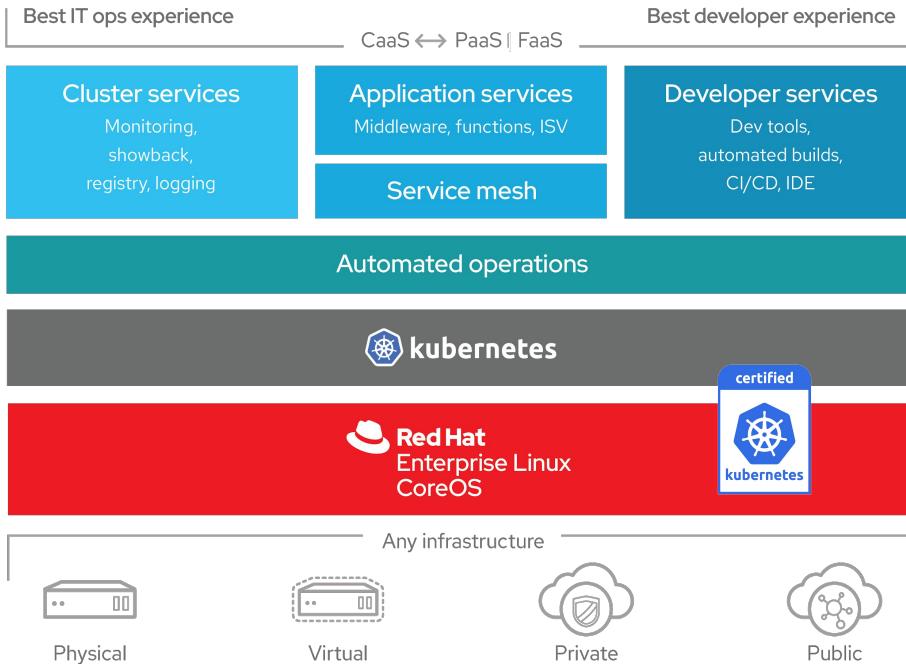




What's New in OpenShift 4.3

Alfred Bach
Partner Enablement Manager Cloud



OpenShift 4 Platform

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

OpenShift 4.3

INSTALLER CUSTOMIZATION



Improvements for disconnected
Internal facing/private clusters
Customer provisioned
VPC/VNet/etc and subnets

SECURITY & COMPLIANCE



FIPS validated crypto
Disk encryption for RHCOS
Encrypted etcd datastore
Kubernetes 1.16

IMPROVED NETWORKING



High performance multicast to
clients outside cluster
SR-IOV graduates to GA
Additional monitoring for OVN

Install & Upgrades

4.3 Supported Providers

Full Stack Automation (IPI)



Microsoft Azure



RED HAT[®]
OPENSTACK[®]
PLATFORM

Pre-existing Infrastructure (UPI)



Microsoft Azure*



IBM Z*



Bare Metal

* Support planned for an upcoming 4.3 z-stream release

Generally Available



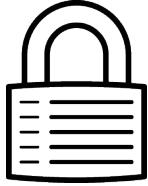
Provider Roadmap & Minimum Supported Version

Provider	Full Stack Automation (Installer provisioned infra)	Pre-existing Infrastructure (User provisioned infra)
 amazon web services	4.1	4.1
 Microsoft Azure	4.2	4.3+ (z-stream)
 Bare Metal	4.4 (TBD)	4.1
 Google Cloud Platform	4.2	4.2
 RED HAT OPENSTACK PLATFORM	4.2	4.4
 RED HAT VIRTUALIZATION	4.4	4.4
 VMware vSphere	4.4	4.1
 IBM Z	-	4.2+ (z-stream)
 IBM Power Systems	-	4.3+ (z-stream)
 Alibaba Cloud	4.5	-

Security

Stronger Platform Security

Defense in Depth



CONTROL Application Security

- [FIPS Compliance](#)
- [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 FIPS 140-2 Compliant Cluster

FIPS ready Services

- When built with RHEL 7 base image

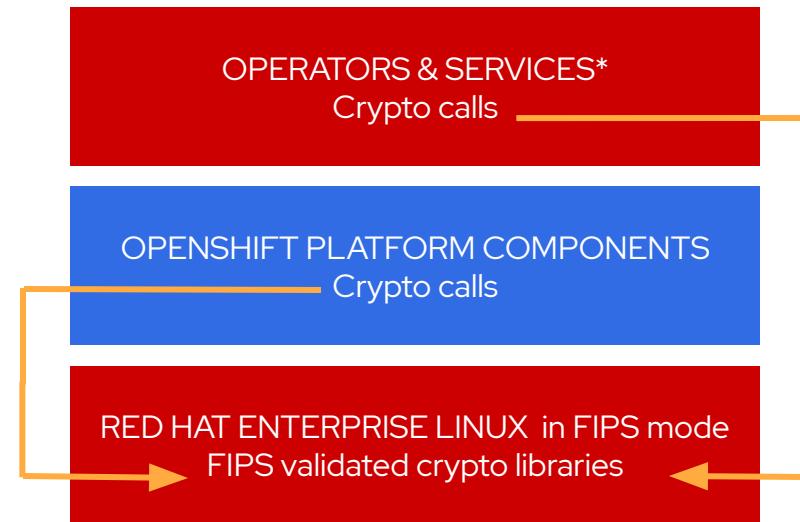
OpenShift calls FIPS validated crypto

- When running on RHEL 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 FIPS validated ciphers for node-level cryptography

Note: products are not FIPS validated, only libraries.



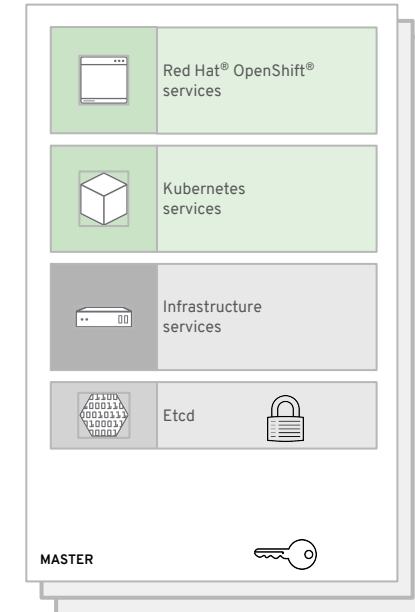
*When built with RHEL base images

[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.



RHEL CoreOS

Red Hat Enterprise Linux CoreOS

4.3 Image Availability: (* = new)

- OpenStack
- GCP
- Azure
- Amazon
- vSphere
- Bare Metal (unified x86_64 image)*
- IBM Z (DASD & FCP via z-stream)*



FIPS mode support:

- Enforces FIPS validated ciphers for node-level cryptography
- Configurable at install/provisioning

Network Bound Disk Encryption:

- Provides encryption for local storage
- Addresses disk/image theft
- Platform/cloud agnostic implementation
- TPM/vTPM (v2) and Tang endpoints for automatic decryption

Kmods via containers:

- A framework to build and load 3rd party kmods
- Viable for drivers unsuitable for the SRO

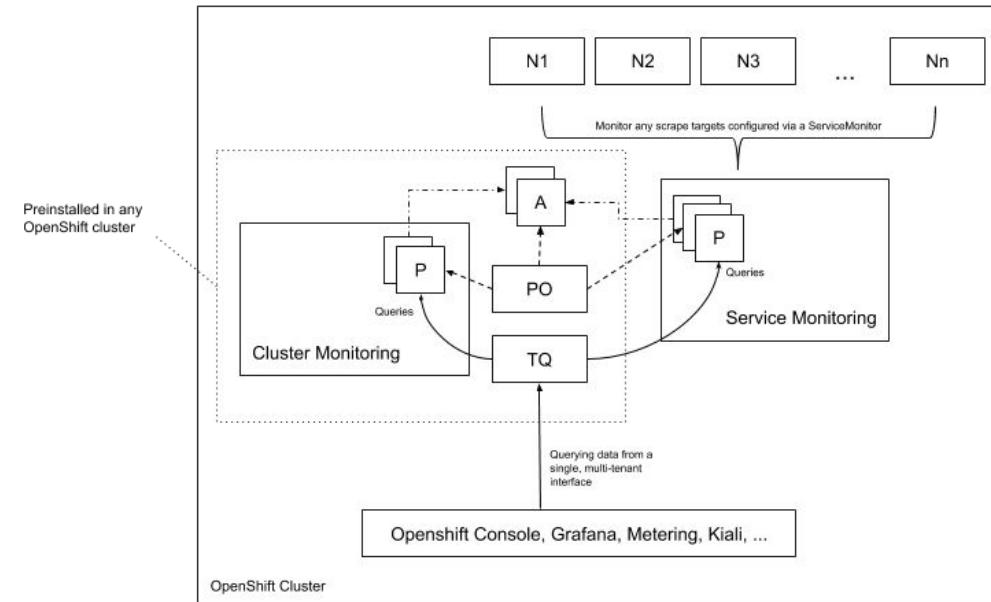
Cluster Monitoring & Logging

Monitoring your own services

Extend existing stack to configure monitoring for any service running on OpenShift.

