



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

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

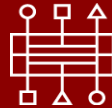
Integrating new capabilities wasn't always simple

Openshift 4 Themes

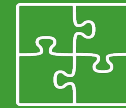
DAY 2
OPERATIONS



IMMUTABLE
INFRASTRUCTURE



OPERATOR
FRAMEWORK



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

Market Expectations for Application Platforms

CLOUD EXPECTATIONS

BROAD APPLICATION SUPPORT

CONTINUOUS SECURITY

ON-DEMAND | AS-A-SERVICE
MARKETPLACES

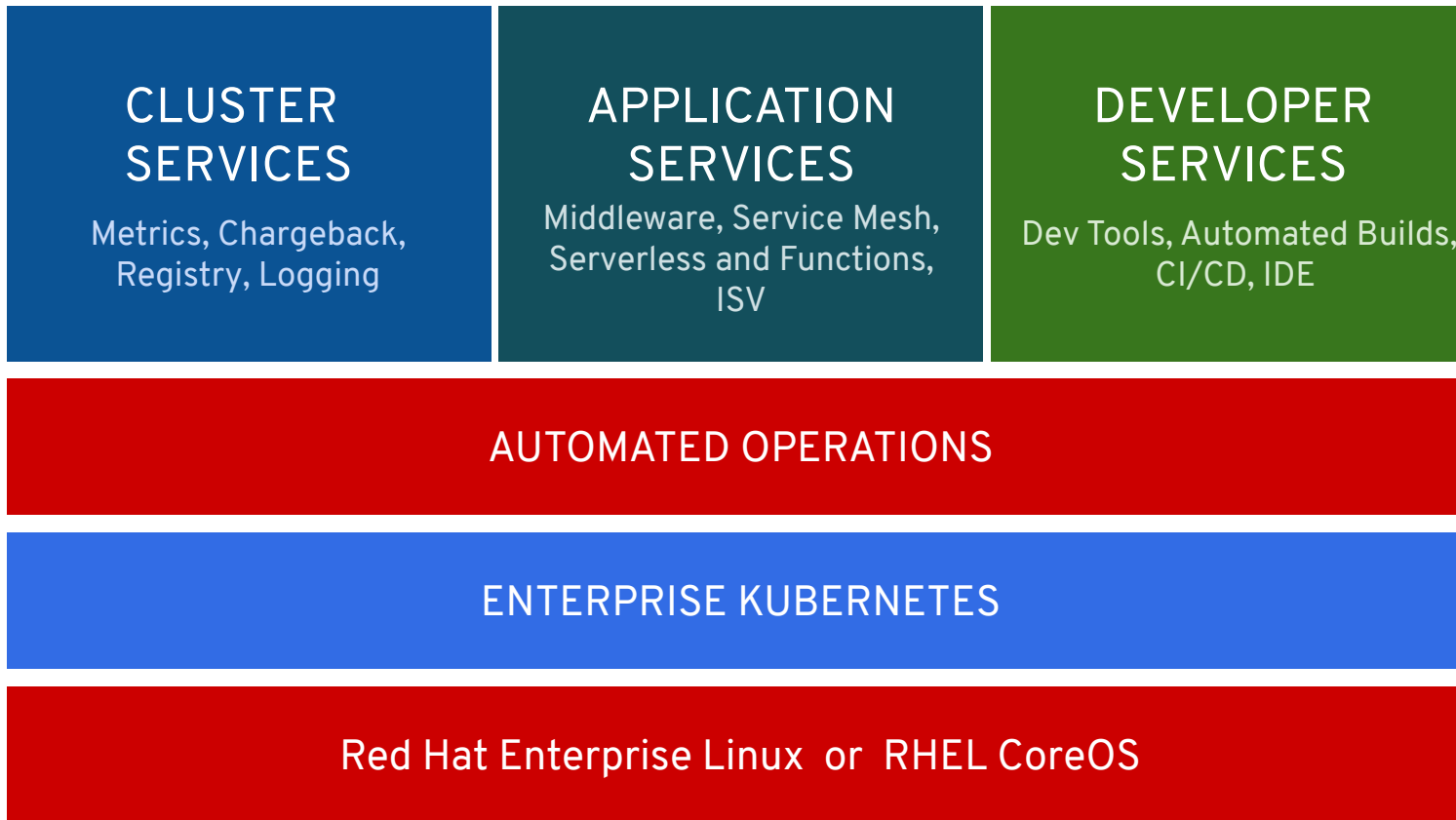
STANDARDS | INTEROPERABILITY |
PORTABILITY | AVOID LOCK-IN

LIMITED OPERATIONS or
MANAGED SERVICES



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

Any infrastructure



Physical



Virtual



Private



Public

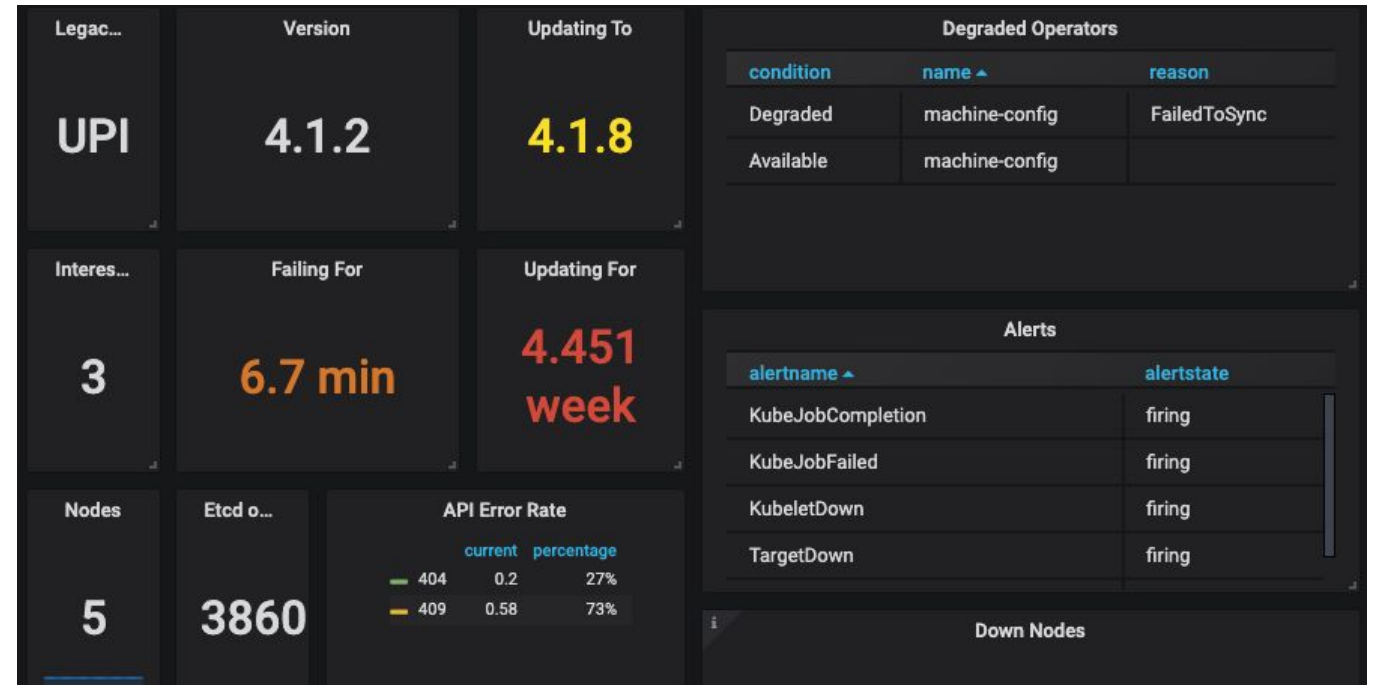
Connected Customer Experience

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



Hosted OpenShift

Get the best of OpenShift without being on call



One Platform, Flexible Consumption Models



Red Hat
OpenShift
Dedicated

Managed service offering on
public cloud



Red Hat Azure
Red Hat
Microsoft OpenShift

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



Red Hat
OpenShift
Container Platform

Enterprise-grade Kubernetes
platform that you manage

HOSTED SERVICES

SELF-MANAGED



Installation and Upgrades

Installation Experiences

OPENSIFT CONTAINER PLATFORM

Full Stack Automation

Simplified opinionated “Best Practices” for cluster provisioning

Fully automated installation and updates including host container OS.



**IPI -
Installer Provisioned
Infrastructure**

Pre-existing Infrastructure

Customer managed resources & infrastructure provisioning

Plug into existing DNS and security boundaries



**UPI -
User Provisioned
Infrastructure**

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.

