Docker

Ahmed Abdelnabi

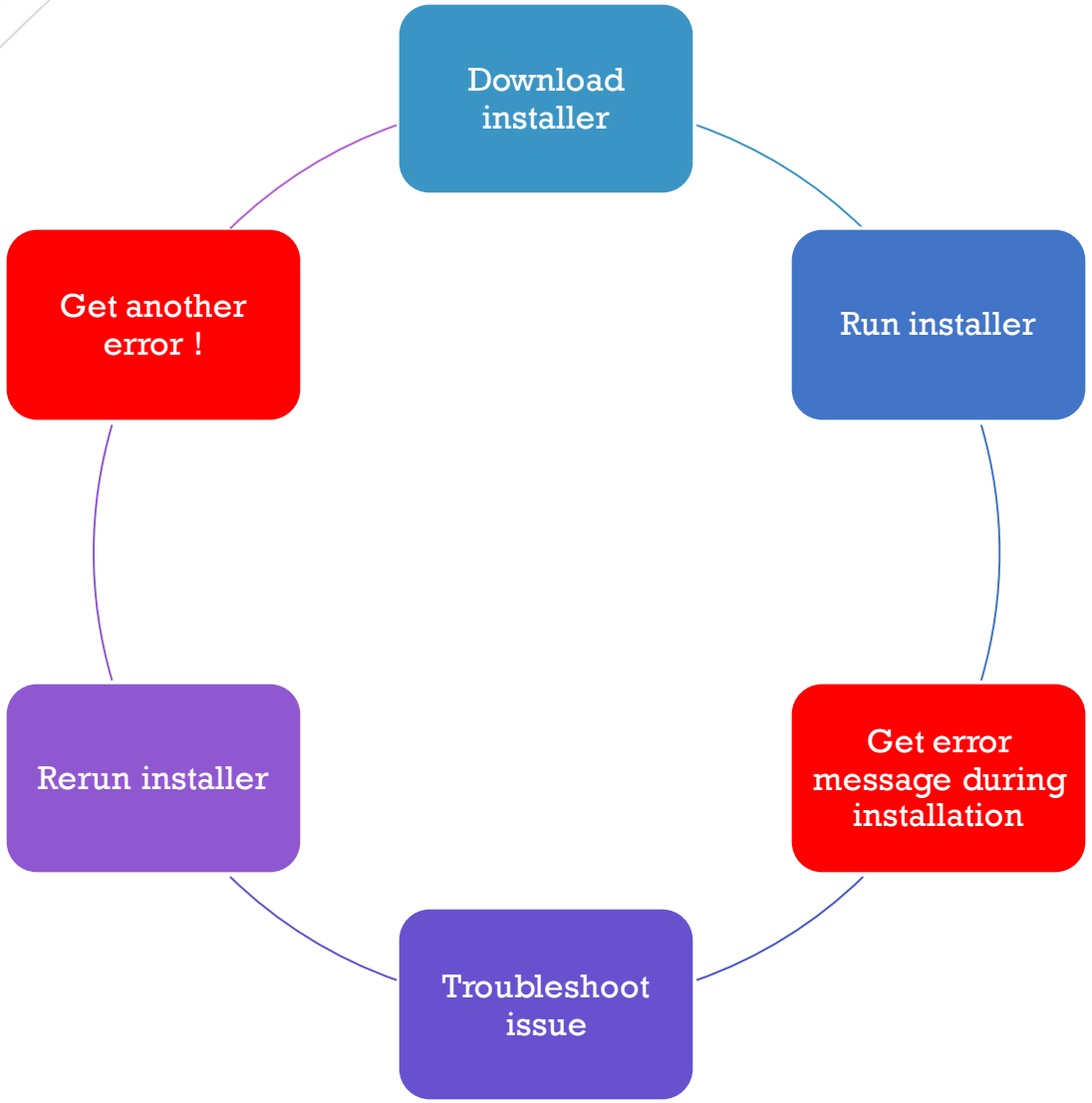
Agenda

- Introduction to Docker
- Docker Image vs Container
- Docker Architecture
- Virtualization vs Containerization
- Container Lifecycle
- Docker Client Commands
- Docker Volumes
- Lab

Introduction to Docker

WHAT IS DOCKER ?

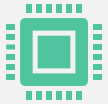
WHY USE DOCKER ?



