



# Introduction to OpenShift 4

**Götz Rieger**

Senior Solution Architect

# Creating value depends on your ability to deliver applications faster

Cloud-native applications



AI & machine learning



Analytics



Internet of Things

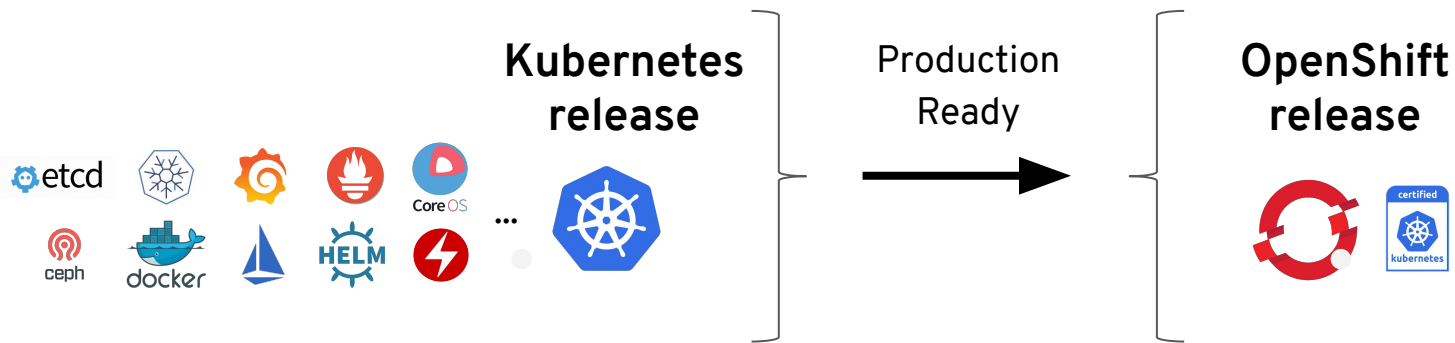


Innovation culture



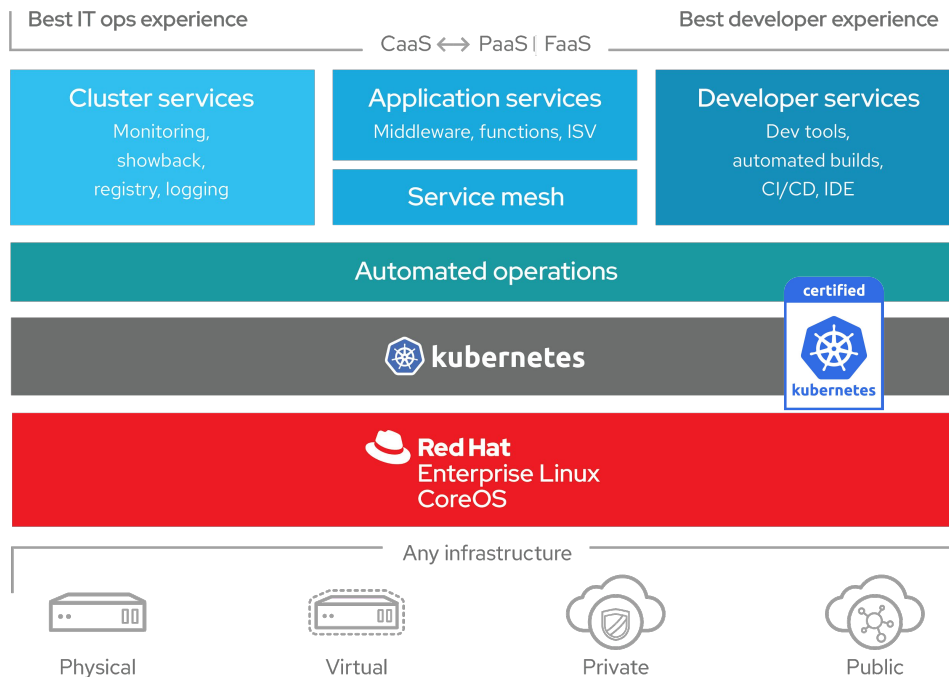
**Containers, Kubernetes, and hybrid cloud** are key ingredients.  
OpenShift is the best platform to deliver container-based applications.

# OpenShift is trusted enterprise Kubernetes

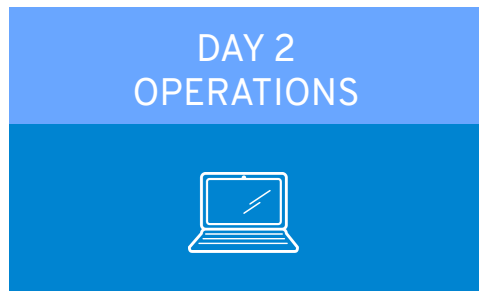


- Hundreds of defect and performance fixes
- 200+ validated integrations
- Certified container ecosystem
- Red Hat is a leading Kubernetes contributor since day 1

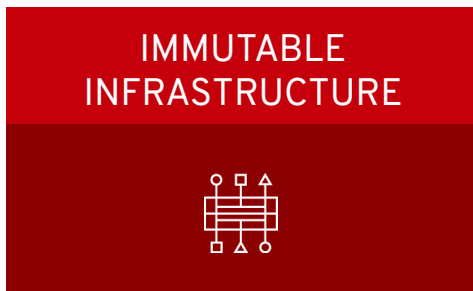
# OpenShift 4 - Enterprise Kubernetes



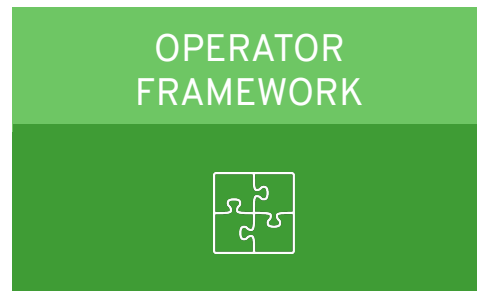
# OpenShift 4.1 Themes



**Installer + bootstrapping**  
**Autoscale out of the box**  
**MachineSet node pools**



**Red Hat Enterprise Linux CoreOS**  
**Discourage SSH/node mutation**  
**Ignition for Machine config**



**SDK & testing tools**  
**OperatorHub for discovery**  
**OLM delivers upper stack services**

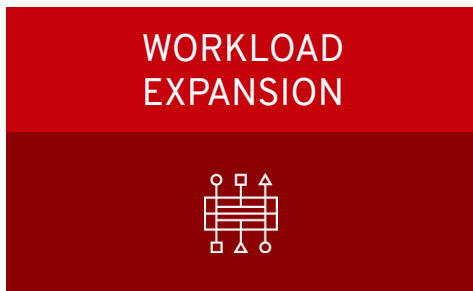
# OpenShift 4.2 Themes



**Disconnected, air-gapped  
installation**

**Full-stack Automation for  
Azure, GCP, OpenStack**

**Migration tooling**



**Operators for Red Hat Integration,  
Business Automation, Runtimes**

**Special resources operator  
For GPU**



**Developer console with  
application topology**

**CodeReady Containers**

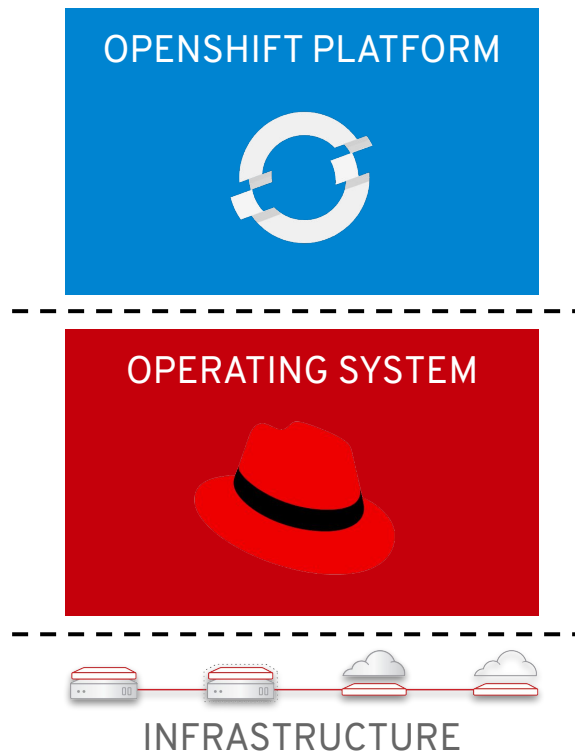
**Service Mesh fully supported**

# Installation & Update



# Full Stack Automation

## OPENSIFT 3



## OPENSIFT 4





# Installation Experiences

## OPENSIFT CONTAINER PLATFORM

### Full Stack Automation

Simplified opinionated “Best Practices” for cluster provisioning

Fully automated installation and updates including host container OS.