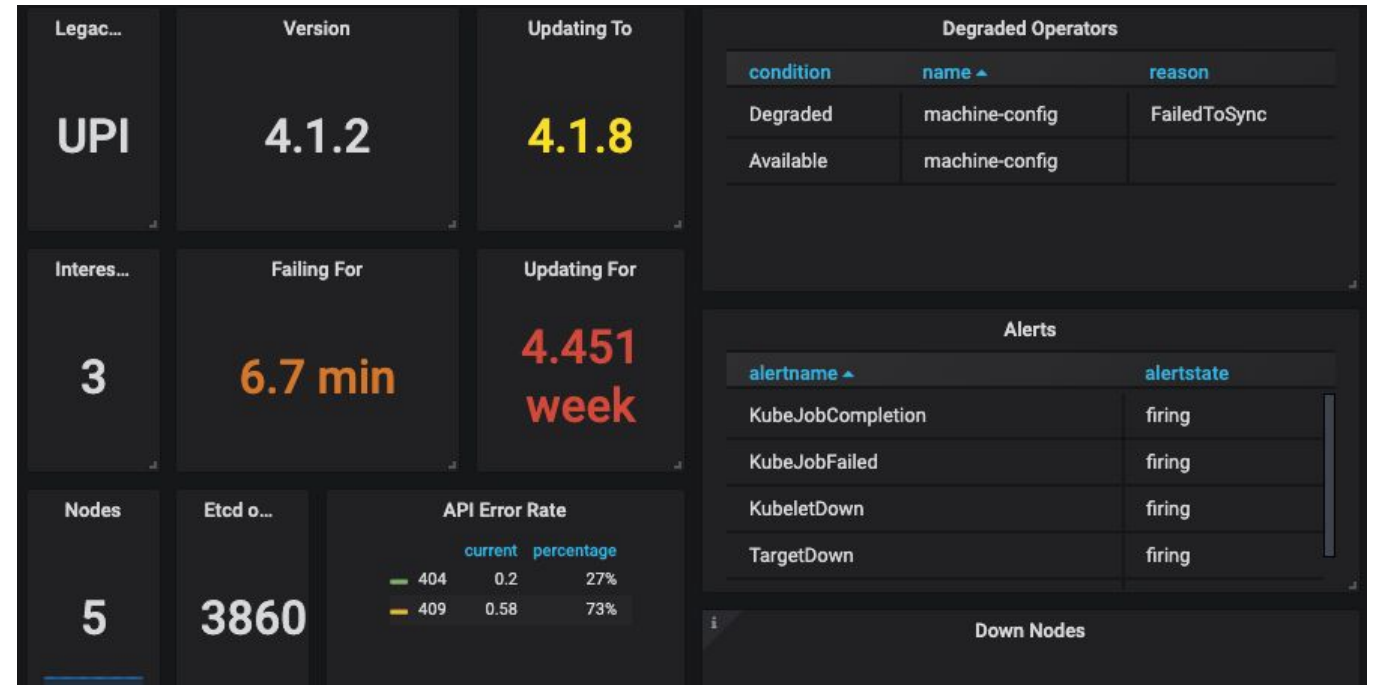
Connected Customer

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



Install OpenShift on Any Cloud

Clusters > Create > OpenShift Container Platform

 **Install OpenShift Container Platform 4**

Select an infrastructure provider



Run on Amazon Web Services



Run on Microsoft Azure



Run on Google Cloud Platform



Run on VMWare vSphere



Run on Bare Metal

IBM Z.
IBM LinuxONE™

Run on IBM Z

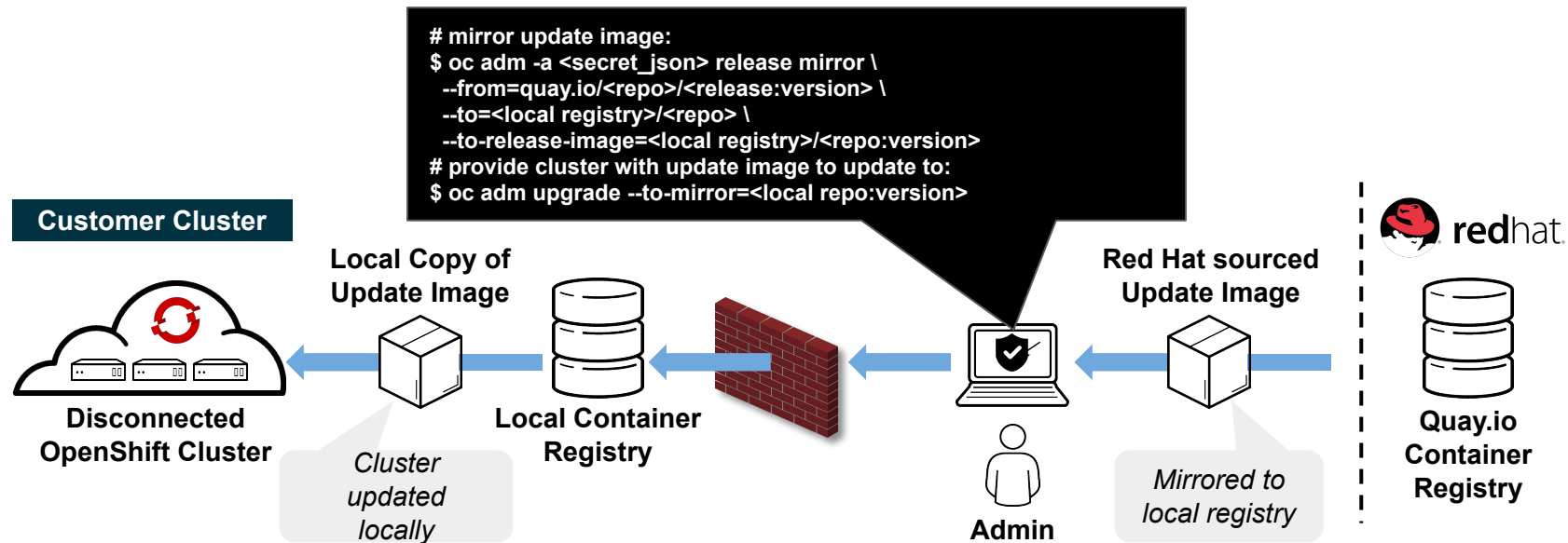


Run on Red Hat OpenStack



Run on Laptop
Powered by Red Hat CodeReady Containers

Disconnected “Air-gapped” Installation & Upgrading



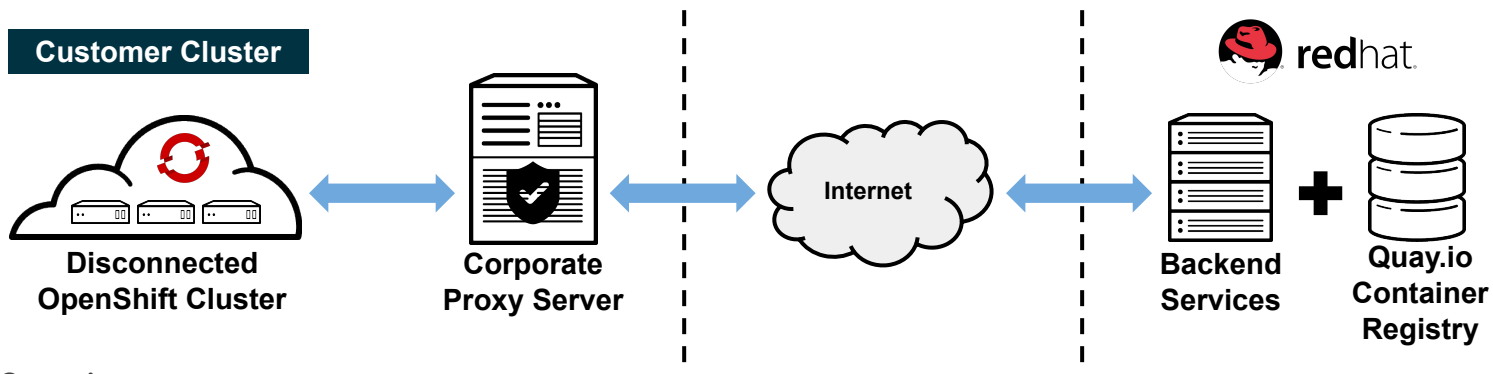
Overview

- 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

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
```

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

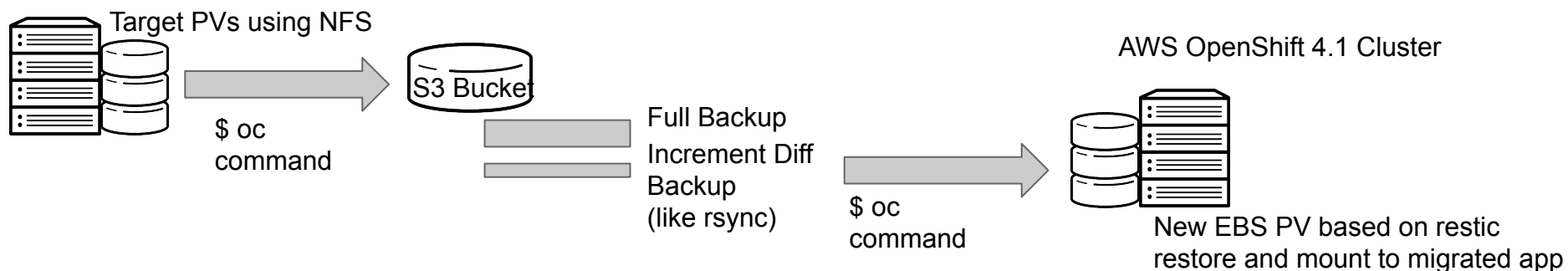
Documentation

- **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

