Running Lessons With Docker

Correlation-One uses `docker` to run Jupyter Notebook training cases. This allows us to eliminate portability issues between different operating systems.

This document describes how to use `docker` to run training cases.

Supported systems include macOS, Windows (7/8/10), and Linux.

Installing Docker

The installation process differs by system. We defer to the official `docker` documentation for the installation instructions.

- Linux: `docker-engine`
- Windows 7/8 and Windows 10 Home (below version 2004): `docker-toolbox`
- macOS (before El Capitan 10.11): `docker-toolbox`
- Windows 10 Home (version 2004 or higher): `docker-desktop`
- Windows 10 Pro/Enterprise/Education (Build 16299 or later): `docker-desktop`
- macOS (from El Capitan 10.11 on): `docker-desktop`

Note: If you are using Windows, you will need to enable virtualization on your machine. This process is machine specific, we defer to this guide for further guidance, but still you may need to search how to do it for your specific machine.

Running the Docker Image

Once `docker` is installed, please follow the below instructions on how to run the relevant `docker` image for ***your*** operating system.

Windows 10 Home (version 2004 or higher)/Pro/Enterprise/Education (Build 16299 or later)

Open the `Command Prompt` as an administrator. In order to do so, please find `CMD` in the start menu, right-click it, and select "Run as Administrator".

Enter the following command to pull and run the `docker` image:

```
docker run -p 8888:8888 -v c:\Users\<User>\path\to\your\notebooks:^
/home/jovyan/work jupyter/scipy-notebook:17aba6048f44
```

Case-writers: The path `c:\Users\<User>\path\to\your\notebooks` should be replaced to the directory **containing** the `starter-kit/` directory otherwise tests will not be executed.

Replace the `<User>` field with your Windows system user-name. The path `c:\Users\<User>\path\to\your\notebooks` should contain the case materials for your lesson. This path is the only part of your local file system visible to the `docker` container! Feel free to change the path to something more convenient if you feel so inclined.

The prompt should pull the `docker` image and run a notebook server. Follow the instructions in the *accessing the notebook server* section.

macOS (from El Capitan 10.11 on) and Linux

On macOS, launch the Terminal application. On Linux, open your preferred terminal.

Enter the following command to pull and run the `docker` image:

```
docker run -p 8888:8888 -v "$PWD":/home/jovyan/work \
  jupyter/scipy-notebook:17aba6048f44
```

Case-writers: Please run the command above from the directory **containing** the `starter-kit/` directory otherwise tests will not be executed.

Students: Ensure that the command is run from a directory containing the cases used in your lesson.

The prompt should pull the `docker` image and run a notebook server. Follow the instructions in the *accessing the notebook server section.*

Accessing the notebook server

After running the `docker run` command with the instructions according to your operating system, you will be able to access the notebook server. You should have been prompted with the following:

```
Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
    http://(b56ec2d7c00a or 127.0.0.1):8888/?token=<some-token>
```

Copy this URL and paste it into your browser (we've found MS Edge not to work properlly), editing the `(b56ec2d7c00a or 127.0.0.1)` portion to be just the `127.0.0.1` component.

Please note that the token field will be different each time you run the `docker run` command. So you cannot reuse the same URL without first changing the `<some-token>` value.

Windows 7/8/10 Home (below version 2004) and macOS (before El Capitan 10.11)

Open the `Docker Quickstart Terminal` application.

Enter the following command to know the ip address on which your `default docker-machine` is running:

```
docker-machine ip default
```

The prompt should display the address of your `default docker-machine`. You will use this address to replace the `<docker-machine ip address>` in a later step.

Then enter the following command to pull and run the `docker` image:

```
docker run -p 8888:8888 -v "$PWD":/home/jovyan/work \
  jupyter/scipy-notebook:17aba6048f44
```

Case-writers: Please run the command above from the directory **containing** the `starter-kit/` directory otherwise tests will not be executed.

Students: Ensure that the command is run from a directory containing the cases used in your lesson.

The prompt should pull the `docker` image and run a notebook server. You should have been prompted with the following:

```
Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
    http://(b56ec2d7c00a or 127.0.0.1):8888/?token=<some-token>
```

This URL will not work on a Windows system using `docker-toolbox` . To make it work, you need to replace the prompted IP address with the IP address of your `docker` machine. Your URL should look something like this:

```
http://<docker-manchine ip address>:8888/?token=<some-token>
```

Where `<docker-manchine ip address>` is the ip address you got when running `docker-manchine ip default` and `<some-token>` is the token outputted by the `docker run` command. Copy the constructed URL into your favorite browser (we've found MS Edge not to work properlly) to open the notebook.

Please note that the token field will be different each time you run the `docker run` command. So you cannot reuse the same URL without first changing the `<some-token>` value.

Environment Management

Case writers: Please ignore this section.

Once a `docker` process is running, you may follow the instructions listed in the `Managing Case Environments` document.

Managing Docker Processes

Running `docker` images persist in the background once opened. Eventually, you should remove these running processes.

List the running `docker` processes by using `docker ps`. Sample output:

```
docker ps
CONTAINER ...    NAMES
aacda4d33aef     elastic_zhukovsky
```

The output above shows a table of running processes. Most of the fields are omitted for clarity. The `NAMES` field is used to reference the process. Names are randomly generated.

To stop a process, use the `docker stop` command with the name of the process.

```
docker stop <name-of-process>
# e.g., docker stop elastic_zhukovsky
```