Lecture 11.2
Virtualization

EN 600.320/420
Instructor: Randal Burns
8 March 2017
Containers

- Not light-weight VM’s

- Analogy: Houses vs Apartment Buildings

- Houses / VM’s
  - Self-contained
  - Own infrastructure

- Apartments / Containers
  - Smaller footprint
  - Shared infrastructure

- Application delivery more then virtualization
Comparing the stacks

From: https://www.docker.com/what-docker#VM
Why Containers?

- Ease of Use – build once, run anywhere
- Speed – lightweight, less resources, boot time
- Distribution – make them publicly available
- Modular and Scalable – a container for each service
Challenges

● Security
  - Containers share the same kernel
  - VM’s don’t share same kernel

● Complicated Management
  - data storage
  - Build process
  - Stateless can be issue for troubleshooting

● No standard eco-system yet
History of Containers

- FreeBSD Jails
  - Released with 4.0 in 2000
  - Chroot ++
  - Segregated users, processes and networks + filesystem

- Solaris Zones
  - Concept of zones
  - Use snapshot and cloning in ZFS
  - Easily duplicate zones

Linux Containers

- Run multiple isolated Linux systems on a single host
- Uses namespaces and cgroups in Linux kernel
- Same as chroot but more isolation
- Provides a virtual operating system not a VM with own CPU, memory, I/O
Container Execution Frameworks

- Google - Kubernetes (Successor to Borg)
- Docker Swarm
- LXD – Canonical Linux Containers
- AWS - Elastic Container Service
Docker

- Started with the aim to build a specialized LXC
- Fundamentally the same, aka use namespaces and cgroups
- More level of abstraction for H/W
Docker Concepts

From: https://medium.freecodecamp.com/a-beginner-friendly-introduction-to-containers-vms-and-docker-79a9e3e119b#.8cbumvzs5
Docker Vocabulary

- **Image** - Base read-only template, Has "layers"

- **Container** – Wraps application in a box, RW filesystem

- **Engine** – Similar to a hypervisor
  - Build, run, manage containers
  - Client – User’s interface
  - Daemon – Run’s on host machine and executes commands

- **Volume** – Data part of a container, Persistent

- **Hub** – Registry service (Image app store)
From: https://www.docker.com/what-docker/#/VM

Lecture 11.2: Virtualization