

TOPIC INDEX

- [Linux Containers](#)
- [OpenShift Concepts Overview](#)
- [OpenShift Architecture](#)
- [OpenShift Installation Architecture](#)
- [Technical Deep Dive](#)
 - [Monitoring Application Health](#)
 - [Networking](#)
 - [Logging & Metrics](#)
 - [Security](#)
 - [Persistent Storage](#)
 - [Service Broker](#)
 - [Operator Framework](#)
 - [Reference Architectures](#)
 - [Build & Deploy Container Images](#)
 - [Continuous Integration \(CI\) & Continuous Delivery \(CD\)](#)
 - [Developer Workflow](#)
 - [Application Services](#)
- [OpenShift 4](#)



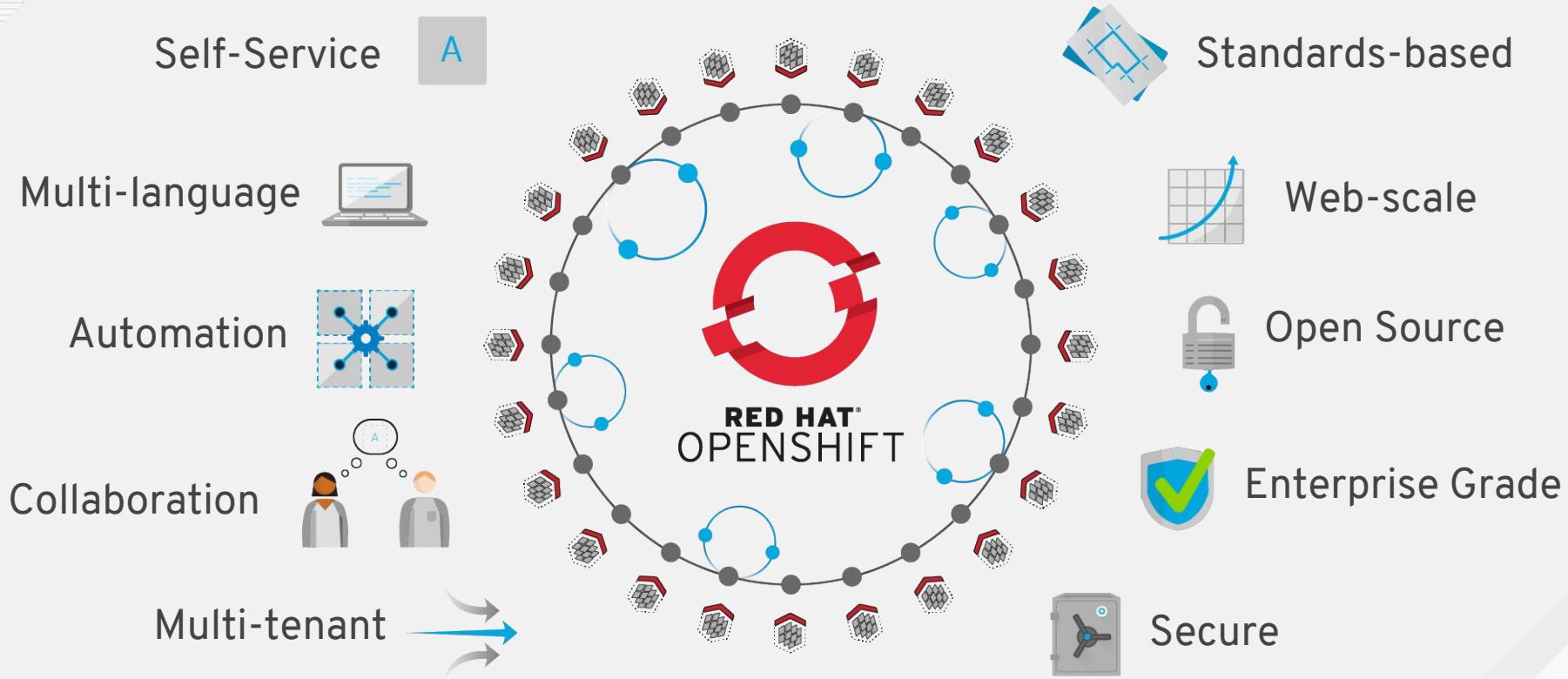
redhat.

OPENSHIFT CONTAINER PLATFORM TECHNICAL OVERVIEW

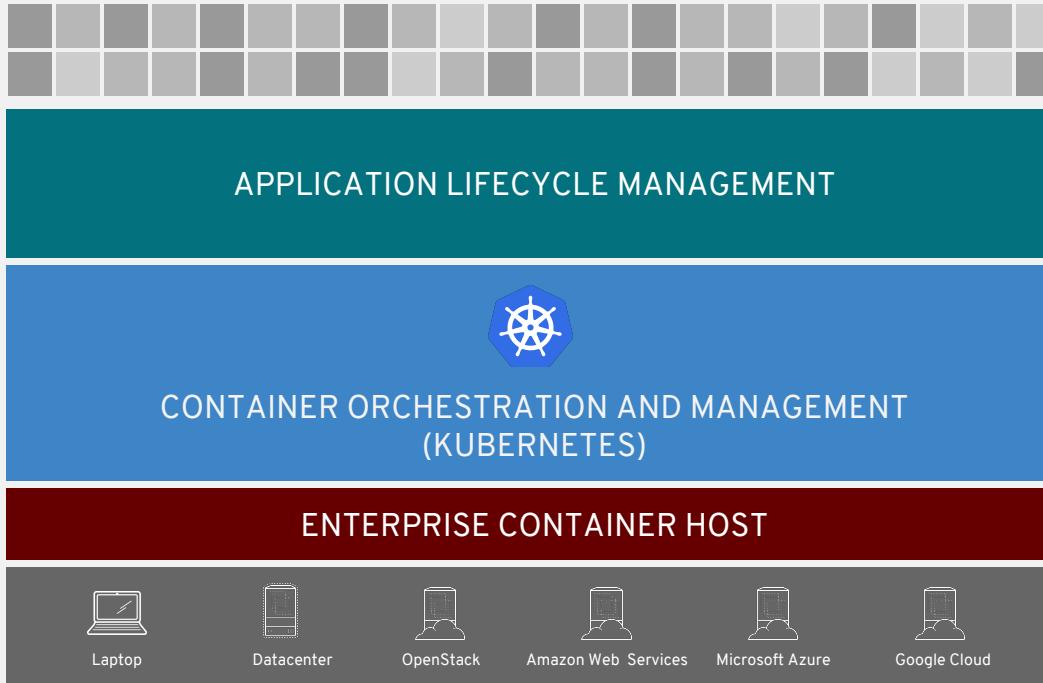
Presenter <Don't overwrite this in the original>

Presenter's title

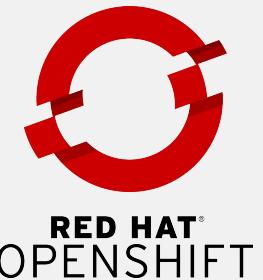
Date



OPENShift CONTAINER PLATFORM

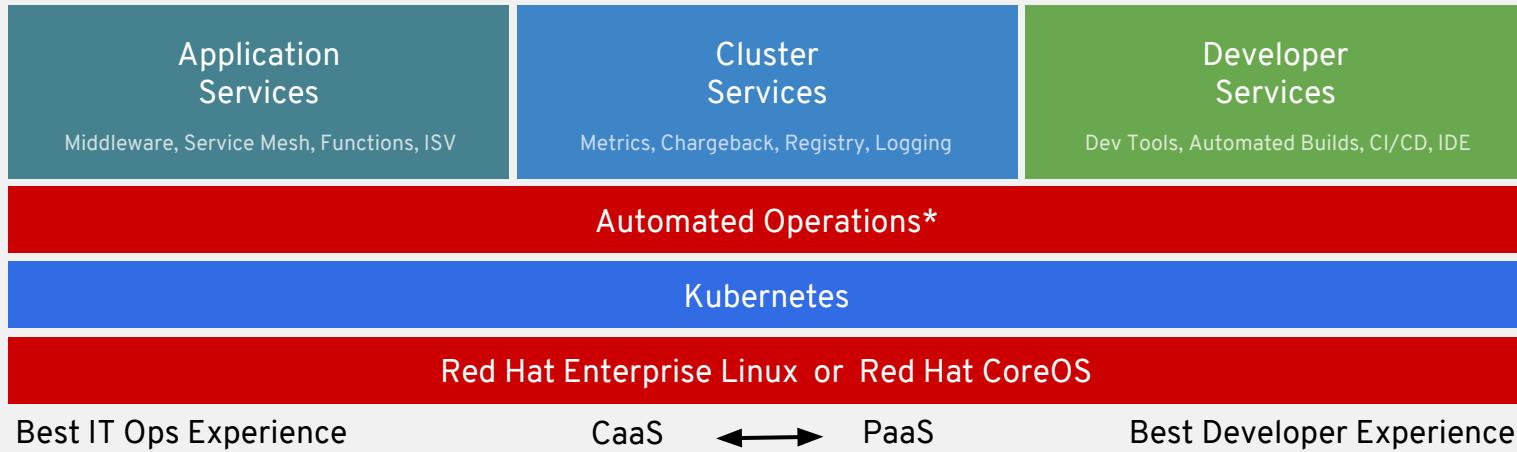


ANY
CONTAINER



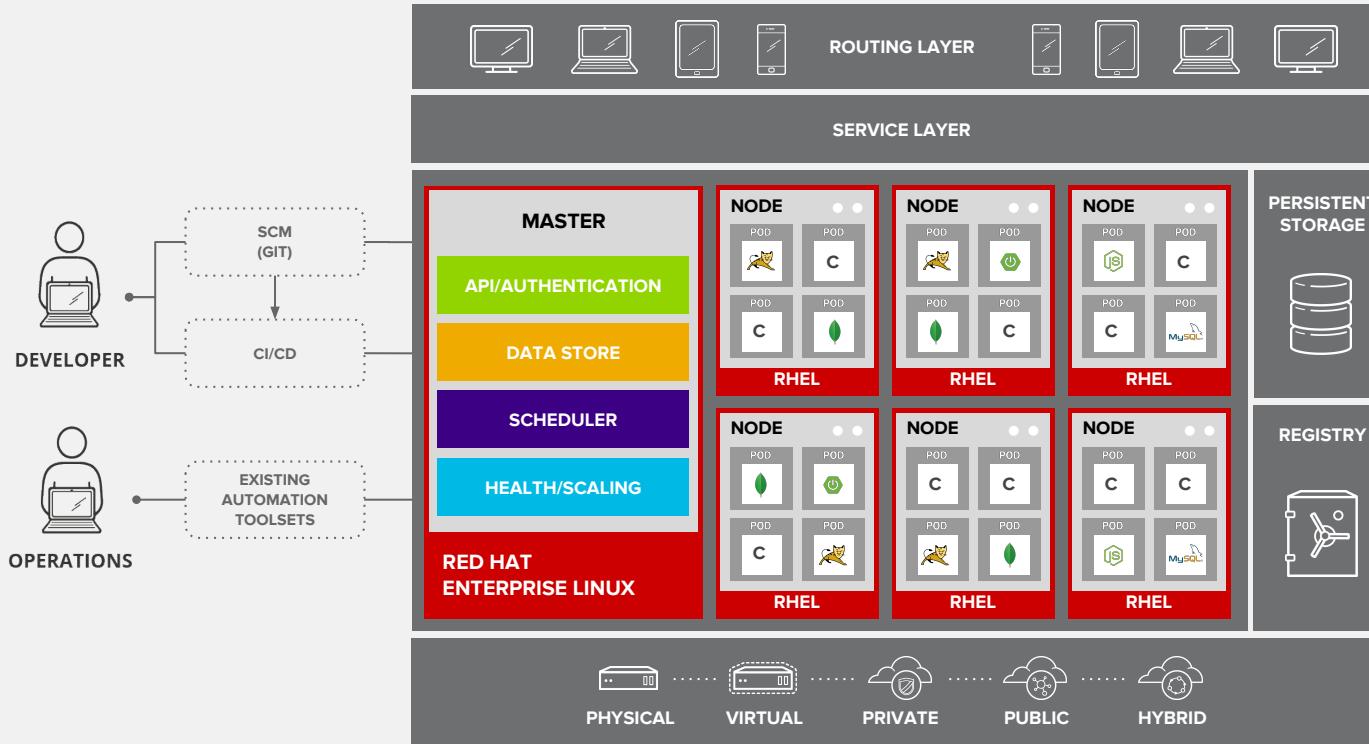
ANY
INFRASTRUCTURE

OPENSHIFT CONTAINER PLATFORM



*coming soon

OPENShift ARCHITECTURE





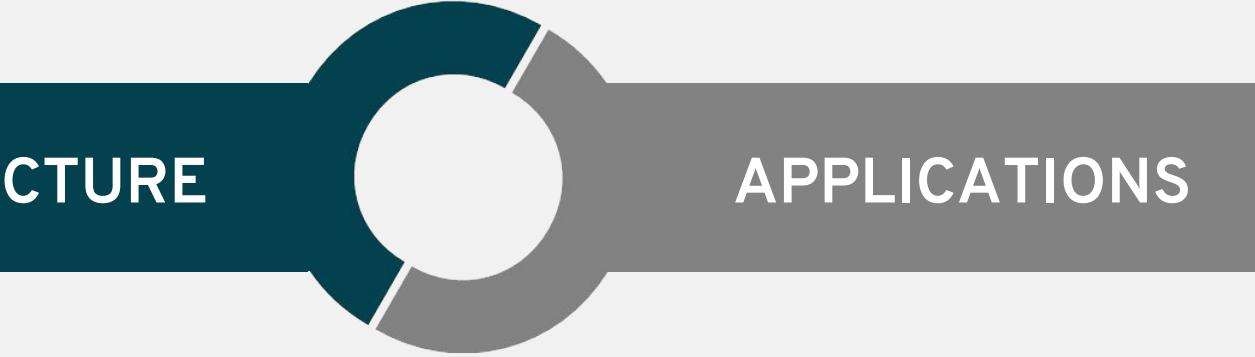
LINUX CONTAINERS

WHAT ARE CONTAINERS?

It Depends Who You Ask

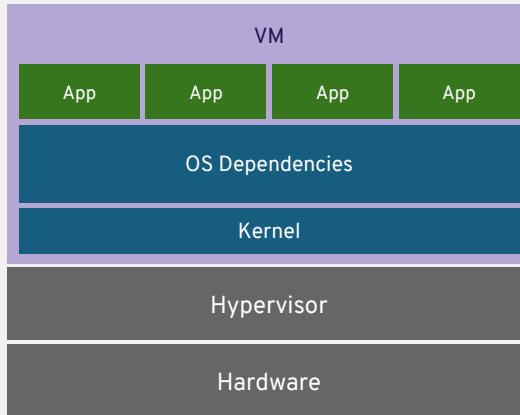
INFRASTRUCTURE

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

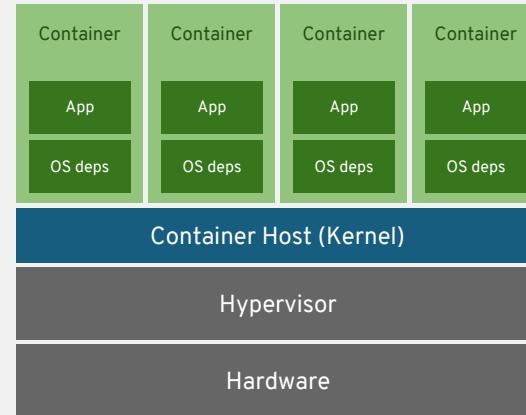
VIRTUAL MACHINES AND CONTAINERS

VIRTUAL MACHINES



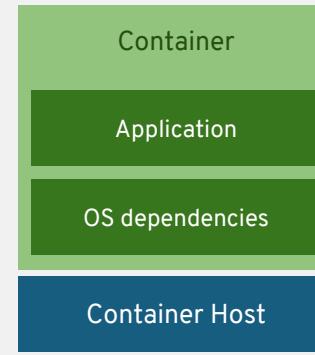
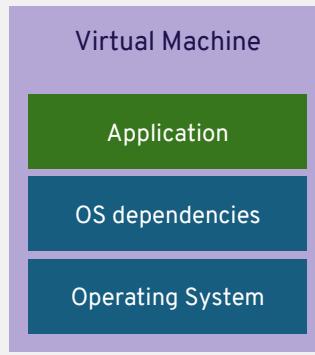
VM isolates the hardware

CONTAINERS



Container isolates the process

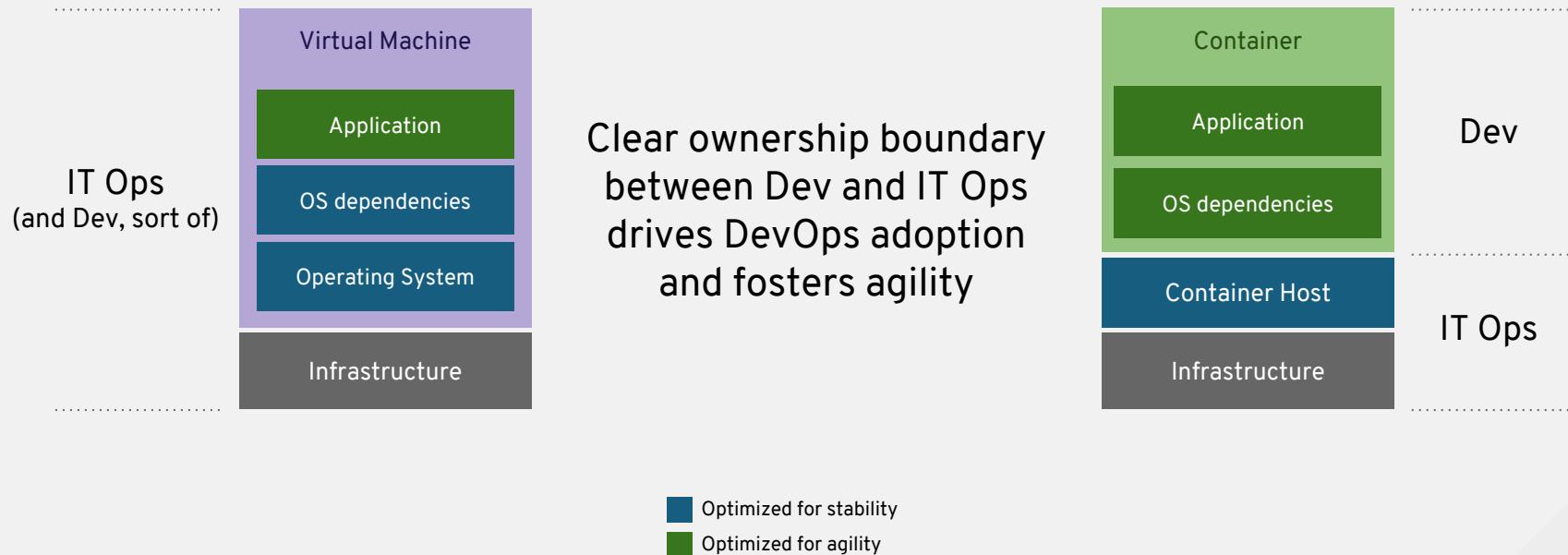
VIRTUAL MACHINES AND CONTAINERS



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

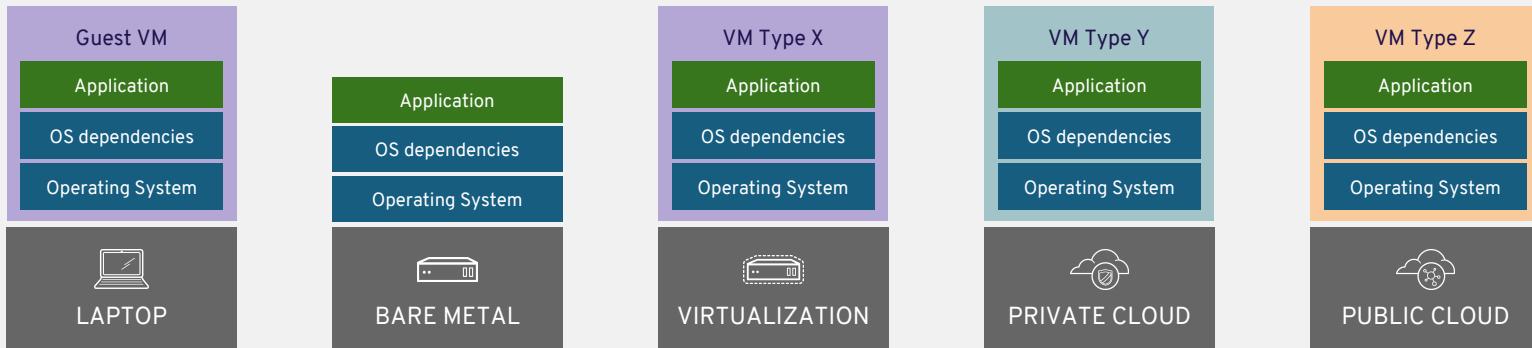
- + Container Isolation
- + Shared Kernel
- + Burstable Compute
- + Burstable Memory
- + Low Resource Usage

VIRTUAL MACHINES AND CONTAINERS



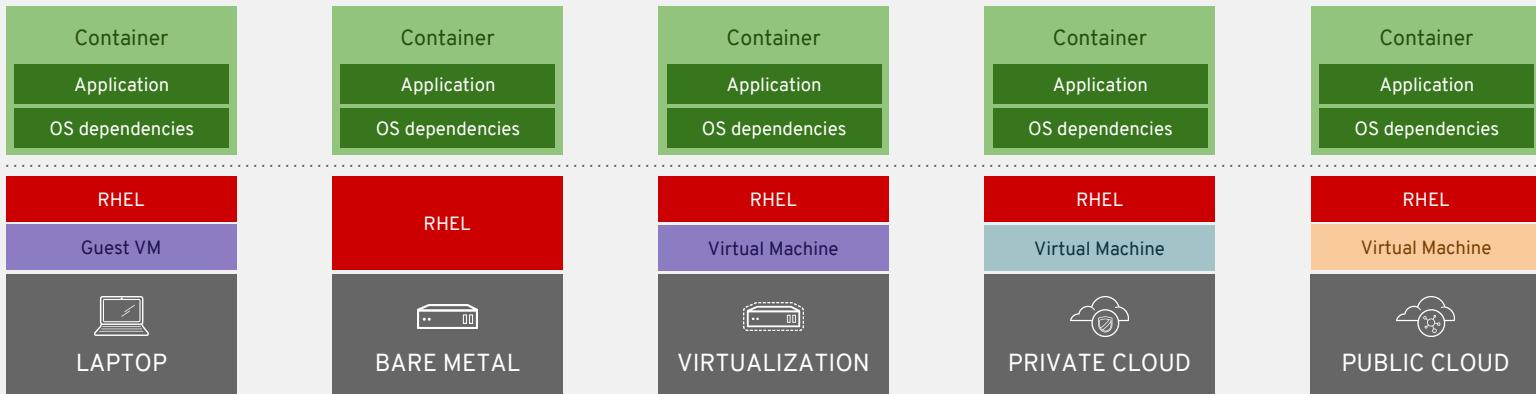
APPLICATION PORTABILITY WITH VM

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

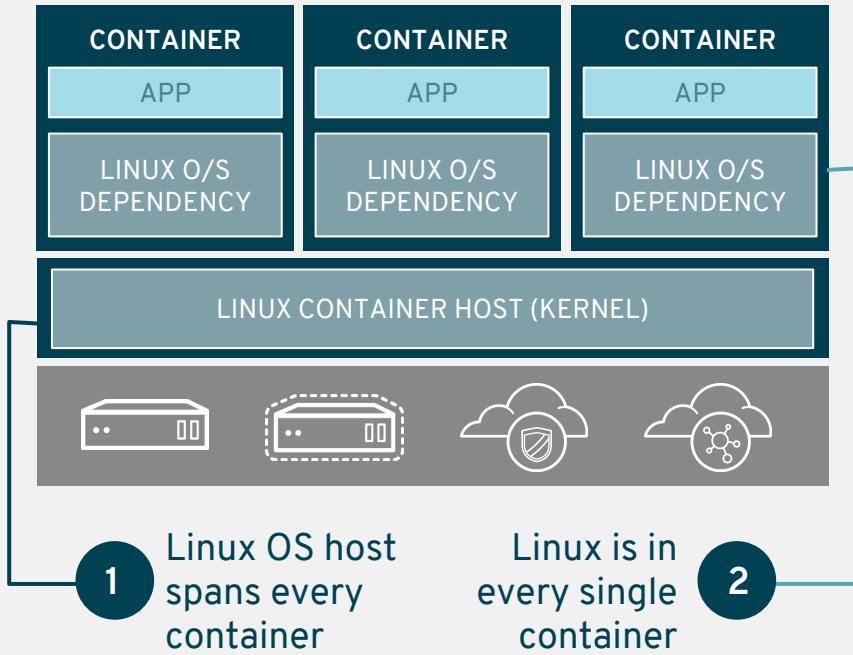


APPLICATION PORTABILITY WITH CONTAINERS

RHEL Containers + RHEL Host = Guaranteed Portability
Across Any Infrastructure



LINUX AND CONTAINER INFRASTRUCTURE

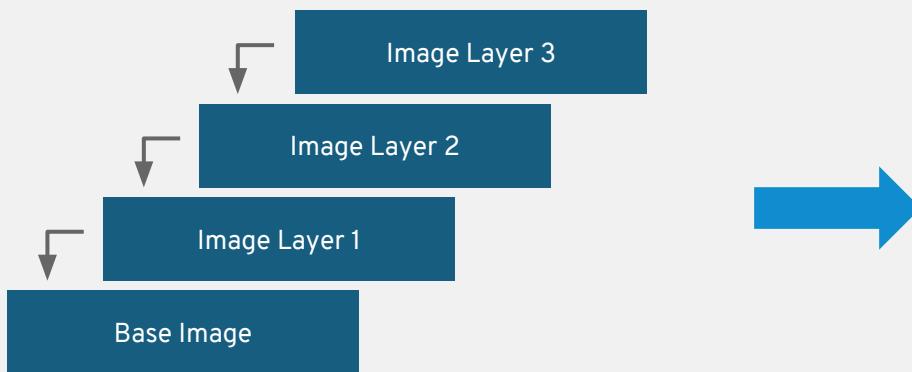


CONTAINERS ARE LINUX

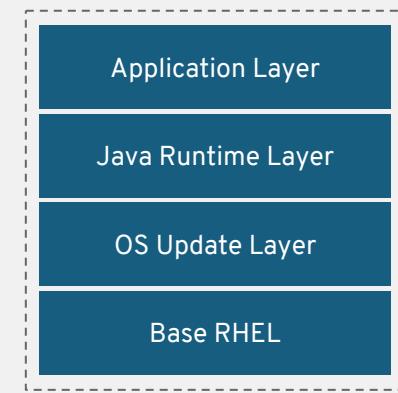
Red Hat
Enterprise Linux
is a leader in paid
Linux

70%
CY2016 paid
Linux share

RAPID SECURITY PATCHING USING CONTAINER IMAGE LAYERING



Container Image Layers



Example Container Image



cri-o

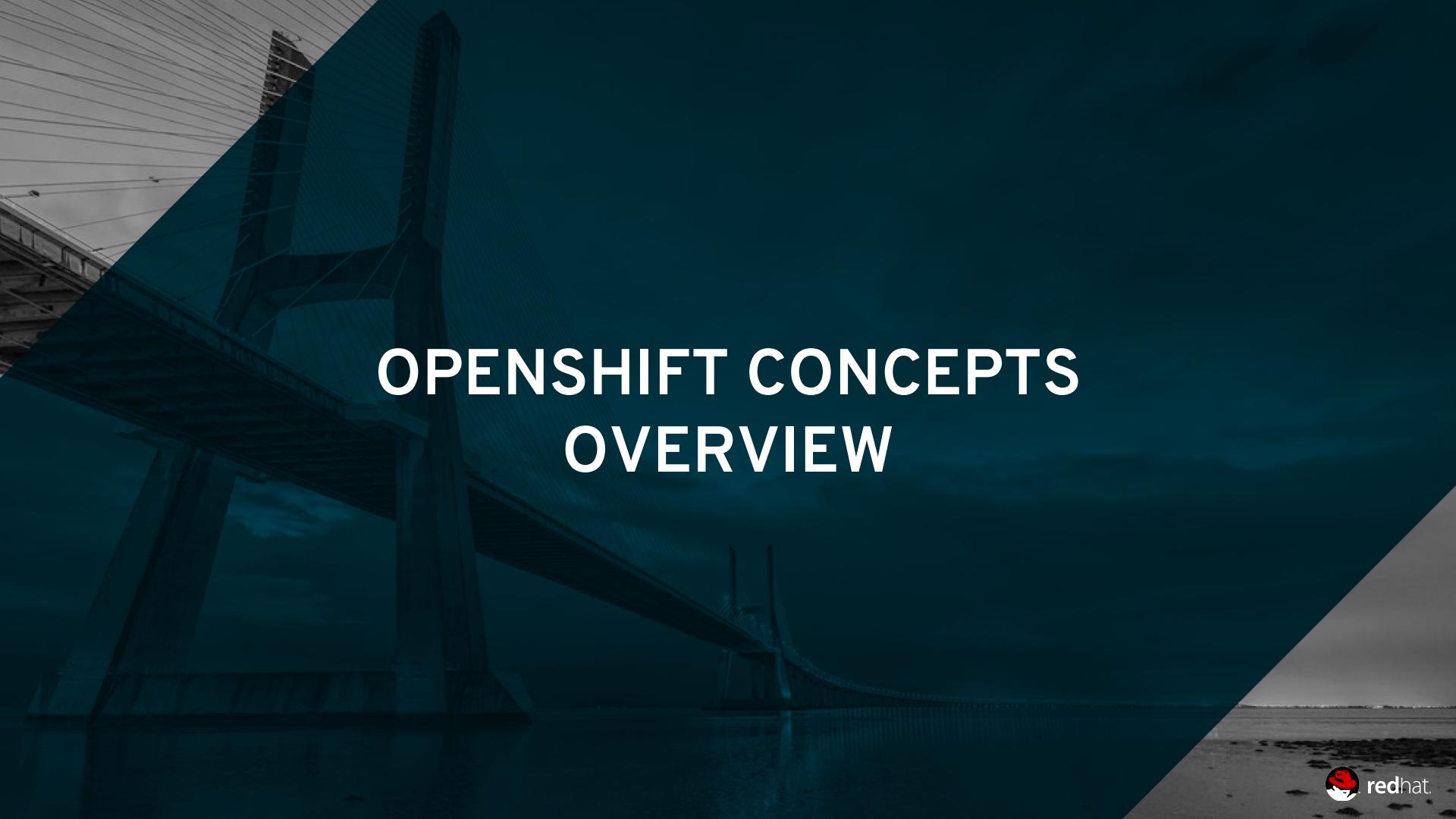
A lightweight, OCI-compliant container runtime

Minimal and Secure
Architecture

Optimized for
Kubernetes

Runs any
OCI-compliant image
(including docker)

Optional runtime in OCP 3.10, default OCP 3.11+



OPENSHIFT CONCEPTS OVERVIEW

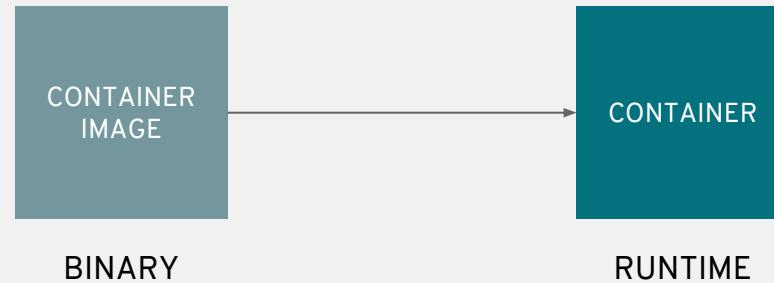


A container is the smallest compute unit

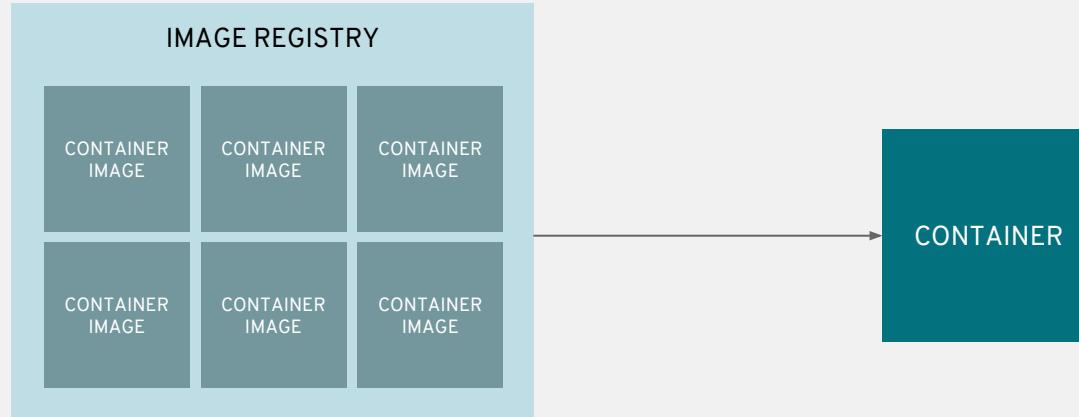




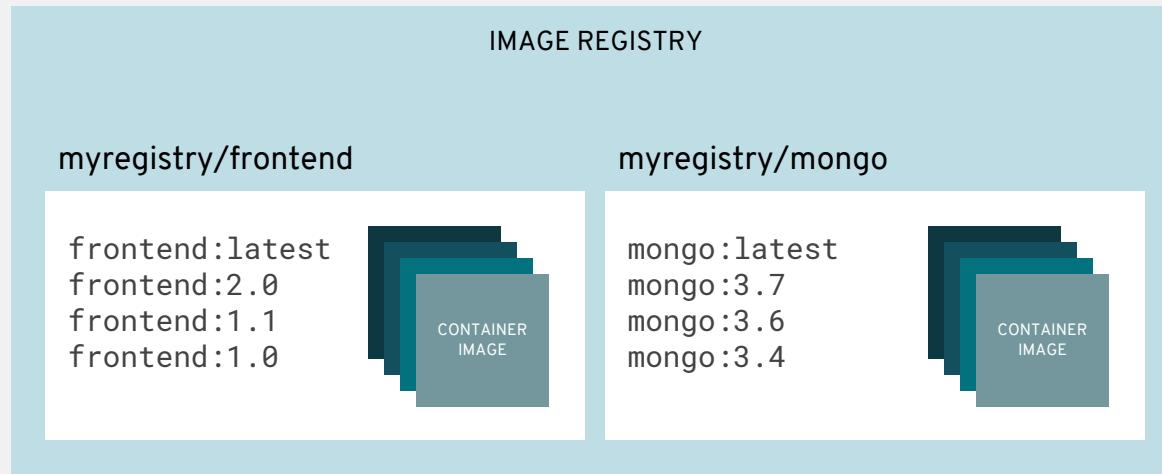
containers are created from container images



container images are stored in an image registry



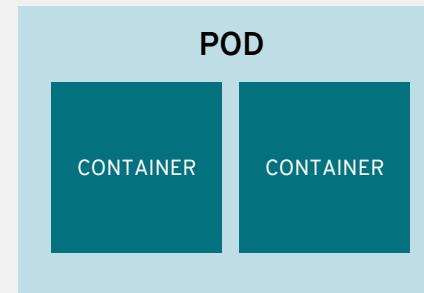
an image repository contains all versions of an image in the image registry



containers are wrapped in pods which are units of deployment and management

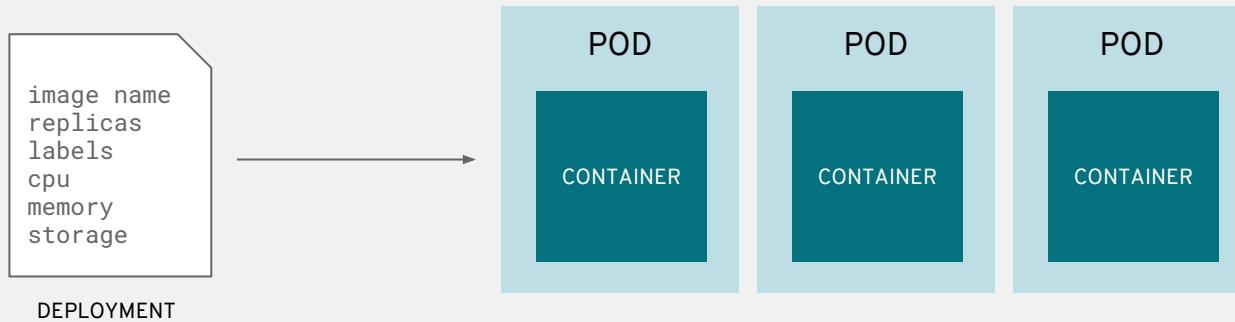


IP: 10.1.0.11

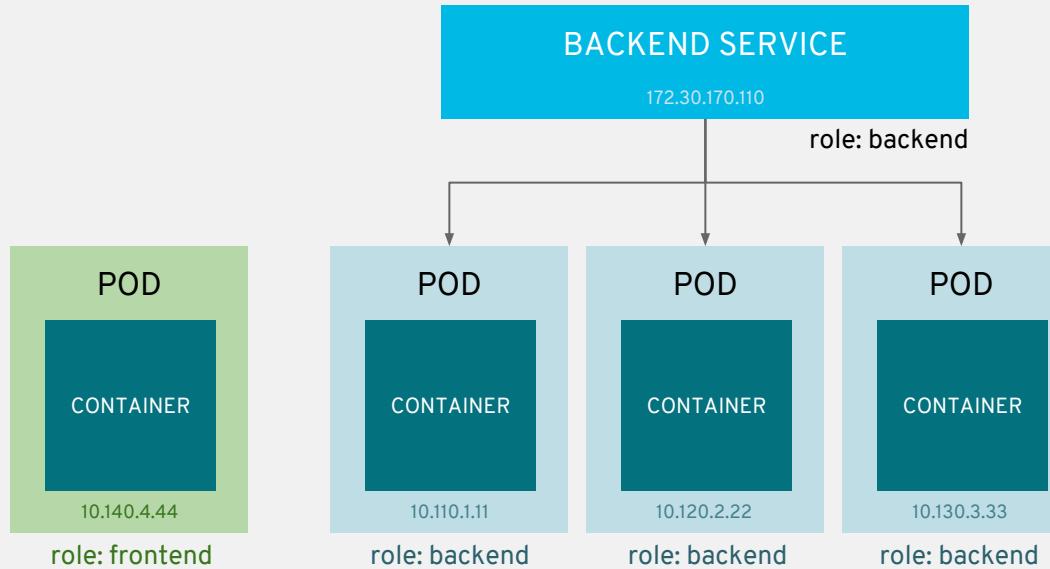


IP: 10.1.0.55

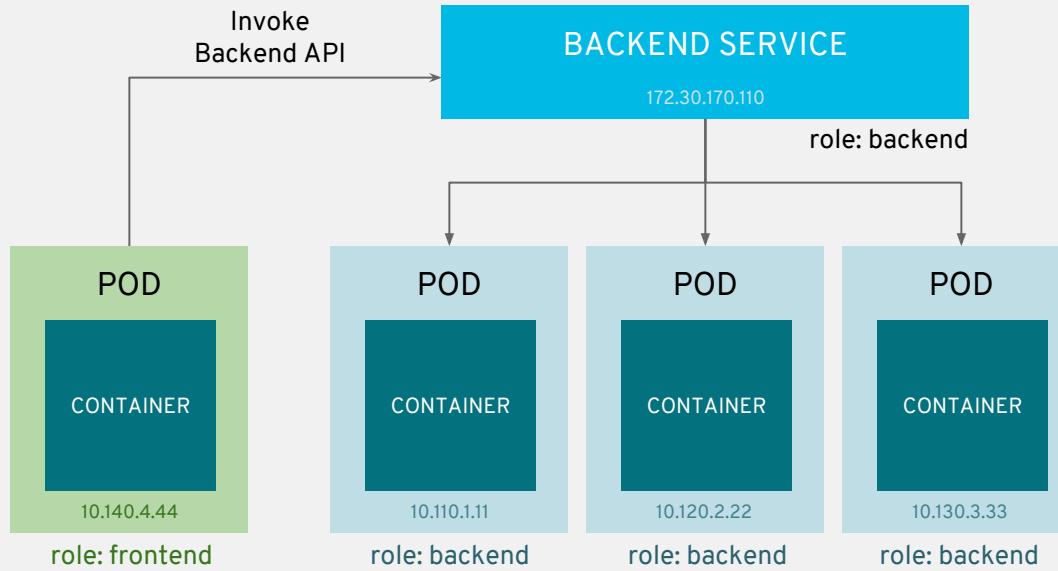
pods configuration is defined in a deployment