```
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...
```

CLUSTER MIGRATION OPENSIFT 3 to 4

vSphere OpenShift 3.10 Cluster



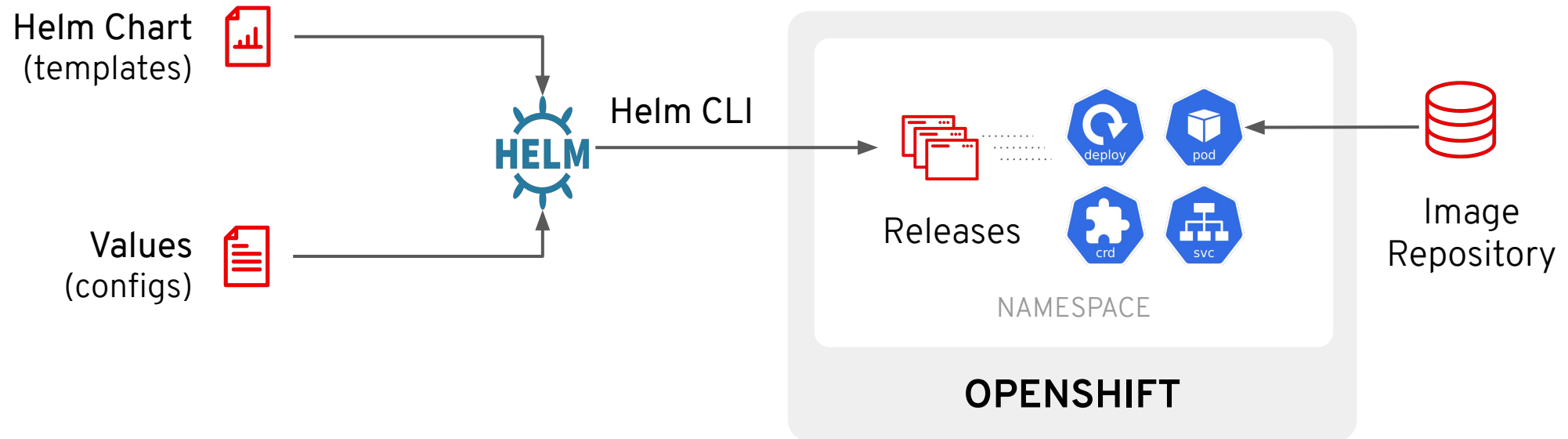
- Deploy a replication of your applications from one OpenShift cluster to a different OpenShift cluster
- Enable cluster specific configuration from OpenShift 3 to work on a OpenShift 4 cluster
- Documentation on how to handle common network, storage, and machine/node re-use scenarios between OpenShift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



Helm Charts

Helm 3 on OpenShift

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



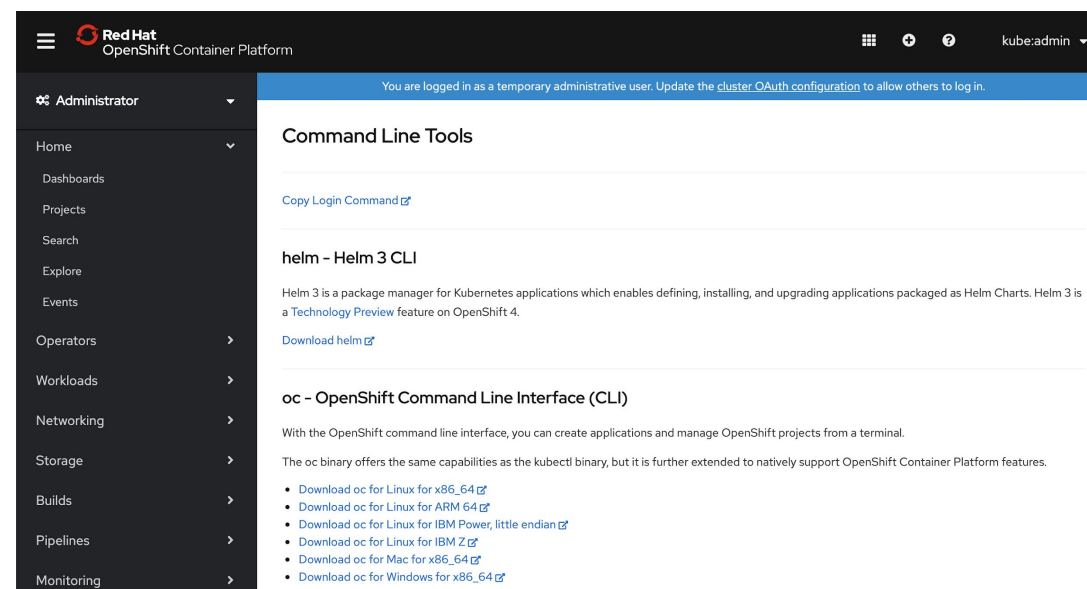
Helm 3 on OpenShift

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



Helm and Operator

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



Day 2 Operations and Cluster Mgm't

Openshift Operators enable ISV Innovation

operatorhub.io

TYPES OF OPERATORS

- Community
- ISV Partners
- Red Hat Products

The image displays two screenshots related to OpenShift Operators. The top screenshot is the OperatorHub.io website, which features a search bar, a 'Contribute' button, and a list of operators categorized by function (e.g., Coordination & Service Discovery, Database, Key Management) and provider (e.g., Auto Pilot, Lifecycle, Installation). The bottom screenshot is the Red Hat OpenShift Operator Hub interface, showing a sidebar with navigation options like Home, Projects, Status, Search, Events, Catalog, and Operator Hub. The main content area displays a list of operators, including 'amq-streams', 'automationbroker', 'cluster-logging', 'couchbase-enterprise', 'descheduler', and 'dynatrace-monitoring', each with a brief description and an 'Installed' status indicator.

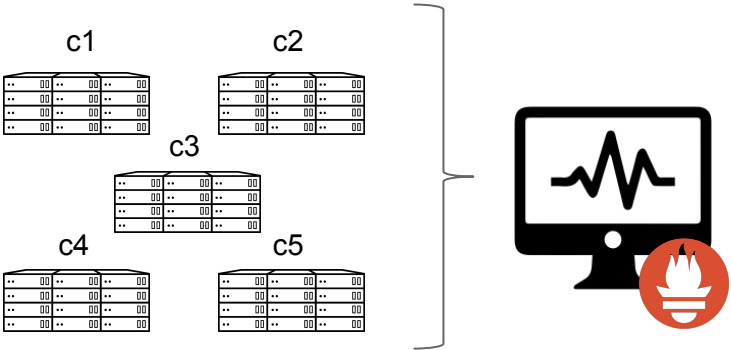
OPENS SHIFT MONITORING



Application Monitoring



Support for Grafana



Multi-cluster monitoring & observability

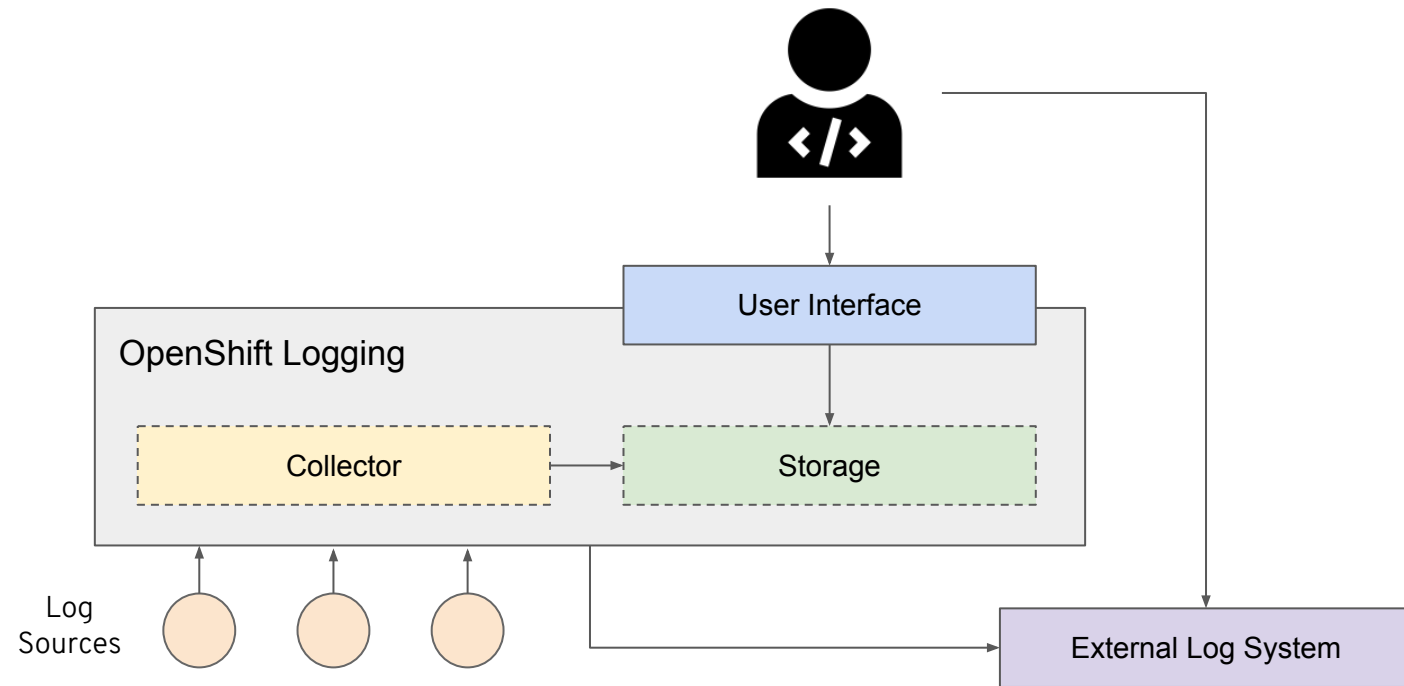
OPENS SHIFT LOGGING

Short term goals:

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

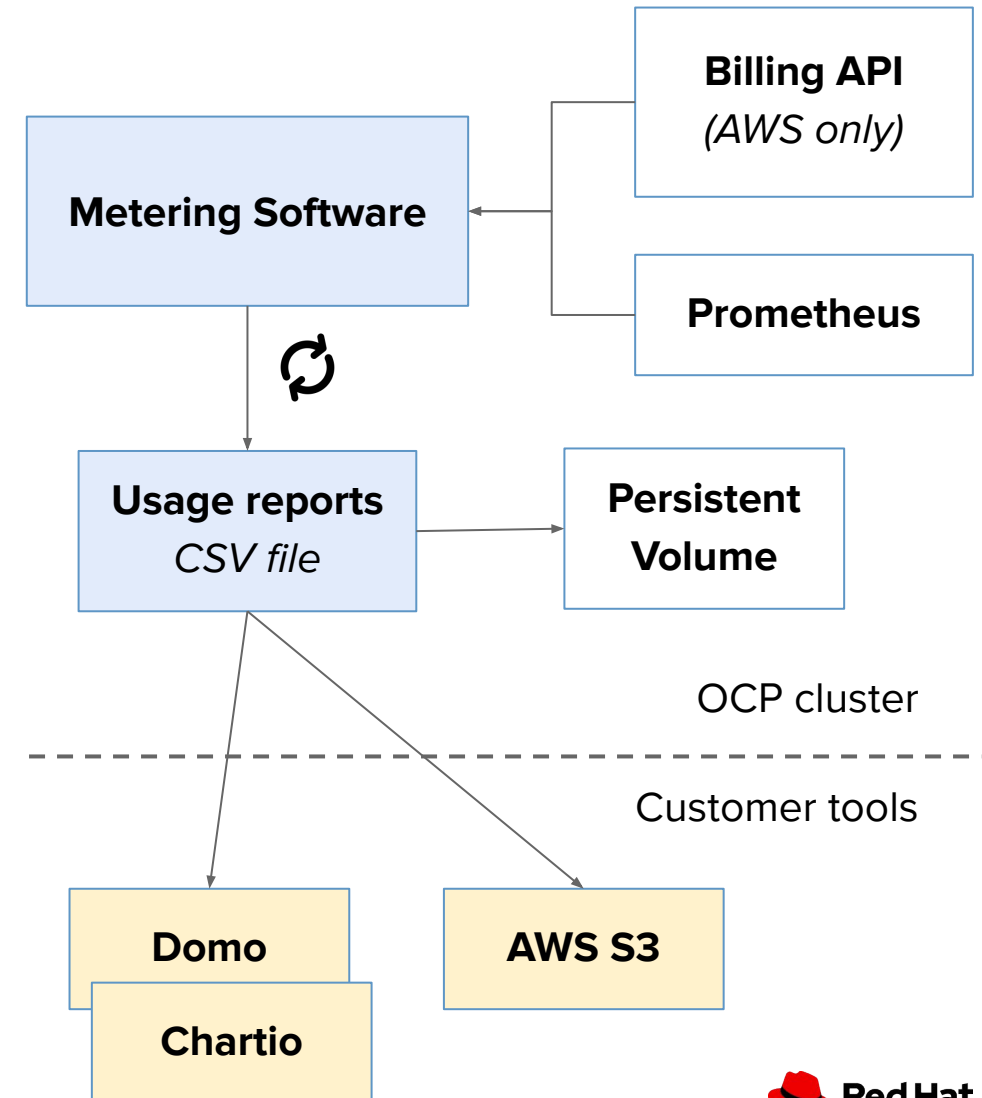
Long term goal:

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



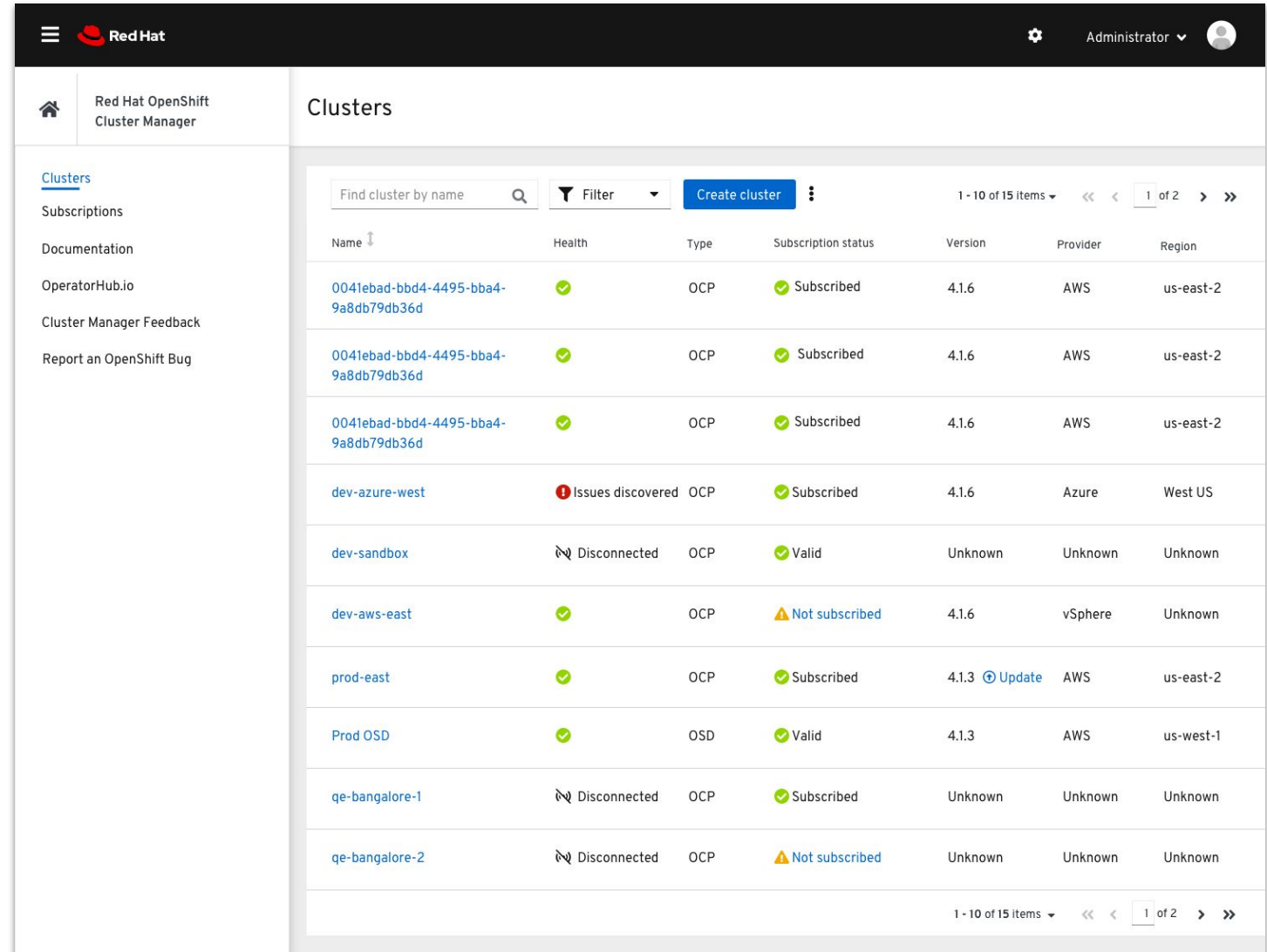
OPENSIFT CLUSTER MANAGER

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



The screenshot displays the Red Hat OpenShift Cluster Manager interface. The top navigation bar includes the Red Hat logo, a settings icon, and the user role 'Administrator'. The main content area is titled 'Clusters' and features a search bar, a filter dropdown, and a 'Create cluster' button. Below this is a table listing various clusters with columns for Name, Health, Type, Subscription status, Version, Provider, and Region. The table shows several clusters in different states, including 'Subscribed', 'Issues discovered', 'Disconnected', and 'Valid'.

Name	Health	Type	Subscription status	Version	Provider	Region
0041ebad-bbd4-4495-bba4-9a8db79db36d	✓	OCP	✓ Subscribed	4.1.6	AWS	us-east-2
0041ebad-bbd4-4495-bba4-9a8db79db36d	✓	OCP	✓ Subscribed	4.1.6	AWS	us-east-2
0041ebad-bbd4-4495-bba4-9a8db79db36d	✓	OCP	✓ Subscribed	4.1.6	AWS	us-east-2
dev-azure-west	! Issues discovered	OCP	✓ Subscribed	4.1.6	Azure	West US
dev-sandbox	🔌 Disconnected	OCP	✓ Valid	Unknown	Unknown	Unknown
dev-aws-east	✓	OCP	⚠ Not subscribed	4.1.6	vSphere	Unknown
prod-east	✓	OCP	✓ Subscribed	4.1.3 Update	AWS	us-east-2
Prod OSD	✓	OSD	✓ Valid	4.1.3	AWS	us-west-1
qe-bangalore-1	🔌 Disconnected	OCP	✓ Subscribed	Unknown	Unknown	Unknown
qe-bangalore-2	🔌 Disconnected	OCP	⚠ Not subscribed	Unknown	Unknown	Unknown

OpenShift Console

The future is now.

**Extending the
Console**

**Improve
Observability**

**Administration
made easy**

**Developer
Focused**

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)

The screenshot shows the Red Hat OpenShift Container Platform interface. The top navigation bar includes the Red Hat logo, 'OpenShift Container Platform', and user information 'kube:admin'. The left sidebar contains a navigation menu with categories like Administrator, Home, Projects, Search, Explore, Events, Operators, Workloads, Networking, Storage, Builds, Monitoring, Compute, User Management, and Administration. The main content area is titled 'Projects > Project Details' and shows the project 'tony' is active. It features several widgets: 'Details' (Name: tony, Requester: kube:admin, Labels: No labels), 'Status' (Active), 'Launcher' (Service Mesh), 'Inventory' (4 Deployments, 4 Pods, 0 PVCs, 1 Service, 0 Routes, 4 Config Maps, 21 Secrets), 'Utilization' (CPU: 8.39m, Memory: 96.31 MiB, Pod count: 4), and 'Activity' (Recent Events). The Utilization widget includes a table and line graphs for CPU, Memory, and Pod count over time.

Resource	Usage	15:10	15:30	15:50
CPU	8.39m	10m	5m	5m
Memory	96.31 MiB	150 MiB	100 MiB	50 MiB
Pod count	4	4	2	2

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.

The screenshot displays the OpenShift console interface. At the top, a green notification bar reads: "This is an example notification message with an optional link. [Optional link text](#)". The main header shows "Red Hat OpenShift Container Platform" and the user "kube:admin".

The "Installed Operators" page is shown, with a table listing operators in the "tony" namespace:

Name	Namespace	Deployment	Status	Provided APIs
AMQ Streams	NS tony	amq-streams-	InstallSucceeded	Kafka

A dropdown menu for "Third Party Applications" is open, showing "Couchbase Server Web Console" with a link icon. A dotted blue line connects this link icon to the "Service Mesh" link in the "Project Details" view below.

The "Project Details" view for the "tony" project shows it is "Active". The "Details" tab is selected, displaying fields for Name (tony), Requester (kube:admin), and Labels (No labels). The "Status" section shows "Active" with a green checkmark. The "Launcher" section contains a link for "Service Mesh". The "Activity" section shows "Ongoing" with a note: "There are no ongoing activities."

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 **Console YAML Samples 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

The screenshot displays the Red Hat OpenShift Console interface. On the left, a navigation menu shows 'Workloads' expanded to 'Jobs'. The main area is titled 'Create Job' and shows a YAML editor with the following content:

