



OpenShift Roadmap & Platform Services

Serverless, Pipelines, Service Mesh, Operators and more

Speakers



Rob Szumski

Sr. Manager of Product Management, OpenShift
Twitter: @robszumski
Lives in Richmond, VA



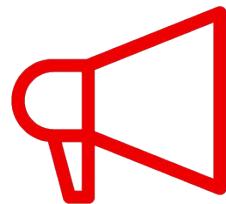
Siamak Sadeghianfar

Principal Product Manager, OpenShift
Twitter: @siamaks
Lives in Toronto, Canada



William Markito Oliveira

Sr. Manager of Product Management, OpenShift
Twitter: @william_markito
Lives in Austin, Texas



OpenShift 4 Themes



Platform Innovation

Kubernetes Operator Framework

Red Hat Enterprise Linux CoreOS

Expanded Workload Support

Enhanced Security



Management Innovation

Full Stack Automated Install & Upgrades

Autoscale Infrastructure

Connected Customer Experience

Multi-Cluster Management



Developer Innovation

OpenShift Service Mesh (Istio)

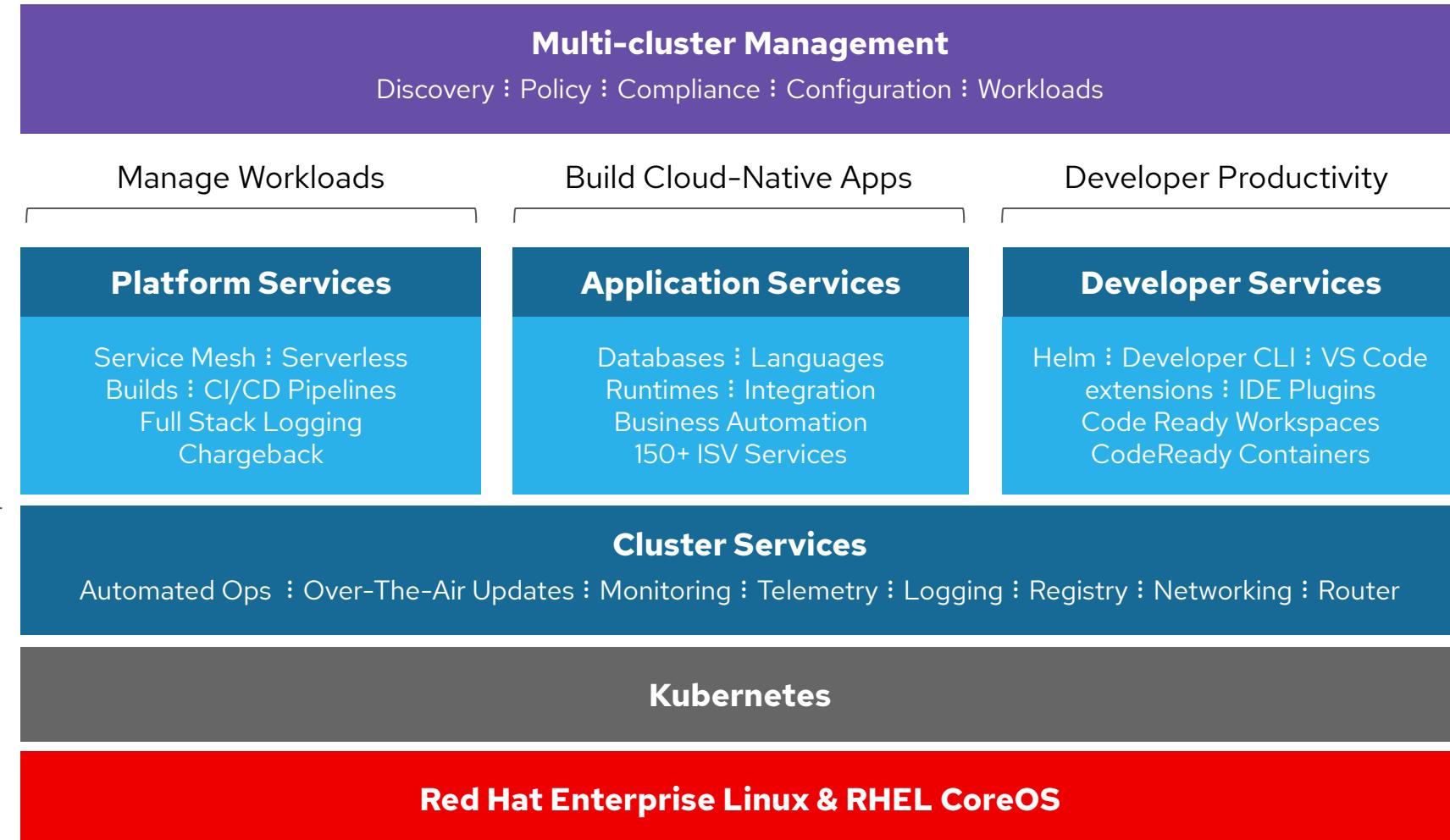
OpenShift Serverless (Knative)

OpenShift Pipelines (Tekton)

CodeReady Workspaces (Che)