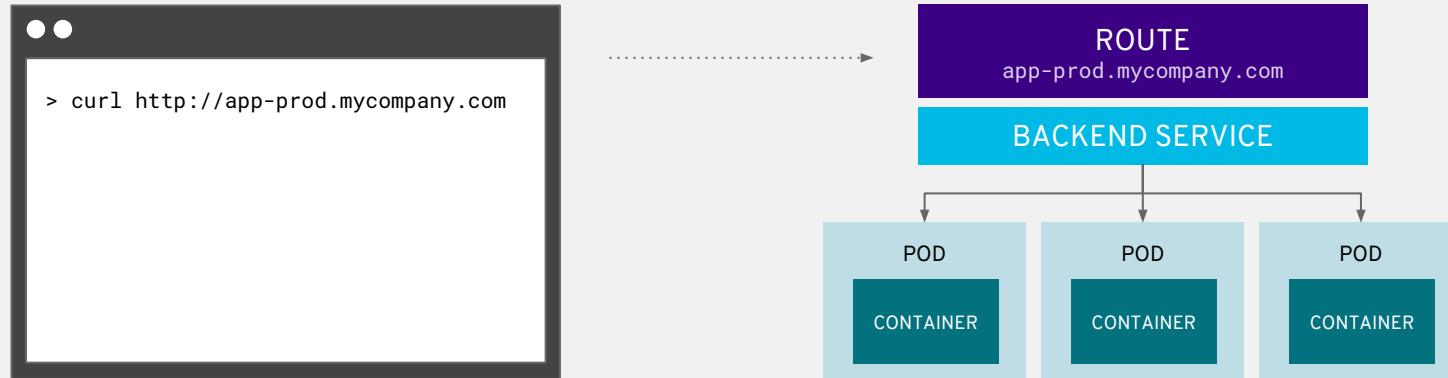
services provide internal load-balancing and service discovery across pods



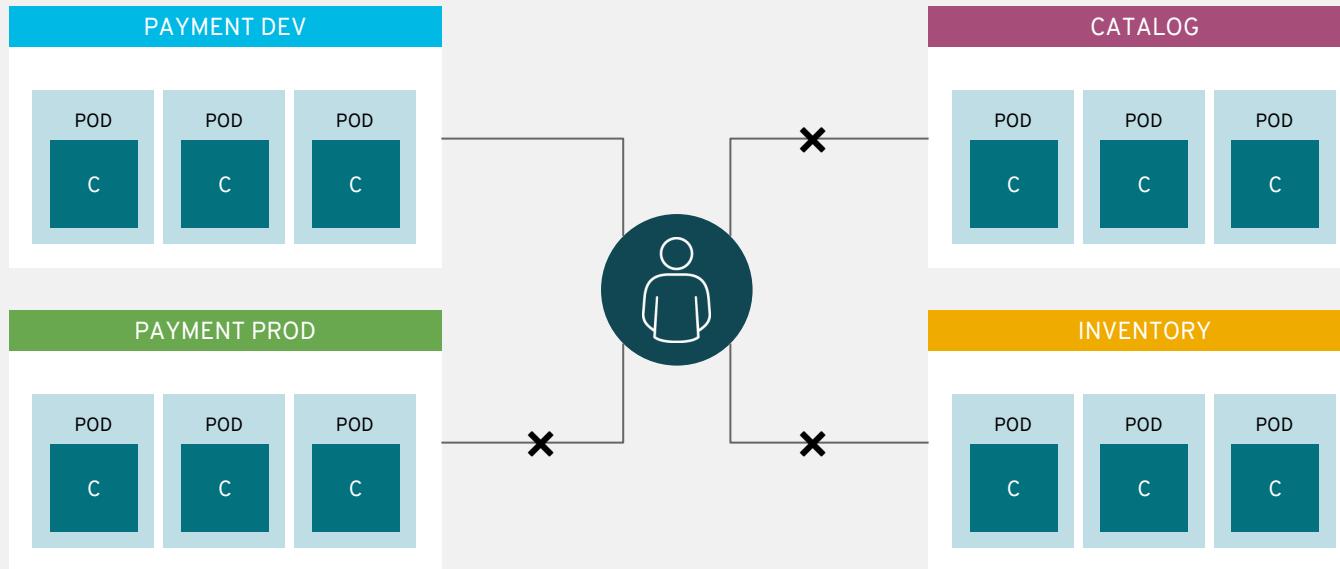
apps can talk to each other via services

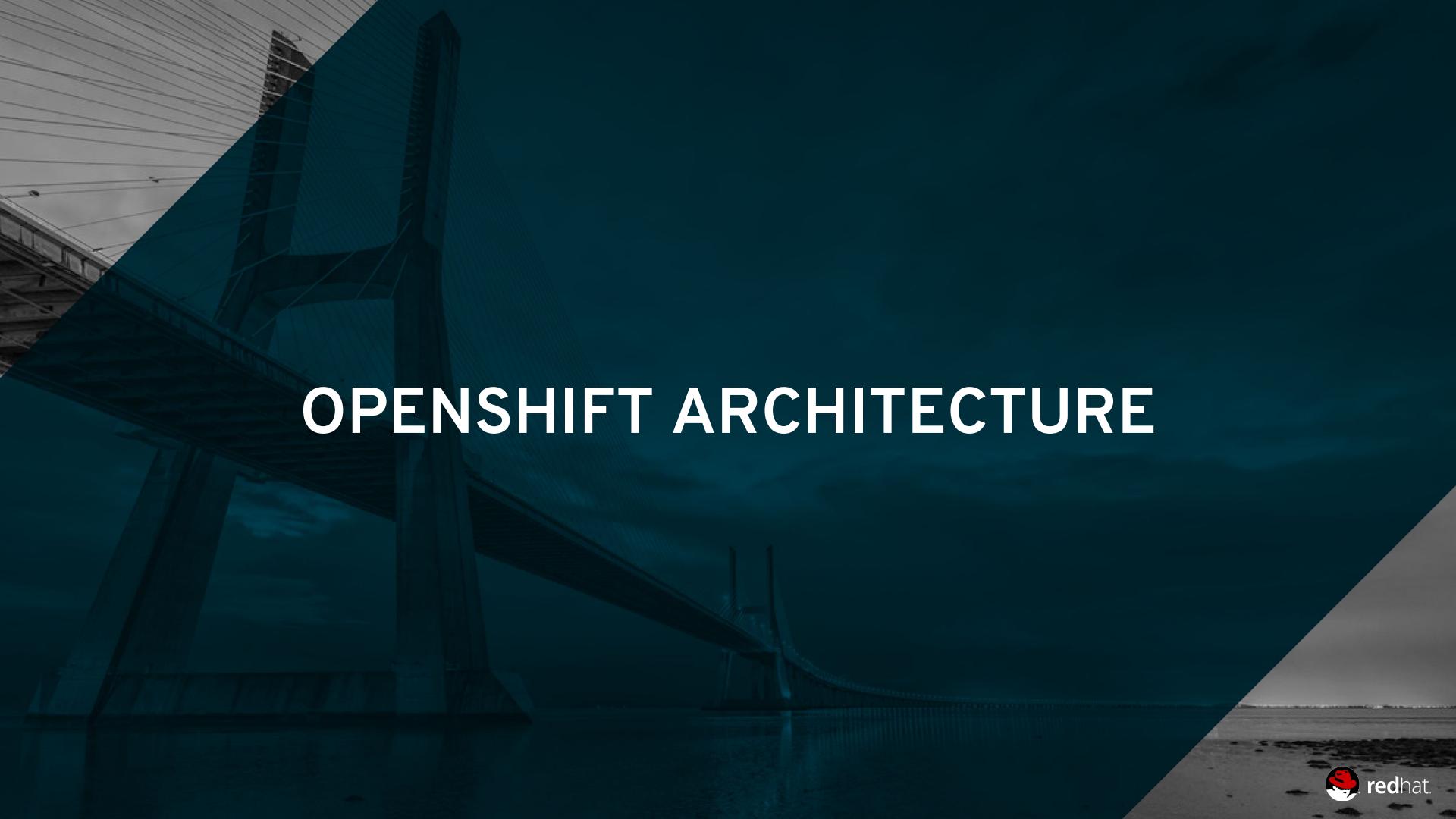


routes add services to the external load-balancer and provide readable urls for the app



