Red Hat Day Events

January 30, Vancouver

OpenShift
The Platform for Big Ideas

Red Hat

OpenShift - Infrastructure & Ops

Brian Gracely

Sr. Director Product Strategy

Red Hat OpenShift

@bgracely

bgracely@redhat.com

Red Hat

If you forget everything else…

Hands on OpenShift Learning : http://learn.openshift.com

Get the OpenShift software : http://try.openshift.com

OpenShift Videos : https://www.youtube.com/user/rhopenshift/playlists

OpenShift Demos : http://demo.openshift.com

OpenShift 3 was great, except…

- Installations were not always easy
- Upgrades were not always easy
- Operations didn’t always have the best visibility
- Integrating new capabilities wasn’t always simple

Openshift 4 Themes

DAY 2
OPERATIONS

IMMUTABLE
INFRASTRUCTURE

OPERATOR
FRAMEWORK

Red Hat

Reimaging OpenShift - OpenShift 4

Drive the Operator Ecosystem	Integrate Application Services	Improve Developer Services
Improve Platform Automation	Enable Telemetry	Multi-Cloud Management
Simplify Installations and Upgrades	Provide Ops Greater Visibility	Simplify Node Management

Red Hat

Market Expectations for Application Platforms

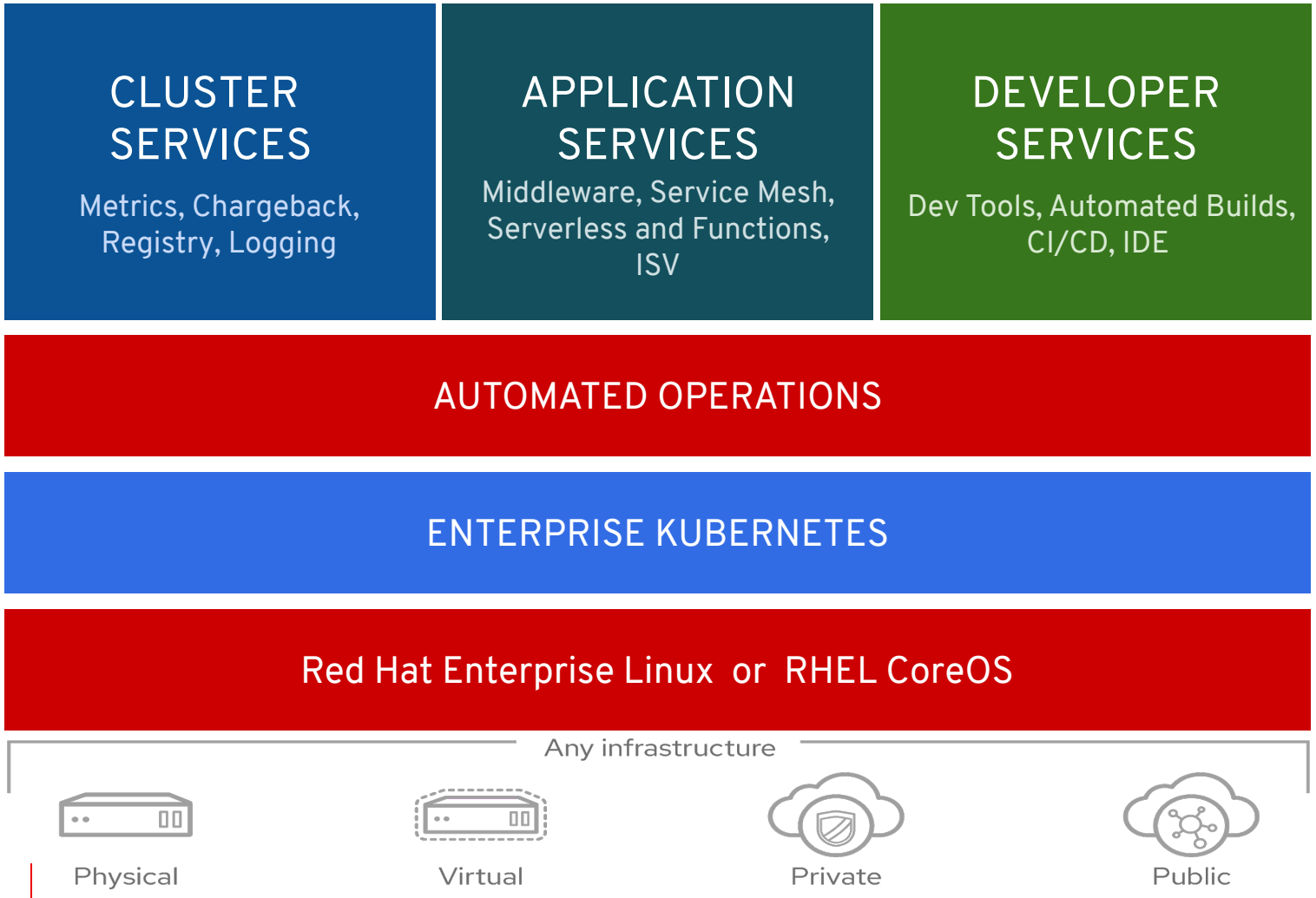
CLOUD EXPECTATIONS

- BROAD APPLICATION SUPPORT
- CONTINUOUS SECURITY
- ON-DEMAND | AS-A-SERVICE MARKETPLACES
- STANDARDS | INTEROPERABILITY | PORTABILITY | AVOID LOCK-IN
- LIMITED OPERATIONS or MANAGED SERVICES

Red Hat OpenShift

OpenShift 4: A smarter Kubernetes platform

Extending the platform boundary, freeing customers to innovate



- Fully integrated and automated
- Seamless Kubernetes deployment
- Fully automated installation
- 1-click platform updates
- Autoscaling of cloud resources

Red Hat

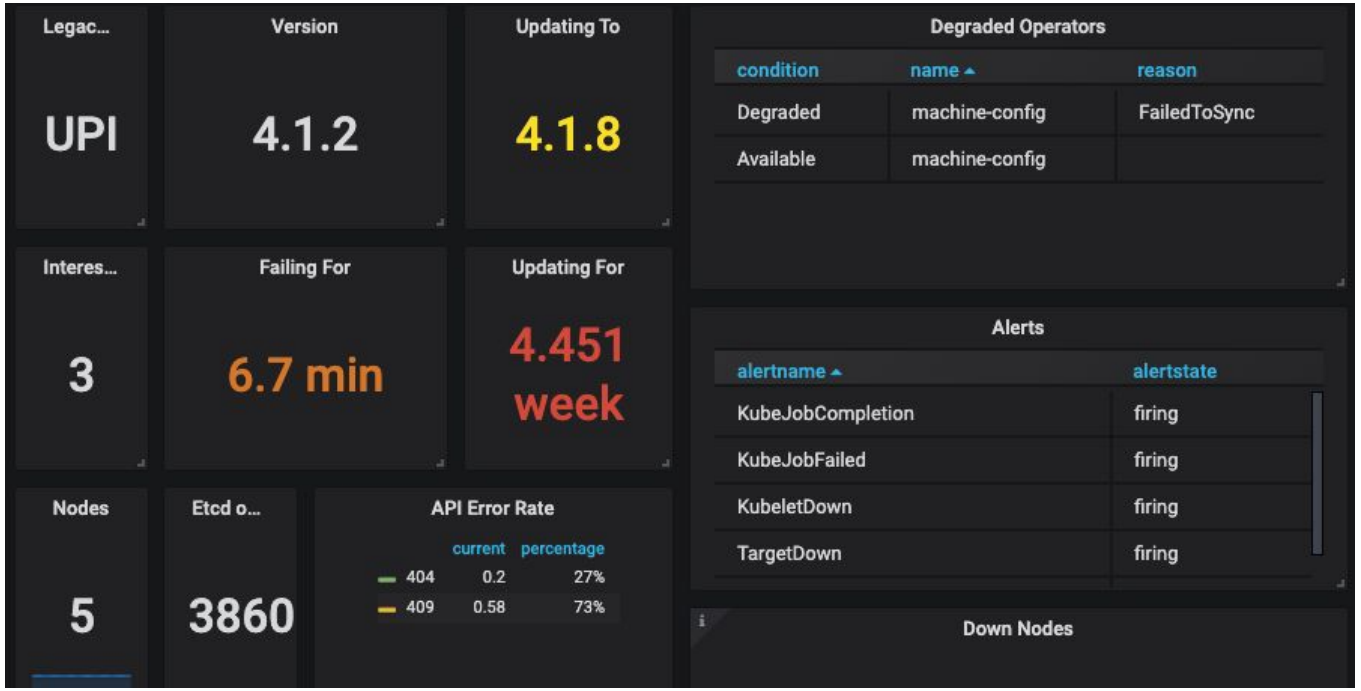
Connected Customer Experience

Proactive support for customer issues

- Active upgrades
- Overall cluster health
- Firing alerts
- Node health

Driving a high quality product

- Monitor and improve upon the health of the customer base
- Prioritize engineering roadmap for platforms and prove they are improving over time
- Active monitoring of fast and stable channels



Red Hat

Hosted OpenShift

Get the best of OpenShift without being on call

Red Hat

One Platform, Flexible Consumption Models

Red Hat OpenShift Dedicated

Managed service offering on public cloud

Red Hat / **Microsoft**

Azure Red Hat OpenShift

Jointly engineered, operated, and supported by Microsoft and Red Hat

HOSTED SERVICES

Red Hat OpenShift Container Platform

Enterprise-grade Kubernetes platform that you manage

SELF-MANAGED

Installation and Upgrades

Installation Experiences

OPENSHIFT CONTAINER PLATFORM

Full Stack Automation

Simplified opinionated "Best Practices" for cluster provisioning

Fully automated installation and updates including host container OS.

Red Hat Enterprise Linux CoreOS

IPI - Installer Provisioned Infrastructure

Pre-existing Infrastructure

Customer managed resources & infrastructure provisioning

Plug into existing DNS and security boundaries

Red Hat Enterprise Linux CoreOS

Red Hat Enterprise Linux

UPI - User Provisioned Infrastructure

HOSTED OPENSHIFT

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.

Red Hat

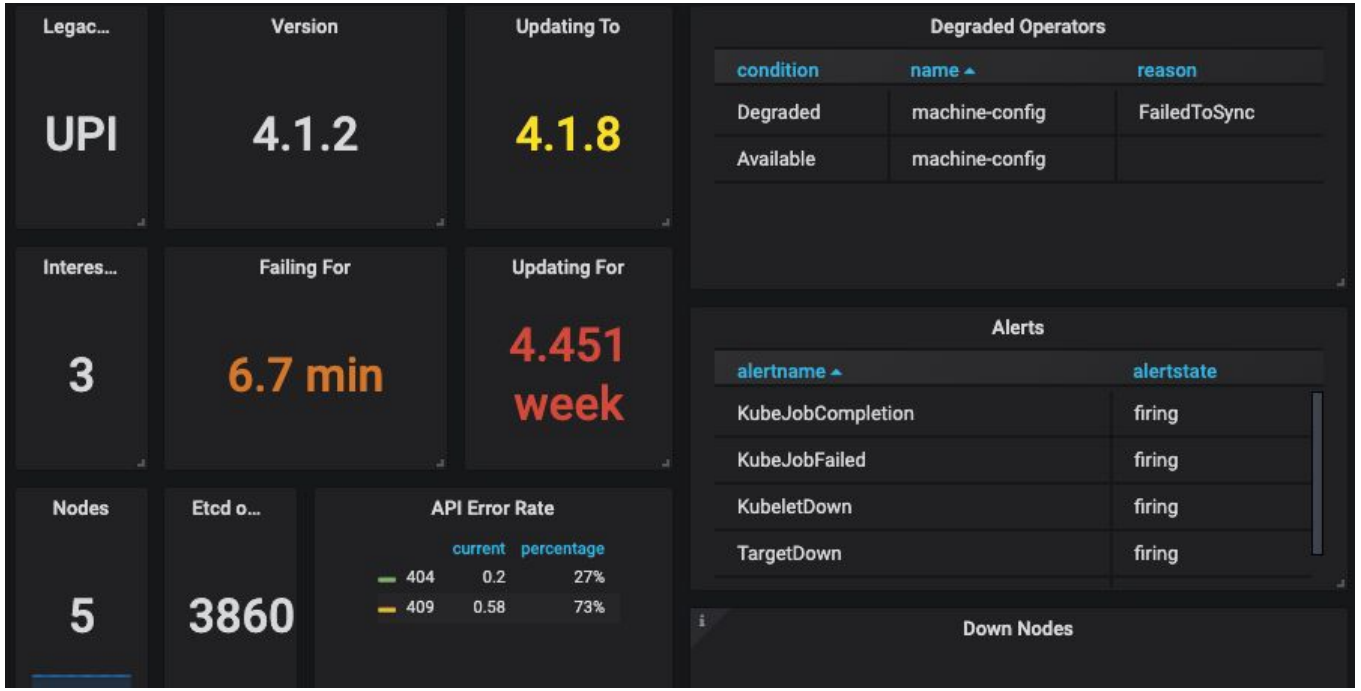
Connected Customer

Proactive support for customer issues

- Active upgrades
- Overall cluster health
- Firing alerts
- Node health

Driving a high quality product

- Monitor and improve upon the health of the customer base
- Prioritize engineering roadmap for platforms and prove they are improving over time
- Active monitoring of fast and stable channels



Red Hat

Install OpenShift on Any Cloud

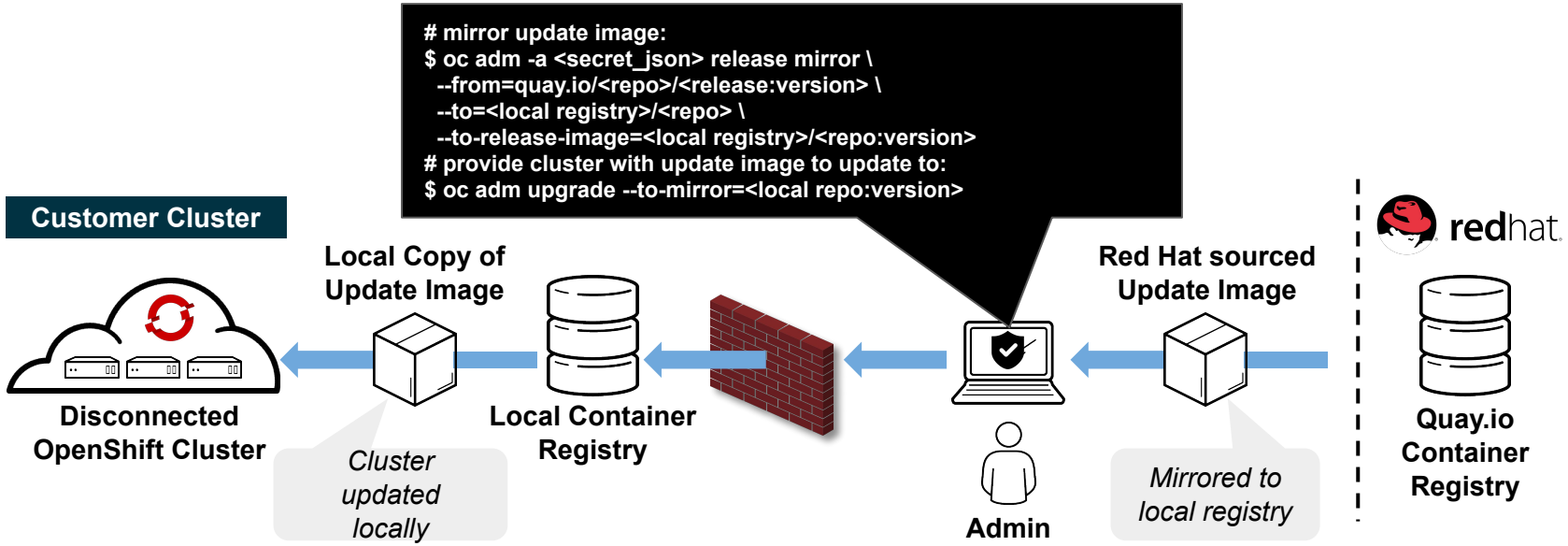
Clusters > Create > OpenShift Container Platform

Red Hat OpenShift | Install OpenShift Container Platform 4

Select an infrastructure provider

- aws — Run on Amazon Web Services
- Azure — Run on Microsoft Azure
- Google Cloud — Run on Google Cloud Platform
- vmware vSphere — Run on VMWare vSphere
- Run on Bare Metal
- IBM Z. IBM LinuxONE — Run on IBM Z
- Red Hat OpenStack Platform — Run on Red Hat OpenStack
- Run on Laptop — Powered by Red Hat CodeReady Containers

