Containerization Technologies

A Technical Overview

Dr Eugene Siow | 28.5.2019

Overview

- A Brief History
- Docker and Containers
- CI/CD and Registries
- Kubernetes
- Service Meshes
- Performance
- Security
- Recommendations of Use

Container Landscape 2019

CI/CD
Gitlab, Travis CI, Argo, Drone, ContainerOps, GitKube, Jenkins X, Circle CI, Weave Flux

Tools
Prometheus, OpenTracing, Kontena, KubeFlow, Jaegar, Helm

Machine Learning
Nvidia Saturn V + NGC Containers
AWS Deep Learning Containers
Google AI Platform
IBM Watson Machine Learning
Azure Stack
HP Blue Data

Service Mesh
Istio, Kiali, Linkerd

Orchestration Platform
Kubernetes, Pivotal Container Service, Apache Mesos, DC/OS, OpenShift, Deis, Docker Enterprise, Vagrant, Tutum

Cloud Platforms
Google Kubernetes Engine, Elastic Container Service, Jelastic, Azure Container Service, Pivotal Cloud Foundry, Apprenda, IBM Kubernetes Cloud, DigitalOcean CS, SUSE CaaS, Aliyun Container Service

Containers & Management
Docker, Kata, CoreOS rkt, LXC, containerd, OpenVZ, runc, Mesos Containeriser

Operating System
CoreOS, Photon, RancherOS, Snappy Ubuntu Core, Atomic, Nano Server

Plugins + Services
GlusterFS, Project Calico, CNI, Consul, Rook, Flocker, CoreDNS

Kernel Technologies
Chroot Jail
FreeBSD Jail
Linux Vserver
Oracle Solaris Containers
cgroups (Control Groups)

Compute
CPUs, FPGAs, GPUs, ASICs, TPUs, Neuro-Chips

Storage
SSDs, Helium Drives, HDDs, NVMe, SAN, Fusion ioMemory, Isilon, NAS

Network
Bridge, Fibre, Router, Switch, Diode, Firewall

Virtualisation
Baremetal
Hardware Virtualisation
Type2 Hypervisor

Security
HSM
Cryptoprocessor
Cryptographic Accelerator
Datacryptors

UNCLASSIFIED

A Brief History of Containerization

1979 to 2013

1979

Isolate a process and its children from the rest of the OS. **Cons:** root process can easily escape.

Chroot Jail was introduced in Version 7 Unix.

2000 – The FreeBSD Jail

2001

Linux Vserver for operating system-level virtualization.

2003

Google starts Borg as a large-scale internal cluster management system.

2004 – Oracle Solaris Containers

2005

OpenVZ for OS-level virtualization adopted by many hosting companies for VPSs.

Cons: Containers have to share the same architecture and kernel version.

2006 – Google's cgroups

2008

LXC (linux containers) used cgroups. Could limit and isolate resource usage.

2009 – Nexus
2010 - Vagrant
2011 – Nexus to Mesos

2013

Docker was introduced to make containers easy-to-use.

2013 – Warden
2013 – Borg to Omega

A Brief History of Containerization

2014 and Beyond

CoreOS (Container Linux) was released as an OS for container clusters.

2014

Istio, a service mesh, was introduced.

2017

2014

Google introduces Kubernetes.

2014 – Docker 1.0
2014 – Google's LMCTFY

2015

Cloud Native Container Foundation (CNCF) was founded.

