Docker

Dr. R. Karthi

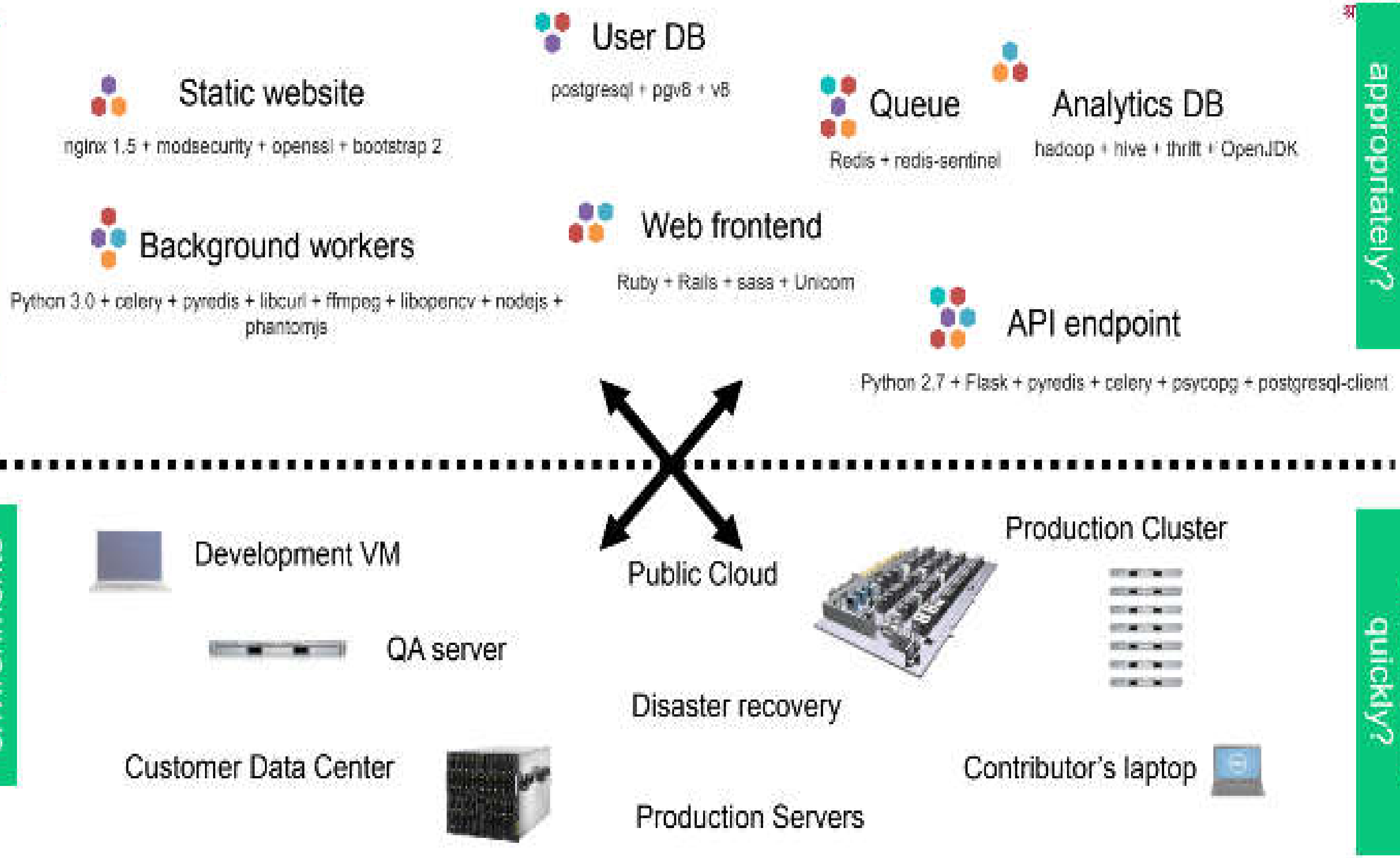
Docker

- Docker is a software containerization platform.
- Helps to build your application, package them along with their dependencies into a container
- Containers can be easily shipped to run on other machine
- Docker is a platform for developers and sysadmins to **develop, deploy, and run** applications with containers.

The Challenge

Multiplicity of Stacks

Do services and apps interact appropriately?