Installing software flow

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

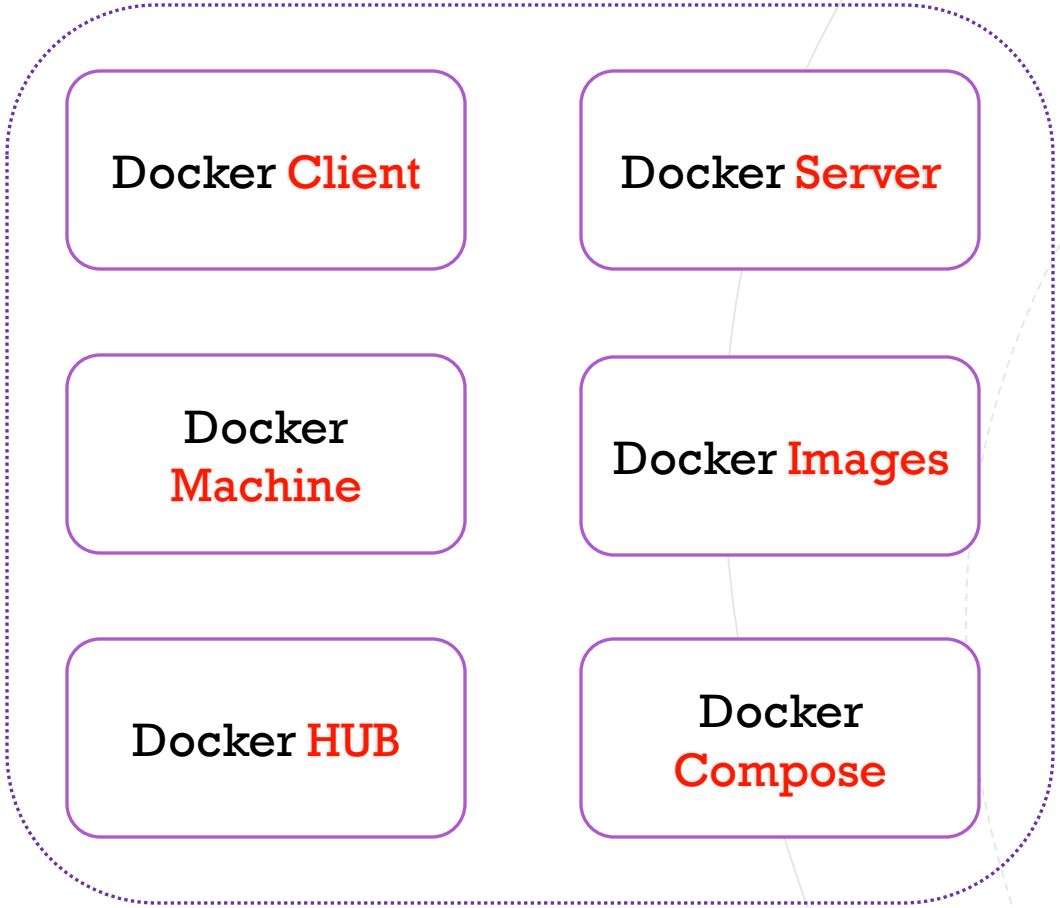
Short setup time.



Different Versions.

Why use Docker !

What is Docker ?



- Docker is a platform or ecosystem around creating and running something called containers.
- Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

