Multiple Containers in Dockers

* Can a Container Run on One Port?

- Each container has its own internal network & can listen on any number of port internally.

means Ex

```
docker run -d -p 3000:3000 container 1
docker run -d -p 3001:3000 container 2
docker run -d -p 3002:3000 container 3
```

3000 → internally

3002 → Host machine map outside container.

→ On the host machine, a port can be mapped to only one container at a time.

→ You can have multiple containers using the same container port internally but map them to different host ports.

* Key Docker Concepts

- Container :- Running instance of an image (isolated Environment).
- Image :- Blueprint for creating containers.
- Port Mapping :- (-p HostPort : containerPort) → connects a Host port to a container's internal port.
- Network :- Allows containers to communicat
- Volume :- Persistent storage between host and container.

* Common Multiple Container Use Cases

1. Same application, different instances (load balancing).
2. Microservices architecture (frontend, backend, database as seperate containers).
3. Scaling Horizontally :- multiple backend containers for handling for more request

*Methods to run Multiple Containers

A. Manual Run

Run containers one by one with unique host ports.

```
docker run -d --name backend1 -p 5000:5000 myapp
docker run -d --name backend2 -p 5001:5000 myapp
docker run -d --name backend3 -p 5002:5000 myapp
docker run -d --name backend4 -p 5003:5000 myapp.
```

*Pros: Simple, no extra cost

*Cons: Hard to manage many containers.

B. Docker Networks

= Internal container connect by network.

```
docker network create my-app-network
docker run -d --name frontend --network my-app-
  network -p 3000:3000 frontend app

docker run -d --name backend --network
my-app-network - p 5000:5000 backend app

docker run -d --name database --network
my-app-network mongo.
```

Inside Containers

- frontend → http://backend:5000
- Backend → mongodb://database:27017

Note:- Why frontend connect to backend & backend connect to the database only.

In docker networks, containers names become DNS hostnames as container name backend & database so docker will automatically resolve the connection problem.

(C) Docker Compose (.yml file)

- .yml → (YAML) is just a text file format used for configuration.

- In docker, we use `docker-compose.yml` to describe multiple containers services in one file.

- It replaces running many long `docker run -p` commands

Example `docker-compose.yml`

yaml

```yaml
version: "3.9"
services:
  frontend:
    build: ./frontend
    ports: - "3000:3000"
    depends_on: - backend

  backend:
    build: ./backend
    ports: - "5000:5000"
    depends_on: - database

  database:
    image : mongo
    ports : "27017:27017"
```

where to write this file

1. Create a file in your project folder called:
   `docker compose.yml`
2. Write the code in this
3. In the same folder, open terminal & run.
   `docker compose up -d`

* Version check when changing code

- When you change your application code and rebuild the image, always use version tags for clarity and rollback safety.
- Avoid using latest in production — use version number (V1.0, V1.1).

Example

```
docker build -t myapp:V1.1
docker run -d -p 3000:3000 myapp:V1.
```

- In docker-compose.yml, update the version tag when the image changes
- To see all versions you've built locally.

```
docker images ls
```

- If running containers still use the old image, restart with the new version.