Disconnected “Air-gapped” Installation & Upgrading



Overview

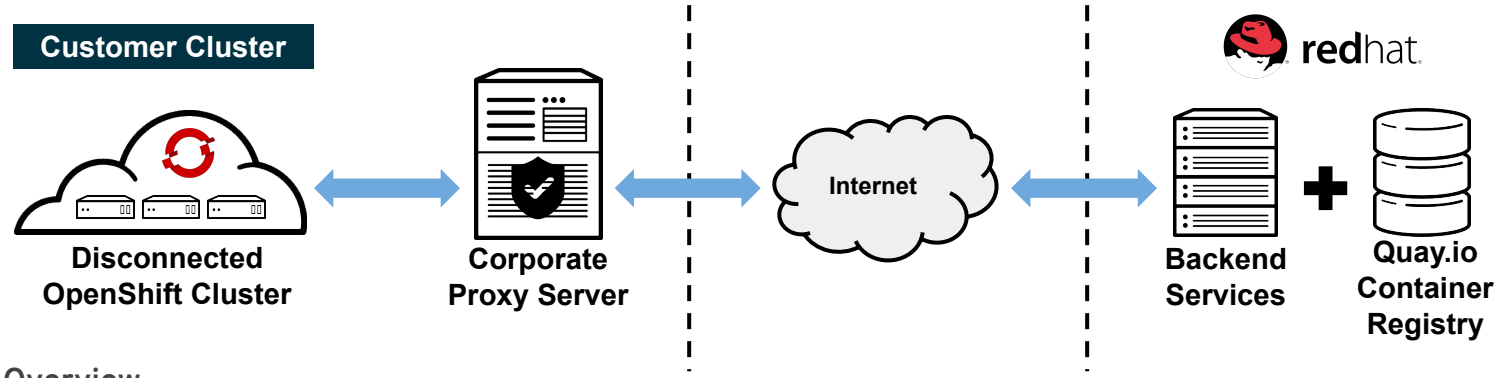
- 4.2 introduces support for installing and updating OpenShift clusters in disconnected environments
- Requires local Docker 2.2 spec compliant container registry to host OpenShift content
- Designed to work with the user provisioned infrastructure deployment method
  - *Note: Will not work with Installer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

Installation Procedure

- Mirror OpenShift content to local container registry in the disconnected environment
- Generate install-config.yaml: $ ./openshift-install create install-config --dir <dir>
  - Edit and add pull secret (PullSecret), CA certificate (AdditionalTrustBundle), and image content sources (ImageContentSources) to install-config.yaml
- Set the OPENSHIFT_INSTALL_RELEASE_IMAGE_OVERRIDE environment variable during the creation of the ignition configs
- Generate the ignition configuration: $ ./openshift-install create ignition-configs --dir <dir>
- Use the resulting ignition files to bootstrap the cluster deployment

Red Hat

Cluster-wide Egress Proxy



Overview

- 4.2 introduces support for installing and updating OpenShift clusters through a corporate proxy server
- Leverages new proxy controller within the cluster-network-operator, which is responsible for:
  - Reconciling a proxy object and writing spec > status upon successful validation.
  - Reconciling user-provided trust bundles referenced by trustedCA, validating the trust bundle certificates, merging the certificates with the system trust bundle and publishing the merged bundle to the openshift-config-managed/trusted-ca-bundle configmap.

Installation Procedure

- Installer will use PROXY* environment variables from the shell it's invoked from
- Generate install-config.yaml: $ ./openshift-install create install-config --dir <dir>
  - Edit proxy information (httpProxy, httpsProxy, & noProxy) and CA certificate (AdditionalTrustBundle) to install-config.yaml
- Installer validates the provided install-config.yaml parameters, renders the necessary assets to create the cluster, and initiates the installation process based on the install method used: $ ./openshift-install create cluster --dir <dir>

An admin with privileges can interact with the proxy object using 'oc' commands (use the 'oc edit' command to modify the proxy information.) Here is an example proxy

```
$ oc get proxy/cluster -o yaml
apiVersion: config.openshift.io/v1
kind: Proxy
metadata:
  creationTimestamp: "2019-08-21T22:36:49Z"
  generation: 2
  name: cluster
  resourceVersion: "24913"
  selfLink: /apis/config.openshift.io/v1/proxies/cluster
  uid: 2a344b01-d267-11f9-a4f3-025de4b59c38
spec:
  httpProxy: http://<username>:<pswd>@<ip>:<port>
  httpsProxy: https://<username>:<pswd>@<ip>:<port>
  noProxy: example.com
  readinessEndpoints:
  - http://www.google.com
  - https://www.google.com
  trustedCA:
    name: user-ca-bundle
status:
  httpProxy: http://<username>:<pswd>@<ip>:<port>
  httpsProxy: https://<username>:<pswd>@<ip>:<port>
  noProxy:
10.0.0.0/16,10.128.0.0/14,127.0.0.1,169.254.169.254,172.30
.0.0/16,api-int.demo.example.com,api.demo.example.openshif
t.com,etcd-0.demo.example.com,etcd-1.demo.example.com,etcd
-2.demo.example.com,example.com,localhost
```

Red Hat

Support for installing to pre-existing VPC/VNet & Subnets

Deploying to shared VPC/VNet & Subnets

- Allows OpenShift clusters to be deployed to a pre-existing, customer managed VPC/VNet on supported public cloud providers
  - Often corporate guidelines prohibit the creation of new VPCs or user accounts don't have the proper permissions to do so

Requirements

- Depending on the cloud provider, OpenShift install no longer creates the following infrastructure objects (refer to documentation for specifics on each provider):
  - *Internet gateways, NAT gateways, Subnets, Route tables, VPCs/VNets, VPC DHCP options, VPC endpoints, or Network Security Groups*
- Admins need to edit the install-config.yaml file to customize the details of your network configuration
  - VPC/VNet validation is performed by the installer to ensure the subnets you provide exist and suitable for deploying OpenShift

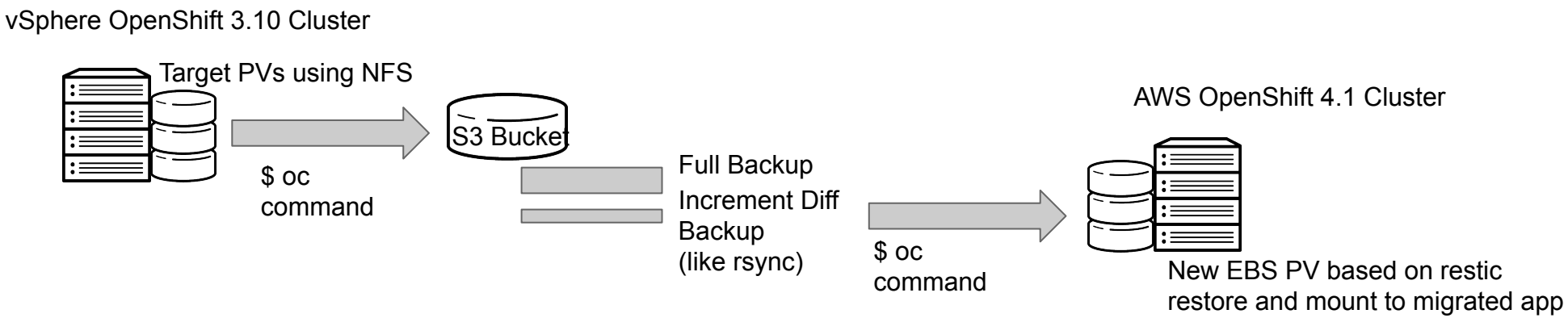
Documentation

- **AWS:** https://docs.openshift.com/container-platform/4.3/installing/installing_aws/installing-aws-vpc.html
- **Azure:** https://docs.openshift.com/container-platform/4.3/installing/installing_azure/installing-azure-vnet.html
- **GCP:** https://docs.openshift.com/container-platform/4.3/installing/installing_gcp/installing-gcp-vpc.html

```yaml
metadata:
  name: test-cluster
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineCIDR: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  aws:
    region: us-west-2
    userTags:
      adminContact: jdoe
      costCenter: 7536
    subnets:
    - subnet-1
    - subnet-2
    - subnet-3
pullSecret: '{"auths": ...}'
fips: false
sshKey: ssh-ed25519 AAAA...
```

Red Hat

CLUSTER MIGRATION OPENSHIFT 3 to 4



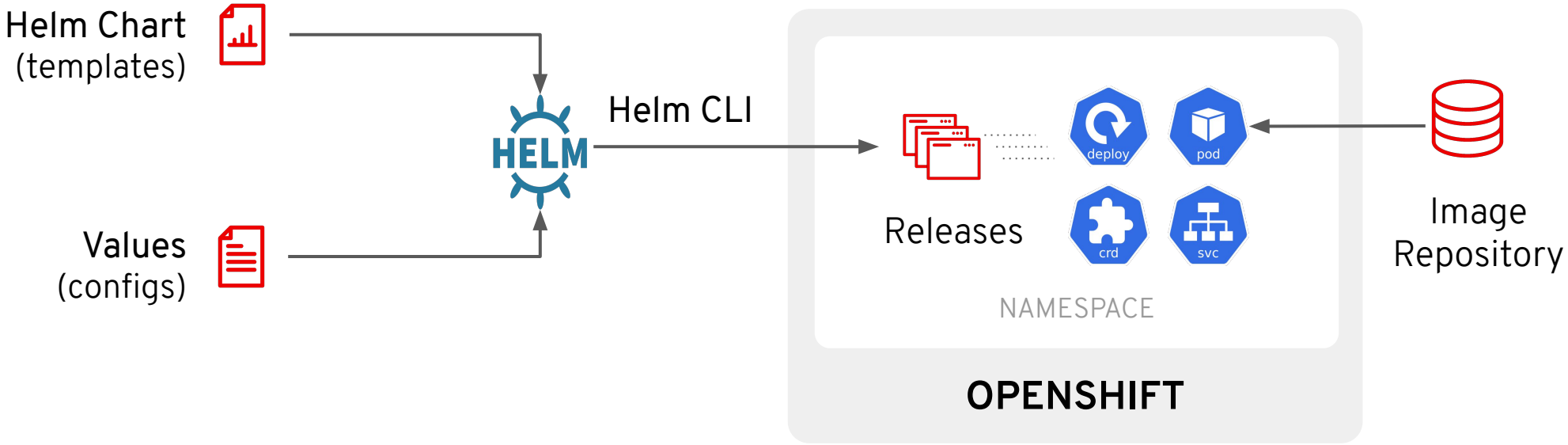
- Deploy a replication of your applications from one OpenShift cluster to a different OpenShift cluster
- Enable cluster specific configuration from OpenShift 3 to work on a OpenShift 4 cluster
- Documentation on how to handle common network, storage, and machine/node re-use scenarios between OpenShift 3 and OpenShift 4 clusters

Red Hat

Helm Charts

Helm 3 on OpenShift

Helm is a package manager for Kubernetes applications and helps to define, install and update apps



Red Hat

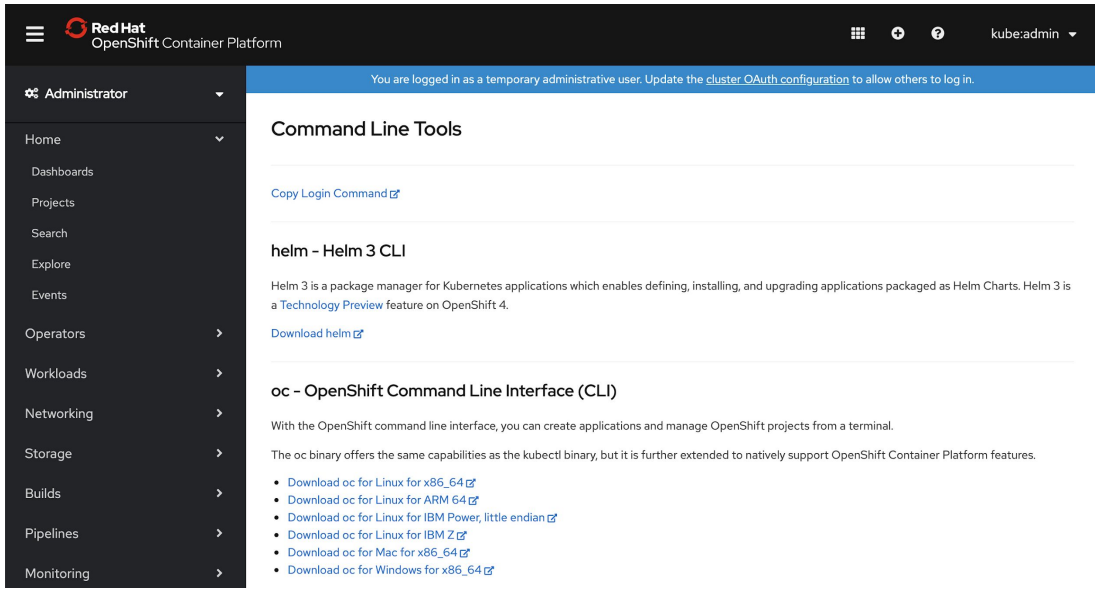
Helm 3 on OpenShift

OpenShift 4.3

- Helm 3 CLI in Tech Preview
- Built and shipped with OpenShift
- Available in Console CLI menu
- Added to OpenShift Docs

OpenShift 4.4+

- Helm 3 in Dev Console
  - Charts in Developer Catalog
  - Releases in Dev Console
  - Update/rollback/delete
- Helm developer guides



Red Hat

Helm and Operators

Package and Install

Automated Day-2 Operations

Helm

Operator

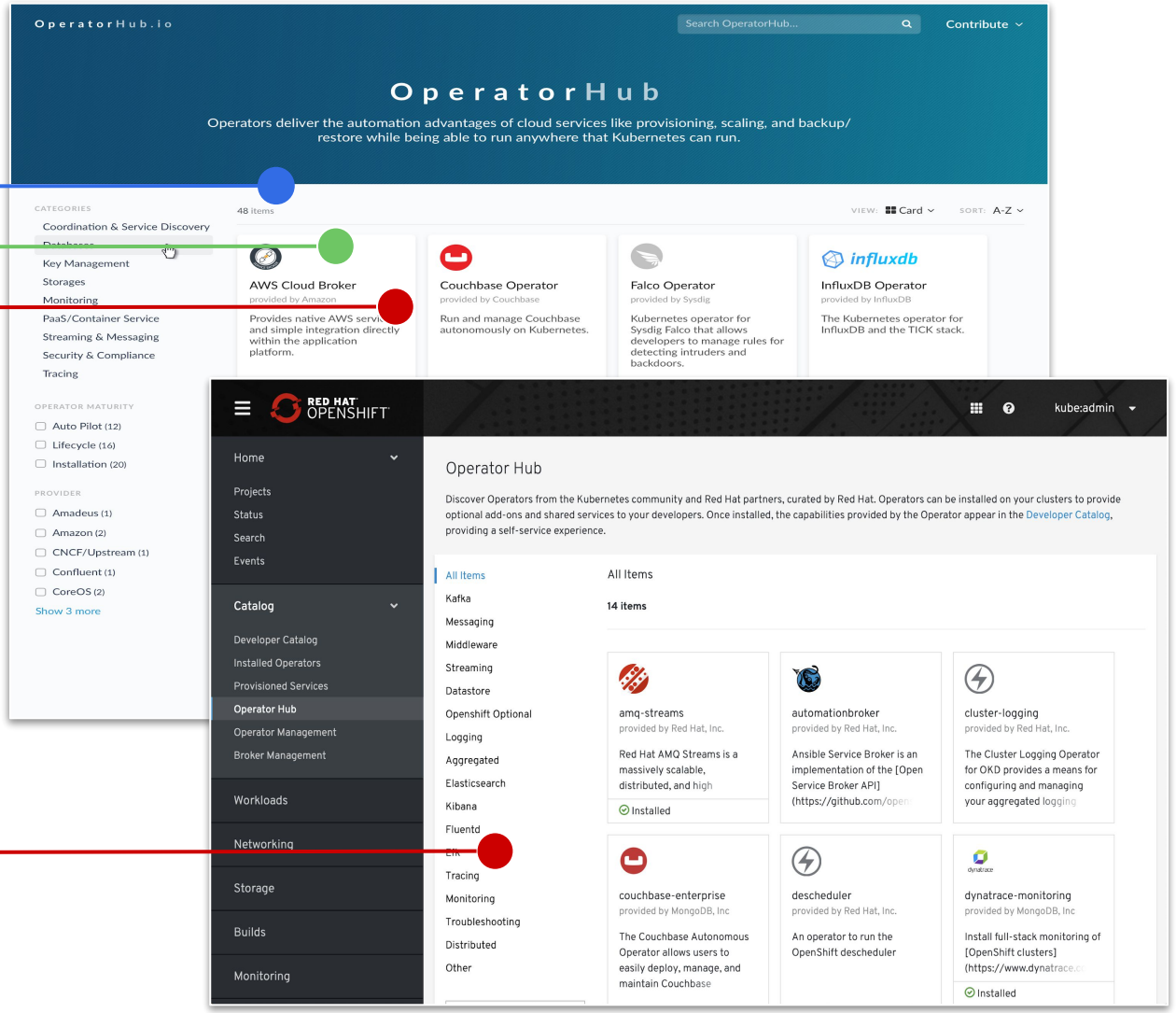
Phase I	Phase II	Phase III	Phase IV	Phase V
Basic Install	Seamless Upgrades	Full Lifecycle	Deep Insights	Auto Pilot
Automated application provisioning and configuration management	Patch and minor version upgrades supported	App lifecycle, storage lifecycle (backup, failure recovery)	Metrics, alerts, log processing and workload analysis	Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning

Red Hat

Day 2 Operations and Cluster Mgm’t

Openshift Operators enable ISV Innovation

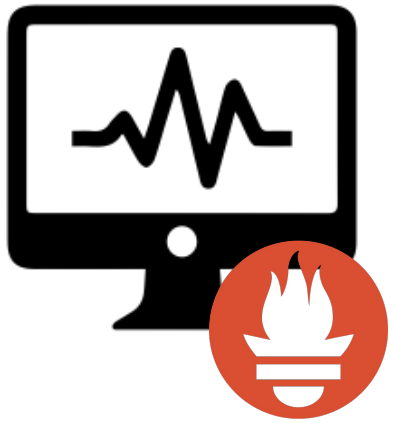
operatorhub.io



TYPES OF OPERATORS

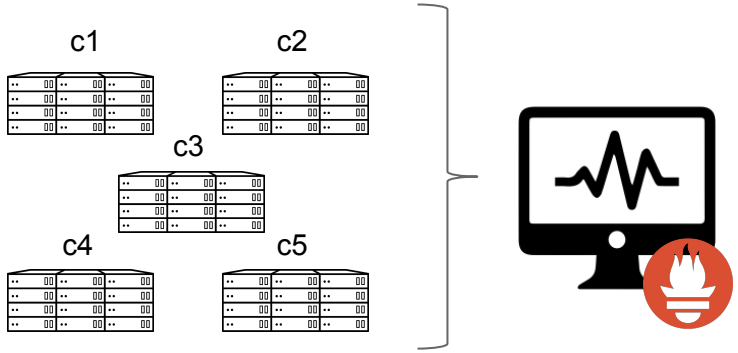
Community
ISV Partners
Red Hat Products

OPENSHIFT MONITORING



Application Monitoring

Support for Grafana



Multi-cluster monitoring & observability

Red Hat

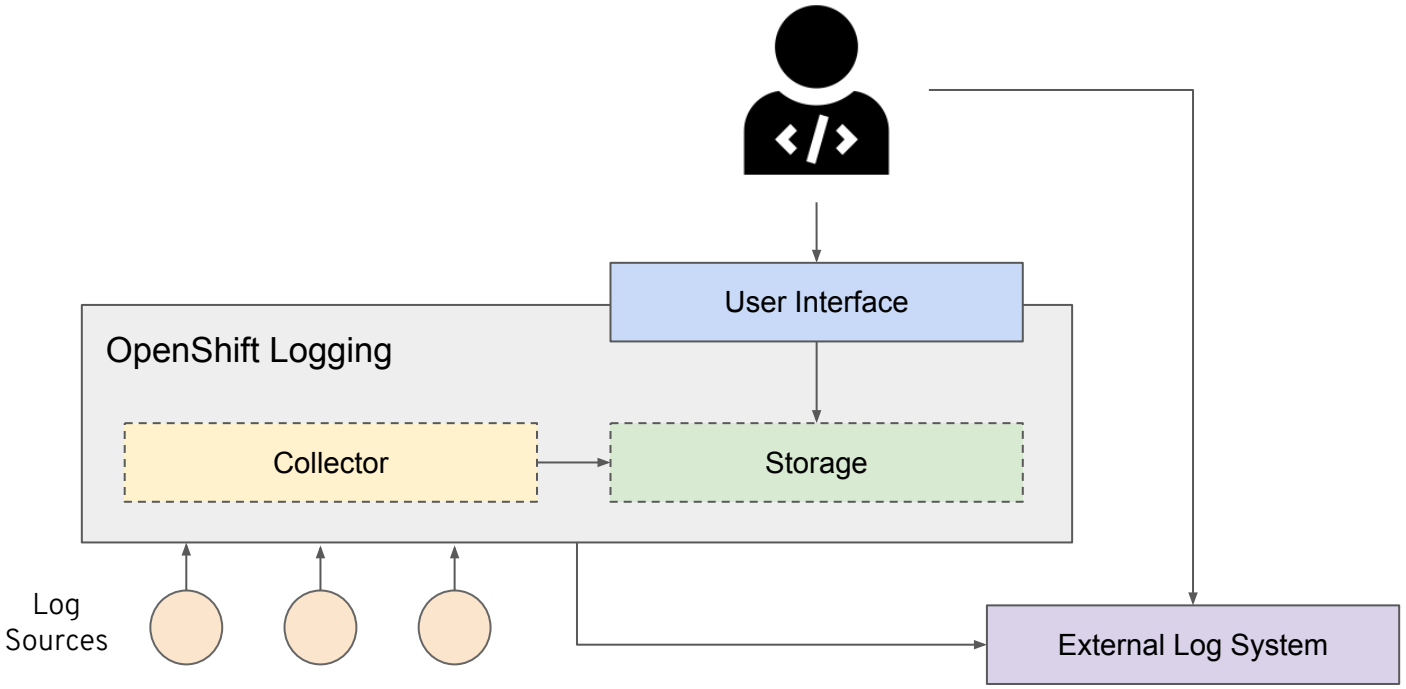
OPENSHIFT LOGGING

Short term goals:

- Enhancing log forwarding to external system(s)
- Newer ElasticSearch version
- Stability

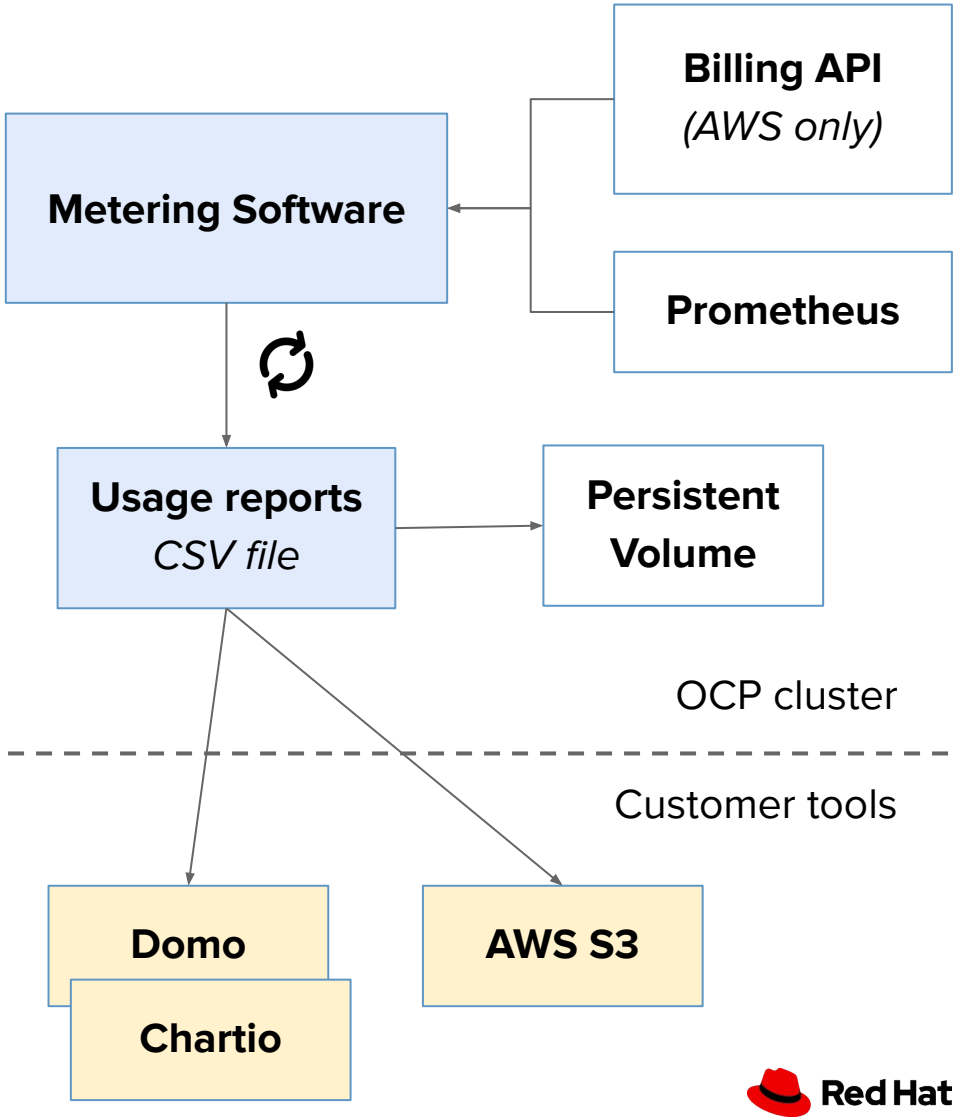
Long term goal:

Provide a more comprehensive, simplified logging stack that is optimized for Observability in mind.



ENABLING BILLING USING METERING OPERATOR

- Periodic reports
  - Requested resources or usage based
  - Reports per pod, node or namespace
  - AWS only: calculate $$ amount for reports
- Only tracks CPU, Memory, Storage to start
- Basis for future consumption based pricing
- Offer basic UI reporting but main use is to plug into customer's BI tool of choice



Red Hat

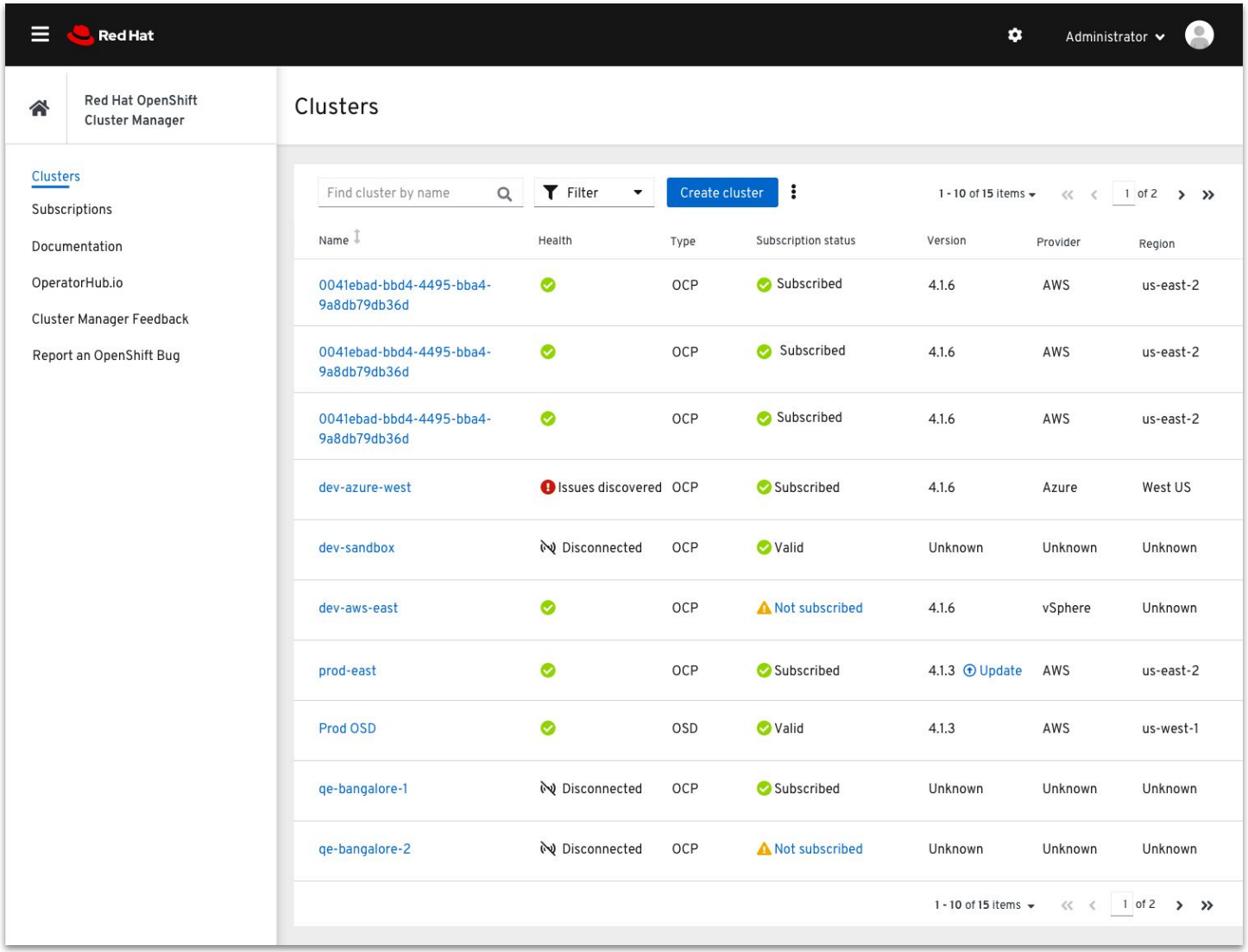
OPENSHIFT CLUSTER MANAGER

Short term goals:

- Create your OpenShift cluster
- Register your cluster with Red Hat
- View high level cluster info at a glance
- Navigate to all of your clusters and relevant Red Hat services

Long term goal:

Centralized cluster: registry, health, updates



Red Hat

OpenShift Console

The future is now.