```
1 apiVersion: batch/v1
2 kind: Job
3 metadata:
4   name: example
5   namespace: brie
6 spec:
7   selector: {}
8   template:
9     metadata:
10      name: pi
11     spec:
12       containers:
13       - name: pi
14         image: perl
15         command:
16         - perl
17         - '-Mbignum=bpi'
18         - '-wle'
19         - print bpi(2000)
20       restartPolicy: Never
```

On the right, a 'Job' details panel shows a 'Samples' tab with a list of samples. One sample is visible: '1. Example Job' with the description 'An example Job YAML sample'. A 'Download YAML' button is present next to it.

Below the main interface, a 'Custom Resource Definition Details' window is open for the 'consoleyamlsamples.console.openshift.io' CRD. It shows the 'Instances' tab with a table of existing samples:

Name	Namespace	Created
example	None	2 minutes ago

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 OpenShift Container Platform

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Administrators

Home

Dashboards

Overview

Projects

Search

Explore

Events

Operators

OperatorHub

Installed Operators

Workloads

Pods

Deployments

Deployment Configs

Stateful Sets

Secrets

Config Maps

Cron Jobs

Details

View settings

Cluster API Address

<https://api.sgoodwin2.devcluster.openshift.com:6443>

Cluster ID

e75320a2-12f0-4f8f-af8c-2812e12c7607

[OpenShift Cluster Manager](#)

Provider

Status

Cluster

Control Plane

Image Security

1 vulnerabilities

Nov 7, 12:09 am

A client in the cluster is using deprecated apps/vbeta2 API that will be removed soon.

Nov 7, 12:09 am

A client in the cluster is using deprecated apps/vbeta2 API that will be removed soon.

Security breakdown

Quay analyzes container images to identify vulnerabilities.

Severity

Fixable

1 High

1 total

Fixable Vulnerabilities

1 namespaces

openssl-libs

RED HAT Quay.io

EXPLORE

TUTORIAL

PRICING

alecmendler/couchbase-server

29abc8c5a3b2

Quay Security Scanner has detected 61 vulnerabilities.

Patches are available for 61 vulnerabilities.

14 High-level vulnerabilities.

33 Medium-level vulnerabilities.

14 Low-level vulnerabilities.