Goals for this milestone are:

- Feedback!
- Enable additional Prometheus servers that your customers own, but are managed by us.
- Configure monitoring for your business critical services not covered by the out-of-the-box monitoring stack.
- Access metrics through a single, multi-tenant interface.
- Maintain notifications in a centralized Alertmanager setup.
- Developers can query metrics through the developer perspective.

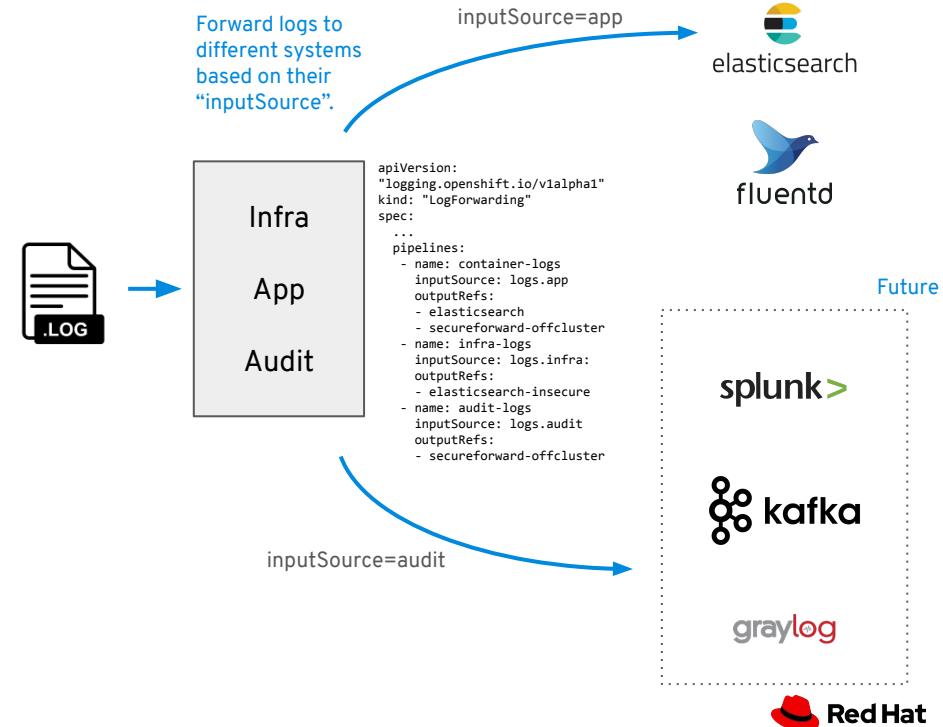


Log forwarding

Foundation for a new ability to configure forwarding logs to various logging systems.

Goals for this milestone are:

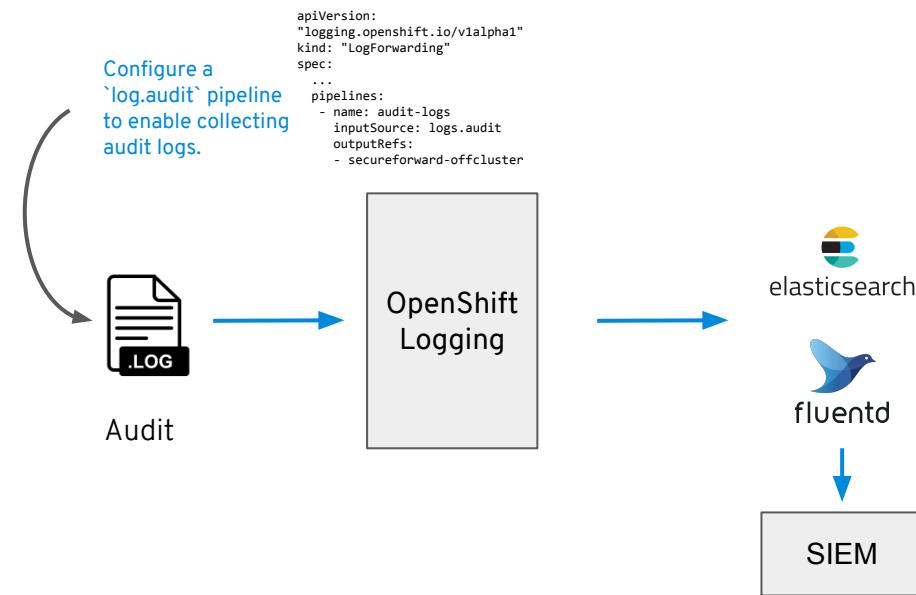
- Feedback!
- Expose singleton LogForwarding CRD to configure OpenShift Logging to forward logs to an external system.
- Supported systems: Elasticsearch and another fluentd via secure-forward.
- Allow deployment of OpenShift Logging without deploying the entirety infrastructure (e.g. Kibana, Elasticsearch)
- Support TLS between the collector and destination if so configured.



Collect audit logs through log forwarding

Collect and forward audit logs to external systems.

- Configure `logs.audit` pipeline to enable a new ability to collect audit logs and to setup and external system you'd like to forward them.
- Either use your own Elasticsearch or your own fluentd via secure-forward as in previous OCP releases; where you can send them to any SIEM system.



Networking and Routing

Support for PTP

- The Precision Time Protocol (PTP) is used to synchronize clocks in a network. When used in conjunction with a timestamping-capable NIC (for incoming and outgoing packets), PTP is capable of sub-microsecond accuracy.
- Requires cluster-admin privileges.
- Installed via the PTP Operator, the protocol implementation is included in the linuxptp package.
- Configuration:
 - Install the PTP Operator, then configure the linuxptp services (ptp4l, phc2sys) by creating the PtpConfig CR.
 - The creation, deletion and update of PtpConfig will be detected by the ptp-operator which updates a per-cluster configmap in openshift-ptp namespace.
 - The linuxptp daemon running on each node reads the configmap change and applies to linuxptp services.
 - PTP's config is applied to a specific hardware NIC interface, and may require a per-node/per-interface configuration (depends on the NIC layout and network connections on each Node).

```
apiVersion: ptp.openshift.io/v1
kind: PtpConfig
metadata:
  name: <name>
  namespace: openshift-ptp
spec:
  profile:
    - name: "profile1"
      interface: "enp134s0f0"
      ptp4lOpts: "-s -2"
      phc2sysOpts: "-a -r"
  recommend:
    - profile: "profile1"
      priority: 4
      match:
        - nodeLabel: "node-role.kubernetes.io/worker="
          nodeName: "worker-1"
```

IP & MAC Address Mgmt for Multus Interfaces

- For applications requiring predictable and recreatable IP address management for Multus *secondary-onward* interfaces.
- The IP address will be associated with a unique Pod ID, so that whenever a pod gets evacuated or re-created, the fixed IP address will always be allocated and assigned to that particular pod interface.
- A user may also request for an IP address to be assigned to an interface according to a predefined mapping list of pod interfaces and IP addresses.
- Provides capability for:
 - Static IP address assignment (no-DHCP env)
 - Static MAC address assignment (DHCP)
 - Multus default route override (no-DHCP env)

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: my-runtimeconfig-def
spec:
  config: '{
    "cniVersion": "0.3.1",
    "plugins": [
      {
        "type": "macvlan",
        "capabilities": { "ips": true },
        "master": "eth1",
        "mode": "bridge",
        "ipam": {
          "type": "static"
        }
      }
    ]
  }'
```

```
apiVersion: v1
kind: Pod
metadata:
  name: my-runtimeconfig
annotations:
  k8s.v1.cni.cncf.io/networks: '[{"name": "my-runtimeconfig-def", "ips": [ "10.1.1.104/24" ]}]'
...
...
```

Generally Available

DNS Forwarding

- Many customers have OpenShift clusters running in the public cloud and connected to their datacenter by VPN. They need to resolve some DNS requests (e.g. other internal devices and services) using the DNS servers in their datacenter.
- OpenShift 4 uses CoreDNS, which has the capability to forward those datacenter DNS requests for configured domains.

