OpenShift Container Platform 4.9

About

Introduction to OpenShift Container Platform

Last Updated: 2023-04-11

OpenShift Container Platform 4.9 About

Introduction to OpenShift Container Platform

Legal Notice

Abstract

This document provides an overview of the OpenShift Container Platform features.

Table of Contents

CHAPTER 1. OPENSHIFT CONTAINER PLATFORM 4.9 DOCUMENTATION

Welcome to the official OpenShift Container Platform 4.9 documentation, where you can learn about OpenShift Container Platform and start exploring its features.

To navigate the OpenShift Container Platform 4.9 documentation, you can use one of the following methods:

- Use the left navigation bar to browse the documentation.
- Select the task that interests you from the contents of this Welcome page.

Start with Architecture and Security and compliance. Then, see the release notes.

1.1. CLUSTER INSTALLER ACTIVITIES

Explore these OpenShift Container Platform installation tasks.

- **OpenShift Container Platform installation overview**: You can install OpenShift Container Platform on installer-provisioned or user-provisioned infrastructure. The OpenShift Container Platform installation program provides the flexibility to deploy OpenShift Container Platform on a range of different platforms.
- **Install a cluster on AWS**: You have many installation options when you deploy a cluster on Amazon Web Services (AWS). You can deploy clusters with default settings or custom AWS settings. You can also deploy a cluster on AWS infrastructure that you provisioned yourself. You can modify the provided AWS CloudFormation templates to meet your needs.
- **Install a cluster on Azure**: You can deploy clusters with default settings, custom Azure settings, or custom networking settings in Microsoft Azure. You can also provision OpenShift Container Platform into an Azure Virtual Network or use Azure Resource Manager Templates to provision your own infrastructure.
- **Install a cluster on Azure Stack Hub**: You can install OpenShift Container Platform on Azure Stack Hub on user-provisioned infrastructure.
- **Install a cluster on GCP**: You can deploy clusters with default settings or custom GCP settings on Google Cloud Platform (GCP). You can also perform a GCP installation where you provision your own infrastructure.
- **Install a cluster on VMware vSphere**: You can install OpenShift Container Platform on supported versions of vSphere.
- **Install a cluster on VMware Cloud**: You can install OpenShift Container Platform on supported versions of VMware Cloud (VMC) on AWS.
- **Install a cluster with z/VM on IBM Z and LinuxONE**: You can install OpenShift Container Platform with z/VM on IBM Z and LinuxONE on user-provisioned infrastructure.
- **Install a cluster with RHEL KVM on IBM Z and LinuxONE**: You can install OpenShift Container Platform with RHEL KVM on IBM Z and LinuxONE on user-provisioned infrastructure.
- **Install a cluster on IBM Power**: You can install OpenShift Container Platform on IBM Power on user-provisioned infrastructure.

- **Install an installer-provisioned cluster on bare metal**: You can install OpenShift Container Platform on bare metal with an installer-provisioned architecture.
- **Install a user-provisioned cluster on bare metal**: If none of the available platform and cloud provider deployment options meet your needs, you can install OpenShift Container Platform on user-provisioned bare metal infrastructure.
- **Install a cluster on Red Hat OpenStack Platform (RHOSP)**: You can install a cluster on RHOSP with customizations, with network customizations, or on a restricted network on installer-provisioned infrastructure.
  You can install a cluster on RHOSP with customizations, with network customizations, or with SR-IOV on user-provisioned infrastructure.
- **Install a cluster on Red Hat Virtualization (RHV)**: You can deploy clusters on Red Hat Virtualization (RHV) with a quick install or an install with customizations.
- **Install a cluster in a restricted network**: If your cluster that uses user-provisioned infrastructure on AWS, GCP, vSphere, IBM Z and LinuxONE with z/VM, IBM Z and LinuxONE with RHEL KVM, IBM Power, or bare metal does not have full access to the internet, then mirror the OpenShift Container Platform installation images and install a cluster in a restricted network.
- **Install a cluster in an existing network**: If you use an existing Virtual Private Cloud (VPC) in AWS or GCP or an existing VNet on Azure, you can install a cluster.
- **Install a private cluster**: If your cluster does not require external internet access, you can install a private cluster on AWS, Azure, or GCP. Internet access is still required to access the cloud APIs and installation media.
- **Check installation logs**: Access installation logs to evaluate issues that occur during OpenShift Container Platform 4.9 installation.
- **Access OpenShift Container Platform**: Use credentials output at the end of the installation process to log in to the OpenShift Container Platform cluster from the command line or web console.
- **Install Red Hat OpenShift Container Storage**: You can install Red Hat OpenShift Container Storage as an Operator to provide highly integrated and simplified persistent storage management for containers.

1.2. DEVELOPER ACTIVITIES

Develop and deploy containerized applications with OpenShift Container Platform. OpenShift Container Platform is a platform for developing and deploying containerized applications. OpenShift Container Platform documentation helps you:

- **Understand OpenShift Container Platform development**: Learn the different types of containerized applications, from simple containers to advanced Kubernetes deployments and Operators.
- **Work with projects**: Create projects from the OpenShift Container Platform web console or OpenShift CLI (**oc**) to organize and share the software you develop.
- **Work with applications**:

Use the **Developer** perspective in the OpenShift Container Platform web console to create and deploy applications.

Use the **Topology** view to see your applications, monitor status, connect and group components, and modify your code base.

- **Connect your workloads to backing services**: The Service Binding Operator enables application developers to easily bind workloads with Operator-managed backing services by automatically collecting and sharing binding data with the workloads. The Service Binding Operator improves the development lifecycle with a consistent and declarative service binding method that prevents discrepancies in cluster environments.

- **Use the developer CLI tool (odo)**: The **odo** CLI tool lets developers create single or multi-component applications easily and automates deployment, build, and service route configurations. It abstracts complex Kubernetes and OpenShift Container Platform concepts, allowing you to focus on developing your applications.

- **Create CI/CD Pipelines**: Pipelines are serverless, cloud-native, continuous integration and continuous deployment systems that run in isolated containers. They use standard Tekton custom resources to automate deployments and are designed for decentralized teams that work on microservice-based architecture.

- **Deploy Helm charts**: Helm is a software package manager that simplifies deployment of applications and services to OpenShift Container Platform clusters. Helm uses a packaging format called charts. A Helm chart is a collection of files that describes the OpenShift Container Platform resources.

- **Understand image builds**: Choose from different build strategies (Docker, S2I, custom, and pipeline) that can include different kinds of source materials (from places like Git repositories, local binary inputs, and external artifacts). Then, follow examples of build types from basic builds to advanced builds.

- **Create container images**: A container image is the most basic building block in OpenShift Container Platform (and Kubernetes) applications. Defining image streams lets you gather multiple versions of an image in one place as you continue its development. S2I containers let you insert your source code into a base container that is set up to run code of a particular type, such as Ruby, Node.js, or Python.

- **Create deployments**: Use **Deployment** and **DeploymentConfig** objects to exert fine-grained management over applications. Manage deployments using the **Workloads** page or OpenShift CLI (**oc**). Learn rolling, recreate, and custom deployment strategies.