Static website

nginx 1.5 + modsecurity + openssl + bootstrap 2

User DB

postgresql + pgv8 + v8

Queue

Redis + redis-sentinel

Analytics DB

hadoop + hive + thrift + OpenJDK

Background workers

Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

Web frontend

Ruby + Rails + sass + Unicorn

API endpoint

Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

Development VM

QA server

Public Cloud

Production Cluster

Disaster recovery

Customer Data Center

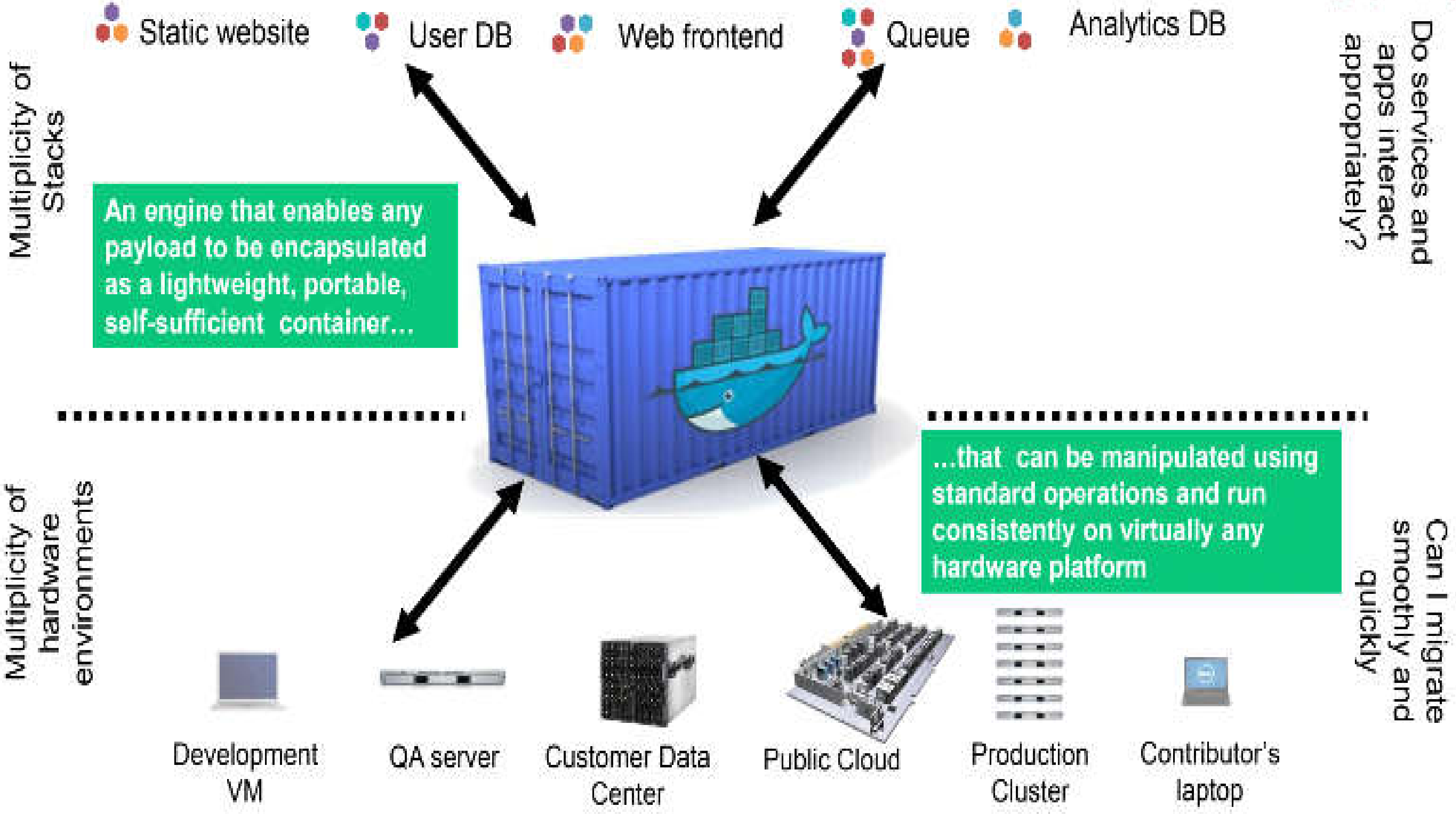
Contributor's laptop

Production Servers

Multiplicity of hardware environments

Can I migrate smoothly and quickly?

Docker is a shipping container system for code



Multiplicity of Stacks

Static website

User DB

Web frontend

Queue

Analytics DB

Do services and apps interact appropriately?

An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container…

…that can be manipulated using standard operations and run consistently on virtually any hardware platform

