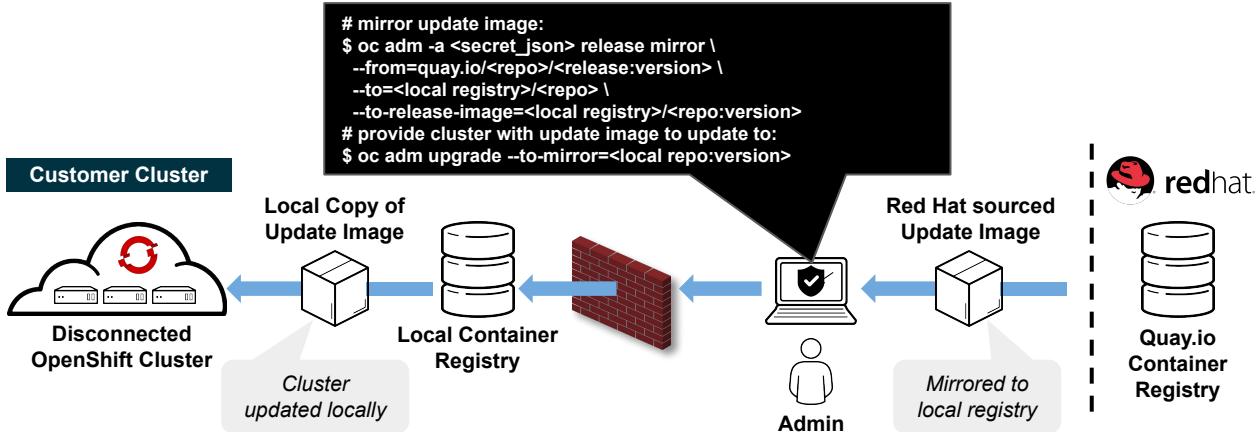




What's New in OpenShift 4.2

Technical view

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

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>
```

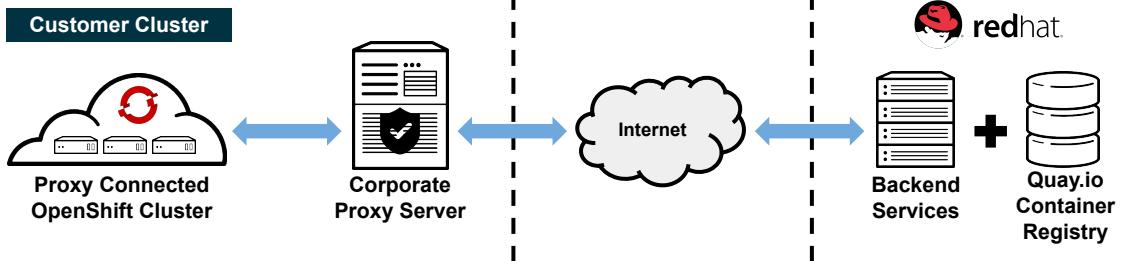
4. Create Operator Catalog Image using this dockerfile:

```
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 image>/example-registry:latest
```

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 config:

```

$ 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.openshift.com,etcd-0.demo.example.com,etcd-1.demo.example.com,etcd-2.demo.example.com,example.com,localhost
  
```

Machine Config Operator (MCO)

Provides cluster-level configuration, enables rolling upgrades, and prevents drift between new and existing nodes. The MCO is the heart of what makes RHCOS a kube-native operating system.

Configure Kernel Arguments for the Cluster

- oc create -f 50-kargs.yaml
- oc edit mc/50-kargs

MCO can be **paused** to suspend operations

Provides control for when changes can be deployed

Custom MachinePools can have inheritance

Enables MachineConfigs to scale

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 50-kargs
spec:
  KernelArguments:
    audit=1
    audit_backlog_limit=8192
    net.ifnames.prefix=net
```

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

NODE LABELS
feature.node.kubernetes.io/kernel-version.full=4.18.0-80.7.2.el8_0.x86_64
feature.node.kubernetes.io/pci-1d0f.present=true
feature.node.kubernetes.io/system-os_release.VERSION_ID.minor=1
beta.kubernetes.io/os=linux feature.node.kubernetes.io/kernel-version.minor=18
failure-domain.beta.kubernetes.io/zone=us-east-1a node.openshift.io/os_version=4.1
feature.node.kubernetes.io/cpu-cpuid.AESNI=true
feature.node.kubernetes.io/cpu-cpuid.RTM=true
feature.node.kubernetes.io/system-os_release.VERSION_ID.major=4
feature.node.kubernetes.io/kernel-version.major=4
feature.node.kubernetes.io/cpu-pstate.turbo=true
feature.node.kubernetes.io/kernel-version.revision=0
feature.node.kubernetes.io/storage-nonrotationaldisk=true
node-role.kubernetes.io/worker failure-domain.beta.kubernetes.io/region=us-east-1
node.openshift.io/os_id=rhcos feature.node.kubernetes.io/cpu-cpuid.ADX=true
beta.kubernetes.io/instance-type=p2.xlarge
feature.node.kubernetes.io/cpu-cpuid.AVX2=true
kubernetes.io/hostname=ip-10-0-137-244
feature.node.kubernetes.io/system-os_release.ID=rhcos
beta.kubernetes.io/arch=amd64 feature.node.kubernetes.io/pci-10de.present=true
feature.node.kubernetes.io/pci-1013.present=true
feature.node.kubernetes.io/system-os_release.VERSION_ID=4.1
feature.node.kubernetes.io/cpu-cpuid.HLE=true
feature.node.kubernetes.io/cpu-hardware_multithreading=true
feature.node.kubernetes.io/cpu-cpuid.FMA3=true
feature.node.kubernetes.io/cpu-cpuid.AVX=true

BOOT ID
34c7c01a-2cad-4e20-9549-323315569d94

CONTAINER RUNTIME
cri-o://1.13.10-0.1.dev.rhaos4.1.git9e2e1de.el8-dev

KUBELET VERSION
v1.13.4+d81afa6ba

KUBE-PROXY VERSION
v1.13.4+d81afa6ba

TRAINTS

Special Resources Operator (SRO)

What is Special Resources Operator (SRO)

As an operator for orchestrating special resources in a cluster that need special handling.

Implements a simple state machine, where each state implements a state and has a validation step.

```
$ make -C special-resource-operator deploy
customresourcedefinition.apiextensions.k8s.io/specialresources.sro.openshift.io created
for obj in manifests/0000_namespace.yaml
<snip>
namespace/openshift-sro-operator created
namespace/openshift-sro created
specialresource.sro.openshift.io/gpu created
$
```

What are we introducing in 4.2?

A reference implementation of the SRO for NVIDIA GPUs.

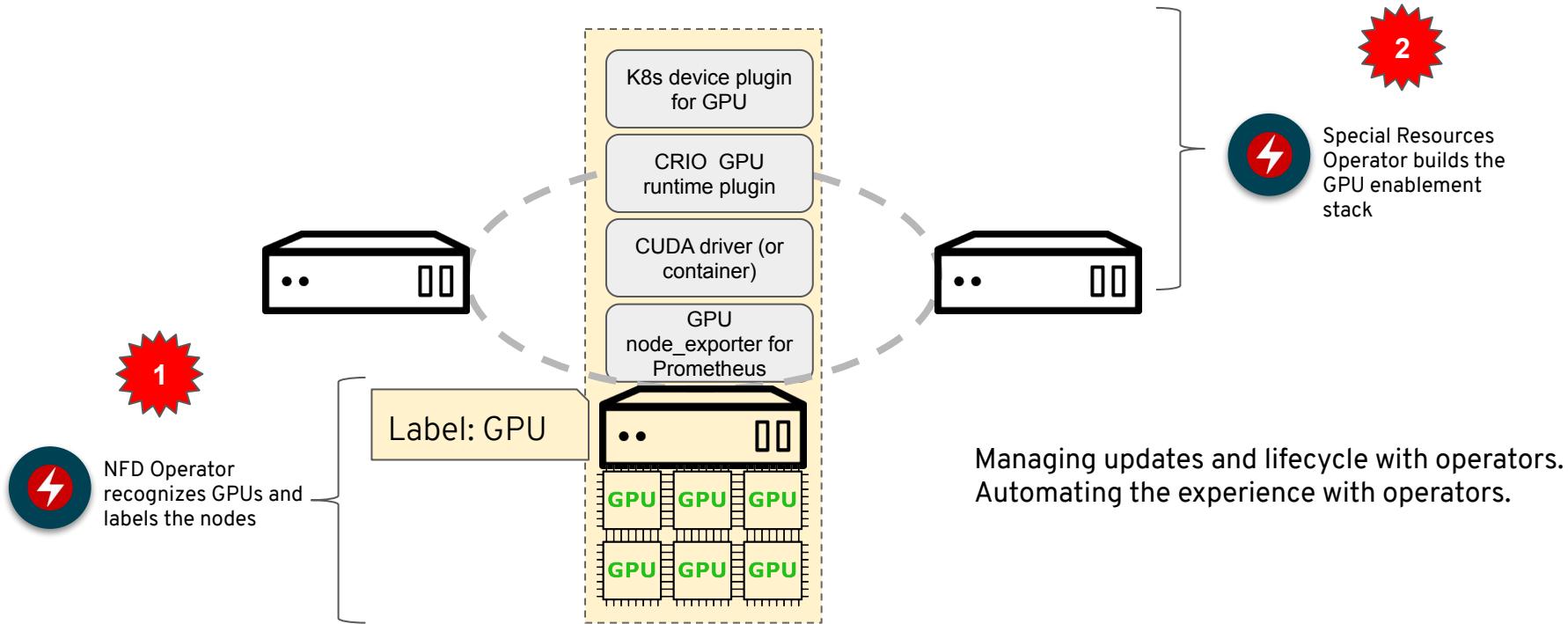
This SRO deploys:

- NVIDIA Cuda driver container
- NVIDIA GPU device plugin deamenoset
- NVIDIA DGCM_exporter for GPU metrics

The screenshot shows the OpenShift web interface with the following details:

- Project:** openshift-sro
- Project Status:** Resources tab is selected.
- Resources:** nvidia-driver-daemonset
- Pod Details:**
 - Name:** nvidia-driver-daemonset
 - Memory:** 3.5 MB
 - Cores:** 0.000
 - Status:** 1 of 1 pods

Enablement of GPUs in an OpenShift Cluster



Node Tuning Operator Enhancements

What is it

- Node Tuning Operator was first introduced in OpenShift 4.1 and manages cluster node-level tuning; The default CR is meant for delivering standard node-level tuning.
- The enhancements in 4.2 allow for customizing the tunings (for things such as High Performance etc)