- **Create templates**: Use existing templates or create your own templates that describe how an application is built or deployed. A template can combine images with descriptions, parameters, replicas, exposed ports and other content that defines how an application can be run or built.

- **Understand Operators**: Operators are the preferred method for creating on-cluster applications for OpenShift Container Platform 4.9. Learn about the Operator Framework and how to deploy applications using installed Operators into your projects.

- **Develop Operators**: Operators are the preferred method for creating on-cluster applications for OpenShift Container Platform 4.9. Learn the workflow for building, testing, and deploying Operators. Then, create your own Operators based on Ansible or Helm, or configure built-in Prometheus monitoring using the Operator SDK.

- **REST API reference**: Learn about OpenShift Container Platform application programming interface endpoints.

1.3. CLUSTER ADMINISTRATOR ACTIVITIES

Manage machines, provide services to users, and follow monitoring and logging reports. This documentation helps you:

- **Understand OpenShift Container Platform management**: Learn about components of the OpenShift Container Platform 4.9 control plane. See how OpenShift Container Platform control plane and worker nodes are managed and updated through the Machine API and Operators.

1.3.1. Manage cluster components

- **Manage machines**: Manage machines in your cluster on AWS, Azure, or GCP by deploying health checks and applying autoscaling to machines.
- **Manage container registries**: Each OpenShift Container Platform cluster includes a built-in container registry for storing its images. You can also configure a separate Red Hat Quay registry to use with OpenShift Container Platform. The Quay.io web site provides a public container registry that stores OpenShift Container Platform containers and Operators.
- **Manage users and groups**: Add users and groups with different levels of permissions to use or modify clusters.
- **Manage authentication**: Learn how user, group, and API authentication works in OpenShift Container Platform. OpenShift Container Platform supports multiple identity providers, including:
  - HTPasswd
  - Keystone
  - LDAP
  - basic authentication
  - request header
  - GitHub
  - GitLab
  - Google
  - OpenID
- **Manage ingress, API server, and service certificates**: OpenShift Container Platform creates certificates by default for the Ingress Operator, the API server, and for services needed by complex middleware applications that require encryption. You might need to change, add, or rotate these certificates.
- **Manage networking**: The cluster network in OpenShift Container Platform is managed by the Cluster Network Operator (CNO). The CNO uses iptables rules in kube-proxy to direct traffic between nodes and pods running on those nodes. The Multus Container Network Interface adds the capability to attach multiple network interfaces to a pod. Using network policy features, you can isolate your pods or permit selected traffic.
- **Manage storage**: OpenShift Container Platform allows cluster administrators to configure persistent storage using Red Hat OpenShift Container Storage , AWS Elastic Block Store, NFS,

iSCSI, Container Storage Interface (CSI), and more. You can expand persistent volumes, configure dynamic provisioning, and use CSI to configure, clone, and use snapshots of persistent storage.

- **Manage Operators**: Lists of Red Hat, ISV, and community Operators can be reviewed by cluster administrators and installed on their clusters. After you install them, you can run, upgrade, back up, or otherwise manage the Operator on your cluster.

1.3.2. Change cluster components

- **Use custom resource definitions (CRDs) to modify the cluster** Cluster features implemented with Operators can be modified with CRDs. Learn to create a CRD and manage resources from CRDs.
- **Set resource quotas**: Choose from CPU, memory, and other system resources to set quotas.
- **Prune and reclaim resources**: Reclaim space by pruning unneeded Operators, groups, deployments, builds, images, registries, and cron jobs.
- **Scale and tune clusters**: Set cluster limits, tune nodes, scale cluster monitoring, and optimize networking, storage, and routes for your environment.
- **Update a cluster**: Use the Cluster Version Operator (CVO) to upgrade your OpenShift Container Platform cluster. If an update is available from the OpenShift Update Service (OSUS), you apply that cluster update from either the OpenShift Container Platform web console or the OpenShift CLI (**oc**).
- **Understanding the OpenShift Update Service**: Learn about installing and managing a local OpenShift Update Service for recommending OpenShift Container Platform updates in disconnected environments.

1.3.3. Monitor the cluster

- **Work with OpenShift Logging**: Learn about OpenShift Logging and configure different OpenShift Logging types, such as Elasticsearch, Fluentd, and Kibana.
- **Monitor clusters**: Learn to configure the monitoring stack. After configuring monitoring, use the web console to access monitoring dashboards. In addition to infrastructure metrics, you can also scrape and view metrics for your own services.
- **Remote health monitoring**: OpenShift Container Platform collects anonymized aggregated information about your cluster. Using Telemetry and the Insights Operator, this data is received by Red Hat and used to improve OpenShift Container Platform. You can view the data collected by remote health monitoring.

CHAPTER 2. LEARN MORE ABOUT OPENSHIFT CONTAINER PLATFORM

Use the following sections to find content to help you learn about and use OpenShift Container Platform.

2.1. ARCHITECT

Learn about OpenShift Container Platform	Plan an OpenShift Container Platform deployment	Additional resources
Enterprise Kubernetes with OpenShift	Tested platforms	OpenShift blog
Architecture	Security and compliance	What's new in OpenShift Container Platform
	Networking	OpenShift Container Platform life cycle
	Backup and restore	

2.2. CLUSTER ADMINISTRATOR

Learn about OpenShift Container Platform	Deploy OpenShift Container Platform	Manage OpenShift Container Platform	Additional resources
Enterprise Kubernetes with OpenShift	Installing OpenShift Container Platform	Using Insights to identify issues with your cluster	Getting Support
Architecture	Post installation configuration	Logging	OpenShift Knowledgebase articles
OpenShift Interactive Learning Portal	Networking	Monitoring	OpenShift Container Platform Life Cycle
	Storage		
	Backup and restore		
	Updating a cluster		

2.3. APPLICATION SITE RELIABILITY ENGINEER (APP SRE)

Learn about OpenShift Container Platform	Deploy and manage applications	Additional resources
OpenShift Interactive Learning Portal	Projects	Getting Support
Architecture	Operators	OpenShift Knowledgebase articles
	Logging	OpenShift Container Platform Life Cycle
	Blogs about logging	
	Monitoring	

2.4. DEVELOPER

Learn about application development in OpenShift Container Platform	Deploy applications
Getting started with OpenShift for developers (interactive tutorial)	Creating applications
Red Hat Developers site	Builds
Red Hat CodeReady Workspaces	Operators
	Images
	Developer-focused CLI

2.5. UNDERSTANDING OPENSHIFT CONTAINER PLATFORM

OpenShift Container Platform is a Kubernetes environment for managing the lifecycle of container-based applications and their dependencies on various computing platforms, such as bare metal, virtualized, on-premise, and in cloud. OpenShift Container Platform deploys, configures and manages containers. OpenShift Container Platform offers usability, stability, and customization of its components.

OpenShift Container Platform utilises a number of computing resources, known as nodes. A node has a lightweight, secure operating system based on Red Hat Enterprise Linux (RHEL), known as Red Hat Enterprise Linux CoreOS (RHCOS).