projects isolate apps across environments,
teams, groups and departments



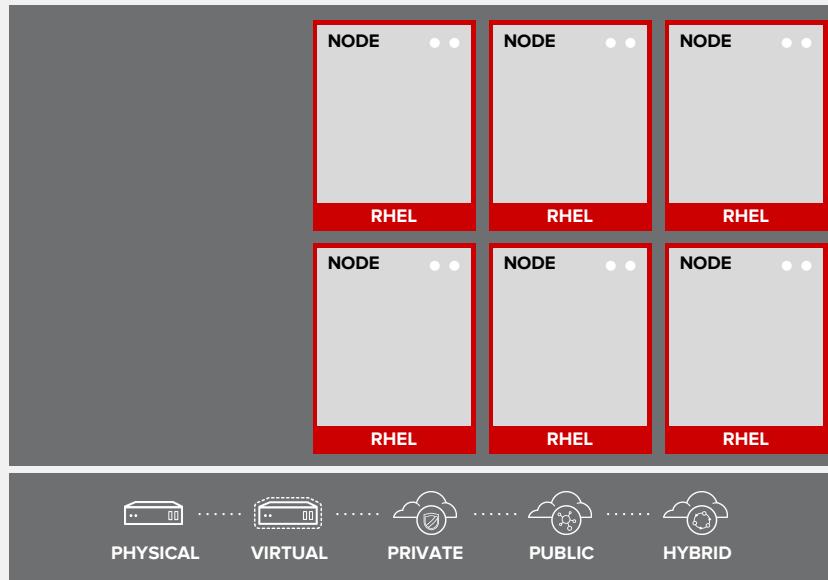


OPENSHIFT ARCHITECTURE

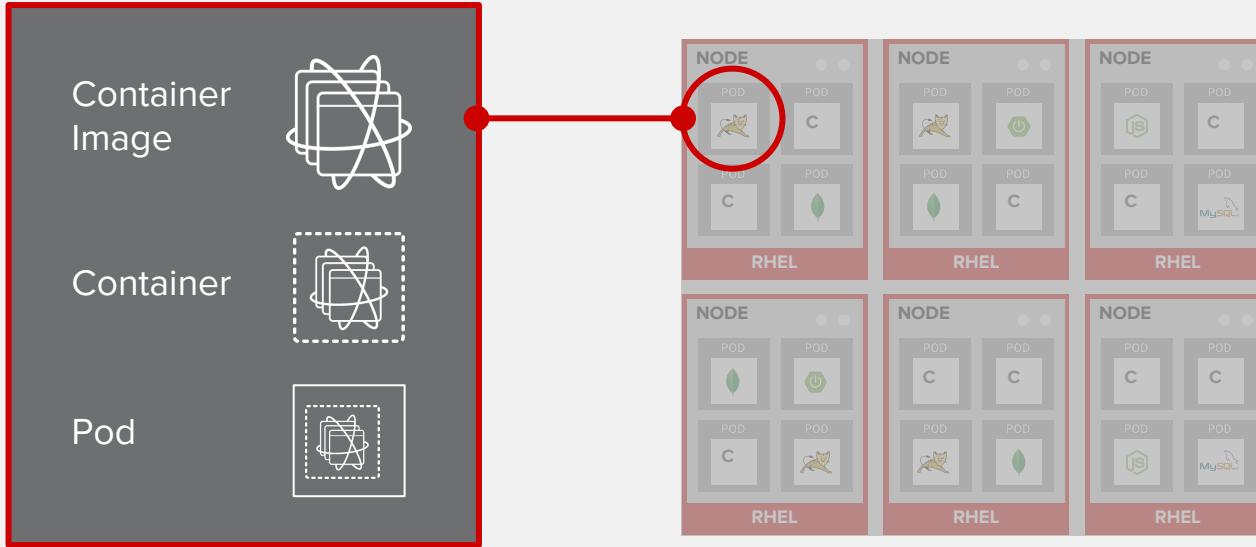
YOUR CHOICE OF INFRASTRUCTURE



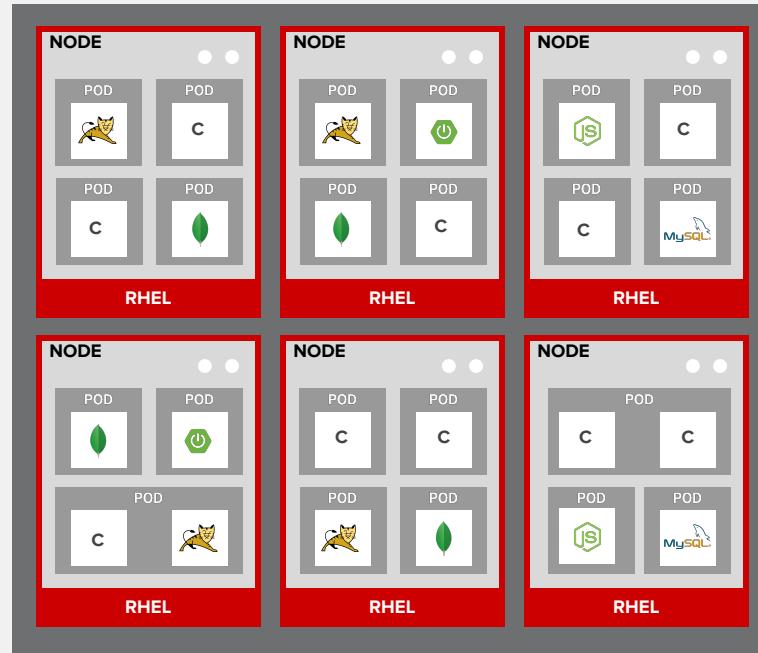
NODES RHEL INSTANCES WHERE APPS RUN



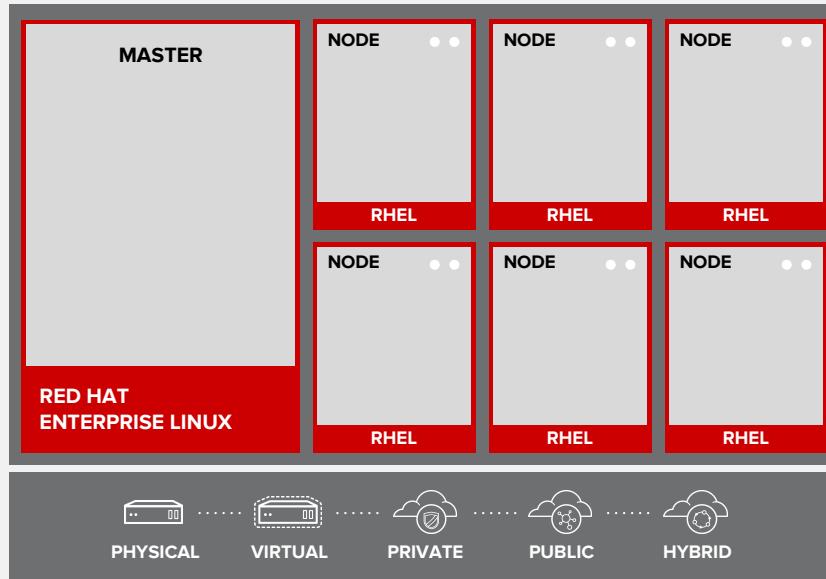
APPS RUN IN CONTAINERS



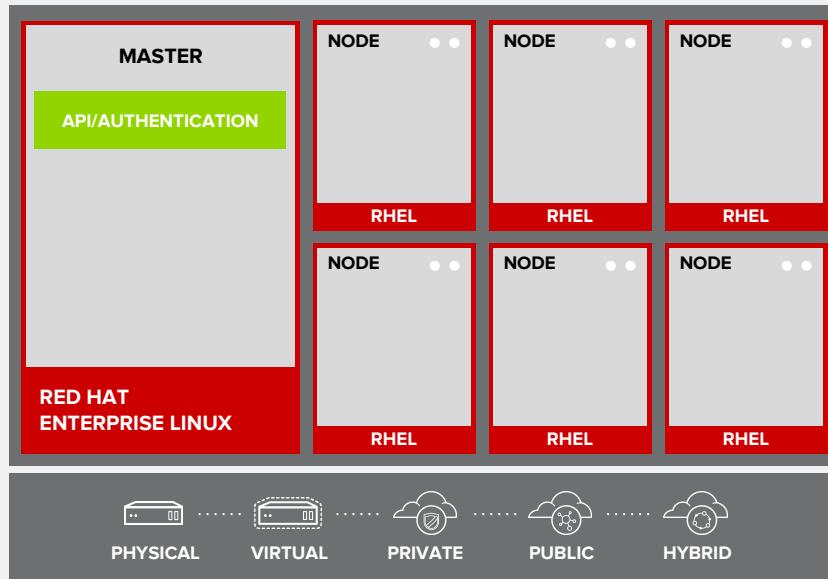
PODS ARE THE UNIT OF ORCHESTRATION



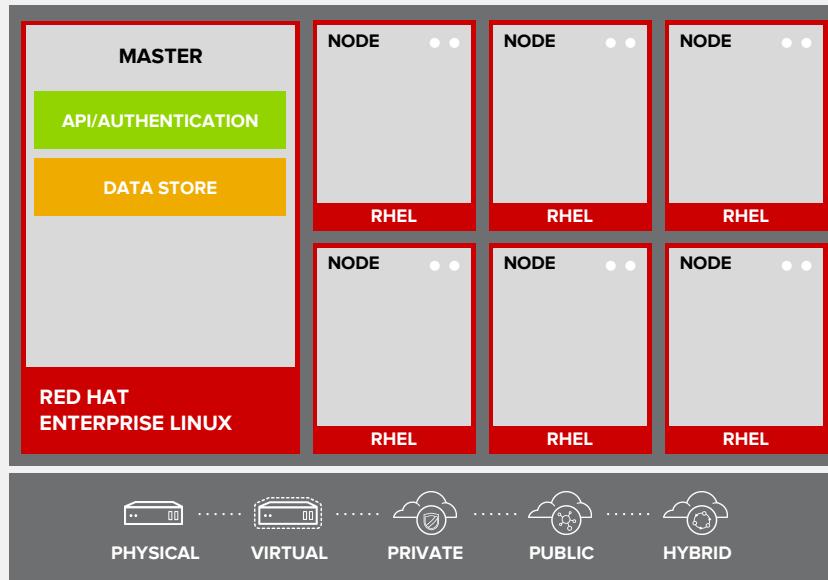
MASTERS ARE THE CONTROL PLANE



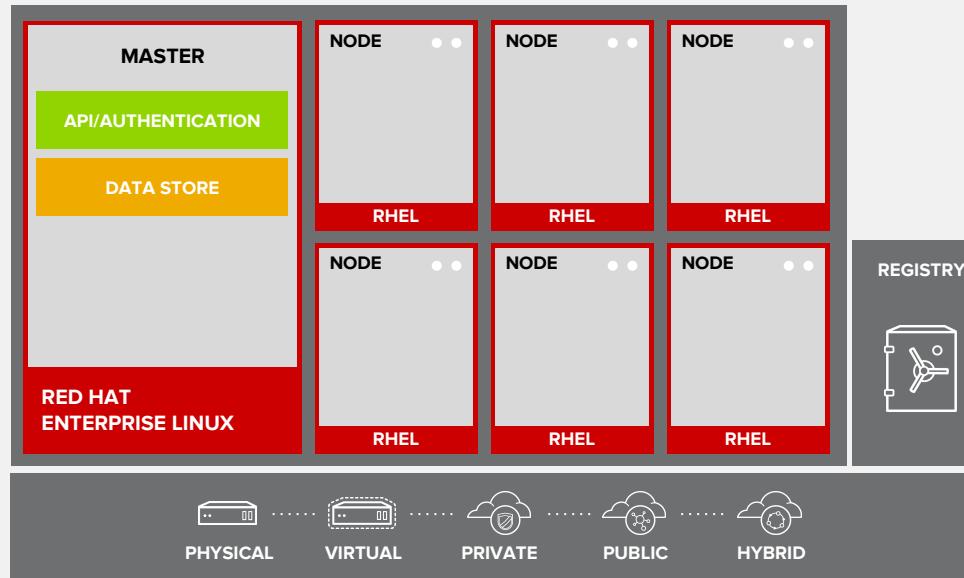
API AND AUTHENTICATION



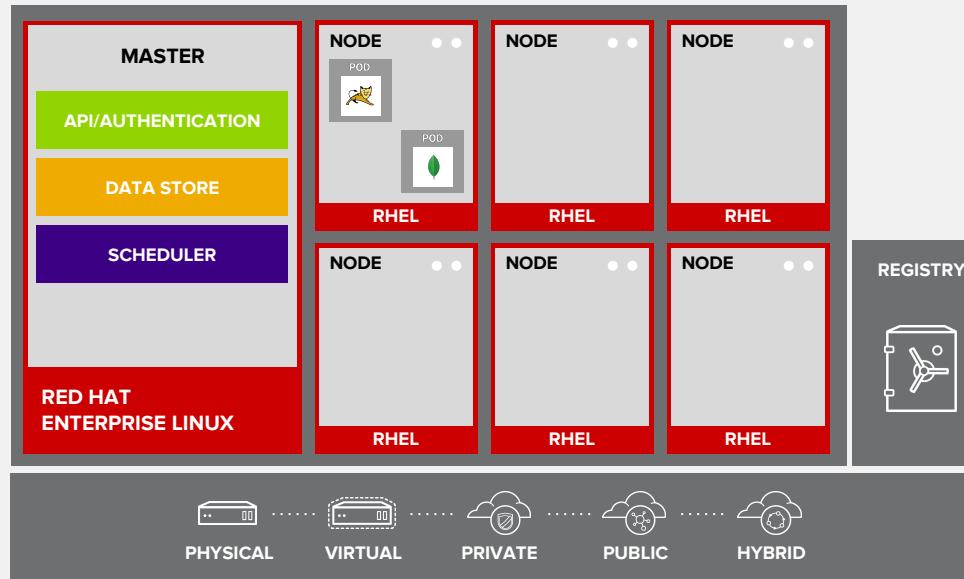
DESIRED AND CURRENT STATE



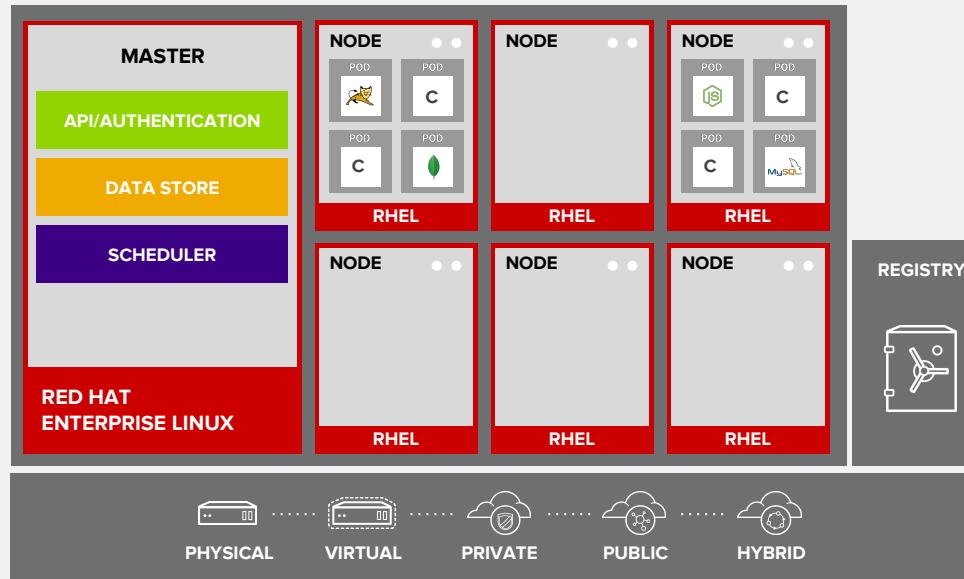
INTEGRATED CONTAINER REGISTRY



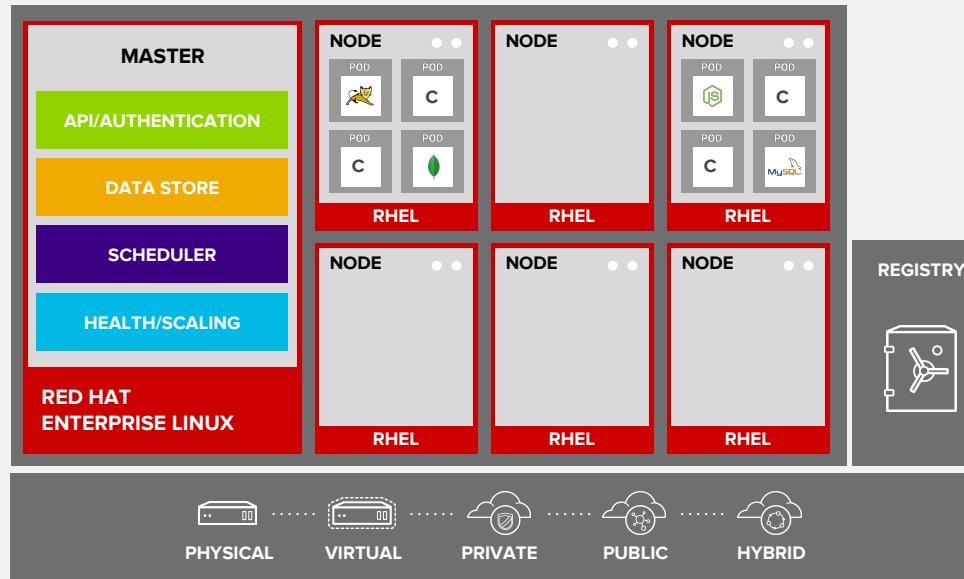
ORCHESTRATION AND SCHEDULING



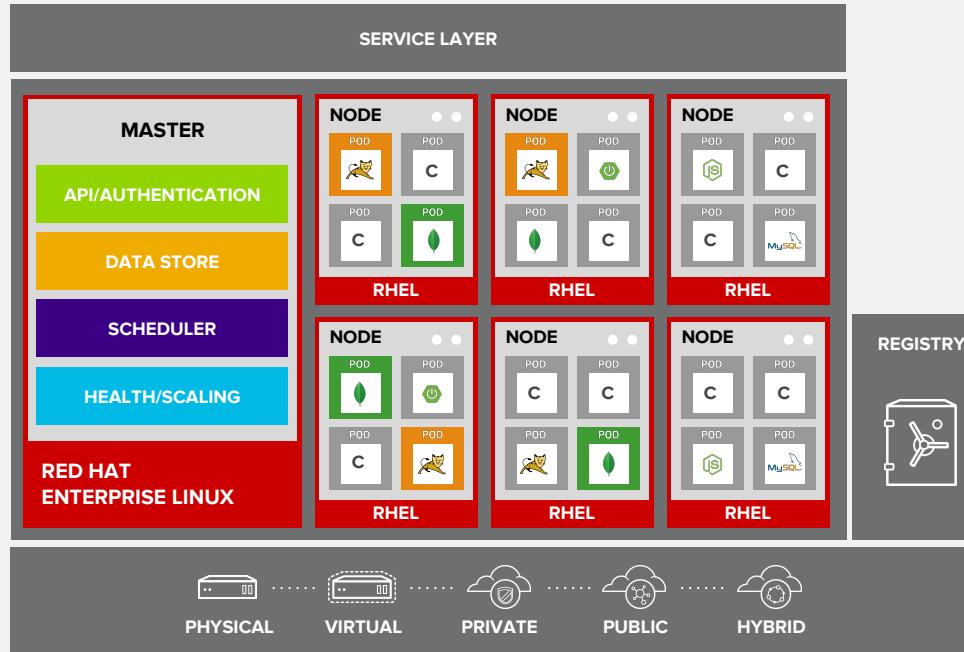
PLACEMENT BY POLICY



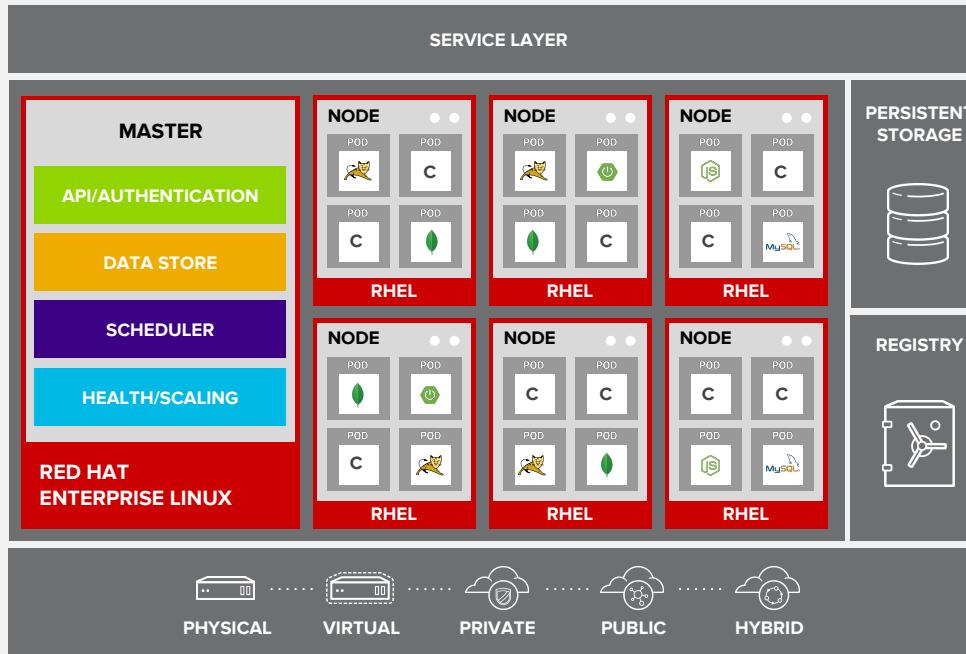
AUTOSCALING PODS



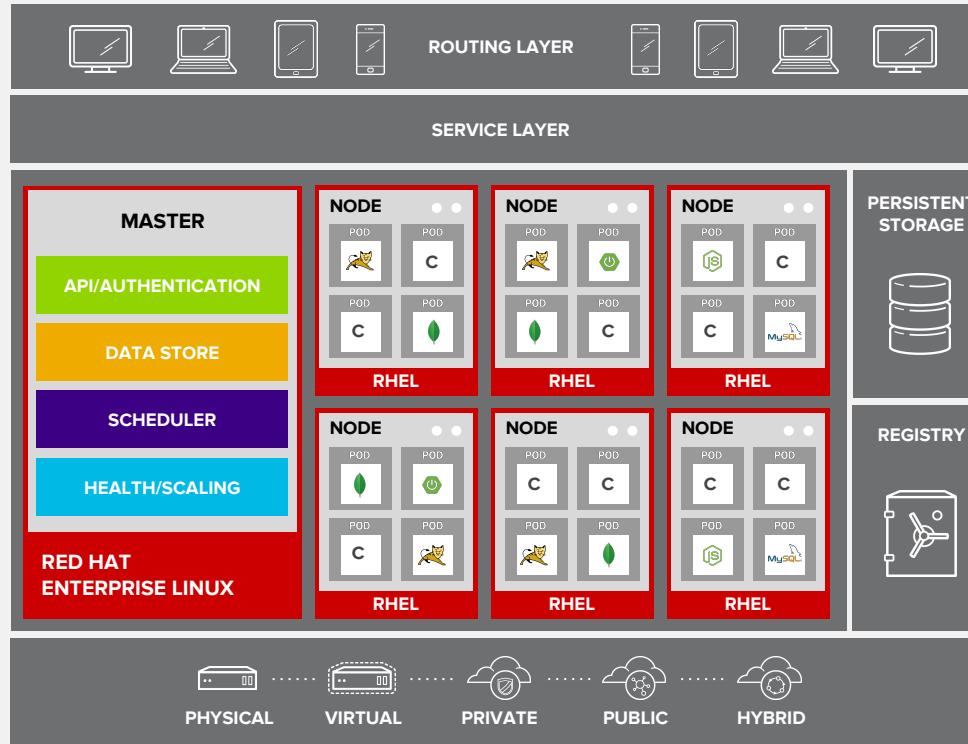
SERVICE DISCOVERY



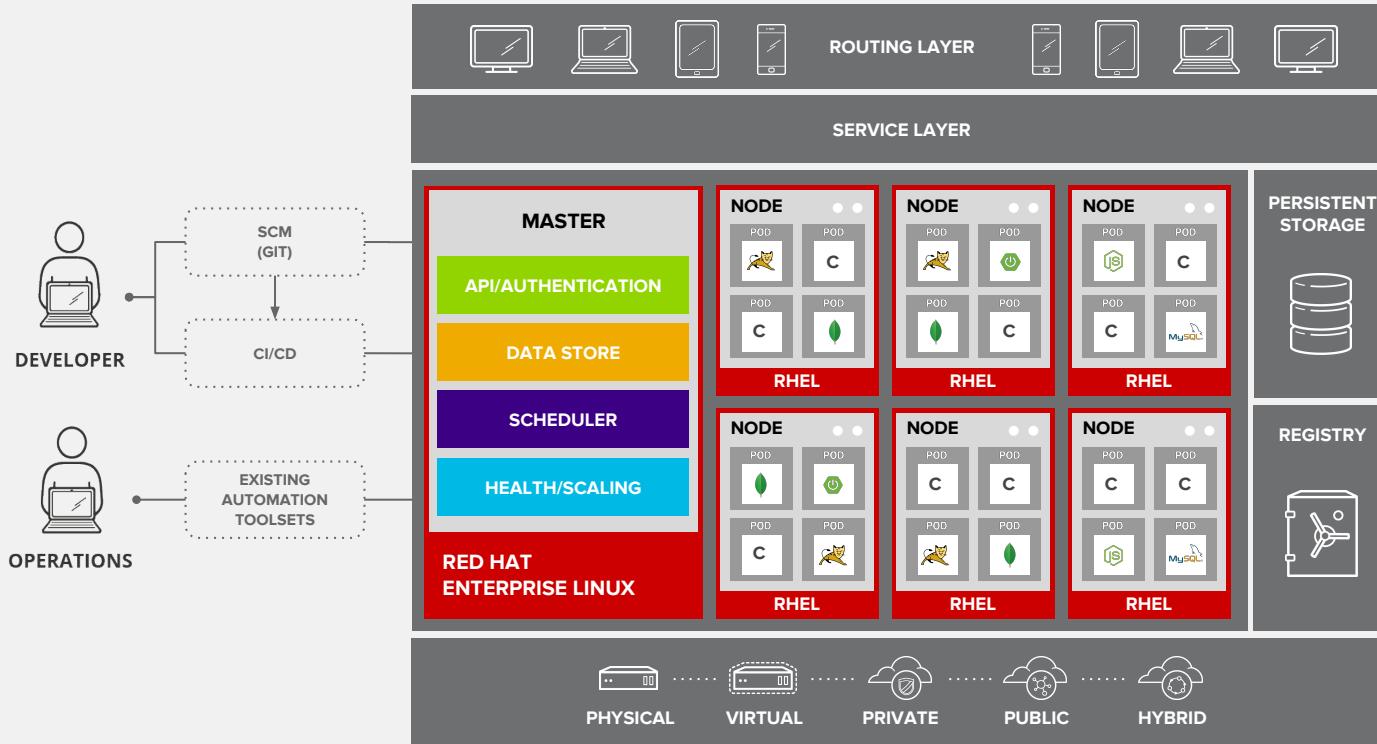
PERSISTENT DATA IN CONTAINERS

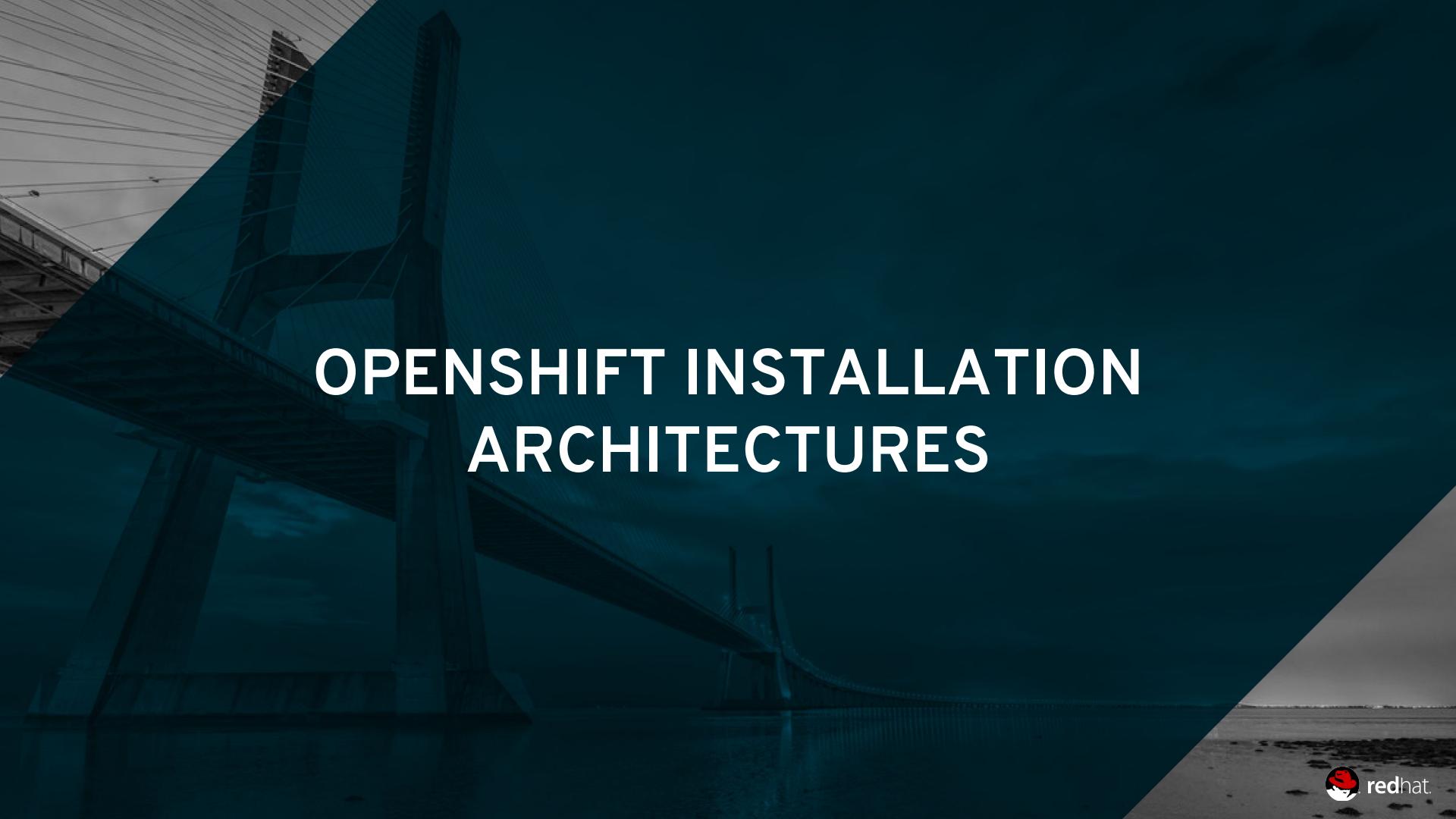


ROUTING AND LOAD-BALANCING



ACCESS VIA WEB, CLI, IDE AND API

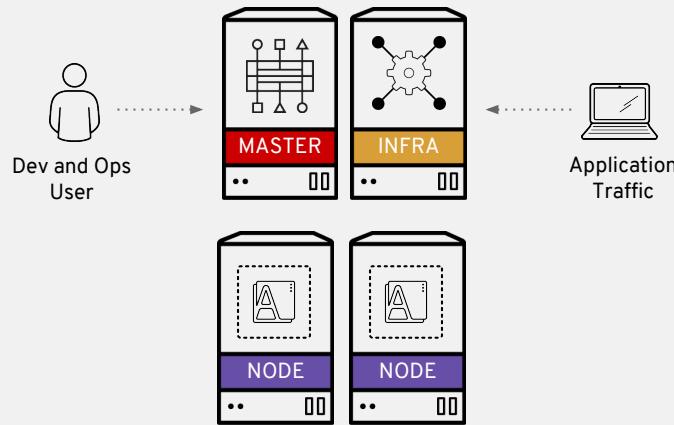




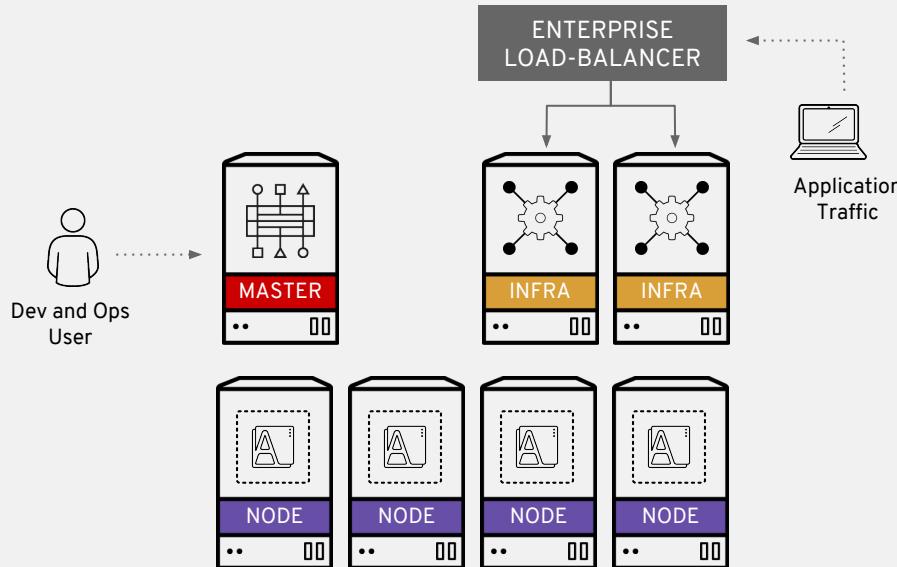
OPENSHIFT INSTALLATION ARCHITECTURES

PROOF-OF-CONCEPT ARCHITECTURE

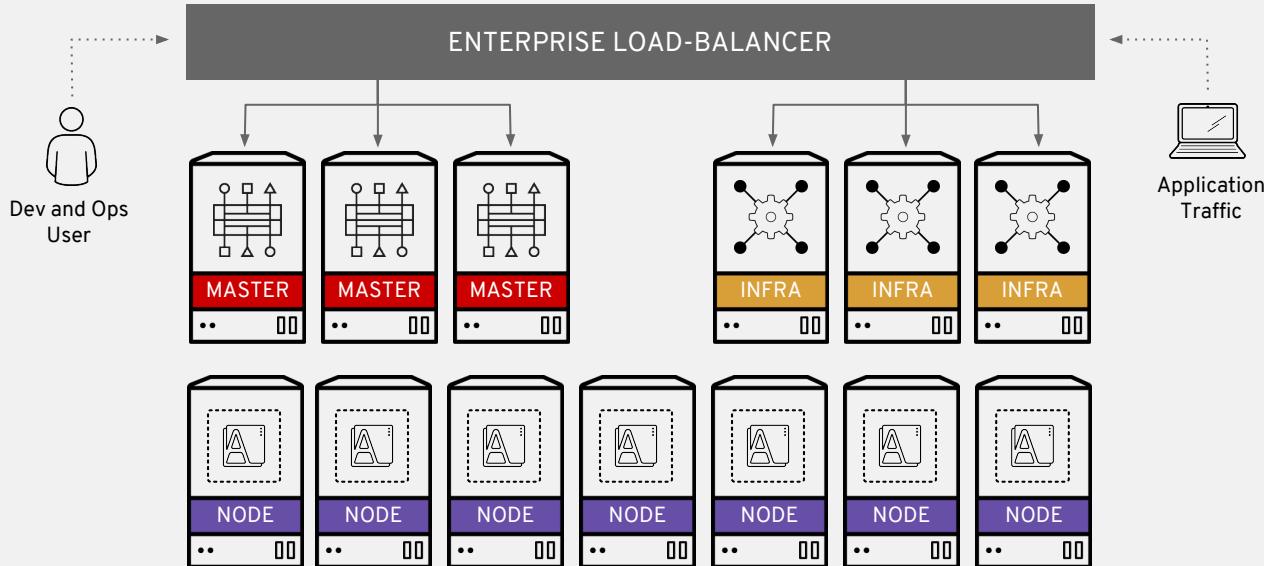
An infrastructure node is a node that is dedicated to infrastructure pods such as router, image registry, metrics, and logs



APP HIGH-AVAILABILITY ARCHITECTURE



FULL HIGH-AVAILABILITY ARCHITECTURE

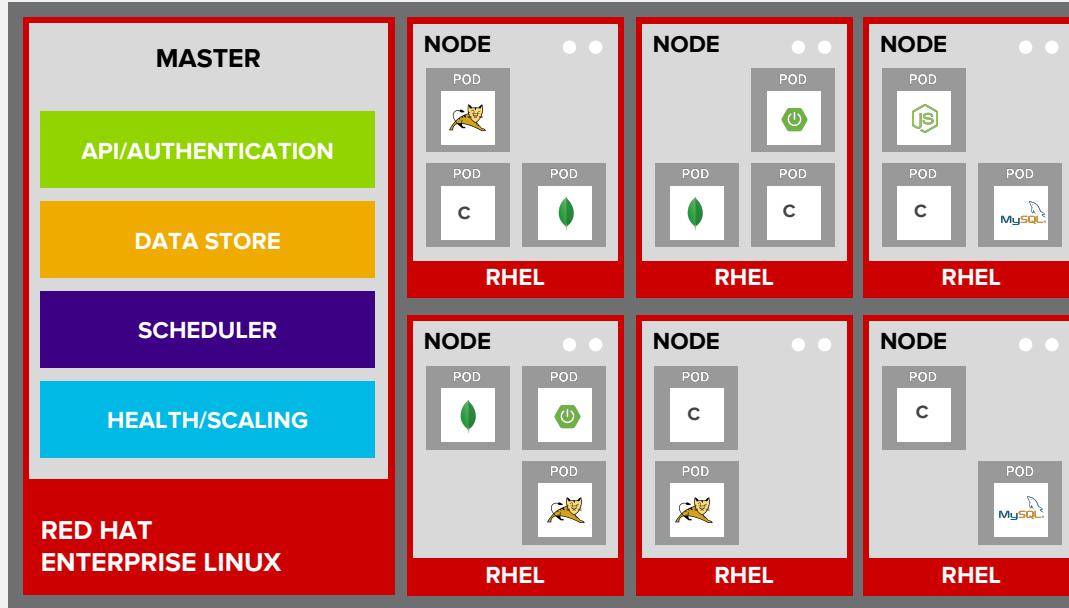




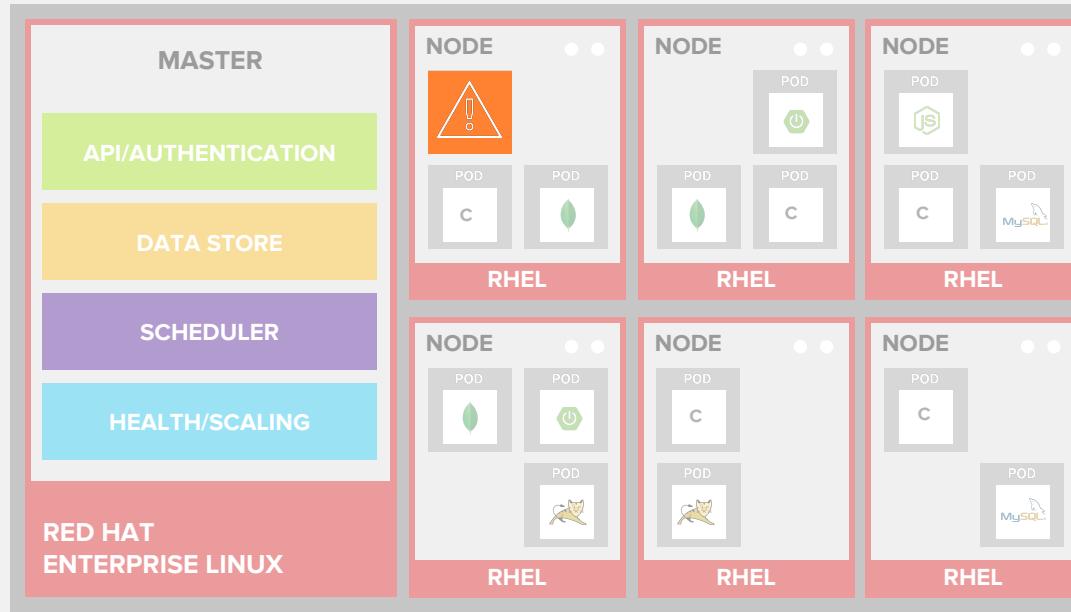
TECHNICAL DEEP DIVE

MONITORING APPLICATION HEALTH

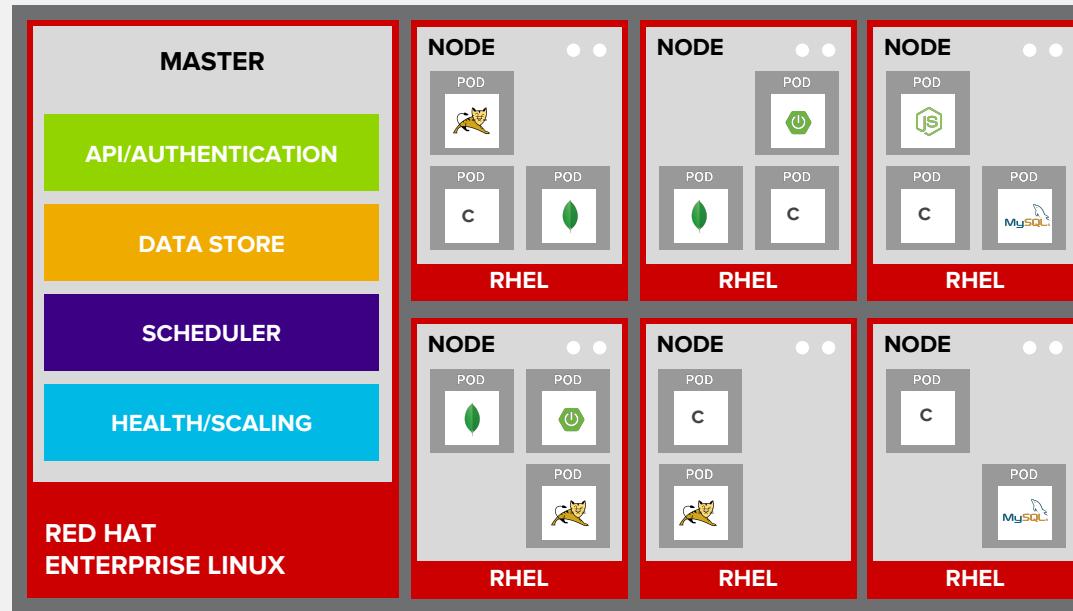
AUTO-HEALING FAILED PODS



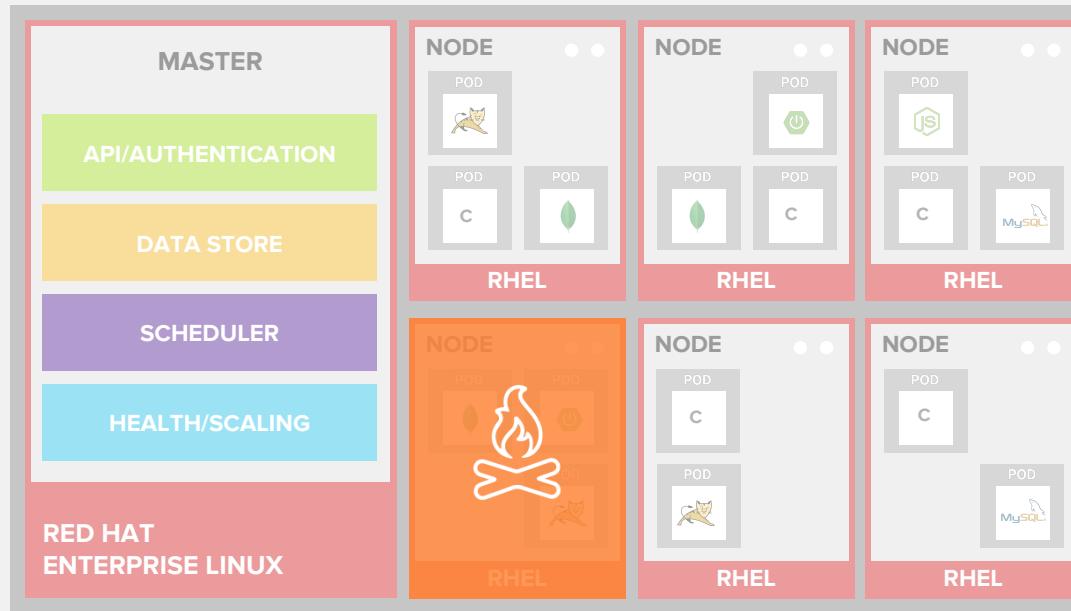
AUTO-HEALING FAILED CONTAINERS



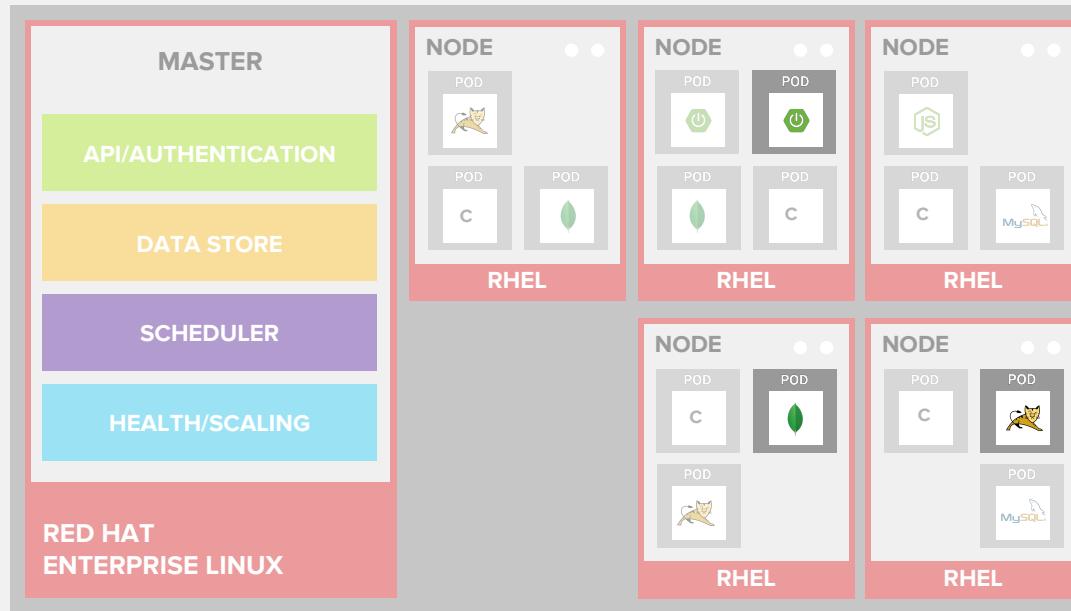
AUTO-HEALING FAILED CONTAINERS



AUTO-HEALING FAILED CONTAINERS

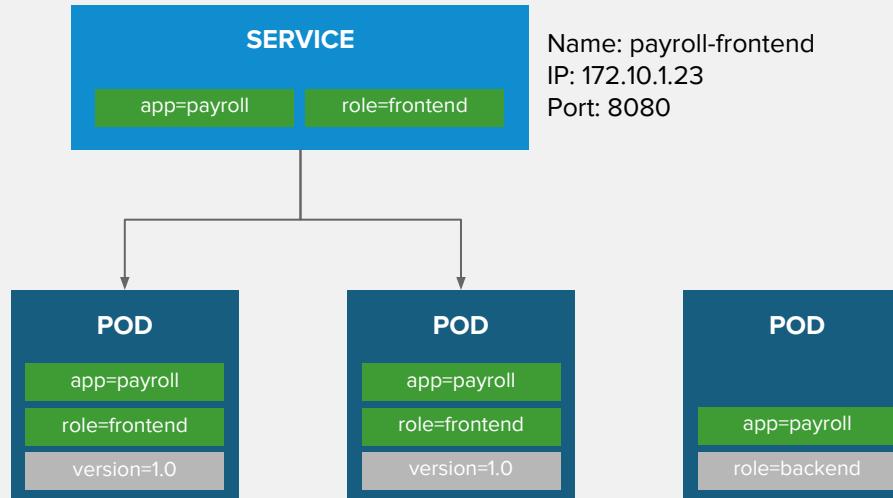


AUTO-HEALING FAILED CONTAINERS

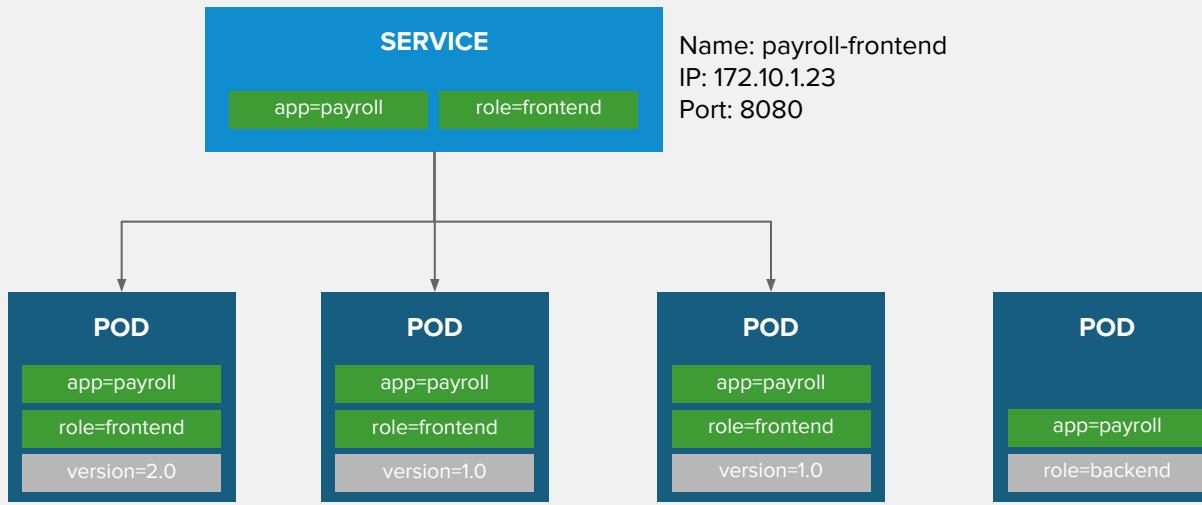


NETWORKING

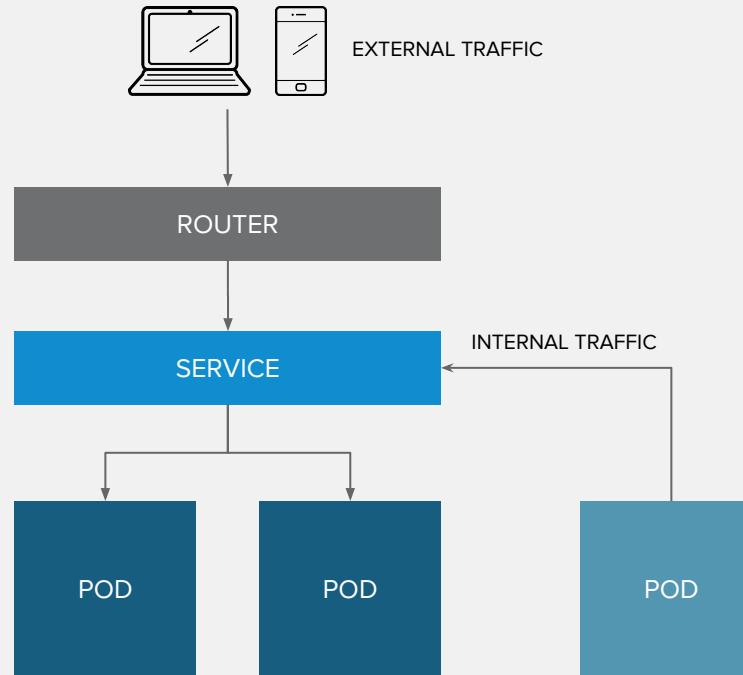
BUILT-IN SERVICE DISCOVERY INTERNAL LOAD-BALANCING



BUILT-IN SERVICE DISCOVERY INTERNAL LOAD-BALANCING

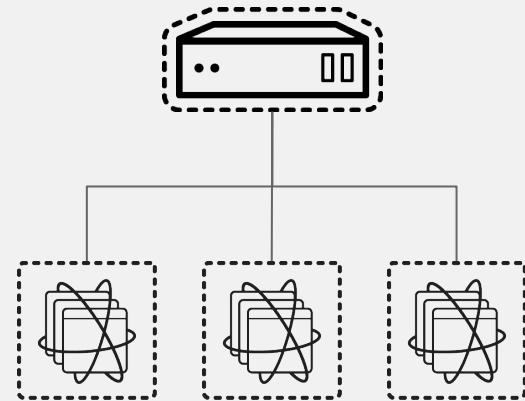


ROUTE EXPOSES SERVICES EXTERNALLY



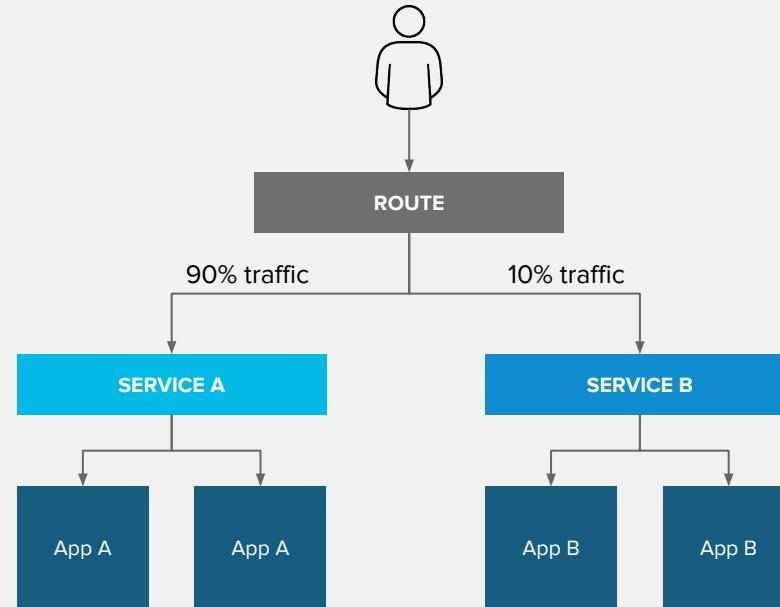
ROUTING AND EXTERNAL LOAD-BALANCING

- 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



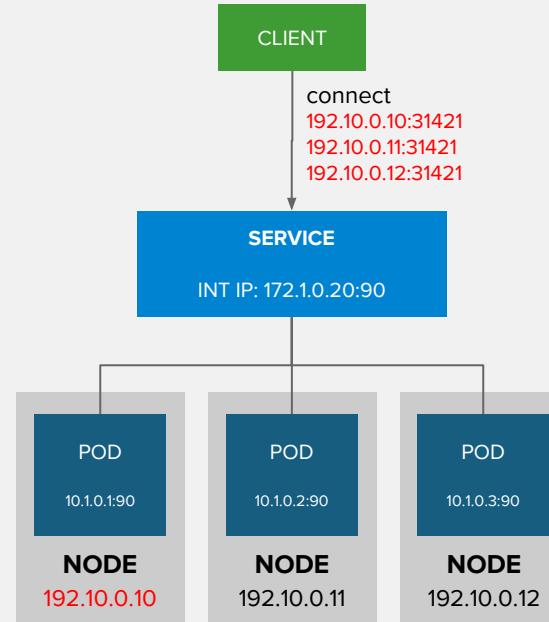
ROUTE SPLIT TRAFFIC

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



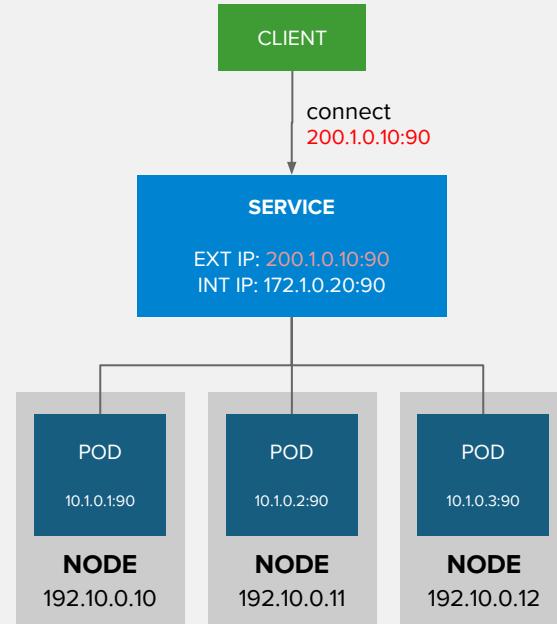
EXTERNAL TRAFFIC TO A SERVICE ON A RANDOM PORT WITH NODEPORT

- 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



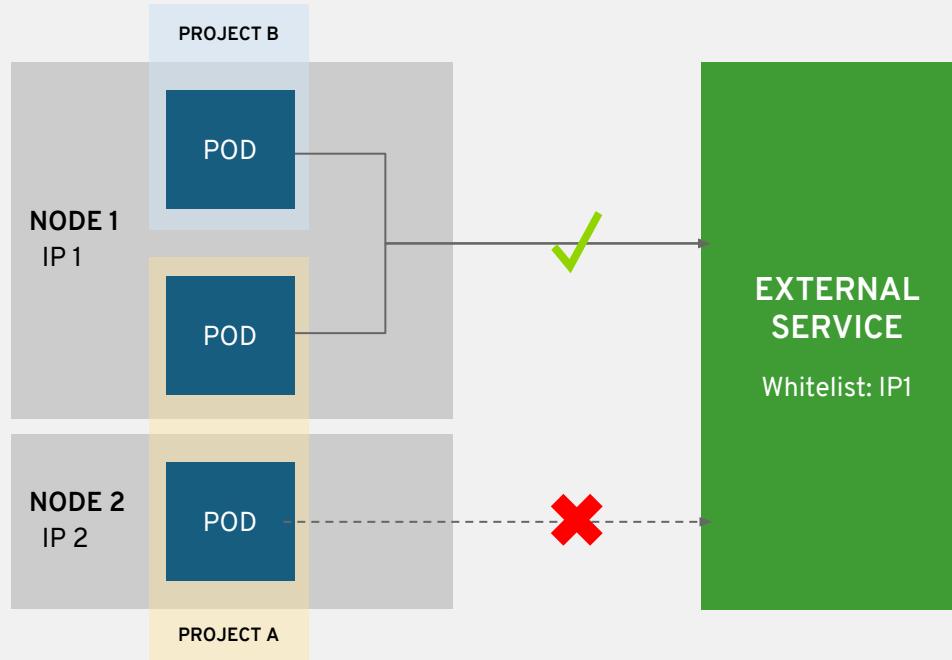
EXTERNAL TRAFFIC TO A SERVICE ON ANY PORT WITH INGRESS

