

CONTAINERS, KUBERNETES AND OPENSIFT

Objectives

- OpenShift Foundations
- Containers
- Container Orchestration
- Kubernetes
- What OpenShift adds to Kubernetes

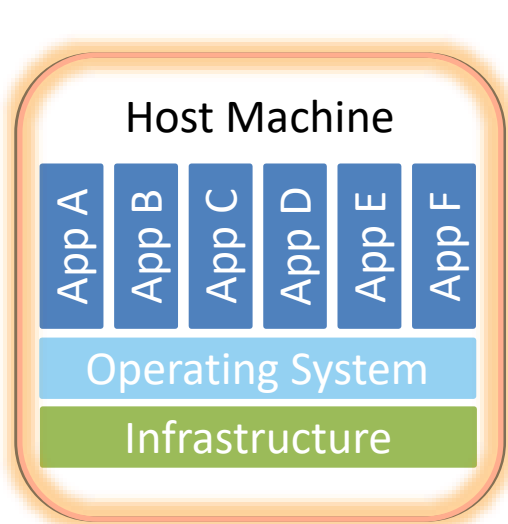
OpenShift Foundations

- OpenShift is built on two open-source technologies, which we briefly introduce here:
 - Containers
 - Kubernetes container orchestration

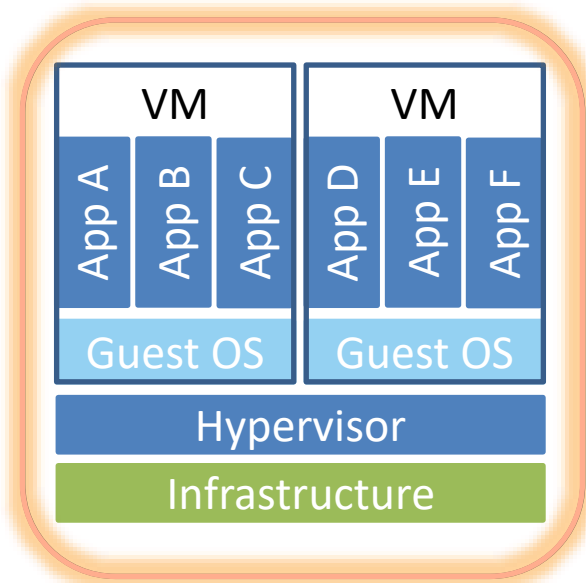
Extending Hardware Usage

- Common requirements:
 - Users - Run app transparently, with all dependencies etc in place
 - SysAdmin - Run multiple workloads with appropriate resource allocation and isolation.
- Two approaches:
 - Virtual Machines: Multiple self-contained computers on one piece of hardware
 - Containers: Self-contained file systems running on a container service on one piece of hardware.

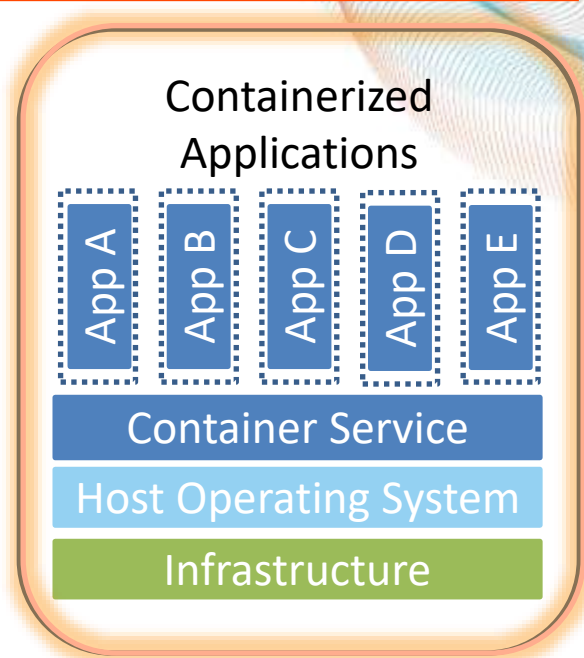
Containers



Single machine architecture



Virtual machine architecture



Container architecture

The Open Container Initiative (OCI)



- An open governance structure to create open industry standards around container image formats and container runtimes.
- Docker donated their runc runtime to OCI.
- Docker's original image format has become the OCI Image Specification
- There are a variety of open-source build tools supporting it
 - **BuildKit**: an optimized build engine
 - **Podman**: a command-line tool
 - **Buildah**: a command line alternative to writing Dockerfiles

The Need for Container Orchestration

- Platforms like Docker handle containers running on a single host well.
- Applications often require more resources than can be available on a single host
- So ... deploy multiple containers on multiple hosts!
- The extra complexity (network, security, service discovery) requires a **Container Orchestrator** to manage.



