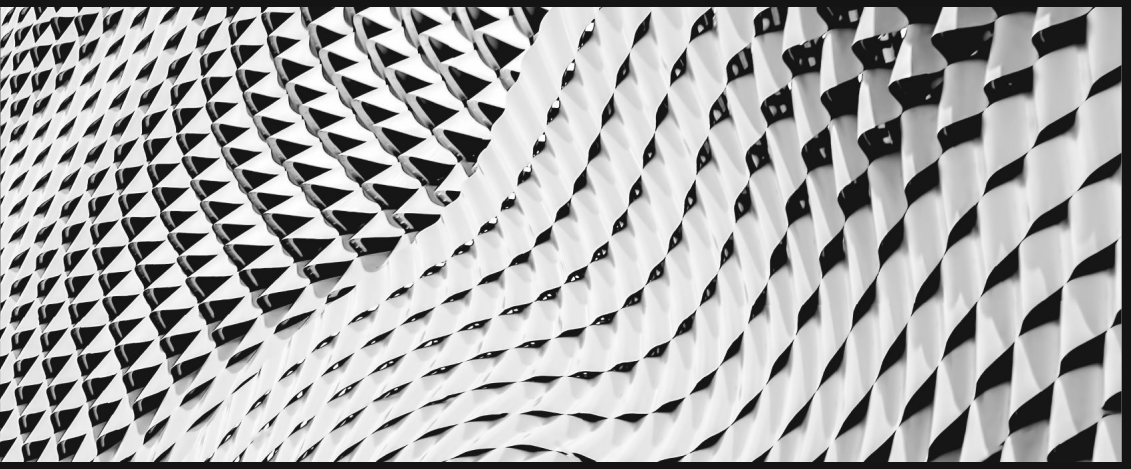
A Brief History Of Containers

Kubernetes In A Nutshell

Deven Phillips
Red Hat AppDev CoE

Red Hat

It all started with CGroups



cgroups (abbreviated from **control groups**) is a Linux kernel feature that limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes.

CGroups Allow For Granular Control

Resource limiting

groups can be set to not exceed a configured memory limit, which also includes the file system cache

Prioritization

some groups may get a larger share of CPU utilization[10] or disk I/O throughput

Accounting

measures a group's resource usage, which may be used, for example, for billing purposes

