Docker & Docker Compose Installation

Important - Operating System Gotchas
You need to install different Docker versions depending on your operating system. Check Docker system requirements carefully, to decide which Docker installation is the right one for you.

Install guide for Docker Desktop for Windows (be sure to check the requirements)
https://docs.docker.com/docker-for-windows/install/

Docker toolbox (for older operating systems, including some versions of Windows 10)
https://docs.docker.com/toolbox/overview/

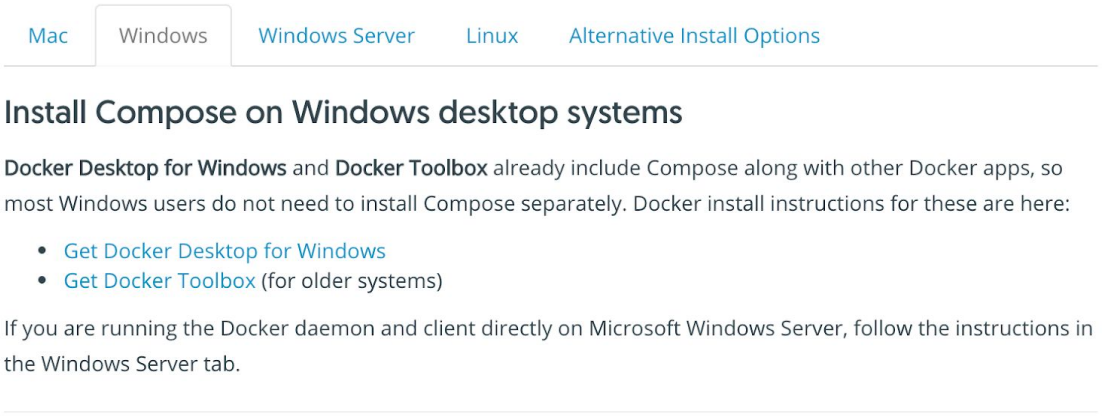
Install guide for Docker Desktop for Mac (be sure to check the requirements)
https://docs.docker.com/docker-for-mac/install/

Note 1: Docker toolbox will also install Oracle VM and Git. If you already have these programmes installed in your computer, **uncheck** them in the docker installation wizard.

Note 2: If installing Docker from the toolbox, or Docker Desktop for Windows or Mac, you will install docker compose as well. So you won't need to install Docker-compose separately. More details below.

1) If necessary, install docker-compose

The docker-compose install docs are the best place to start
https://docs.docker.com/compose/install/



Docker Compose relies on Docker Engine for any meaningful work, so make sure you have

Docker Engine installed either locally or remote, depending on your setup.

- On desktop systems like Docker Desktop for Mac and Windows, Docker Compose is included as part of those desktop installs.
- On Linux systems, first install the Docker for your OS as described on the Get Docker page, then come back here for instructions on installing Compose on Linux systems.
- To run Compose as a non-root user, see Manage Docker as a non-root user.

Commands

Start your Docker Desktop application if you haven't already (e.g. Docker Desktop for Mac, Docker Toolbox etc.)

If you are able to run this command:

```
docker-compose --help
```

Then your installation is complete

Windows Specific Docker Issue

Relevant Resources

- **Challenges of accessing containers running on windows via localhost:**

https://docs.docker.com/docker-for-windows/troubleshoot/#limitations-of-windows-containers-for-localhost-and-published-ports

- **Find IP address with Docker Toolbox**
  https://devilbox.readthedocs.io/en/latest/howto/docker-toolbox/find-docker-toolbox-ip-address.html
- **What is a TCP port?**
  https://www.bullguard.com/bullguard-security-center/pc-security/computer-security-resources/tcp-ip-ports

Commands (Windows only)

`git checkout {Create notebooks commit hash}`

Make sure you are in the *`testing-and-monitoring-ml-deployments/exercise_notebooks/utility_scripts`*

directory

Start the command prompt **as an Administrator**

Run the script in this directory:

`MapPortsForDocker.cmd 5000`

The script should take about 10 seconds to run.

To undo the script

use `netsh interface portproxy reset`

And manually clear hosts file. See:
https://www.howtogeek.com/howto/27350/beginner-geek-how-to-edit-your-hosts-file/

Docker Compose Exercise

Relevant Resources

- Redis introduction: https://redis.io/topics/introduction
- Redis Python client: https://github.com/andymccurdy/redis-py
- Base image used in the exercise Dockerfile (python:3.7-alpine):
  https://github.com/docker-library/python/blob/5f1e3cbcb02c508a5357b5637f3b7a51937e4b5d/3.7/alpine3.10/Dockerfile

Commands (Windows only)

```
git checkout {Create notebooks commit hash}
```

Navigate to the docker_exercise directory which should be reachable in the following paths: testing-and-monitoring-ml-deployments\exercise_notebooks\docker_exercise

Make sure you are in the exercise directory, then start the flask and redis containers by running this command:

```
docker-compose up
```

then CTRL+C to stop

You should see the container being built, which may take a couple of minutes. Once this is complete, you should be able to see the web application running by navigating to **http://localhost:5000**

Docker Cheatsheet

Docker:

https://www.docker.com/sites/default/files/d8/2019-09/docker-cheat-sheet.pdf

Docker Compose:

https://devhints.io/docker-compose

Debugging Docker with logs
https://hackernoon.com/to-boldly-log-debug-docker-apps-effectively-using-logs-options-tail-and-grep-53d2e655abcb

Warning: Docker eats disk space like Cookie Monster eats cookies

Everytime you create a new docker image, hundreds of megabytes of disk space is consumed. Manage this disk usage by:

1. Removing all containers (running or not) - this is more to prevent containers fighting over the same TCP ports rather than saving disk space.

Windows (powershell)

```
docker ps -aq | foreach {docker rm $_}
```

OS X / Linux

```
docker rm $(docker ps -a -q)
```

2. Removing docker images - this will free up loads of disk space and is the command you want to use the most, just be aware that docker containers now have to be built from scratch and will take *much* longer to spin up as a result.

View images (and their IDs for use below)

```
docker images
```

Remove one specific image

```
docker rmi {IMAGE_ID}
```

Remove all images (careful, you may want to keep the base images for faster image building)

```
docker rmi $(docker images -q)
```