Kubernetes

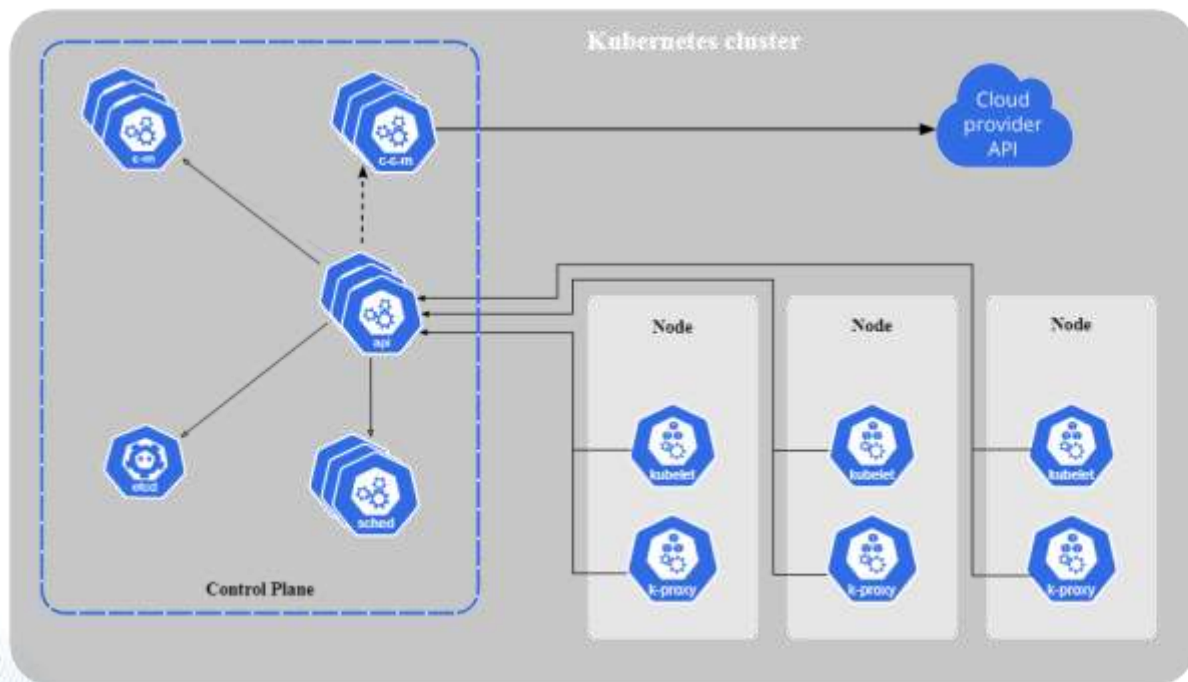
- According to their website:
“Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.”
- Comes from Greek **κυβερνήτης**: helmsman or ship pilot. Can also be referred to as k8s (pronounced Kate's)
- Started by Google and donated to CNCF in July 2015.



Kubernetes Features

- Automatic container scheduling to maximize utilization without affecting availability
- Self-healing and automatic traffic re-routing from failed nodes
- Manually or automatically scale your cluster
- Service discovery and load balancing within the cluster
- Automated rollouts and rollbacks
- Secrets management and configuration details management

Kubernetes Architecture



Kubernetes Architecture: Control Plane

- This runs on the master node
- Responsible for managing cluster state - needs to be kept running!
- It consists of the following
 - Distributed key-value store for the most critical system data
 - API server that coordinates all the administrative tasks
 - Scheduler that assigns workloads to worker nodes
 - Controller manager running controllers to regulate cluster state.

Kubernetes Cluster: Worker Nodes

- Provide running environment for client applications.
- Applications are run in containers, encapsulated in Pods, controlled by the control plane on the master node.
- They run the following components:
 - Container runtime
 - Node agent to interface with the API server
 - Proxy for communication between pods

kubectl command line manager

- Kubernetes provides a command line tool - kubectl - for communicating with a cluster's control plane.
- The following is the standard syntax for kubectl commands:

kubectl [command][TYPE][NAME][flags]

command: the operation you want to perform, e.g.
create, get, describe

TYPE: the resource type

NAME: the name of the resource (if required)

flags: optional flags

What Does Kubernetes Do?