```
$ oc edit dns.operator/default
spec:
  servers:
    - forwardPlugin:
        upstreams:
          - "<DNS1_server_IP>"
          - "<DNS2_server_IP>"
      name: test
    zones:
      - foo.com
      - bar.com

Validate the forwarder:

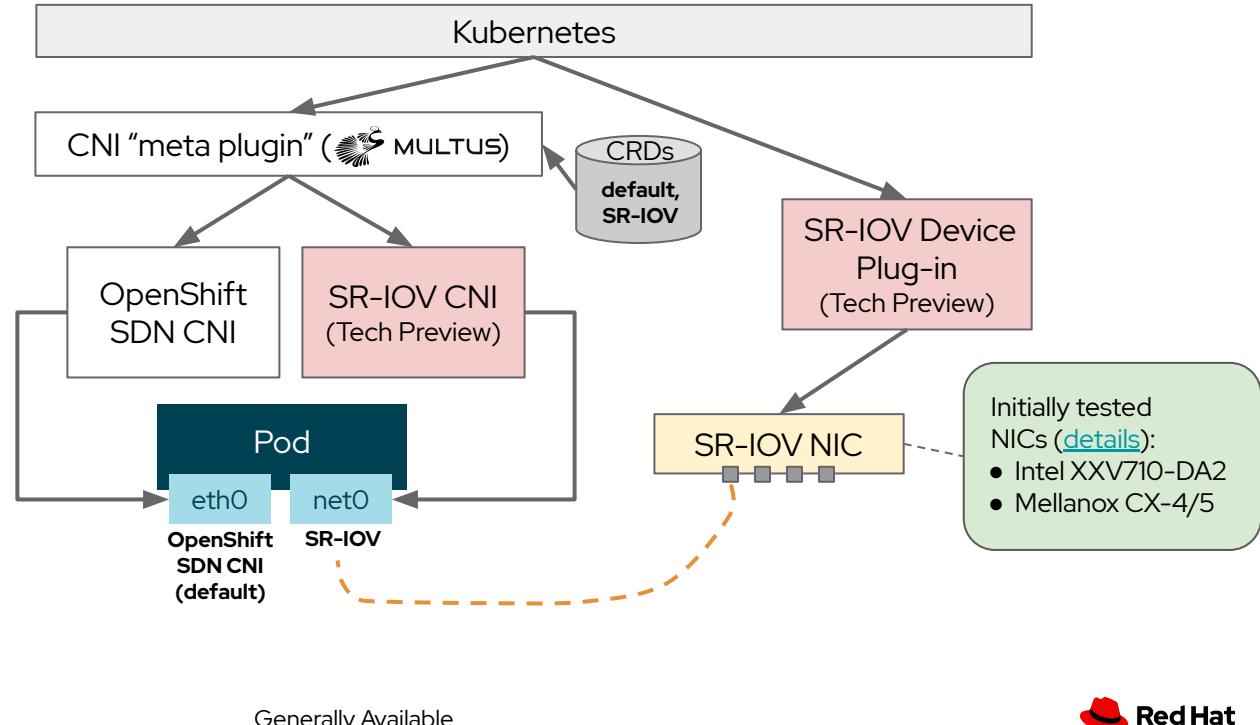
$ oc get cm dns-default -n openshift-dns -o yaml
data:
  Corefile: |
    # test
    foo.com:5353 bar.com:5353 {
      forward . "<DNS1_server_IP>" "<DNS2_server_IP>"
    }
```

Generally Available

SR-IOV High-Performance Networking

New for OCP 4.3:

- Admission Controller
- Metrics and Alerting
- Telemetry
- Topology Mgr support
- CNI runtimeConfig support for MAC and IP
- Create pod with static MAC & IP addresses by SR-IOV CNI
- SR-IOV VF Security and QoS Flags
- Available in OperatorHub



High-Performance Multicast

- Streaming media applications like ITV and multipoint videoconferencing require high-bandwidth multicast capability.
- Prior to 4.3, OpenShift SDN supported multicast between pods on the default network for low-bandwidth coordination or service discovery – it was not a high-bandwidth solution.
- In OpenShift 4.3, high-performance multicast is fully supported using SR-IOV to provide near-native networking performance.
- Multicast packets must be sent or received by a pod through the SR-IOV interface.
- The physical network which connects the SR-IOV interfaces decides the multicast routing and topology, and is not controlled by OpenShift.
- [Configuration](#) requires cluster-admin role.

1) Create an `SriovNetworkNodePolicy` CR to init the VF (brief excerpt shown):

```
spec:  
  resourceName: example  
  nodeSelector:  
    feature.node.kubernetes.io/network-sriov.capable: "true"  
  numVfs: 4  
  nicSelector:  
    vendor: "8086"  
    pfName: ['ens803f0']  
    rootDevices: ['0000:86:00.0']
```

2) Create an `SriovNetwork` CR to create a net-attach-def (brief excerpt shown):

```
spec:  
  networkNamespace: default  
  ipam: |  
  {  
    "type": "host-local", (1)  
    "subnet": "10.56.217.0/24",  
    "rangeStart": "10.56.217.171",  
    "rangeEnd": "10.56.217.181",  
    "routes": [  
      {"dst": "224.0.0.0/5"},  
      {"dst": "232.0.0.0/5"}  
    ],  
    "gateway": "10.56.217.1"  
  }
```

3) Create a pod with the multicast application....

Storage

Storage Devices

Continued improvements

- iSCSI support to GA
- Raw block support additions
 - Raw block with iSCSI to GA
 - Raw Block with Cinder to Tech Preview
- CSI
 - Ember driver Tech Preview
- Continued focus on partner enablement

OCP Supported		
AWS EBS	Fibre Channel	
Azure File & Disk	HostPath	
GCE PD	Local Volume	
VMware vSphere Disk	Raw Block	IMPROVED
NFS	iSCSI	NEW
Supported via OCS		
File , Block, Raw Block, Object		
Supported via OSP		
Cinder		

CSI Partner Certification Pipeline

In
Test

- CSI Ember driver in testing soon (tech preview for 4.3)
- CSI Manila and Cinder being investigated



Pilot



- Initial pilot of certification process
- Normal operator process plus test suite and TOI

Initial
Talks

- In discussions

Initial
Talks

- In discussions

Re-
connect

- Re-connecting to this opportunity

OpenShift Container Storage 4.2

GA with OpenShift Container Platform 4.3



Portability

Seamless data placement and access across clouds

Multi Cloud Data Portability/Hybrid Cloud with S3

Consistent set of management tools across clouds

AWS (UPI + IPI), VMware (UPI)



Simplicity

Operator driven install, upgrade, expand through OLM

Integrated OCP + OCS monitoring and management

Dynamic provisioning of persistent volumes for RWX, RWO, S3 in Converged Mode



Scalability

Support Traditional and Emerging OCP Workloads

Easily share data across geo-locations and platforms

5,000 PV's in a 10 node setup

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 'Project Details' page for the 'tony' project. The top navigation bar includes the Red Hat logo, 'OpenShift Container Platform', and user information ('kube:admin'). The left sidebar has a 'Projects' section with options like Home, Dashboards, Search, Explore, Events, Operators, Workloads, Networking, Storage, Builds, Monitoring, Compute, User Management, and Administration. The main content area has tabs for Dashboard, Overview, YAML, Workloads, and Role Bindings. The 'Dashboard' tab is selected, showing sections for Details, Status, Inventory, Utilization, and Activity.

Details

- Name: tony
- Requester: kube:admin
- Labels: No labels

Status

- Active

Inventory

- 4 Deployments
- 4 Pods
- 0 PVCs
- 1 Service
- 0 Routes
- 4 Config Maps
- 21 Secrets

Utilization

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

Activity

No project messages

Recent Events

- 16:00 P Successfully as... >
- 15:09 CSV install strate... >
- 15:08 P Started contain... >
- 15:08 P Created contai... >
- 15:08 P Successfully pul... >
- 15:07 CSV install strate... >
- 15:07 P Successfully pul... >

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.

The screenshot displays two views of the Red Hat OpenShift Container Platform interface. The top view shows the 'Installed Operators' page under the 'Operators' menu. It lists 'AMQ Streams' as an installed operator, with details including Namespace (tony), Deployment (amq-streams-), Status (InstallSucceeded), and Provided APIs (Kafka). The bottom view shows a project-scoped interface for the 'tony' project, also under the 'Operators' menu. This view shows the 'tony' operator with status 'Active'. Both views include a sidebar with various navigation options like Home, Dashboards, Projects, and Compute.

"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.

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 shows the Red Hat OpenShift Container Platform interface. On the left, a sidebar menu lists various workloads: Operators, OperatorHub, Installed Operators, Workloads (Pods, Deployments, Deployment Configs, Stateful Sets, Secrets, Config Maps, Cron Jobs, Jobs, Daemon Sets, Replica Sets, Replication Controllers, Horizontal Pod Autoscalers, Serverless, Networking), and a 'Custom Resource Definitions' section. The 'Jobs' option under 'Workloads' is selected. The main content area shows a 'Create Job' screen with a YAML editor containing a sample Job configuration. A tooltip indicates that the 'Samples' tab is selected. Below this, a modal window displays the 'Job' CRD details, specifically the 'Samples' tab, which contains an example Job YAML sample and a 'Try it' button. At the bottom, the 'consoleyamlsamples.console.openshift.io' CRD details page is shown, listing an instance named 'example' with its creation timestamp.

