Docker :

Container :

1. A way to package application with all the necessary dependency and configuration.
2. Portable can easily share between other member or client.
3. Make development and deployment easy.
4. Live in container repository (that can be private or public e.g., Docker Hub)
5. Own isolated environment.
6. Package with all needed configuration and one command to install the app.

So basically container is the layers of the images, mostly Linux base image because of small size.
Application image on the top.

Difference in Virtual Machine and Docker :

- Docker virtualise the application layer of the OS. As it uses the host kernel as it doest have its personal.
- VM virtualise at the OS kernel layer. It uses it own kernel so uses more hardware and computational power.

Container and Image:

- Container is a running environment for image.
- Application images are the applications like Postgres, redis, Django etc
- Container consist Port minded: talk to application running inside the container.
- Images can exist without containers, whereas a container needs to run an image to exist. Therefore, containers are dependent on images and use them to construct a run-time environment and run an application. The two concepts exist as essential components (or rather phases) in the process of running a Docker container.

Basic Command :

Image Installation:

```
docker pull image_name:image_version
```

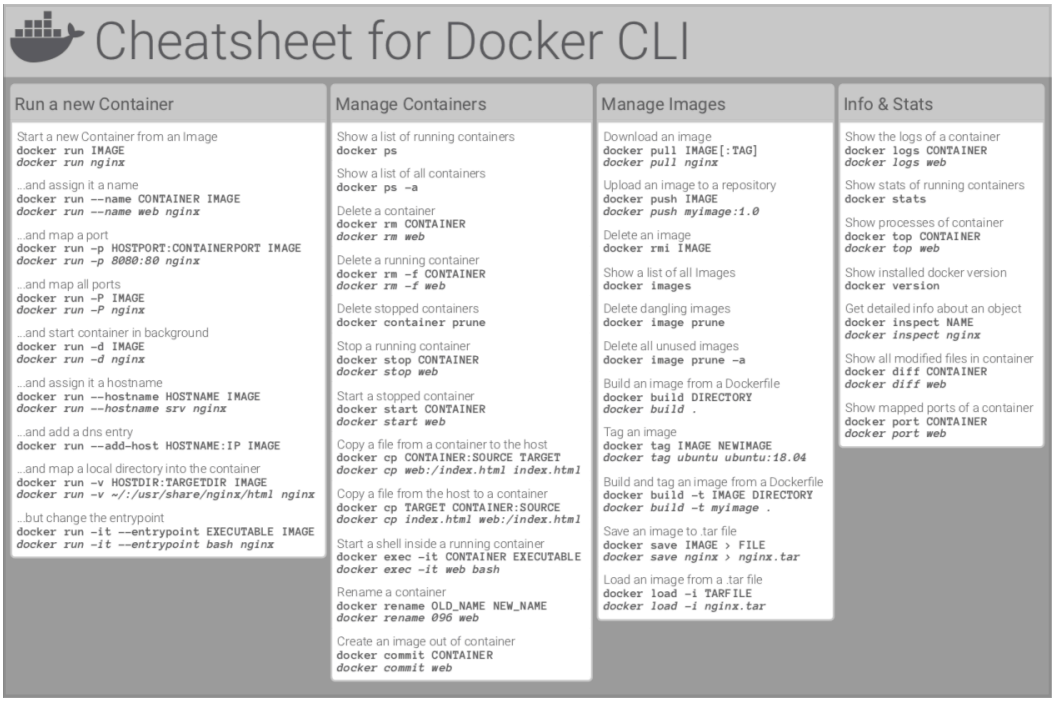
- Here image_name can be the name of the application image like postgres, redis, django etc.
- And image version is the version of the package, leave blank if need latest version

All Image:

docker images

Shows all the images available or downloaded.

