Lesson 8

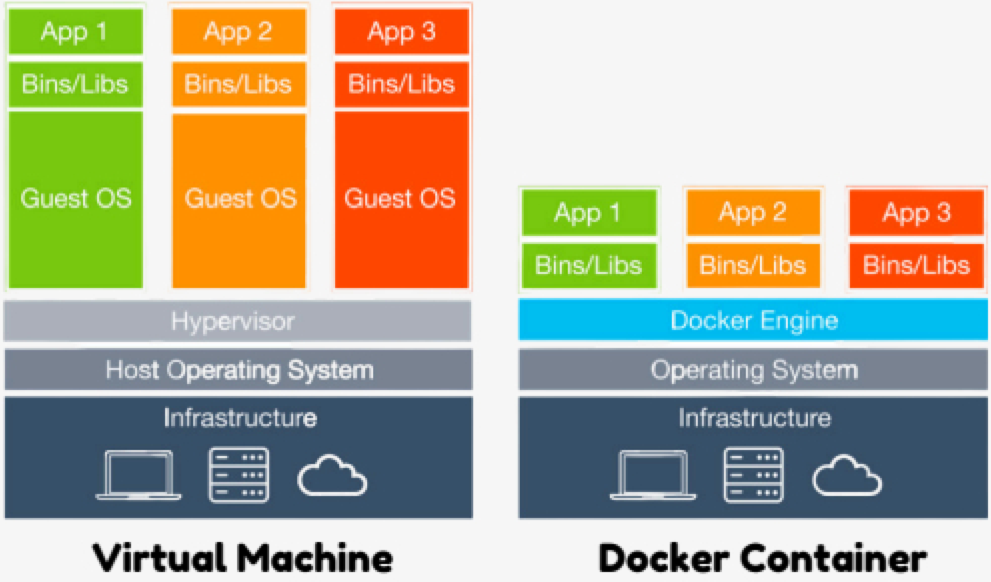
Docker

It is time to package our apps ;)

Learn About

- Virtual Machine and Docker
- Docker Compose
- Dockerfile Best Practices

Virtual Machine and Docker (1)



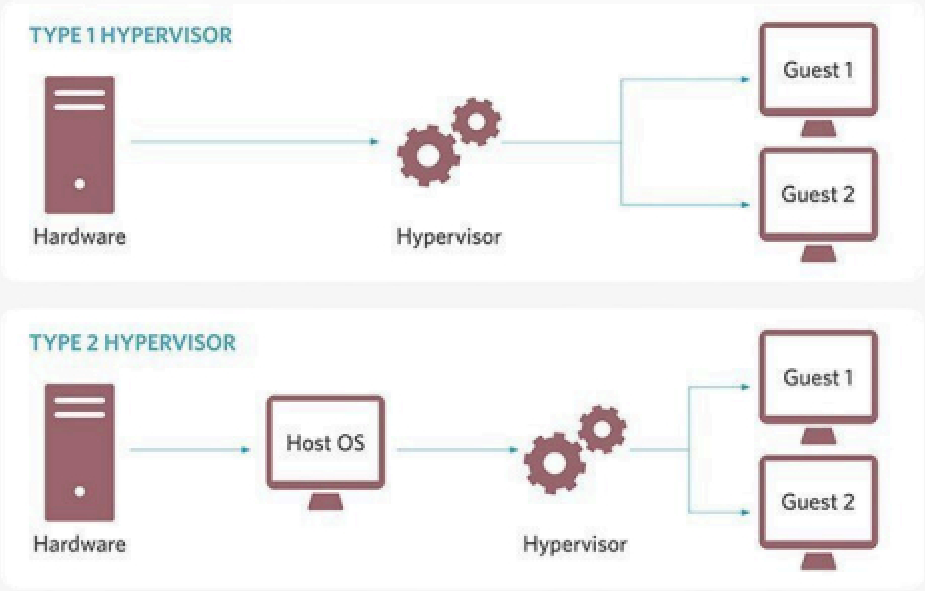
Virtual Machine and Docker (2)

	Virtual Machine	Docker Container
Size	Heavyweight	Lightweight
OS	Isolated OS	Share the host OS with other containers
Startup time	Slow	Very fast
Memory	More	Less
Security	Fully isolated and hence more secure	Process-level isolation, possibly less secure

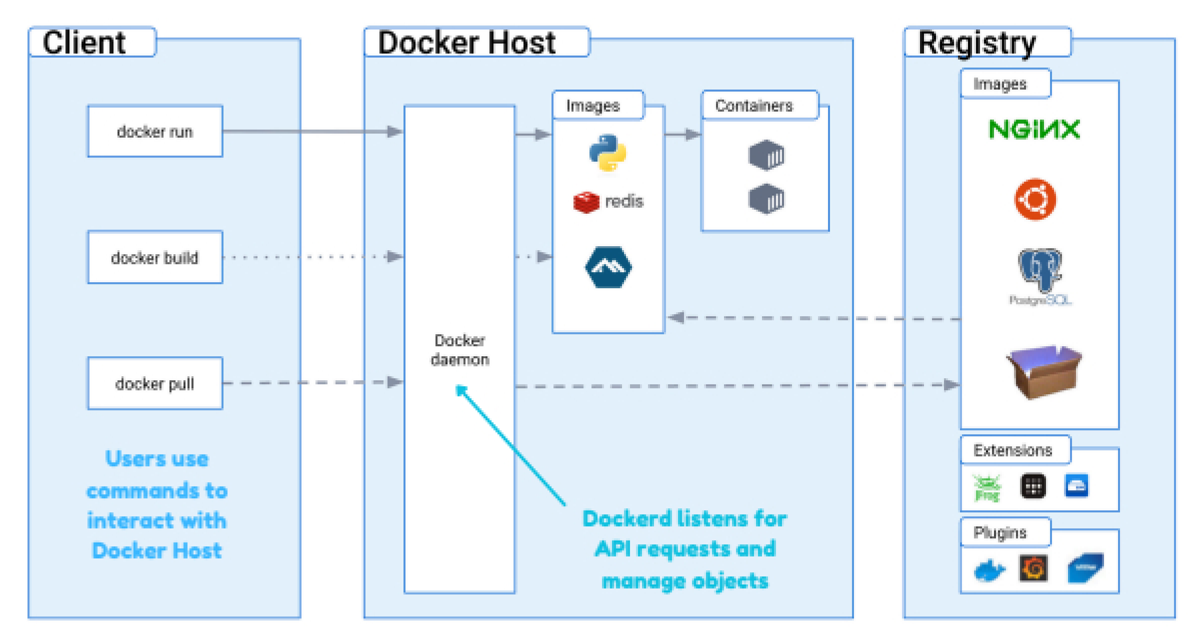
Hypervisors (I)

	Type 2 hypervisor (Virtual Box or QEMU)	Type 1/Bare metal hypervisor (Hyper-V or KVM)
Performance	Worse, it depends on the OS to allocate computing resources	Better, since it runs directly on the host machine's physical hardware
Workload	Small and medium	Large
Complexity	Low	High

Hypervisors (2)



Docker Architecture



Users use commands to interact with Docker Host

Dockerd listens for API requests and manage objects

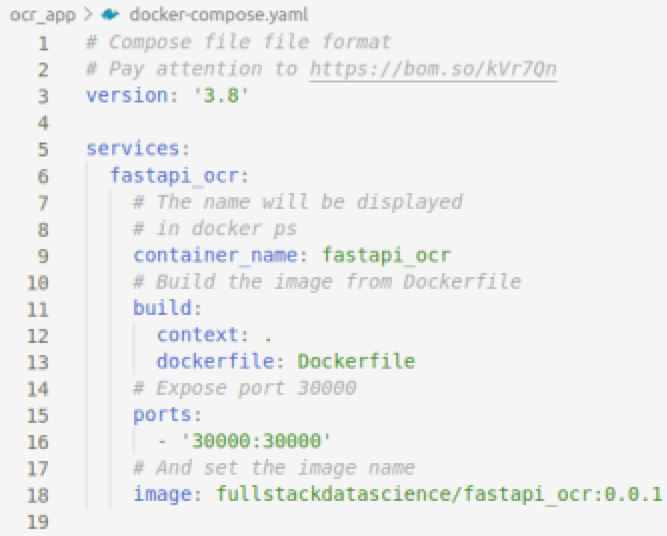
Docker Basic Commands

Commands	Explanation
docker build -t quandv/ocr:v1.0.0 .	Build a docker image using current context path ('.')
docker tag quandv/ocr:v1.0.0 fullstackdatascience/ocr:v1.0.0	Rename an image from quandv to fullstackdatascience
docker run -ti fullstackdatascience/ocr:v1.0.0 bash	Start a container, access to it, and play around as if our machine
docker exec -ti contname bash (or sh)	Access to a running container
docker pull nginx:latest	Pull an image from dockerhub
docker stats or ctop	CPU and memory usage of containers

Debug Docker Containers

Commands	Explanation
docker logs contname	Get current logs of the container
docker exec -ti contname1 ping contname2	Call to other container to test connection
docker events	List all events that happened in all containers (e.g. delete a container)
docker run -v .:/data -e myvar=10 -ti nginx:latest bash	Mount a current local dir to folder /data in the container, and add an environment variable myvar to the container
docker inspect contname	Investigate information of a container

Docker Compose



ocr_app > docker-compose.yaml

```yaml
# Compose file file format
# Pay attention to https://bom.so/kVr7Qn
version: '3.8'

services:
  fastapi_ocr:
    # The name will be displayed
    # in docker ps
    container_name: fastapi_ocr
    # Build the image from Dockerfile
    build:
      context: .
      dockerfile: Dockerfile
    # Expose port 30000
    ports:
      - '30000:30000'
    # And set the image name
    image: fullstackdatascience/fastapi_ocr:0.0.1

```

Docker Compose Basic Commands

Commands	Explanation
docker compose -f docker-compose.yaml up -d	Start all services in the compose (running on the background)
docker compose -f docker-compose.yaml down	Stop and remove all containers, networks, volumes, images create by the up command
docker compose -f docker-compose.yaml start (or stop)	Start/stop all containers in the compose (w/o remove)
docker copmpose ls	List all composes

Dockerfile Best Practices

The what	The how
Small image size	Using multistage build, a minimal base image, reduce number of layers and remove unnecessary libs or dependencies
Build fast	Leverage build cache (remember don't use alpine images :(, the standard is slow to pull, use the slim)
Easy to maintain	Don't use too old base images or libs, use LABEL in Dockerfile to note maintainers names
Secured	Scan for vulnerabilities with Docker Scout

Thank You!

Created by
Quan Dang