Control

freezing groups of processes, their checkpointing and restarting

Source:
Ihttps://en.wikipedia.org/wiki/Cgroups

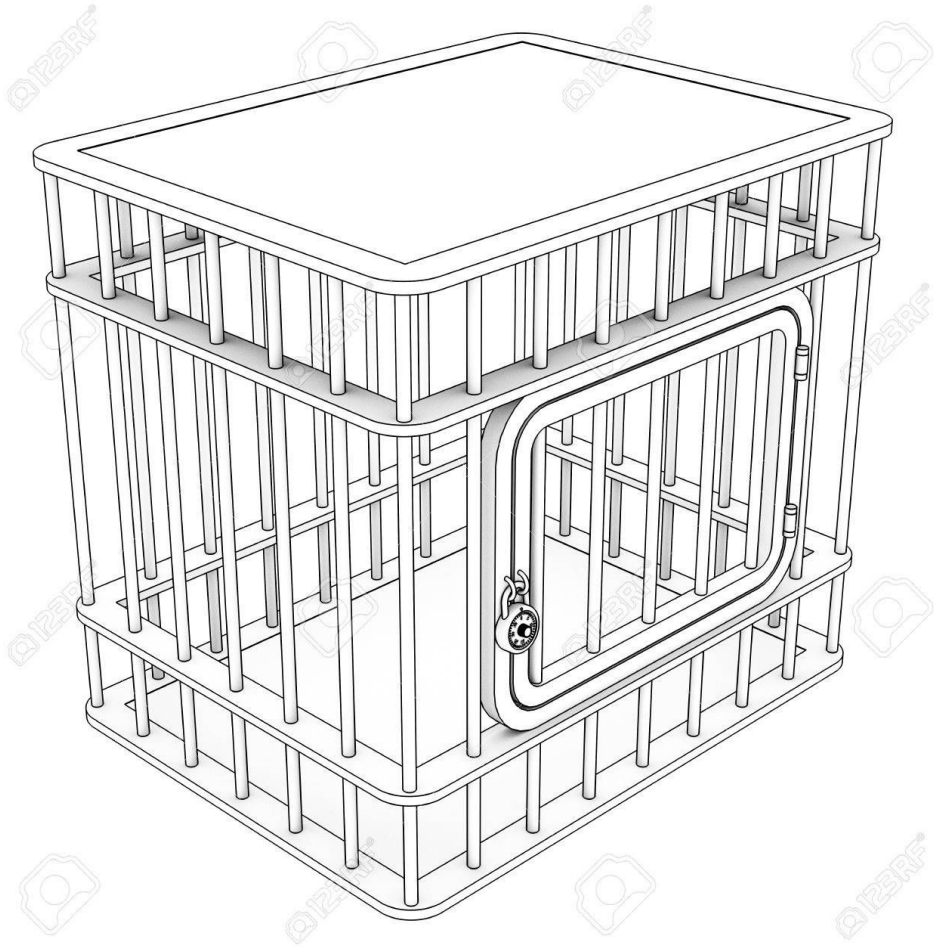
LXC - Linux Containers

System Level Virtualization

Allows for a single kernel to control multiple operating system “instances”

Source:
https://en.wikipedia.org/wiki/LXC

Docker

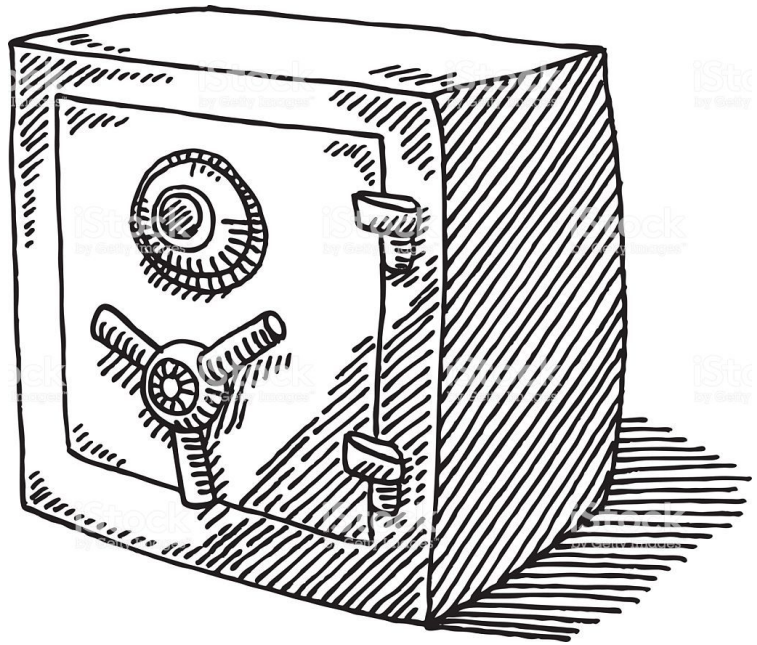


CGroups with A Packaged Filesystem

No shared filesystem unless explicitly configured

Source:
https://docs.docker.com/

OCI Containers



Like Docker, But Less Vulnerable

Runs without ‘root’ privileges.