Here is the some command cheats below :

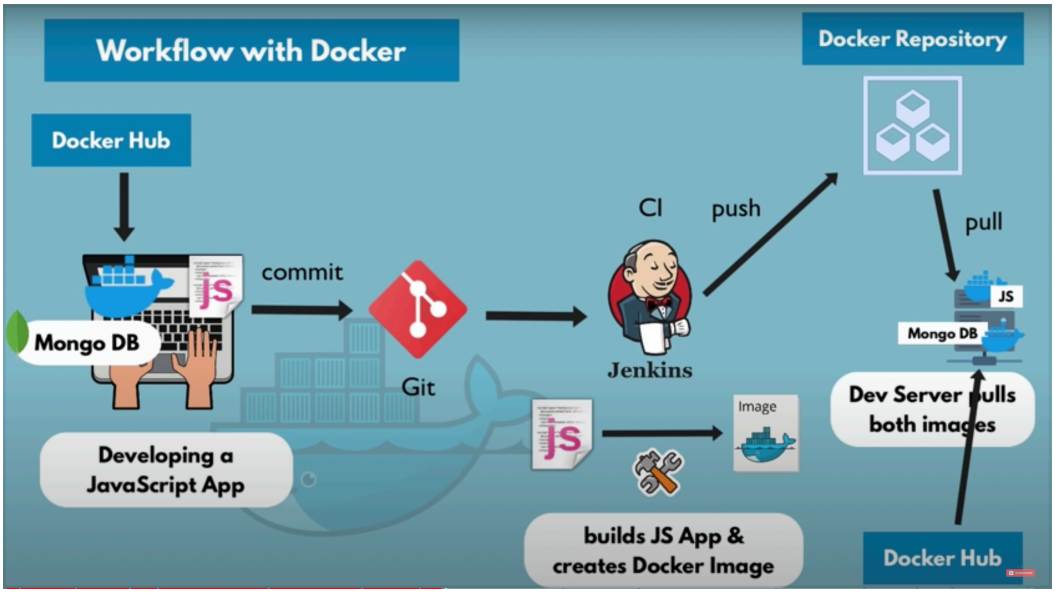


For more details regarding docker command visit :

The Ultimate Docker Cheat Sheet | do...
Docker - Beginners Intermediate Advanced
dockerlabs.collabnix.com

Docker run is used to create a new container for the image but docker start is use to restart the existing container using its name or container id.

Docker Workflow :



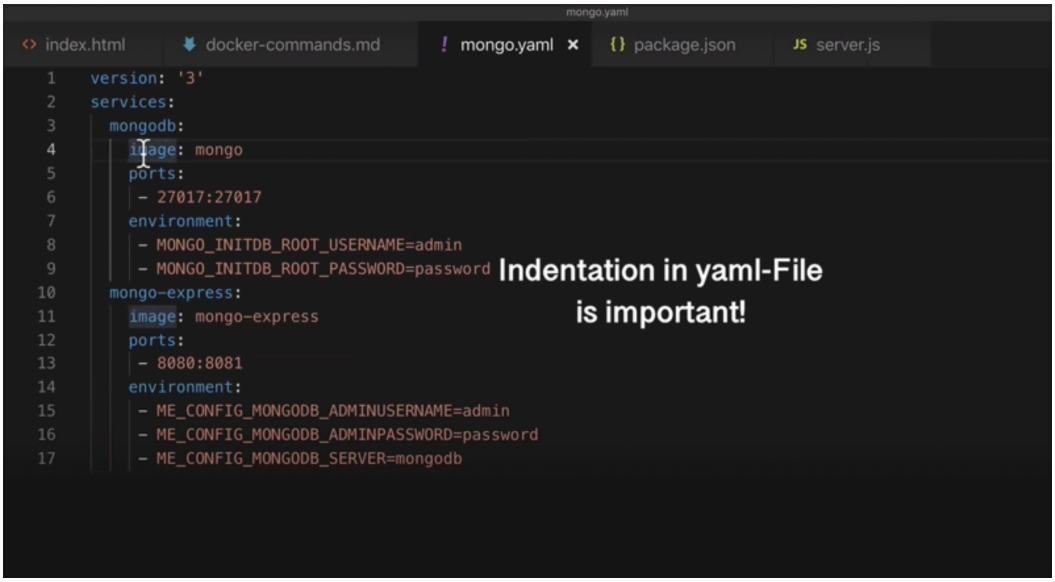
Docker Compose:

It is a structure way to contain basic commands of the docker. We use docker compose as it is easy to change in a file like environment variable etc.
So we create a docker compose file inside we add our docker commands.
We define the version, image name, ports, name, environmental variable in the file.

We don't mention the network in the docker compose as it itself take care of the common network.

File extension of docker compose file is
"file_name.yml"

Example file ;



Using docker compose :

We can use docker compose using the command,

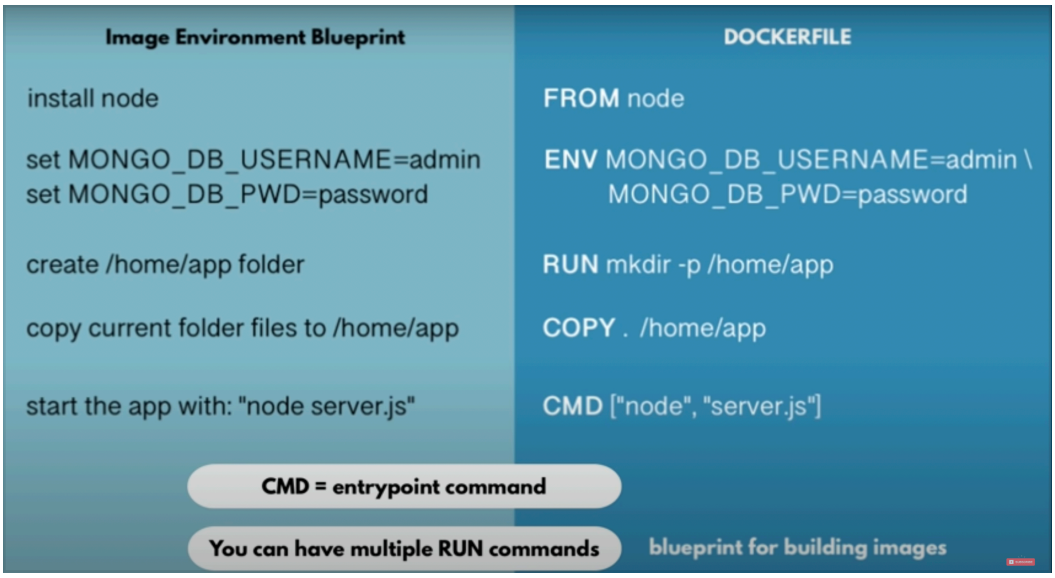
‘docker-compose -f file_name.yml up’

Here, -f denotes the file and up means start the container, to stop container use down instead of up.

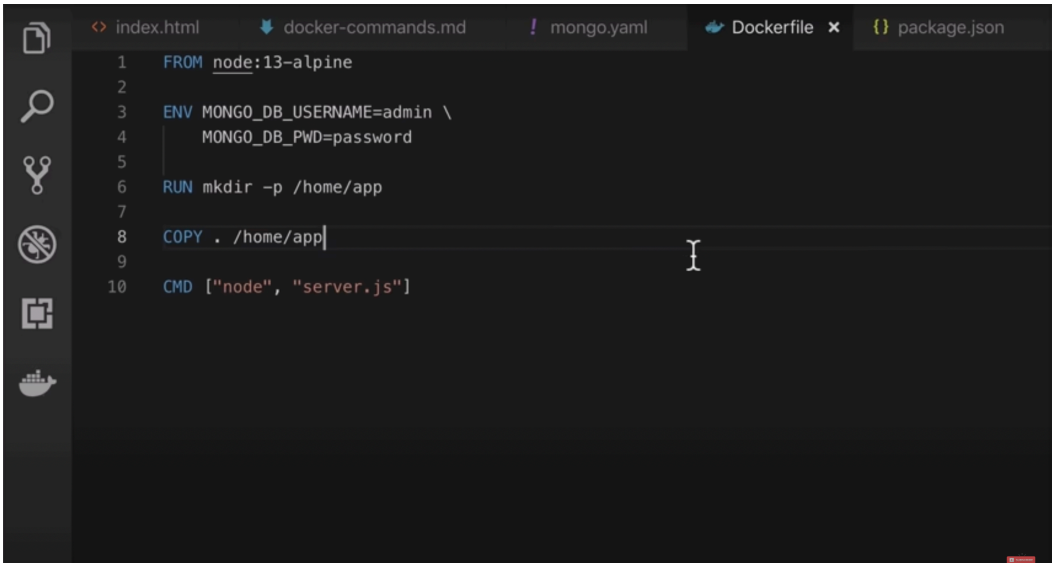
When we restart the container every configuration and data of the container will be lost. And it create new network every time when we start the container and deletes the network when we stop the container.

Docker File :

It is a blueprint for building the images.
When we deploy our application on the server before that we create the docker image of the application. So to achieve this we create the docker file. This file contains the artefacts of the application.



Example File :



Docker file name should be “Dockerfile”

After creating the docker file, to build the image use the following command,

“docker build -t image_name:tag .”

Here this command takes 2 parameter,
First is -t which is tag name, here we have to give image name and in tag we can give version-1 or 1.0
Second parameter is the allocation of the docker file. If the directory is same folder as Dockerfile so we can use “ . ”.

“App including Dockerfile is committed to git”

“When we adjust the docker file, we must rebuild the image.”

Docker Volume :

Docker Volume is used for the data persistence.
Basically if we restart or stop the container, then the data in the container will get removed or deleted. So to resolve that issue we use the docker volume.

So what docker volume does is, it connect the host machine physical file system with the directory of the folder in the virtual file system of the docker or can say connect with the container.
As a result data get automatically replicated from the virtual file system to the host file system and vice versa.

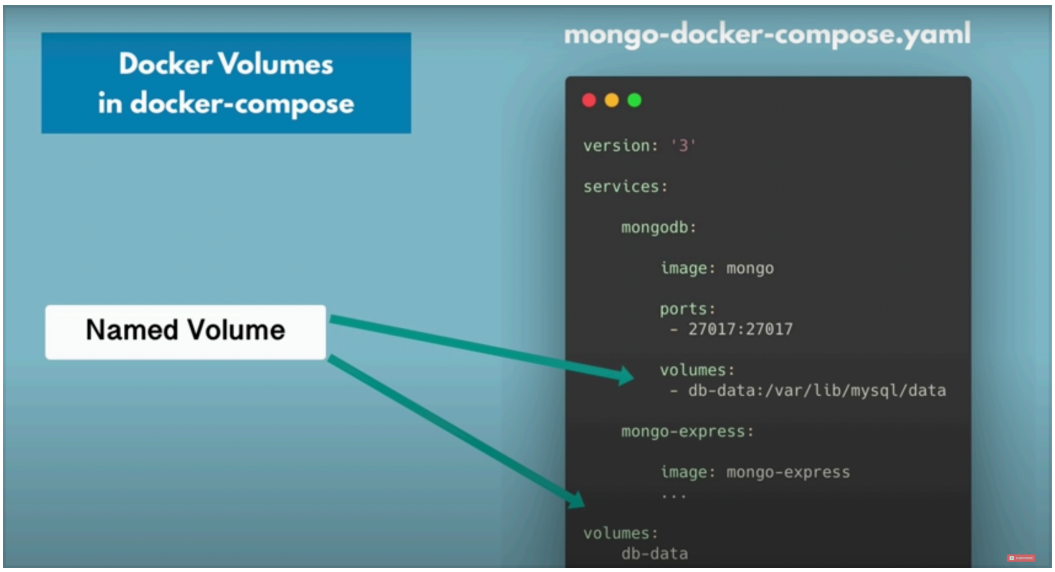
So we can use volume in the following way.

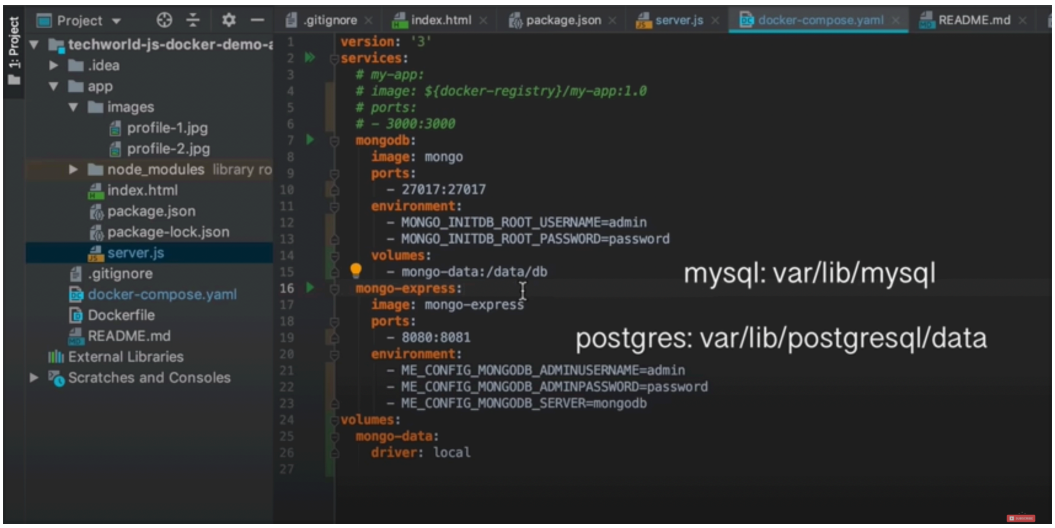
Using Command :
So in command we specify the -v property in the run command. i.e.,

“Docker run -v host_file-directory:virtual-file_directory image_name”

So here we can give host file directory Or the folder name in the place of host_file_directory
We can also set it blank, in that case docker will take care of the host file directory.

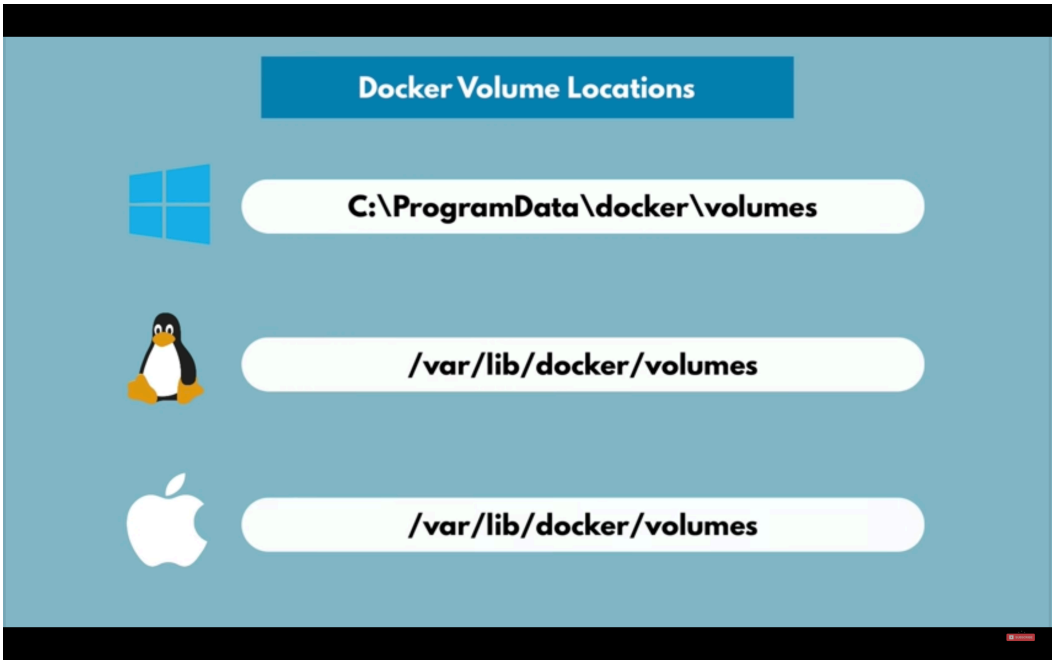
Using Docker Compose :





Here the volume define at service level and used in the image level.

Data Volume Location



Docker for Mac create a linux virtual machine and store all the docker data here.