How it works

- For custom tuning, create your own tuned CRs. Newly created CRs will be combined with the default CR and custom tuning applied to nodes based on node/pod labels and profile priorities.

```
# The CR applies custom node-level tuning for nodes that run an ingress pod with label
tuned.openshift.io/ingress-pod-label=ingress-pod-label-value
oc create -f <<_EOF_
apiVersion: tuned.openshift.io/v1
kind: Tuned
metadata:
  name: ingress
  namespace: openshift-cluster-node-tuning-operator
spec:
  profile:
    - data: |
        [main]
        summary=A custom OpenShift ingress profile
        include=openshift-control-plane
        [sysctl]
        net.ipv4.ip_local_port_range="1024 65535"
        net.ipv4.tcp_tw_reuse=1
        name: openshift-ingress
    recommend:
      - match:
          - label: tuned.openshift.io/ingress-pod-label
            value: "ingress-pod-label-value"
            type: pod
            priority: 10
            profile: openshift-ingress
_EOF_
```

Whitelisting of sysctl for OpenShift 4 and RHEL CoreOS

What is it

- Procedure to allow system admins to whitelist sysctl on a per-node basis
- All safe sysctls are enabled by default; All unsafe sysctls are disabled by default, and the cluster administrator must manually enable them on a per-node basis.
- New functionality introduces this feature for OpenShift 4 and RHEL CoreOS

How it works

- Add a label to the node's MachineConfigPool
- Create a KubeletConfig custom resource on the node with the specified whitelist of sysctls
- Set sysctls on pods using the pod's `securityContext`

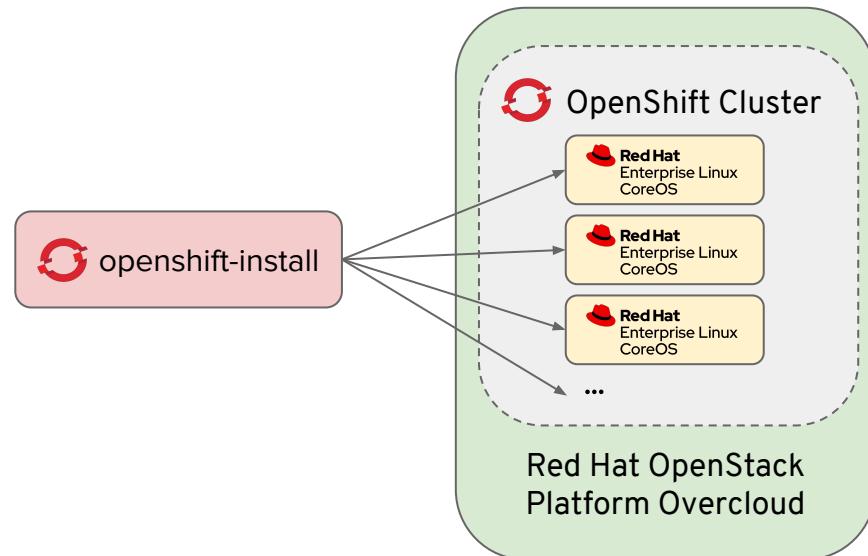
```
# Add a label to the MachineConfigPool where the containers
# where containers with the unsafe sysctls will run
$oc edit machineconfigpool worker
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfigPool
metadata:
  creationTimestamp: 2019-02-08T14:52:39Z
  generation: 1
  labels:
    custom-kubelet: sysctl (1)
# Add a key: pair label.
# Create a KubeletConfig Custom Resource (CR)
apiVersion: machineconfiguration.openshift.io/v1
kind: KubeletConfig
metadata:
  name: custom-kubelet
spec:
  machineConfigPoolSelector:
    matchLabels:
      custom-kubelet: sysctl (1)
  kubeletConfig:
    allowedUnsafeSysctls: (2)
      - "kernel.msg*"
      - "net.ipv4.route.min_pmtu"
$ oc apply -f set-sysctl-worker.yaml
```

INSTALLER PROVISIONED OPENSIFT-ON-OPENSTACK

OSP 13 (Long Life Release) and OSP 15

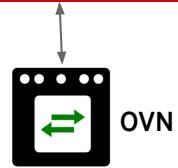
The OCP installer provisions (no customization):

- Networks
- Internal Load Balancers¹
- Internal DNS¹
- OpenStack Instances
- RHEL CoreOS
- CoreOS Ignition Configs
- OpenShift Worker Nodes
- OpenShift Cluster Resources



Kuryr:

- Manually enabled (not default)
- Target OSP requires Neutron trunk ports² + Octavia
- Supports OCP network policies
- IP Block support



OVN TECHNICAL PREVIEW

If we are to continue to grow OpenShift as a product, we need to make it easier to grow and improve our networking capabilities, with less investment, while maintaining stability at scale.

OVN has many advantages, including acceleration of customer-driven feature requirements, some of which are pre-enabled:

- Low barrier to integration (implementation of virtual networking via OVS)
- SDN portfolio consolidation / common network technology
- Virtually eliminate iptables scale issues
- Heterogeneous Linux/Windows clusters
- Ability to have a cluster network that spans on-prem nodes and cloud nodes
- Full Network Policy Support
- Egress IP per pod
- Distributed L4 Ingress/Egress firewall
- Distributed services LB
- Multi-Network/Interface
- Heterogeneous clusters w/ Windows nodes
- Capability to span on-prem & cloud nodes
- Traffic isolation / Multi-tenancy
- DPDK support
- Encrypted tunnels
- IPv6 / DHCPv6
- QoS, Control/Data plane separation
- ...

KUBERNETES CNI PLUG-IN ADDS & ENHANCEMENTS

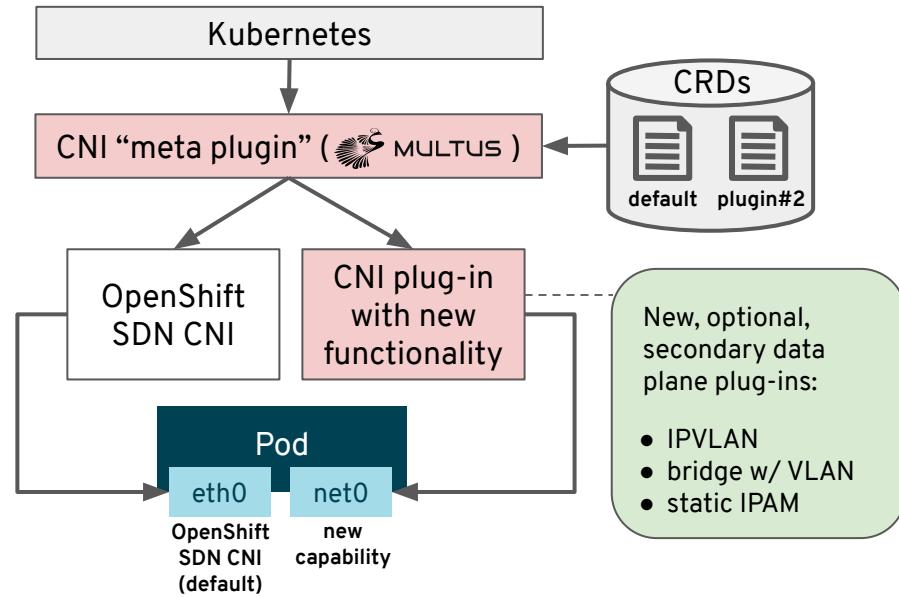
Several Kubernetes CNI Plug-Ins are added or enhanced in OCP 4.2 to grow capability.

SR-IOV Solution:

- Remains Technical Preview at 4.2
- RDMA / RoCE Support
- DPDK Mode for SR-IOV VFs
- Admission Controller
- Operator

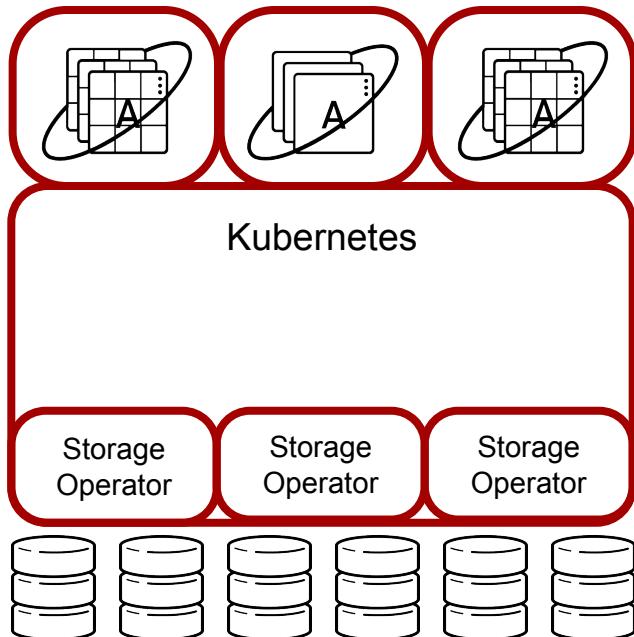
New CNI Plug-Ins:

- IPVLAN
- Bridge with VLAN
- Static IPAM

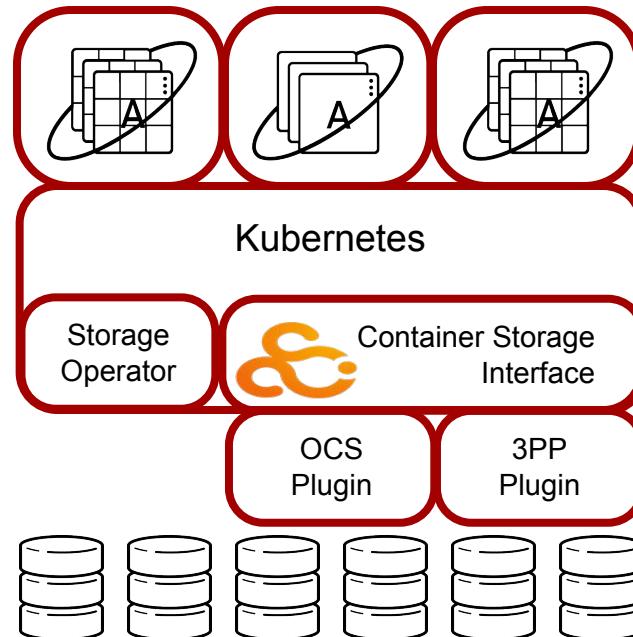


OPENSIFT CSI

4.1



4.2



CSI: ENABLING OCS AND
PARTNERS

STORAGE DEVICES

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, CSI versions to follow later
- New GA storage in 4.2
 - Local Volume
 - Raw Block
 - Cloud providers (AWS, GCP, Azure, vSphere)
 - Local Volume

Supported	
AWS EBS	iSCSI
Azure File & Disk	Fibre Channel
GCE PD	HostPath
VMware vSphere Disk	Local Volume NEW
NFS	Raw Block NEW

