Docker Compose

Docker Compose

- Manages building/running docker images/containers
- Build and run with one command
  - docker compose up --build --force-recreate
- No need to use docker build and docker run
- Will be used later to manage multiple containers
  - Separate container for your database

- Let's walk through a docker-compose.yml file

Docker Compose

docker-compose.yml

```yaml
version: '3'
services:
  app:
    build: .
    ports:
      - '8080:8000'
```

Docker Compose

docker-compose.yml

```yaml
version: '3'
services:
  app:
    build: .
    ports:
      - '8080:8000'
```

- Specify the docker compose file format version
- Version 3 is the current latest version

Docker Compose

docker-compose.yml

```yaml
version: '3'
services:
  app:
    build: .
    ports:
      - '8080:8000'
```

- List all of the services for docker compose to run
- A docker container is created for each service
- Later, we will create a second service for our database

Docker Compose

docker-compose.yml

```yaml
version: '3'
services:
  app:
    build: .
    ports:
      - '8080:8000'
```

- Name each service
- We have one service that we name "app"
- This name becomes the hostname when communicating between containers
  - This will be important when we add a database

Docker Compose

docker-compose.yml

```yaml
version: '3'
services:
  app:
    build: .
    ports:
      - '8080:8000'
```

- Use 'build' to specify the path to build from
  - Docker Compose will look in this directory for a Dockerfile and use it to build the image
- Same as the trailing '.' when building an image

Docker Compose

docker-compose.yml

```yaml
version: '3'
services:
  app:
    build: .
    ports:
      - '8080:8000'
```

- Map a local port to a container port
- Same as using "-p 8080:8000" when running a single container

Docker Compose

docker-compose.yml

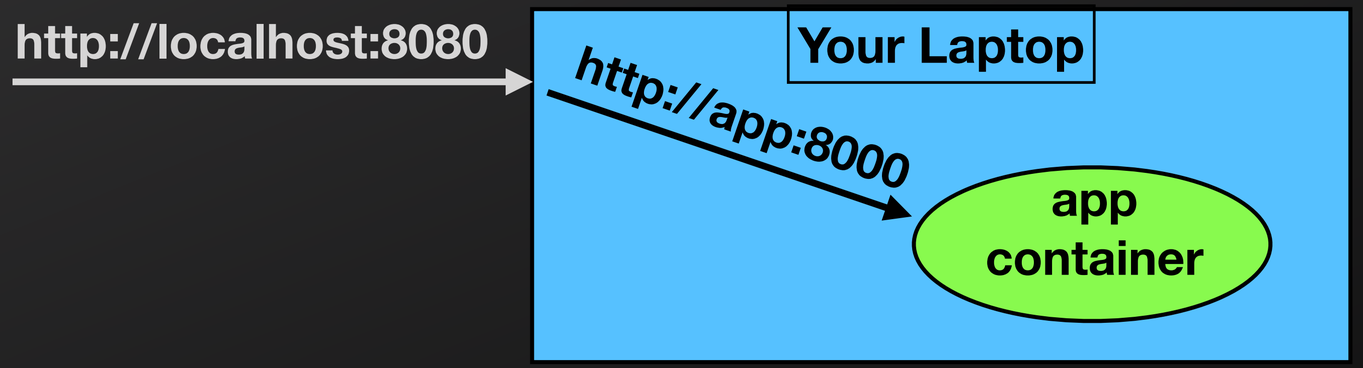
```yaml
version: '3'
services:
  app:
    build: .
    ports:
      - '8080:8000'
```

- Mapping a port allows your app [inside the container] to be accessed from your machine
- This line maps your local port 8080 [On your machine] to port 8000 inside your container

Docker Compose

docker-compose.yml

```yaml
version: '3'
services:
  app:
    build: .
    ports:
      - '8080:8000'
```



- When your machine receives a request for the mapped port
  - Docker forwards the request to the container on the specified port

Docker Compose

docker-compose.yml

```yaml
version: '3'
services:
  app:
    build: .
    ports:
      - '8080:8000'
```

- This file is then used to build an image and run the container using docker compose

Running Your App

- To run your app
  - docker compose up
- To run in detached mode
  - docker compose up -d
- To **rebuild** and restart the containers
  - docker compose up --build --force-recreate
  - *This is the best command to use!
- To restart the container without rebuilding
  - docker compose restart

Running Your App

- To **rebuild** and restart the containers
  - docker compose up --build --force-recreate

- Use this command during development
- Very important to rebuild your images after you change code
  - If you don't, you will not see your changes since you'll be running your old code

Demos