- 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



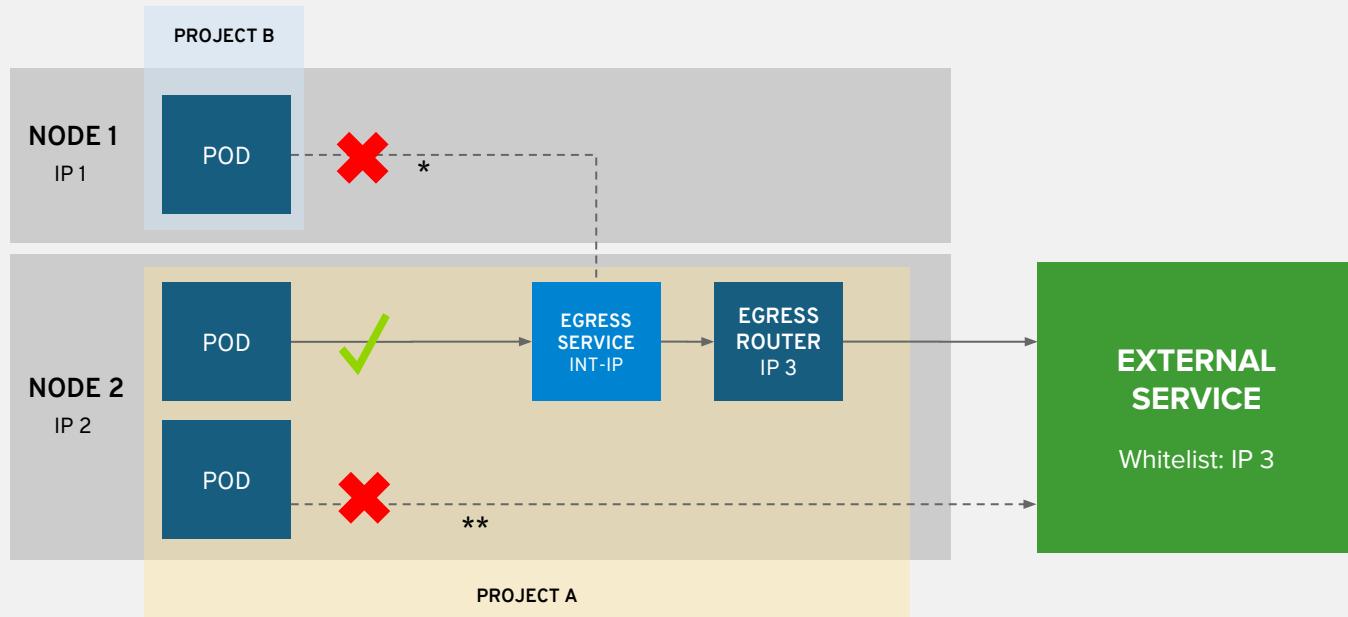
CONTROLLING EGRESS TRAFFIC

Default Kubernetes Behaviour



CONTROLLING EGRESS TRAFFIC

Egress Router

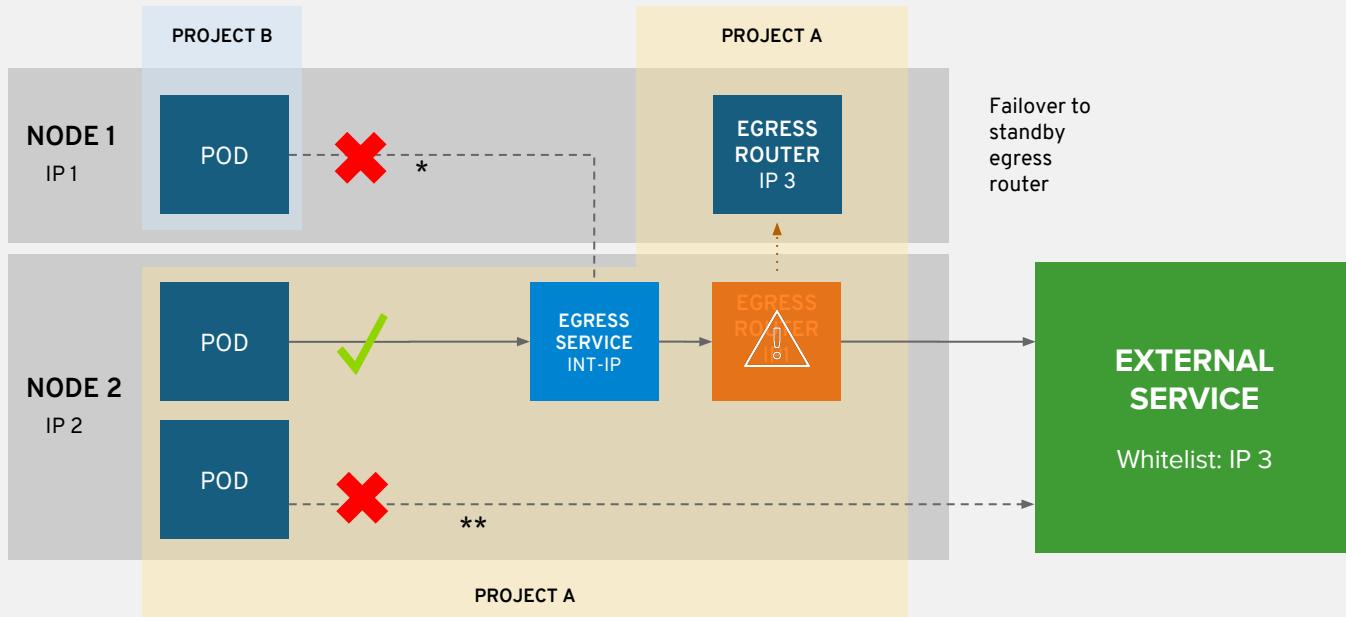


* Blocked by multi-tenant network plugin

** Blocked by external service

CONTROLLING EGRESS TRAFFIC

Egress Router

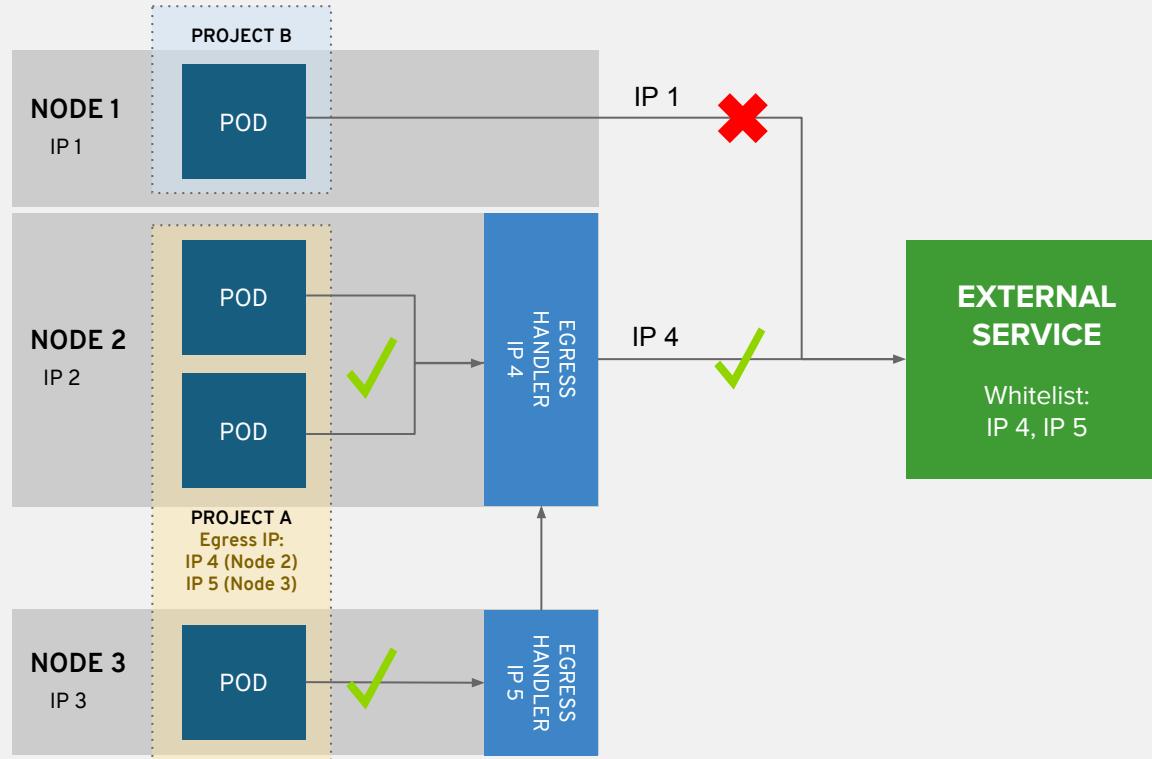


* Blocked by multi-tenant network plugin

** Blocked by external service

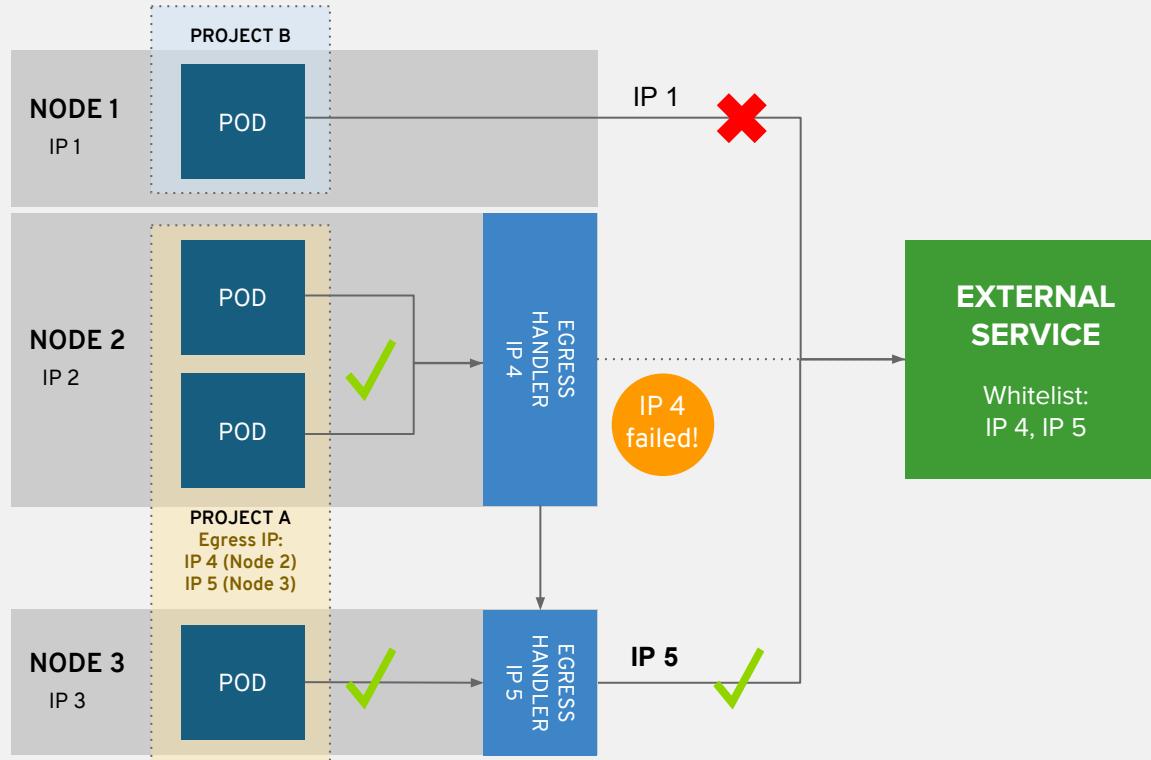
CONTROLLING EGRESS TRAFFIC

Egress IP High Availability (multiple IPs)



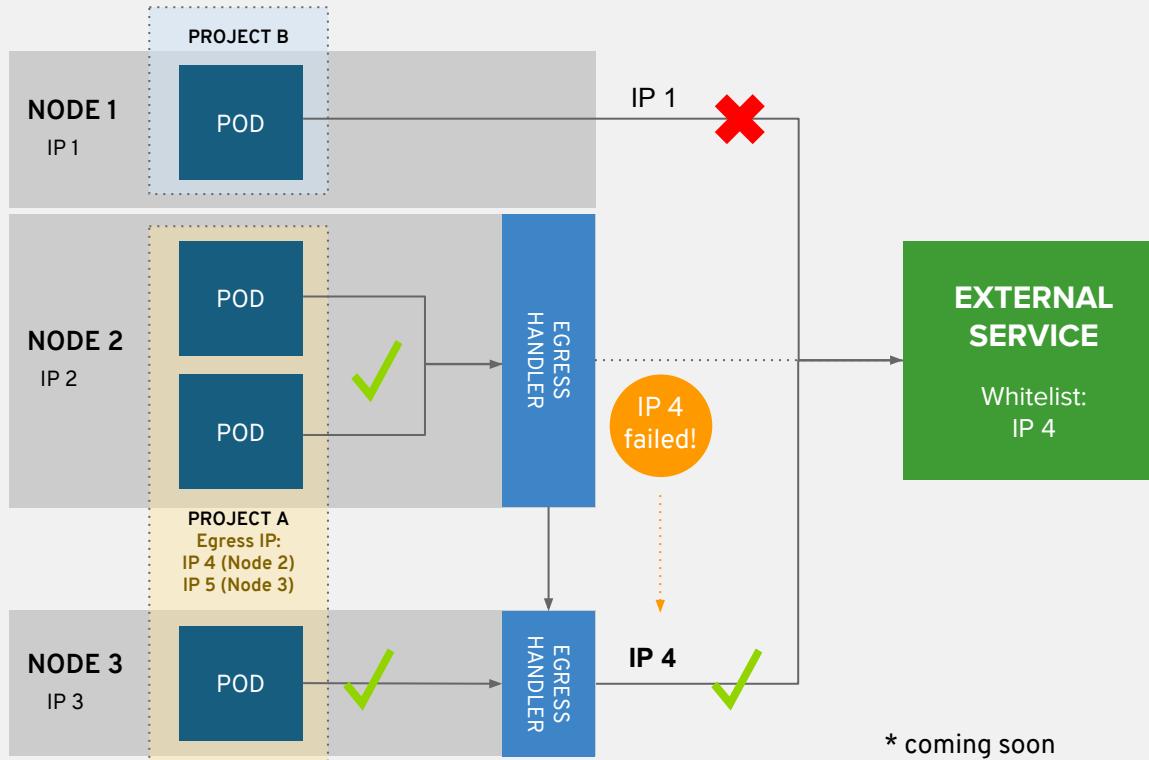
CONTROLLING EGRESS TRAFFIC

Egress IP High Availability (multiple IPs)



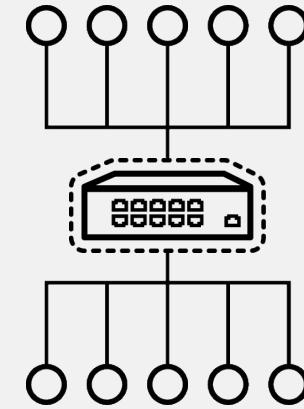
CONTROLLING EGRESS TRAFFIC

Egress IP High Availability (single IP)*

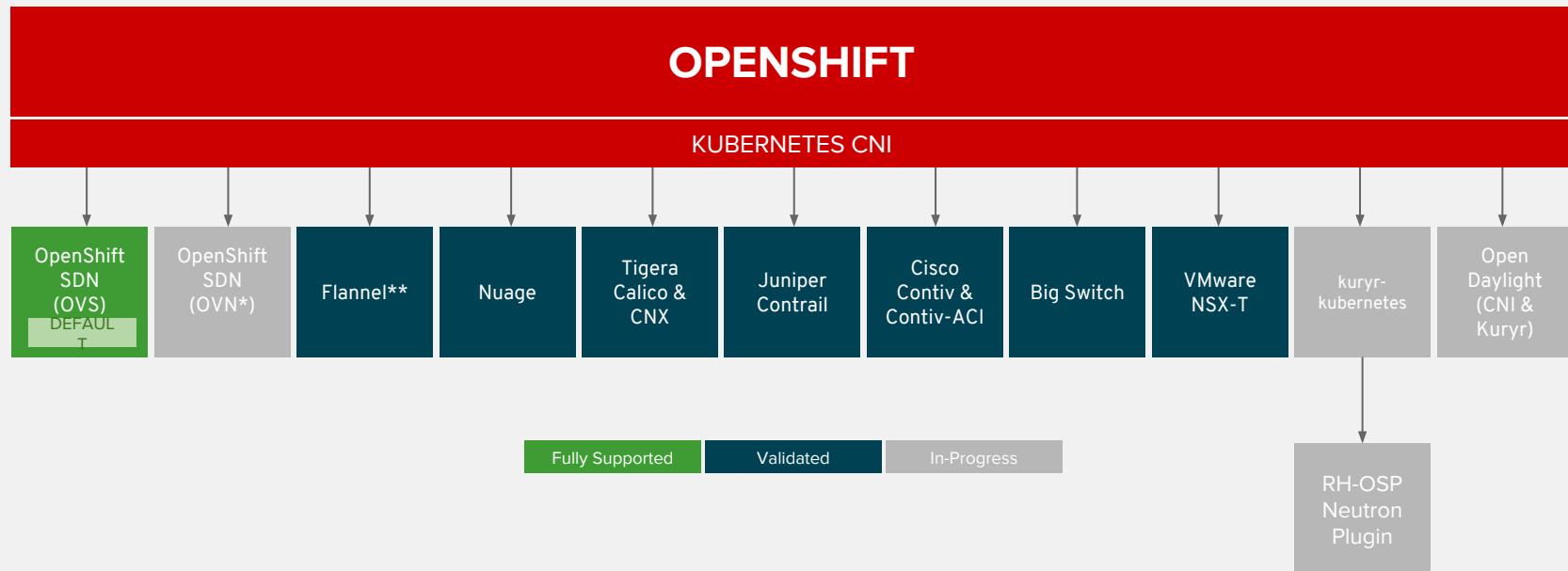


OPENShift NETWORKING

- Built-in internal DNS to reach services by name
- Split DNS is supported via SkyDNS
 - Master answers DNS queries for internal services
 - Other name servers 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



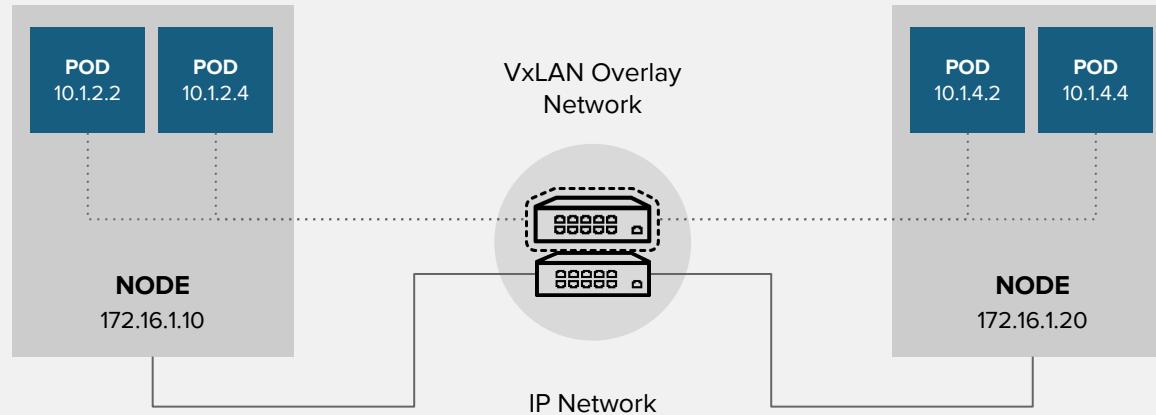
OPENShift NETWORK PLUGINS



* Coming as default in OCP 4.1

** 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)

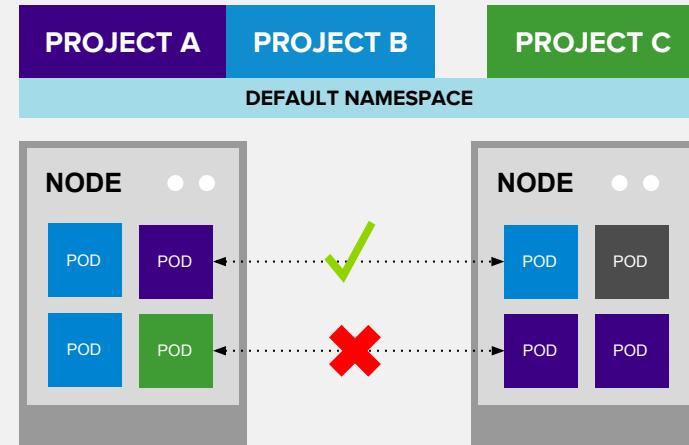
- All pods can communicate with each other across projects

