infra	3
management-infra	3
default	2
cloudforms	1
openshift-infra	0
openshift	0
management-infra	0

Relationships	
Containers Provider	OpenShift Container Platform
Project	cicd
Container Services	1
Replicator	gogs-1
Containers	1
Node	ocp-node-2.lab.example.com
Underlying Instance	ocp-node-2.lab.example.com
Container Images	1

1 Providers
1

6 Nodes
26 Pods
26 Containers
43 Services
3 Registries
14 Projects

Aggregated Node Utilization

CPU: 83 Available of 84 Cores
Memory: 95 Available of 152 GB

1 Cores Used
57 GB Used

1301 Kbps
Last 30 Days

New Image Usage Trend
Images Last 30 Days

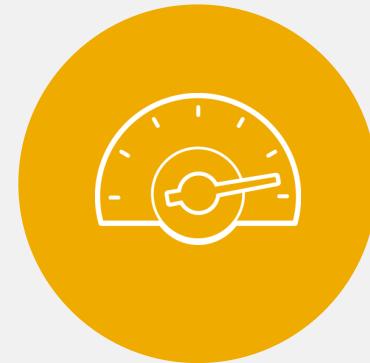
Node Utilization

CPU: > 90% (red), 80-90% (orange), 70-80% (yellow), < 70% (light blue)
Memory: > 90% (red), 80-90% (orange), 70-80% (yellow), < 70% (light blue)

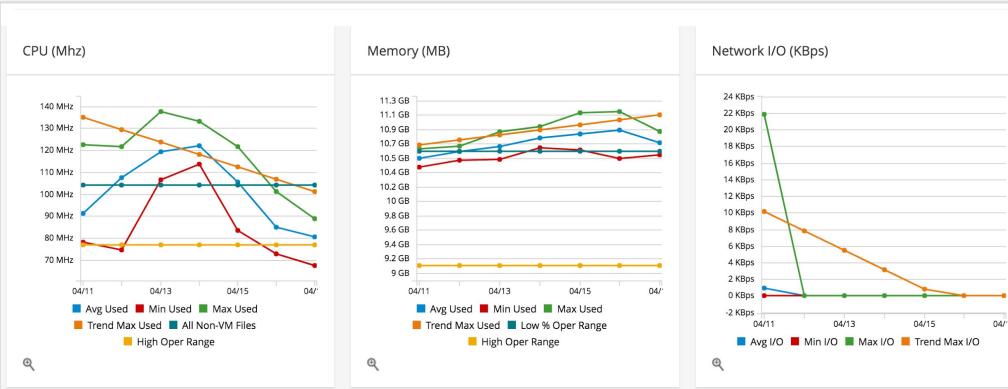
Pod Creation and Deletion Trends
Created (blue), Deleted (green) Last 30 Days

SERVICE HEALTH

- CloudForms monitors resource consumption and shows trends
- CloudForms alerts on performance thresholds or other events
- CloudForms offers right-sizing recommendations
- CloudForms enforces configuration and tracks it over time.



SERVICE HEALTH



Time Stamp	Type	Name	Event Type	Severity	Message
01/02/18 07:23: 10 UTC	Cluster / Deployment Role	Raleigh	Memory Usage	1	Memory - Peak Aggregate Used for Child VMs for Collected Intervals (MB) is projected to reach 765.6 GB (100% of Memory Max Total)
11/01/17 06:18: 52 UTC	Cluster / Deployment Role	Raleigh	Memory Usage	1	Memory - Peak Aggregate Used for Child VMs for Collected Intervals (MB) is projected to reach 689 GB (90% of Memory Max Total)
07/31/17 04:42: 25 UTC	Cluster / Deployment Role	Raleigh	Memory Usage	2	Memory - Peak Aggregate Used for Child VMs for Collected Intervals (MB) is projected to reach 574.2 GB (75% of Memory Max Total)
02/26/17 02:01: 39 UTC	Cluster / Deployment Role	Raleigh	Memory Usage	3	Memory - Peak Aggregate Used for Child VMs for Collected Intervals (MB) is projected to reach 382.8 GB (50% of Memory Max Total)

Normal Operating Ranges (up to 30 days' data)

	Max	High	Average	Low
CPU	745.90 MHz	705.74 MHz	663.99 MHz	622.23 MHz
CPU Usage	100.00%	15.36%	14.10%	12.84%
Memory	7.7 GB	7.57 GB	7.37 GB	7.18 GB
Memory Usage	65.00%	63.46%	61.78%	60.11%

Right-Sizing (Conservative - derived from Absolute Maximum)

	Current	Recommended	% Savings	Savings
Processors	4	5	-25.0%	-1
Memory	12288 MB	7988 MB	35.0%	4300 MB

Right-Sizing (Moderate - derived from High NORM)

	Current	Recommended	% Savings	Savings
Processors	4	1	75.0%	3
Memory	12288 MB	7800 MB	36.5%	4488 MB

Right-Sizing (Aggressive - derived from Average NORM)

	Current	Recommended	% Savings	Savings
Processors	4	1	75.0%	3
Memory	12288 MB	7596 MB	38.2%	4692 MB

SECURITY & COMPLIANCE

- CloudForms finds and marks nodes non-compliant with policy.
- CloudForms allows reporting on container provenance.
- CloudForms scans container images using OpenSCAP.
- CloudForms tracks genealogy between images and containers.



SECURITY & COMPLIANCE

Compliance			
Status	Non-Compliant as of 5 Days Ago		
History	Available		
Configuration			
Packages	 528		
OpenSCAP Results	 431		
OpenSCAP HTML	Available		
Last scan	Tue, 28 Mar 2017 11:05:54 +0000		
OpenSCAP Failed Rules Summary			
Medium	1		
High	3		
Name	Result	Severity	
 xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20170386	Fail	High	
 xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20170372	Fail	High	
 xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20170294	Fail	High	
 xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20170286	Fail	Medium	
 xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20140685	Pass	High	
 xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20140686	Pass	High	
 xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20140679	Pass	High	
 xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20140703	Pass	Medium	
 xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20140684	Pass	High	
 xccdf_com.redhat.rhsa_rule_oval-com.redhat.rhsa-def-20140704	Pass	Medium	

Compliance and Scoring

The target system did not satisfy the conditions of 4 rules! Please review rule results and consider applying remediation.

Rule results

427 passed 4 failed

Severity of failed rules

1 medium 3 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf.scoring:default	99.071922	100.000000	99.07%

Rule Overview

pass fail
 fixed error informational notchecked notselected unknown notapplicable

Search through XCCDF rules

Group rules by: Default ▾

Title	Severity	Result
Automatically generated XCCDF from OVAL file: com.redhat.rhsa-RHEL7.xml	4x fail	
RHSA-2017:0286: openssl security update (Moderate)	medium	fail
RHSA-2017:0294: kernel security update (Important)	high	fail
RHSA-2017:0372: kernel-aarch64 security and bug fix update (Important)	high	fail
RHSA-2017:0386: kernel security, bug fix, and enhancement update (Important)	high	fail

FINANCIAL MANAGEMENT

- Define cost models for infrastructure and understand your cost.
- Rate schedules per platform and per tenant with multi-tiered and multi-currency support
- CloudForms shows top users for CPU, memory, as well as cost.
- Chargeback/showback to projects based on container utilization.



FINANCIAL MANAGEMENT

Top CPU Consumers (Last Hour)			
VM Name	CPU Usage	Allocated vCPUs	VM Vendor
overcloud1-telus	21.4%	8	vmware
manageiq-euwe-2	18.3%	4	redhat
manageiq-euwe-3	14.0%	4	redhat
Lenovo XClarity Administrator - Do not delete	9.1%	2	vmware
vcenter6	8.0%	4	vmware

Updated December 21, 2016 20:17 | Next January 11, 2017 23:45

Top Memory Consumers (last hour)		
VM Name	Memory Usage	Allocated Memory
CF41_DB	100.0%	16 GB
CF42_UI2	97.7%	8 GB
CF42_UI1	97.7%	8 GB
manageiq-euwe-3	97.6%	8 GB
CF42_google1	97.0%	8 GB

Updated January 11, 2017 23:49 | Next January 12, 2017 00:05

Saved Report "ChargeBack by Project - Tue, 18 Apr 2017 17:59:28 +0000"

Date Range	Project Name	Project Uid	Cpu Cores Used Cost	Memory Used Cost	Total Cost
04/17/2017	cicd	b8f35aeee974-11e6-89d9-fa163ec3f31d	\$24.00	\$30.33	\$66.34
04/17/2017	default	4c767b2b-df4d-11e6-8850-fa163ec3f31d	\$24.00	\$4.90	\$40.90
04/17/2017	ifixied	acc6113d-ed77-11e6-8c6a-fa163ec3f31d	\$24.00	\$28.77	\$64.77
04/17/2017	jritenour-demo	47ee9d2a-efae-11e6-8c6a-fa163ec3f31d	\$24.00	\$28.80	\$64.80
04/17/2017	mlbparks	4666e252-e296-11e6-8a49-fa163ec3f31d	\$24.00	\$406.96	\$442.96
04/17/2017	openshift-infra	4e37af93-df4d-11e6-8850-fa163ec3f31d	\$24.06	\$992.75	\$1,290.78
04/17/2017	stage	b771432a-e974-11e6-89d9-fa163ec3f31d	\$24.00	\$491.89	\$527.89
Totals:					
			\$168.07	\$1,984.40	\$2,498.43
All Rows					
			\$168.07	\$1,984.40	\$2,498.43



redhat.[®]



redhat.[®]

LINUX CONTAINERS

WHAT ARE CONTAINERS?

It Depends Who You Ask

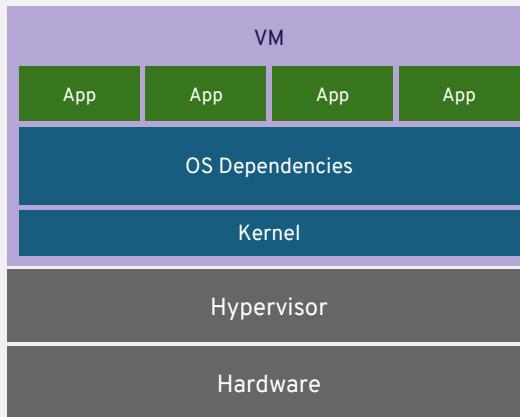
INFRASTRUCTURE

APPLICATIONS

- 
- Application processes on a shared kernel
 - Simpler, lighter, and denser than VMs
 - Portable across different environments
 - Package apps with all dependencies
 - Deploy to any environment in seconds
 - Easily accessed and shared

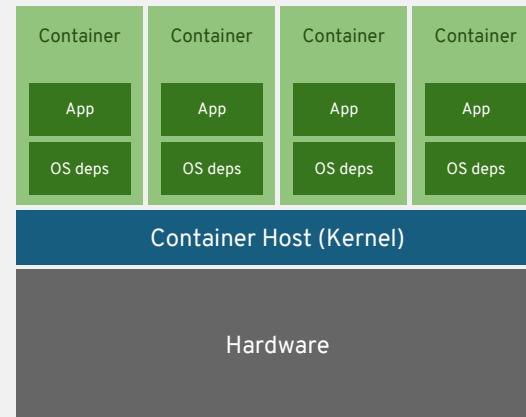
VIRTUAL MACHINES AND CONTAINERS

VIRTUAL MACHINES



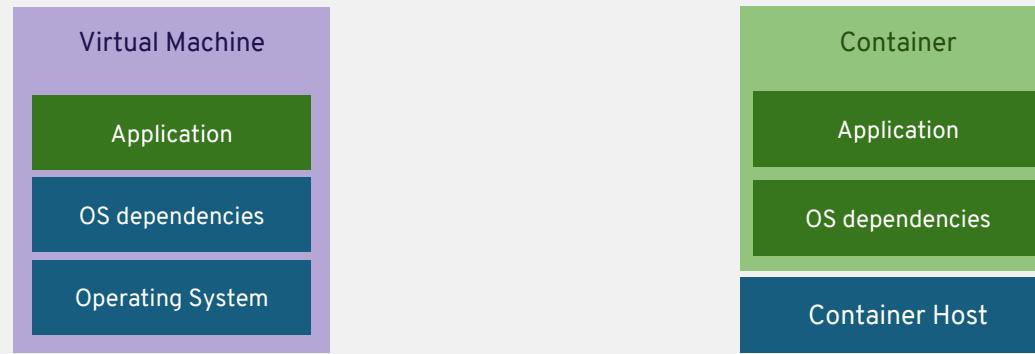
virtual machines are isolated
apps are not

CONTAINERS



containers are isolated
so are the apps

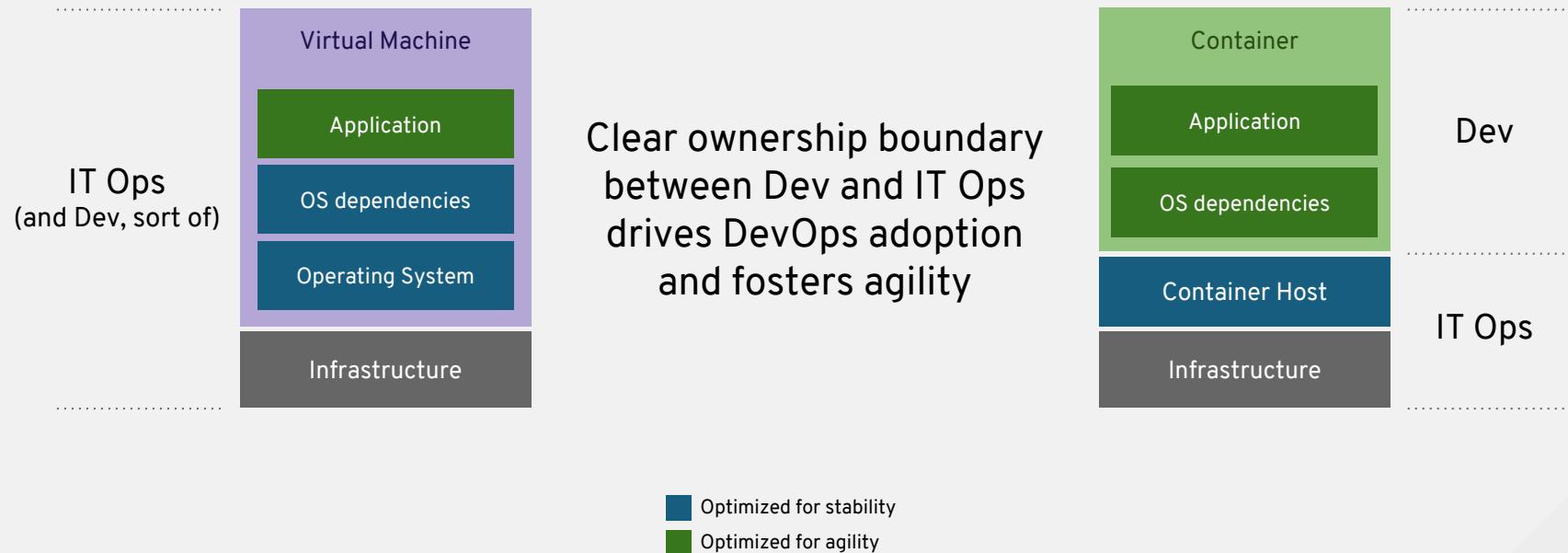
VIRTUAL MACHINES AND CONTAINERS



- + VM Isolation
- Complete OS
- Static Compute
- Static Memory
- High Resource Usage

