Red Hat OpenShift

Installation
Day 2 Operations and BackUp

Red Hat

Red Hat OpenShift

Installation

Red Hat

Installation Paradigms

OPENSHIFT CONTAINER PLATFORM

Full Stack Automated (IPI)

Simplified opinionated "Best Practices" for cluster provisioning

Fully automated installation and updates including host container OS.

Red Hat Enterprise Linux CoreOS

Pre-existing Infrastructure (UPI)

Customer managed resources & infrastructure provisioning

Plug into existing DNS and security boundaries

Red Hat Enterprise Linux CoreOS

Red Hat Enterprise Linux

HOSTED OPENSHIFT

Red Hat OpenShift on IBM Cloud * (ROKS)

Deploy directly from the IBM Cloud console. An IBM service, master nodes are managed by IBM Cloud engineers.

Azure Red Hat OpenShift ** (ARO)

Deploy directly from the Azure console. A MSFT service, jointly managed by Red Hat and Microsoft

OpenShift on AWS (ROSA)

Get a powerful cluster, fully managed by Red Hat engineers and support; a Red Hat service.

* Based on OCP v4.3 GA slated for March; public beta available now

** Entitlements of OCP obtained through a Cloud Pak purchase are not transferable to these environments

Red Hat

Hosted OpenShift

Red Hat

IBM Cloud
Dashboard
Quick start
Build
Explore IBM Cloud with this selection of easy starter tutorials and services.
Build a Virtual Cloud (VPC)
Create your ow space in the IB
7 min
IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
1830391 - Alfr...
Orchestration service
Select the container platform type and version for your cluster. For more information about versions, including links to the container platform community release notes, see the docs.
OpenShift
4.5.24
Infrastructure
Choose which network and compute environment to run your cluster on. Learn more about the differences.
Classic
Run your cluster with native subnet and VLAN networking on our classic infrastructure.
VPC
Create a fully customizable, software-defined virtual network with superior isolation using IBM Cloud VPC.
Location
Choose your location and configure your VLANs. Learn more about this.
Resource group
Default
Geography
Asia Pacific
Europe
North America
South America
Availability
Single zone
Multizone
Worker zone
Frankfurt 02
No VLANs exist.
VLANs will be created for you.
Summary
OpenShift cluster
3 Worker nodes
€1.11/hr
b3c.4x16 - 4 vCPUs 16GB RAM
Total estimated cost
€801.84/mo
Additional charges for networking and bandwidth might apply.
Actual monthly total will vary with tiered pricing.
Create
Add to estimate

Red Hat

OpenShift Container Platform (OCP)

Red Hat

OpenShift 4.13 Supported Providers

Installation Experiences

aws Outposts
AWS Local Zones
Azure
Azure Stack Hub
Alibaba Cloud
Bare Metal
Google Cloud
vmware vSphere
IBM Cloud
IBM Power Systems
IBM Z
NUTANIX
RED HAT OPENSTACK PLATFORM
RED HAT VIRTUALIZATION

Full Stack Automation

Installer Provisioned Infrastructure

- Auto-provisions infrastructure
- *KS like
- Enables self-service

Pre-existing Infrastructure

User Provisioned Infrastructure

- Bring your own hosts
- You choose infrastructure automation
- Full flexibility
- Integrate ISV solutions

Interactive – Connected

Assisted Installer

- Hosted web-based guided experience
- Agnostic, bare metal, and vSphere only
- ISO Driven

Local – Disconnected

Agent-based Installer

- Disconnected bare metal deployments
- Automated installations via CLI
- ISO driven

Red Hat

Product Manager(s): Marcos Entenza (AWS*, Azure*, GCP, IBM Cloud, Nutanix), Gaurav Singh (Alibaba), Ramon Acedo (BM, VMware), Peter Lauterbach (RHV & OCP Virtualization), Gil Rosenberg (OpenStack), & Duncan Hardie (IBM Z & Power)