OpenShift Container Storage 4.2

Persistent data services for OCP Hybrid Cloud

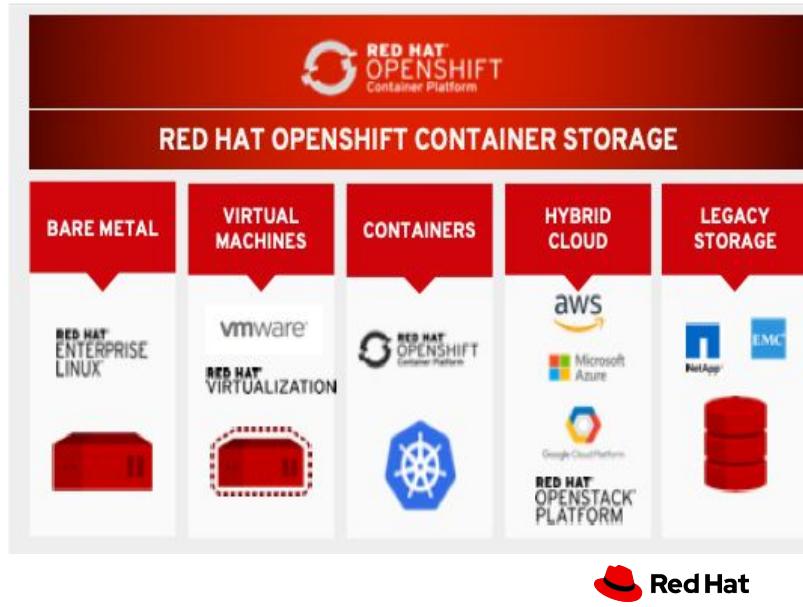
- Complete Data Services: RWO, RWX & **S3(new)** (block, file & object)
- Persistent storage for all OCP Infra and Applications
- Build and deploy anywhere -Consistent Storage Consumption, management, and operations

OCS 4.2 support with OCP 4.2

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

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 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 3.11 to OCP + OCS 4.2**

OCS 3



OCS 4



OCS - Deeper integration with OCP

- **Operator** driven - install, expand, manage,
- Integrated **monitoring** using OCP dashboard
- Integrated **management** - Operator driven expand
- Integrated Prometheus metrics, telemetry, must-gather, alerts

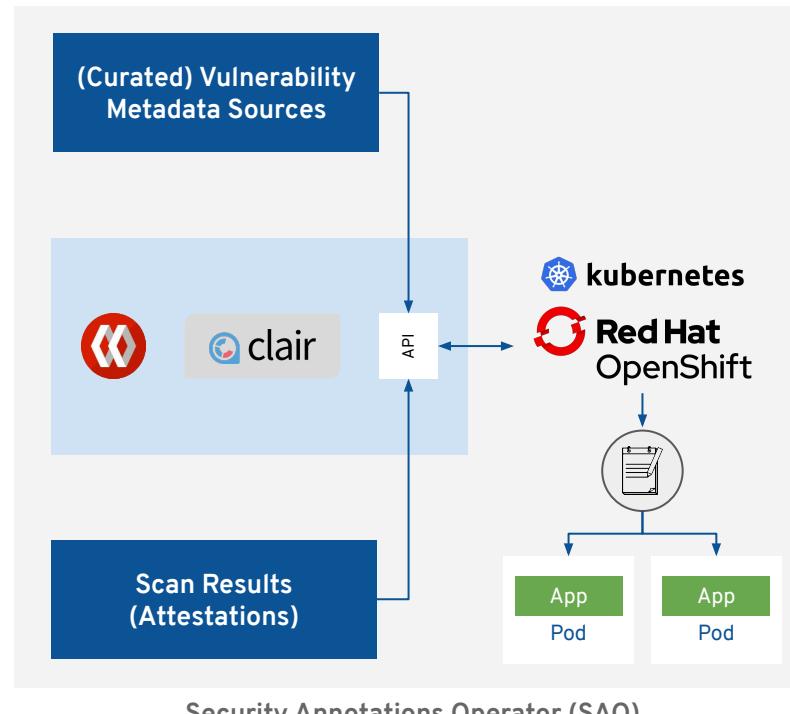
The screenshot shows the Red Hat OpenShift Container Platform dashboard. On the left, there's a sidebar with various navigation options like Home, Dashboards, Projects, Search, Explore, Operators, Workloads, Serverless, Networking, Storage, Builds, Monitoring, Compute, and Administration. The Storage section is expanded, showing Persistent Volumes, Persistent Volume Claims, Storage Classes, and a sub-section for AWS. The main content area has tabs for Dashboards (Overview, Persistent Storage, Object Service), Cluster Details (Cluster ID: 31947d-0000-0000-0000-000000000000, Version: 4.2.0-0.ci-2019-08-20-173300), Cluster Inventory (6 Nodes, 423 Pods, 6 PVcs, VMs, 3 Bare-Metal Hosts), and Cluster Health (multiple errors, cluster health is degraded). Two specific sections are circled in red: 'Persistent Storage' and 'Object Service'. Below the dashboard, the text 'Health, Capacity Configuration, Alerts' is displayed.

Progress towards making the OCP + OCS as one integrated container platform

Red Hat Quay v3.2 and beyond

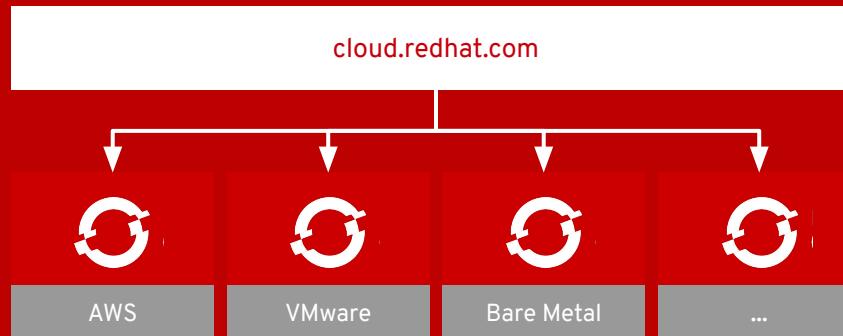
Features targeted for Quay v3.2 (Dec '19)

- Open Sourcing Quay
- Security Annotations Operator (SAO)
- Clair Scan Results shown in OpenShift Console (project, pod, cluster views)
- Repository Mirroring Enhancements
- Quay - OpenShift Integration Operator
- Support for RHOCs4 via NooBaa MCG



Cloud-like Simplicity, Everywhere

Full-stack automated operations across any on-premises,
cloud, or hybrid infrastructure



OpenShift Cluster Manager cloud.redhat.com/openshift

Enhanced OpenShift Web Console Integration

Bi-directional navigation to and from the OpenShift web console for cluster administrators. Deep linking from OCM to the console where relevant.

OpenShift Dedicated cluster management

Self-service cluster provisioning, scaling, and basic management for OpenShift Dedicated customers (4.1+).

Cluster Monitoring

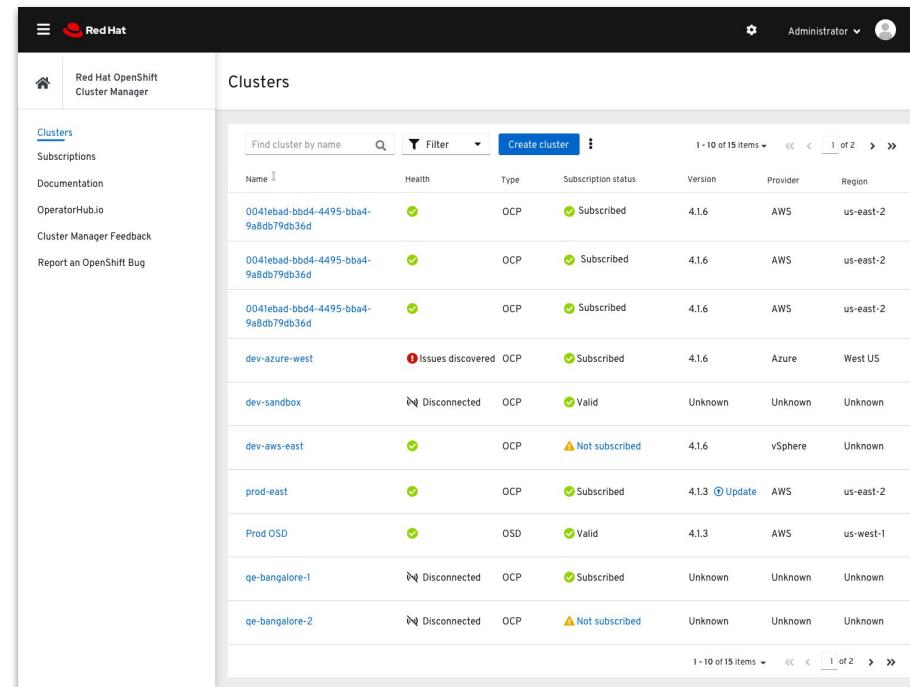
New tab available on all cluster detail pages helps cluster administrators discover critical issues impacting their clusters.

Cluster Updates & New Metrics

OCM provides a link to your cluster's settings page when updates are available for your cluster. Infrastructure provider and region are now captured and displayed for all clusters, where available.

More OpenShift Install Options

New infrastructure providers, including CodeReady Containers, are now listed as install options.



The screenshot shows the Red Hat OpenShift Cluster Manager web interface. The top navigation bar includes the Red Hat logo and the title "Red Hat OpenShift Cluster Manager". On the left, there is a sidebar with links for "Clusters", "Subscriptions", "Documentation", "OperatorHub.io", "Cluster Manager Feedback", and "Report an OpenShift Bug". The main content area is titled "Clusters" and displays a table of 15 items. The columns in the table are: Name, Health, Type, Subscription status, Version, Provider, and Region. The data in the table is as follows:

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
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

Subscription Management cloud.redhat.com/openshift

Disconnected Cluster Registration

Register your disconnected clusters with Red Hat to receive support and updates using the online process.

Subscription Compliance Summary

View subscription compliance status across all of your clusters at a glance.

Public Documentation

Outlining the OpenShift 4 subscription management process, added to the public OCP 4.X docs.

OCP 4 Subscription Management

Product Manager: Jake Lucky

Generally Available

Improved Metrics Collection

RHEL workers, entitled via RHSM, are automatically excluded from remaining entitlement required for your cluster.

Infrastructure Node Labeling

Nodes labeled with the infra node role will be excluded from entitlement required for your cluster.

