Containers

a.k.a. OS-Level Virtualization

Containers

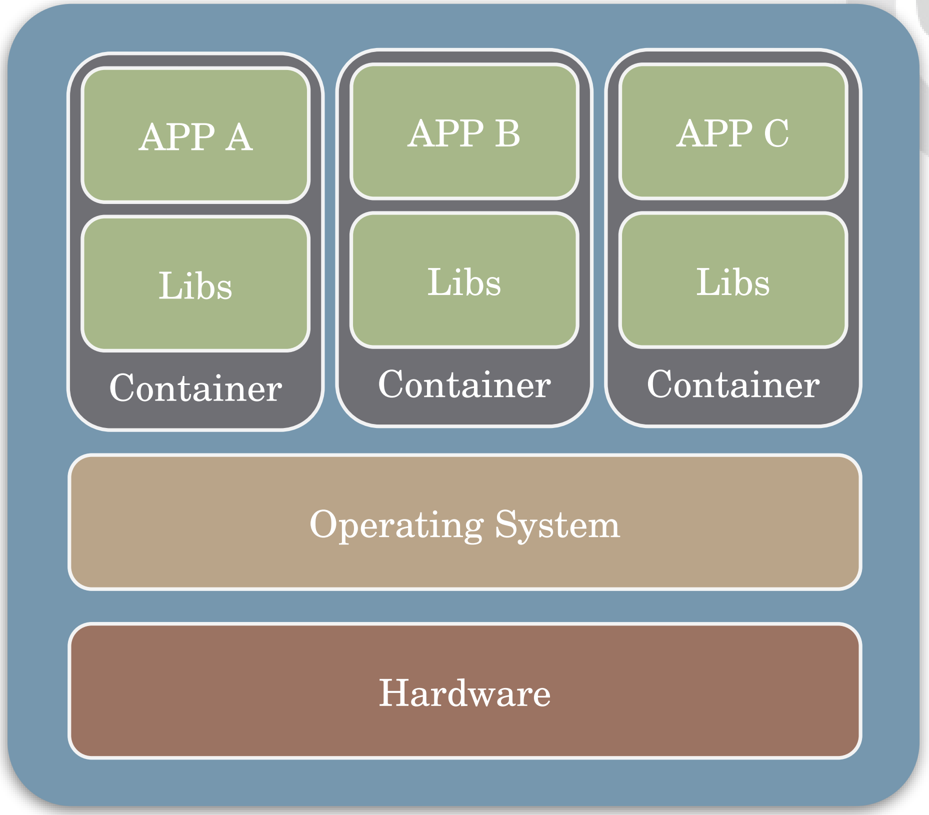
- Form of operating system virtualization
- OS Kernel creates multiple isolated **user space** instances
  - Process from one user space sees only it’s space
  - Process from one user space “cannot” affect process of another user space
- This makes them more lightweight and portable than VMs
- The “de facto” standard for modern microservices architecture

Serios vulnerabilities have been reported many times with container isolation mechanisms. There is always a risks of container escape vulnerability. This makes then less secure than virtual machines and not preferred approach for some use cases.

Containers

- Namespace isolation
  - Process tree
  - Networking
  - User IDs
  - File Systems
  - IPC

- Resource limitation
  - CPU
  - Memory
  - I/O
  - Network



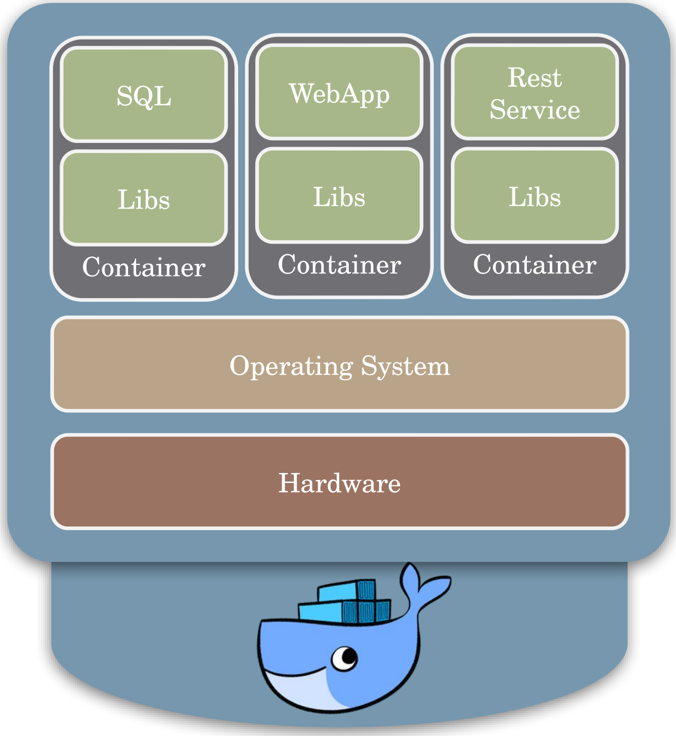
Brief History of Containers

- 1979 - Chroot
- 2000 - FreeBSD Jails
- 2001 - Linux VServer
- 2004 - Solaris Containers
- 2005 - Open VZ (Open Virtuozzo)
- 2007 - Control groups (cgroups)
- 2008 - LXC (LinuX Containers)
- 2013 – Docker
- 2015 – Kubernetes
- 2016 – Windows Native Containers
- More…

Docker

What is Docker?

- Creating, working with, and managing containers
- Standardized packaging for software
- Simplify building, shipping, running apps
- Isolate apps from each other
- Share the same OS kernel
- Works for all major Linux distributions
- Open Source platform



Docker components

- **Docker Image**

  The basis of a Docker container. Represents a full application

- **Docker Container**

  The standard unit in which the application service resides and executes

- **Docker Engine**

  Creates, ships and runs Docker containers on physical or virtual

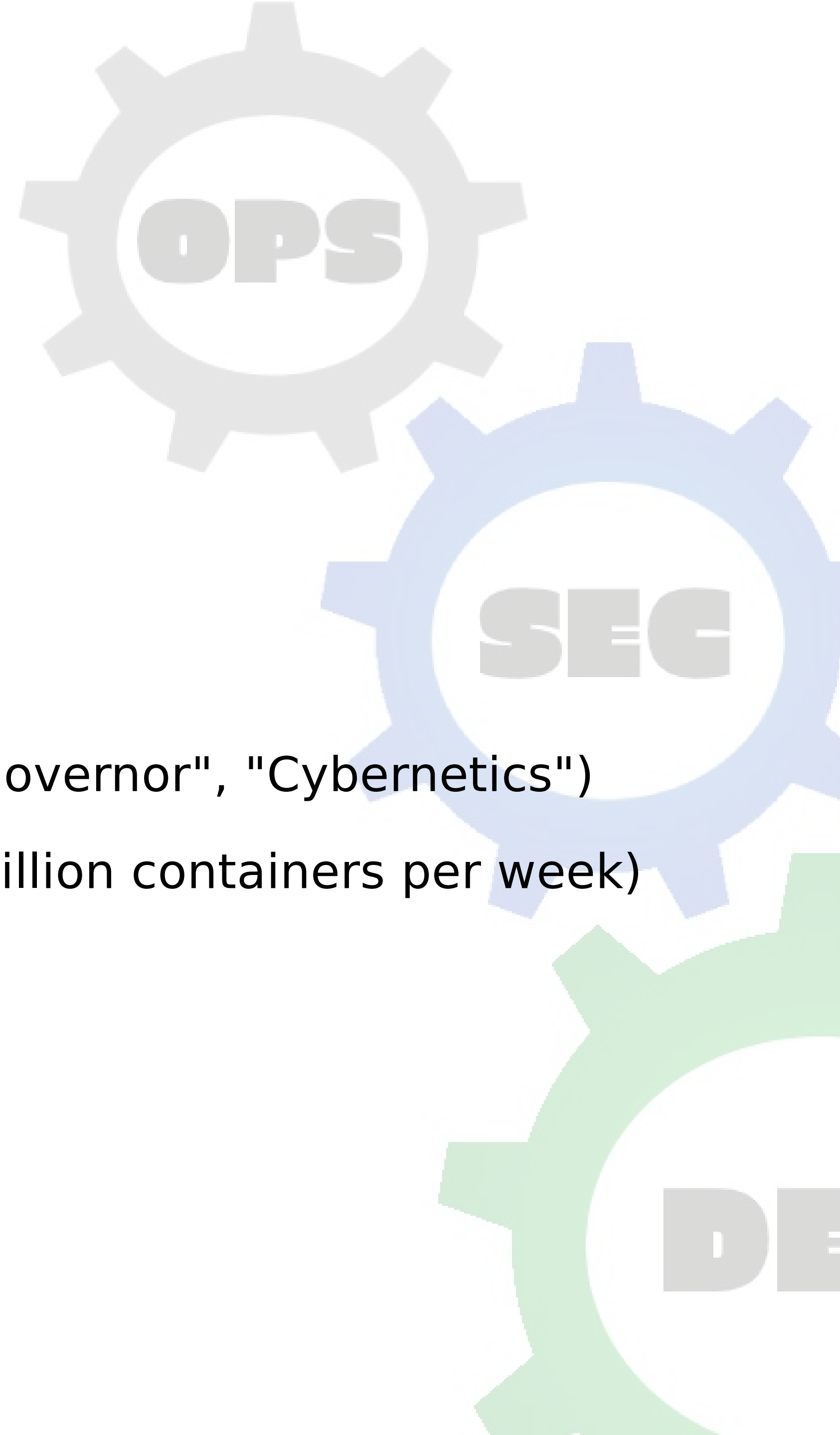
- **Registry Service (Docker Hub(Public) or Docker Trusted Registry(Private))**

  Cloud or server-based storage and distribution service for your images

Kubernetes

What is Kubernetes?

- Automates containerized and distributed applications
- Container orchestration platform
- Cluster management orchestration platform
- Open Source platform
- Also know as k8s
- Ancient Greek word for "Helmsman" (Root of the word "Governor", "Cybernetics")
- Born from a Google internal project (Google launches 2 billion containers per week)



Core Features

- Service discovery
- Load balancing
- Self-healing ideology
- Declarative approach

Kubernetes Cluster

- The cluster is the heart of Kubernetes
- Set of nodes that run containerized applications
- Aims to run containers in efficient, automated, distributed, and scalable manner
- Master and worker nodes
- Decouples the containers from the underlying hardware layer

Kubernetes Nodes

- Runs container workloads
- May be a virtual or physical machine
- May be in cloud or on-prem
- Managed by k8s cluster control plain
- Contains k8s services required to manage containers and PODs
- Typically you have several nodes in a cluster
  - You can have a single node cluster for learning

PODs

- Basic building block
- Smallest deployable units
- Group of one or more containers
  - Deeply coupled
  - Shared network
  - Shared storage
- Each POD has a unique dynamic IP address

Service

- Service discovery
- Load balancing

Namespaces

- Mechanism for isolating groups of resources
- Single cluster – Multiple namespaces
- Names of resources need to be unique within a namespace
- Applicable only fro namespaced objects (e.g. Deployments, Services, etc)
  - Not applicable for cluster-wide objects (e.g. StorageClass, Nodes, PersistentVolumes, etc).
- Intended for use in environments with many users and teams

Configuration Units

- ConfigMaps
  - Object used to store non-confidential data
  - Do not store confidential data!
  - Key-value pairs
  - Can be consumed as environment variables or file
- Secrets
  - Object used to store sensitive data
  - Passwords, tokens, encryption keys, etc.
  - Can be consumed as environment variables or file

Other Units

- ReplicaSets
  - Ensures that the number of desired pods "replicas" are running at any time.
- Deployments
  - Describe the desired state of the application (pods, replica sets).
  - Easy version updates for any software
- DaemonSets
  - Runs a POD on every node in a cluster
- StatefulSets:
  - Clustered applications (e.g. PostgreSQL, MongoDB, Elasticsearch)
  - Startup/shutdown ordering
  - Stable hostname and storage