**Docker is a set of platforms as a service (PaaS)** products that use Operating system-level virtualization to deliver software in packages called containers

Docker is an open-source containerization platform by which you can pack your application and all its dependencies into a standardized unit called a container

Q. WHAT IS DOCKER FILE?

The Docker file uses DSL (Domain Specific Language) and contains instructions for generating a Docker image. Docker file will define the processes quickly produce an image.

While creating your application, you should create a Docker file in order since the Docker daemon runs all of the instructions from top to bottom.

Q. WHAT IS DOCKER IMAGES?

It is a file, comprised of multiple layers, used to execute code in a Docker container. They are a set of instructions used to create Docker containers.

**Docker Image** is an executable package of software that includes everything needed to run an application. This image informs how a container should instantiate,

Determining which software components will run and how. Docker Container is a virtual environment

That bundles application code with all the dependencies required to run the application. The application runs quickly and reliably from one computing environment to another.

Q. WHAT IS DOCKER CONTAINER?

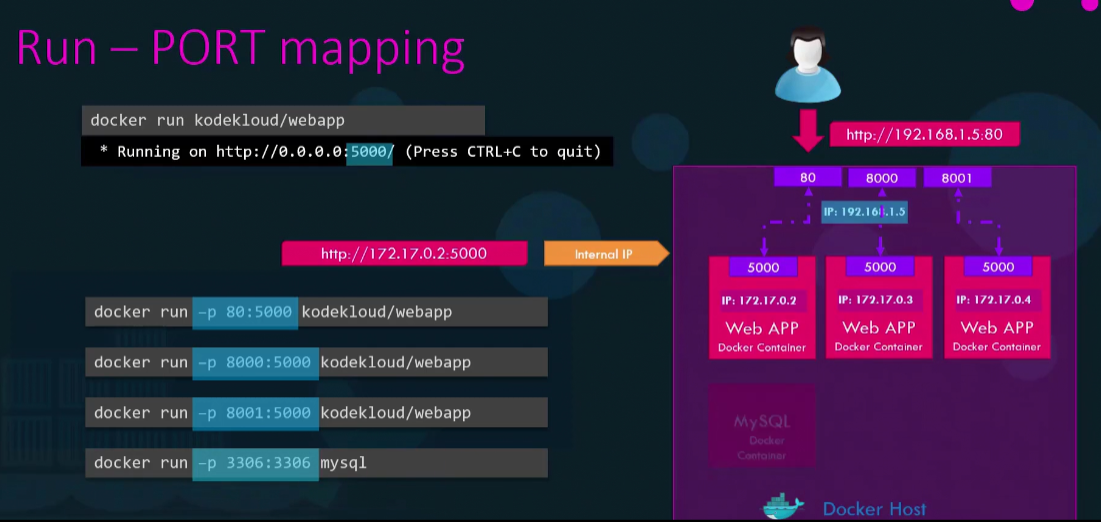
Containers are meant to run a specific task or process such as to host an instance of a web server, or an application server, database or simply carryout some computational or analysis tasks.

**Docker container** is a runtime instance of an image. Allows developers to package applications with all parts needed such as libraries and other dependencies. Docker Containers are runtime instances of Docker images.

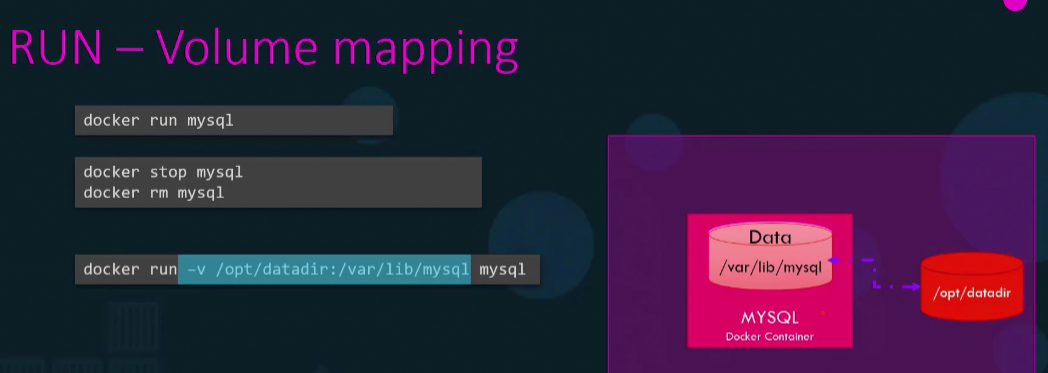
* Underlying host where Docker is installed is call Docker host or Docker engine
* Every Docker container gets an IP assigned by default 🡪 that is internal IP that can be accessible with in the Docker hub
* If we want to access that from the outside then we have to use Docker hub IP address

For that, we have to map port 80 to 5000

Docker run –p 80:5000 /kodekloud



* You can’t map the same port on the Docker host more than once
* If we would like to persist data, you would want to map a directory outside the container on the Docker host to a directory inside the container.[if we didn’t map after stopping it entire data will be deleted]
* Docker run –v /opt/datadir: /var/lib/mysql mysql

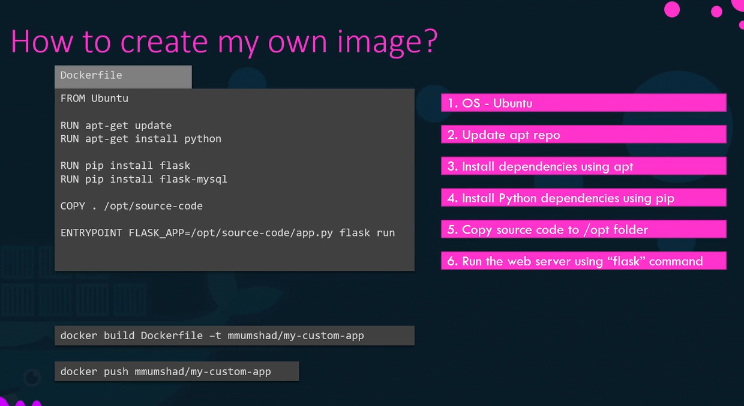


* Docker inspect <container name or ID> 🡪 to get detailed information about a container [that will return all details about the container in the json format]
* Docker logs <container name> 🡪To see the logs we ran in the background [-d – detatch mode]

Why we need to create image?

Developing will dockerised for ease of shipping and deployment.

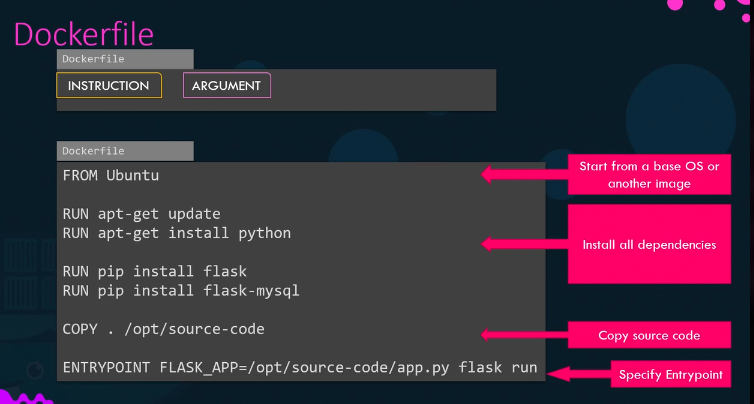
How to create image?



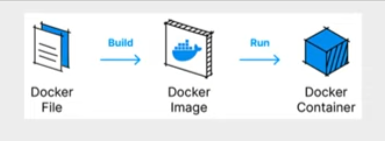
**Docker file** is a text file written in a specific format that Docker can understand.

It is an instructions and arguments format

Instructions 🡪 these instructs Docker to perform specific actions while creating the image.



ENTRYPOINT🡪specify the program that will run when the container starts.



**Docker Registry:** it is a central repository of all Docker images.

**Docker Compose**

**docker-compose.yml**

To set up and run multiple containers as part of a single application

It is a configuration file describes how different services, networks, and volumes should work together**.**

****

**Container orchestration** contains multiple Docker hosts that can host containers.

A container orchestration solution allows you to deploy hundreds or thousands of instances of your application with a single command.

Orchestration solutions

Docker swarm from Docker

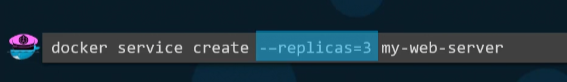
Kubernetes from google

MESOS from apache

**Docker swarm** helps us to combine multiple Docker machines together into a single cluster.

Docker swarm take care of distributing your services or your application instances into separate hosts for high availability for load balancing across different systems and hardware’s.

**Docker services** are one or more instances of a single application or service that can runs across the nodes in the swarm cluster.



Docker service command must run on the manager node not in the slave node.

**Cluster** is a set of nodes grouped together.

**Master** is a node with a kubernetes control plane components installed. The master watches over the nodes in the cluster and is responsible for the actual orchestration of the containers on the worker node.