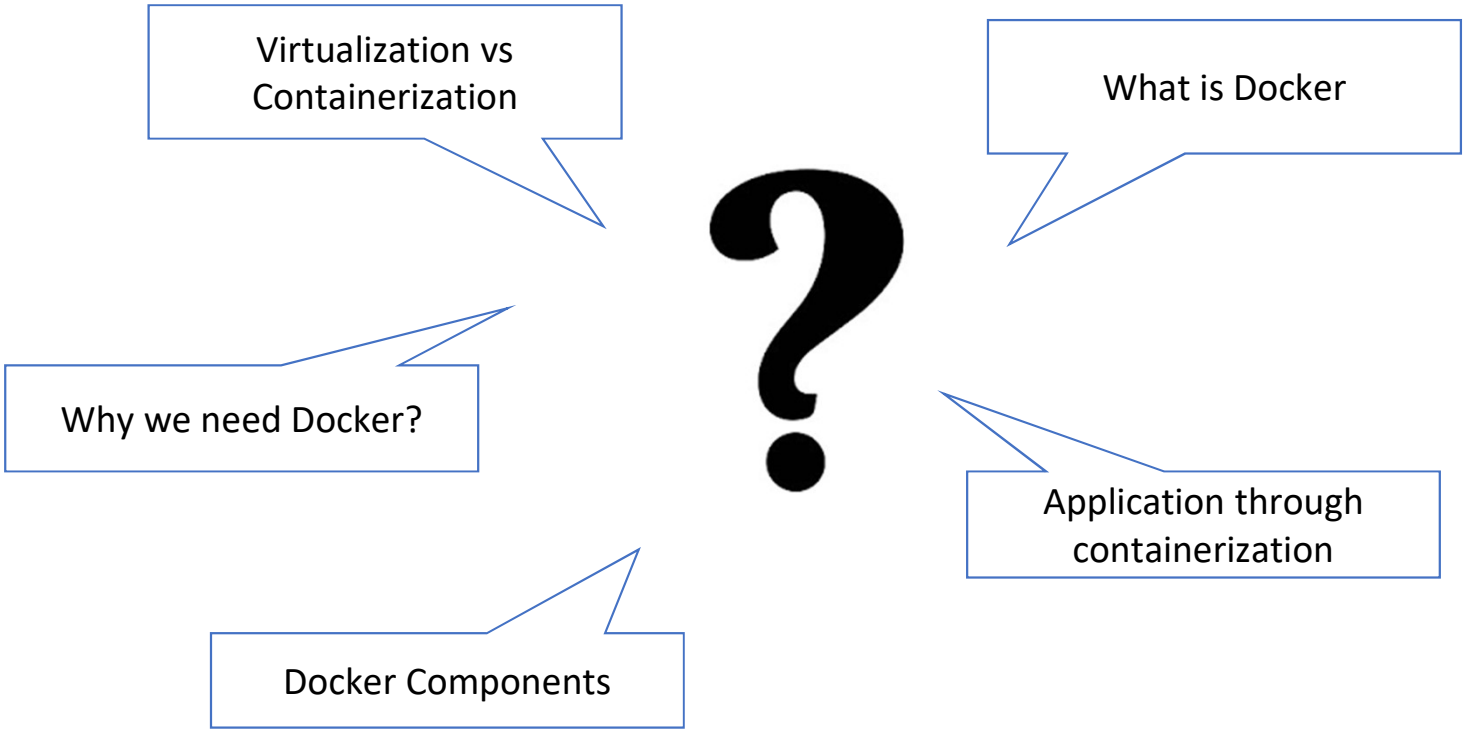
AGENDA


Virtualization vs Containerization
What is Docker
Why we need Docker?
Application through containerization
Docker Components