Vulnerabilities

CVE	SEVERITY	PACKAGE	CURRENT VERSION	FIXED IN VERSION	INTRODUCED IN LAYER
RHSA-2019-0710	High	python-libs	2.7.5-68.el7	0.2.7.5-77.el7_6	
RHSA-2019-1587	High	python-libs	2.7.5-68.el7	0.2.7.5-80.el7_6	
RHSA-2019-0368	High	systemd-libs	219-57.el7	0.219-62.el7_6.5	
RHSA-2019-0049	High	systemd-libs	219-57.el7	0.219-62.el7_6.2	
RHSA-2019-0679	High	libssh2	1.4.3-10.el7_2.1	0.1.4.3-12.el7_6.2	
RHSA-2018-2285	High	yum-plugin-ovl	1.1.31-45.el7	0.1.1.31-46.el7_5	
RHSA-2018-3181	High	rpm	2.0.22.4.el7	0.2.0.22.5.el7_6	

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

The screenshot displays the OpenShift console interface. On the left, a dark sidebar contains a navigation menu with 'User Management' expanded to show 'Users' and 'Groups'. The main content area is split into two panels. The top panel, titled 'Groups', features a 'Create Group' button and a table with columns 'Name' and 'Users'. The table lists two groups: 'admins' and 'app_devs', both with a count of 2 users. The bottom panel, titled 'Users', shows a 'You are logged in' notification and a list of users: 'developer' and 'user'. A context menu is open over the 'user' entry, providing actions: 'Impersonate User "user"', 'Edit Labels', 'Edit Annotations', 'Edit User', and 'Delete User'.

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

The image shows a screenshot of the Red Hat OpenShift Container Platform interface. The main window displays the 'Alerting' configuration page, which includes sections for 'Overview', 'Alert Routing', and 'Receivers'. The 'Alert Routing' section shows 'Group By' set to 'job' and 'Group Wait' set to '30s'. The 'Receivers' section has a 'Create Receiver' button. A modal dialog box titled 'Create Receiver' is open in the foreground, showing the following configuration options:

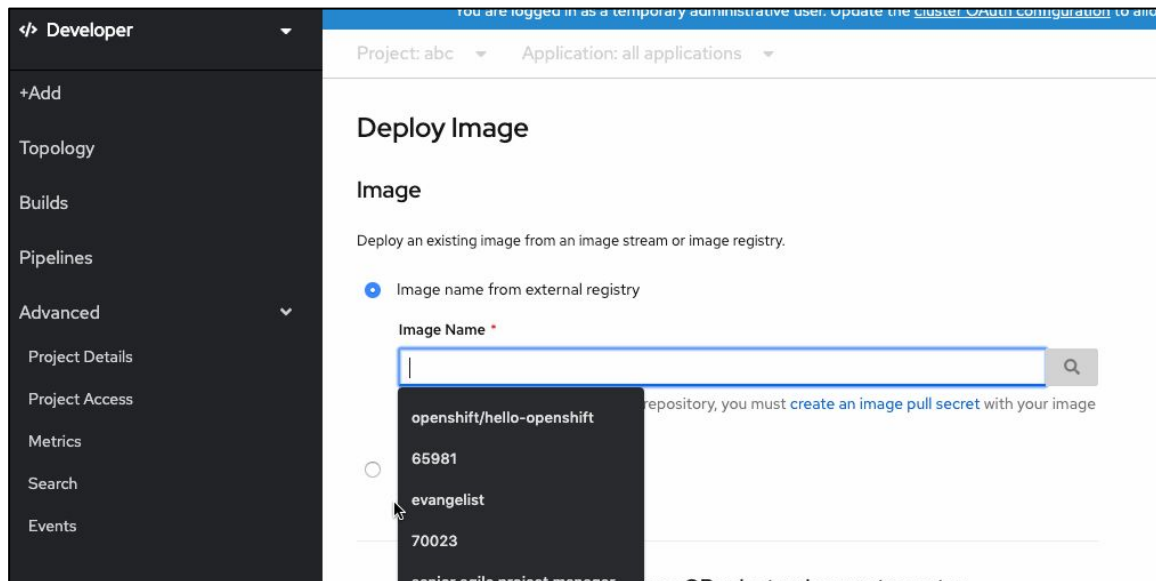
- Receiver Name ***: my-new-receiver
- Receiver Type ***: PagerDuty
- PagerDuty Configuration**
 - Integration Type**: Events API v2 Prometheus
 - Routing Key ***: thisissometextthatwillblurverysoon
 - PagerDuty integration key
