# Docker Commands – Practical Notes (From Your Session)

## 1. docker images

Command:
**docker images**

What it does:
Lists all Docker images available locally.

Why we use it:
• Check which images are downloaded
• Understand disk usage before running containers

## 2. docker system

Command:
**docker system**

What it does:
Displays Docker system subcommands.

Why we use it:
• Explore Docker system■level management commands

## 3. docker system df

Command:
**docker system df**

What it does:
Shows Docker disk usage for images, containers, volumes, and cache.

Why we use it:
• Identify space consumption
• Find reclaimable disk space

## 4. docker container run

```
Base command:
docker container run IMAGE_NAME

What it does:
Creates and starts a new container from an image.
```

## Using -d (Detached Mode)

Why we use it:
• Runs the container in the background

Example:

```
docker container run -d in28min/hello-world-java:0.0.1.RELEASE
```

## Using -p (Port Mapping)

Why we use it:
• Exposes container application to host machine

Syntax:
**-p HOST_PORT:CONTAINER_PORT**

Example:

```
docker container run -p 5002:5000 in28min/hello-world-java:0.0.1.RELEASE
```

## Using -m (Memory Limit)

Why we use it:
• Prevents container from consuming too much RAM

Example:

```
docker container run -m 512m in28min/hello-world-java:0.0.1.RELEASE
```

## Using --cpu-quota (CPU Limit)

Why we use it:
• Restricts CPU usage per container

Reference:
• 100000 = 1 full CPU core
• 50000 = 50% of one CPU core

Example:

```
docker container run --cpu-quota=50000 in28min/hello-world-java:0.0.1.RELEASE
```

## Using --name (Container Name)

Why we use it:
• Makes container identification easier

Example:

```
docker container run --name hello-java in28min/hello-world-java:0.0.1.RELEASE
```

## Using --rm (Auto Remove Container)

Why we use it:
• Automatically removes container after it stops

Example:

```
docker container run --rm in28min/hello-world-java:0.0.1.RELEASE
```

## Combining Multiple Flags (Real Usage)

Why we use it:
• Common in real projects to control networking and resources

Example:

```
docker container run -d -p 5002:5000 -m 512m --cpu-quota=50000 --name hello-java
in28min/hello-world-java:0.0.1.RELEASE
```

## 5. docker container ls

Command:
**docker container ls**

What it does:
Lists only running containers.

Why we use it:
• Check running containers and port mappings

## 6. docker container stop

Command:
**docker container stop <container_id>**

What it does:
Gracefully stops a running container.

Technical detail:
• Sends SIGTERM (allows cleanup)

## 7. docker container ls -a

Command:
**docker container ls -a**

What it does:
Lists all containers (running + stopped).

## 8. docker system info

Command:
**docker system info**

What it does:
Displays detailed Docker environment information.

## 9. docker system events

Command:
**docker system events**

What it does:
Displays real■time Docker events.