

# Docker & Docker Compose

---

## 1. Docker Basics

- **What is Docker?**
  - Packages app + dependencies into containers.
  - Ensures app runs the same on all systems.
- **Why Docker?**
  - Solves “works on my machine” problem.
  - Faster setup for new developers.
  - Standardized deployments.
- **Architecture:**
  - **Image** → blueprint (Node, React, Mongo).
  - **Container** → running instance of an image.
  - **Docker Engine** → runs & manages containers.

---

## 2. Basic Commands

```
# Verify install
docker --version

# Run first container
docker run hello-world

# Run nginx web server
docker run -d -p 8080:80 nginx

# List containers
docker ps

# Stop & remove
docker stop <id>
docker rm <id>
```

```
# Remove image
docker rmi <image>
```

---

### 3. Dockerfile (Backend Example)

server/Dockerfile

```
FROM node:18-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 5000
CMD ["node", "index.js"]
```

• **Build & Run:**

```
cd server
docker build -t mern-backend .
docker run -p 5000:5000 mern-backend
```

---

### 4. Dockerfile (Frontend Example)

client/Dockerfile

```
FROM node:18-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 5173
CMD ["npm", "run", "dev"]
```

• **Build & Run:**

```
cd client
docker build -t mern-frontend .
docker run -p 5173:5173 mern-frontend
```

---

## 5. Docker Compose

- **Why Compose?**

- Run multiple containers (backend, frontend, db) with one command.
- Handles networking & volumes automatically.

- **docker-compose.yml Example:**

```
version: '3.8'
services:
  mongo:
    image: mongo
    volumes:
      - mongo-data:/data/db

  backend:
    build: ./server
    ports:
      - "5000:5000"
    environment:
      - MONGO_URI=mongodb://mongo:27017/dev_db
    depends_on:
      - mongo

  frontend:
    build: ./client
    ports:
      - "3000:5173"
    environment:
      - VITE_API_URL=http://localhost:5000/api

volumes:
  mongo-data:
```

- **Commands:**

```
docker-compose up --build -d
docker-compose ps
docker-compose logs backend
docker-compose down
```

## 6. Applying in MERN Project

- **Backend connects to Mongo inside Compose:**

```
MONGO_URI=mongodb://mongo:27017/dev_db
```

- **Frontend connects to backend via env:**

```
VITE_API_URL=http://localhost:5000/api
```

- **Run all at once:**

```
docker-compose up --build
```

---

✅ **Summary:** Docker standardizes app environments, and Docker Compose orchestrates multi-container MERN apps with one config file.