### Pre-existing Infrastructure

Customer managed resources & infrastructure provisioning

Plug into existing DNS and security boundaries



## HOSTED OPENSIFT

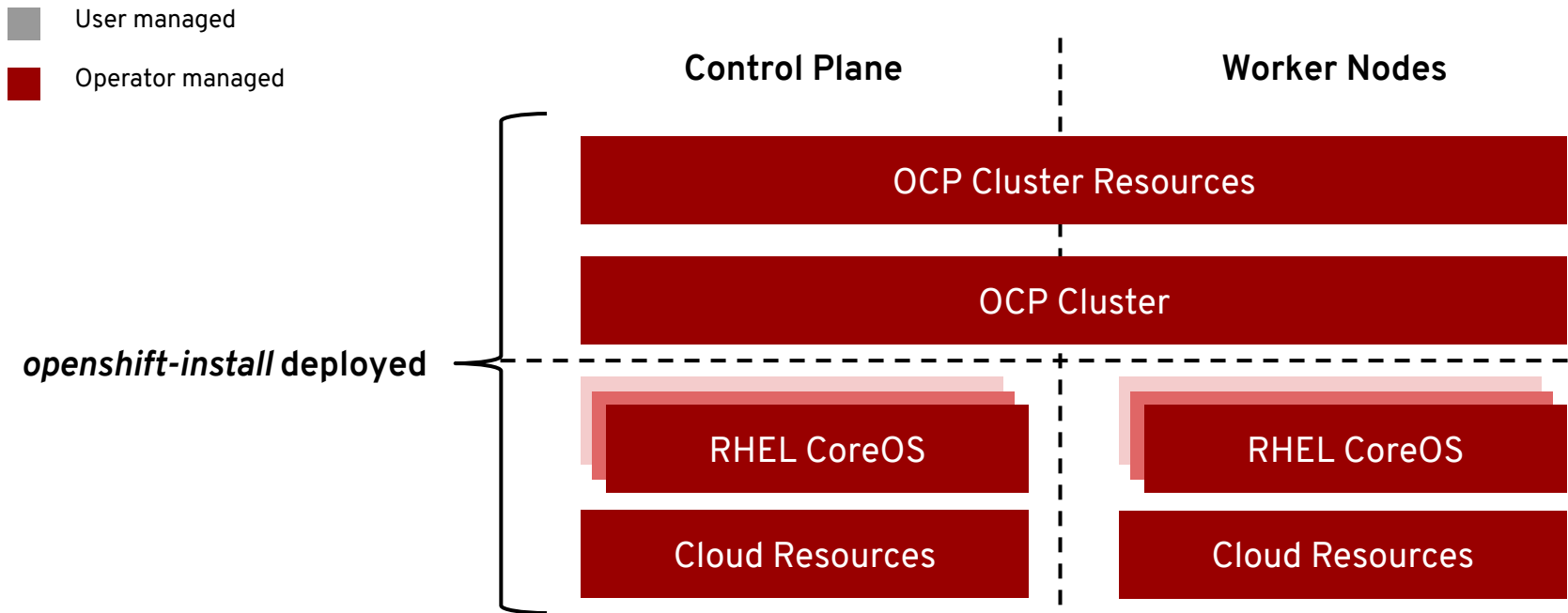
### Azure Red Hat OpenShift

Deploy directly from the Azure console. Jointly managed by Red Hat and Microsoft Azure engineers.

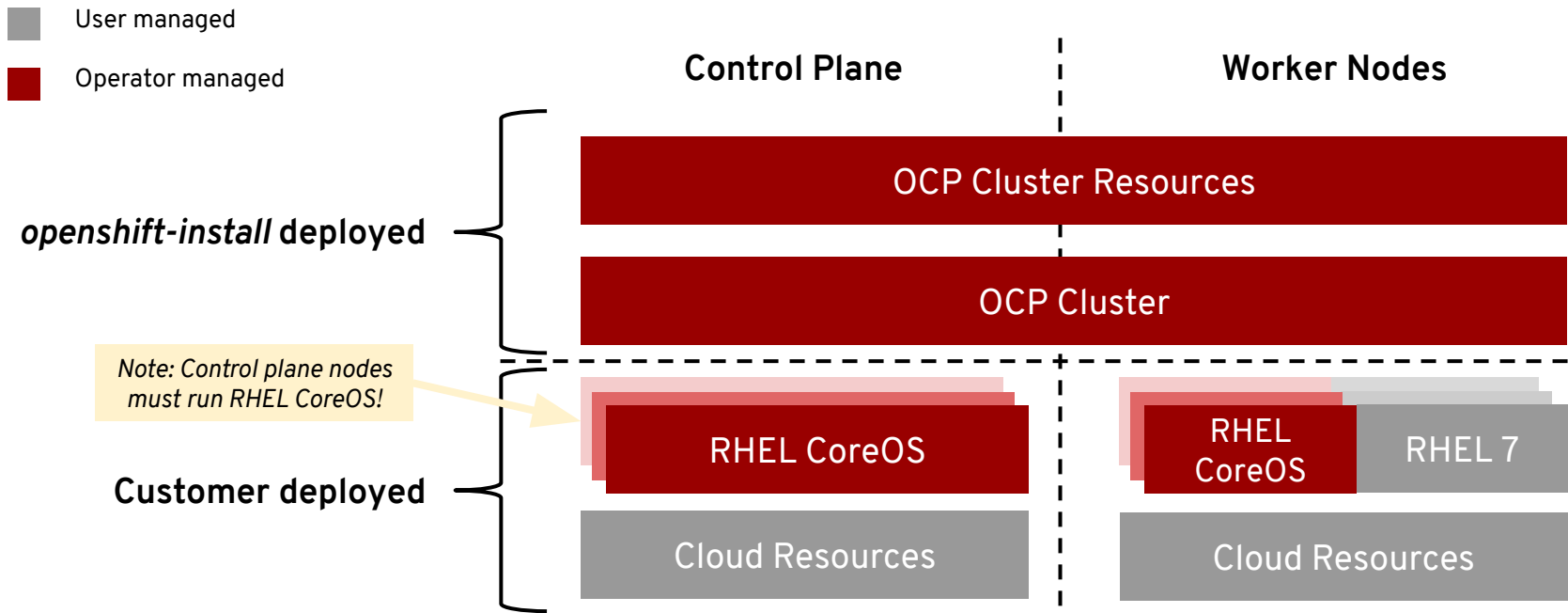
### OpenShift Dedicated

Get a powerful cluster, fully Managed by Red Hat engineers and support.

# Full Stack Automated Deployments



# Deploying to Pre-existing Infrastructure



# Supported Providers 4.1

Full Stack Automated



Pre-existing Infrastructure



**Bare Metal**



*Requires Internet connectivity*

# Supported Providers 4.2

## Full Stack Automation (IPI)



## Pre-existing Infrastructure (UPI)

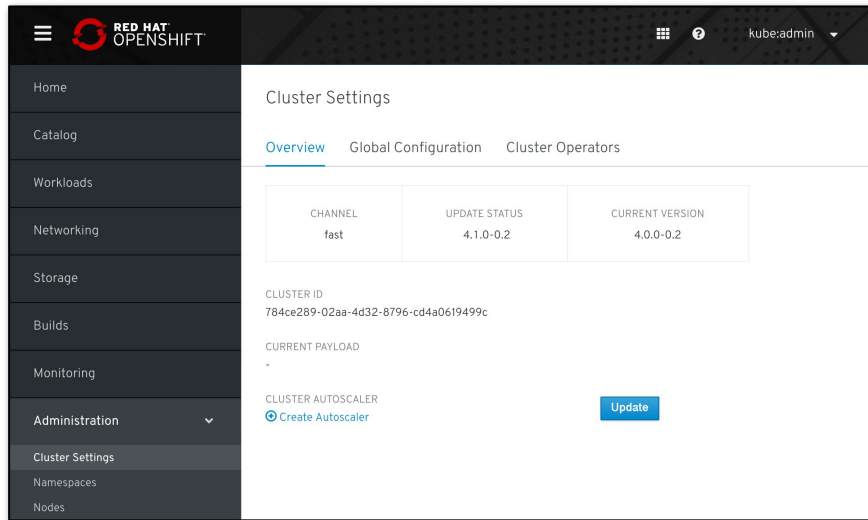


**Bare Metal**

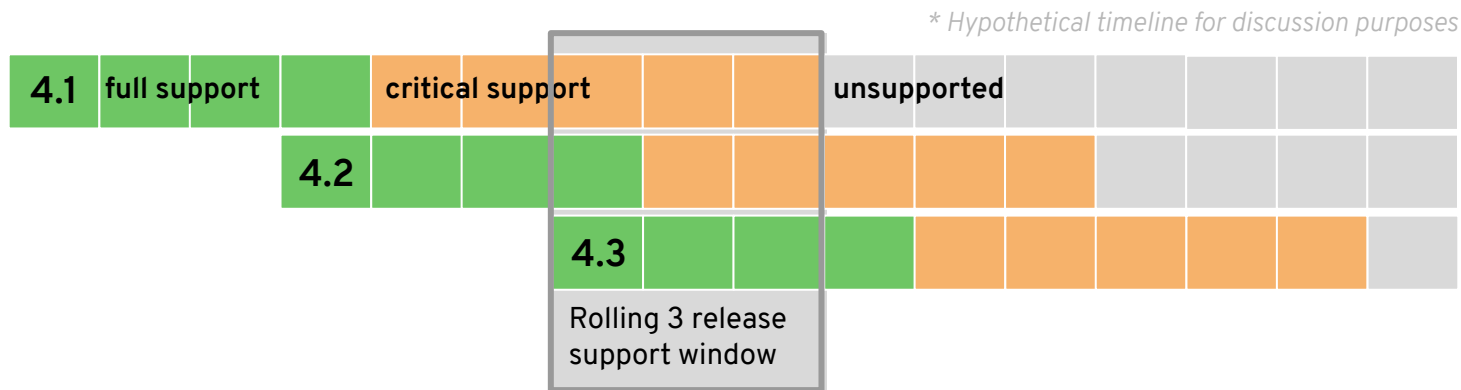
*\* Support for full stack automated installs to pre-existing VPC & subnets and deploying as private/internal clusters is planned for 4.3.*

# Over-the-Air Updates

- OpenShift retrieves list of available updates
- Admin selects the target version
- OpenShift is updated over the air
- Auto-update support
- RHCOS only



# OpenShift 4 Lifecycle



## New model

Release based, not date based. Rolling three release window for support.