Podman

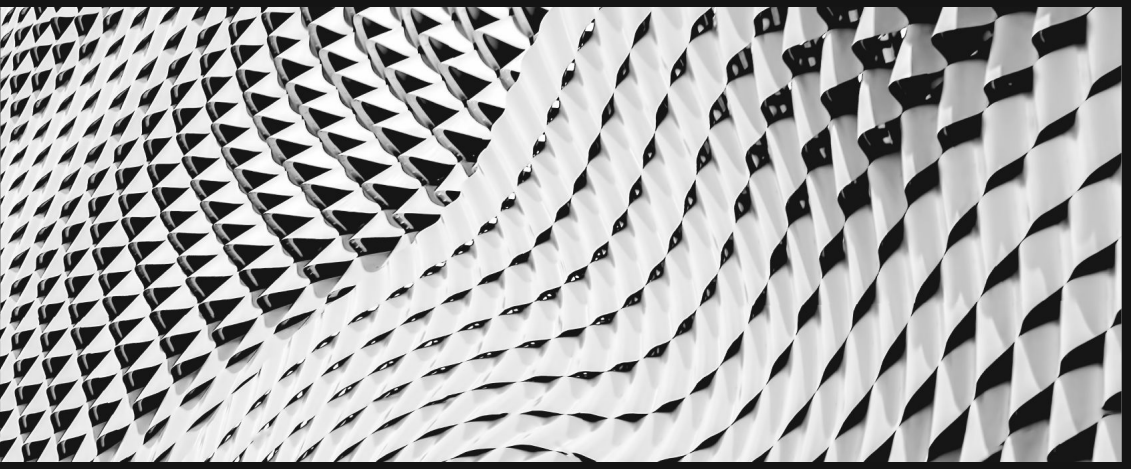
Runs containers (including Docker containers)

Buildah

Builds containers, like `docker build . . .`

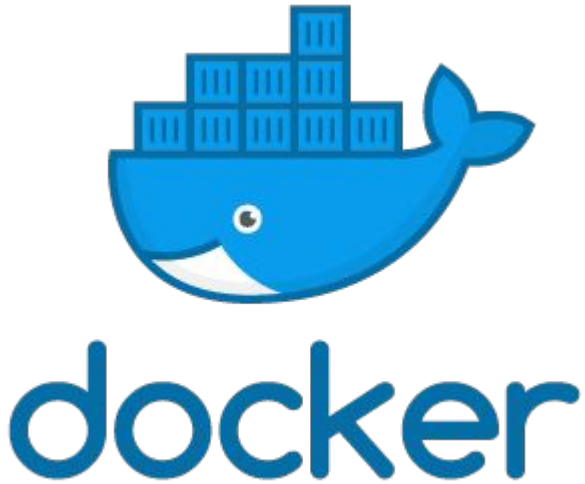
Source:
https://www.opencontainers.org

Containers Are Great . . . UNTIL



Red Hat

Docker “run”



