CLOUD COMPUTING APPLICATIONS

Containers

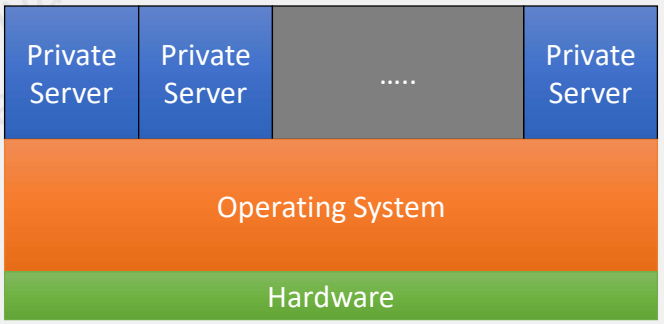
Prof. Reza Farivar

Isolation

- “I once heard that hypervisors are the living proof of operating system's incompetence”
  - -Glauber Costa, 2012
- hypervisors have indeed provided a remedy for certain deficiencies in operating system design
- for some cases, containers may be an even better remedy for those deficiencies

Operating System-Level Virtualization

- Virtualizing a physical server at the operating system level, enabling multiple isolated and secure virtualized servers to run on a single physical server
- Examples:
  - Solaris Containers (2004)
  - FreeBSD Jails (2000)
  - Linux Containers
    - Linux Vserver (2001)
    - OpenVZ (2005)
    - Process Container (2006) → cgroups
    - LXC (2008)
    - Docker (2013)



OS-Virtualization / Containers

- OS (operating system) virtualization is how we generally refer to this type of "light-weight" virtualization
- processes think they see a virtual kernel, but are all sharing the same real kernel under the hood
- kernel acts as a sort of hypervisor in ensuring that container/virtualization boundaries are not crossed
- The goal of containers is to support all of the resource-isolation use cases, without the overhead and complexity of running multiple kernel instance

Operating System-Level Virtualization

- Hypervisor (VM)
- One real HW, many virtual HWs, many OSs
- High versatility – can run different Oss
- Lower density, performance, scalability
- Performance overhead is mitigated by new hardware features (such as VT-D)

- Containers (CT)
- One real HW (no virtual HW), one kernel, many userspace instances
- Higher density, natural page sharing
- Dynamic resource allocation
- Native performance: [almost] no overhead