BDBI Platform Workshop

Container Orchestration with Kubernetes (K8s)

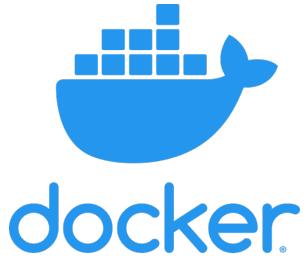
3/9/2023

What is Kubernetes (K8s)?

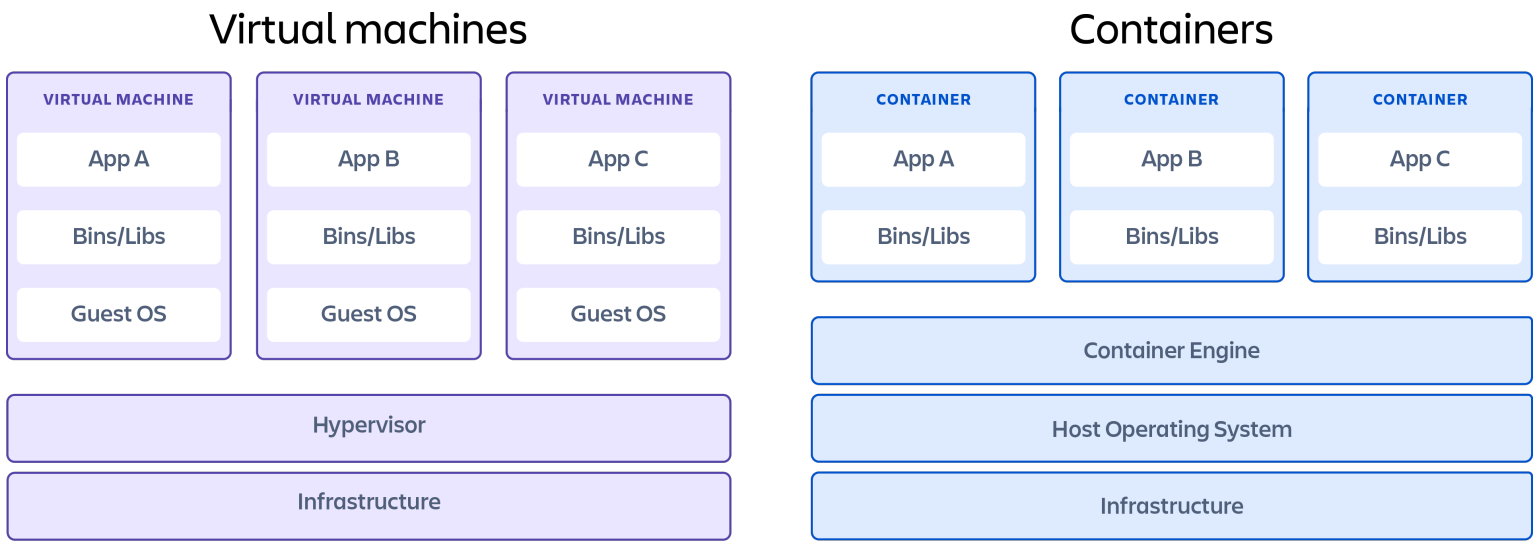
- container orchestration tool
- developed by Google
- manage containerized applications in different deployment environments (physical, cloud, etc.)

Google

What is a Container?



- lightweight way to run app that only virtualizes application layer (rather than application layer + kernel)
  - SW layers vs. HW layers
- contain dependencies required to execute contained app

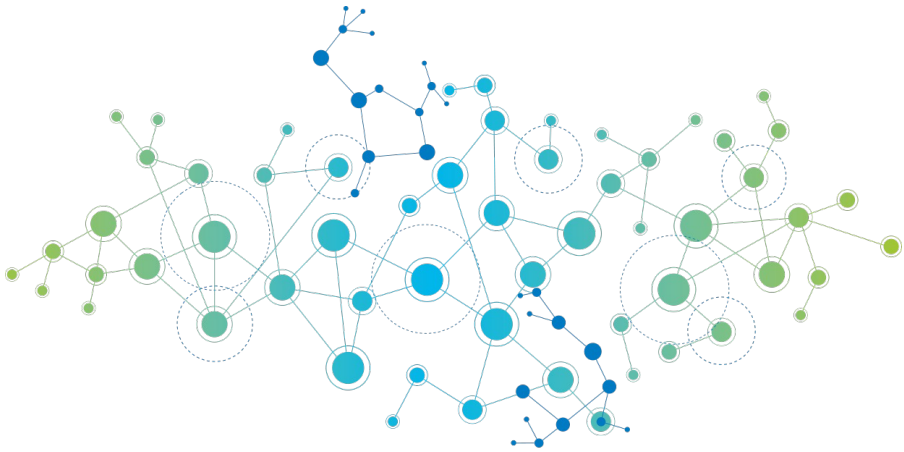


Why K8s?

- tons of microservices
- hard to manage thousands of containers
- create distributed systems
  - scalability + less downtime
- backup + restore

K8s Architecture

- node
  - virtual or physical machine
- cluster
  - master node
    - manages cluster
  - worker nodes
    - runs a kubelet process (K8s interface)
    - runs container(s)
    - where “work” is done
- virtual network
  - spans all nodes in cluster



The Master Node (Control Plane)

- API Server = entry point to cluster
- Controller Manager = keeps track of what's happening in cluster
- Scheduler = ensures pods replacement based on available resources & workloads
- etcd = backing store
  - (key, value)
  - snapshots for backup/restore

- Typically 2+ master nodes in prod clusters

K8s Components

Pods

- smallest unit
- abstraction over container
- typically one app/pod
- die easily
- unique IP address → new IP on recreation

K8s Components

Service

- permanent IP address attached to each pod
  - not connected to pod lifecycle
- external & internal
- connect to cloned pods (replicas)
  - load balancer

Ingress

- route traffic to services

K8s Components

ConfigMap

- external configuration of your application
- references
- non-sensitive data

Secret

- base64 encoded
  - 3rd party encryption
- credentials

K8s Components

Deployment

- blueprint for stateless pods
- abstraction of pods, we typically deal with Deployments
- replicas

StatefulSet

- for stateful apps (MySQL, mongoDB, Elastic)
- synchronize read/writes to DB, but difficult

K8s Config Files (YAML or JSON)

3 parts to every config file

1. metadata
2. specification
3. status
    a. automatically generated by K8s
    b. desired vs. actual state
    c. status data from etcd

Using K8s

minikube

https://minikube.sigs.k8s.io/docs/start/

- virtual node (one cluster)
- for local dev
- Docker pre-installed

kubectl

minikube dependency

- CLI
- interact with cluster

Demo

Scan QR Code for GitHub repo

Demo Overview

A web app that uses mongoDB as a backend, all deployed on K8s cluster

- Deployment, Service, ConfigMap, Secret

Links:

https://kubernetes.io/docs/home/

https://hub.docker.com/repository/docker/nanajanashia/k8s-demo-app