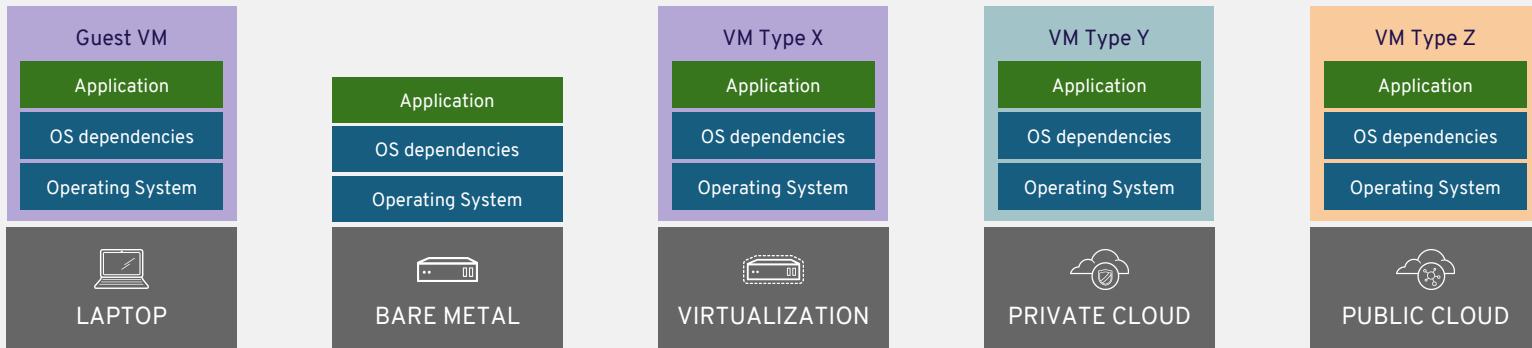
- + Container Isolation
- + Shared Kernel
- + Burstable Compute
- + Burstable Memory
- + Low Resource Usage

VIRTUAL MACHINES AND CONTAINERS



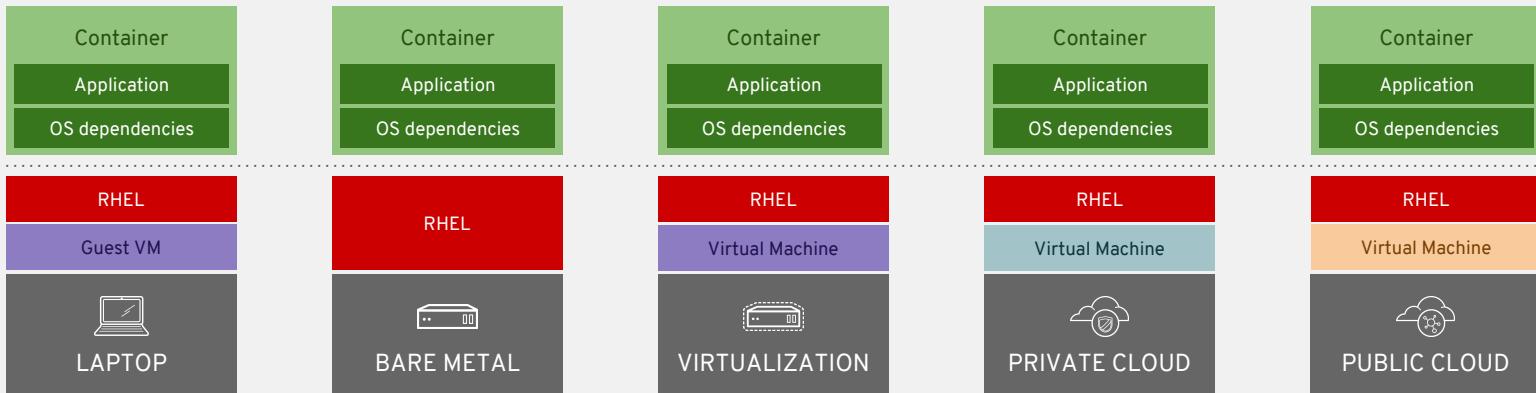
APPLICATION PORTABILITY WITH VM

Virtual machines are **NOT** portable across hypervisor and
do **NOT** provide portable packaging for applications

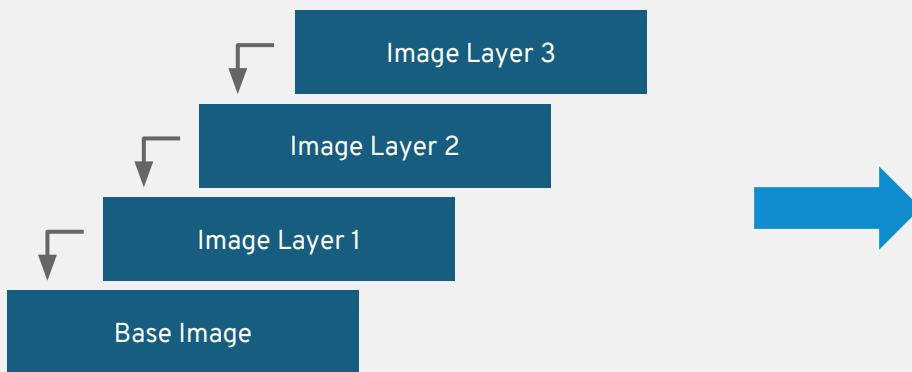


APPLICATION PORTABILITY WITH CONTAINERS

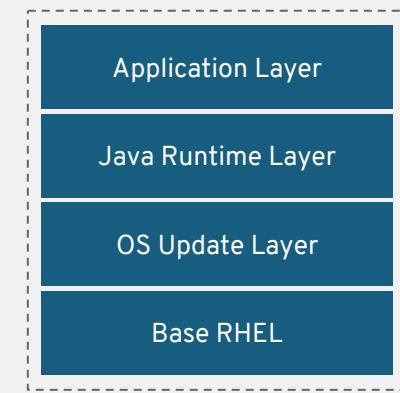
RHEL Containers + RHEL Host = Guaranteed Portability
Across Any Infrastructure



RAPID SECURITY PATCHING USING CONTAINER IMAGE LAYERING



Container Image Layers



Example Container Image



cri-o

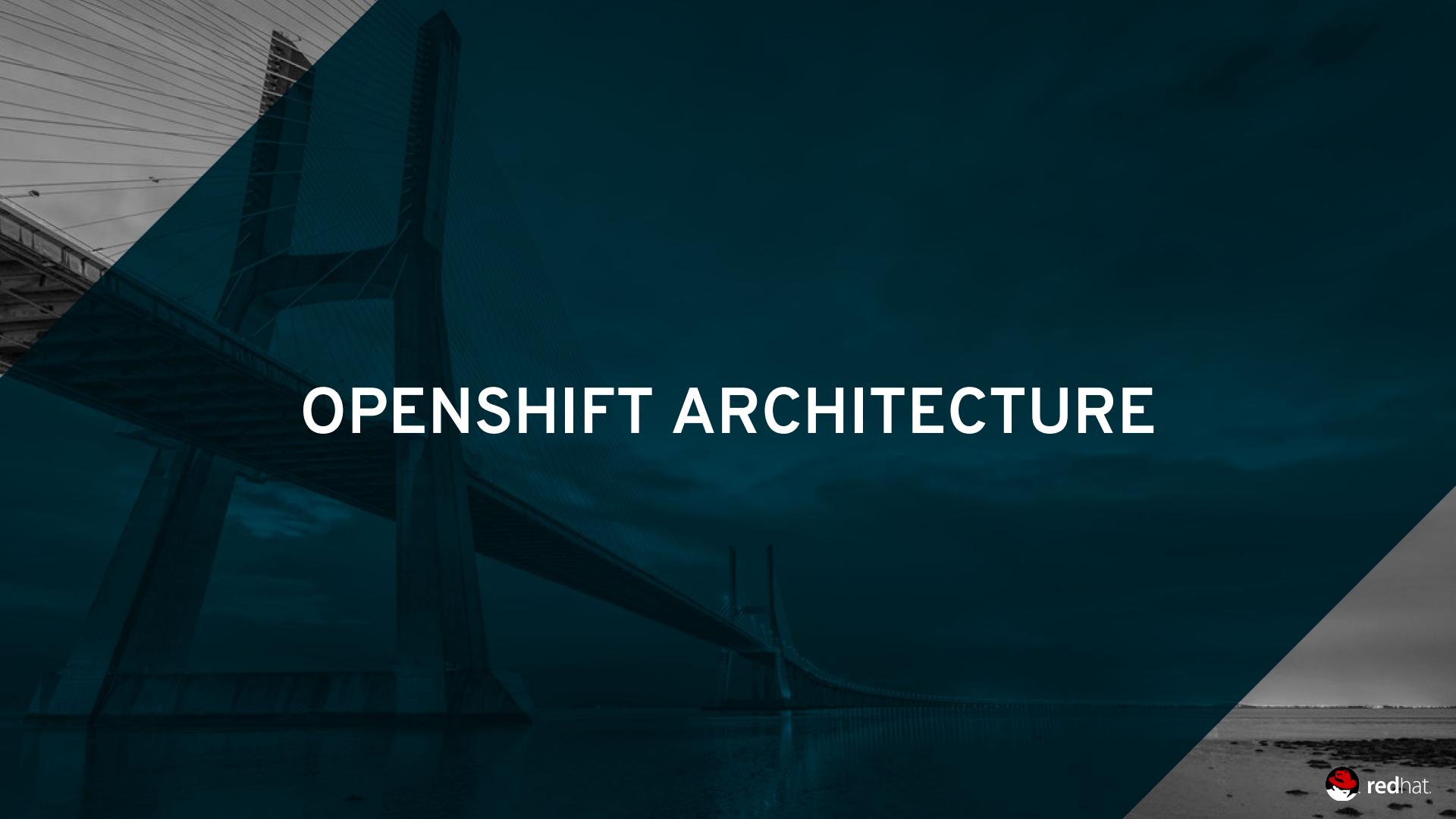
A lightweight, OCI-compliant container runtime

Optimized for
Kubernetes

Any OCI-compliant
container from any
OCI registry
(including docker)

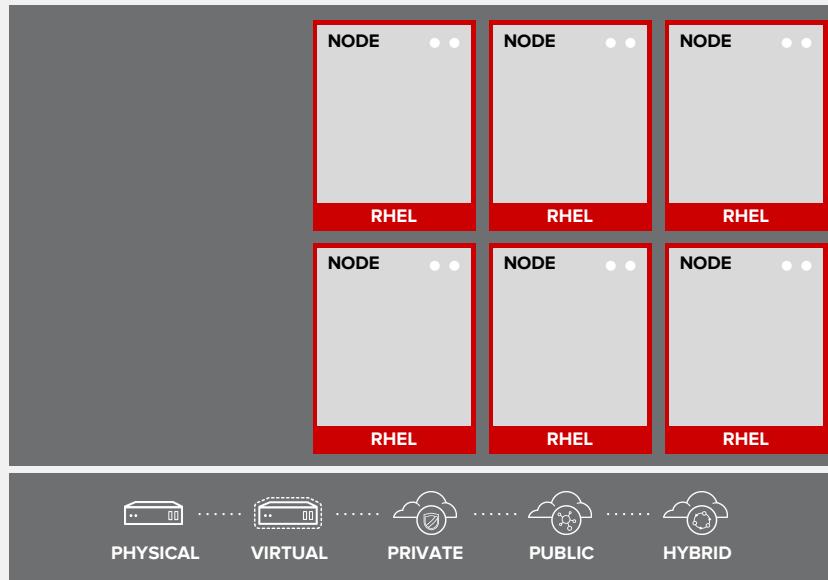
Improve Security and
Performance at scale

