# **Why Docker**

Docker is a centralized platform for packaging, deploying and running applications. Before docker, many users faced the problem that the code was working on developer’s environment but it was not working on users environment, hence , to overcome this problem, docker came into existence.

**What is Docker**

Docker is an open source, centralized platform to create , deploy and run applications. Docker uses containers on the host operating system to run applications. Containers ensure that the application works in any environment, be it production, test or development.

**Docker Container**

Docker containers are the lightweight alternatives of the virtual machine. It allows developers to package up the applications with all its libraries and dependencies and ship it as a single package. We don’t need to allocate RAM or space to containers, it is done automatically.

**Advantages**

* It runs the containers in seconds instead of minutes.
* It uses less memory.
* It provides lightweight virtualization.
* It uses application dependencies to reduce the risk.

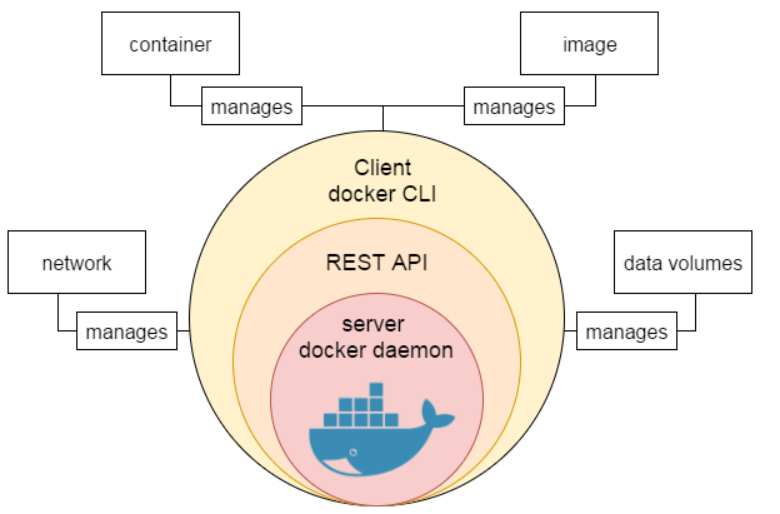
**Disadvantages**

* It increases complexity due to an additional layer.
* It is difficult to manage large amount of containers.

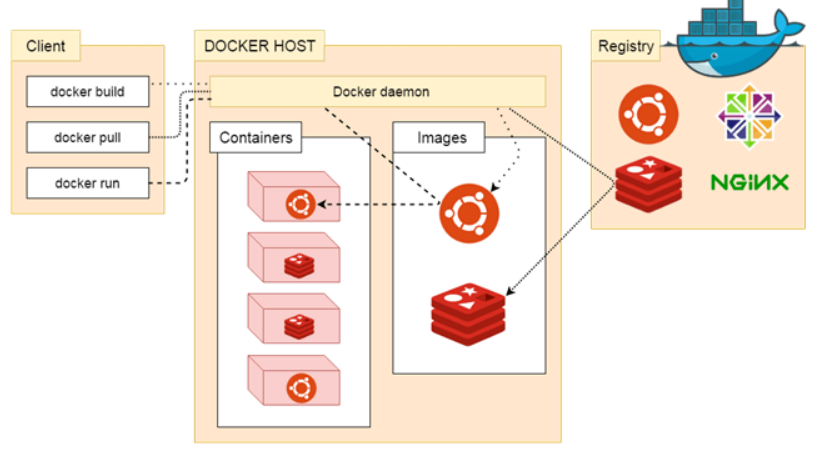
**Docker Engine**

It is a client server application that contains following major components:

* A server which a long running program called daemon process
* The REST Api is used to specify interfaces that programs can be used to talk to daemon and instruct it what to do.
* A command line interface client.



**Docker Architecture**



* **Docker Client** : It uses Rest Apis and commands to talk to docker daemon process. When a client runs any docker command, it is sent to docker daemon, which processes it and executes it.
* **Docker Host** : It is used to provide an environment to execute and run applications. It contains the docker daemon, images, containers, networks and storage.
* **Docker Registry** : It manages and stores the docker images. Ex -DockerHub
* **Docker Images**: Docker images are read only binary templates which are used to create docker containers.
* **Docker Networking** : Docker networking is used when there needs to be interaction between two containers. It acts as a bridge.

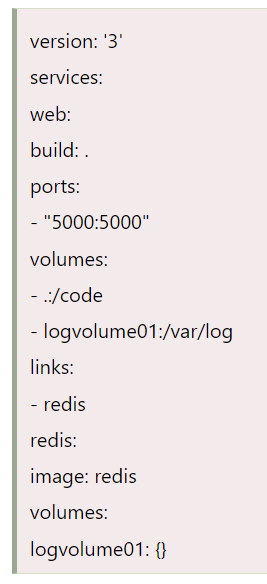
**Dockerfile**

A dockerfile is a text document which contains commands to assemble a docker image. Docker builds images automatically by reading the commands from command line.

* FROM - Creates a layer from the ubuntu:18.04
* PULL - Adds files from your Docker repository
* RUN - Builds your container
* CMD - Specifies what command to run within the container
* COPY-This instruction is used to copy new files or directories from source to the filesystem of the container at the destination.
* WORKDIR - The WORKDIR is used to set the working directory for any RUN, CMD and COPY instruction that follows it in the Dockerfile. If work directory does not exist, it will be created by default

When we are having multiple containers and we want interaction between them, then either it can be done with docker networking or it can be achieved with docker compose.

Docker Compose.yml



**Imp Docker commands**

* docker search MySQL - We can use the command docker search to search for public images on the Docker hub. It will return information about the image name, description, stars, official and automated.
* docker pull --platform linux/x86\_64 mysql - Now that we know the name of the image, we can pull that from the Docker hub using the command docker pull.
* docker images - now we should have some images in our local machine, and to confirm
* docker run --env MYSQL\_ROOT\_PASSWORD=my-secret-pw --detach mysql-now that we have some images, we can try to create a container. Here we used the --env option to set a mandatory environment variable and --detach option to run the container in the background.
* docker ps - list all the running containers
* docker ps –all - list all the containers
* docker stop f8c52bedeecc - use the docker stop command with either the container id or container name.
* docker rm test\_db - In case we want to remove a container, we can use the following command.