Subscriptions

Entitlement status

Properly entitled

Subscription amount required

45 vCPU

Support level

Premium

Subscription manager ID

de278845-2ed3-4a60-b4b5-a7ba6c47c615

Operating system

Red Hat Enterprise Linux CoreOS

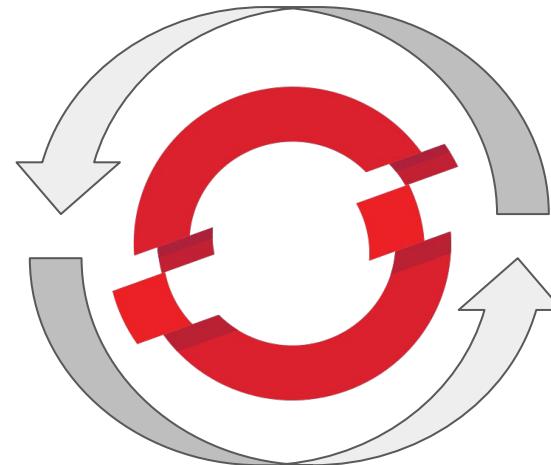
[Manage subscriptions](#)



More visibility for OpenShift Components

Our goal is to provide a 360° view for Openshift

- In every release, we are increasing the number of OpenShift components monitored out of the box.
- If you don't see a component being yet monitored, please reach out to the respective engineering team.
- More than 30 OpenShift-related services we monitor w/ OCP 4.2 (compared to ~5-8 in 3.11).



Configure tolerations for monitoring components

Allow customers to deploy Prometheus and other components of the cluster monitoring stack on tainted nodes.

.

1. Edit the cluster-monitoring-config ConfigMap:

```
$ oc -n openshift-monitoring edit configmap cluster-monitoring-config
```