Multiplicity of hardware environments

Development VM

QA server

Customer Data Center

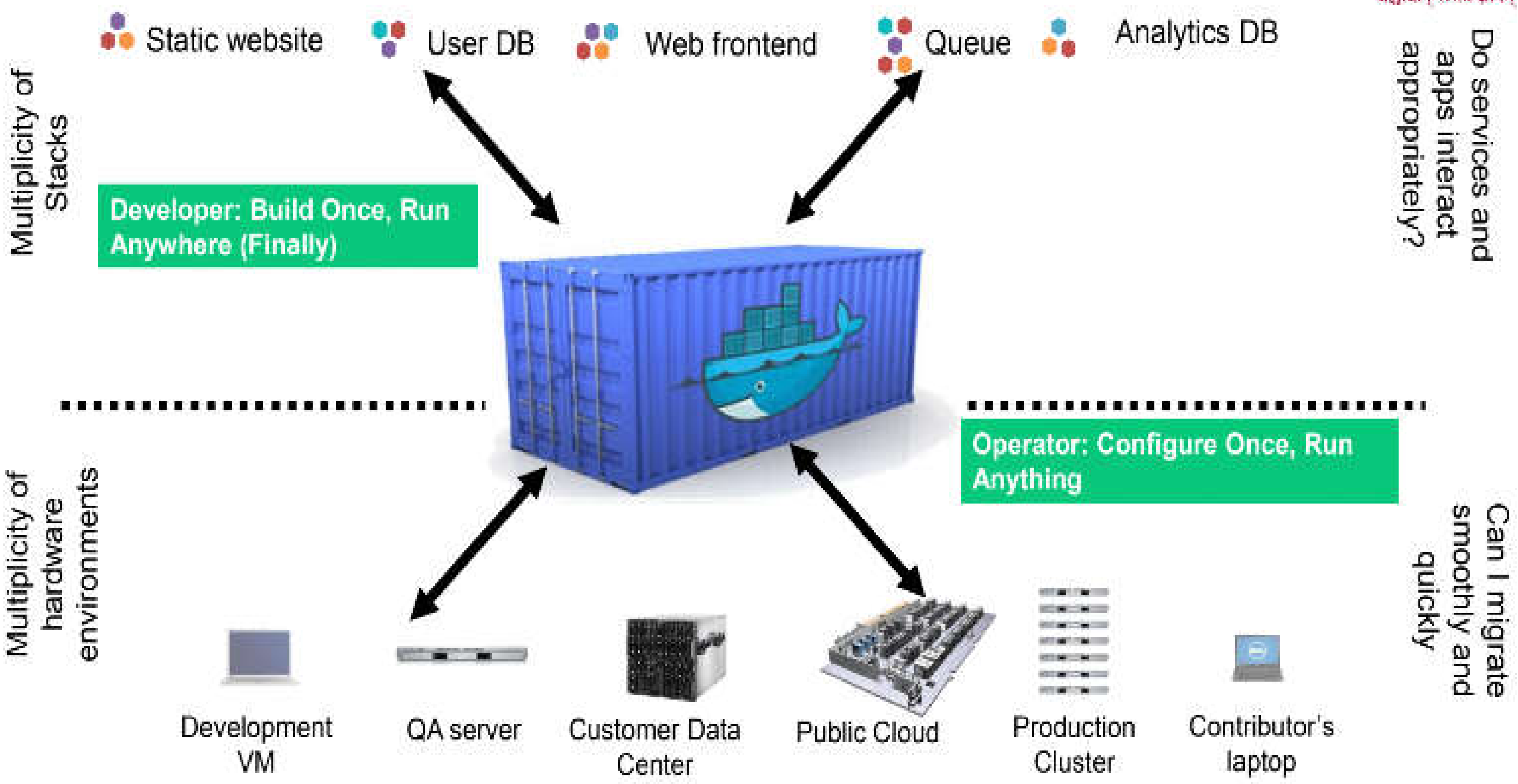
Public Cloud

Production Cluster

Contributor's laptop

Can I migrate smoothly and quickly

Or…put more simply



Static website

User DB

Web frontend

Queue

Analytics DB

Multiplicity of Stacks

Do services and apps interact appropriately?

Developer: Build Once, Run Anywhere (Finally)

Operator: Configure Once, Run Anything

Multiplicity of hardware environments

Can I migrate smoothly and quickly

Development VM

QA server

Customer Data Center

Public Cloud

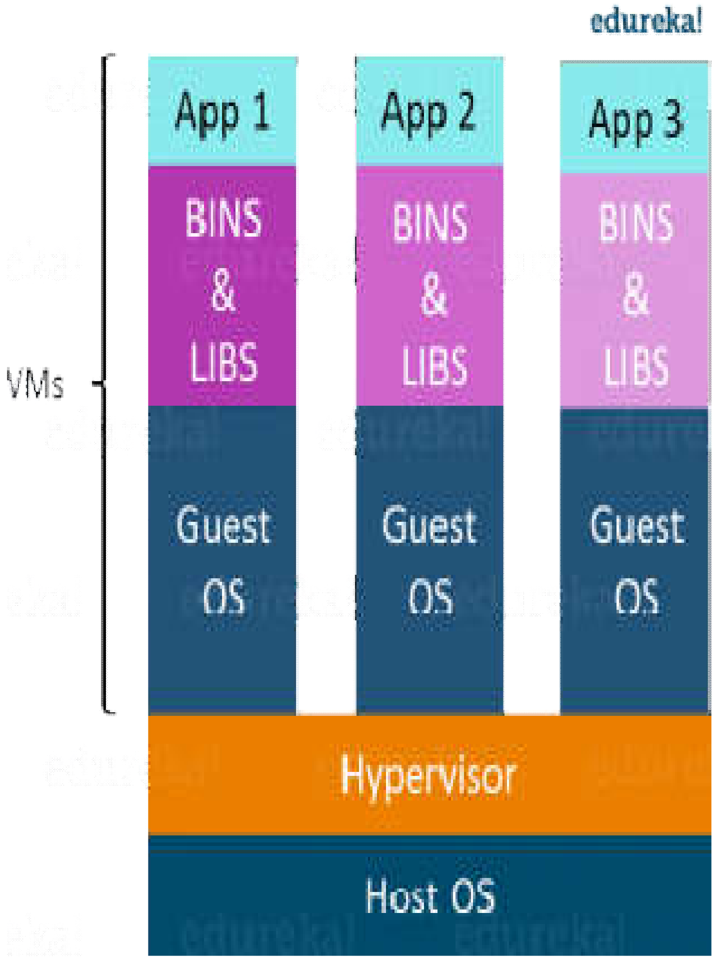
Production Cluster

Contributor's laptop

Creating a container

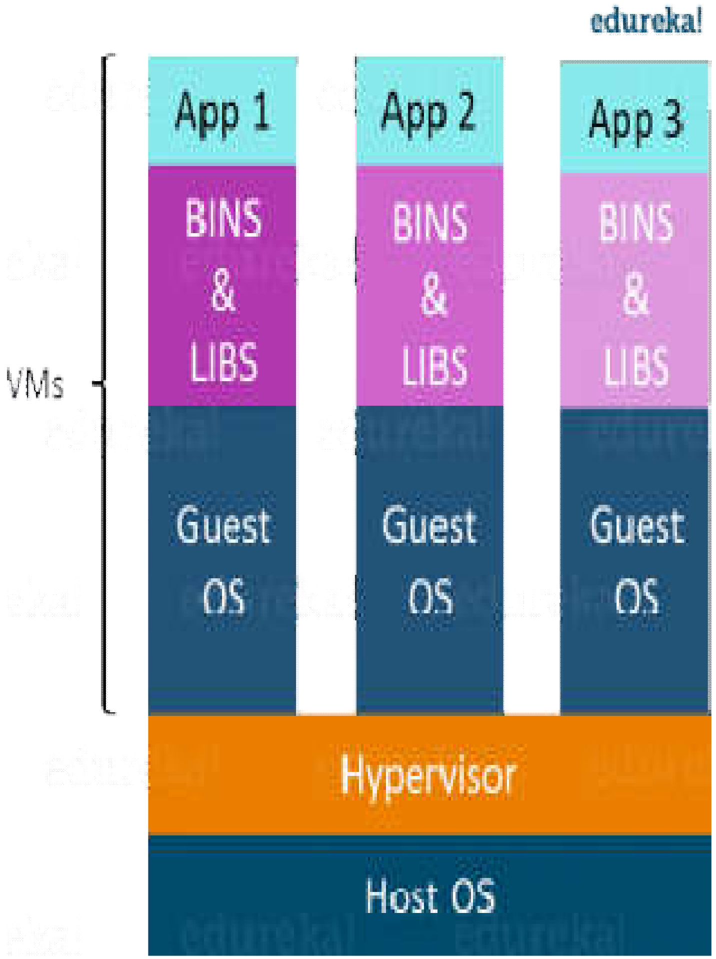
- Linux based application which has been written both in Ruby and Python.
- This application requires a specific version of linux, Ruby and Python.
- A linux docker container can be created with the required versions of Ruby and Python installed along with the application.
- End users can use the application easily by running this container without worrying about the dependencies or any version conflicts.

Virtualization



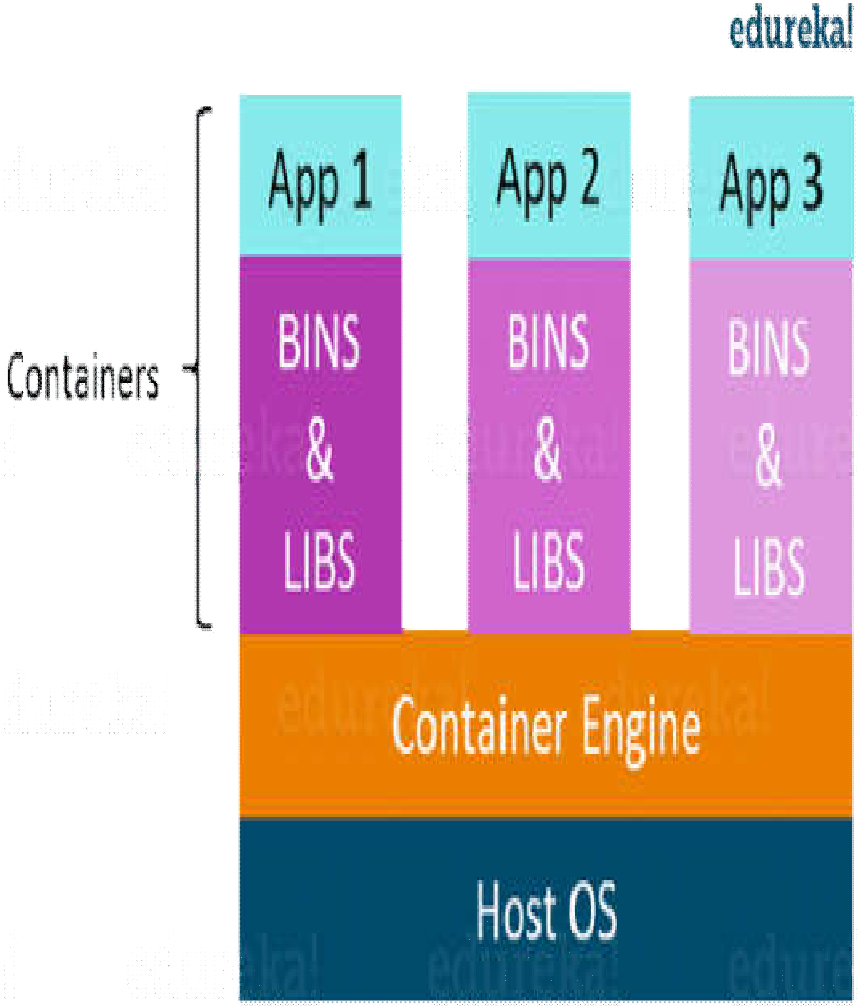
- Virtualization is the technique of importing a Guest operating system on top of a Host operating system.
- Developers can run multiple operating systems in different virtual machines all running on the same host.
- The advantages of Virtual Machines or Virtualization are:
  - Multiple operating systems can run on the same machine
  - Maintenance and Recovery were easy in case of failure conditions
  - Total cost of ownership was also less due to the reduced need for infrastructure

Virtualization



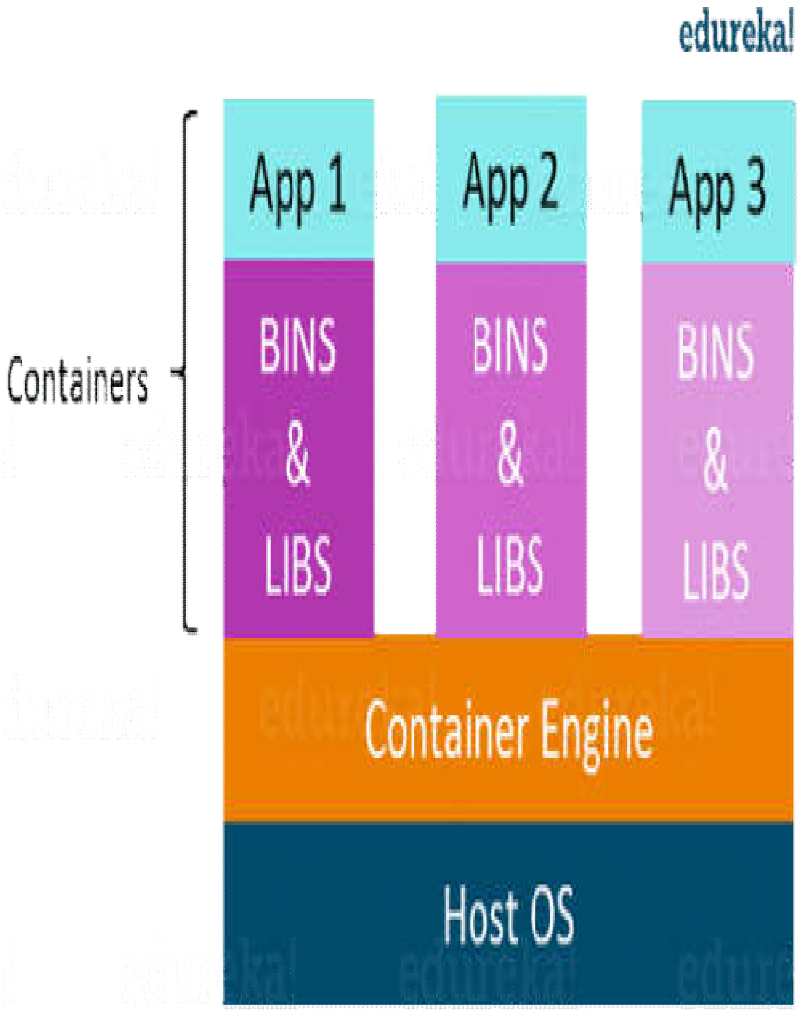
- Disadvantages of Virtualization are
  - Running multiple Virtual Machines in the same host operating system leads to performance degradation.
  - Guest OS is running on top of the host OS, which will have its own kernel and set of libraries and dependencies.
  - large chunk of system resources, i.e. hard disk, processor and especially RAM is used
  - Hypervisors are not as efficient as the host operating system
  - Boot up process is long and takes time

Containerization



- Containerization is the technique of bringing virtualization to the operating system level.
- Containerization has no guest OS and utilizes a host's operating system.
- Application specific binaries and libraries of containers run on the host kernel, which makes processing and execution very fast.

Containerization

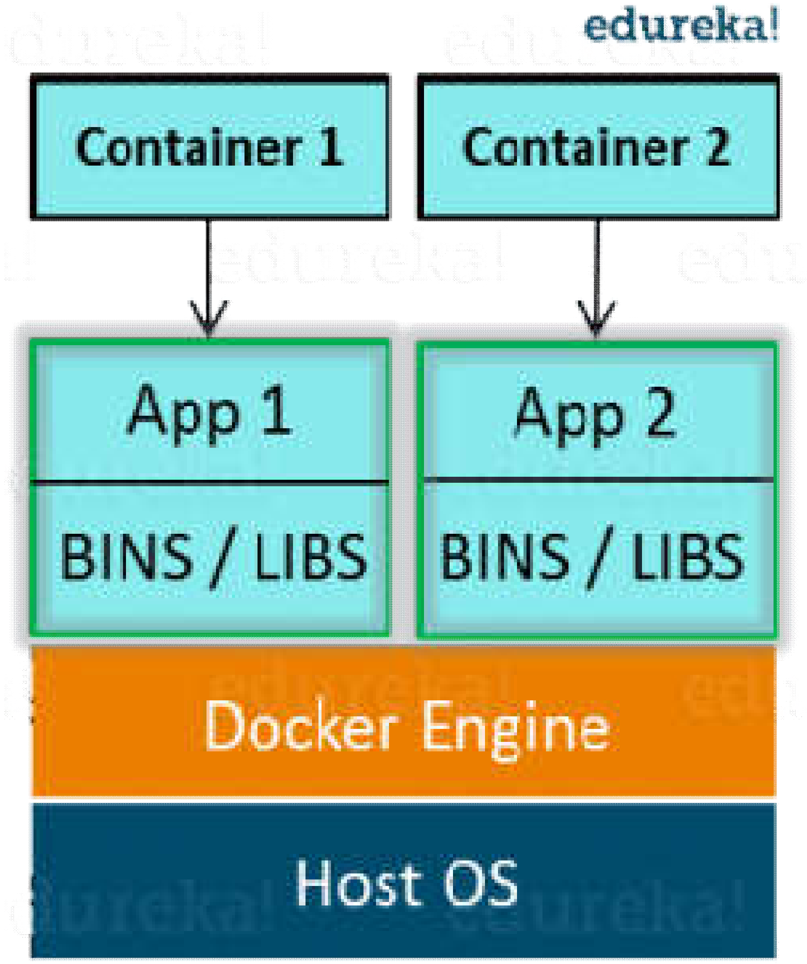


Advantages of Containerization over Virtualization:

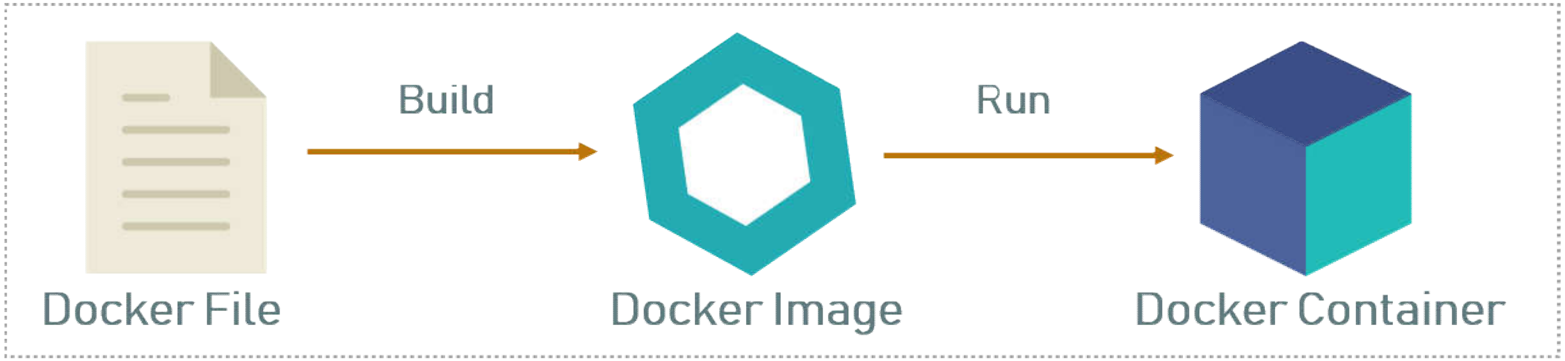
- Containers on the same OS kernel are lighter and smaller
- Better resource utilization compared to VMs
- It runs a discrete process, taking no more memory than any other executable, making it lightweight.
- Boot-up process is short and takes few seconds

Docker

- Docker is a containerization platform
- It packages your application and all its dependencies together in the form of Containers
- Helps application to works seamlessly in any environment.
- Ensures process level isolation
- Docker is a platform for developers and sysadmins to **develop, deploy, and run** applications with containers.
- The use of Linux containers to deploy applications is called *containerization*.



Dockerfile, Images & Containers



Dockerfile is built, it becomes a Docker Image
Docker Image is run, it becomes a Docker Container.

Docker

- **Dockerfile:** A Dockerfile is a text document which contains all the commands called to build an image.

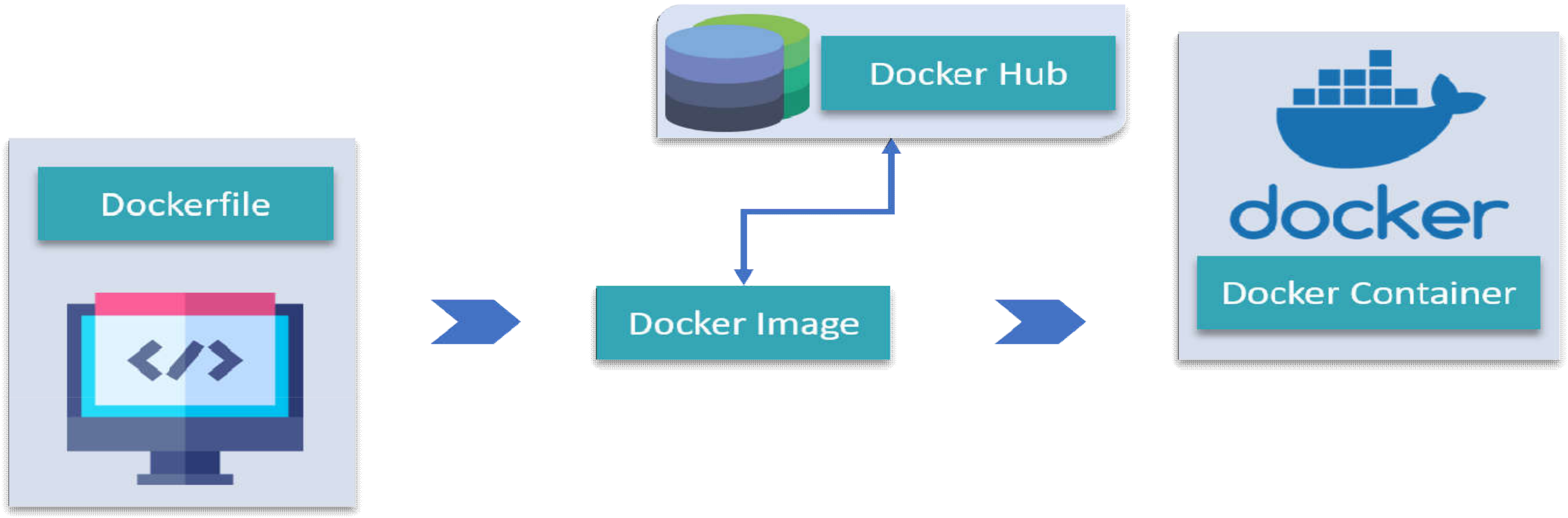
Dockerfile

```
FROM node:10
WORKDIR /app
COPY package.json /app
RUN npm install
COPY . /app
CMD node index.js
EXPOSE 9000
```

Docker Image and Containers

- An **Docker image** is an executable package that includes everything needed to run an application--the code, a runtime, libraries, environment variables.
- Read-only templates and are used to create containers
- A **container** is a runtime instance of an image
- Container is what the image becomes in memory when executed (that is, **an image with state, or a user process**).