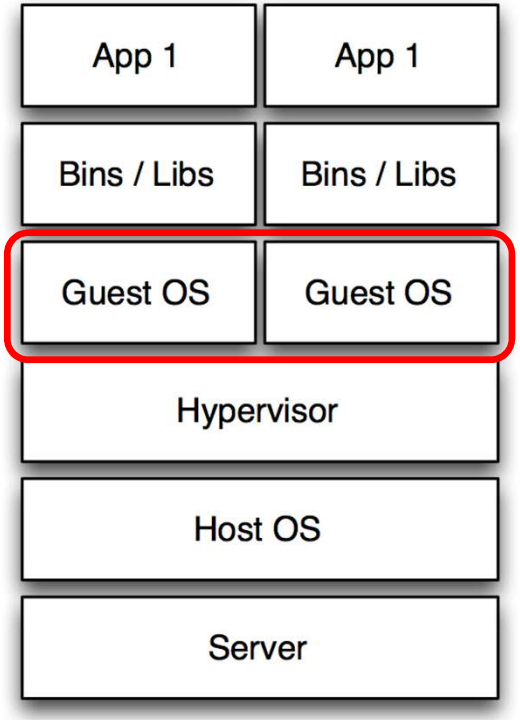

THE DEPENDENCIES WILL KILL YOU


docker

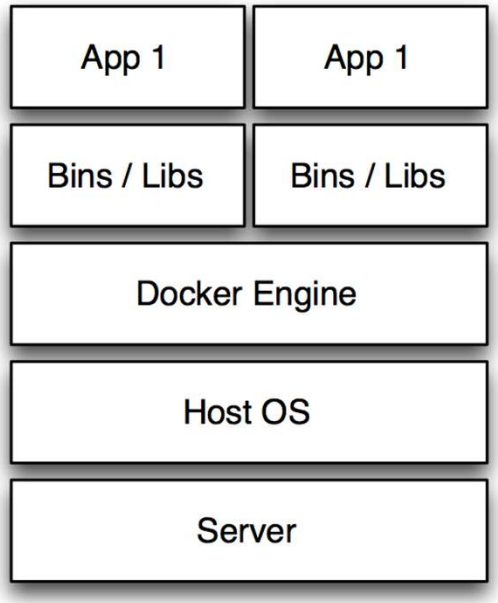
Docker: Containerization for Software

VIRTUALIZATION
vs
CONTAINERIZATION

VM vs. Docker (Containers)



