



# Co nowego w OpenShift 4

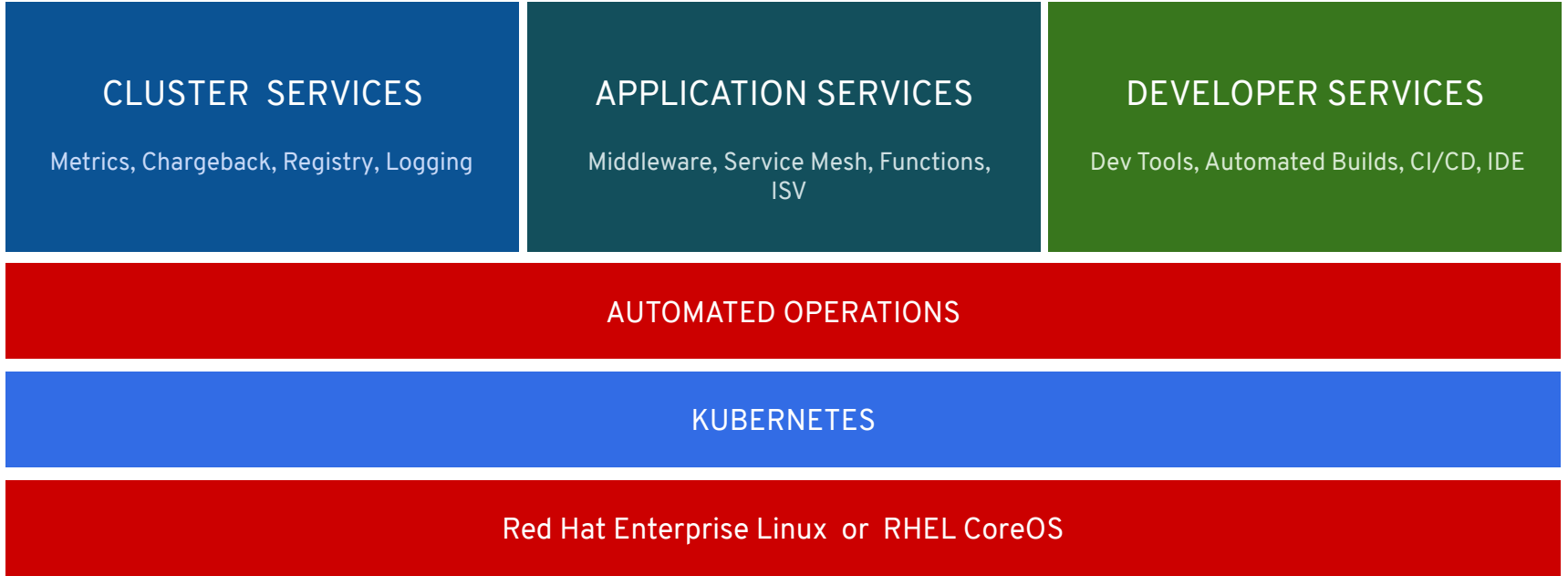
Jarosław Stakuń

RHCA, Senior Solutions Architect

Red Hat CEE



# OpenShift 4 Platform



Best IT Ops Experience

CaaS ↔ PaaS ↔ FaaS

Best Developer Experience



# Red Hat OpenShift 4

## Trusted enterprise Kubernetes

- Trusted host, content, platform
- Full-stack automated install
- Seamless updates & day 2 management

## A cloud-like experience, everywhere

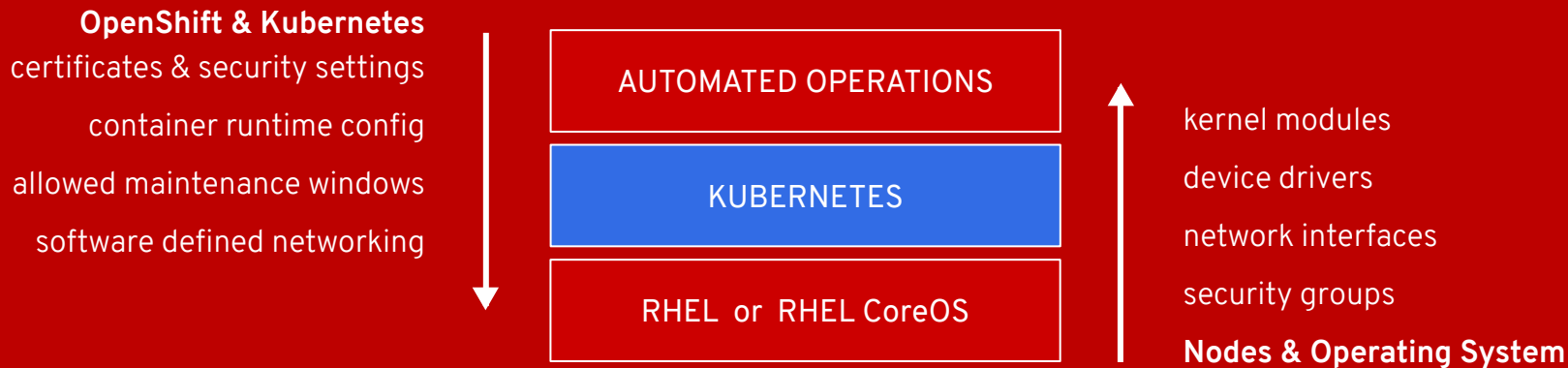
- Operator Framework
- Operator Hub & certified ISVs
- Hybrid, multicluster management

## Empowering developers to innovate

- OpenShift service mesh / Istio
- OpenShift serverless / Knative
- CodeReady Workspaces / Che

# The New Platform Boundary

OpenShift 4 is aware of the entire infrastructure and brings the Operating System under management



# Installation Experiences

## OPENSIFT CONTAINER PLATFORM

### Full Stack Automated

Simplified opinionated “Best Practices” for cluster provisioning

Fully automated installation and updates including host container OS.



### Pre-existing Infrastructure

Customer managed resources & infrastructure provisioning

Plug into existing DNS and security boundaries



## HOSTED OPENSIFT

### Azure Red Hat OpenShift

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

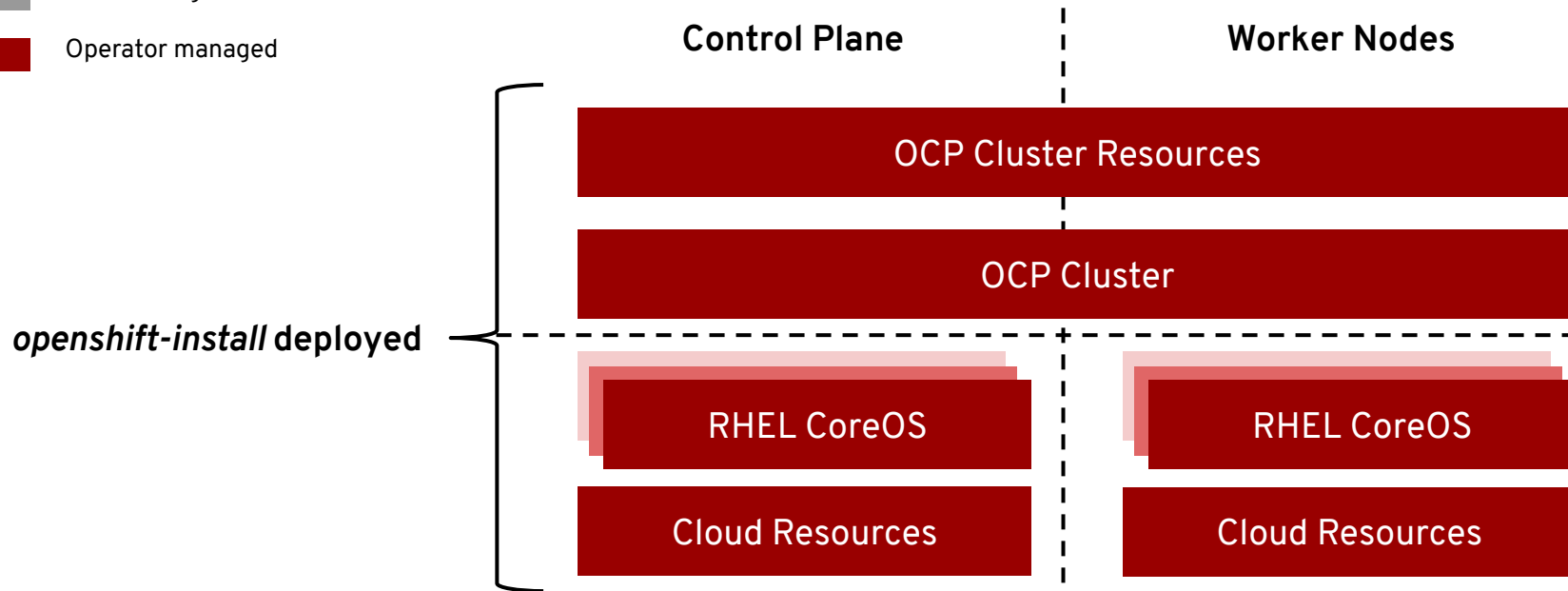
### OpenShift Dedicated

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

# Full Stack Automated Deployments

Day 1: openshift-install - Day 2: Operators

- User managed
- Operator managed



# Full Stack Automated Deployments

## Simplified Cluster Creation

Designed to easily provision a “best practices” OpenShift cluster

- New CLI-based installer with interactive guided workflow that allows for customization at each step
- Installer takes care of provisioning the underlying Infrastructure significantly reducing deployment complexity
- Leverages RHEL CoreOS for all node types enabling full stack automation of installation and updates of both platform and host OS content