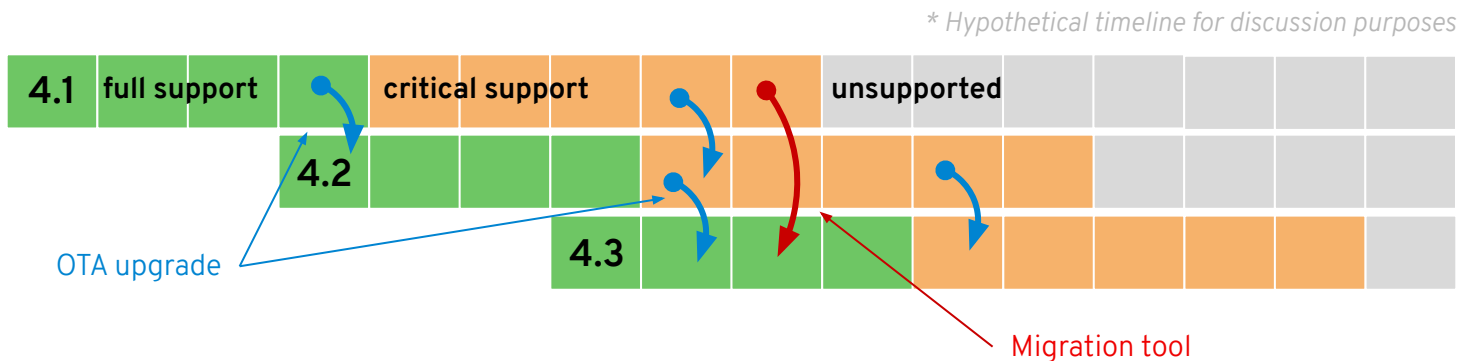
The overall 4 series will be supported for at least three years

- Minimum two years full support (likely more)
- One year maintenance past the end of full support

## EUS release planned

Supported for 14 months of critical bug and critical security fixes instead of the normal 5 months. If you stay on the EUS for its entire life, you must use the application migration tooling to move to a new cluster

# OpenShift 4 Upgrades



## OTA Upgrades

Works between two minor releases in a serial manner.

## Happy path = migrate through each version

On a regular cadence, migrate to the next supported version.

## Optional path = migration tooling

If you fall more than two releases behind, you must use the application migration tooling to move to a new cluster.

## Current minor release

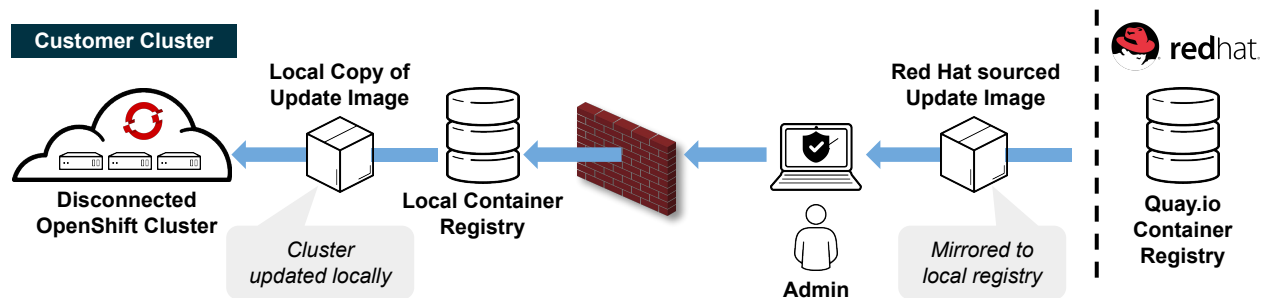
Full support for all bugs and security issues  
1 month full support overlap with next release to aid migrations

## Previous minor release

Fixes for critical bugs and security issues for 5 months



# Disconnected “Air-Gapped” Installation & Upgrading



- Support for installing and updating OpenShift clusters in disconnected environments
- Requires local Docker 2.2 compliant container registry to host OpenShift content
- Designed to work with the UPI deployment method
  - *Note: Will not work with Installer provisioned infrastructure deployments*

# Disconnected “Air-gapped” OLM

## 1. Turn off default operator sources that comes with OpenShift:

```
# Update the spec of operatorhub.config.openshift.io/v1:
apiVersion: config.openshift.io/v1
kind: OperatorHub
spec:
  disableAllDefaultSources: true
```

## 2. Retrieve package lists for the default operator sources:

```
$ curl https://quay.io/cnr/api/v1/packages?namespace=redhat-operators > packages.txt
$ curl https://quay.io/cnr/api/v1/packages?namespace=community-operators >> packages.txt
$ curl https://quay.io/cnr/api/v1/packages?namespace=certified-operators >> packages.txt
```

## 3. Pull operator content; for example:

```
$ curl \
https://quay.io/cnr/api/v1/packages/redhat-operators/openshifttemplateservicebroker/3.0.0
# From that JSON take the digest and use to pull the gzipped:
$ curl -XGET \
https://quay.io/cnr/api/v1/packages/redhat-operators/openshifttemplateservicebroker/blobs/
sha256/9bddc82c0fee05b243d01c6e2f0110c413eb9165cd7bb8adc25f1e88997854c9 -o
openshifttemplateservicebroker.tar.gz
$ tar fczP openshifttemplateservicebroker.tar.gz manifests/
```

In the file called ‘bundle.yaml’, you need to break data.clusterServiceVersion (each file in the list), data.Package, and data.customResourceDefinition (each file in the list) into their own files:

```
Manifests
├── openshifttemplateservicebroker
│   ├── clusterserviceversion.yaml
│   ├── customresourcedefinition.yaml
│   └── package.yaml
├── etcd-XXXX
└── <CSV's and CRDs and a package file>
```

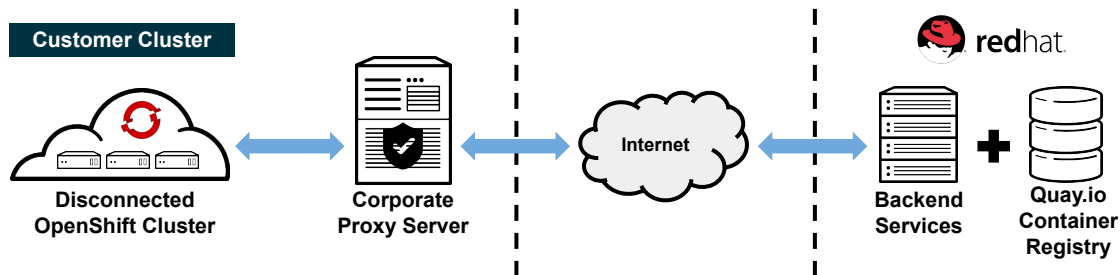
## 4. Create Operator Catalog Image using this dockerfile:

```
FROM registry.redhat.io/openshift4/ose-operator-registry:4.2.0 as builder
COPY manifests manifests
RUN ./bin/initializer -o ./bundles.db
FROM scratch
COPY --from=builder /build/bundles.db /bundles.db
COPY --from=builder /build/bin/registry-server /registry-server
COPY --from=builder /bin/grpc_health_probe /bin/grpc_health_probe
EXPOSE 50051
ENTRYPOINT ["/registry-server"]
CMD ["--database", "bundles.db"]
```