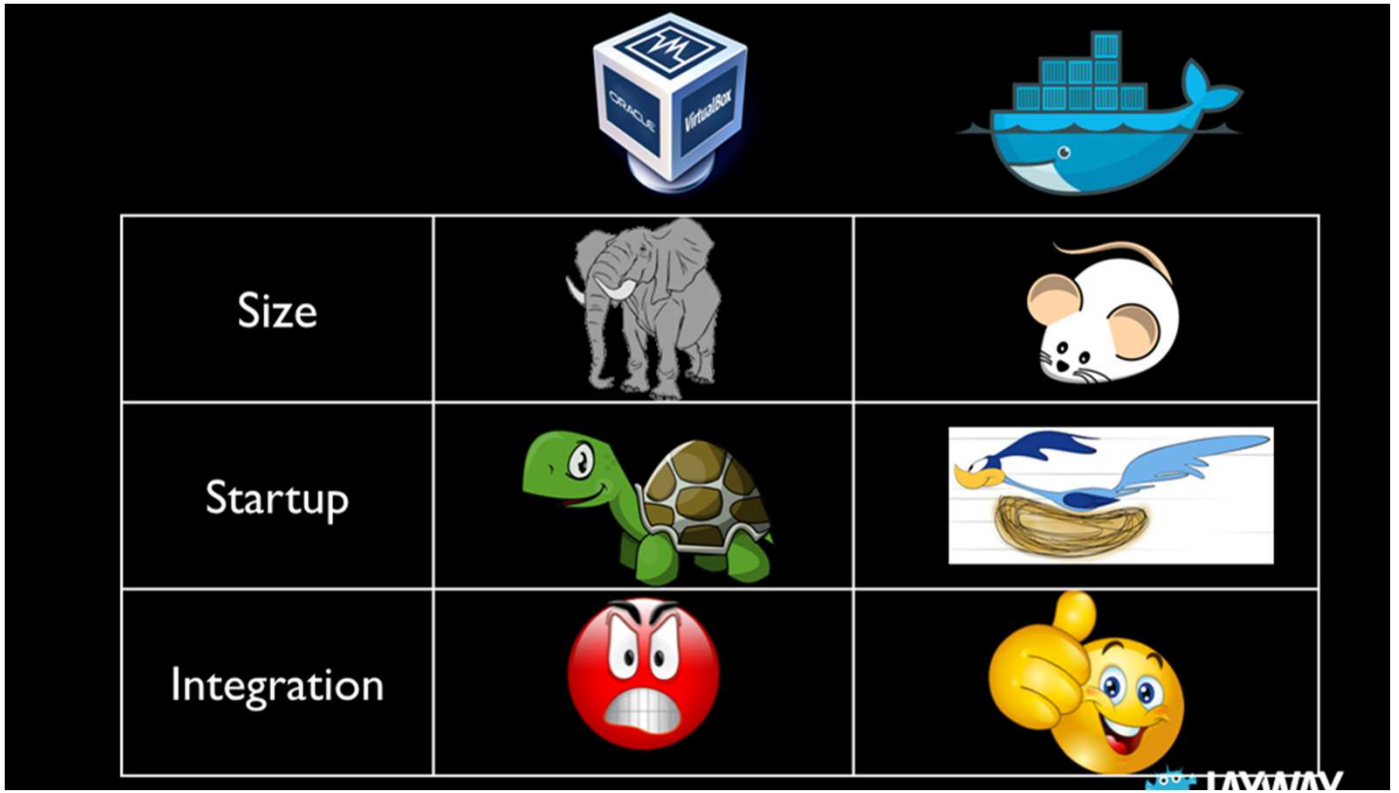
Virtual Machines



Docker

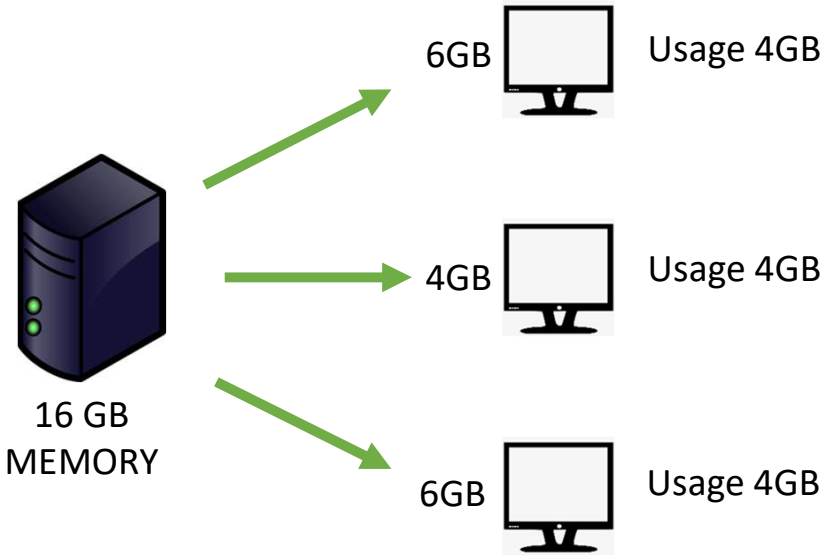
Docker Engine
Docker engine is the layer on which Docker runs.
It's a lightweight runtime and tooling that manages containers, builds, and more.