OpenShift Container Platform



Physical



Virtual



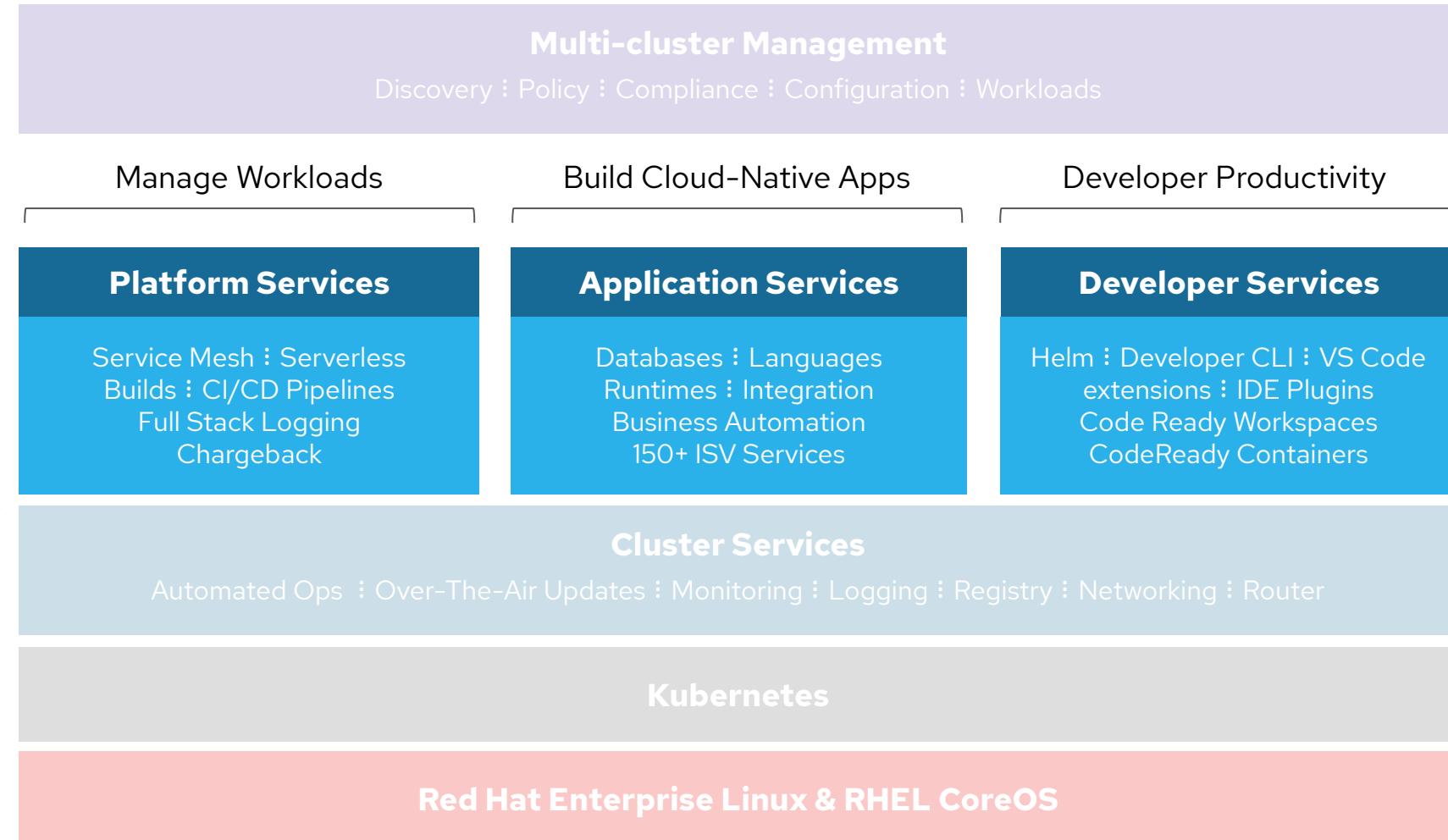
Private cloud



Public cloud



OpenShift Container Platform



Physical



Virtual



Private cloud



Public cloud



Next-generation cloud-native application stack



Istio - Service Mesh to connect, secure and observe services



Knative - Kubernetes-Native Serverless to enable hybrid FaaS



Tekton - Kubernetes-Native CI/CD for app build & deployment pipelines



Eclipse Che - Kubernetes-Native IDE for development & collaboration



Quarkus - Kubernetes-Native Java stack for next-generation apps

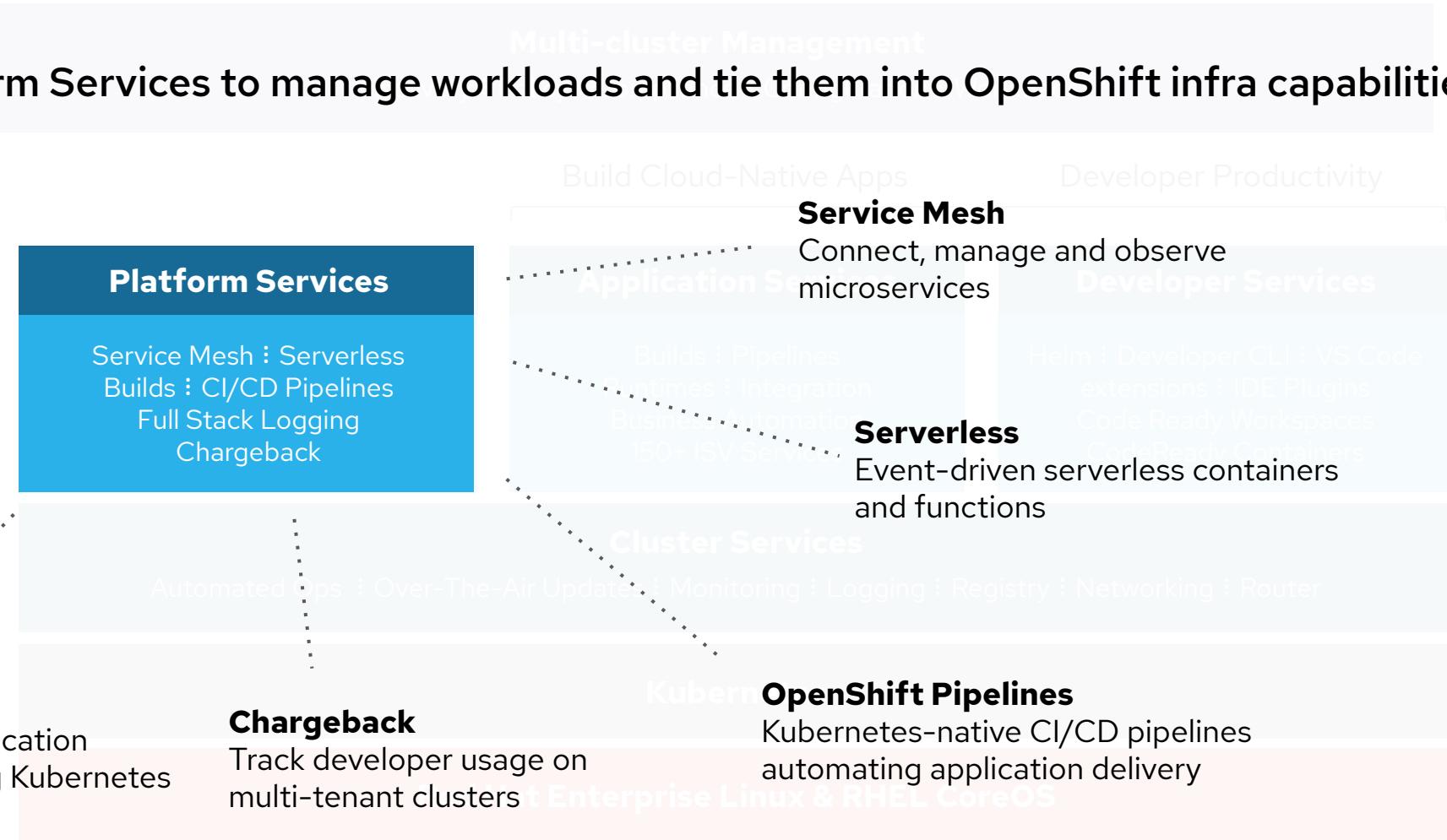


Operator Framework for building managed services on Kubernetes

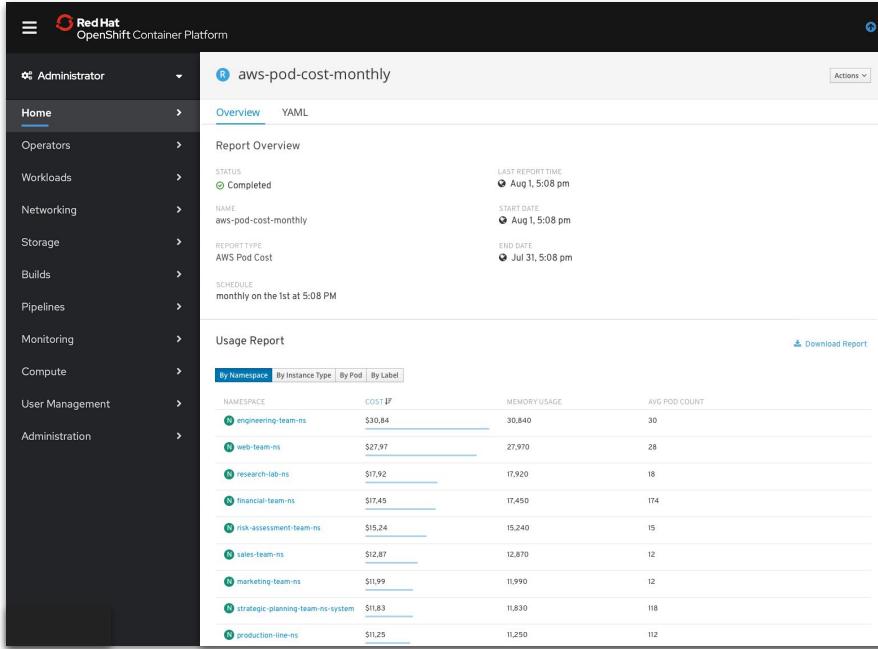
Platform Services

OpenShift Container Platform

Platform Services to manage workloads and tie them into OpenShift infra capabilities



Metering & Chargeback



The screenshot shows the Red Hat OpenShift Container Platform web interface. On the left, there's a dark sidebar with a navigation menu. The main area displays a report titled 'aws-pod-cost-monthly'. At the top of the report, it says 'Report Overview' and shows the status as 'Completed'. It includes fields for 'NAME', 'REPORTTYPE', and 'SCHEDULE'. Below this is a 'Usage Report' section with a table showing costs by namespace. The table has columns for 'NAMESPACE', 'COST IF', 'MEMORY USAGE', and 'AVG POD COUNT'. The namespaces listed are: engineering-team-ns, web-team-ns, research-lab-ns, financial-team-ns, risk-assessment-team-ns, sales-team-ns, marketing-team-ns, strategic-planning-team-ns-system, and production-line-ns.

NAMESPACE	COST IF	MEMORY USAGE	AVG POD COUNT
engineering-team-ns	\$30.84	30,840	30
web-team-ns	\$27.97	27,970	28
research-lab-ns	\$17.92	17,920	18
financial-team-ns	\$17.45	17,450	174
risk-assessment-team-ns	\$15.24	15,240	15
sales-team-ns	\$12.87	12,870	12
marketing-team-ns	\$11.99	11,990	12
strategic-planning-team-ns-system	\$11.83	11,830	118
production-line-ns	\$11.25	11,250	112

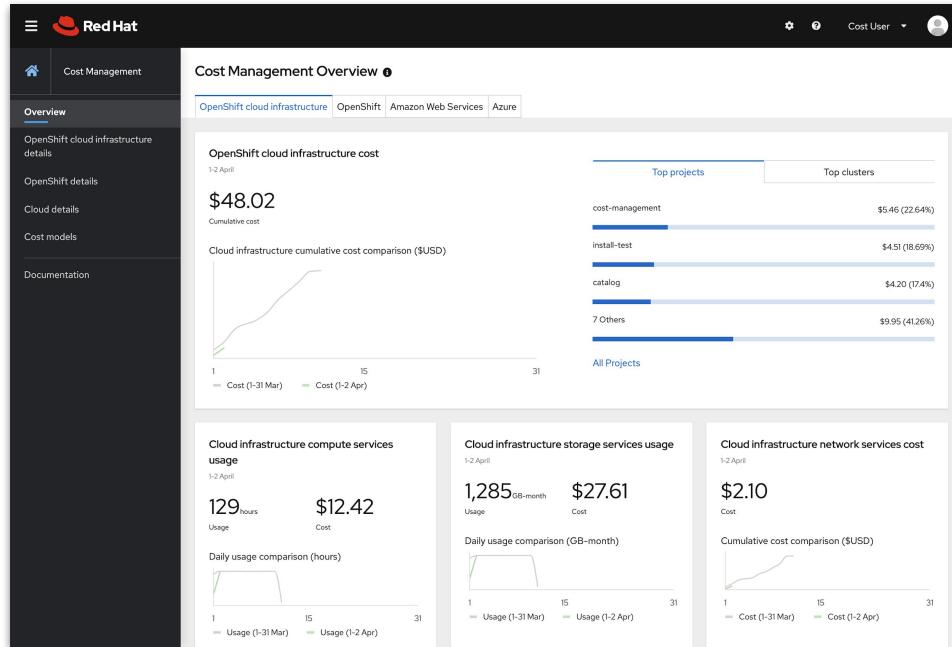
OpenShift Metering

- Track usage at Node, Pod, Namespace level
- Installed from OperatorHub
- Scheduled reports that can be queried by API
- Create custom reports for label queries and more

Used as the technical foundation for Red Hat Cost Management and Red Hat Marketplace.

