Ubuntu on Windows for computational biology

James Lloyd

Abstract

With the Fall Creators update to **Windows 10** comes the ability for all to install **Ubuntu** on your Windows 10 machine by simply downloading the app from the official **Windows App Store**. Rather than using a tool like **Cygwin**, which replicates a **UNIX(-like)**experience with some of the same command line tools, **Ubuntu on Windows** is the full **Linux** experience. Unlike using Cygwin, you can install bioinformatics software like **Salmon** or **Cufflinks** with ease. This protocol from Bad Grammar, Good Syntax provides a detailed method of how to install Ubuntu, setting up Python, and installing binaries of software for compbio.

Citation: James Lloyd Ubuntu on Windows for computational biology. **protocols.io**
dx.doi.org/10.17504/protocols.io.sfuebnw
Published: 10 Aug 2018

Guidelines

In the past, you could try to dual boot your system so that it has two operating systems or run Ubuntu in a virtual machine within Windows, but I think this method is nice and clean. Others will disagree.

What you won't get is the Ubuntu graphical user interface. You are still stuck with running Windows 10. If that is a problem for you, then dual booting or a virtual machine is probably your best option. Instead, what Ubuntu on Windows offers is a simple way to get access to the command line from Ubuntu. Not only does that mean that you can run software designed and compiled to run on Linux, but you can also use all the cool tools of Ubuntu like apt-get to seamlessly install programs, keep them up-to-date, and fetch dependences.

An alternative guide to Linux on Windows using Conda:
https://github.com/kapsakcj/win10-linux-conda-how-to/blob/master/README.md

Before start

You need a computer with Windows 10 installed and kept up to date.

Protocol

Installing Ubuntu

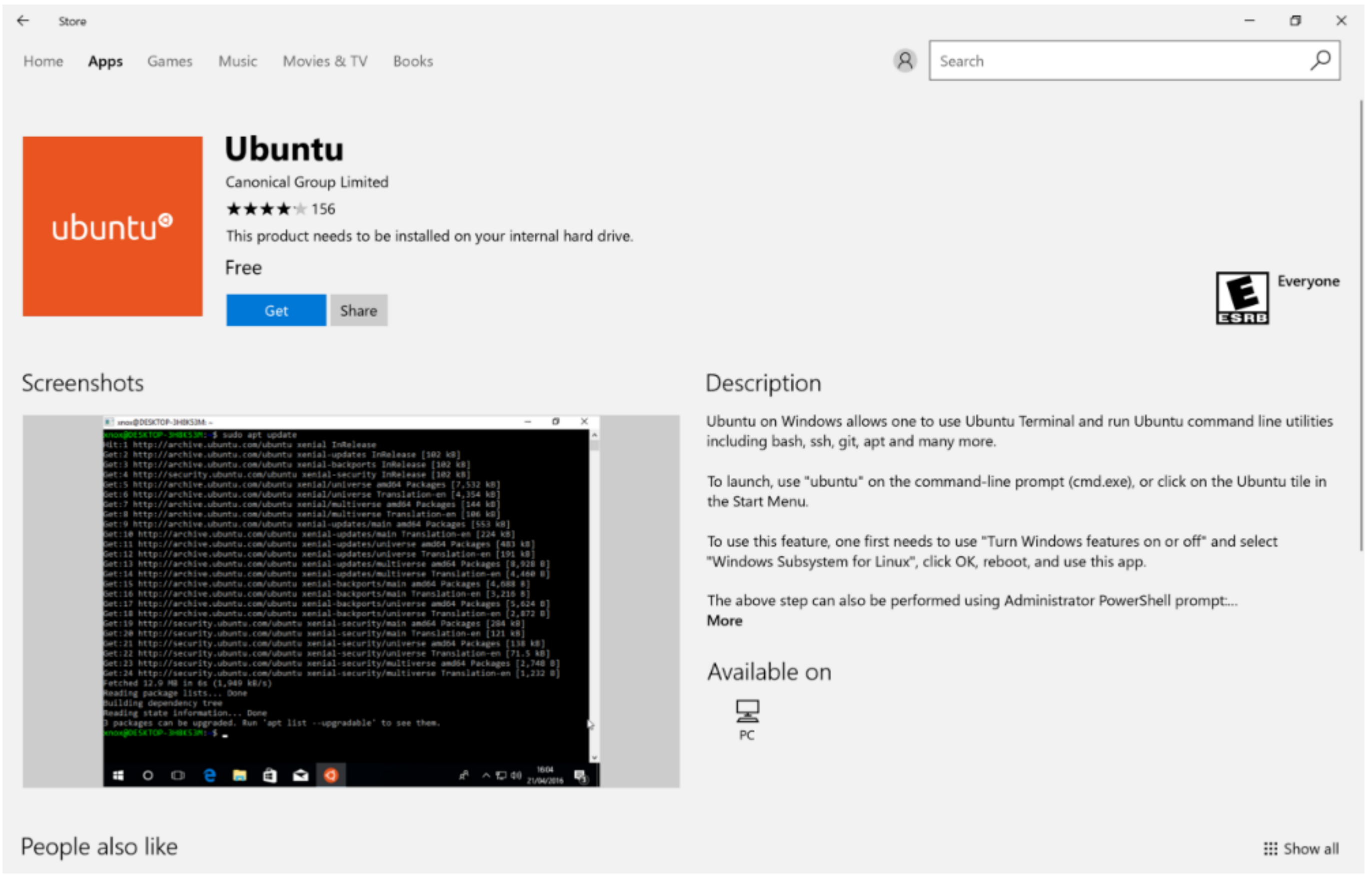
Step 1.