```
apiVersion: batch/v1
kind: Job
metadata:
  name: example
  namespace: brie
spec:
  selector: {}
  template:
    metadata:
      name: pi
    spec:
      containers:
        - name: pi
          image: perl
          command:
            - perl
            - "-Mbignum=bpi"
            - "-wle"
            - "print bpi(2000)"
  restartPolicy: Never
```

Custom Resource Definitions > Custom Resource Definition Details
CRD consoleyamlsamples.console.openshift.io

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

The screenshot shows the Red Hat OpenShift Container Platform dashboard with a sidebar menu on the left. The sidebar includes options like Home, Dashboards, Projects, Search, Explore, Events, Operators, OperatorHub, Installed Operators, Workloads, Pods, Deployments, Deployment Configs, Stateful Sets, Secrets, Config Maps, Cron Jobs, and Logs. A blue arrow points from the 'Dashboards' option in the sidebar to the 'Dashboards' section of the main content area. The main content area displays a 'Dashboards' overview with tabs for 'Overview' (selected), 'Details', and 'Status'. The 'Overview' tab shows a summary of vulnerabilities: 61 total, 14 High-level, 33 Medium-level, and 14 Low-level. It also lists recent events: 'Nov 7, 12:09 am' (A client in the cluster is using deprecated apps/v1beta2 API that will be removed soon.) and 'Nov 7, 12:09 am' (A client in the cluster is using deprecated apps/v1beta2 API that will be removed soon.). The 'Status' tab shows the status of the Cluster (green), Control Plane (green), and Image Security (red). Below the overview, there is a 'Vulnerabilities' table with columns: CVE, SEVERITY, PACKAGE, CURRENT VERSION, FIXED IN VERSION, and INTRODUCED IN LAYER. The table lists several vulnerabilities, such as RHSA-2019-0710, RHSA-2019-1587, RHSA-2019-0368, RHSA-2019-0049, RHSA-2019-0679, RHSA-2018-2285, and RHSA-2018-7184, along with their respective package names (python-lbs, python-lbs, systemd-lbs, systemd-lbs, libssh2, yum-plugin-novl, openssh), current versions, fixed versions, and introduction layers. To the right of the main content area, there is a 'Security breakdown' panel with a circular chart showing the distribution of vulnerabilities by severity (High, Medium, Low) and a list of fixable vulnerabilities for the 'openssl-libs' package across 1 namespace, with log entries for successful assignments.

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 illustrates the new User Management section in the OpenShift 4.3 console. It features a navigation sidebar on the left and two main content panels on the right.

Navigation Sidebar:

- Groups
- Projects
- Search
- Explore
- Events
- Operators >
- Workloads >
- Networking >
- Storage >
- Builds >
- Monitoring >
- Compute >
- User Management > (highlighted)
- Users
- Groups (highlighted)
- Service Accounts
- Roles
- Role Bindings
- Administration >

Groups Panel:

Create Group

Name	Users
G admins	2
G app_devs	2

Users Panel:

You are logged in as Administrator

Users are automatically added the first time they log in.

Name
U developer
U user

Context Menu (Impersonate User "user"):

- Impersonate User "user"
- Edit Labels
- Edit Annotations
- Edit User
- 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 screenshot displays the Red Hat OpenShift Container Platform web interface. On the left, a sidebar menu includes options like Home, Operators, Workloads, Networking, Storage, Builds, Monitoring, Compute, User Management, Administration (with Cluster Settings, Namespaces, Resource Quotas, Limit Ranges, and Custom Resource Definitions), Configuration Resources (APIServer, Alertmanager, Authentication, Build, ClusterVersion, Console, DNS, FeatureGate), and Global Configuration. The main content area shows the 'Alerting' section under the 'Alerting' tab. It features a 'Create Receiver' dialog box. The 'Receiver Name' field is set to 'my-new-receiver'. The 'Receiver Type' dropdown is set to 'PagerDuty'. Under 'PagerDuty Configuration', the 'Integration Type' is selected as 'Events API v2'. The 'Routing Key' field contains the value 'thisis sometext that will blur very soon'. The 'Routing Labels' section allows setting alerts with specific labels. A table lists a single entry: 'severity' with 'warning' as its value. There are buttons for 'Create' and 'Cancel' at the bottom of the dialog.

Deploy Applications streamlining flows

Deploy Image from Internal Registry

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

You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow...

Project: abc Application: all applications

Deploy Image

Image

Deploy an existing image from an image stream or image registry.

Image name from external registry

Image Name *

openshift/hello-openshift

repository, you must create an image pull secret with your image

65981
evangelist
70023

Auto-detect builder image

- Recommends builder images based on detected language by git provider

Import from git

Git

Git Repo URL *

https://github.com/scrlorg/nodejs-ex

Validated

Show Advanced Git Options

Builder

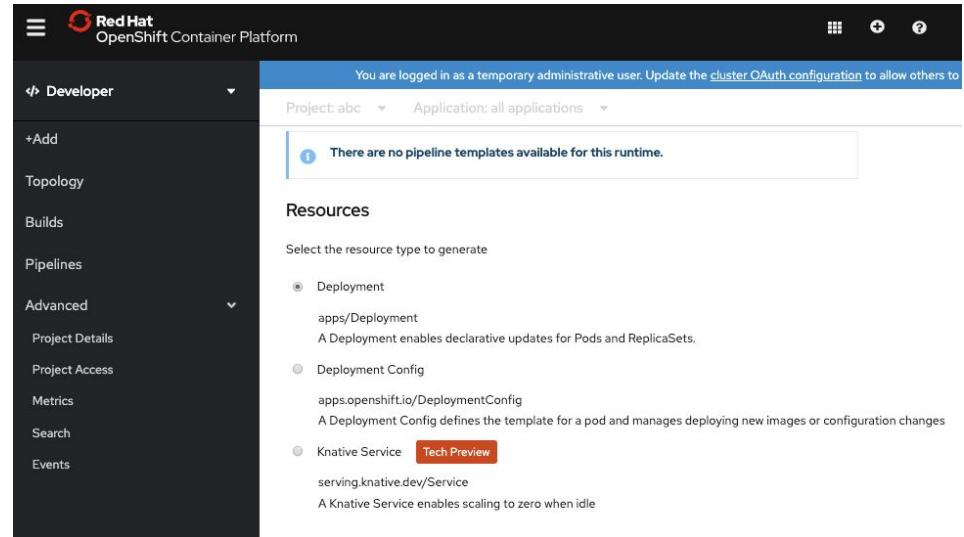
Builder Image *

Builder image(s) detected.
Recommended builder images are represented by ★ icon.

Perl	PHP	Nginx	JS	Httpd	.NET
Go	Ruby	Python	Java	Node.js	.NET Core

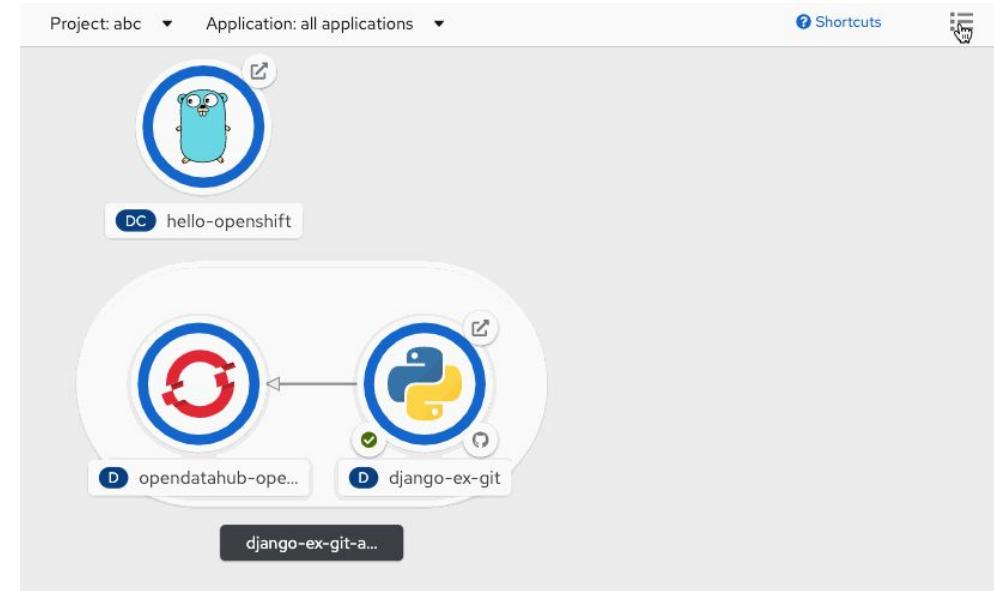
Deploy Applications alternate deployment targets

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



Application Topology streamlined flows

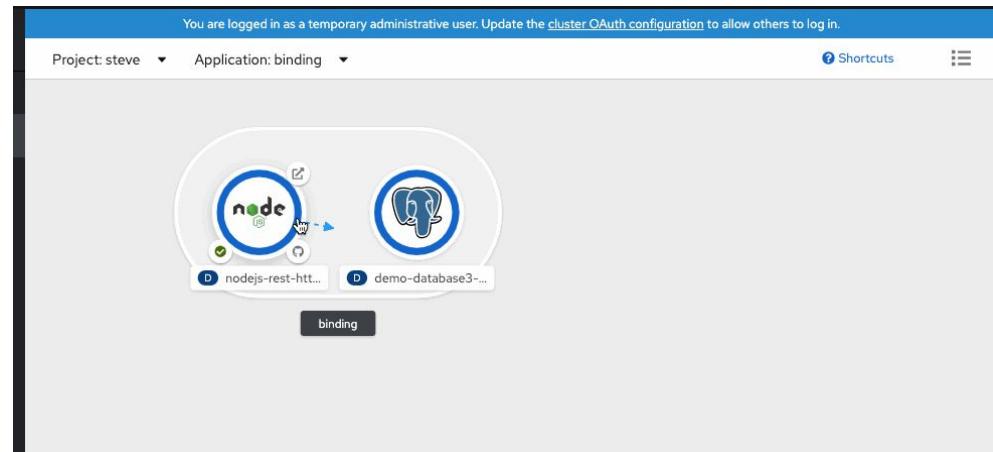
- Toggle between List and Topology views
- Easily group applications
- Connect/bind applications easily
- Contextual actions
- Quickly delete applications



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