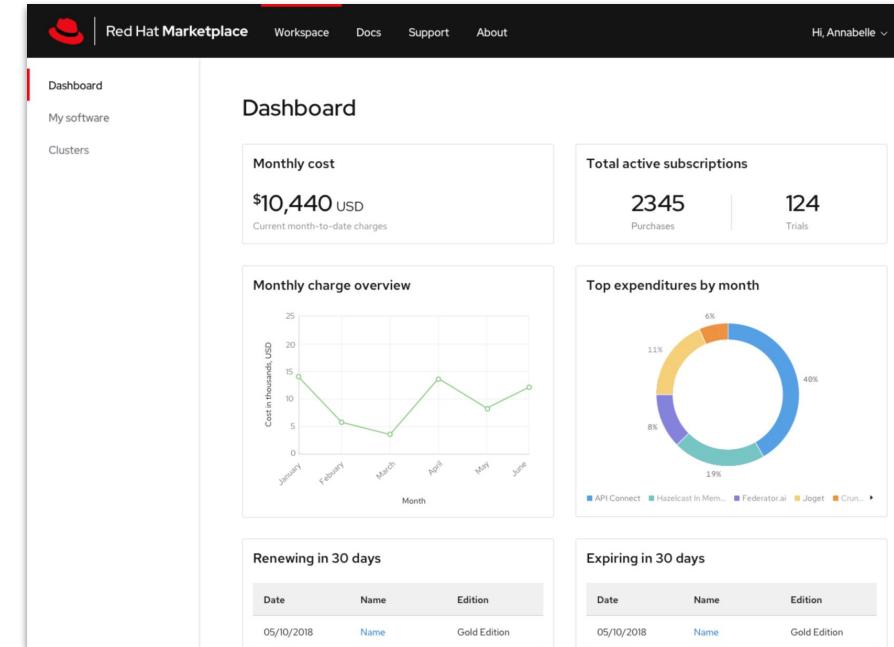
Metering & Chargeback

Red Hat Cost Management



- Cost visibility for all stakeholders
- Cost mapping to teams
- Combines cloud IaaS costs and Red Hat subscription usage

Red Hat Marketplace



- Install and track software usage across clusters
- Developers can find the right certified software solutions quickly in one place
- Simplify approvals, delegation—with visibility and control over software use

Operator Developer Enhancements

Improved workflow for offline catalogs and upgrade testing



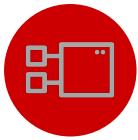
Single "Operator" object

Simplify registering/running Operators for all users, but is very helpful during testing and development.



Semver-based upgrade logic

Add a simpler mechanism to track upgrade logic to sit alongside the options for a complex graph



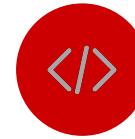
Bundle custom functional tests

Provide developers a tool to package up tests that cover important use-cases for their software.



Build catalogs with Kubernetes tools

Use tools that you are familiar with to package up your Operator for consumption.



Improve tools for offline mirroring of images

Easily mirror all of the content required to run Operators behind a firewalled network.

COMING SOON:



CLOUD NATIVE
SANDBOX



OpenShift Service Mesh

Enhanced Application Flexibility through Istio, Kiali, and Jaeger



Enhanced gRPC support for efficient microservice development



Managed Updates to the Control Plane via Operator Driven Deployment



Enhanced Authorization Traffic Controls



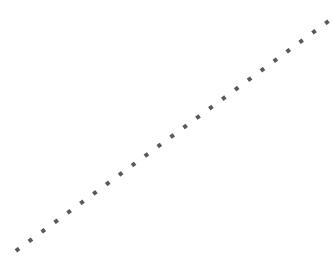
Increased visibility with new Kiali and Jaeger functionality



Increased Traffic Management Capabilities



Time-range selection in Kiali





OpenShift Service Mesh

Istio Enhancements in Detail

- Streamlined mutual TLS setup
- Improved troubleshooting with **istioctl analyze**
- Mirroring a percentage of traffic
- Automatic rotation of expired root certificates managed by Citadel
- Authorization Policy graduating to **beta**
- Adoption of OpenAPI v3 schemas for Istio CRDs
- More efficient Regular Expression handling for header checks

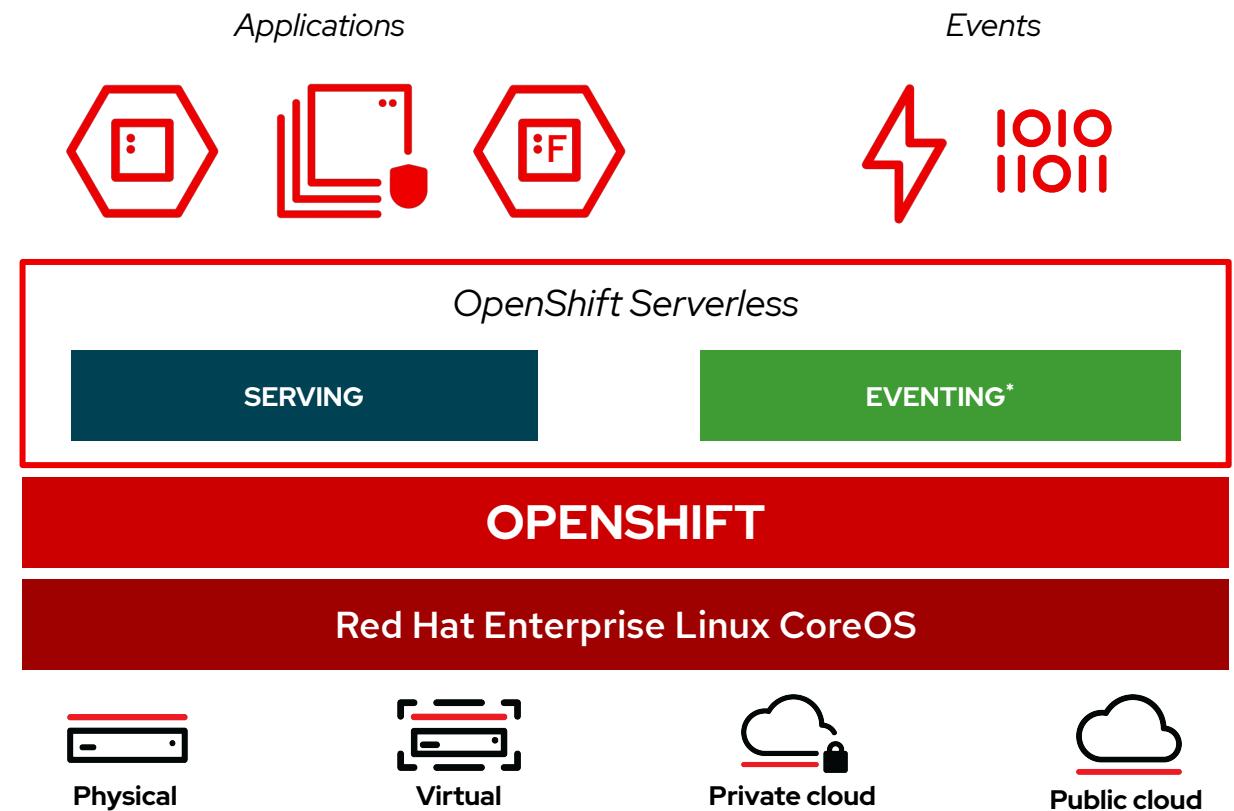


Generally Available

OpenShift Serverless

Event-driven serverless containers and functions

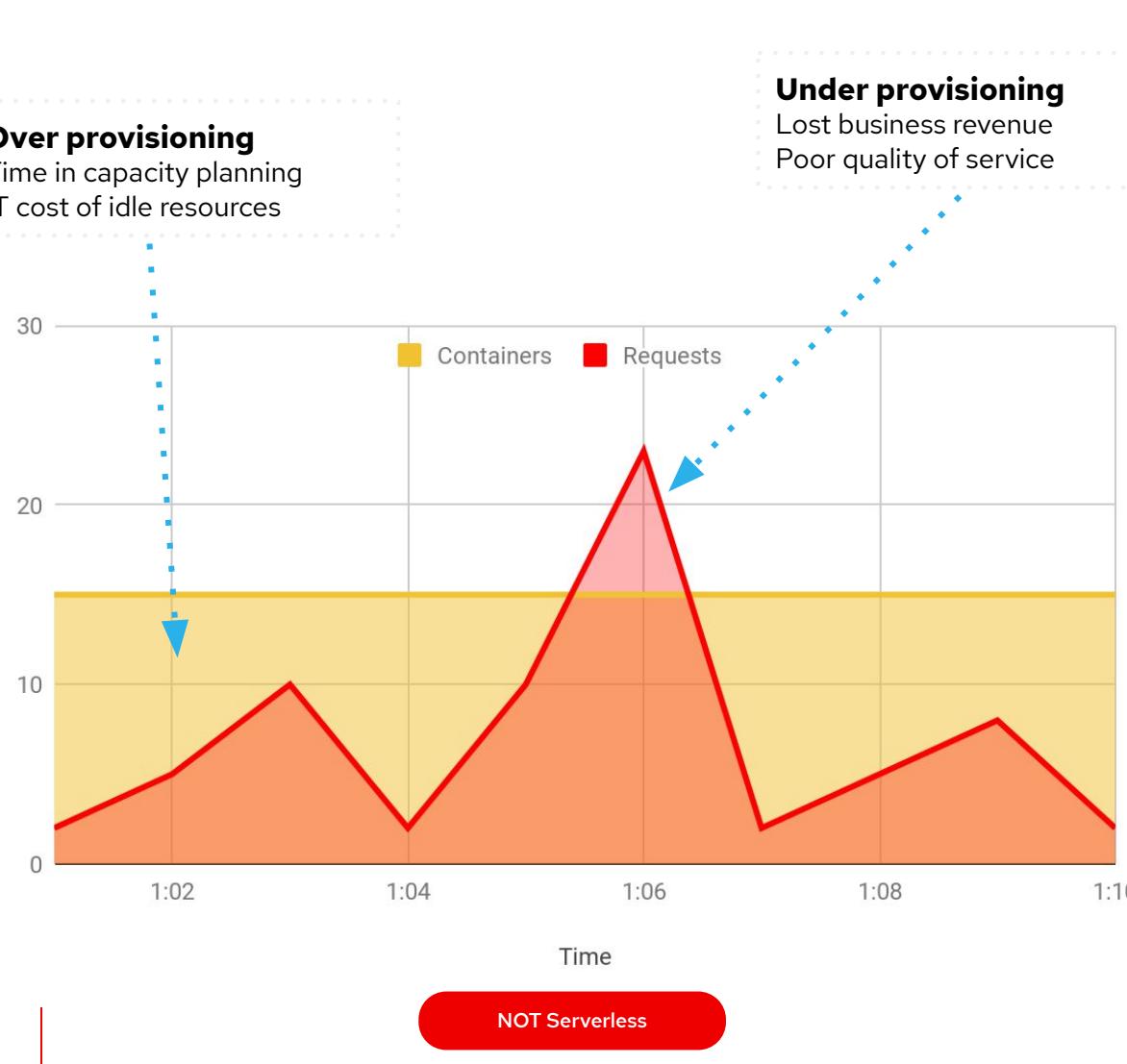
- Deploy and run **serverless containers**
- Use any programming language or runtime
- Modernize existing applications
- Powered by a rich ecosystem of event sources
- Manage serverless apps natively in Kubernetes
- Based on open source project **Knative**
- Run anywhere OpenShift runs



Serverless Operational Benefits

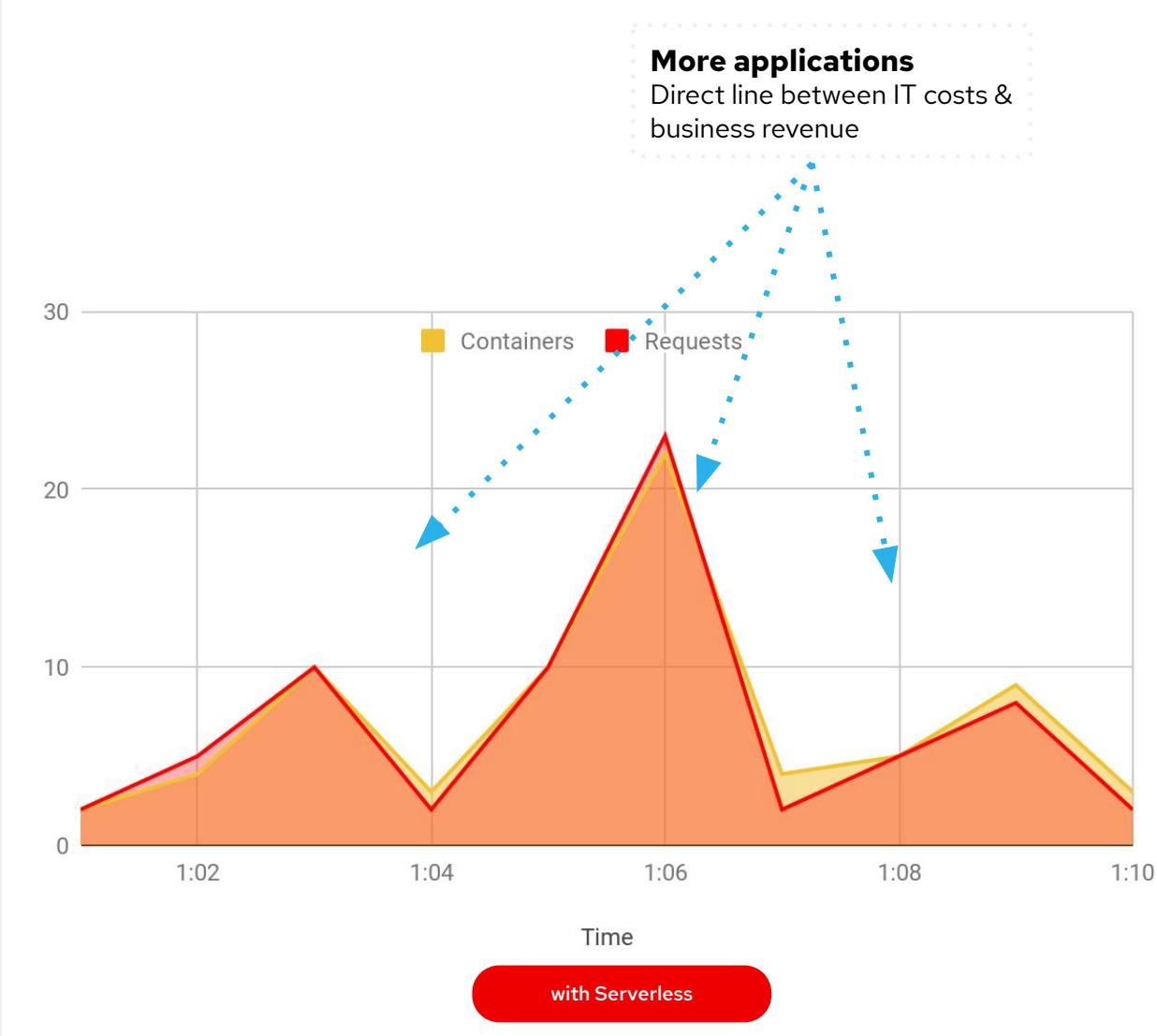
Over provisioning

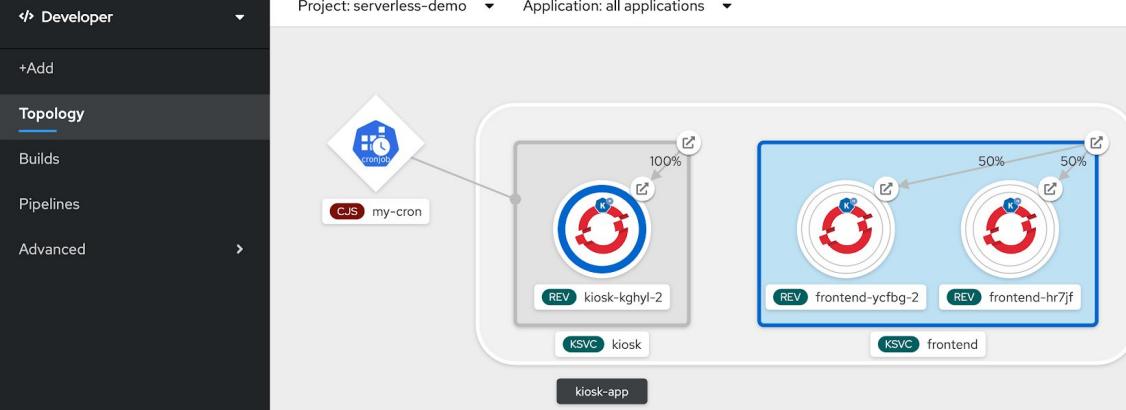
Time in capacity planning
IT cost of idle resources



More applications

Direct line between IT costs & business revenue





KSVC frontend

Overview Resources

Revisions

REV frontend-hr7jf 50%

D frontend-hr7jf-deployment 0

REV frontend-ycfbg-2 50%

Actions Set Traffic Distribution

Serverless User Experience



OpenShift Serverless Operator

1.5.0 provided by Red Hat, Inc.

Install

OPERATOR VERSION

1.5.0

PROVIDER TYPE

Red Hat

The Red Hat Serverless Operator provides a collection of APIs to enable containers, microservices and functions to run "serverless".

It includes a number of built-in event sources and integrates with other OpenShift services.

```
markito@anakin ~ % kn service create store-app --image=markito/store-app:v1
markito@anakin ~ % kn service create store-app --image=markito/store-app:v1
```

OpenShift Builds

Build lean images from application source code and binary using
Kubernetes tools on OpenShift



Build slim runtime
images without the
build dependencies



Use Kubernetes build
tools (e.g. buildah, S2I,
CNB¹, Kaniko, etc)



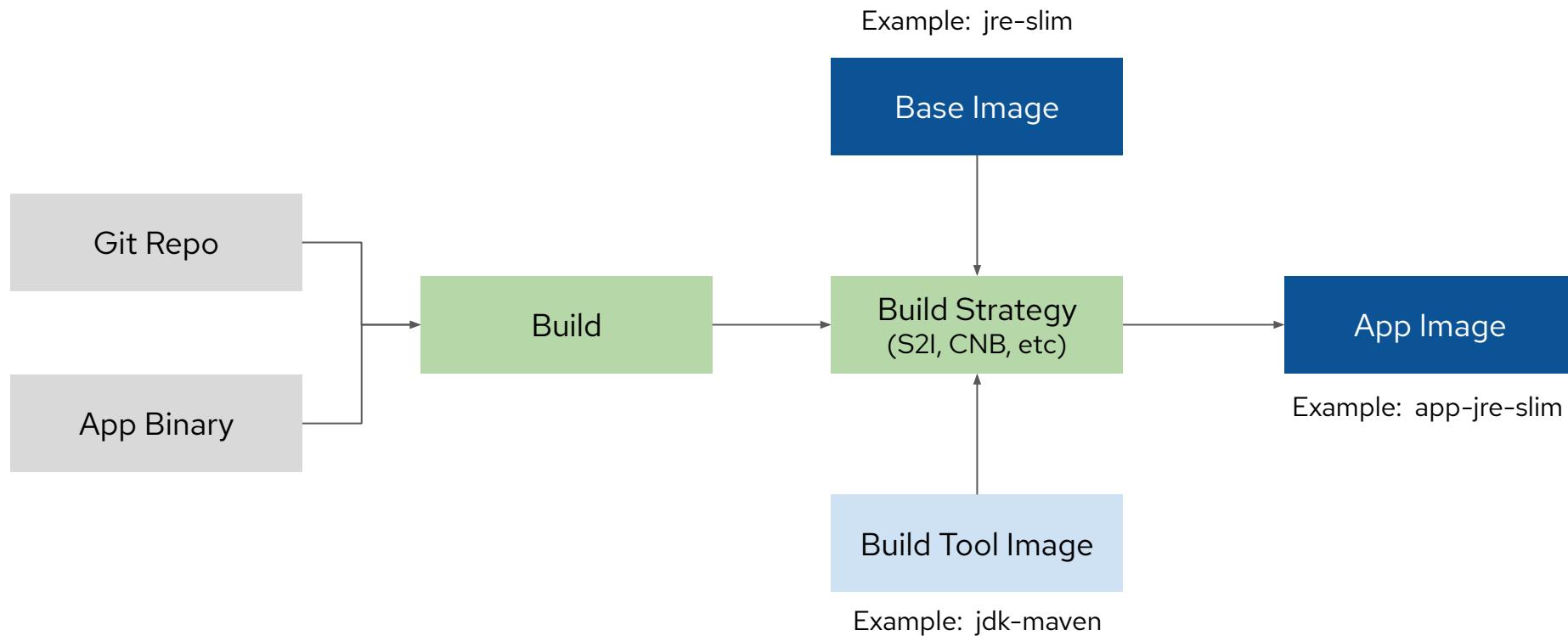
Extensible and
customizable with
your own build tools



Portable builds to
any Kubernetes
platform

¹Cloud-Native Buildpacks

OpenShift Builds



OpenShift Builds

Cloud-Native Buildpacks

```
kind: Build
metadata:
  name: myapp-cnb-build
spec:
  source:
    url: https://github.com/myorg/myapp
  strategy:
    name: buildpacks-v3
  builder:
    image: heroku/buildpacks:18
  output:
    image: quay.io/myorg/myapp:v1
```

Source-to-Image (S2I)

```
kind: Build
metadata:
  name: myapp-s2i-build
spec:
  source:
    url: https://github.com/myorg/myapp
  strategy:
    name: source-to-image
  builder:
    image: registry.redhat.io/openjdk/openjdk-11-rhel8
  output:
    base: docker.io/openjdk:11-jre-slim
    image: quay.io/myorg/myapp:v1
```

OpenShift Pipelines

Cloud-Native CI/CD with Tekton on OpenShift



Kubernetes-native
declarative
Pipelines with
Tekton



Serverless CI/CD
with no single server
to share and
maintain



Run pipelines in
isolated containers with
all required
dependencies



Standard and
portable to any
Kubernetes
platform



Web, CLI, and
Visual Studio
Code and IDE
plugins

Red Hat OpenShift Container Platform

Developer

+Add

Topology

Monitoring

Builds

Pipelines

More

Project: demo

Pipeline Runs > Pipeline Run Details

PLR petclinic-deployment-pipeline-vivla6 Running

Dev Preview

Actions

Details YAML Logs

Pipeline Run Details

Name: petclinic-deployment-pipeline-vivla6

Namespace: NS demo

Red Hat OpenShift Container Platform

Developer

+Add

Topology

Monitoring

Builds

Pipelines

More

Project: demo

Pipeline Runs > Pipeline Run Details

PLR petclinic-deployment-pipeline-vivla6 Running

Details YAML Logs

unit-tests

slow-unit-tests

build-image

build-image

STEP-GENERATE

```
{"level": "info", "ts": 1585670155.8098009, "logger": "fallback-logger", "caller": "logging/config.g"} Application dockerfile generated in /gen-source/Dockerfile.gen
```

STEP-BUILD

```
{"level": "info", "ts": 1585670174.8911083, "logger": "fallback-logger", "caller": "logging/config.g"} STEP 1: FROM registry.access.redhat.com/redhat-openjdk-18/openjdk18-openshift
```

Getting image source signatures

Copying blob sha256:6c858ac87d44df4b64d7c273886fc5aed55a28422df33dc641884ffa419db218

Copying blob sha256:51e9f237b750efcdad2d5755785cd8bd089d51585ae35d368e4fb29a11b1994

Copying blob sha256:41bc10db03d6e2f00d82a33d2fdca62e3a50cd17b8cfdf59cbe650e278af

Copying config sha256:fc8de886c1471711ed33a0c1ab46143e433e02870d342b00683be0175a92a64f

Writing manifest to image destination

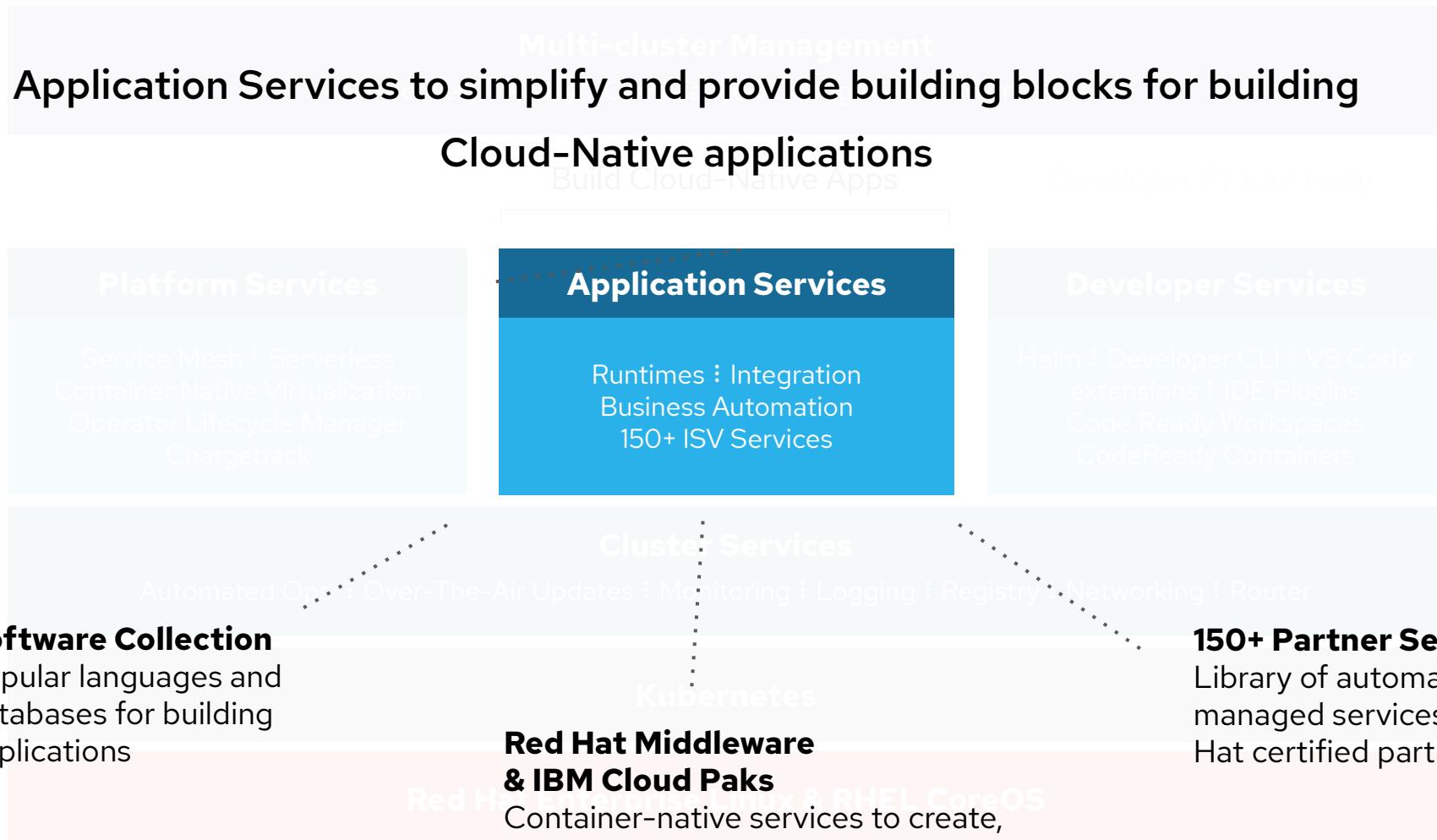
Storing signatures

pipeline.yaml — pipelines-examples

```
private > tmp > ! pipeline.yaml > {} spec > [ ] tasks > {} 2 > [ ] params > {} 0 > [ ] value
resources:
  - name: source
    resource: app-git
  outputs:
    - name: image
      resource: app-image
  - name: slow-unit-tests
taskRef:
  name: mvn
runAfter:
  - unit-tests
resources:
  inputs:
    - name: source
      resource: app-git
  params:
    - name: GOALS
      value:
        - verify
  - name: deploy
taskRef:
  name: oc-redeploy
runAfter:
  - build-image
  - slow-unit-tests
params:
  - name: DEPLOYMENT
    value: spring-petclinic
```

Application Services

OpenShift Container Platform



Languages and Runtimes

A collection of programming languages, runtimes and databases for
building containerized cloud-native applications

 Java

 Node.js

 Python

 Go

 Ruby

 Ruby on Rails

 PHP

 Perl

 .NET Core

 MySQL

 PostgreSQL

 MariaDB

 MongoDB

 Redis

 Apache HTTP Server

 JBoss Web Server (Tomcat)

 Nginx

 Red Hat SSO

Red Hat Middleware

Create , integrate, and automate applications with cloud-native runtimes, integration, and business process automation frameworks



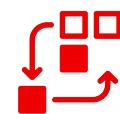
Red Hat Runtimes

- Red Hat JBoss Enterprise Application Platform
- Cloud-Native Runtimes
(Quarkus, Spring Boot, Vert.x)
- Red Hat AMQ (Broker)
- Red Hat Data Grid
- Red Hat Application Migration Toolkit
- Open Liberty



Red Hat Integration

- Red Hat Runtimes
- Red Hat Fuse
- Red Hat 3scale API Management
- Red Hat AMQ
(Broker, Interconnect, Streaming)



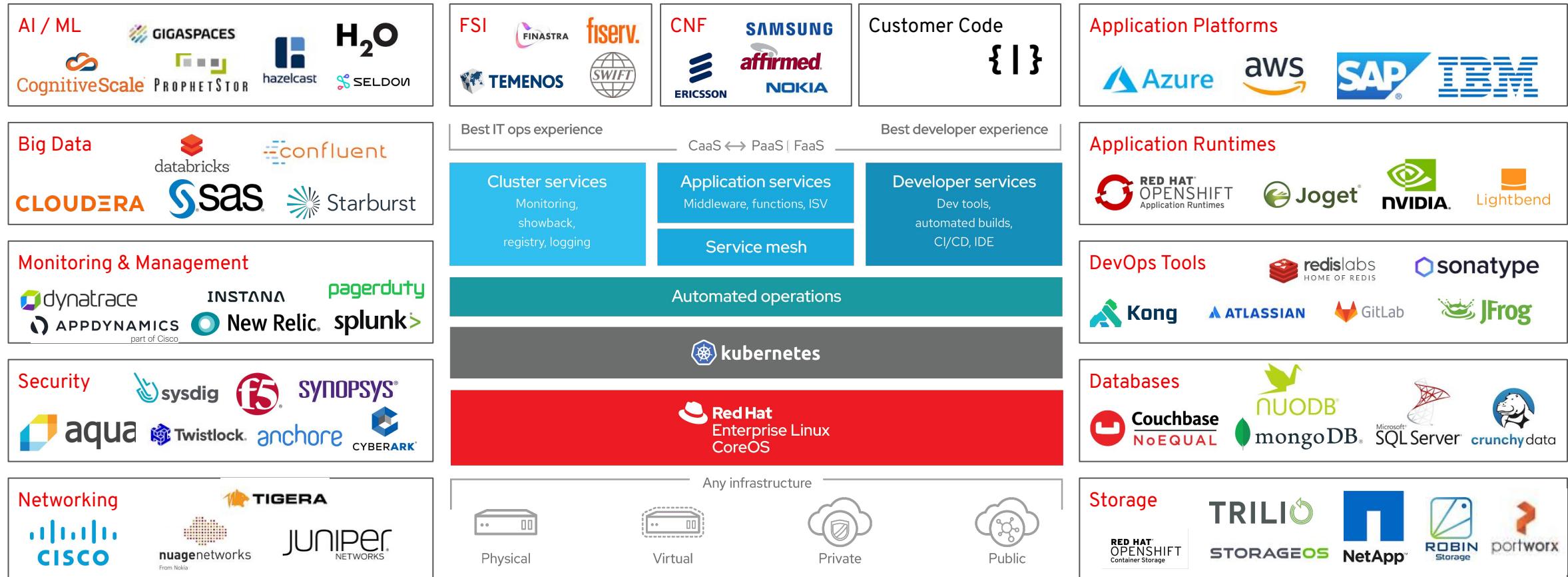
Red Hat Process Automation

- Red Hat Runtimes
- Red Hat Decision Manager
- Red Hat Process Automation Manager



Cloud Paks for Apps, Data, Integration, Automation, Multi-Cloud and Security

150+ Partner Services



Application Binding Operator

```
apiVersion: apps.openshift.io/v1alpha1
kind: ServiceBindingRequest
metadata:
  name: binding-request
  namespace: service-binding-demo
spec:
  applicationSelector:
    resourceRef: nodejs-rest-http-crud
    group: apps
    version: v1
    resource: deployments
  backingServiceSelector:
    group: postgresql.baiju.dev
    version: v1alpha1
    kind: Database
    resourceRef: db-demo
```



Connect two Kubernetes resources together, similar to the binding model in OpenServiceBroker.

Label selectors make this very powerful and dynamic as resources come and go.

Installed from OperatorHub

Works for all objects

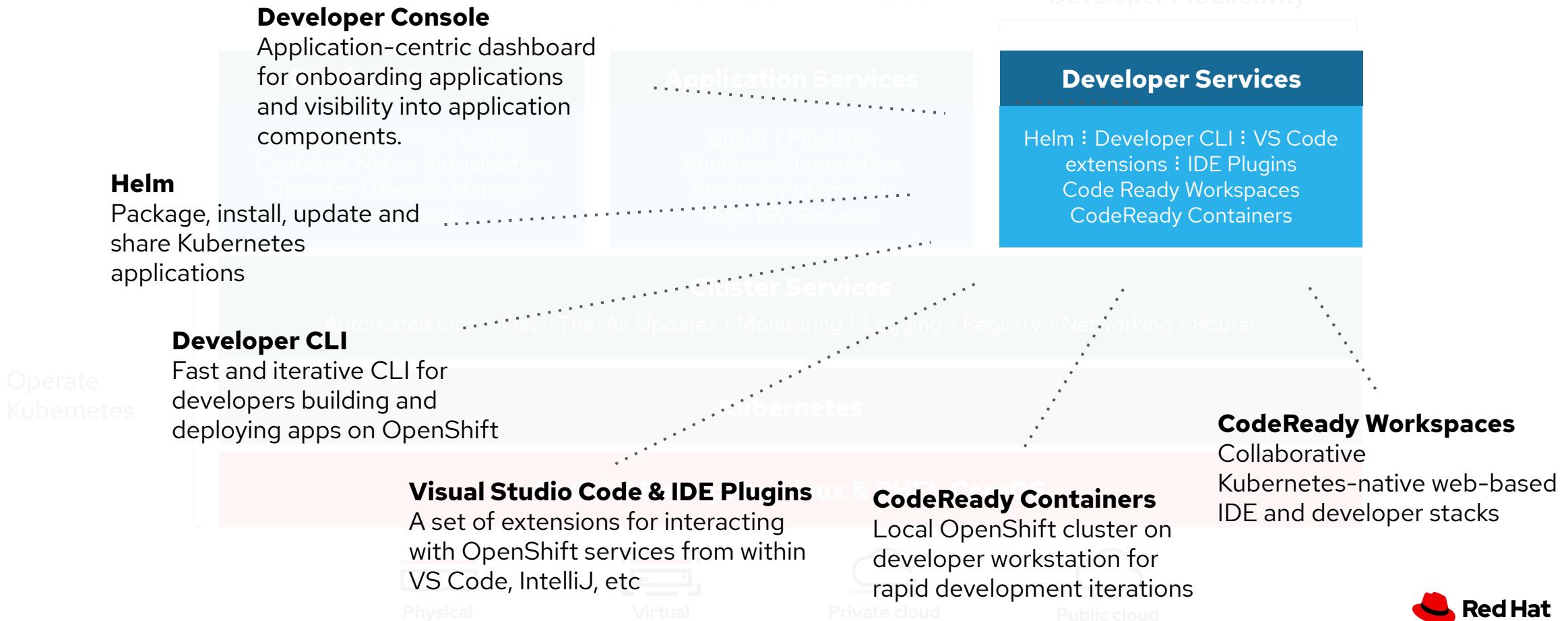
- Operator↔Operator
- Operator↔Deployments
- Pod↔Pod

Developer Services

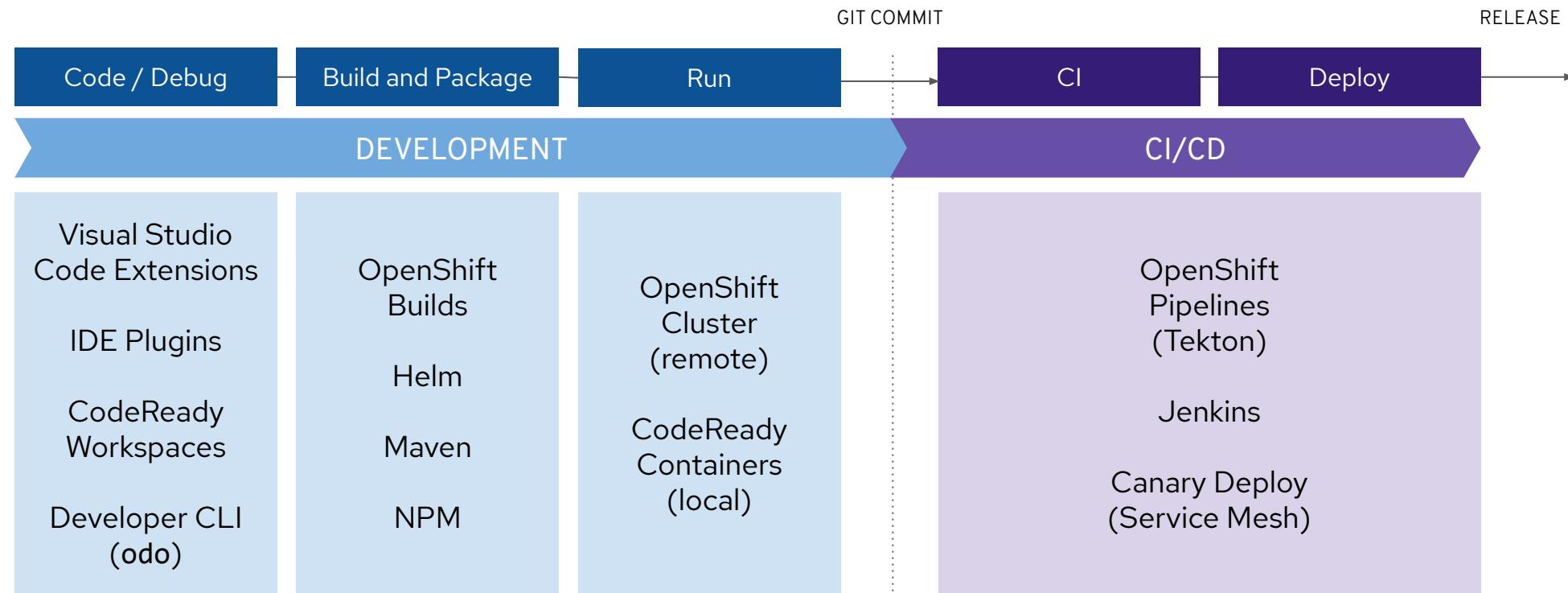
OpenShift Container Platform

Multi-cluster Management

Increase developer productivity on Kubernetes with developer-focused tools and services



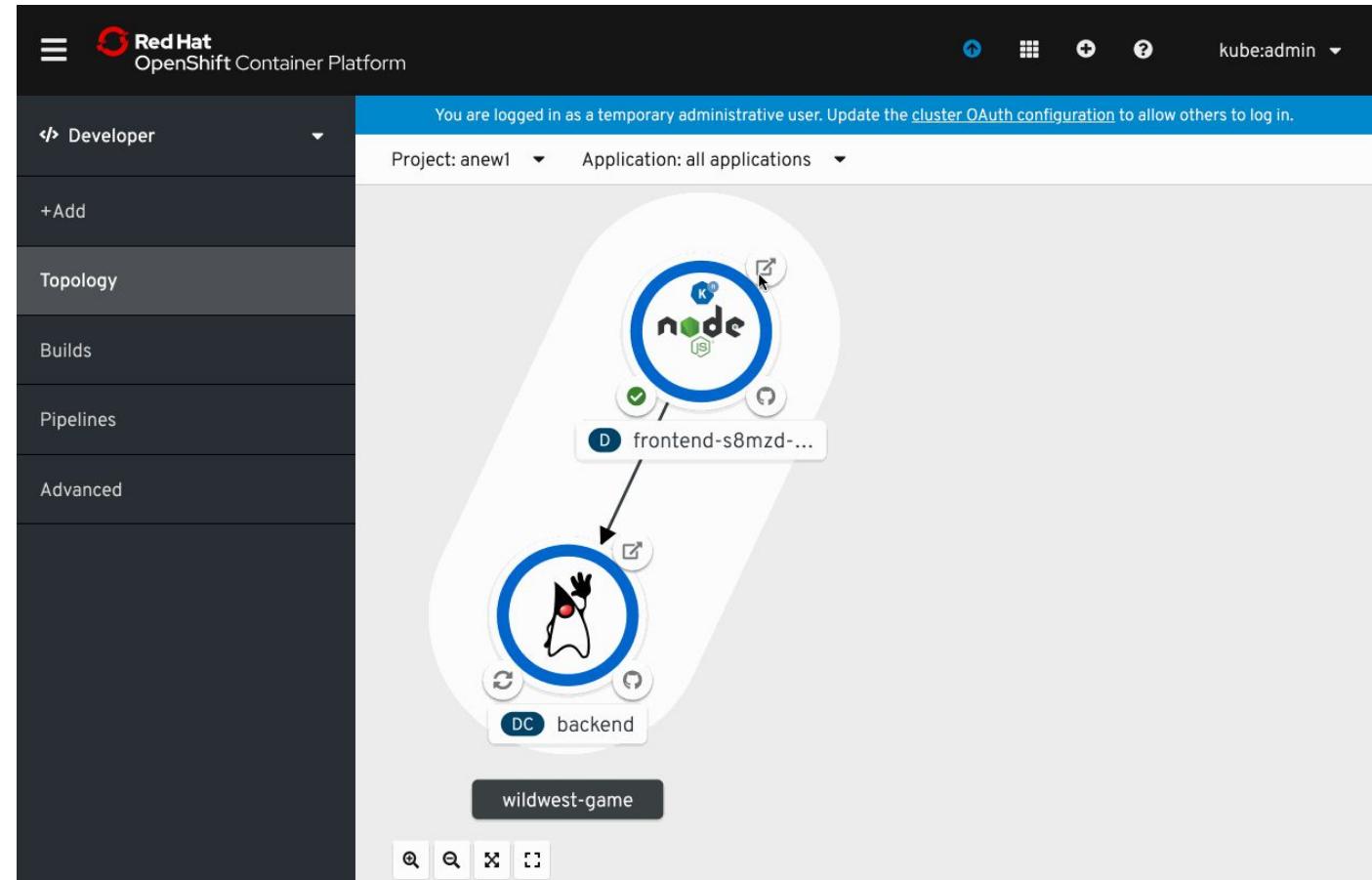
Development services and tools simplify onboarding for new users, while allowing expert growth



Developer Console

An application-focused perspective in the OpenShift Console that will sit beside the Admin console and focus on developer and application use cases.

