1. What is Docker? / Docker म्हणजे काय?

Docker is a platform that allows you to **create, deploy, and run applications inside containers**. Containers are lightweight, portable, and ensure that your application runs consistently on any system.

Docker ही एक platform आहे जी तुम्हाला **containers मध्ये applications तयार, deploy आणि चालवायला** मदत करते. Containers हलके, पोर्टेबल आणि consistent असतात, म्हणजे तुमचे application कुठेही चालवले तरी त्याचा परिणाम सारखाच राहतो.

---

2. Why do we need Docker? / Docker ची गरज का आहे?

- Avoid "it works on my machine" problem.
- Applications run consistently in any environment.
- Faster deployment compared to virtual machines.
- Resource efficient.

- "माझ्या मशीनवर चालतं, तुमच्या मशीनवर का नाही?" हा problem टाळता येतो.
- Application कोणत्याही environment मध्ये सारखं चालतं.
- Virtual machines पेक्षा जलद deployment.
- कमी resources वापरतो.

---

3. Architecture of Docker / Docker चे Architecture

Docker has three main components:

1. **Docker Client:** User interacts with Docker (CLI commands).
2. **Docker Daemon:** Runs on the host machine, builds, runs, and manages containers.
3. **Docker Registry:** Stores Docker images (like Docker Hub).

---

4. Docker Flow / Docker चा flow

1. Developer writes application → creates **Dockerfile**
2. Docker builds **Image** from Dockerfile
3. Docker runs **Container** from Image
4. Application runs consistently in the container

5. What is a Dockerfile? / Dockerfile म्हणजे काय?

A **Dockerfile** is a text file that contains instructions to build a Docker image.

Example:

```
FROM ubuntu:20.04
RUN apt-get update
RUN apt-get install -y python3
COPY app.py /app.py
CMD ["python3", "/app.py"]
```

Dockerfile हा एक text file आहे ज्यात Docker image बनवण्यासाठी instructions दिल्या असतात.

---

6. What is Docker Registry? / Docker Registry म्हणजे काय?

A Docker registry stores **Docker images** so they can be shared. Example: Docker Hub.

Docker registry हे **Docker images** साठवण्याचं ठिकाण आहे, जे इतरांसोबत share करता येतात. उदा.: Docker Hub.

---

7. What is a Container? / Container म्हणजे काय?

A container is a **running instance of a Docker image**. It has all the necessary dependencies and configuration for application needs to run.

Container म्हणजे **Docker image चालू असलेला instance**. यात application चालवण्यासाठी सर्व काही असतं.

---

8. What is Docker Compose? / Docker Compose म्हणजे काय?

Docker Compose allows you to **define and run multi-container applications** using a YAML file.

Docker Compose तुम्हाला **multi-container applications** तयार आणि चालवायला मदत करतो YAML file वापरून.

9.Differece between Docker and Virtual machine

Feature	Docker	Virtual Machine (VM)
Architecture	Shares host OS kernel; runs in isolated containers	Runs full guest OS on top of hypervisor
Size	Lightweight (~MBs)	Heavy (~GBs)
Startup Time	Fast (~seconds)	Slow (~minutes)
Resource Usage	Low; shares host resources	High; each VM allocates full resources
Isolation	Process-level isolation	Full OS-level isolation
Portability	Highly portable	Less portable; depends on hypervisor
Performance	Near-native performance	Slightly slower due to full OS
Use Case	Microservices, Dev/Test, CI/CD	Legacy apps, full OS environments

10. What is Docker Image? / Docker Image म्हणजे काय?

A **Docker image** is a **read-only template** that contains everything needed to run an application – including the code, libraries, dependencies, and environment variables. You can create a container from a Docker image.

Docker image हा एक **read-only template** आहे ज्यात application चालवण्यासाठी आवश्यक असलेले सर्व काही असते – जसे की code, libraries, dependencies, आणि environment variables. Docker image वरून तुम्ही container तयार करू शकता.

11. What is Docker Hub? / Docker Hub म्हणजे काय?

Docker Hub is a **cloud-based registry service** where you can **store, share, and manage Docker images**. It allows developers to **download existing images** or **upload their own images** to share with others.

Docker Hub हे **cloud-based registry service** आहे जिथे तुम्ही **Docker images साठवू, share करू आणि manage करू** शकता. हे developers ला **existing images डाउनलोड** करण्याची आणि **स्वतःचे images अपलोड** करण्याची सुविधा देते.