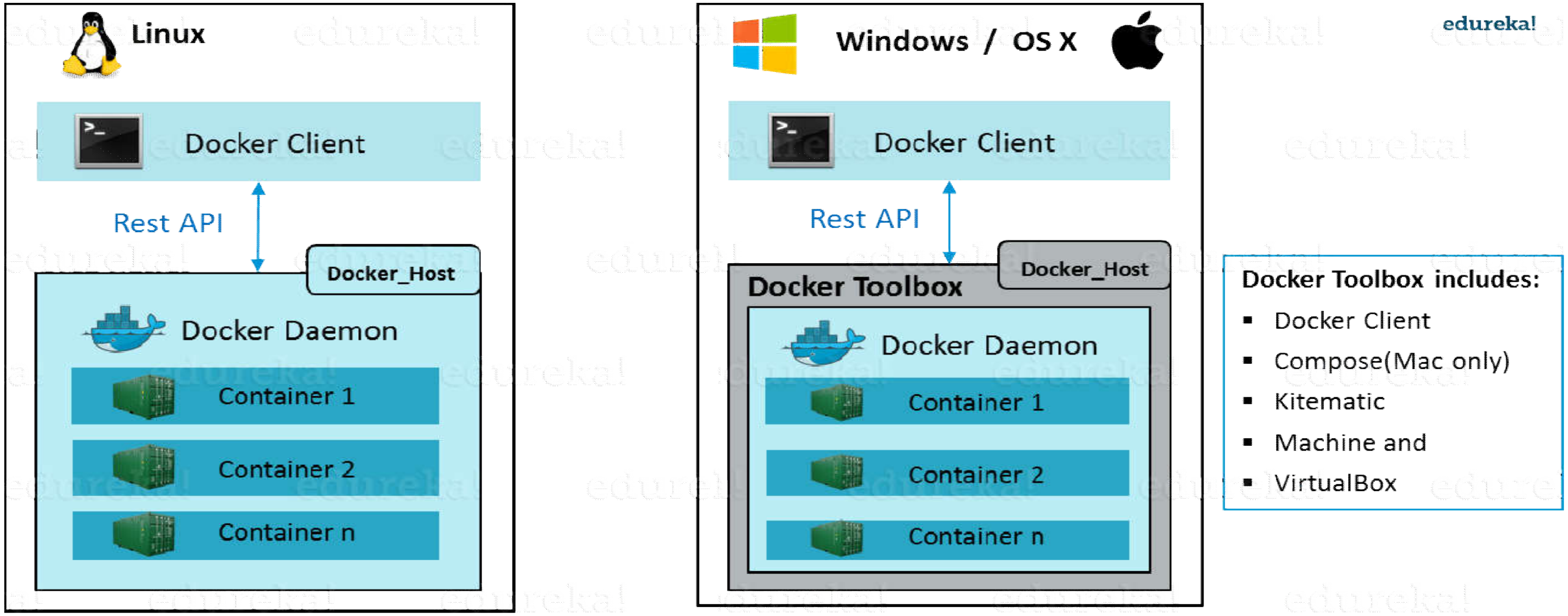
Docker Hub



- Docker Hub is like GitHub for Docker Images.
- It is a cloud registry where you can find Docker Images uploaded by different communities
- Develop your own image and upload on Docker Hub

