Docking TM5

Maarten Krol

WAGENINGEN UNIVERSITY
WAGENINGEN UR

SRON
Netherlands Institute for Space Research

Institute for Marine and
Atmospheric research Utrecht

First Python Notebooks, Now Docking? Come On!

- What is DOCKER?
- Docker Example: MOGUNTIA
- Docker possibilities: TM5

What is DOCKER?

www.docker.com

Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run:

- code
- runtime
- system tools
- system libraries
- anything else that can be installed on a server

This guarantees that the software will always run the same, regardless of its environment.

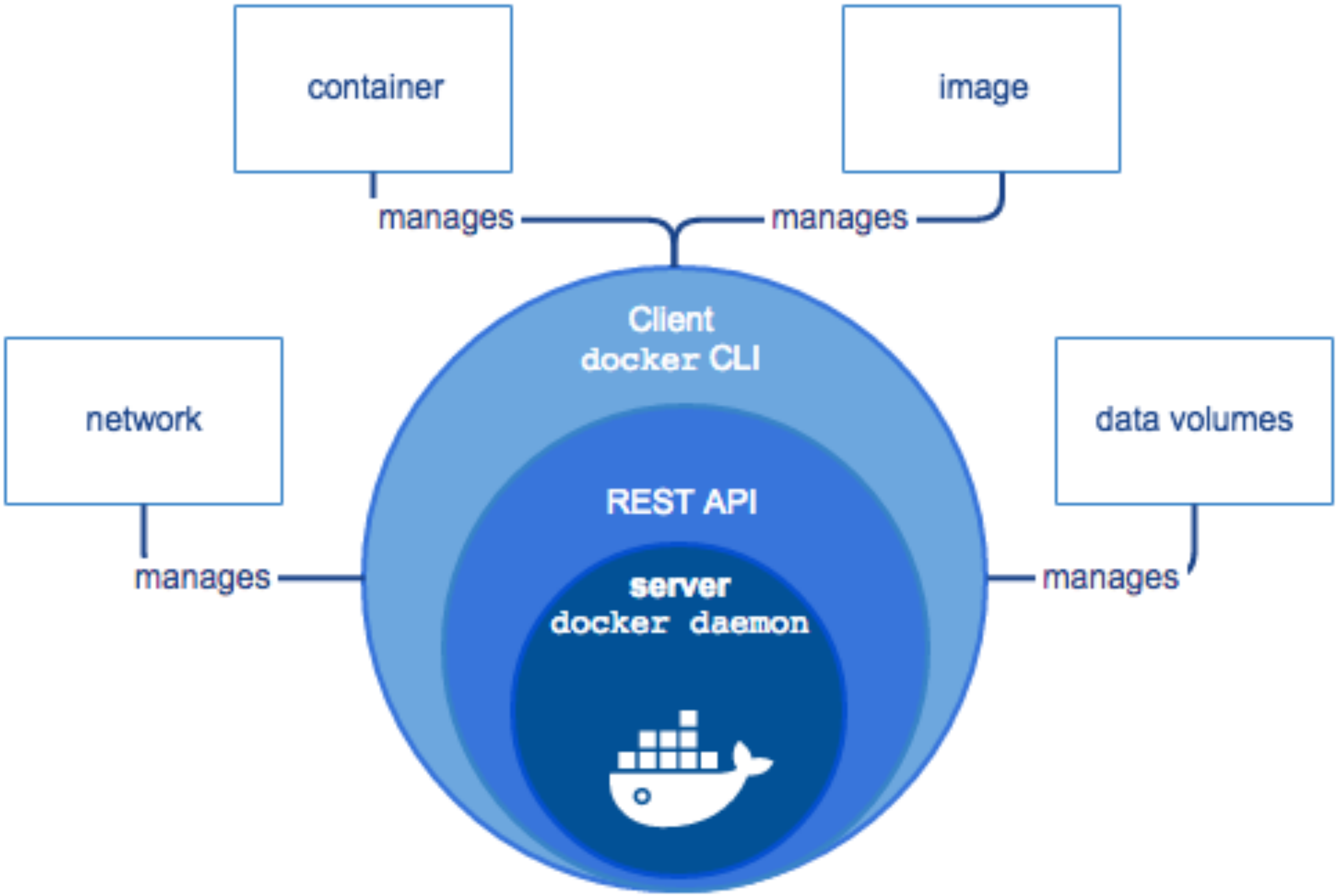
This is what we need for TM5!

What is DOCKER?

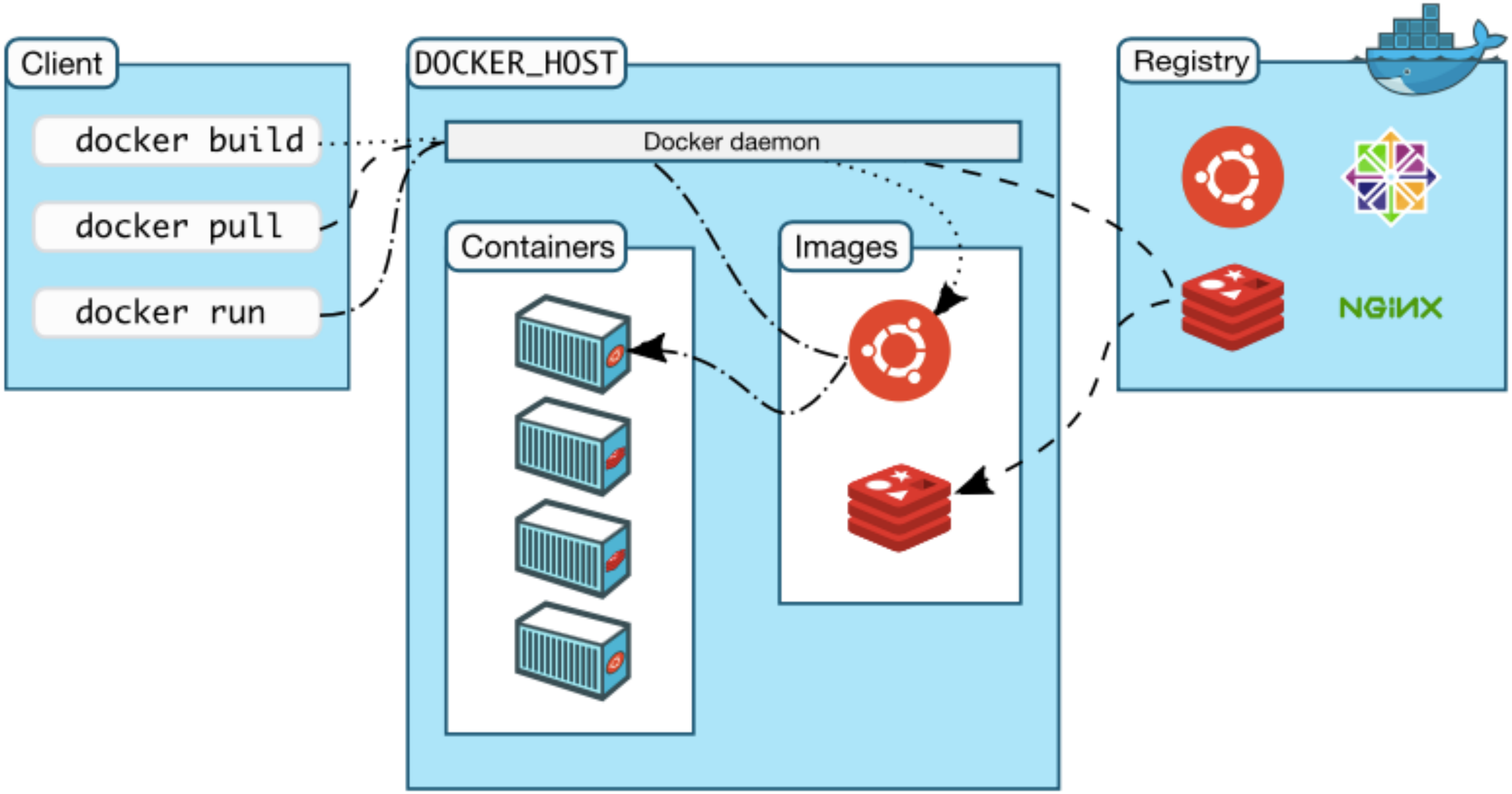
Well, this is all technical stuff….

And would this help TM5?

OK: lets investigate first how it works…..



Docker_host can be the same, or remote machine



WAGENINGEN UNIVERSITY
WAGENINGEN UR
SRON
Netherlands Institute for Space Research
Institute for Marine and
Atmospheric research Utrecht

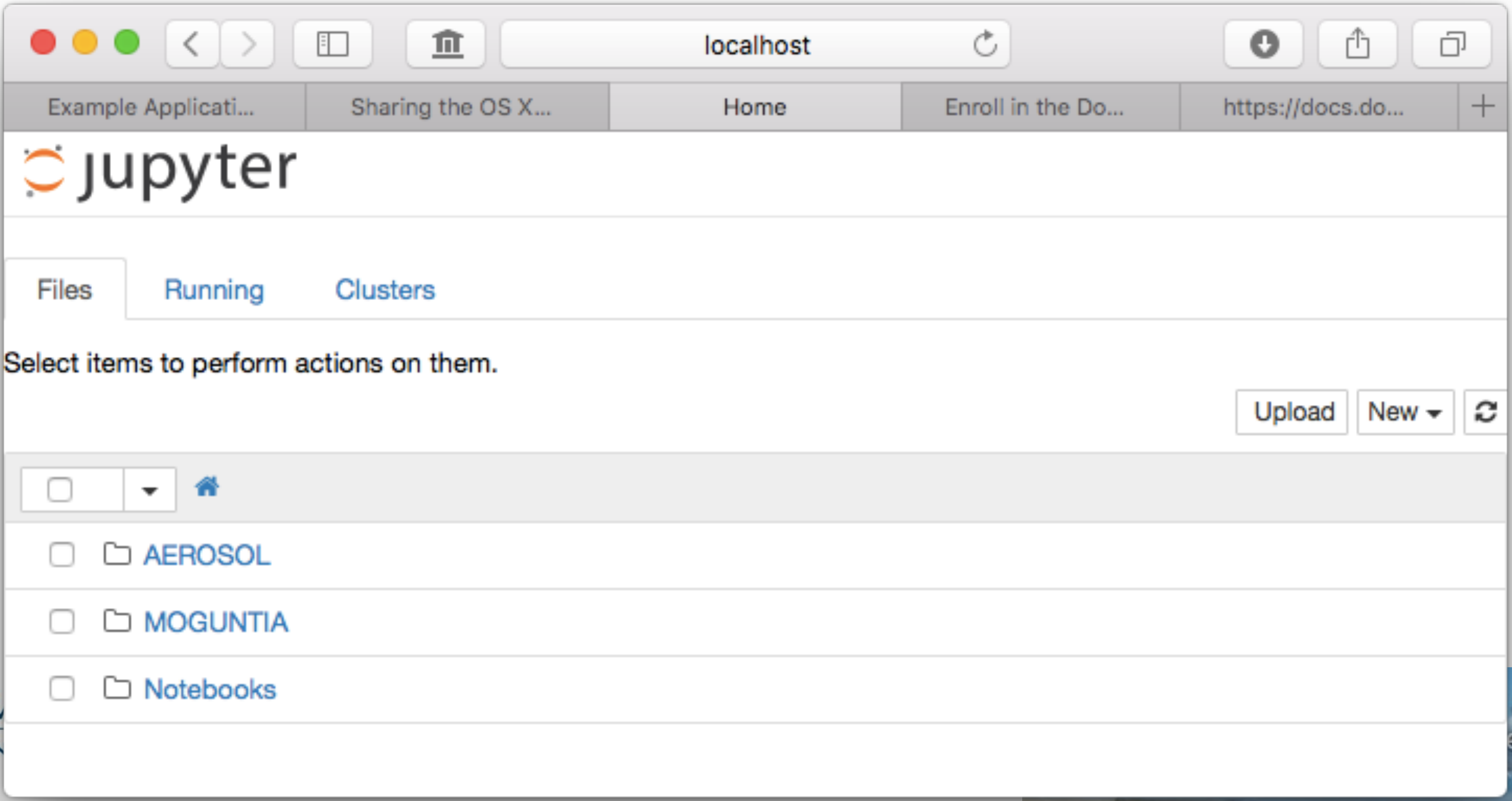
Some example….

Docker beta installed on MAC OS X
This contains a “VM” machine running natively
Below a container is started from my public domain docker cloud image “maartenkrol/notebooks-test”:

```
Maartens-MacBook-Air:Docker krol$ docker images
REPOSITORY                   TAG                 IMAGE ID            CREATED             SIZE
tm5                          latest              25bf5799174f        45 hours ago        5.924 GB
alpine                       latest              13e1761bf172        7 weeks ago         4.797 MB
maartenkrol/notebooks-test   latest              fcca36fcca39        7 weeks ago         5.804 GB


Maartens-MacBook-Air:Docker krol$ docker run -d -p 8888:8888 maartenkrol/notebooks-test
edee3115ba0acc46a2efa4b95bcd867340095497c98be13ac6716a8b25eb1f58

Maartens-MacBook-Air:Docker krol$ docker ps
CONTAINER ID        IMAGE                        COMMAND                   CREATED             
STATUS              PORTS                    NAMES
edee3115ba0a        maartenkrol/notebooks-test   "tini -- start-notebo"    25 seconds ago      Up 24
seconds        0.0.0.0:8888->8888/tcp   prickly_thompson
```

What does this do?

- A container will run on the native VM
- The container has python notebooks installed
- Jupyter server is enabled
- Container sends on port 8888 (port mapping)
- We can access the running container by a browser!

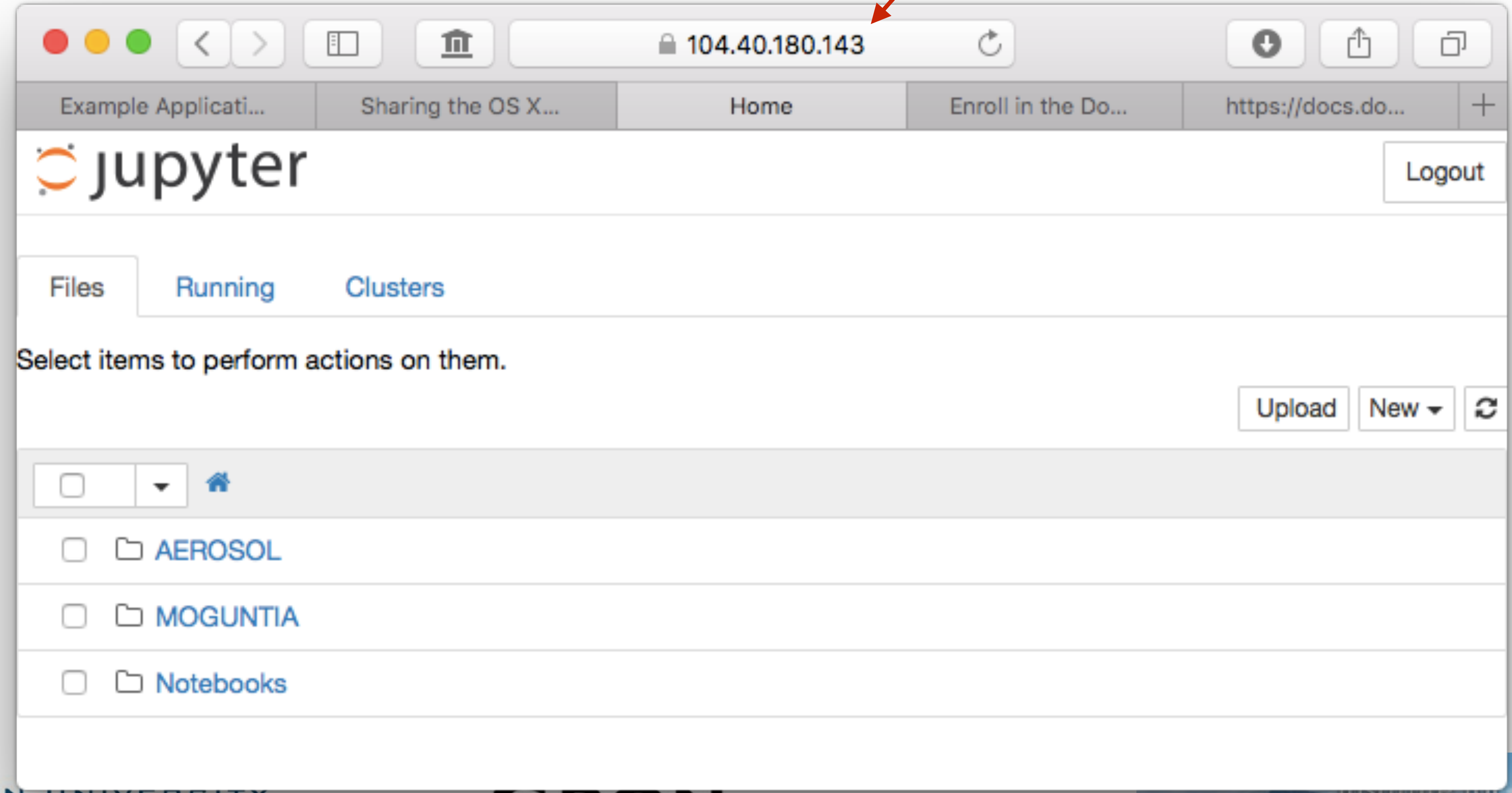


Select items to perform actions on them.

WOW!

Azure IP

- So you can test & develop locally…
- Later run it in the “cloud” in exactly the same way!
- E.g. on Microsoft AZURE
- Could this work for TM5?



Select items to perform actions on them.

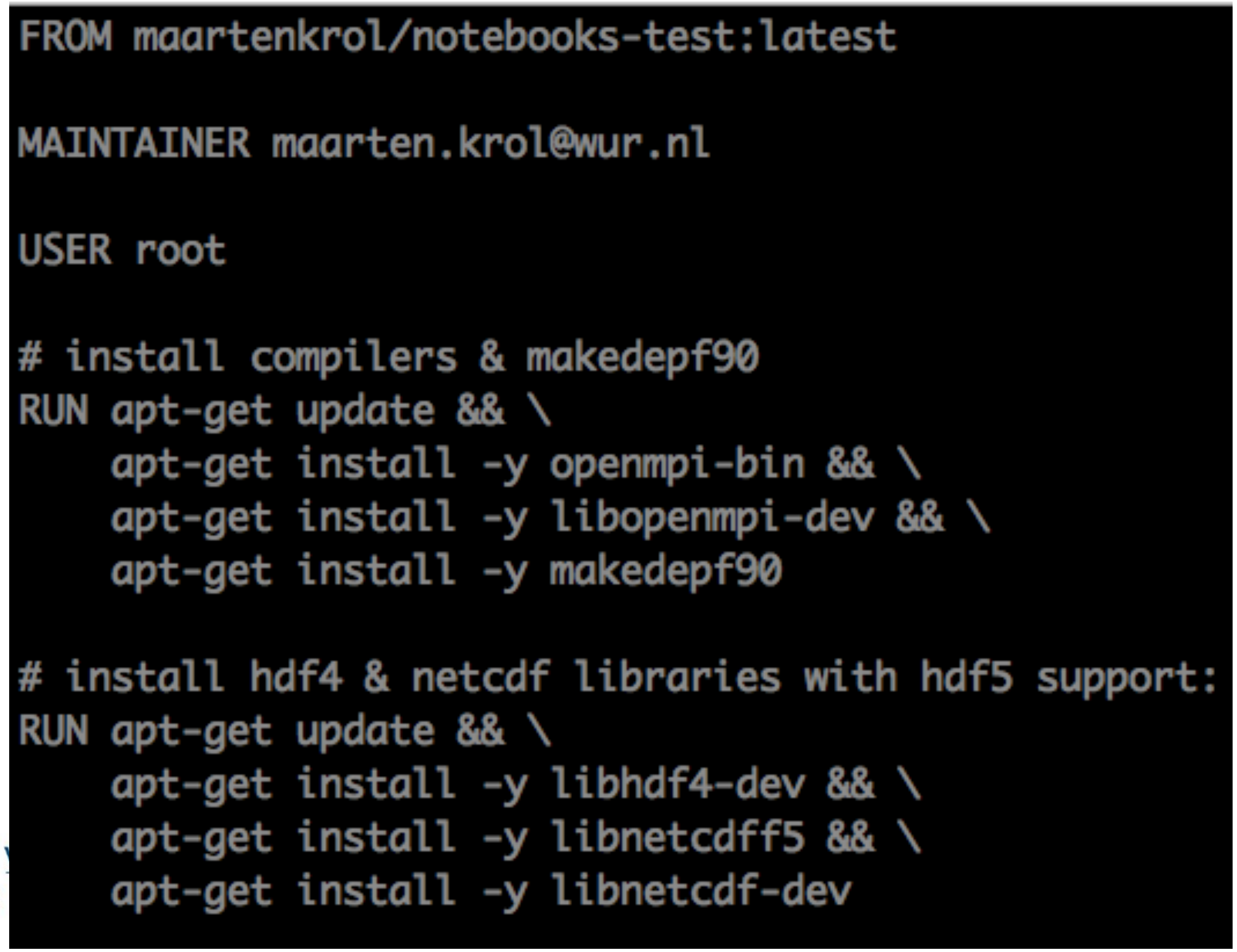
□ AEROSOL

□ MOGUNTIA

□ Notebooks

TM5: We will, we will, Dock You

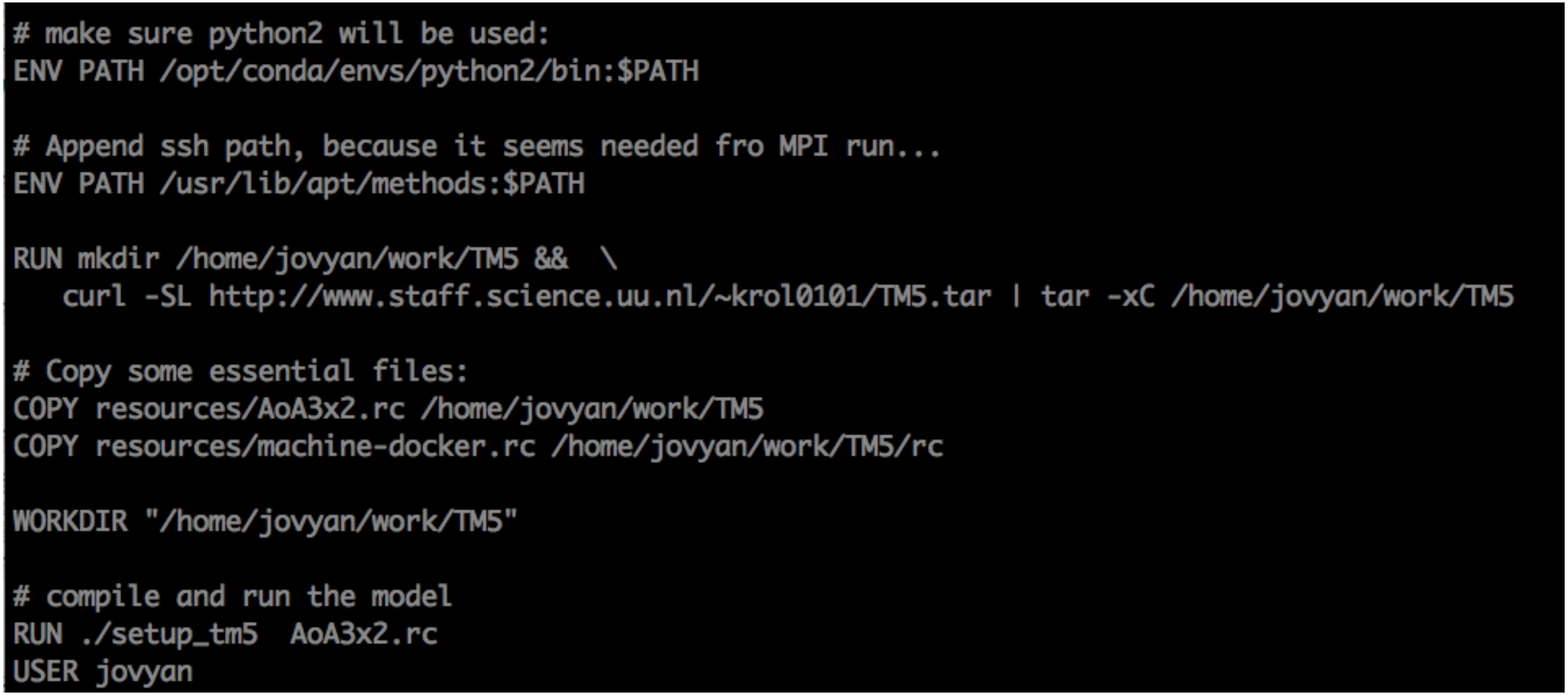
- “docker build” lets you build images
- You need a “Dockerfile” and type:
- >> docker build -t tm5 ./



```
FROM maartenkrol/notebooks-test:latest

MAINTAINER maarten.krol@wur.nl

USER root

# install compilers & makedepf90
RUN apt-get update && \
    apt-get install -y openmpi-bin && \
    apt-get install -y libopenmpi-dev && \
    apt-get install -y makedepf90

# install hdf4 & netcdf libraries with hdf5 support:
RUN apt-get update && \
    apt-get install -y libhdf4-dev && \
    apt-get install -y libnetcdff5 && \
    apt-get install -y libnetcdf-dev
```