2. Specify tolerations for the component:

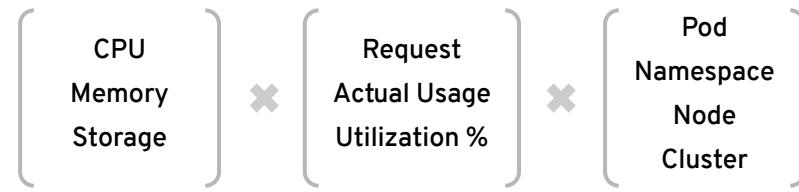
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cluster-monitoring-config
  namespace: openshift-monitoring
data:
  config.yaml: |
    prometheus:
      nodeSelector:
        foo: bar
      tolerations:
        - key: "key1"
          operator: "Equal"
          value: "value1"
          effect: "NoSchedule"
```

Metering

ShowBack/ChargeBack Reports available from OperatorHub

- Base functionality on all providers
- Tie into cloud providers for \$\$
- Included reports for 80% use-case
- Customers can write custom reports and time periods
- Offer basic UI reporting but main use is to plug into customer's BI tool of choice

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



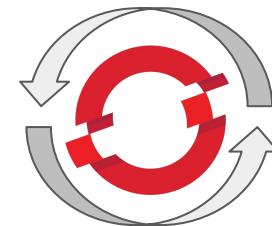
Cluster Logging

Overall performance improvements in collecting logs in 4.x w/ fluentd

- Tripled the amount of logs we collect and at the same time reduced resource consumption by ~50%

3.11	4.1
750 - 1000 1Kb messages/second/fluentd over 10 namespaces at ~600Mb RSS memory used and 75-85% utilization of a single core.	2250 - 2500 1Kb messages/second/fluentd over 10 namespaces at ~325Mb RSS memory used and 35% utilization of a single core.

Be notified when something is wrong with Fluentd with new alerting rules in OCP 4.2.



OpenShift Console

The future is now.

Extending the
Console

Improve
Observability

Administration
made easy

Scaling your
Cluster

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
- Cluster Utilization
- Top Consumers

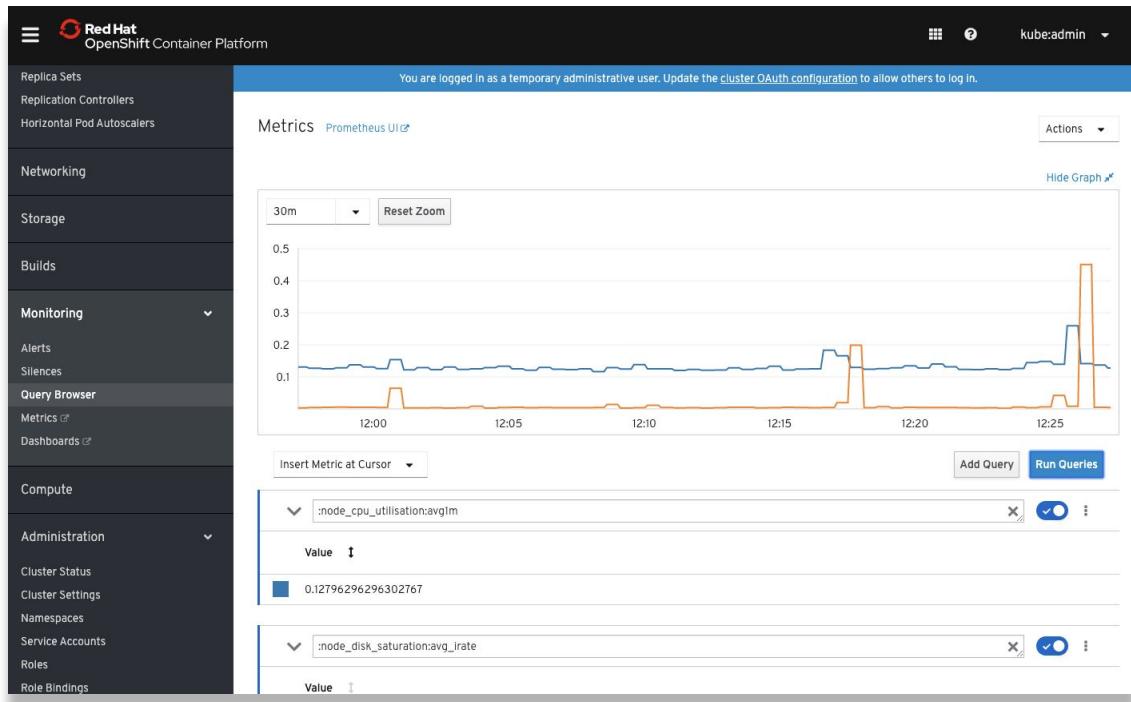
The screenshot shows the OKD Cluster-wide Dashboard interface. The left sidebar has a dark theme with white text and includes sections for Home, Dashboards, Projects, Search, Events, Operators, Workloads, Networking, Storage, Builds, Monitoring, Compute, and Administration. The main content area has a light blue header bar with the text "You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in." Below this is a "Dashboards" section with tabs for Overview, Persistent Storage, and Object Service. The Overview tab is active and displays several cards: "Cluster ID" (47cf4c68-9022-482b-8d31-6cb0faa3e4c2), "Provider" (AWS), "OpenShift Version" (4.2.0-0.ci-2019-07-24-105737), "Cluster Inventory" (6 Nodes, 203 Pods, 3 PVCs, 2 VMs, 0 Bare Metal Hosts), "Cluster Health" (warning: Multiple errors, Cluster health is degraded), "Alerts" (multiple alerts about throttling and StatefulSets), and "Cluster Capacity" (CPU: 14% used, Memory: 5% used, Storage: No Data, Network: 0% used). To the right, there are "Events" (with a "View all" link) and a "Top Consumers" section with dropdowns for "Pods" and "By CPU". The top right corner shows the user is "kubeadmin" and has a "Logout" button.

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.

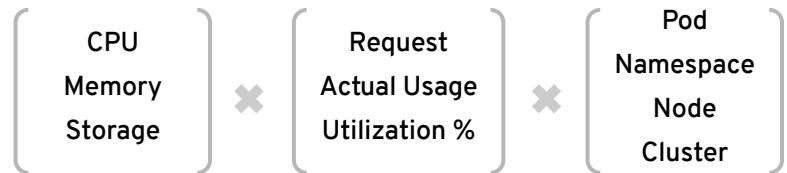


ChargeBack Reports for metering

Managing your Chargeback Reports inside the Console

- Chargeback>Showback** - Break down the reserved and utilized resources requested by applications.
- Pod, Namespace, Node & Cluster Reports** - Built in reports exist to break down CPU, RAM and Storage in any way you desire.
- 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. Use this to send reminder emails, integrate into your ERP system, or graph on a dashboard.

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

Operator UI Improvements for Operator Lifecycle Manager (OLM)

Streamlined navigation and Operator management flows

New OLM UI provides a user-focussed, task-oriented approach when it comes to surfacing the OLM concepts to a user.

The screenshot displays two views of the Red Hat OpenShift Container Platform interface, illustrating the improvements in the Operator UI.

Left View (Streamlined Navigation):

- The top navigation bar includes the Red Hat logo, project dropdown (Project: testing), and user dropdown (kube:admin).
- The left sidebar has a dark theme with a vertical navigation menu:
 - Home
 - Operators** (selected)
 - OperatorHub
 - Installed Operators** (selected)
 - Workloads
 - Networking
 - Storage
 - Builds
 - Monitoring
 - Compute
 - Administration

Right View (Operator management flows):

- The top navigation bar includes the Red Hat logo, project dropdown (Project: testing), and user dropdown (kube:admin).
- The main content area is titled "Installed Operators". It shows a table of installed operators:

Name	Namespace	Deployment	Status	Provided APIs
Couchbase Operator	NS testing	couchbase-operator	InstallSucceeded Up to date	Couchbase Cluster
Dynatrace OneAgent				
etcd				
MongoDB				
Crunchy Postgres Cluster				
- A modal window is open over the main content, titled "Installed Operators > Operator Details" for the "Couchbase Operator". The modal contains tabs for Overview, YAML, Subscription (which is selected), Events, and Couchbase Cluster.
- The bottom of the right view shows a "Subscription Overview" section with tabs for Channel (preview), Approval (Automatic), and Upgrade Status (Up to date, 0 installing).

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