Extending the Console	**Improve Observability**
Administration made easy	**Developer Focused**

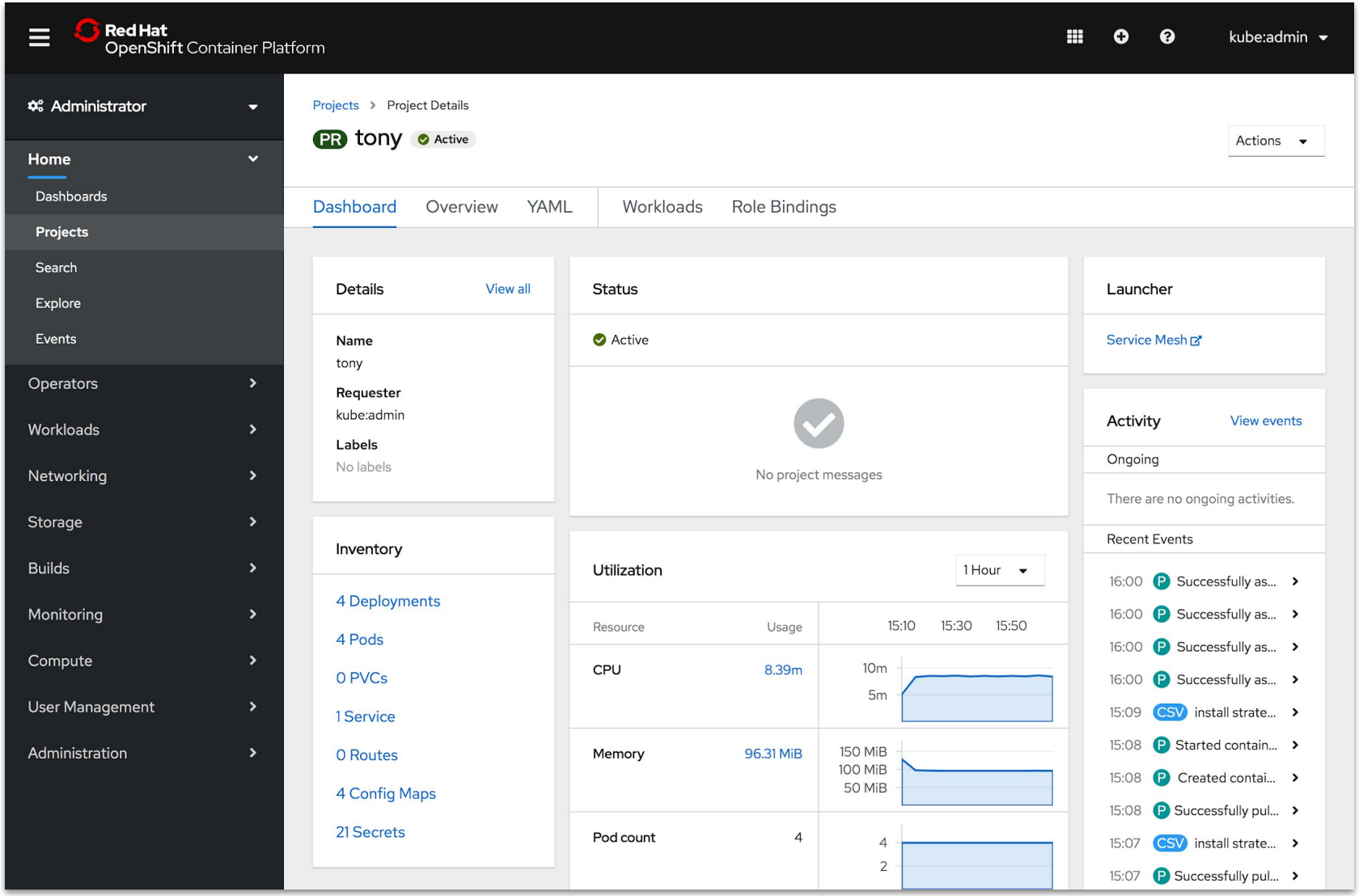
Red Hat

Enhanced Visibility with the New Project Dashboard

Project-scope Dashboard gives Developer Clear Insights

Drill down in context from the new project dashboard widgets:

- Project Details
- Project Status/Health
- Project External Links (Launcher)
- Project Inventory
- Project Utilization
- Project Resource Quota
- Project Activity (Top consumers)



Red Hat

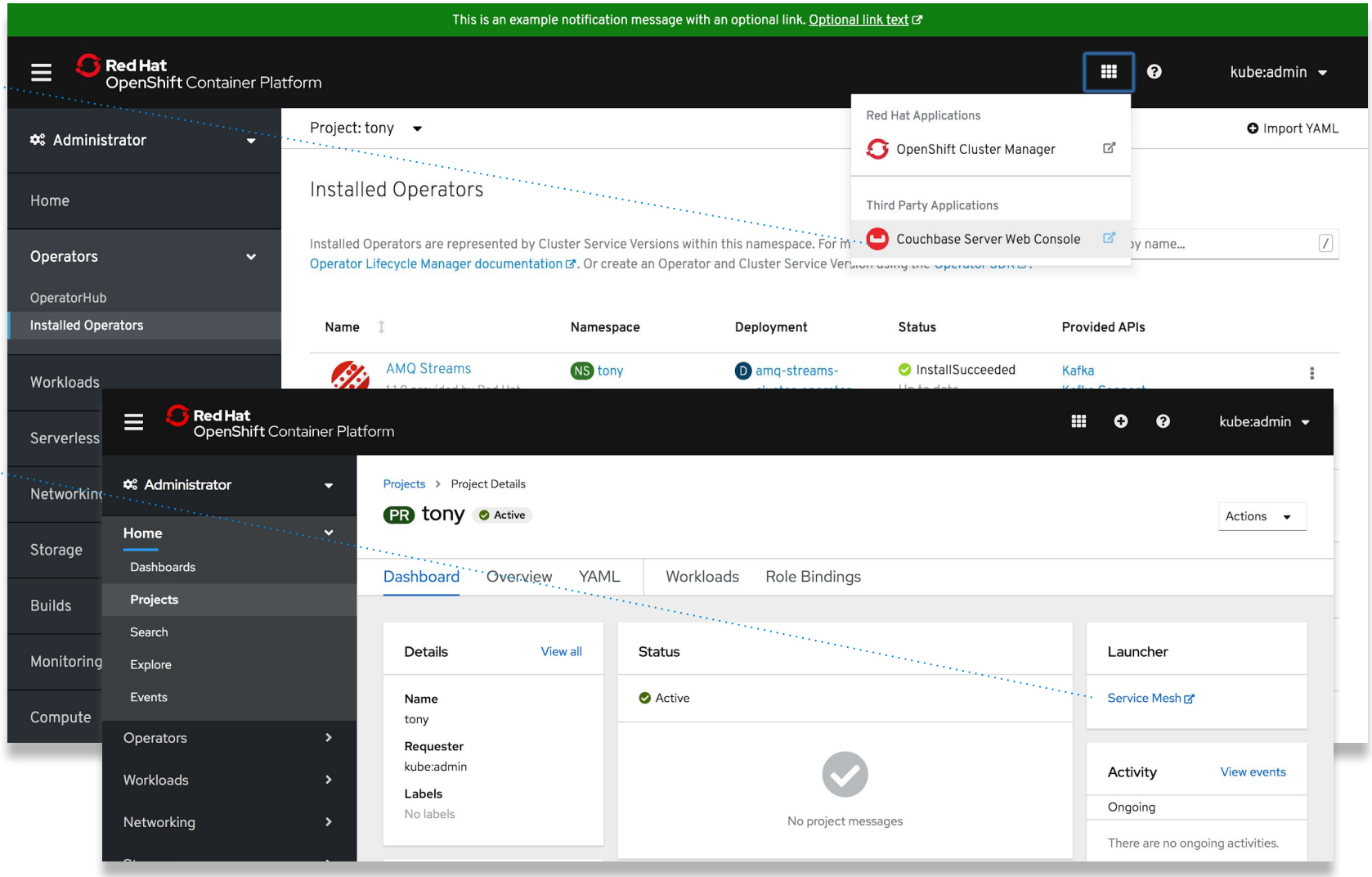
Expose Third Party App Console for Operator-backed Services

"Cluster-wide" ConsoleLink CRD

- Easily integrate/onboard **cluster-wide** third-party user interfaces to develop, administer, and configure Operator-backed services.

"Project-scoped" ConsoleLink CRD

- Customize the access to integrated **project-scoped** third-party user interfaces for your users.
- With the project-scoped external link launch mechanism, **link in context** to your interface.

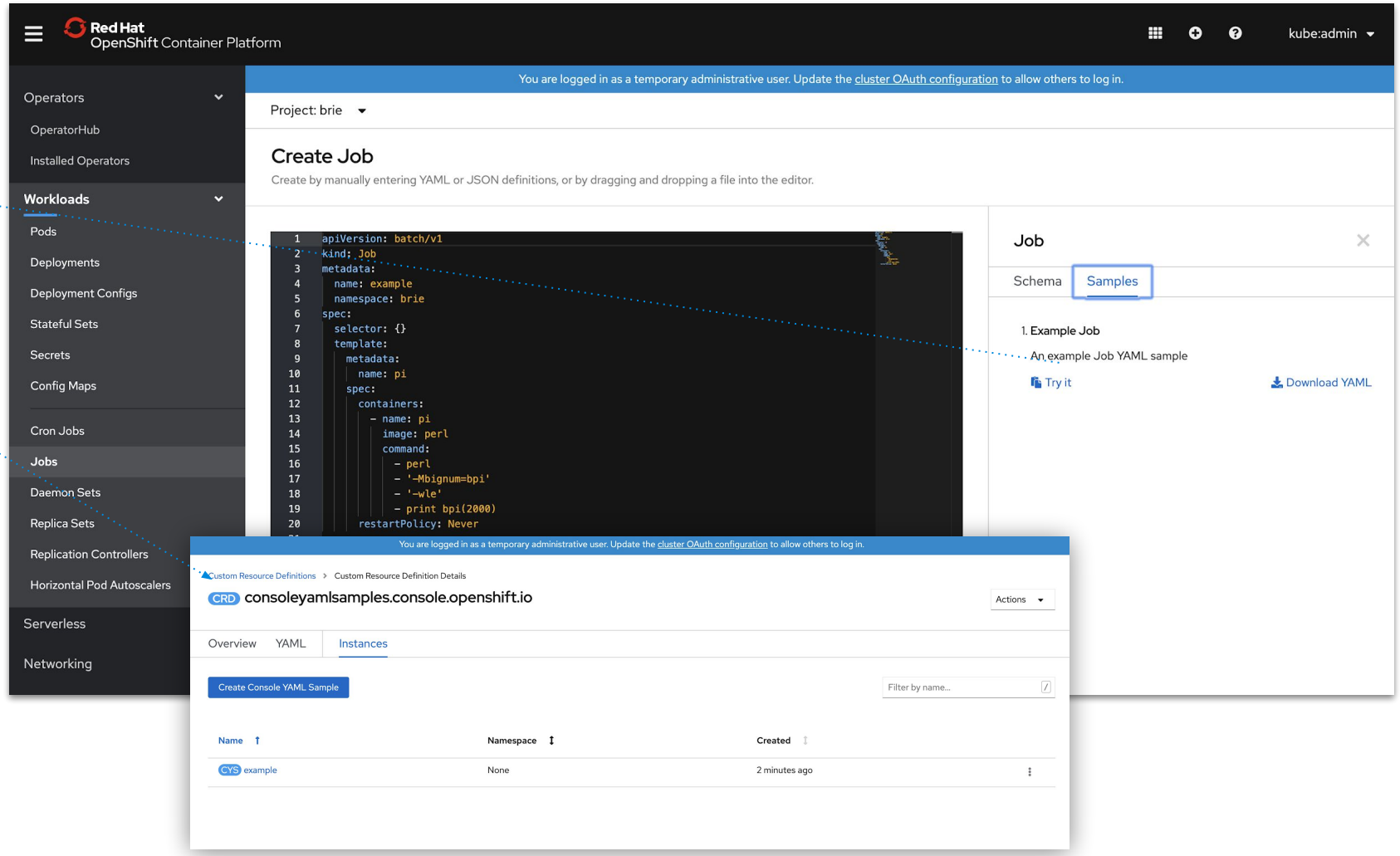


Red Hat

Add YAML Samples for a specific resource

Educate your Users with an Easy Way to Understand Kubernetes Resources

- You can now add cluster-wide samples to any Kube Resource with ***ConsoleYAMLSamples CRD***.
- Each team that manages kube resources owns their samples and should make it part of their Operator.
- Any Operators can add YAML samples including Third-Party ISVs

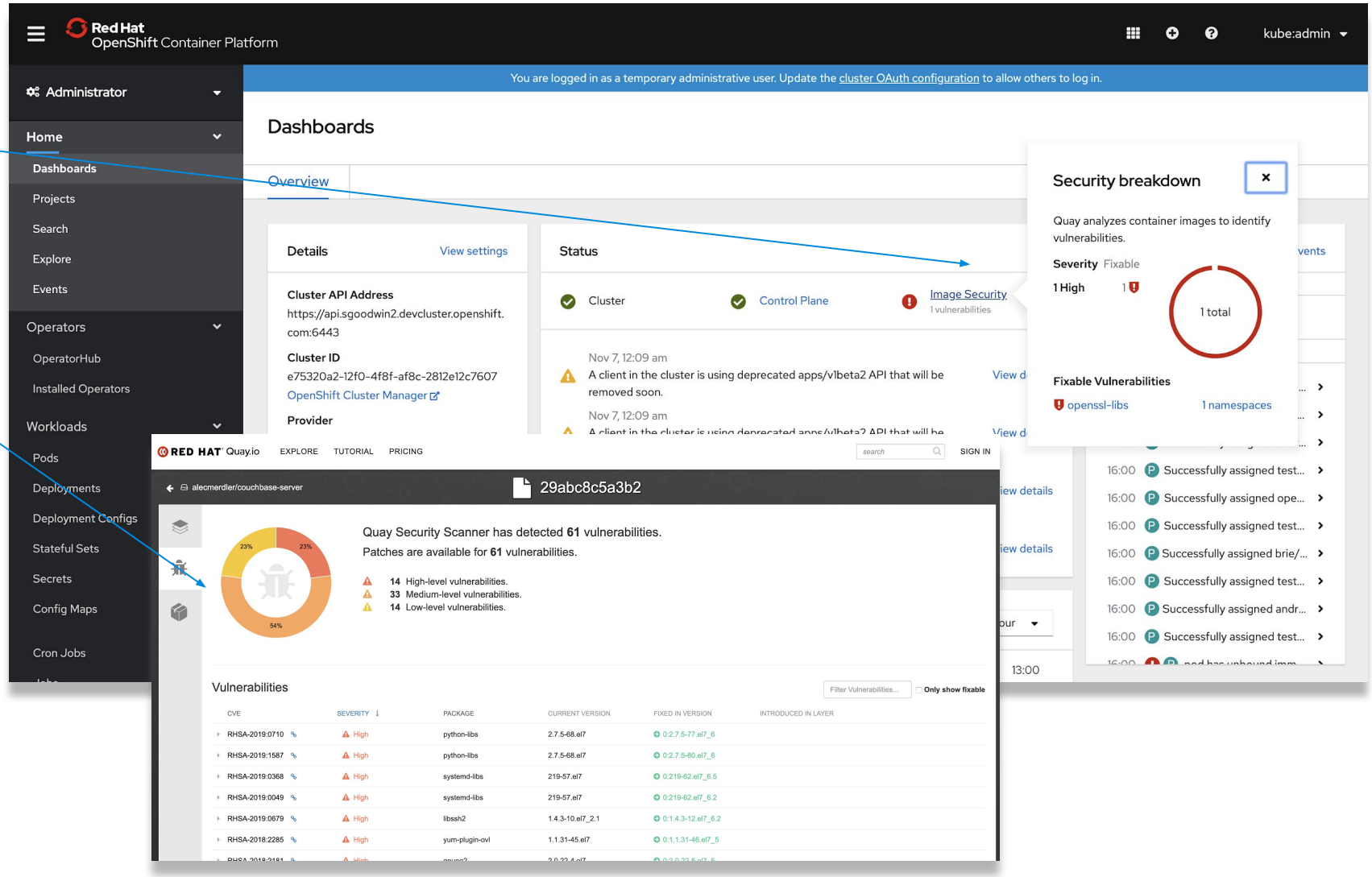


Red Hat

View Security Vulnerabilities with the Quay Operator

See all your Container Vulnerabilities right from the Console Dashboard

- Link out to **Red Hat Quay** for more in depth information
- The Quay Operator supports both **On-premise and External** Quay Registries
- Currently uses **Clair for Security Scan**; Planning to expand to other Vendors(TwistLock, Aqua, e.g.)
- *Only works for images managed by Quay*

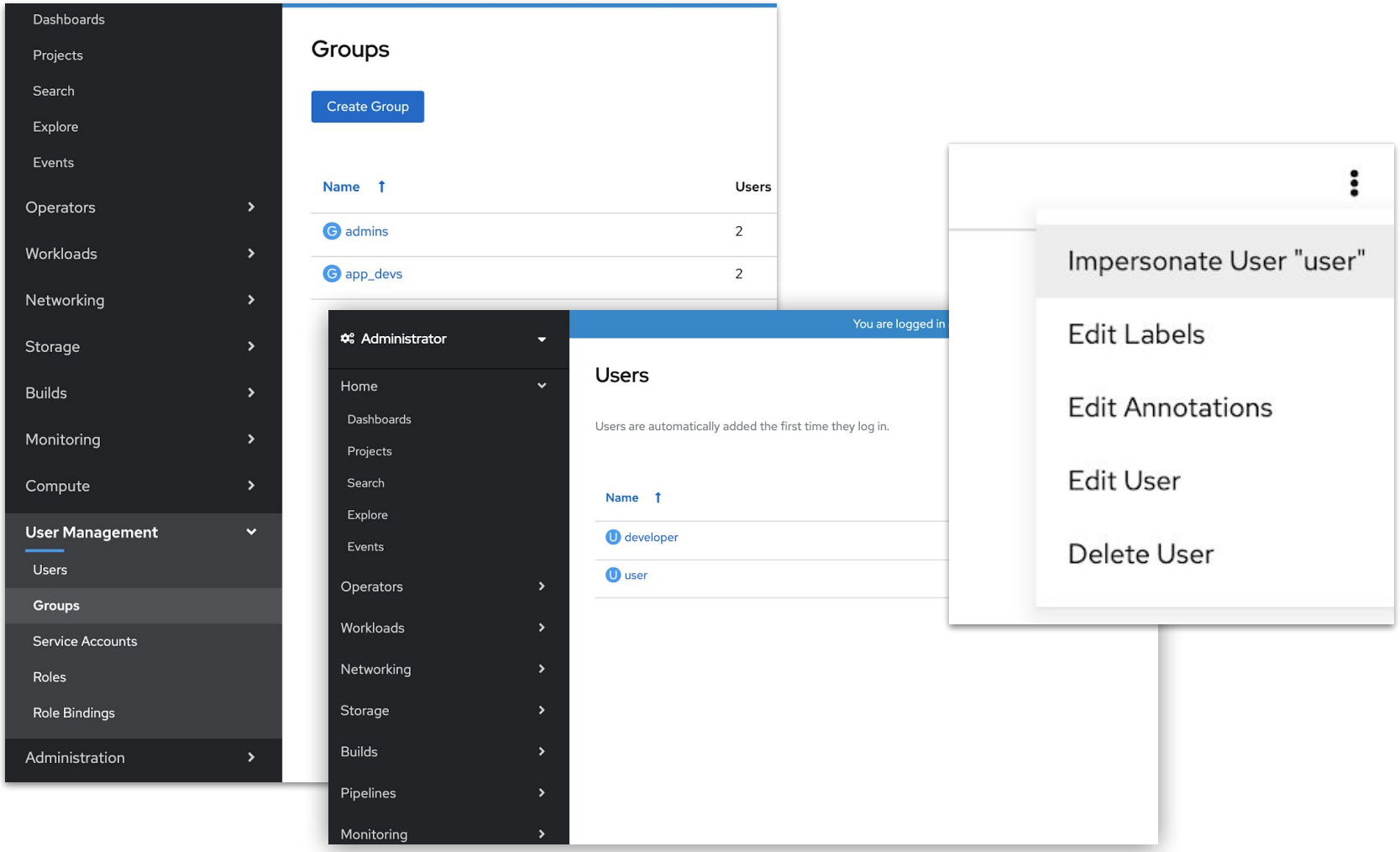


Red Hat

New User Management Section with the Console

Allow cluster admins to easily see who has access to the cluster and how they are organized