OpenShift 4.12 Supported Providers & Installation Experiences

Provider	**Full Stack Automation** *Installer-provisioned infrastructure*	**Pre-existing Infrastructure** *User-provisioned infrastructure*	**Interactive – Connected** *Assisted Installer*	**Local – Disconnected** *Agent-based Installer*	**Hosted Control Planes** *(via Multicluster Engine for Kubernetes)*
Alibaba	Technology Preview				
AWS	x	x			Technology Preview
AWS Local Zones		x			
AWS Outposts	x				
Azure	x	x			Developer Preview
Azure Stack Hub	x	x			
Bare Metal	x	x	x	x	Technology Preview
Google Cloud Platform	x	x			
IBM Cloud VPC	x				
IBM Power Systems		x			
IBM Z or LinuxONE		x			
Nutanix AOS	x				
OpenShift Virtualization	Post-installation option	Post-installation option	x		Developer Preview
Red Hat OpenStack Platform	x	x			
Red Hat Virtualization	x	x			
VMware vSphere	x	x	x	x	
Agnostic (untested platform)		x	x	x	

Red Hat

Product Manager(s): Marcos Entenza (AWS*, Azure*, GCP, IBM Cloud, Nutanix), Gaurav Singh (Alibaba), Ramon Acedo (BM, VMware), Peter Lauterbach (RHV & OCP Virtualization), Gil Rosenberg (OpenStack), & Duncan Hardie (IBM Z & Power)

OpenShift 4 installation

Installer and user-provisioned infrastructure, bootstrap, and more

Red Hat

OpenShift Bootstrap Process: Self-Managed Kubernetes

How to boot a self-managed cluster:

- OpenShift 4 is unique in that management extends all the way down to the operating system
- Every machine boots with a configuration that references resources hosted in the cluster it joins enabling cluster to manage itself
- Downside is that every machine looking to join the cluster is waiting on the cluster to be created
- Dependency loop is broken using a bootstrap machine, which acts as a temporary control plane whose sole purpose is bringing up the permanent control plane nodes
- Permanent control plane nodes get booted and join the cluster leveraging the control plane on the bootstrap machine
- Once the pivot to the permanent control plane takes place, the remaining worker nodes can be booted and join the cluster

OpenShift Bootstrap Process: Step by Step

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot. Runs one instance of etcd

OpenShift Bootstrap Process: Step by Step

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot. Runs one instance of etcd
2. Master machines fetch the remote resources from the bootstrap machine and finish booting.

OpenShift Bootstrap Process: Step by Step

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot. Runs one instance of etcd
2. Master machines fetch the remote resources from the bootstrap machine and finish booting.
3. Master machines use the bootstrap node to scale the etcd cluster to 4 total instances.

OpenShift Bootstrap Process: Step by Step

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot. Runs one instance of etcd
2. Master machines fetch the remote resources from the bootstrap machine and finish booting.
3. Master machines use the bootstrap node to scale the etcd cluster to 4 total instances.
4. The Etcd operator scales itself down off the bootstrap node, leaving the etcd instance count to 3

OpenShift Bootstrap Process: Step by Step

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot. Runs one instance of etcd
2. Master machines fetch the remote resources from the bootstrap machine and finish booting.
3. Master machines use the bootstrap node to scale the etcd cluster to 3 instances.
4. The Etcd operator scales itself down off the bootstrap node, then scales back up to 3; all on the Masters
5. Bootstrap node starts a temporary Kubernetes control plane using the newly-created etcd cluster.

OpenShift Bootstrap Process: Step by Step

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot. Runs one instance of etcd
2. Master machines fetch the remote resources from the bootstrap machine and finish booting.
3. Master machines use the bootstrap node to scale the etcd cluster to 3 instances.
4. The Etcd operator scales itself down off the bootstrap node, then scales back up to 3; all on the Masters
5. Bootstrap node starts a temporary Kubernetes control plane using the newly-created etcd cluster.
6. Temporary control plane schedules the production control plane to the master machines.