Copy the image to the internal registry and create a catalog source pointing to the newly created image:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: example-manifests
  namespace: default
spec:
  sourceType: grpc
  image: <path to the the image>/example-registry:latest
```

# Cluster-wide Egress Proxy



- Support for installing and updating OpenShift clusters through a corporate proxy server
- Leverages new proxy controller within the cluster-network-operator

# Node Feature Discovery Operator

## **What is Node Feature Discovery (NFD)**

Detects hardware features available on each node and advertises those features using node labels.

## **List of features managed by NFD**

CPU Features: cpuid, hardware\_multithreading, power, pstate

Kernel Features: config, selinux\_enabled, version, os\_version

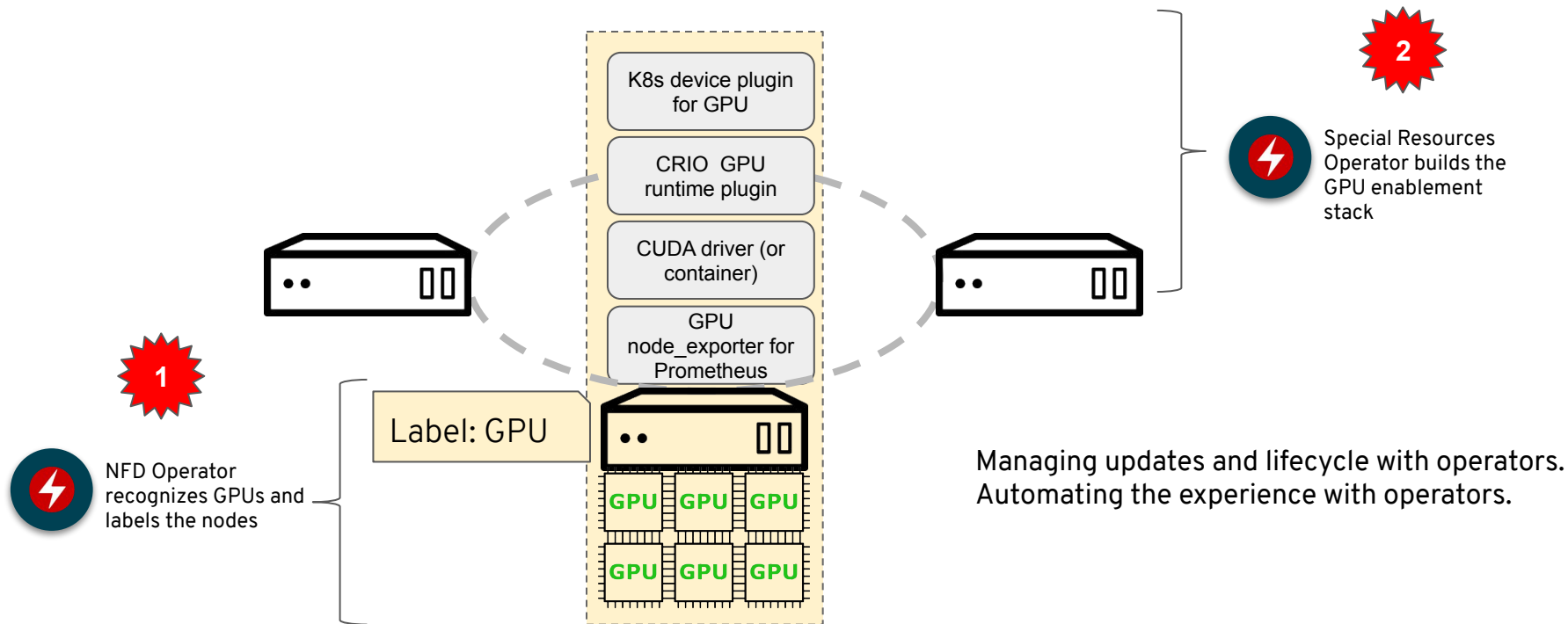
Others: **Presence of NVMe, NUMA, SR-IOV and GPUs**

## **Node Feature Discovery (NFD) Operator**

Manages the install and lifecycle of the NFD DaemonSet

NFD operator can be accessed via the embedded OperatorHub

# Enablement of GPUs in an OpenShift Cluster



**A cloud-like experience,  
everywhere**



# A consistent container application platform

FROM YOUR DATACENTER TO THE CLOUD



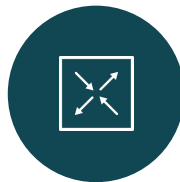
Automated  
operations



Multi-tenant



Secure by  
default



Network  
traffic control



Over-the-air  
updates



Monitoring  
& chargeback



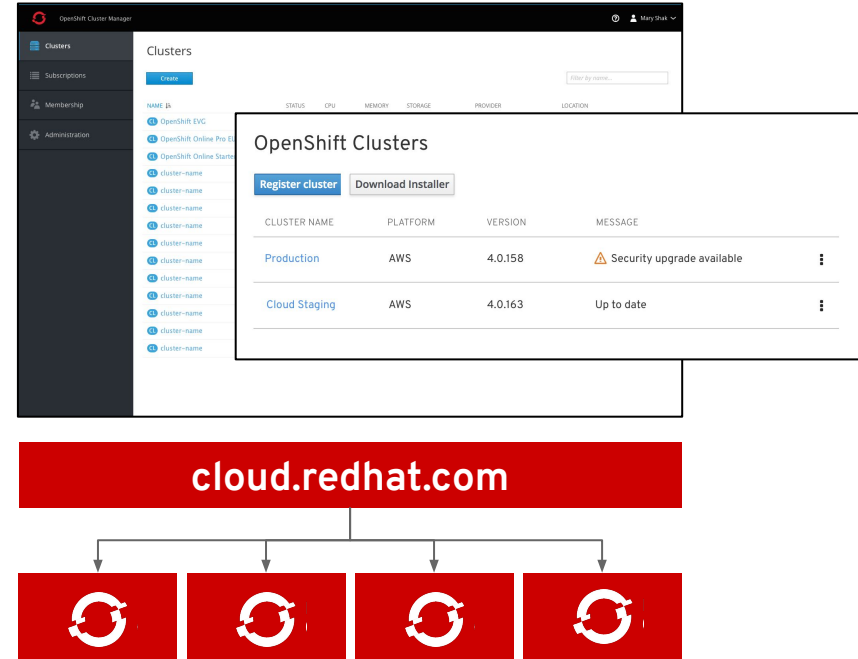
Pluggable  
architecture



Bare metal, VMware vSphere, Red Hat Virtualization, Red Hat OpenStack Platform,  
Amazon Web Services, Microsoft Azure, Google

# Delivering Kubernetes Everywhere

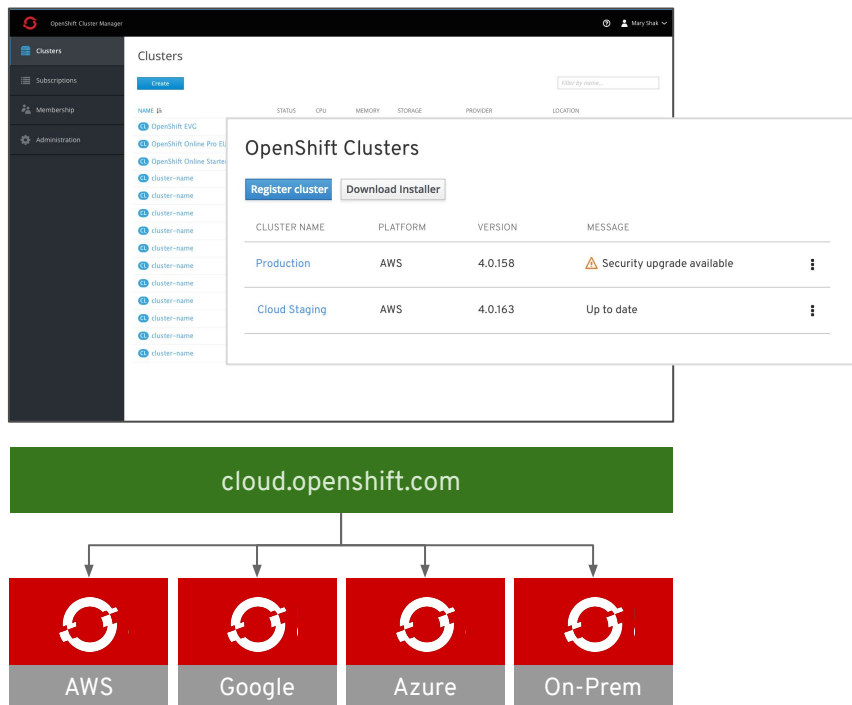
- Manage multiple OpenShift clusters, across multiple cloud and on-premises environments
- Install and update OpenShift across all your cloud environments
- Centrally manage policy and deployments





# Unified Hybrid Cloud

- Cloud-based multicloud management
  - New clusters on AWS, Azure, Google, vSphere, OpenStack, and bare metal
  - Register existing clusters
  - Including OpenShift Dedicated
- Management operations
  - Install new clusters
  - View all registered clusters
  - Update clusters



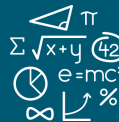
# Console & (Day 2) Operations



# Kubernetes-Native Day 2 Management



Flexible app  
architectures



No reinvention  
of core concepts



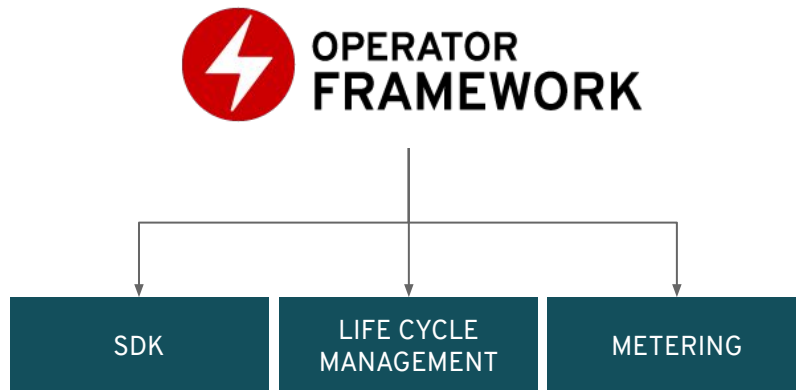
Uniform deploy  
and debug



Truly hybrid

# Operator Framework

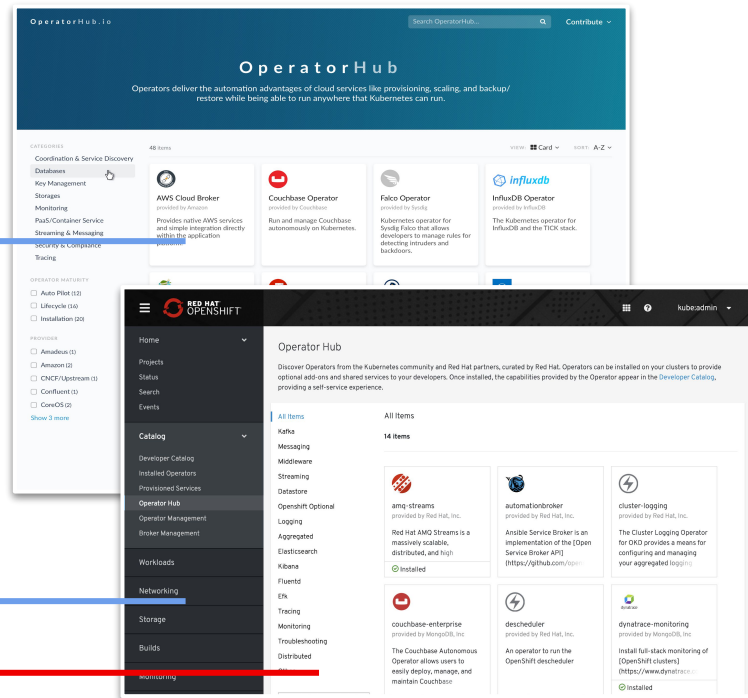
Operators codify operational knowledge and workflows to automate life cycle management of containerized applications with Kubernetes



# OperatorHub and Certified Operators

- OperatorHub.io launched by Red Hat, AWS, Microsoft and Google
- OpenShift Operator Certification
- OperatorHub integrated into OpenShift 4

COMMUNITY OPERATORS  
OPENSHIFT CERTIFIED OPERATORS



# Build Operators for your apps

Helm Chart



Helm SDK



Build operators from  
Helm chart, without any  
coding

Ansible Playbooks  
APBs



Ansible SDK



Build operators from  
Ansible playbooks and  
APBs

Go SDK



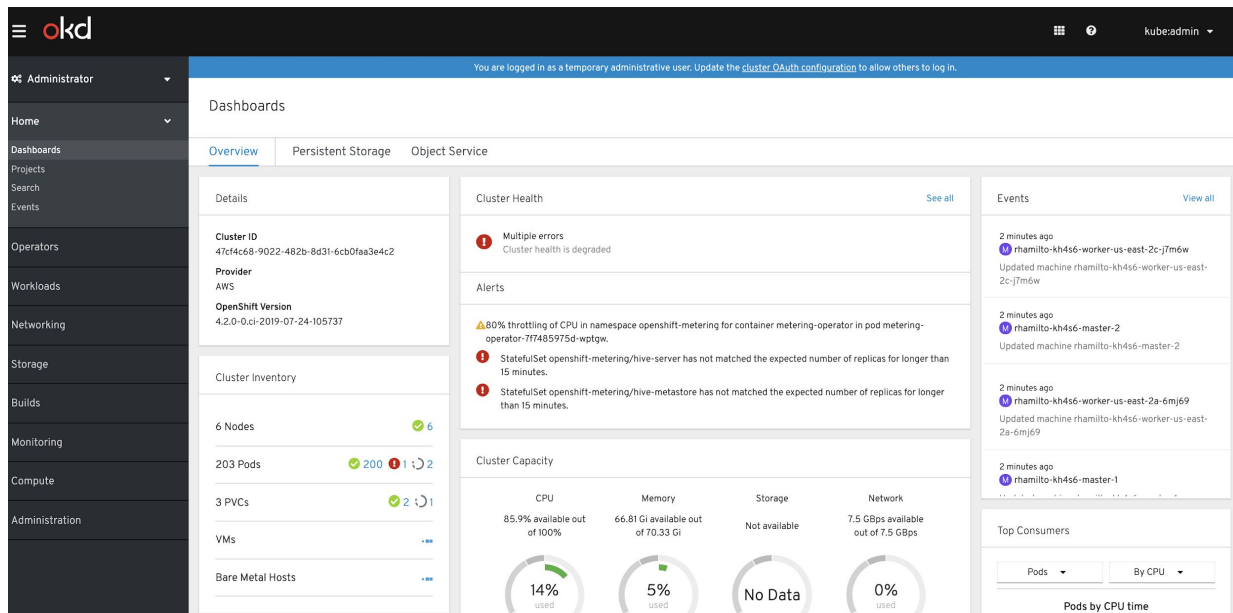
Build advanced operators  
for full lifecycle  
management

# Enhanced Visibility with the New Dashboard

## Cluster-wide Dashboard gives Admins Clear Insights

Drill down in context from the new dashboard widgets:

- Cluster Details
- Cluster Health
- Cluster Inventory