## Faster Install

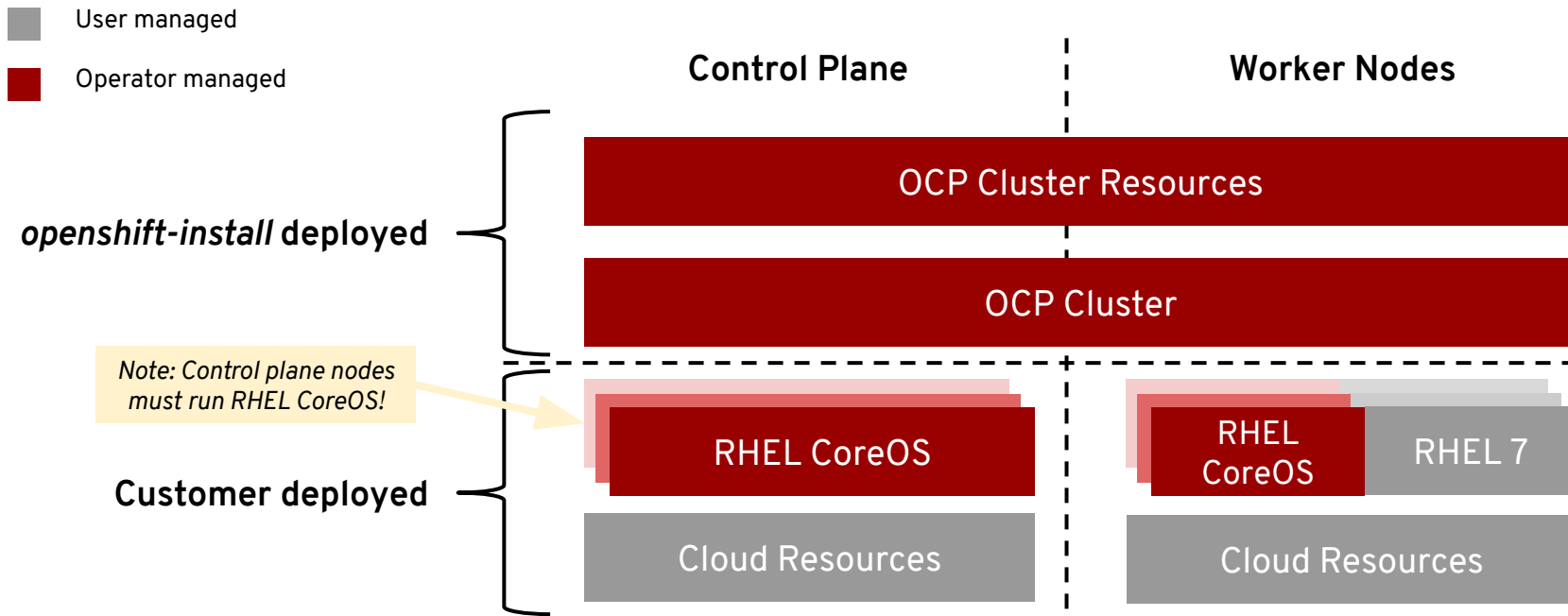
The installer typically finishes within 30 minutes

- Only minimal user input needed with all non-essential install config options now handled by component operator CRD's
- IPI installer 4.1 provides support for AWS deployments with additional provider support planned in future releases
- [See the OpenShift documentation for more details](#)

```
$ ./openshift-install --dir ./demo create cluster
? SSH Public Key /Users/demo/.ssh/id_rsa.pub
? Platform aws
? Region us-west-2
? Base Domain example.com
? Cluster Name demo
? Pull Secret [? for help]
*****
INFO Creating cluster...
INFO Waiting up to 30m0s for the Kubernetes API...
INFO API v1.11.0+c69f926354 up
INFO Waiting up to 30m0s for the bootstrap-complete event...
INFO Destroying the bootstrap resources...
INFO Waiting up to 10m0s for the openshift-console route to be created...
INFO Install complete!
INFO Run 'export KUBECONFIG=<your working directory>/auth/kubeconfig' to
manage the cluster with 'oc', the OpenShift CLI.
INFO The cluster is ready when 'oc login -u kubeadmin -p <provided>'
succeeds (wait a few minutes).
INFO Access the OpenShift web-console here:
https://console-openshift-console.apps.demo.example.com
INFO Login to the console with user: kubeadmin, password: <provided>
```

# Deploying to Pre-existing Infrastructure

Day 1: openshift-install - Day 2: Operators + Customer Managed Infra & Workers





# Deploying to Pre-existing Infrastructure

## Customized OpenShift Deployments

Enables OpenShift to be deployed to user managed resources and pre-existing infrastructure.

- Customers are responsible for provisioning all infrastructure objects including networks, load balancers, DNS, hardware/VMs and performing host OS installation
- Deployments can be performed both on-premise and to the public cloud
- OpenShift installer handles generating cluster assets (such as node ignition configs and kubeconfig) and aids with cluster bring-up by monitoring for bootstrap-complete and cluster-ready events
- While RHEL CoreOS is mandatory for the control plane, either RHEL CoreOS or RHEL 7 can be used for the worker/infra nodes
- Node auto-scaling can be setup for providers with OpenShift Machine API support
- [See the OpenShift documentation for more details](#)

```
$ cat ./demo/install-config.yaml
apiVersion: v1
baseDomain: example.com
compute:
- name: worker
  replicas: 0
controlPlane:
  name: master
...

$ ./openshift-install --dir ./demo create ignition-config
INFO Consuming "Install Config" from target directory

$ ./openshift-install --dir ./demo wait-for bootstrap-complete
INFO Waiting up to 30m0s for the Kubernetes API at
https://api.demo.example.com:6443...
INFO API v1.11.0+c69f926354 up
INFO Waiting up to 30m0s for the bootstrap-complete event...
$ ./openshift-install --dir ./demo wait-for cluster-ready

INFO Waiting up to 30m0s for the cluster at
https://api.demo.example.com:6443 to initialize...
INFO Install complete!
```

# Comparison between deployments methods

|                                  | Full Stack Automation  | Pre-existing Infrastructure                           |
|----------------------------------|------------------------|---|
| Build Network                    | Installer              | User  |
| Setup Load Balancers             | Installer              | User  |
| Configure DNS                    | Installer              | User  |
| Hardware/VM Provisioning         | Installer              | User  |
| OS Installation                  | Installer              | User  |
| Generate Ignition Configs        | Installer              | Installer   |
| OS Support                       | Installer: RHEL CoreOS | User: RHEL CoreOS + RHEL 7                            |
| Node Provisioning / Autoscaling  | Yes                    | Only for providers with OpenShift Machine API support |
| Customization & Provider Support | Best Practices: AWS    | Yes: AWS, Bare Metal, & VMware                        |

# Red Hat Enterprise Linux

## RED HAT<sup>®</sup> ENTERPRISE LINUX<sup>®</sup>

General Purpose OS

## RED HAT<sup>®</sup> ENTERPRISE LINUX CoreOS

Immutable container host

### BENEFITS

- 10+ year enterprise life cycle
- Industry standard security
- High performance on any infrastructure
- Customizable and compatible with wide ecosystem of partner solutions

- Self-managing, over-the-air updates
- Immutable and tightly integrated with OpenShift
- Host isolation is enforced via Containers
- Optimized performance on popular infrastructure

### WHEN TO USE

When customization and integration with additional solutions is required

When cloud-native, hands-free operations are a top priority

# Red Hat Enterprise Linux CoreOS

## 4.1 Image Availability:

- Amazon: AMIs
- vSphere: OVA
- Bare Metal: UEFI & BIOS

## Installation Requirements:

- RHCOS image + ignition config (installer generated)

## RHCOS Details

- RHEL 8 bits (4.18 kernel)
- Includes all packages required for OpenShift
- Over-The-Air updates encompass OCP & RHCOS
- Transactional host updates
- Read-only OS binaries
- Preconfigured for most environments

## Bare Metal Installer (ISO or PXE):

```
coreos.inst=yes
coreos.inst.install_dev=sda
coreos.inst.image_url=http://10.10.10.1/rhcos-metal-uefi.raw.gz
coreos.inst.ignition_url=http://10.10.10.1/master.ign
```

# Immutable Operating System

## Red Hat Enterprise Linux CoreOS is versioned with OpenShift

CoreOS is tested and shipped in conjunction with the platform. Red Hat runs thousands of tests against these configurations.