2015 – Kubernetes 1.0 & 1.1
2015 – GKE on Google Cloud

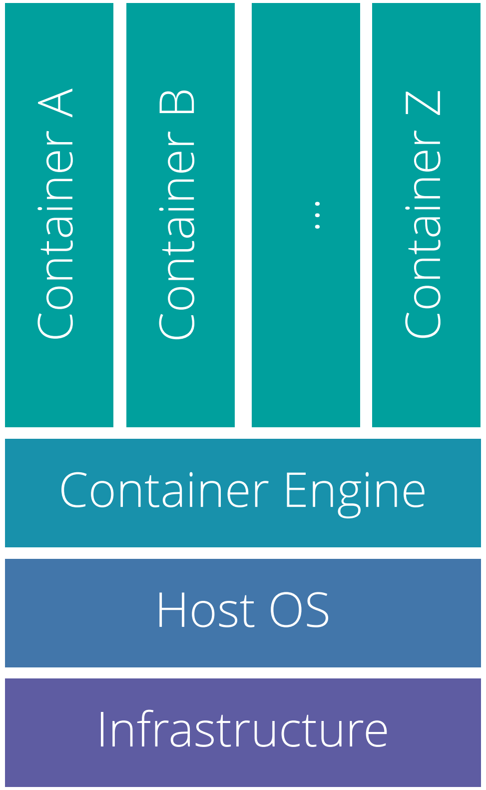
2017 – Moby Project
2018 – Istio 1.0

UNCLASSIFIED

Docker and Containers

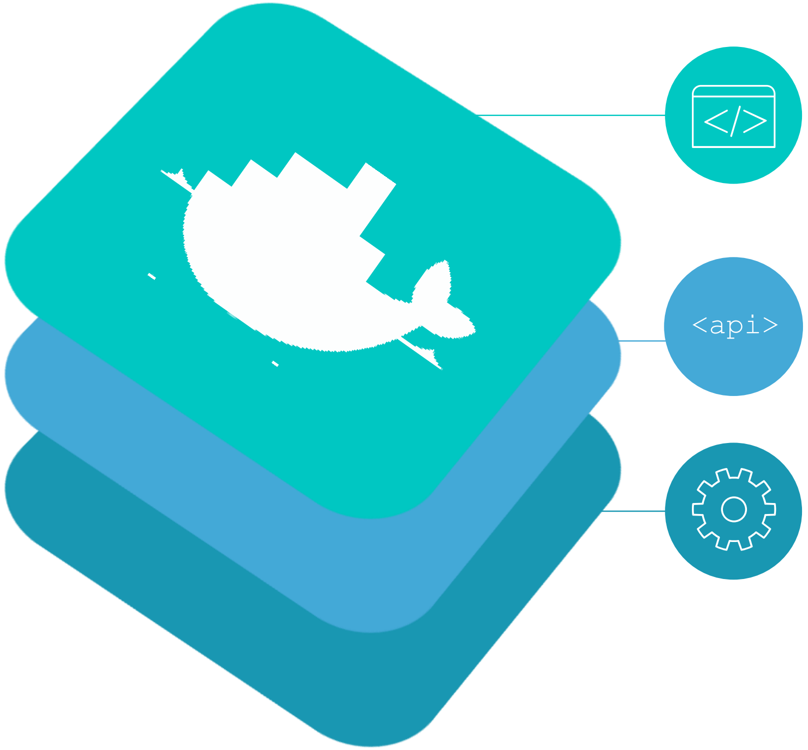
Container Architecture

OS-Level Virtualization



Docker Engine

Making Containers Easy-to-Use



Command Line Interface (CLI)

