

# Installing the CH Jupyter Environment

---

Bernhard Liebl, [liebl@informatik.uni-leipzig.de](mailto:liebl@informatik.uni-leipzig.de), October 2020

A very large part of running computations involves installing software that actually works. This document helps you set up the environment we use in the course.

Download `ch_docker.zip` and unzip it. Then follow the descriptions below, depending on your operating system, i.e. look for one of these sections below:

- Installation on Linux
- Installation on macOS
- Installation on Windows 10

For macOS and Windows, we provide text-only and video versions of the setup. Choose what suits you best. The text versions are more geared towards people who are familiar to work in a terminal.

If you are running macOS or Windows, you also have the option of running a full Ubuntu Linux as VM and then following the steps for the Linux installation. We only recommend this as a last resort if all other options fail.

If you run into problems, please contact us.

## The Software We Use

### Jupyter Lab

Jupyter Lab is a software for running interactive computations in notebooks. While it is mainly used for Python, it also supports many other programming languages.

You can find more information under [https://jupyterlab.readthedocs.io/en/stable/getting\\_started/overview.html](https://jupyterlab.readthedocs.io/en/stable/getting_started/overview.html)

### Docker

Docker allows us to give you a standardized Jupyter Lab installation (i.e. containing various prepackaged scientific libraries) on many different platforms. If you want to learn more about Docker, read the Docker section at the end of this document.

---

## Installation on Linux

### Docker or not?

For Linux, we describe two ways of installation: (1) without Docker or (2) using our prepackaged Docker container to run Jupyter in an isolated environment that comes with all necessary packages.

While we feel that (2) is generally the neater solution as it isolates itself from your system installation, some users might prefer (1) as it is faster to perform.

### Option A: Installing without Docker

You need to install the packages seen in `ch_docker/docker/Dockerfile` directly on your machine. First, install the needed system packages. On Ubuntu/Debian, you would run:

```
sudo apt update

sudo apt install \
    build-essential \
    git \
    libtesseract-dev \
    libleptonica-dev \
    tesseract-ocr \
    tesseract-ocr-deu \
    tesseract-ocr-eng
```

Next, build an isolated Python environment that runs Jupyter Lab and other necessary libraries. We suggest you first install conda, e.g. via miniconda3 as described under <https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html>. For example:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
chmod +x Miniconda3-latest-Linux-x86_64.sh

# IMPORTANT: for the next command, use default settings, but
# answer "yes" when asked if you want to run "conda init"
./Miniconda3-latest-Linux-x86_64.sh
```

Now open a new terminal window - this will make sure you see the newly installed `conda` command.

Then run:

```
conda create --name=ch2020 python=3.8
conda activate ch2020

# check that you are now inside the (ch2020) env
pip install -r /path/to/ch_docker/docker/requirements.txt
python -m spacy download en_core_web_md && python3 -m spacy
download de_core_news_md
```

In a new terminal, you are now able to run Jupyter Lab via:

```
conda activate ch2020
# check that you are now inside the (ch2020) env
jupyter lab
```

The last command will launch a web browser that takes you to Jupyter Lab.

## Option B: Installing with Docker

### Installing Docker Itself

First, install Docker for Linux. The exact steps depend on your Linux distribution. For Ubuntu, instructions are at <https://docs.docker.com/engine/install/ubuntu/>. We suggest you follow the steps in the five boxes under the "Install using the repository" option, i.e. starting with "sudo apt-get update" and ending with "sudo apt-get install docker-ce docker-ce-cli containerd.io".

Once Docker is installed, you should be able to run `docker` in a terminal without an error. You then also need to install `docker-compose`. On Ubuntu this can be achieved through `sudo apt install docker-compose`.

To automatically launch the Docker server on boot, you need to take additional steps. For details, see "Configure Docker to start on boot" under <https://docs.docker.com/engine/install/linux-postinstall/>. On most distros, you want to run `sudo systemctl enable docker`.

Note that configuring Docker in this way means always running it in the background upon each machine launch, which can have unwanted side effects. Please read the section "General Advice on Docker" for further information.

## Installing Jupyter as Docker Container

Basically, we now need to run `docker-compose up`. However, since we share the home folder and need to run `docker-compose up` with `sudo` on a default Docker installation, our `$HOME` directory will be wrong. Also, there are some problems with Docker's volume sharing and user ids. We provide an `up.sh` script that fixes these issues and lets you read and write your HOME directory from Jupyter.

Run:

```
cd /path/to/ch_jupyter/launchers/linux
chmod +x up.sh
cat up.sh # verify what you are running
./up.sh
```

This will take a while on the first run.

Once the installation is done - this can take up to 30 minutes on the first run - and you see messages with `jupyter_1`, you can access Jupyter via a web browser on `localhost:8888`. To test file access works, try to create and save a new Jupyter notebook in one of your folders.

When done, stop the container/server by hitting Control-C in the terminal.

---

## Installation on macOS

### Text Only Version

You need to install Docker Desktop for Mac, then open the application at `ch_docker/launchers/macOS/Jupyter`. Alternatively, if macOS will not let you launch the app, or you want to use the Terminal, you can just `cd` to `ch_docker/docker` and run `docker-compose up`.

Once this command is started - this can take up to 30 minutes on the first run - and you see messages with `jupyter_1`, you can access a preconfigured Jupyter Lab installation via a web browser on `localhost:8888`.

To stop Jupyter hit "Cancel" and then "Quit" if you are using the `launchers/macOS/Jupyter` app, or - if you are using the Terminal - by hitting Control-C. You can also use the Docker Desktop Dashboard to check and manage running containers.

Note that Docker is by default configured to always run in the background upon each machine launch, which might not be what you want. Please read the section "General Advice on Docker" for further information.

## Video Version

A step-by-step video of the whole installation procedure is here:

<https://www.dropbox.com/s/j24cj4ndijfkfkj/macOS.mp4?dl=0>

Note that Docker is by default configured to always run in the background upon each machine launch, which might not be what you want. Please read the section "General Advice on Docker" for further information.

## Useful Links

Docker Desktop for Mac:

<https://hub.docker.com/editions/community/docker-ce-desktop-mac>

---

## Installation on Windows 10

You need to install Docker Desktop for Windows with the WSL2 backend. When properly installed, this will allow you to launch the `ch_jupyter` environment using the `ch_docker/launchers/win10/launch.bat` batch file.

## Text Only Version

The installation procedure is unfortunately rather involved. If you are tech savvy, we suggest the following steps:

- Obtain a Docker Hub ID by signing up on Docker Hub (<http://hub.docker.com>). Your Docker ID is a username you choose yourself.
- Enable hardware virtualization in the BIOS settings. Steps for doing this are here: <https://bce.berkeley.edu/enabling-virtualization-in-your-pc-bios.html>
- Check that your Windows 10 version has version  $\geq$  1903. If not, update accordingly.
- Install WSL2 using the steps described in <https://docs.microsoft.com/de-de/windows/wsl/install-win10>
- Install Docker Desktop for Windows from <https://hub.docker.com/editions/community/docker-ce-desktop-windows>
- Log into Docker, either using the UI or `docker login` on the console using your Docker ID from above
- Once Docker is running, either launch

`ch_docker/launchers/win10/launch.bat` or perform the steps yourself in a terminal (look at what `ch_docker/launchers/win10/launch.bat` does).

- Once the container installation is done - this can take up to 30 minutes on the first run - and you see messages starting with `jupyter_1`, you can access a preconfigured Jupyter Lab installation via a web browser on localhost:8888.
- When done, stop the container/server by hitting Control-C. You can also use the Docker Desktop Dashboard to check and manage running containers.

Note that Docker is by default configured to always run in the background upon each machine launch, which might not be what you want. Please also read the section "General Advice on Docker" for further information.

## Video Version

The following video is a step-by-step approach to finishing the installation:

<https://www.dropbox.com/s/e099h1vk4z4qzo2/win10.mp4?dl=0>

Two corrections to the video:

- The video mentions a separate video on how to obtain a Docker ID. This video does not exist, as the process is really simple. Simply go to <http://hub.docker.com> and sign up. Your Docker ID is a username you choose yourself.
- The mention to Hyper-V in the BIOS setting is wrong and misleading. What is actually meant is that if problems with starting up Docker persist, you should enable hardware virtualization in your BIOS and described here: <https://bce.berkeley.edu/enabling-virtualization-in-your-pc-bios.html>

Note that Docker is by default configured to always run in the background upon each machine launch, which might not be what you want. Please also read the section "General Advice on Docker" for further information.

## Useful Links

Docker Desktop for Windows:

<https://hub.docker.com/editions/community/docker-ce-desktop-windows>

Information on Installing WSL2:

<https://docs.microsoft.com/de-de/windows/wsl/install-win10>

Docker Hub:

<https://hub.docker.com>

# Docker

## What is Docker?

Docker is a software that allows you to run software isolated inside a defined environment, a concept not completely unsimilar to virtual machine (VM) solutions like VirtualBox, Parallels or VMWare. However, Docker is much more lightweight, smaller and faster than VMs. In short, Docker allows us to provide you with a defined software stack - consisting of JupyterLab and an installation of Python3 - that contains all needed software packages for the course, runs fast and is resource friendly.

While it is possible to install various necessary packages for the course in isolated Python environments directly on your machine (e.g. using `conda`), this approach breaks down for some other components we use in the course that need compilation for installing them on some platforms. Docker allows us to provide all course participants with the same working environment, regardless of the operating system you are using. This also makes it easier to us to provide an environment that actually works for everybody and has been tested.

While the installation of Docker is a bit technical on Linux and can be rather cumbersome on Windows, once installed, you will have obtained a fast and tested environment that contains all you need for the course. You won't have to second guess whether installing packages used in the courses will actually work on your machine.

For a short but more detailed introduction of what Docker is and what it is about, watch [https://www.youtube.com/watch?v=TvnZTi\\_gaNc](https://www.youtube.com/watch?v=TvnZTi_gaNc)

## Some General Advice on Docker

Docker runs as a background system service that, once installed, usually always runs in the background once you start up your computer. It might also restart containers that were not closed on previous shutdowns in the background, which means your machine might run the CH Jupyter notebook all the time even if you are not using it.

If you experience performance problems, configure open up the preferences in Docker Desktop and disable "launch on startup". You can also quit the Docker Desktop app which usually shut downs the whole Docker service and any connected containers.

On macOS and Windows you can also use the Docker Desktop Dashboard - accessible via the Docker menu - to check and manage running containers.