MULTI-TENANT NETWORK

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

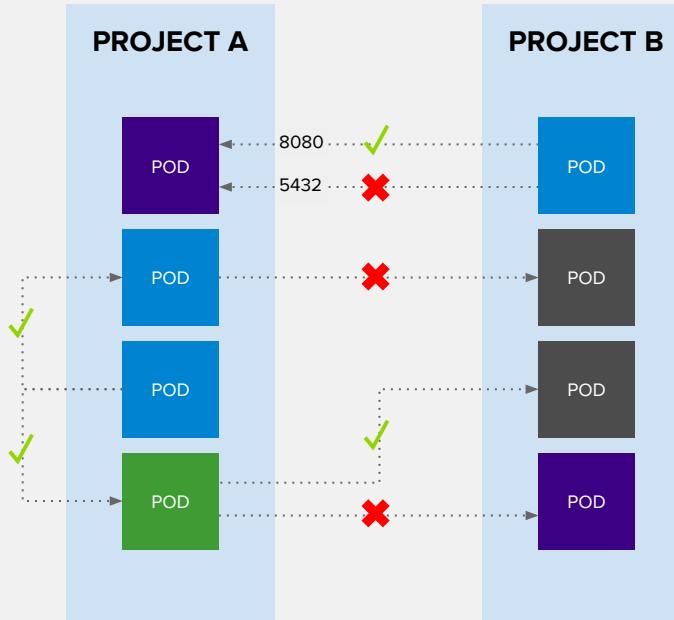
NETWORK POLICY

- Granular policy-based isolation



Multi-Tenant Network

OPENShift SDN - NETWORK POLICY



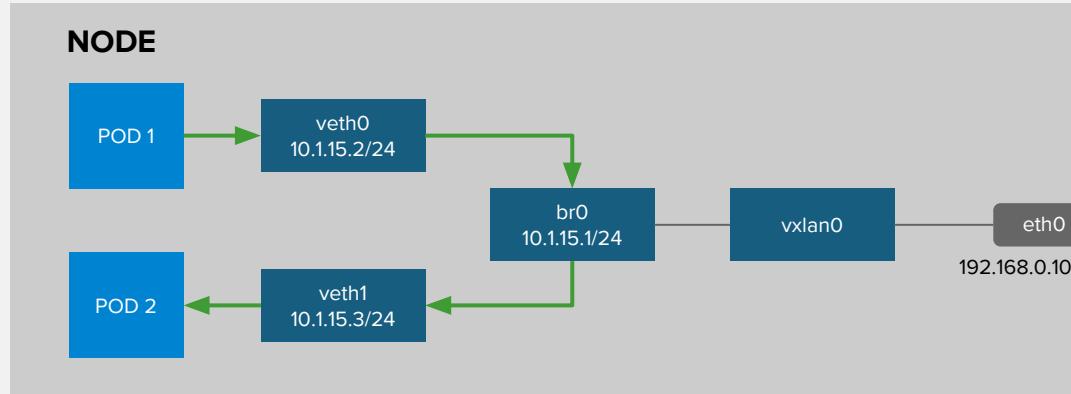
Example Policies

- 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
```

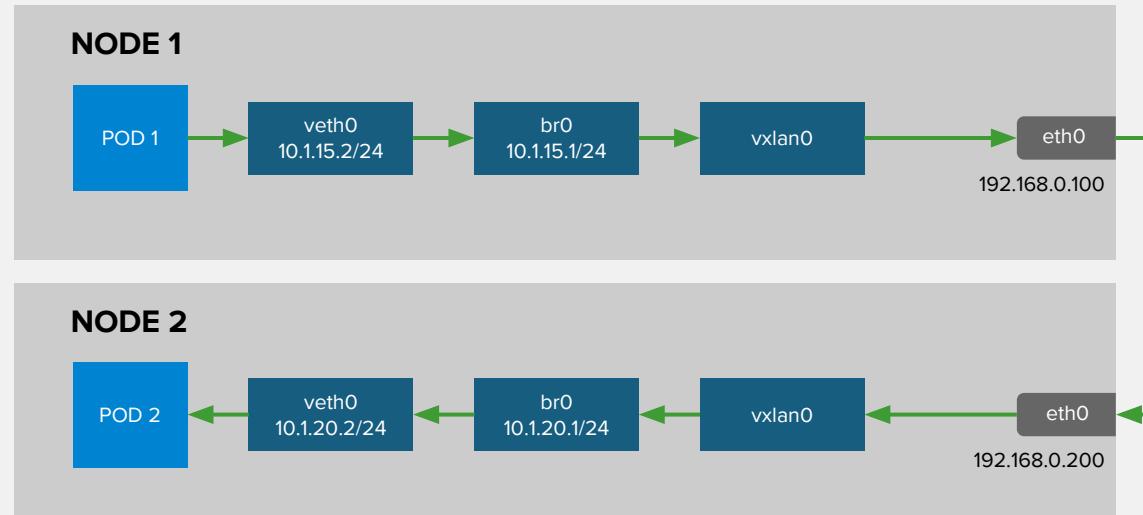
OPENShift SDN - OVS PACKET FLOW

Container to Container on the Same Host



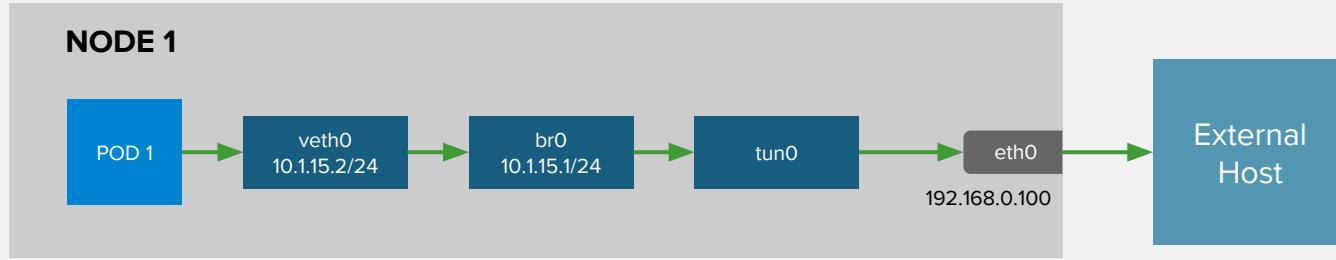
OPENShift SDN - OVS PACKET FLOW

Container to Container on the Different Hosts

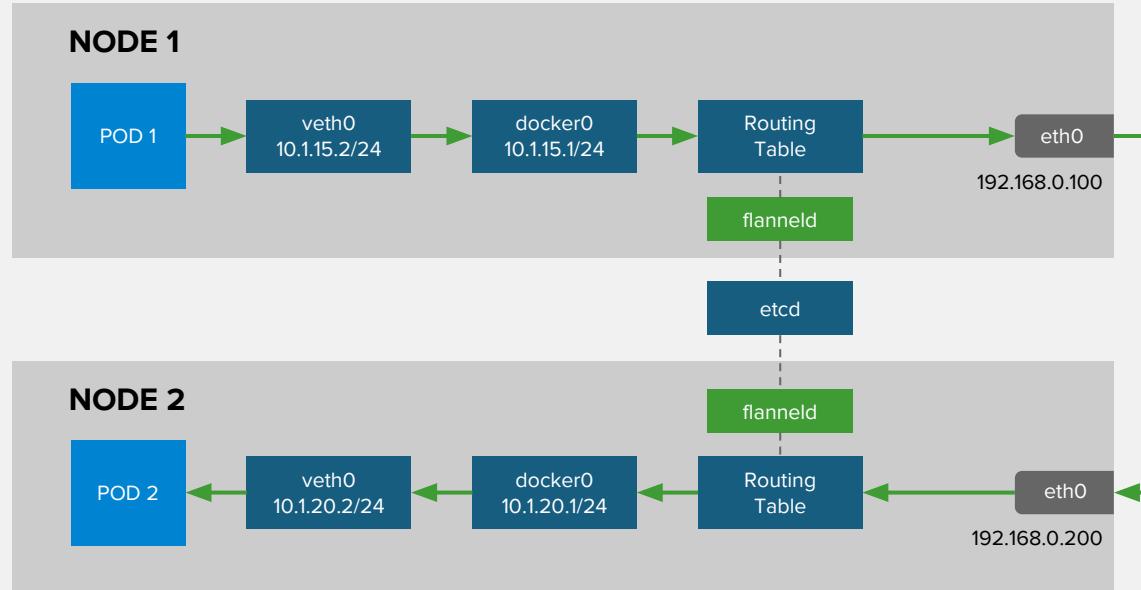


OPENShift SDN - OVS PACKET FLOW

Container Connects to External Host



OPENShift SDN WITH FLANNEL FOR OPENSTACK

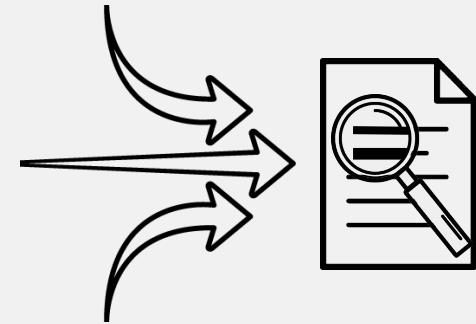


Flannel is minimally verified and is supported only and exactly as deployed in the OpenShift on OpenStack reference architecture <https://access.redhat.com/articles/2743631>

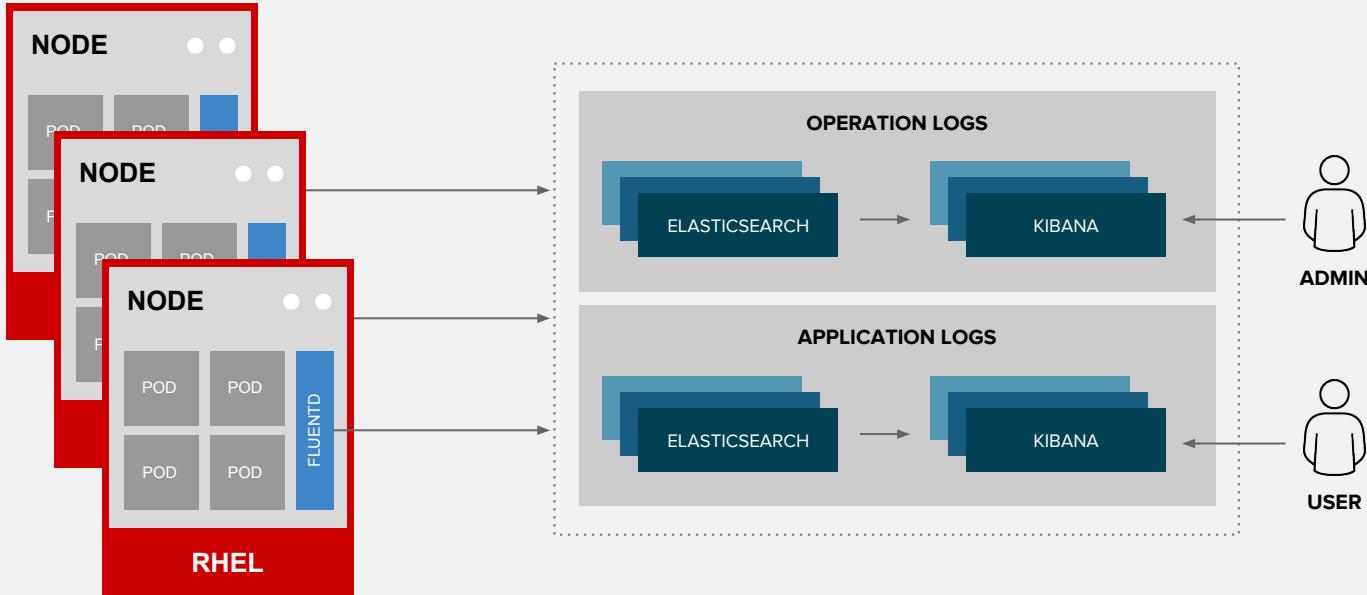
LOGGING & METRICS

CENTRAL LOG MANAGEMENT WITH EFK

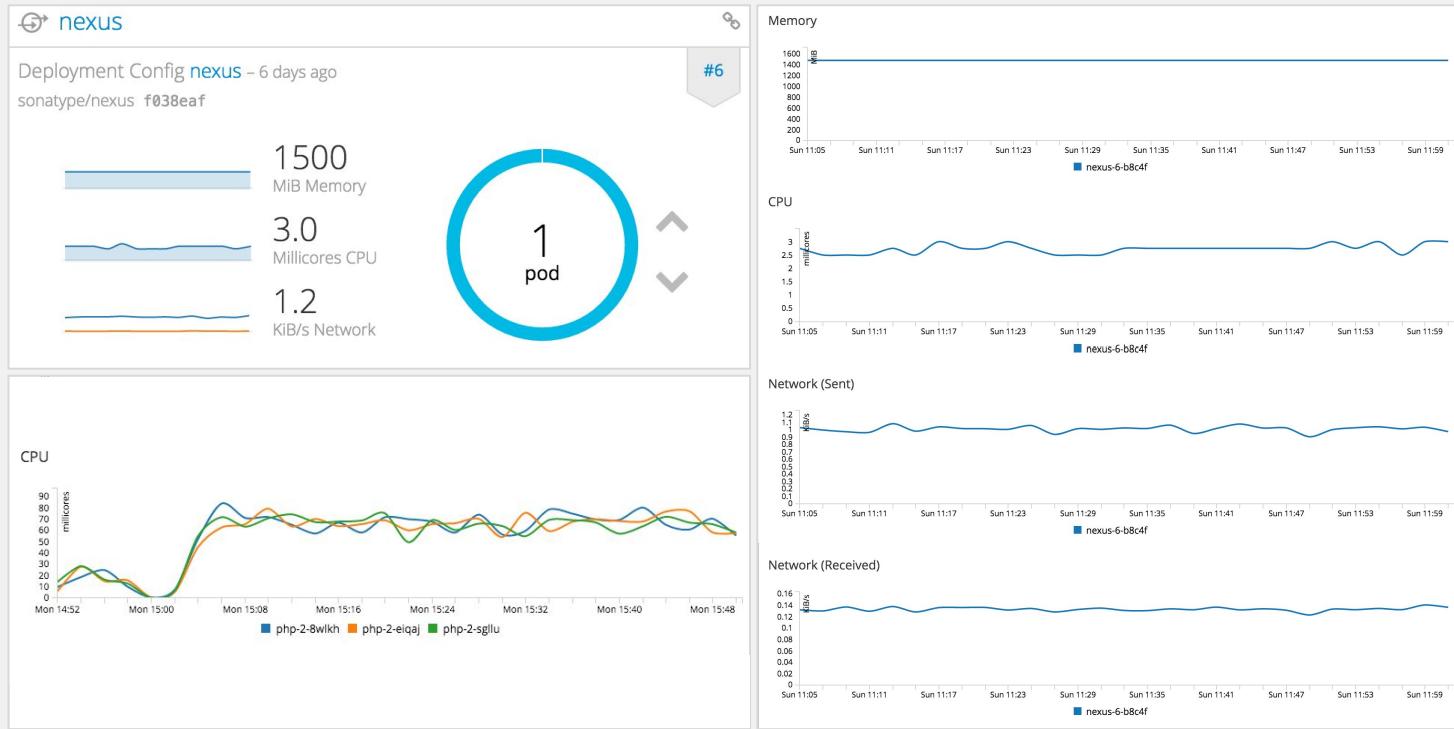
- EFK stack to aggregate logs for hosts and applications
 - **Elasticsearch:** a search and analytics engine to store logs
 - **Fluentd:** gathers logs and sends to Elasticsearch.
 - **Kibana:** A web UI for Elasticsearch.
- Access control
 - Cluster administrators can view all logs
 - Users can only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



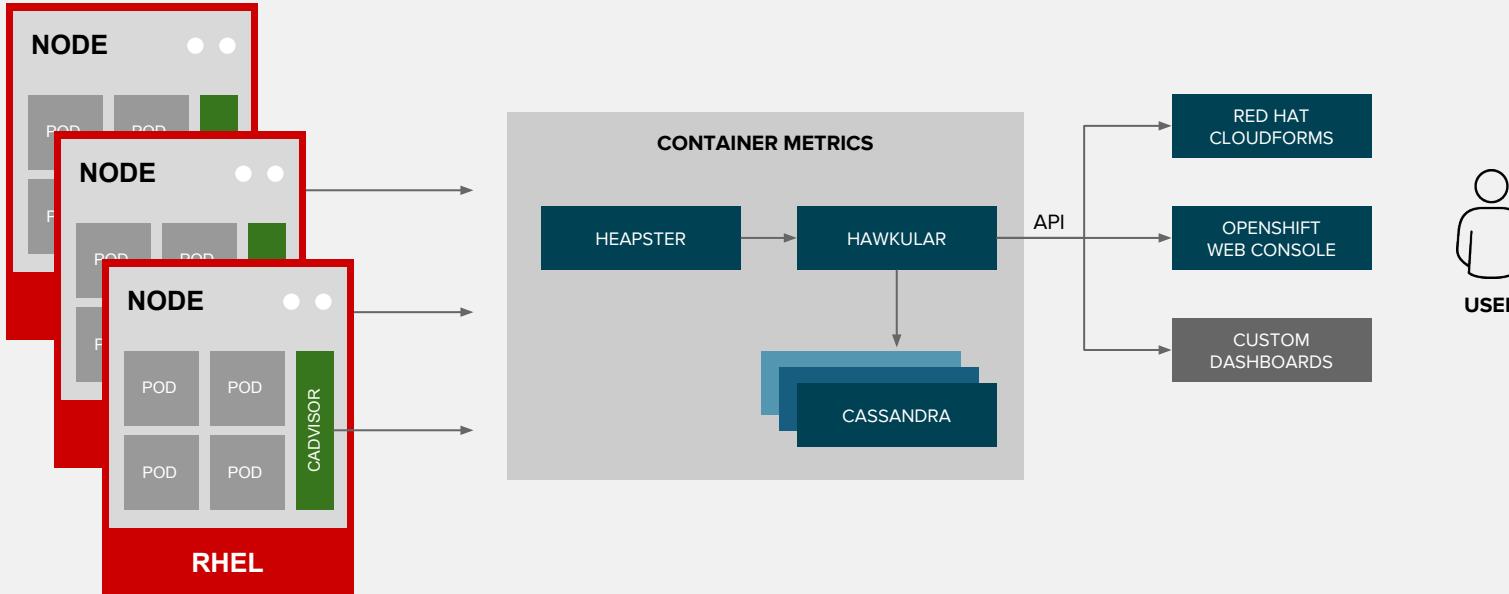
CENTRAL LOG MANAGEMENT WITH EFK



CONTAINER METRICS



CONTAINER METRICS



SECURITY

AUTOMATED & INTEGRATED SECURITY



CONTROL

Application Security

Container Content

CI/CD Pipeline

Container Registry

Deployment Policies



DEFEND

Infrastructure

Container Platform

Container Host Multi-tenancy

Network Isolation

Storage

Audit & Logging

API Management

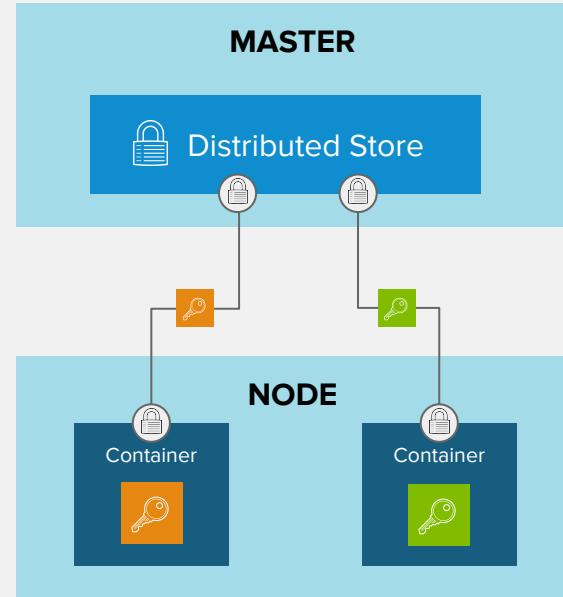


EXTEND

Security Ecosystem

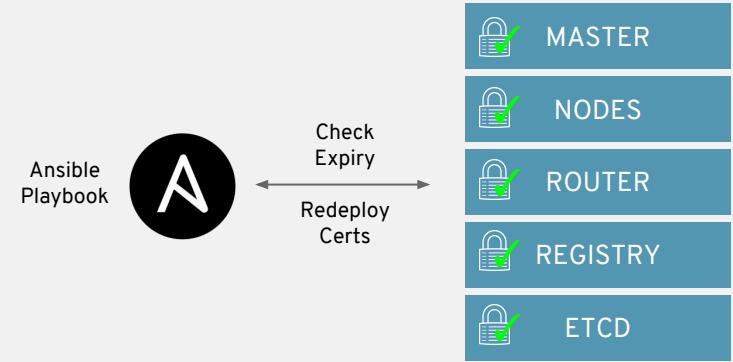
SECRET MANAGEMENT

- 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 and at rest
- Never rest on the nodes



CERTIFICATE MANAGEMENT

- Certificates are used to provide secure connections to
 - master and nodes
 - router and registry
 - etcd
- Ansible playbooks to automate redeployment
- Redeploy all at once or specific components
- Certificate expiry report generator



CERTIFICATE EXPIRY REPORT

OpenShift Certificate Expiry Report

Redeploying Certificates Expiry Role Documents

m01.example.com

Checked 14 total certificates. Expired/Warning/OK: 0/0/14. Warning window: 30 days

- **Expirations checked at:** 2017-03-15 16:02:53.563634
- **Warn after date:** 2017-04-14 16:02:53.563634

ocp_certs						
Certificate Common/Alt Name(s)	Serial	Health	Days Remaining	Expiration Date	Path	
CN:172.30.0.1,DNS:kubernetes,DNS:kubernetes.default,DNS:kubernetes.default.svc,DNS:kubernetes.default.svc.cluster.local,DNS:m01.example.com,DNS:openshift,DNS:openshift.default,DNS:openshift.default.svc,DNS:openshift.default.svc.cluster.local,DNS:172.30.0.1,DNS:192.168.122.241,IP Address:172.30.0.1,IP Address:192.168.122.241	int(4)/hex(0x4)	ok	694	2019-02-07 18:12:40	/etc/origin/master/master.server.crt	
CN:192.168.122.241,DNS:m01.example.com,DNS:192.168.122.241,IP Address:192.168.122.241	int(14)/hex(0xe)	ok	694	2019-02-07 18:19:36	/etc/origin/node/server.crt	
CN:openshift-signer@1486491158	int(1)/hex(0x1)	ok	1789	2022-02-06 18:12:38	/etc/origin/master/ca-bundle.crt	
CN:openshift-signer@1486491158	int(1)/hex(0x1)	ok	1789	2022-02-06 18:12:38	/etc/origin/node/ca.crt	
etcd						
Certificate Common/Alt Name(s)	Serial	Health	Days Remaining	Expiration Date	Path	
CN:etcd-signer@1486490714	int(13699604374018404644)/hex(0xbe1ec59834c13124L)	ok	1789	2022-02-06 18:06:00	/etc/etcd/ca.crt	
CN:m01.example.com, IP Address:192.168.122.241	int(1)/hex(0x1)	ok	1789	2022-02-06 18:06:16	/etc/etcd/server.crt	
CN:m01.example.com, IP Address:192.168.122.241	int(2)/hex(0x2)	ok	1789	2022-02-06 18:06:16	/etc/etcd/peer.crt	
kubeconfigs						
Certificate Common/Alt Name(s)	Serial	Health	Days	Expiration	Path	

CERTIFICATE CHECKS

- master and nodes
- router and registry service certificates from etcd secrets
- master, node, router, registry, and kubeconfig files for cluster-admin users
- etcd certificates

PERSISTENT STORAGE

PERSISTENT STORAGE

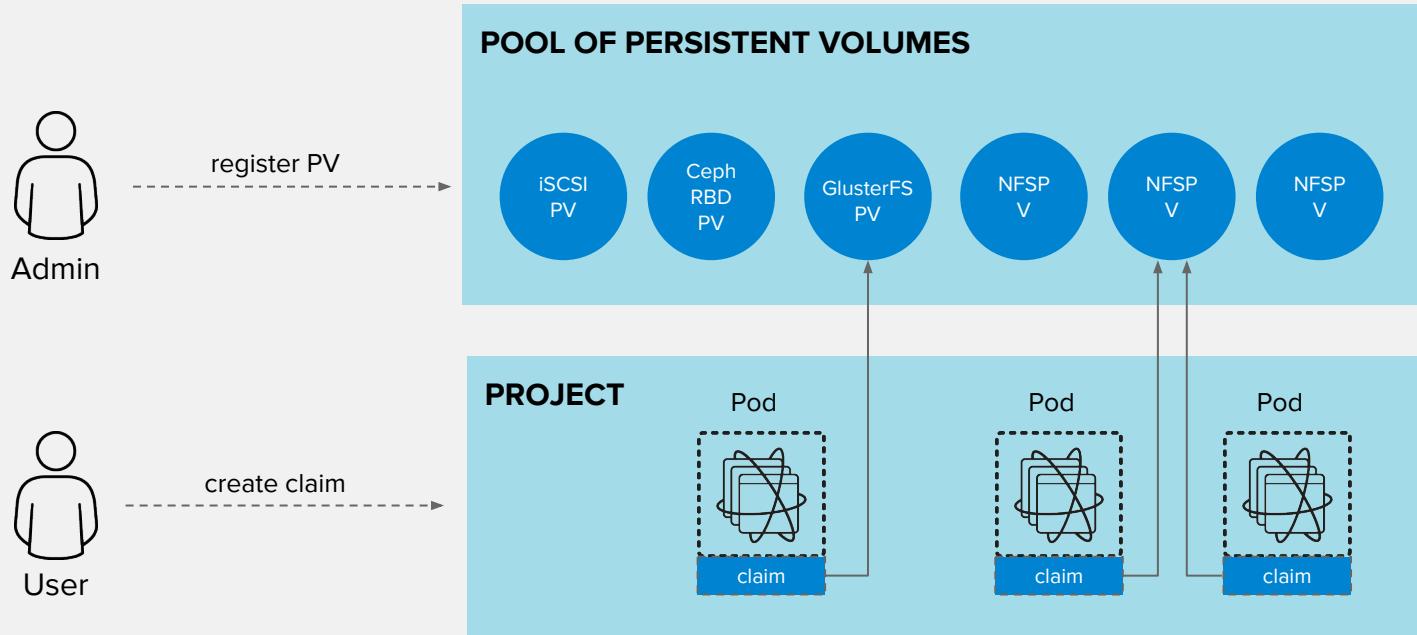
- Persistent Volume (PV) is tied to a piece of network storage
- Provisioned by an administrator (static or dynamically)
- Allows admins to describe storage and users to request storage
- Assigned to pods based on the requested size, access mode, labels and type

NFS	OpenStack Cinder	iSCSI	Azure Disk	AWS EBS	FlexVolume
GlusterFS	Ceph RBD	Fiber Channel	Azure File	GCE Persistent Disk	VMWare vSphere VMDK
NetApp Trident*			Container Storage Interface (CSI)**		

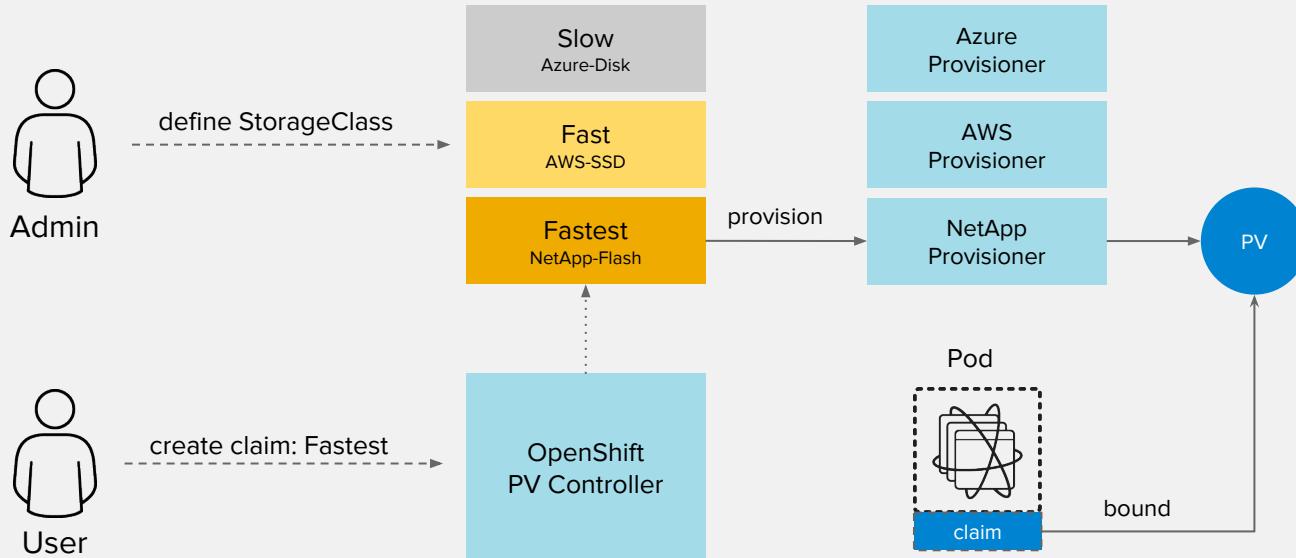
* Shipped and supported by NetApp via TSANet

** Tech Preview

PERSISTENT STORAGE

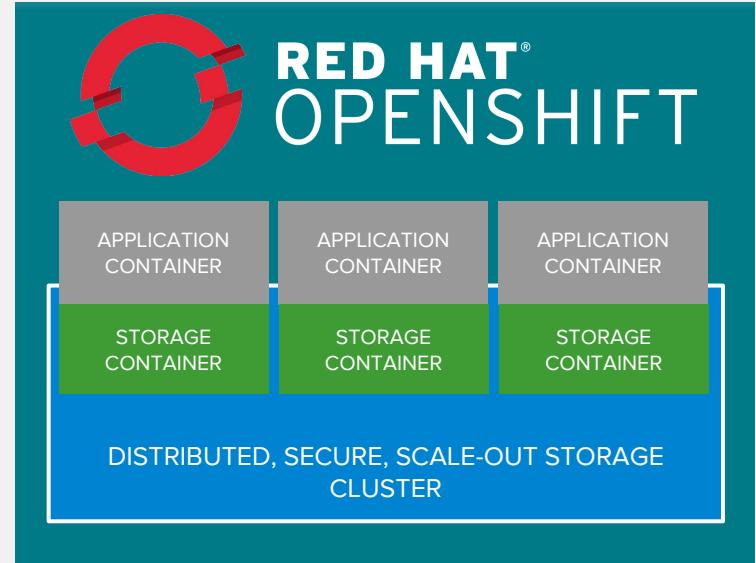


DYNAMIC VOLUME PROVISIONING

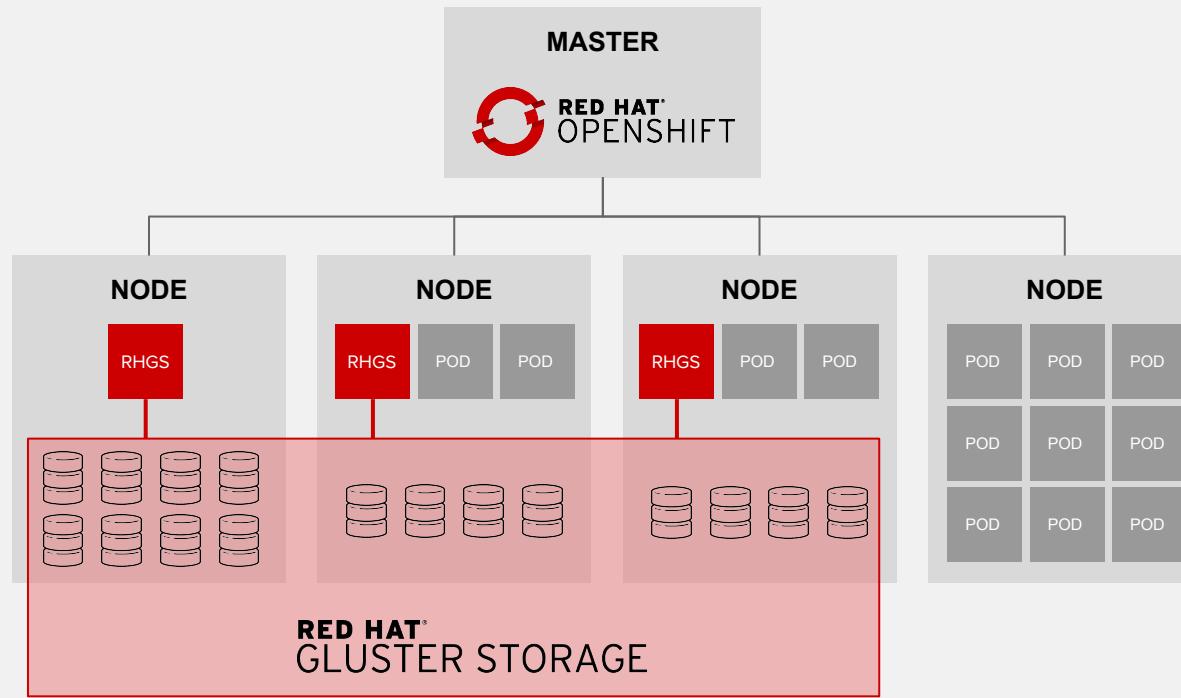


OPENShift CONTAINER STORAGE

- Containerized Red Hat Gluster Storage
- Native integration with OpenShift
- Unified Orchestration using Kubernetes for applications and storage
- Greater control & ease of use for developers
- Lower TCO through convergence
- Single vendor Support

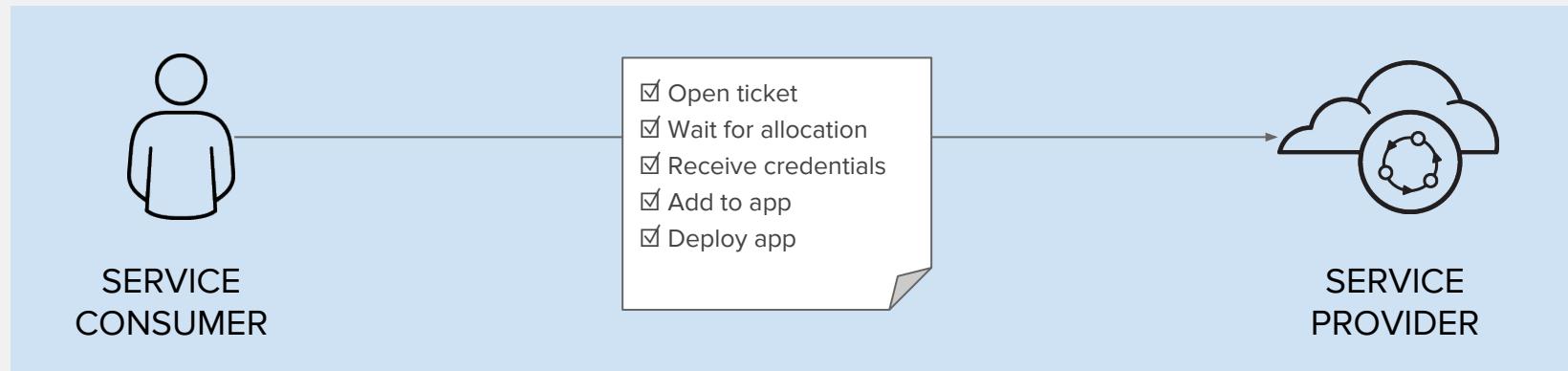


OPENShift CONTAINER STORAGE



SERVICE BROKER

WHY A SERVICE BROKER?



Manual, Time-consuming and Inconsistent



OPEN SERVICE BROKER API™

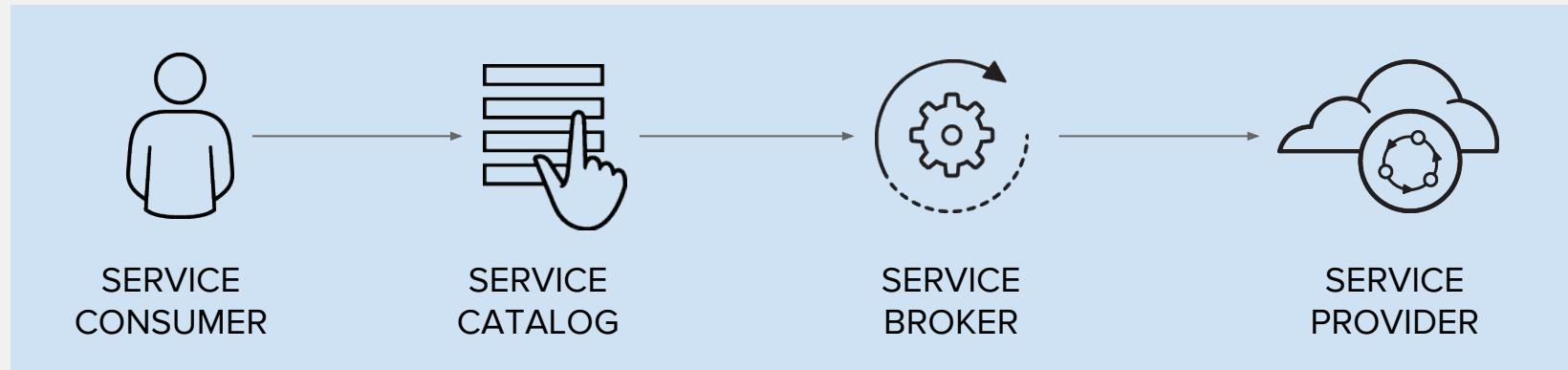
A multi-vendor project to standardize how services are consumed on cloud-native platforms across service providers

FUJITSU Pivotal.

IBM redhat.

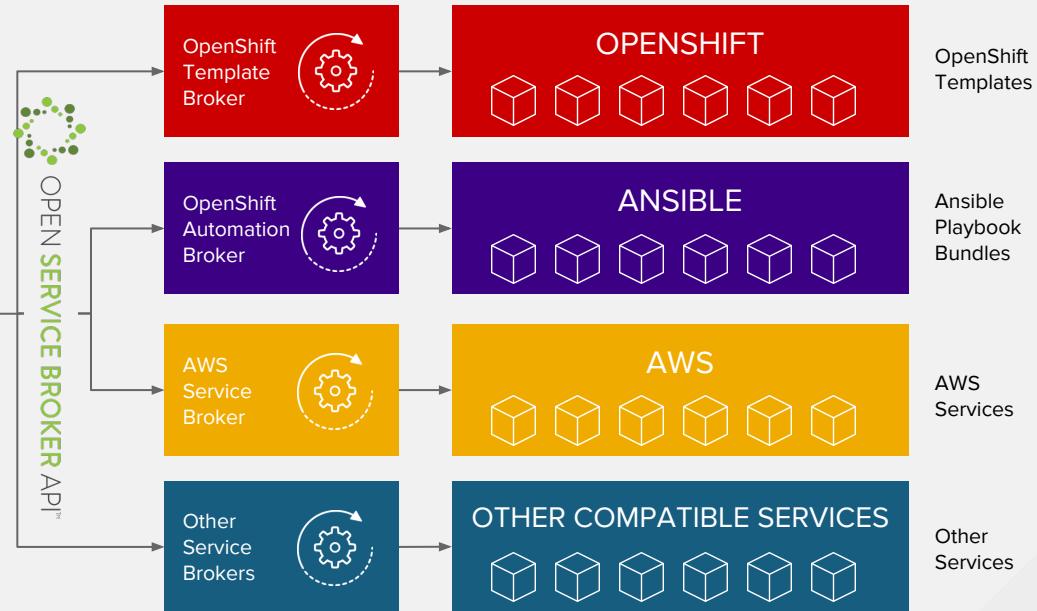
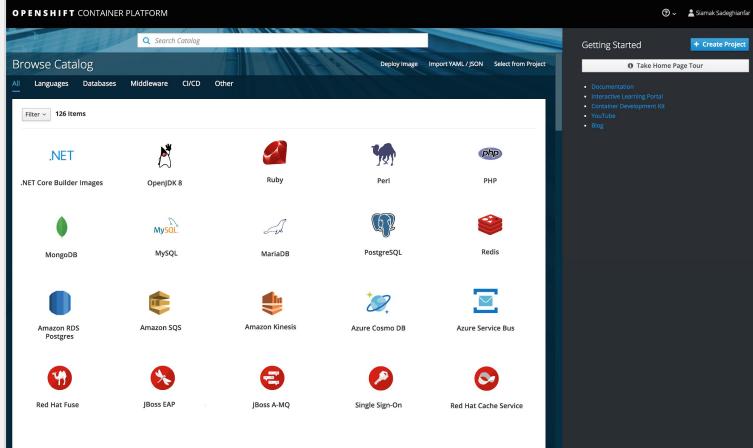
Google SAP®

WHAT IS A SERVICE BROKER?



Automated, Standard and Consistent

OPENShift SERVICE CATALOG



OPENSIFT SERVICE CATALOG

SERVICE BROKER CONCEPTS

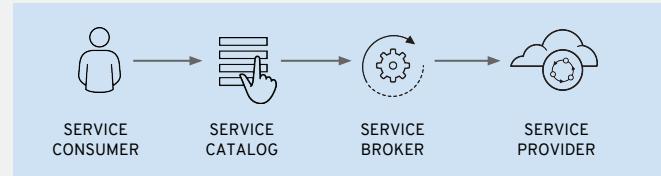
SERVICE: an offering that can be used by an app e.g. database

PLAN: a specific flavor of a service e.g. Gold Tier

SERVICE INSTANCE: an instance of the offering

PROVISION: creating a service instance

BIND: associate a service instance and its credentials to an app



HOW TO ADD A SERVICE BROKER

- Deploy service broker on or off OpenShift
- Register the broker referring to the deployed broker

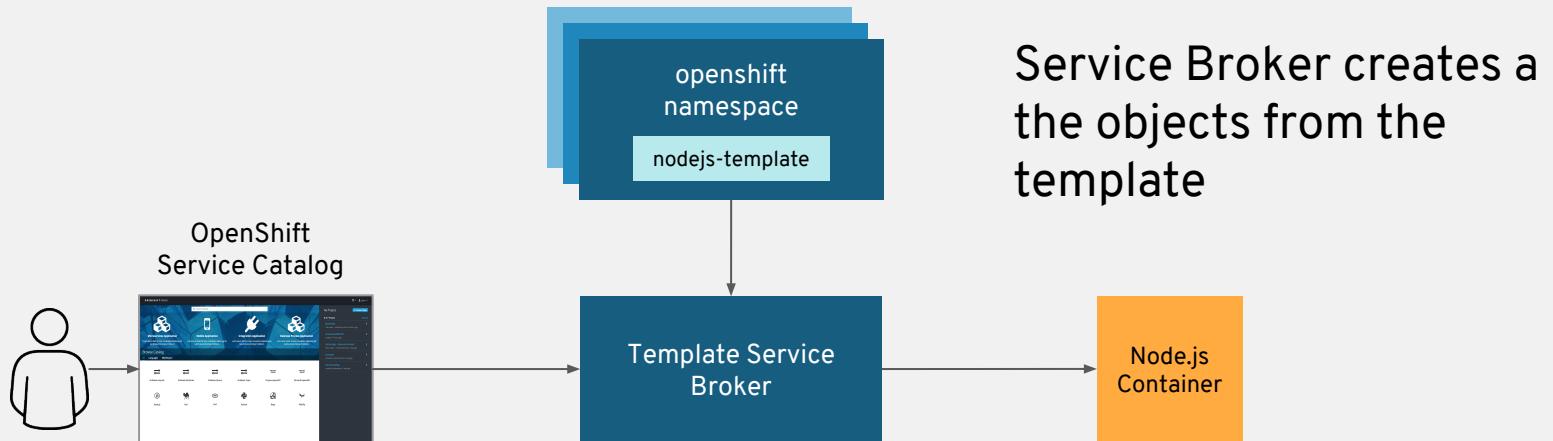
```
apiVersion: servicecatalog.k8s.io/v1alpha1
kind: Broker
metadata:
  name: asb-broker
spec:
  url: https://asb-1338-ansible-service-broker.10.2.2.15.nip.io
```

- Register the broker services by creating ServiceClass resources
(the service broker might automatically perform this step)

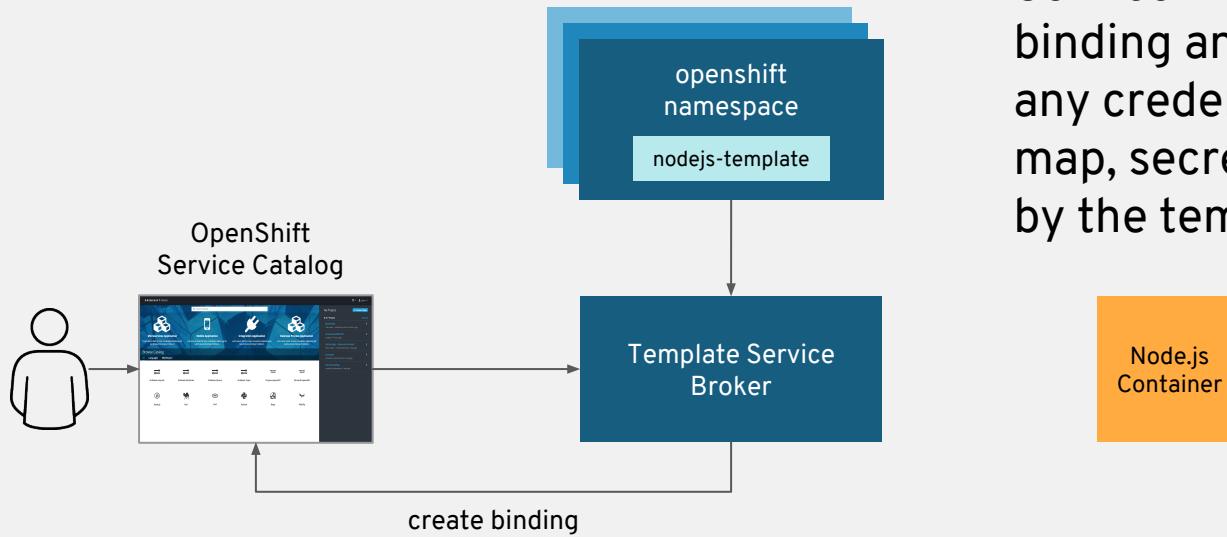
TEMPLATE SERVICE BROKER

- Exposes Templates and Instant Apps in the Service Catalog
- Pulled from openshift namespace by default
- Multiple namespaces can be configured for template discovery

TEMPLATE SERVER BROKER PROVISIONING



TEMPLATE SERVICE BROKER BINDING



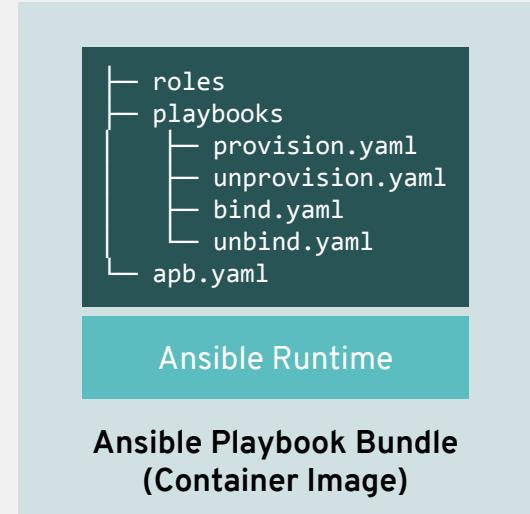
Service Broker creates a binding and secret for any credentials (config map, secret, etc) created by the template

OPENShift ANSIBLE BROKER

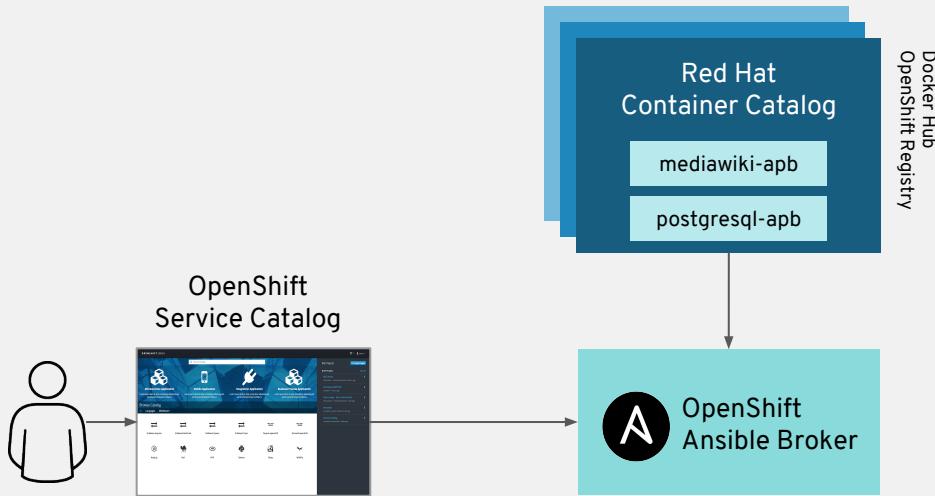
- Use Ansible on OpenShift
 - Deploy containerized applications
 - Provision external services (e.g. Oracle database)
 - Provision cloud services (e.g. AWS RDS)
 - Orchestrate multi-service solutions
 - Conditional logic for control on deployments (e.g. database is initialized)
- Leverage existing Ansible playbooks
- Anything you can do with Ansible, you can do with OAB

ANSIBLE PLAYBOOK BUNDLES (APB)

- Lightweight application definition
- Packaged as a container image