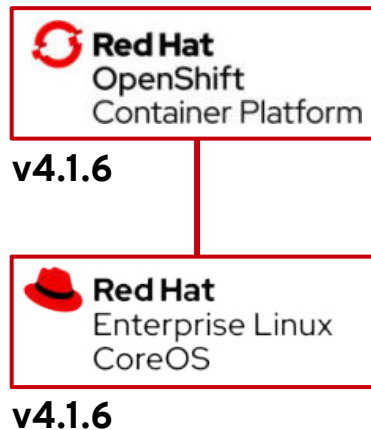
## Red Hat Enterprise Linux CoreOS is managed by the cluster

The Operating system is operated as part of the cluster, with the config for components managed by Machine Config Operator:

- CRI-O config
- Kubelet config
- Authorized registries
- SSH config

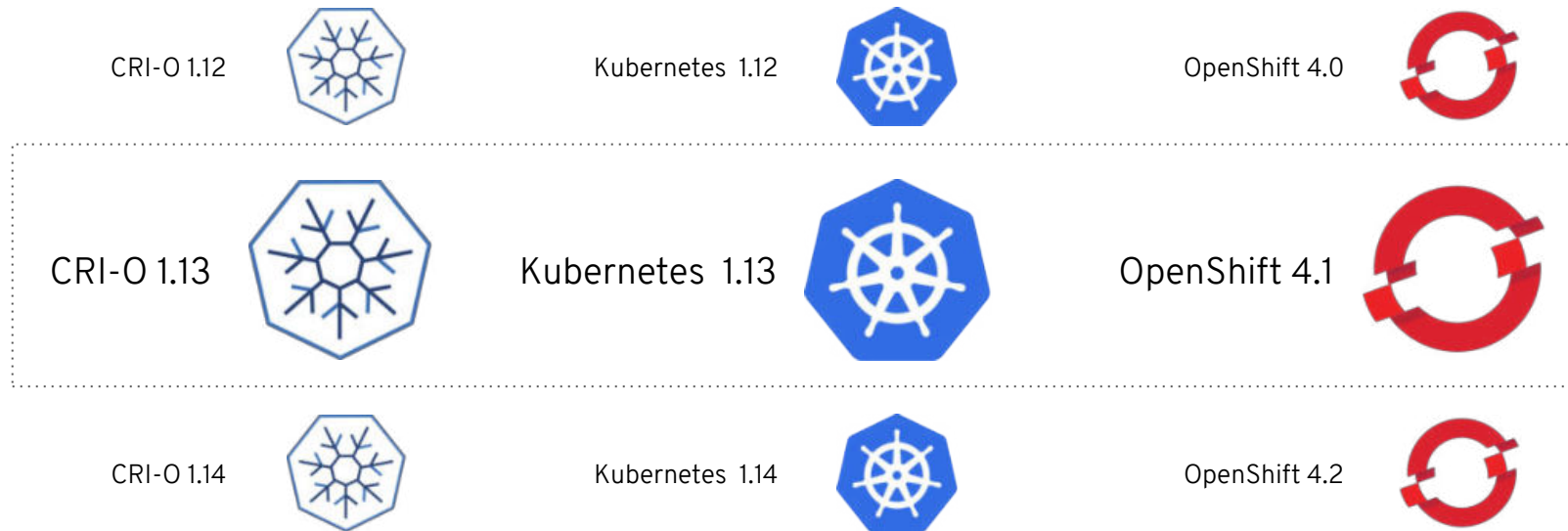
RHEL CoreOS admins are responsible for:

Nothing. 🤖 🙌



# CRI-O Support in OpenShift

CRI-O tracks and versions identical to Kubernetes, simplifying support permutations



# Graphical Re-configuration

## Global Configuration

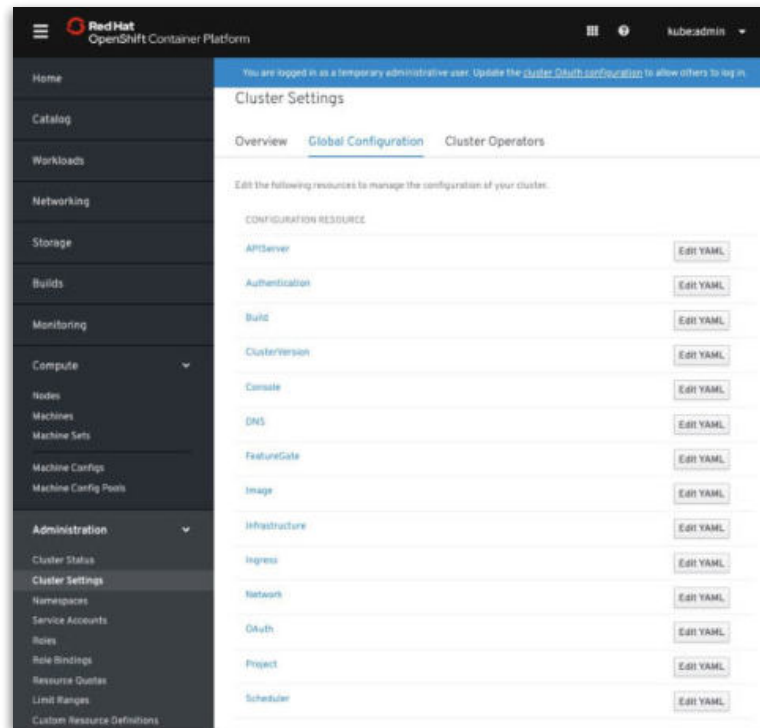
You complete most of the cluster configuration and customization after you deploy your OpenShift Container Platform cluster.

## Change via Cluster Settings screen

Once you have discovered your desired settings, changes can be made via Console or CLI.

## Operators apply these updates

One or more Operators are responsible for propagating these settings through the infrastructure



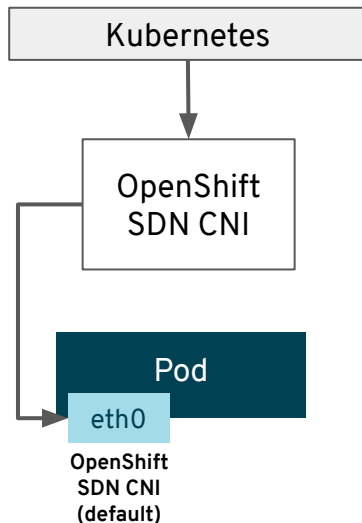
# Networking Plug-ins

## Multus Enables Multiple Networks & New Functionality to Existing Networking

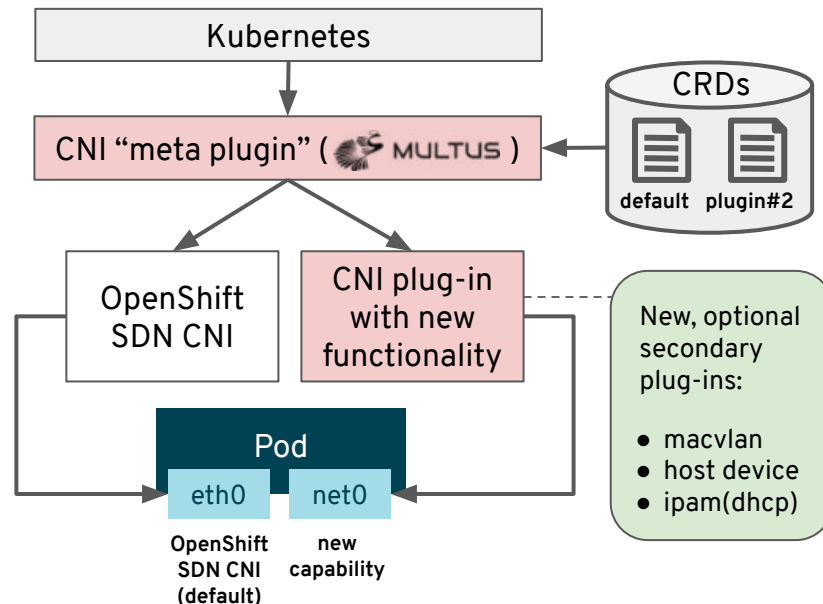
The Multus CNI “meta plugin” for Kubernetes enables one to create multiple network interfaces per pod, and assign a CNI plugin to each interface created.

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

3.x Capability...



4.x Capability...