1. **All user management** resources under **one navigation section**
2. **Dedicated pages** to view **Users** and **Groups** for the cluster have been added
3. Ability to **impersonate a user;** view exactly what they can see

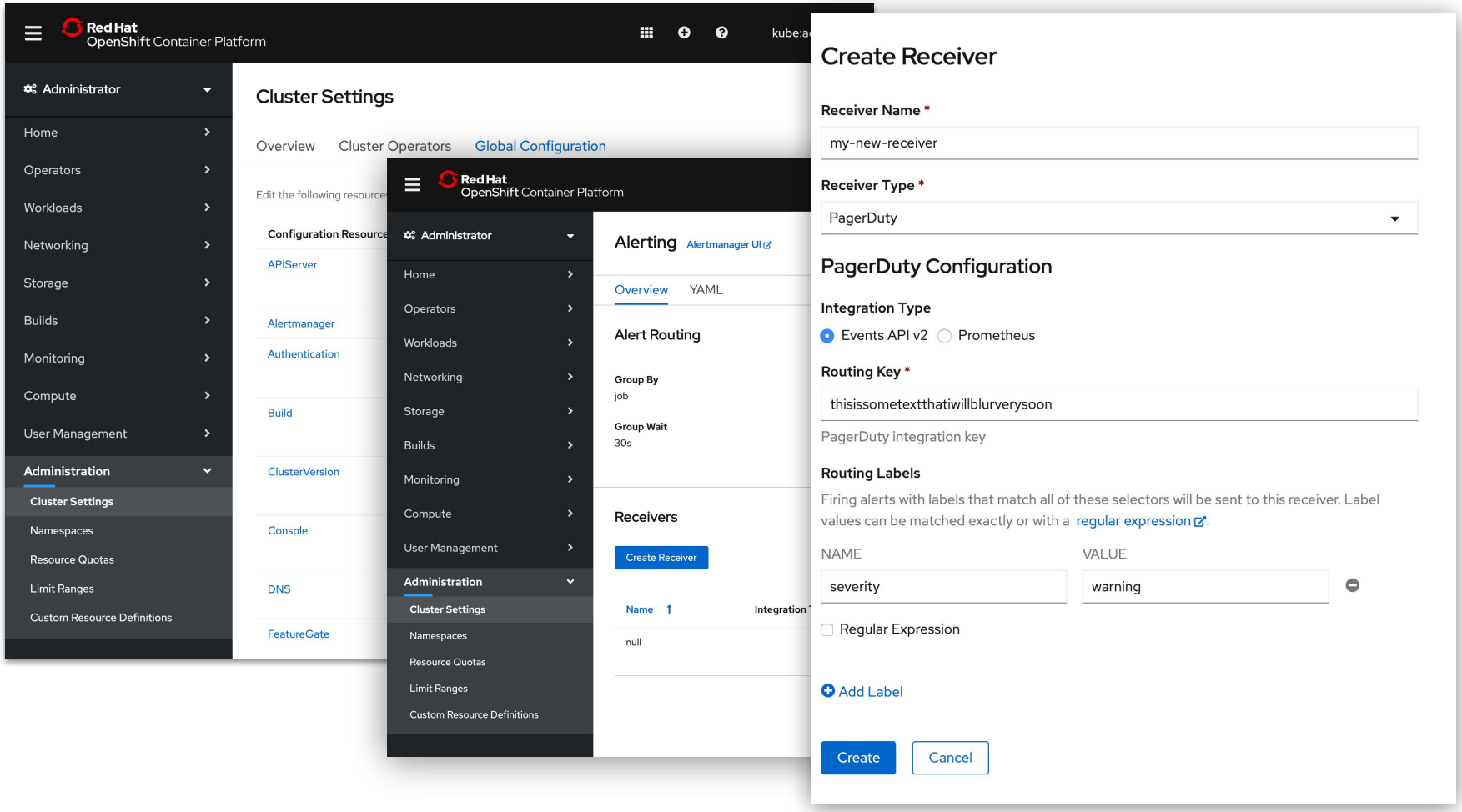


Red Hat

Be Informed with the Alert Receivers

Alerts are only useful if you know about them!

- **Reduce** your **Mean Time To Resolution** (MTTR)
- Create alerts receivers for:
  - **Pager Duty**
  - **Webhooks**
- **More receivers** to come in **future** releases
- Send alerts to the teams that need them; **Reduce the noise** for teams that don't
- Default receiver in place as a **catch all**

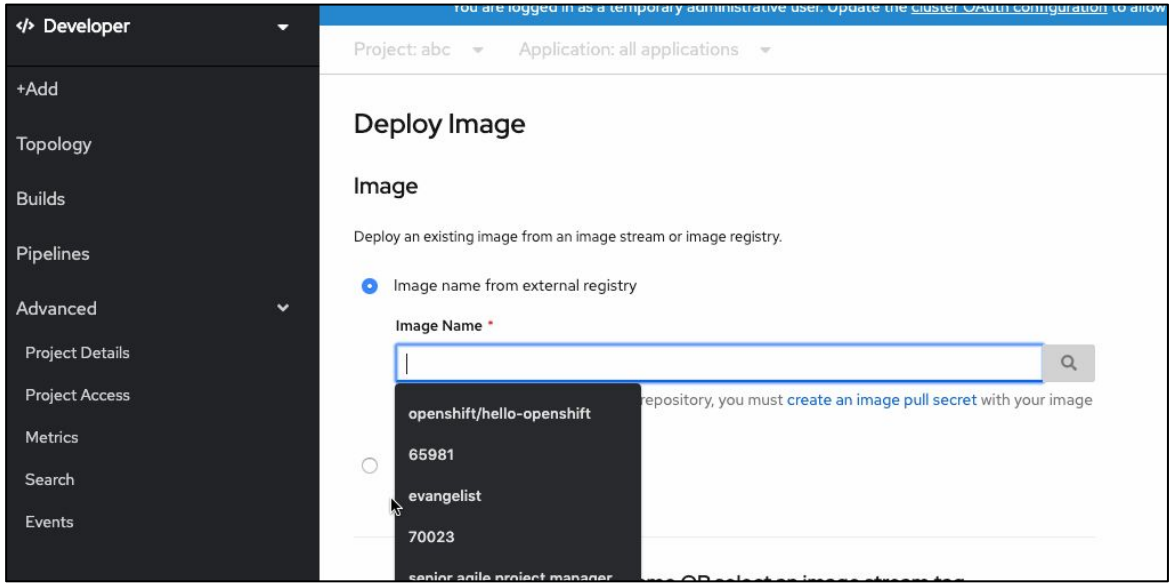


Red Hat

Deploy Applications streamlining flows

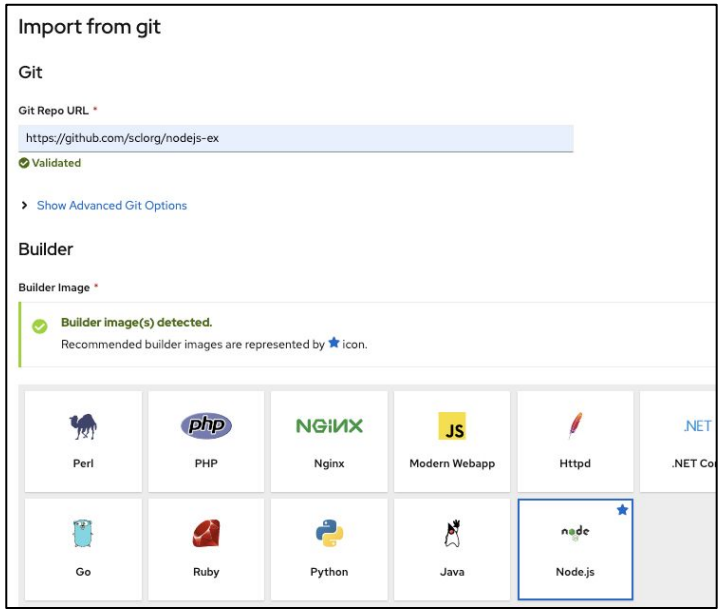
Deploy Image from Internal Registry

- Allow for rapidly deploying with alternate paths
- No need to repush/pull images



Auto-detect builder image

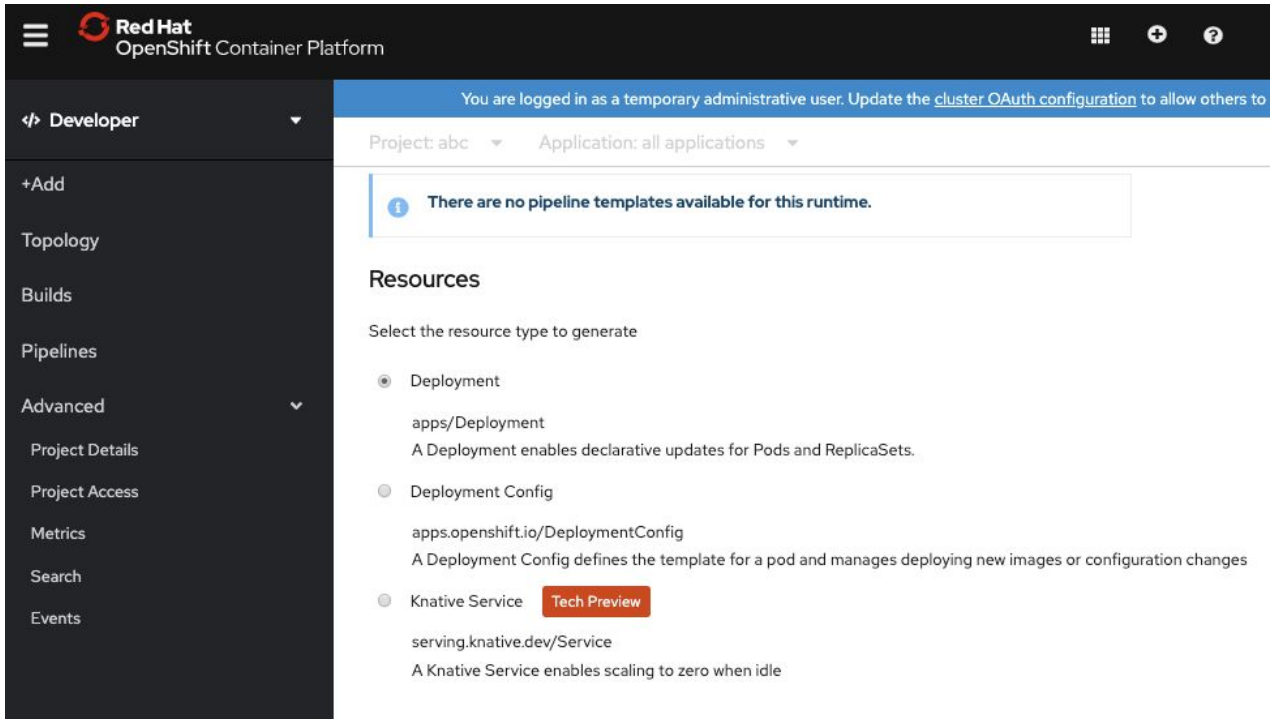
- Recommends builder images based on detected language by git provider



Red Hat

Deploy Applications alternate deployment targets

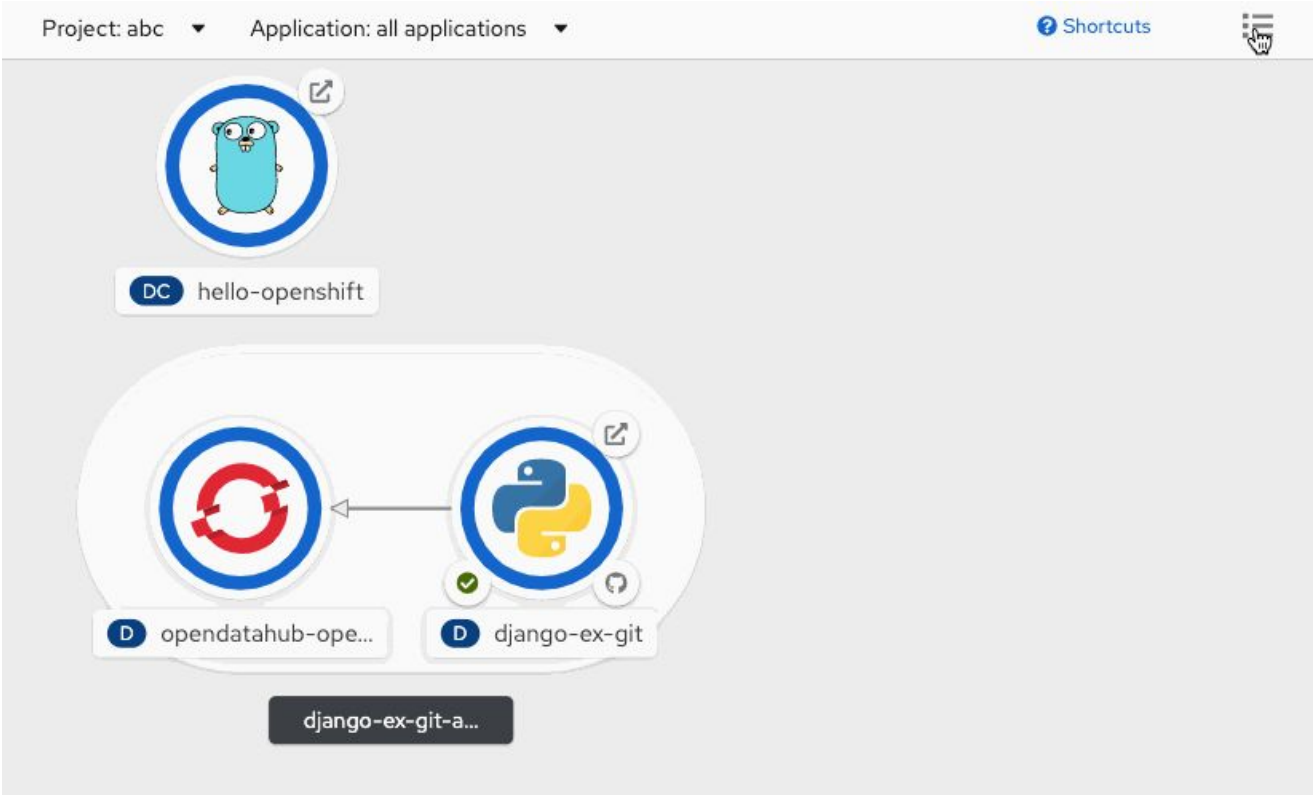
- Default to Kubernetes Deployments
- Alternately can use OpenShift's DeploymentConfigs or Knative Service (tech preview) objects
- Advanced options changes accordingly