---

12. What is Docker Daemon? / Docker Daemon म्हणजे काय?

Docker Daemon (`dockerd`) is the **background service** that runs on the host machine. It is responsible for **building, running, and managing Docker containers and images**. The daemon listens for requests from the Docker Client and performs the requested tasks.

Docker Daemon (`dockerd`) हा host machine वर चालणारा **background service** आहे. हे **Docker containers आणि images तयार, चालवणे आणि manage करणे** यासाठी जबाबदार असते. Daemon Docker Client कडून येणाऱ्या requests ऐकतो आणि त्या पूर्ण करतो.

---

13. What is Docker Client? / Docker Client म्हणजे काय?

Docker Client is the **command-line tool (CLI)** or interface that allows users to interact with the **Docker Daemon**. It is the main way you communicate with Docker — by typing commands like `docker build`, `docker run`, `docker ps`, etc.

Docker Client म्हणजे एक **command-line tool (CLI)** आहे ज्याच्या मदतीने user **Docker Daemon** शी संवाद साधतो. याच्याद्वारे आपण Docker ला commands देतो — जसे `docker build`, `docker run`, `docker ps` इत्यादी.

14. What is Docker Host? / Docker Host म्हणजे काय?

A **Docker Host** is the **physical or virtual machine** where the **Docker Daemon (dockerd)** runs. It provides the **environment** to build, run, and manage containers.
It includes the **Docker Daemon**, **containers**, **images**, **networks**, and **storage** used by Docker.

Docker Host म्हणजे ती **physical किंवा virtual machine** जिथे **Docker Daemon (dockerd)** चालतो. हे **environment** प्रदान करते ज्यात containers तयार, चालवले आणि manage केले जातात.
Docker Host मध्ये **Docker Daemon**, **containers**, **images**, **networks**, आणि **storage** यांचा समावेश असतो.

---

15. What is Docker Volume? / Docker Volume म्हणजे काय?

A **Docker Volume** is a **storage mechanism** used to **store data persistently** even after the container is stopped or deleted.
By default, container data is temporary (it is lost when the container is removed).
Using **volumes**, we can **save and share data** between multiple containers.

Docker Volume म्हणजे एक **storage mechanism** आहे जे **data कायमस्वरूपी साठवण्यासाठी** वापरले जाते, जरी container थांबवला किंवा delete केला तरीही डेटा टिकून राहतो.
सामान्यतः container मधील डेटा तात्पुरता असतो, पण **volume** वापरल्यास आपण **डेटा सेव्ह आणि शेअर** करू शकतो अनेक containers मध्ये.

---

16. Difference between Docker attach vs Docker Exec?

Aspect	docker attach	docker exec
Purpose	Reconnect to a container's **main process**	Run a **new command or shell** inside a running container
Affects main process?	✅ Yes — connects to it directly	❌ No — runs independently
Stops container if you exit?	⚠ Possibly (if signals are sent, e.g. Ctrl+C)	⚠ No — container keeps running
Multiple sessions allowed?	❌ No, only one attach per main process	✅ Yes, multiple exec sessions
Output shown	Shows **existing output** of main process	Shows **output of the new command**
Typical use case	Reattach to an interactive session you started earlier	Debug, inspect, or run commands in a running container

17. Difference between CMD & Entrypoint?

Feature	CMD	ENTRYPOINT
Purpose	Sets *default command* or *parameters* for the container	Sets *main command* that always runs
Can be overridden by docker run **arguments**	✅ Yes — running docker run image <args> replaces CMD	❌ No — docker run image <args> passes args *to* ENTRYPOINT
Preferred form	CMD ["executable", "param1", "param2"]	ENTRYPOINT ["executable", "param1", "param2"]
Shell form	CMD command param1 param2 (runs in /bin/sh -c)	ENTRYPOINT command param1 param2 (runs in /bin/sh -c)
Common use	Provide *default arguments* for ENTRYPOINT or as standalone command	Define *fixed executable* that alway

---

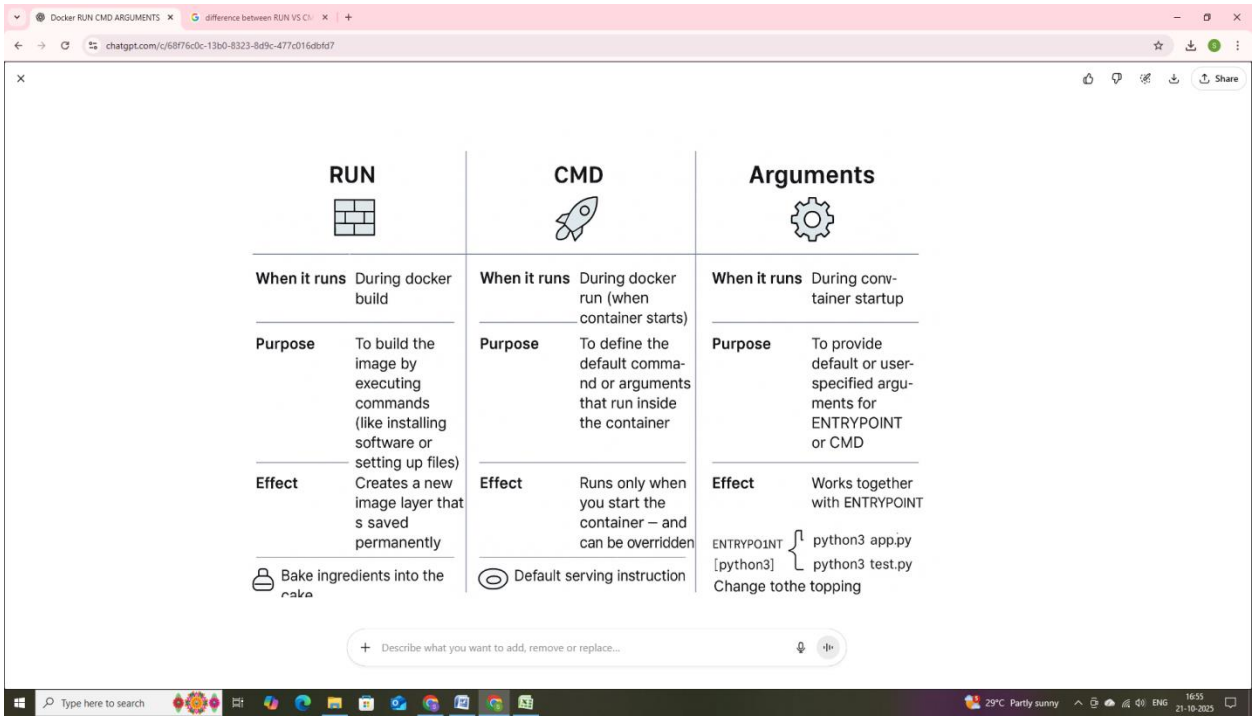
18. Difference between Expose and Publish

Feature	EXPOSE	Publish (-p / --publish)
Purpose	Documents which port the container *intends* to listen on	Actually maps a *container port* to a *host port*
Effect	Informational only (no real network change)	Makes the port accessible from outside the container
Where it's used	Inside the **Dockerfile**	When running the container (docker run)
Visibility	Only inside Docker network	Accessible from host or external machines
Example	EXPOSE 80	docker run -p 8080:80 nginx
Result	Tells Docker "this app listens on port 80"	Maps container's port 80 → host's port 8080

19. Difference between COPY VS ADD

Feature	COPY	ADD
Primary Purpose	Copy files and directories from the local build context into the image	Copy files and directories **plus** support extra features
Copies local files/folders	✅ Yes	✅ Yes
Automatically extracts compressed archives (.tar, .tar.gz, etc.)	❌ No	✅ Yes
Can fetch files from remote URLs	❌ No	✅ Yes
Behavior predictability	✅ Simple and explicit	⚠ Sometimes confusing (due to auto-extraction or downloads)
Performance	✅ Faster, minimal features	⚠ Slightly slower if extracting or downloading

19. Difference between RUN vs CMD vs Arguments



. Docker Version & Info Commands / Docker आवृत्ती आणि माहिती

Command	Description (English)	Description (Marathi)
`docker --version`	Shows Docker version	Docker ची आवृत्ती दाखवते
`docker version`	Shows client and server version	Client आणि Server दोन्हीची आवृत्ती दाखवते
`docker info`	Shows system-wide Docker information	संपूर्ण सिस्टमवरील Docker माहिती दाखवते