# Storage

## Storage Focus

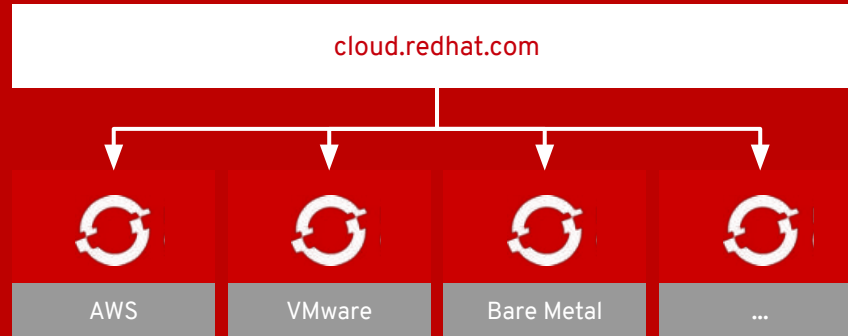
- Cluster Storage Operator
  - Sets up the default storage class
  - Looks through cloud provider and sets up the correct storage class
- Drivers themselves remain in-tree for now
- Focus has been on RHEL7 and RHCOS8 validating:
  - AWS EBS
  - vSphere Default Storage Class
- Openshift Container Storage v4 will be based on Ceph
  - Rook operator

| Supported           | Dev Preview   |
|---------------------|---------------|
| AWS EBS             | Snapshot*     |
| VMware vSphere Disk | EFS*          |
| NFS                 | Local Volume* |
| iSCSI               | Raw Block     |
| Fibre Channel       |               |
| HostPath            |               |

\* via external provisioner

# Cloud-like Simplicity, Everywhere

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



# OpenShift Cluster Manager on cloud.redhat.com

## Automatic registration of OpenShift clusters

View cluster versions and capacity in one place, no matter what infrastructure you are running on. Integrated with **RHSM**.

## OpenShift Dedicated cluster management

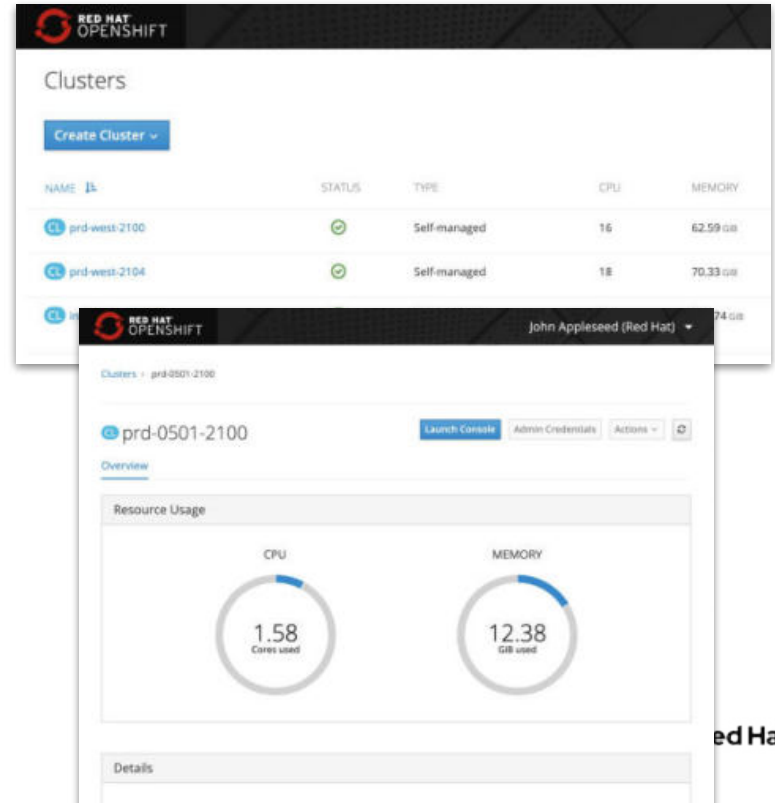
Self-service cluster deployment, scaling, and management for OpenShift Dedicated coming soon.

## Azure Red Hat OpenShift

Information about these clusters will be coming at a later date.

## Hosted in the United States

Other geographies may come later. You can [opt-out](#) too.



# Rolling Machine Updates

## Single-click updates

- RHEL CoreOS version & config
- Kubernetes core components
- OpenShift cluster components

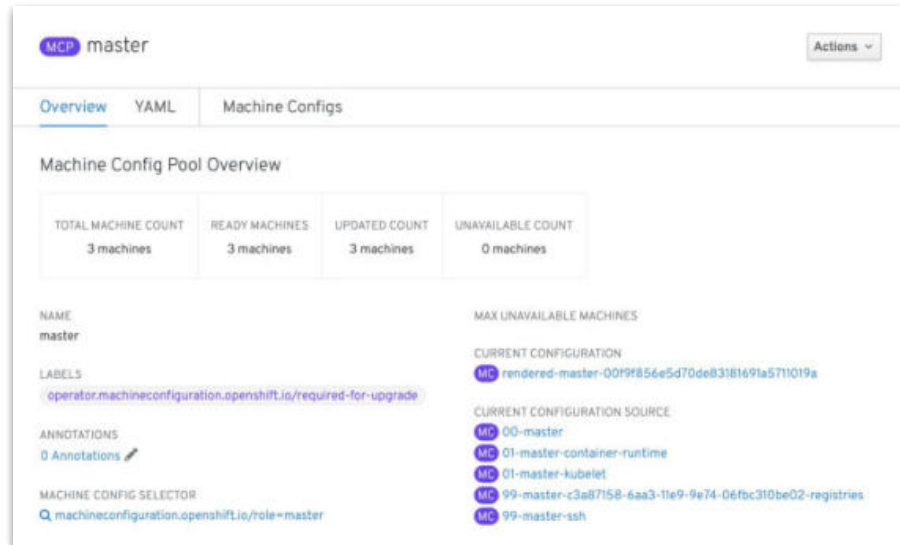
## Configure how many machines can be unavailable

Set the “maxUnavailable” setting in the MachineConfigPool to maintain high availability while rolling out updates.

The default is 1.

## Machine Config Operator (MCO) controls updates

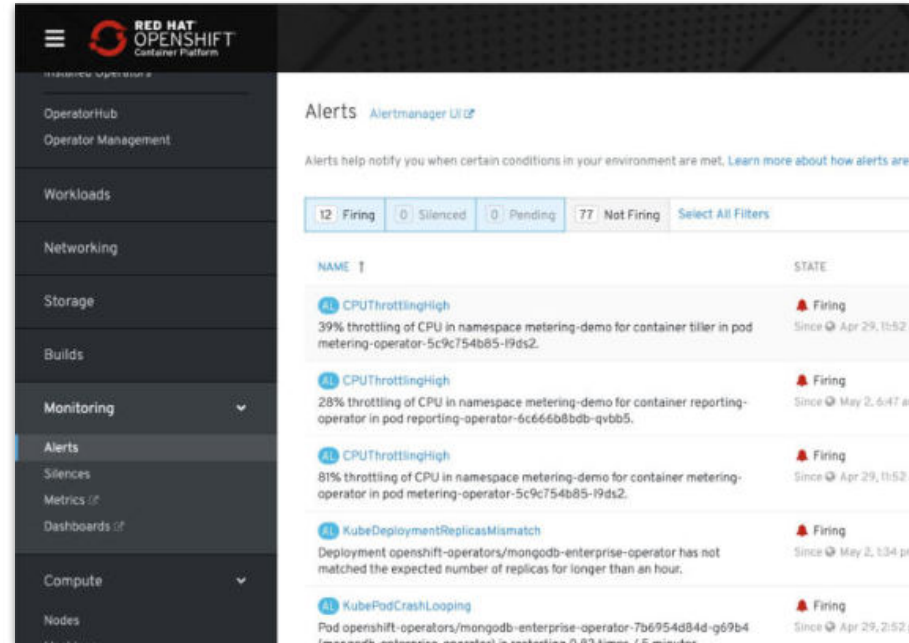
This is a DaemonSet that runs on all Nodes in the cluster. When you upgrade with `oc adm upgrade`, the MCO executes these changes.



# Cluster Monitoring

## Cluster monitoring is installed by default

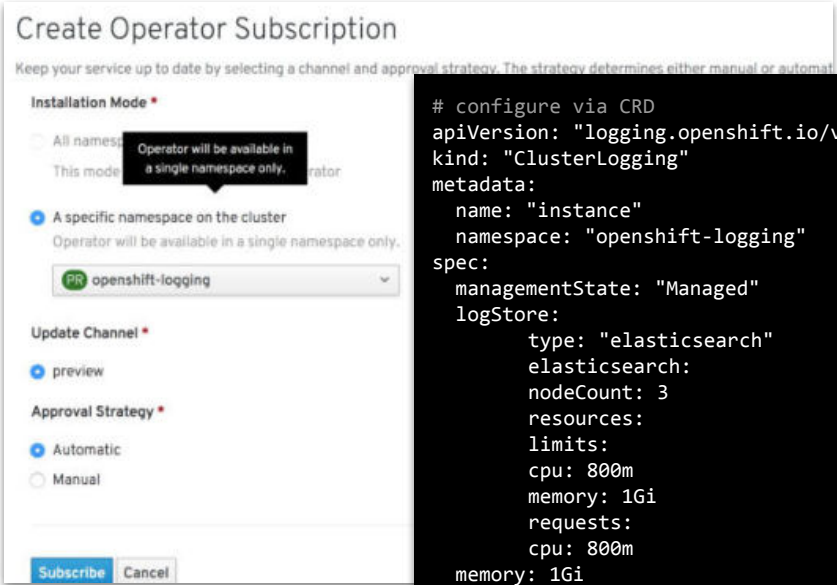
- Exposes resource metrics for Horizontal Pod **Autoscaling** (HPA) by default
  - HPA based on custom metric is tech preview
- No manual etcd monitoring configuration anymore
- New screens for managing Alerts & Silences
- More metrics available for troubleshooting purposes (e.g. HAproxy)
- Configuration via ConfigMaps and Secrets



