Docker

Presentation of Docker

What is docker? Why is it so popular?

- It's a platform for **building**, **running** and **shipping application** in a ***consistent manner***
- It will **optimize** the platform in **different environment**.
  - Environment variables can be different in different devices
  - One or more file can be missing on different machines
  - Software can mismatch
- It can **easily package applications** with every necessary files.
- Docker can **automatically download and install all dependencies** that is contained within container.
- Docker can **remove all application packages at once.**
  - Docker-compose up / Docker-compose down --rmi all

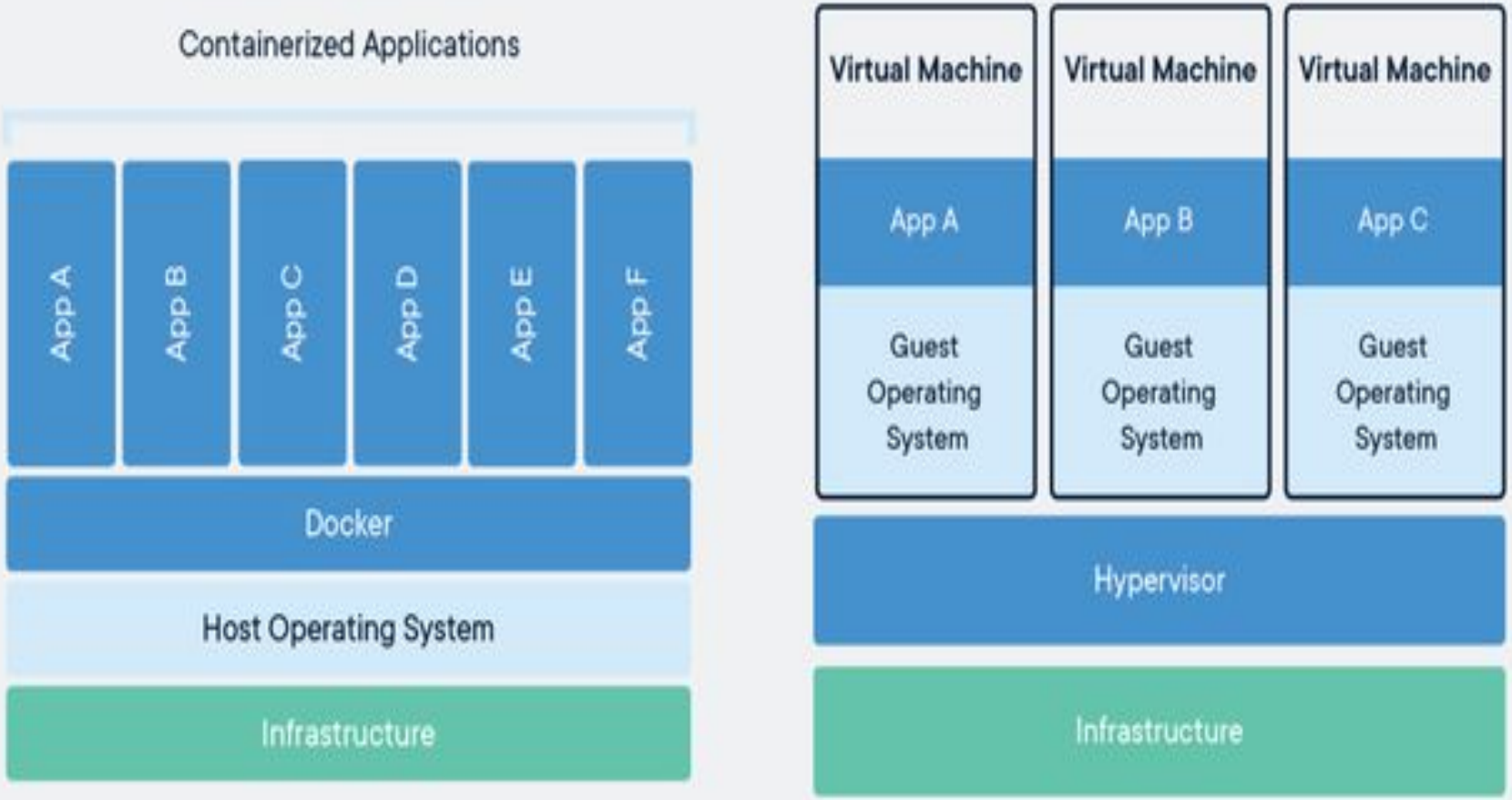
Virtual Machine VS container

VIRTUAL MACHINES:

- **Abstraction of a machine**
- Benefit is run applications in isolation
- But VM needs **full OS system**
- **Intense power** and it can be slow

CONTAINER:

- **Isolated environment for running applications**
- Also allows to **run applications in isolation**
- **Lighter power**, so need less hardware resources
- No need of full OS, and **starts quickly**

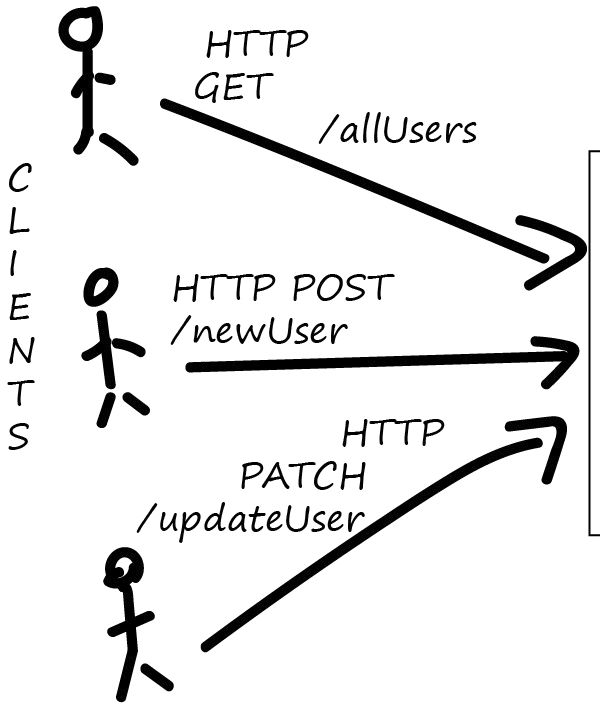

Containerized Applications
App A
App B
App C
App D
App E
App F
Docker
Host Operating System
Infrastructure
Virtual Machine
Virtual Machine
Virtual Machine
App A
App B
App C
Guest Operating System
Guest Operating System
Guest Operating System
Hypervisor
Infrastructure

How does it work?

- It's like **client -> server architecture** similar to socketserver, and it uses **Restful API**. Server is called **"Docker Engine"**
- All its container share the **kernel of the host**, which is the core of the OS
- What is restful API: It is a **contract between information provider** and the information receiver, which can relate to the **connection between client and the server** and **connects to database.**
- Restful API Architecture:
  - ***Database* <-> *Web Server* <-> *RESTFUL API* <-> *Application/Website***

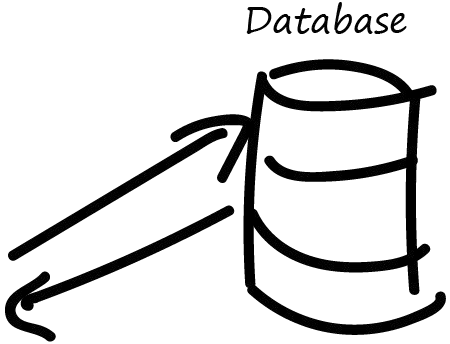
Rest API Basics

Typical HTTP Verbs:
GET -> Read from Database
PUT -> Update/Replace row in Database
PATCH -> Update/Modify row in Database
POST -> Create a new record in the database
DELETE -> Delete from the database



Rest API

Recieves HTTP requests from Clients and does whatever request needs. i.e create users



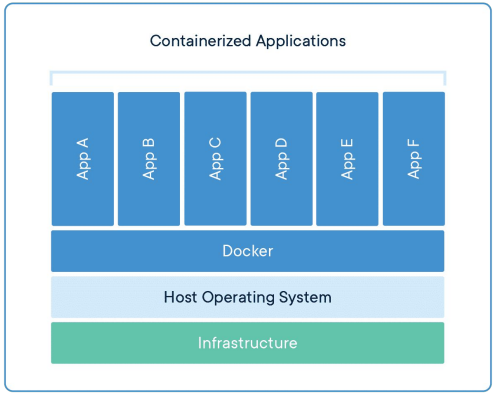
Our Rest API queries the database for what it needs

Our Clients, send HTTP Requests and wait for responses

Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.

To make it work

- **Build** an **image** as a **package** of all the source-codes, files, packages, and dependencies, so that it can all be in one “image” and can be functioned at once.
- **Run** the **image** as **container** for Docker, and with the connection to Docker file with the commands, the application will run when the dockerfile is running.



Step-by step details

1. **Open terminal** or command window on your device
2. Use **“cd”** command to take you to file location where there’s dockerfile existing.
3. Use the command **“Docker build -t hello-world . ”** to build the image from the docker (assuming that the name of the container is hello-world)
4. You can use the command **“docker images”** to see when’s the last time you build the images
5. Use the command **“Docker run -p 80:80 hello-world”** to run the container that has images
6. Check **localhost on web browser app** to see if the image is running correctly.