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

Project: steve

PR steve Active

Dashboard Overview Y/A

Status

Active

Project Access

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

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

Project: steve

Project Access

Project Access allows you to add or remove a user's access to the project. More advanced management of role-based control appear in [Roles](#) and [Role Bindings](#). For more information, see the [role-based access control documentation](#).

Name	Role
kube-admin	Admin
pipeline	Edit
steve	Select a role

Add Access

Select a role

Admin

View

You made changes to this page.

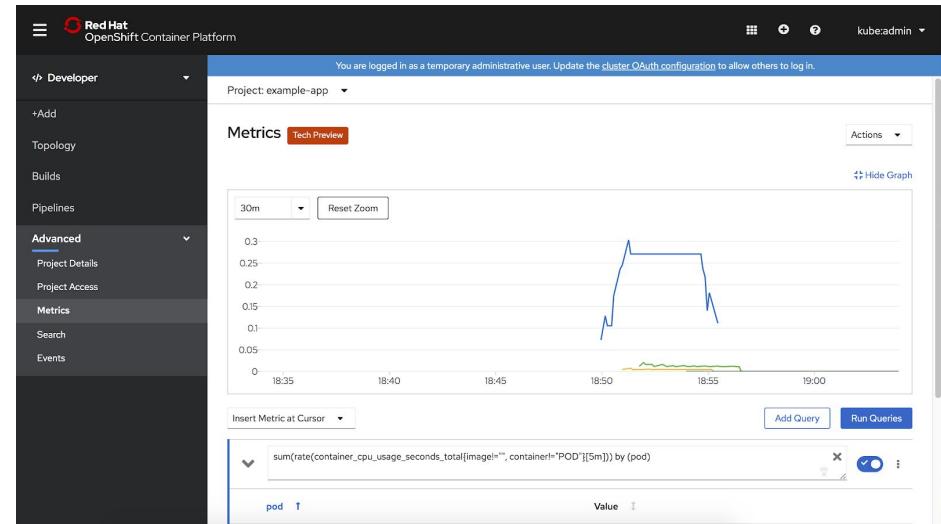
Click Save to save changes or Reload to cancel.

Save Reload

Metrics

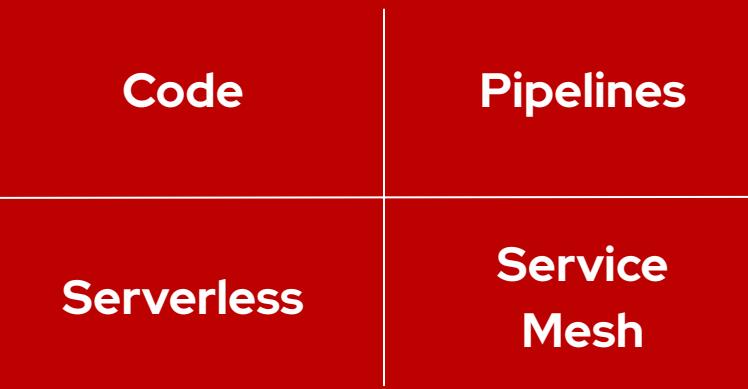
Quick access to key application metrics

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



Cloud Native Development

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

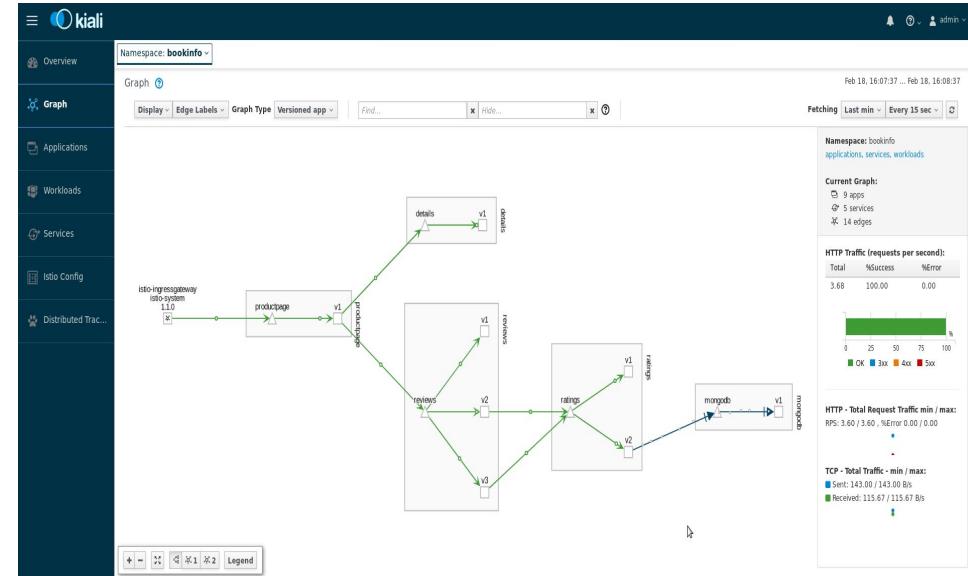


Service Mesh

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



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 % 

Evenly distribute traffic

Hide Advanced Options

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



Add LoadBalancer: OFF

Add Gateway: OFF

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	-

Serverless

OpenShift Serverless in 4.3

Key features and updates

- **Serverless Operator v1.3.0**
- **Knative v0.10**
- **OLM dependency resolution for Service Mesh**
- Dropped support for Kubernetes 1.14 (OCP 4.1)

Learn more

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

[Knative Tutorial](#)

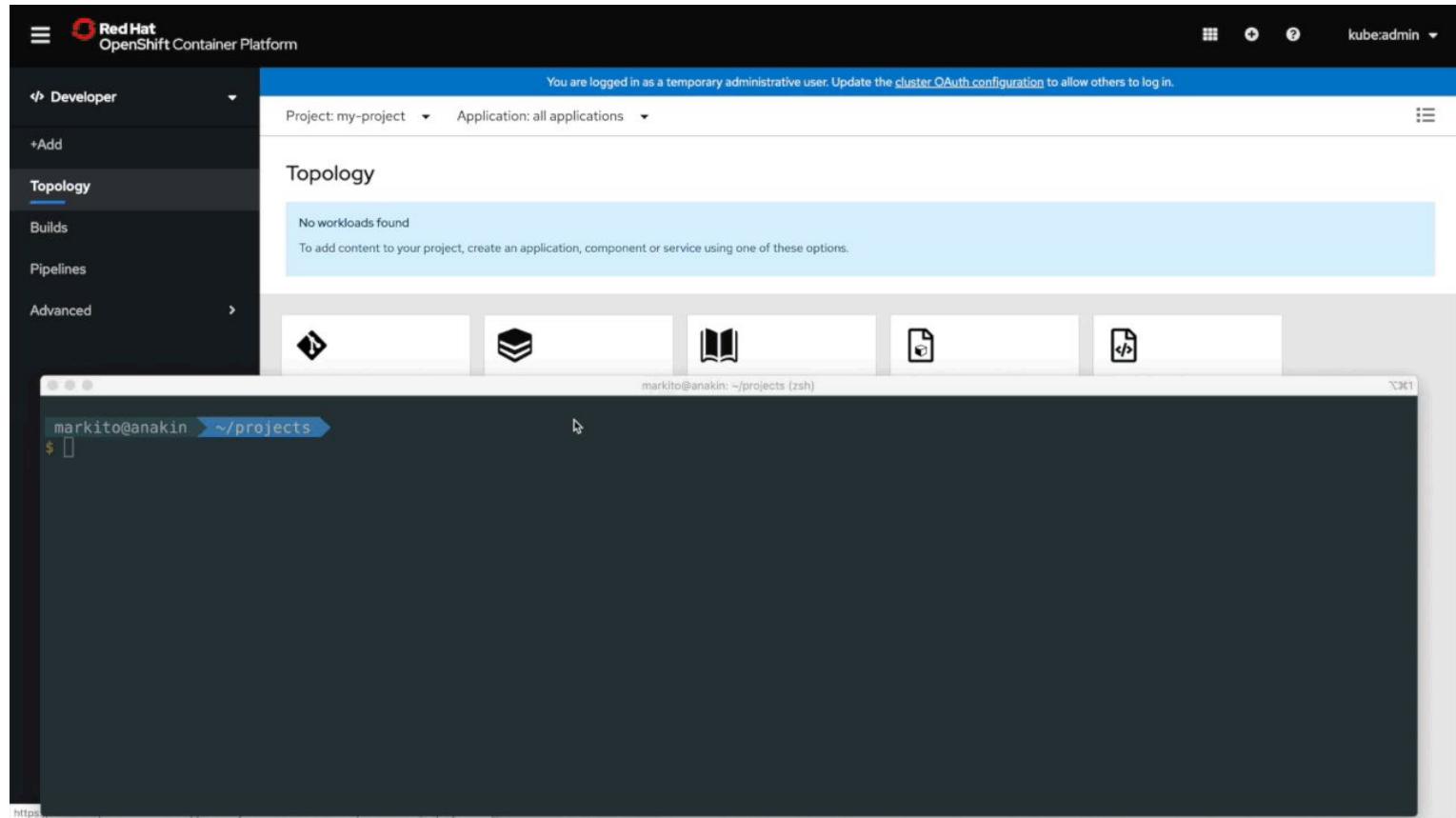
The screenshot shows the OpenShift web console interface. At the top, there is a navigation bar with 'Installed Operators' and 'Operator Details'. Below this, the 'OpenShift Serverless Operator' is listed as version 1.3.0 provided by Red Hat, Inc. The interface includes tabs for 'Overview', 'YAML', 'Events', and 'Knative Serving'. The 'Overview' tab is selected, showing a summary of the operator's status and a link to the Red Hat OpenShift Container Platform provider. On the right side, a detailed view of a Knative application deployment named 'spring-petclinic-bchpw-deployment' is shown. This view includes a circular progress indicator showing '4 scaling to 10', and a table with columns for Name, Namespace, Labels, Update Strategy, MaxUnavailable, MaxSurge, ProgressDeadline, and MinReadySeconds. The deployment is currently at 25% greater than 10 pods, with a progress deadline of 2m 0s and no minimum ready seconds. The labels include app=spring-petclinic-bchpw, app.kubernetes.io/...=springBootApp, and serving.knative.dev/configured=1. The Red Hat logo is visible in the bottom right corner.

OpenShift Serverless in 4.3

Traffic Split for Revisions

The screenshot illustrates the OpenShift Serverless interface for managing application revisions. On the left, a main panel displays a service named "store-app" with two revision icons: "store-app-lcvcb..." and "store-app-bbgc...". A dashed arrow indicates a 50% traffic split between these two revisions. Below the main panel are search and filter buttons. On the right, a detailed view for the "store-app" service shows the "Resources" tab selected. Under the "Revisions" section, two revisions are listed with their names and traffic percentages: "store-app-bbgc-1" at 50% and "store-app-lcvcb-2" at 50%. A "Set Traffic Distribution" button is located in this section. The "Routes" section is also visible, showing a single route entry for "store-app".

Revision	Traffic (%)
R store-app-bbgc-1	50%
R store-app-lcvcb-2	50%



OpenShift Serverless in 4.3

Event Sources

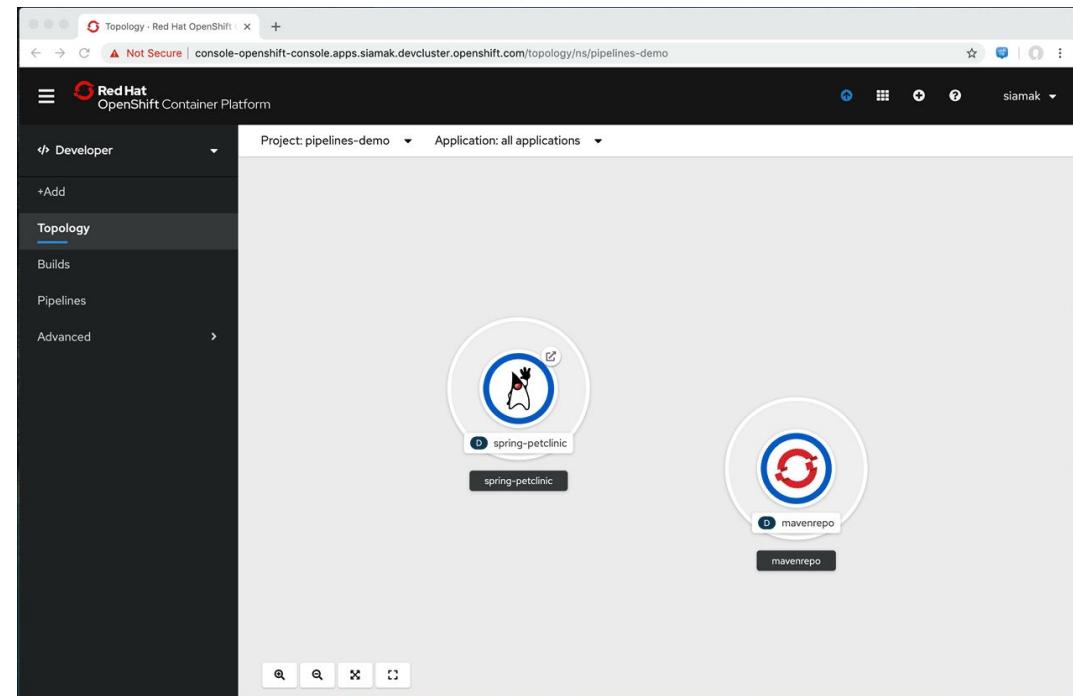
- KafkaSource
- CamelSource
- CronJobSource
- ContainerSource
- ApiServerSource

The screenshot shows the Red Hat OpenShift Container Platform web interface. The top navigation bar includes the Red Hat logo and the text "OpenShift Container Platform". The left sidebar has a "Developer" dropdown set to "Developer" and a "+Add" button. Below these are links for "Topology" (which is underlined), "Builds", "Pipelines", and "Advanced". The main content area has a blue header bar with the message: "You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in." It also shows "Project: my-project" and "Application: all applications". The main view displays a diagram of a CronJob source connected to two pods labeled "store-app". The pods have "Knative" icons and "50%" completion status. Labels include "CJS test-cronjob-so...", "R store-app-lcvccb...", and "R store-app-bbggc...". At the bottom are search and filter icons.

Pipelines

Cloud-native CI/CD with OpenShift Pipelines

- Based on Tekton Pipelines
- Runs serverless (no CI engine!)
- Containers as building blocks
- Build images with Kubernetes tools
(s2i, buildah, kaniko, jib, buildpack, etc)
- Pipelines portable to any Kubernetes
- Available in OperatorHub
- Tekton CLI



OpenShift Pipelines in OCP 4.3

- Git triggers (webhook)
- Automated RBAC setup
- Default curated tasks
- Pipeline metrics in Prometheus
- Pipeline samples and Task ref snippets in YAML editor

The screenshot shows the Red Hat OpenShift Container Platform web interface. On the left, a sidebar menu includes 'Developer', '+Add', 'Topology', 'Builds', 'Pipelines' (which is selected), and 'Advanced'. The main area is titled 'Create Pipeline' with the sub-instruction 'Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor.' A 'Dev Preview' button is visible in the top right of this area. Below this, a large code editor displays the following YAML configuration:

```
1  apiVersion: tekton.dev/v1alpha1
2  kind: Pipeline
3  metadata:
4    name: docker-build-and-deploy
5  spec:
6    params:
7      - name: IMAGE_NAME
8      | type: string
9    resources:
10      - name: app-source
11      | type: git
12      - name: app-image
13      | type: image
14    tasks:
15      - name: build
16        taskRef:
17          name: buildah
18          kind: ClusterTask
19        resources:
20          inputs:
21            - name: source
22            | resource: app-source
23          outputs:
24            - name: image
25            | resource: app-image
26  params:
```

To the right of the code editor is a sidebar titled 'Pipeline' with tabs for 'Samples' and 'Snippets'. The 'Snippets' tab is selected, showing a snippet titled 'S2I-Java-8 Task' with the following description: 'An S2I task to build java 8 based source. Source-to-Image (S2I) is a toolkit and workflow for building reproducible container images from source code. S2I produces images by injecting source code into a base S2I container image and letting the container prepare that source code for execution. The base S2I container images contains the language runtime and build tools needed for building and running the source code.' Below the description are buttons for 'Insert Snippet' and 'Show YAML >'. The top right corner of the interface shows the user's name, 'siamak'.

OpenShift Pipelines in OCP 4.3

- Default pipeline on app import (+Add) in Dev Console
- Pipeline objects in Admin Console
- New Tekton CLI commands
 - Start pipelines
 - Start tasks
 - Create resources

The screenshot displays two side-by-side views of the Red Hat OpenShift Container Platform interface.

Left View (Dev Console): Shows the navigation bar "Red Hat OpenShift Container Platform" and the "Administrator" dropdown. The main menu includes Home, Operators, Workloads, Networking, Storage, Builds, Pipelines, Tasks, Task Runs, Cluster Tasks, and Monitoring.

Right View (Admin Console): Shows the navigation bar "Red Hat OpenShift Container Platform" and the "Developer" dropdown. The main menu includes +Add, Topology, Builds, Pipelines, and Advanced. Under Pipelines, there are sub-options for Pipelines, Pipeline Runs, Pipeline Resources, Tasks, Task Runs, Cluster Tasks, and Monitoring.

Details Panel: Shows a component named "Red Hat OpenJDK 11" with the description "Builder Java OpenJDK 11. Build and run Java applications using Maven and OpenJDK 11. Sample repository: <https://github.com/jboss-openshift/openshift-quickstarts>".

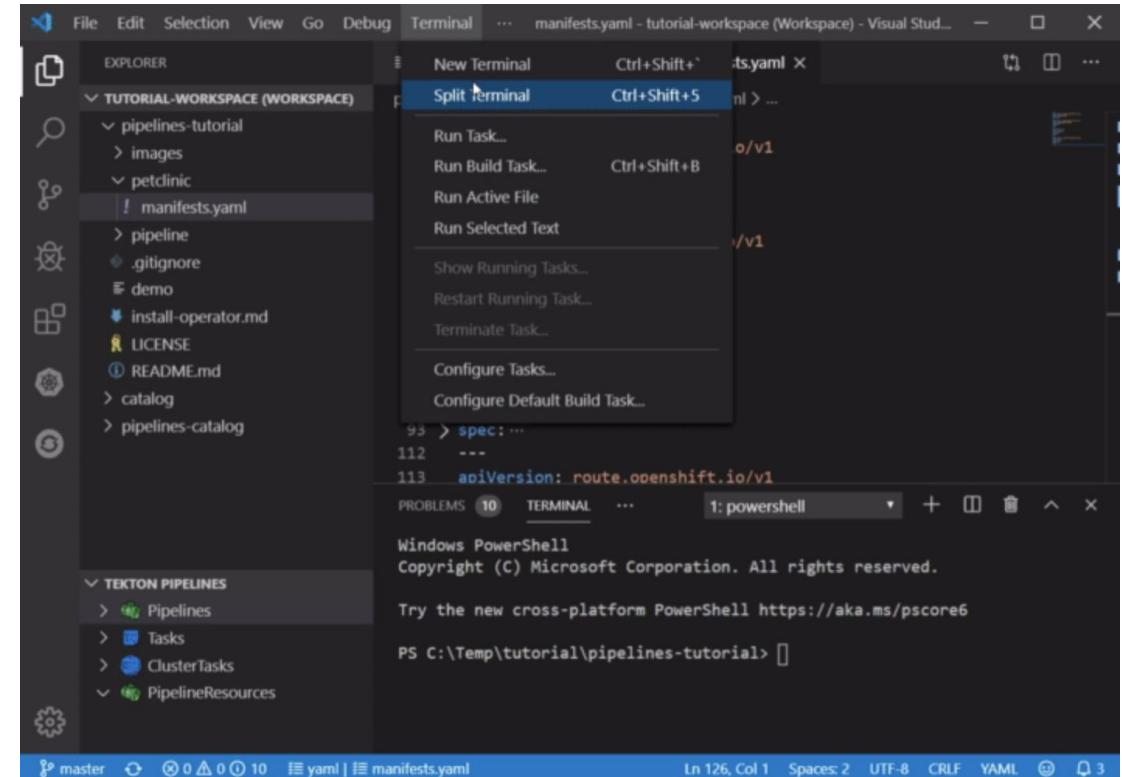
Form Fields: A form for creating a new pipeline. It includes fields for "Application" (set to "Triggers"), "Name" (input field), and a description. Below the form is a note: "A unique name given to the component that will be used to name associated resources."

Pipelines Tab: A tab labeled "Pipelines" with a "Dev Preview" badge. It contains a checkbox for "Add pipeline" and a "Hide pipeline visualization" dropdown.

Diagram: A small diagram showing a sequence of nodes: "build" and "deploy".

Tekton Pipelines VSCode Extension

Create, triggers and manage
Tekton Pipelines on OpenShift
and Kubernetes from Visual
Studio Code



Jenkins

- Jenkins server on JDK 8 & 11
- Jenkins agents
 - JDK 11
 - Node.js 10
- Official Jenkins Operator
 - github.com/jenkinsci/kubernetes-operator
 - Available in OperatorHub.io
 - Developer Preview on OCP 4.3
 - Collaboration upstream

The screenshot shows the Jenkins Operator page on OperatorHub.io. At the top, there's a header with the OperatorHub logo, a search bar, and a 'Contribute' button. Below the header, the title 'Jenkins Operator' is displayed next to a Jenkins logo. A sub-header states: 'Kubernetes native operator which fully manages Jenkins on Kubernetes.' Below this, a breadcrumb navigation shows 'Home > Jenkins Operator'. The main content area has a heading 'Jenkins Operator' and a large 'Install' button. To the right, there are sections for 'CHANNEL' (set to 'alpha'), 'VERSION' (set to '0.2.0 (Current)'), 'CAPABILITY LEVEL' (set to 'Basic Install'), and 'PROVIDER' (set to 'VirtusLab'). The central text area contains a question 'What's the Jenkins Operator?' followed by a detailed description of its purpose and benefits.

CodeReady / Dev Tools

odo - OpenShift's Dev-Focused CLI

Focus on additional stability & customer usage (46 issues fixed)

Improve output when showing list of components

Focus on R&D/spike for new use cases: Knative, other runtimes, devfile support, etc