- Embedded Ansible runtime
- Metadata for parameters
- Named playbooks for actions
- Leverage existing Ansible playbooks
- Registry is queried to discover APBs

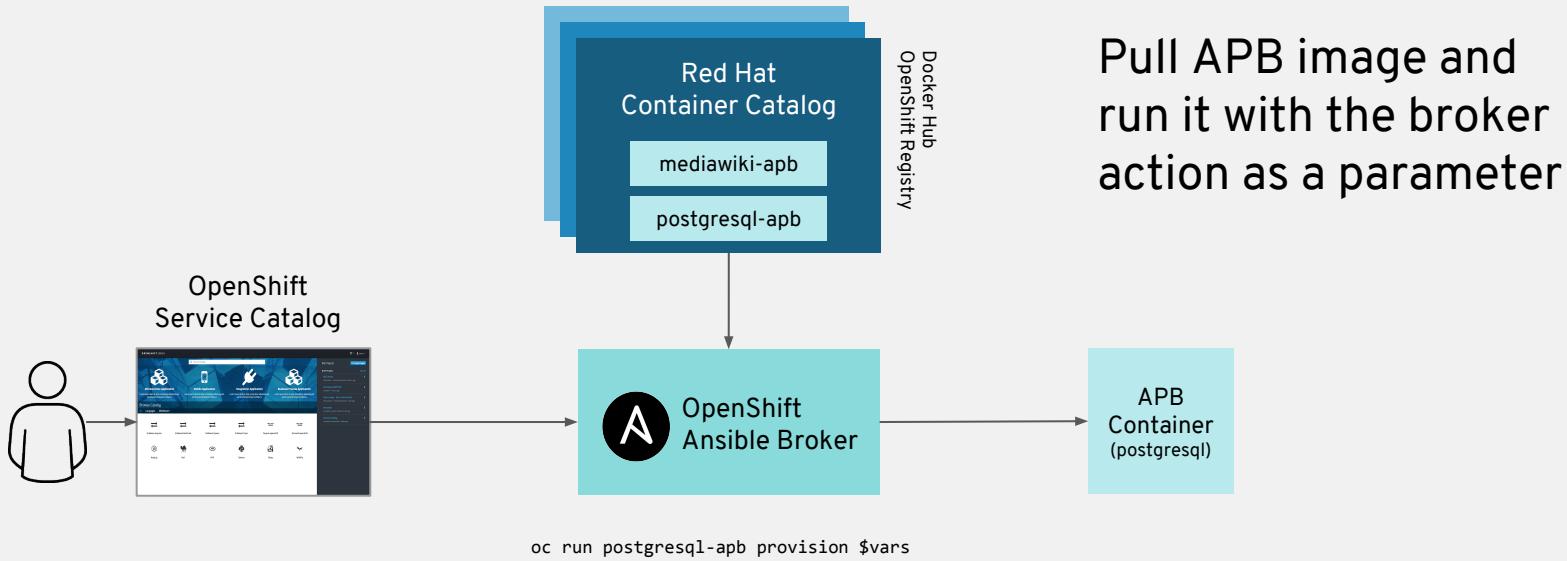


OPENShift ANSIBLE BROKER PROVISIONING

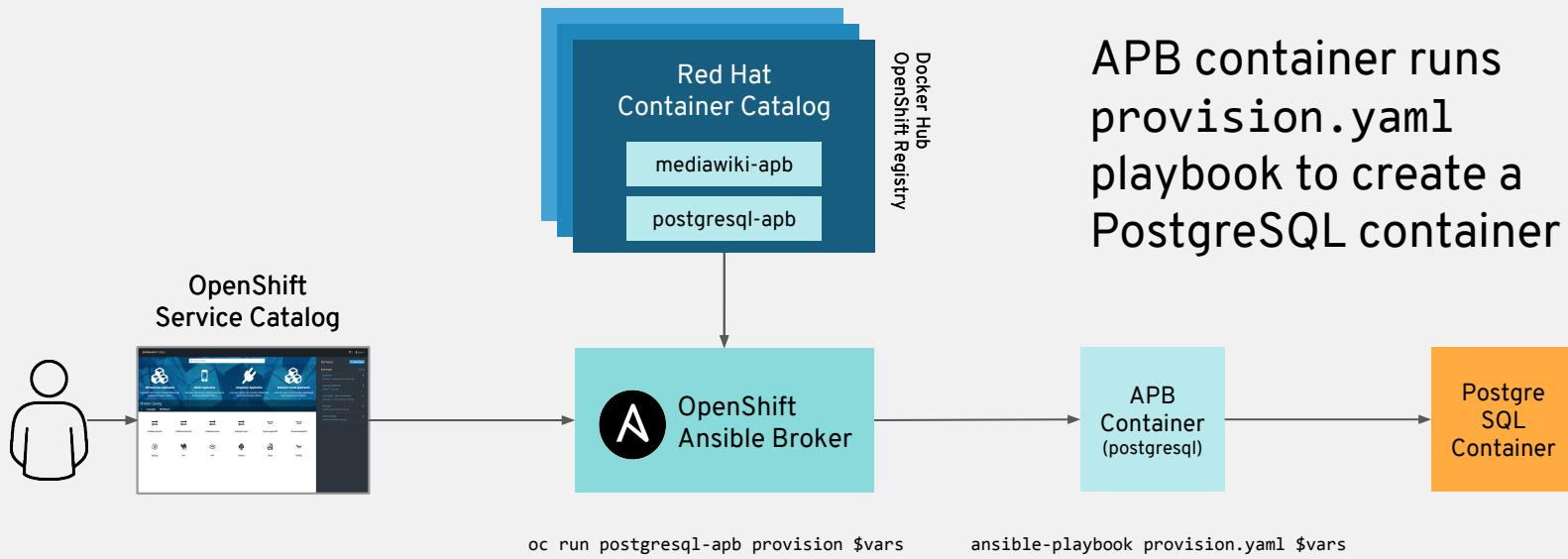


Discover and list APBs from the configured image registries

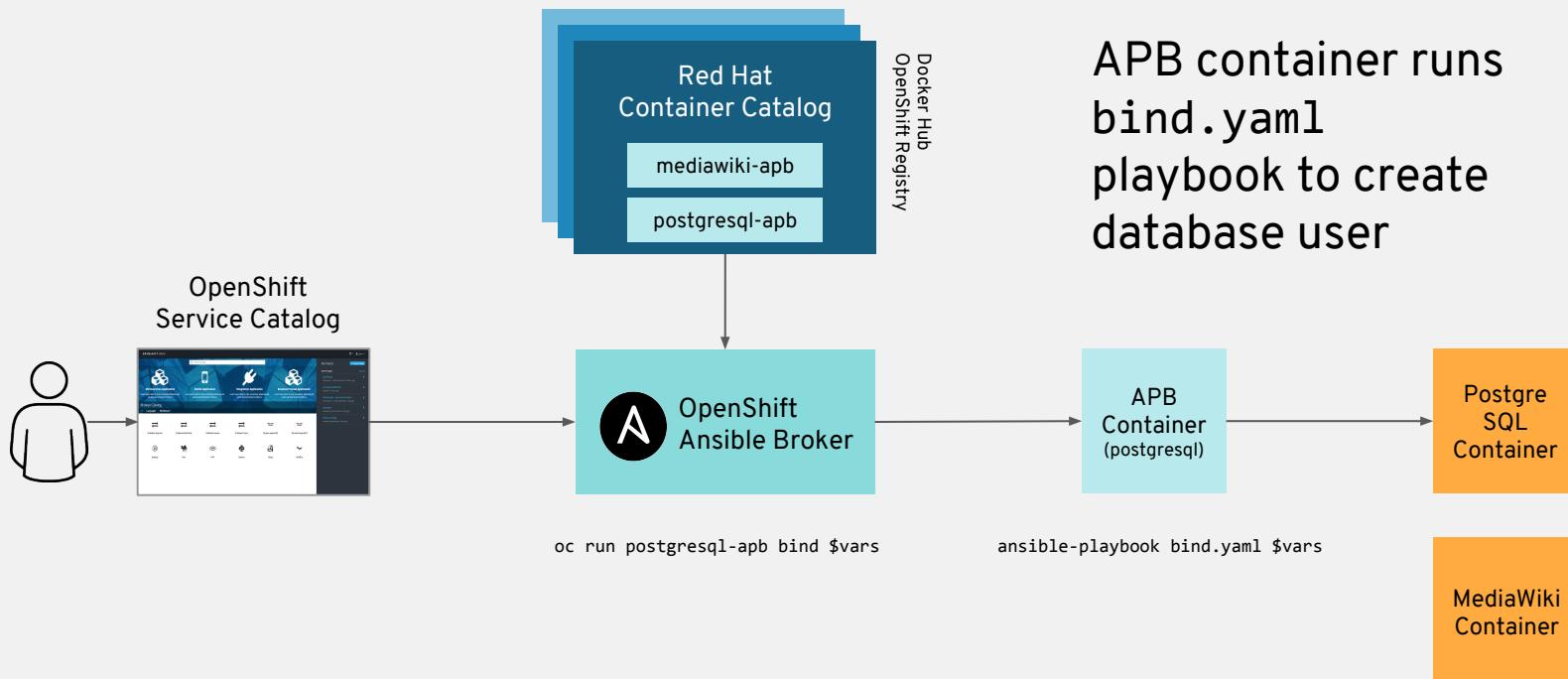
OPENShift ANSIBLE BROKER PROVISIONING



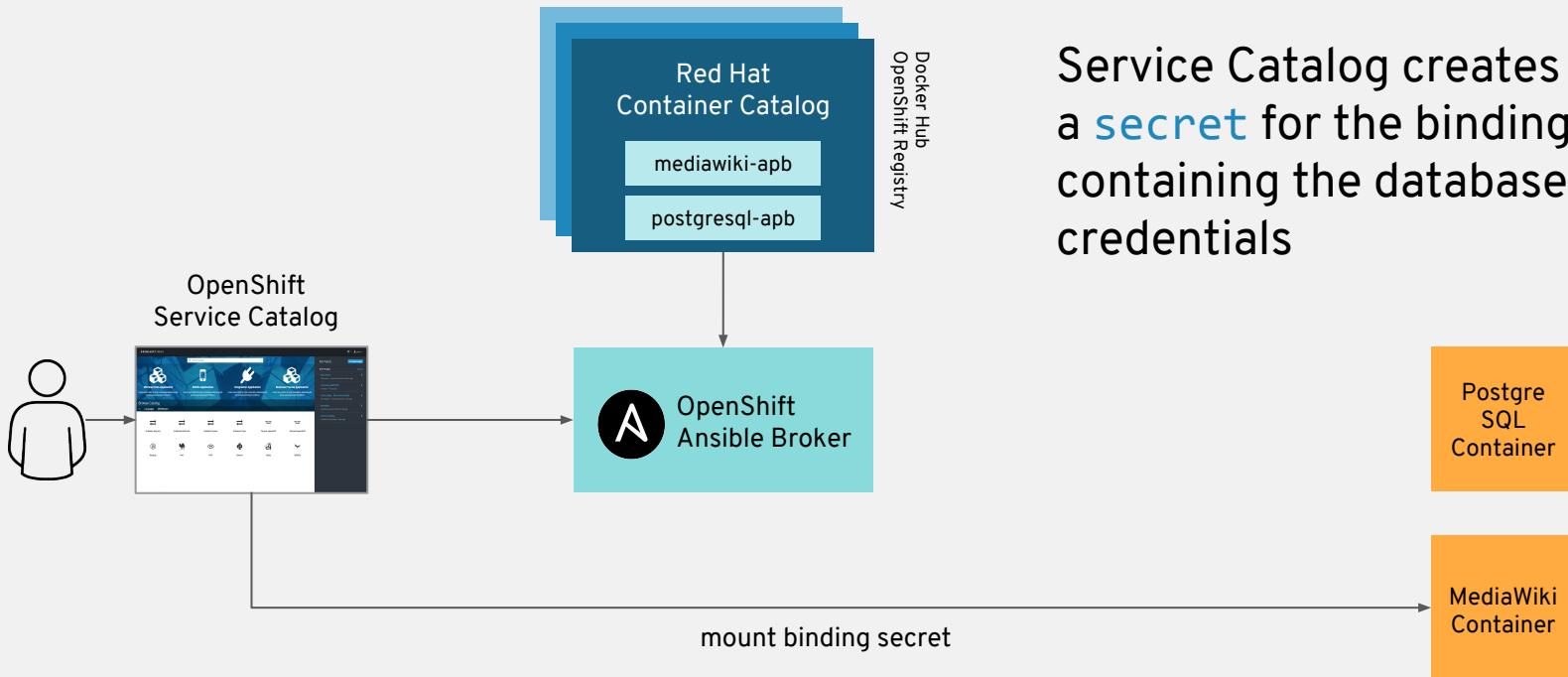
OPENShift ANSIBLE BROKER PROVISIONING



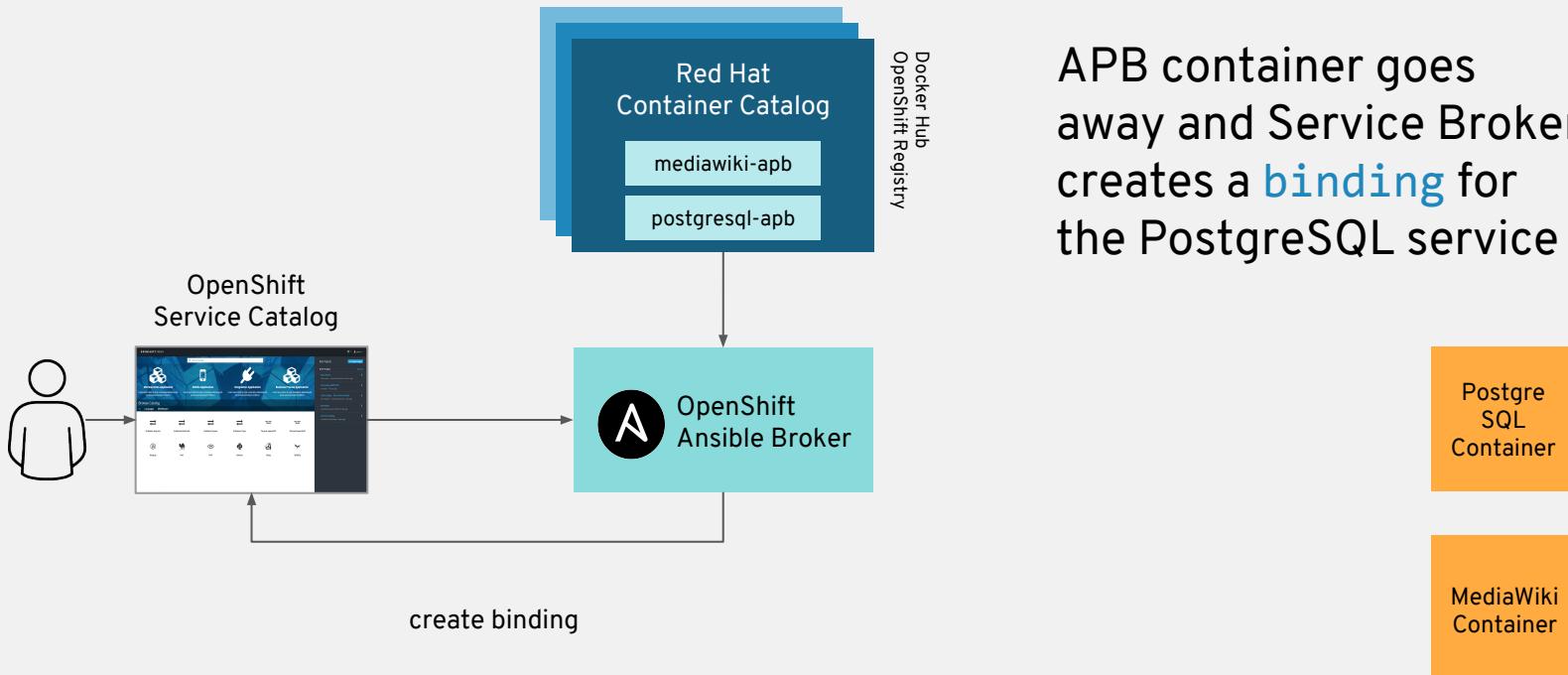
OPENShift ANSIBLE BROKER BINDING



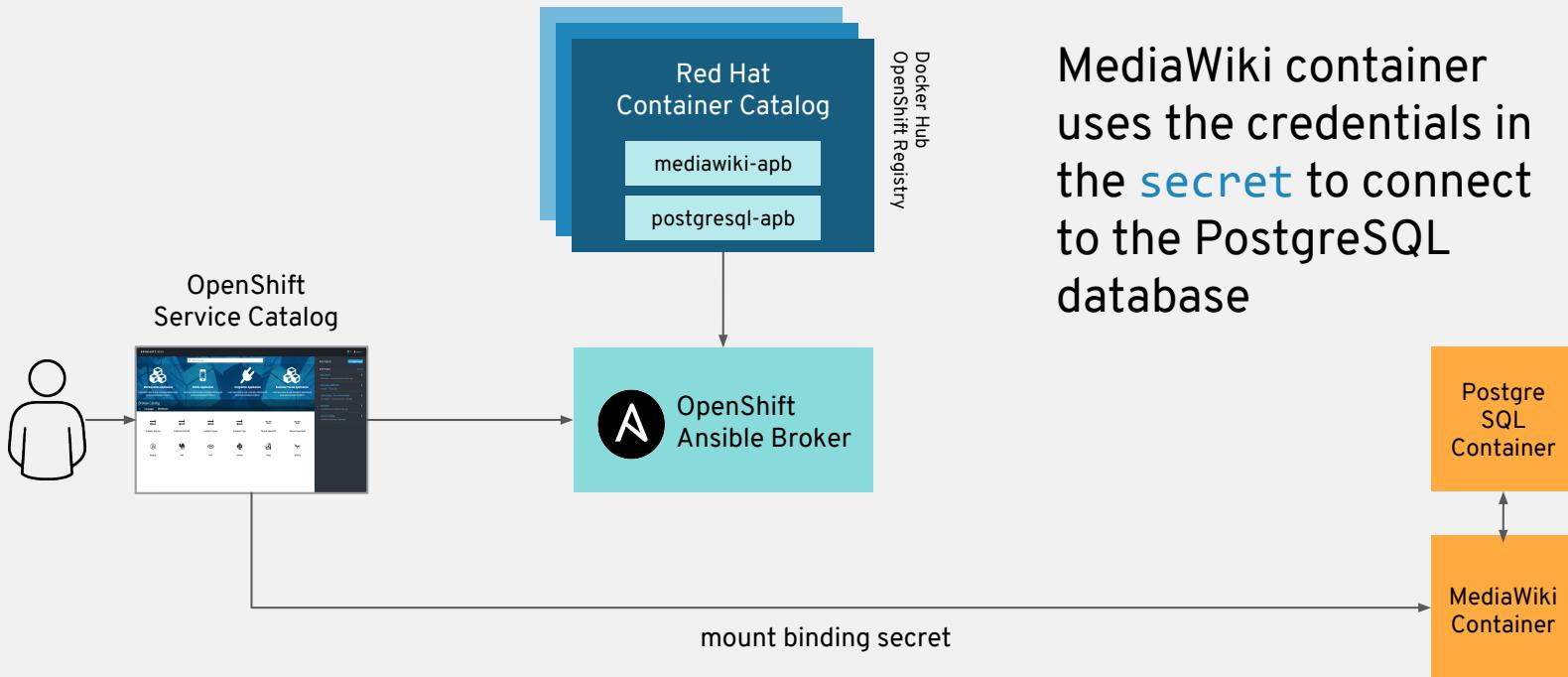
OPENShift ANSIBLE BROKER BINDING



OPENShift ANSIBLE BROKER BINDING



OPENShift ANSIBLE BROKER BINDING



AWS SERVICE BROKER

OPENShift ORIGIN

Search Catalog

Browse Catalog Deploy Image Import YAML / JSON Select from Project

All Languages Databases CI/CD Other

Filter 28 Items

The screenshot shows the OpenShift Origin AWS Service Broker interface. On the left, there's a grid of service icons and names. On the right, a sidebar lists 'My Projects' with a 'Create Project' button.

Service	Description
Amazon DynamoDB	
Amazon EMR (APB)	
Amazon RDS (APB)	
Amazon Redshift	
Amazon Route 53 (APB)	
Amazon S3	
Amazon SNS (APB)	
Amazon SQS Queue (APB)	
Apache HTTP Server (httpd)	
CakePHP + MySQL (Persistent)	
Dancer + MySQL (Persistent)	
dh-elasticache-apb	
Django + PostgreSQL (Persistent)	
Jenkins (Ephemeral)	
Jenkins (Persistent)	
MariaDB (Persistent)	
MongoDB (Persistent)	
MySQL (Persistent)	
Node.js	
Node.js + MongoDB (Persistent)	

My Projects

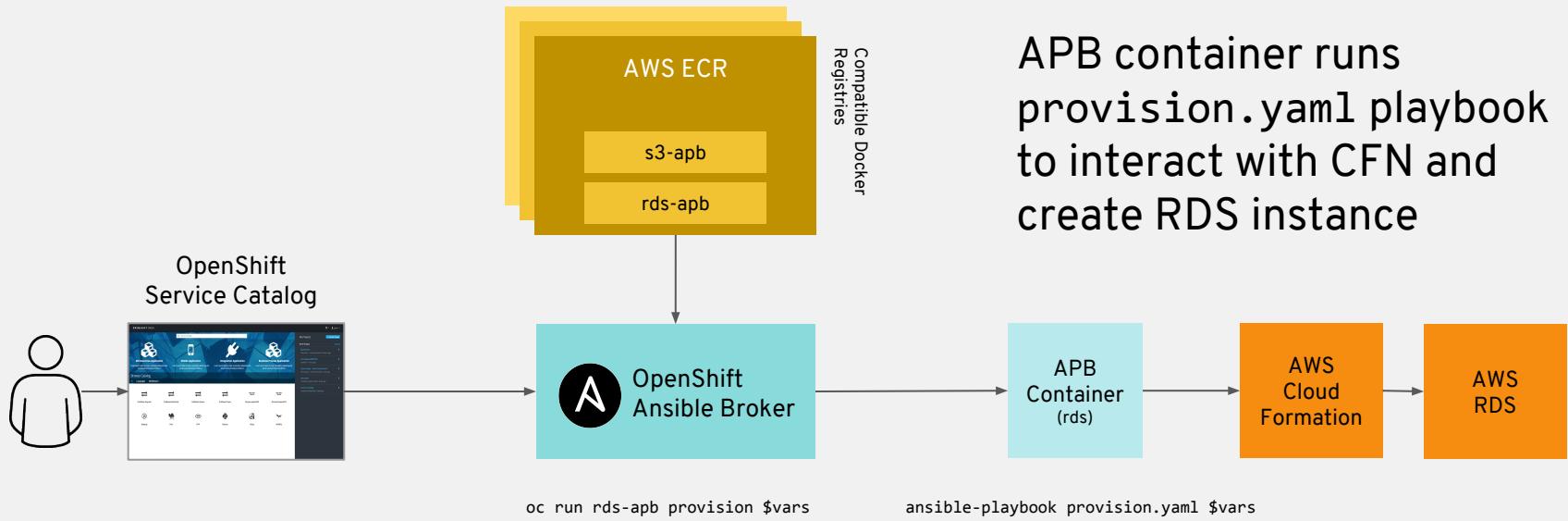
+ Create Project

5 of 6 Projects View All

- kube-service-catalog
- aws-service-broker
- kube-system
- My Project
- openshift-infra

- Amazon Athena
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon EMR
- Amazon Kinesis Data Streams
- Amazon KMS
- Amazon Lex
- Amazon Polly
- Amazon RDS for MariaDB
- Amazon RDS for MySQL
- Amazon RDS for PostgreSQL
- Amazon RedShift
- Amazon Rekognition
- Amazon Route 53
- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Translate

AWS PROVISIONING



AZURE SERVICE BROKER

The screenshot shows the OpenShift Container Platform Service Catalog interface. On the left, there's a grid of service icons categorized by type. The categories include:

- .NET Core + PostgreSQL (Persistent)**, **.NET Core Example**, **.NET Core Runtime Example**, **Apache HTTP Server**, and **Apache HTTP Server (httpd)**.
- Azure Container Instances**, **Azure Cosmos DB**, **Azure Cosmos DB (Graph API)**, **Azure Cosmos DB (MongoDB)**, and **Azure Cosmos DB (MongoDB)**.
- Azure Database for MySQL**, **Azure Database for MySQL - Database Only**, **Azure Database for MySQL - DBMS Only**, **Azure Database for PostgreSQL**, and **Azure Database for PostgreSQL - Database Only**.
- Azure Database for PostgreSQL - DBMS Only**, **Azure Event Hubs**, **Azure Key Vault**, **Azure Redis Cache**, and **Azure Search**.
- Azure Service Bus**, **Azure SQL Database**, **Azure SQL Server (Database Only)**, **Azure SQL Server (DBMS Only)**, and **Azure Storage**.

On the right side of the interface, there's a sidebar titled "My Projects" which lists the user's projects. It shows 5 of 13 projects, including "jzcosmosdb-test", "osba", "default", "kube-service-catalog", and "jzpostgres". Below this, there's a "Recently Viewed" section with icons for "Azure Database for MySQL", "Azure Cosmos DB (MongoDB)", and "Azure Key Vault".

Available on OpenShift on Azure managed-service and Azure Stack

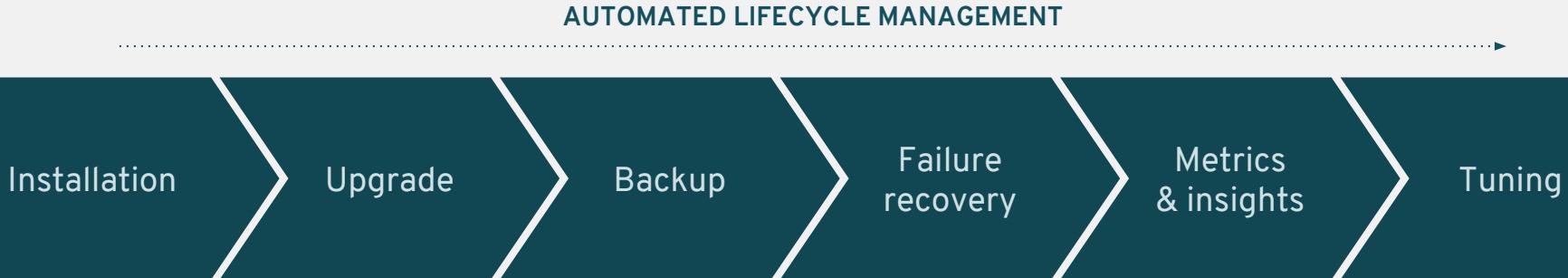
- Azure Cosmos DB
- Azure KeyVault
- Azure Storage
- Azure Redis Cache
- Azure DocumentDB
- Azure Service Bus and Event Hub
- Azure SQL Database
- Azure SQL Database Failover Group
- Azure Database for MySQL
- Azure Database for PostgreSQL

OPERATOR FRAMEWORK

(coming soon)

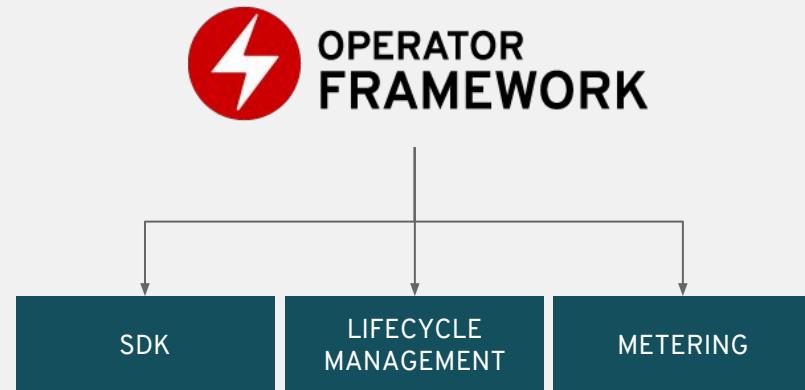
KUBERNETES OPERATOR FRAMEWORK

Operator Framework is an open source toolkit to manage application instances on Kubernetes in an effective, automated and scalable way.



KUBERNETES OPERATOR FRAMEWORK

Operators codify operational knowledge and workflows to automate lifecycle management of containerized applications with Kubernetes



WHY OPERATOR FRAMEWORK?



OPERATOR LIFECYCLE MANAGER

The image displays two screenshots of the OpenShift Container Platform interface.

Left Screenshot: Service Catalog

This screenshot shows the "Service Catalog" page under "OPENSHIFT CONTAINER PLATFORM". The top navigation bar includes "Service Catalog" and a search bar labeled "Search Catalog". Below the header, there are tabs for "Deploy Image", "Import YAML / JSON", and "Select from Project". The main area is titled "Browse Catalog" and lists categories: All, Languages, Databases, Middleware, CI/CD, and Other. A filter bar at the top indicates "8 of 49 Items" and filters for "Publisher: CoreOS, Inc" and "Couchbase". The catalog lists several services with icons and versions:

- Chargeback 0.5.1
- Couchbase 0.8.0
- etcd 0.6.1
- etcd 0.9.0
- Prometheus 0.14.0
- Prometheus 0.15.0
- Vault 0.1.5
- Vault 0.1.9

Right Screenshot: Cluster Console

This screenshot shows the "Cluster Console" page for the project "demo". The top navigation bar includes "Project: demo" and a "Cluster Console" dropdown. The left sidebar contains navigation links: Overview, Applications, Cluster Service Versions, Open Cloud Catalog (which is selected), Subscriptions, Install Plans, Workloads, Networking, Troubleshooting, Administration, Projects, Namespaces, Nodes, Persistent Volumes, Cluster Settings, Service Accounts, and Storage Classes. The main content area is titled "Open Cloud Services" and lists a single entry:

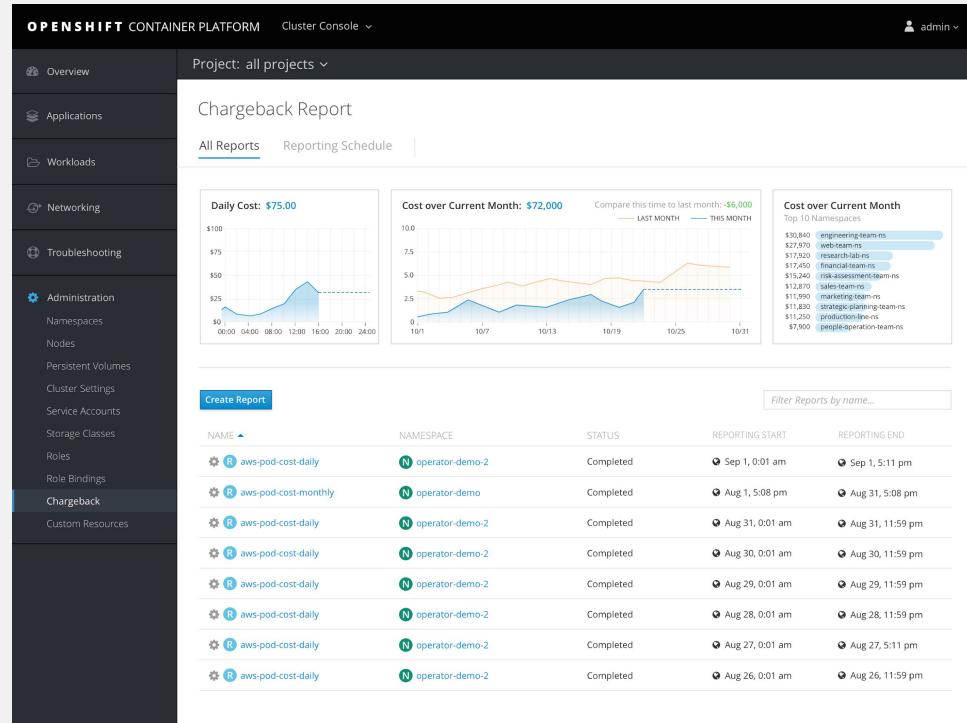
NAME	PUBLISHER
Tectonic Open Cloud Services	CoreOS, Inc.

Below this, the "Applications" section lists the installed services:

NAME	LATEST VERSION
Chargeback	0.5.1 (alpha)
Couchbase	0.8.0 (alpha)
etcd	0.9.0 (alpha)
Prometheus	0.15.0 (alpha)

OPERATOR METERING

- Based on Prometheus
- Reports namespace, pods and custom label query
- Easy to process by accounting or custom software



THE INDUSTRY IS ALIGNING BEHIND THE KUBERNETES OPERATOR FRAMEWORK



60+ Certified ISV Operators in Red Hat Early Access Program

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](#)

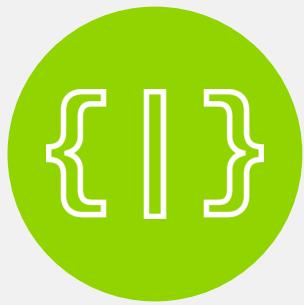
[Spring Boot Microservices on Red Hat OpenShift](#)

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

[App CI/CD on OCP with Jenkins](#)

BUILD AND DEPLOY CONTAINER IMAGES

BUILD AND DEPLOY CONTAINER IMAGES



DEPLOY YOUR
SOURCE CODE

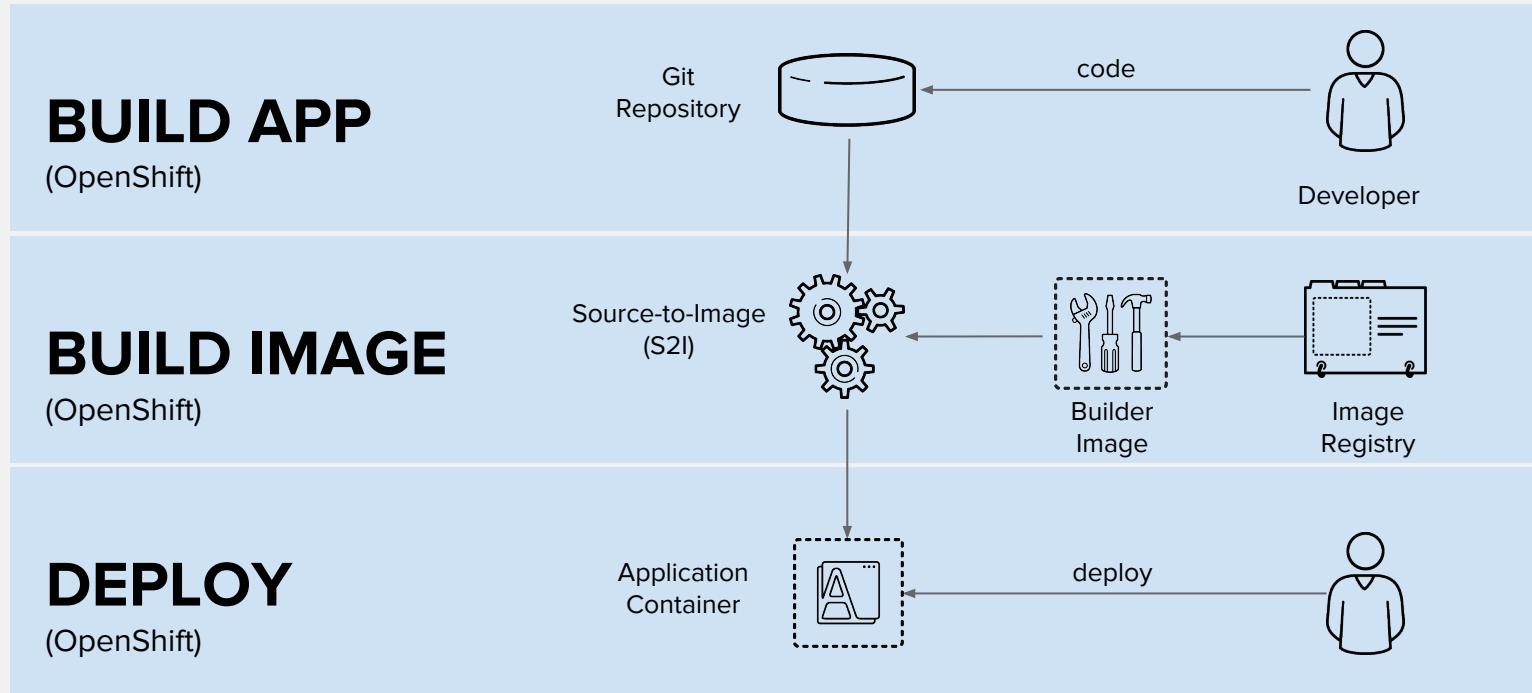


DEPLOY YOUR
APP BINARY

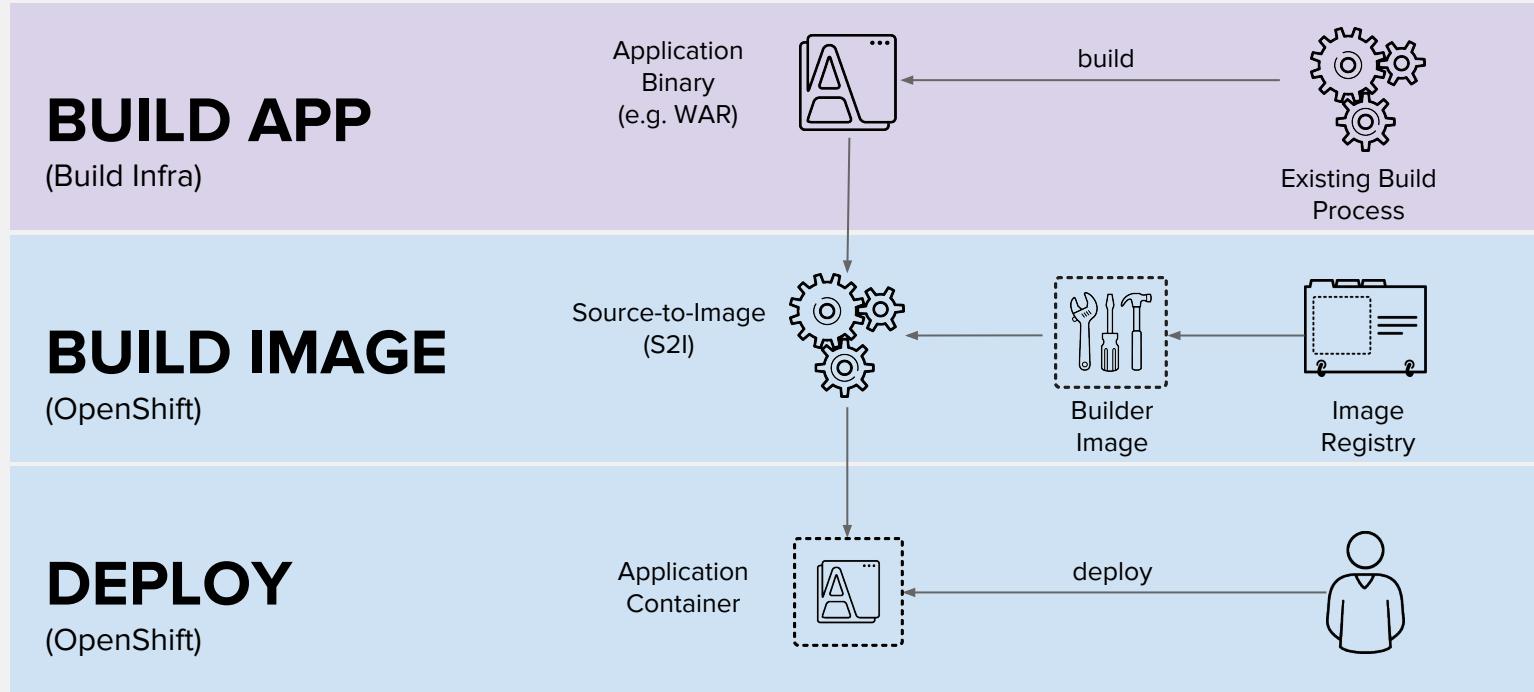


DEPLOY YOUR
CONTAINER IMAGE

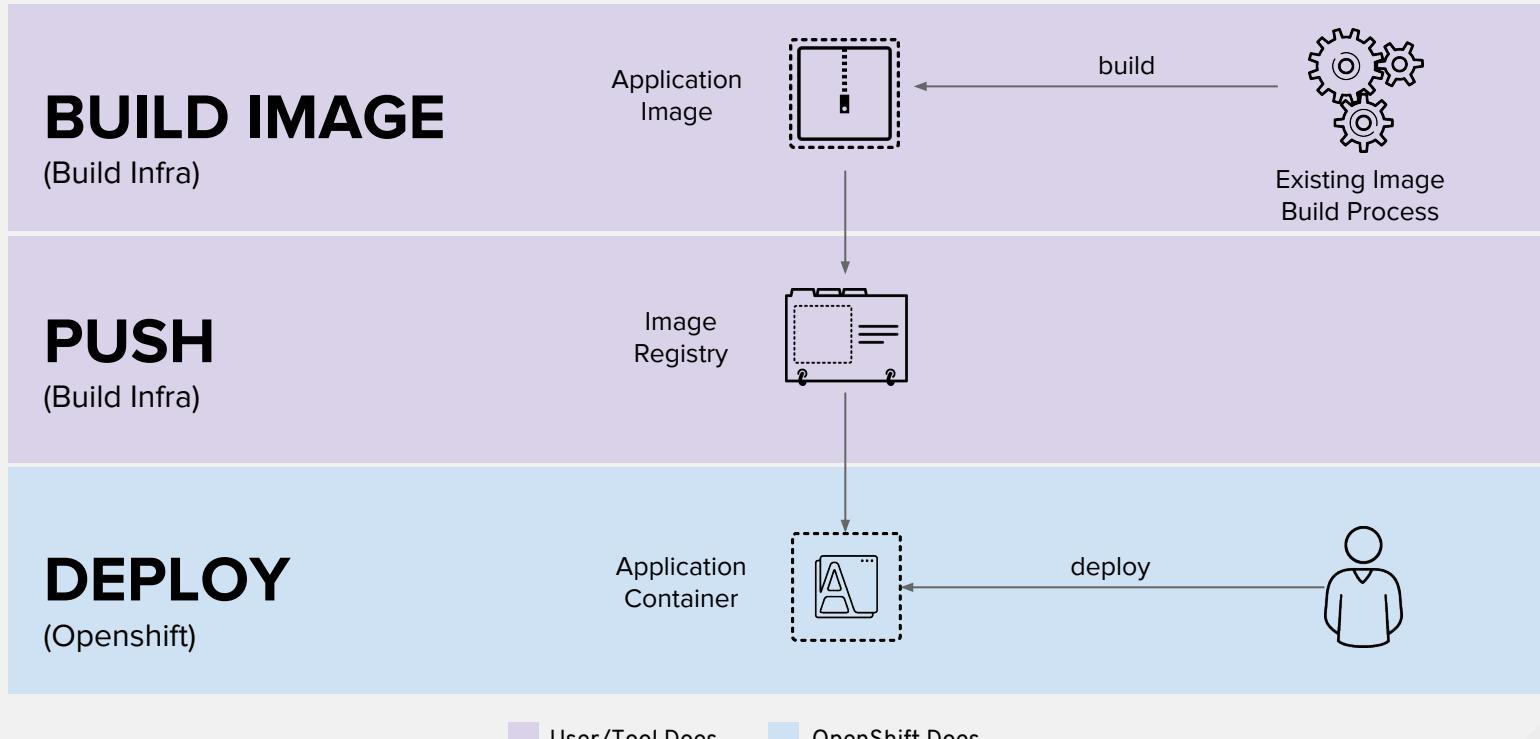
DEPLOY SOURCE CODE WITH SOURCE-TO-IMAGE (S2I)



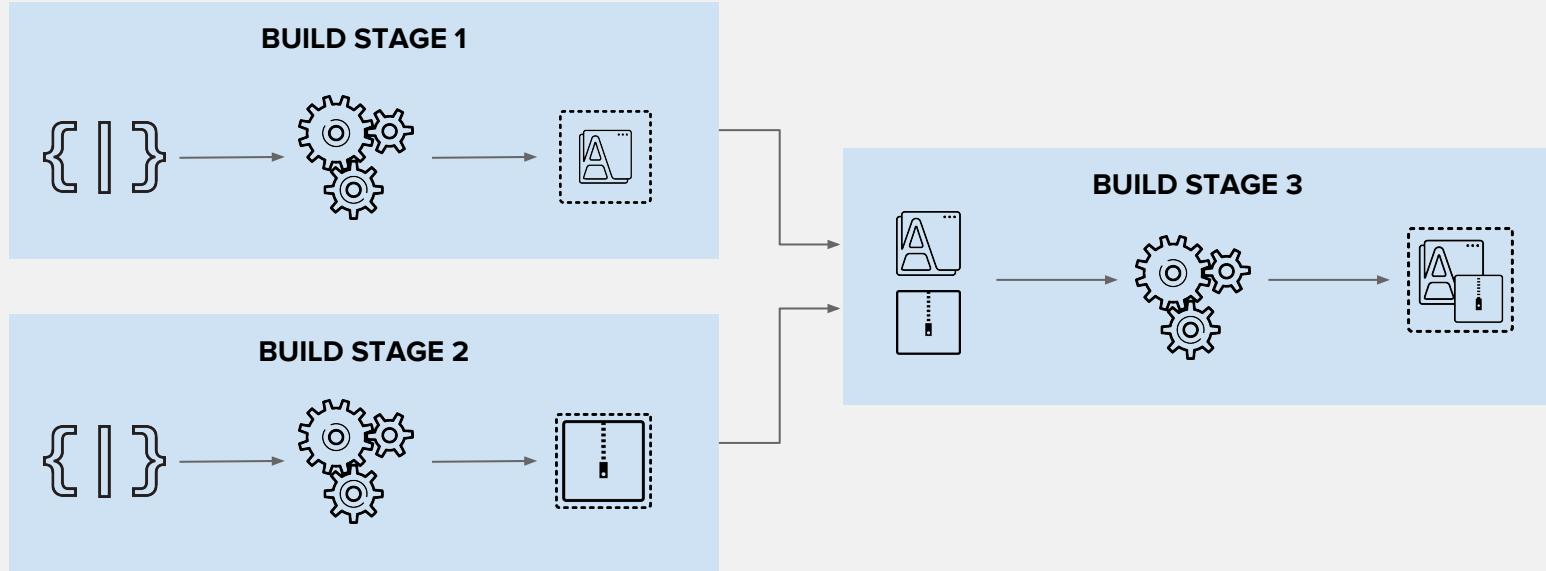
DEPLOY APP BINARY WITH SOURCE-TO-IMAGE (S2I)



DEPLOY DOCKER IMAGE

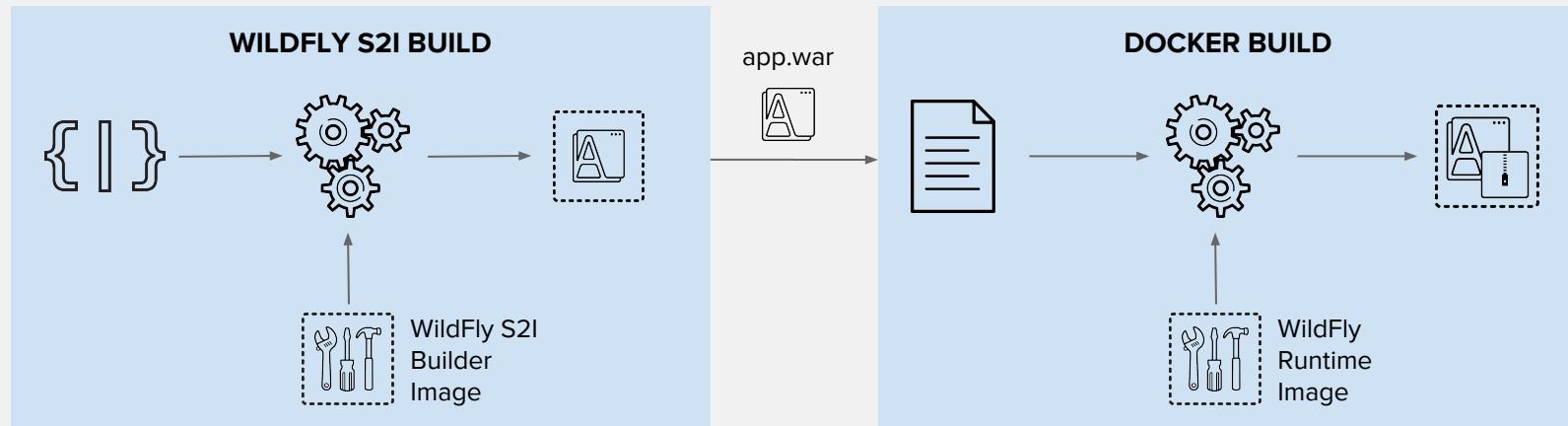


BUILD IMAGES IN MULTIPLE STAGES



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

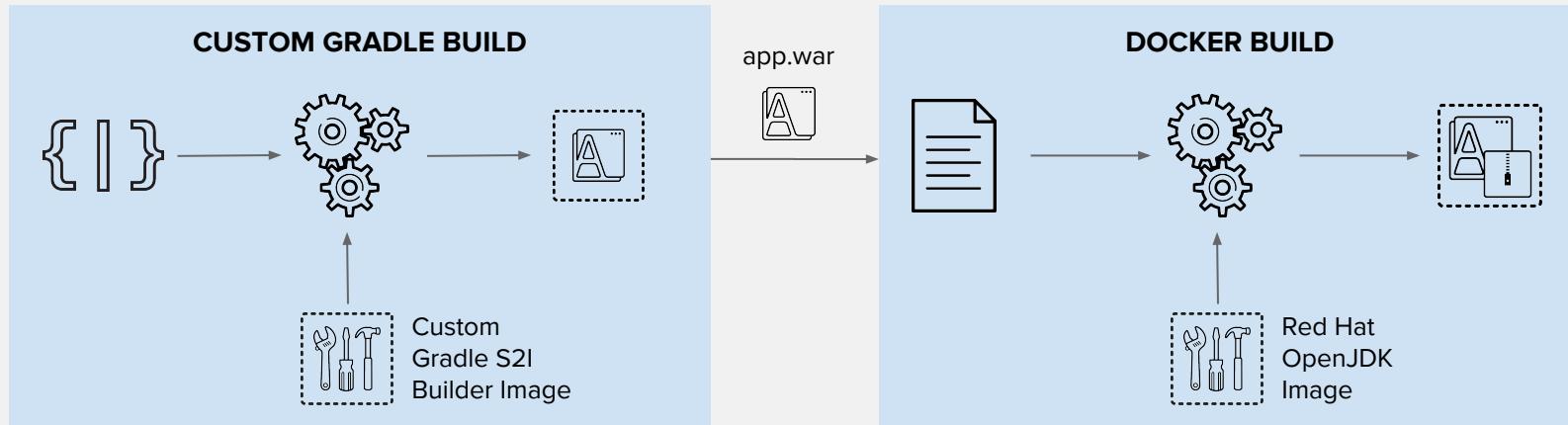
Use Source-to-Image to build app binaries and deploy on lean vanilla runtimes



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

EXAMPLE: USE ANY BUILD TOOL WITH OFFICIAL RUNTIME IMAGES

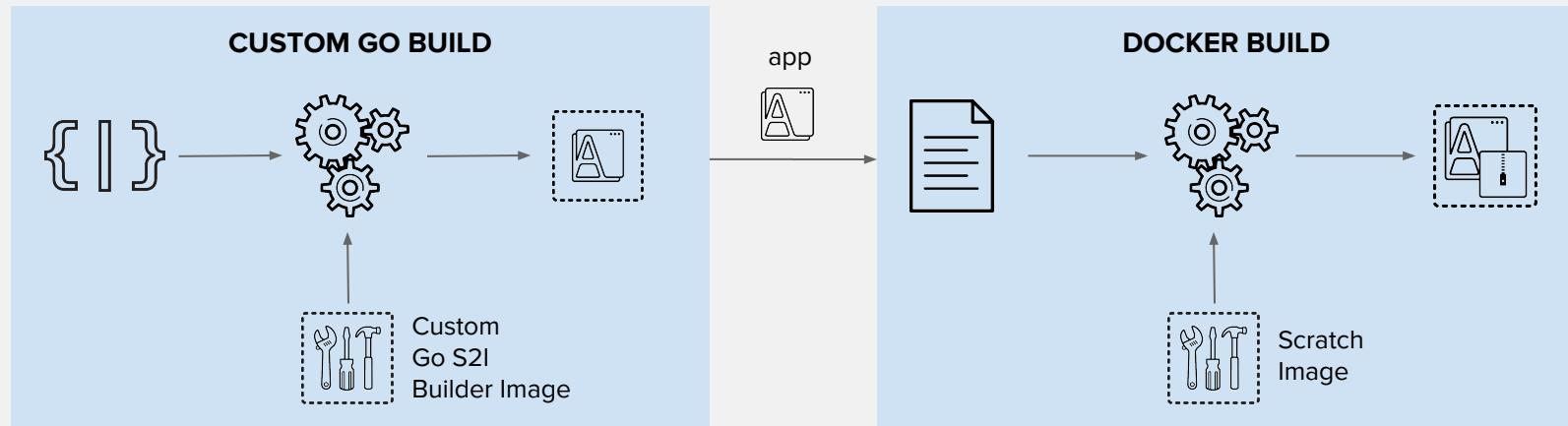
Use your choice of build tool like Gradle and deploy to official images like the JDK image



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

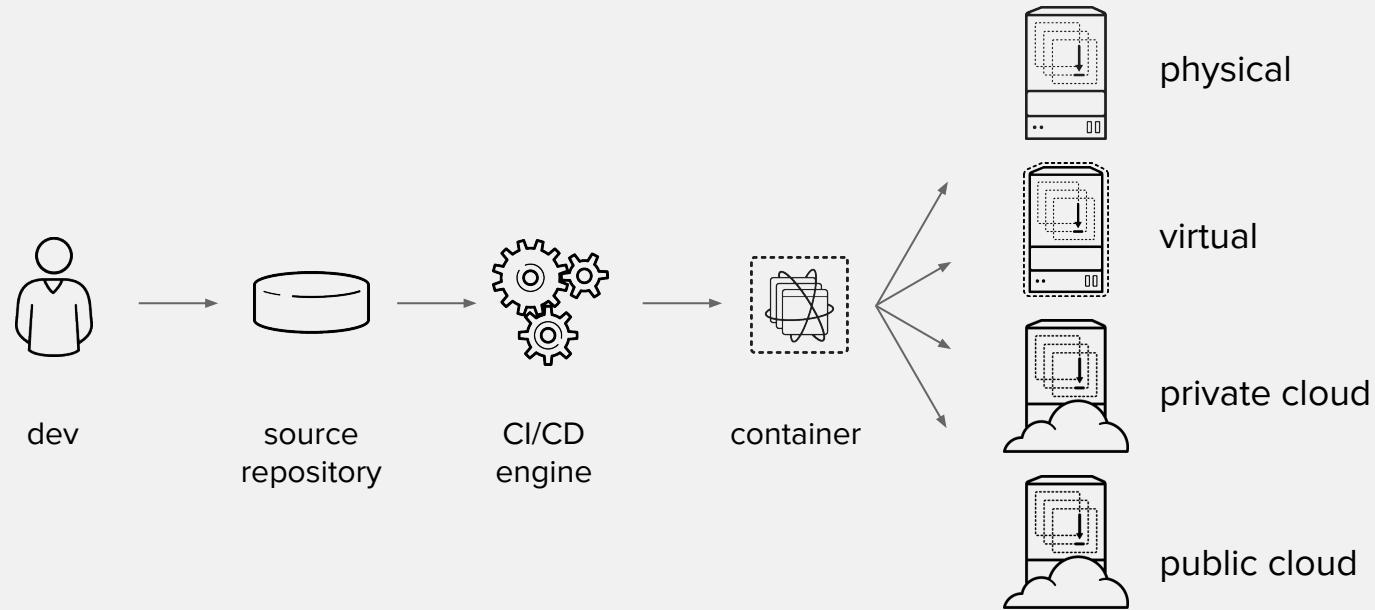
BUILDS

- 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



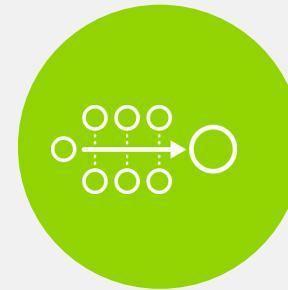
OPENSHIFT LOVES CI/CD



JENKINS-AS-A SERVICE
ON OPENSHIFT