- Cluster Capacity
- Cluster Utilization
- Top Consumers

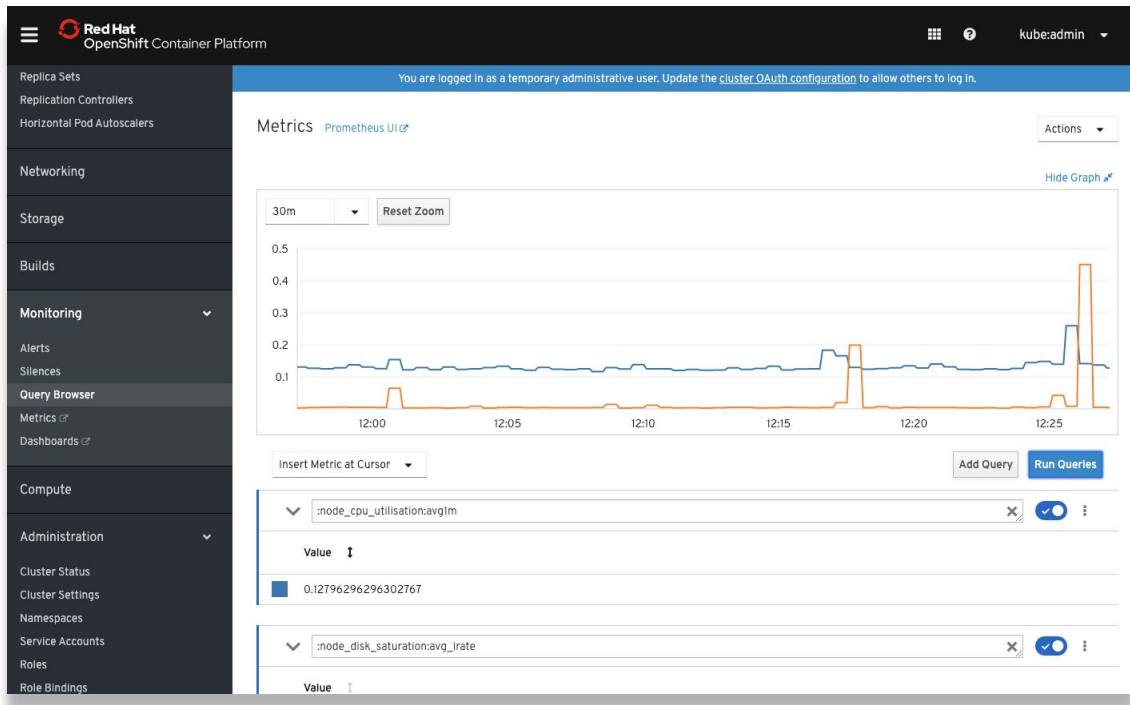


# Metrics with the new Query Browser

## Ad-hoc Query Browser

A native interface to create graphs on the fly to help triage what was going on in your cluster.

- Use PromQL to query cluster-level metrics and plot results into a graph.
- Decide whether you want to see your results in a graph or as text put into a table view.





# Chargeback Reports for metering

## Managing your Chargeback Reports inside the Console

- **Chargeback/Showback** - Break down reserved and utilized resources requested by applications.
- **Pod, Namespace, Node & Cluster Reports** - Built in reports to break down CPU, RAM and Storage.
- **Scheduled Reports** - Schedule reports to run on a standard interval, eg. daily or monthly
- **Post-Processing** - Reports are generated in CSV format and stored in persistent storage for further post-processing.

Project: openshift-metering

You are logged in as a temporary administrative user. Update the cluster `Default` configuration to allow others to log in.

Chargeback Reporting

All Reports Report Queries

Create Report Query

Filter by name...

Name	Namespace	Labels	Created At
cluster-cpu-capacity	openshift-metering	operator-metering=true	7 minutes ago
cluster-cpu-capacity-raw	openshift-metering	operator-metering=true	7 minutes ago
cluster-cpu-usage	openshift-metering	operator-metering=true	7 minutes ago
cluster-cpu-usage-raw	openshift-metering	operator-metering=true	7 minutes ago
cluster-cpu-utilization	openshift-metering	operator-metering=true	7 minutes ago
cluster-memory-capacity	openshift-metering	operator-metering=true	7 minutes ago
cluster-memory-capacity-raw	openshift-metering	operator-metering=true	7 minutes ago
cluster-memory-usage	openshift-metering	operator-metering=true	7 minutes ago
cluster-memory-usage-raw	openshift-metering	operator-metering=true	7 minutes ago
cluster-memory-utilization	openshift-metering	operator-metering=true	7 minutes ago

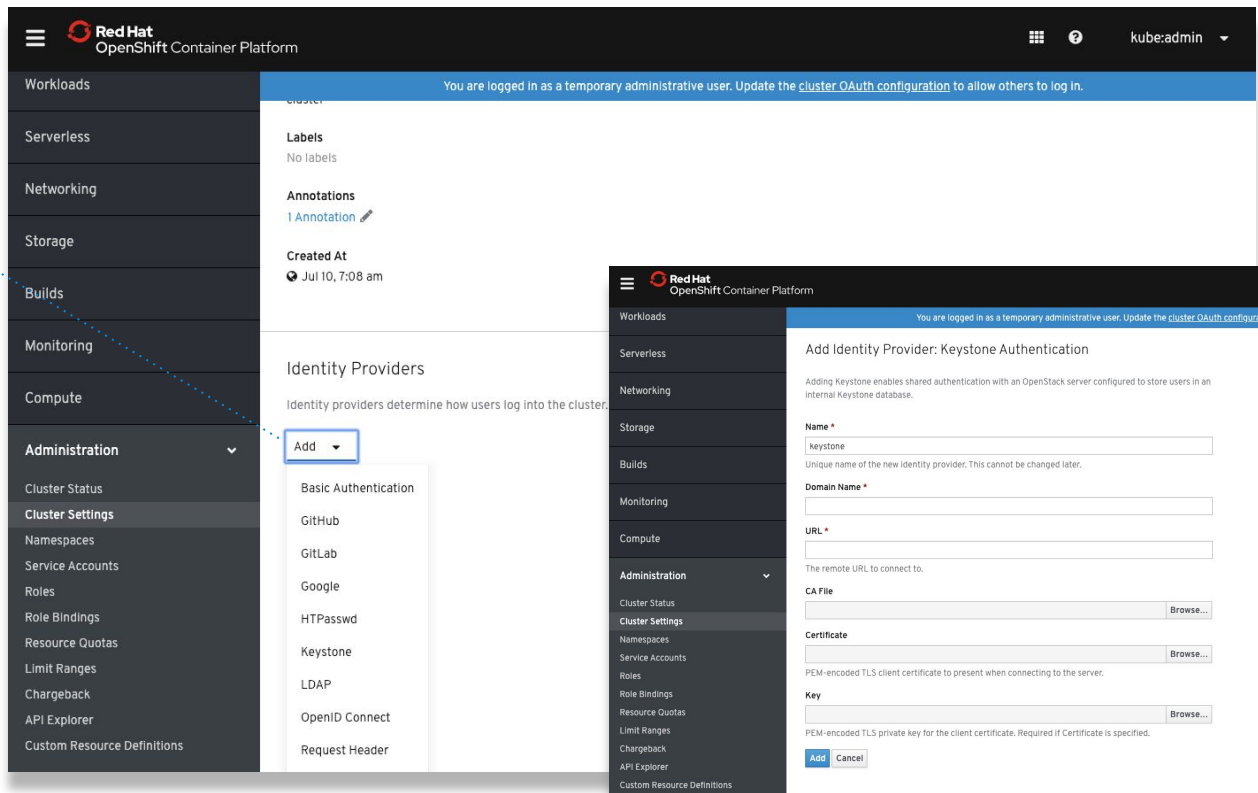
Generally Available



# Configuring Authentication for your desired Identity Providers

Customize and determine how users log into the cluster

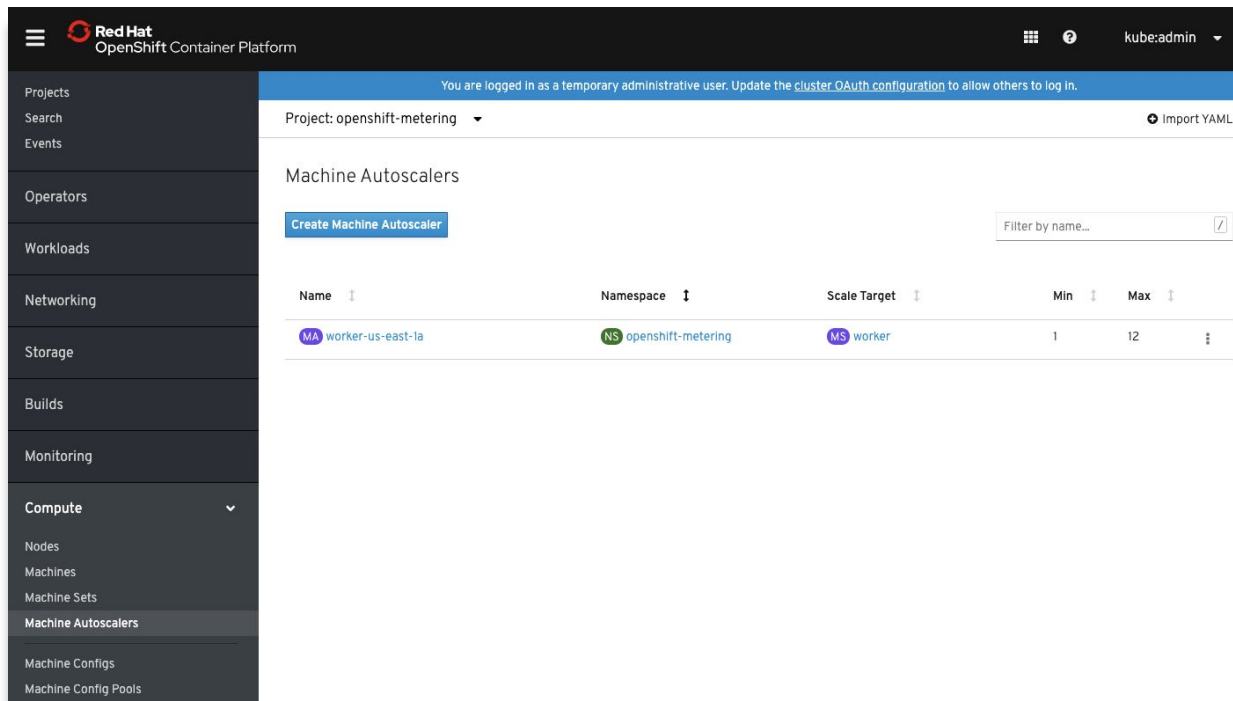
- Basic Authentication
- GitHub
- GitLab
- Google
- HTPasswd
- Keystone
- LDAP
- OpenID Connect
- Request Header



# Scaling Your Cluster with the Machine Autoscaler

**Machine Autoscaler** adjusts the number of Machines in the MachineSets being deployed in your cluster.

- Increase Machines when the cluster runs out of resources to support more deployments.
- Any changes such as the minimum or maximum number of instances, are immediately applied to the MachineSet that MachineAutoscalers target.



The screenshot displays the Red Hat OpenShift Container Platform console interface. The left sidebar contains a navigation menu with the following items: Projects, Search, Events, Operators, Workloads, Networking, Storage, Builds, Monitoring, Compute (expanded), Nodes, Machines, Machine Sets, Machine Autoscalers (selected), Machine Configs, and Machine Config Pools. The main content area shows the 'Machine Autoscalers' page for the 'openshift-metering' project. It includes a 'Create Machine Autoscaler' button and a table listing existing autoscalers.

Name	Namespace	Scale Target	Min	Max
MA worker-us-east-1a	NS openshift-metering	MS worker	1	12