`docker pull dso/image:latest`

REST API

`POST http:/v1.24/containers/create`

Docker Daemon

Container Engine

Docker and Containers

How Docker Works

libcontainer and cgroups

Docker Daemon

libcontainer

Linux Kernel

Netlink | cgroups

Namespaces | Capabilities

SELinux | AppArmor

Infrastructure

Docker Concepts

Registry

Images

Containers

Networks

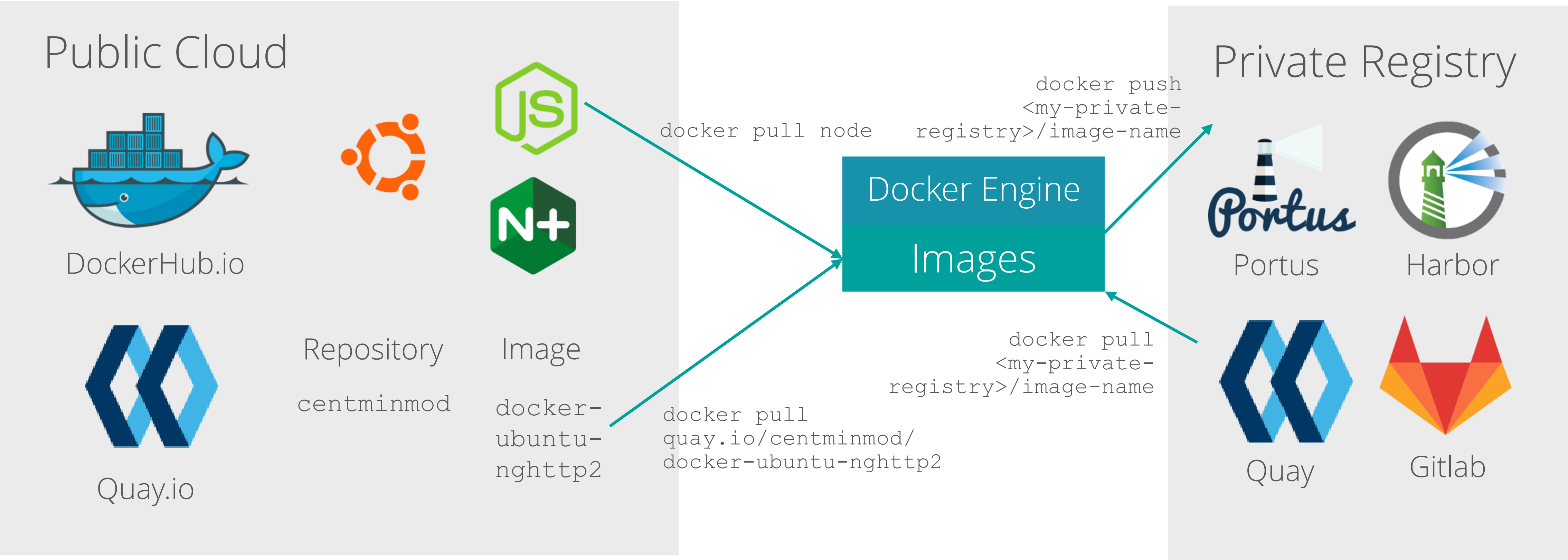
Volumes

Compose + Swarm/K8s

CI/CD and Registries

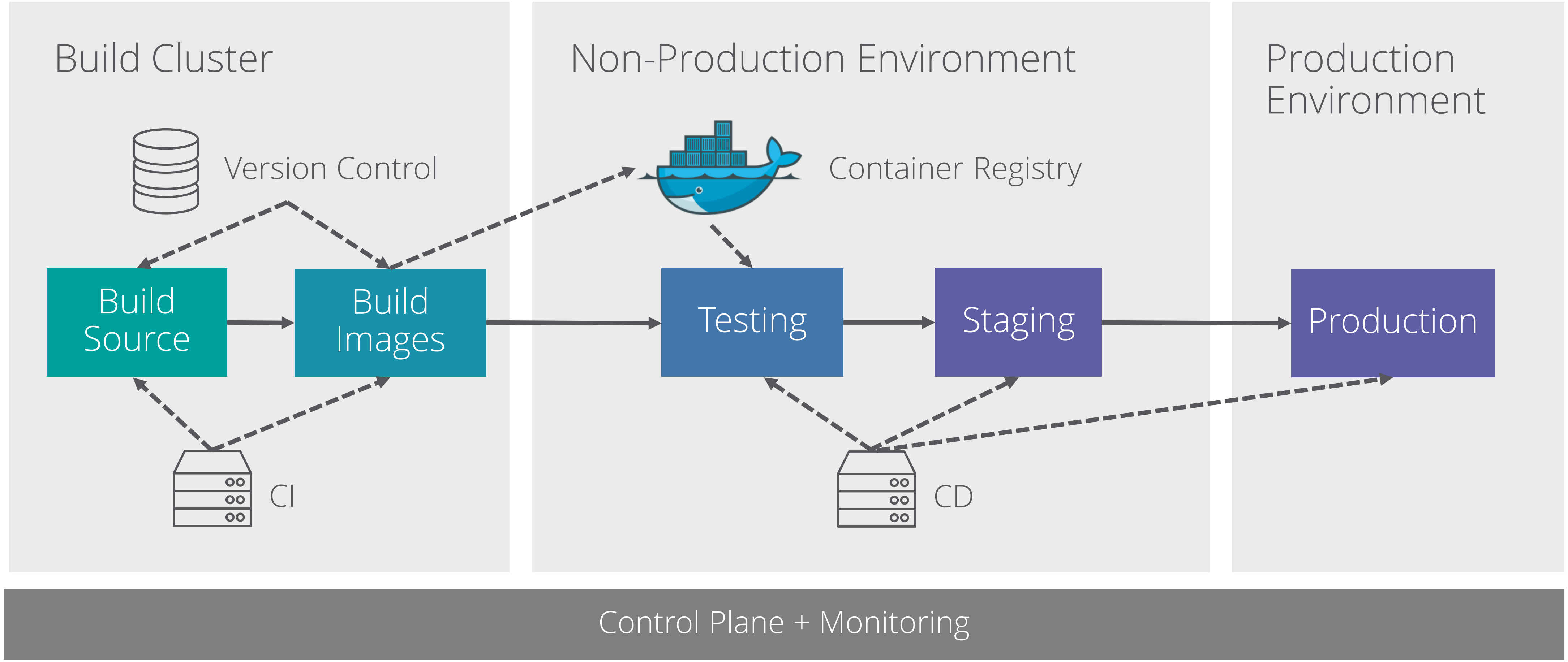
Container Registry

A container registry is a storage and distribution system for named container images.