 - Routing Labels**: Firing alerts with labels that match all of these selectors will be sent to this receiver. Label values can be matched exactly or with a [regular expression](#).
 - Table with columns NAME and VALUE:

NAME	VALUE
severity	warning
 - Regular Expression
 - [+ Add Label](#)
- Create** and **Cancel** buttons.

Deploy Applications streamlining flows

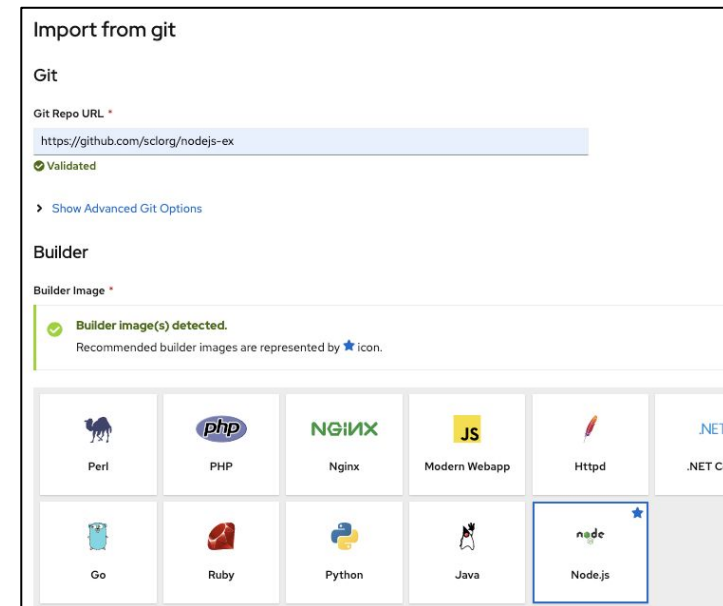
Deploy Image from Internal Registry

- 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



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

The screenshot shows the Red Hat OpenShift Container Platform Developer console. The top navigation bar includes the Red Hat logo and the text "OpenShift Container Platform". A blue notification bar at the top right states: "You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to". Below this, the breadcrumb navigation shows "Project: abc" and "Application: all applications". A message box indicates: "There are no pipeline templates available for this runtime." The main content area is titled "Resources" and prompts the user to "Select the resource type to generate". Three options are listed: "Deployment" (selected), "Deployment Config", and "Knative Service" (marked as "Tech Preview").

Red Hat
OpenShift Container Platform

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to

Project: abc Application: all applications

There are no pipeline templates available for this runtime.

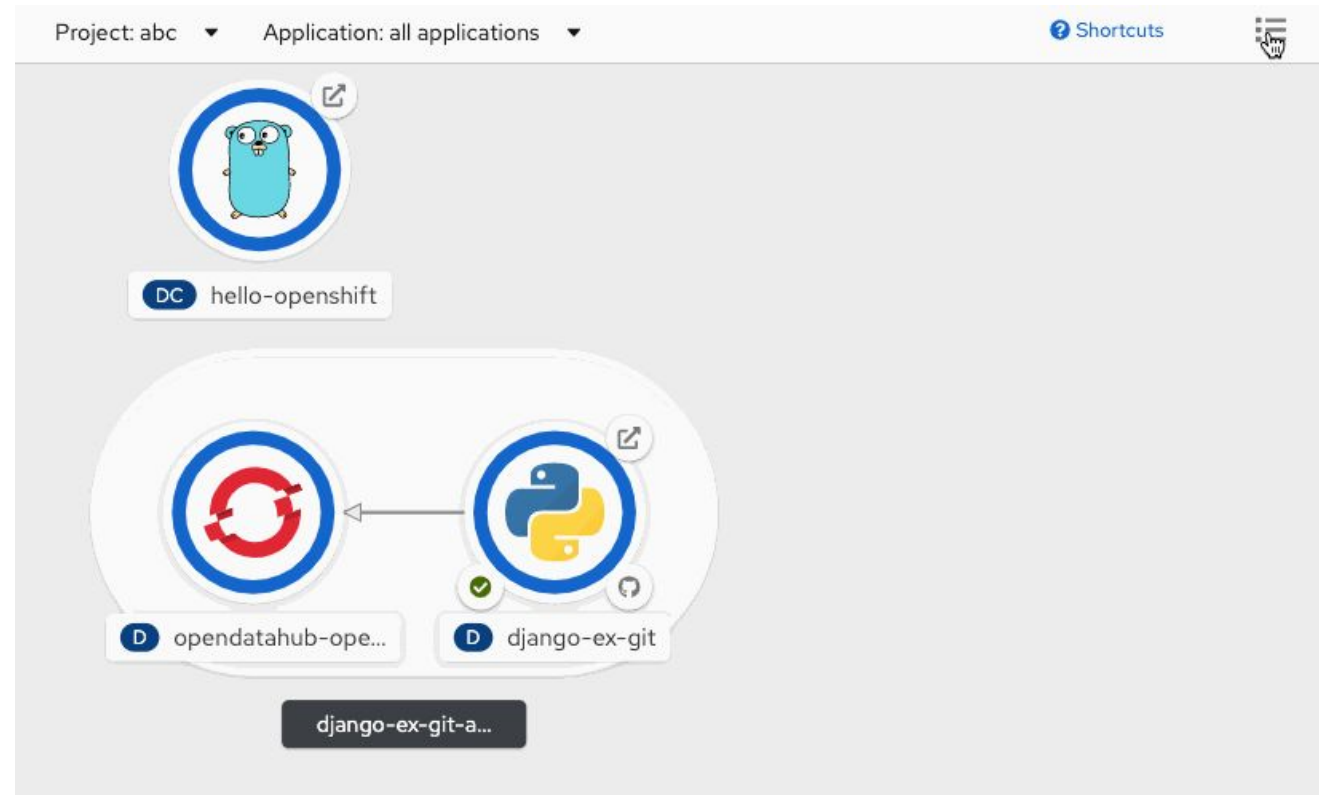
Resources

Select the resource type to generate

- Deployment
apps/Deployment
A Deployment enables declarative updates for Pods and ReplicaSets.
- Deployment Config
apps.openshift.io/DeploymentConfig
A Deployment Config defines the template for a pod and manages deploying new images or configuration changes
- Knative Service **Tech Preview**
serving.knative.dev/Service
A Knative Service enables scaling to zero when idle

Application Topology streamlined flows

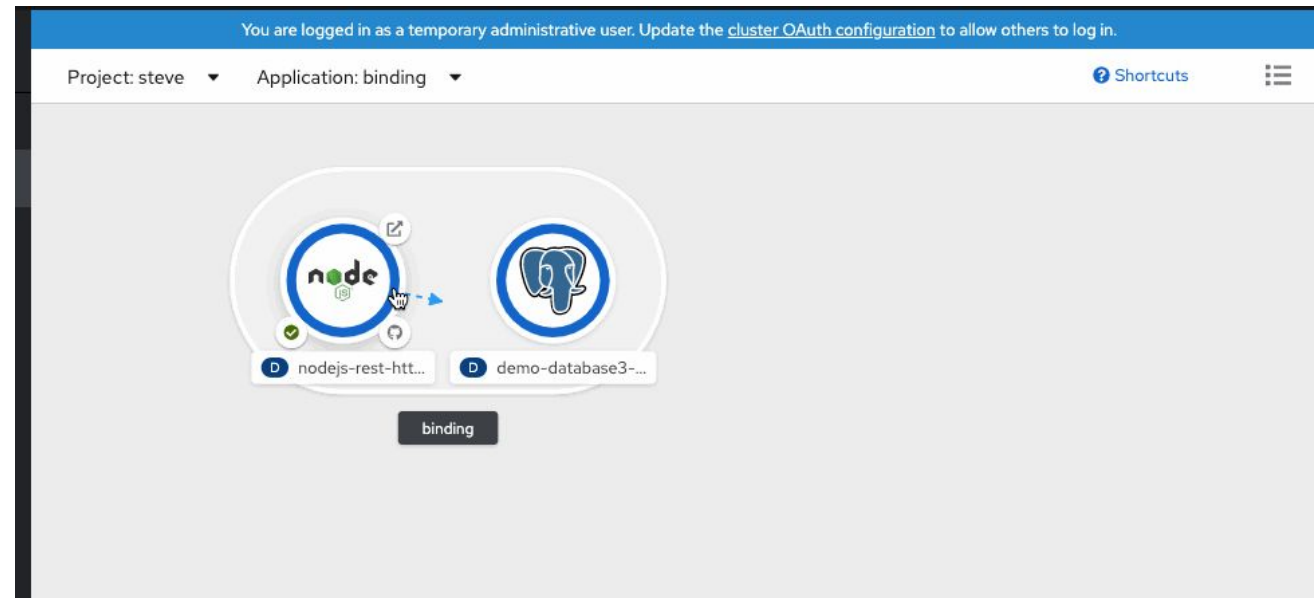
- Toggle between List and Topology views
- Easily group applications
- Connect/bind applications easily
- Contextual actions
- Quickly delete applications