# Cluster Logging

## Cluster Logging is lifecycle managed via Operator Lifecycle Management

- Install the Elasticsearch and Cluster Logging Operators from OperatorHub
- Create an instance of Cluster Logging. fluentd, Elasticsearch and Kibana (with Operators) are created
- Changing the out-of-box configuration:
  - CPU, memory requests and limits, PVC sizes etc can be changed by editing the Cluster Logging Operator YAML
- Direct Elasticsearch and Kibana Deployments to dedicated Nodes (recommended for production usage)



```
# configure via CRD
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
  elasticsearch:
    nodeCount: 3
  resources:
    limits:
      cpu: 800m
      memory: 1Gi
      requests:
        cpu: 800m
        memory: 1Gi
  storage:
    storageClassName: gp2
    size: 100G
    redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana"
    kibana:
      replicas: 1
```

# Infrastructure Machine Sets

## MachineSets that are purpose built for Infrastructure Services

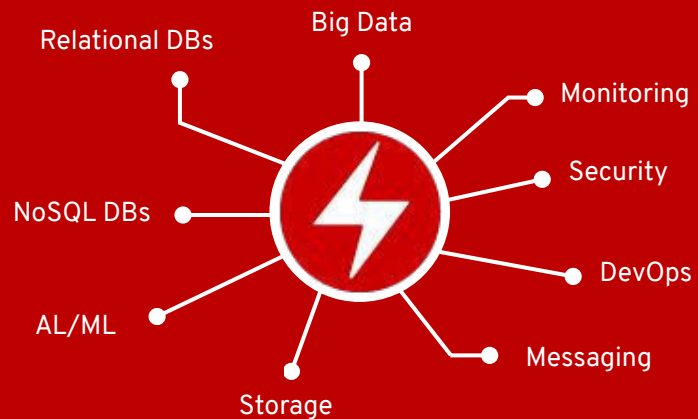
- Elasticsearch, Prometheus, Router, Registry
- Out-of-box installer does not create a MachineSets dedicated for Infra services
- Create a MachineSet via console or cli and label them with desired roles
- Redeploy Infra Services with nodeSelector set to the designated role

[Documentation: Creating Infrastructure MachineSets](#)

```
1 apiVersion: machine.openshift.io/v1beta1
2 kind: MachineSet
3 metadata:
4   labels:
5     machine.openshift.io/cluster-api-cluster: <clusterID>
6   name: <clusterID>-<role>-us-east-1a
7   namespace: openshift-machine-api
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      machine.openshift.io/cluster-api-cluster: <clusterID>
13      machine.openshift.io/cluster-api-machine-role: <role>
14      machine.openshift.io/cluster-api-machine-type: <role>
15      machine.openshift.io/cluster-api-machine-set: <clusterID>-<role>-us-east-1a
16  template:
17    metadata:
18      labels:
19        machine.openshift.io/cluster-api-cluster: <clusterID>
20        machine.openshift.io/cluster-api-machine-role: <role>
21        machine.openshift.io/cluster-api-machine-type: <role>
22        machine.openshift.io/cluster-api-machine-set: <clusterID>-<role>-us-east-1a
23  spec:
24    provider:
25      labels:
26        node-role.kubernetes.io/role: **
27    providerSpec:
28      apiVersion:
29      id: <apiID>
30      apiVersion: awsproviderconfig.openshift.io/v1beta1
31      blockDevices:
32        - ebs:
33          fs:
34            type:
35              volumeSize: 128
36              volumeType: gp2
37          credentialsSecret:
38            name: aws-cloud-credentials
39            deviceIndex: 0
40          subInstanceProfile:
41            id: <clusterID>-worker-profile
42            instanceType: m4.large
43            kind: AWSMachineProviderConfig
44          placement:
45            availabilityZone: us-east-1a
46            region: us-east-1
47          securityGroups:
48            - filters:
49              - name: tag:None
```

# A broad ecosystem of workloads

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





# Build Operators for your apps

Helm Chart

Helm Chart



Helm SDK

Helm SDK



Build operators from  
Helm chart, without any  
coding

Ansible Playbooks  
APBs

Ansible Playbooks  
APBs



Ansible SDK

Ansible SDK



Build operators from  
Ansible playbooks and  
APBs

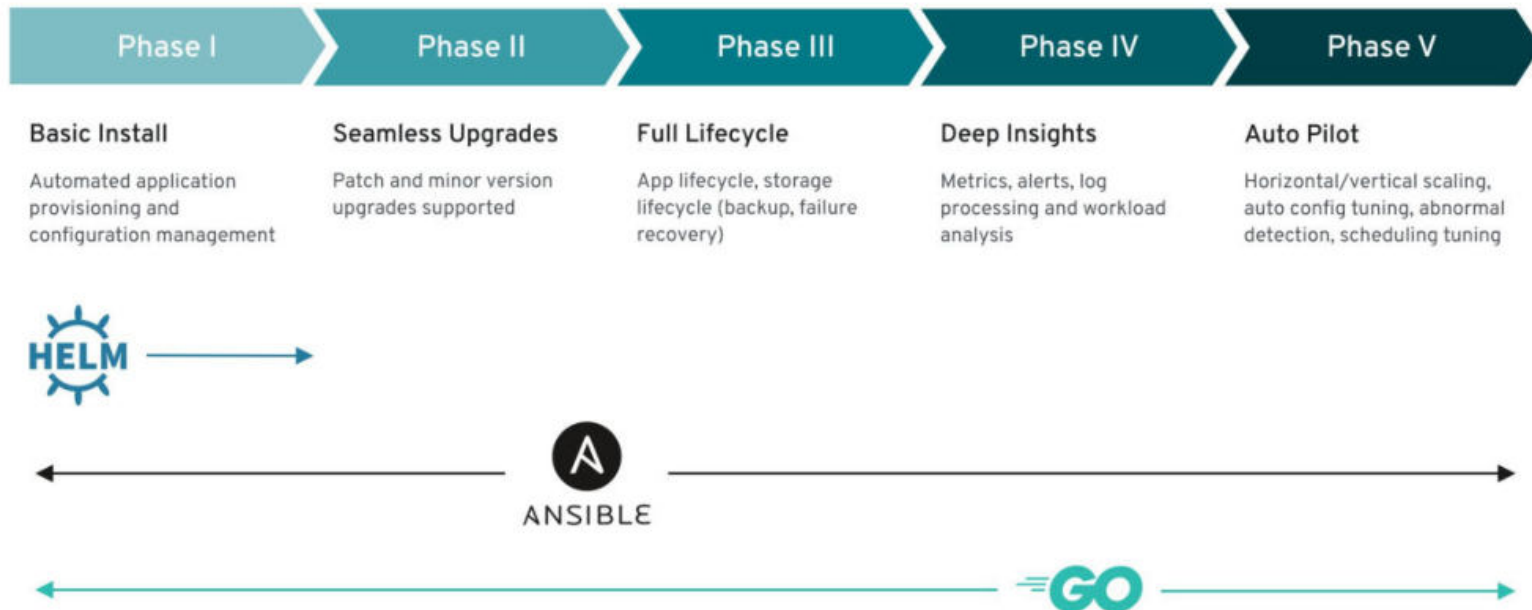
Go SDK

Go SDK



Build advanced operators  
for full lifecycle  
management

# Operator maturity model



# OperatorHub data sources

## Requires an online cluster

- For 4.1, the cluster must have connectivity to the internet
- Later 4.x releases will add offline capabilities

## Operator Metadata

- Stored in quay.io
- Fetches channels and available versions for each Operator

## Container Images

- Red Hat products and certified partners come from RHCC
- Community content comes from a variety of registries

The screenshot displays the OperatorHub interface. At the top, it says "Project: default" and "OperatorHub". Below this, a brief description states: "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, operators appear in the Developer Catalog, providing a self-service experience."

The main content area is a grid of operator cards. On the left, there is a sidebar with a list of categories: All Items, AI/Machine Learning, Application Monitoring, Big Data, Database, Developer Tools, Integration & Delivery, Logging & Tracing, Monitoring, Networking, OpenShift Optional, Security, Security Policy Management, Storage, Streaming & Messaging, and Other. Below the sidebar is a search bar labeled "Filter by keyword:" and a section for "INSTALL STATE" with radio buttons for "Installed (26)" and "Not Installed (40)".

The operator cards are arranged in a grid. Each card includes an icon, a title, a provider name, and a short description. For example, the first card is "AMD Streams" provided by Red Hat, Inc., with the description: "Red Hat AMD Streams is a massively scalable, distributed, and high-performance data stream processing engine." Other cards include "AppDynamics ClusterAgent", "Aqua Security Operator", "Automation Broker Operator", "Camel K Operator", "CockroachDB", "Community Jaeger Operator", "Couchbase Operator", "Crytic PostgreSQL Enterprise", "Descheduler", "Elasticsearch Operator", "Federation", "Federator", "FederatorAI", and "Hazelcast Operator".

