Docker for Machine Learning

- First steps – just use image
- More advanced – modify a public image
- More realistic – use (changed) images together

Just use image

Sources for Jupyter Notebooks

- https://hub.docker.com/u/jupyter
- https://github.com/jupyter/docker-stacks
- https://github.com/Kaggle/docker-python
- ...

Let‘s pick a universal one:

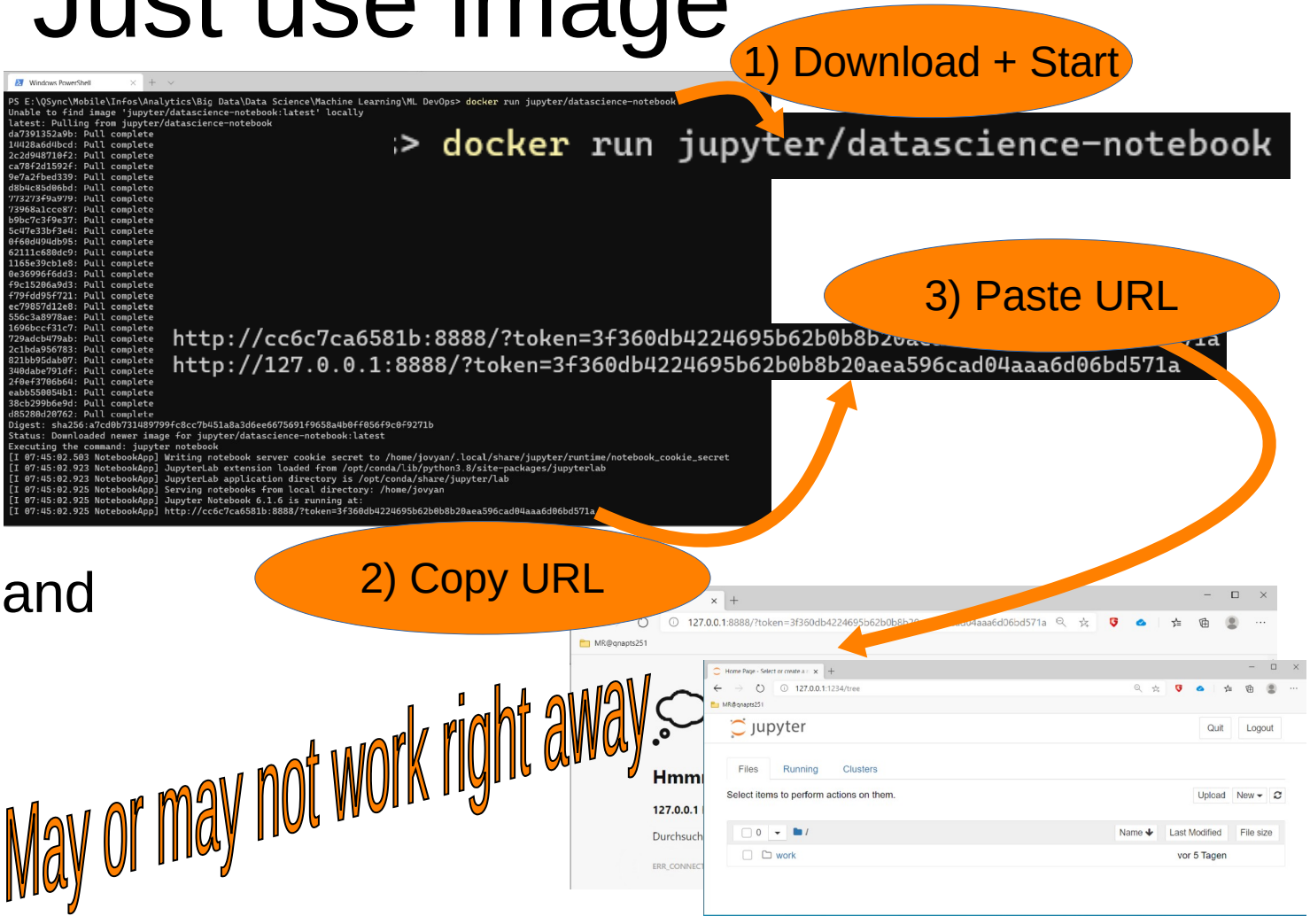
https://hub.docker.com/r/jupyter/datascience-notebook