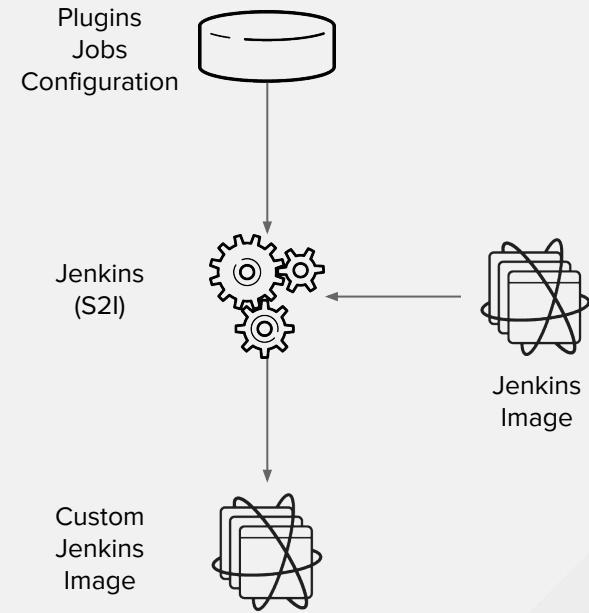
HYBRID JENKINS INFRA
WITH OPENSHIFT



EXISTING CI/CD
DEPLOY TO OPENSHIFT

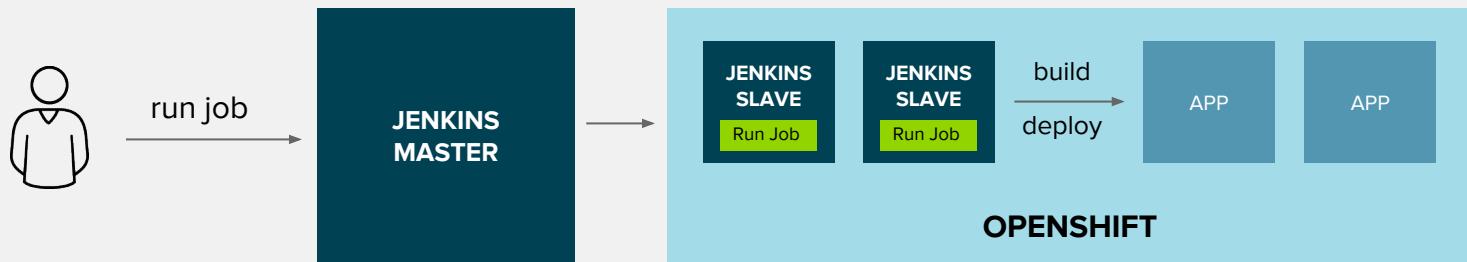
JENKINS-AS-A-SERVICE ON OPENSIFT

- 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



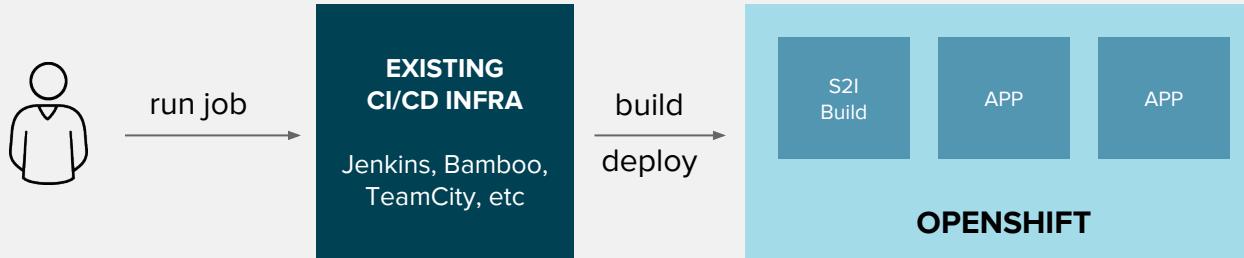
HYBRID JENKINS INFRA WITH OPENSHIFT

- 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

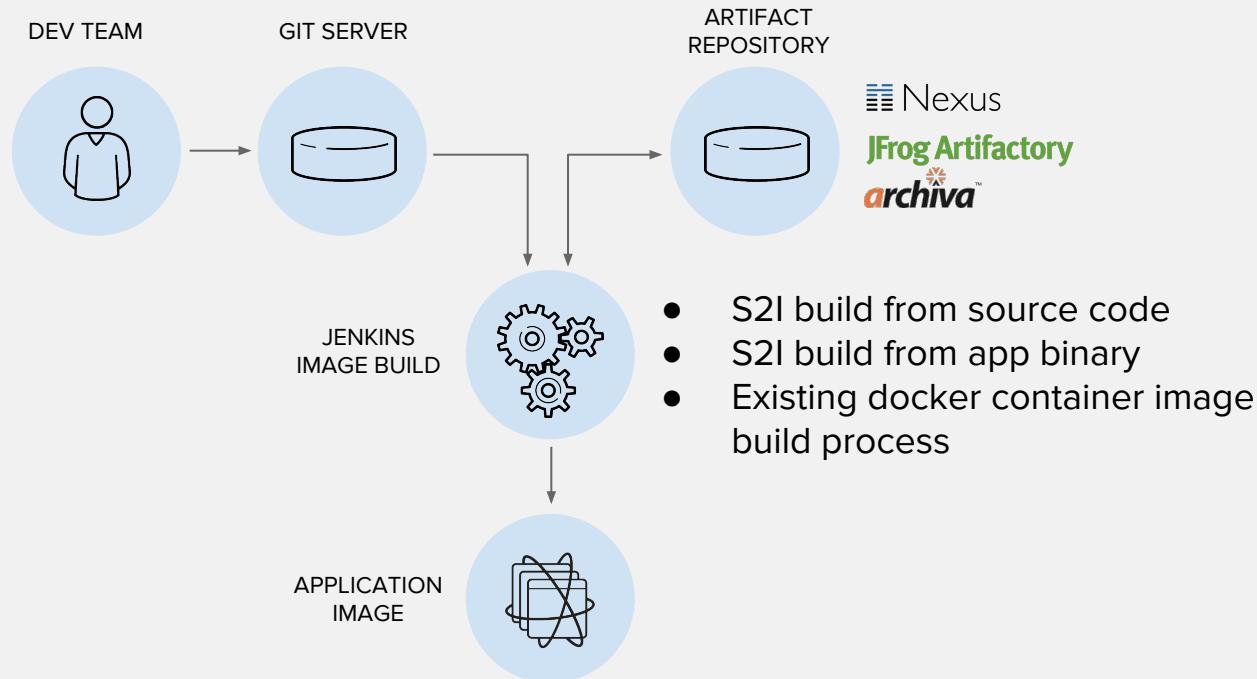
— Average: 1m 55s

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 time taken for each stage is indicated below the bar.

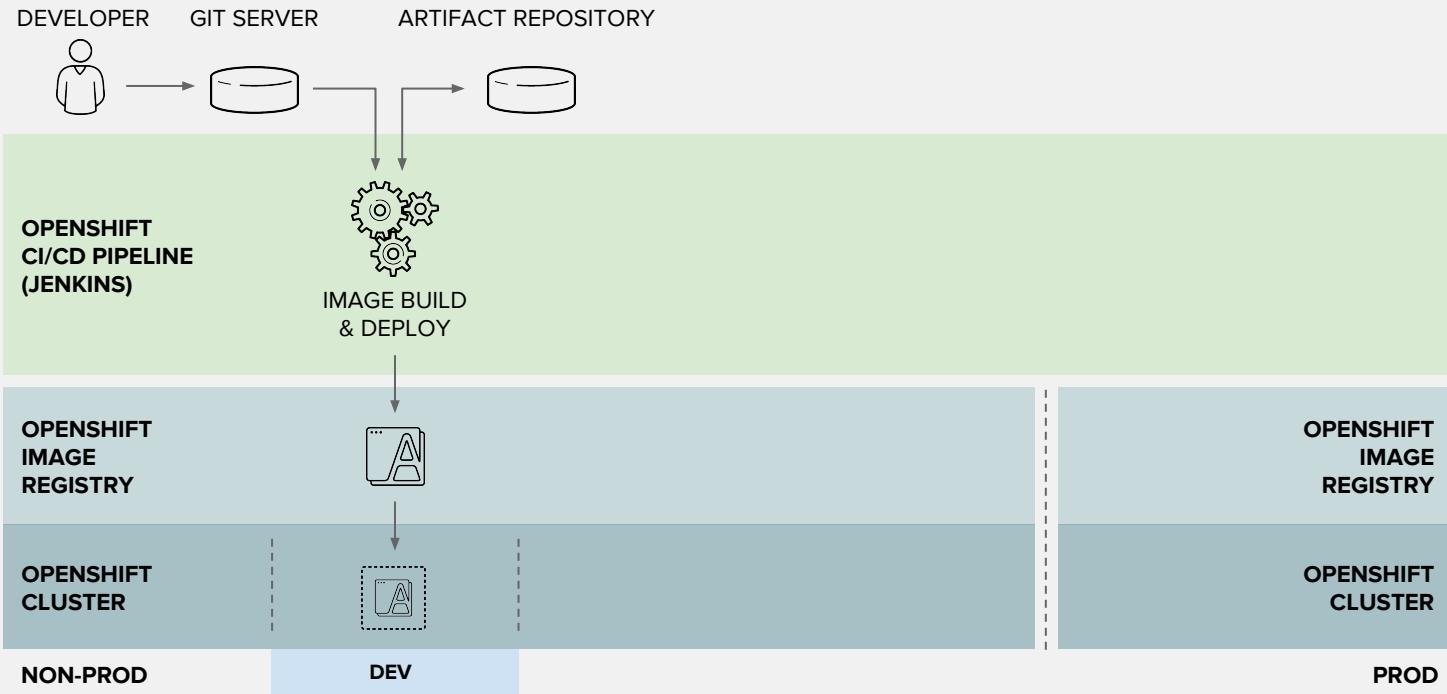
Build	Stage	Time
Build #11	build app	25s
	build image	16s
	deploy	45s
Build #10	build app	26s
	build image	16s
	deploy	47s

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

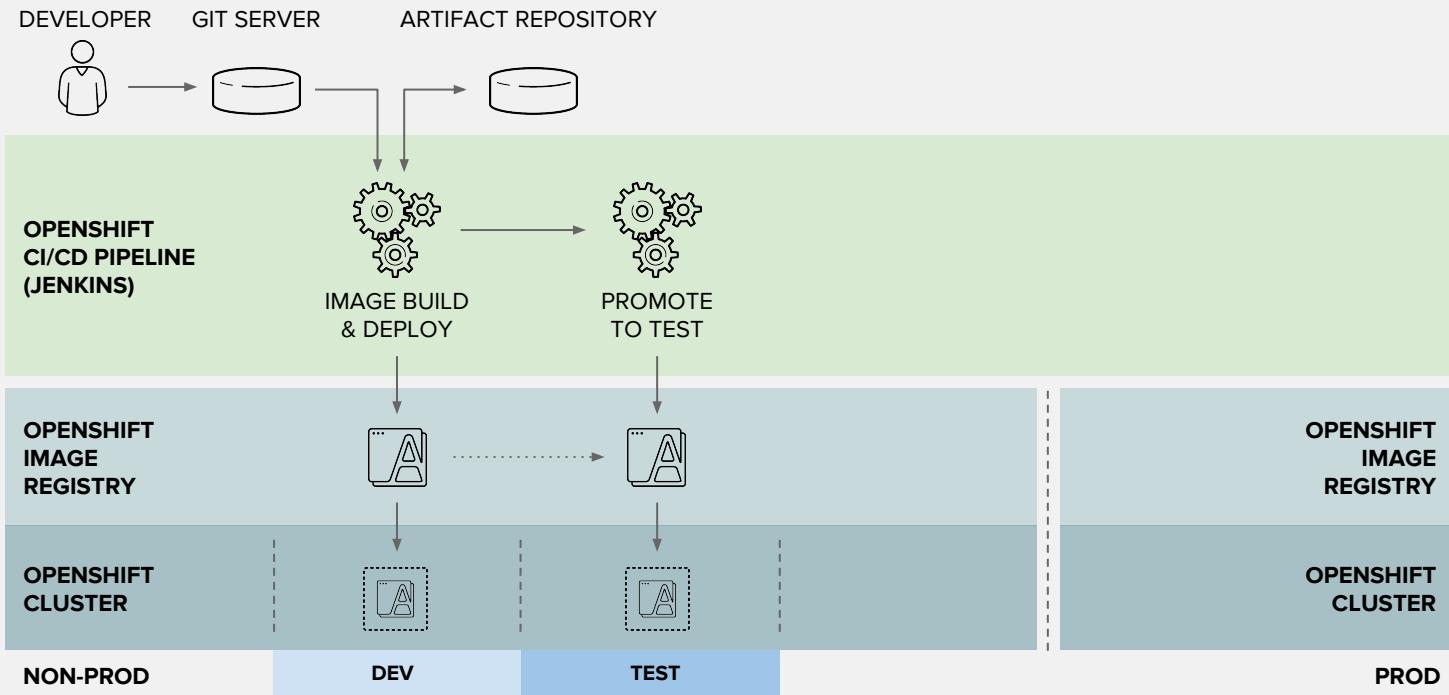
CONTINUOUS DELIVERY PIPELINE



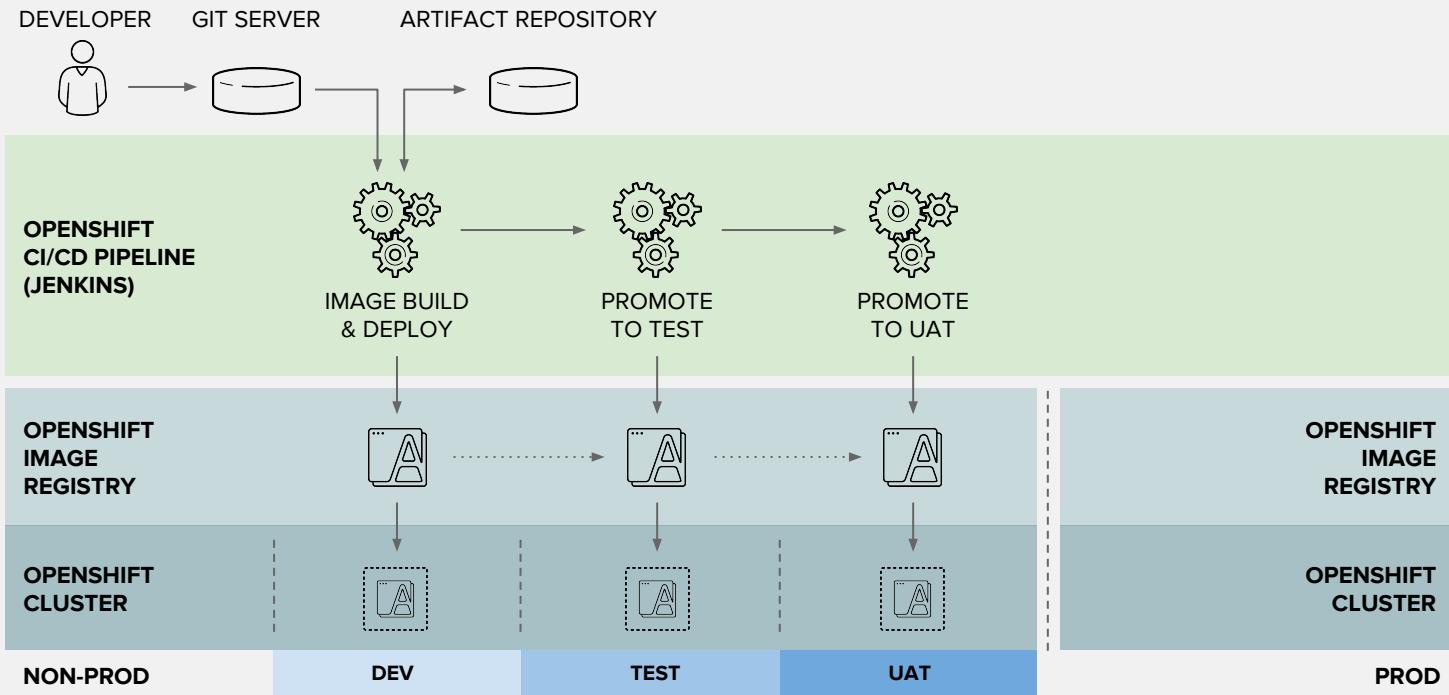
CONTINUOUS DELIVERY PIPELINE



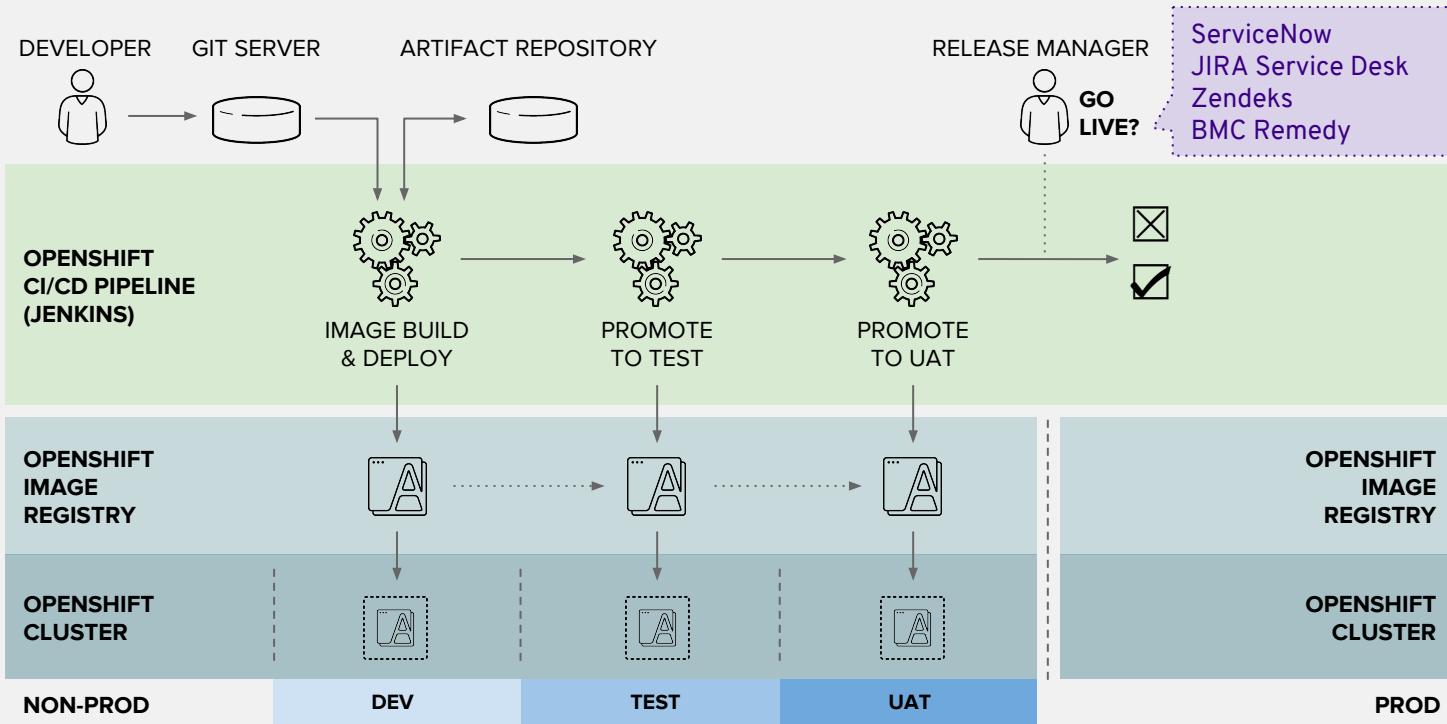
CONTINUOUS DELIVERY PIPELINE



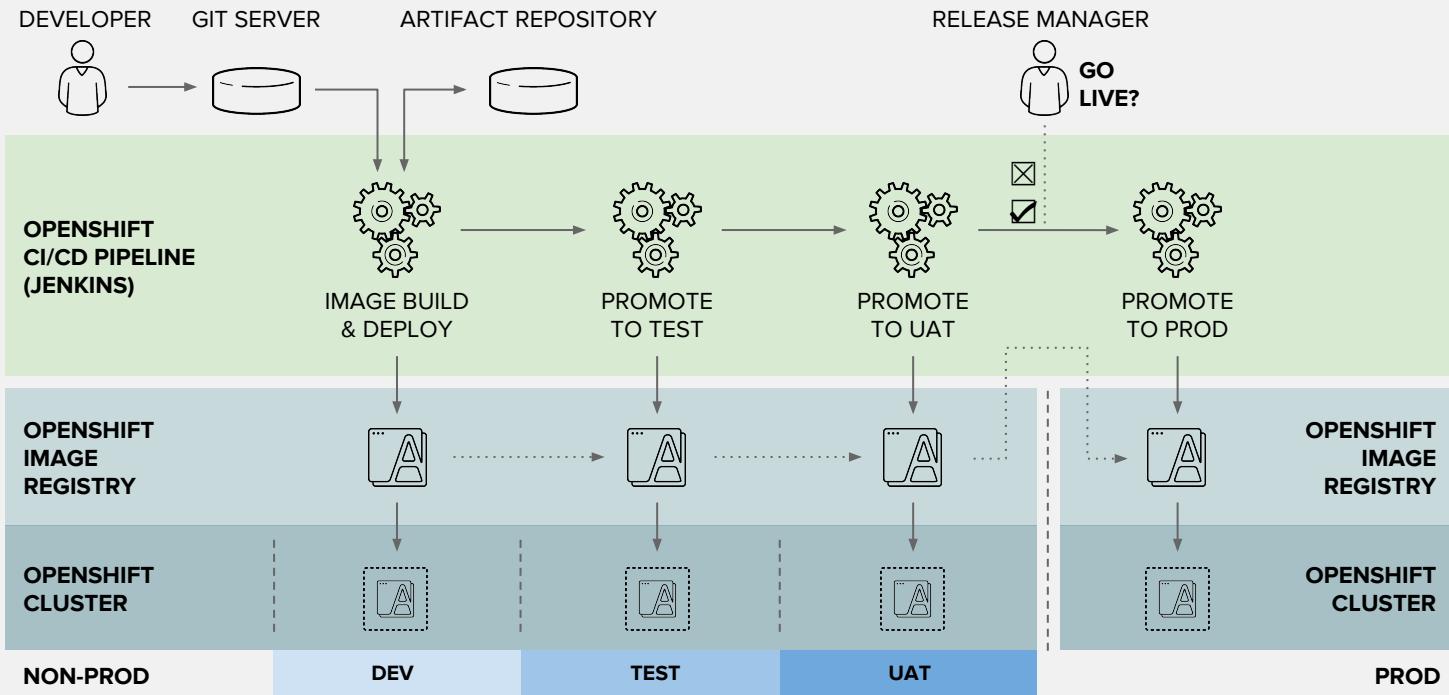
CONTINUOUS DELIVERY PIPELINE



CONTINUOUS DELIVERY PIPELINE



CONTINUOUS DELIVERY PIPELINE

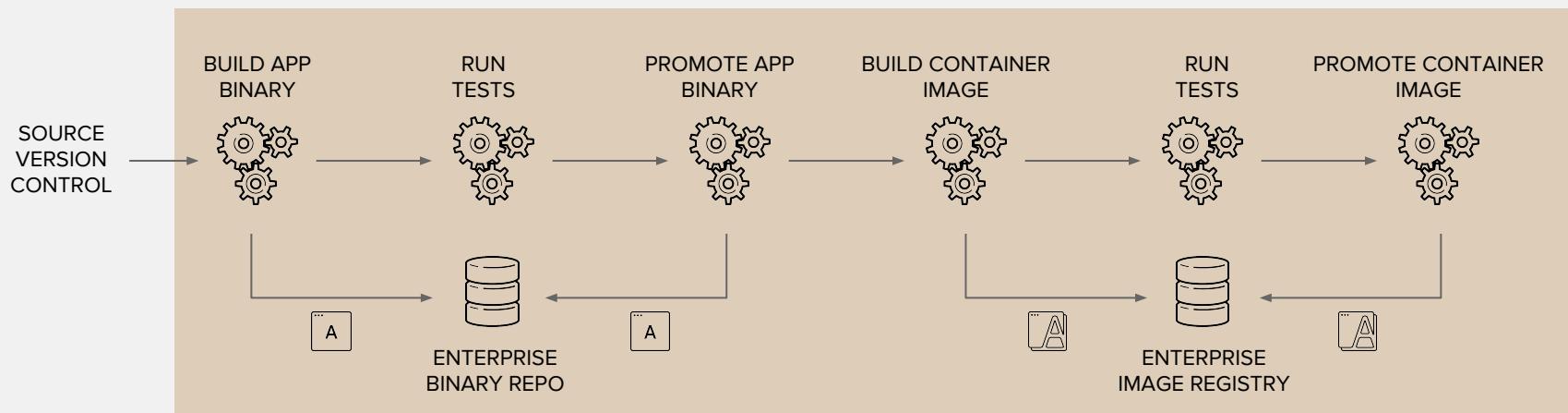




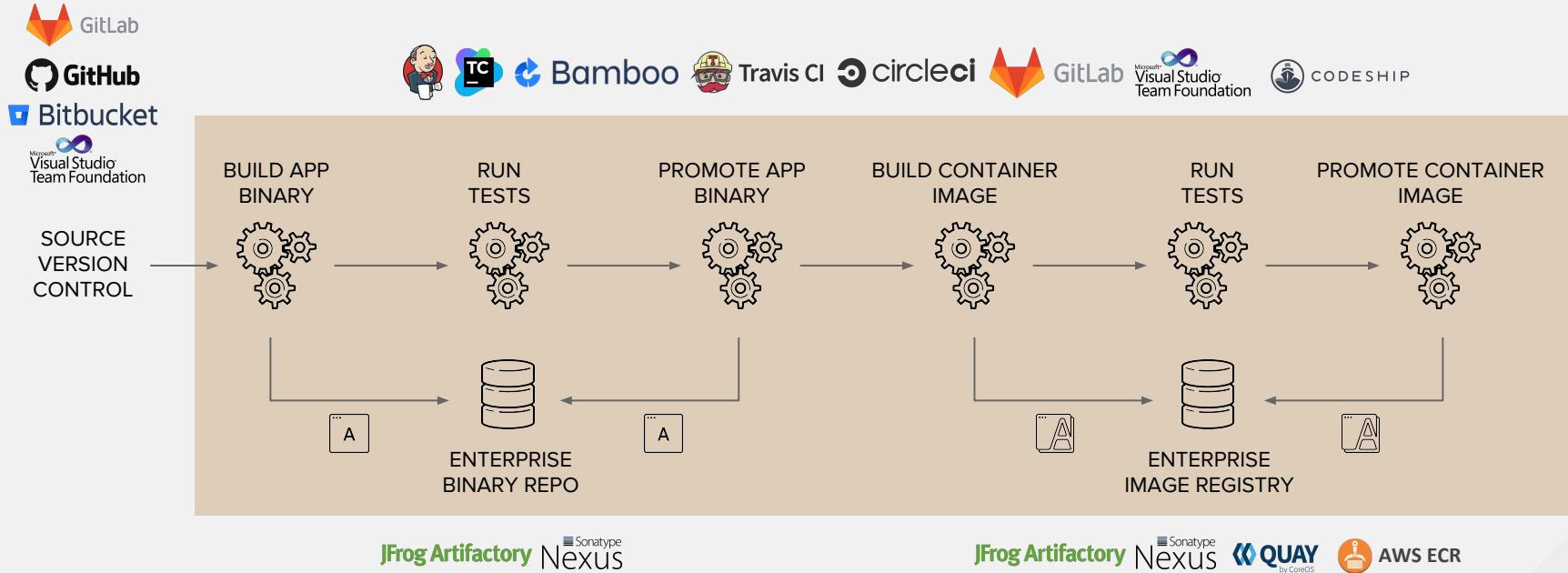
BUT...

**SOME TEAMS ALREADY HAVE
AUTOMATED DELIVERY PIPELINES**

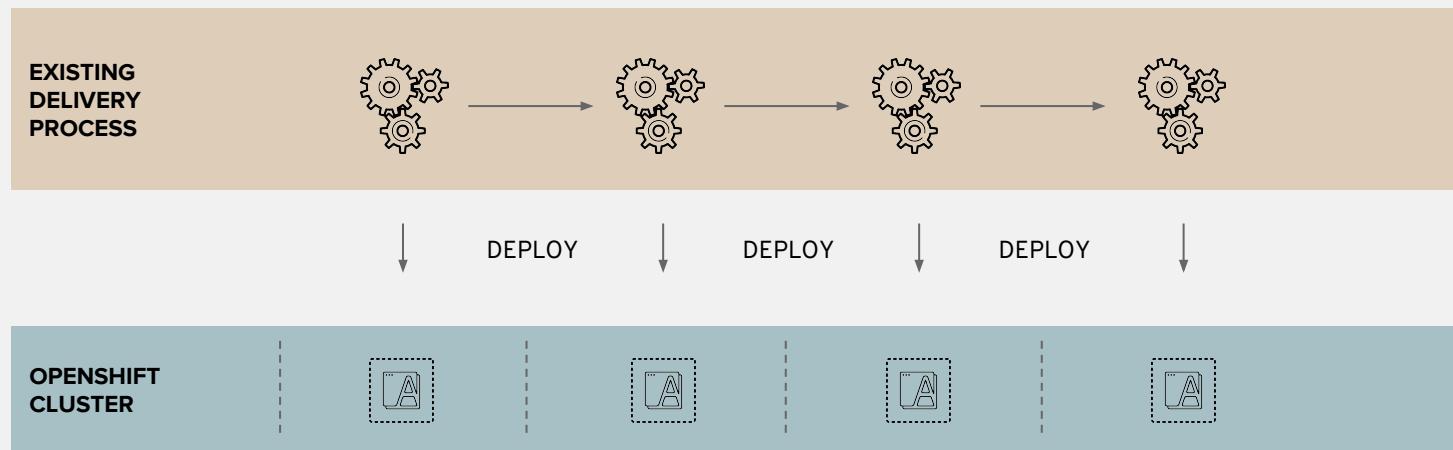
WHAT IF THERE ARE EXISTING DELIVERY PROCESSES?



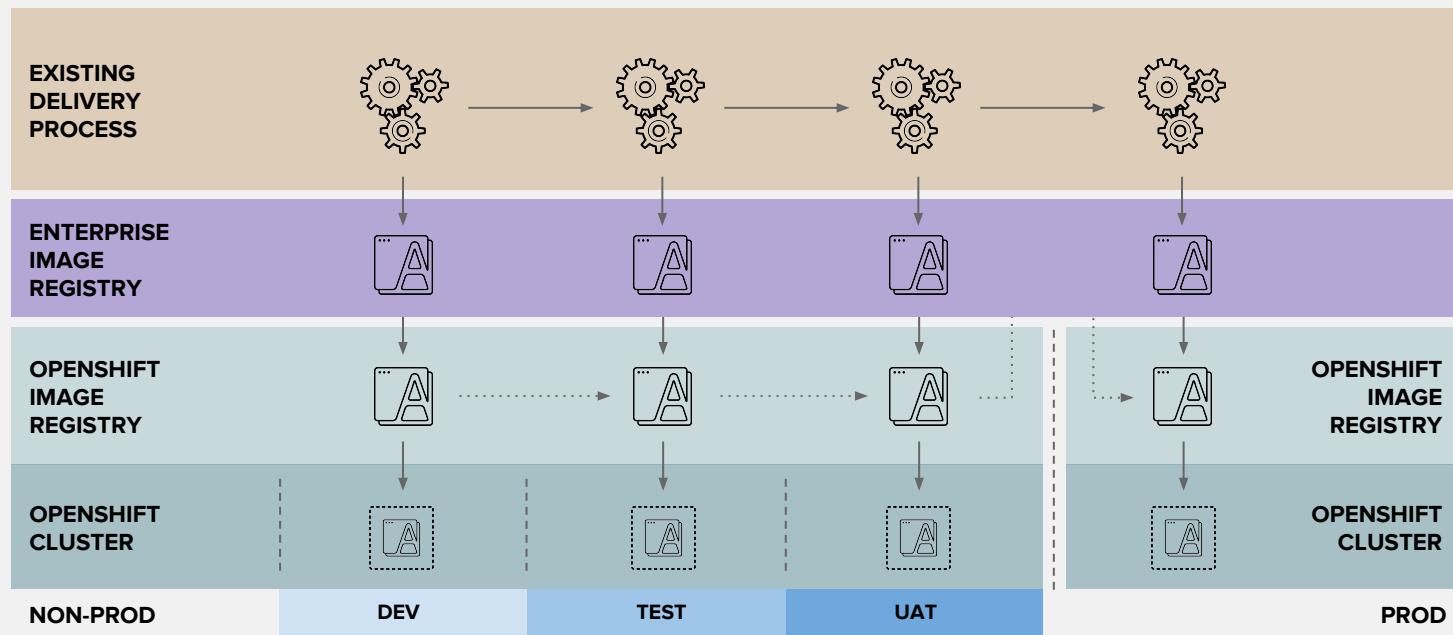
WHAT IF THERE ARE EXISTING DELIVERY PROCESSES?



ENRICHING EXISTING DELIVERY PROCESSES WITH OPENSHIFT



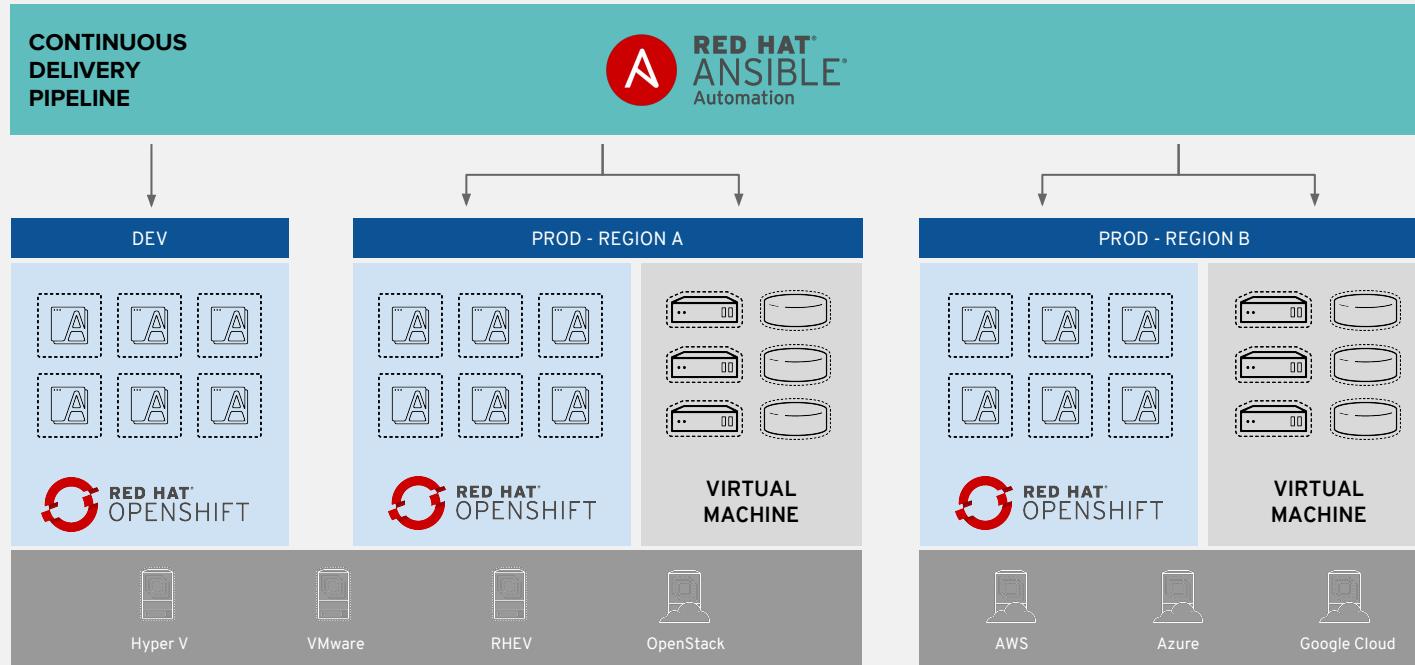
ENRICHING EXISTING DELIVERY PROCESSES WITH OPENSHIFT



HYBRID APPLICATION AUTOMATION WITH OPENSHIFT AND ANSIBLE

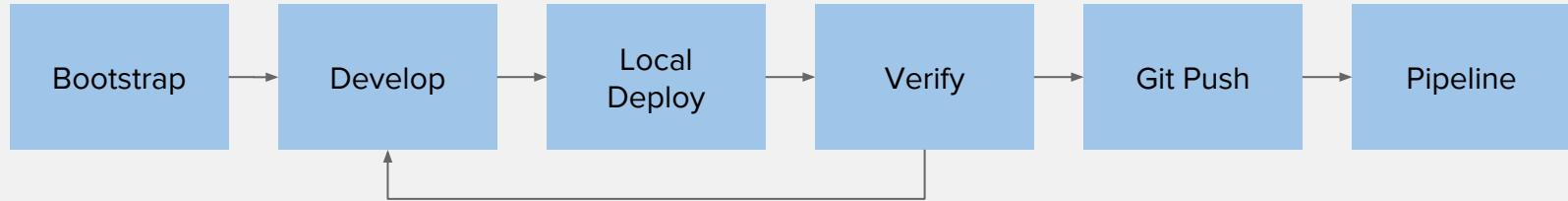


HYBRID APPLICATION AUTOMATION WITH OPENSHIFT AND ANSIBLE

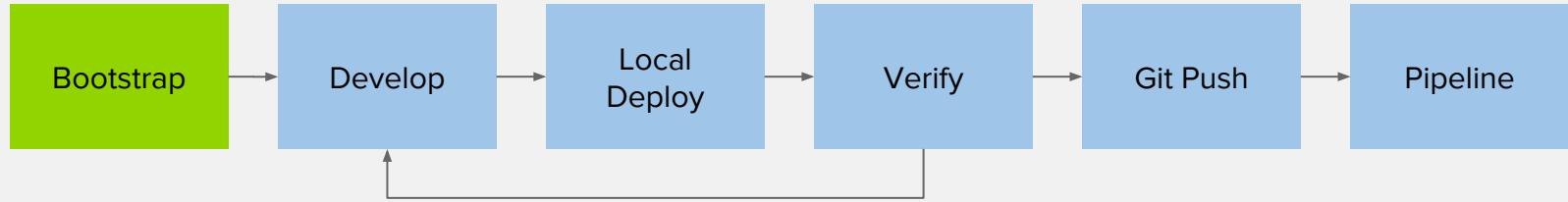


DEVELOPER WORKFLOW

LOCAL DEVELOPMENT WORKFLOW



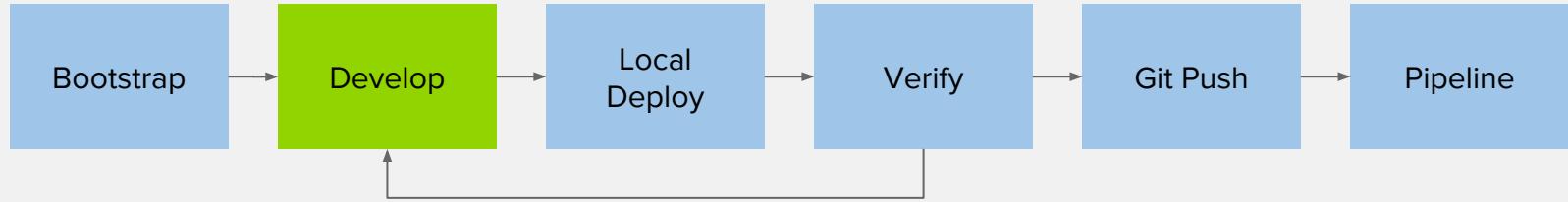
LOCAL DEVELOPMENT WORKFLOW



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

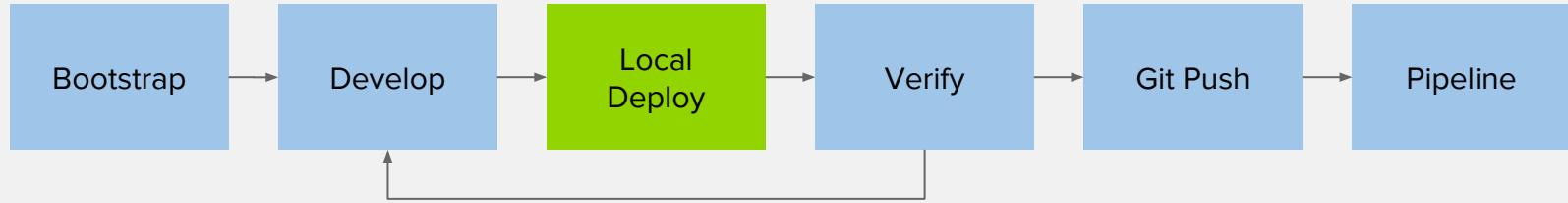
LOCAL DEVELOPMENT WORKFLOW



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

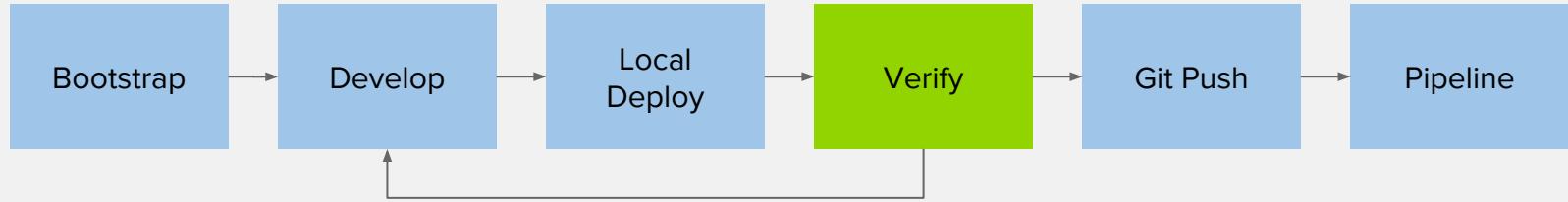
LOCAL DEVELOPMENT WORKFLOW



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

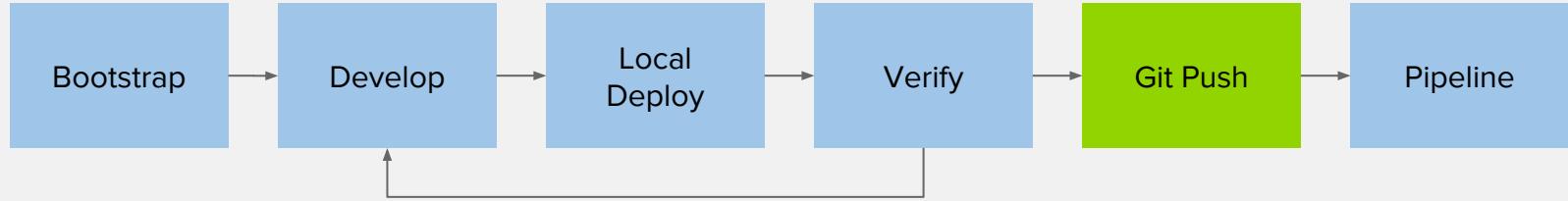
LOCAL DEVELOPMENT WORKFLOW



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

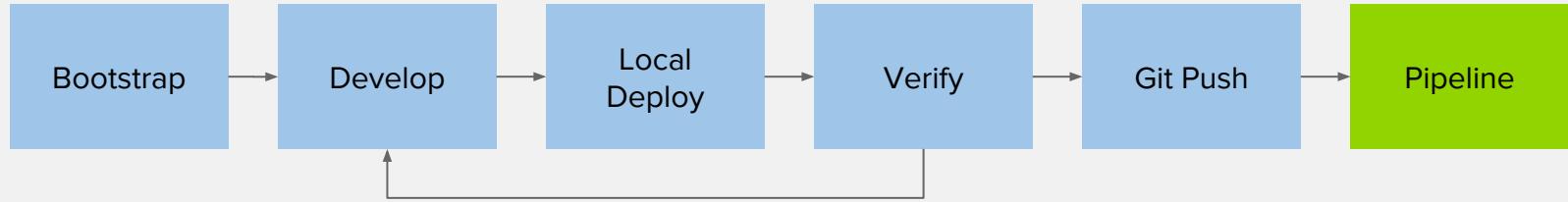
LOCAL DEVELOPMENT WORKFLOW



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

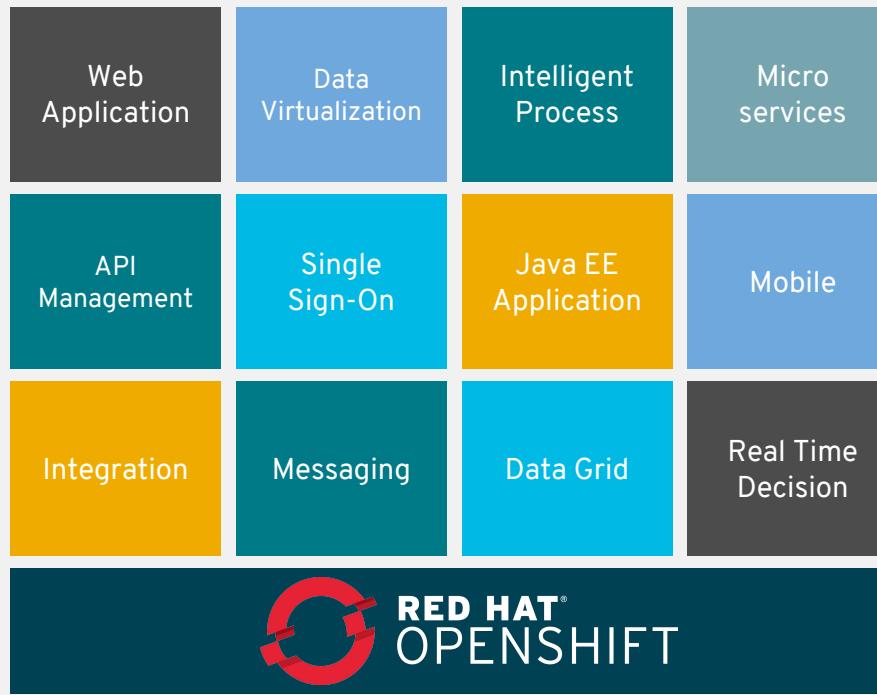


PIPELINE

- 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

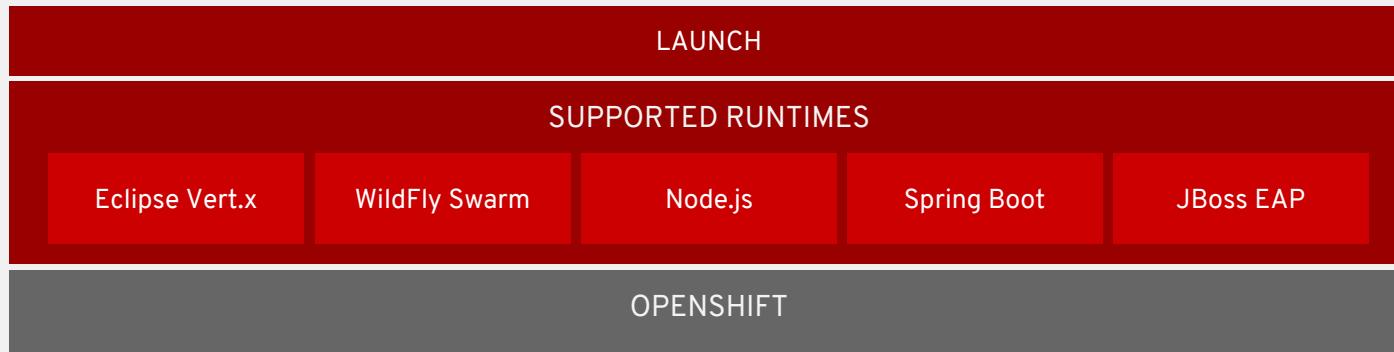
APPLICATION SERVICES

A PLATFORM THAT GROWS WITH YOUR BUSINESS



TRUE POLYGLOT PLATFORM

LANGUAGES	Java	NodeJS	Python	PHP	Perl	Ruby	.NET Core	Third-party Language Runtimes	
DATABASES	MySQL	PostgreSQL	MongoDB	Redis	<p>...and virtually any docker image out there!</p>				CrunchyData GitLab Iron.io Couchbase Sonatype EnterpriseDB NuoDB Fujitsu and many more
WEB SERVERS	Apache HTTP Server	nginx	Varnish	Phusion Passenger	Tomcat	Third-party App Runtimes			
MIDDLEWARE	Spring Boot	Wildfly Swarm	Vert.x	JBoss Web Server	JBoss EAP	JBoss A-MQ	JBoss Fuse	Third-party Middleware	
	3SCALE API mgmt	JBoss BRMS	JBoss BPMS	JBoss Data Virt	JBoss Data Grid	RH Mobile	RH SSO	Third-party Middleware	



Modern, Cloud-Native Application Runtimes and
an Opinionated Developer Experience

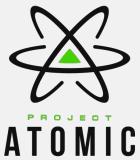
MICROSERVICES INFRASTRUCTURE: ISTIO SERVICE MESH

REFER TO OFFICIAL
ISTIO PRESENTATION



OPENSHIFT 4

IMMUTABLE INFRASTRUCTURE WITH RED HAT COREOS



RED HAT®
CoreOS

- Minimal Linux distribution
- Optimized for running containers
- Decreased attack surface
- Over-the-air automated updates
- Immutable foundation for OpenShift
- Bare-metal and cloud host configuration

AUTOMATED OPERATIONS

Fully automated day-1 and day-2 operations for Kubernetes

INSTALL

DEPLOY

HARDEN

OPERATE

AUTOMATED OPERATIONS

Infra provisioning

Full-stack deployment

Secure defaults

Multi-cluster aware

Embedded OS

On-premises and cloud

Network isolation

Monitoring and alerts

Unified experience

Audit and logs

Full-stack patch & upgrade

Signing and policies

Zero downtime upgrades

Vulnerability scanning

OPERATOR AND DEVELOPER CONSOLES

The image displays two side-by-side screenshots of the OpenShift Cluster Console and Application Console interfaces.

Cluster Console (Left):

- Navigation:** Shows a sidebar with links for Stateful Sets, Jobs, Cron Jobs, Pods, Build Configs, Builds, Image Streams, Config Maps, Secrets, Resource Quotas, and Horizontal Pod Autoscalers (HPAs).
- Projects Section:** A table titled "Projects" lists 17 entries. Each entry includes a small icon, a name (e.g., PR_00, PR_11, PR_99, PR_abcd, bens-app-operator, bens-project, bens-project-2, chris-project, default, defgh, Jeff-project, jhadvig, jons-project, kube-public, kube-system), its status (Active), requester (e.g., benjaminapetersen, admin@example.com, TheRealJon), and labels (No labels). A "Create Project" button is located at the top left of the table.

Application Console (Right):

- My Projects Section:** A table titled "My Projects" lists 12 projects. Each row shows the project name, a brief description, and creation date. A "Create Project" button is located at the top right of the table.
- Project Details Examples:**
 - CoolStore:** cptech - created by ssadeghi@redhat.com a month ago
 - PetStore DEV:** created by ssadeghi@redhat.com a year ago
 - PetStore STAGE:** created by ssadeghi@redhat.com a year ago
 - CI/CD:** created by scitronp@redhat.com 10 months ago
 - Jenkins Workshop:** created by ssadeghi@redhat.com 4 months ago
 - Roadshow:** created by ssadeghi@redhat.com a year ago
 - Payroll Integrations DEV:** created by ssadeghi@redhat.com a year ago
 - Payroll Integrations PROD:** created by scitronp@redhat.com 10 months ago
 - Kubevirt Demo:** created by ssadeghi@redhat.com 4 months ago
 - Tasks List App:** created by ssadeghi@redhat.com a year ago

OPERATOR CONSOLE

The image displays two side-by-side screenshots of the OpenShift Operator Console interface.

Left Screenshot (Services View):

- Header:** okd Cluster Console, Project: all projects, Siamak Sadeghianfar.
- Left Sidebar:** Overview, Workloads, Networking (Ingress, Routes), Services (selected), Monitoring, Administration (Projects, Namespaces, Nodes, Persistent Volumes, Service Accounts, Storage Classes, Roles, Role Bindings, CRDs).
- Section:** Services
- Buttons:** Create Service, Filter Services by name...
- Table:** Shows a list of services with columns: NAME, NAMESPACE, LABELS, POD SELECTOR, LOCATION.
- Sample Data:**
 - alertmanager-main (openshift-monitoring) - alertmanager=main
 - alertmanager-operated (openshift-monitoring) - operated-alertmanager=true
 - console (openshift-console) - app=openshift-console, component=ui
 - docker-registry (default) - docker-registry=default
 - etcd-restore-operator (olm-dev) - na... =etcd-operator-alm-ow...
 - example (chris-project) - No labels, app=MyApp
 - example (tonys-project) - No labels, app=MyApp
 - example (jhadwig) - No labels, app=MyApp
 - grafana (openshift-monitoring) - No labels, app=grafana

Right Screenshot (Service Details View):

- Header:** okd Cluster Console, Project: jons-project, Siamak Sadeghianfar.
- Left Sidebar:** Overview, Workloads, Networking (Ingress, Routes), Services (selected), Monitoring, Administration (Projects, Namespaces, Nodes, Persistent Volumes, Service Accounts, Storage Classes, Roles, Role Bindings, CRDs).
- Section:** nodejs-ex
- Buttons:** Actions, Save Changes, Reload, Cancel, Download.
- Table:** Overview, YAML, Pods.
- YAML View (Selected):**

