Computer Setup Programming

Robin Hellmers

October 22, 2021

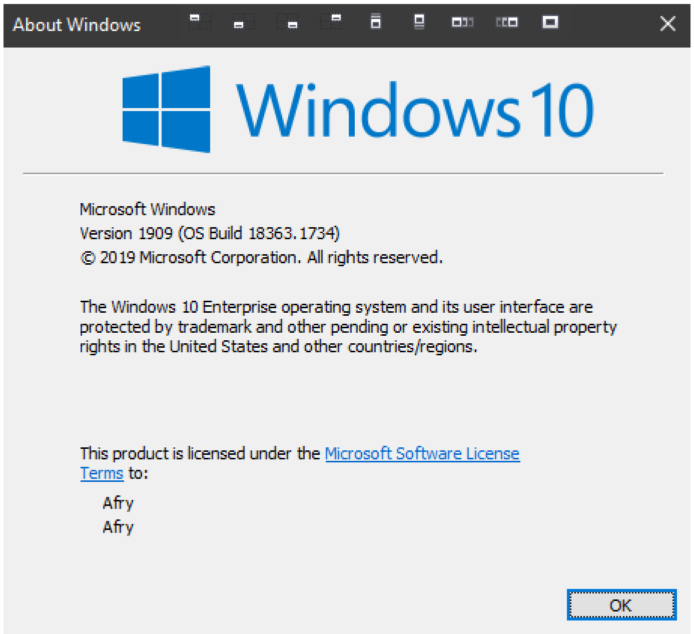
Contents

1 Windows Subsystem for Linux 2 (WSL2)

1.1 Installation

Installation method of WSL depends on which windows version you are running.

Check your Windows version and build number by pressing
`Win + R` -> Entering `winver`



Microsoft Windows
Version 1909 (OS Build 18363.1734)
© 2019 Microsoft Corporation. All rights reserved.

The Windows 10 Enterprise operating system and its user interface are protected by trademark and other pending or existing intellectual property rights in the United States and other countries/regions.

This product is licensed under the Microsoft Software License Terms to:

Afry
Afry

Depending on the version you have, see the below sections.

If you have:

- Windows version 2004 and higher, with build 1904 and higher

Then you can do the **Automatic installation** section.

If that is not the case, then you must at least have:

- Windows version 1903 or higher, with build 18362 or higher

If so, then if running the version

- 1903, the build number must be 18362.1049 or higher
- 1909, the build number must be 18363.1049 or higher

If any of those is true for your system, then you can do the **Manual installation** section.

1.1.1 Automatic installation

A simple command in PowerShell: `wsl --install`

1.1.2 Manual installation

1.1.2.1 Enable Windows Subsystem for Linux

Open PowerShell as Admin and run:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

1.1.2.2 Enable Virtual Machine feature

Open PowerShell as Admin and run:

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

Restart your computer.

1.1.2.3 Download the Linux kernel update package

Download the latest package through **this link**.

1.1.2.4 Set WSL 2 as the default version

Every new distribution installed should run with the WSL version set as default.

Open PowerShell and run:

```
wsl --set-default-version 2
```

1.2 Distribution installation

1.2.1 Downloading and installing

First we are going to download the Ubuntu distribution with some PowerShell commands.

Open PowerShell and go to somewhere in your main drive where you want to install Ubuntu:
When unzipping later, a directory named Ubuntu will be created.

```
cd C:\Users \<username>\
```

The download is done with the command:

```
Invoke-WebRequest -Uri <distro url> -OutFile ./Ubuntu.zip -UseBasicParsing
```

The `<distro url>` must be chosen depending on which distribution you want *(Recommending 18.04)*:

- Ubuntu 20.04 **https://aka.ms/wslubuntu2004**
- Ubuntu 18.04 **https://aka.ms/wsl-ubuntu-1804**

Unzip:

```
Expand-Archive ./Ubuntu.zip
```

Close down PowerShell.

Open the File Explorer at the location which you have the unzipped files e.g. `C:\Users \<username>\Ubuntu \`.

Start the installation by running the `.exe` file e.g. `ubuntu2004.exe`.

1.2.2 Setup of the distribution

When installing, you will be prompted for a username.