Just use image

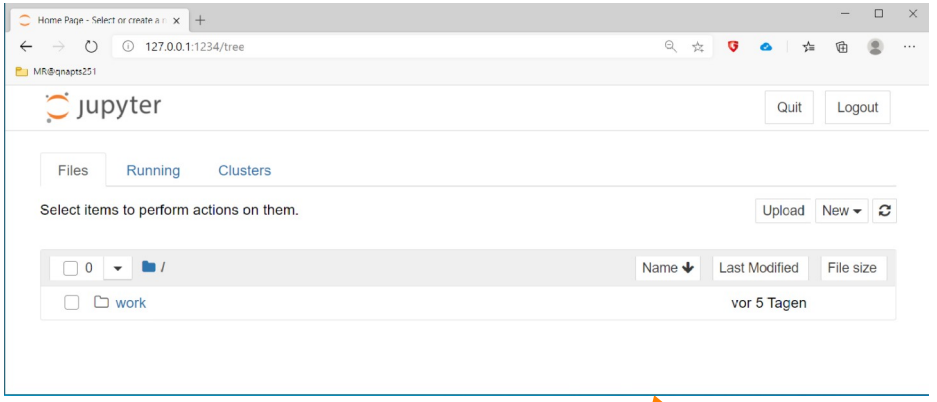
Steps

- Pull Image
- Run it

Can be one command



Just use image



Troubleshooting

1. Stop running container
   - `<Ctrl+C>`
1. Remove running container
   - `docker container rm -f <mycontainer>`
2. Restart container with parameter(s)
   - `docker run -p 1234:8888 jupyter/datascience-notebook`

Just use image

Working with Container

- Name container
  - `docker run -p 1234:8888 --name jupyter_ds --rm jupyter/datascience-notebook`
- Use data files (*.csv, etc.)
  - `docker cp some.csv jupyter_ds:/home/jovyan/`
- Persist *.ipynb, etc. outside container
  - `docker run -p 1234:8888 -v C:\Users\Mike_64\Documents\MLOps_data:/home/jovyan --rm --name jupyter_ds jupyter/datascience-notebook`

Modify a Public Image

Create individual Docker image with Dockerfile:

- Pull an image
- Customize it
- Save it to own Docker registry
- Run containers of the modified image

Modify a Public Image

Steps

1)Code Dockerfile
2)Build image
3)Run container off image

```
FROM jupyter/datascience-notebook

#base image is Ubuntu (https://g...                    /jupyter/docker-stacks/issues/949)
#to install, user needs to be root
USER root

#update base packages
RUN apt-get update && \
    apt-get --yes install \
    apt-utils \
    vim

#adding Python packages
RUN pip install mysql.connector sqlalchemy

#preparing Notebook environment
#RUN mkdir /home/jovyan/.jupyter/
COPY jupyter_notebook_config.py /home/jovyan/.jupyter/

#back from roo
USER $NB_UID
```

1

```
PS E:\QSync\Mobile\Infos\Analytics\Big Data\Data Science\Machine l
[+] Building 0.2s (9/9) FINISHED
 => [internal] load build definition from Dockerfile                                    0.0s
 => => transferring dockerfile: 616B                                                    0.0s
 => [internal] load .dockerignore                                                       0.0s
 => => transferring context: 2B                                                         0.0s
 => [internal] load metadata for docker.io/jupyter/datascience-notebook:latest          0.0s
 => [1/4] FROM docker.io/jupyter/datascience-notebook                                   0.0s
 => [internal] load build context                                                       0.0s
 => => transferring context: 49B                                                        0.0s
 => CACHED [2/4] RUN apt-get update &&     apt-get --yes install  apt-utils     vim    0.0s
 => CACHED [3/4] RUN pip install mysql.connector sqlalchemy                             0.0s
 => [4/4] COPY jupyter_notebook_config.py /home/jovyan/.jupyter/                        0.0s
 => exporting to image                                                                  0.1s
 => => exporting layers                                                                 0.0s
 => => writing image sha256:6935a67c44b3a450a135ec27fa4e50b4aa2e4e1771aa37c127de8e3719141cd0  0.0s
 => => naming to docker.io/library/mike_jn                                              0.0s
```

```
docker build -t mike_jn .
```

2

```
docker run -p 1234:1234 -v C:\Users\Mike_64\Documents:/home/jovyan --rm --name jupyter_ds mike_jn
```

```
[I 07:50:39.946 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 07:50:39.948 NotebookApp] Serving notebooks from local directory: /home/jovyan
[I 07:50:39.948 NotebookApp] Jupyter Notebook 6.1.6 is running at:
[I 07:50:39.948 NotebookApp] http://e60e0faba498:1234/?token=b1e99e5af04b7d7d5f21c1ca8e94bd315abe47a8f3eee8e2
[I 07:50:39.948 NotebookApp]  or http://127.0.0.1:1234/?token=b1e99e5af04b7d7d5f21c1ca8e94bd315abe47a8f3eee8e2
[I 07:50:39.948 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 07:50:39.959 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/jovyan/.local/share/jupyter/runtime/nbserver-7-open.html
    Or copy and paste one of these URLs:
        http://e60e0faba498:1234/?token=b1e99e5af04b7d7d5f21c1ca8e94bd315abe47a8f3eee8e2
```

3

```
http://127.0.0.1:1234/?token=b1e99e5af04b7d7d5f21c1ca8e94bd315abe47a8f3eee8e2
```

Modify a Public Image

Worth noting:

- Dockerfile name must be „Dockerfile“
- Docker build must be issued in directory of Dockerfile

Modify a Public Image

For best practices:
https://docs.docker.com/develop/develop-images/dockerfile_best-practices/
A good tutorial:
https://takacsmark.com/dockerfile-tutorial-by-example-dockerfile-best-practices-2018/

Use (Changed) Images Together

Pull / change multiple image to form „apps“:

- Pull images
- Customize them
- Save them to own Docker registry
- Run them as services with docker-compose

Use (Changed) Images Together

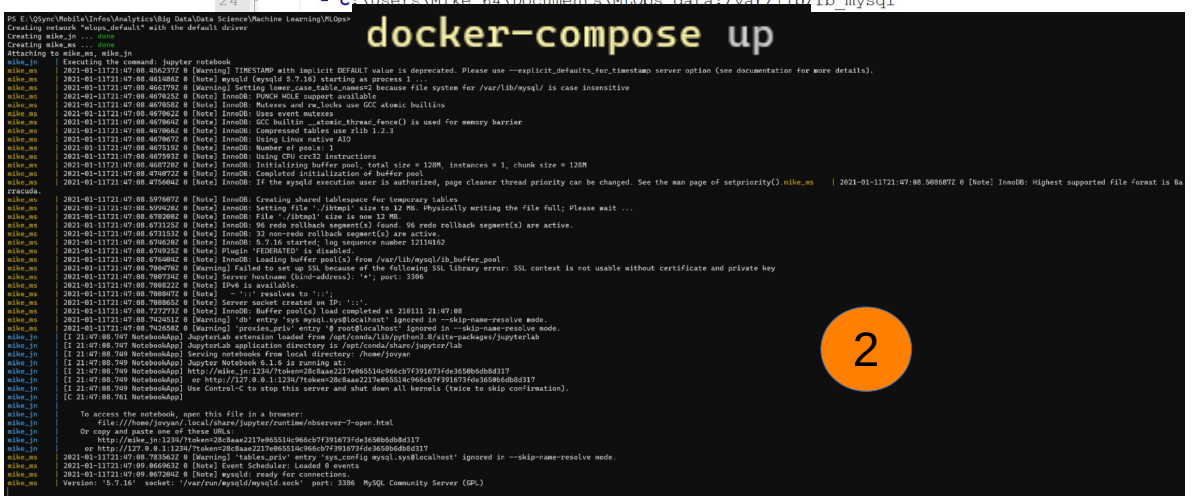
Steps

1)Code docker-compose-yml file

2)Run docker-compose

```yaml
version: '3'
services:
  mysql:
    image: mike_ms
    hostname: mike_ms
    container_name: mike_ms
    ports:
      # <Port exposed> : < MySQL Port running inside container>
      - '2345:3306'
    expose:
      # Opens port 3306 on the container
      - '3306'
      # Where our data will be persisted
      #restart: always
    environment:
      MYSQL_DATABASE: 'dsdb'
      # So you don't have to use root, but you can if you like
      MYSQL_USER: 'dsdb_user'
      # You can use whatever password you like
      MYSQL_PASSWORD: 'dsdb_user_pwd'
      # Password for root access
      MYSQL_ROOT_PASSWORD: 'root_pwd'
    volumes:
      - C:\Users\Mike_64\Documents\MLOps_data:/var/lib/fb_mysql
```

1



2

Use (Changed) Images Together

Worth noting:

- docker-compose file name: „docker-compose.yml“ (*.yaml)
- docker-compose up / down: issued in *.yml file directory
- Services may error out when started at once
  → start one service at a time:

  docker-compose up mysql
  docker-compose up jupyter
- `Docker-compose down` removes all containers
- Stop container(s) with
  `docker container mike_jn stop`

Use (Changed) Images Together

Working with services

- Name container
  - `docker run -p 1234:8888 -name jupyter_ds --rm jupyter/datascience-notebook`
- Use data files (*.csv, etc.)
  - `docker cp some.csv jupyter_ds:/home/jovyan/`
- Persist *.ipynb, etc. outside container
  - `docker run -p 1234:8888 -v C:\Users\Mike_64\Documents:/home/jovyan --rm --name jupyter_ds jupyter/datascience-notebook`