Red Hat

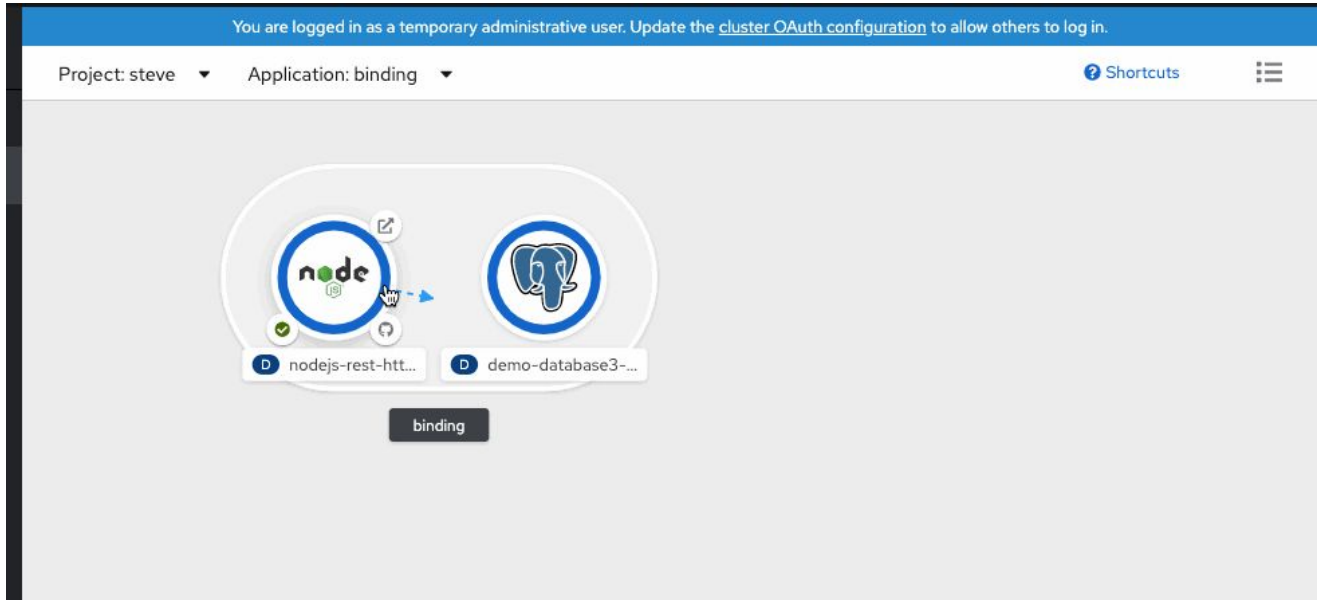
Application Topology streamlined flows

- Toggle between List and Topology views
- Easily group applications
- Connect/bind applications easily
- Contextual actions
- Quickly delete applications



Service Binding easily connecting apps

- Leverages new ServiceBindingRequest and Operator to handle binding requests
- Easily create in Topology by dropping connector to valid drop target
- Injects config into source pod template as environment variables as a secret
- Pods are redeployed to pick up binding credentials



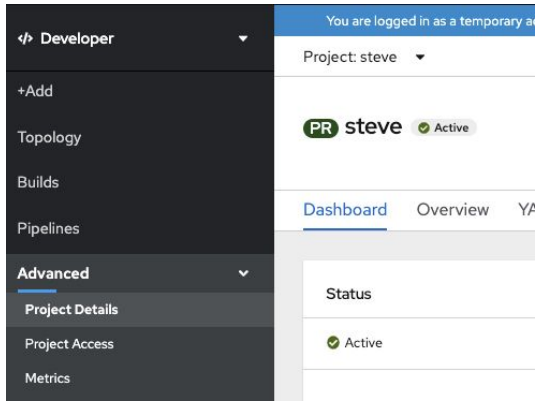
Learn more about service binding:
https://github.com/redhat-developer/service-binding-operator

Red Hat

Project Details & Access

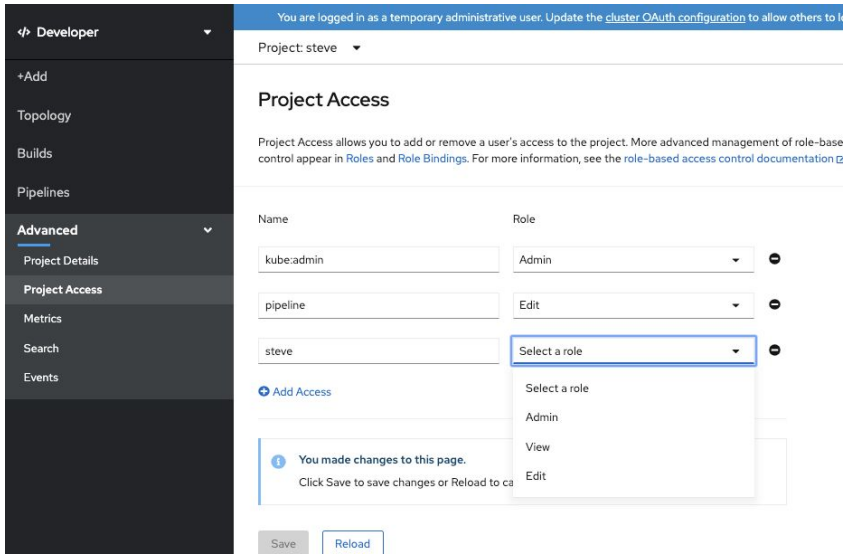
Project Details

- Quick access to current project details
- View dashboard for status and resource utilization
- Actions for edit or delete



Project Access

- Simplify sharing projects
- Reduces to a simple set of Roles that developer frequently use

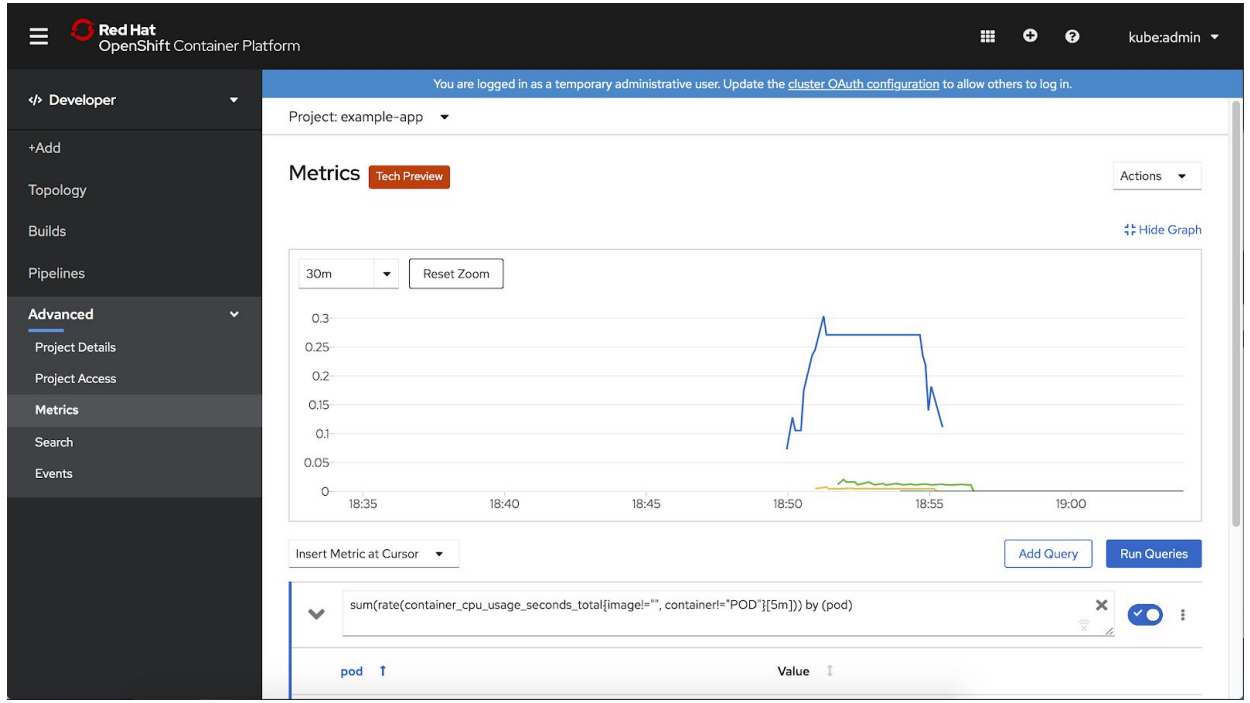


Red Hat

Metrics

Quick access to key application metrics

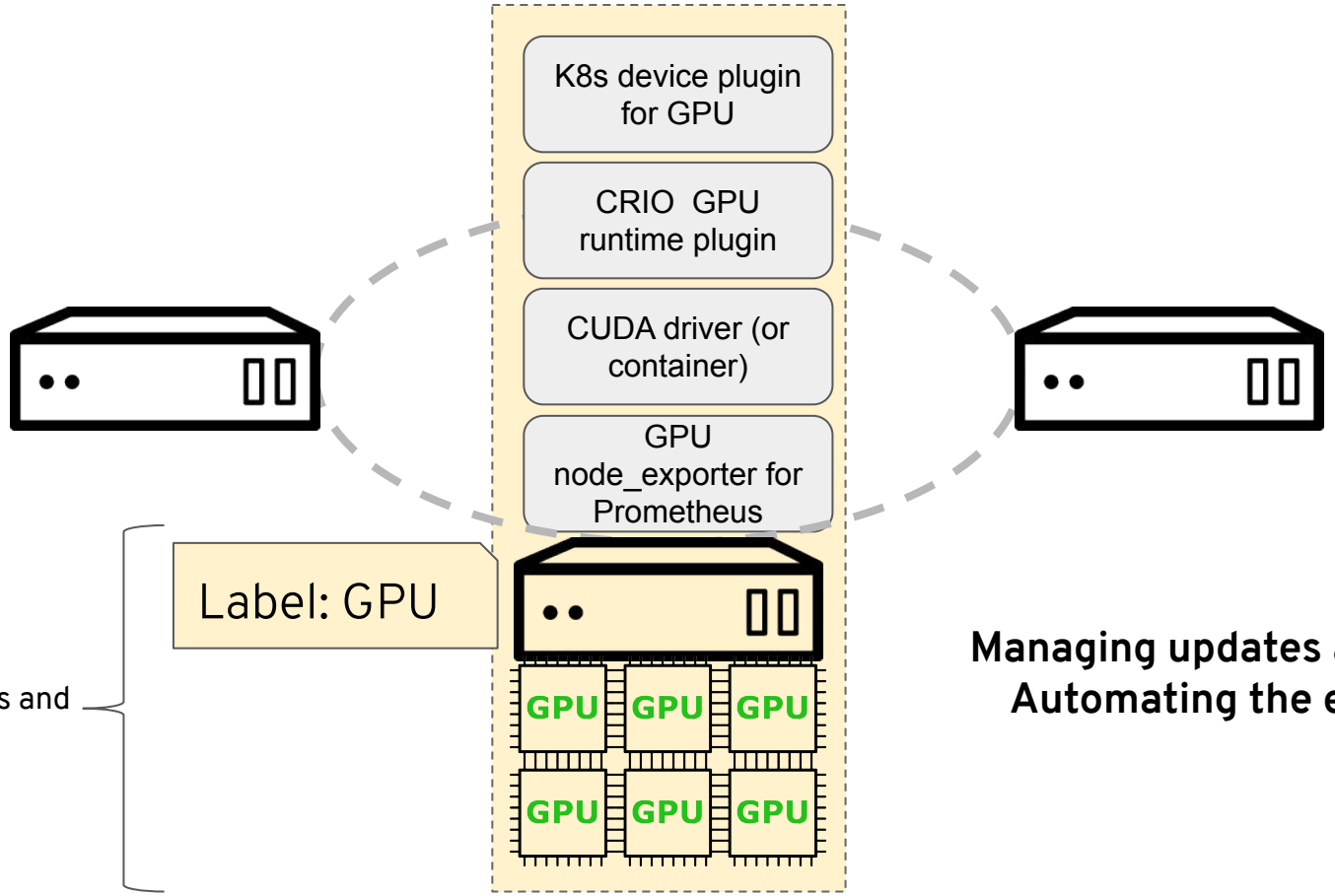
- Use of Prometheus Query Language
- Easily build up queries and plot to visualize application and component trends



Red Hat

Compute

Automating the enablement of GPUs in an OpenShift Cluster



1

NFD Operator recognizes GPUs and labels the nodes

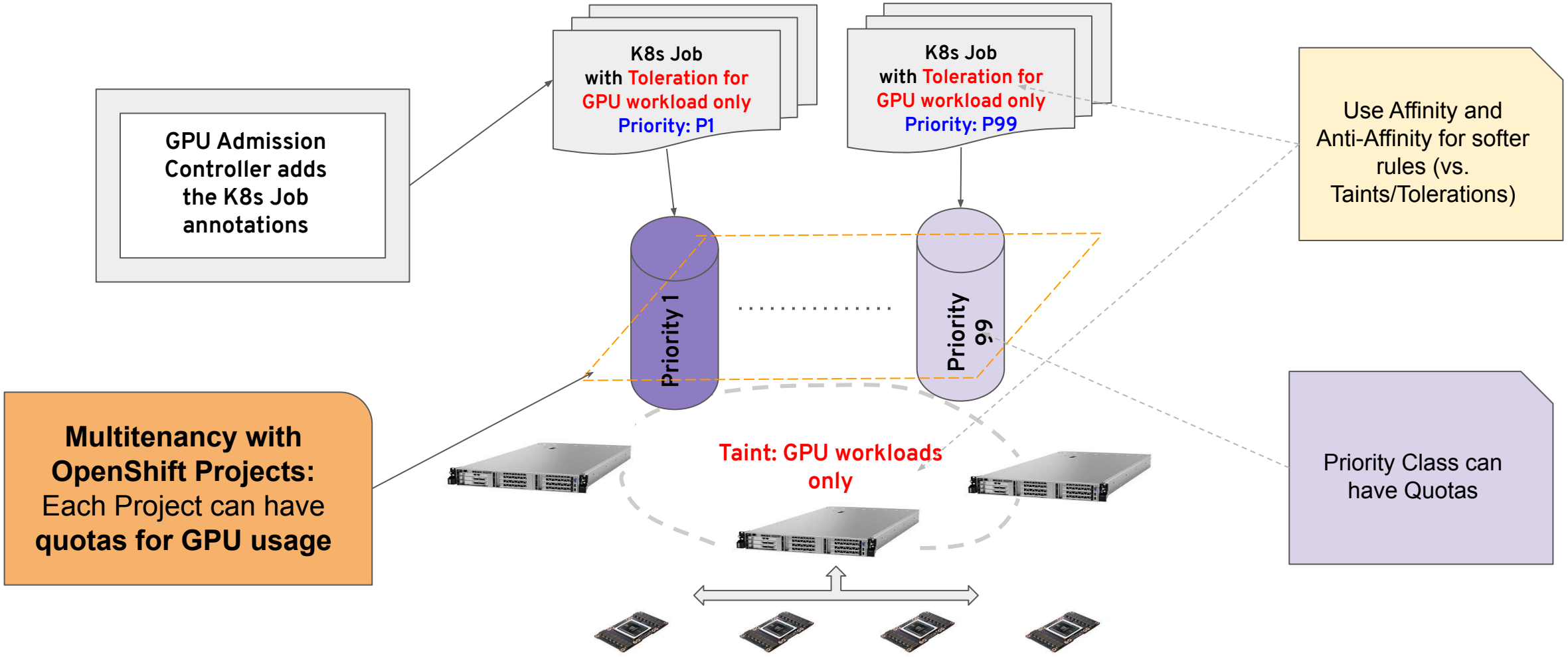
2

Special Resources Operator builds the GPU enablement stack

Managing updates and lifecycle with operators.
Automating the experience with operators.