Ensure your Windows10 has the 2018 Fall Creators update (version 1709 or greater).

Installing Ubuntu

Step 2.

Go to the Windows App store and download Ubuntu app:



Ubuntu

Canonical Group Limited

★★★★☆ 156

This product needs to be installed on your internal hard drive.

Free

Description

Ubuntu on Windows allows one to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

To launch, use "ubuntu" on the command-line prompt (cmd.exe), or click on the Ubuntu tile in the Start Menu.

To use this feature, one first needs to use "Turn Windows features on or off" and select "Windows Subsystem for Linux", click OK, reboot, and use this app.

The above step can also be performed using Administrator PowerShell prompt:...
More

Available on

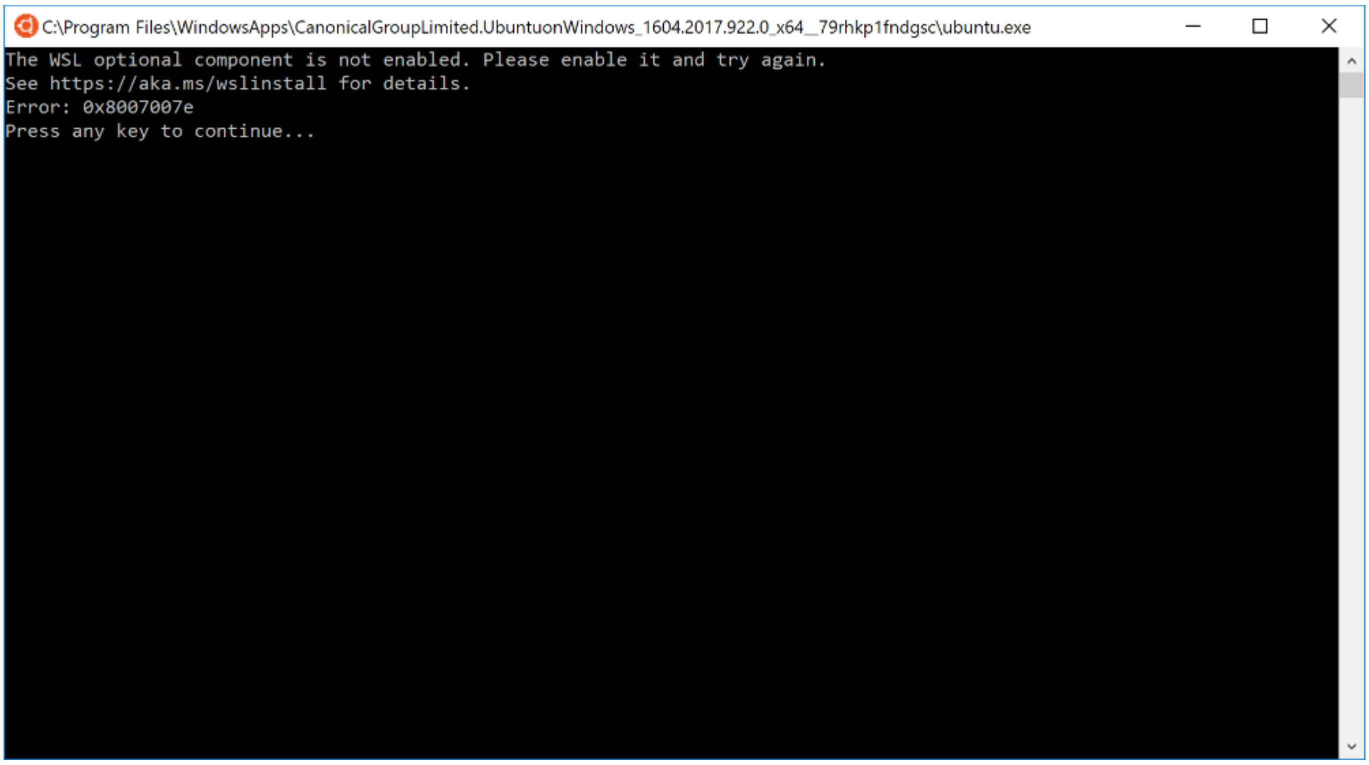
```
xnox@DESKTOP-3H8K53M: ~
xnox@DESKTOP-3H8K53M:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Get:3 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Get:5 http://archive.ubuntu.com/ubuntu xenial/universe amd64 Packages [7,532 kB]
Get:6 http://archive.ubuntu.com/ubuntu xenial/universe Translation-en [4,354 kB]
Get:7 http://archive.ubuntu.com/ubuntu xenial/multiverse amd64 Packages [144 kB]
Get:8 http://archive.ubuntu.com/ubuntu xenial/multiverse Translation-en [106 kB]
Get:9 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [553 kB]
