6. Virtualization and Cloud Computing

IT5406 - Systems and Network Administration

Level III - Semester 5

List of sub topics

6.1. Cloud Computing [Ref 1: Pg. (270-293)]

6.2. Virtualization [Ref 1: Pg. (900-910)]

6.3 Containers

6.1. Cloud Computing

- Cloud computing is the practice of leasing computer resources from a pool of shared capacity.
- Users of cloud services provision resources on demand and pay a metered rate for whatever they consume.
- The transition from servers in private data centers to the now ubiquitous cloud has been rapid and dramatic.
    - Cloud providers create technically advanced infrastructure
    - They design custom server chassis that maximize energy efficiency and minimize maintenance.
    - They automate aggressively to allow rapid expansion and reduce the likelihood of human error.
    - They use purpose-built network infrastructure with custom hardware and software fine-tuned to their internal networks.

Ref 1: Pg (270 - 272)

6.1. Cloud Computing

6.1.1. Cloud Platform Choices

- Factors influence for choosing a cloud provider
  - Cost
  - Past experience
  - Compatibility with existing technology
  - Security, or compliance requirements
  - Internal politics

Ref 1: Pg (273)

6.1. Cloud Computing

6.1.2. Cloud Service Fundamentals

- Three (3) categories in cloud services,
  - Infrastructure-as-a-Service (IaaS)
  - Platform-as-a-Service (PaaS)
  - Software-as-a-Service(SaaS)

Ref 1: Pg (276)

6.1. Cloud Computing

6.1.2. Cloud Service Fundamentals

- Access to the cloud - Most cloud providers' primary interface is some kind of web-based GUI. New system administrators should use this web console to create an account and to configure their first few resources.
- Regions and availability zones - Cloud providers maintain data centers around the world.
- Virtual private servers - It is a virtual machine (instance) that runs on the provider's hardware.
- Networking - Cloud providers let you create virtual networks with custom topologies that isolate your systems from each other and from the Internet.
- Storage - Cloud providers have the largest and most advanced storage systems.
- Identity and authorization
- Automation
- Serverless functions - Cloud functions are a model of code execution that do not require any long-lived infrastructure.

Ref 1: Pg (276-283)

6.1. Cloud Computing

6.1.3. Clouds – Amazon, Google, Digital Ocean

- Major public cloud providers
  - Amazon Web Services (AWS) https://aws.amazon.com/
  - Google Cloud Platform (GCP)
  - DigitalOcean (DO).

Ref 1: Pg (273-276)

6.1. Cloud Computing

6.1.4. Cost Control

- Components in Cloud service and pricing
  - The compute resources of virtual private servers (consumes CPU cycles) - Pricing is per hour of use.
  - Internet data transfer - Pricing is per GiB or TiB transferred.
  - Storage - Pricing is per GiB or TiB stored per month.

- Customers should try their best to design and apply methods to reduce the cost for using the cloud services.

Ref 1: Pg (291-293)

6.2. Virtualization

6.2.1. Virtualization

- Hypervisors -
  - is a software/ hardware that mediates between virtual machines (VMs) and the underlying hardware on which they run.
  - Hypervisors are responsible for sharing system resources among the guest operating systems, which are isolated from one another and which access the hardware exclusively through the hypervisor.
  - Guest operating systems are independent.

Ref 1: Pg (901)