```
$ 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
```

CodeReady Containers: OpenShift on your Laptop

New in 4.3:

- Automatic certificate rotation for internal node<->master communication
- 4.3 embedded GA version targeted for February 4th
- Ongoing updates with 4.2 z-stream updates
- Deprecated: removed VirtualBox support
- crc version outputs embedded OCP version number
- Many stability fixes around host networking

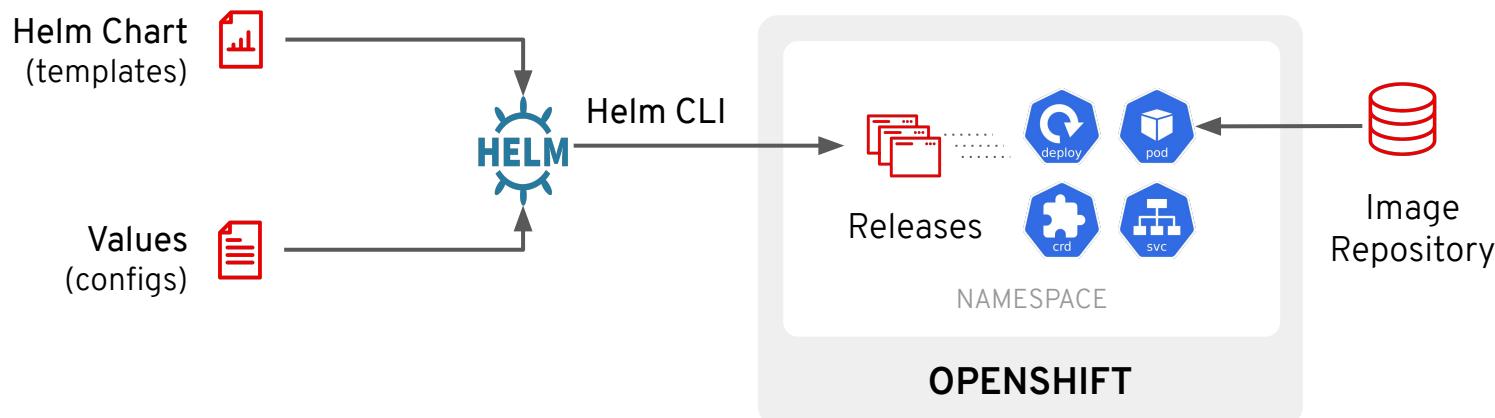
Provides a pre-built development environment based on **Red Hat Enterprise Linux** and **OpenShift** for quick container-based application development. Use with OpenShift on-premises or cloud.