Get:10 http://archive.ubuntu.com/ubuntu xenial-updates/main Translation-en [224 kB]
Get:11 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [483 kB]
Get:12 http://archive.ubuntu.com/ubuntu xenial-updates/universe Translation-en [191 kB]
Get:13 http://archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 Packages [8,928 B]
Get:14 http://archive.ubuntu.com/ubuntu xenial-updates/multiverse Translation-en [4,460 B]
Get:15 http://archive.ubuntu.com/ubuntu xenial-backports/main amd64 Packages [4,688 B]
Get:16 http://archive.ubuntu.com/ubuntu xenial-backports/main Translation-en [3,216 B]
Get:17 http://archive.ubuntu.com/ubuntu xenial-backports/universe amd64 Packages [5,624 B]
Get:18 http://archive.ubuntu.com/ubuntu xenial-backports/universe Translation-en [2,872 B]
Get:19 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [284 kB]
Get:20 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [121 kB]
Get:21 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [138 kB]
Get:22 http://security.ubuntu.com/ubuntu xenial-security/universe Translation-en [71.5 kB]
Get:23 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 Packages [2,748 B]
Get:24 http://security.ubuntu.com/ubuntu xenial-security/multiverse Translation-en [1,232 B]
Fetched 12.9 MB in 6s (1,949 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
xnox@DESKTOP-3H8K53M:~$ _
```