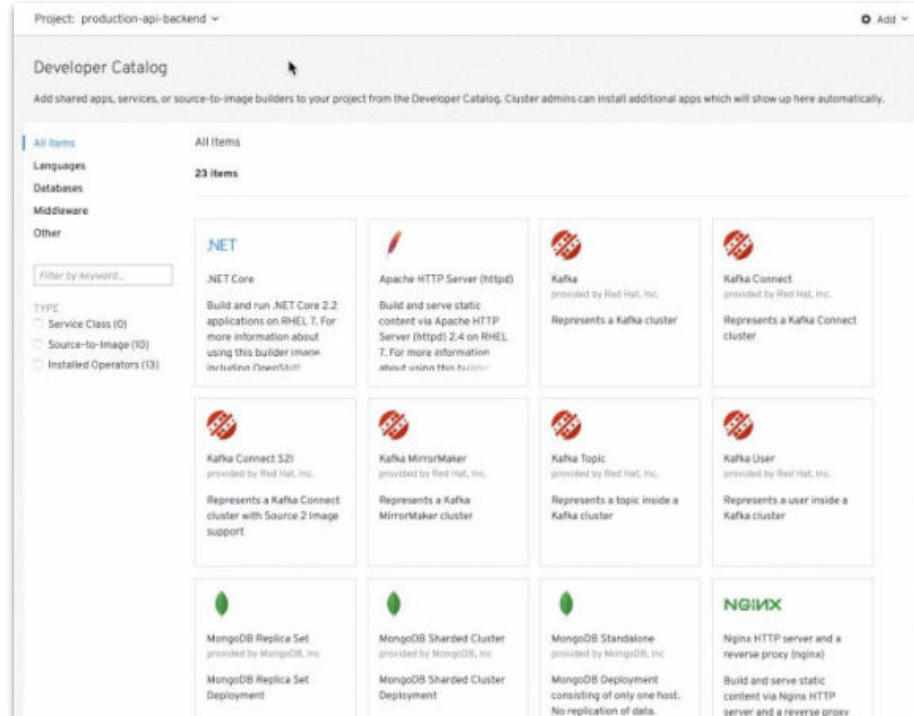
# Services ready for your developers

## New Developer Catalog aggregates apps

- Blended view of Operators, Templates and Broker backed services
- Operators can expose multiple CRDs. Example:
  - MongoDBReplicaSet
  - MongoDBSharded Cluster
  - MongoDBStandalone
- Developers can't see any of the admin screens

## Self-service is key for productivity

- Developers with access can change settings and test out new services at any time

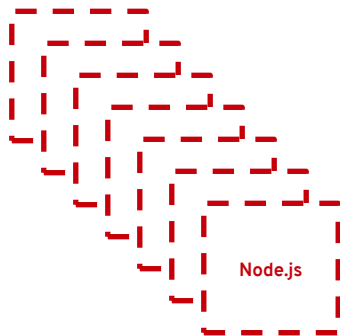


# Red Hat Universal Base Image

Enable an ecosystem of freely distributable operators for Kubernetes/OpenShift



Base  
Images

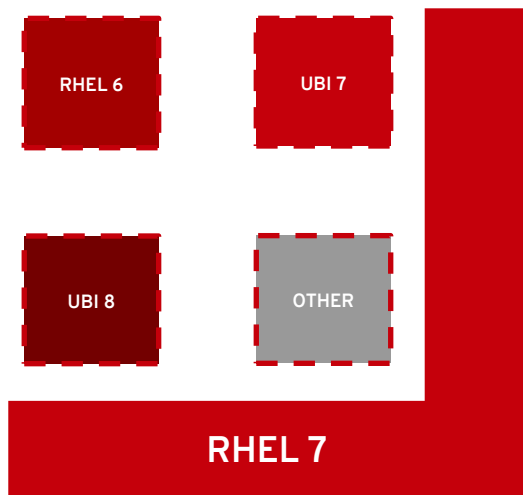


Pre-Built  
Language  
Images

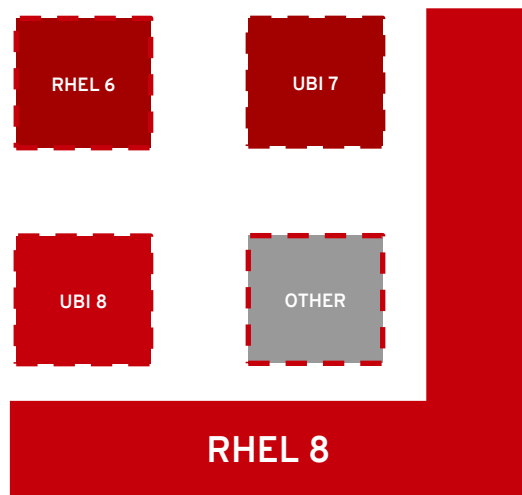


Package  
Subset

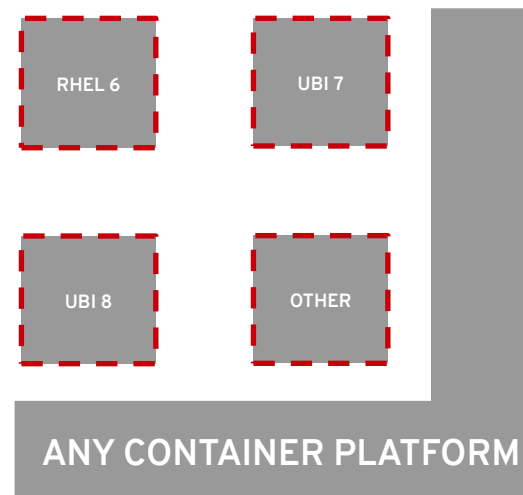
# UBI and Host interactions



Red Hat Enterprise Linux 7



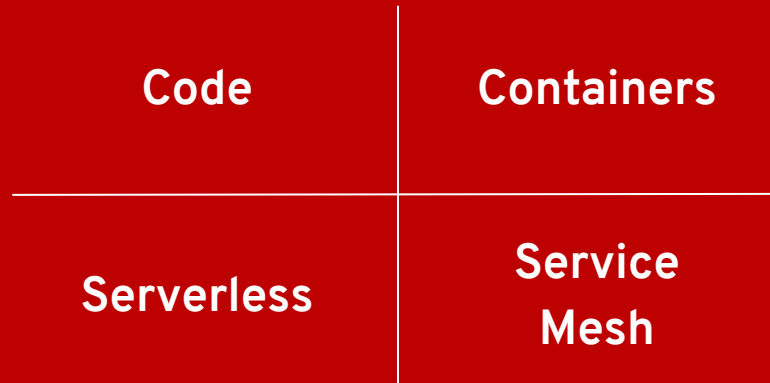
Red Hat Enterprise Linux 8



Like any community distro

# Next wave of developer tools

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



# 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
- Pipelines portable to any Kubernetes
- Available in OperatorHub

The screenshot displays the OpenShift Pipelines console interface. The left sidebar shows navigation options: Developer, Add, Topology, Builds, Pipelines (selected), and Advanced. The main content area is titled 'Pipelines' and shows a table of pipeline runs. A search bar at the top right allows filtering by name. The table columns are NAME, LAST PIPELINE RUN, LAST RUN STATUS, TASK COMPLETED, and LAST RUN STARTED. The data rows are as follows:

| NAME       | LAST PIPELINE RUN | LAST RUN STATUS | TASK COMPLETED | LAST RUN STARTED |
|------------|-------------------|-----------------|----------------|------------------|
| Pipeline-A | Pipeline-run-1    | Running         | 2 of 4         | 3 seconds ago    |
| Pipeline-B | Pipeline-run12    | Running         | 3 of 5         | 2 minutes ago    |
| Pipeline-C | Pipeline-run23    | Succeeded       | 3 of 3         | 4 minutes ago    |
| Pipeline-D | Pipeline-run4     | Failed          | 2 of 4         | 6 minutes ago    |
| Pipeline-E | Pipeline-run34    | Succeeded       | 2 of 2         | 8 minutes ago    |

Below the table, there are several informational cards: 'Storage', 'Builds', 'Monitoring', 'Developer Tools' (with sub-items: Integration & Delivery, Logging & Tracing, Monitoring, Networking), 'Knative Serving Operator' (provided by Red Hat, Knative Serving builds on Kubernetes to support deploying and serving of serverless applications), and 'OpenShift Pipelines Operator' (provided by Red Hat, Inc., OpenShift Pipelines is a cloud-native CI/CD solution for building pipelines).