One-Offs

Each container needs to be started using the CLI or the Docker API

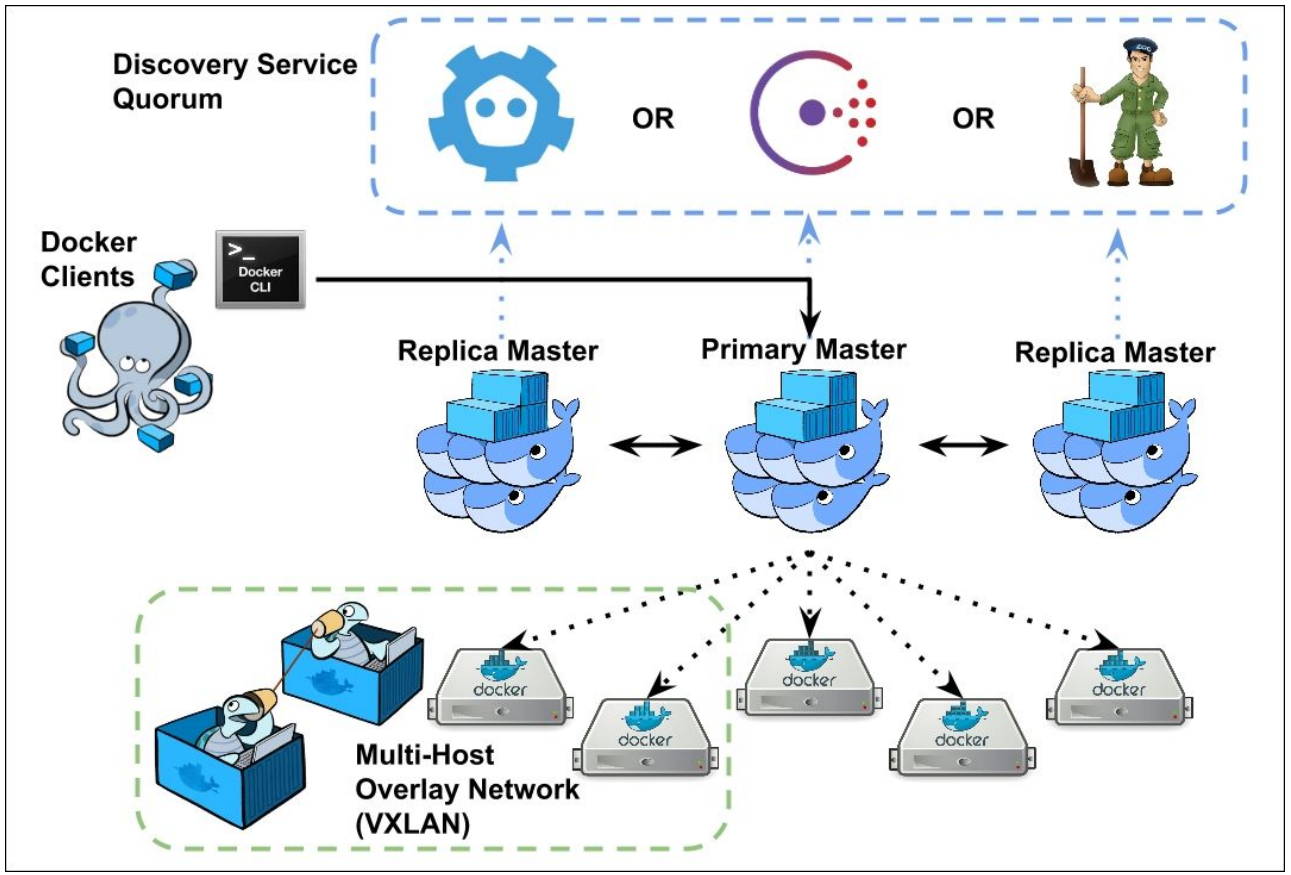
Storage

Storage is passed through to the underlying OS

Networking

Has to be configured manually for each container

Docker Swarm



Discovery Server

Raft/Zookeeper/etcd

Networking

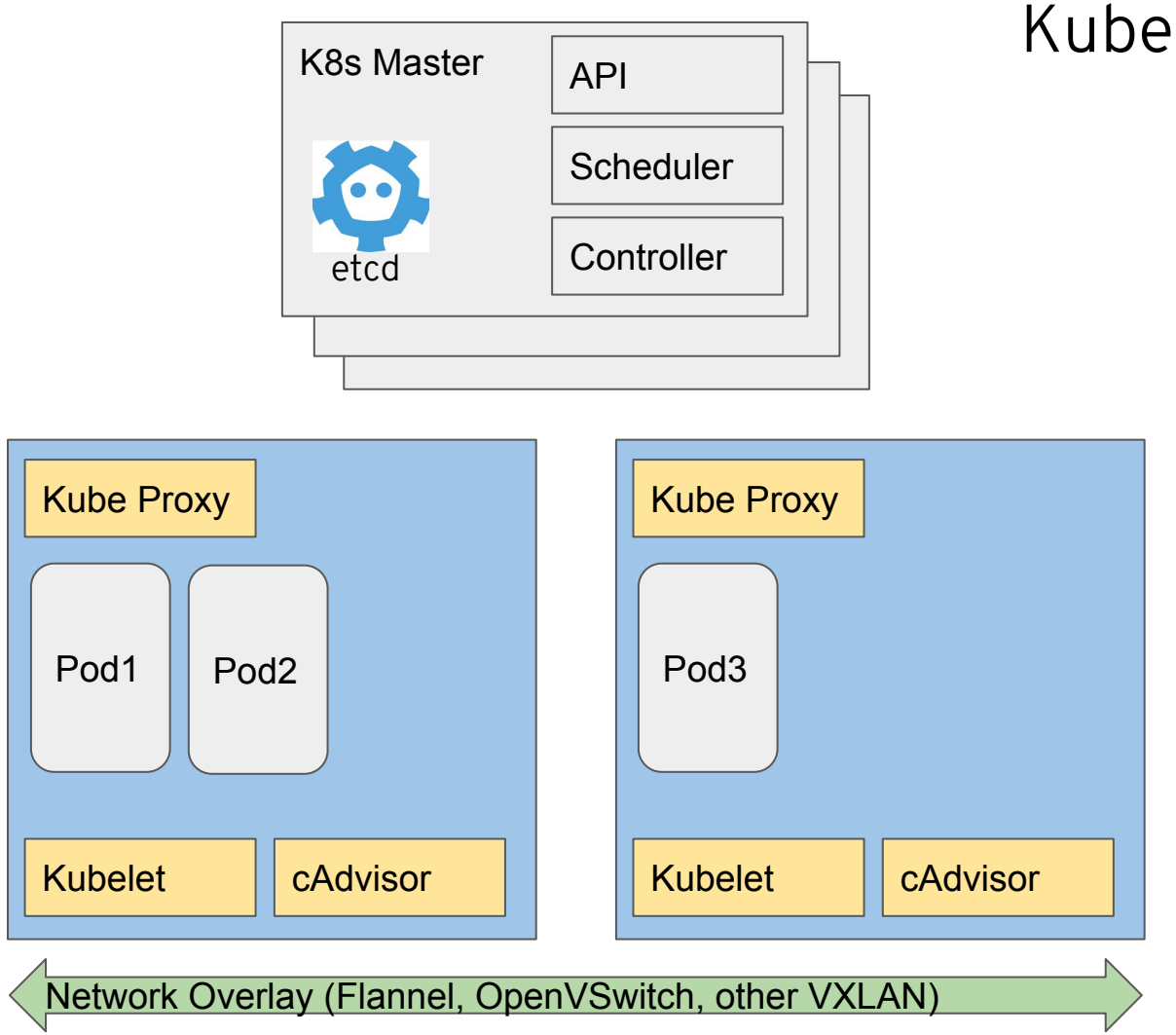
Overlay Network Using VXLANs

Storage

Not really addressed by Swarm, but you could use other distributed filesystems

Source:
https://docs.docker.com/engine/swarm/

Kubernetes



Kube Master

etcd, API, Controller; written in GoLang

Networking

Overlay Network Using VXLANs

Storage

Several cloud-native storage options now available, or use external Gluster/NFS/Ceph/S3/etc…

