



Spring 2021 Setup Guide [For Q1]

Getting Started

A video tutorial has been created to help walk you through the steps in this document. The setup video if you have:

- **Mac:** <https://youtu.be/fHej6YzEupE>
- **Windows 10:** <https://youtu.be/p31blzqPsIM>

To get started with Docker follow the documentation below.

1. The first step to get started is to download and install Docker according to the instructions for your operating system found here: <https://docs.docker.com/get-docker/>
 - a. Windows 10 (Pro, Enterprise, or Education): <https://docs.docker.com/docker-for-windows/install/>
 - b. Windows 10 (Home): <https://docs.docker.com/docker-for-windows/install-windows-home/>
 - i. Before installing Docker on Windows 10 Home, be sure the [system requirements](#) are met. Specifically, ensure Windows is updated to version 2004 or higher and WSL 2 is enabled by following steps 1 – 4 of the [Windows Subsystem for Linux Installation Guide for Windows 10](#)
 - c. Mac: <https://docs.docker.com/docker-for-mac/install/>
 - d. Linux: <https://docs.docker.com/engine/install/>

2. Open a terminal (Windows users, use Powershell, not CMD) inside your HW3 directory.

3. From inside the HW3 directory run the following command:

- ```
docker container run -d -v ${PWD}/Q1:/root -p 127.0.0.1:6242:8888 --name hw3 polochau/cse6242-2021spring
```
- a. Note: Linux and Mac users may require adding `sudo` before the command.
  - b. The first time you run this command it will download the `polochau/cse6242-2021spring` image from DockerHub: <https://hub.docker.com/r/polochau/cse6242-2021spring>. The Docker image is approximately 1GB so the time it takes to download will depend on your internet connection.
  - c. The command is reproduced below on the left with line breaks and shorthand options extended, plus comments to the right explaining what each part does:

|                                                                                                                                                                    |                                                                                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>docker container run \<br/>--detach \<br/>--volume \${PWD}/Q1:/root \<br/>--port 127.0.0.1:6242:8888 \<br/>--name hw3 \<br/>polochau/cse6242-2021spring</pre> | <pre># Start running a container<br/># Run in the background<br/># Make Q1 dir available at /root<br/># Forward port 6242 to 8888<br/># Name the container "hw3"<br/># Use this container image</pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- d. This command will start a container with the name *hw3*. You can change the name to whatever you'd like, but the remainder of the commands in this document assume the container name is *hw3*
  - e. The `-p` flag will publish port 8888 on the container to port 6242 on the host. This will allow you to access Jupyter by loading <http://localhost:6242> in your browser, which is set to accept connections on port 8888 inside the container.
  - f. The `-v` flag will share a local directory on your computer (which should be the HW3 directory's Q1 subdirectory) with the `/root` directory on the container, which is the directory that Jupyter will start in, making your files accessible.
4. At this point you can run **docker container ls** to confirm the container is started.

```
(base) + Q1 docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b11bc5afe30d polochau/cse6242-2020fall "jupyter notebook --..." 10 seconds ago Up 9 seconds 127.0.0.1:6242->8888/tcp hw3
(base) + Q1
```

5. Once the container has been started you may go to <http://localhost:6242> in your browser to access Jupyter

## Helpful Commands

- Show all running containers:
  - `docker container ls`
- Show all containers (started and stopped):
  - `docker container ls -a`
- Force kill and remove a container (useful if you made a mistake and need to start a new container with the same name):
  - `docker container rm -f <container>`
  - Ex: `docker container rm -f hw3`
- Show all images:
  - `docker image ls`
- Attach to a running container (type `exit` to detach from the container):
  - `docker container exec -it <container> /bin/bash`
  - Ex: `docker container exec -it hw3`
- Start a stopped container:
  - `docker container start <container>`
  - Ex: `docker container start hw3`

## FAQ

1. After I restart my computer, I am unable to attach to the container

- a. After restarting your computer the Docker container may not automatically restart. See the helpful commands above on how to start a stopped container. Additionally, check to make sure the Docker daemon/service is running on your host platform. If it doesn't start automatically, find and run the Docker desktop application, which should start the Docker service.
2. How do I transfer my code to the docker container?
  - a. There is no need to manually transfer code between your local environment and the docker container. Since the container was created with a [bind mount](#) using the -v flag, all the files are automatically synchronized between your local environment and the container. If you need to access a file inside the container, make sure you place it at the point that you bound the volume (The Q1 subdirectory, if using the example command in the instructions above)
3. I'm getting an "Error response from daemon: Conflict. The container name "/hw3" is already in use by container" error. How do I fix it?
  - a. This error typically means you already have a container named "hw3." Instead of creating a new container, you should list all containers and either start the container if it's stopped (status is Exited) or attach to it if it's running (status is Up). If it is misconfigured and you need to completely remove the "hw3" container and create a new one, you can do so by killing the container and removing it using the commands above.
4. I'm getting an "Error response from daemon: driver failed programming external connectivity on endpoint" error. How do I fix it?
  - a. This typically means you have another application (or another container) using the port specified in the -p flag. Check that you don't already have a hw3 container running, or try changing the port from 6242 to 8888 and use that to access Jupyter instead.
5. While installing Docker, I get an error "Installation failed: one prerequisite is not fulfilled"
  - a. For Windows 10 Home, Docker is only compatible with Build 19018+. So you can check from System Information before installing Docker and update Windows if required.
6. When installing WSL 2 on Windows, I'm getting the error "Elevated permissions are required to run DISM."
  - a. When installing WSL 2, you will need to do so as an administrator. You can start an elevated command prompt by accessing the Power Users menu by pressing Windows+X or right-clicking the Start button. On the Power Users menu, choose "Command Prompt (Admin)" and run the WSL 2 installation commands.
7. When trying to upgrade WLS, I'm getting the error: "This update only applies to machines with the Windows Subsystem for Linux."
  - a. There are a few reasons you will see this error. The most common reason is that "after you enabled WSL, a reboot is required for it to take effect, reboot your machine and try again." Additional details may be found in the [Windows Subsystem for Linux Installation Guide for Windows 10](#)

**Notes:**

1. Docker containers should be treated as ephemeral. In situations where the Docker bind mount does not correctly work and the Jupyter notebook is not shared between the Docker container and the local machine, the Jupyter notebook may be uploaded manually as a workaround. However, since the containers should be treated as ephemeral, you must manually download the notebook by navigating to File -> Download as -> Notebook otherwise the notebook may be deleted and work may be lost.
2. There have been issues when running Docker inside a directory that is synced with OneDrive or Dropbox.