

Introduction to Kubernetes and OpenShift Objectives

- Describe container orchestration and the features of Red Hat OpenShift.

Kubernetes Overview

Kubernetes is an orchestration service that simplifies the deployment, management, and scaling of containerized applications. It manages complex pools of resources, such as CPU, RAM, storage, and networking. Kubernetes provides high uptime and fault tolerance for containerized application deployments, removing the concern that developers might have regarding how their applications use resources.

The smallest manageable unit in Kubernetes is a pod, which represents a single application and consists of one or more containers, including storage resources and an IP address.

Kubernetes Features

Kubernetes clusters provide a modern container platform that addresses the concerns and challenges of running applications at scale. No matter the deployment size, Kubernetes implementations deliver a robust infrastructure and ease of management:

Service discovery and load balancing

Kubernetes enables inter-service communication by assigning a single DNS entry to each set of containers. To enable the cluster to change the container's location and IP address, the requesting service must know the target's DNS name. As a result, Kubernetes can load-balance the request across the pool of containers that provide the service. For example, Kubernetes can evenly split incoming requests to an NGINX web server by taking into account the availability of the NGINX pods.

Horizontal scaling

Applications can scale up and down manually or automatically with a configuration set, with either the Kubernetes command-line interface or the web UI.

Self-healing

Kubernetes can use user-defined health checks to monitor pods to restart and reschedule them in the event of failure.

Automated rollout

Kubernetes can gradually roll out updates to your application's containers while checking their status. If something goes wrong during the rollout, then Kubernetes can roll back to the previous version of the deployment.

Secrets and configuration management

You can manage the configuration settings and secrets of your applications without rebuilding containers. Application secrets can be usernames, passwords, and service endpoints, or any configuration setting that must be kept private.

Operators

Operators are packaged Kubernetes applications that also bring the knowledge of the application's lifecycle into the Kubernetes cluster. Applications that are packaged as operators use the Kubernetes API to update the cluster's state, and react to changes in the application state.

Red Hat OpenShift Container Platform Overview

Red Hat OpenShift Container Platform (RHOC) is a set of modular components and services that are built on top of the Kubernetes container infrastructure. RHOC adds capabilities to a production platform, such as remote management, multitenancy, increased security, monitoring and auditing, application lifecycle management, and self-service interfaces for developers.

Red Hat OpenShift Container Platform Features

RHOC adds the following features to a Kubernetes cluster:

Developer workflow

Integrates a built-in container registry, *Continuous Integration/Continuous Delivery (CI/CD)* pipelines, and *Source-to-Image (S2I)*, a tool to build artifacts from source repositories to container images.

Routes

Exposes services to the outside world easily.

Metrics and logging

Includes a built-in and self-analyzing metrics service and aggregated logging.

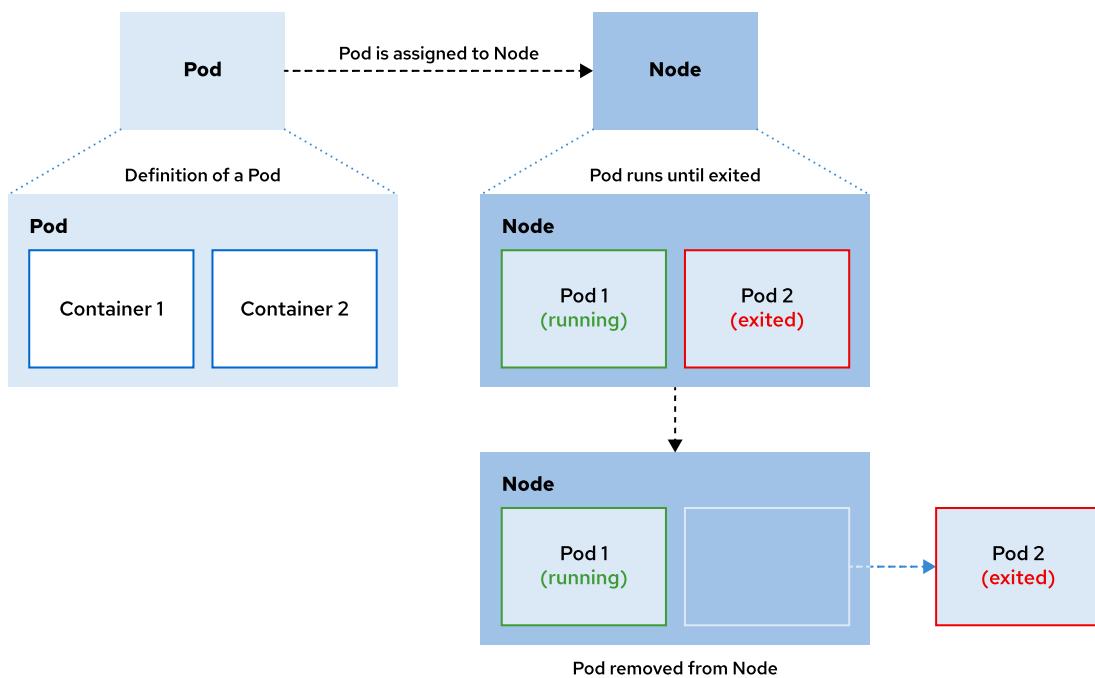
Unified UI

Provides unified tools and an integrated UI to manage the different capabilities.

Lifecycle of Applications in Red Hat OpenShift Container Platform

The following figure illustrates the basic lifecycle of an application that is deployed in a RHOC cluster:

1. Starts with the definition of a pod and the containers that it is composed of, which contain the application.
2. Pods are assigned to a healthy node.
3. Pods run until their containers exit.
4. Pods and their containers are removed from the node.



Depending on policy and exit code, RHOCOP might remove pods after exiting, or might retain them to enable access to the pod container logs.

REFERENCES

For more information about Kubernetes, see *Kubernetes Documentation* at <https://kubernetes.io/docs/home/>

For more information about Red Hat OpenShift Container Platform, refer to the documentation at https://access.redhat.com/documentation/en-us/openshift_container_platform/4.18