16:04
21/04/2016

Installing Ubuntu

Step 3.

Once the app is installed, open it up. You will see this terminal appear:



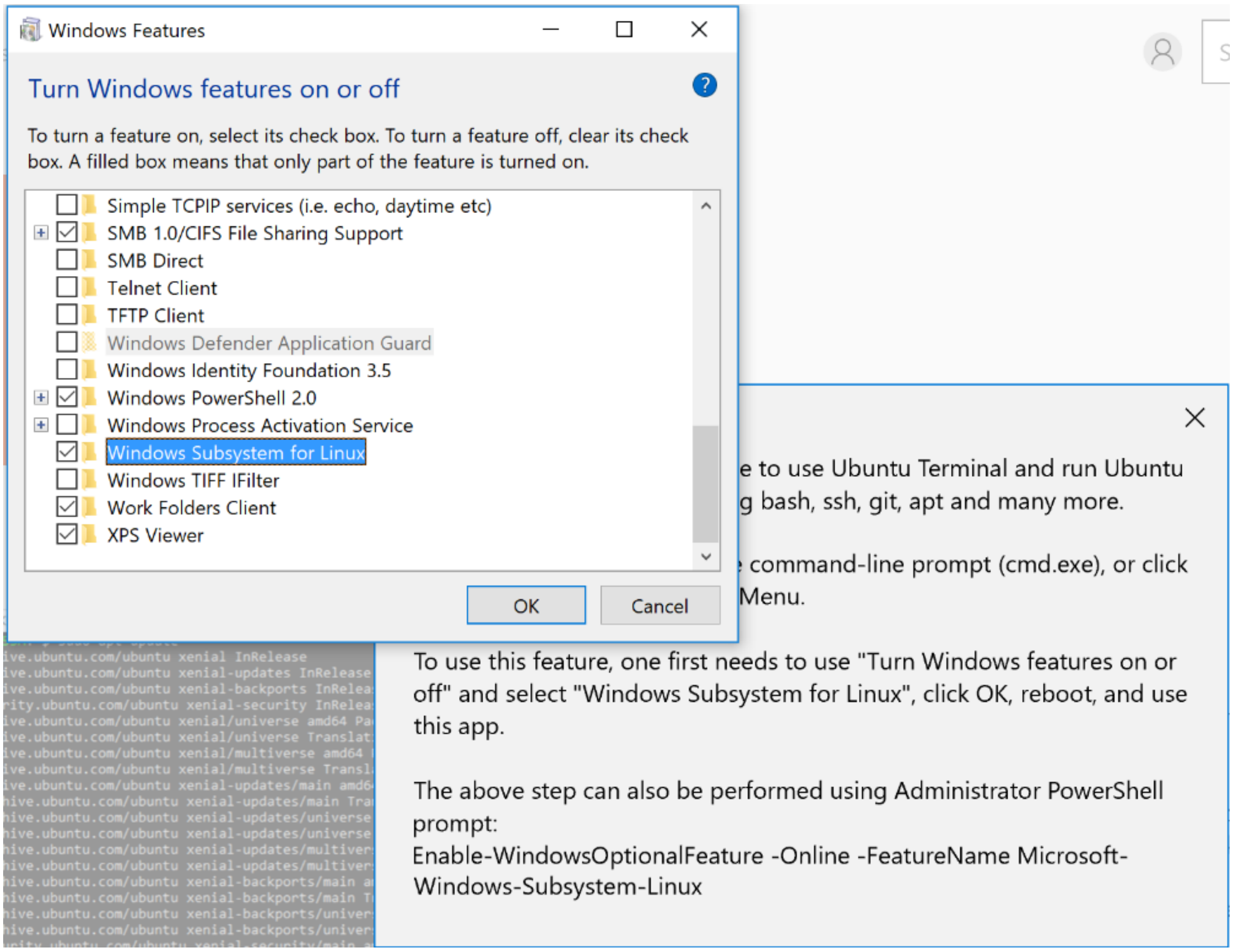
C:\Program Files\WindowsApps\CanonicalGroupLimited.UbuntuonWindows_1604.2017.922.0_x64__79rhkp1fndgsc\ubuntu.exe

```
The WSL optional component is not enabled. Please enable it and try again.
See https://aka.ms/wslinstall for details.
Error: 0x8007007e
Press any key to continue...
```

Installing Ubuntu

Step 4.

You need to enable an option in Windows first. Open the Start menu on Windows and start typing 'Windows Features' until a program called **Turn Windows Features on or off** appears. Click on it and find 'Windows Subsystems for Linux, then enable it (tick the box):



To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- [ ] Simple TCPIP services (i.e. echo, daytime etc)
- [✓] SMB 1.0/CIFS File Sharing Support
- [ ] SMB Direct
- [ ] Telnet Client
- [ ] TFTP Client
- [ ] Windows Defender Application Guard
- [ ] Windows Identity Foundation 3.5
- [✓] Windows PowerShell 2.0
- [ ] Windows Process Activation Service
- [✓] Windows Subsystem for Linux
- [ ] Windows TIFF IFilter
- [✓] Work Folders Client
- [✓] XPS Viewer

e to use Ubuntu Terminal and run Ubuntu
g bash, ssh, git, apt and many more.

command-line prompt (cmd.exe), or click
Menu.

