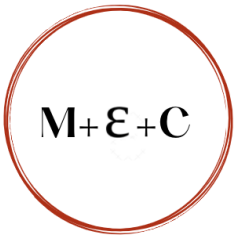


Math+Econ+Code Masterclass

June 2021

Equilibrium Transport and Matching Models in Economics

Setting up Docker[1]

1 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 operative systems,
3. save a significant amount of computing resources.

To do so, Docker uses **containers** and **images**.

1.1 Docker containers

A **container** is a standard unit of software. It is a "package" containing an application and all of its dependencies (libraries, settings, etc.) 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!**

[1]This tutorial is based on Jules Baudet's precious work for the June 2020 'math+econ+code' masterclass. For any further information, contact Gabriele Buontempo: gabriele.buontempo@sciencespo.fr. Support from the ERC CoG-866274 EQUIPRICE grant is acknowledged.

1.2 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 the command **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**.

For more details on Docker, you may read: https://docs.docker.com/get-started/overview/.

2 Using Docker for the M+E+C Masterclass

2.1 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).

Once you have installed Docker, open the Docker Desktop application.

2.2 Downloading the image

You are now ready to download the **image** of the course.

1. Setup a work directory on your local hard drive (avoid synced folders). For me, that is:

   ```
   /Users/gabrielebuontempo/Desktop/mec_june
   ```

2. Open the shell, navigate to folder created for the course. For me, that is:

   ```
   cd /Users/gabrielebuontempo/Desktop/mec_june
   ```

3. Download the course image from docker hub with the following command:

   ```
   docker pull alfredgalichon/mec_equil:2021-06
   ```

The last instruction will start the process of downloading the image (which may take a while).

2.3 Running the container

Now that you have installed the image, you are ready to run the container. To do so, I use the following command in shell:

```
docker run -it --rm -p 8888:8888 -v /Users/gabrielebuontempo/Desktop/mec_june:/home alfredgalichon/mec_equil:2021-06
```

NOTE: you have to use the command above with **the path to the work directory in your hard drive**, i.e. you have to replace

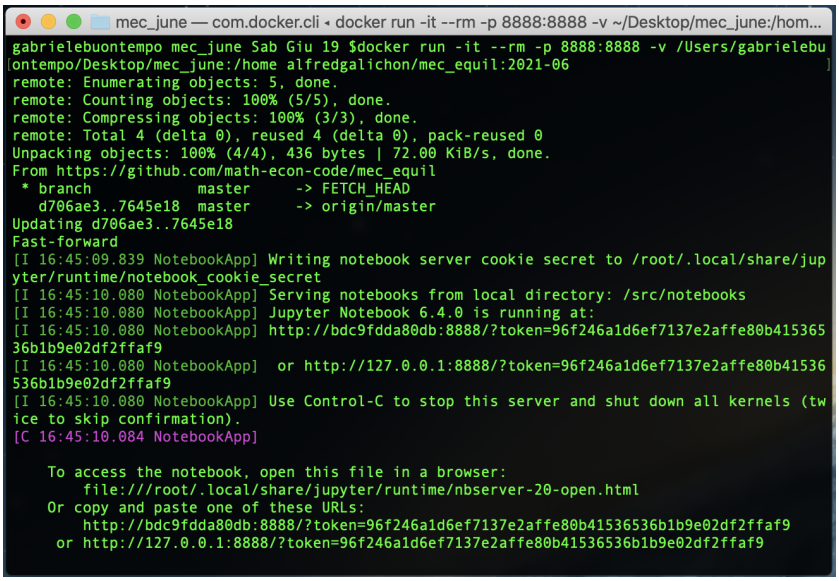
```
/Users/gabrielebuontempo/Desktop/mec_june
```

with the path to the folder you created for the course on your machine.

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 versa.[2]

2.4 Opening Jupyter

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



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

[2] You can find out the meaning of the other flags -p, -it, etc. here: https://docs.docker.com/engine/reference/run/.

3 Good practices

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 in section 2.3.
- 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!**

You now have everything you need to start the class!

If needed, do not hesitate to contact me at gabriele.buontempo@sciencespo.fr.