Available in OpenShift Online (soon)
Tech Preview in OCP 3.7, GA in OCP 3.8

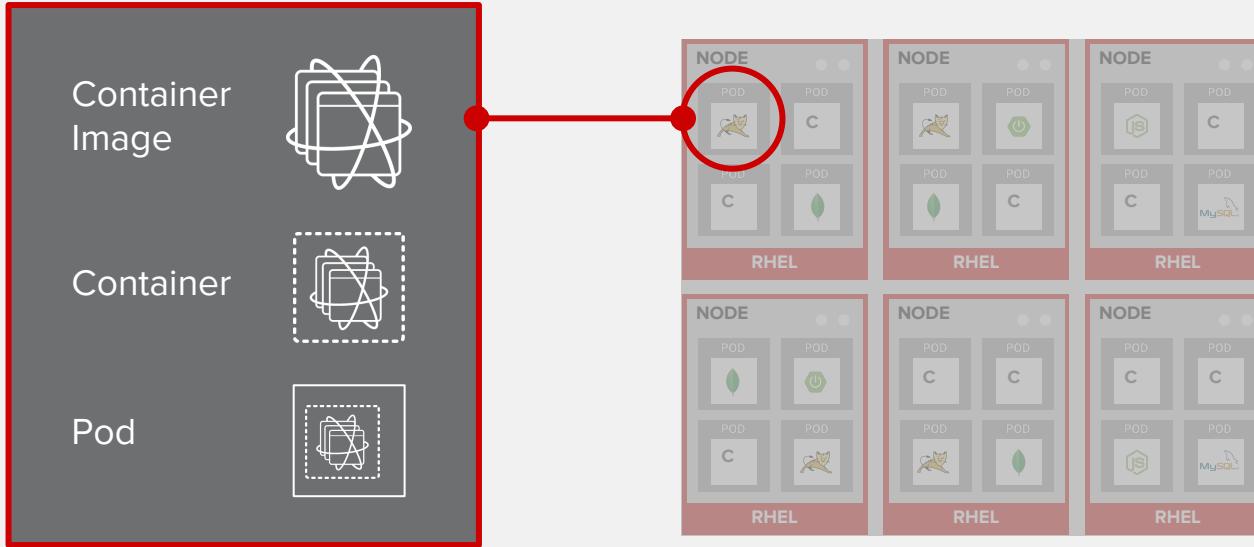


OPENSHIFT ARCHITECTURE

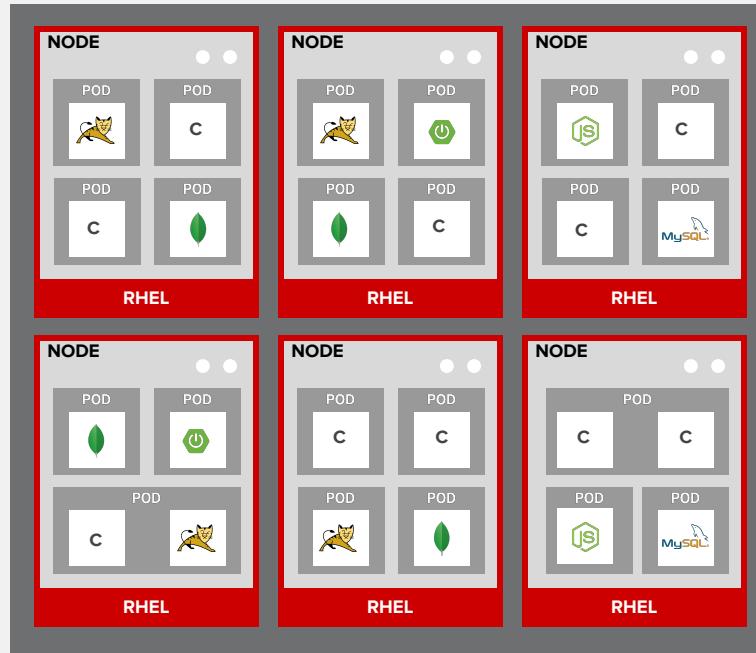
NODES RHEL INSTANCES WHERE APPS RUN



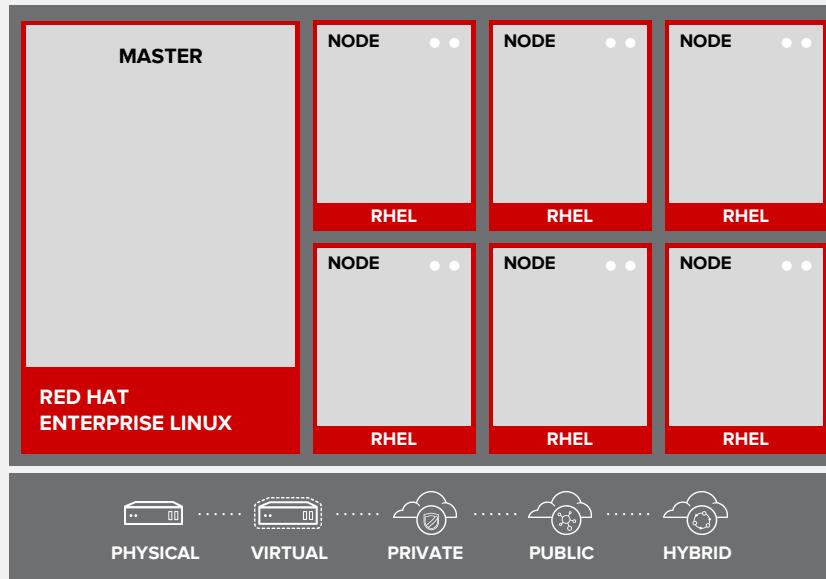
APPS RUN IN CONTAINERS



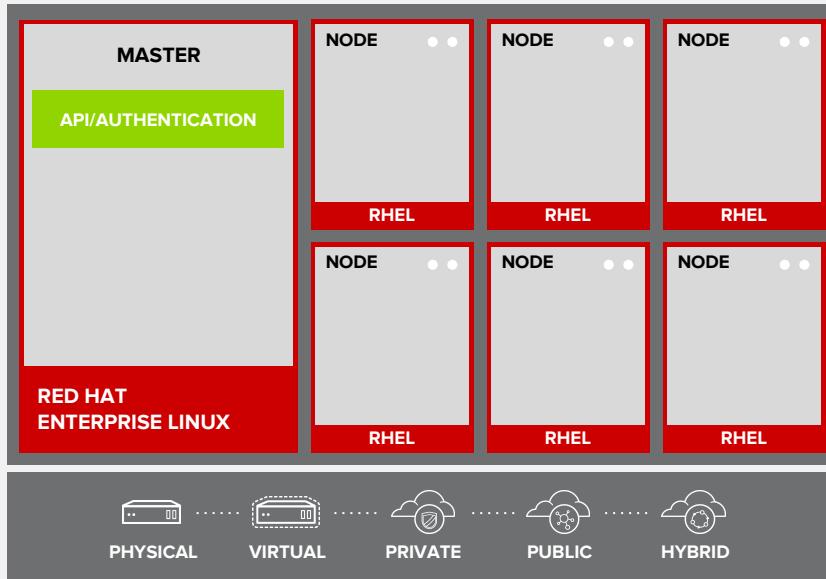
PODS ARE THE UNIT OF ORCHESTRATION



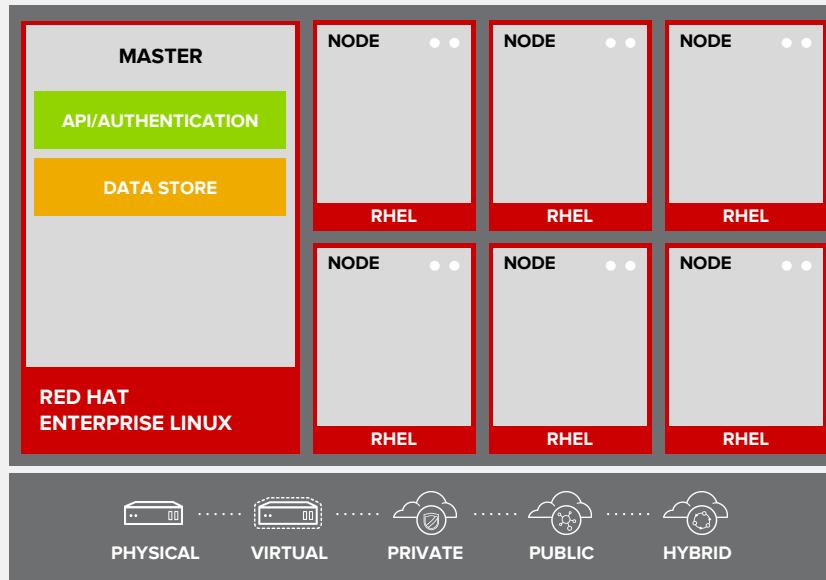
MASTERS ARE THE CONTROL PLANE



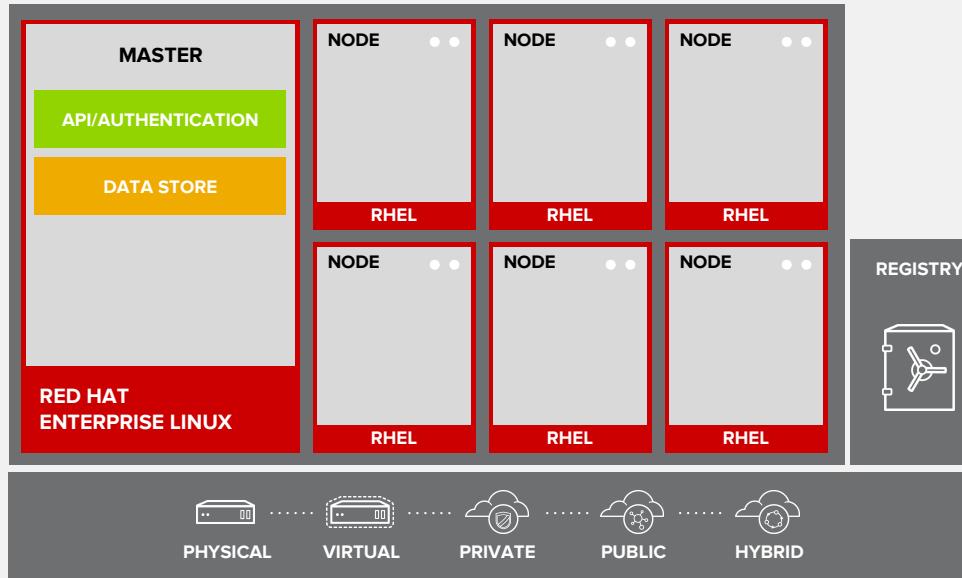
API AND AUTHENTICATION



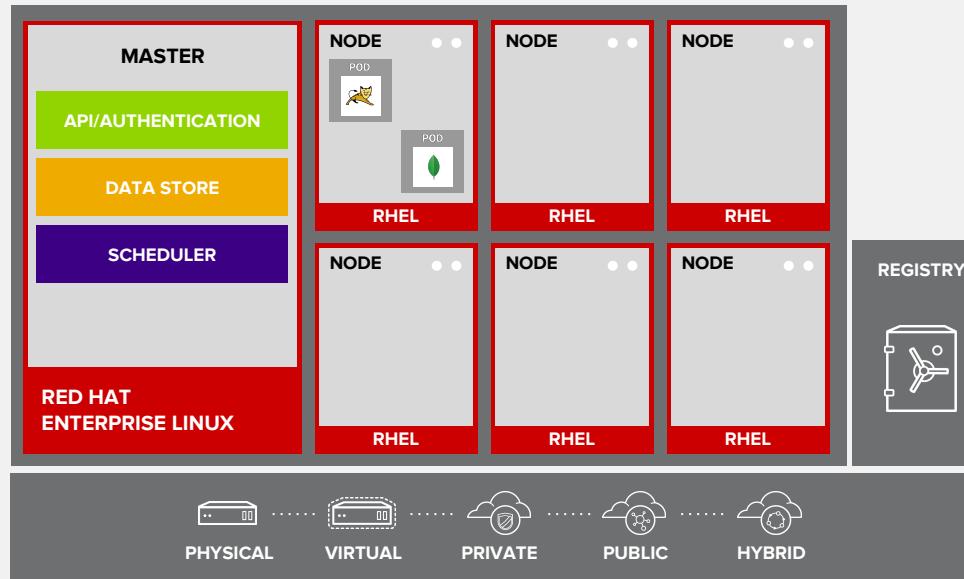
DESIRED AND CURRENT STATE



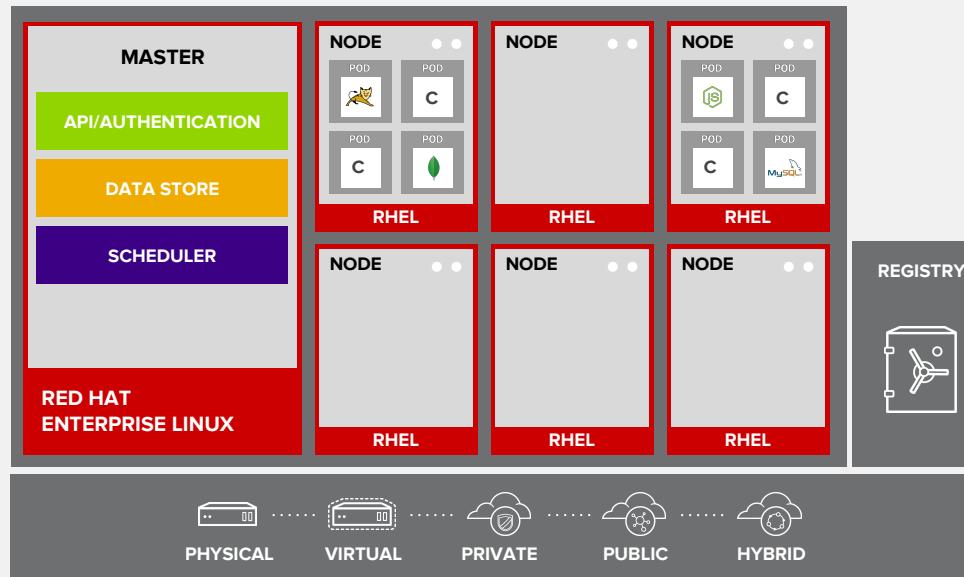
INTEGRATED CONTAINER REGISTRY



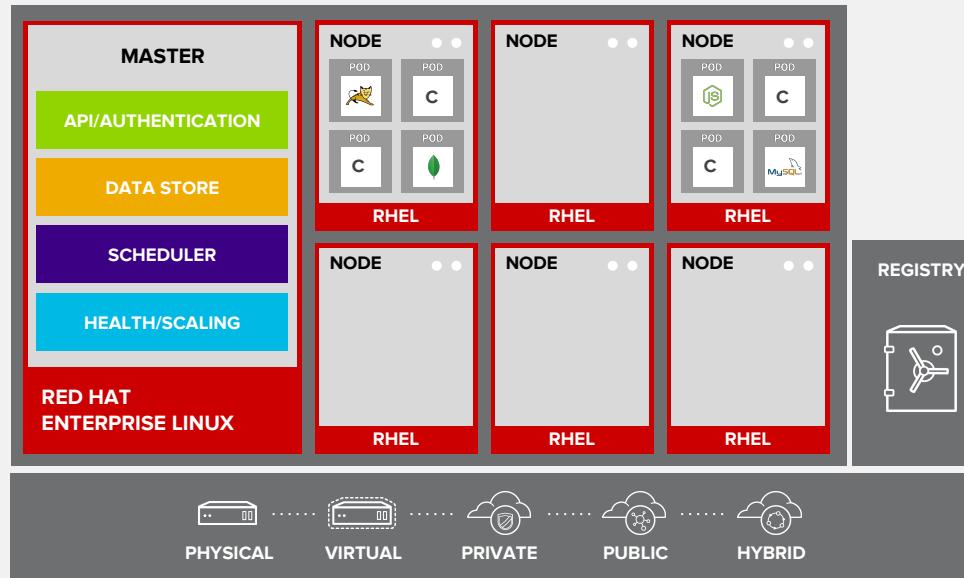
ORCHESTRATION AND SCHEDULING



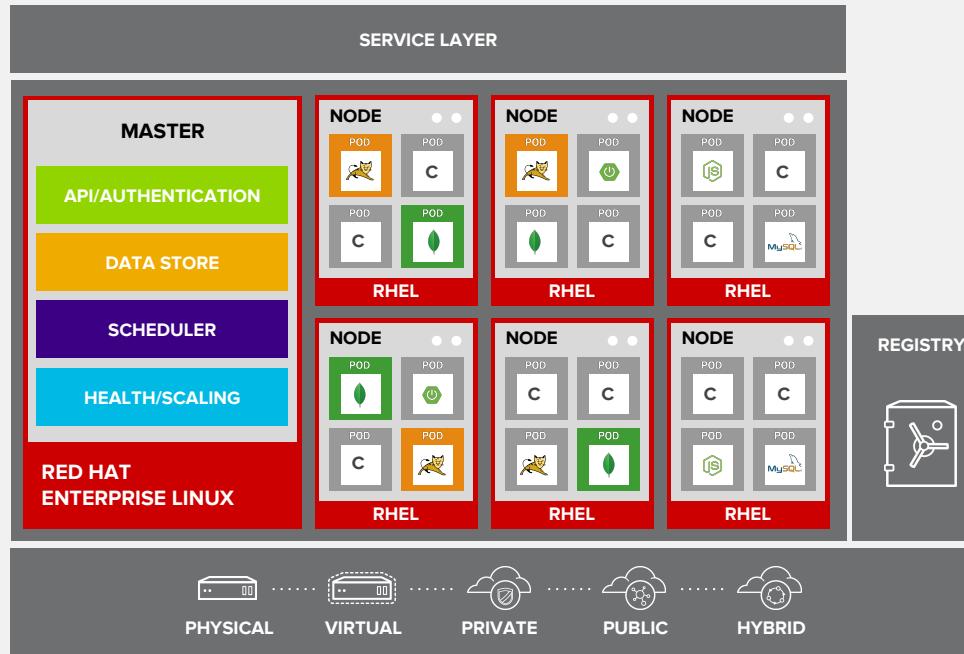
PLACEMENT BY POLICY



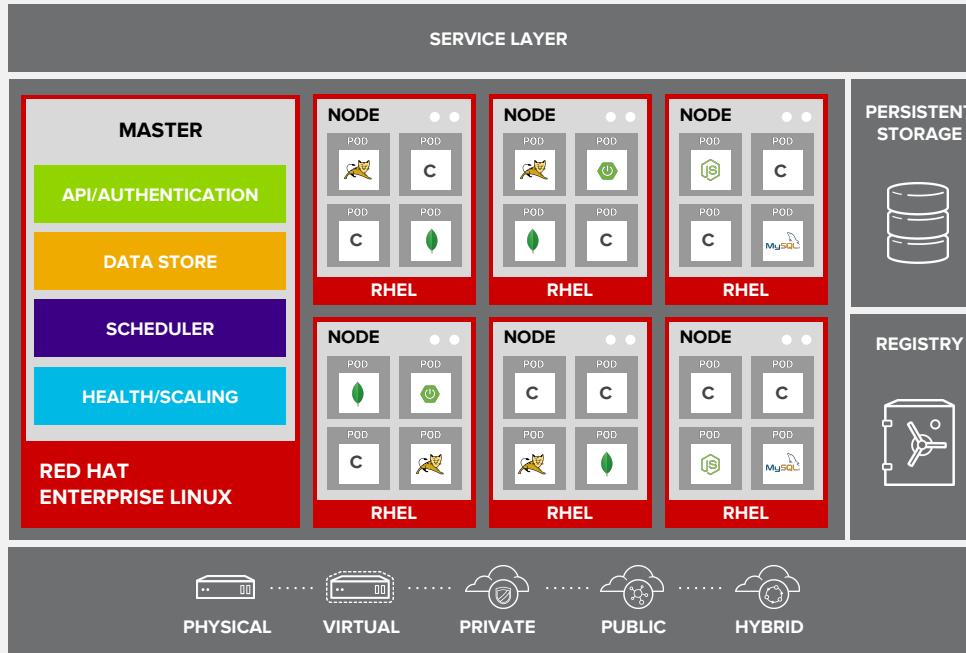
AUTOSCALING PODS



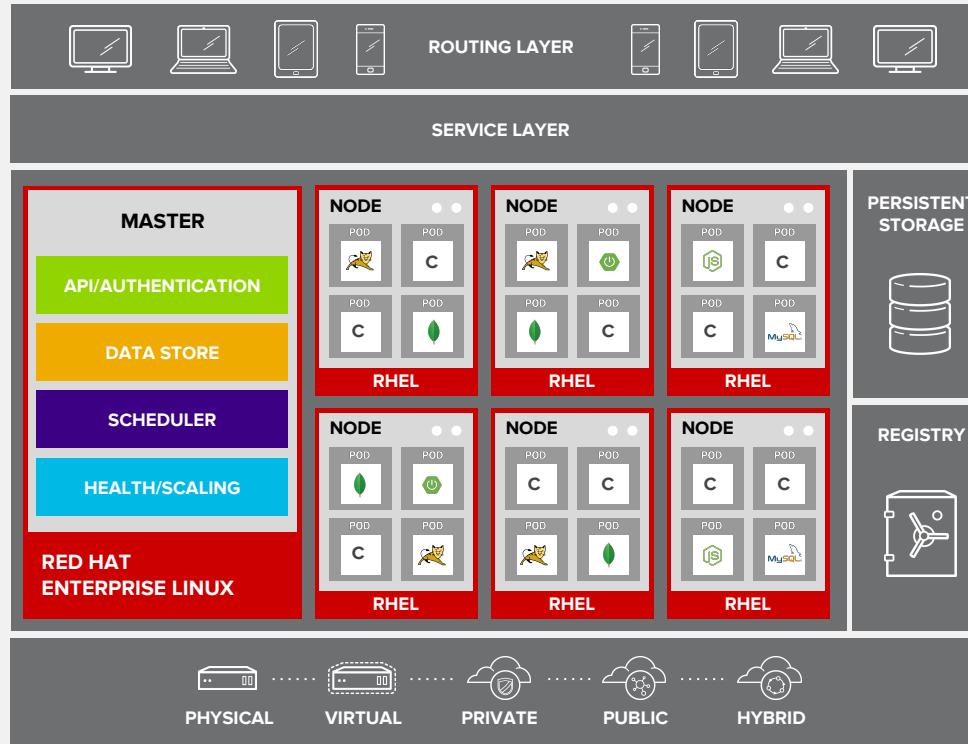
SERVICE DISCOVERY



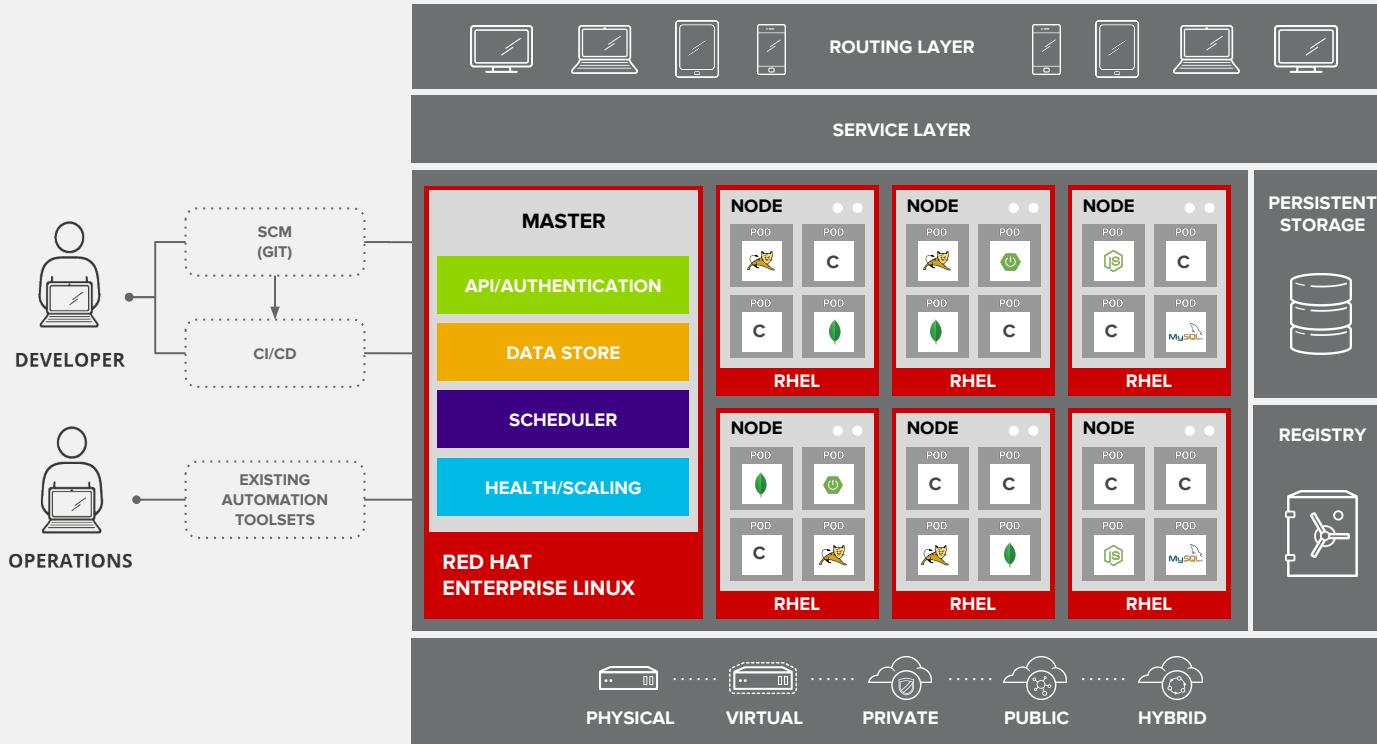
PERSISTENT DATA IN CONTAINERS



ROUTING AND LOAD-BALANCING



ACCESS VIA WEB, CLI, IDE AND API



SECURITY

TEN LAYERS OF CONTAINER SECURITY

Container Host & Multi-tenancy

Federated Clusters

Container Platform

API Management

Network Isolation

Deploying Container

Container Registry

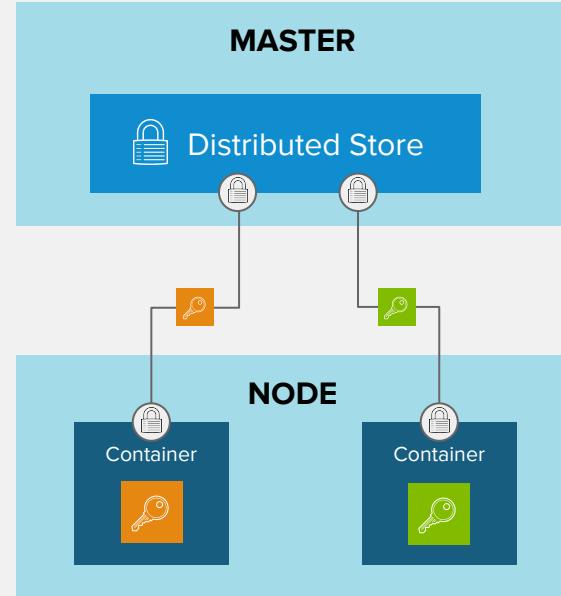
Container Content

Storage

Building Containers

SECRET MANAGEMENT

- Secure mechanism for holding sensitive data e.g.