1.2.2.1 If the username can't be chosen

If the username can't be chosen because of using e.g. a dot (.), press `Ctrl + C` and the window should close.

Run the `.exe` file again and you should now have skipped the first user initialization and be logged in as `root`.

```
adduser -aG --force-badname <username>
```

Set a password and you can skip entering all of your details by just pressing enter till it's done.

Now we are going to give the new user `sudo` privileges by adding it to the `sudo` group:

```
usermod -aG sudo <username>
```

Verify that the new user is in the `sudo` group with: `groups <username>`

Which should show: `<username> : <username> sudo`

Log into the new user:

```
su - <username>
```

1.2.2.2 Setting the default user

In order to set the default user to the newly created one, run the line below but replace `<username>`:

```
sudo bash -c 'printf "[user]\ndefault =<username>\n " >> /etc/wsl.conf'
```

Verify that the file has been created with the contents:

```
cat /etc/wsl.conf
```

Close the Ubuntu terminal.

Open up the Command Prompt `cmd` and see if the Ubuntu distribution is still running:

```
wsl -l -
```

If the `STATE` is `Running`, then terminate the session with:

```
wsl -t <distro name>
```

Verify that it has `Stopped`:

```
wsl -l -v
```

Close the `cmd`.

Then open up Ubuntu again with the `.exe`. Now it should be logged in with the new user, instead of `root`.

1.2.3 Update the system

Update the list of packages, but doesn't install:

```
sudo apt update
```

Install new version of packages and say yes to every question:

```
sudo apt upgrade -y
```

1.2.4 Set WSL to version 2

We are now going to check which WSL version which is used with the newly installed Ubuntu distribution.

Close any Ubuntu terminal open.

Open up the Command Prompt `cmd` and run:

```
wsl -l -v
```

Which should show all WSL distributions and which version they are running.

If your Ubuntu WSL version is 1, close any Ubuntu window and then run:

```
wsl --set-version <distro name> 2
```

After it is done, recheck that the version now is 2:

```
wsl -l -v
```

1.3 Relevant programs & setups

1.3.1 Visual Studio Code

Download and install Visual Studio Code on Windows

Go to **Extensions** with `Ctrl + Shift + X`.

Search for `Remote - WSL` and press **Install**.

You can now open any file or directory within Ubuntu by using `code <file/directory>`.

See Visual Studio Code's connection to the Ubuntu WSL by clicking on the **Remote Explorer Icon** to the left. Make sure that the dropdown menu at the top shows **WSL Targets**.

If there is a green symbol at your Ubuntu distribution, it is connected. If not,
right click and press **Connect to WSL**.

You can also access your Ubuntu distribution terminal by clicking `Terminal -> New Terminal`.
Press the **dropdown arrow button** at the top right of the newly opened terminal and press **Ubuntu**.

1.3.2 Docker for WSL2

In Windows:

Download Docker Desktop

Install and make sure that the following checkboxes are ticked:

- Enable Hyper-V Windows Features (Probably not necessary because of using WSL2 and not WSL1, but won't hurt)
- Install required Windows components for WSL 2

Run the Docker Desktop program in Windows. If the engine never starts, try to reboot your computer.

In Docker Desktop go into settings through the gear icon in the top right and then:

- **General** make sure that
  - The **Use the WSL 2 based engine** checkbox is checked.
- **Resources** → **WSL INTEGRATION** and make sure that

  - The **Enable integration with my default WSL distro** checkbox is checked.
  - Your installed distribution is enabled.

Open up Ubuntu and check if docker is available:

```
docker --version
```

1.3.2.1 Test docker connection between distribution and Windows

Time to test the connection between Windows docker and Ubuntu docker which both runs through WSL2.

In **Ubuntu terminal**, see which containers that exists:

```
docker ps -a
```

Which should be empty right now.

Open up `cmd` in **Windows** and run `hello-world`:

```
docker run hello-world
```

When it is done, see which containers that exists

```
docker ps -a
```

Which should show `hello-world` as `IMAGE` with a `CONTAINER ID`.

Go back to the **Ubuntu terminal** and see which containers that exists:

```
docker ps -a
```

Which should show `hello-world` as `IMAGE` with the same `CONTAINER ID` as in Windows.