```
$ crc setup  
Prepare your machine for running OpenShift  
  
$ crc start  
Start with the Hyperkit 4.3 bundle  
  
$ crc status  
Get the status of the cluster
```

Helm

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

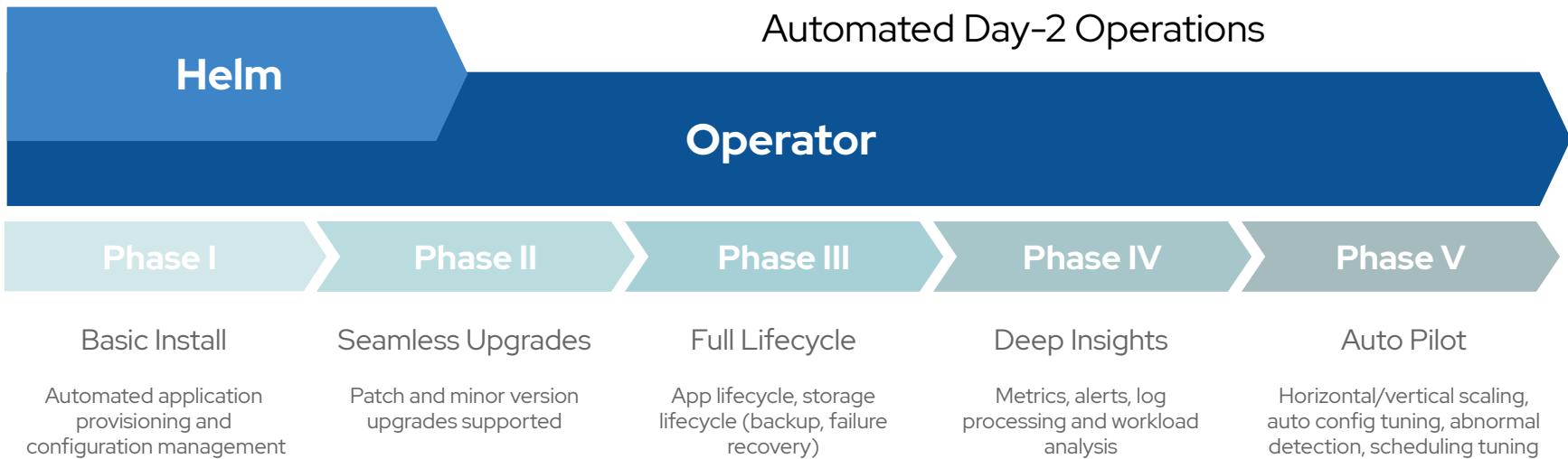
OpenShift 4.4+

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

The screenshot shows the Red Hat OpenShift Container Platform interface. On the left is a navigation sidebar with options like Home, Dashboards, Projects, Search, Explore, Events, Operators, Workloads, Networking, Storage, Builds, Pipelines, and Monitoring. The main content area has a header "Command Line Tools" and a sub-section "helm - Helm 3 CLI". It contains a link "Copy Login Command" and a link "Download helm". Below this, there's a section titled "oc - OpenShift Command Line Interface (CLI)" with a note about its capabilities and links to download the command-line tools for various architectures.

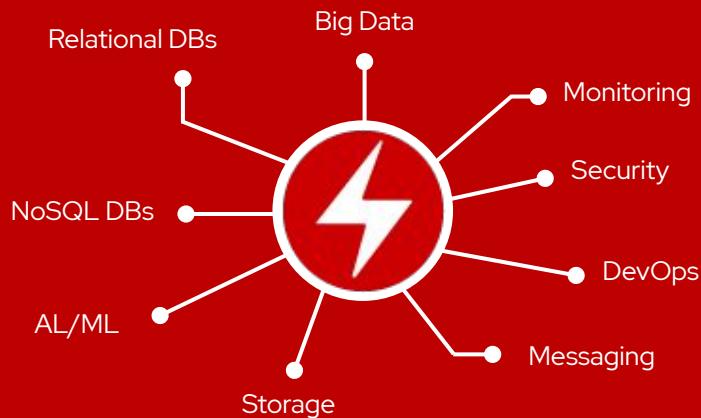
Helm and Operators

Package and Install



A broad ecosystem of workloads

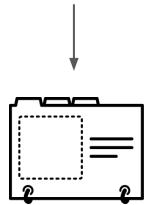
Operator-backed services allow for a SaaS experience on your own infrastructure



Simplified Mirroring of OperatorHub



Operator Catalog
on quay.io



- crunchy/postgres-server:9.5
- strimzi/kafka-image:latest
- kubevirt/virt-api:1.0.1
- ...



1. Mirror Operator Catalog into container image & push to disconnected registry

```
oc adm catalog build...
```

2. Parse referenced Operator and app images & push to disconnected registry

```
oc adm catalog mirror...
```

3. Enable mirror catalog in disconnected cluster

```
oc apply -f ./manifests
```

Operator Telemetry & Alerts

OpenShift 4.3 Cluster

OPERATOR LIFECYCLE MANAGER

aqua v1.0.1 mongoDB v2.0.0 MEMSQL v1.3.5 PlanetScale v4.1.2



OLM reports installed Operators name, version, channel and health state to Red Hat

Alert	Start	End	Severity
ClientCertificateExpiration	May 15, 2019	May 15, 2019	Warning
ClientCertificateExpiration	May 15, 2019	May 15, 2019	Critical
ClientCertificateExpiration	May 15, 2019	May 15, 2019	Pending

OLM fires alerts about Operator transitioning into failure state

Red Hat Operator SDK

Operator Testing (scorecard v2)

- Operator tests now categorized as required/optional
- Configurable test selection and pass/fail behavior
- Ship Custom tests

Framework Integration

- Single command to deploy OLM:
`operator-sdk alpha [install|uninstall|status] olm`

Ansible-based Operator

- Support for Prometheus Metrics
- Uses UBI base-image
- Molecule-based e2e testing

Helm-based Operator

- Helm v3 support (starting SDK 0.14.0)
- SDK automatically generates RBAC for your chart

Golang-based Operator

- Generate OpenAPI spec
- Supporting Kubernetes 1.14
- Go module support
- Controller-runtime 0.2.0
- Support for Prometheus Metrics
- Upstreamed Operator SDK features into k8s

OPENSHIFT ROADMAP

Q3 CY2019 OpenShift 4.2		Q1 CY2020 OpenShift 4.3		CY2020 OpenShift 4.4+	
HOSTED	PLATFORM	HOSTED	PLATFORM	HOSTED	PLATFORM
HOSTED	PLATFORM	HOSTED	PLATFORM	HOSTED	PLATFORM
<ul style="list-style-type: none">• Insights Operator• Azure Red Hat OpenShift new features (monitoring, logging)	<ul style="list-style-type: none">• Kubernetes 1.14 w/ CRI-O runtime• Disconnected Install and Update• Automated Installer for Azure, OSP, GCP• Pre-existing Infra Installer for GCP• Cluster-wide Egress Proxy• OVN Tech Preview• OpenShift Container Storage 4.2 (1 month after)	<ul style="list-style-type: none">• Developer Console GA• OpenShift Serverless (Knative) - TP• OpenShift Pipelines (Tekton) DP3• CodeReady Containers GA• Developer CLI (odo) GA <ul style="list-style-type: none">• OperatorHub Enhancements• Operator Deployment Field Forms• Application Migration Console	<ul style="list-style-type: none">• Kubernetes 1.16 w/ CRI-O runtime• Automated Installer for RHV• Private/Internal Clusters support from the installer• Deploy to pre-existing VPC & Subnets• FIPS• Pre-existing Infra Installer for OSP• OpenShift Container Storage 4.3 <ul style="list-style-type: none">• Subscription Mgmt Improvements (cloud.redhat.com)• Azure Red Hat OpenShift new features (private clusters)• Azure Red Hat OpenShift preview of 4.x• OSD on Google Cloud preview on 4.x	<ul style="list-style-type: none">• OpenShift Pipelines (Tekton) TP• Helm 3 TP <ul style="list-style-type: none">• Metering for Services• Windows Containers (Planned)• GPU Metering• Application Operator Binding - DP	<ul style="list-style-type: none">• OpenShift Serverless (Knative) GA• Guided application creation• OpenShift Pipelines (Tekton) GA• Helm 3 GA <ul style="list-style-type: none">• Monitor application workloads• Simplify OLM interactions• Improving native developer console for monitoring and troubleshooting <ul style="list-style-type: none">• Compact 3 node clusters• OVN GA w/ Windows Networking Integration (Planned)• Windows Containers GA• Multi-cluster summary dashboards• Centralized cluster updates• Compliance operator• Node problem detector• IPv6 (single/dual on control plane)• HTTP/2 Support• CSI certification suite <ul style="list-style-type: none">• Enhanced consumption building• Regulatory compliance• Machine autoscaling• Google cloud platform



Thank you

 [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