 - Passwords and credentials
 - SSH Keys
 - Certificates
- Secrets are made available as
 - Environment variables
 - Volume mounts
 - Interaction with external systems
- Encrypted in transit
- Never rest on the nodes



REFERENCE ARCHITECTURES

REFERENCE ARCHITECTURES

[OpenShift on VMware vCenter](#)

[OpenShift on Red Hat OpenStack Platform](#)

[OpenShift on Amazon Web Services](#)

[OpenShift on Google Cloud Platform](#)

[OpenShift on Microsoft Azure](#)

[OpenShift on Red Hat Virtualization](#)

[OpenShift on HPE Servers with Ansible Tower](#)

[OpenShift on VMware vCenter 6 with Gluster](#)

[Deploying an OpenShift Distributed Architecture](#)

[OpenShift Architecture and Deployment Guide](#)

[OpenShift Scaling, Performance, and Capacity Planning](#)

[Application Release Strategies with OpenShift](#)

[Building Polyglot Microservices on OpenShift](#)

[Building JBoss EAP 6 Microservices on OpenShift](#)

[Building JBoss EAP 7 Microservices on OpenShift](#)

[Business Process Management with JBoss BPM Server on OpenShift](#)

[Build and Deployment of Java Applications on OpenShift](#)

[Building Microservices on OpenShift with Fuse Integration...](#)

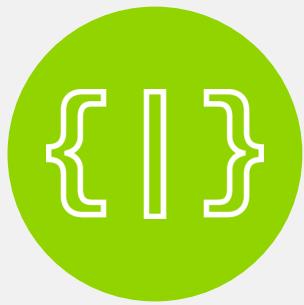
[JFrog Artifactory on OpenShift Container Platform](#)

[Spring Boot Microservices on Red Hat OpenShift](#)

[API Management with Red Hat 3scale on OpenShift](#)

BUILD AND DEPLOY CONTAINER IMAGES

BUILD AND DEPLOY CONTAINER IMAGES



DEPLOY YOUR
SOURCE CODE

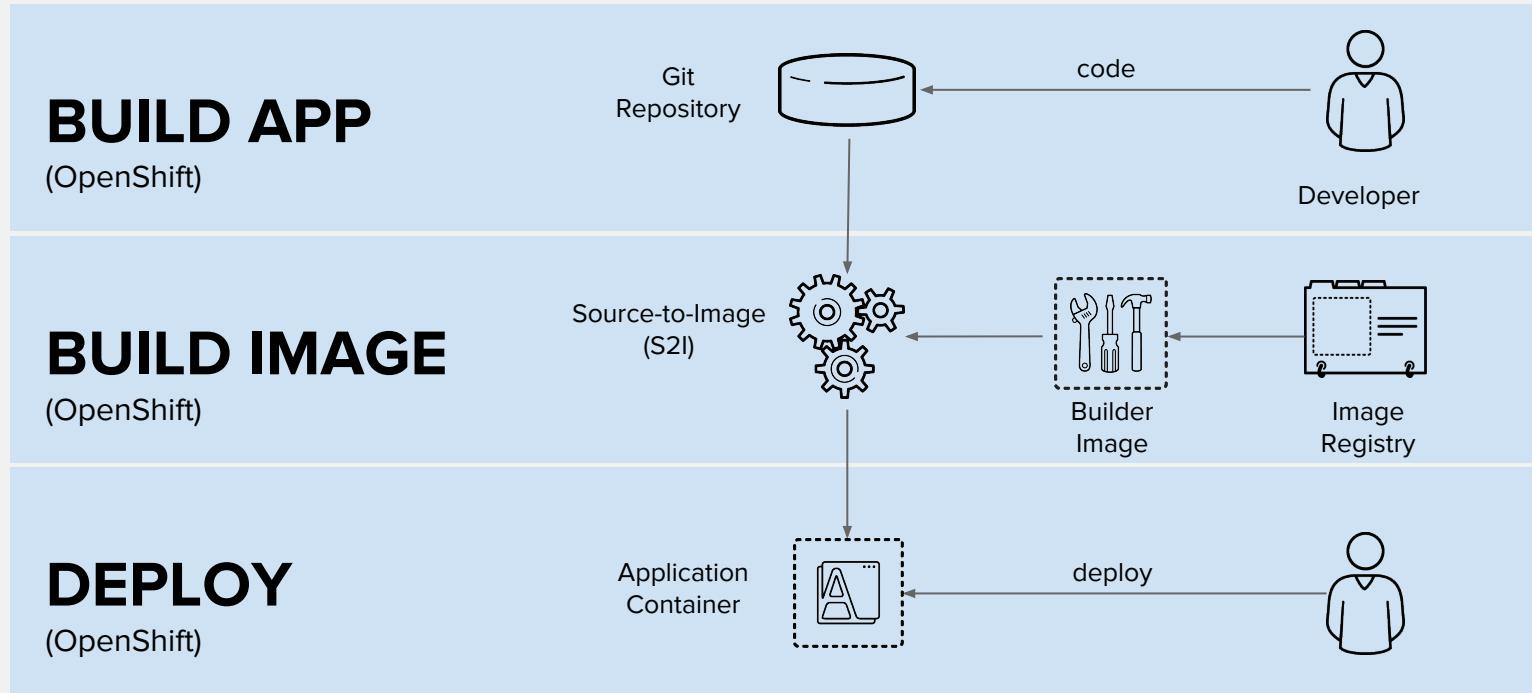


DEPLOY YOUR
APP BINARY

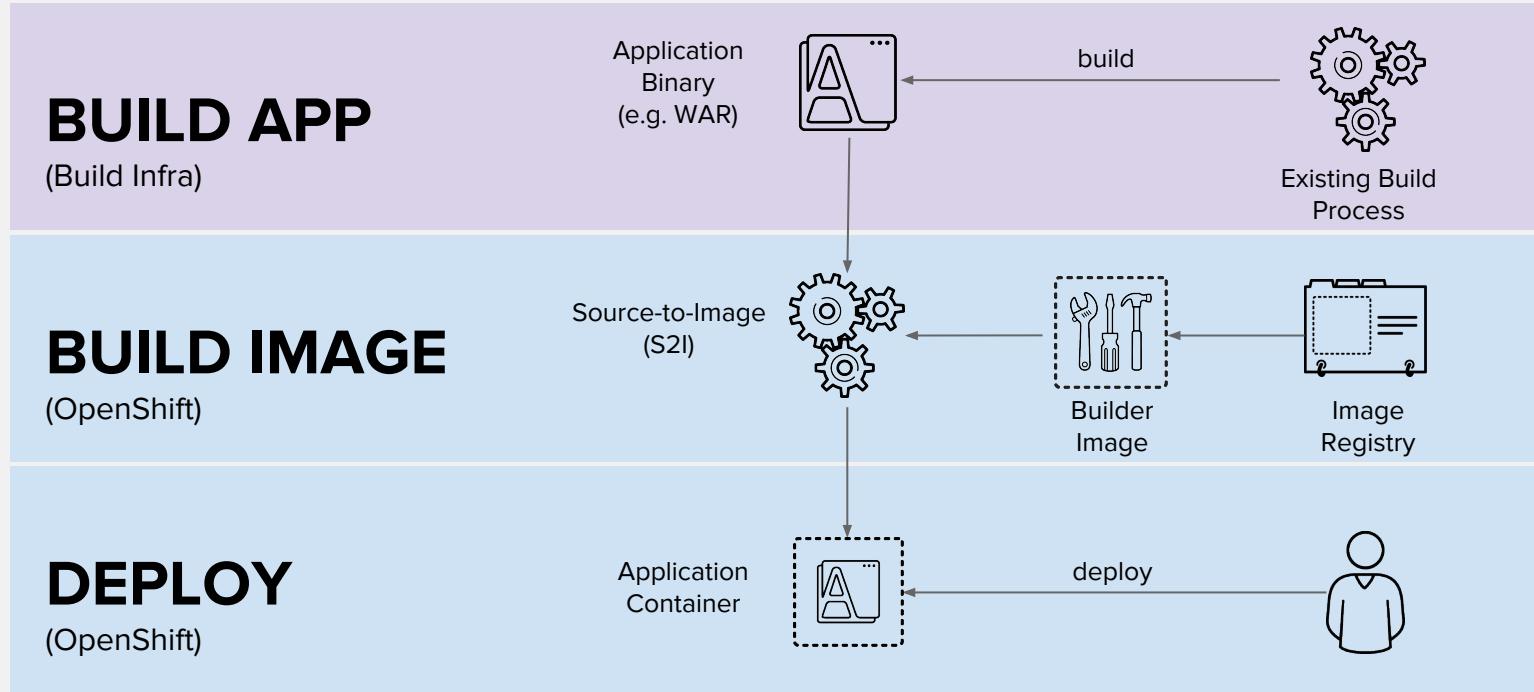


DEPLOY YOUR
CONTAINER IMAGE

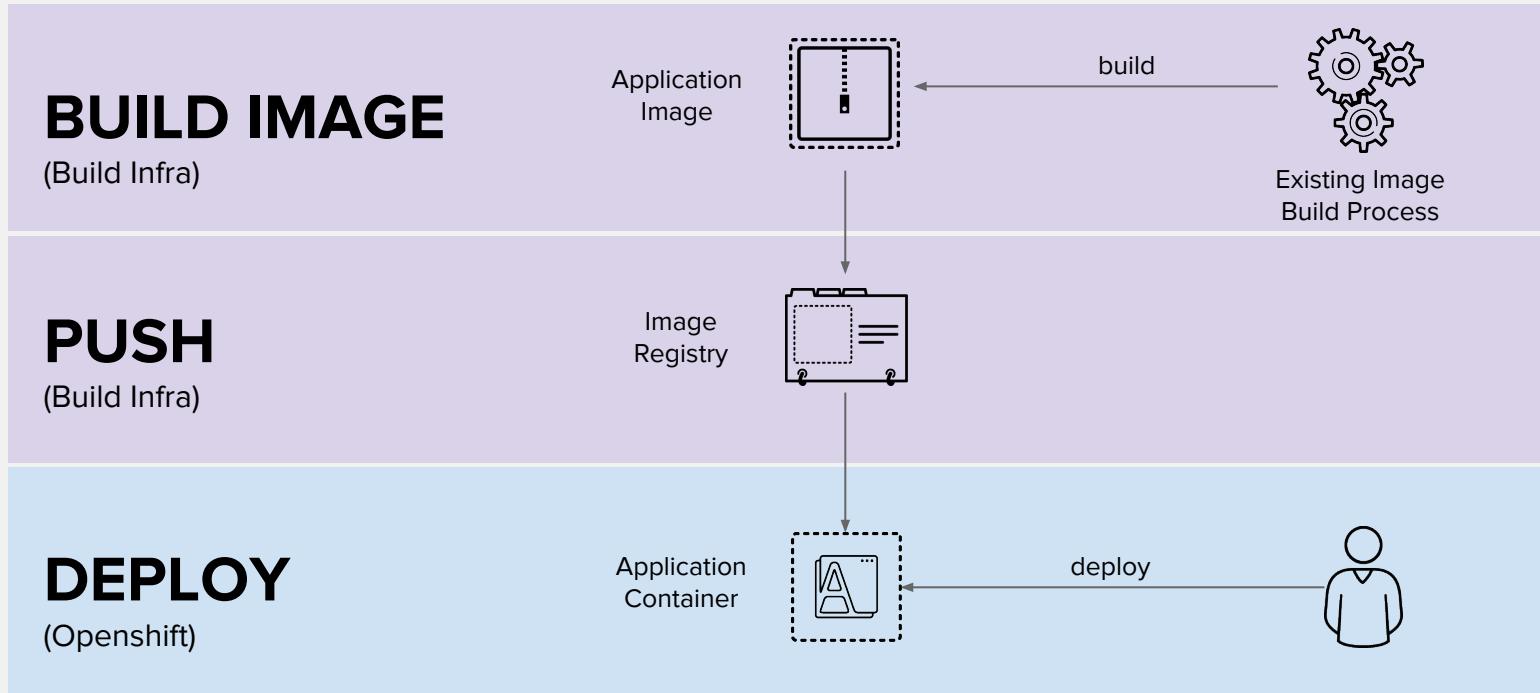
DEPLOY SOURCE CODE WITH SOURCE-TO-IMAGE (S2I)



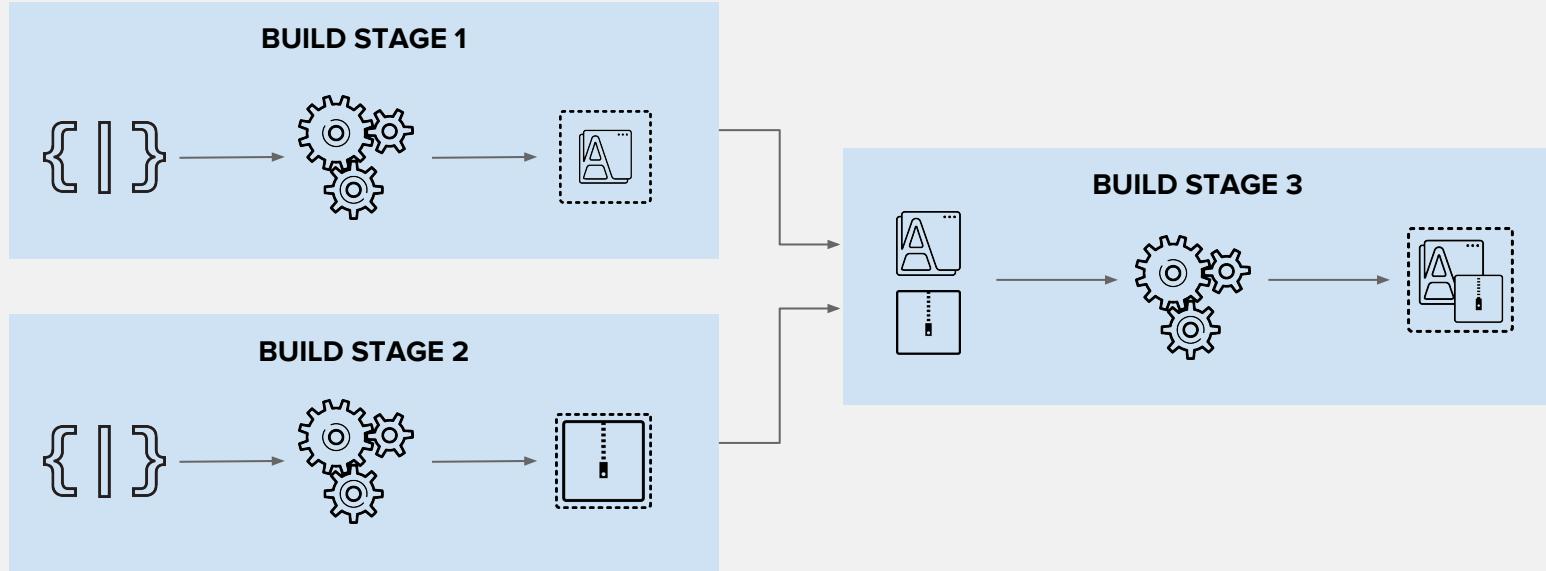
DEPLOY APP BINARY WITH SOURCE-TO-IMAGE (S2I)



DEPLOY DOCKER IMAGE

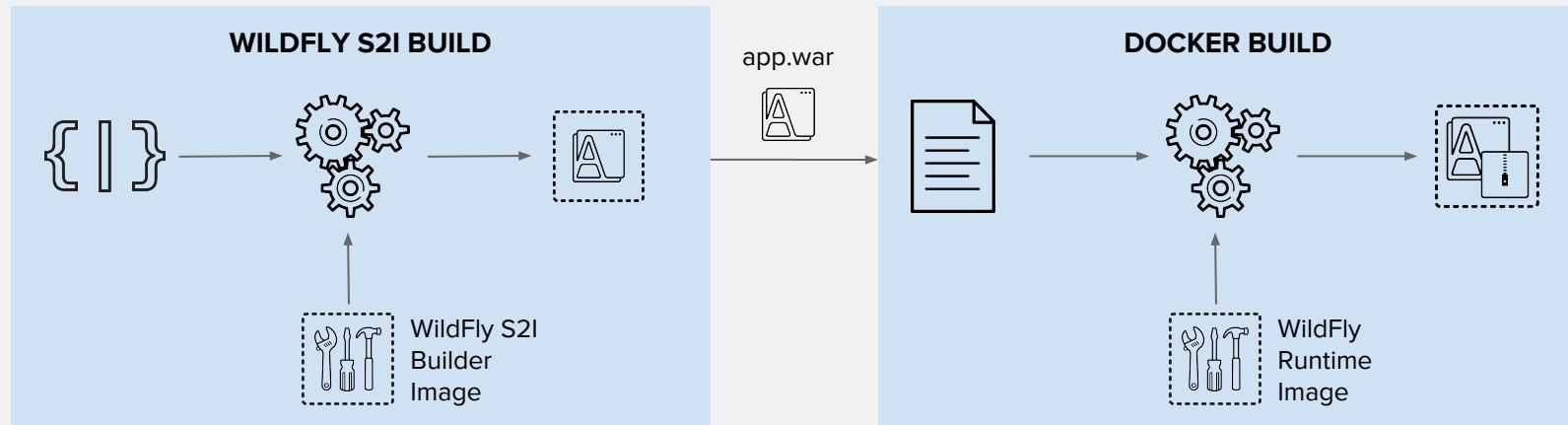


BUILD IMAGES IN MULTIPLE STAGES



EXAMPLE: USE ANY RUNTIME IMAGE WITH SOURCE-TO-IMAGE BUILDS

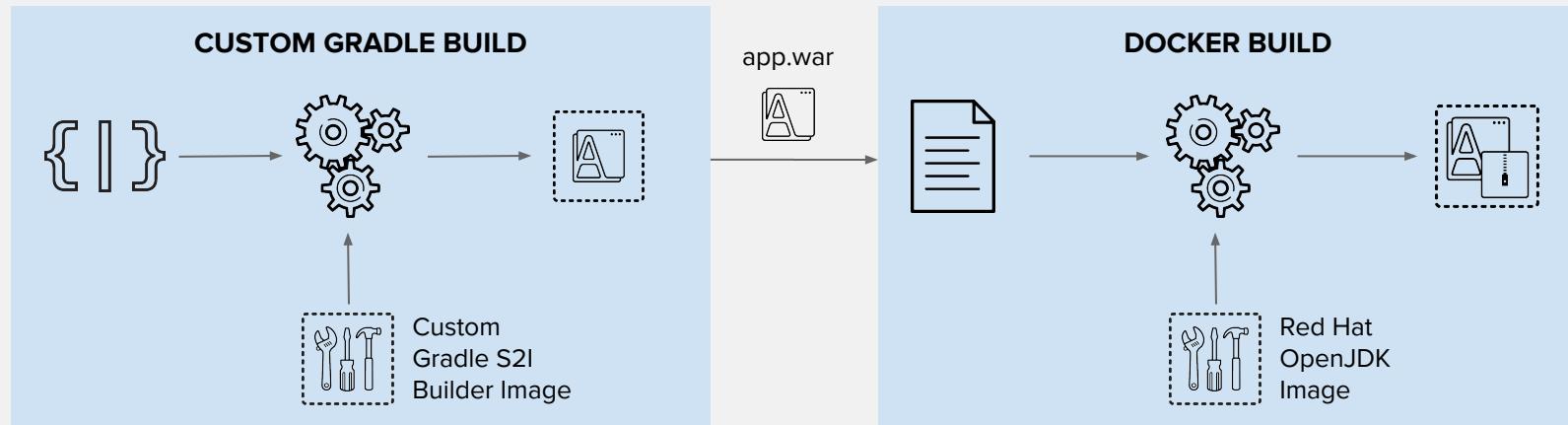
Use Source-to-Image to build app binaries and deploy on lean vanilla runtimes



read more on <https://blog.openshift.com/chaining-builds/>

EXAMPLE: USE ANY BUILD TOOL WITH OFFICIAL RUNTIME IMAGES

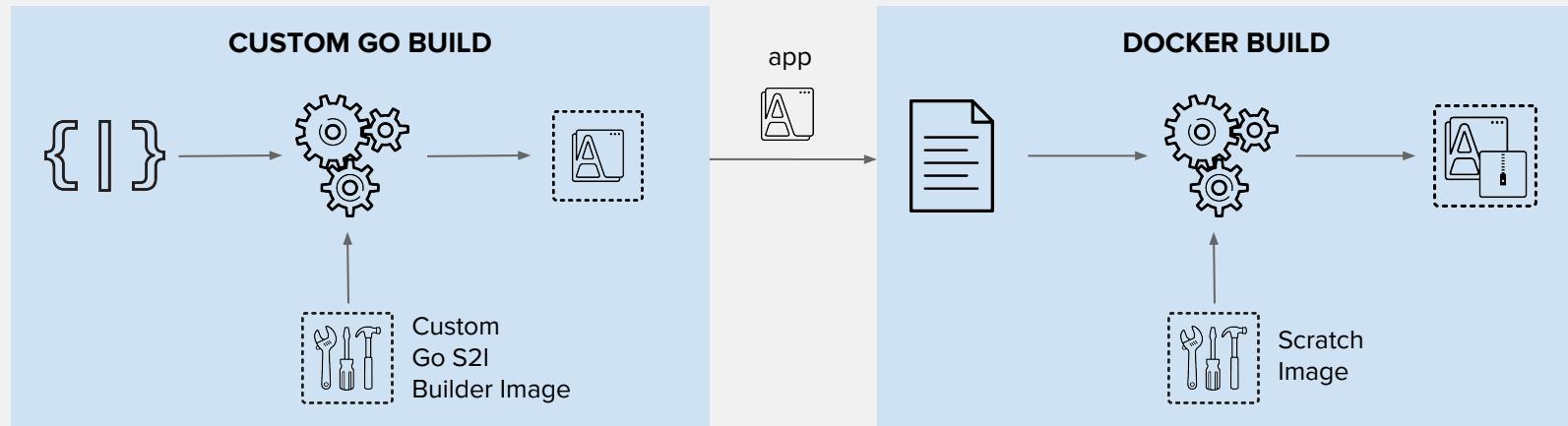
Use your choice of build tool like Gradle and deploy to official images like the JDK image



read more on <https://blog.openshift.com/chaining-builds/>

EXAMPLE: SMALL LEAN RUNTIMES

Build the app binary and deploy on small scratch images



read more on <https://blog.openshift.com/chaining-builds/>

CONTINUOUS INTEGRATION (CI) CONTINUOUS DELIVERY (CD)