Red Hat

GPUs in OpenShift



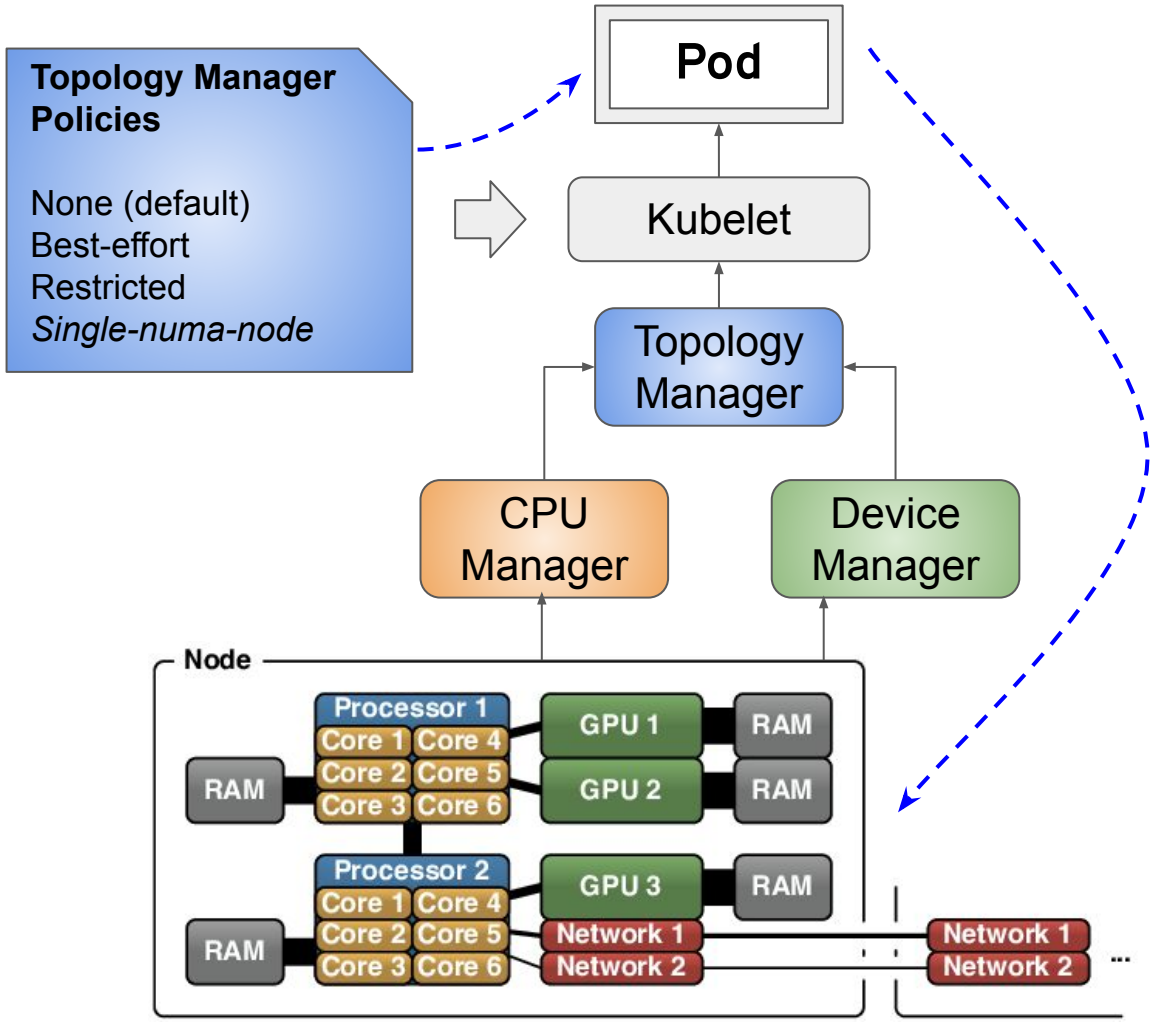
Node Topology Manager

Description

- Performs optimizations related to CPU isolation and memory and device locality
- Rationalizes the (independent) decisions of CPU Manager and Device Manager on the node

Benefits

- Optimize the CPU, Memory and Device assignments for specific workloads
- Better handle High Performance and Low Latency applications



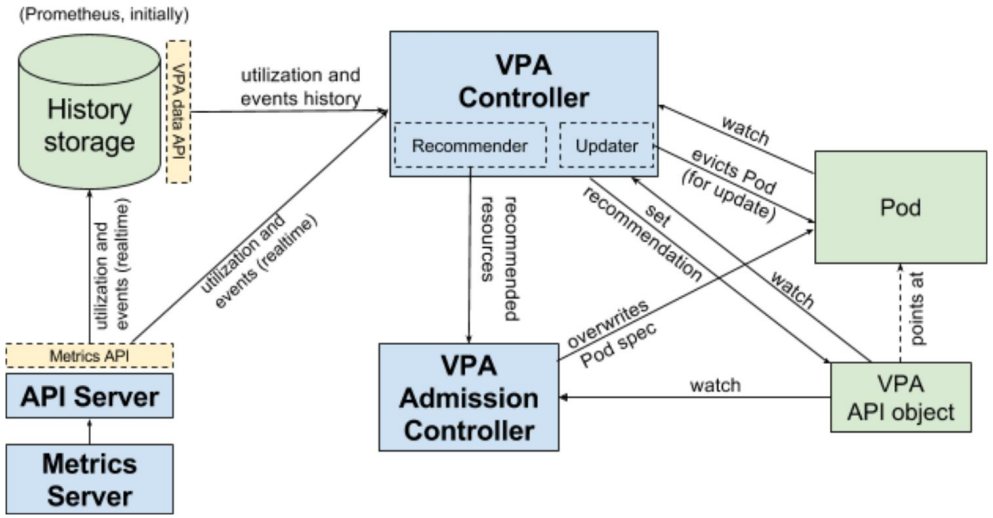
Vertical Pod Autoscaler

Description

- Recommends values for CPU and memory requests based on historical trends.
- OpenShift Users do not have to think about what values to specify for CPU and memory requests

Benefits

- Better node efficiency- pods use what they need
- Reduces pod starvation - Pods are scheduled onto nodes that have the appropriate resources available
- Better cluster utilization
- Self-maintaining - The autoscaler can adjust CPU and memory requests over time.



Red Hat

Networking

Networking Themes

Stability and Security

Operators
Traffic Isolation
Metrics, Alerting, Telemetry
Security Policy API Enhancements

Performance and Scale

SR-IOV
High-Performance Multicast
RDMA, GPUDirect
Multi-Cluster

New and Flexible Features

Multus Plug-ins
IPAM, IPv6, External DNS
Multi-Network
Platform Native Support

Telco Enablement

Foundational Capabilities
CNF Onboarding
Host features (e.g. PTP)
Platform integrations (e.g. OSP)

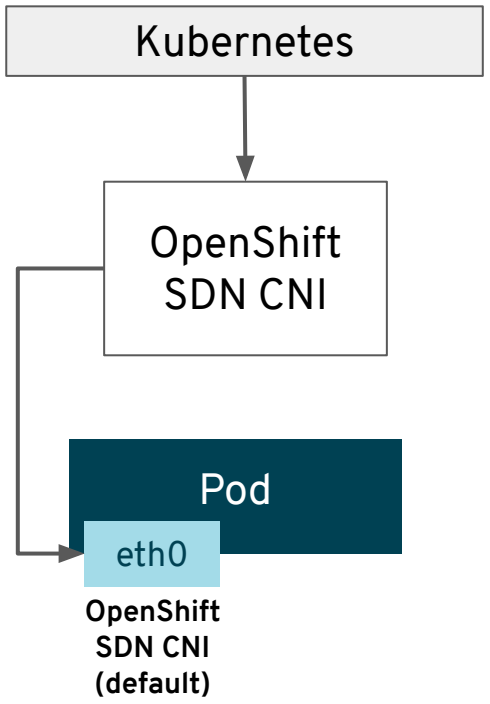
Capability-Building Feature: Networking Plugins

Multus Enables Multiple Networks & New Functionality to Existing Networking

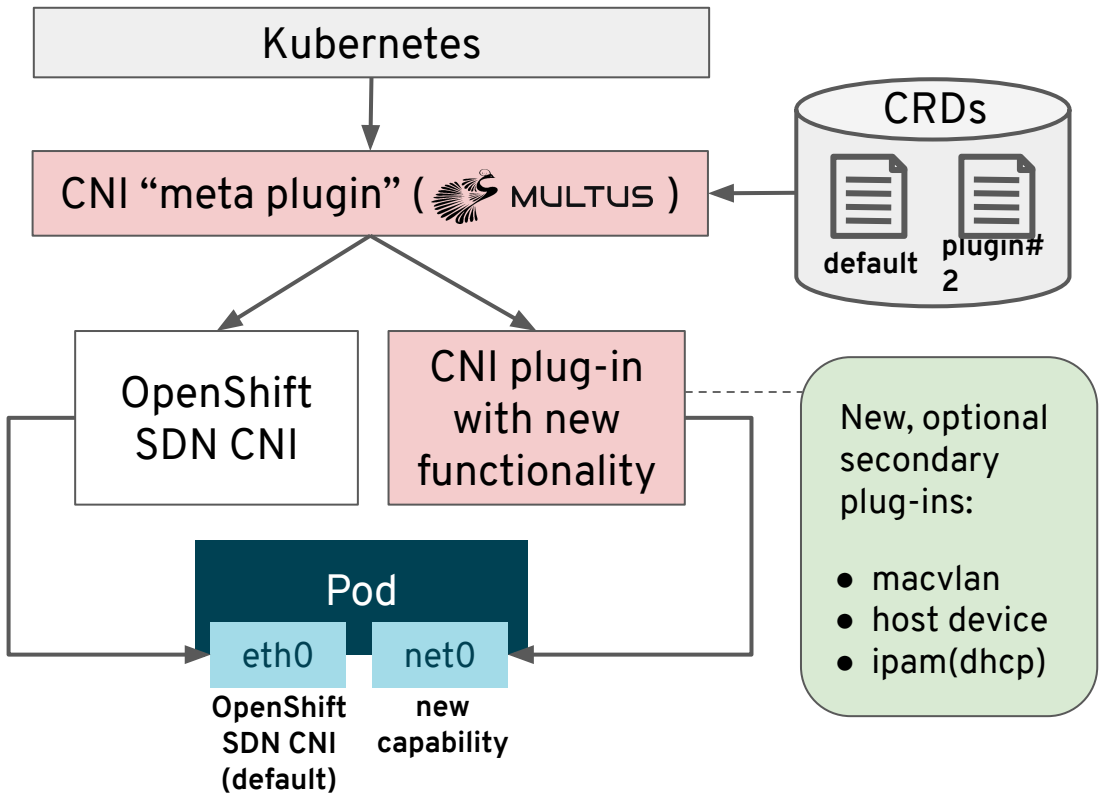
The Multus CNI "meta plugin" for Kubernetes enables one to create multiple network interfaces per pod, and assign a CNI plugin to each interface created.

1. Create pod annotation(s) to call out a list of intended network attachments...
2. ...each pointing to CNI network configurations packed inside CRD objects

3.x Capability...



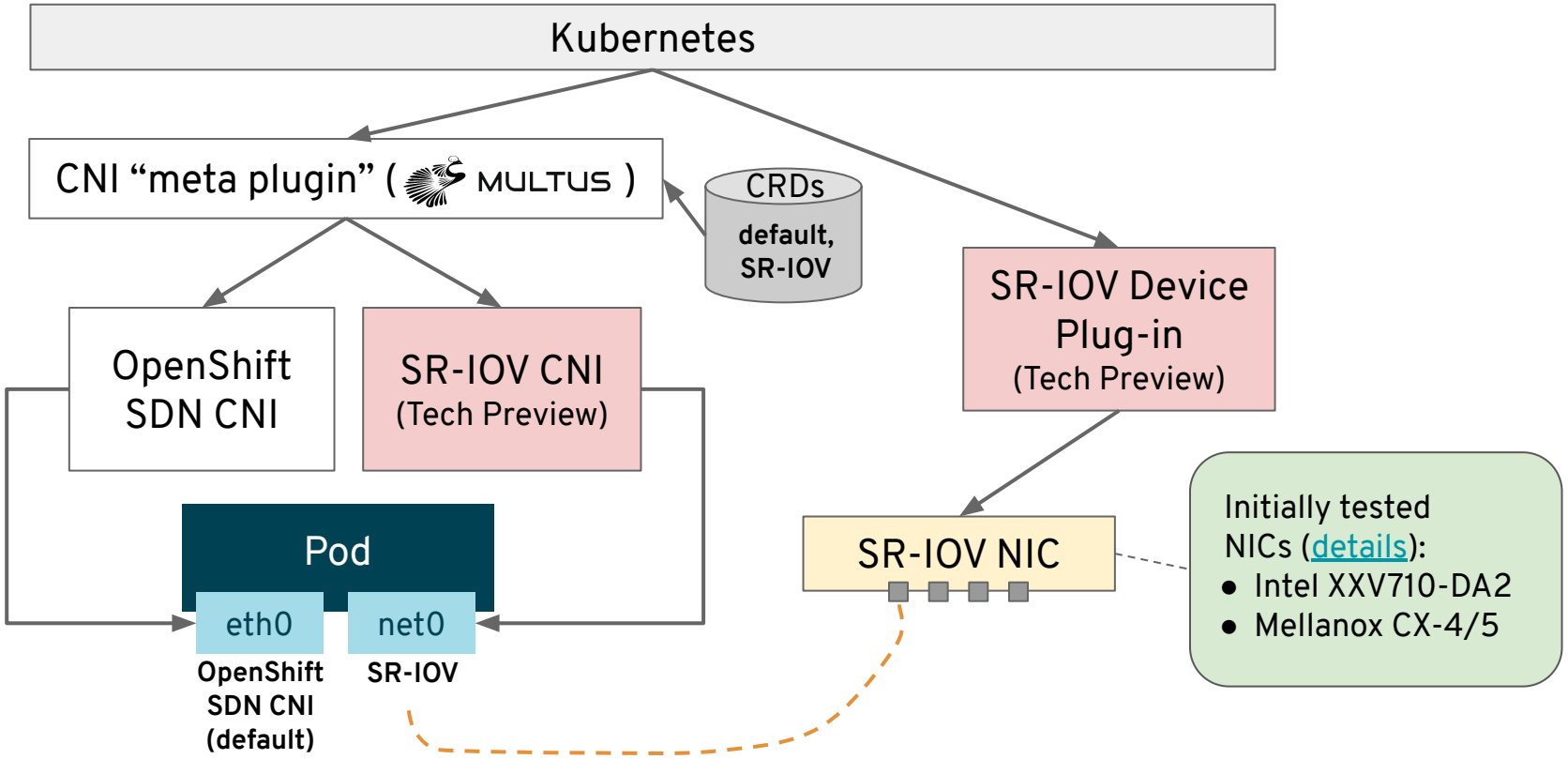
4.x Capability...