# Explore Kube's APIs Interactively via API Explorer

## Explore and manage various Kubernetes APIs interactively

- View the **schema** for each API and what **parameters** being supported
- Manage the **instances** of the API
- Review the **access** of each API

The image displays four overlapping screenshots of the Red Hat OpenShift Container Platform API Explorer interface, demonstrating its capabilities for exploring and managing Kubernetes APIs.

**Top Screenshot: Explore API Resources**

This view shows a list of API resources. The table below represents the data shown:

Kind	Group	Version	Namespaced	Description
ComponentStatus	-	v1	false	ComponentStatus (and ComponentStatusList) holds the cluster validation info.
ConfigMap	-	v1	true	ConfigMap holds configuration data for pods to consume.
Event	-	v1	true	Event is a report of an event somewhere in the cluster.

**Bottom Left Screenshot: Role Access Review**

This view shows the 'Access Review' tab for a Role. It lists subjects that can create roles in the namespace 'testing'.

Subject
system:cluster-admins
system:masters
system:admin

**Bottom Middle Screenshot: Role Instances**

This view shows the 'Instances' tab for a Role. It lists the instances of the Role in the namespace 'testing'.

Name	Namespace
cockroachdb.v2.1.1-wg64r	testing
couchbase-operator.v1.0-ndjs	testing
dynatrace-monitoring.v0.3.1-j8x4	testing
etcdoperator.v0.9.4-jgbvc	testing

**Bottom Right Screenshot: Role Schema**

This view shows the 'Schema' tab for a Role. It displays the schema for the Role, including the 'apiVersion' and 'kind' fields.

• **apiVersion** string

APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest format in the API, unless a previous version was explicitly requested. More info: <https://git.k8s.io/community/contributors/devel/api-conventions.md#resources>

• **kind** string

Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client invokes. More info: <https://git.k8s.io/community/contributors/devel/api-conventions.md#types-kinds>

# Container Storage 4.2



# OpenShift Container Storage 4.2

## Persistent data services for OCP 4 Hybrid Cloud

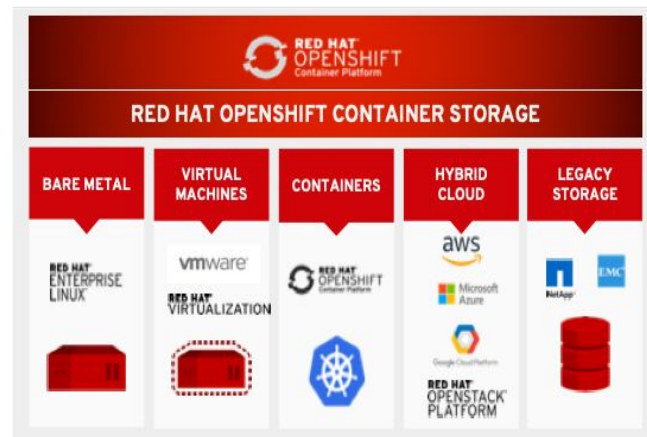
- **Complete Data Services: RWO, RWX & S3 (new) (block, file & object)**
- **Persistent storage for all OCP Infra and Applications**
- **Build and deploy anywhere**

## OCS 4.2 support with OCP 4.2

- **Platform support:** AWS and VMware
- **Converged Mode support :** Run as a service on OCP Cluster
- **Consistent S3 across hybrid cloud**

## Roadmap (OCS 4.3)

- Additional Platform: Bare Metal, Azure Cloud
- Independent Mode : Run OCS outside of OCP Cluster
- Hybrid and Multi-cloud S3



# OCS 4.2: Change in Technology Stack

## Goal to have complete storage for OCP

- **Need for scalable S3 object stack**
  - **Noobaa** is consistent S3 interface over Ceph RGW, AWS S3, Azure Blob
- **Red Hat Ceph**: scalable object stack with block and file
- **Rook**: operator framework for simple install, manage, expand

No change in OCS SKU or pricing

Full integrated migration support from OCP + OCS 3.11 to OCP + OCS 4.2  
(not in-place!)

OCS 3



OCS 4



# Empowering Developers to Inovate





	Databases	Data Warehouse	Streaming	Languages & Frameworks	SCM	Registry Services	Application Definition	CI / CD	Services as Code	API management
Application Definition & Development	              	     	       	              	  	     	    	            	        	         

	Scheduling & Orchestration	Coordination & Service Discovery	Service Management