CI/CD WITH BUILD AND DEPLOYMENTS

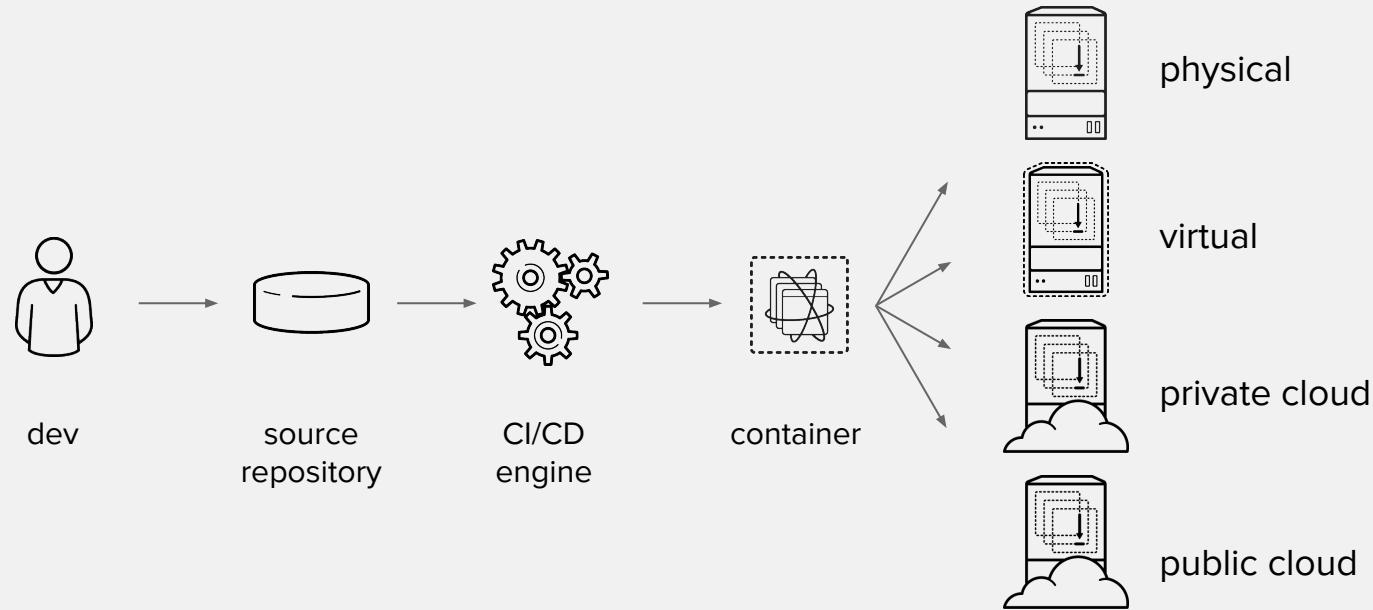
BUILDS

- Webhook triggers: build the app image whenever the code changes
- Image trigger: build the app image whenever the base language or app runtime changes
- Build hooks: test the app image before pushing it to an image registry

DEPLOYMENTS

- Deployment triggers: redeploy app containers whenever configuration changes or the image changes in the OpenShift integrated registry or upstream registries

CONTINUOUS DELIVERY WITH CONTAINERS



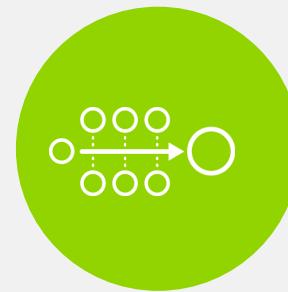
OPENSHIFT LOVES CI/CD



JENKINS-AS-A SERVICE
ON OPENSHIFT



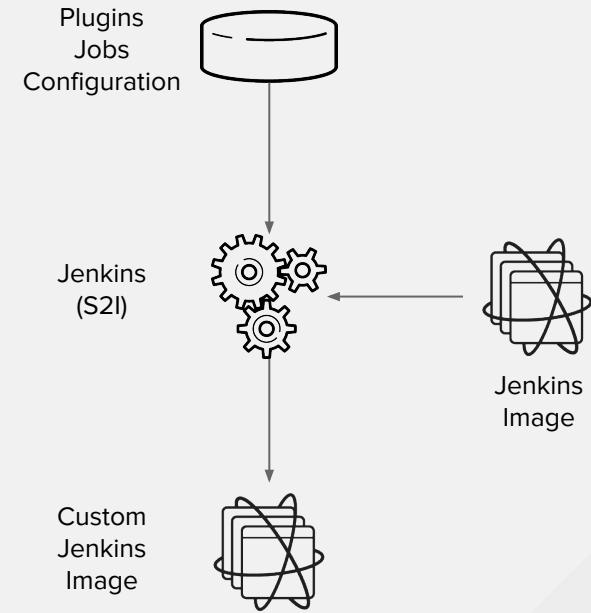
HYBRID JENKINS INFRA
WITH OPENSHIFT



EXISTING CI/CD
DEPLOY TO OPENSHIFT

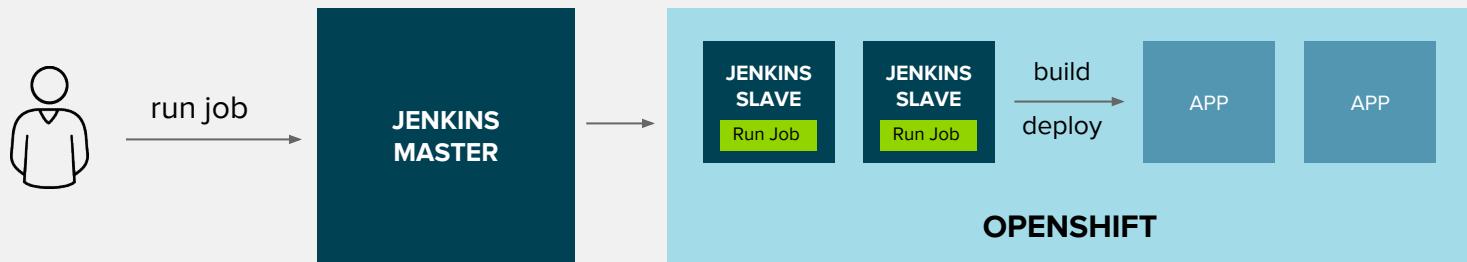
JENKINS-AS-A-SERVICE ON OPENSIFT

- Certified Jenkins images with pre-configured plugins
 - Provided out-of-the-box
 - Follows Jenkins 1.x and 2.x LTS versions
- Jenkins S2I Builder for customizing the image
 - Install Plugins
 - Configure Jenkins
 - Configure Build Jobs
- OpenShift plugins to integrate authentication with OpenShift and also CI/CD pipelines
- Dynamically deploys Jenkins slave containers



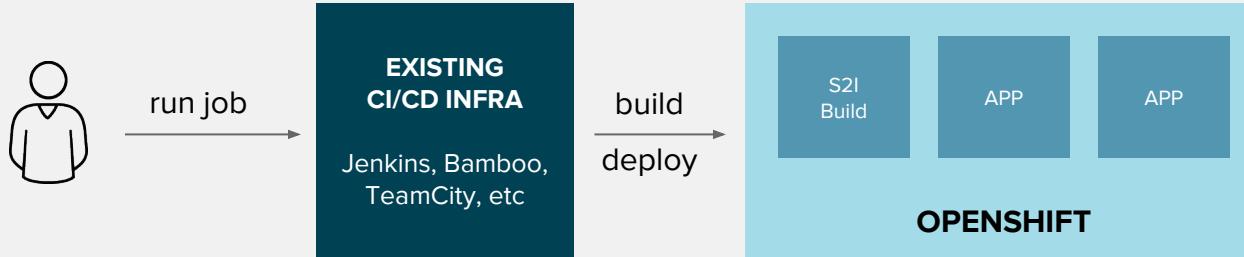
HYBRID JENKINS INFRA WITH OPENSHIFT

- Scale existing Jenkins infrastructure by dynamically provisioning Jenkins slaves on OpenShift
- Use Kubernetes plug-in on existing Jenkins servers



EXISTING CI/CD DEPLOY TO OPENSHIFT

- Existing CI/CD infrastructure outside OpenShift performs operations against OpenShift
 - OpenShift Pipeline Jenkins Plugin for Jenkins
 - OpenShift CLI for integrating other CI Engines with OpenShift
- Without disrupting existing processes, can be combined with previous alternative



OPENShift PIPELINES

- OpenShift Pipelines allow defining a CI/CD workflow via a Jenkins pipeline which can be started, monitored, and managed similar to other builds
- Dynamic provisioning of Jenkins slaves
- Auto-provisioning of Jenkins server
- OpenShift Pipeline strategies
 - Embedded Jenkinsfile
 - Jenkinsfile from a Git repository

```
apiVersion: v1
kind: BuildConfig
metadata:
  name: app-pipeline
spec:
  strategy:
    type: JenkinsPipeline
    jenkinsPipelineStrategy:
      jenkinsfile: |->
        node('maven') { <----->
          stage('build app') {
            git url: 'https://git/app.git'
            sh "mvn package"
          }
          stage('build image') {
            sh "oc start-build app --from-file=target/app.jar"
          }
          stage('deploy') {
            openshiftDeploy deploymentConfig: 'app'
          }
        }
```

Provision a Jenkins slave for running Maven

OpenShift Pipelines in Web Console

app-pipeline created 32 minutes ago

[Start Build](#) [Actions](#)

[Summary](#) Configuration

✓ Latest build #11 complete. [View Log](#)
started 16 minutes ago

A bar chart comparing the duration of recent builds. The y-axis represents Duration in seconds, ranging from 20s to 2m 20s. The x-axis lists build numbers #2, #3, #8, #9, #10, and #11. Builds #2 and #3 are red bars indicating failure, while builds #8, #9, #10, and #11 are blue bars indicating completion. The average duration is 1m 55s.

Build Number	Status	Duration
#2	Failed	2m 20s
#3	Failed	2m 0s
#8	Complete	1m 40s
#9	Complete	1m 20s
#10	Complete	1m 0s
#11	Complete	40s

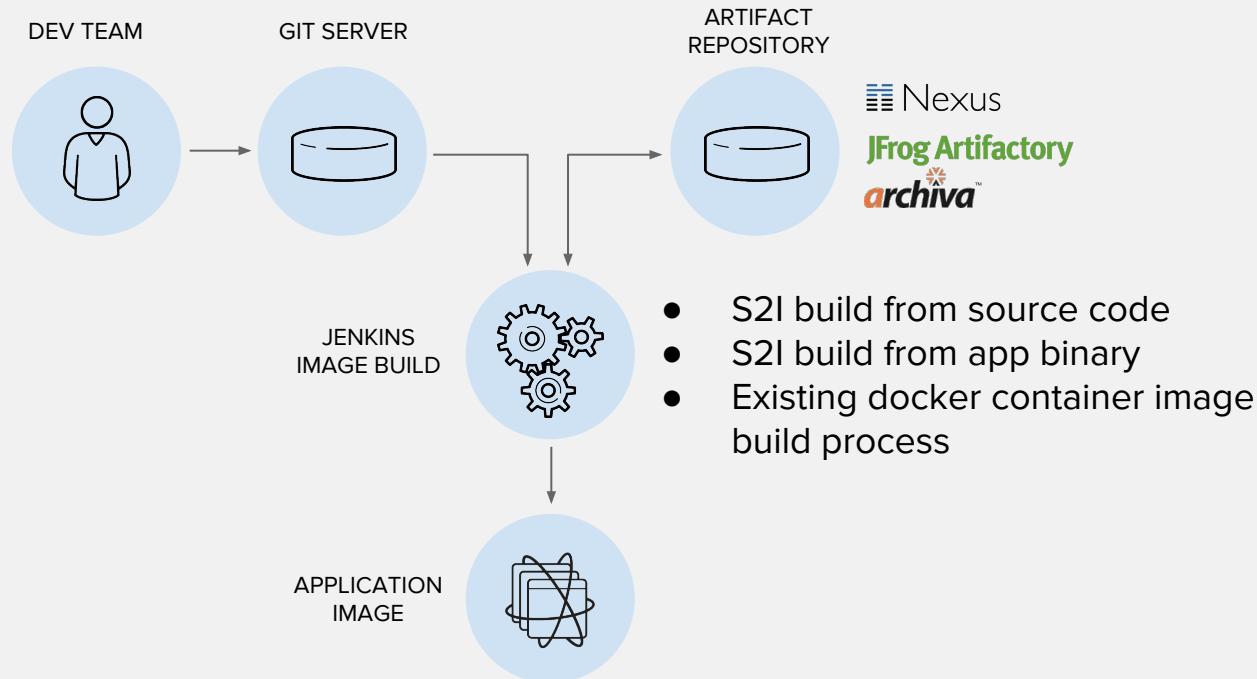
— Average: 1m 55s

The timeline view shows the sequence of stages for two completed builds: Build #11 and Build #10. Each stage is represented by a green horizontal bar with a checkmark at the start. The stages are: build app, build image, and deploy. The duration for each stage is indicated below the bar.

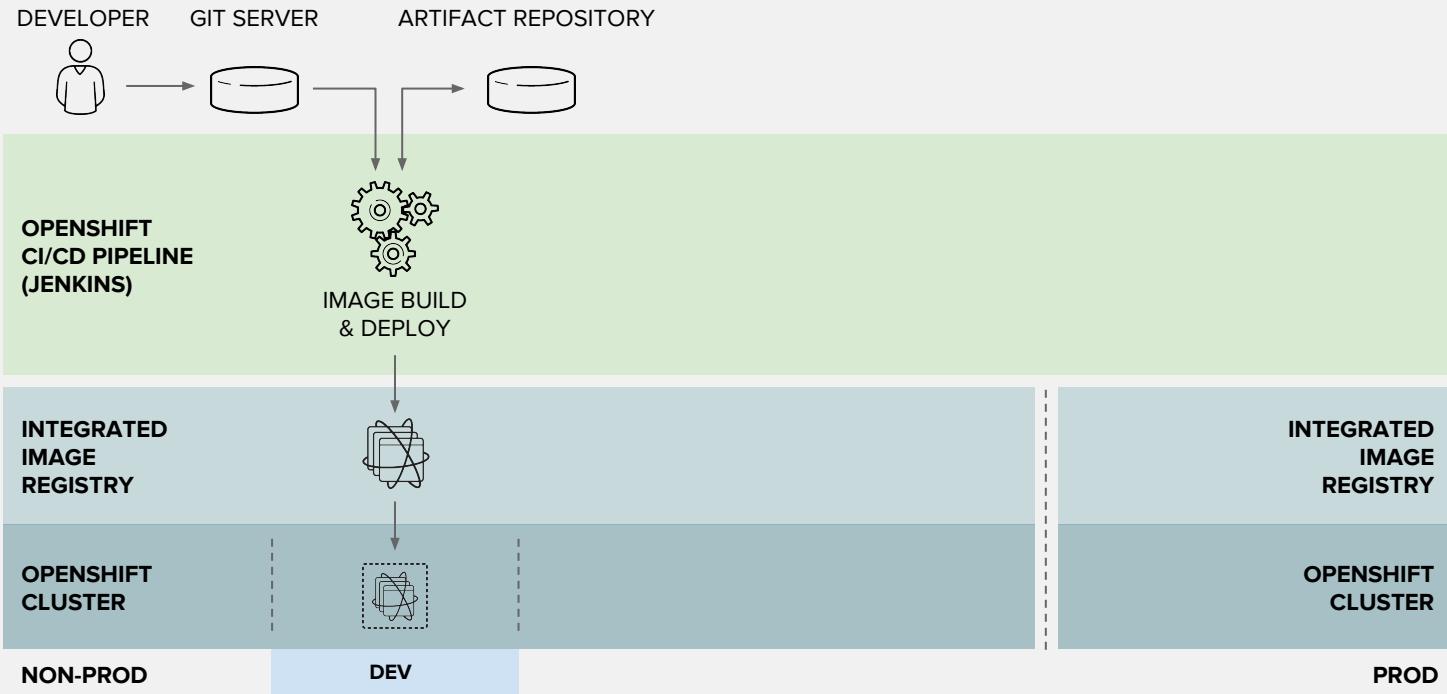
Build	Stage	Duration
Build #11	build app	25s
	build image	16s
	deploy	45s
Build #10	build app	26s
	build image	16s
	deploy	47s