OpenShift Bootstrap Process: Step by Step

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot. Runs one instance of etcd
2. Master machines fetch the remote resources from the bootstrap machine and finish booting.
3. Master machines use the bootstrap node to scale the etcd cluster to 3 instances.
4. The Etcd operator scales itself down off the bootstrap node, then scales back up to 3; all on the Masters
5. Bootstrap node starts a temporary Kubernetes control plane using the newly-created etcd cluster.
6. Temporary control plane schedules the production control plane to the master machines.
7. Temporary control plane shuts down, yielding to the production control plane.

Red Hat

OpenShift Bootstrap Process: Step by Step

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot. Runs one instance of etcd
2. Master machines fetch the remote resources from the bootstrap machine and finish booting.
3. Master machines use the bootstrap node to scale the etcd cluster to 3 instances.
4. The Etcd operator scales itself down off the bootstrap node, then scales back up to 3; all on the Masters
5. Bootstrap node starts a temporary Kubernetes control plane using the newly-created etcd cluster.
6. Temporary control plane schedules the production control plane to the master machines.
7. Temporary control plane shuts down, yielding to the production control plane.
8. Bootstrap node injects OpenShift-specific components into the newly formed control plane.

OpenShift Bootstrap Process: Step by Step

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot. Runs one instance of etcd
2. Master machines fetch the remote resources from the bootstrap machine and finish booting.
3. Master machines use the bootstrap node to scale the etcd cluster to 3 instances.
4. The Etcd operator scales itself down off the bootstrap node, then scales back up to 3; all on the Masters
5. Bootstrap node starts a temporary Kubernetes control plane using the newly-created etcd cluster.
6. Temporary control plane schedules the production control plane to the master machines.
7. Temporary control plane shuts down, yielding to the production control plane.
8. Bootstrap node injects OpenShift-specific components into the newly formed control plane.
9. Installer then tears down the bootstrap node or if user-provisioned, this needs to be performed by the administrator.

OpenShift Bootstrap Process: Step by Step

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot. Runs one instance of etcd
2. Master machines fetch the remote resources from the bootstrap machine and finish booting.
3. Master machines use the bootstrap node to scale the etcd cluster to 3 instances.
4. The Etcd operator scales itself down off the bootstrap node, then scales back up to 3; all on the Masters
5. Bootstrap node starts a temporary Kubernetes control plane using the newly-created etcd cluster.
6. Temporary control plane schedules the production control plane to the master machines.
7. Temporary control plane shuts down, yielding to the production control plane.
8. Bootstrap node injects OpenShift-specific components into the newly formed control plane.
9. Installer then tears down the bootstrap node or if user-provisioned, this needs to be performed by the administrator.
10. Worker machines fetch remote resources from masters and finish booting.

How everything deployed comes under management

Masters (Special)

- Full Stack Automated: Installer provisions minimal viable masters
- User Provisioned: User/Administrator provisions minimal viable masters
- Machine API adopts existing masters post-provision
- Each master is a standalone Machine object
- Termination protection (avoid self-destruction)

Workers

- Each Machine Pool corresponds to MachineSet
- Optionally autoscale (min,max) and health check (replace if not ready > X minutes)

Multi-AZ

- MachineSets scoped to single AZ
- Installer stripes N machine sets across AZs by default
- Post-install best effort balance via cluster autoscaler

One Touch provisioning via Ignition

Machine generated; Machine validated

Ignition applies a declarative node configuration early in the boot process. Unifies kickstart and cloud-init.