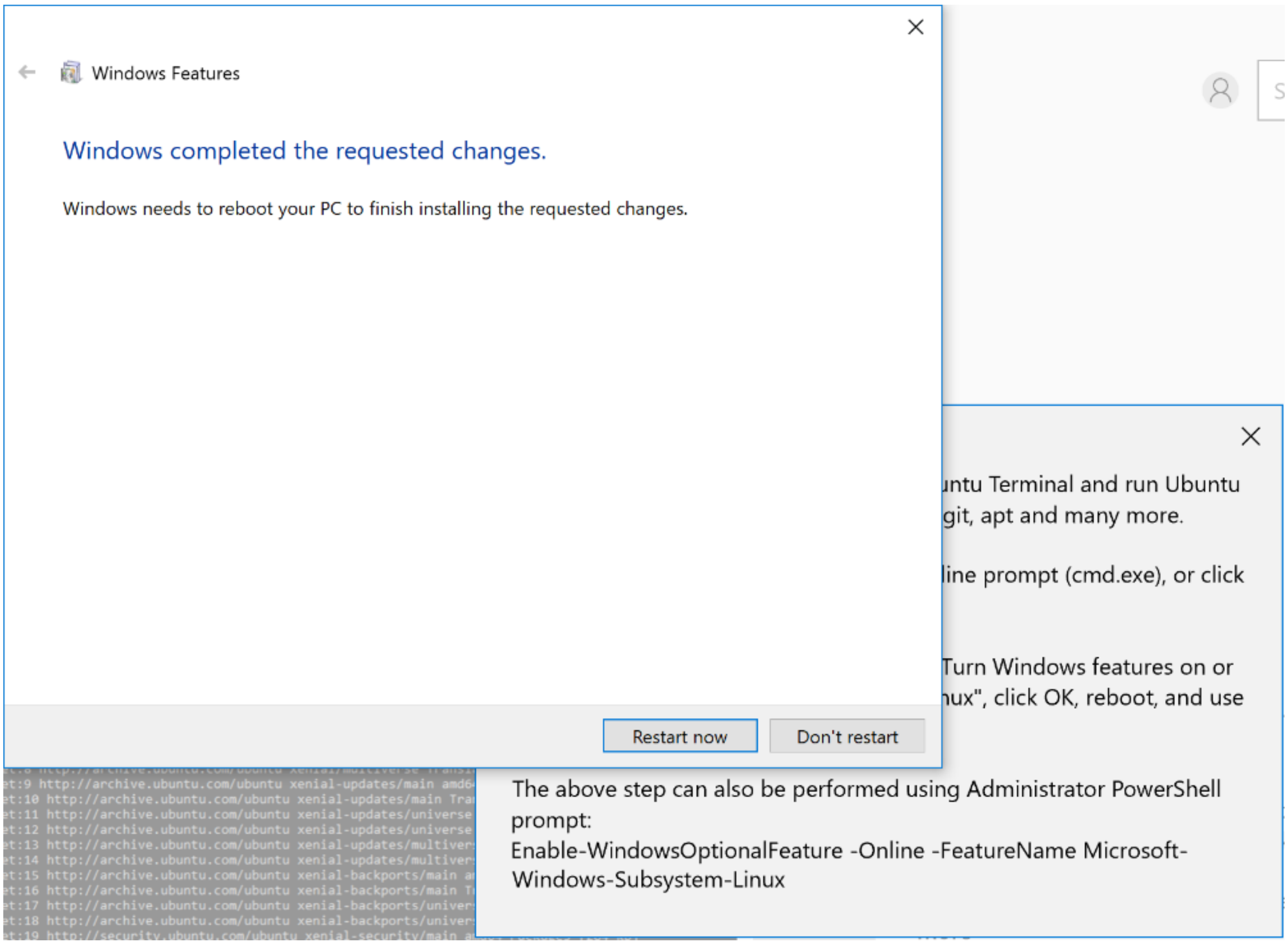
To use this feature, one first needs to use "Turn Windows features on or off" and select "Windows Subsystem for Linux", click OK, reboot, and use this app.

The above step can also be performed using Administrator PowerShell prompt:
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

Installing Ubuntu

Step 5.

Restart your PC:



Windows completed the requested changes.

Windows needs to reboot your PC to finish installing the requested changes.

untu Terminal and run Ubuntu
git, apt and many more.