Docker Image vs Container

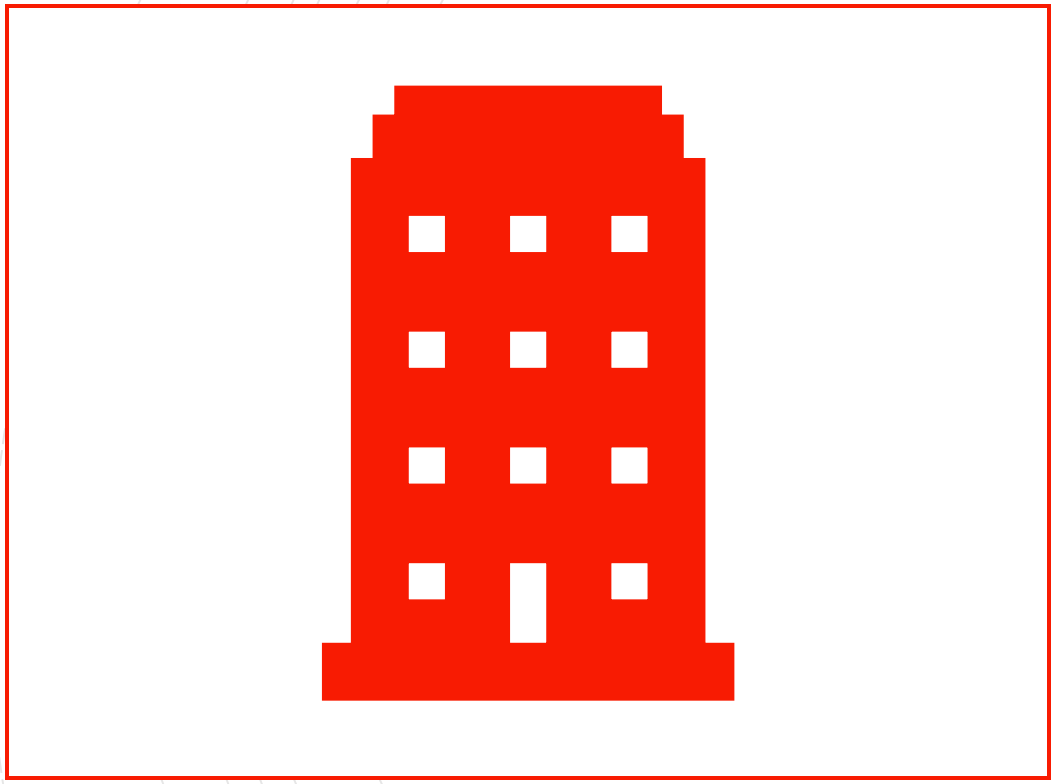
Image

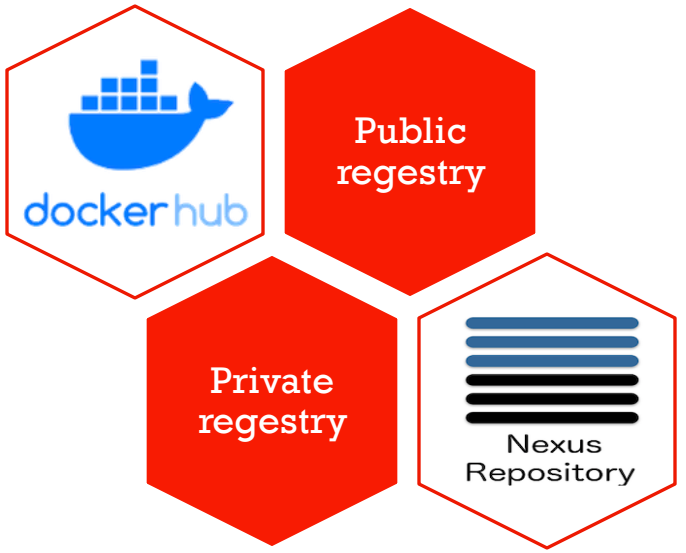
- Single file with all the dependences and config required to run a program

Container

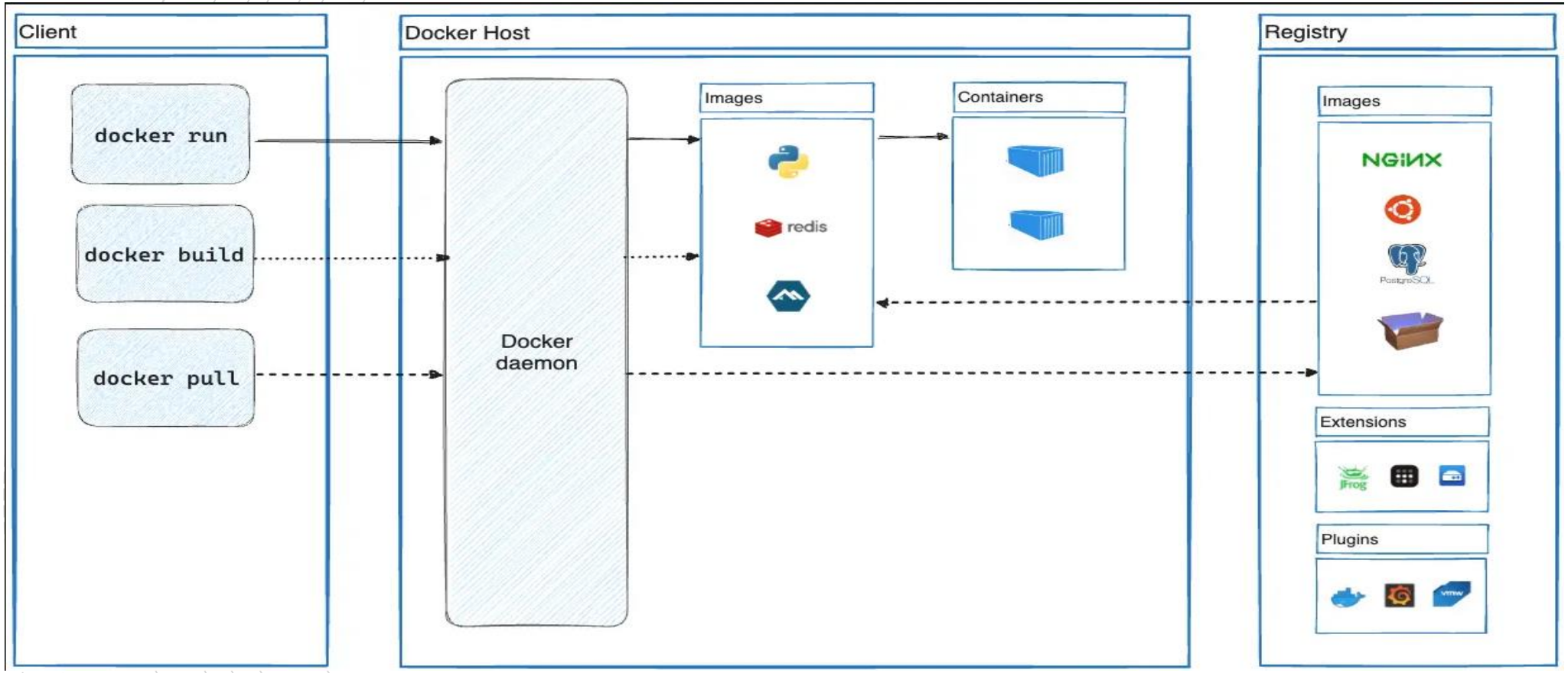
- Instance of an image.
- Runs a program.

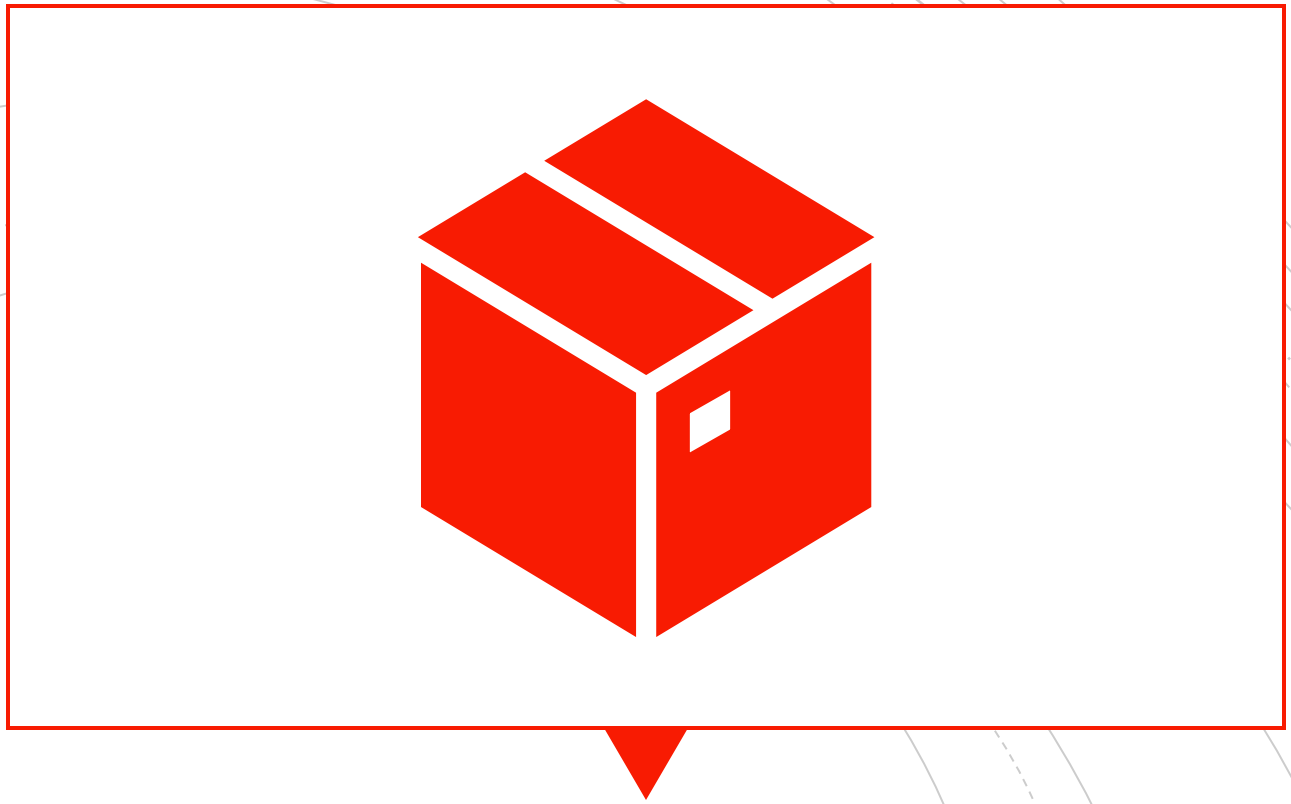
Types of Docker regestries





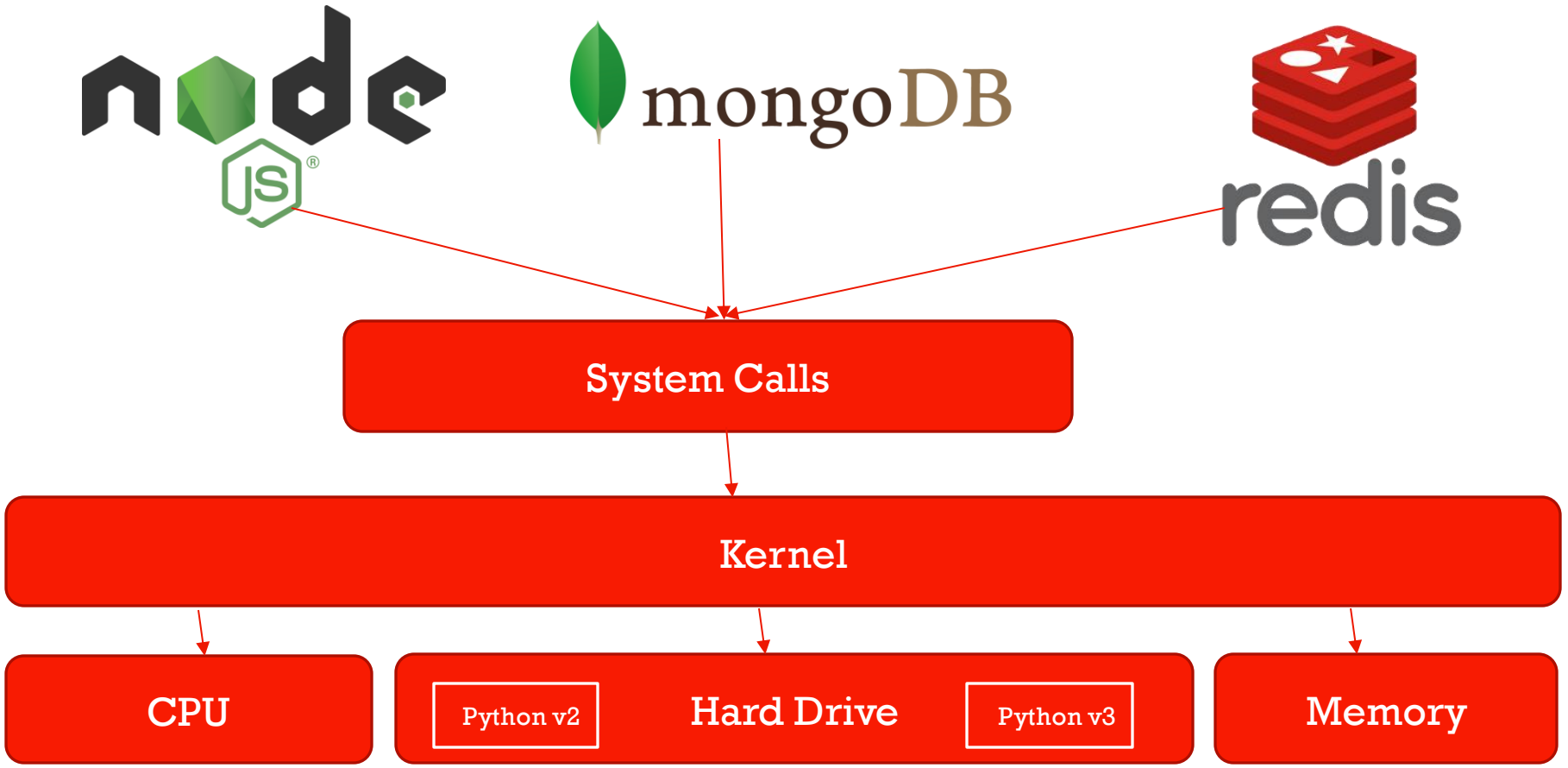
Docker Architecture



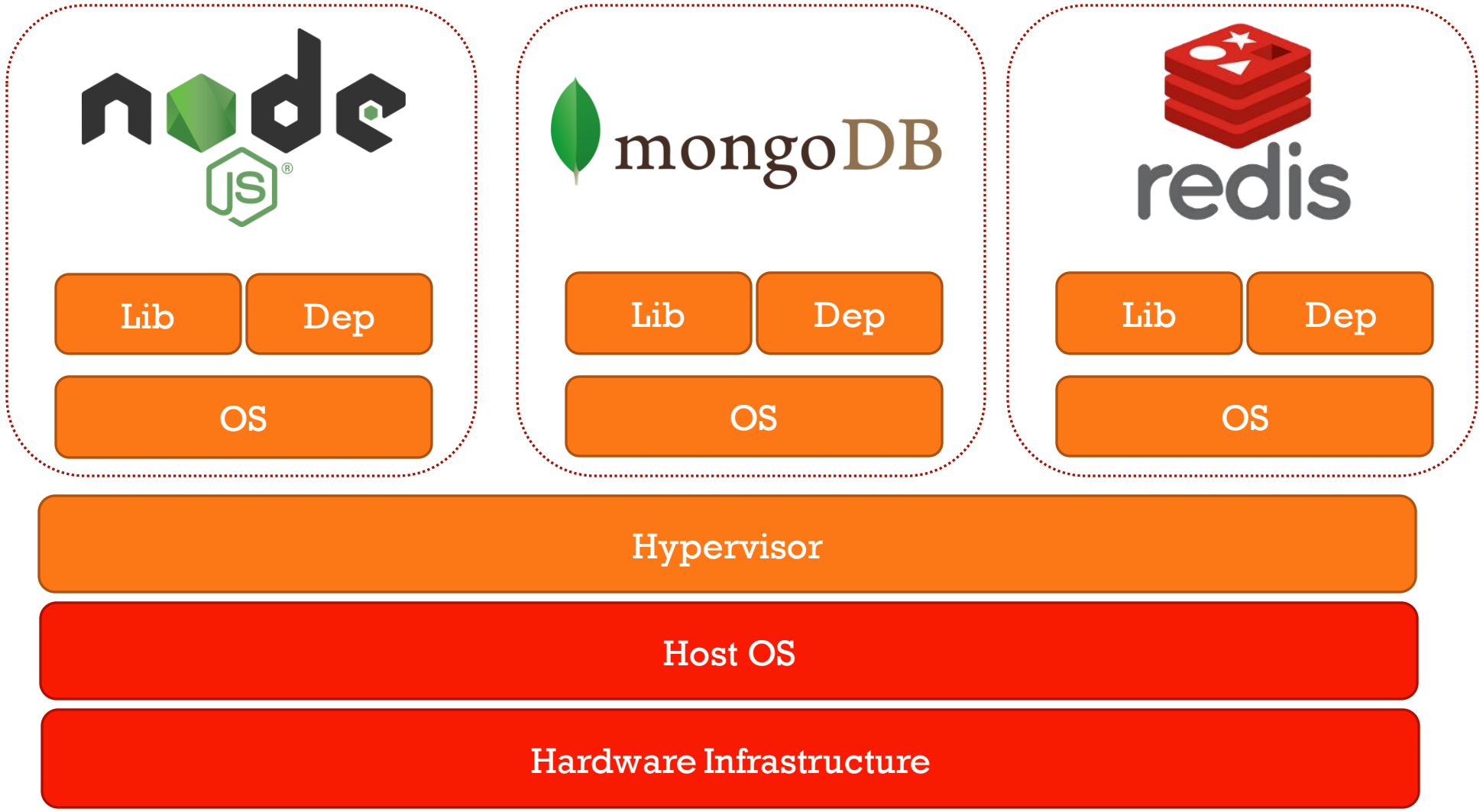


Virtualization VS Containerization

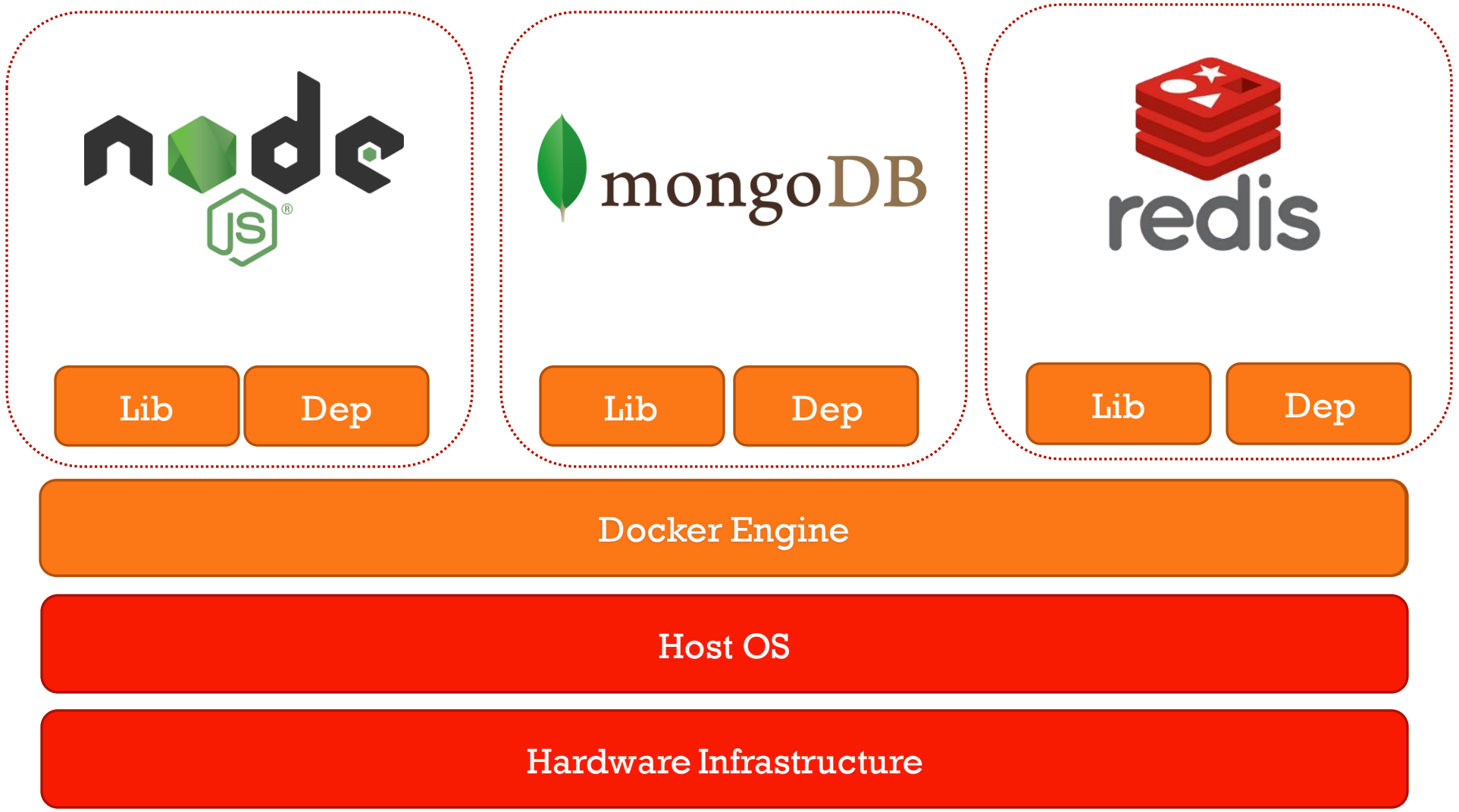
How Process Runs



Virtualization



Containerization



Containerization vs Virtualization

UTILIZATION

SIZE

BOOT UP