- Generated via openshift-install
- Configures storage, systemd units, users, & remote configs
- Executed in the initramfs
- Configuration for masters & workers is served from the control plane and sourced from Machine Configs

```
{
  "ignition": {
    "config": {},
    "timeouts": {},
    "version": "2.1.0"
  },
  "passwd": {
    "users": [
      {
        "name": "core",
        "passwordHash": "$6$43y3tkl...",
        "sshAuthorizedKeys": [
          "key1"
        ]
      }
    ]
  },
  "storage": {},
  "systemd": {}
}
```

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
- [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>
```


Red Hat

Pre-existing Infrastructure Installation (aka UPI)

Red Hat

Comparison of Paradigms

	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 8
Node Provisioning / Autoscaling	Yes	Only for providers with OpenShift Machine API support

Disconnected “Air-gapped” Installation & Upgrading

Overview

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

Installation Procedure

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

Red Hat

Mirror the Registry with QUAY

3.2.1. Prerequisites

- An OpenShift Container Platform subscription.
- **Red Hat Enterprise Linux (RHEL) 8 with Podman 3.3 installed.**
- Fully qualified domain name for the Red Hat Quay service, which must resolve through a DNS server.
- Passwordless `sudo` access on the target host.
- Key-based SSH connectivity on the target host. SSH keys are automatically generated for local installs. For remote hosts, you must generate your own SSH keys.

Red Hat

OpenShift Update Service

Update manager for your clusters in restricted or disconnected networks

- OpenShift Update Service (OSUS) is the on-premise release of Red Hat's hosted update service
- Supports the publishing of upgrade graph information to clusters in restricted networks
- Provides clusters with a list of next recommended update versions based on the current version installed on the cluster
- Comprised of two services:
  - **Graph Builder:** Fetches OpenShift release payload information (primary metadata) from any container registry (compatible with Docker registry V2 API) and builds a directed acyclic graph (DAG) representing valid upgrade edges
  - **Policy Engine:** Responsible for selectively serving updates to every cluster by altering a client's view of the graph with a set of filters
- GA release planned for post-4.6 and will be distributed on Operator Hub as an optional add-on operator
- Blog post announcing OpenShift Update Service
- https://github.com/openshift/cincinnati-operator

Red Hat

Red Hat
OpenShift

OpenShift
Day two operations

Alfred Bach
PSA Red Hat EMEA

Red Hat

You have installed your first OpenShift Cluster and logged in first time with “kubeadmin”

Now let's configure the basics

User Management

Alert receivers

Connect to an User Source

Click on the Blue Line of your graphical Console and ad an usersource e.g. HTpasswd

Create a group with cluster admin privileges

And ad one or two users to this group

Configure the alert receivers

To receive alerts from the OCP CLuster

Red Hat

Section 1

User Management

Alert Receivers

Add some useful tools and add persistence storage to the registry

Customize your OpenShift environment

Add a custom banner easy identify your cluster

With a custom resource definition - console notification

Add a browser based terminal

With the terminal operator

extend the lifetime of the communication certificates

With the oc command in the terminal window

```
oc -n openshift-kube-apiserver-operator get secret
kube-apiserver-to-kubelet-signer -o
jsonpath='{.metadata.annotations.auth\.openshift\.io/certificate-not-aft
er}'
```

Configure you registry to use persistent storage

With the oc command in the terminal window

```
oc patch configs.imageregistry.operator.openshift.io cluster --type merge
--patch '{"spec":{"managementState":"Managed"}}'
oc patch configs.imageregistry.operator.openshift.io cluster --type merge
--patch '{"spec":{"storage":{"emptyDir":{}}}}'
```

Remove the “kubeadmin” user

With the oc command in the terminal window