line prompt (cmd.exe), or click

Turn Windows features on or
nux", click OK, reboot, and use

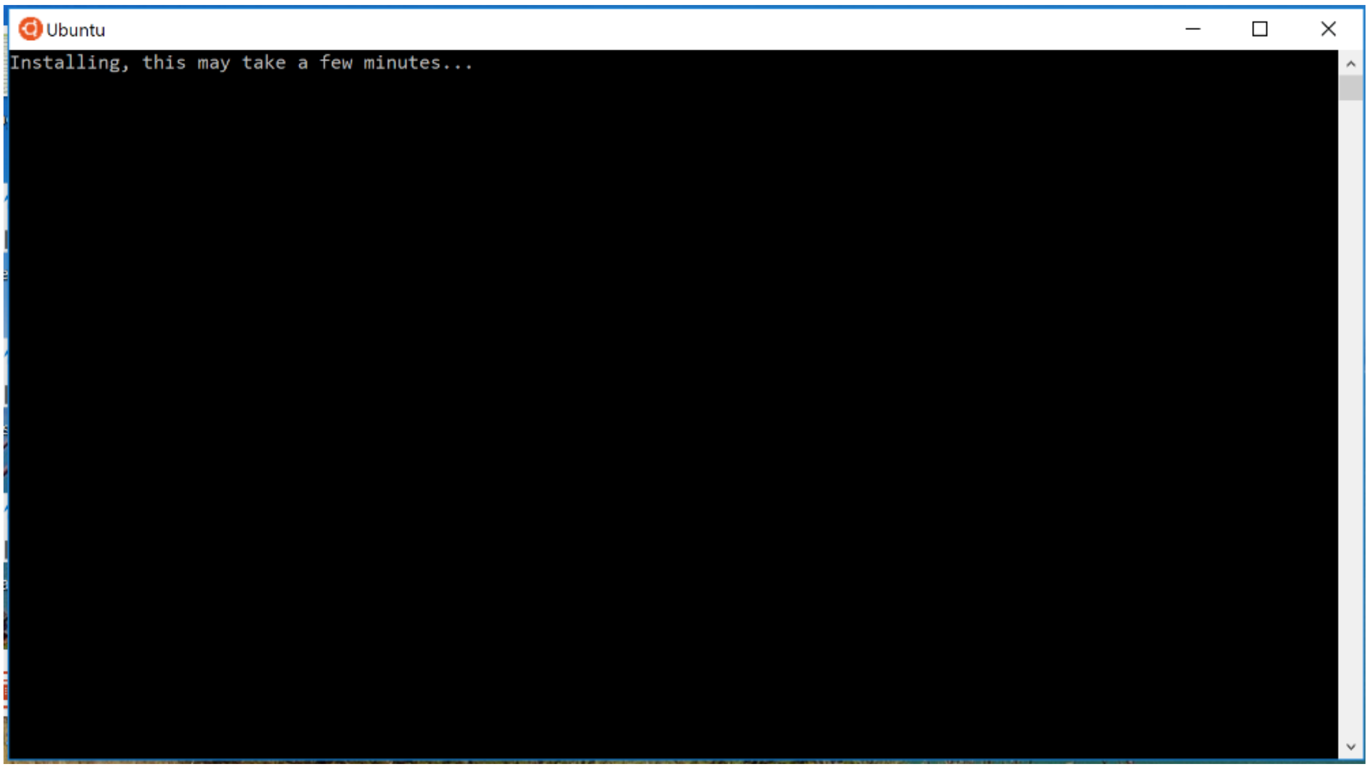
The above step can also be performed using Administrator PowerShell prompt:
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

```
et:9 http://archive.ubuntu.com/ubuntu xenial-updates/main amd6
et:10 http://archive.ubuntu.com/ubuntu xenial-updates/main Tra
et:11 http://archive.ubuntu.com/ubuntu xenial-updates/universe
et:12 http://archive.ubuntu.com/ubuntu xenial-updates/universe
et:13 http://archive.ubuntu.com/ubuntu xenial-updates/multiver
et:14 http://archive.ubuntu.com/ubuntu xenial-updates/multiver
et:15 http://archive.ubuntu.com/ubuntu xenial-backports/main a
et:16 http://archive.ubuntu.com/ubuntu xenial-backports/main T
et:17 http://archive.ubuntu.com/ubuntu xenial-backports/univer
et:18 http://archive.ubuntu.com/ubuntu xenial-backports/univer
et:19 http://security.ubuntu.com/ubuntu xenial-security/main a
```

Installing Ubuntu

Step 6.

Now try opening the Ubuntu app again. This time it should take a few moments to install itself:



```
Installing, this may take a few minutes...
```