VM vs Docker



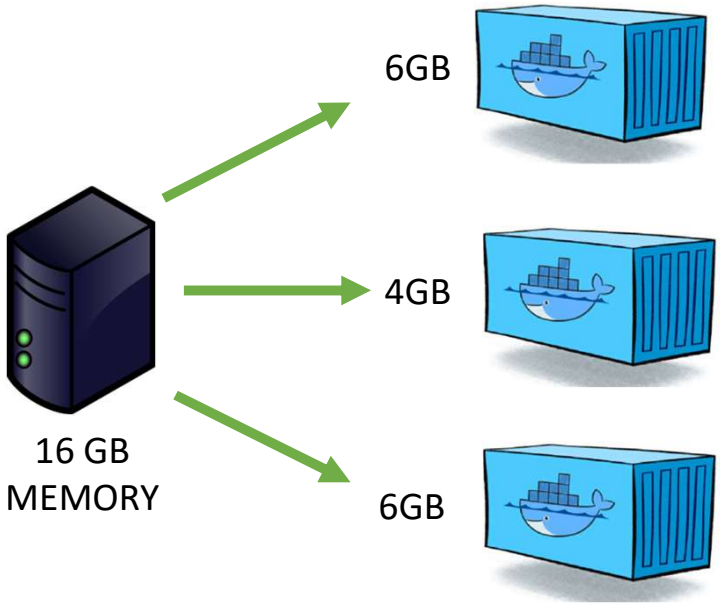
VM vs Docker

In case of VM



4 GB MEMORY remain unused and cannot be allocated to another VM

In case of Docker



4 GB MEMORY remain unused and can be allocated to another container as containers share resources

VM vs Docker

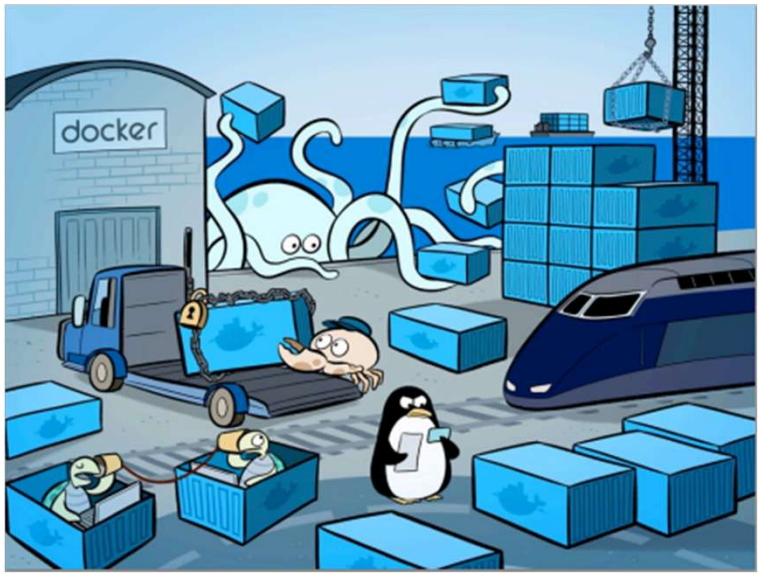
Deploying Application through installer requires multiple VMs

Integration in Docker is cheaper, faster and scalable

Costly due to Infrastructure Requirements

WHAT IS DOCKER?

Docker



“Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications”

So why Docker?

- Containers are far from new;
  - Google has been using their own container technology for years.
  - Others Linux container technologies include
    - Solaris Zones,
    - BSD jails, and
    - LXC, which have been around for many years.
- Docker is an open-source project based on Linux containers. It uses Linux Kernel features.

Docker Benefits

1. Local development environments can be set up that are exact replicas of a live environment/server.
2. It simplifies collaboration by allowing anyone to work on the same project with the same settings, irrespective of the local host environment.
3. Multiple development environments can be run from the same host each one having different configurations, operating systems, and software.
4. Projects can be tested on different servers.
5. It gives you instant application portability. Build, ship, and run any application as a portable container that can run almost anywhere.

Why Docker?

