Setting up WSL / Ubuntu and Docker on a Windows PC

# Installing WSL1/2 and Ubuntu

1. Enable these Windows features:
   * Windows subsystem for Linux
   * Virtual Machine platform

This will require a restart after the features have installed.

1. Check the version of Windows you have. If it is 1909.535, skip to step 5 - you are using WSL1. If it's 1909.1409+, or if it's major build number is > 1909, carry on - you are using WSL2.
2. Download and install the Linux Kernel update package for WSL2:

<https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi>

1. Set WSL2 as your default version, by using the following command in a Windows terminal
   * wsl --set-default-version 2
2. Install Ubuntu. The version you can install depends on whether you are WSL1 or WSL2. For WSL1, you can only use Ububtu 18.04; for WSL2 you can use either Ubuntu 20.04 or Ubuntu 18.04.

If your Windows installation supports it, use the Microsoft Store to find and install Ubuntu.

If Windows Store is not available, use this approach instead:

* Download Ubuntu as an .appx file (URL depends on version):
  + <https://aka.ms/wsl-ubuntu-1804>
  + <https://aka.ms/wslubuntu2004>
* In Powershell, run this command (using the path of the .appx file you downloaded)
  + add-appxpackage <path of appx file downloaded>

1. Run Ubuntu for the first time, to set up the initial user account.
   * Type in the username to be created
   * Type int he password you want to be associated with the new user
2. Now whenever you run Ubuntu, this is the user account it will automatically use by default

# Install Windows Terminal

The new Windows Terminal provides a simple, convenient way to access your Ubuntu command line.

Windows Terminal can be installed from the Microsoft Store app.

If you can't access the Microsoft Store, use this approach instead:

* Navigate to the latest release on the project's github page:

<https://github.com/microsoft/terminal/releases/tag/v1.5.10271.0>

* Scroll to the bottom of the page where the assets are listed, and click on the downlaod link for the MSIX bundle version of the application (i.e. an installer, not the source code)
* Run the downloaded installer - this installs Windows Terminal
* Note that by using this approach, you won't get automatic updates to the application

To access your Ubuntu command line using the terminal, just launch Windows Terminal, then in the top toolbar, click the down arrow, pick Ubuntu, and a new tab will open connecting you to Ubuntu.

# Install Docker in Windows

You'll need to install Docker actually in Windows itself. For WSL1, that's all you can do. For WSL2, you have the option to also install the Docker CLI into Ubuntu, but even then you get the best performance by linking the two Dockers together, so effectively there's only one.

To install Docker, just navigate to the main Docker website, click to download Docker Desktop for Windows, then run the downloaded installer.

To test Docker is working, try this command (from Windows, not from Ubuntu):

* docker run hello-world

To run a sample website in Docker, run this command (again, from Windows not Ubuntu):

* docker run --name web-demo -d -p 3000:3000 scottyc/webapp
* Then, browse to <http://localhost:3000>

# Installing Docker CLI in Ubuntu

curl https://get.docker.com | sudo bash

sudo moduser -aG docker $(whoami)

docker --version

# Installing docker-compsose in Ubuntu

# Check for the latest version of docker-compose via https://github.com/docker/compose/releases

sudo curl -L https://github.com/docker/compose/releases/download/1.28.5/docker-compose-`uname -s`-`uname -m` -o /usr/bin/docker-compose

chmod +x /usr/bin/docker-compose

docker-compose --version

# Linking docker CLI in Ubuntu (WSL1) to Docker in Windows

# In Windows, configure Docker Desktop to expose docker on tcp://localhost:2375

# In Ubuntu, edit your ~/.bashrc file to add this line to the end:

export DOCKER\_HOST=tcp://localhost:2375

# Re-run your .bashrc file

source ~/.bashrc

# Check it worked

echo ${DOCKER\_HOST}

# Testing it all worked

In Ubuntu, run:

docker info