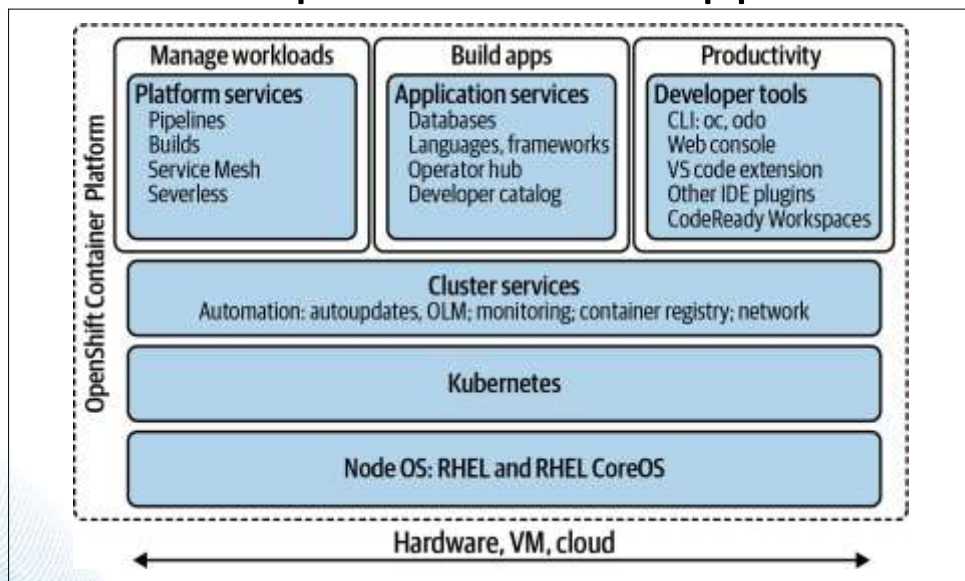
- Kubernetes is like an operating system for a distributed platform.
- It defines a set of common resources, and an API for utilizing them.
- Kubernetes is *declarative*, which means you declare what you would like your system to look like, then Kubernetes works to match your desired state.
 - This is called the *reconcile loop* of the cluster.

What Kubernetes Doesn't Do

- To really use a Kubernetes cluster you need to add some extra components on top of Plain Old Kubernetes.
- In a fully DevOps system these include
 - Tools to build, compile and run apps.
 - Tools to manage workloads on the cluster and deploy updates
 - Networking, monitoring and security.

OpenShift is Kubernetes++

- OpenShift builds on Kubernetes and containers to add features and the components that support them.



OPENSIFT 4



Best IT ops experience

CaaS ↔ PaaS | FaaS

Best developer experience



Cluster services

Monitoring,
showback,
registry, logging

Application services

Middleware, functions, ISV

Service mesh

Developer services

Dev tools,
automated builds,
CI/CD, IDE



Automated operations



 **Red Hat**
Enterprise Linux
CoreOS



Any infrastructure



Physical



Virtual



Private



Public



OpenShift Components: Containers

- OpenShift adds some key container management components to the Kubernetes standard:
 - **CRI-O** - container runtime (lightweight alternative to Docker)
 - **Podman** - container engine for developing, managing and running OCI containers on Linux
 - **Skopeo** - tool for moving container images between different types of storage (e.g. docker to internal registry)
 - **Buildah** - tool to build OCI container images.
 - **Quay** - container registry

The OpenShift Web Console

- A key addition OpenShift makes to Kubernetes is its **Web Console**
 - Kubernetes has a web console, but it currently has very limited functionality.
- This is a full featured UI to your cluster, networking and CI/CD capabilities.

OpenShift CLI tools

- The main CLI tool OpenShift offers is the `oc` tool. **We will focus on this**
- This is a souped up version of `kubectl`
 - It uses the `kubectl` syntax and can be used with all `kubectl` subcommands
 - It also adds a suite of commands only available on OpenShift.
- The `odo` command line tool is a tool for developing, testing, debugging and deploying microservices-based applications on a k8s cluster without having to understand the platform deeply.
- There are also CLI tools for Tekton pipelines (`tkn`), serverless (`kn`), and Operator development and management (Operator SDK and `opm`).

The OpenShift App Store

- The Web Console aggregates software catalogues
 - Application templates to simplify standard deployments
 - Kubernetes Operators for full lifecycle management of backend services
- Find and deploy databases, message queues, etc

CI/CD Pipelines

- OpenShift sets itself to add full CI/CD capabilities to Kubernetes.
- **OpenShift pipelines** based on Tekton sets you up to automatically
 - Build
 - Test
 - Package
 - Release
- OpenShift also hooks in easily to Jenkins build pipelines.

Networking and Service Mesh

- In k8s setting up connections within your cluster and to the outside world can be pretty cumbersome.
- OpenShift addresses this with
 - OpenShift Routes: A layer 7 reverse proxy for external HTTP connections to internal, load-balanced cluster Services.
 - [Istio](#): A bolt on service mesh that measures and controls how services connect with each other and the outside world.

Metrics, Monitoring and Alerts

- OpenShift also bolts on a number of monitoring resources based around Prometheus.
 - [Prometheus](#) is an open-source systems monitoring and alerting toolkit originally from SoundCloud.
- This gives accurate resource monitoring for all levels of granularity right down to a running container.

Summary

- OpenShift Foundations
- Containers
- Container Orchestration
- Kubernetes
- What OpenShift adds to Kubernetes

Questions and Comments?