The screenshot shows the Red Hat OpenShift Container Platform web interface. On the left, there's a navigation sidebar with categories like Workloads, Serverless, Networking, Storage, Builds, Monitoring, Compute, and Administration. Under Administration, there are sub-options: Cluster Status, Cluster Settings, Namespaces, Service Accounts, Roles, Role Bindings, Resource Quotas, Limit Ranges, Chargeback, API Explorer, and Custom Resource Definitions. A dropdown menu is open over the 'Administration' section, listing 'Add' (with a dropdown arrow), followed by 'Basic Authentication', 'GitHub', 'GitLab', 'Google', 'HTPasswd', 'Keystone', 'LDAP', 'OpenID Connect', and 'Request Header'. To the right of the sidebar, there are two main content areas. The top area is titled 'Cluster' and has sections for Labels (No labels), Annotations (1 Annotation), and Created At (Jul 10, 7:08 am). It also contains a message: 'You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.' The bottom area is titled 'Identity Providers' and has a sub-section: 'Identity providers determine how users log into the cluster.' Below this, there's a form for adding a new identity provider. The form title is 'Add Identity Provider: Keystone Authentication'. It includes fields for 'Name' (keystone), 'Domain Name', 'URL', 'CA File', 'Certificate' (with a 'Browse...' button), and 'Key' (with a 'Browse...' button). At the bottom of the form are 'Add' and 'Cancel' buttons.

Knative on OpenShift Serverless via Knative Operators

Build and deploy Serverless applications using an event-driven infrastructure on OpenShift

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

Top Screenshot (OperatorHub):

- Left Sidebar:** Shows navigation links for Home, Operators (selected), OperatorHub (selected), Installed Operators, Workloads, and Networking.
- Header:** Red Hat logo and "OpenShift Container Platform".
- Project:** testing
- OperatorHub Section:** Title "OperatorHub". Subtext: "Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. Operators can be installed on your clusters to provide optional add-ons and shared services to your developers. Once installed, the capabilities provided by the Operator appear in the [Developer Catalog](#), providing a self-service experience." Filter bar shows "knative" (highlighted). Buttons: "All Items" (selected), "All Items", and "5 items". Categories: AI/Machine Learning, Application Monitoring, Application Runtime, Big Data.

Bottom Screenshot (Serverless):

- Left Sidebar:** Shows navigation links for Home, Operators, Workloads, Serverless (selected), Services, Revisions (selected), Configurations, Networking, and Storage.
- Header:** Red Hat logo and "OpenShift Container Platform".
- Project:** openshift-operators
- Revisions Section:** Title "Revisions", "Create Revision" button, table with columns: Name, Namespace, Service. One row shown: helloworld-nodejs-cxdq, NAMESPACE, helloworld-nodejs.
- Community Operators Section:** Four cards:
 - Knative Apache Camel Operator** (Community): Provided by Red Hat. Description: "The Knative Camel addon provides a collection of eventing sources from the".
 - Knative Apache Kafka Operator** (Community): Provided by Red Hat. Description: "Knative Eventing Kafka manages the Kafka source and channel provisioner fo".
 - Knative Serving Operator** (Community): Provided by Red Hat. Description: "Knative Serving builds on Kubernetes to support deploying and serving of serverless applications an".
 - TriggerMesh** (Community): Provided by TriggerMesh, Inc. Description: "A serverless management platform that runs on Knative. TriggerMesh provides continuous delivery of".

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 shows the Red Hat OpenShift Container Platform web interface. The left sidebar has a dark theme with the following navigation items:

- Projects
- Search
- Events
- Operators
- Workloads
- Networking
- Storage
- Builds
- Monitoring
- Compute
 - Nodes
 - Machines
 - Machine Sets
 - Machine Autoscalers** (highlighted)
 - Machine Configs
 - Machine Config Pools

The main content area has a header bar with the Red Hat logo and "Red Hat OpenShift Container Platform". It displays a message: "You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in." The project dropdown is set to "openshift-metering". On the right, there is a "Import YAML" button and a search bar labeled "Filter by name...".

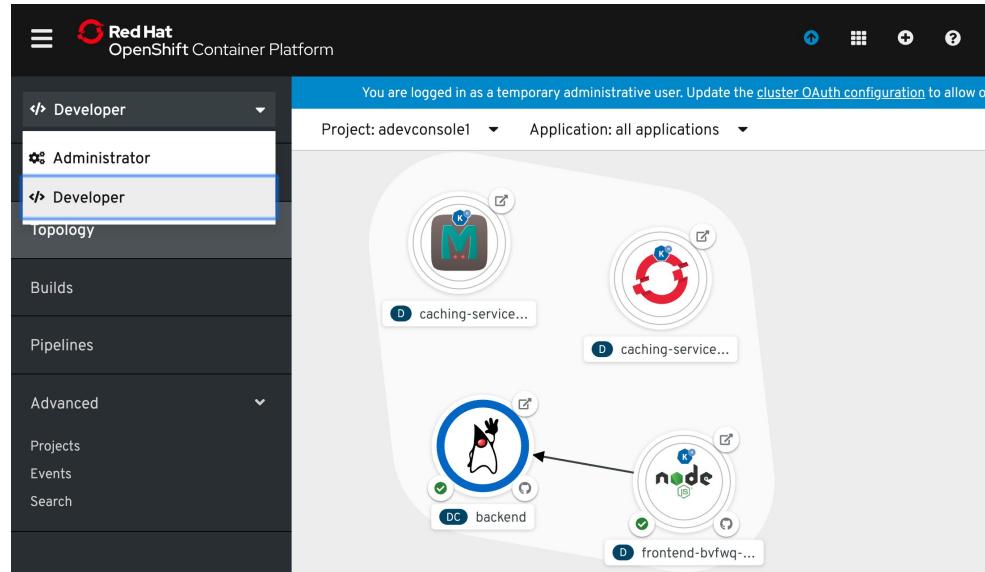
The central part of the screen is titled "Machine Autoscalers" and contains a table with the following data:

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

Web Console - Developer Perspective

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

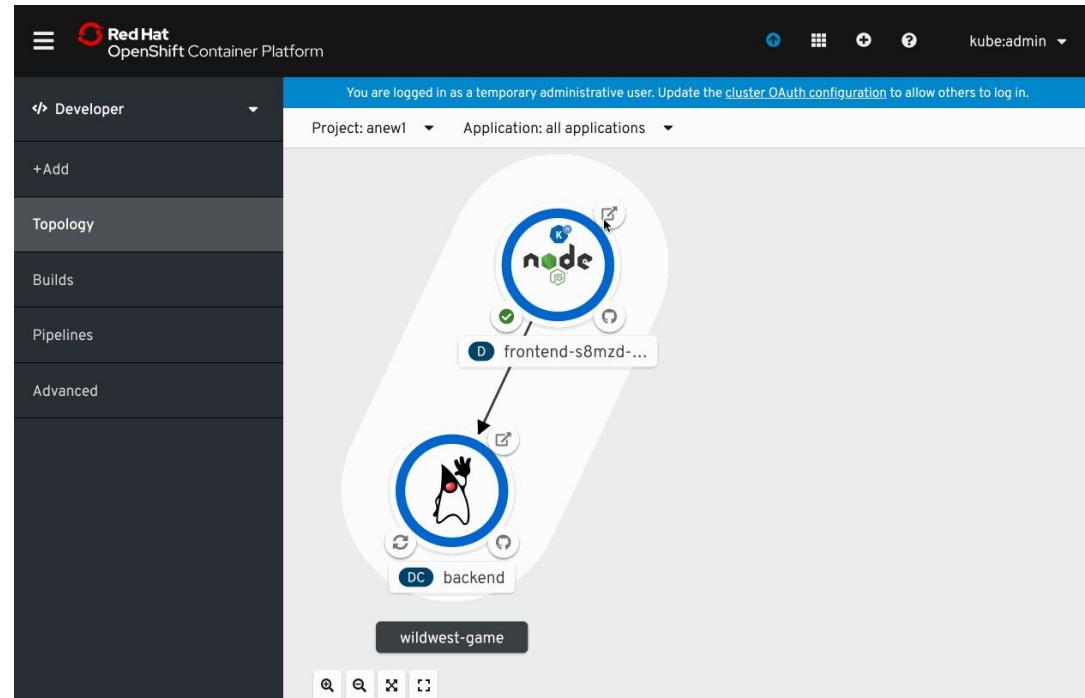
All OpenShift developer tool UIs will be surfaced here...though some (like CodeReady Workspaces) will be links out to unique UIs.



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



Next wave of developer tools

OpenShift has all of the latest tools to make
your devs more productive



CodeReady Workspaces

The collaborative OpenShift-Native IDE. Free for any customer of OpenShift Dedicated or OpenShift Container Platform.

Container Workspaces



Workspace replicas to end “works on my machine” and enable team collaboration.

DevOps Integrations



Reference developer workspaces from any issue, failed build, or git notification.

Protect Source Code

Full access to source code without any of it landing on hard-to-secure laptops.

Based on the open Eclipse Che project

Red Hat Linux and Application Infrastructure

Plugin model for extensibility

Serverless support (coming later)

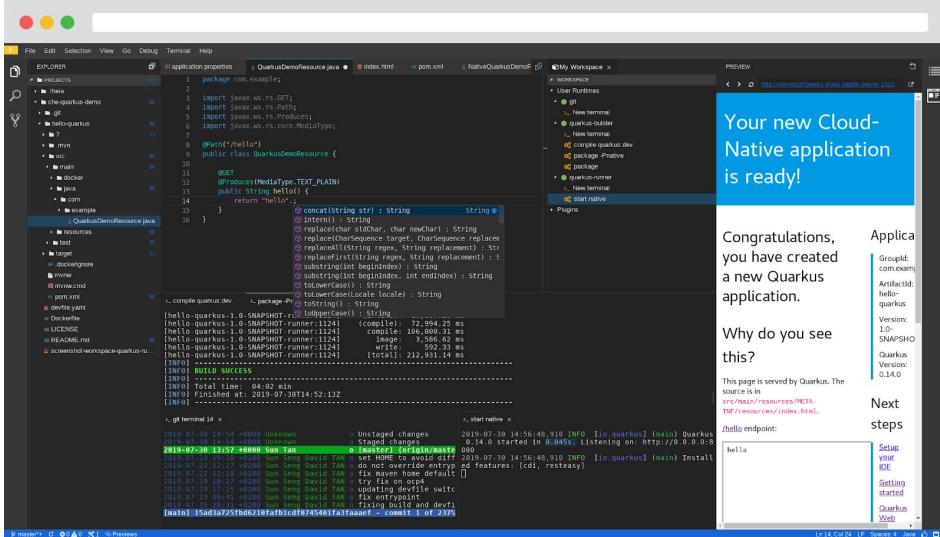
Use It To: Replace VDI for devs, and enable true container-based DevOps.

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



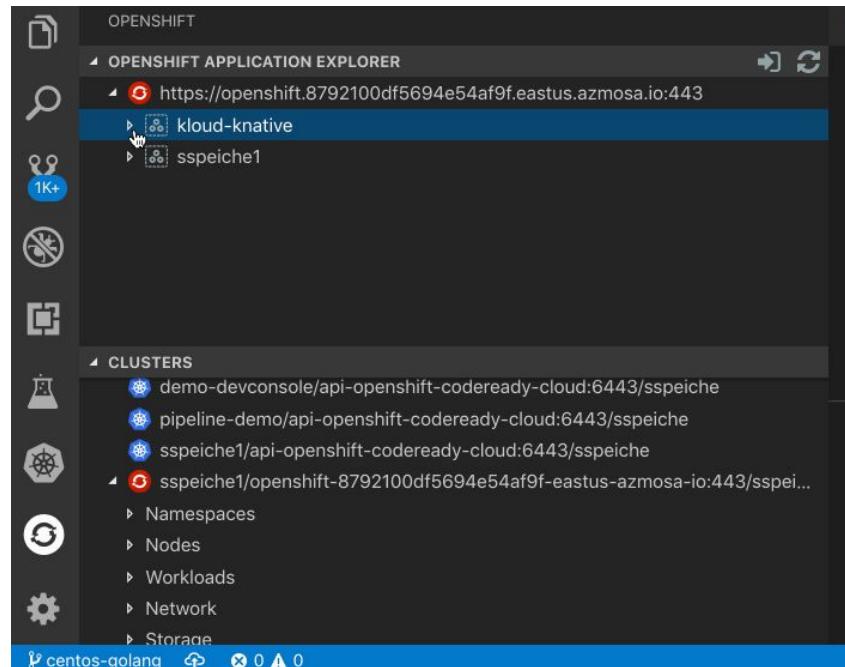
VS Code Kubernetes Extension

Kubernetes Extension Improvements

- Collaboration spearheaded by Red Hat and OpenShift needs
- Many improvements around:
 - Non-cluster-admin use cases
 - Auto-hide Helm features when no Tiller installed
 - Add nodes to navigator

OpenShift Improvements

- OpenShift logo on OpenShift clusters
- Add: Routes, DeploymentConfig, Projects, ImageStreams