Orchestration & Management	    	        	          

	OS	Cloud-Native Storage	Container Runtime	Cloud-Native Network
Runtime	    	                	      	                     

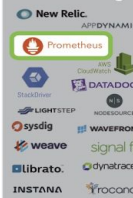
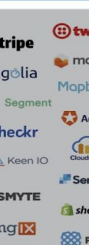
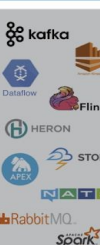
	Infrastructure Automation	Host Management / Tooling	Secure Images
Provisioning	     	    	   

	Infrastructure

Observability & Analysis
<p>Monitoring</p> <p>New Relic</p> <p>Prometheus</p> <p>DATADOG</p> <p>Wavefront</p> <p>Signal FX</p> <p>Librato</p> <p>INSTANA</p> <p>Trocano</p> <p>Opsclarity</p> <p>Nagios</p> <p>ZABBIX</p> <p>RIEMANN</p> <p>InfluxDB</p> <p>Graphite</p> <p>StatsDB</p> <p>Datadog</p> <p>COSCALE</p> <p>meros</p> <p>Applica</p> <p>beat</p> <p>catchpoint</p>
<p>Logging</p> <p>elastic</p> <p>loggly</p> <p>sumologic</p> <p>splunk</p> <p>graylog</p> <p>logz.io</p> <p>papertail</p>
<p>Event-based compute</p> <p>AWS Lambda</p> <p>Cloud Functions</p> <p>Firebase</p> <p>webtask</p> <p>StackHut</p> <p>SERVERLESS</p> <p>CFEngine</p> <p>fission</p> <p>Lever OS</p>
<p>Tracing</p> <p>Zipkin</p> <p>dtrace</p> <p>JAEGER</p>

## CNCF Projects

github.com/cncf/landscape



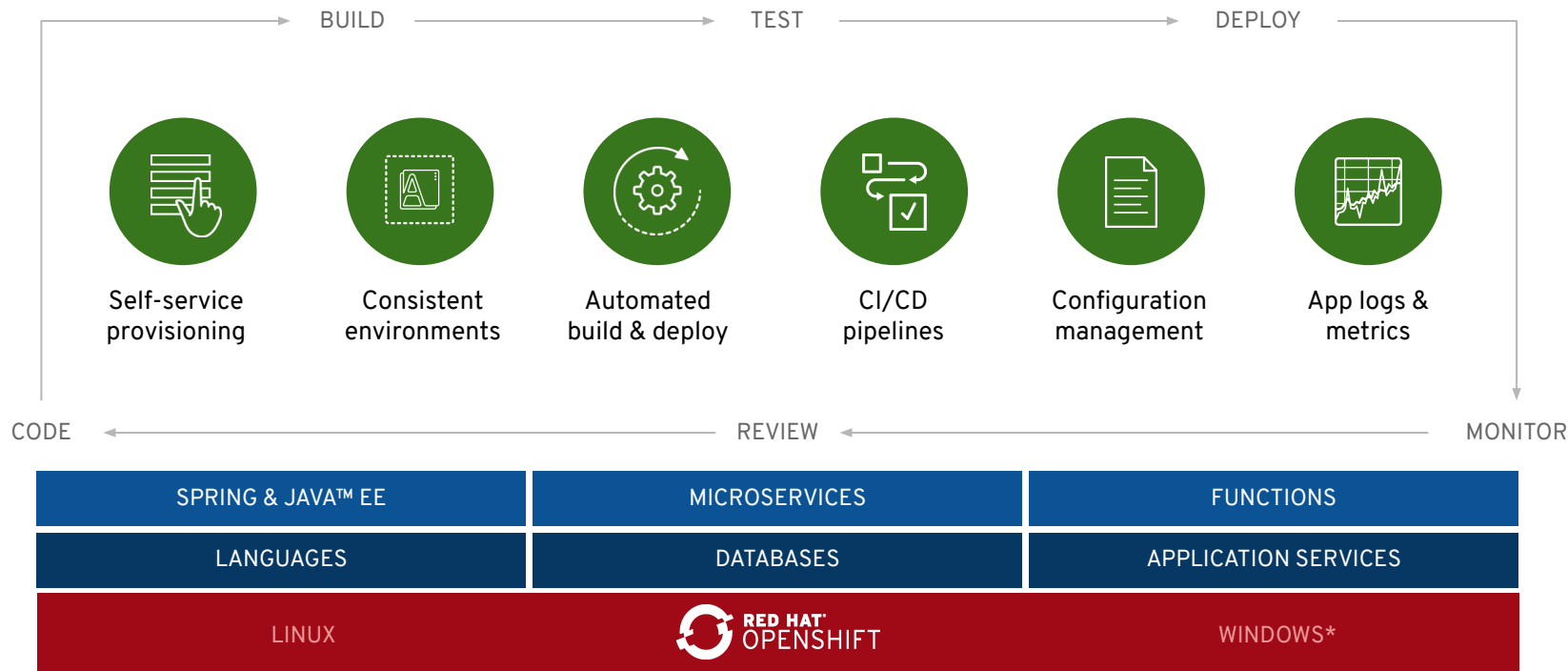
# THE CLOUD-NATIVE APP DEV CHALLENGE



CNCF Projects

[github.com/cncf/landscape](https://github.com/cncf/landscape)

# OpenShift enables developer productivity



\* coming soon

# Enabling greater developer productivity

## CodeReady Workspaces

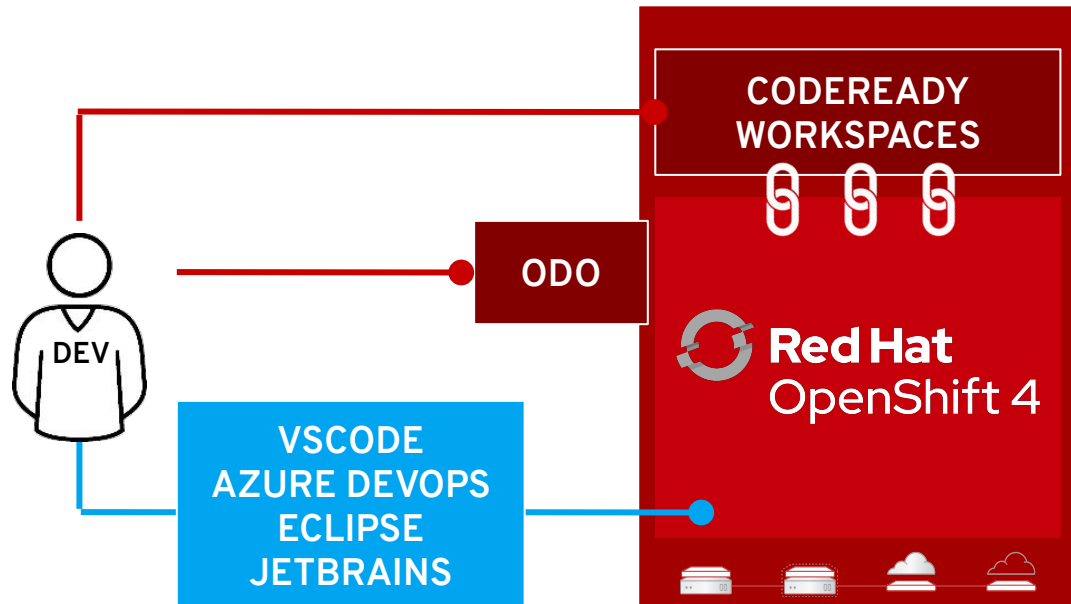
Web-Based IDE (Eclipse Che), Collaborative Development, integrated with CI/CD.

## OpenShift ODO

Advanced developer CLI

## OpenShift Plugins

Integration plugins - VScode, Azure DevOps, Eclipse IDE, JetBrains

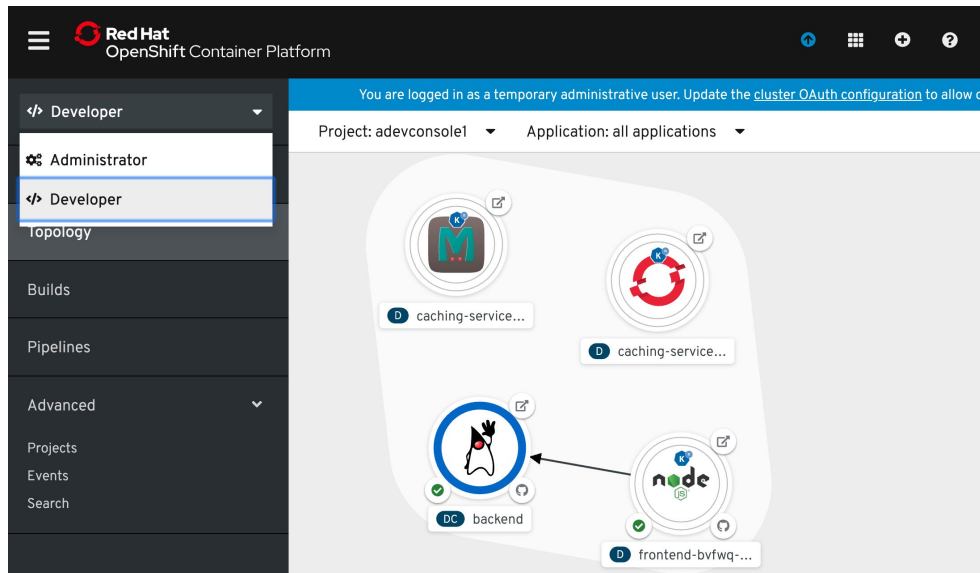


# Web Console - Developer Perspective

## Key Features

An alternative perspective in the OpenShift UI that will sit beside the admin console and focus on developer use cases.

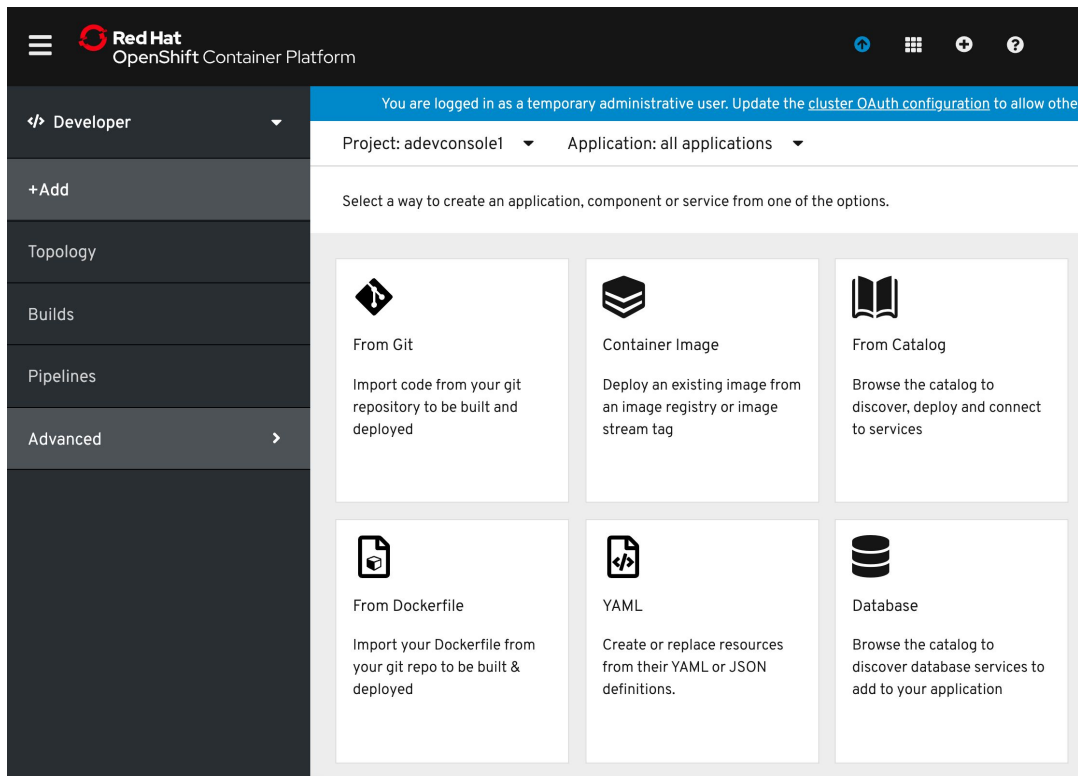
All OpenShift developer tool UIs will be surfaced here...though some (like CodeReady Workspaces) will be links out to unique UIs.



# Easily Create Application

## Key Features

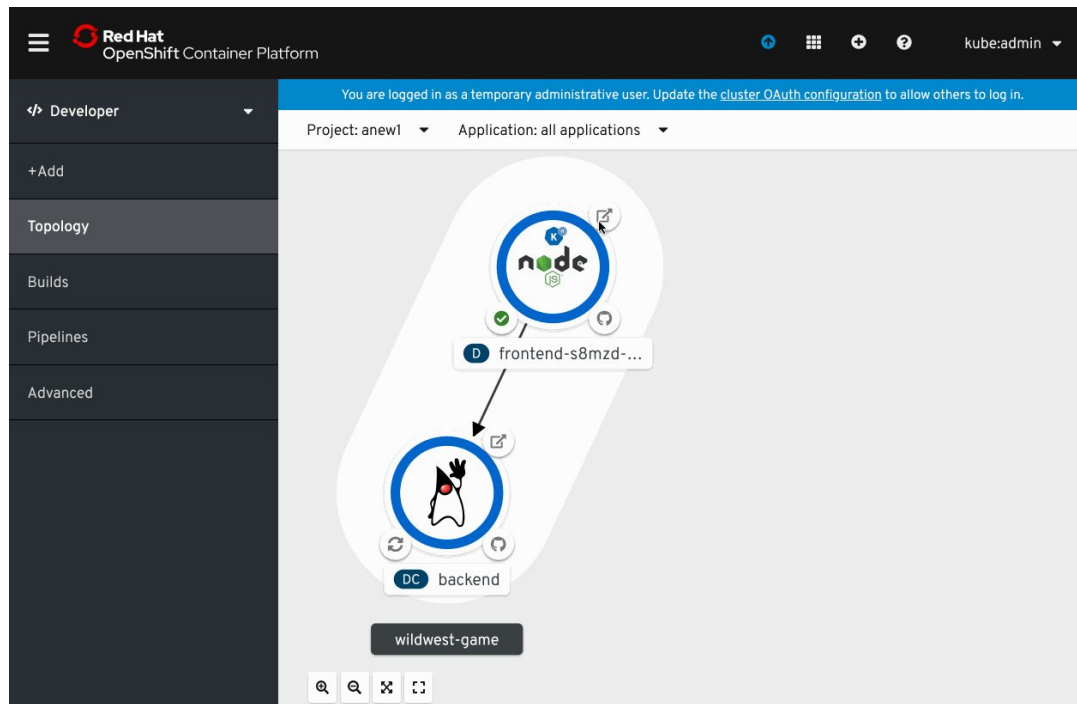
- Source in Git
- Existing container image
- YAML definition
- Build from Dockerfile
- Explore catalog
- Databases from catalog



# Application Topology

## Key Features

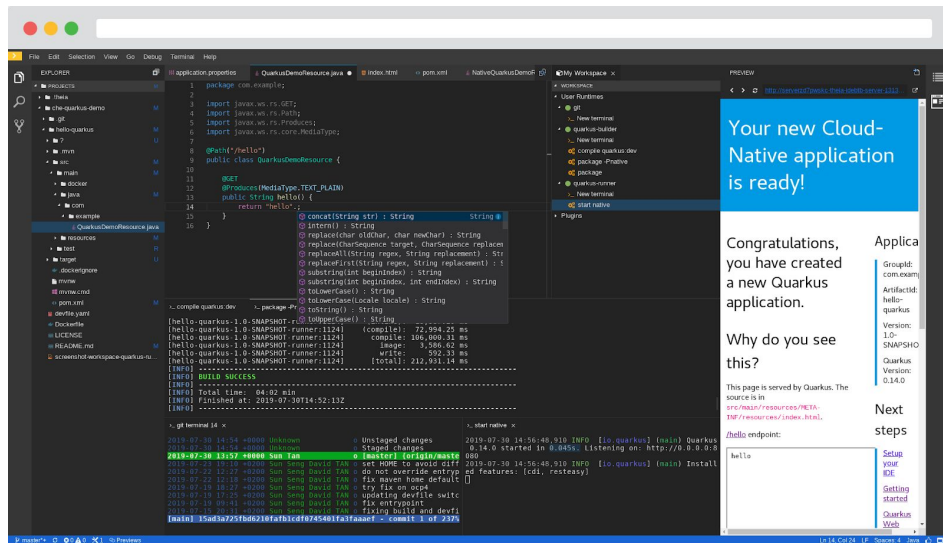
- View structure and status of app components
- Drill into specific workloads
- Quickly navigate to pod logs
- Manually scale
- Pod donut!
- Access route/URL
- Linked build and source



# CodeReady Workspaces 2.0

## Based on Eclipse Che 7

- **Kubernetes-based developer workspaces:** Fully containerized developer workspaces allowing to bring your K8S application runtime easily in your dev environment.
- **New Editor:** New default web-based editor provides a VSCode like experience in the browser.
- **Devfile:** Configure a devfile for your project and get reproducible and portable developer environments.
- **VSCode plug-ins compatibility**
- **Swappable Editor**
- **OpenShift VSCode Plug-in**
- **Easier to Monitor and Administrate:** Prometheus and Grafana dashboards.



Shipped independently from 4.2, shortly after

Planned for Q3FY20



# odo - Dev-Focused CLI

Developer-focused CLI  
for rapid development  
iterations on OpenShift

Simplifies building of  
microservices  
applications on  
OpenShift.