Scaling

Pods can be “replicated” to scale horizontally

Management

Via CLI, Web, or API

Centralized Logging

Elasticsearch/Fluentd/Kibana

Source:
https://kubernetes.io

Kubernetes Operators

Extend Kubernetes API

Custom Resource Definitions

Integrate With Kubernetes Controller

Can watch for changes and respond automatically

Add New Functionality

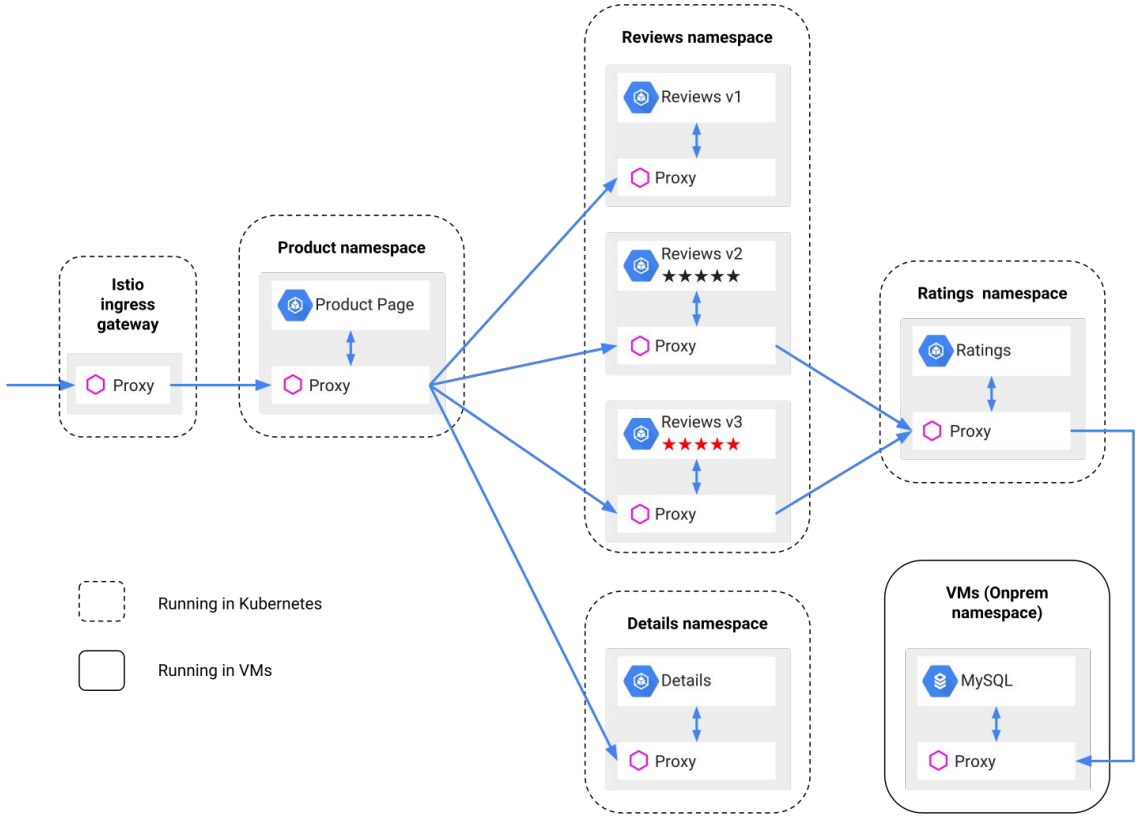
Operators for storage, databases, security, messaging, etc…
Think of is like AWS servers on your own cloud

Codify Domain Expertise

Make services self-managing (as much as possible)