- Ability to set Project context



odo - OpenShift's 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.

NEXT WAVE OF DEVELOPER TOOLS



OpenShift Serverless

Key Features

- Familiar to Kubernetes users. Native.
- Scale to 0 and autoscale to N based on demand
- Applications and functions. Any container workload.
- Powerful eventing model with multiple event sources.
- Operator available via OperatorHub
- Knative v0.7.1 (v1beta1 APIs)
- No vendor lock in

Learn more

<https://openshift.com/learn/topics/serverless>

<https://redhat-developer-demos.github.io/knative-tutorial>

The screenshot shows the Red Hat OpenShift Container Platform interface. On the left, the navigation sidebar is open, showing options like Home, Dashboards, Projects, Search, Explore, Events, Operators (selected), Workloads, Serverless, Services, Revisions, Routes, and Networking. The main content area displays the 'Installed Operators' section under 'Operators'. A card for the 'Serverless Operator' is shown, indicating it is version 1.0.0 provided by Red Hat. Below this, a detailed view of a deployment named 'spring-petclinic-bchpw-deployment' is presented. The deployment summary shows a scaling factor of 4 out of 10. The deployment details include:

Name	spring-petclinic-bchpw-deployment	Update Strategy	RollingUpdate
Namespace	markito-rhte	Max Unavailable	25% of 10 pods
Labels	app=spring-petclinic-bchpw, app.kubernetes.io/...=springbilo..., app.kubernetes.io/l...=spring-pe..., serving.knative.dev/configurat...=1, serving.knat...=bb3d3ed9f4cb1..., serving.knative.dev/...=spring-pe..., serving.knative.dev/...=spring-pe..., serving.knative.dev/...=spring-pe...	Max Surge	25% greater than 10 pods
		Progress Deadline	2m 0s
		Min Ready Seconds	Not Configured

The deployment status indicates that 4 pods are scaling to 10. The deployment resources tab shows the current state of the deployment, with 4 pods running and 0 pending. The deployment logs tab shows the latest log entries for the deployment.

NEXT WAVE OF DEVELOPER TOOLS

Cloud-native CI/CD with OpenShift Pipelines

- Based on Tekton Pipelines
- Runs serverless (no babysitting!)
- Containers as building blocks
- Deploy to multiple platforms
- Standard CRDs
- Build images with Kubernetes tools
(s2i, buildah, kaniko, jib, buildpack, etc)
- Pipelines portable to any Kubernetes
- Available in OperatorHub

The screenshot shows the Red Hat OpenShift Container Platform web interface. On the left, a sidebar menu includes 'Developer', '+ Add', 'Topology', 'Builds', 'Pipelines', and 'Advanced'. The main area is titled 'Project: Project01' and shows a 'Pipelines' section with a pipeline run named 'PR pipelinerun01a' in the 'Running' state. The 'Overview' tab is selected, displaying a pipeline run overview with five sequential steps: 'code compile', 'compile & test', 'unit test', 'image build', and 'security check'. Each step has a green checkmark icon. Below the steps, the pipeline run details are listed: Name (pipelinerun01a), Namespace (NS project01), Labels (app=dummy-mongo-pod-test, bap.me/environment=dev, bap.me/track=experimental, bap.me/tier=backend), Annotations (0 Annotations), and Created At (2018-08-23T18:28:53Z). The 'Logs' tab is also visible at the bottom, showing log output for the 'image build' step. The log output includes several error messages related to core errors before and after the eval hook, such as 'hook.core.error:before in (eval 306)', 'Plack::Sandbox::2f0pt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsg:54] core @2018-08-23 18:28:53> looking [Plack::Sandbox::2f0pt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsg:54] core @2018-08-23 18:28:53> looking [Plack::Sandbox::2f0pt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsg:54] core @2018-08-23 18:28:53> Entering hook.core.error:init in (eval 306)', and 'Plack::Sandbox::2f0pt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsg:54] core @2018-08-23 18:28:53> Entering hook.core.error:before in (eval 306)'.

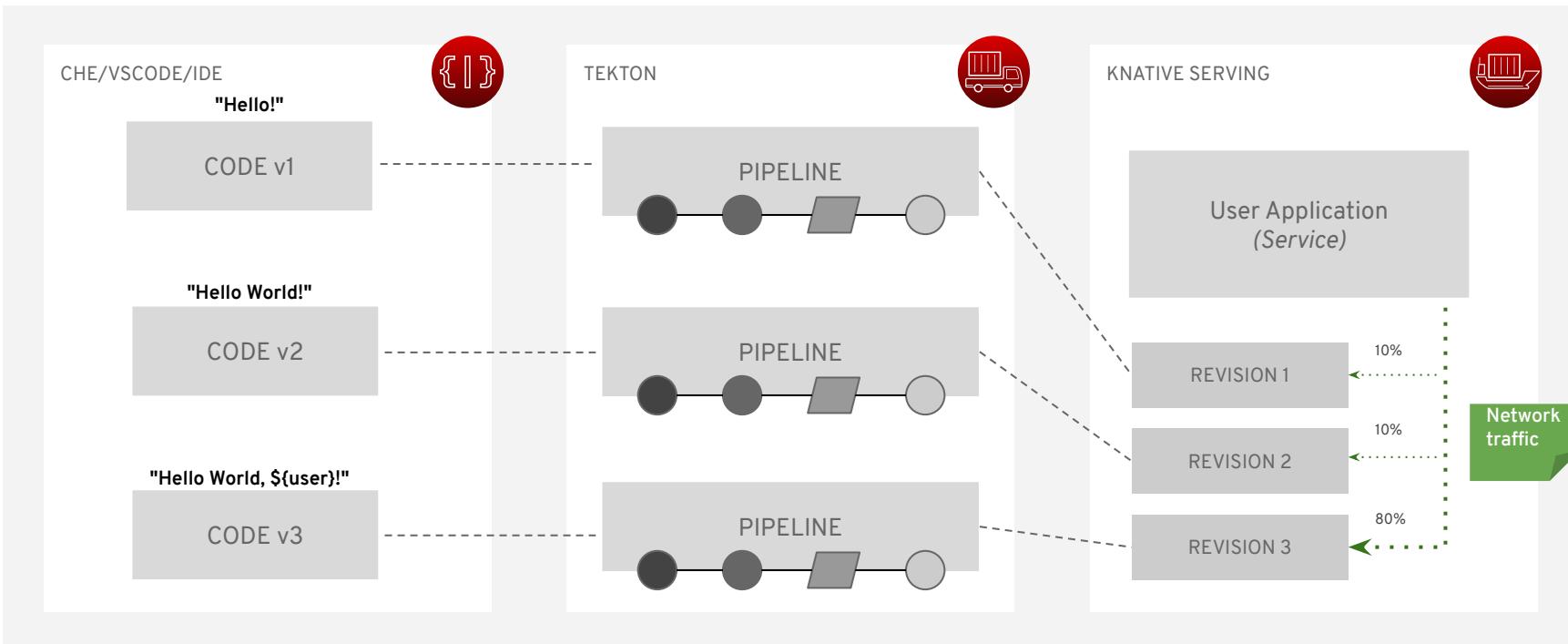
NEXT WAVE OF DEVELOPER TOOLS

OpenShift Serverless - Lifecycle

CODE

BUILD

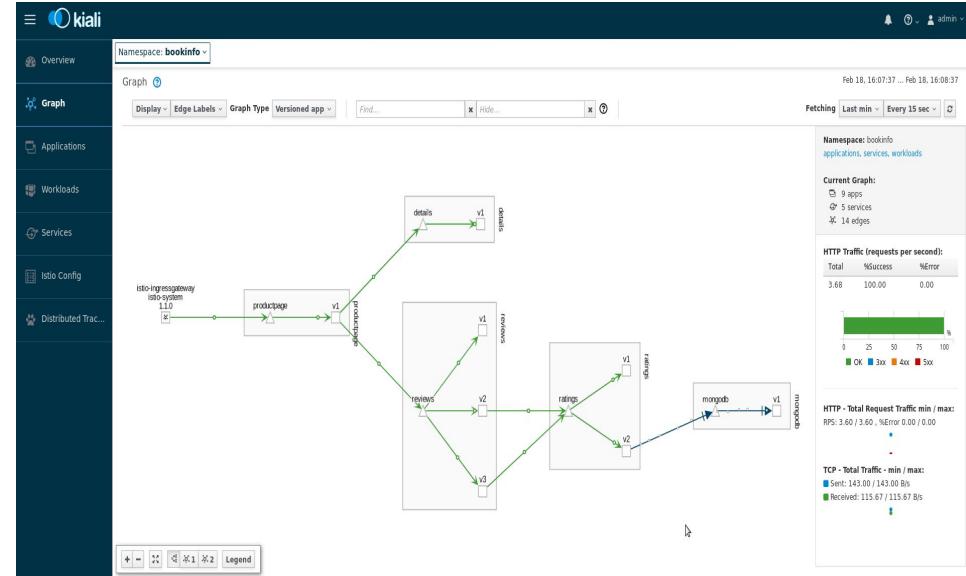
RUN



Red Hat Service Mesh

Key Features

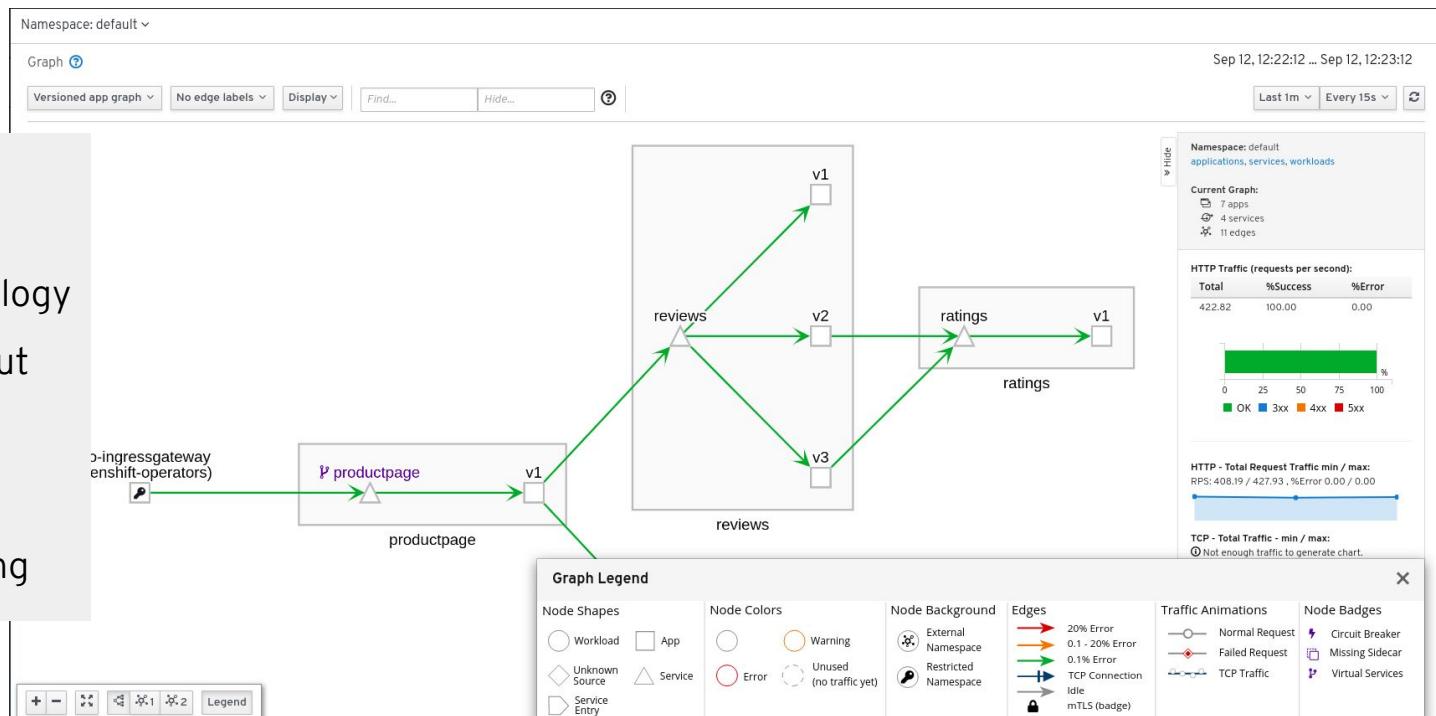
- A dedicated network for service to service communications
- Observability and distributed tracing
- Policy-driven security
- Routing rules & chaos engineering
- Powerful visualization & monitoring
- Will be available via OperatorHub



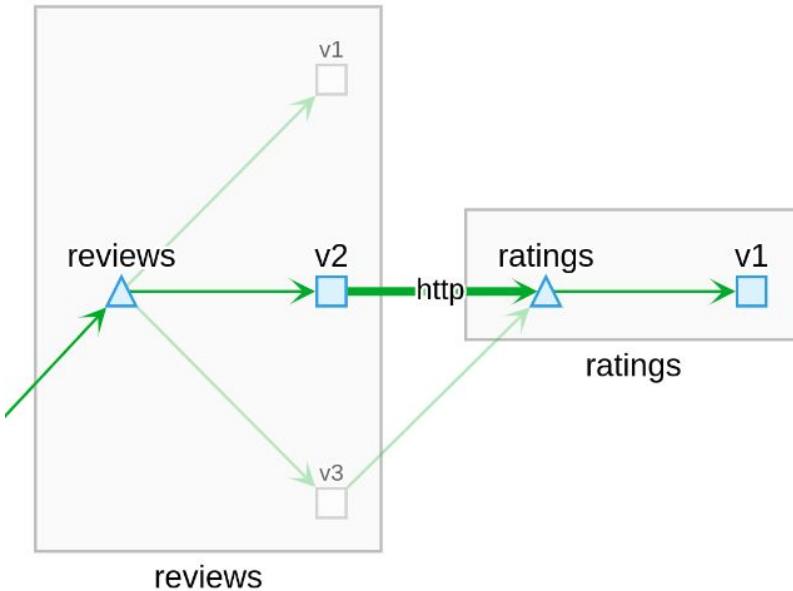
Enhanced Visualization of Cluster Traffic With Kiali

Visualization of what
Matters most:

- Application Topology
- Traffic throughput
- Error Rates
- Service Latency
- Service Versioning



Protocol Specific Analysis and Drill down



Convenient Overviews of Individual Services

Services > Namespace: default > Service: reviews

reviews (Show on graph)

Overview Traffic Inbound Metrics

Last 1m ▾ Actions ▾

Labels
[app](#) [reviews](#) [service](#) [reviews](#)

Selectors

[app](#) [reviews](#)

Type ClusterIP

IP 172.30.17.223

Created at 9/12/2019, 12:06:32 PM

Resource Version 5094640

Ports

TCP http (9080)

Endpoints

10.128.2.27 : reviews-v1-989d5ffdf-w8gmn
10.128.2.28 : reviews-v3-757c4f7849-rs7sw
10.131.0.35 : reviews-v2-6ff8648d69-tlqhn

Health

Healthy

Error Rate over last 1m: 0.00%

Workloads (3) Virtual Services (0) Destination Rules (0)

Name	Type	Labels	Created at	Resource version
reviews-v1	Deployment	app reviews version v1	9/12/2019, 12:06:32 PM	5095051
reviews-v2	Deployment	app reviews version v2	9/12/2019, 12:06:32 PM	5094998
reviews-v3	Deployment	app reviews version v3	9/12/2019, 12:06:32 PM	5095043

Guided Configuration of Traffic Policies

