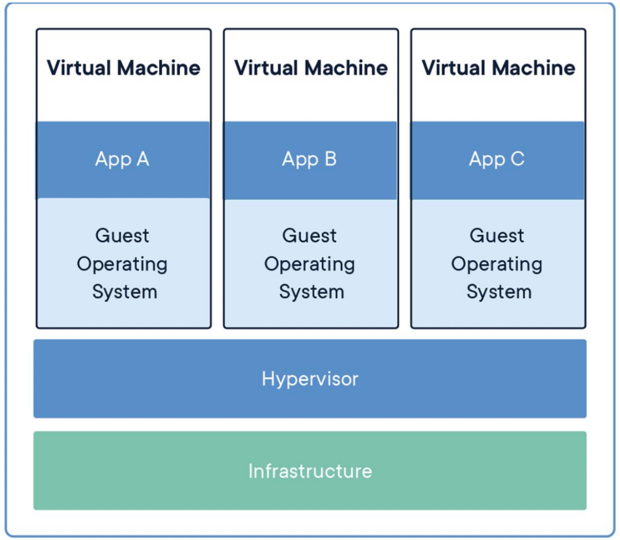
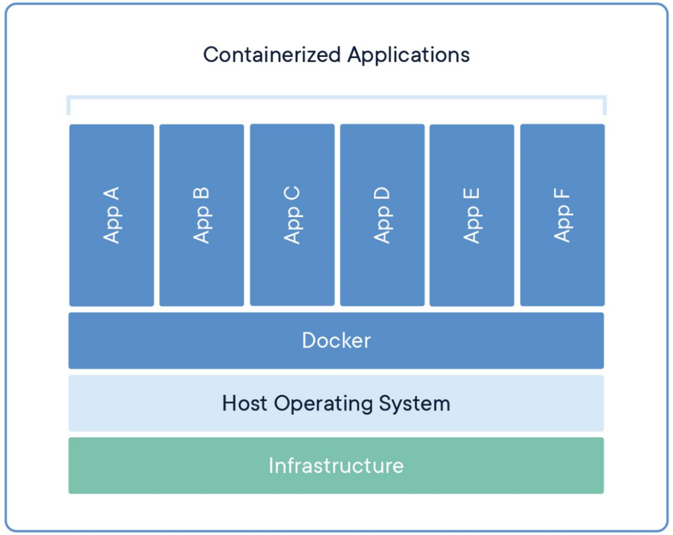
Docker

VM vs Docker





VM:

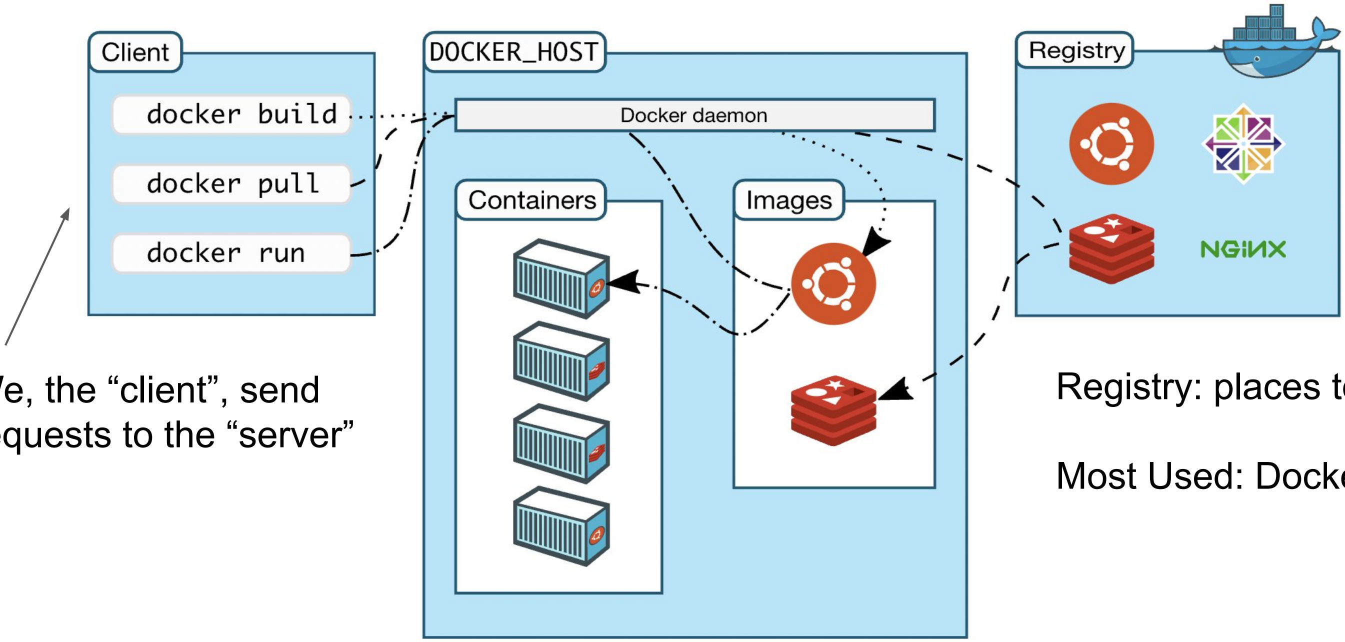
1. abstraction of physical hardwares
2. own OS

Docker:

1. abstraction of apps and dependencies
2. share the same OS kernel, different filesystems
3. small and fast

Img: https://www.docker.com/resources/what-container

Docker Architecture: Client-Server Model



We, the “client”, send requests to the “server”

Registry: places to store Docker Images.

Most Used: Docker Hub / GitHub

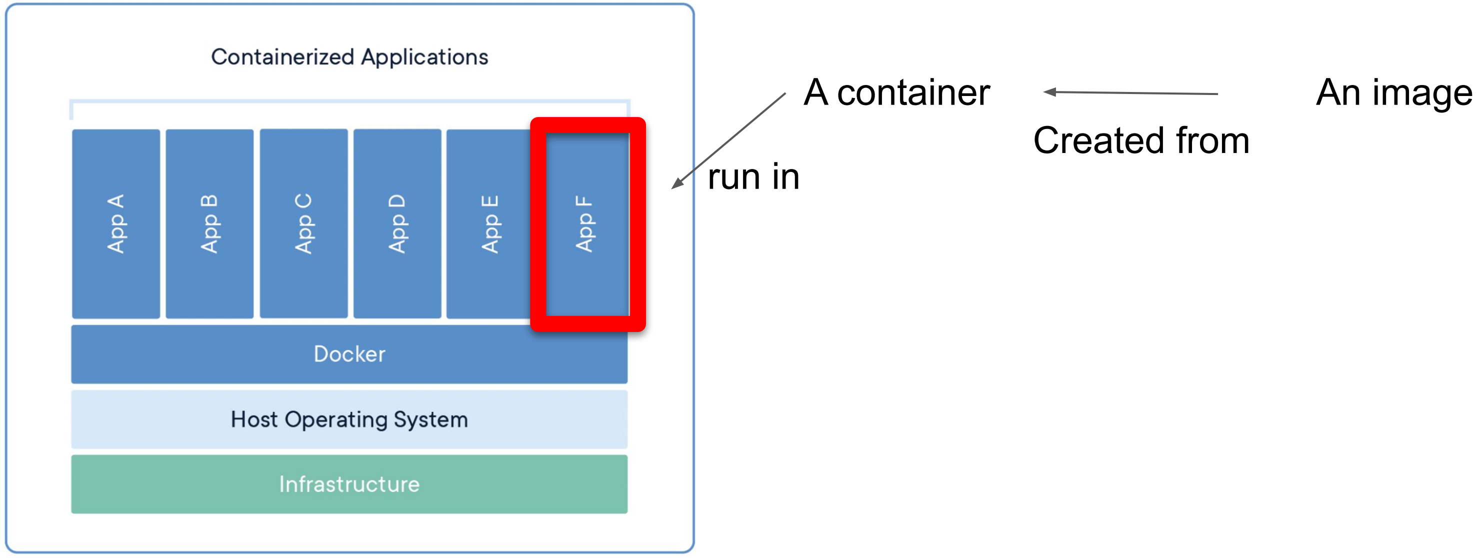
Docker Daemon, a background process, process the requests.

Img: https://docs.docker.com/get-started/overview/

Images And Containers

Image: read-only template, tells daemon how to create a **container**.

Container: a runnable instance of an **image**. App is run in container.



Images And Containers

Image: read-only template, tells how to create a **container**.

Container: a runnable instance of an **image**.

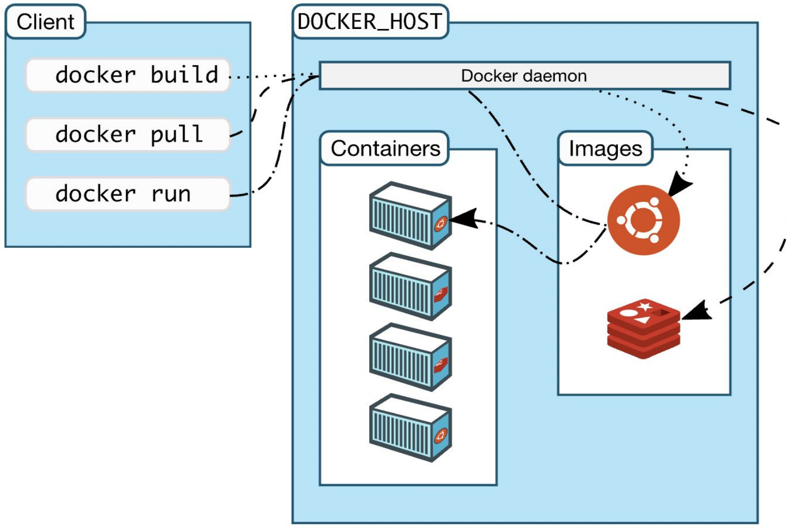
1. Containers from same image: initially look the same.
2. Containers: well isolated
3. Control How Isolated.
    a. Network: expose port
    b. Storage: mount
    c. Subsystems

Images And Dockerfile

Image: describes to daemon how to create a container

Dockerfile: saves the descriptions.

Dockerfile: tells daemon how to create an image



Container and Image: Creation

Create a container:
1. Only from a local image (e.g. local in your laptop)
2. If image not found locally, will first download a copy from Registry (the cloud)

Create an image:
1. Pull an existing image from Registry.
2. Save a container as a new image.
3. Build from dockerfile.

Storage:

Containers can share the same folder.

A container and the host machine can share the same folder.

Installation: on ACFS/RedHat/CentOS

1. https://docs.docker.com/engine/install/centos/
2. ACFS ==> Red Hat Enterprise Linux 8.5 (Ootpa)
3. Need to install CentOS version.
4. Require root/sudo rights.

Commands:

1. sudo yum install -y yum-utils
2. sudo yum-config-manager --add-repo \ https://download.docker.com/linux/centos/docker-ce.repo
3. sudo yum install docker-ce docker-ce-cli containerd.io
4. sudo systemctl start docker
5. sudo systemctl enable docker.service
6. sudo systemctl enable containerd.service

Post Installation:

1. Both installing and using docker need root/sudo right.
2. Don’t want “sudo” when **using** docker:
    a. create a Unix group called “docker”
    b. add users to it.
    c. Users in “docker” group can run “docker run …” instead of “sudo docker run …”

Commands:

1. sudo groupadd docker
2. sudo usermod -aG docker $USER

From:
https://docs.docker.com/engine/install/linux-postinstall/

Problem:

1. Needs root/sudo, or in the “docker” group
2. Anyone using docker effectively has access to all files. (by mounting)
3. Meaning multi user isolation is difficult. (maybe we can run docker inside docker)

Installation: on Mac

https://docs.docker.com/desktop/mac/install/

1. Download the dmg,
2. Doubleclick dmg, give root permissions

Installation: on Ubuntu

https://docs.docker.com/desktop/mac/install/

1. Download the dmg, then doubleclick. Done.

```
1.   sudo apt-get update
2.   sudo apt-get install ca-certificates curl gnupg lsb-release
3.   curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo \
     gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
4.   echo \
      "deb [arch=$(dpkg --print-architecture) \
     signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \
     https://download.docker.com/linux/ubuntu \
     $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
5.   sudo apt-get update
6.   sudo apt-get install docker-ce docker-ce-cli containerd.io
```