Break

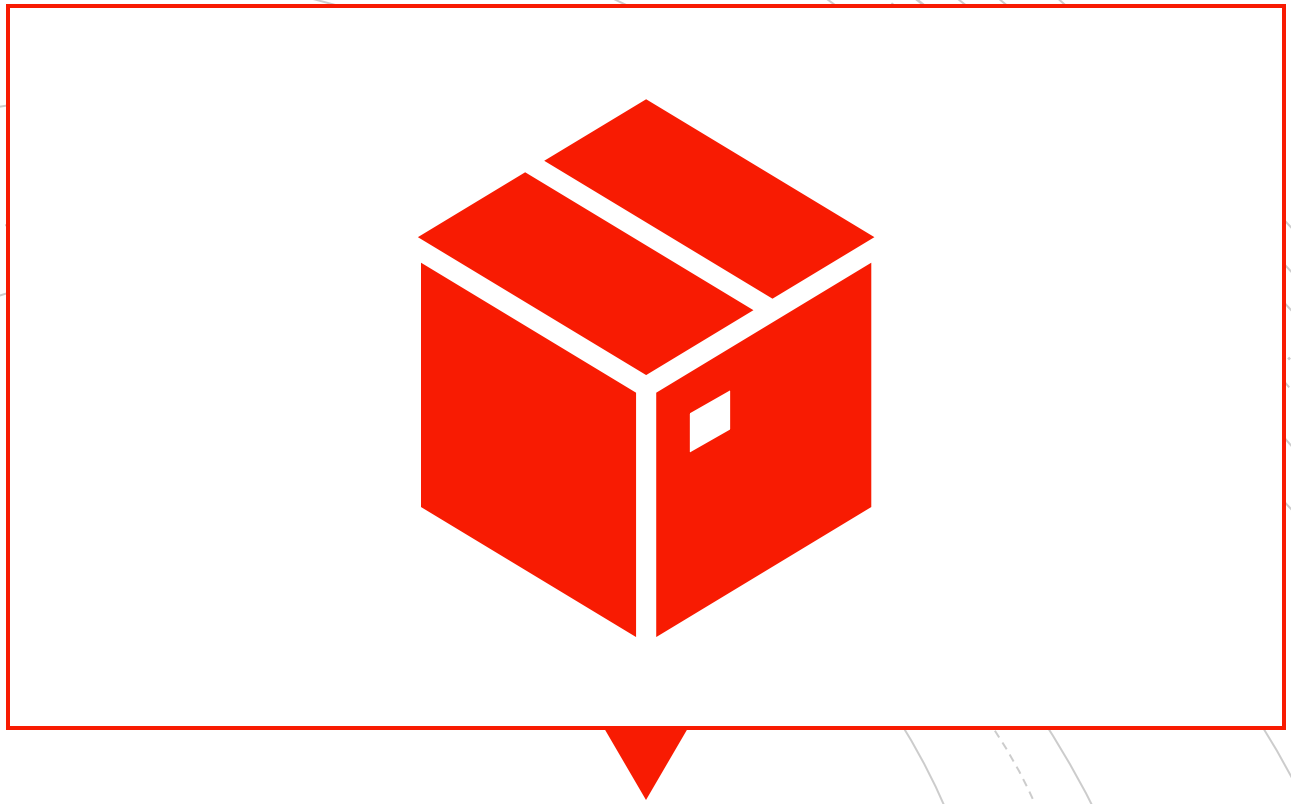
Container Lifecycle

- docker run:
  - Create: take file system snapshot and add as a process ready to run in hard drive.
  - Start: Run start up command
  - Exited: finished its job and exit.
- docker stop: terminate single but wait
- docker kill: force terminate
- docker prune: clear docker caches, images and stopped containers.