Docker Engine

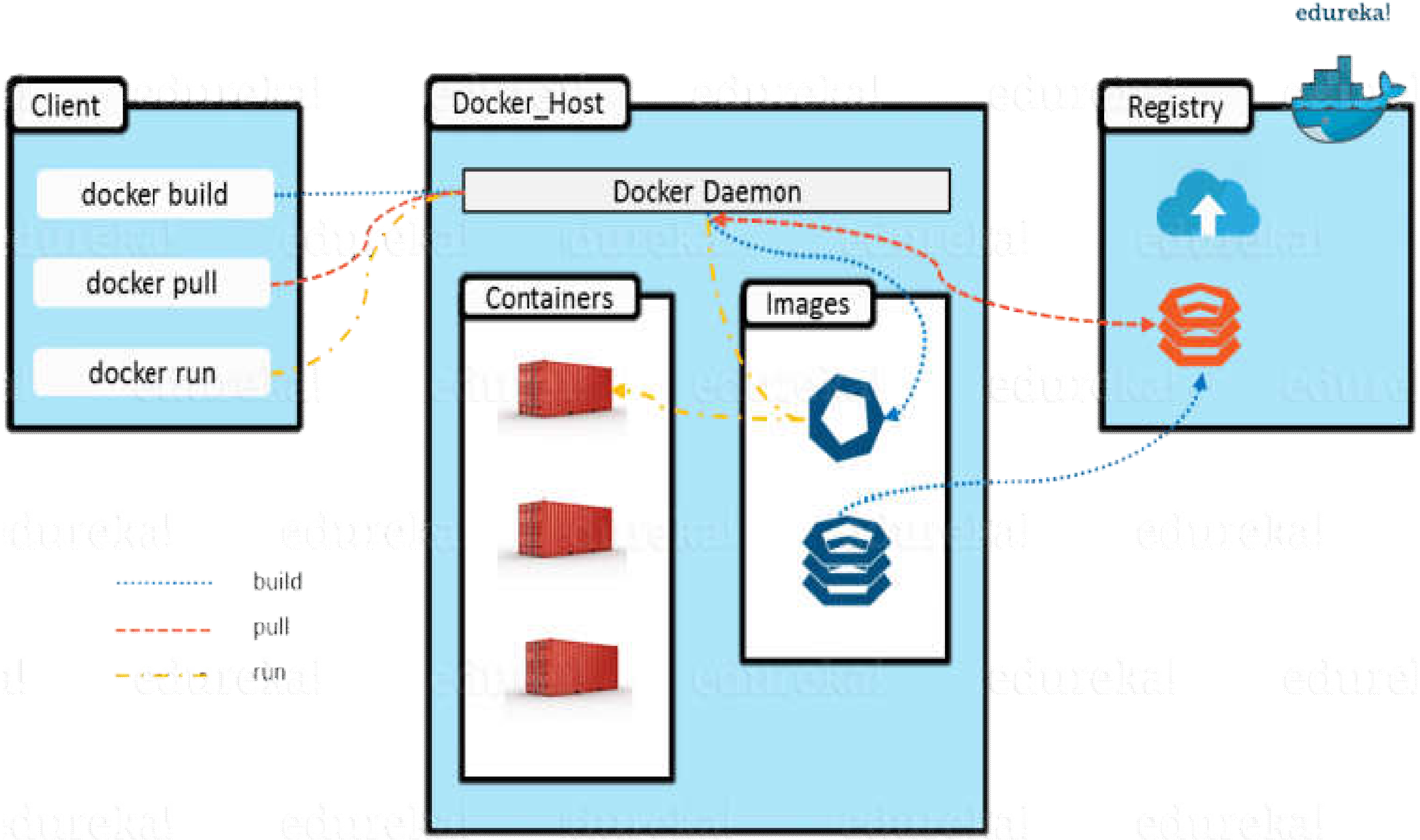
- ✓ Docker Engine is docker application that is installed on host machine.
- ✓ It is a client-server application which uses:
  - A **server** - a daemon process
  - A **client -** command line interface (CLI)
  - REST API is used for communication between the CLI client and Docker Daemon



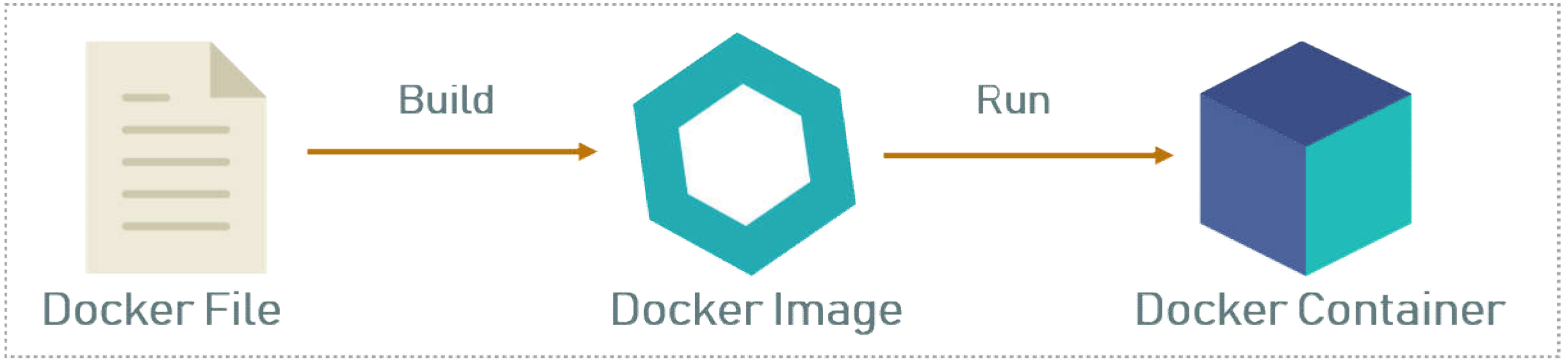
Docker Toolbox includes:

- Docker Client
- Compose(Mac only)
- Kitematic
- Machine and
- VirtualBox

Docker Architecture



Dockerfile, Images & Containers



Dockerfile is built, it becomes a Docker Image
Docker Image is run, it becomes a Docker Container.

Docker Commands

- docker pull - pull images from the docker repository
- docker push - used to push an image to the docker hub repository
- docker images - lists all the locally stored docker images
- docker run - create a container from an image
- docker ps - List the running containers
- docker ps –a - show all the running and exited containers
- docker exec - is used to access the running container
- docker stop - stops a running container
- docker commit - creates a new image of an edited container on the local system
- docker login - login to the docker hub repository
- docker rm - delete a stopped container
- docker build - build an image from a specified docker file

Dockerfile

```
FROM node:10
WORKDIR /app
COPY package.json /app
RUN npm install
COPY . /app
CMD node index.js
EXPOSE 9000
```

Index.js

```
var express = require('express');
var app = express()

app.get('/',function(req,res) {
   res.send(" Hello World Welcome");
})

app.listen(9000, function(){
   console.log("Hai in function")
})
```