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



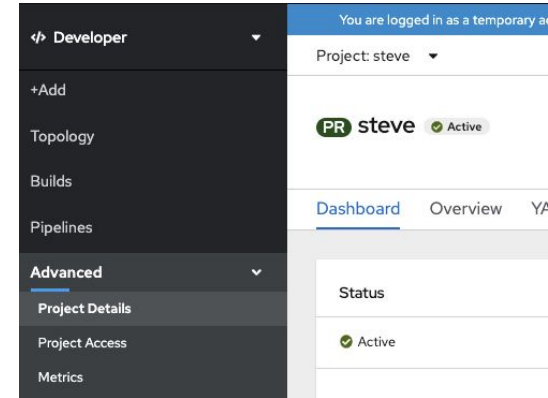
Learn more about service binding:

<https://github.com/redhat-developer/service-binding-operator>

Project Details & Access

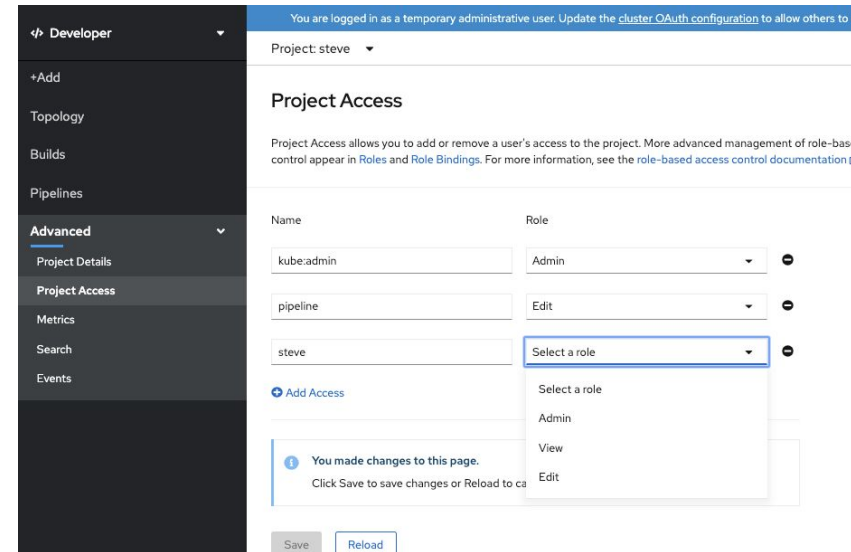
Project Details

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



Project Access

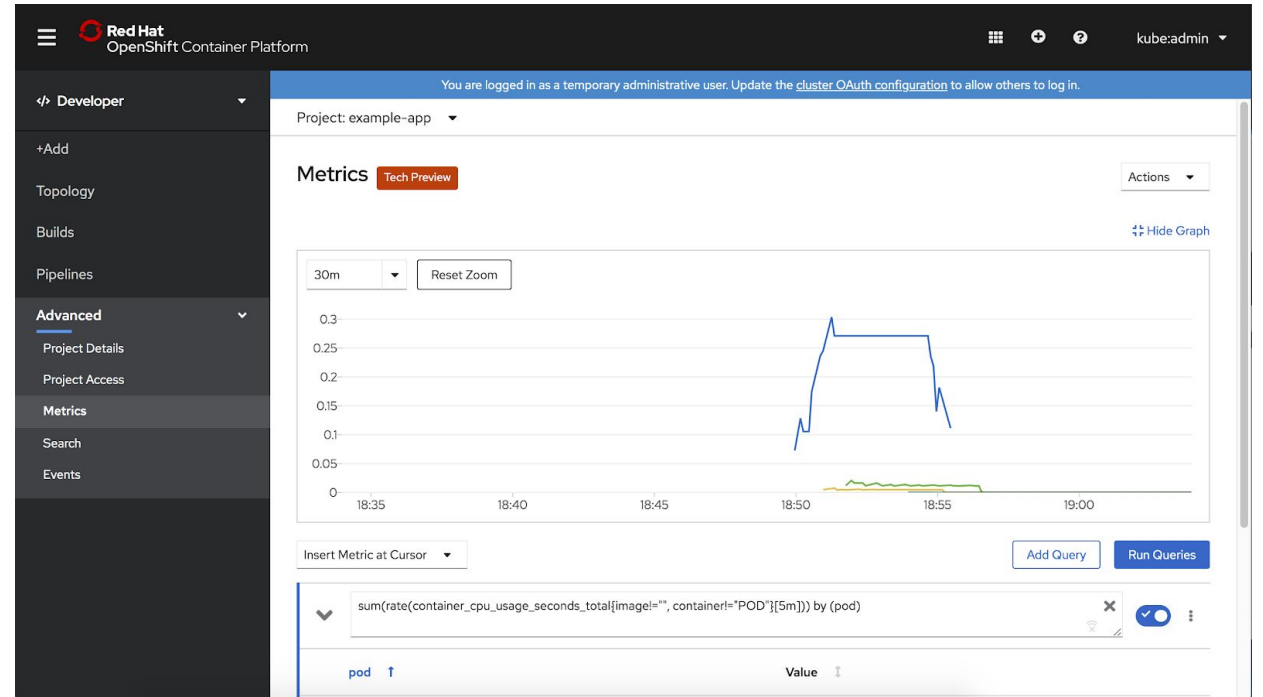
- Simplify sharing projects
- Reduces to a simple set of Roles that developer frequently use



Metrics

Quick access to key application metrics

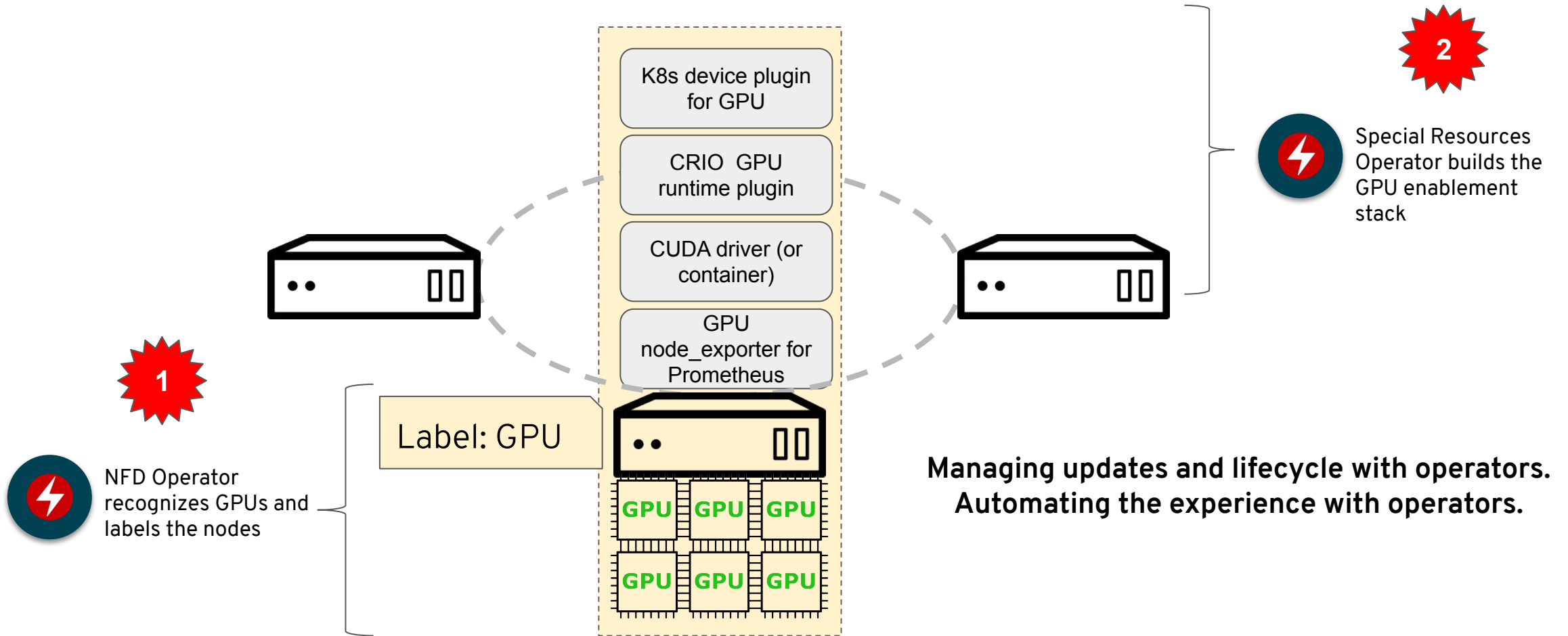
- Use of Prometheus Query Language
- Easily build up queries and plot to visualize application and component trends



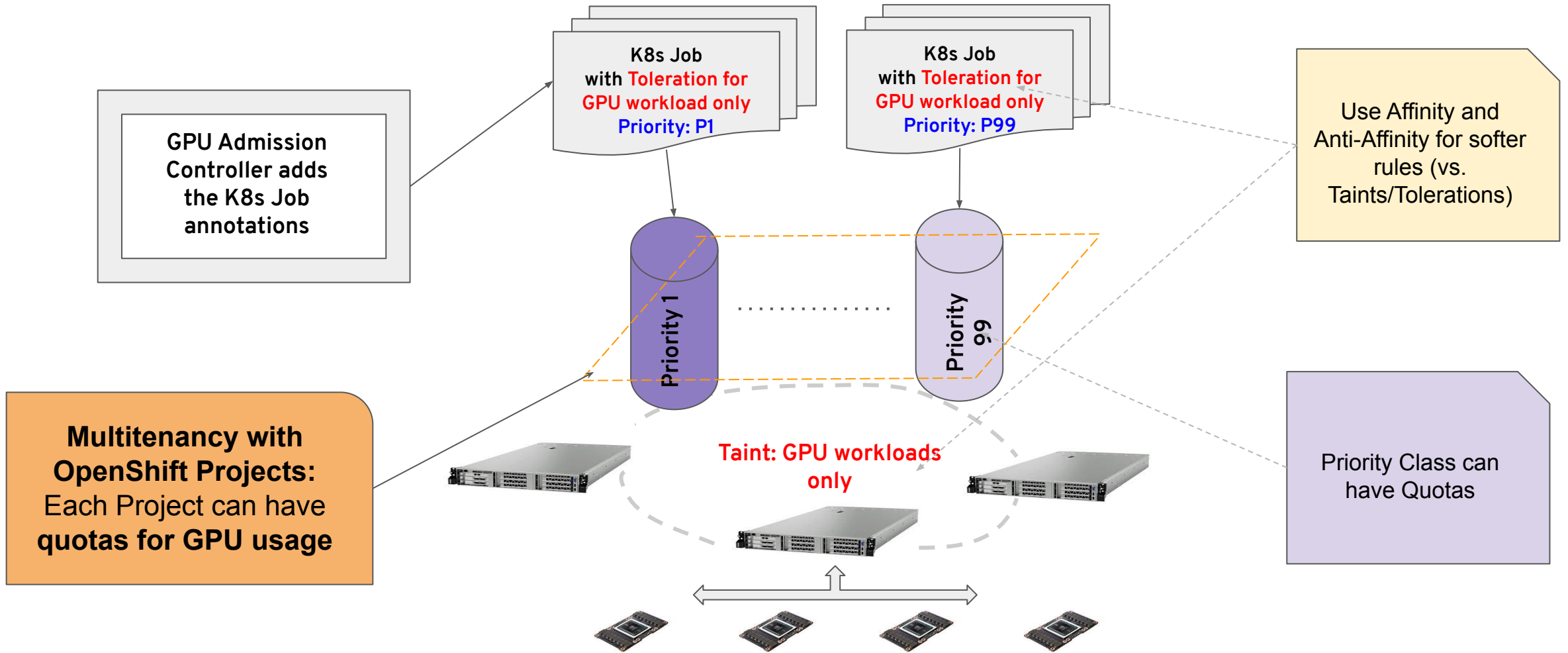


Compute

Automating the enablement of GPUs in an OpenShift Cluster



GPUs in OpenShift



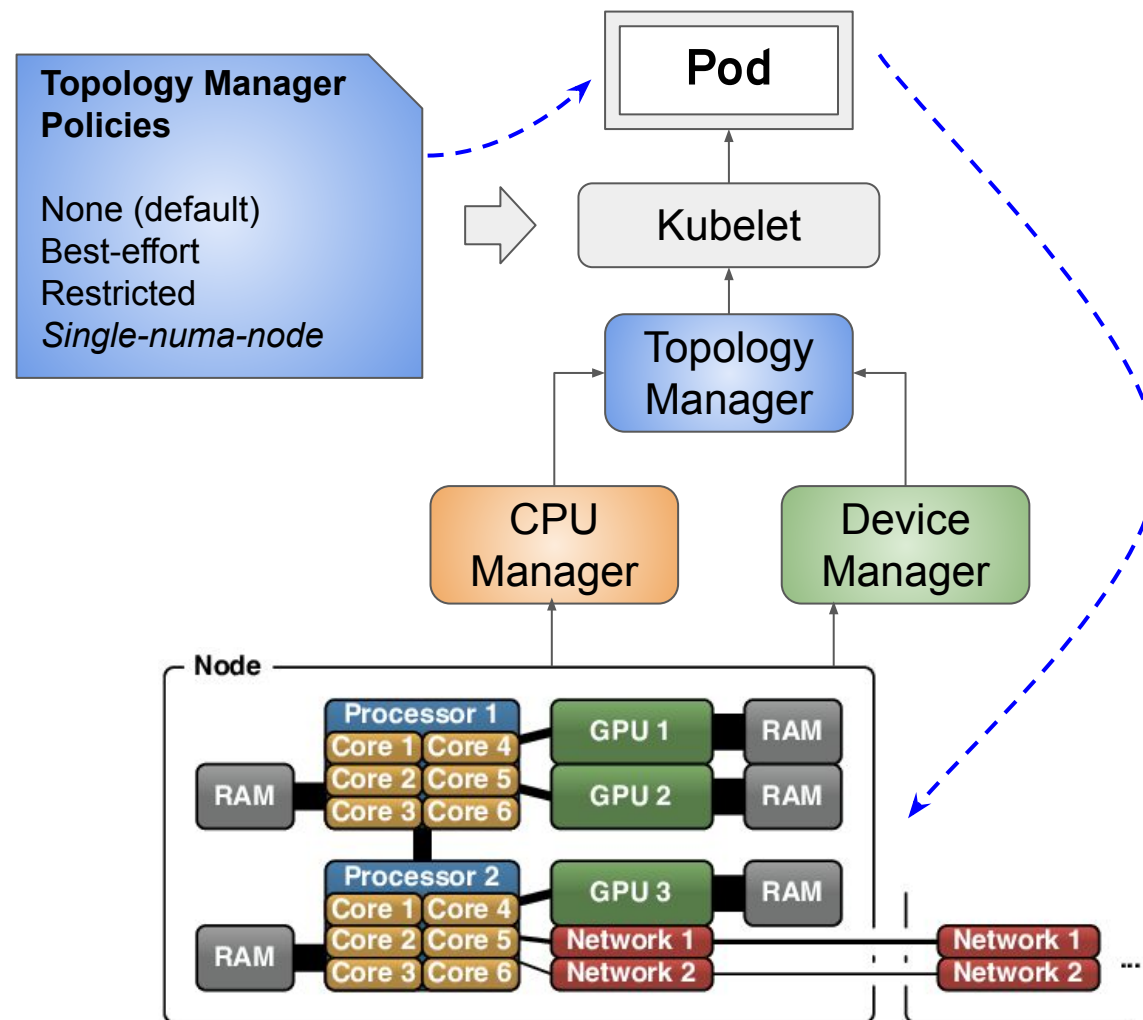
Node Topology Manager

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



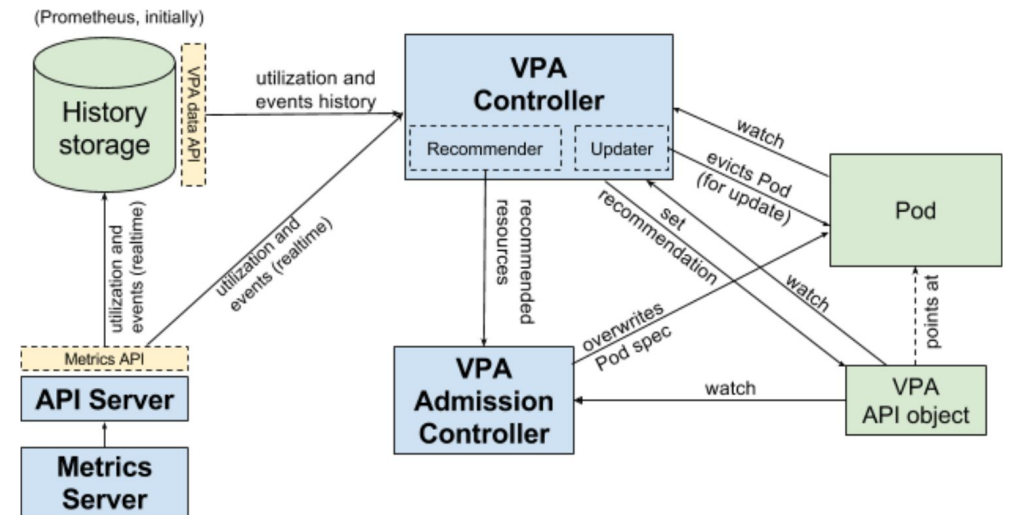
Vertical Pod Autoscaler

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.





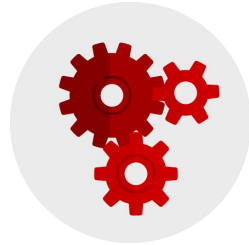
Networking

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)

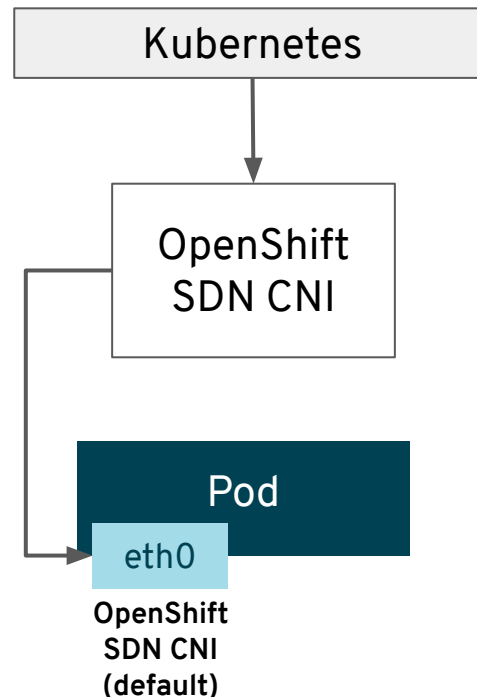
Capability-Building Feature: Networking Plugins

Multus Enables Multiple Networks & New Functionality to Existing Networking

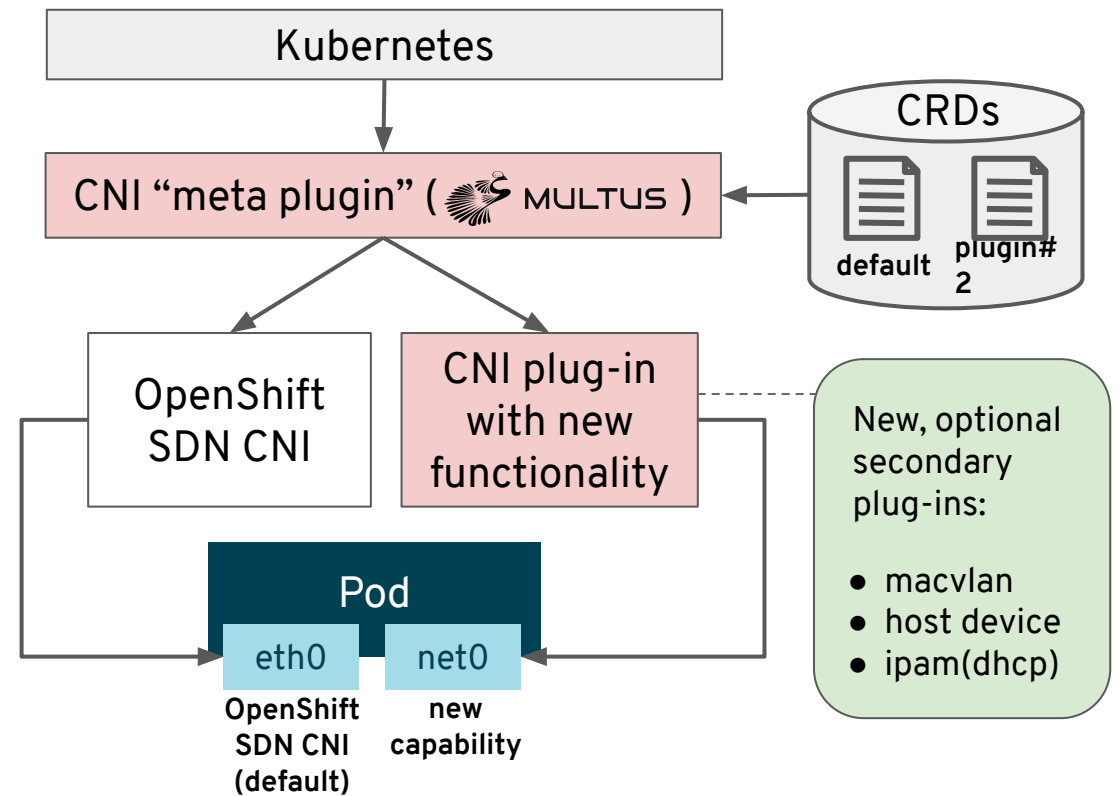
The Multus CNI “meta plugin” for Kubernetes enables one to create multiple network interfaces per pod, and assign a CNI plugin to each interface created.

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...



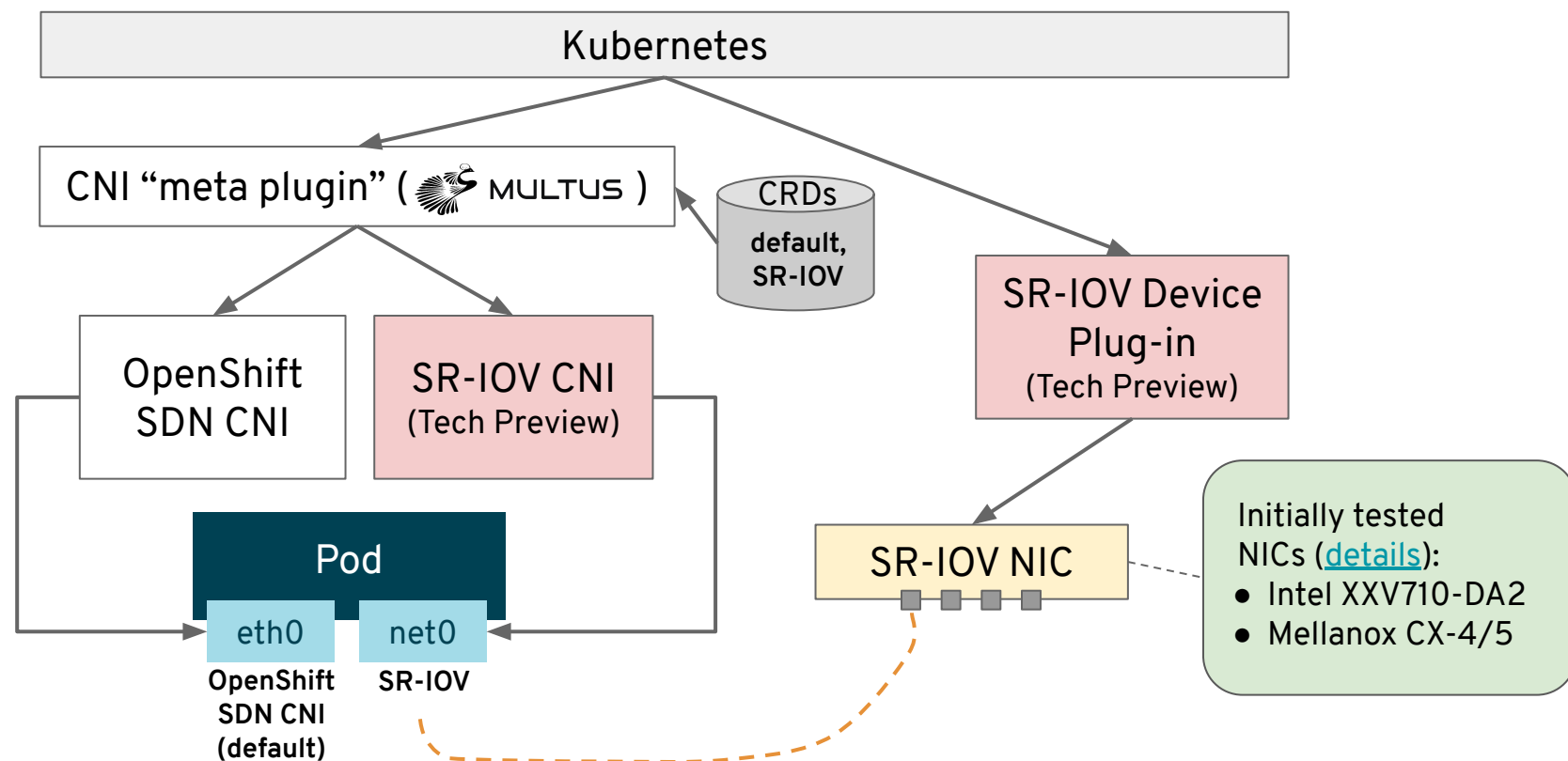
4.x Capability...



High-Performance Networking

SR-IOV Solution:

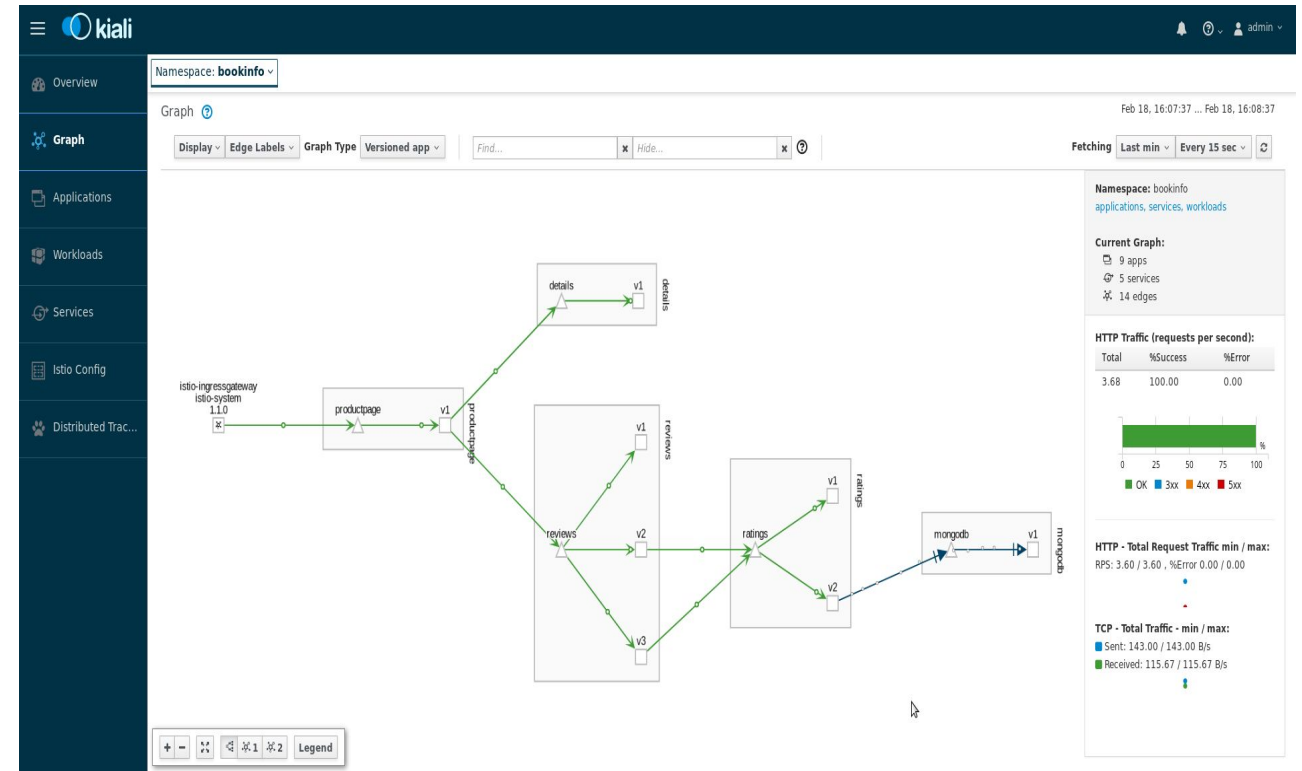
- Tech Preview since Jun 2019
- Fully Supported Dec 2019
- CNI Plug-In
- Device Plug-In
- RDMA / RoCE Support
- DPDK Mode for SR-IOV VFs
- Admission Controller
- Operator
- VF Security & QoS Flags



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

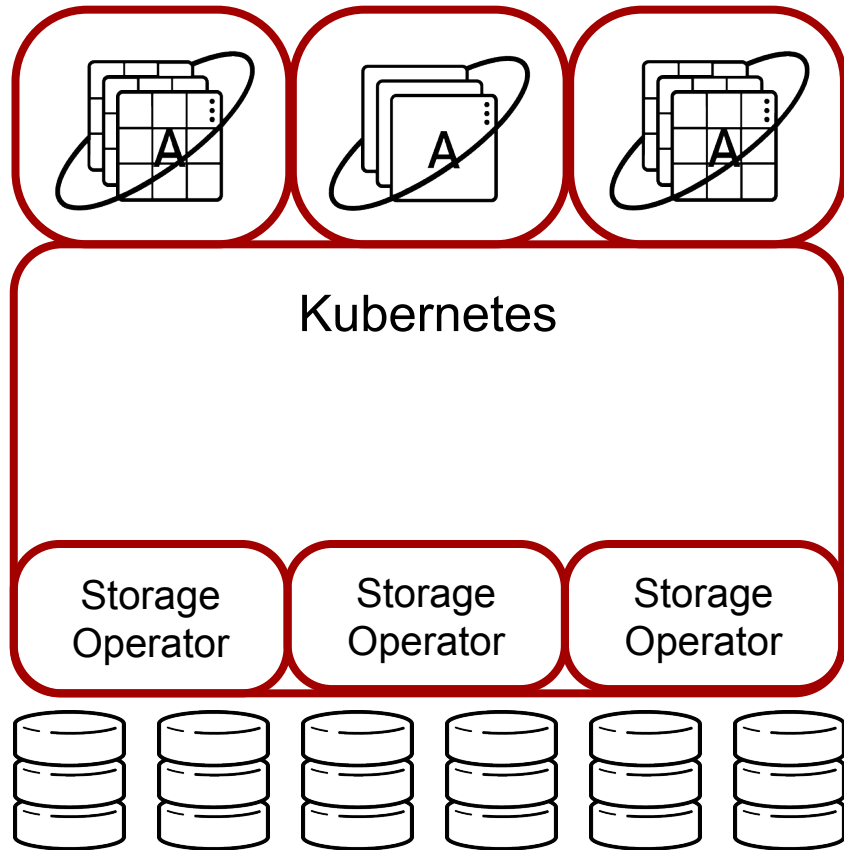




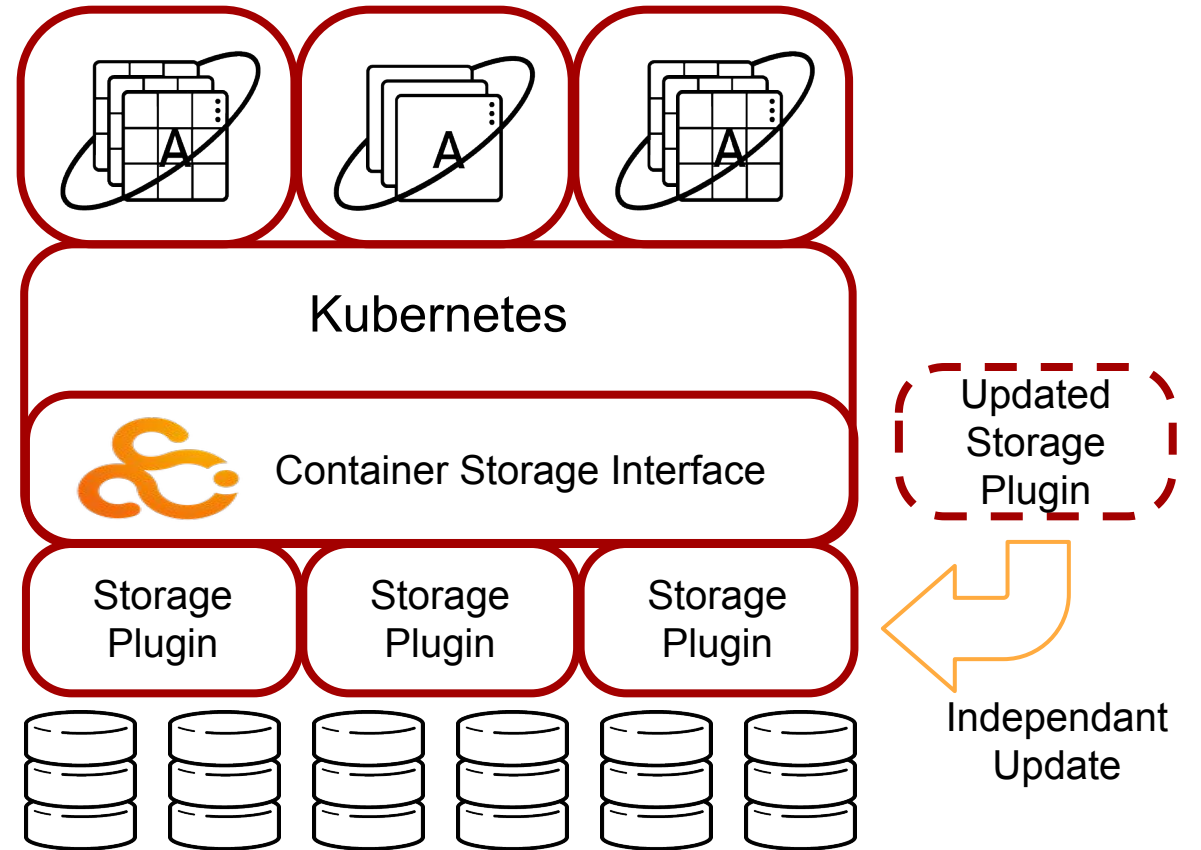
Storage

OpenShift CSI

4.1