Installing Ubuntu

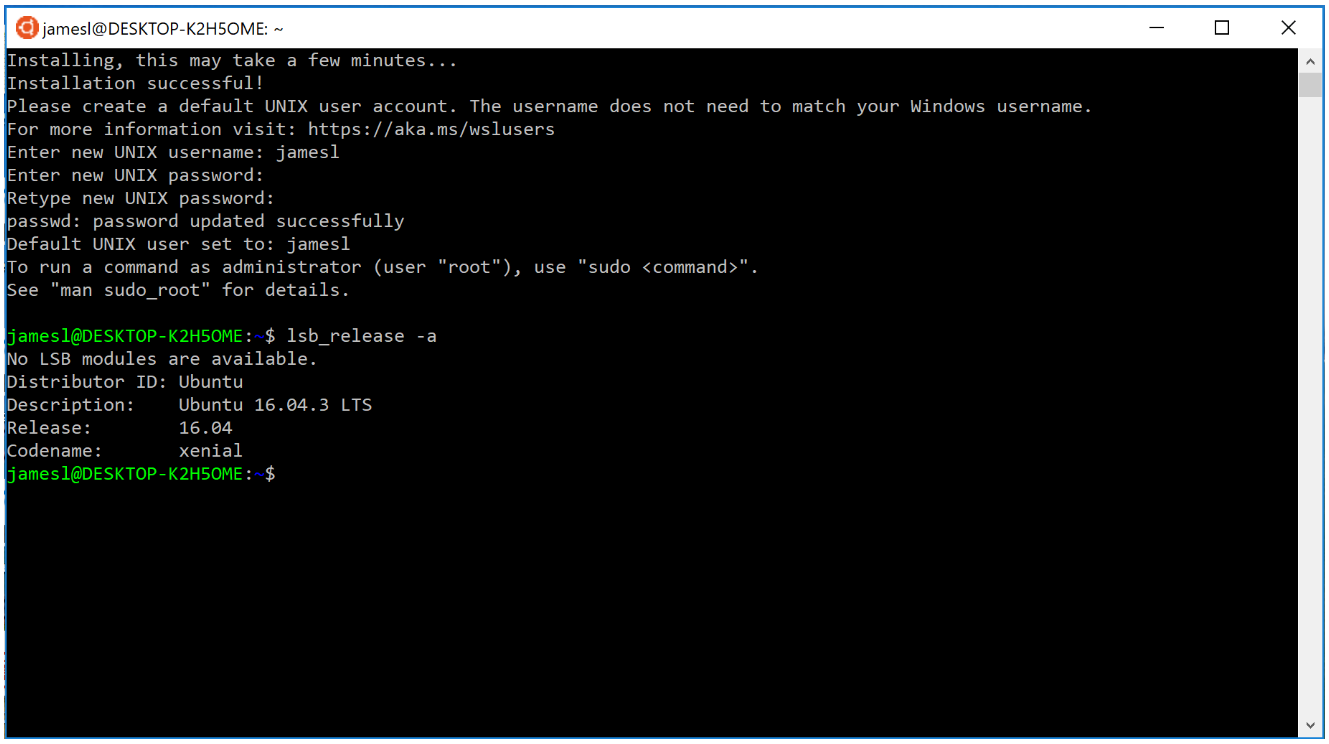
Step 7.

Once it has finished installing, it will ask for your to select a username and password. These are independent from your Windows username and password. Select a username and passphrase that you will remember (or store in a password manager) and then enter them. Then you can check the exact version you are using with this command:

cmd COMMAND

```
$ lsb_release -a
```

NOTES



```
Installing, this may take a few minutes...
Installation successful!
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: jamesl
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Default UNIX user set to: jamesl
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

jamesl@DESKTOP-K2H5OME:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.3 LTS
Release:        16.04
Codename:       xenial
jamesl@DESKTOP-K2H5OME:~$
```

Installing Ubuntu
Step 8.

Now you are ready to start using your Ubuntu on Windows.

NOTES

One thing that you might notice is that you cannot do Ctrl+C or Ctrl+P for copy/paste. This is a pain but to copy text that you have highlighted in the Ubuntu app, you can right-click with your mouse. When you have texted highlighted, right-click will then paste what you have copied.

Installing Ubuntu
Step 9.

Right now, you are in a special drive for your Ubuntu files that are separate from your normal storage drive. To access your normal files (what is normally the C drive), go to where they are mounted at:

cmd COMMAND
```
$ /mnt/c
```

NOTES

This is where you wish you had never used spaces in file/folder names, as bash does not handle spaces in filenames well. If this is a big issue for you, I would suggest going back and renaming files/folders you plan to access from the command line.

Installing Ubuntu
Step 10.

Now update the Ubuntu OS that is running.

cmd COMMAND

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

NOTES

While my version number did not increase, doing this cleared up an issue I was having with some other software I was trying to install - so I advice that you do update it now with the two commands above.

If you are new to using a UNIX(-like) system, sudo makes you a superuser: you are able to change important things in you system you cannot do normally. It will ask for your password you set Ubuntu up with.

Setting up Python

Step 11.

If you use a lot of **Python** in your work, it is rather simple to get Python running on Ubuntu. There are many ways to install Python. One way would be to use Ubuntu's apt-get. Another way is to use Conda. Each of these methods installs Python in a different environment, so just be aware. Ubuntu comes with an install of Python 3 out of the box

cmd COMMAND

```
$ which python3
/usr/bin/python3
$ python3 --version
Python 3.5.2
```

Setting up Python

Step 12.

Given that a lot of the software and code used is from Python version 2.7, it is worth having a legacy version around. To install Python 2.7, use apt-get like so:

cmd COMMAND

```
$ sudo apt install python
```

NOTES

This will ask for the password that you set Ubuntu up with.

Setting up Python

Step 13.