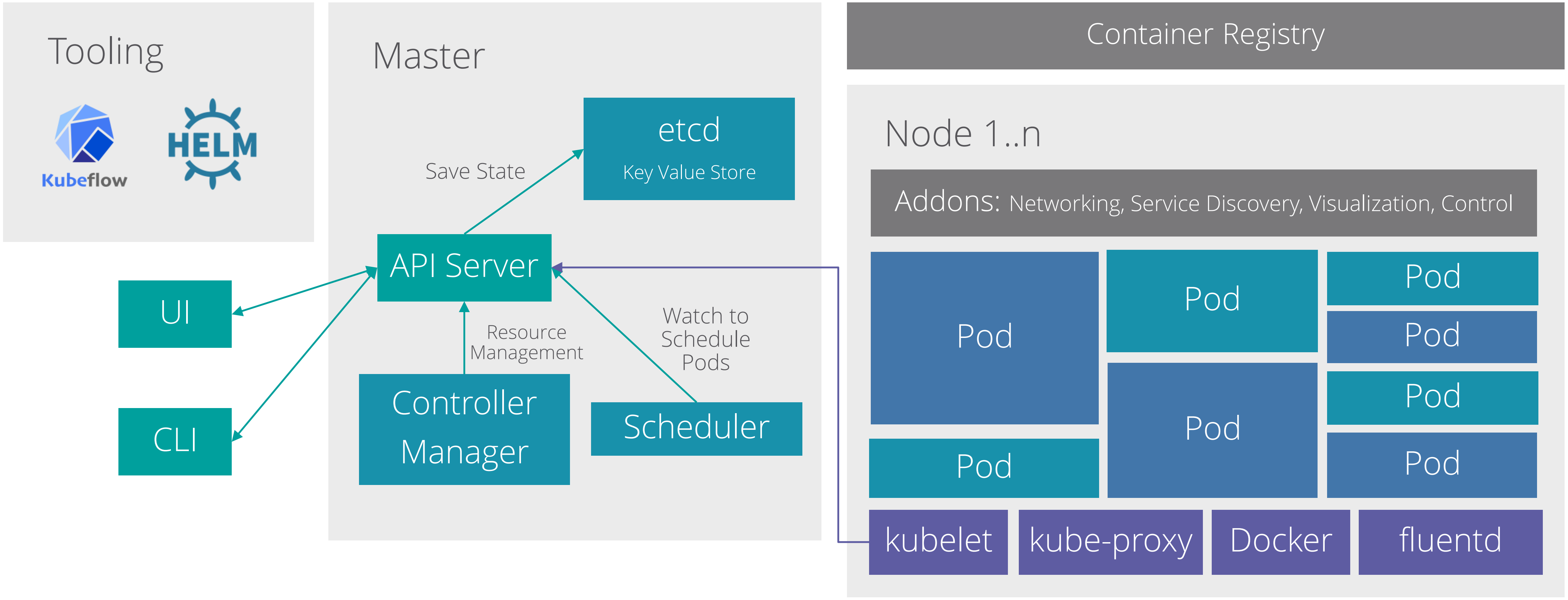
CI/CD and Registries

Continuous Integration (CI)/Continuous Delivery (CD) Pipeline



Kubernetes (K8s)

Kubernetes is a system for automating deployment, scaling, and management of containerized applications.



Kubernetes (K8s)

