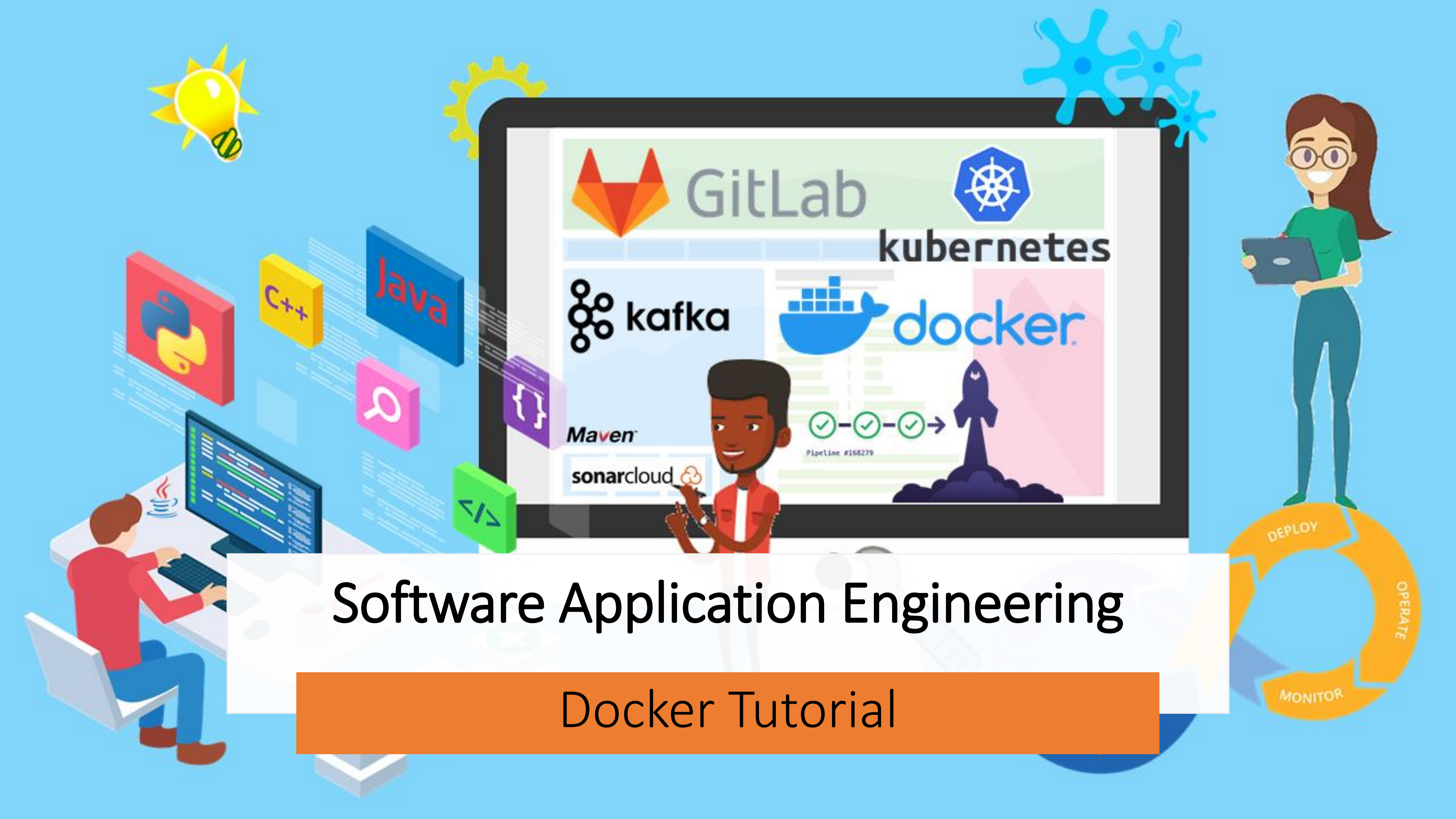
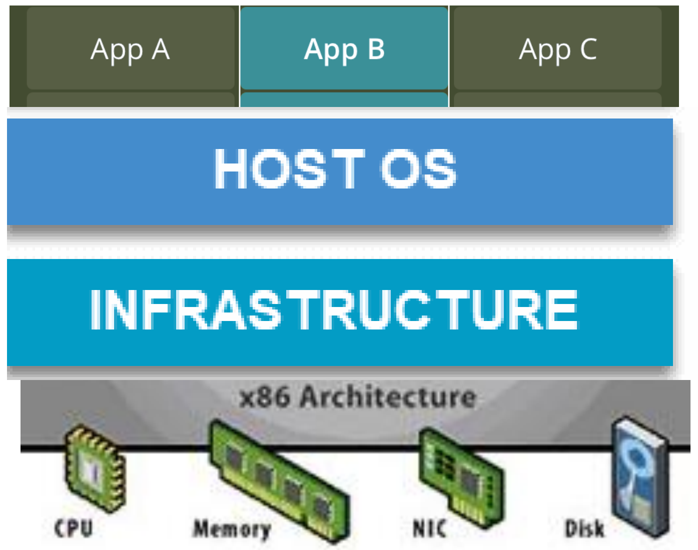


Software Application Engineering

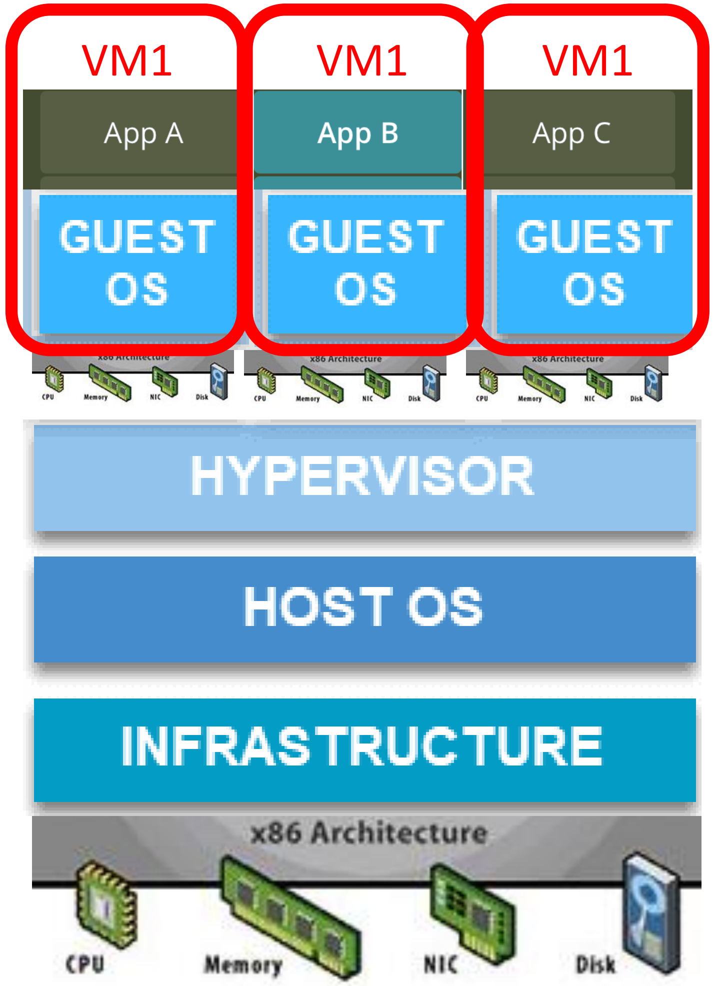
Docker Tutorial

VM and Containers

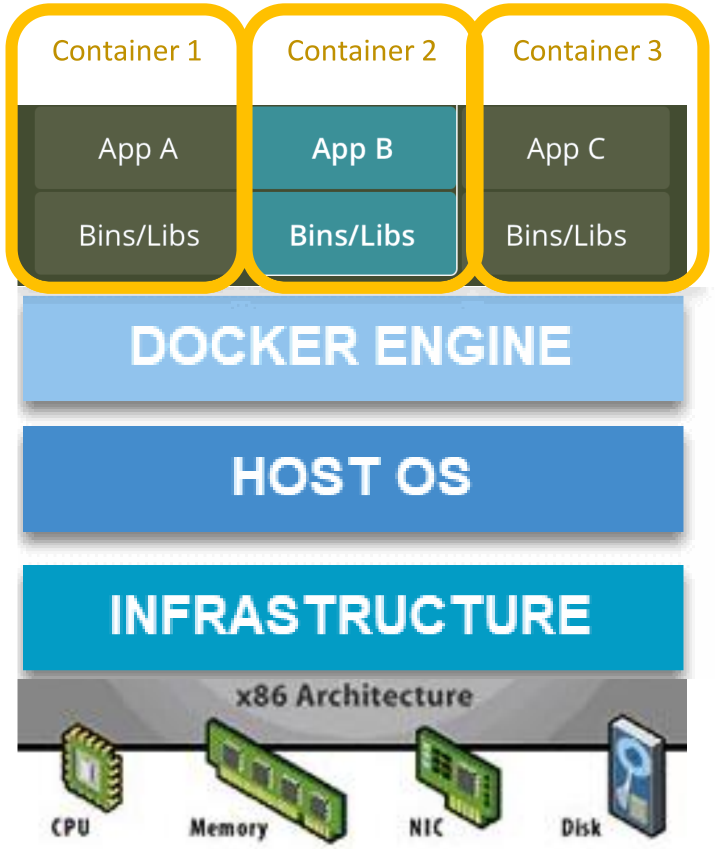
bare-metal server



VM abstracts the entire hardware server



Container abstracts the OS kernel

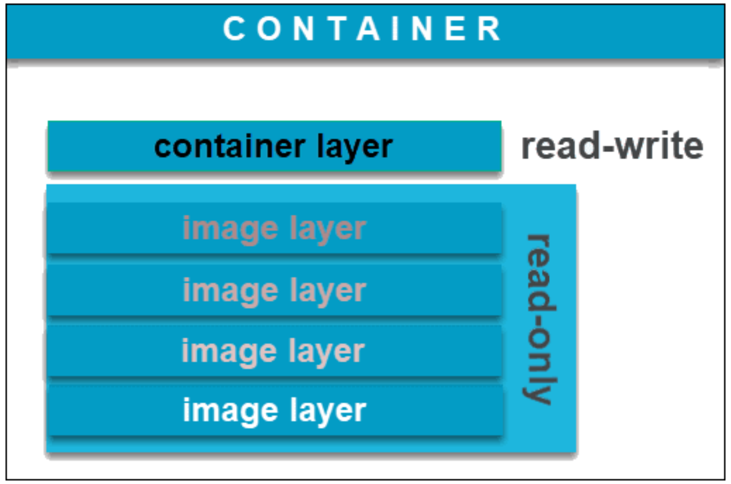


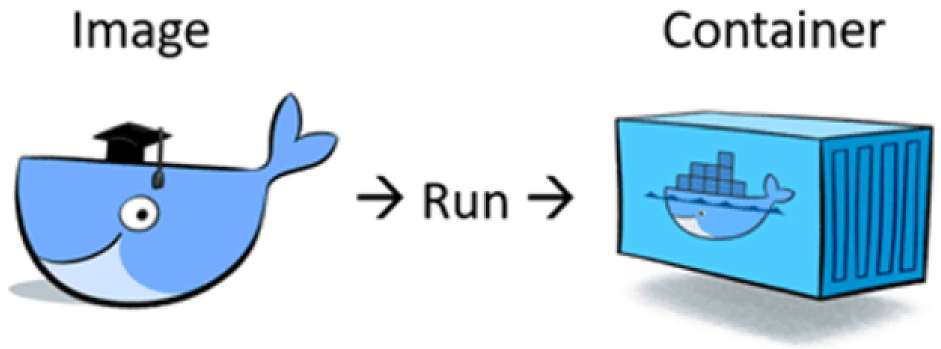
Images and containers

- Image:
  - immutable/unchangeable/read-only file that contains the source code, libraries, dependencies, tools, and other files needed for an application to run
  - static template
- Container:
  - running instance
  - add a writable layer on top of an immutable base image

https://phoenixnap.com/kb/docker-image-vs-container

`docker create` command: create a container layer from an image

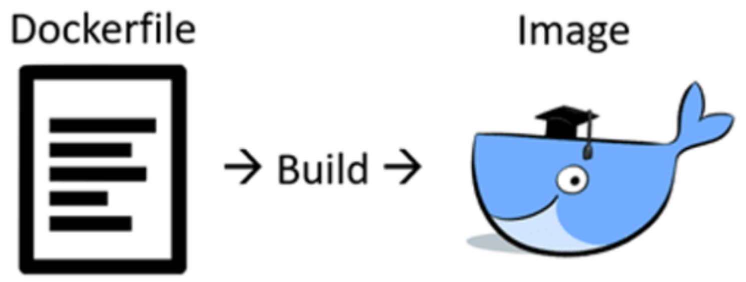


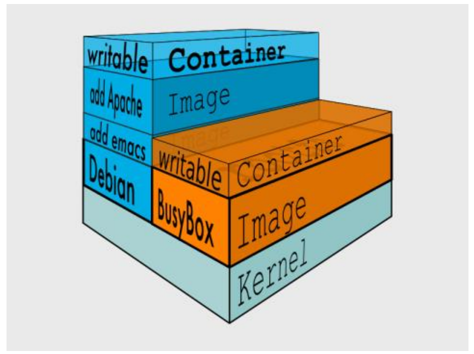


https://www.saagie.com/blog/your-first-steps-into-docker/

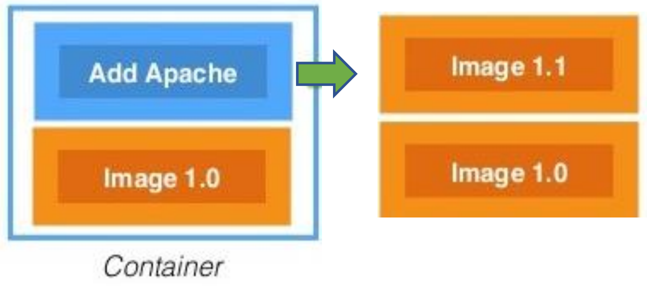
How to create images?

- often based on another image: ex: Ubuntu base + install a JDK
- Dockerfile: script of instructions defining how to build an image, each instruction creates a new layer in the image





- Take a snapshot of a container (the container layer becomes a new image layer)



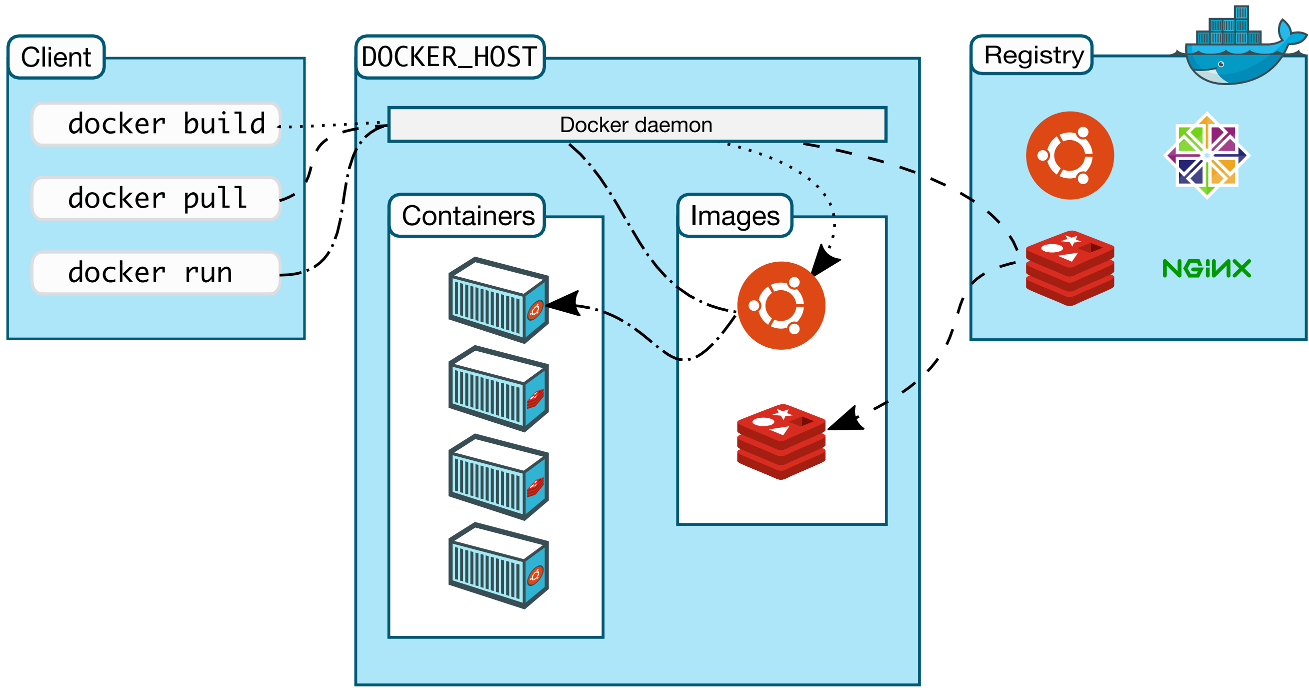
https://www.saagie.com/blog/your-first-steps-into-docker/

http://www.slideshare.net/FabioFerrari31/docker-containers-talk-linux-day-2015

https://linoxide.com/linux-how-to/dockerfile-create-docker-container/

Docker architecture

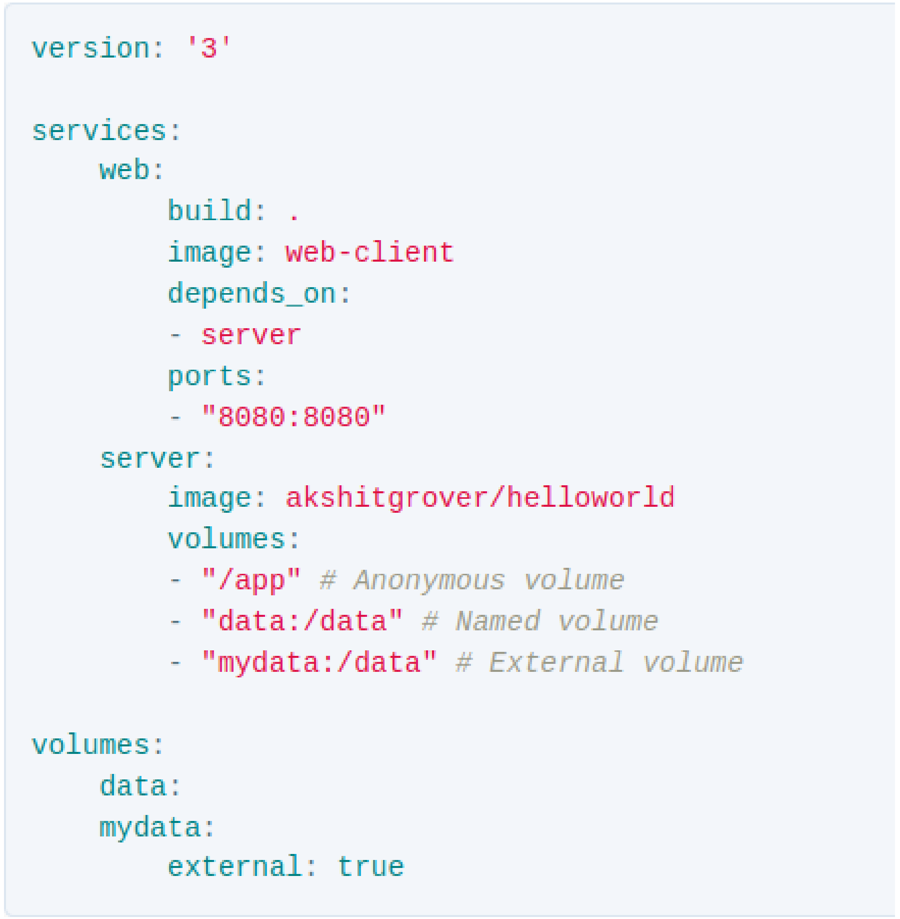
- Docker daemon:
  - persistent background server process that manages network, container, image and data volumes
  - runs on a docker host
- Docker host:
  - provides the environment to execute the application
  - hosts containers, images and data volumes
  - connects containers to external networks
- Docker client:
  - used to interact with a docker daemon (local or remote)
  - direct CLI (Command-Line Interface) commands or scripts using the REST API
- Docker registry:
  - stores many images
  - On-premises, or public cloud (Docker Hub) or vendor-specific clouds (AWS Container Registry, Google Container registry…)



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

Docker-compose

- define multi-container app in YAML file
- use a single command to deploy the entire app



```yaml
version: '3'

services:
    web:
        build: .
        image: web-client
        depends_on:
        - server
        ports:
        - "8080:8080"
    server:
        image: akshitgrover/helloworld
        volumes:
        - "/app" # Anonymous volume
        - "data:/data" # Named volume
        - "mydata:/data" # External volume

volumes:
    data:
    mydata:
        external: true
```

https://dockerlabs.collabnix.com/intermediate/docker-compose/#compose-file-used-in-examples

Tutorial #1: run containers

- get help on CLI commands
- workflow behind a “docker run”
- manage containers and images on your machine
- browse DockerHub
- execute a command within the container as it starts
- open an interactive terminal in the container
- persist data in the writable layer

Tutorial #2: create your own images

- Step 1:
  - Start a container in interactive mode
  - Install software in the container and copy necessary files
  - `docker commit`
- Step 2:
  - Script actions taken in step 1 in a dockerfile
  - `docker build`
- Step 3:
  - Publish to DockerHub repository

Tutorial #3: transfer files between host and container

- At image-creation time : `COPY`
- While a container is running : `docker cp`
- Share a repository
  - Direct mount
  - “Volume” handled by docker

Tutorial #4: networking

- Complex topic!
- A few examples:
  - Isolated container
  - Default bridge
  - User-defined bridge
  - Host network
  - Publish services outside of Docker