4.2



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	

OCS 4.2: Change in Technology Stack

Goal to have complete storage for OCP whatever the needs

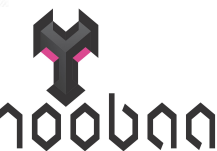
- **Need for scalable S3 object stack** (New apps, infra like chargeback, metering)
- **Red Hat Ceph** is scalable object stack with block and file
- **Recently acquired Noobaa** - consistent S3 interface over Ceph RGW, AWS S3, Azure Blob; Federation & multi-cloud capable
- **Rook** operator framework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

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

OCS 3



OCS 4



OpenShift Container Storage 4.2

Persistent data services for OCP Hybrid Cloud

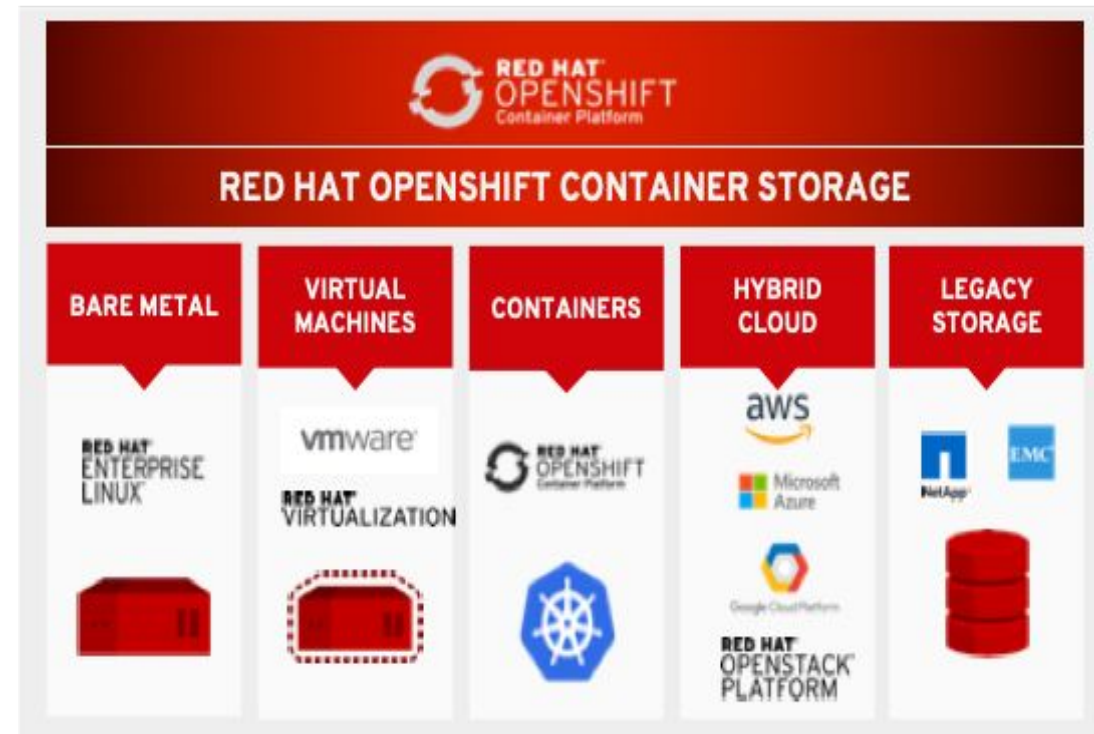
- Complete Data Services: RWO, RWX & **S3(new)** (block, file & object)
- Persistent storage for all OCP Infra and Applications
- Build and deploy anywhere -Consistent Storage Consumption, management, and operations

OCS 4.2 support with OCP 4.2

- Platform support: AWS and VMware
- Converged 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





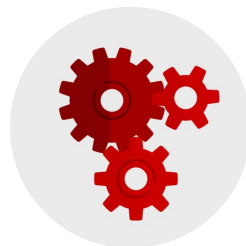
Security

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

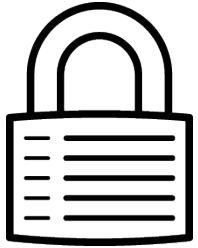


Automate Compliance

Disconnected / air-gapped install
FIPS compliance
Cipher Suite Configuration
Compliance Operator

Stronger Platform Security

Defense in Depth



CONTROL

Application Security

- [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 \(tech preview\)](#)



DEFEND

Infrastructure



EXTEND

OpenShift 4 and Fips 140-2

FIPS ready Services

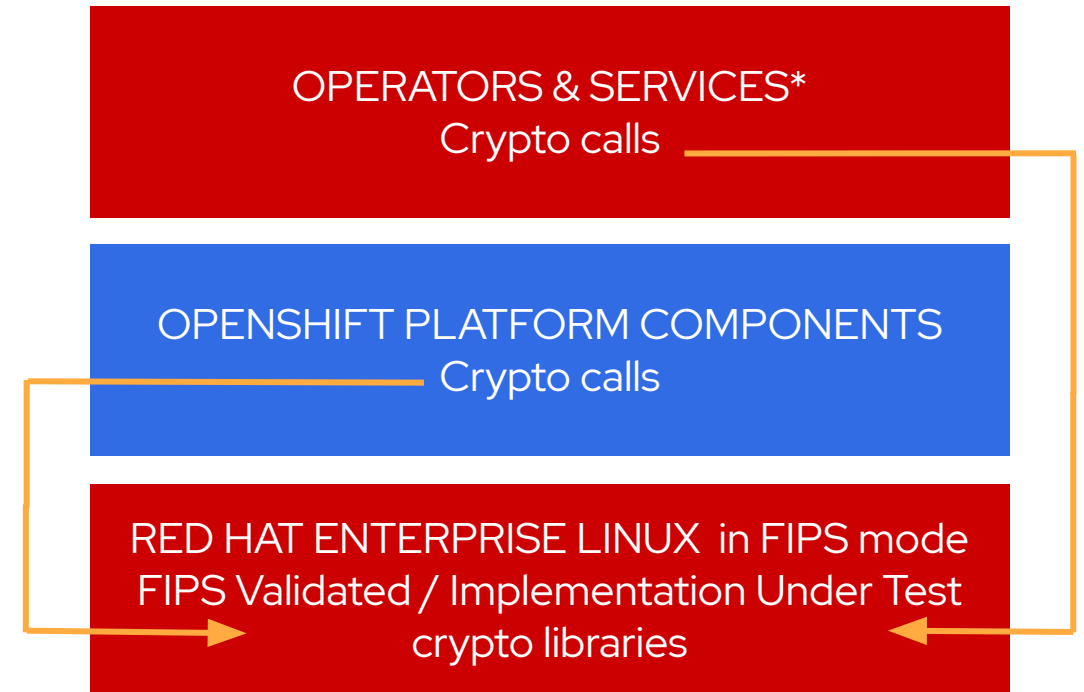
- When built with RHEL 7 base image

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



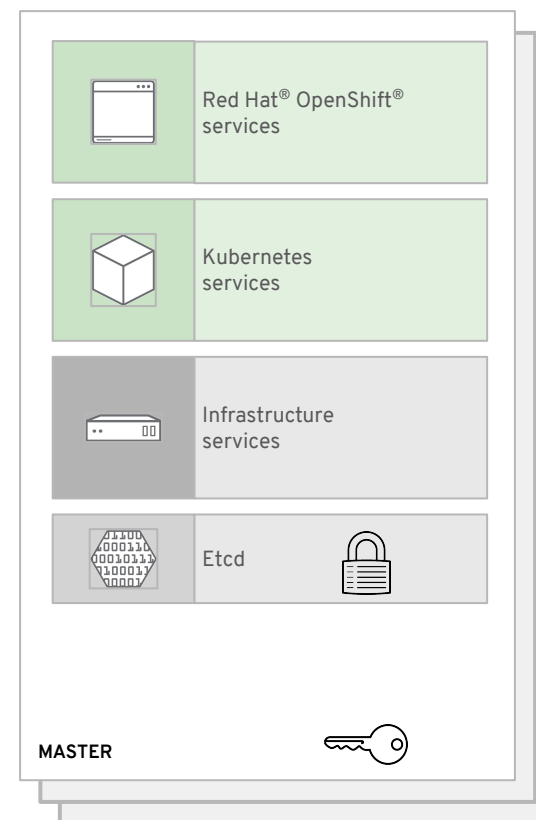
*When built with RHEL base images

[More about RHEL go and FIPS 140-2](#)

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"`
```




Thank You



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



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



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



twitter.com/RedHat