[Filter by label](#) [Add](#)

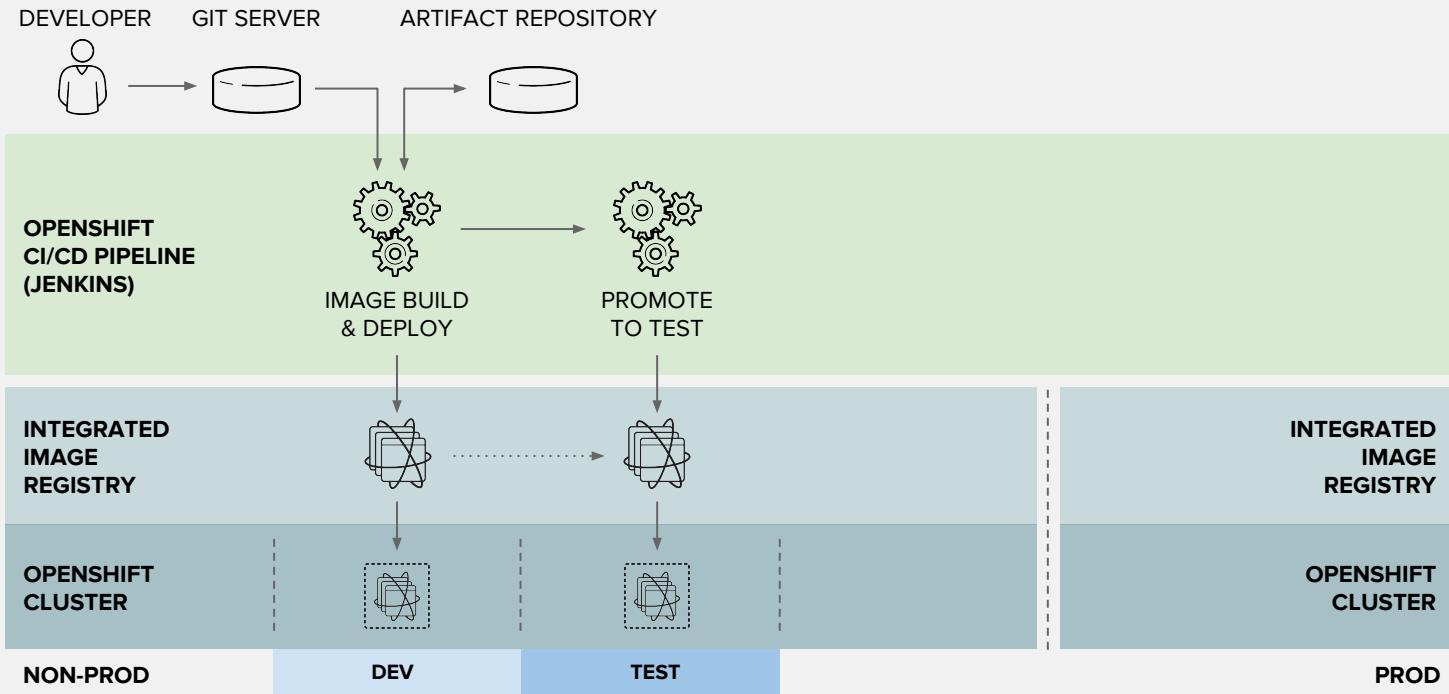
CONTINUOUS DELIVERY PIPELINE



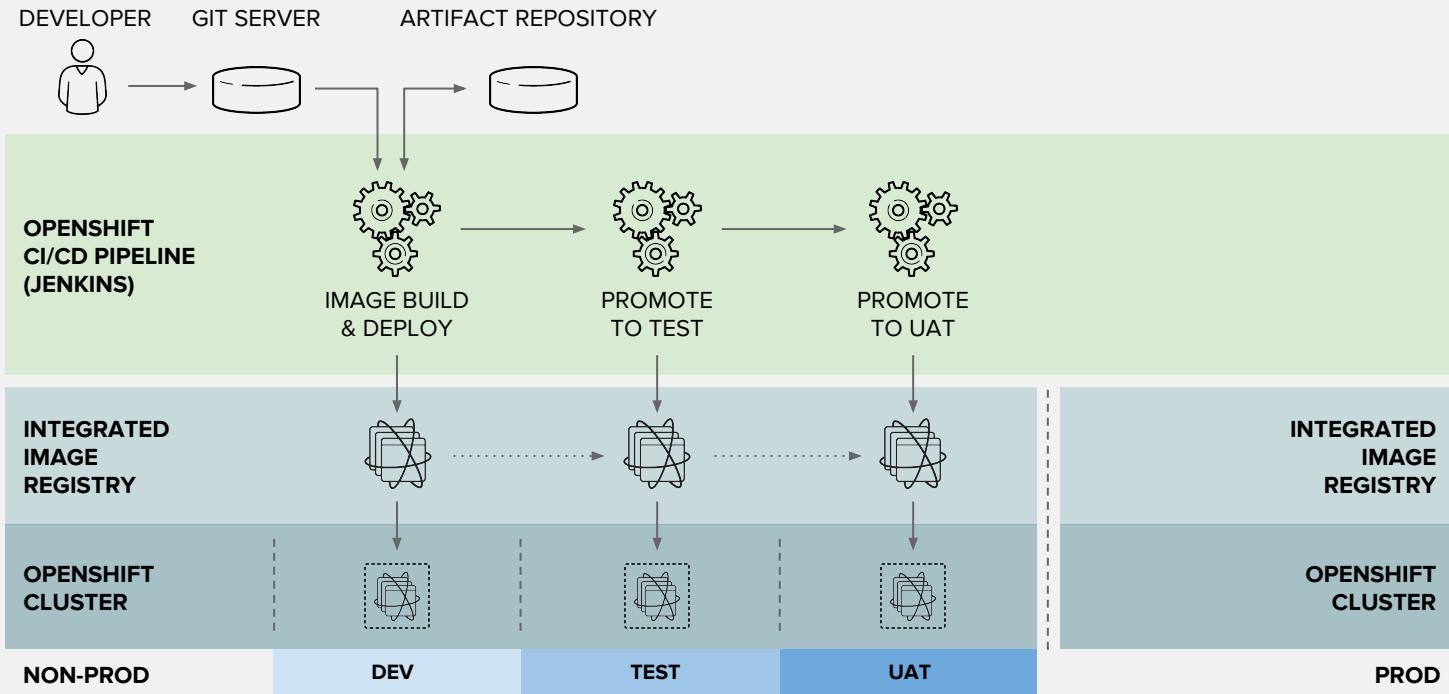
CONTINUOUS DELIVERY PIPELINE



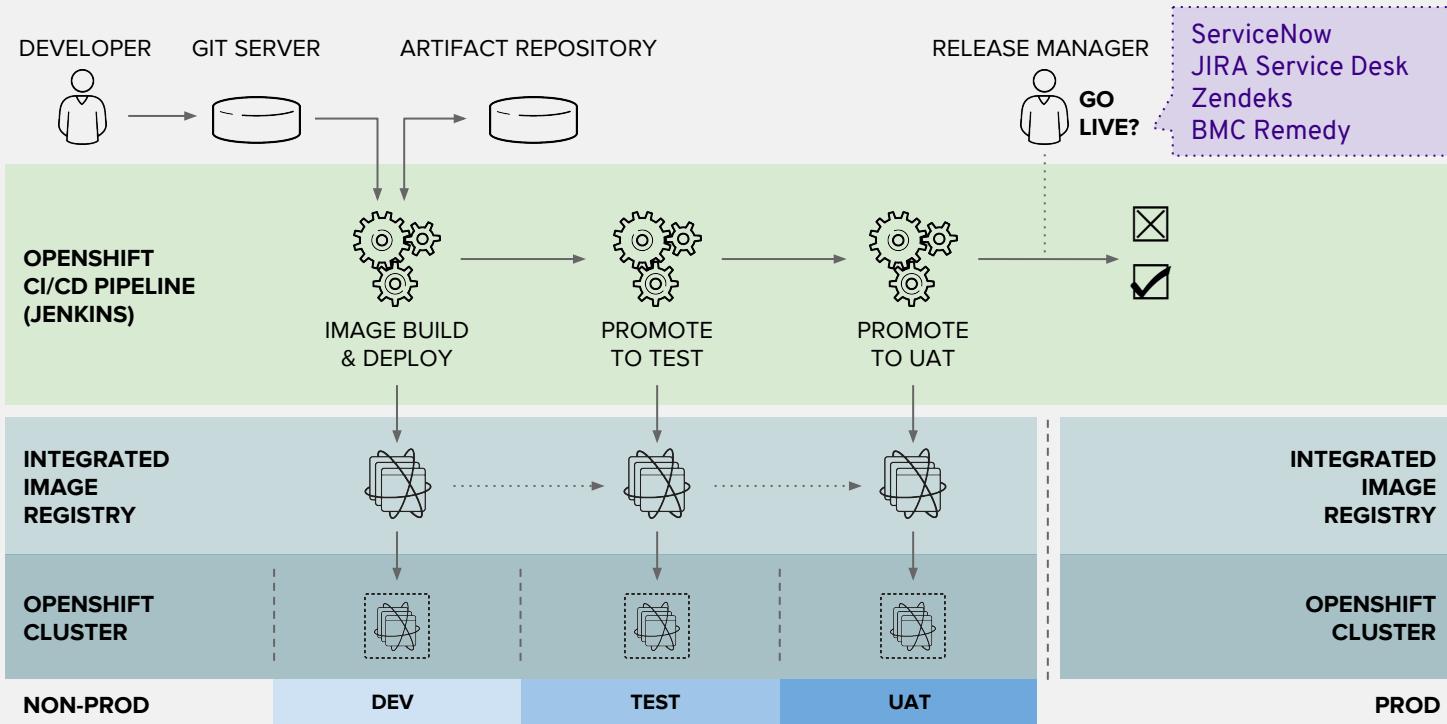
CONTINUOUS DELIVERY PIPELINE



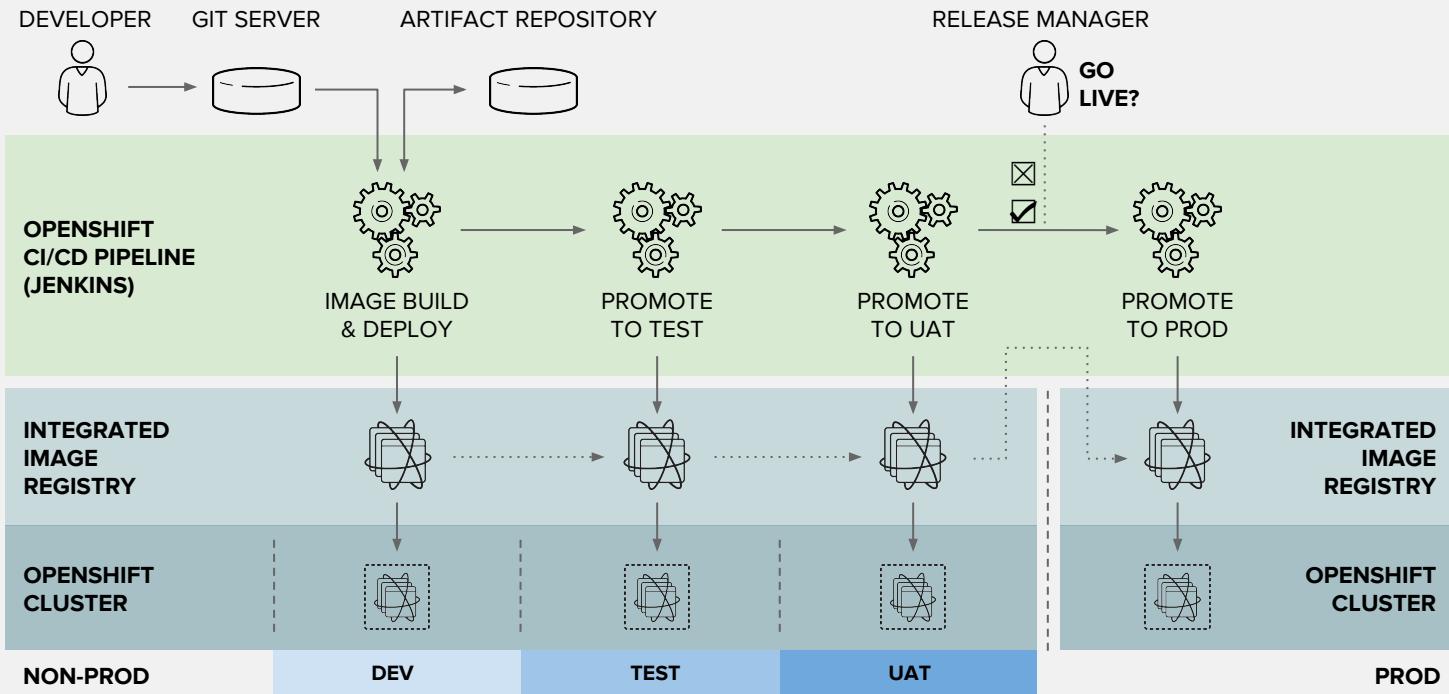
CONTINUOUS DELIVERY PIPELINE



CONTINUOUS DELIVERY PIPELINE

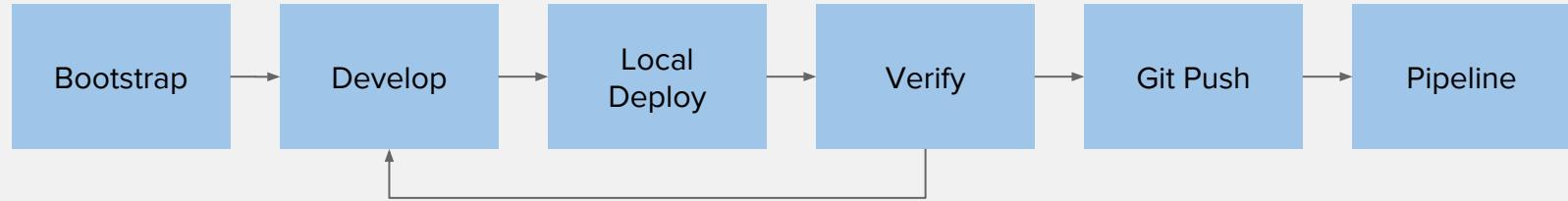


CONTINUOUS DELIVERY PIPELINE

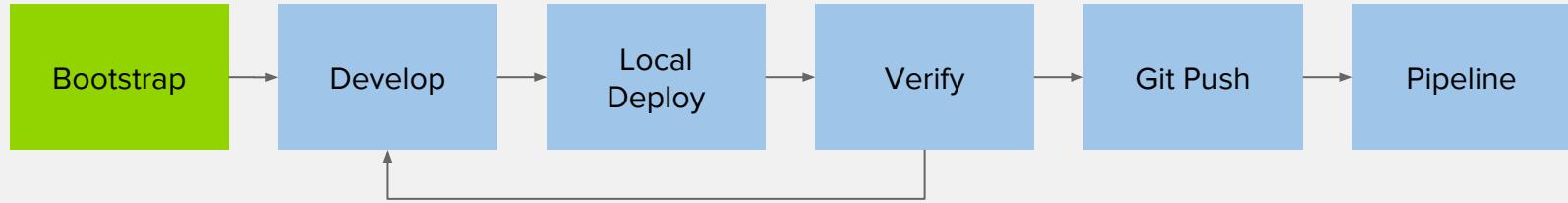


DEVELOPER WORKFLOW

LOCAL DEVELOPMENT WORKFLOW



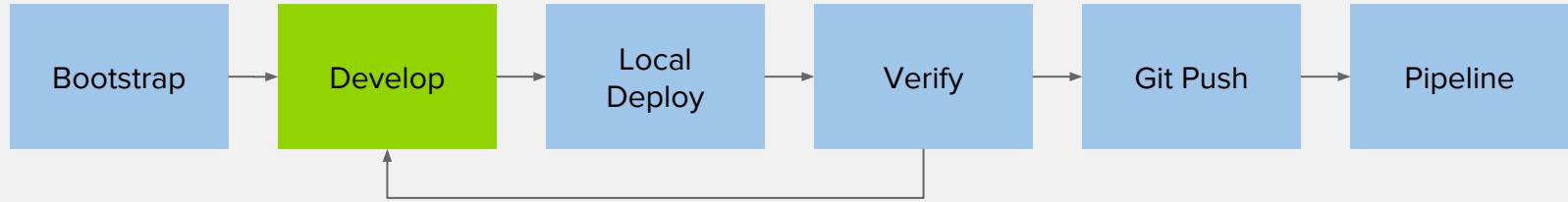
LOCAL DEVELOPMENT WORKFLOW



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

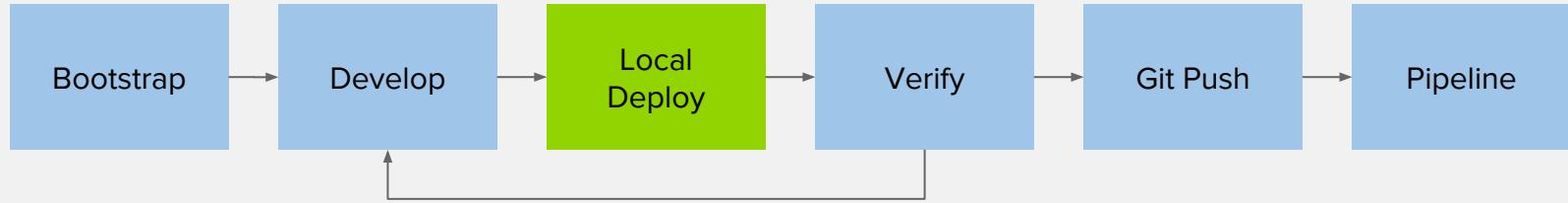
LOCAL DEVELOPMENT WORKFLOW



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

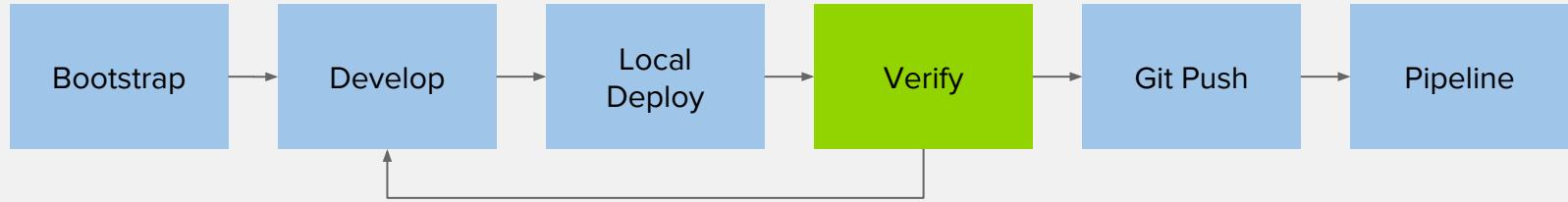
LOCAL DEVELOPMENT WORKFLOW



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

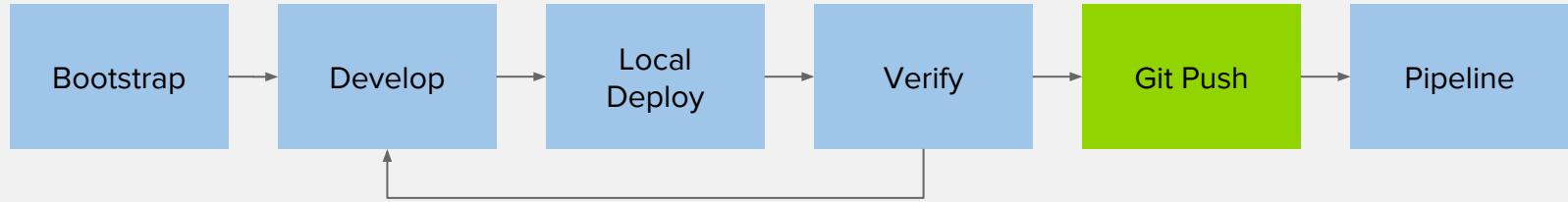
LOCAL DEVELOPMENT WORKFLOW



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

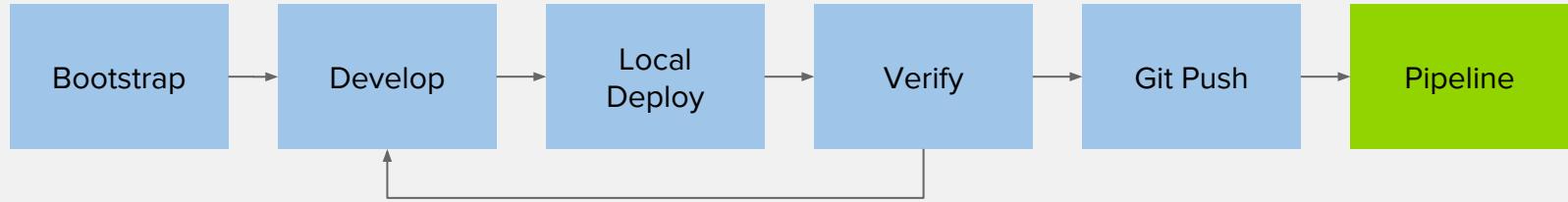
LOCAL DEVELOPMENT WORKFLOW



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

Modern App Dev?

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



Process



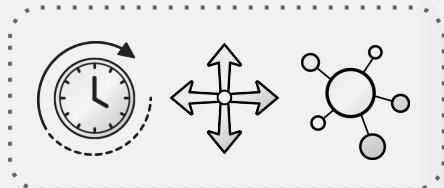
Infrastructure



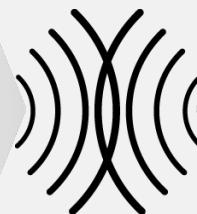
Architecture

PROCESS ?

PROBLEM:

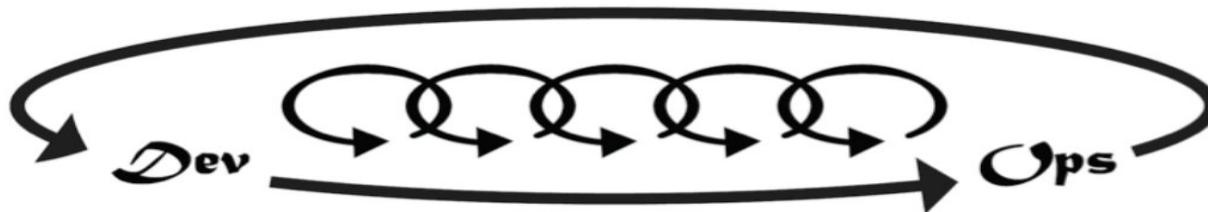


DEVELOPERS



I.T. OPERATIONS

SOLUTION:



Key concepts:

- Small changes -> Less Risk
- Delivery pipeline = **Automation!**