```
1 kind: Service
2 apiVersion: v1
3 metadata:
4   name: nodejs-ex
5   namespace: jons-project
6   selfLink: /api/v1/namespaces/jons-project/services/nodejs-ex
7   uid: b091842c-065b-11e8-a2ec-42010c8e0002
8   resourceVersion: 1001
9   creationTimestamp: '2018-08-02T21:40:35Z'
10+
11   labels:
12     app: nodejs-ex
13   annotations:
14     openshift.io/generated-by: OpenShiftWebConsole
15   spec:
16     ports:
17       - name: 8080-tcp
18         protocol: TCP
19         port: 8080
20         targetPort: 8080
21     selector:
22       deploymentconfig: nodejs-ex
23     clusterIP: 172.30.167.250
24     type: ClusterIP
25     sessionAffinity: None
26     status:
27       loadBalancer: {}
```

OPERATOR CONSOLE

okd Cluster Console ▾

Slamak Sadeghianfar ▾

Nodes

Compact Expand

NODE NAME ↑	STATUS
ui-preserve-ig-m-j9nk	✓ Ready
ui-preserve-ig-n-nwq4	✓ Ready
ui-preserve-ig-n-p6w6	✓ Ready
ui-preserve-ig-n-qqm4	✓ Ready

okd Cluster Console ▾

Slamak Sadeghianfar ▾

ui-preserve-ig-m-j9nk

Actions ▾

Overview YAML Pods Events

Node Overview

Node Conditions

TYPE	STATUS	REASON	UPDATED	CHANGED
NetworkUnavailable	False	RouteCreated	Jan 1, 1970 1:00 am	Jul 27, 4:21 pm
OutOfDisk	False	KubeletHasSufficientDisk	a few seconds ago	Jul 27, 4:20 pm
MemoryPressure	False	KubeletHasSufficientMemory	a few seconds ago	Jul 27, 4:20 pm
DiskPressure	False	KubeletHasNoDiskPressure	a few seconds ago	Jul 27, 4:20 pm
PIDPressure	False	KubeletHasSufficientPID	a few seconds ago	Jul 27, 4:20 pm
Ready	True	KubeletReady	a few seconds ago	Jul 27, 4:23 pm

Images

NAME	SIZE
docker.io/openshift/origin-node:v3.11	1.31 GB
docker.io/openshift/origin-control-plane:v3.11.0	830.6 MB
docker.io/openshift/origin-deployer:v3.11.0-alpha.0	673.3 MB
quay.io/coreos/cluster-monitoring-operator:v0.1.0	460.4 MB

INFRASTRUCTURE MONITORING

ckd Cluster Console ▾ Siamak Sadeghianfar ▾

Overview Project: all projects ▾

Cluster Status

Health

Kubernetes API	OpenShift Console	Alerts Firing	Crashlooping Pods
UP All good	UP All good	6 Alerts	0 Pods

Control Plane Status

API Servers Up	Controller Manage...	Schedulers Up	API Request Succe...
100%	100%	100%	97%

Capacity Planning

CPU Usage	Memory Usage	Disk Usage	Pod Usage
99%	27%	4%	16%

Events

All Types ▾ All Categories ▾ Filter Events by message...

Streaming events... Showing 500 of 500+ events

nodejs-auth-5d7f7876886-2z8w Generated from kubelet on ui-preserve-ig-n-p6w6
Generated from kubelet on ui-preserve-ig-n-p6w6
Back-off pulling image "registry.gitlab.com/benjaminapetersen/nodejs-auth"

6 less than a minute ago 37514 times in the last 6 days

ckd Cluster Console ▾ Siamak Sadeghianfar ▾

Jobs Cron Jobs Pods Build Configs Builds Image Streams Config Maps Secrets Resource Quotas HPA

Documentation Full Documentation

From getting started with creating your first application, to trying out more advanced build and deployment techniques, these resources provide what you need to set up and manage your environment as a cluster administrator or an application developer.

Get Started with the CLI

With the OpenShift command line interface (CLI), you can create applications and manage projects from a terminal. Learn how to install and use the oc client tool.

Additional Support

- Interactive Learning Portal
- Local Development
- YouTube
- Blog

Software Info

Kubernetes	v1.11.0+d4cacc0
OpenShift	v3.11.0-alpha.0+10b4e56-568

Node Overview

ui-preserve-ig-m-j9nk

Overview YAML Pods Events Actions ▾

RAM CPU Number of Pods

Network In Network Out Filesystem

NODE NAME ui-preserve-ig-m-j9nk OPERATING SYSTEM Linux

EXTERNAL ID - ARCHITECTURE AMD64

NODE ADDRESSES External IP: 35.237.156.140 KERNEL VERSION 3.10.0-862.9.1.el7.x86_64

Hostname: ui-preserve-ig-m-j9nk BOOT ID 076464f5-a4f8-4b13-8b34-be6731900f20

Internal IP: 10.142.0.2

NODE LABELS node-role.kubernetes.io/infra=true beta.kubernetes.io/os=linux failure-domain.beta.kubernetes.io/zone=us-east1-c failure-domain.beta.kubernetes.io/region=us-east1

beta.kubernetes.io/master=true beta.kubernetes.io/instance-type=m1-standard-2 kubernetes.io/hostname=ui-preserve-ig-m-j9nk beta.kubernetes.io/arch=amd64 role=infra

ANNOTATIONS 1 Annotation ▾



redhat.

THANK YOU



plus.google.com/+RedHat



linkedin.com/company/red-hat



youtube.com/user/RedHatVideos



facebook.com/redhatinc



twitter.com/RedHatNews

DEPRECATED SLIDES

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

Replicators Pods Containers Services Routes Nodes VMs Hosts

Aggregated Node Utilization

Category	Value
Providers	1
Nodes	6
Containers	26
Pods	26
Services	43
Registries	3
Images	346
Projects	14
Routes	23

CPU Memory

83 Available of 84 Cores 95 Available of 152 GB

1 Cores Used 57 GB Used

Relationships

Container 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

Node Utilization

CPU Memory

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

Network Utilization Trend

1301 Kbps

New Image Usage Trend

Images

Last 30 Days

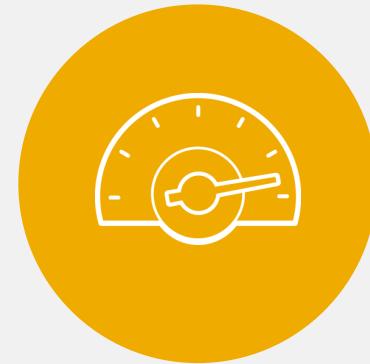
Pod Creation and Deletion Trends

Created Deleted

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

CPU (MHz)

Legend: Avg Used (Blue), Min Used (Red), Max Used (Green), Trend Max Used (Orange), All Non-VM Files (Teal), High Oper Range (Yellow)

Memory (MB)

Legend: Avg Used (Blue), Min Used (Red), Max Used (Green), Trend Max Used (Orange), Low % Oper Range (Teal), High Oper Range (Yellow)

Network I/O (Kbps)

Legend: Avg I/O (Blue), Min I/O (Red), Max I/O (Green), Trend Max I/O (Orange)

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

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)

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 notchecked
 informational unknown notselected
 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-euhe-2	18.3%	4	redhat
manageiq-euhe-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_U12	97.7%	8 GB
CF42_U11	97.7%	8 GB
manageiq-euhe-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

MICROSERVICES INFRASTRUCTURE: ISTIO SERVICE MESH

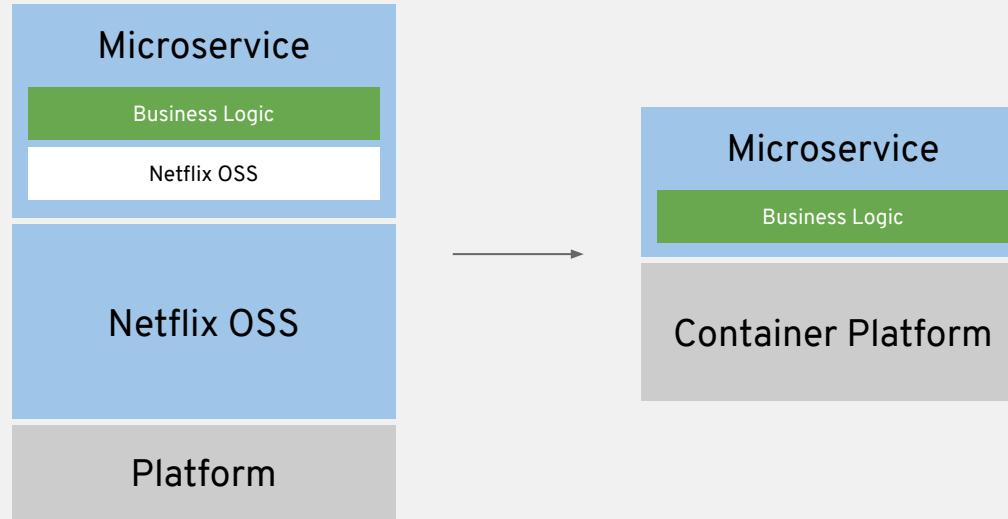
WHAT DO YOU NEED FOR MICROSERVICES?



WHAT YOU NEED FOR MICROSERVICES?



MICROSERVICES EVOLUTION

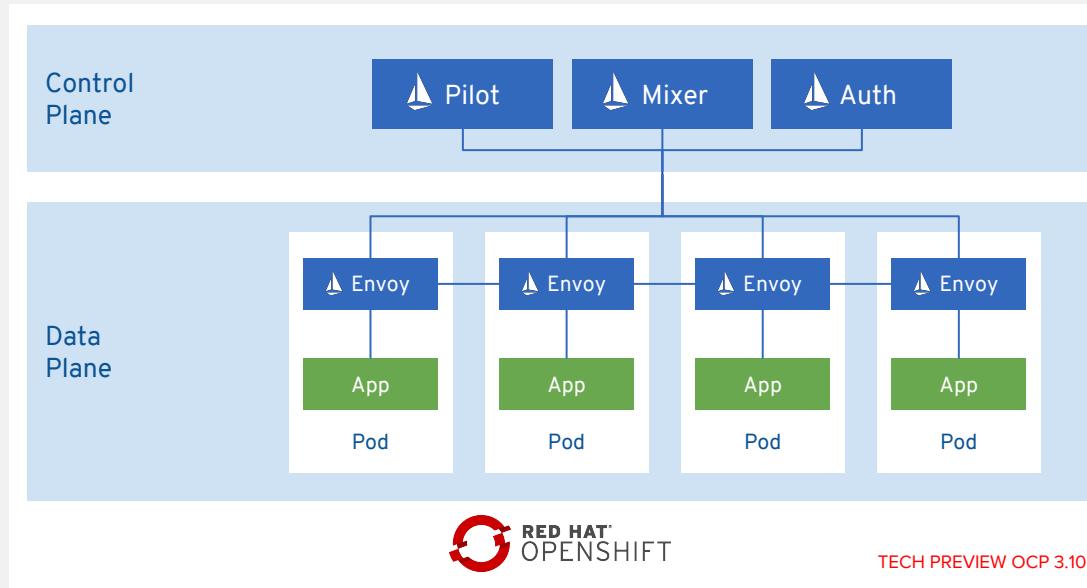


WHAT YOU NEED FOR MICROSERVICES?

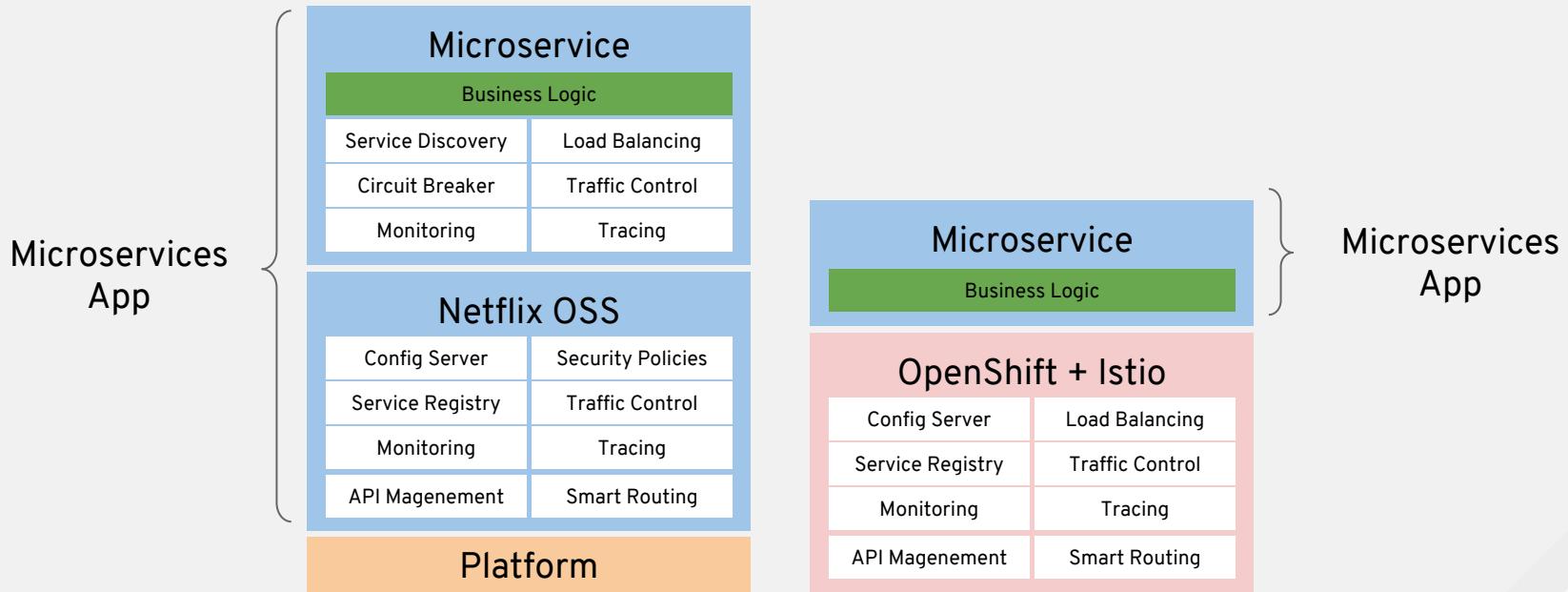


WHAT IS ISTIO?

a service mesh to connect, manage, and secure microservices



NETFLIX OSS VS ISTIO



CONTROL OUTGOING TRAFFIC SOURCE IP WITH EGRESS ROUTER