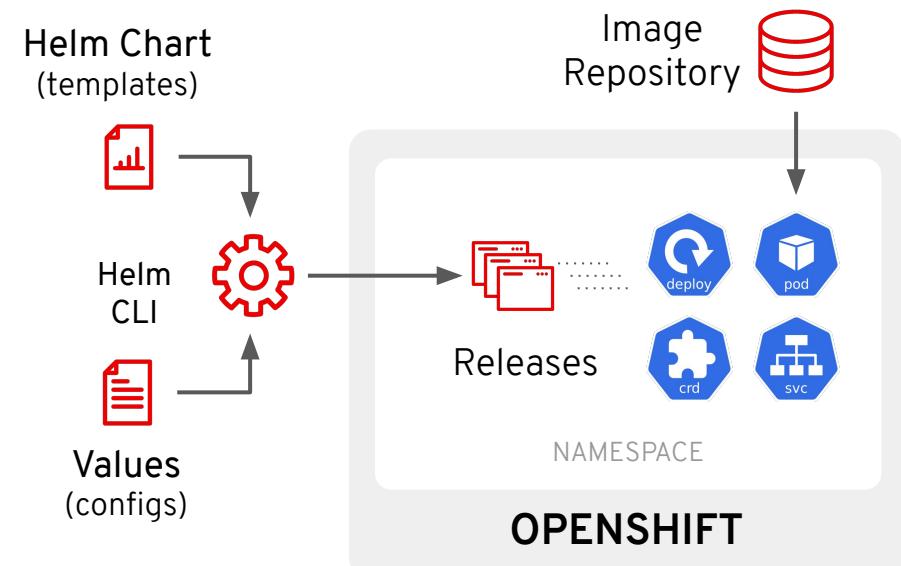
- Import applications
- Insight into components
- Application metrics
- Image builds
- CI/CD Pipelines



Helm 3 on OpenShift

Package, install, update and share Kubernetes applications

- Developer-focused application packaging
- Install, update and rollback Helm Charts
- Share charts via charts repositories
- Compatible with Kubernetes RBAC
- Helm charts in OpenShift Console
- Build Helm Operators for automated chart instantiation through the developer catalog



```
$ helm install myapp myorg/myapp
```

Helm 3 on OpenShift

The image displays three screenshots of the Red Hat OpenShift Container Platform interface, illustrating the integration of Helm 3.

- Screenshot 1: Developer Catalog**
Shows the 'Developer Catalog' page for the 'demo' project. It lists various shared apps, services, or source-to-image builders. A search bar and filter options are present. A modal window for 'Helm Charts' is open, showing a chart for 'Nodejs Ex K v0.1.0'.
- Screenshot 2: Helm Release Details**
Shows the 'Helm Release Details' page for the 'mynodejs-app-1' release in the 'demo' namespace. It lists resources such as BuildConfig, DeploymentConfig, ImageStream, Route, and Service, all created 3 minutes ago.
- Screenshot 3: Application Overview**
Shows the application overview for 'mynodejs-app-1' in the 'demo' namespace. It includes a summary card for the 'nodejs-example' deployment and a detailed view of the 'nodejs-example' deployment resource.

CodeReady Containers

Local **OpenShift 4** development cluster
on your workstation for quick
container-based application
development iterations.

Replaces the 3.x experiences around:

- Minishift
- CDK
- oc cluster up

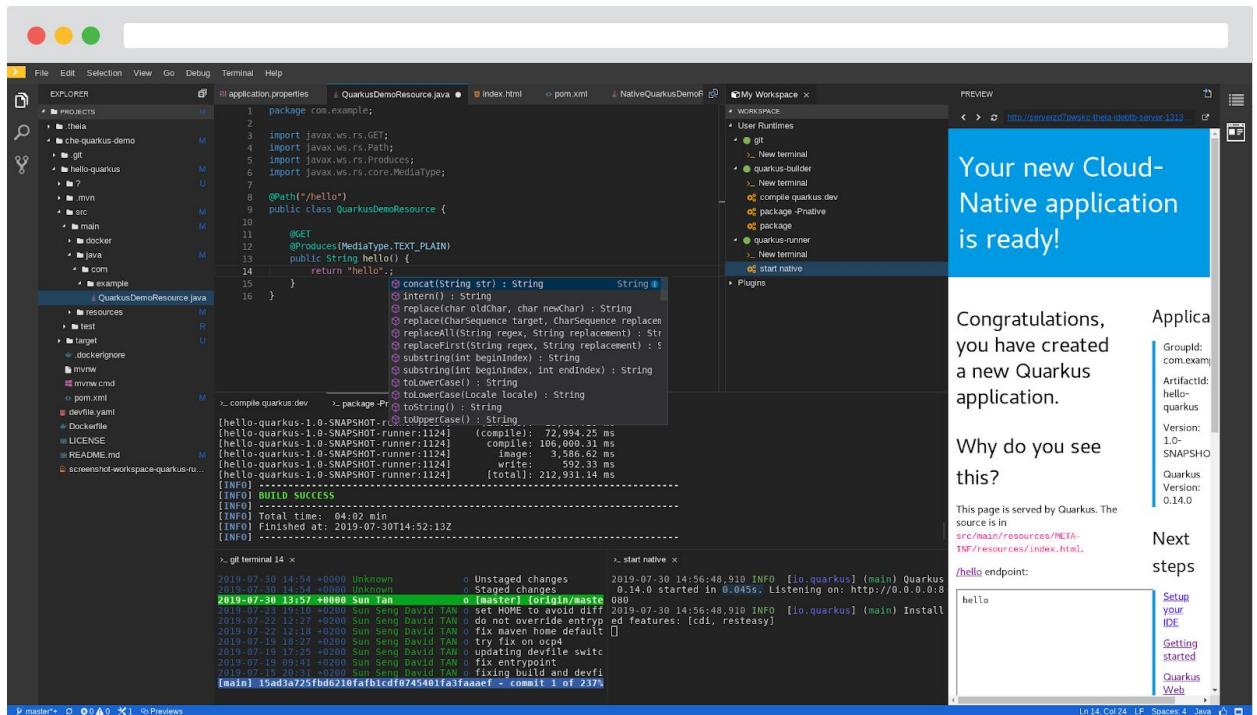
```
$ crc setup
Prepare your machine for running OpenShift

$ crc start
Start with the Hyperkit 4.3 bundle
```

```
$ crc status
Get the status of the cluster
```

CodeReady Workspaces

- Web-based developer workspaces
- Fully containerized and Kubernetes-native
- Familiar experience similar to Visual Studio Code with plugin extensions compatibility
- Reproducible and portable developer environments defined in git
- Prometheus and Grafana dashboards
- Based on Eclipse Che 7+



Roadmap (Q3/Q4 CY 2020)

Manage Workloads

Platform Services

Service Mesh : Serverless
Builds : CI/CD Pipelines
Full Stack Logging
Chargeback

- Operators: simplify OLM interactions
- Operators: ship catalogs in container images
- Operators: install older versions instead of latest
- Service Mesh: disconnected install
- Serverless: Eventing GA
- Serverless: Functions Tech Preview
- Builds Tech Preview
- Pipelines GA
- Jenkins Operator Tech Preview
- Marketplace: expansion of offerings
- Marketplace: run a disconnected Marketplace
- Logging: GA of log forwarding

Build Cloud-Native Apps

Application Services

Databases : Languages
Runtimes : Integration
Business Automation
150+ ISV Services

- Red Hat Runtimes:
 - Event Sources for SSO & Data Grid
 - Runtimes for IBM Power
 - Quarkus Native support GA
- Red Hat Integration & Serverless
 - Camel-K Event source GA
 - 3Scale integration
 - Apache Kafka (Strimzi)
- Operators: expanding partner support for:
 - FIPS cryptography
 - Disconnected
 - Egress proxies
- Operators: improved tools for curated catalogs
- Operators: enhanced upgrade testing

Developer Productivity

Developer Services

Helm : Developer CLI : VS Code extensions : IDE Plugins
Code Ready Workspaces
CodeReady Containers

- Helm 3 GA
- Helm in Dev Console
- VS Code Plugins
 - OpenShift,
 - Knative
 - Tekton
- odo AppStak/DevfileV2
- Pipeline Triggers in Dev Console
- GitOps View
- CRC system tray for Mac/Win
- CRW startup improvements
- CRW shared language servers



Thank you



linkedin.com/company/Red-Hat



facebook.com/RedHatinc



youtube.com/user/RedHatVideos



twitter.com/RedHat