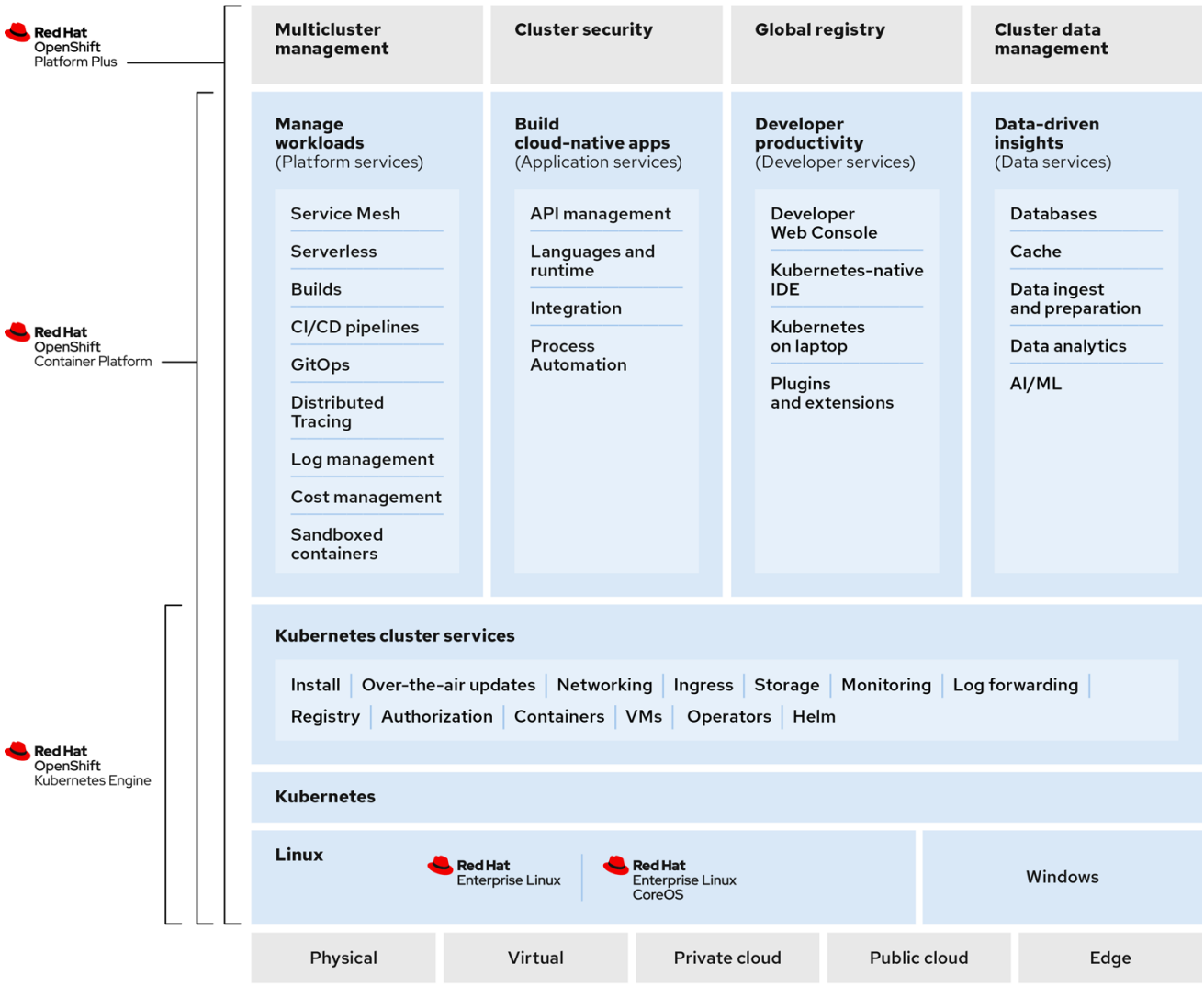
After a node is booted and configured, it obtains a container runtime, such as CRI-O or Docker, for managing and running the images of container workloads scheduled to it. The Kubernetes agent, or kubelet schedules container workloads on the node. The kubelet is responsible for registering the node with the cluster and receiving the details of container workloads.

OpenShift Container Platform configures and manages the networking, load balancing and routing of the cluster. OpenShift Container Platform adds cluster services for monitoring the cluster health and performance, logging, and for managing upgrades.

The container image registry and OperatorHub provide Red Hat certified products and community built softwares for providing various application services within the cluster. These applications and services manage the applications deployed in the cluster, databases, frontends and user interfaces, application runtimes and business automation, and developer services for development and testing of container applications.

You can manage applications within the cluster either manually by configuring deployments of containers running from pre-built images or through resources known as Operators. You can build custom images from pre-build images and source code, and store these custom images locally in an internal, private or public registry.

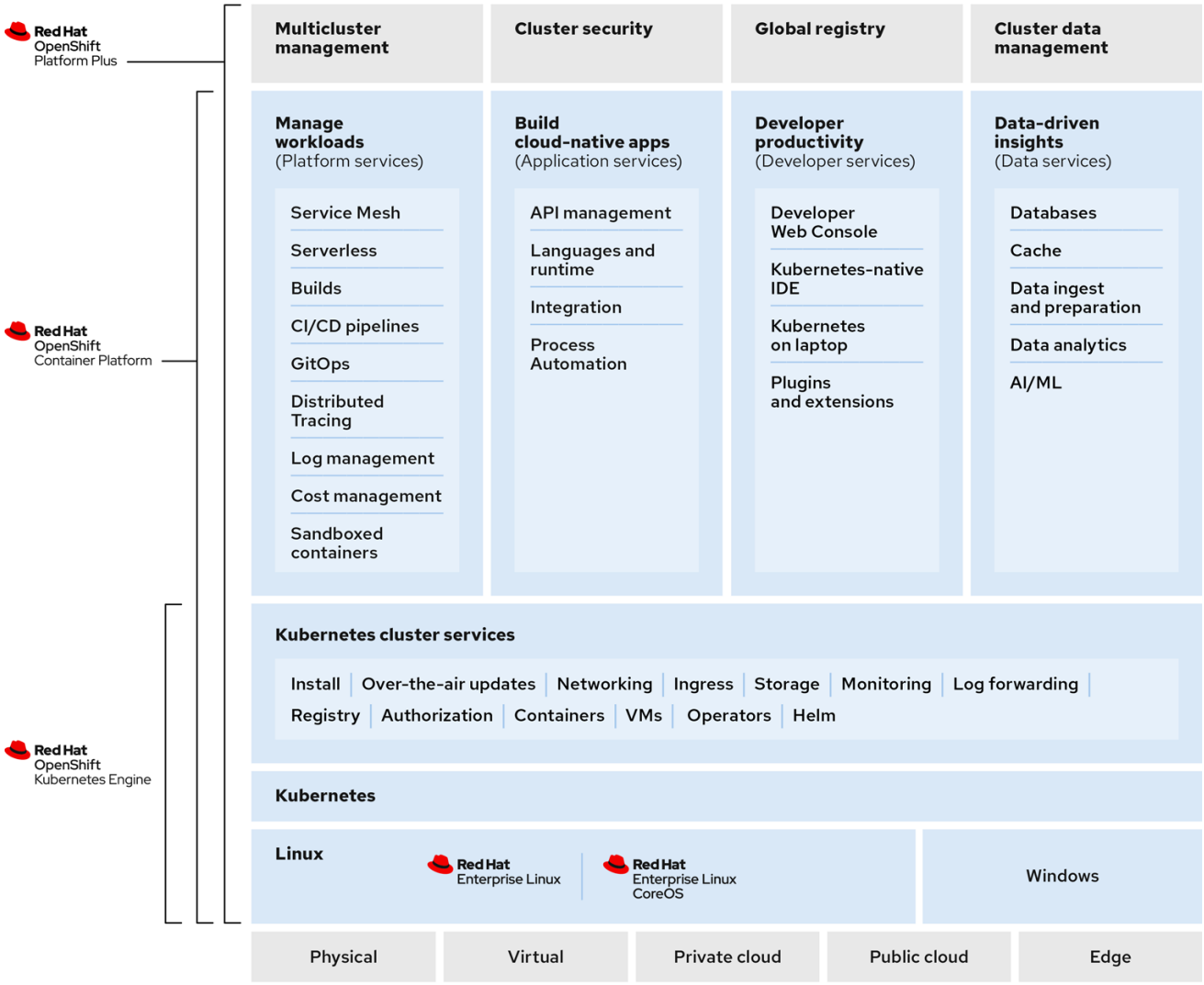
The Multicluster Management layer can manage multiple clusters including their deployment, configuration, compliance and distribution of workloads in a single console.