6.2. Virtualization

6.2.1. Virtualization

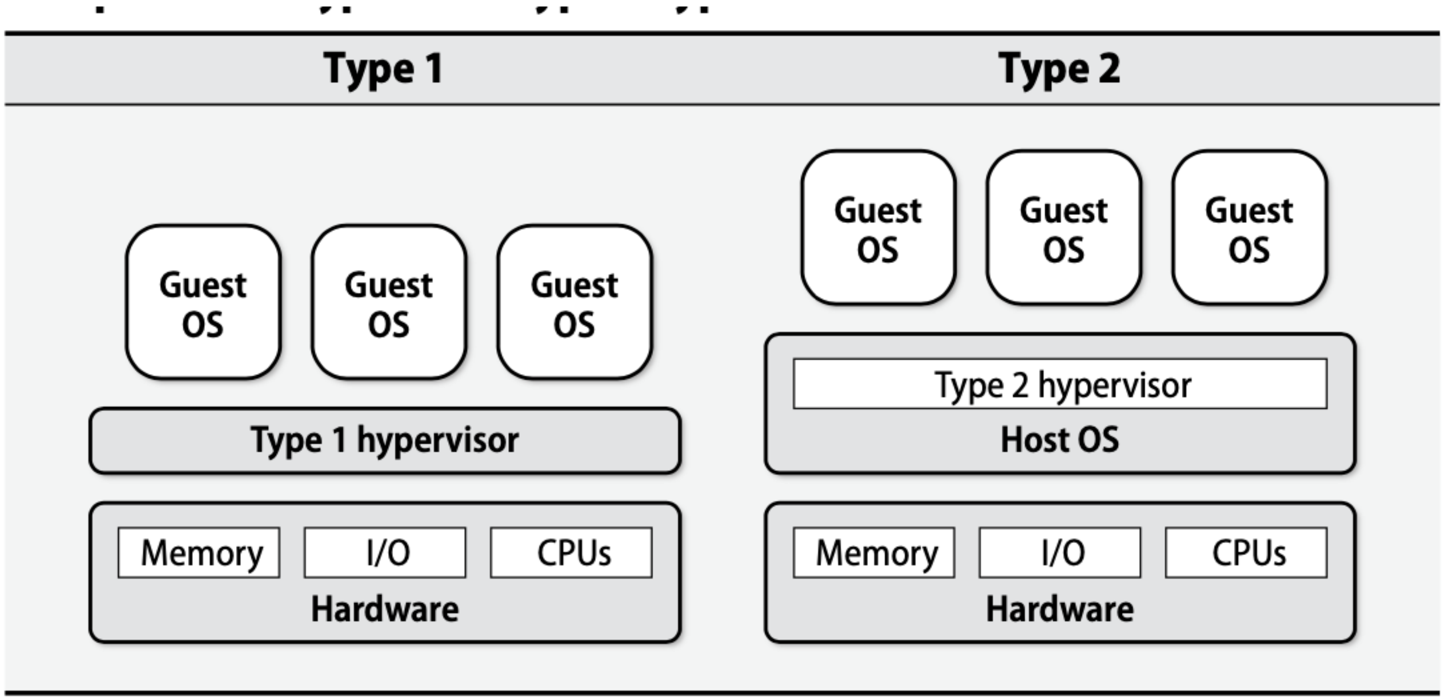
- Types of hypervisors
  - Full virtualization
  - Paravirtualization
  - Hardware-assisted virtualization
  - Paravirtualized drivers

Ref 1: Pg (901-903)

6.2. Virtualization

6.2.1. Virtualization

- Type 1 vs. type 2 hypervisors



ID: Recreate this diagram

Ref 1: Pg (903)

© 2022 e-Learning Centre, UCSC

6.2. Virtualization

6.2.1. Virtualization

- Live migration - Virtual machines can move between hypervisors running on different physical hardware in real time, in some cases without interruptions in service or loss of connectivity.
- Virtual machine images - Virtual servers are created from images, which are templates of configured operating systems that a hypervisor can load and execute.
- Containerization - OS-level virtualization—or containerization—is a different approach to isolation that does not use a hypervisor. Instead, it relies on kernel features that isolate processes from the rest of the system.

Ref 1: Pg (904-905)

6.2. Virtualization

6.2.2. Virtualization with Linux

- Xen and KVM are the leading open source virtualization projects for Linux.
- Xen, now a project of the Linux Foundation, powers some of the largest public clouds, including Amazon Web Services and IBM's SoftLayer.
- KVM is the kernel-based virtual machine integrated into the mainline Linux kernel.

Ref 1: Pg (905)

6.2. Virtualization

6.2.2. Virtualization with Linux

- Xen
  - Paravirtual hypervisor
  - Open source
  - Xen is a bare-metal hypervisor that runs directly on the physical hardware.

https://xenproject.org/

Ref 1: Pg (906-908)

6.2. Virtualization

6.2.2. Virtualization with Linux

- KVM
  - Kernel-based Virtual Machine
  - Full virtualization platform
  - Under KVM, the Linux kernel itself serves as the hypervisor.
  - Memory management and scheduling are handled through the host’s kernel, and guest machines are normal Linux processes.

https://www.linux-kvm.org/page/Main_Page

Ref 1: Pg (908-910)