Now you have Python 2.7 (which you can call with python or python2 as opposed to python3):

cmd COMMAND

```
$ which python
/usr/bin/python
$ python --version
Python 2.7.12
```

Setting up Python

Step 14.

With advice from **here**, install these very useful tools for Python on Ubuntu:

cmd COMMAND
```
$ sudo apt-get install build-essential
$ sudo apt-get install python-dev
$ sudo apt-get install python3-dev
```

Setting up Python

Step 15.

Now for something really important, to install **pip**. pip is the official package manager for Python modules.

cmd COMMAND
```
$ sudo apt-get install python-pip
$ sudo apt-get install python3-pip
```

NOTES

It makes installing new packages for Python SOOOO much easier in the long run.

The issue with apt-get is that while it deals with all of the dependency issues (which is really important), it usually installs a really out of date version of the python package. In constrast, pip doesn't deal with the dependencies but it does install the latest version of the package. So I suggest initally installing with apt-get and then updating with pip (see below).

Setting up Python

Step 16.

Now to install pandas using apt-get:

cmd COMMAND
```
$ sudo apt-get install python-pandas
$ sudo apt-get install python3-pandas
```

NOTES

By installing pandas with apt-get, we also install many other important Python packages like numpy and matplotlib as dependances, which is a massive timesaver, compared to going after each on individually.

Setting up Python

Step 17.

The latest scipy version is 1.0.0. If you look at the version of scipy that has been installed, you can see that we are well out of date:

cmd COMMAND
```
$ pip show scipy
Name: scipy
```

```
Version: 0.17.0
Summary: SciPy: Scientific Library for Python
Home-page: http://www.scipy.org
Author: SciPy Developers
Author-email: scipy-dev@scipy.org
License: BSD
Location: /usr/lib/python2.7/dist-packages
Requires:
```

Setting up Python

Step 18.

You can update scipy from version 0.17.0 to 1.0.0 with pip:

cmd COMMAND

```
$ sudo pip install scipy --upgrade
```

Setting up Python

Step 19.

Now you have the latest version of scipy:

cmd COMMAND

```
$ pip show scipy
Name: scipy
Version: 1.0.0
Summary: SciPy: Scientific Library for Python
Home-page: https://www.scipy.org
Author: SciPy Developers
Author-email: scipy-dev@python.org
License: BSD
Location: /usr/local/lib/python2.7/dist-packages
Requires: numpy
```

Setting up Python

Step 20.

This is great, but beware that the scipy for Python3 has NOT been updated! So do not forget to do that one too!

cmd COMMAND

```
$ sudo pip3 install scipy --upgrade
```

Installing binaries of software for compbio

Step 21.

There are a number of different programs out there for computational biologists that are not going to be downloadable with apt-get. With these, you might need to download the source code and compile the software yourself. Or you can install them with Conda/bioconda (highly reccomended when possible) or you can down load a pre-compiled binary. You can download a binary and upzip it like so:

cmd COMMAND

```
$ cd /usr/local/bin
$ sudo wget https://github.com/COMBINE-lab/salmon/releases/download/v0.8.2/Salmon-0.8.2_lin
ux_x86_64.tar.gz
$ sudo tar -xvzf Salmon-0.8.2_linux_x86_64.tar.gz
```

NOTES

Luckily, many developers of these tools put out binaries for systems (Mac, Linux, and sometimes even Windows). A binary is a pre-compiled file that you can call to run the program. So here I will go through the simple task of downloading the latest binary for the RNA-seq quantification software **Salmon**. But a similar approach should work for other programs available as binaries.

Installing binaries of software for compbio

Step 22.

To run Salmon, simply enter the path to the binary in the terminal.

cmd COMMAND

```
 /usr/local/bin/Salmon-0.8.2_linux_x86_64/bin/salmon
Salmon v0.8.2
Usage:  salmon -h|--help or
        salmon -v|--version or
        salmon -c|--cite or
        salmon [--no-version-check] <COMMAND> [-h | options]
Commands:
     cite  Show salmon citation information
     index Create a salmon index
     quant Quantify a sample
     swim  Perform super-secret operation
```

Installing binaries of software for compbio

Step 23.

Now you can run Salmon on your Windows PC (through Ubuntu)!

NOTES

The issue here is having to remember this long path every time you wanted to run Salmon. You could modify your bashrc to add Salmon to your path. But one advantage of entering the path every time, is that you know exactly which version of Salmon (or any software) you are calling every time you use it.

Hopefully now you can start to perform some of your work on your local (Windows) machine but with the comfort of having all of the tools for Linux at your fingertips.