```
$ odo create wildfly backend
Component 'backend' was created.

$ odo push
Pushing changes to component: backend

$ odo create php frontend
Component 'frontend' was created.
To push source code to the component run 'odo push'

$ odo push
Pushing changes to component: frontend

$ odo url create
frontend - http://frontend-myapp.192.168.99.100.nip.io

$ odo watch
Waiting for something to change in /dev/frontend
```

Use It To: Enable the 'git push' flow developers love, but with OpenShift Kubernetes.

# CodeReady Containers: OpenShift on your Laptop

Provides a pre-built development environment based on **Red Hat Enterprise Linux** and **OpenShift** for quick container-based application development. Use with OpenShift on-premises or cloud.

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

$ crc start -b
crc-hyperkit-4.2.0.crcbundle
Start with the Hyperkit 4.2 bundle

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

- Based on OpenShift 4.x
- Linux (libvirt)
- Windows (Hyper-V)
- MacOS (Virtualbox)
- External beta available
- Replaces the 3.x experiences around:
  - Minishift
  - CDK
  - oc cluster up

**Use It To:** Simplify direct-to-OpenShift development on laptops.

# Roadmap



# 2019 Roadmap

Q2 CY2019 OpenShift 4.1	
DEV	<ul style="list-style-type: none"> <li>• OpenShift Serverless (Knative) - DP</li> <li>• OpenShift Pipelines (Tekton) DP2</li> <li>• CodeReady Workspaces</li> <li>• CodeReady Containers Alpha</li> <li>• Developer CLI (odo) Beta</li> </ul>
APP	<ul style="list-style-type: none"> <li>• OperatorHub</li> <li>• Operator Lifecycle Manager</li> <li>• Service Mesh (~4 month after)</li> </ul>
PLATFORM	<ul style="list-style-type: none"> <li>• Kubernetes 1.13 with CRI-O runtime</li> <li>• RHEL CoreOS, RHEL7</li> <li>• Automated Installer for AWS</li> <li>• Pre-existing Infra Installer for Bare Metal, VMware, AWS</li> <li>• Automated, one-click updates</li> <li>• Multus (Kubernetes multi-network)</li> <li>• Quay v3</li> </ul>
HOSTED	<ul style="list-style-type: none"> <li>• cloud.redhat.com - Multi-Cluster Mgmt</li> <li>• OCP Cluster Subscription Management</li> <li>• OpenShift Dedicated consumption pricing</li> </ul>

Q3 CY2019 OpenShift 4.2	
DEV	<ul style="list-style-type: none"> <li>• Developer Console GA</li> <li>• OpenShift Serverless (Knative) - TP</li> <li>• OpenShift Pipelines (Tekton) DP3</li> <li>• CodeReady Containers GA</li> <li>• Developer CLI (odo) GA</li> </ul>
APP	<ul style="list-style-type: none"> <li>• OperatorHub Enhancements</li> <li>• Operator Deployment Field Forms</li> <li>• Application Migration Console</li> </ul>
PLATFORM	<ul style="list-style-type: none"> <li>• Kubernetes 1.14 w/ CRI-O runtime</li> <li>• Disconnected Install and Update</li> <li>• Automated Installer for Azure, OSP, GCP</li> <li>• Pre-existing Infra Installer for GCP</li> <li>• Cluster-wide Egress Proxy</li> <li>• OVN Tech Preview</li> <li>• OpenShift Container Storage 4.2 (1 month after)</li> </ul>
HOSTED	<ul style="list-style-type: none"> <li>• Insights Operator</li> <li>• Azure Red Hat OpenShift new features (monitoring, logging)</li> </ul>

Q4 CY19/Q1 CY20 OpenShift 4.3	
DEV	<ul style="list-style-type: none"> <li>• <b>OpenShift Serverless (Knative) - GA</b></li> <li>• OpenShift Pipelines (Tekton) TP</li> <li>• Helm 3 TP</li> </ul>
APP	<ul style="list-style-type: none"> <li>• Metering for Services</li> <li>• <b>Windows Containers (Planned)</b></li> <li>• GPU Metering</li> <li>• Application Operator Binding - DP</li> </ul>
PLATFORM	<ul style="list-style-type: none"> <li>• <b>Kubernetes 1.16 w/ CRI-O runtime</b></li> <li>• <b>Automated Installer for RHV</b></li> <li>• Private/Internal Clusters support from the installer</li> <li>• Deploy to pre-existing VPC &amp; Subnets</li> <li>• OVN GA w/ Windows Networking Integration (Planned)</li> <li>• FIPS</li> <li>• <b>Pre-existing Infra Installer for OSP</b></li> <li>• OpenShift Container Storage 4.3</li> </ul>
HOSTED	<ul style="list-style-type: none"> <li>• cloud.redhat.com - Subscription Mgmt Improvements</li> <li>• Azure Red Hat OpenShift new features (private clusters)</li> <li>• Azure Red Hat OpenShift preview of 4.x</li> <li>• OSD on Google Cloud preview on 4.x</li> </ul>

# Thank You



[linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://facebook.com/redhatinc)



[twitter.com/RedHat](https://twitter.com/RedHat)