6.3 Containers

6.3.1. Background and Core Concepts

Kernel support

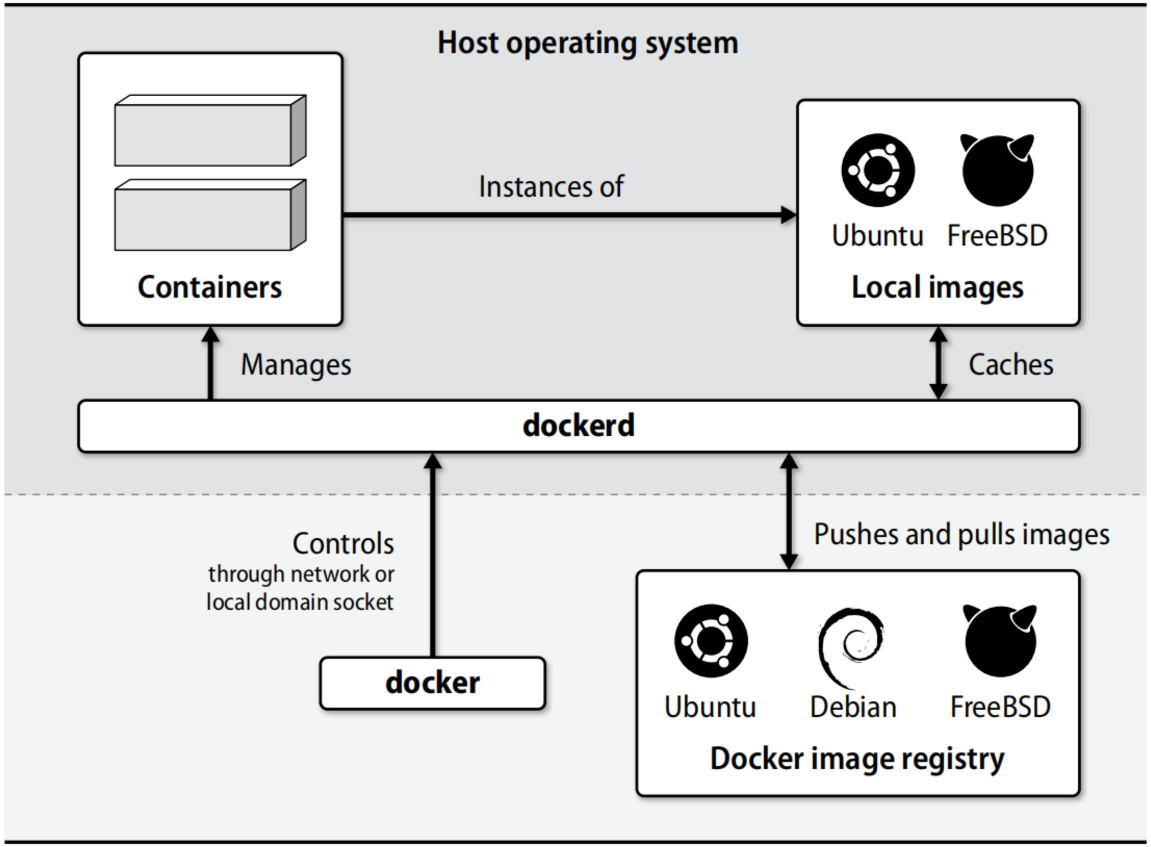
- Namespaces
- Control groups
- Capabilities
- Secure computing mode

Docker now relies on an improved, standards-based container run time dubbed containerd. It too relies on Linux namespaces, cgroups, and capabilities to isolate containers.

6.3 Containers

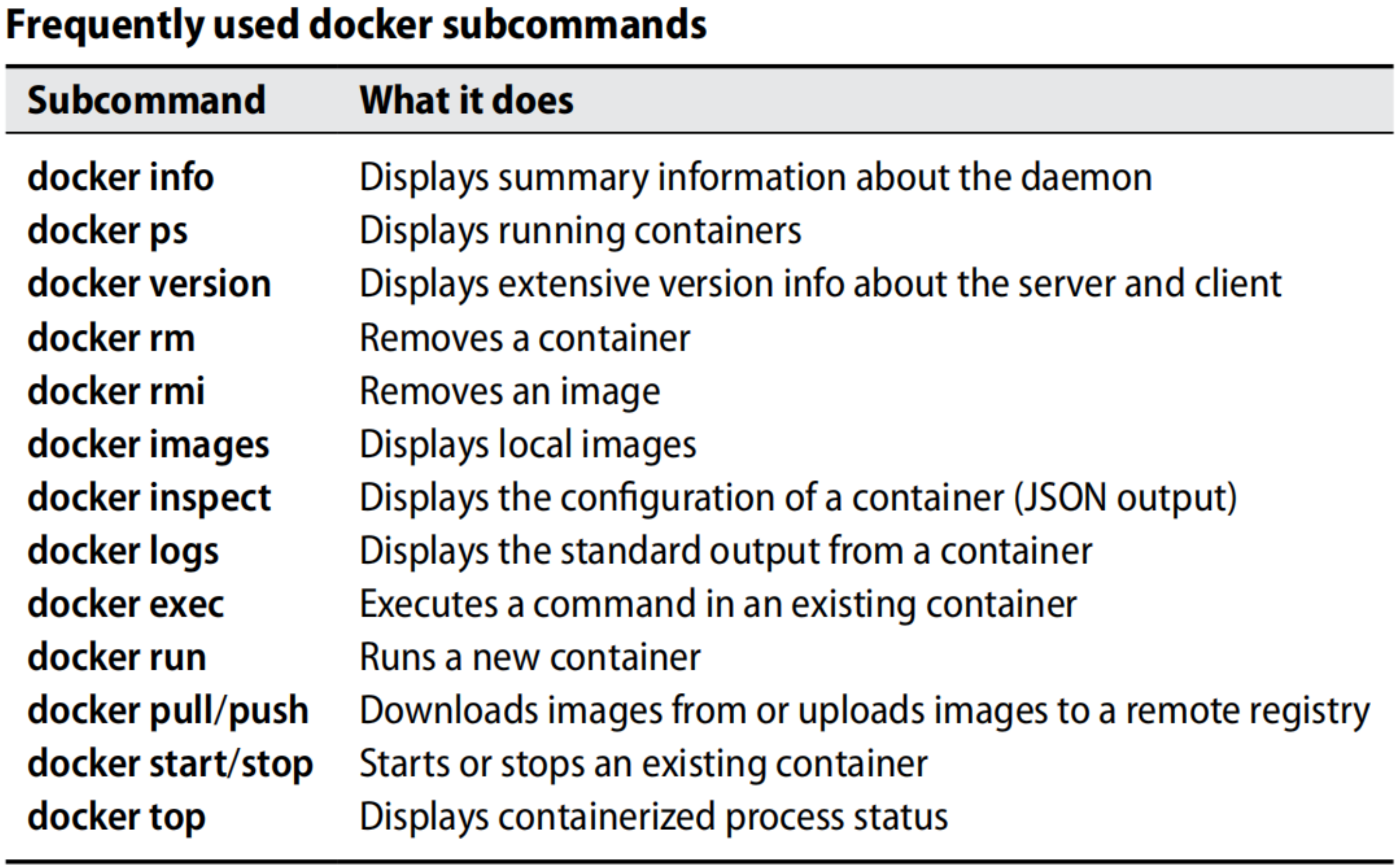
6.3.2. The Open Source Container Engine – Docker

- Basic architecture



6.3 Containers

6.3.2. Open Source Container Engine – Docker



Frequently used docker subcommands

Subcommand	What it does
docker info	Displays summary information about the daemon
docker ps	Displays running containers
docker version	Displays extensive version info about the server and client
docker rm	Removes a container
docker rmi	Removes an image
docker images	Displays local images
docker inspect	Displays the configuration of a container (JSON output)
docker logs	Displays the standard output from a container
docker exec	Executes a command in an existing container
docker run	Runs a new container
docker pull/push	Downloads images from or uploads images to a remote registry
docker start/stop	Starts or stops an existing container
docker top	Displays containerized process status

6.3 Containers

6.3.2. Open Source Container Engine – Docker

- Installation
- Client setup
- The container experience
- Volumes
- Data volume containers
- Docker networks
- Storage drivers
- dockerd option editing

Refer the coresponding topics in the reference book for practical activities

6.3 Containers

6.3.2. The Open Source Container Engine – Docker

- Image building
    - Choosing a base image
    - Building from a Dockerfile
    - Composing a derived Dockerfile
- Registries

Refer the coresponding topics in the reference book for practical activities

6.3 Containers

6.3.3. Containers in Practice

- Few rules of thumb to help you adjust to life inside a container:
  - Use the cron daemon from the host (or a systemd timer) to schedule a short-lived container that runs the job and exits
  - Need to log in and check out what a process is doing? Don’t run sshd in your container. Log in to the host with ssh, then use docker exec to open an interactive shell.
  - If possible, set up your software to accept its configuration information from environment variables.

6.3 Containers

6.3.3. Containers in Practice

- Ignore the commonly dispensed advice “one process per container.” Split processes into separate containers only when it makes sense to do so.
- Focus on the automatic creation of containers for your environment.
- If you’re accessing a container manually to fix something, figure out what the problem is, resolve it in the image, then replace the container.

6.3 Containers

6.3.3. Containers in Practice

- Logging
- Security advice
    - Restrict access to the daemon
    - Use TLS
    - Run processes as unprivileged users
    - Use a read-only root filesystem
    - Limit capabilities
    - Secure images
- Debugging and troubleshooting

Refer the coresponding topics in the reference book for details and practical activities

6.3 Containers

6.3.4. Container clustering and management

- Container management systems’ features
- A synopsis of container management software
- Kubernetes
- Mesos and Marathon
- Docker Swarm
- AWS EC2 Container Service

Refer the coresponding topics in the reference book for details

Reference

- Ref 1. Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley and Dan Mackin “UNIX and Linux System Administration Handbook ” (5th Edition), Pearson Education, Inc., 2018.