----------Docker Syntax meaning----------

A. curl -fsSL https://get.docker.com -o install-docker.sh

1. **curl** = "Client URL"

Internet se file download karne ya URL se data fetch karne ke liye use hota hai.

2.-**fsSL**

Ye 4 flags ka combined use hai:

- **-f** → Fail silently on HTTP errors

Agar URL 404, 500 jaisa error de, curl error message Nahi dikhaata Directly exit ho jata hai (good for scripting)

- **-s** → Silent mode

Download progress bar, messages hide karta hai Output clean rakhta hai

- **-S** → Show error

Jab -s use hota hai to errors hide ho jaate hain , -S bolta hai: "Silent rehna, but agar error aaye to dikhana"

Combination:

-sS = chup-chap download karo, par error aaye to batao

- **-L** → Follow redirects

Agar website URL redirect kare (HTTP 301/302, curl automatically new link follow Karega

- **get.docker.com**
  usually redirect hota hai actual script URL par — isliye important hai
- **-o install-docker.sh**

  -o = output file name

  Jo file download hogi, uska naam install-docker.sh rakhega

B. docker run -it ubuntu

Yeh command Ubuntu container ko interactive mode me start kar deta hai, jisme aap seedha terminal ke andar jaa kar commands chala sakte ho.

1) docker run

Naya container create karta hai

Agar image local me nahi hai, to pull (download) karta hai (Yahaan: ubuntu image)

2) -i (interactive)

Container ka STDIN (keyboard input) open rakhta hai

Matlab: aap container ke andar likh sakte ho

3) -t (allocate a pseudo-TTY)

Container ko ek terminal screen provide karta hai

ls, cd, apt, etc commands properly chalengi

User experience bilkul Linux terminal jaisa ho jata hai

-it ka combined effect:

➡️ Aap Ubuntu container ke shell ke andar jaa sakte ho

➡️ Jaise local Linux system me ho

4) ubuntu

Ubuntu image ka naam

Yeh latest Ubuntu Linux environment download + run karega

- **docker ps -a**

  Meaning: Ye command sabhi containers dikhati hai — running + stopped.

Use-case:

1. Kaunsa container stop hua?
2. Kaun fail hua?
3. Kaun run ho kar ruk gaya?
   Example:
   docker ps -a
   CONTAINER ID IMAGE STATUS
   a1b2c3d4e5 nginx Exited (0)
   f6g7h8i9j0 ubuntu Up 5 minutes

- **docker ps**

  Meaning: Sirf running containers list karta hai.

Use-case:

- Kaun container abhi active hai?
- Kya port pe run ho raha hai?

- **docker images**
  Meaning: System me jo images stored hai un sabki list.

Use-case:

- Kaun-si image kitni size ki hai?
- Old images delete karne ke liye dekhna ho.
- Kya image local me already available hai?

docker logs <container-id-or-name>

Meaning: Container ka output / error logs dikhaata hai.

Use-case:

- Container crash kyu hua?
- App ka output kya dikha raha hai?
- Debugging.

Ex. docker logs my-nginx

- **docker exec -it <container> /bin/bash**

Meaning: Running container ke andar jaane ke liye use hota hai.
-it = interactive terminal
/bin/bash = bash shell run karega

Use-case:

- Container ke filesystem me jaake debugging
- Files dekhna
- Configuration check

Example:

docker exec -it my-container /bin/bash
Note: Agar bash available nahi ho to /bin/sh use karna.

- **docker rm $(docker ps -aq)**

Meaning: Saare containers (running + stopped) delete kar deta hai.

Breakdown:

docker ps -aq = sabhi containers ka list

$() = unko input ke tarah use karo

docker rm = un sabko delete kar do

⚠ Warning:

Running containers pe error aa sakta hai.
Agar sabko force delete karna hai:

docker rm -f $(docker ps -aq)

- **docker rmi <image-id>**

Meaning: Local machine se image delete kar deta hai.

Use-case:

- Disk space free karna
- Old / unused images remove
- Clean-up

Example:

docker rmi nginx:latest

⚠ Agar koi container us image ko use kar raha ho — delete nahi hogi.

C. docker run --name docker-nginx -p 80:80 -d -v ~/docker-nginx/html:/usr/share/nginx/html nginx

- **--name docker-nginx**

  Container ka naam set karta hai:

  ➡ docker-nginx

Use-case:

- Easy to manage (logs, exec, stop, rm)
- IDs ke jagah naam use kar sakte ho
- Example:
  docker logs docker-nginx
  docker stop docker-nginx

- **-p 80:80**

  Ye port mapping hoti hai.

Format:

-p <host-port>:<container-port>
Meaning:

Host ka 80 → container ka 80

Matlab: browser me jab http://localhost open karoge,
request NGINX container ke port 80 tak phunchti hai.

Use-case:

Web server ko internet ya system se access karna

Default Nginx port 80 hota hai

- **-d** -d = detached mode

Meaning:

- Container background me run karega
- Terminal free ho jayega
- Aap logs use karke output dekh sakte ho

docker logs docker-nginx

- **-v ~/docker-nginx/html:/usr/share/nginx/html**

Ye volume / bind mount hai.

