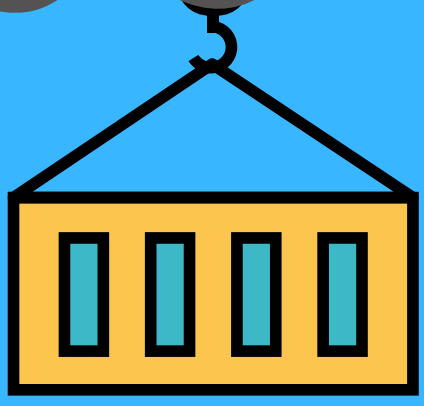
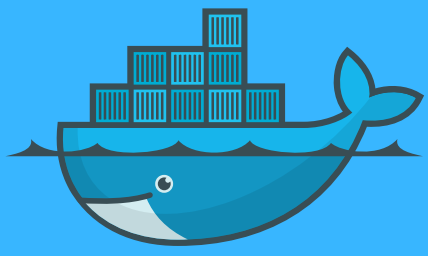
QUICK TIPS

@mayank ahuja

UNDERSTANDING DOCKER



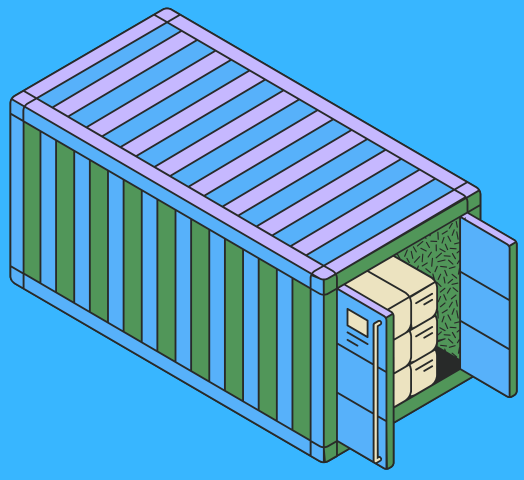


Handbook

REPOST

Let's first try to understand a Container

In simple terms, a container is a software unit that packages the application code and all its dependencies or everything that the application needs to run.



We can also say that a Container is

actually set of one or more processes that are isolated from rest of the system.

@mayank ahuja

Container vs VM

VM =>

Hardware level virtualization
Overhead of Guest OS in every instance.

Containers =>

OS level virtualization (Resources managed using kernel features - Namespaces & Control Groups).

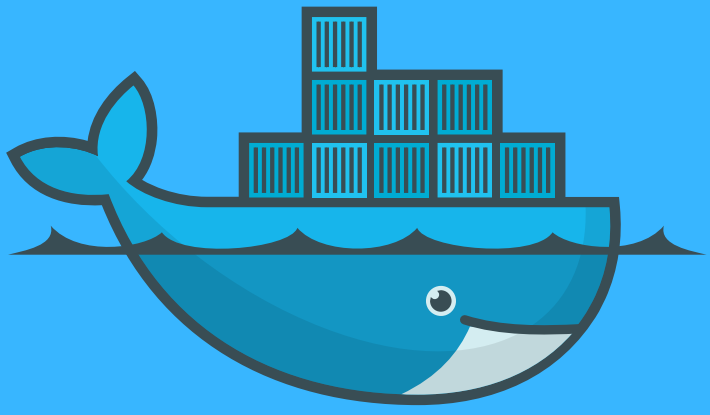
No Guest OS required (leverages the features/resources of the host OS).

@mayank ahuja

What is Docker?

We often use the terms Docker and Containers interchangeably.

Docker is actually a platform that provides the ability to package and run application in a container.

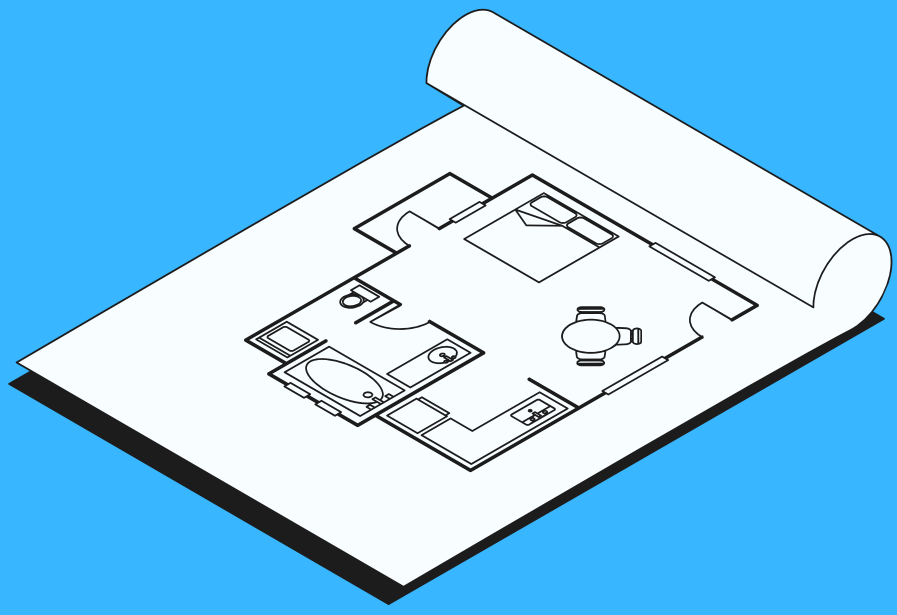


When we talk about containers we usually refer to Docker Engine which manages the container lifecycle.