□ 2. Docker Images Commands / Docker Images साठी Commands

Command	Description (English)	Description (Marathi)
`docker images`	Lists all images	सर्व images दाखवते
`docker pull <image>`	Downloads image from Docker Hub	Docker Hub वरून image डाउनलोड करते
`docker push <image>`	Uploads image to Docker Hub	Image Docker Hub वर upload करते
`docker build -t <name> .`	Builds image from Dockerfile	Dockerfile वरून image तयार करते
`docker rmi <image>`	Removes image	Image delete करते
`docker tag <image> <repo>:<tag>`	Tags image with name and version	Image ला नाव आणि tag लावते
`docker history <image>`	Shows image build history	Image तयार करण्याचा इतिहास दाखवते
`docker inspect <image>`	Displays detailed info	Image बद्दल सविस्तर माहिती दाखवते

□ 3. Docker Container Commands / Docker Containers साठी Commands

Command	Description (English)	Description (Marathi)
`docker ps`	Lists running containers	चालू असलेले containers

Command	Description (English)	Description (Marathi)
		दाखवते
`docker ps -a`	Lists all containers (including stopped)	सर्व containers दाखवते (थांबलेलेही)
`docker run <image>`	Runs a new container	नवीन container चालवते
`docker run -d <image>`	Runs container in background (detached mode)	Container background मध्ये चालवते
`docker run -it <image> bash`	Runs container interactively	Container मध्ये interactively प्रवेश करते
`docker start <container>`	Starts a stopped container	थांबलेला container पुन्हा चालू करते
`docker stop <container>`	Stops a running container	चालू container थांबवते
`docker restart <container>`	Restarts container	Container पुन्हा सुरू करते
`docker pause <container>`	Pauses processes in container	Container मधील प्रक्रिया थांबवते
`docker unpause <container>`	Resumes paused container	Paused container पुन्हा सुरू करते
`docker rm <container>`	Removes container	Container delete करते
`docker logs <container>`	Shows container logs	Container चे logs दाखवते
`docker exec -it <container> bash`	Opens shell inside container	Container मध्ये shell उघडते
`docker inspect <container>`	Shows container details	Container ची सविस्तर माहिती दाखवते

□ 4. Docker Volume Commands / Docker Volume साठी Commands

Command	Description (English)	Description (Marathi)
`docker volume create <name>`	Creates new volume	नवीन volume तयार करते
`docker volume ls`	Lists all volumes	सर्व volumes दाखवते
`docker volume inspect <name>`	Shows volume info	Volume बद्दल सविस्तर माहिती
`docker volume rm <name>`	Removes volume	Volume delete करते

---

5. Docker Network Commands / Docker Network साठी Commands

Command	Description (English)	Description (Marathi)
`docker network ls`	Lists networks	सर्व networks दाखवते
`docker network create <name>`	Creates new network	नवीन network तयार करते
`docker network inspect <name>`	Shows network details	Network बद्दल सविस्तर माहिती दाखवते
`docker network connect <network> <container>`	Connects container to network	Container network शी जोडते
`docker network disconnect <network> <container>`	Disconnects container from network	Container network पासून वेगळे करते
`docker network rm <name>`	Removes network	Network delete करते

---

6. Docker System Commands / सिस्टम व्यवस्थापनासाठी Commands

Command	Description (English)	Description (Marathi)
`docker system df`	Shows disk usage by Docker	Docker चा disk usage दाखवते
`docker system prune`	Removes unused data	वापरात नसलेले डेटा delete करते
`docker system info`	Shows overall Docker system info	संपूर्ण Docker सिस्टम माहिती दाखवते
`docker stats`	Displays real-time container resource usage	Container चा live resource usage दाखवते

---

7. Docker Compose Commands / Docker Compose साठी Commands

Command	Description (English)	Description (Marathi)
`docker compose up`	Starts services defined in docker-compose.yml	docker-compose.yml मधील services चालवते
`docker compose down`	Stops and removes containers, networks, volumes	सर्व services थांबवते आणि delete करते
`docker compose`	Shows running services	चालू असलेल्या services दाखवते

Command	Description (English)	Description (Marathi)
`ps`		
`docker compose build`	Builds images defined in docker-compose.yml	Compose फाइलमधील images तयार करते
`docker compose logs`	Shows service logs	Services चे logs दाखवते

□ 8. Docker Cleanup Commands / Cleanup साठी Commands

Command	Description (English)	Description (Marathi)
`docker rm $(docker ps -aq)`	Removes all containers	सर्व containers delete करते
`docker rmi $(docker images -q)`	Removes all images	सर्व images delete करते
`docker volume prune`	Removes all unused volumes	न वापरलेले volumes delete करते
`docker network prune`	Removes all unused networks	न वापरलेले networks delete करते
`docker system prune -a`	Removes everything not in use	वापरात नसलेले सर्व काही delete करते