Format:

-v <host-path>:<container-path>
Meaning:

- Host folder: ~/docker-nginx/html
- Container folder: /usr/share/nginx/html

Important:
/usr/share/nginx/html = Nginx ka default web root

Matlab:

➡️ Host system me jo HTML files hongi,

➡️ Container Nginx wahi serve karega.

Benefits:

Code edit karo host me → container me auto update

No need to rebuild image

Useful for development

Example:
Host me:

~/docker-nginx/html/index.html
Aap browser me dekho:

http://localhost
Nginx usi file ko serve karega.

D. docker build -t mybuild19nov25

1. **docker build**
    1) Dockerfile se image banane ke liye use hota hai.
    2) Ye command Docker Engine ko bolta hai:
       "Is folder me Dockerfile read karke ek image banao."

2. -t mybuild19nov25

-t = tag

Meaning: Image ka naam set karna

Tag basically image ko identify karne ka naam hota hai

Example:

mybuild19nov25:latest

Agar tag specify nahi kiya gaya ho to default :latest lag jata hai.

Use-case:

1. Images ko easily identify karna
2. Versioning karna
3. Registry me push karna

Example:

docker run mybuild19nov25

3. Dot = build context

Meaning:

1. Current directory ko build context ke roop me Docker ko send kiya jaata hai
2. Yahi directory ke andar Dockerfile hota hai
3. Yahi se COPY, ADD commands files utha sakti hain

docker pull

docker pull <image-name>

docker pull nginx

docker pull ubuntu:22.04

1. docker pull ka matlab hota hai: Docker Hub ya kisi registry se image ko download karna.
2. Docker image = ek packaged software + its dependencies.

tags

Tag version ya variant hota hai.

Format:

<image>:<tag>

Dockerfile Syntax

1. FROM

Base image define karta hai (sab Dockerfile yahi se start hota hai).
FROM ubuntu:22.04
Meaning: Image is Ubuntu 22.04 ke upar banegi.

Use-case:

- OS choose karna (ubuntu, alpine, debian)
- Runtime choose karna (node, python, java)

2. **LABEL**
   Image me metadata add karta hai.
   LABEL maintainer="rohit@example.com"
   LABEL version="1.0"

   Use-case:
   - Authors, version, description add karna

RUN

Image build time me command execute karta hai.

RUN apt update && apt install -y nginx
Ye steps image ke layer me store ho jate hain.

Use-case:

- Packages install karna
- Directory create karna
- Config download Karna

COPY

Host machine se file container ke andar copy karta hai.

COPY index.html /usr/share/nginx/html/

Use-case:

App code add karna

Config files add karna

ADD

COPY jaisa hi hota hai, but extra features bhi deta hai.

ADD app.tar.gz /app/
ADD https://example.com/file.zip /downloads/

Extra features:

- Remote URL download
- Archive auto extract

NOTE:
Best practice → COPY use karo, ADD sirf jab zarurat ho.

WORKDIR

Container me current working directory set karta hai.

WORKDIR /app

Use-case:

- Commands yahin run honge
- COPY, RUN, CMD yahi path me execute honge

EXPOSE

Container ke andar ka port declare karta hai.

EXPOSE 3000

Meaning: Ye batata hai ki container internally 3000 port use karega

Actual port mapping run time me hoti hai
Example: **docker run -p 3000:3000**

ENV

Environment variables define karta hai.

ENV NODE_ENV=production
ENV APP_PORT=8080

Use-case:

- Config variables
- API keys (secure nahi → secrets mat rakho)

CMD

Container start hone par default command run karta hai.

CMD ["nginx", "-g", "daemon off;"]

Features:

- Sirf one CMD effective hota hai (last wins)
- Run time me override kar sakte ho:

docker run myapp echo hello

ENTRYPOINT

Container ka fixed startup command set karta hai.

ENTRYPOINT ["python3", "app.py"]

Difference vs CMD:

- ENTRYPOINT overwrite nahi hota easily
- CMD arguments de sakta hai

Example:

ENTRYPOINT ["ping"]
CMD ["google.com"]

Run:

docker run myimage
→ ping google.com

VOLUME

Volume define karta hai (data persist ke liye)

VOLUME ["/data"]
Use-case:

- Database files
- Logs

USER

Container ke andar ka default user set karta hai.

USER root
Production me: **USER nonroot**

ARG

Build time variable (runtime nahi).

ARG version=1.0
RUN echo $version

Build:

docker build --build-arg version=2.0 .
Use-case:

- dynamic builds
- versioning

📘 Small Example Dockerfile

FROM node:18-alpine
WORKDIR /app

COPY package.json .
RUN npm install

COPY . .

EXPOSE 3000
CMD ["npm", "start"]

Meaning:

Node image use karo

App directory set karo

Files copy karo

Dependencies install karo

Port expose karo

Start script run karo