@mayank ahuja

Brief about Docker Architecture

A classical client-server architecture.

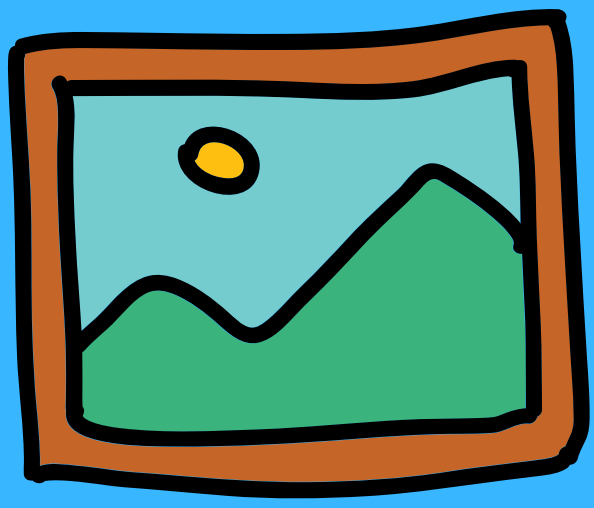


You need to perform/run docker commands using the docker client, request (via Rest API call) then goes to the Docker Daemon (server side) which helps you build and manage docker objects like containers, images etc.

@mayank ahuja

Let's now talk about Images

Images are like foundation for building containers.
You create container from an image.



An image is like a recipe (including ingredients) that will help you make your dish (container).

Basically, image is the bundlet hat contains all dependencies required to run a container.

@mayank ahuja

3 Ways to Build Images

You can use an existing image from the registry.

Use 'docker search' - to search an image
'docker pull' - to pull the image to local and use it.

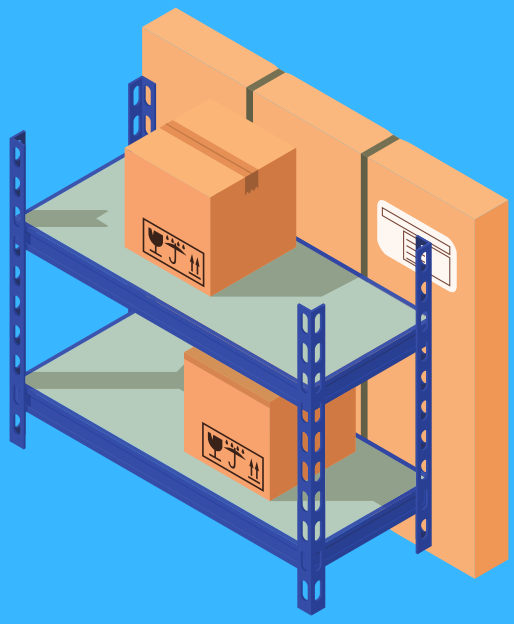
You can write a Dockerfile and build an image out of it, using 'docker build'

You can create image out of a container as well, using the 'docker commit' command.

@mayank ahuja

Image Registry

It's a service - could be private or public - where you can store the images and search & retrieve them.



You can also store different versions of the image, hence you get feature of image version control.

@mayank ahuja

Dockerfile

One of the way which can help you build a docker image.

A Dockerfile is like a blueprint, where you basically write instructions to build an image.

@mayank ahuja

Dockerfile Instructions

FROM - specifies the base image your new image will be built upon

WORKDIR - sets the working directory inside the container for subsequent instructions

COPY/ADD - copies files and directories from your local machine (build context) into the container's filesystem

RUN - executes commands inside the container during the build process

ENV - sets environment variables that will be available within the container

CMD - specifies the default command to run when the container starts
This is typically your application's main process.

*** **Do read about ENTYRYPOINT.**

@mayank ahuja

Some Important Facts

Internally an Image actually is a set of layers.

You can see the image layers using command 'docker history image'.

The layering helps to speed up the image builds and intermediate layers are also cached for reuse.

When you create an image you usually create it on top of a parent image.

@mayank ahuja

5 essential Docker commands

docker run - to create and start a Docker container from an image

docker ps - to list all running Docker containers

docker build - to build a Docker image from a Dockerfile.

docker pull - to download a Docker image from a registry (like Docker Hub)

docker exec - to execute a command inside a running Docker container

@mayank ahuja

@mayank ahuja

THANK YOU

learn SOMETHING new EVERYDAY

REPOST