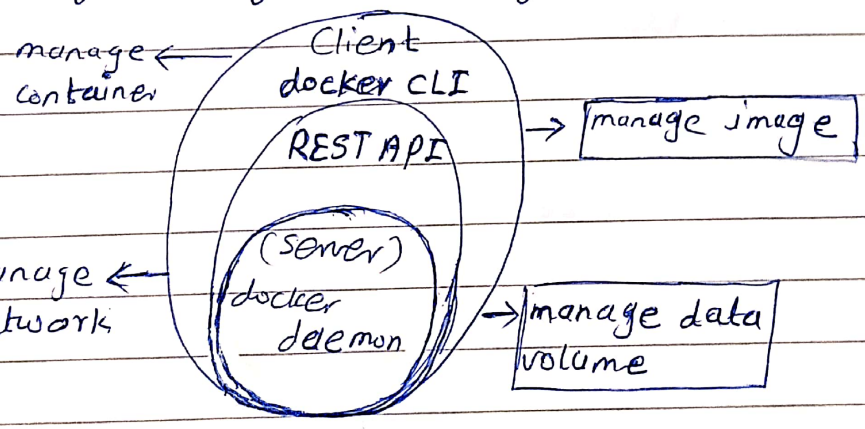
⊛ Docker

Date 28-6-24

- what is docker?
- docker is a platform designed to help developers build, share & run container applications.
- it is provide lightweight, portable & consistent environment for running application.
- each docker container includes the app & all its dependencies
- no more installing things manually
- easy to testing & debugging. run different version of software without affecting your system.
- build once, run anywhere
- portability : docker image run any OS
- docker integrates easily with jenkins, github Action, gitlab CI etc

* Docker engine :

- it is core component of platform, responcible for creating, running & managing docker containes.



* Docker image :

- A docker image is a lightweight, stand-alone & executable software package that includes everything needed to run a piece of software, such as code, runtime, libraries, environment variable & configuration files.

* Components of docker image:

1) Creation: image are created using the docker build command, which processes the instructions in a dockerfile to create the image layers.

2) Storage: image are stored locally the host machine.

* components of docker image:

1) Base image: The starting point for building an image. It could be minimal OS image like alpine alpine, a full fledged OS like ubuntu, or even another application image like python or node.

2) Application code: The actual code & files necessary for the application to run.

3) dependencies: libraries, frameworks & packages required by the application.

4) metadata: Information about the image, such as environment variables, labels & exposed ports.

* Docker image lifecycle

1) Creation: Image are created using the docker build command, which pocesses the instruction in a docker file to create the image layers.

2) Storage: Image are stored locally on the host machine

3) distribution: image can be shared by pushing them to a docker registry registry, allowing to pull and use the same image

4) Execution: image are executed by running containers, which are instance of these images.

* Dockerfile : (text file)
- It is the set of instruction that tell to docker daemon to build docker image
- key components of dockerfile :
1) Base image : specifies the starting point for the image, which could be a minimal OS, a specific version of a language runtime, or another image.
(ex : ubuntu : 20.04.01
2) label : Add metadata to image, such as version, description or maintainer.
(ex: LABEL version = "1.0.0" description = "My Application"
3) Run Commands (RUN) : execute commands in the image during the build process, typically used to install software package.
ex: RUN apt-get updates && apt-get-install-y python3
4) Copy files (COPY) : copies files or directories from the host system to the image.
ex: COPY ./app
5) Environment variables (ENV) : set environment variable in the image
ex: ENV PATH/app/bin : $PATH
6) Work directory (WORKDIR) : set of working directory for subsequent instructions.
ex: WORKDIR/app
7) Expose Ports (EXPOSE) : Informs docker that the container listens on specified network port
ex: EXPOSE 8080

8) Command (CMD): provide default command to run when the container starts
ex: CMD ["python", "app.py"]

9) Volume (VOLUME): create mount point with a specified path & mark it as holding externally mounted volumes from the host or other containers.
ex: VOLUME ["/data"]

10) Arguments (ARG): defines build-time variables
ex: ARG VERSION = 1.0

* Docker container:
- It is light-weight, portable & isolated environment that encapsulate an application & dependencies & allow to run consistently.
- It is create instance of the image & run the project.

* Registry:
- It is service that stores & distributes docker image
- it is act as repository where we push, pull & manage docker image. (it is ~~called~~ storage of image life cycle)
- docker hub is the most well-known public ~~reposi~~ registry

* key component:
1) Repositories:
- it is collection of docker image, you can store diff. version of same node with tages. tags representing the version of image.

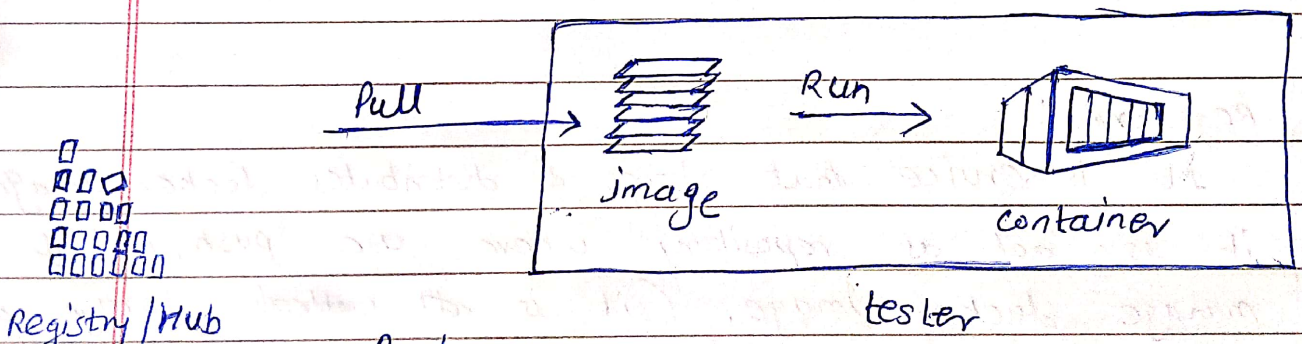
2) Tags: tags are used to version within a repository
ex: myapp:1.0, myapp:2.0

* Types of docker registry :

1) Docker Hub : It is public registry provided docker to access & by publicly & sharing anyone
URL : hub.docker.com

2) Private Registries : Custom registry set up by organizations to securely store & manage their own docker image

3) third - party registries :
- integrating with cloud platforms for seamless development and management of image within cloud infrastructure.
ex : Amazon - Elastic container Registry (ECR)
Google - google container Registry (GCR)
Azure - " container Registry (ACR)



* Use cases :

1) Microservices & architecture :
- breakdown the application into smaller, independent services, each running its own container.
Benefites : simplifies deployment, scalling & maintenance and each service can be update, developed & deployed independently.

2) Continuous integration & continuous deployment (CI/CD
- streamlines the CICD pipelines, reduces discrepacies betn environments & speeds up testing & deployment process.

3) Cloud migration : you can able to move your hosting one cloud platform to another using docker

4) Scalable web application : easy deploy & easy scaling
5) testing & QA : consistent environment to testing.
6) ML & AI :
7) API deployment & developement : deploying in container.

ex

```
#base image
 From python : 3.9
# Workdir
 WORKDIR /app
# copy
 COPY ./app
# Run
 RUN pip install -r requirement.txt
# Port
 EXPOSE 5000
# command
 CMD ["python", "./app.py"]
```

Dockerfile example