High-Performance Networking

SR-IOV Solution:

- Tech Preview since Jun 2019
- Fully Supported Dec 2019
- CNI Plug-In
- Device Plug-In
- RDMA / RoCE Support
- DPDK Mode for SR-IOV VFs
- Admission Controller
- Operator
- VF Security & QoS Flags

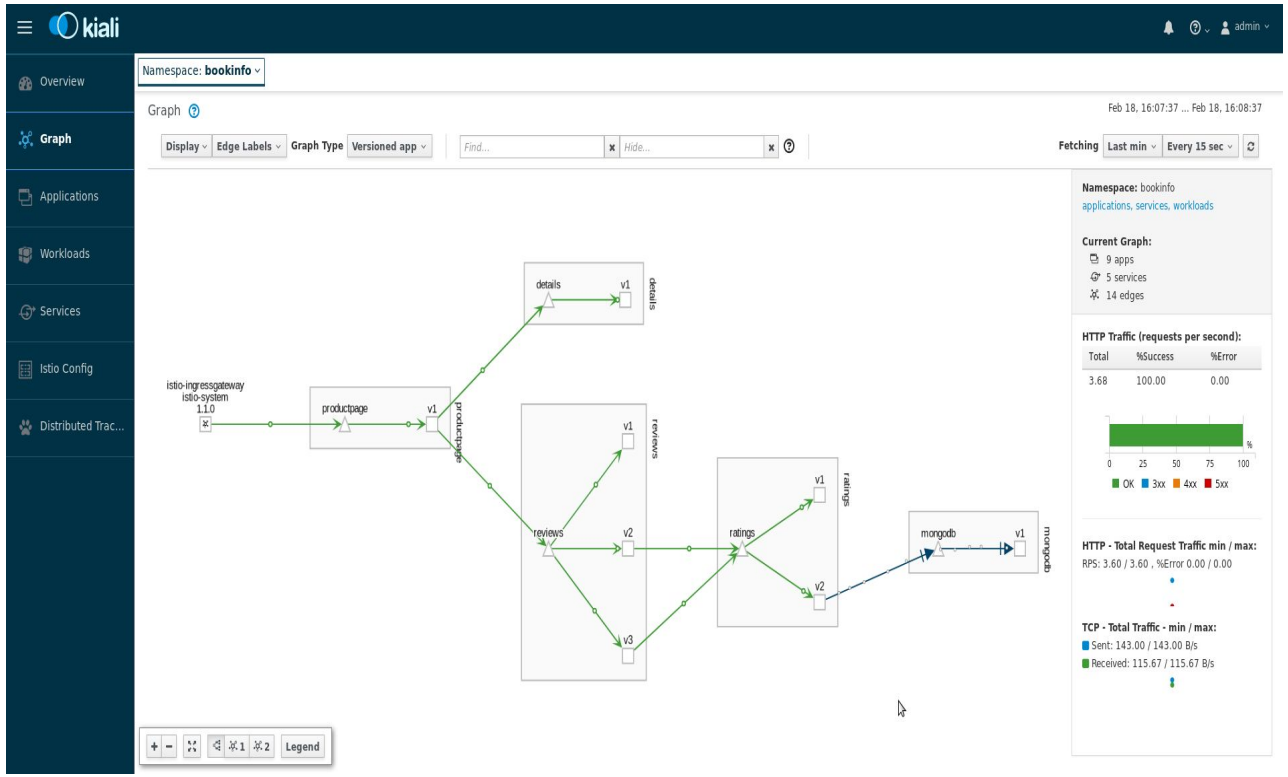


Red Hat

OpenShift Service Mesh

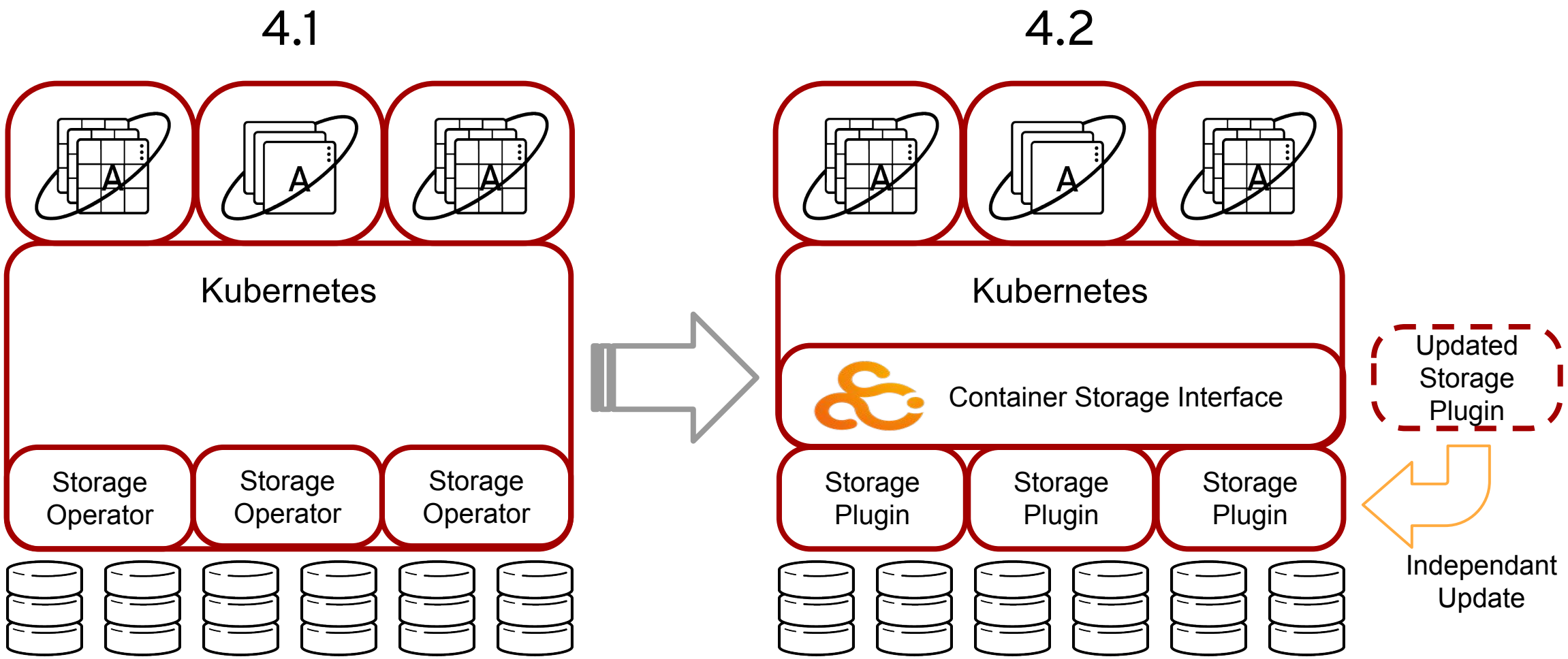
Key Features & Updates

- Version 1.1 coming mid-February
- Upgrade Istio to version 1.4
- Direct links from OCP Console
- Labeled HAProxy routes into the mesh
- Kiali has been updated to Patternfly4
- Jaeger streaming support via Kafka
- Allow Jaeger to be used with an external Elasticsearch instance



Storage

OpenShift CSI



Red Hat

Storage

Storage Focus

- Cluster Storage Operator
  - Sets up the default storage class
  - Looks through cloud provider and sets up the correct storage class
- Drivers themselves remain in-tree for now
- Focus has been on RHEL7 and RHEL CoreOS validating:
  - AWS EBS
  - vSphere Default Storage Class
- New GA storage in 4.2
  - Local Volume
  - Raw Block
    - Cloud providers
    - Local Volume

Supported	Dev Preview
AWS EBS	CSI Snapshot NEW
VMware vSphere Disk	CSI Clone NEW
NFS	CSI Resize NEW
iSCSI	EFS
Fibre Channel	
HostPath	
Local Volume NEW	
Raw Block NEW	

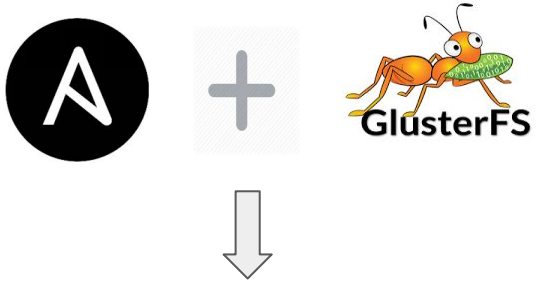
Red Hat

OCS 4.2: Change in Technology Stack

Goal to have complete storage for OCP whatever the needs

- **Need for scalable S3 object stack (**New apps, infra like chargeback, metering)
- **Red Hat Ceph is** scalable object stack with block and file
- **Recently acquired Noobaa -** consistent S3 interface over Ceph RGW, AWS S3, Azure Blob; Federation & multi-cloud capable
- **Rook** operator framework for simple install, manage, expand

OCS 3



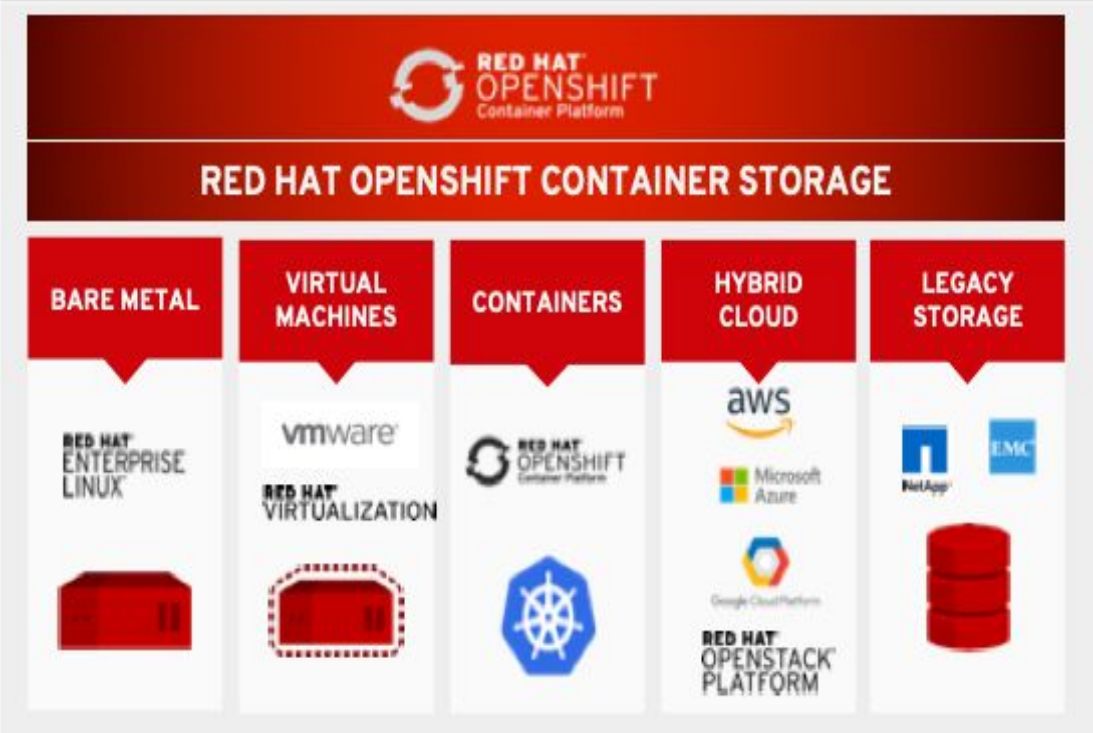
OCS 4

- **No change in OCS SKU or pricing**
- **Full integrated migration support from OCP +OCS 3 to OCP + OCS 4**

OpenShift Container Storage 4.2

Persistent data services for OCP Hybrid Cloud

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



OCS 4.2 support with OCP 4.2

- **Platform support**: AWS and VMware
- **Convergred Mode support** : Run as a service on OCP Cluster
- Hybrid and multi cloud S3 (Tech Preview)

OCS 4.3

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

Red Hat

Security

Security Themes

Control Application Security

Connect workload identity to Cloud provider authorization

Application certificate lifecycle management

Defend the Infrastructure

Encrypt etcd datastore

Enhanced certificate management

RHEL CoreOS disk encryption

VPN / VPC support

Consume group membership from Identity Provider

External Keycloak integration

Automate Compliance

Disconnected / air-gapped install

FIPS compliance

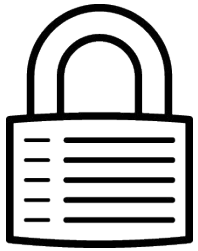
Cipher Suite Configuration

Compliance Operator

Red Hat

Stronger Platform Security

Defense in Depth



CONTROL

Application Security

DEFEND

Infrastructure

EXTEND

- Support for FIPS validated cryptography
- Encrypt etcd datastore
- RHEL CoreOS network bound disk encryption
- Private clusters with existing VPN / VPC
- Internal ingress controller
- Ingress Cipher & TLS Policy Configuration
- Log forwarding (tech preview)

Red Hat

OpenShift 4 and Fips 140-2

FIPS ready Services

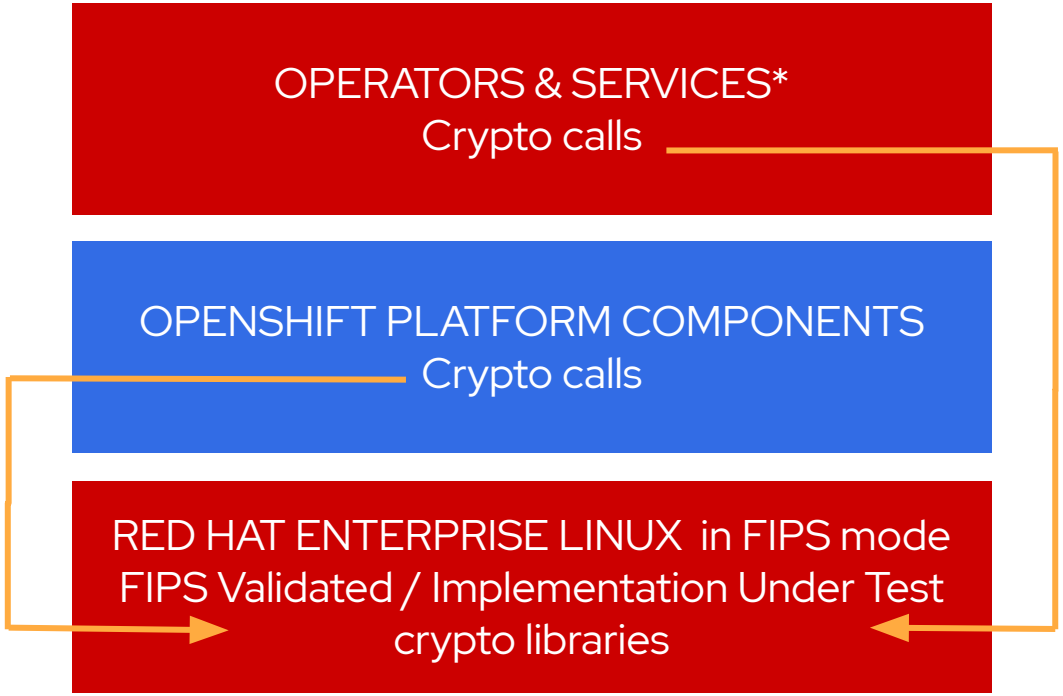
- When built with RHEL 7 base image

OpenShift calls FIPS validated crypto

- When running on RHEL 7.6 in FIPS mode, OpenShift components bypass go cryptographic routines and call into a RHEL FIPS 140-2 validated cryptographic library
- This feature is specific to binaries built with the RHEL go compiler and running on RHEL

RHEL CoreOS FIPS mode

- Configure at install to enforce use of FIPS Implementation Under Test* modules



*When built with RHEL base images

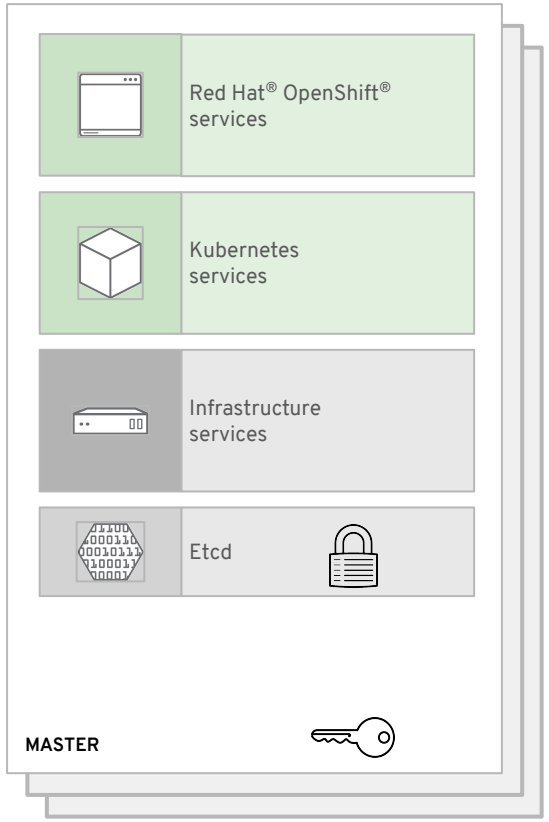
More about RHEL go and FIPS 140-2

Red Hat

OpenShift 4 etcd Encryption

Encrypt secrets, config maps...

- Encryption of the etcd datastore is optional. Once enabled, encryption cannot be disabled.
- The aes-cbc cipher is used.
- Keys are created and automatically rotated by an operator and stored on the master node's file system.
- Keys are available as a secret via the kube API to a cluster admin.
- Assuming a healthy cluster: after enabling encryption, within a day, all relevant items in etcd are encrypted
- Backup: The etcd data store should be backed up separately from the file system with the key.
- Disaster recovery: a backup of both the encrypted etcd data and encryption keys must be available.



Ingress Cipher Suite Configuration

- Allow customers to meet policies requiring them to use specific cipher suites and/or to ensure that disallowed ciphers are not available.
- The Ingress Operator TLSSecurityProfile defines the schema for a TLS security profile. Type is one of Old, Intermediate, or Custom. The Modern profile is currently not supported because it is not yet well adopted by common software libraries
- Cipher suites for the API server will be addressed in a future release

```
// custom is a user-defined TLS security profile. Be extremely careful using a custom
// profile as invalid configurations can be catastrophic. An example custom profile
// looks like this:
//
//   ciphers:
//     - ECDHE-ECDSA-CHACHA20-POLY1305
//     - ECDHE-RSA-CHACHA20-POLY1305
//     - ECDHE-RSA-AES128-GCM-SHA256
//     - ECDHE-ECDSA-AES128-GCM-SHA256
//   minTLSVersion: TLSv1.1
//
// +optional
// +nullable
Custom *CustomTLSProfile `json:"custom,omitempty"`
```

Red Hat

Red Hat OpenShift

Thank You

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat