



Setting up Docker

Math+Econ+Code

Jules Baudet, jules.baudet99@gmail.com

January 2021

Support from the ERC CoG-866274 EQUIPRICE grant is acknowledged.

Table of contents

1. What is Docker?

2. Installing Docker

If you are already familiar with Docker, you can go directly to slide 8.

What is Docker?

What is Docker?



Docker is a technology that allows developers to:

1. Deliver softwares in bundles called **containers**
2. Make their application run smoothly on different OSs
3. Save a significant amount of computing resources

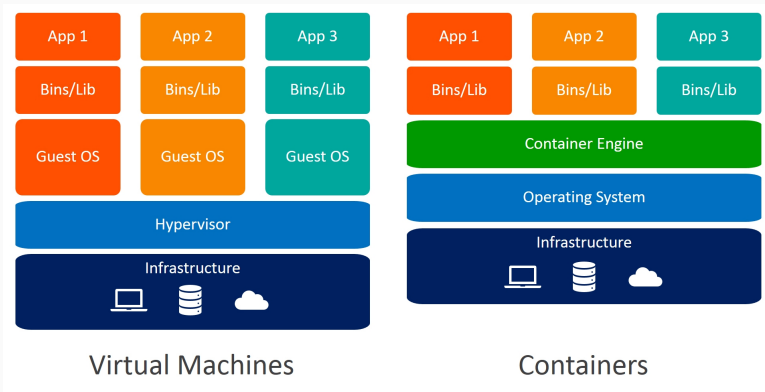
What is a container?

A container is a standard unit of software. It is a "package" containing an application and all of its dependencies (libraries, settings...), allowing it to run reliably on different environments.

Containers allow to isolate applications from the computer environment. They run on Docker Engine, a runtime compatible with most Linux and Windows-based systems.

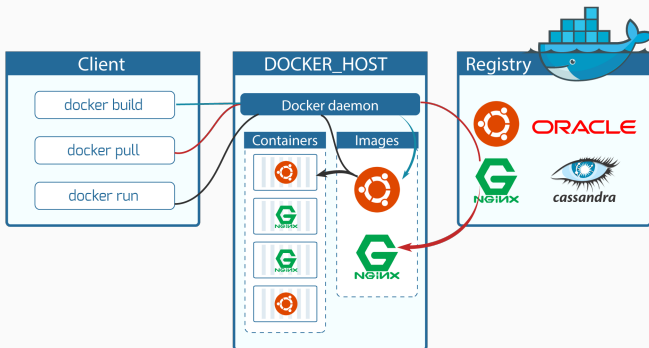
This tutorial will allow you to download a container gathering all the material for the course (jupyter notebooks) and the dependencies needed to run them. **You will thus have everything you need in one place!**

Containers vs Virtual Machines



For VMs to work, a hypervisor is used to create virtual guest OSs able to access an infrastructure's resources. Each app running on the cloud runs on a separate guest OS. On the other hand, containers all run on the same kernel, allowing to limit the resource use, but offering poorer security guarantees. Source: <https://www.weave.works/blog/a-practical-guide-to-choosing-between-docker-containers-and-vm>

DOCKER COMPONENTS



To understand how docker in more details, you may read:
<https://docs.docker.com/get-started/overview/>

Docker images

A Docker **image** is an immutable file containing the source code, libraries, dependencies, etc. needed for the application to run. It can be *built* (i.e. created) through the command `docker build`, or downloaded from an online storage (e.g. DockerHub) through `docker pull`. We will use the latter option.

An image is a **read-only** file, whereas the container is a virtualized run-time environment where your application will actually be running. Launching a container is done with the command `docker run`.

Creating such images is achieved by writing a **Dockerfile**, which is a step-by-step configuration file of the commands needed to create your image.


```

m-e-c.Dockerfile X
Users > julesbaudet > Documents > 4. Research > mec_equil > Docker > m-e-c.Dockerfile
1 FROM ubuntu:latest
2
3 RUN apt-get update && apt-get -y update
4 RUN apt-get install -y build-essential python3.7 python3-pip python3-dev wget
5 RUN pip3 -q install pip --upgrade
6
7 RUN mkdir src
8 WORKDIR src/
9 COPY ./content .
10
11 RUN pip3 install -r requirements.txt
12 RUN apt-get install -y --fix-missing curl autoconf libtool
13 RUN curl -L https://github.com/libspatialindex/libspatialindex/archive/1.8.5.tar.gz | tar -xz
14 RUN cd libspatialindex-1.8.5 && ./autogen.sh && ./configure && make && make install && ldconfig
15 RUN pip3 install osmnx
16
17 RUN pip3 install jupyter
18
19
20 WORKDIR /src/notebooks
21
22 # Add Tini. Tini operates as a process subreaper for jupyter. This prevents kernel crashes.
23 ENV TINI_VERSION v0.6.0
24 ADD https://github.com/krallin/tini/releases/download/${TINI_VERSION}/tini /usr/bin/tini
25 RUN chmod +x /usr/bin/tini
26 ENTRYPOINT ["usr/bin/tini", "--"]
27
28 EXPOSE 8888
29
30 CMD ["jupyter", "notebook", "--port=8888", "--no-browser", "--ip=0.0.0.0", "--allow-root"]

```

Example of a Dockerfile for creating an image allowing to run Jupyter notebooks.

Installing Docker

Installing Docker

To download the course container, you will first need to install Docker.

Follow the instructions corresponding to your machine:

- For Mac: <https://docs.docker.com/docker-for-mac/install/>
- For Windows: <https://docs.docker.com/docker-for-windows/wsl/>
(Please make sure that you meet the prerequisites! In particular, you need to install WSL 2
<https://docs.microsoft.com/en-us/windows/wsl/install-win10>)

Downloading the image

Now that you have installed the container, you are ready to download the image of the course.

1. Setup a work directory on your local hard drive (avoid synced folders). For me, `/Users/julesbaudet/Desktop/mec_optim`
2. Store the *gurobi.lic* file I sent by email somewhere on your local hard drive (avoid synced folders). For me, `/Users/julesbaudet/Desktop/gurobi`.
Please do not share the gurobi.lic file outside of the class.
3. Open the shell, navigate to folder created for the course.
For me, `cd /Users/julesbaudet/Desktop/mec_optim`
4. Download the course image from docker hub.
`docker pull alfredgalichon/mec_optim:2021-01`

The last instruction will start the process of downloading the image. This may take a while: it is normal!

Running the container

Now that you have installed the image, you are ready to run the container!

```
First, run the command: docker run -it --rm -p 8888:8888 -v
/Users/julesbaudet/Desktop/mec_optim/volume:/src/notebooks/volume
-v
/Users/julesbaudet/Desktop/gurobi/gurobi.lic:/opt/gurobi/gurobi.lic:ro
alfredgalichon/mec_optim:2021-01
```

Make sure to change the paths to match those you chose for your work directory and gurobi licence file.

The flag `-v ADDRESS_1:ADDRESS_2` creates a *volume*, i.e. a folder shared by your local system at the address `ADDRESS_1` and your container at `ADDRESS_2`. It will allow you to send files from your computer to your container and vice-versa¹.

¹You can find out the meaning of the other flags `-p`, `-it` etc on <https://docs.docker.com/engine/reference/run/>

Opening Jupyter

After running the command `docker run`, your terminal should display something like this:

```
mec_test — com.docker.cli • docker run -it --rm -p 8888:8888 -v ~/Documents/4. Resear...
ase.
[W 2021-01-17 06:29:00.760 LabApp] 'allow_root' has moved from NotebookApp to ServerApp. T
his config will be passed to ServerApp. Be sure to update your config before our next rele
ase.
[I 2021-01-17 06:29:00.774 LabApp] JupyterLab extension loaded from /usr/local/lib/python3
.7/site-packages/jupyterlab
[I 2021-01-17 06:29:00.774 LabApp] JupyterLab application directory is /usr/local/share/ju
pyter/lab
[I 06:29:00.781 NotebookApp] Serving notebooks from local directory: /src/notebooks
[I 06:29:00.781 NotebookApp] Jupyter Notebook 6.1.6 is running at:
[I 06:29:00.781 NotebookApp] http://8fa243478d0e:8888/?token=91e33a20b16768ab42a2207c0f951
e04f48cce39eae0e224
[I 06:29:00.781 NotebookApp] or http://127.0.0.1:8888/?token=91e33a20b16768ab42a2207c0f95
1e04f48cce39eae0e224
[I 06:29:00.781 NotebookApp] Use Control-C to stop this server and shut down all kernels (
twice to skip confirmation).
[C 06:29:00.786 NotebookApp]

To access the notebook, open this file in a browser:
file:///root/.local/share/jupyter/runtime/nbserver-1-open.html
Or copy and paste one of these URLs:
http://8fa243478d0e:8888/?token=91e33a20b16768ab42a2207c0f951e04f48cce39eae0e224
or http://127.0.0.1:8888/?token=91e33a20b16768ab42a2207c0f951e04f48cce39eae0e224
```

Copy one of the URLs displayed in a browser. You should now have access to Jupyter running in your container!

Throughout the class, we advise you to work as follows:

- Whenever you want to work on a notebook, navigate to your work directory and use the `docker run` command presented at slide 10.
- When you are done working on a notebook, save your modifications and **make sure to copy this notebook in your volume!**
Otherwise, when you will close your container, **all your progress will be lost.**

We have created a Discord server for you to discuss the material, ask questions etc.

<https://discord.gg/j5gQjhByZt>

Congratulations! You now have everything you need to start the class!

If needed, do not hesitate to contact me at
`jules.baudet99@gmail.com`