```
oc delete secrets kubeadmin -n kube-system
```

Section 2

Banner,
Terminal
and registry

Red Hat

No we add persistent storage to our cluster and use it for storing net-observe data

Adding external persistent storage and use it for Net-Observe

Add a separate disk to your cluster and use it with LVM

Adding an empty virtual disc and installing the LVM Storage operator

Installing the Net-Observe Operator

And configure it that it runs with LOKI on local storage

Looking up our cluster in the Red Hat Portal

And investigate in OpenShift Insights

Red Hat

Section 3

External storage

Net Observe

Red Hat

What you have seen in this section

Red Hat

Red Hat OpenShift

Backup and Disaster Recovery

Red Hat

BACKUP an OPENSHIFT Cluster

Backup Solutions for Red Hat OpenShift

Business Continuity

OpenShift OADP 1.1 - Native backup utility with 4.12

- **Application** level (Namespace), **consistent backups with OADP**
- CLI based **scheduling and management** of backups
- **Built-in data mover** enables CSI-based storage snapshots to be backed up to a remote S3 compatible object store.
  - **Plugable DataMover** support is Tech Preview (ie. VolSync)
- **Supports all OpenShift storage provisioners** that also support CSI Snapshots

Product Manager: Jeff Brent

OADP Operator

OpenShift API for Data Protection (OADP) supports the following features:

Backup

You can back up all resources in your cluster or you can filter the resources by type, namespace, or label.

OADP backs up Kubernetes objects and internal images by saving them as an archive file on object storage. Restore

You can restore resources and PVs from a backup. You can restore all objects in a backup or filter the restored objects by namespace, PV, or label.

Schedule

You can schedule backups at specified intervals.

Hooks

You can use hooks to run commands in a container on a pod

Red Hat

Business Continuity and Disaster recovery

Red Hat

OCP Application Backup – Key Components

- Third party backup Applications /native CLI
  - Handles Backup policy management, backup scheduling, retention and restore management and data movement
- OADP – OpenShift App Data Protection API
  - Enables namespace or label scoped backups with all ensuing cluster resources and application data (PVs)
  - Ensures OCP version independence and works across storage providers (via plug-ins)
- ODF PV Snapshots via CSI
  - PV/PVC backups of ODF volumes through standard CSI interfaces
  - Can be used with or without OADP

Resiliency Solutions for different service level objectives

- Comprehensive protection solutions against wide spectrum of failures
- Beyond Data Protection -→ Full Application protection
- Resiliency built into the platform – Available to all stateful and stateless applications on OpenShift
- OCP + ODF + ACM integrated stack lends towards Automated and Simplified Application granular protection

Red Hat

OCP Integrated, Full Stack DR Protection

Multi-Site Multi Cluster Manager

RHACM

- ODR Hub Operator – Orchestrates & Automates DR operations across clusters

Platform

Red Hat OpenShift

- ODR Cluster Operator - Manages and synchronizes cluster meta data and application data

Persistence Layer

Red Hat OpenShift Data Foundation

- Asynchronous replication of Application volumes – or-
- Synchronous Mirroring of Application volumes

- Easy configuration DR across cluster and sites as part of application deployment
- Automated DR Failover and Failback operations reduces RTO
- Manage and Monitor DR across clusters and Apps
- Same consistent DR operations for both Metro-DR and Regional-DR
- Both Application Data and State is protected and used for Application granular protection
- Consistent Data replication or mirroring or both based on infrastructure and desired protection.

OCP Cluster HA + DR for stateful Applications

	Cluster HA Metro-DR	Regional-DR	
Topology	Single OCP+ODF clusters deployed over multiple AZs in a single region	Multi OCP + ODF clusters spread over multiple regions	Multi OCP clusters + single external ODF stretched cluster deployed over low latency networks
RTO (Downtime)	RTO=0 (Continuous)*	RTO = minutes DR Automation from ACM+ODF reduces RTO	RTO = minutes DR Automation from ACM+ODF reduces RTO
RPO (Data loss exposure)	RPO=0 No Data loss due due to Synchronous mirroring of ODF data	RPO > 0; Usually 5 min or higher Depends upon network bandwidth & change rate	RPO=0 No Data loss due due to Synchronous mirroring of ODF data
Infra Requirements	Multi-AZ supported public clouds (vSphere support in OCP 4.10)	All ODF supported platforms No network latency limits	On-prem only (vSphere, bare metal) <10ms network latency between sites

* Subject to RWO PV limitations

Red Hat

OpenShift in vSphere is Zone Aware

- Create highly-available OpenShift clusters in vSphere with installer provisioned infrastructure (IPI)
- Applies zonal tags (regions and zones) to multiple vCenter datacenters and clusters in a single vCenter
- Excludes User Provisioned infrastructure (UPI) deployments

Technology Preview

Product Manager: Ramon Acedo Rodriguez, Ju Lim

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make
Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat