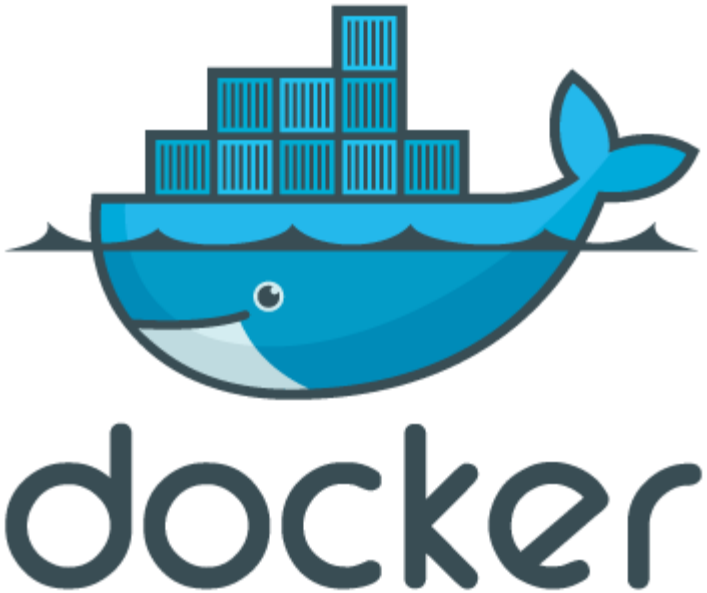
Docker

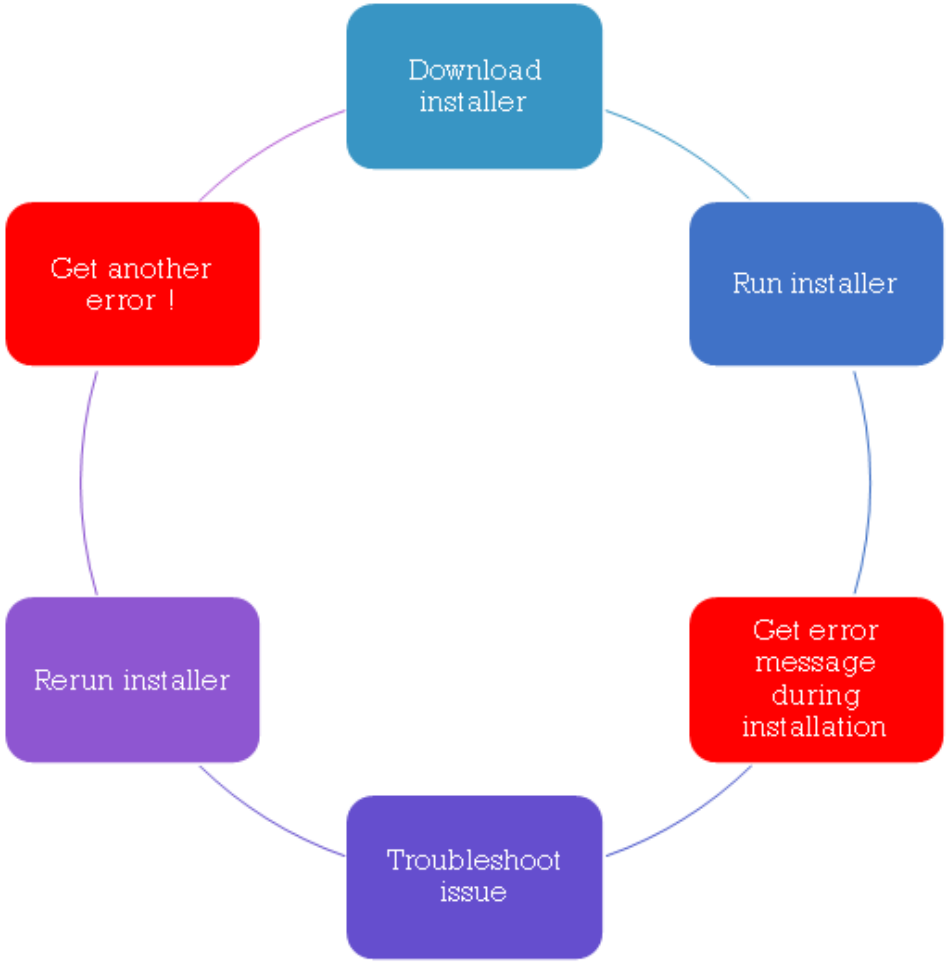
Presented by:
Jospheen Boles



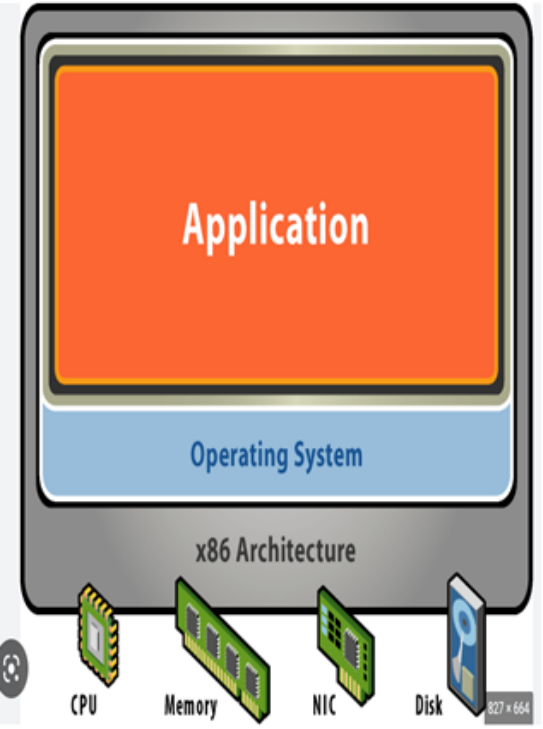
Outline

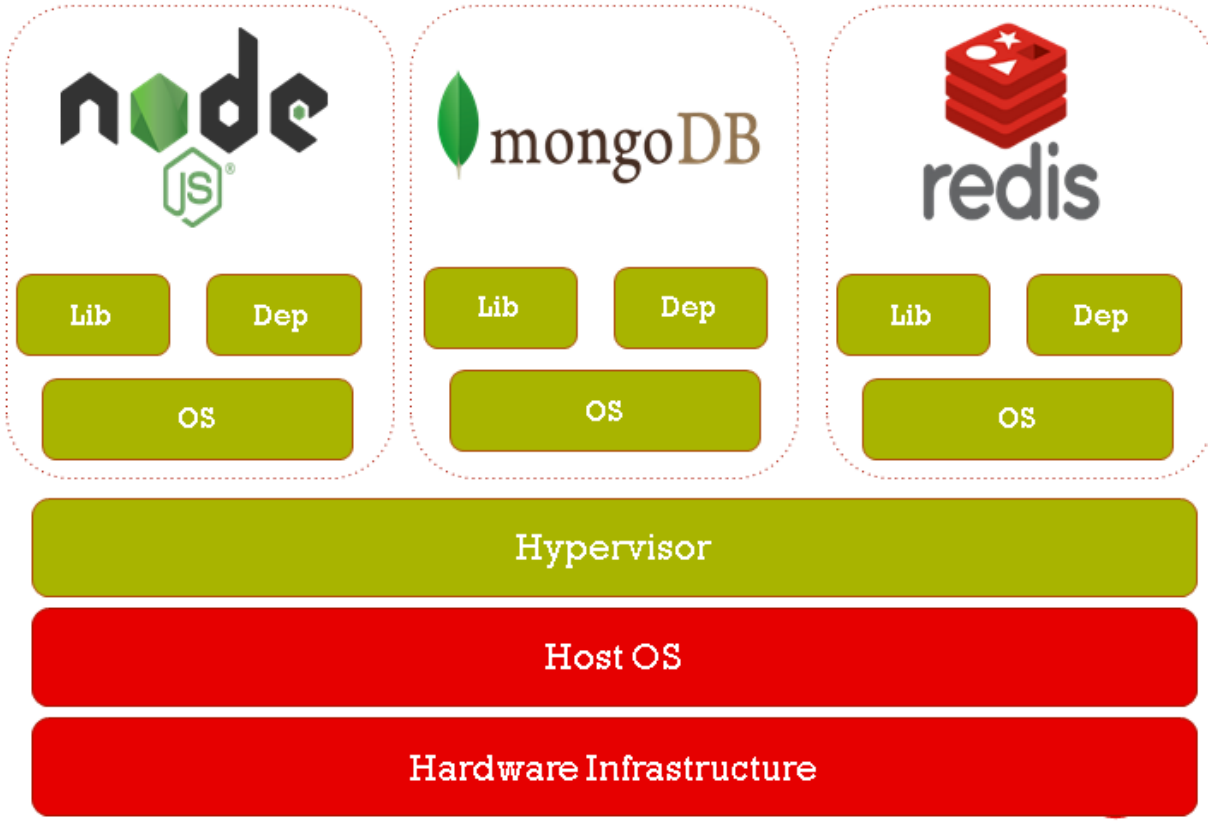
- Containerization VS Virtualization (Container vs VM)
- What's Docker and its architecture?
- Docker images
- Docker Container
- Docker registry
- Docker components (Client, Host and Daemon)
- Common docker commands
- Dockerfile and its instructions

Installing software flow

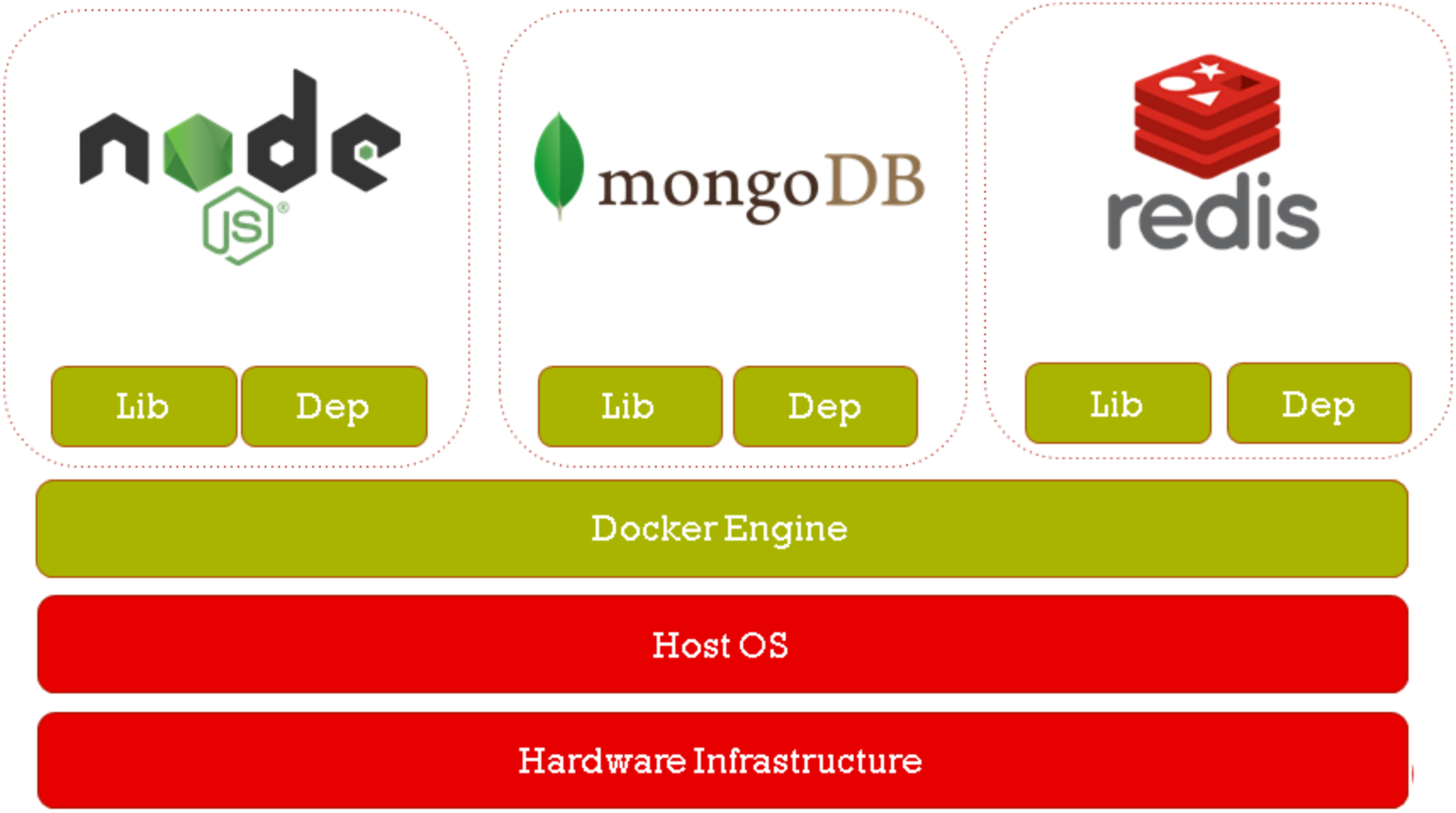


Virtualization

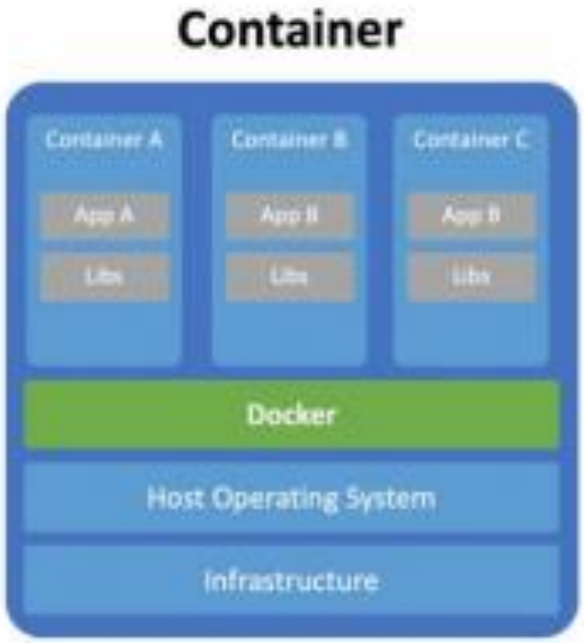


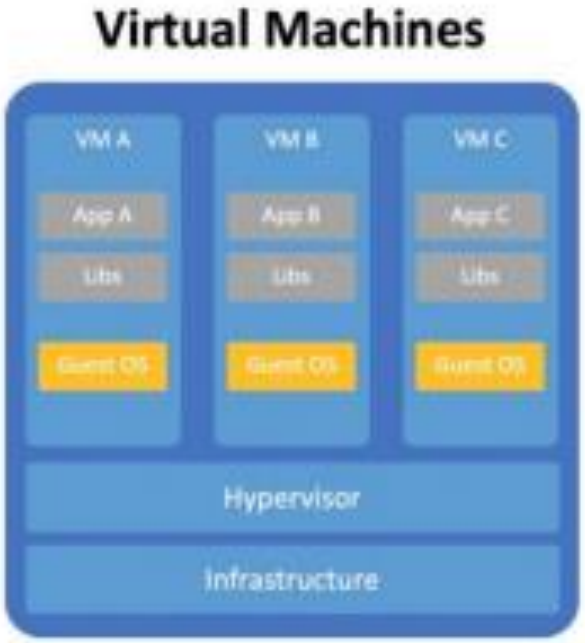


Containerization



Container VS Virtual Machine

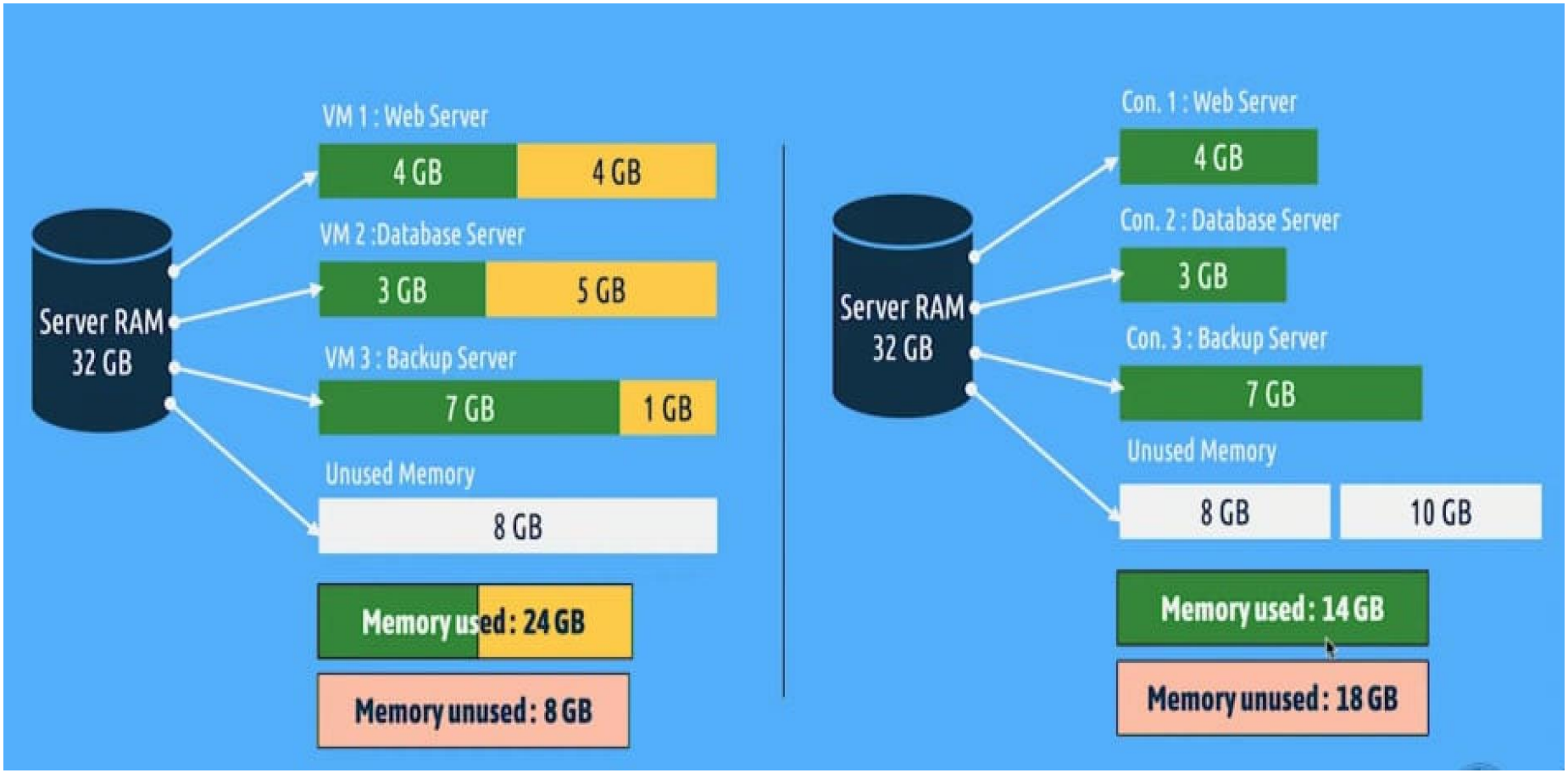




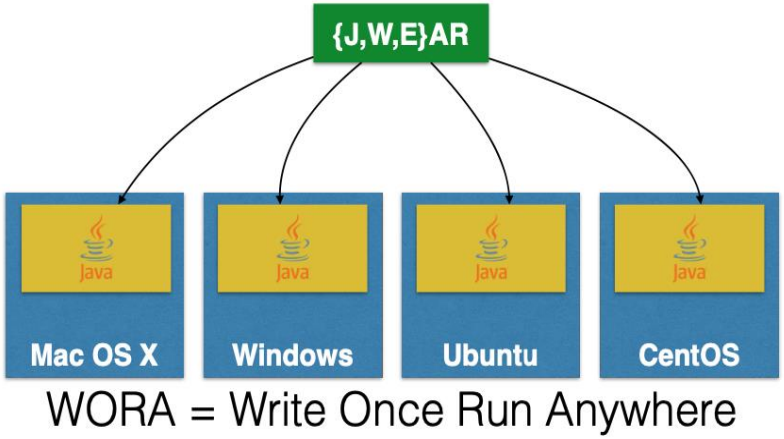
Why use Docker ?

Container advantages.

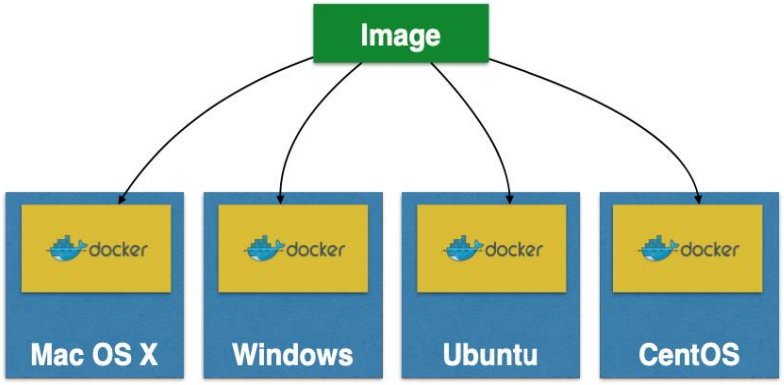
Using


VM 1 : Web Server
4 GB
4 GB
VM 2 :Database Server
3 GB
5 GB
VM 3 : Backup Server
7 GB
1 GB
Unused Memory
8 GB
Server RAM
32 GB
Memory used : 24 GB
Memory unused : 8 GB
Con. 1 : Web Server
4 GB
Con. 2 : Database Server
3 GB
Con. 3 : Backup Server
7 GB
Unused Memory
8 GB
10 GB
Server RAM
32 GB
Memory used : 14 GB
Memory unused : 18 GB

Why use Docker ?



WORA = Write Once Run Anywhere



PODA = Package Once Deploy Anywhere

Why use Docker !

Docker makes it really easy to install and run software without worrying about setup or dependencies.

Short setup time.

Different Dev/Test/Prod environments.

What is Docker?

Docker

- Docker runs on client-server architecture. The client communicates with the docker service (daemon), which does the work of compiling, running, and distributing the containers. The registry is in charge of storing the docker images.

Container Engines

Docker Architecture

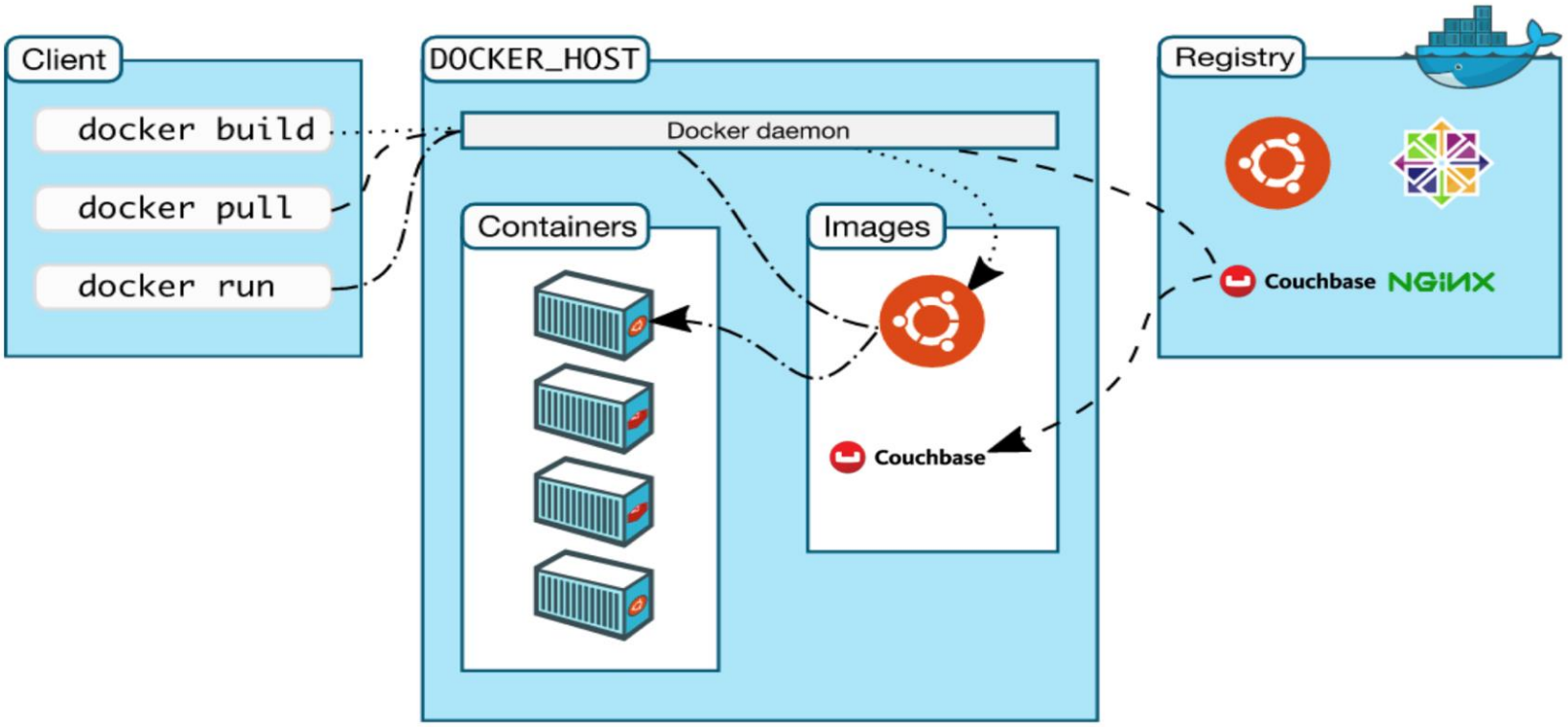


Image VS Container

Image

- Single file with all the deps and config required to run a program

Container

- Instance of an image. Runs a program.

Inside Docker.

- **Images**

Read-only template with the instructions for creating a container, usually created from another image with additional customization.

- **Layers**

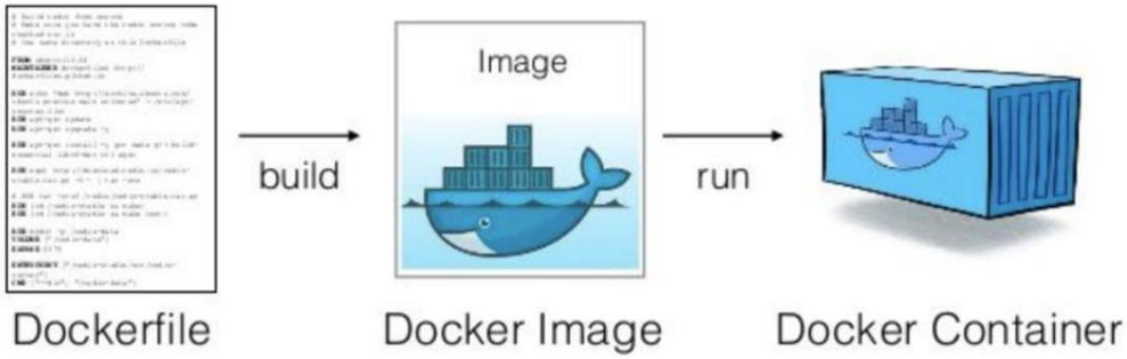
In an image, a layer is the modification of the image, represented by an instruction in the Dockerfile. Layers are applied in sequence to the base image to create the final image.

- **Containers**

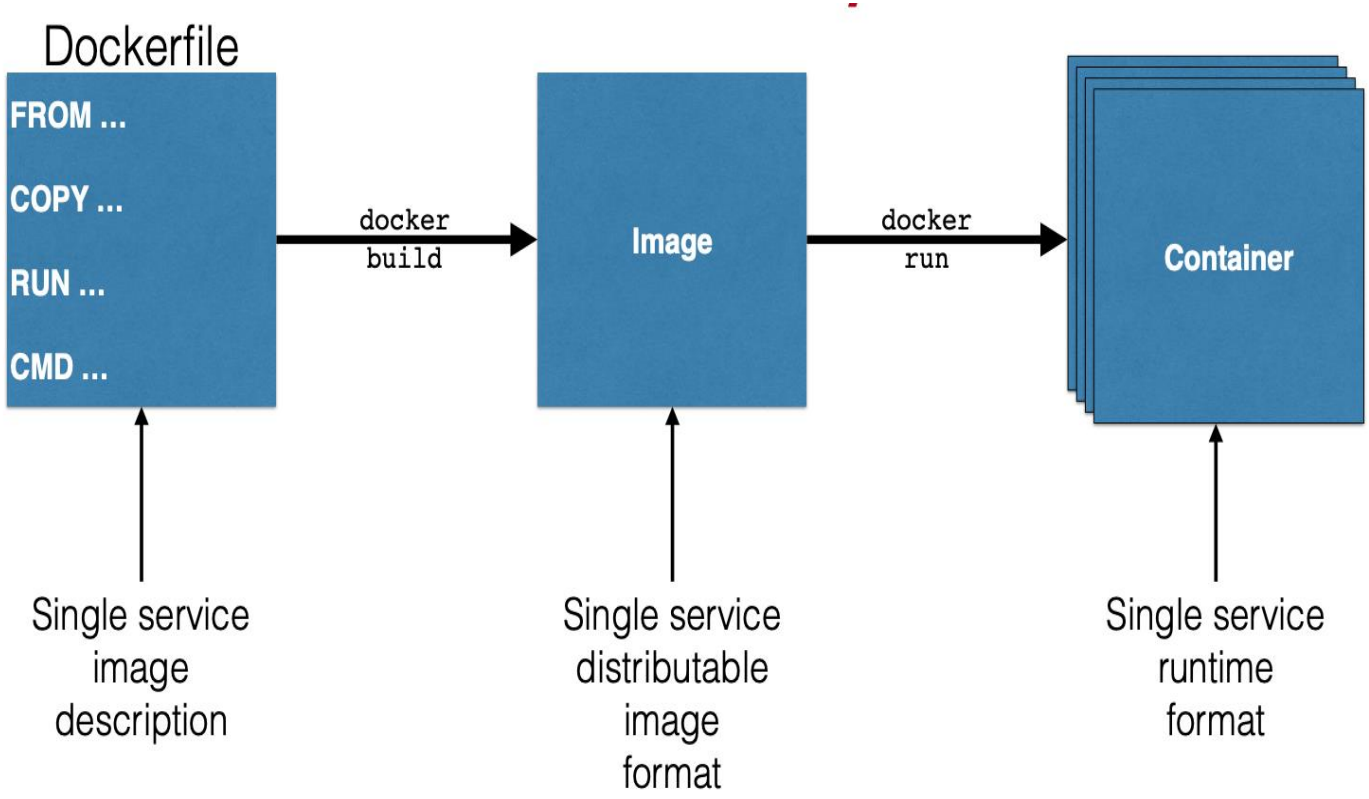
It is an executable instance of an image, which is relatively isolated from other containers and from its host.

Dockerfile

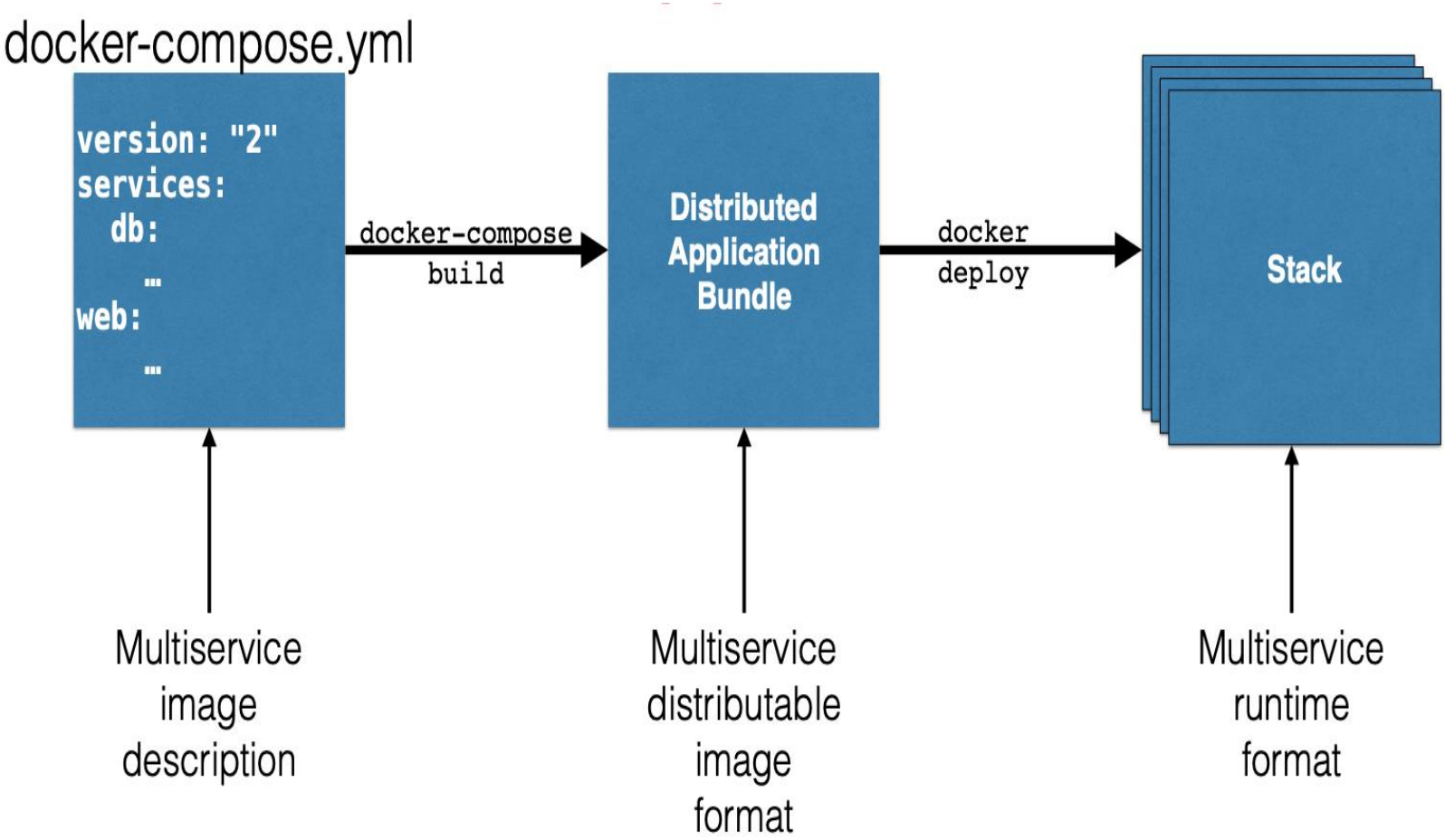
- A Dockerfile is a text file which contains a series of commands or instructions.
- These instructions are executed in the order in which they are written.
- Execution of these instructions takes place on a base image.
- On building the Dockerfile, the successive actions form a new image from the base parent image.
- Will share later the arguments like (FROM, COPY, ADD, RUN, WORKDIR, CMD, ENTRYPOINT)
- Multistage Dockerfile.



Docker Lifecycle for single service.



Docker Lifecycle for Multi service.



Docker for Mac/Windows

- Native application and UI
- Auto update capability
- No additional software required
- Download: https://docs.docker.com/desktop/windows/install/.
- Requires
  - Yosemite 10.10+
  - Windows 10 64-bit (Hyper-V)

Lab 1

- Install docker
- Sign up your Docker hub account
- Run your first container
- Delete all containers on the system.