Dev Preview on OCP 4.1 (June)



# Cloud-native CI/CD with OpenShift Pipelines

```
apiVersion: tekton.dev/v1alpha1
kind: Pipeline
metadata:
  name: funky-deploy-pipeline
spec:
  resources:
    ... # git, images, etc
  tasks:
  - name: build-app
    taskRef:
      name: mvn-build
    ...
  - name: build-image
    taskRef:
      name: s2i-java
    ...
  - name: deploy
    taskRef:
      name: openshift-cli
    ...
```

Inputs (e.g. git repo) to and outputs (e.g. images) from the pipeline

Provided task library:  
s2i, buildah, oc, jib, kaniko, etc

User can create custom ones



# OpenShift Serverless

Functions

Apps

Microservices

Containers

Platform



*Application*



*Infrastructure*



RED HAT  
OPENSHIFT



# 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 v0.6 (v1beta1 APIs)
- No vendor lock in

## Learn more

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

<http://bit.ly/knative-tutorial>

The screenshot displays the Red Hat OpenShift Container Platform console. The left sidebar shows navigation options like Home, Projects, Status, Search, Events, Catalog, OperatorHub, Workloads, Networking, and Storage. The main content area shows the 'Project: openshift-operators' page with a list of operators. The 'Knative Serving Operator' is highlighted with a red box, showing it is 'Installed'. Below the console, there is a diagram of the Knative architecture showing components like 'knative-ingressgateway', 'helloworld-go-00001', 'autoscaler', 'knative-serving', 'activator', and 'knative-serving-00001'. A small graph shows resource usage over time.



```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
  labels:
    app: guestbook
spec:
  selector:
    matchLabels:
      app: guestbook
      tier: frontend
  replicas: 1
  template:
    metadata:
      labels:
        app: guestbook
        tier: frontend
    spec:
      containers:
        - image: markusthoemmes/guestbook
          name: guestbook
          resources:
            requests:
              cpu: 100m
              memory: 100Mi
          env:
            - name: GET_HOSTS_FROM
              value: dns
          ports:
            - containerPort: 80

```

## Kubernetes

```

---
apiVersion: v1
kind: Service
metadata:
  name: frontend-service
  labels:
    app: guestbook
    tier: frontend
spec:
  ports:
    - port: 80
  selector:
    app: guestbook
    tier: frontend
---
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: frontend-route
spec:
  to:
    kind: Service
    name: frontend-service

```

53 lines

## Knative

```

apiVersion: serving.knative.dev/v1alpha1
kind: Service
metadata:
  name: frontend
spec:
  template:
    metadata:
      labels:
        app: guestbook
        tier: frontend
    spec:
      containers:
        - image: markusthoemmes/guestbook
          resources:
            requests:
              cpu: 100m
              memory: 100Mi
          env:
            - name: GET_HOSTS_FROM
              value: dns
          ports:
            - containerPort: 80

```

22 lines



# OpenShift Serverless + Azure Functions

## Key Features

- Enable FaaS in OpenShift
- Familiar developer experience using VS Code and Azure CLI
- Polling based auto-scaling for Azure Queues, Kafka...
- Reuse Knative event sources, HTTP auto-scaling
- On premise or Any cloud.

## Learn more

<https://github.com/kedacore/keda>



In partnership with

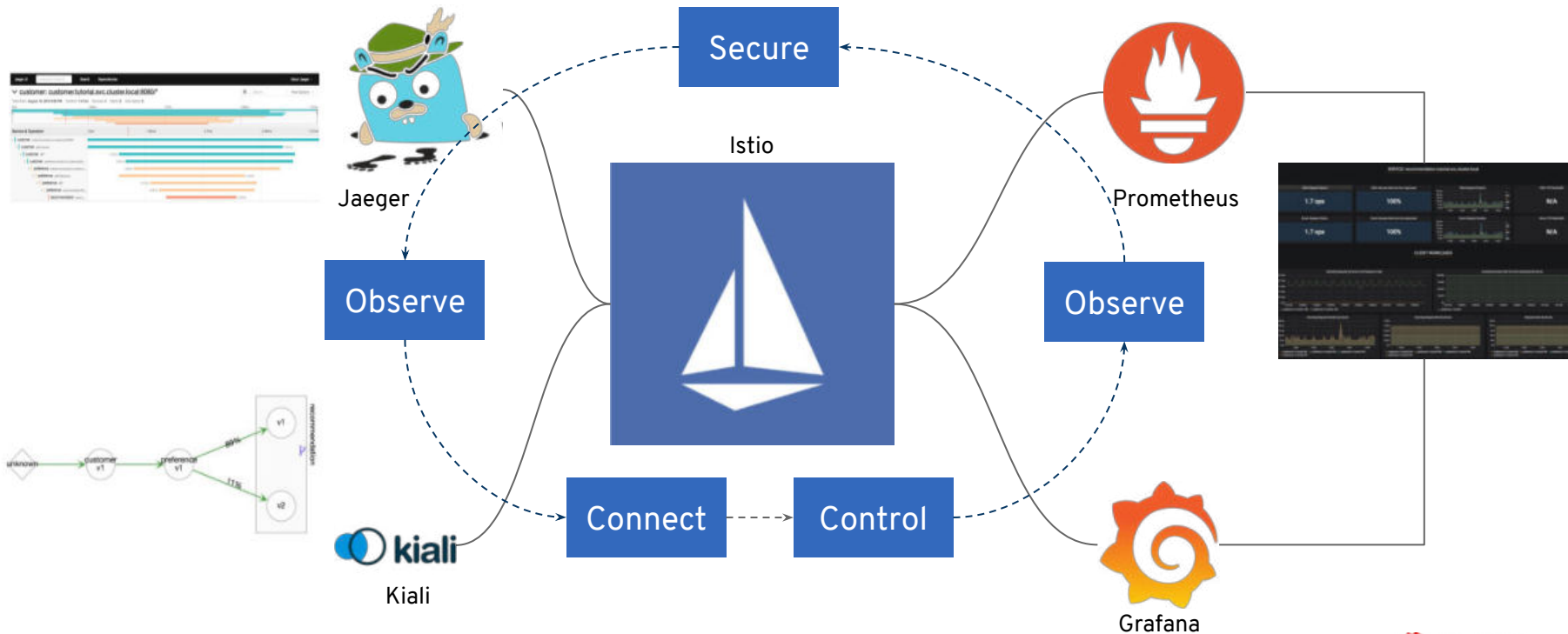


**Red Hat**



**Microsoft Azure**

# Red Hat Service Mesh



# CodeReady Workspaces

Collaborative web IDE

Supported Eclipse Che

Available in OperatorHub

Included in OCP/OSD SKU

The screenshot shows the Red Hat OpenShift Container Platform OperatorHub interface. The left sidebar contains navigation options: Home, Projects, Status, Search, Events, Catalog, Developer Catalog, Installed Operators, OperatorHub, Operator Management, Workloads, Networking, Storage, Builds, Monitoring, Compute, and Administration. The main content area displays the details for the 'Red Hat CodeReady Workspaces' operator. The operator version is 'LLD', provided by 'Red Hat, Inc.'. The repository is 'https://github.com/eclipse/che-operator'. The container image is 'registry.access.redhat.com/codeready-workspaces/server-operator:1.1'. The operator was created on '2019-03-06 11:59:59'. The support is provided by 'Red Hat, Inc.'. The description states: 'A collaborative Kubernetes-native development solution that delivers OpenShift workspaces and in-browser IDE for rapid cloud application development. This operator installs PostgreSQL, Red Hat SSO, and the Red Hat CodeReady Workspaces server, as well as configures all three services.' The 'Pre-Req' section lists requirements for a ClusterRole and a service account. The 'SUPPORT' section indicates that the operator is supported by Red Hat, Inc. The 'INSTALL STATE' section is partially visible at the bottom.

Project: pipelines-tutorial

## OperatorHub

Discover Operators from the Kubernetes ecosystem and install them on your OpenShift cluster.

All Items  
AI/Machine Learning  
Application Monitoring  
Big Data  
Cloud Provider  
Database  
Developer Tools  
Integration & Delivery  
Logging & Tracing  
Monitoring  
Networking  
OpenShift Optional  
Security  
Security Policy Management  
Storage  
Streaming & Messaging

Filter by keyword...

INSTALL STATE

### Red Hat CodeReady Workspaces

LLD provided by Red Hat, Inc.

**Install**

OPERATOR VERSION  
LLD

PROVIDER TYPE  
Red Hat

PROVIDER  
Red Hat, Inc.

REPOSITORY  
https://github.com/eclipse/che-operator

CONTAINER IMAGE  
registry.access.redhat.com/codeready-workspaces/server-operator:1.1

CREATED AT  
2019-03-06 11:59:59

SUPPORT  
Red Hat, Inc.

A collaborative Kubernetes-native development solution that delivers OpenShift workspaces and in-browser IDE for rapid cloud application development. This operator installs PostgreSQL, Red Hat SSO, and the Red Hat CodeReady Workspaces server, as well as configures all three services.