Docker Client Commands

Docker Basic Commands

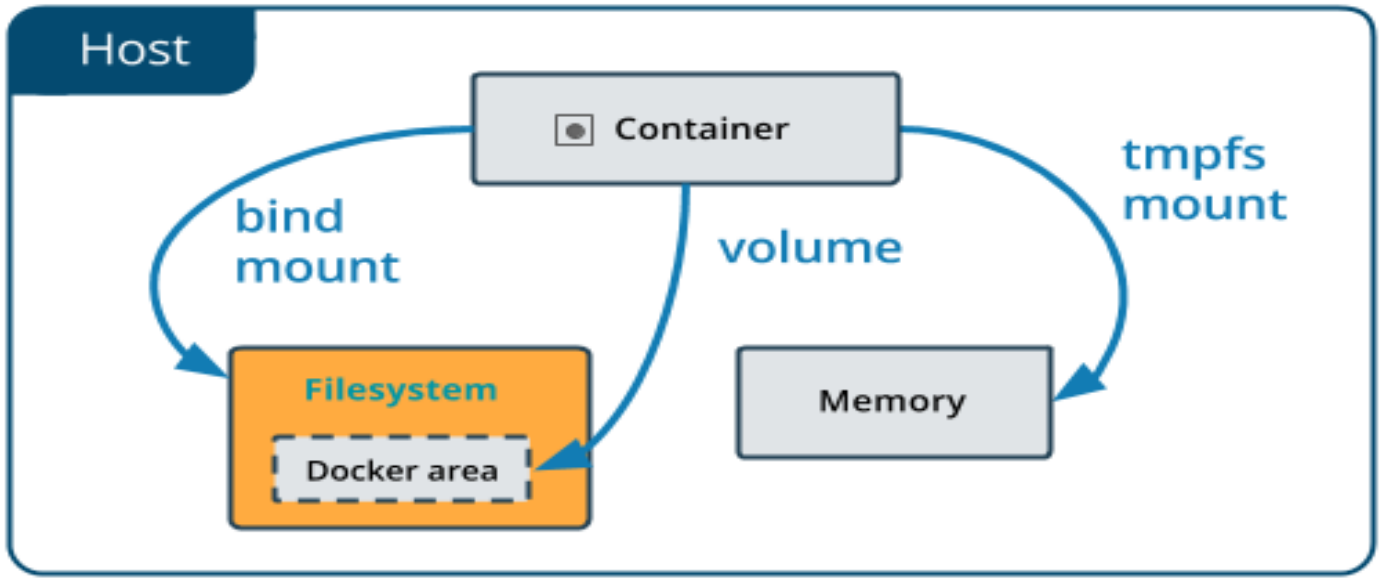
Run Container	•$ docker run <image-name>
Overriding default command	•$ docker run <image-name> <command!>
Attached and Detached	•$ docker run –d <image-name> •$ docker attach <container ID>
Interactive mode	•$ docker run -i <image-name>
Port mapping	•$ docker run –p host:container <image-name>
Volume mapping	•$ docker run –v host:container <image-name>
Inspect container	•$ docker inspect <container-name>
Stop container	• $ docker stop CONTAINER
List all images	•$ docker images -a
List all containers	•$ docker ps -a



Docker Volume

Docker Volumes

- Bind Mount:
  - a file or directory on the *host machine* is mounted into a container
  - The file or directory is referenced by its absolute path on the host machine
- Volume:
  - a new directory is created within Docker's storage directory on the host machine, and Docker manages that directory's contents.



Lab

Problem 1

- Run the container hello-world
- Check the container status
- Start the stopped container
- Remove the container
- Remove the image

Problem 2

- Run container centos or ubuntu in an interactive mode
  - Run the following command in the container "echo docker "
  - touch a file named hello-docker
- Stop the container and remove it.
  - What is your comment about the file hello-docker?
- Remove all stopped containers

Problem 3

- Run a container nginx with name mynginx and attach a volume to the container
  - Volume for containing static html file
- Remove the container
- Run a new container with the following:
  - Attach the volume that was attached to the previous container
  - Map port 80 to port 9898 on you host machine
  - Access the html files from your browser

Problem 4

- Run a container using nginx image "without attaching any volumes"
- Add html static files to the container and make sure they are accessible
- Commit the container with image name "my-nginx"
- Run new container with the new image my-nginx
  - What will happen to the static file?

Problem 5

- Create a volume called mysql_data, then deploy a MySQL database called app-database. Use the mysql latest image, and use the -e flag to set MYSQL_ROOT_PASSWORD to P4sSw0rd0!.M ount the mysql_data volume to /var/lib/mysql. The container should run in the background.

Thanks for attention