277_OpenShift_1122

CHAPTER 3. ABOUT OPENSHIFT KUBERNETES ENGINE

As of 27 April 2020, Red Hat has decided to rename Red Hat OpenShift Container Engine to Red Hat OpenShift Kubernetes Engine to better communicate what value the product offering delivers.



Red Hat OpenShift Kubernetes Engine is a product offering from Red Hat that lets you use an enterprise class Kubernetes platform as a production platform for launching containers. You download and install OpenShift Kubernetes Engine the same way as OpenShift Container Platform as they are the same binary distribution, but OpenShift Kubernetes Engine offers a subset of the features that OpenShift Container Platform offers.

3.1. SIMILARITIES AND DIFFERENCES

You can see the similarities and differences between OpenShift Kubernetes Engine and OpenShift Container Platform in the following table:

Table 3.1. Product comparison for OpenShift Kubernetes Engine and OpenShift Container Platform

	OpenShift Kubernetes Engine	OpenShift Container Platform
Fully Automated Installers	Yes	Yes
Over the Air Smart Upgrades	Yes	Yes
Enterprise Secured Kubernetes	Yes	Yes
Kubectl and oc automated command line	Yes	Yes
Operator Lifecycle Manager (OLM)	Yes	Yes
Administrator Web console	Yes	Yes
OpenShift Virtualization	Yes	Yes
User Workload Monitoring		Yes
Metering and Cost Management SaaS Service		Yes
Platform Logging		Yes
Developer Web Console		Yes
Developer Application Catalog		Yes
Source to Image and Builder Automation (Tekton)		Yes
OpenShift Service Mesh (Maistra, Kiali, and Jaeger)		Yes
OpenShift distributed tracing (Jaeger)		Yes
OpenShift Serverless (Knative)		Yes
OpenShift Pipelines (Jenkins and Tekton)		Yes
Embedded Component of IBM Cloud Pak and RHT MW Bundles		Yes

3.1.1. Core Kubernetes and container orchestration

OpenShift Kubernetes Engine offers full access to an enterprise-ready Kubernetes environment that is easy to install and offers an extensive compatibility test matrix with many of the software elements that you might use in your data center.

OpenShift Kubernetes Engine offers the same service level agreements, bug fixes, and common

vulnerabilities and errors protection as OpenShift Container Platform. OpenShift Kubernetes Engine includes a Red Hat Enterprise Linux (RHEL) Virtual Datacenter and Red Hat Enterprise Linux CoreOS (RHCOS) entitlement that allows you to use an integrated Linux operating system with container runtime from the same technology provider.

The OpenShift Kubernetes Engine subscription is compatible with the Red Hat OpenShift support for Windows Containers subscription.

3.1.2. Enterprise-ready configurations

OpenShift Kubernetes Engine uses the same security options and default settings as the OpenShift Container Platform. Default security context constraints, pod security policies, best practice network and storage settings, service account configuration, SELinux integration, HAproxy edge routing configuration, and all other standard protections that OpenShift Container Platform offers are available in OpenShift Kubernetes Engine. OpenShift Kubernetes Engine offers full access to the integrated monitoring solution that OpenShift Container Platform uses, which is based on Prometheus and offers deep coverage and alerting for common Kubernetes issues.

OpenShift Kubernetes Engine uses the same installation and upgrade automation as OpenShift Container Platform.

3.1.3. Standard infrastructure services

With an OpenShift Kubernetes Engine subscription, you receive support for all storage plugins that OpenShift Container Platform supports.

In terms of networking, OpenShift Kubernetes Engine offers full and supported access to the Kubernetes Container Network Interface (CNI) and therefore allows you to use any third-party SDN that supports OpenShift Container Platform. It also allows you to use the included Open vSwitch software defined network to its fullest extent. OpenShift Kubernetes Engine allows you to take full advantage of the OVN Kubernetes overlay, Multus, and Multus plugins that are supported on OpenShift Container Platform. OpenShift Kubernetes Engine allows customers to use a Kubernetes Network Policy to create microsegmentation between deployed application services on the cluster.

You can also use the **Route** API objects that are found in OpenShift Container Platform, including its sophisticated integration with the HAproxy edge routing layer as an out of the box Kubernetes ingress controller.

3.1.4. Core user experience

OpenShift Kubernetes Engine users have full access to Kubernetes Operators, pod deployment strategies, Helm, and OpenShift Container Platform templates. OpenShift Kubernetes Engine users can use both the **oc** and **kubectl** command line interfaces. OpenShift Kubernetes Engine also offers an administrator web-based console that shows all aspects of the deployed container services and offers a container-as-a service experience. OpenShift Kubernetes Engine grants access to the Operator Life Cycle Manager that helps you control access to content on the cluster and life cycle operator-enabled services that you use. With an OpenShift Kubernetes Engine subscription, you receive access to the Kubernetes namespace, the OpenShift **Project** API object, and cluster-level Prometheus monitoring metrics and events.

3.1.5. Maintained and curated content

With an OpenShift Kubernetes Engine subscription, you receive access to the OpenShift Container Platform content from the Red Hat Ecosystem Catalog and Red Hat Connect ISV marketplace. You can access all maintained and curated content that the OpenShift Container Platform eco-system offers.

3.1.6. OpenShift Container Storage compatible

OpenShift Kubernetes Engine is compatible and supported with your purchase of OpenShift Container Storage.

3.1.7. Red Hat Middleware compatible

OpenShift Kubernetes Engine is compatible and supported with individual Red Hat Middleware product solutions. Red Hat Middleware Bundles that include OpenShift embedded in them only contain OpenShift Container Platform.

3.1.8. OpenShift Serverless

OpenShift Kubernetes Engine does not include OpenShift Serverless support. Use OpenShift Container Platform for this support.

3.1.9. Quay Integration compatible

OpenShift Kubernetes Engine is compatible and supported with a Red Hat Quay purchase.

3.1.10. OpenShift Virtualization

OpenShift Kubernetes Engine includes support for the Red Hat product offerings derived from the kubevirt.io open source project.

3.1.11. Advanced cluster management

OpenShift Kubernetes Engine is compatible with your additional purchase of Red Hat Advanced Cluster Management (RHACM) for Kubernetes. An OpenShift Kubernetes Engine subscription does not offer a cluster-wide log aggregation solution or support Elasticsearch, Fluentd, or Kibana based logging solutions. Similarly, the chargeback features found in OpenShift Container Platform or the console.redhat.com Cost Management SaaS service are not supported with OpenShift Kubernetes Engine. Red Hat Service Mesh capabilities derived from the open source istio.io and kiali.io projects that offer OpenTracing observability for containerized services on OpenShift Container Platform are not supported in OpenShift Kubernetes Engine.

3.1.12. Advanced networking

The standard networking solutions in OpenShift Container Platform are supported with an OpenShift Kubernetes Engine subscription. OpenShift Container Platform's Kubernetes CNI plugin for automation of multi-tenant network segmentation between OpenShift Container Platform projects is entitled for use with OpenShift Kubernetes Engine. OpenShift Kubernetes Engine offers all the granular control of the source IP addresses that are used by application services on the cluster. Those egress IP address controls are entitled for use with OpenShift Kubernetes Engine. OpenShift Container Platform offers ingress routing to on cluster services that use non-standard ports when no public cloud provider is in use via the VIP pods found in OpenShift Container Platform. That ingress solution is supported in OpenShift Kubernetes Engine. OpenShift Kubernetes Engine users are supported for the Kubernetes ingress control object, which offers integrations with public cloud providers. Red Hat Service Mesh, which is derived from the istio.io open source project, is not supported in OpenShift Kubernetes Engine. Also, the Kourier ingress controller found in OpenShift Serverless is not supported on OpenShift Kubernetes Engine.

3.1.13. Developer experience

With OpenShift Kubernetes Engine, the following capabilities are not supported:

- The CodeReady developer experience utilities and tools, such as CodeReady Workspaces.
- OpenShift Container Platform's pipeline feature that integrates a streamlined, Kubernetes-enabled Jenkins and Tekton experience in the user's project space.
- The OpenShift Container Platform's source-to-image feature, which allows you to easily deploy source code, dockerfiles, or container images across the cluster.
- Build strategies, builder pods, or Tekton for end user container deployments.
- The **odo** developer command line.
- The developer persona in the OpenShift Container Platform web console.

3.1.14. Feature summary

The following table is a summary of the feature availability in OpenShift Kubernetes Engine and OpenShift Container Platform. Where applicable, it includes the name of the Operator that enables a feature.

Table 3.2. Features in OpenShift Kubernetes Engine and OpenShift Container Platform

Feature	OpenShift Kubernetes Engine	OpenShift Container Platform	Operator name
Fully Automated Installers (IPI)	Included	Included	N/A
Customizable Installers (UPI)	Included	Included	N/A
Disconnected Installation	Included	Included	N/A
Red Hat Enterprise Linux (RHEL) or Red Hat Enterprise Linux CoreOS (RHCOS) entitlement	Included	Included	N/A
Existing RHEL manual attach to cluster (BYO)	Included	Included	N/A
CRIO Runtime	Included	Included	N/A
Over the Air Smart Upgrades and Operating System (RHCOS) Management	Included	Included	N/A

Feature	OpenShift Kubernetes Engine	OpenShift Container Platform	Operator name
Enterprise Secured Kubernetes	Included	Included	N/A
Kubectl and oc automated command line	Included	Included	N/A
Auth Integrations, RBAC, SCC, Multi-Tenancy Admission Controller	Included	Included	N/A
Operator Lifecycle Manager (OLM)	Included	Included	N/A
Administrator web console	Included	Included	N/A
OpenShift Virtualization	Included	Included	OpenShift Virtualization Operator
Compliance Operator provided by Red Hat	Included	Included	Compliance Operator
File Integrity Operator	Included	Included	File Integrity Operator
Gatekeeper Operator	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Gatekeeper Operator
Klusterlet	Not Included - Requires separate subscription	Not Included - Requires separate subscription	N/A
Kube Descheduler Operator provided by Red Hat	Included	Included	Kube Descheduler Operator
Local Storage provided by Red Hat	Included	Included	Local Storage Operator
Node Feature Discovery provided by Red Hat	Included	Included	Node Feature Discovery Operator
Performance Add-on Operator	Included	Included	Performance Add-on Operator

Feature	OpenShift Kubernetes Engine	OpenShift Container Platform	Operator name
PTP Operator provided by Red Hat	Included	Included	PTP Operator
Service Telemetry Operator provided by Red Hat	Included	Included	Service Telemetry Operator
SR-IOV Network Operator	Included	Included	SR-IOV Network Operator
Vertical Pod Autoscaler	Included	Included	Vertical Pod Autoscaler
Cluster Monitoring (Prometheus)	Included	Included	Cluster Monitoring
Device Manager (for example, GPU)	Included	Included	N/A
Log Forwarding (with fluentd)	Included	Included	Red Hat OpenShift Logging Operator (for log forwarding with fluentd)
Telemeter and Insights Connected Experience	Included	Included	N/A
Feature	**OpenShift Kubernetes Engine**	**OpenShift Container Platform**	**Operator name**
OpenShift Cloud Manager SaaS Service	Included	Included	N/A
OVS and OVN SDN	Included	Included	N/A
MetalLB	Included	Included	MetalLB Operator
HAProxy Ingress Controller	Included	Included	N/A
Red Hat OpenStack Platform (RHOSP) Kuryr Integration	Included	Included	N/A
Ingress Cluster-wide Firewall	Included	Included	N/A

Feature	OpenShift Kubernetes Engine	OpenShift Container Platform	Operator name
Egress Pod and Namespace Granular Control	Included	Included	N/A
Ingress Non-Standard Ports	Included	Included	N/A
Multus and Available Multus Plugins	Included	Included	N/A
Network Policies	Included	Included	N/A
IPv6 Single and Dual Stack	Included	Included	N/A
CNI Plugin ISV Compatibility	Included	Included	N/A
CSI Plugin ISV Compatibility	Included	Included	N/A
RHT and IBM middleware à la carte purchases (not included in OpenShift Container Platform or OpenShift Kubernetes Engine)	Included	Included	N/A
ISV or Partner Operator and Container Compatibility (not included in OpenShift Container Platform or OpenShift Kubernetes Engine)	Included	Included	N/A
Embedded OperatorHub	Included	Included	N/A
Embedded Marketplace	Included	Included	N/A
Quay Compatibility (not included)	Included	Included	N/A

Feature	OpenShift Kubernetes Engine	OpenShift Container Platform	Operator name
RHEL Software Collections and RHT SSO Common Service (included)	Included	Included	N/A
Embedded Registry	Included	Included	N/A
Helm	Included	Included	N/A
User Workload Monitoring	Not Included	Included	N/A
Metering and Cost Management SaaS Service	Not Included	Included	N/A
Platform Logging	Not Included	Included	Red Hat OpenShift Logging Operator
OpenShift Elasticsearch Operator provided by Red Hat	Not Included	Cannot be run standalone	N/A
Developer Web Console	Not Included	Included	N/A
Developer Application Catalog	Not Included	Included	N/A
Source to Image and Builder Automation (Tekton)	Not Included	Included	N/A
OpenShift Service Mesh	Not Included	Included	OpenShift Service Mesh Operator
Service Binding Operator	Not Included	Included	Service Binding Operator
Feature	**OpenShift Kubernetes Engine**	**OpenShift Container Platform**	**Operator name**
Red Hat OpenShift Serverless	Not Included	Included	OpenShift Serverless Operator

Feature	OpenShift Kubernetes Engine	OpenShift Container Platform	Operator name
Web Terminal provided by Red Hat	Not Included	Included	Web Terminal Operator
Jenkins Operator provided by Red Hat	Not Included	Included	Jenkins Operator
Red Hat OpenShift Pipelines Operator	Not Included	Included	OpenShift Pipelines Operator
Embedded Component of IBM Cloud Pak and RHT MW Bundles	Not Included	Included	N/A
Red Hat OpenShift GitOps	Not Included	Included	OpenShift GitOps
Red Hat CodeReady Workspaces	Not Included	Included	CodeReady Workspaces
Red Hat CodeReady Containers	Not Included	Included	N/A
Quay Bridge Operator provided by Red Hat	Not Included	Included	Quay Bridge Operator
Quay Container Security provided by Red Hat	Not Included	Included	Quay Operator
Red Hat OpenShift distributed tracing platform	Not Included	Included	Red Hat OpenShift distributed tracing platform Operator
Red Hat OpenShift Kiali	Not Included	Included	Kiali Operator
Metering provided by Red Hat (deprecated)	Not Included	Included	N/A
Migration Toolkit for Containers Operator	Not Included	Included	Migration Toolkit for Containers Operator
Cost management for OpenShift	Not included	Included	N/A

Feature	OpenShift Kubernetes Engine	OpenShift Container Platform	Operator name
Red Hat JBoss Web Server	Not included	Included	JWS Operator
Red Hat Build of Quarkus	Not included	Included	N/A
Kourier Ingress Controller	Not included	Included	N/A
RHT Middleware Bundles Sub Compatibility (not included in OpenShift Container Platform)	Not included	Included	N/A
IBM Cloud Pak Sub Compatibility (not included in OpenShift Container Platform)	Not included	Included	N/A
OpenShift Do (odo)	Not included	Included	N/A
Source to Image and Tekton Builders	Not included	Included	N/A
OpenShift Serverless FaaS	Not included	Included	N/A
IDE Integrations	Not included	Included	N/A
Windows Machine Config Operator	Community Windows Machine Config Operator included - no subscription required	Red Hat Windows Machine Config Operator included - Requires separate subscription	Windows Machine Config Operator
Red Hat Quay	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Quay Operator
Red Hat Advanced Cluster Management	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Advanced Cluster Management for Kubernetes
Red Hat Advanced Cluster Security	Not Included - Requires separate subscription	Not Included - Requires separate subscription	N/A

Feature	OpenShift Kubernetes Engine	OpenShift Container Platform	Operator name
OpenShift Container Storage	Not Included - Requires separate subscription	Not Included - Requires separate subscription	OpenShift Container Storage
Feature	**OpenShift Kubernetes Engine**	**OpenShift Container Platform**	**Operator name**
Ansible Automation Platform Resource Operator	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Ansible Automation Platform Resource Operator
Business Automation provided by Red Hat	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Business Automation Operator
Data Grid provided by Red Hat	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Data Grid Operator
Red Hat Integration provided by Red Hat	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Red Hat Integration Operator
Red Hat Integration - 3Scale provided by Red Hat	Not Included - Requires separate subscription	Not Included - Requires separate subscription	3scale
Red Hat Integration - 3Scale APICast gateway provided by Red Hat	Not Included - Requires separate subscription	Not Included - Requires separate subscription	3scale APIcast
Red Hat Integration - AMQ Broker	Not Included - Requires separate subscription	Not Included - Requires separate subscription	AMQ Broker
Red Hat Integration - AMQ Broker LTS	Not Included - Requires separate subscription	Not Included - Requires separate subscription	
Red Hat Integration - AMQ Interconnect	Not Included - Requires separate subscription	Not Included - Requires separate subscription	AMQ Interconnect
Red Hat Integration - AMQ Online	Not Included - Requires separate subscription	Not Included - Requires separate subscription	
Red Hat Integration - AMQ Streams	Not Included - Requires separate subscription	Not Included - Requires separate subscription	AMQ Streams
Red Hat Integration - Camel K	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Camel K

Feature	OpenShift Kubernetes Engine	OpenShift Container Platform	Operator name
Red Hat Integration - Fuse Console	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Fuse Console
Red Hat Integration - Fuse Online	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Fuse Online
Red Hat Integration - Service Registry Operator	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Service Registry
API Designer provided by Red Hat	Not Included - Requires separate subscription	Not Included - Requires separate subscription	API Designer
JBoss EAP provided by Red Hat	Not Included - Requires separate subscription	Not Included - Requires separate subscription	JBoss EAP
JBoss Web Server provided by Red Hat	Not Included - Requires separate subscription	Not Included - Requires separate subscription	JBoss Web Server
Smart Gateway Operator	Not Included - Requires separate subscription	Not Included - Requires separate subscription	Smart Gateway Operator
Kubernetes NMState Operator	Included	Included	N/A

3.2. SUBSCRIPTION LIMITATIONS

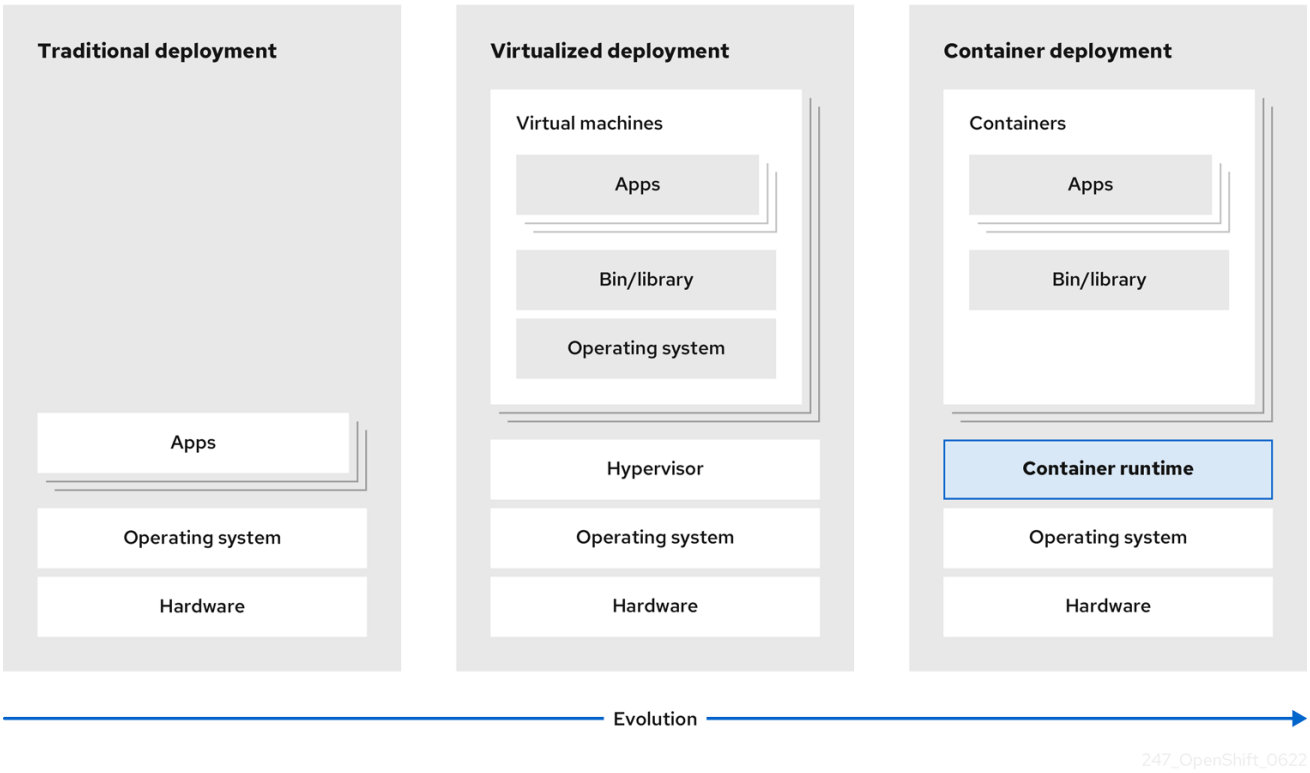
OpenShift Kubernetes Engine is a subscription offering that provides OpenShift Container Platform with a limited set of supported features at a lower list price. OpenShift Kubernetes Engine and OpenShift Container Platform are the same product and, therefore, all software and features are delivered in both. There is only one download, OpenShift Container Platform. OpenShift Kubernetes Engine uses the OpenShift Container Platform documentation and support services and bug errata for this reason.

CHAPTER 4. KUBERNETES OVERVIEW

Kubernetes is an open source container orchestration tool developed by Google. You can run and manage container-based workloads by using Kubernetes. The most common Kubernetes use case is to deploy an array of interconnected microservices, building an application in a cloud native way. You can create Kubernetes clusters that can span hosts across on-premise, public, private, or hybrid clouds.

Traditionally, applications were deployed on top of a single operating system. With virtualization, you can split the physical host into several virtual hosts. Working on virtual instances on shared resources is not optimal for efficiency and scalability. Because a virtual machine (VM) consumes as many resources as a physical machine, providing resources to a VM such as CPU, RAM, and storage can be expensive. Also, you might see your application degrading in performance due to virtual instance usage on shared resources.

Figure 4.1. Evolution of container technologies for classical deployments



To solve this problem, you can use containerization technologies that segregate applications in a containerized environment. Similar to a VM, a container has its own filesystem, vCPU, memory, process space, dependencies, and more. Containers are decoupled from the underlying infrastructure, and are portable across clouds and OS distributions. Containers are inherently much lighter than a fully-featured OS, and are lightweight isolated processes that run on the operating system kernel. VMs are slower to boot, and are an abstraction of physical hardware. VMs run on a single machine with the help of a hypervisor.

You can perform the following actions by using Kubernetes:

- Sharing resources
- Orchestrating containers across multiple hosts
- Installing new hardware configurations
- Running health checks and self-healing applications

- Scaling containerized applications

4.1. KUBERNETES COMPONENTS

Table 4.1. Kubernetes components

Component	Purpose
kube-proxy	Runs on every node in the cluster and maintains the network traffic between the Kubernetes resources.
kube-controller-manager	Governs the state of the cluster.
kube-scheduler	Allocates pods to nodes.
etcd	Stores cluster data.
kube-apiserver	Validates and configures data for the API objects.
kubelet	Runs on nodes and reads the container manifests. Ensures that the defined containers have started and are running.
kubectl	Allows you to define how you want to run workloads. Use the **kubectl** command to interact with the **kube-apiserver**.
Node	Node is a physical machine or a VM in a Kubernetes cluster. The control plane manages every node and schedules pods across the nodes in the Kubernetes cluster.
container runtime	container runtime runs containers on a host operating system. You must install a container runtime on each node so that pods can run on the node.
Persistent storage	Stores the data even after the device is shut down. Kubernetes uses persistent volumes to store the application data.
container-registry	Stores and accesses the container images.
Pod	The pod is the smallest logical unit in Kubernetes. A pod contains one or more containers to run in a worker node.

4.2. KUBERNETES RESOURCES

A custom resource is an extension of the Kubernetes API. You can customize Kubernetes clusters by using custom resources. Operators are software extensions which manage applications and their components with the help of custom resources. Kubernetes uses a declarative model when you want a fixed desired result while dealing with cluster resources. By using Operators, Kubernetes defines its states in a declarative way. You can modify the Kubernetes cluster resources by using imperative

commands. An Operator acts as a control loop which continuously compares the desired state of resources with the actual state of resources and puts actions in place to bring reality in line with the desired state.

Figure 4.2. Kubernetes cluster overview

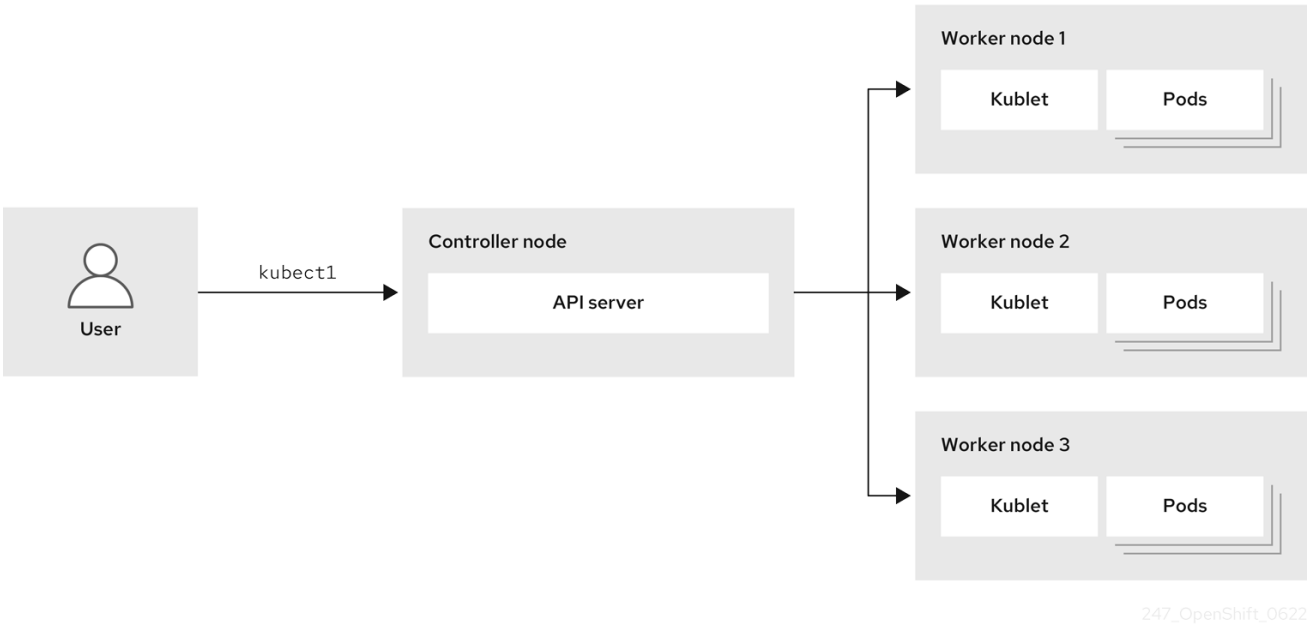
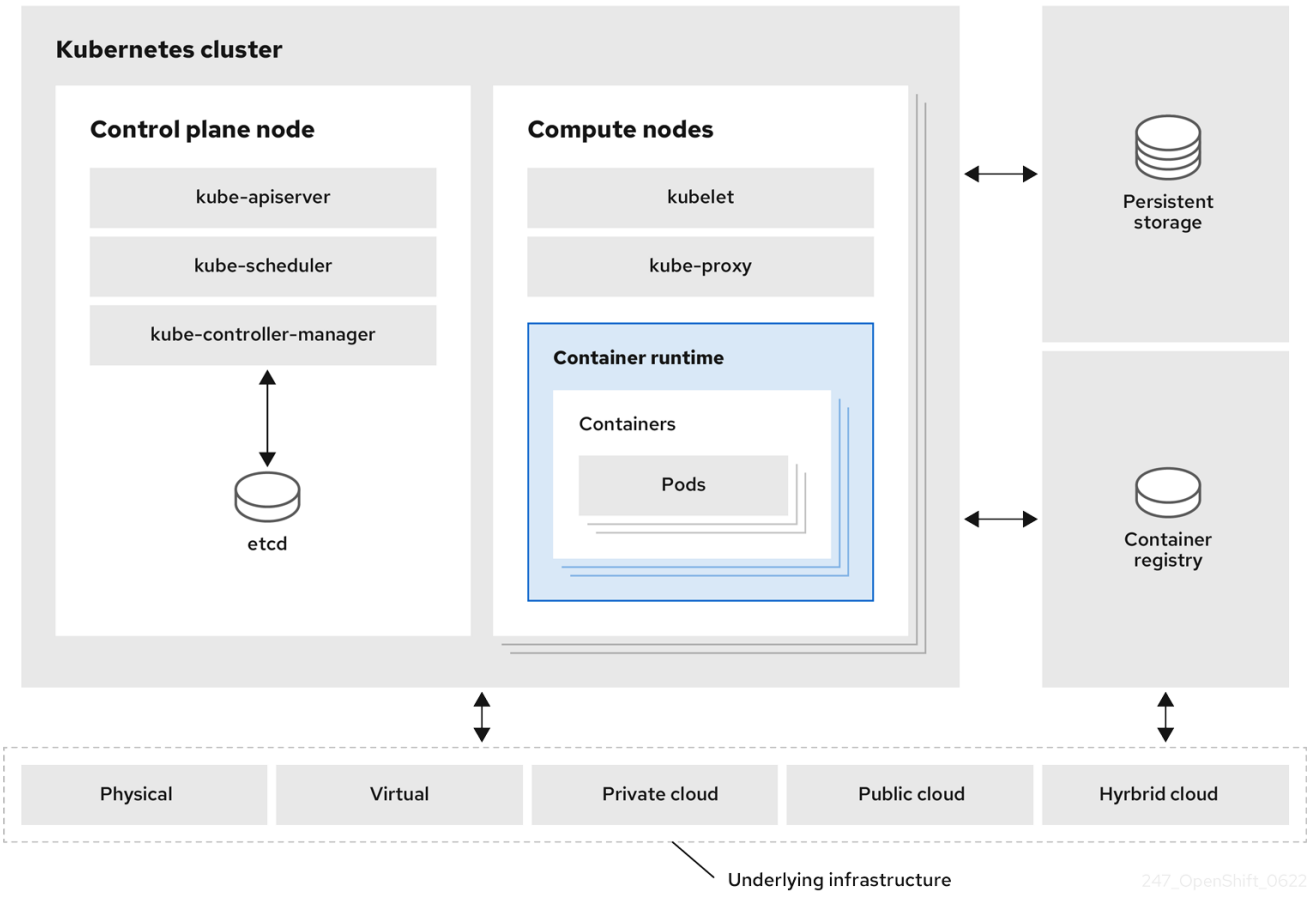


Table 4.2. Kubernetes Resources

Resource	Purpose
Service	Kubernetes uses services to expose a running application on a set of pods.
ReplicaSets	Kubernetes uses the **ReplicaSets** to maintain the constant pod number.
Deployment	A resource object that maintains the life cycle of an application.

Kubernetes is a core component of an OpenShift Container Platform. You can use OpenShift Container Platform for developing and running containerized applications. With its foundation in Kubernetes, the OpenShift Container Platform incorporates the same technology that serves as the engine for massive telecommunications, streaming video, gaming, banking, and other applications. You can extend your containerized applications beyond a single cloud to on-premise and multi-cloud environments by using the OpenShift Container Platform.

Figure 4.3. Architecture of Kubernetes



A cluster is a single computational unit consisting of multiple nodes in a cloud environment. A Kubernetes cluster includes a control plane and worker nodes. You can run Kubernetes containers across various machines and environments. The control plane node controls and maintains the state of a cluster. You can run the Kubernetes application by using worker nodes. You can use the Kubernetes namespace to differentiate cluster resources in a cluster. Namespace scoping is applicable for resource objects, such as deployment, service, and pods. You cannot use namespace for cluster-wide resource objects such as storage class, nodes, and persistent volumes.

4.3. KUBERNETES CONCEPTUAL GUIDELINES

Before getting started with the OpenShift Container Platform, consider these conceptual guidelines of Kubernetes:

- Start with one or more worker nodes to run the container workloads.
- Manage the deployment of those workloads from one or more control plane nodes.
- Wrap containers in a deployment unit called a pod. By using pods provides extra metadata with the container and offers the ability to group several containers in a single deployment entity.
- Create special kinds of assets. For example, services are represented by a set of pods and a policy that defines how they are accessed. This policy allows containers to connect to the services that they need even if they do not have the specific IP addresses for the services. Replication controllers are another special asset that indicates how many pod replicas are required to run at a time. You can use this capability to automatically scale your application to adapt to its current demand.

The API to OpenShift Container Platform cluster is 100% Kubernetes. Nothing changes between a container running on any other Kubernetes and running on OpenShift Container Platform. No changes to the application. OpenShift Container Platform brings added-value features to provide enterprise-ready enhancements to Kubernetes. OpenShift Container Platform CLI tool (**oc**) is compatible with **kubectl**. While the Kubernetes API is 100% accessible within OpenShift Container Platform, the **kubectl** command-line lacks many features that could make it more user-friendly. OpenShift Container Platform offers a set of features and command-line tool like **oc**. Although Kubernetes excels at managing your applications, it does not specify or manage platform-level requirements or deployment processes. Powerful and flexible platform management tools and processes are important benefits that OpenShift Container Platform offers. You must add authentication, networking, security, monitoring, and logs management to your containerization platform.