Source:
https://github.com/operator-framework/
https://operatorhub.io/

Istio Operator

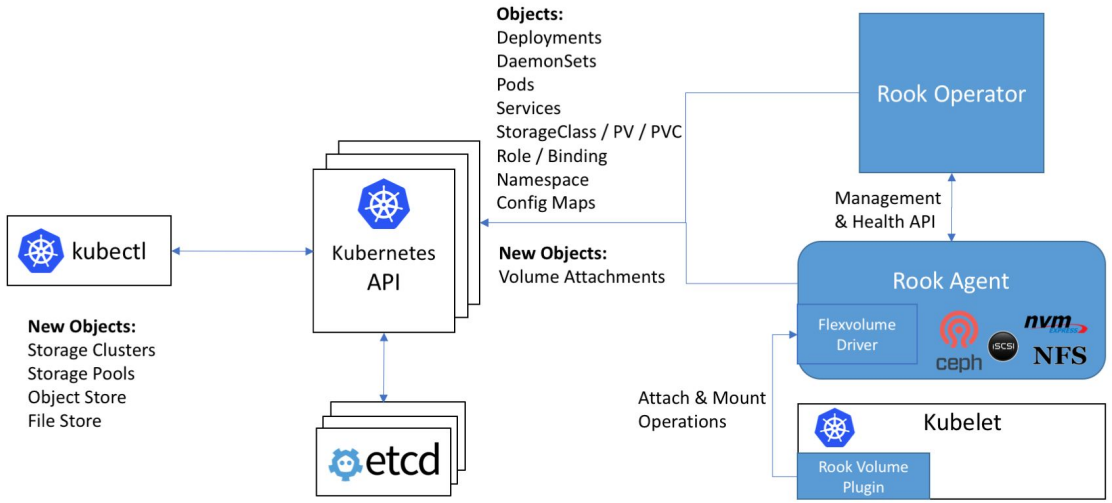


- End-To-End Encryption
- Control Plane
- Tracing
- Traffic Management
- Metrics

Source:
https://istio.io/

Red Hat

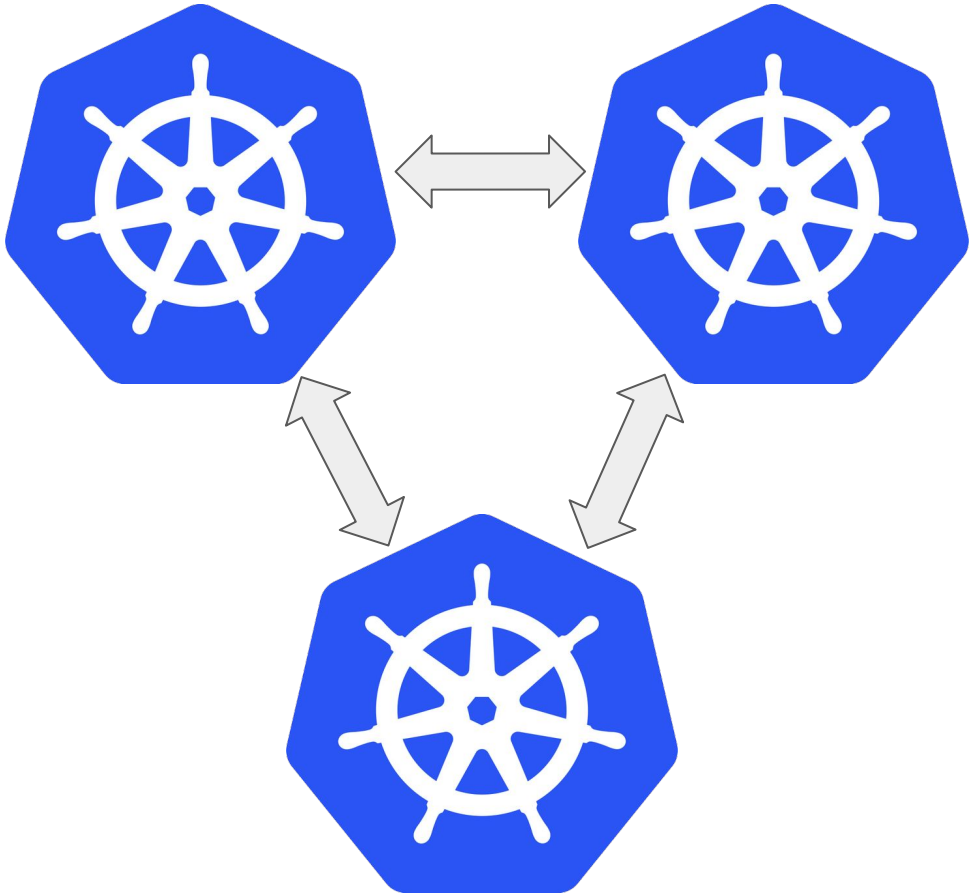
Rook Operator



- Scalable Storage
- Storage Management
- Portable Storage
- Replication
- Volumes
- Block Devices
- Object Stores

Source:
https://rook.io/

Kubernetes Federation



Sync Resources Across Kubernetes Clusters

etcd Sync

Migrate Workloads/Storage/Configuration

Allow workloads to migrate from one DC to another

Source:
https://kubernetes.io/docs/concepts/cluster-administration/federation/

Further Resources

Rancher Kubernetes

OpenShift Kubernetes

- Operator Hub - A Marketplace for K8s Operators
- OpenVSwitch - An Open Source virtual switch
- Operator Framework - How to create Operators
- Mini-Kubernetes For Testing/Experimentation
  - K3s
  - MiniKube
  - MicroK8s

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