Docker Tutorial — Lesson Notes

What is Docker?

Docker is a platform for building, shipping, and running applications in lightweight, portable containers. Containers package code and all its dependencies so the app runs consistently everywhere.

Core Concepts

Concept	Description
Image	A read-only template with instructions for creating a container
Container	A running instance of an image
Dockerfile	A script with instructions to build an image
Registry	A repository for Docker images (e.g., Docker Hub)
Volume	Persistent data storage outside the container

Dockerfile Example

```dockerfile
FROM python:3.11-slim

WORKDIR /app

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

EXPOSE 5000

CMD ["gunicorn", "-b", "0.0.0.0:5000", "app:create_app()"]
```

Essential Commands

```
# Build an image
docker build -t myapp:1.0 .

# Run a container
docker run -d -p 8080:5000 --name web myapp:1.0

# List running containers
docker ps

# View logs
docker logs web

# Stop and remove
docker stop web && docker rm web

# Enter a running container
docker exec -it web bash
```

Docker Compose

```
version: "3.8"
services:
  web:
    build: .
    ports:
      - "5000:5000"
    environment:
      - DATABASE_URL=postgresql://user:pass@db/mydb
    depends_on:
      - db
  db:
    image: postgres:15
    volumes:
      - pgdata:/var/lib/postgresql/data
volumes:
  pgdata:
```

Key Takeaways

1. Containers are not VMs — they share the host OS kernel and are much lighter.
2. Use multi-stage builds to keep images small.
3. Docker Compose simplifies multi-container setups.
4. Never store secrets in images — use environment variables or secrets managers.