- Fail fast and recover vs. Never fail
- Culture change: Acceptance of failure

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



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

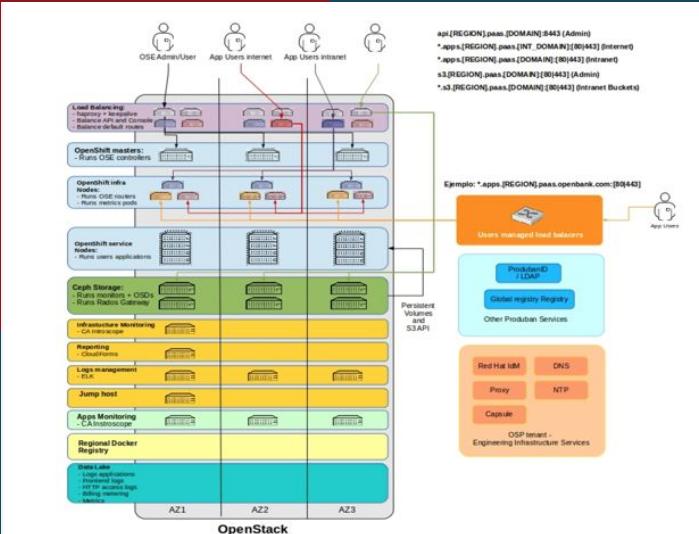
- International Airport, 3rd busiest in Europe, 64M passengers per year.
- Goal to become the world's best digital airport by 2018.
- Needed to accelerate application development time through cloud-agnostic platform.
- Deployed OpenShift across Private Cloud, AWS and Azure.
- Leverage JBoss FUSE and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

Read the [press release](#)



redhat.[®]

Evolving Development Process at Produban (Santander Bank)



Increased collaboration between Dev and Ops

Reduced deployment time & improved time to market

Automated deployment tasks, reduce failures

Provide a global service to reduce infrastructure and operation costs.

Enabled continuous software delivery & reduced resolution time of problems

Evolving Application Architecture at Volvo

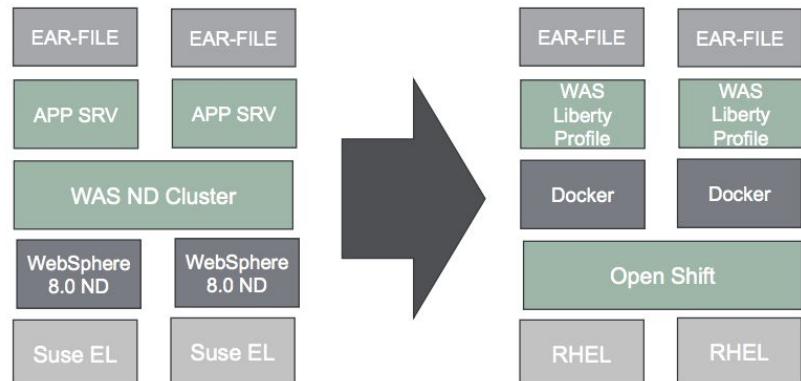
785 apps across 560 app servers

OpenShift provides build, distribution & runtime environment

Platform for DevOps and Microservices

Running OpenShift on Azure,
automatically provisioned with
Ansible

OUR NEW ENVIRONMENT



amadeus



Datacenter



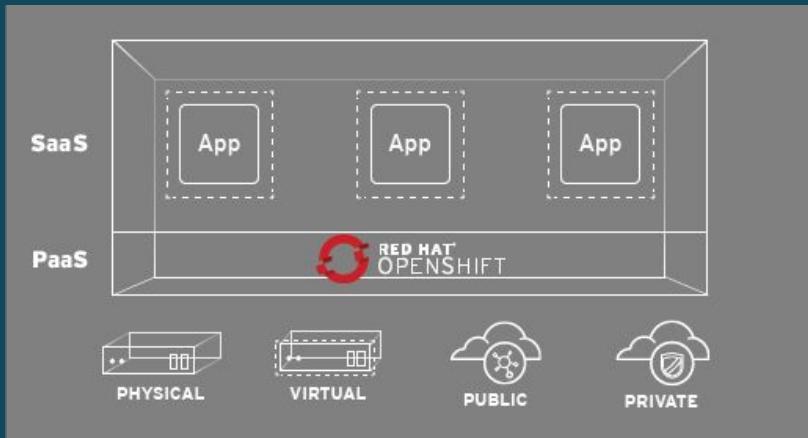
Hosted



Hybrid

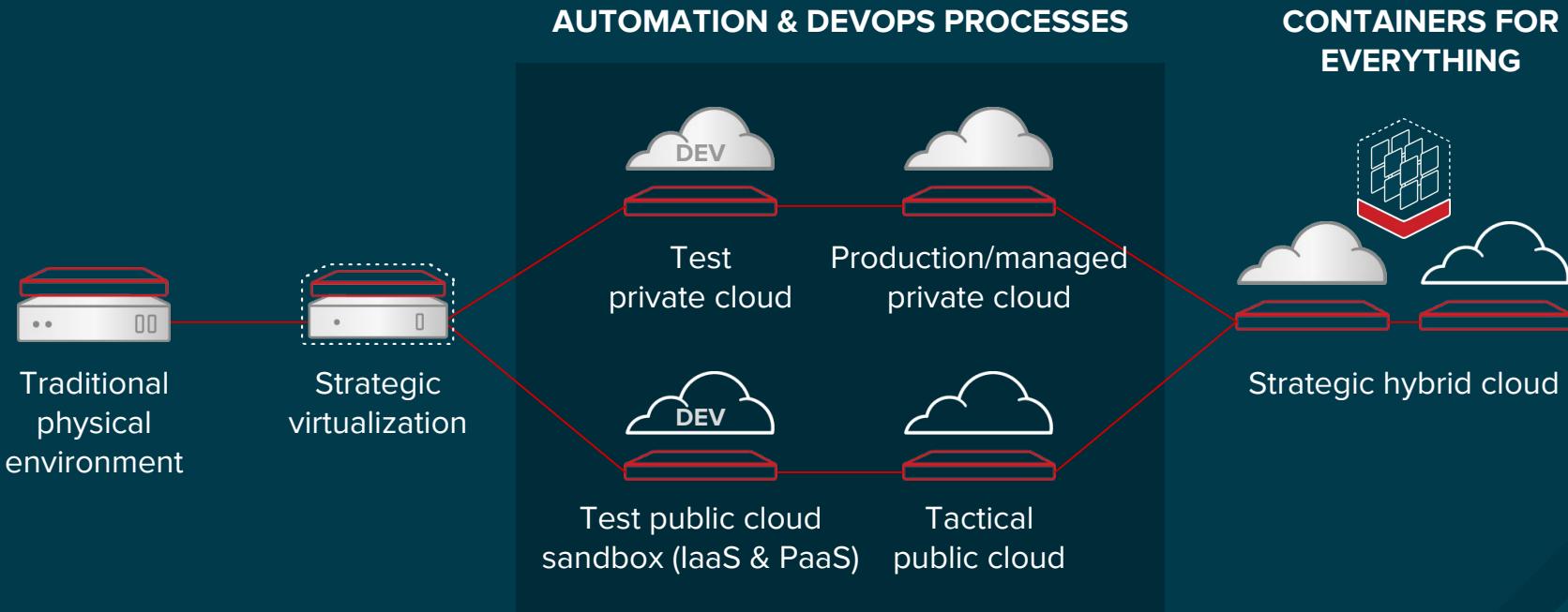
Platform Infrastructure

- 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





redhat.[®]