### Pre-Req

In addition to a standard namespaced role, this operator may require a **ClusterRole** that allows the operator service account to:

- read secrets in openshift-ingress namespace (the operator will get TLS secret in openshift-ingress namespace to extract certificate and add it to Java trust store for Red Hat CodeReady Workspaces server)
- list, get, create, update, patch, watch and delete oauthclients at a cluster scope (the operator creates oAuthclient and OpenShift v3 identity provider in Keycloak to enable Login With OpenShift in Red Hat CodeReady Workspaces)

The operator service account will require extra privileges if you enable either `server.selfSignedCerts` or `auth.openshiftOAuth` which are false in CR template by default. After the operator is installed, grant the `codeready-operator` service account such privileges: When `auth.openshiftOAuth` is enabled:

```
oc create clusterrole codeready-operator --resource=oauthclients --verb=get,create,delete,update,list,watch
oc create clusterrolebinding codeready-operator --clusterrole=codeready-operator --serviceaccount=$(NAMESPACE):codeready-operator
```

When `server.selfSignedCerts` is enabled:

```
oc create role secret-reader --resource=secrets --verb=get --noopenshift-ingress
oc create rolebinding codeready-operator --role=secret-reader --serviceaccount=$(NAMESPACE):codeready-operator --noopenshift-ingress
```

\$(NAMESPACE) is an OpenShift project where you installed the operator.

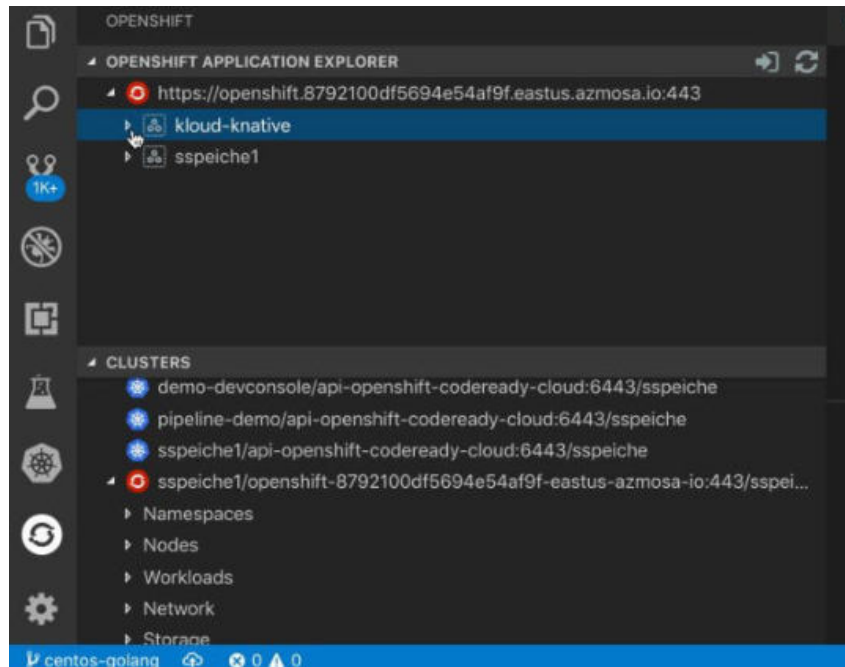
# VS Code Kubernetes Extension

## 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

A developer-focused command-line tool for rapid development iterations on OpenShift.

Available for download from Web Console

```
$ 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
```

odo - Developer-focused CLI for OpenShift

Tech Preview

OpenShift Do (odo) is a fast, iterative, and straightforward CLI tool for developers who write, build, and deploy applications on OpenShift.

odo abstracts away complex Kubernetes and OpenShift concepts, thus allowing developers to focus on what is most important to them: code.

[Download odo](#)

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

# CodeReady Containers

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.

## Details

- Linux (KVM) provides a single machine (node) instance
- Commands: setup, start, stop, delete
- PoCs exist for Windows and MacOS (VirtualBox)

Use It To: Simplify direct-to-OpenShift 4 development on laptops.

# Hosted OpenShift

Get the best of OpenShift without managing it



# Hosted OpenShift Benefits

## OPENSIFT CONTAINER PLATFORM

Full Stack Automated

Pre-existing Infrastructure

**Skip the on-call rotation**

**Red Hat engineers keep you up to date**

**Expand capacity without hassle**

## HOSTED OPENSIFT

**Azure Red Hat OpenShift**

Deploy directly from the Azure console.

Jointly managed by Red Hat and Azure engineers.

Free your team from the distraction of ops

**OpenShift Dedicated**

Powerful cluster, no maintenance needed

Managed by Red Hat engineers and support

Free your team from the distraction of ops

# Azure Red Hat OpenShift

Jointly engineered, operated, and supported by both Microsoft and Red Hat with an integrated support experience

## Experience OpenShift as a native Microsoft Azure service.

- Create fully managed OpenShift clusters in minutes using `az openshift create`
- Add or remove compute nodes to match resource demand using `az openshift scale`
- 99.9% SLA
- Inherit Azure regulatory compliance
- Pricing available at

<https://azure.microsoft.com/en-us/pricing/details/openshift/>



# OpenShift Dedicated

## **Dedicated with OpenShift 3**

Available today, hosted on Amazon Web Services

Consumption based billing now available

Bring Your Own Cloud Account

## **Dedicated with OpenShift 4**

Initial availability June 2019

Broader availability in fiscal Q2

### **OperatorHub**

Red Hat products and certified Operators will be added in a curated catalog later in the year.

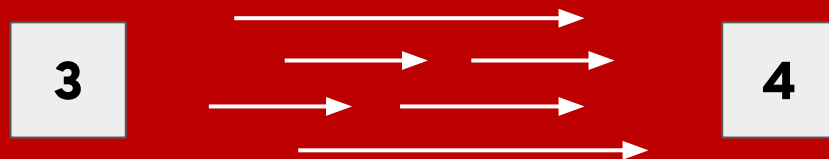
The Service Catalog and Brokers will not migrate to Dedicated due to their deprecation.

### **Connected to cloud.redhat.com**

Clusters will appear beside other self-managed installs

# Migrating to OpenShift 4

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



# App migration experience

## Using open source tooling based on Velero

Velero is an upstream project previously known as Ark. Check out [this video](#) if you are curious and want to get a sneak peek at our capabilities.

## What's moved during a migration

- Namespaces
- Persistent Volumes (move or copy)
- All important resource objects (Deployments, StatefulSets, etc)

## Available in OpenShift 4.2

Customers are anxious to get their hands on this, but we want to get it right. We would love to receive sample application workloads to test.

| Name      | Migrations | Source                     | Target         | Repository   | Persistent Volumes | Last Status           |
|-----------|------------|----------------------------|----------------|--------------|--------------------|-----------------------|
| demo plan | 2          | Summit Demo Source Cluster | Target cluster | mydemobucket | 2                  | Migrated Successfully |
| demo2     | 2          | Summit Demo Source Cluster | Target cluster | mydemobucket | 2                  | Migrated Successfully |



# DEMO

# 2019 Roadmap

## 4 June 2019 OpenShift 4.1

DEV

- OpenShift Serverless (Knative) - DP
- OpenShift Pipelines (Tekton) Dev Preview
- CodeReady Workspaces
- CodeReady Containers Alpha
- Developer CLI (odo) Beta

APP

- OperatorHub
- Operator Lifecycle Manager
- Service Mesh (~2 month after)

PLATFORM

- Kubernetes 1.13 with CRI-O runtime
- RHEL CoreOS, RHEL7
- Automated Installer for AWS
- Pre-existing Infra Installer for Bare Metal, VMware, AWS
- Automated, one-click updates
- Multus (Kubernetes multi-network)
- Quay v3

HOSTED

- cloud.redhat.com - Multi-Cluster Mgmt
- OCP Cluster Subscription Management
- Azure Red Hat OpenShift
- OpenShift Dedicated consumption pricing

## Q3/Q4 CY2019 OpenShift 4.2

DEV

- Developer Console GA
- OpenShift Serverless (Knative) - TP
- OpenShift Pipelines (Tekton) Tech Preview
- CodeReady Containers GA
- Developer CLI (odo) GA

APP

- GPU metering
- OperatorHub Enhancements
- Operator Deployment Field Forms
- Application Binding with Operators
- Application Migration Console

PLATFORM

- Kubernetes 1.14 w/ CRI-O runtime
- **Disconnected Install and Update**
- Automated Installer for Azure, OSP, GCP
- OVN Tech Preview
- FIPS
- Federation Workload API
- Automated App cert rotation
- OpenShift Container Storage 4.2

HOSTED

- cloud.redhat.com - Multi-Cluster Deployment
- Proactive Support Operator

## Q4 CY19/Q1 CY20 OpenShift 4.3

DEV

- OpenShift Serverless (Knative) - GA
- OpenShift Pipelines (Tekton) GA

APP

- Metering for Services
- Windows Containers

PLATFORM

- Kubernetes 1.15 w/ CRI-O runtime
- Automated Installer for IBM Cloud, Alibaba, RHV, Bare Metal Hardware Appliance
- Pre-existing Infra Installer for Azure, OSP, GCP
- OVN GA w/ Windows Networking Integration


HOSTED

- cloud.redhat.com - Subscription Mgmt Consumption Improvements




# Thank you & [try.openshift.com](https://try.openshift.com)

[jarek@redhat.com](mailto:jarek@redhat.com)

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

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

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

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