- **Ease of use.** It allows anyone to package an application on their laptop, which in turn can run unmodified anywhere
  - The mantra is: “build once, run anywhere.”
- **Speed.** Docker containers are very lightweight and fast. Since containers are just sandboxed environments running on the kernel, they take up fewer resources. You can create and run a Docker container in seconds, compared to VMs which might take longer because they have to boot up a full virtual operating system every time.
- **Docker Hub.** Docker users also benefit from the increasingly rich ecosystem of Docker Hub, which you can think of as an “app store for Docker images.” Docker Hub has tens of thousands of public images created by the community that are readily available for use.
- **Modularity and Scalability.** Docker makes it easy to break out your application’s functionality into individual containers. With Docker, it’s become easier to link containers together to create your application, making it easy to scale or update components independently in the future.

How Docker works for you

Powerful, easy to use, delivers a great user experience so you can focus on what you love – writing great code

DEV

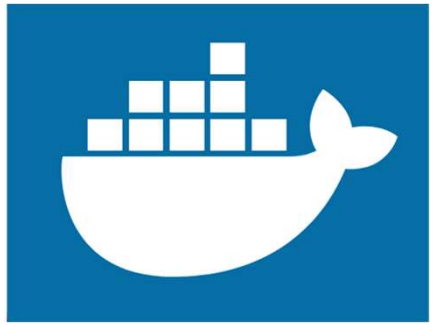
OPS

Deploy and run applications is a way that makes best sense for your customers and business

BUSINESS LEADERS

Drive your digital transformation by accelerating new innovation and dramatically driving down existing IT costs

Docker Components

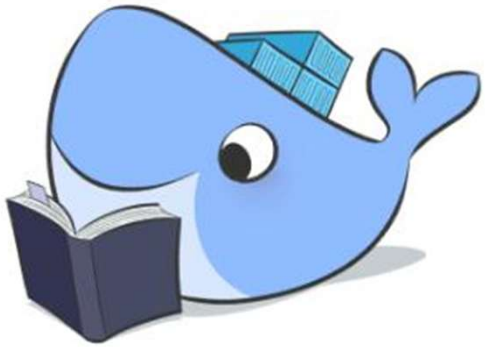


- Image

  An executable package that includes everything needed to run an application--the code, a runtime, libraries, environment variables, and configuration files.
- Container