Concepts

Manages ReplicaSets

D Deployment

J Job

Finite task, run to completion

R ReplicaSet

Ensures n pods are running

At least 1 pod per node running

DS DaemonSet

SS StatefulSet

Pods that are not interchangeable across nodes

Objects

Pod
Basic Unit, a Process on Cluster

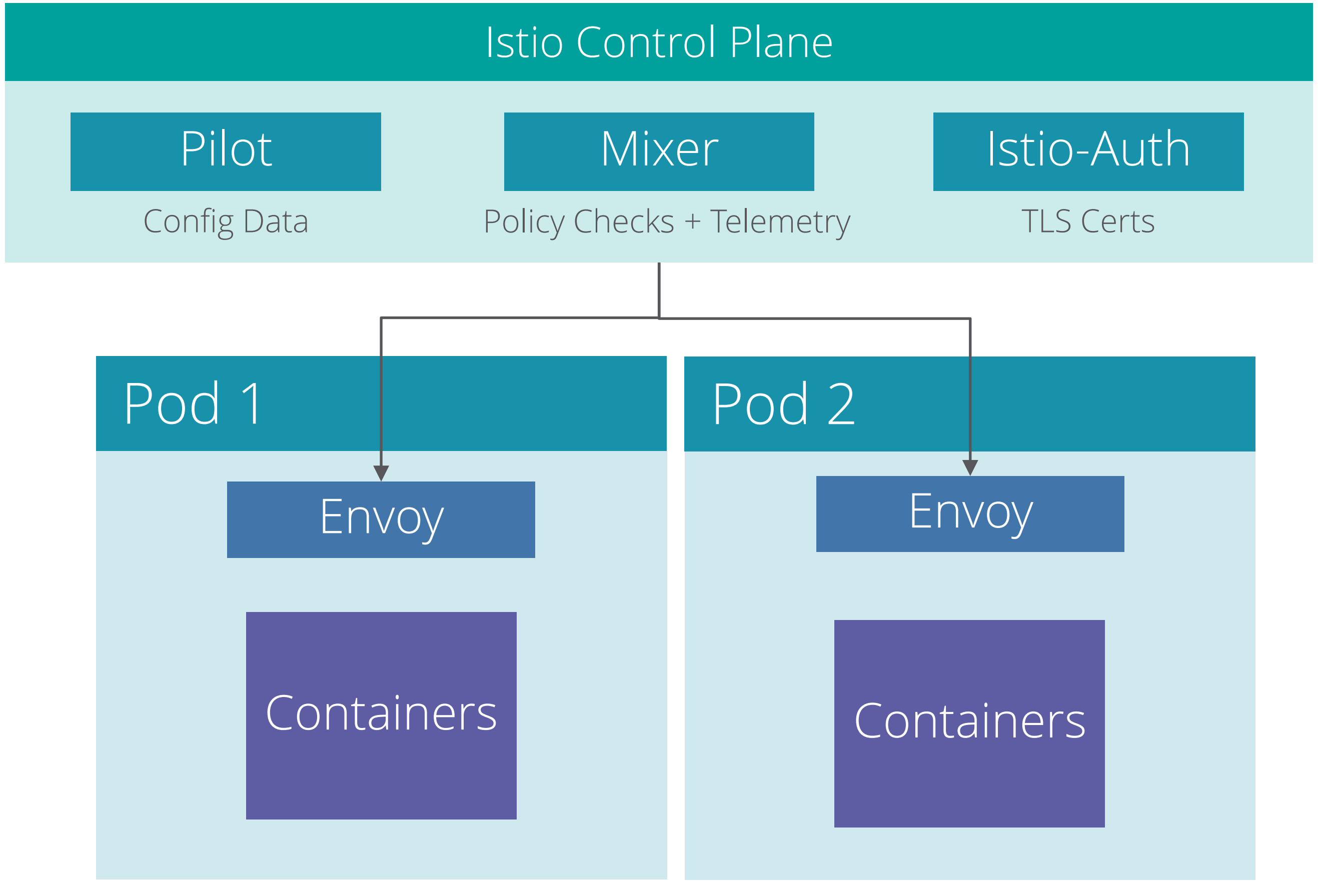
Volume
A storage abstraction for Pods

Service
Micro-service, Logical set of Pods and a Access Policy

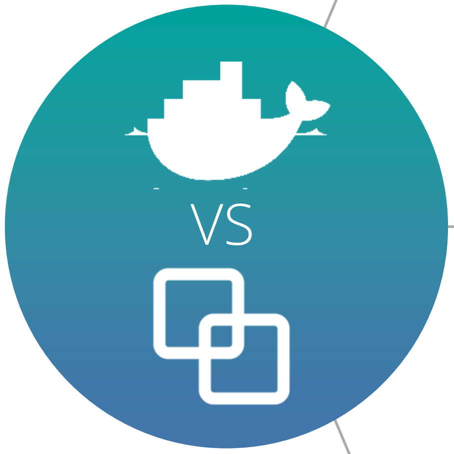
Namespace
A virtual cluster representation

Service Meshes

Istio is a service mesh that runs on top of container orchestration platforms like Kubernetes or Mesos.



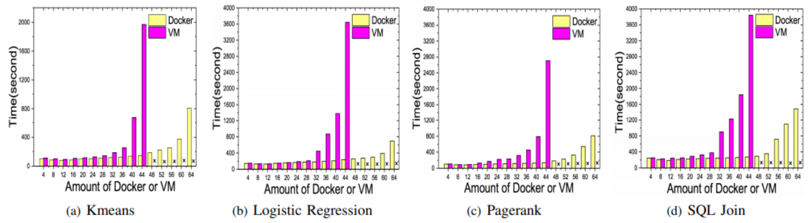
Performance



Scalability

Containers outperform VMs in execution time as the number of containers/VM reaches saturation.
Containers have better scalability.

Execution time across Spark tasks on a big data cluster as number of containers and VMs are increased.

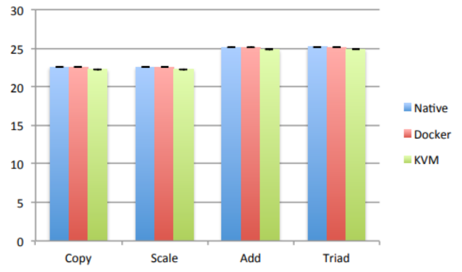


Zhang, Qi, et al. "A comparative study of containers and virtual machines in big data environment." *arXiv preprint arXiv:1807.01842* (2018).

Memory Bandwidth

Containers and VMs have **almost the equal** memory performance on the STREAM benchmark.
Due to hardware TLB pre-fetching.

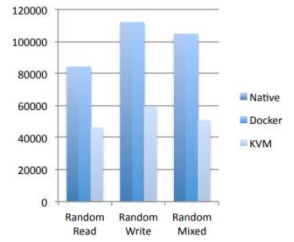
Stream performance in GB/s on one socket (eight cores) measuring sustainable memory bandwidth when performing simple operations on vectors.



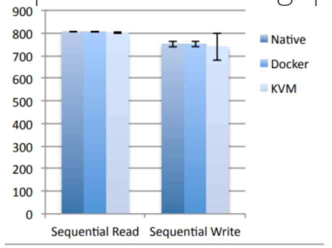
Random IOPs

Containers offer close to native performance whereas **VMs have half the performance** as each IOP goes through QEMU.
<u>Note: sequential read performance is different, VMs are equal.</u>

Random I/O throughput (IOPS).



Sequential I/O throughput.



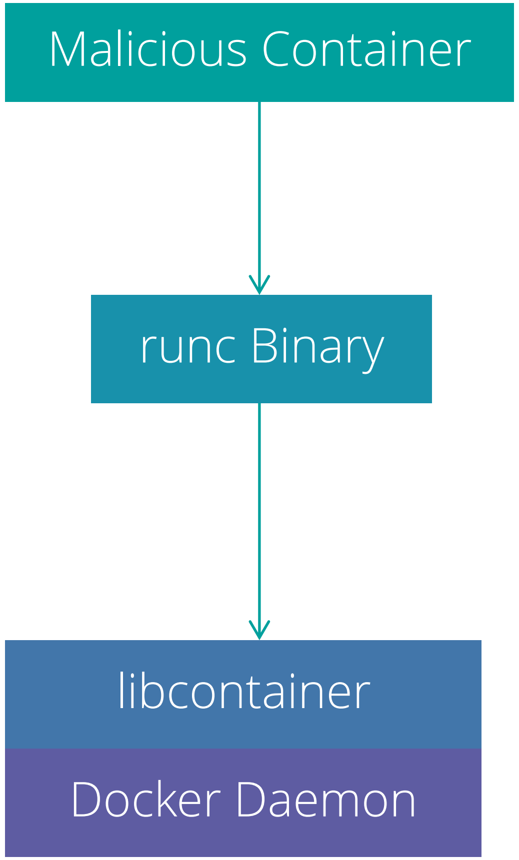
Felter, Wes, et al. "An updated performance comparison of virtual machines and linux containers." *2015 IEEE international symposium on performance analysis of systems and software (ISPASS)*. IEEE, 2015.

Security

Container Isolation

Process-level isolation?

Flaw: CVE-2019-5736



Replacing the target binary in the container with one that refers back to the runc binary. Either by:

1. Attaching a privileged container
2. Starting it with a malicious image and making it execute itself.

The Linux kernel normally would not allow the runc binary on the host to be overwritten while runc is executing.

The attacker can instead open a file descriptor to the process file and then proceed to reopen the binary and try to write to it in a busy loop from a separate process.

The attacker can then run any command as root within a container and can take over the container host.