Create Weighted Routing

WORKLOAD	TRAFFIC WEIGHT
 reviews-v1	<input type="range" value="5"/> 5 %
 reviews-v2	<input type="range" value="80"/> 80 %
 reviews-v3	<input type="range" value="15"/> 15 %

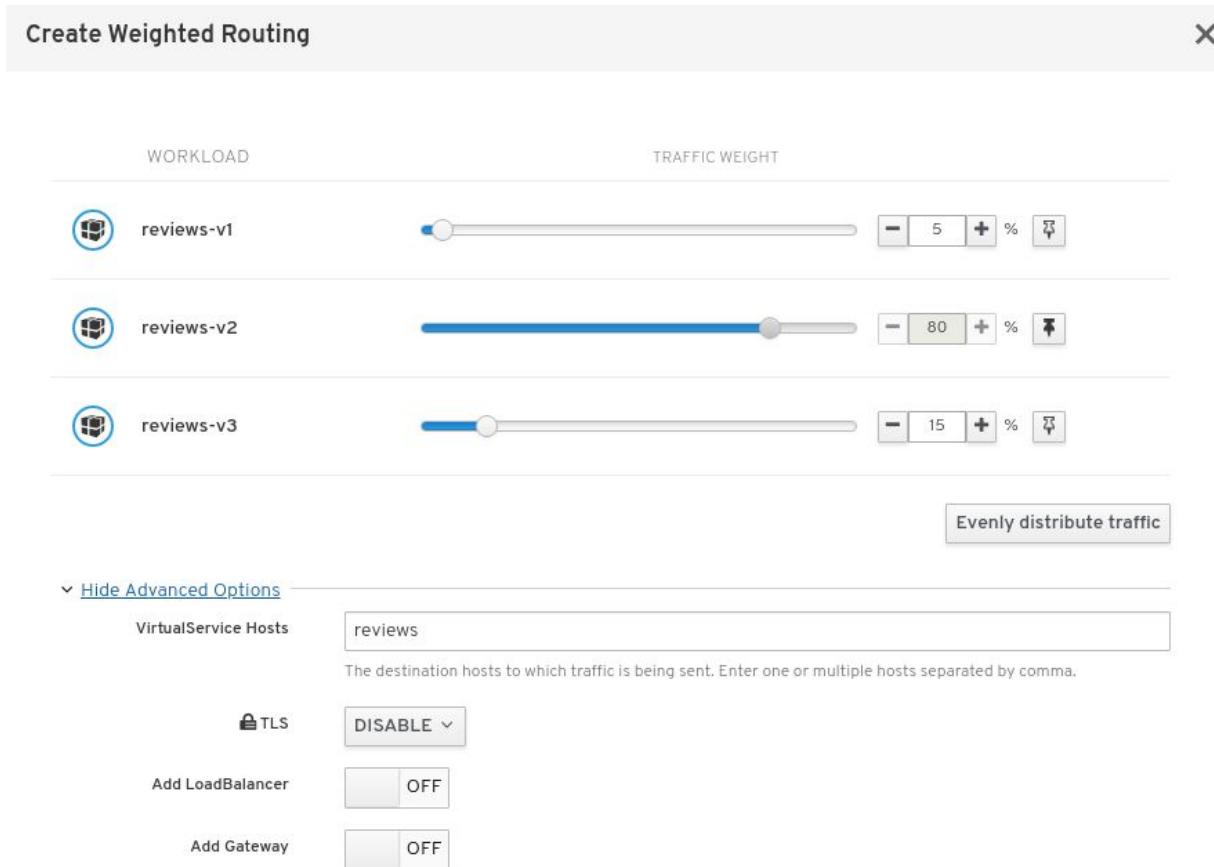
[▼ Hide Advanced Options](#)

VirtualService Hosts The destination hosts to which traffic is being sent. Enter one or more hosts separated by comma.



Add LoadBalancer

Add Gateway



Management of URI Matching for Virtual Services

Istio Config > Namespace: default > Istio Object Type: virtualservices > Istio Object: bookinfo

[Overview](#) [YAML](#)

VirtualService: bookinfo

Created at: 9/12/2019, 12:06:46 PM

Resource Version: 5094784

Hosts

*

Gateways

[bookinfo-gateway](#)

HTTP Route

Match

Uri

[Exact] /productpage

Uri

[Exact] /login

Uri

[Exact] /logout

Uri

[Prefix] /api/v1/products

Status	Destination			Weights
	Host	Subset	Port	
	productpage ↗	-	9080	-

Service Mesh Roadmap

Q4 CY19

Beginning of multi channel releases:

- Tech Preview
- Stable
- LTS (To come later)

Establish baseline for performance improvements

Expanding automated testing matrix

Upgrade stabilization

Migrate Kiali to Patternfly 4

1H CY20

Release of OSM v1.1 (approx Jan 2020)

- Update to Istio 1.3 (Possibly 1.4)
- FIPS-140-2 Certification for core components
- [Telepresence.io](#) Support
- Initial eBPF support
- Additional configuration via Kiali UI
- Multi-cluster mesh support
- Mesh Expansion (Federation of resources *into* the mesh from outside OpenShift)

2H CY20

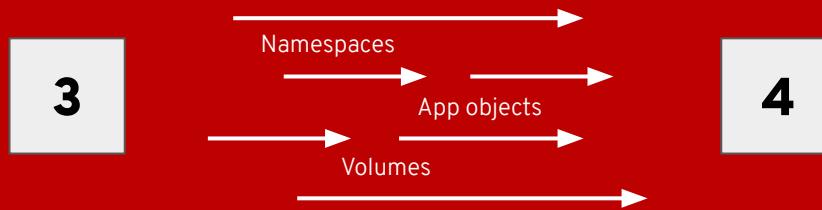
Initial integration with cloud.redhat.com/openshift for cluster management

Additional protocol support

- MySQL
- MongoDB
- Kafka

Migrating to OpenShift 4

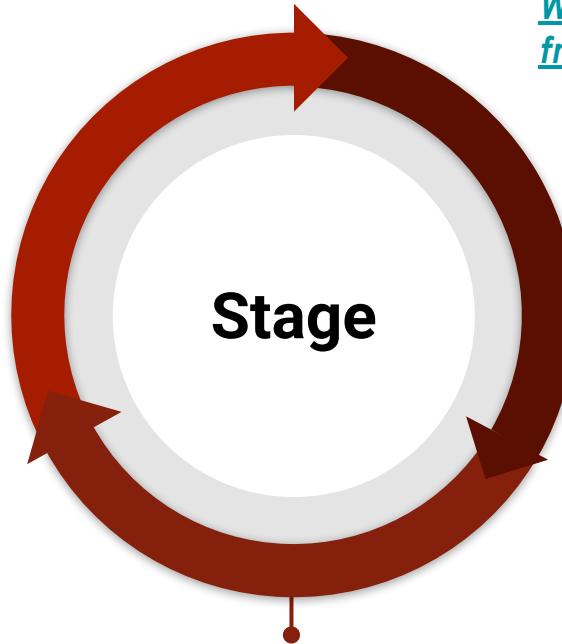
Tooling and advice for moving from OpenShift 3.x to 4.x



Application Migration: Migration Process

Plan

1. Select Source Cluster
2. Select Namespaces
3. Choose Copy or Move for each PV
4. Specify Destination



Stages the data from Source to Destination.

May be run multiple times.
Applications are running
no Downtime during step

[Watch a migration of MS-SQL Server from OCP 3.11 to OCP 4.1!](#)

Migrate

Quiesce Application

Migrate any delta bits not captured in stage.

Granularity of Namespace(s) & ‘cluster-admin’ required

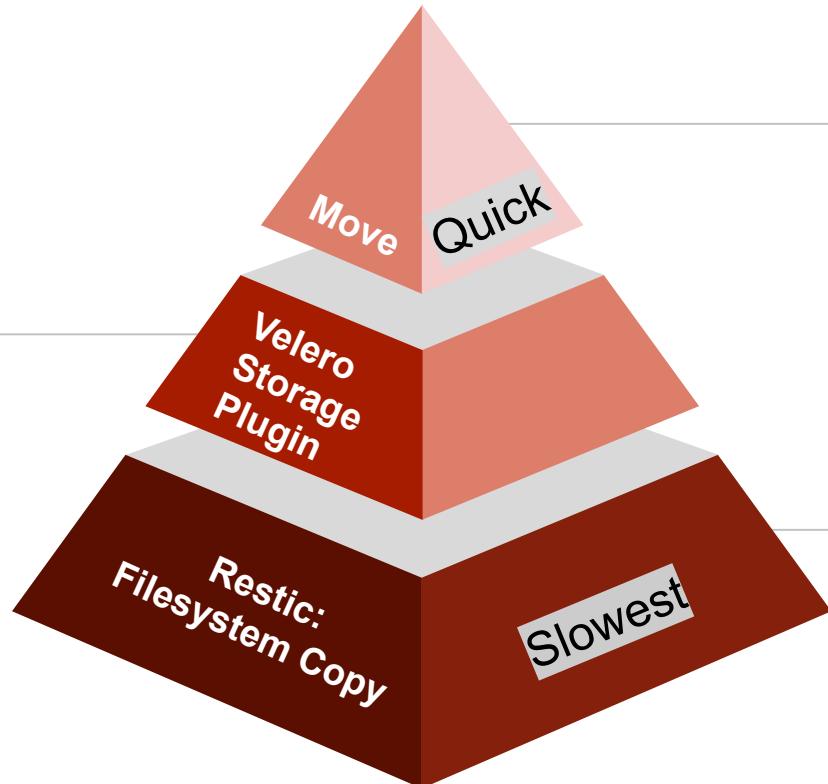
- **Migration is at scope of a Namespace.**
 - *Future will allow selecting resources inside of a Namespace*
- **Cluster Scoped Resources are not handled**
 - Cluster Role Bindings, SCCs, etc are not handled with migration.
 - Expectation is that cluster admin handles cluster scoped resources ahead of running a Migration.
- **‘cluster-admin’ required for initial release targeting OCP 4.2**
 - Future plans to allow end user to migrate what they own post OCP 4.2+

Persistent Volume Handling

Copy: Storage Provider Plugin

(AWS EBS, Google, Azure)

Leverage snapshot support from the storage provider such as Amazon EBS snapshots..



Move “Swing the PV”

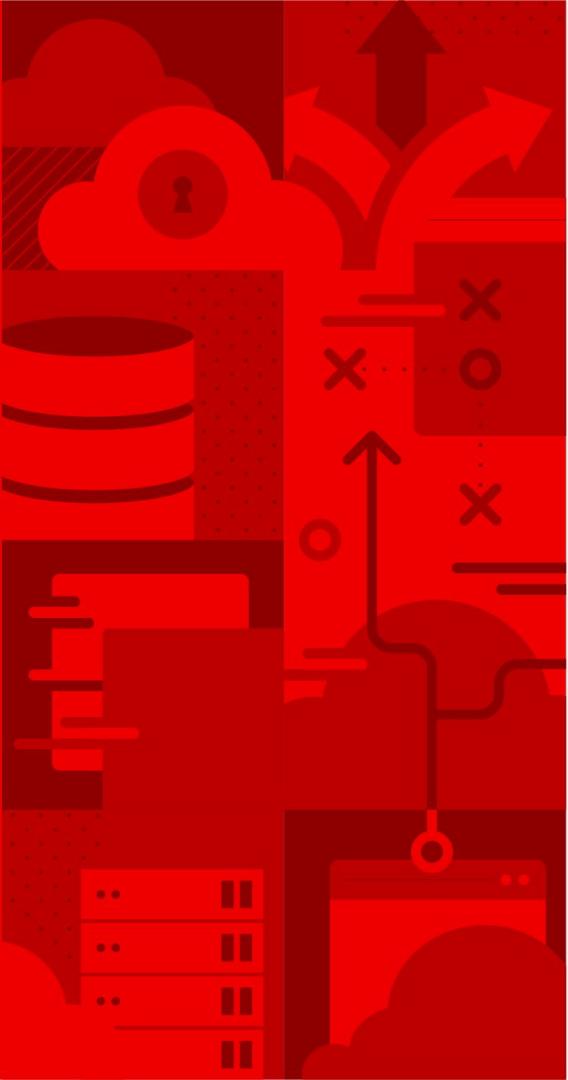
The data remains where it is in the volume. The definition of the volume is essentially ‘swung’ from the source cluster to the destination cluster. Quickest migration strategy as it involves no processing of the data. No rollback capability, recommended to ensure a backup exists prior to moving.

Copy: Restic Filesystem level copy

Catchall, if no other plugins exist fall through to a filesystem level copy of the data on a PV. Most flexible approach, yet likely not as performant as other methods.

2019 Roadmap

Q2 CY2019 OpenShift 4.1		Q3 CY2019 OpenShift 4.2		Q4 CY19/Q1 CY20 OpenShift 4.3	
HOSTED	PLATFORM	HOSTED	PLATFORM	HOSTED	PLATFORM
HOSTED	PLATFORM	HOSTED	PLATFORM	HOSTED	PLATFORM
	DEV		DEV		DEV
<ul style="list-style-type: none">OpenShift Serverless (Knative) - DPOpenShift Pipelines (Tekton) DP2CodeReady WorkspacesCodeReady Containers AlphaDeveloper CLI (odo) Beta		<ul style="list-style-type: none">Developer Console GAOpenShift Serverless (Knative) - TPOpenShift Pipelines (Tekton) DP3CodeReady Containers GADeveloper CLI (odo) GA	<ul style="list-style-type: none">OperatorHub EnhancementsOperator Deployment Field FormsApplication Migration Console	<ul style="list-style-type: none">OpenShift Serverless (Knative) - GAOpenShift Pipelines (Tekton) TPHelm 3 TP	<ul style="list-style-type: none">Metering for ServicesWindows Containers (Planned)GPU MeteringApplication Operator Binding - DP
<ul style="list-style-type: none">Kubernetes 1.13 with CRI-O runtimeRHEL CoreOS, RHEL7Automated Installer for AWSPre-existing Infra Installer for Bare Metal, VMware, AWSAutomated, one-click updatesMultus (Kubernetes multi-network)Quay v3	APP	<ul style="list-style-type: none">Kubernetes 1.14 w/ CRI-O runtimeDisconnected Install and UpdateAutomated Installer for Azure, OSP, GCPPre-existing Infra Installer for GCPCluster-wide Egress ProxyOVN Tech PreviewOpenShift Container Storage 4.2 (1 month after)	<ul style="list-style-type: none">Insights OperatorAzure Red Hat OpenShift new features (monitoring, logging)	<ul style="list-style-type: none">Kubernetes 1.16 w/ CRI-O runtimeAutomated Installer for RHVPrivate/Internal Clusters support from the installerDeploy to pre-existing VPC & SubnetsOVN GA w/ Windows Networking Integration (Planned)FIPSPre-existing Infra Installer for OSPOpenShift Container Storage 4.3	<ul style="list-style-type: none">cloud.redhat.com - Multi-Cluster MgmtOCP Cluster Subscription ManagementOpenShift Dedicated consumption pricing



Questions?

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

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

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

 twitter.com/RedHat