  Running instance of an image

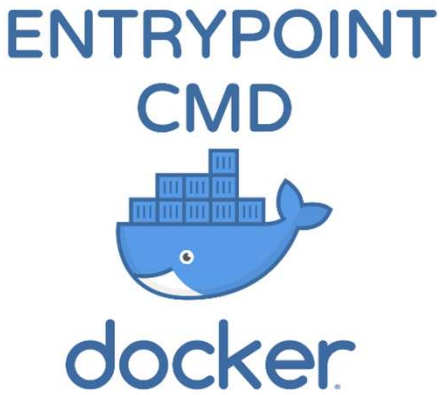
Docker Commands



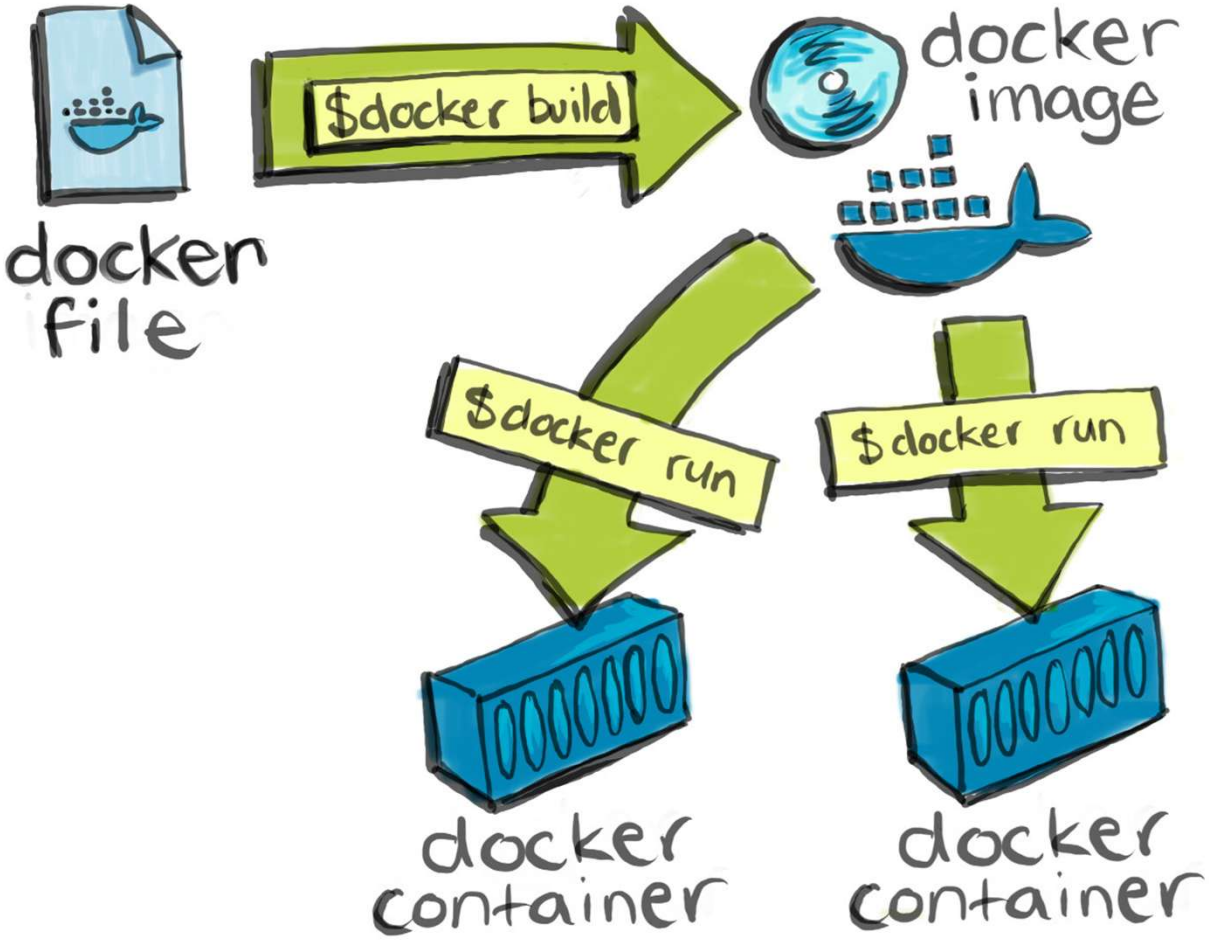
- Check version
  docker --version
- Create an image
  docker build . –t tagname
- Show images
  docker images
  docker image ls
- Remove image
  docker rmi imageID

- Run container
  docker run imageName
- Start container
  docker start containerId
- Stop container
  docker stop containerId
- List container
  docker ps -a
- Remove container
  docker rm containerId

Dockerfile



- A text document that contains all the commands a user could call on the command line to assemble an image.
- Executes several command-line instructions in succession.

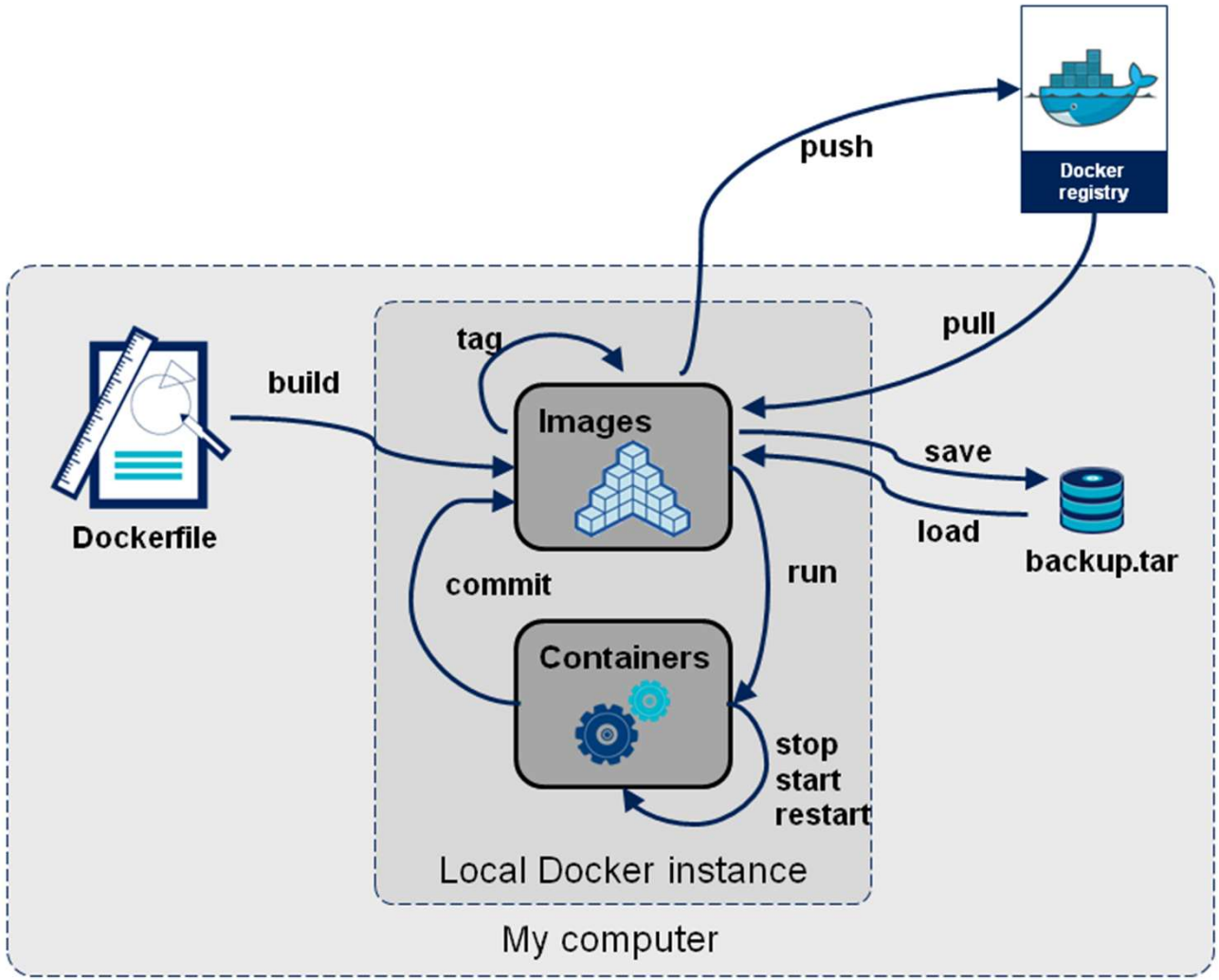

docker file
$docker build
docker image
$docker run
$docker run
docker container
docker container

Docker Compose

- Compose is a tool for defining and running multi-container Docker applications
- Uses a yaml file to configure application's services
  docker-compose.yml
- A single command creates and starts all the services
  docker-compose up

Docker Registry

- A registry is a storage and content delivery system
- A stateless, highly scalable server side application that stores and lets you distribute Docker images.
- Users interact with a registry by using Docker push and pull commands.


Docker registry
push
pull
tag
build
Images
save
load
backup.tar
Dockerfile
commit
run
Containers
stop
start
restart
Local Docker instance
My computer