Institute for Marine and
Atmospheric research Utrecht

Docking TM5



```
# make sure python2 will be used:
ENV PATH /opt/conda/envs/python2/bin:$PATH

# Append ssh path, because it seems needed fro MPI run...
ENV PATH /usr/lib/apt/methods:$PATH

RUN mkdir /home/jovyan/work/TM5 &&  \
   curl -SL http://www.staff.science.uu.nl/~krol0101/TM5.tar | tar -xC /home/jovyan/work/TM5

# Copy some essential files:
COPY resources/AoA3x2.rc /home/jovyan/work/TM5
COPY resources/machine-docker.rc /home/jovyan/work/TM5/rc

WORKDIR "/home/jovyan/work/TM5"

# compile and run the model
RUN ./setup_tm5  AoA3x2.rc
USER jovyan
```

WAGENINGEN UNIVERSITY
WAGENINGEN UR

SRON
Netherlands Institute for Space Research

Institute for Marine and
Atmospheric research Utrecht

WOW!!!

- This will build TM5 in an image that can run as a docker “container”
- Platform independent….
- Proper libraries can be installed by proficient users like.. Philippe, Arjo
- But how do we run a “tm5” container after build?

```
docker run -d  -p 8889:8888 \
-v /Users/krol/TESTDIR:/home/jovyan/work/TESTDIR tm5
```

- Maps port 8889—>port 8888 in container
- Mounts “/Users/krol/TESTDIR to dir in container

Good to know:

- You can access the running container by:

```
docker ps (get container id=xxxxxx)

docker exec -u root -it xxxxxx /bin/bash
```

- You stop the container by:

```
docker stop xxxxxxxx
```

DONE?…Critical reflection

Why on Earth do we want this?

- TM5 needs high performance platforms
- TM5 needs huge amounts of input files..

Possible Answers & Possibilities..

- New possibilities may become available also on HPC platforms (Docker is still quite new)
- New users can learn code, etc. before they go to high-performance
- Specific applications (CO_2 simulations, specific tracers, back-plumes, user-supplied emissions, ..)
- One of the platforms to test (different Linux OS's)