Big Data, organisation and analysis

Building a REST API service part II

Steffen M. Noe, Spring 2023

Recap on building a REST API
Docker

- Docker has a 2-stage build system

- Step 1: Create/Build the Docker image

```
docker build -t <tag_image_name> <directory_with_Dockerfile>
```

- Step 2: Create a Docker container

```
docker run —name <container_name> -p <nn:mm> -d <tag_image_name>
```

Docker build

- The build process relies on the Dockerfile

```
Dockerfile > ...
  1    FROM python:3.10-slim
  2
  3    # set the working directory
  4    WORKDIR /code
  5
  6    # install dependencies
  7    COPY ./requirements.txt ./
  8    RUN pip install --no-cache-dir --upgrade -r requirements.txt
  9
 10    # copy the src and data to the folder
 11    COPY ./src ./src
 12    COPY ./data ./data
 13
 14    # start the server
 15    CMD ["uvicorn", "src.main:app", "--host", "0.0.0.0", "--port", "80", "--reload"]
```

Parent system from Dockerhub

Landing point inside the container

Copy into container and install dependencies

Copy python source code and necessary data into the container

Starting the app using uvicorn inside the container

Docker build

The image has its own filesystem

Dockerfile and source
on local computer

Container image

Dockerhub hosts
parent images

```
Dockerfile
1    FROM python:3.10-slim
2
3    # set the working directory
4    WORKDIR /code
5
6    # install dependencies
7    COPY ./requirements.txt ./
8    RUN pip install --no-cache-dir --upgrade -r requirements.txt
9
10   # copy the src and data to the folder
11   COPY ./src ./src
12   COPY ./data /data
13
14   # start the server
15   CMD ["uvicorn", "src.main:app", "--host", "0.0.0.0", "--port", "80", "--reload"]
```

```
src > main.py
1    # Application that implements a REST API
2
3    from fastapi import FastAPI
4
5    import pandas as pd
6    import io
7
8    # import a file responder
9    from fastapi.responses import FileResponse
10
11   # import a streaming responder
12   from fastapi.responses import StreamingResponse
13
14   # Link the app with the REST API
15   app = FastAPI()
16
17   # default endpoint, returns a message
18   @app.get("/")
19   def read_root():
20       return {"Hello": "World"}
```

```
requirements.txt
1    fastapi
2    pydantic
3    uvicorn
4    pandas
5    redis
6    debugpy
```

Files in the container:

```
/-+-root/
  .
  |
  +-code/+-src/-main.py
  |       +-data/
  |       +-
  .
  +-lib/…
```

Docker run

Create (instantiate) the container from the image

Container image

Files in the container:

```
/-+-root/
  .
  |
  +-code/+-src/-main.py
  |      +-data/
  |      +-
  .
  +-lib/...
```
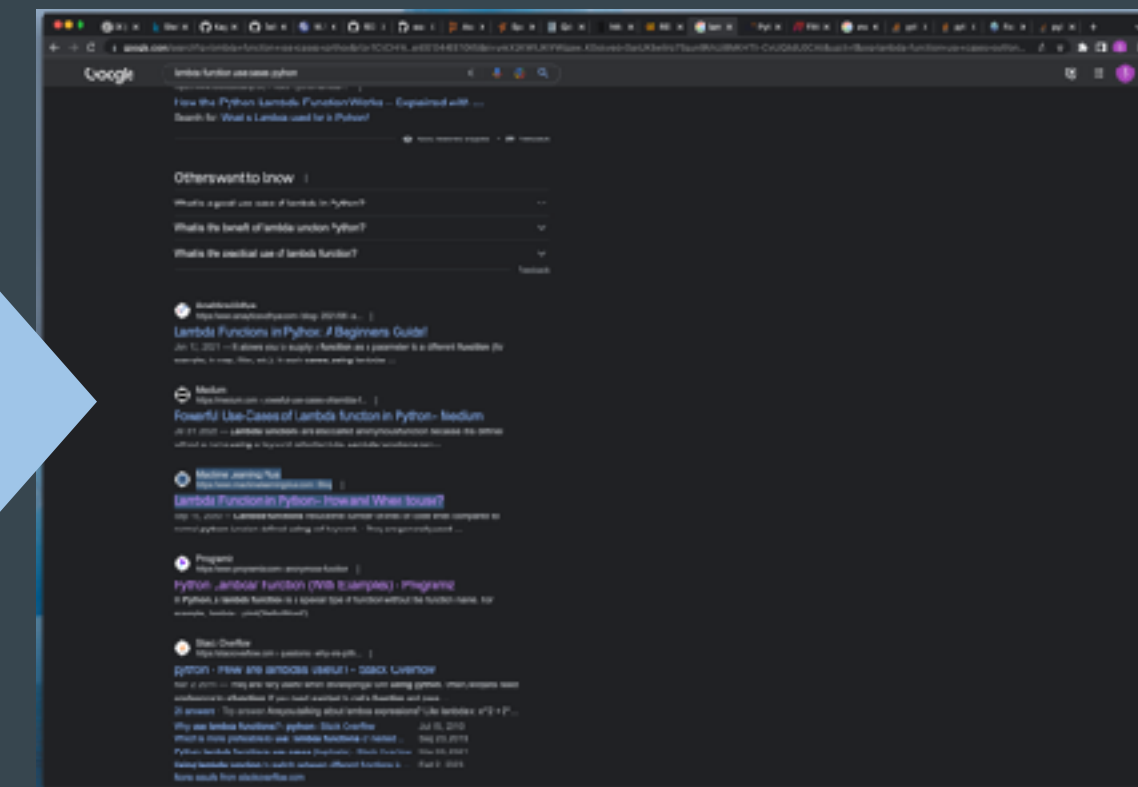
`docker run ...`

Container

serve REST API
on port 80

http://0.0.0.0/
http://0.0.0.0/data/
http://0.0.0.0/pandas/

this IP because we "simulate" a
network on our own computer!

Webbrowser

Docker commands we need

Docker images

```
docker build -t <tag_image_name> <directory_with_Dockerfile>
```

```
docker images
```
show images

```
docker image rm
```
remove images

Docker containers

```
docker run --name <container_name> -p <nn:mm> -d <tag_image_name>
```

```
docker ps
```
show running containers

```
docker ls
```
show running containers

```
docker ls -a
```
show all containers (running and non-running)

```
docker stop <container_name>
```
stop containers

```
docker rm <container_name>
```
remove containers