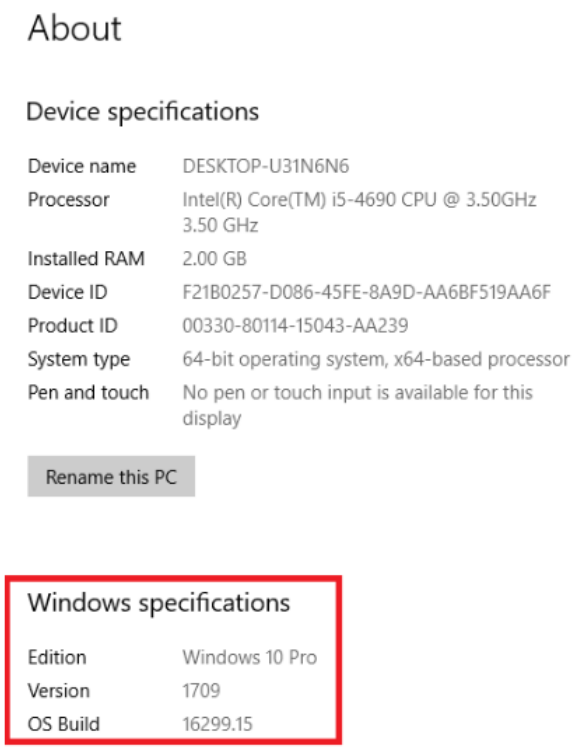
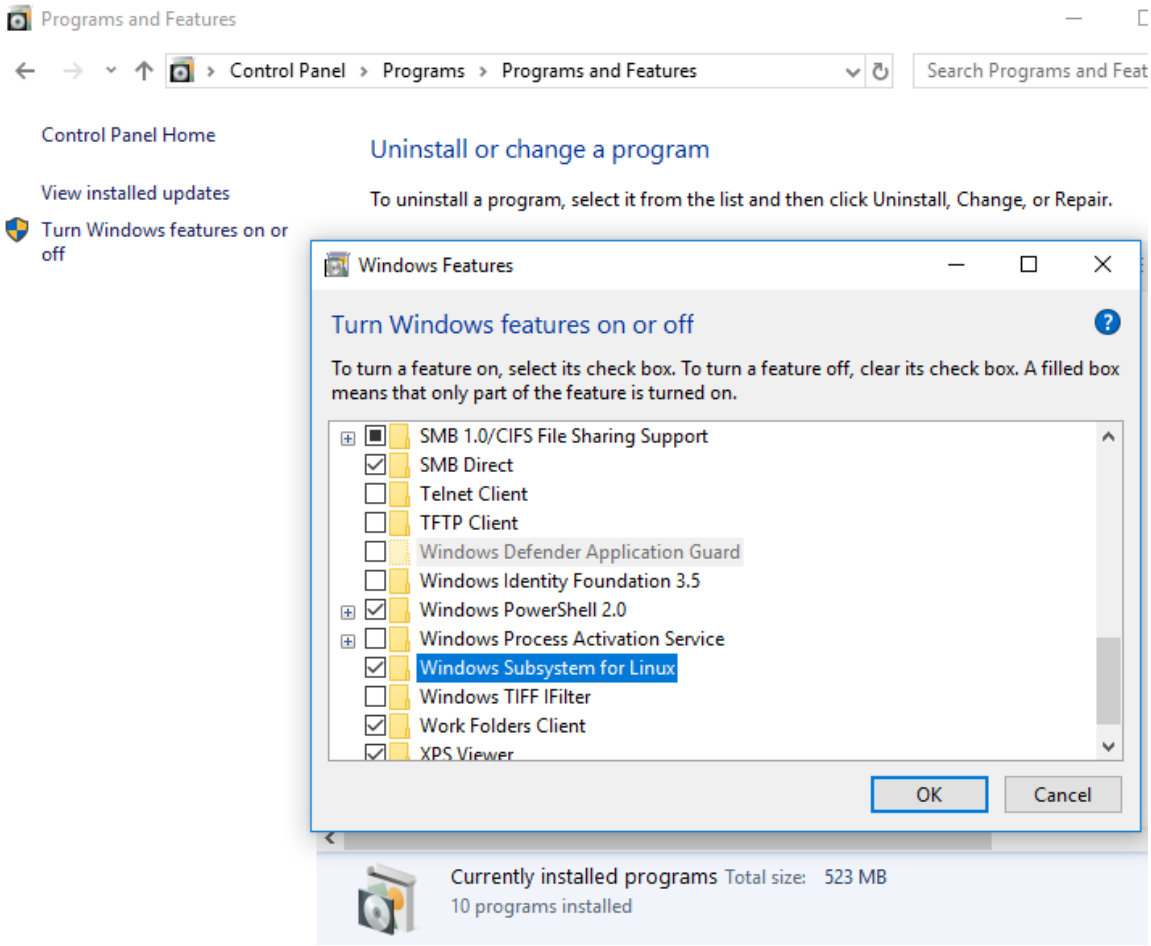
Install Linux Subsystem (WSL) and OpenSSH in Windows 10

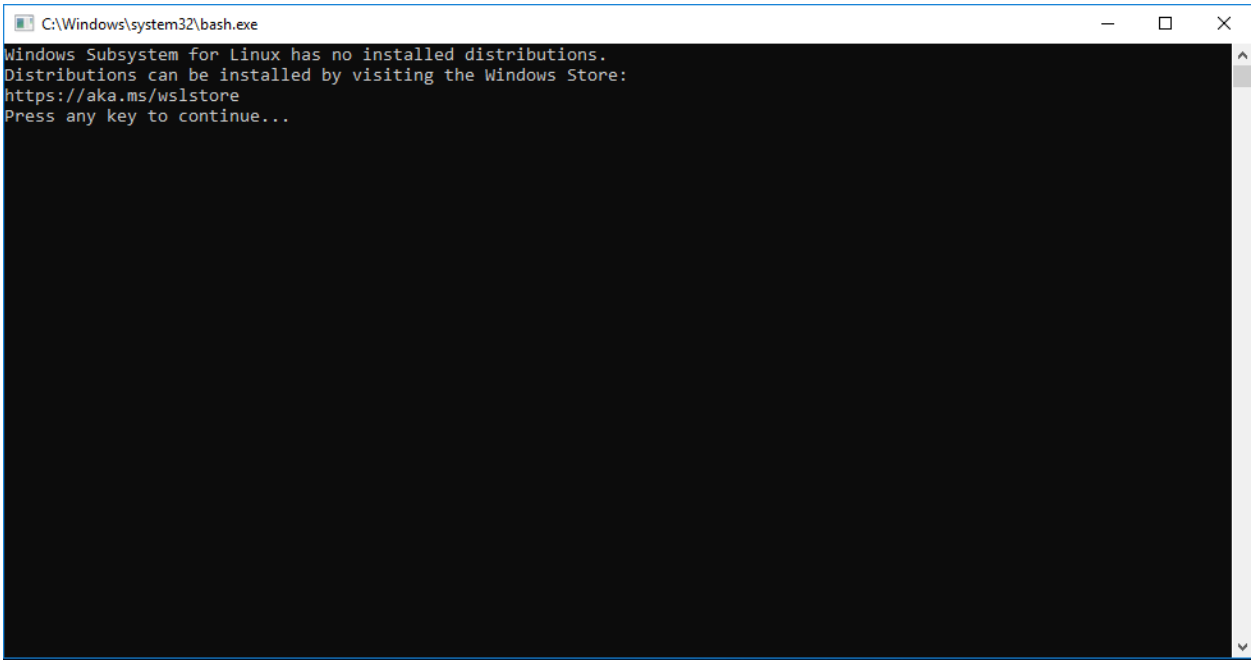
1. First, you'll want to check your Version of Windows 10 and upgrade to a Version that supports the Linux Subsystem: Settings -> System -> About



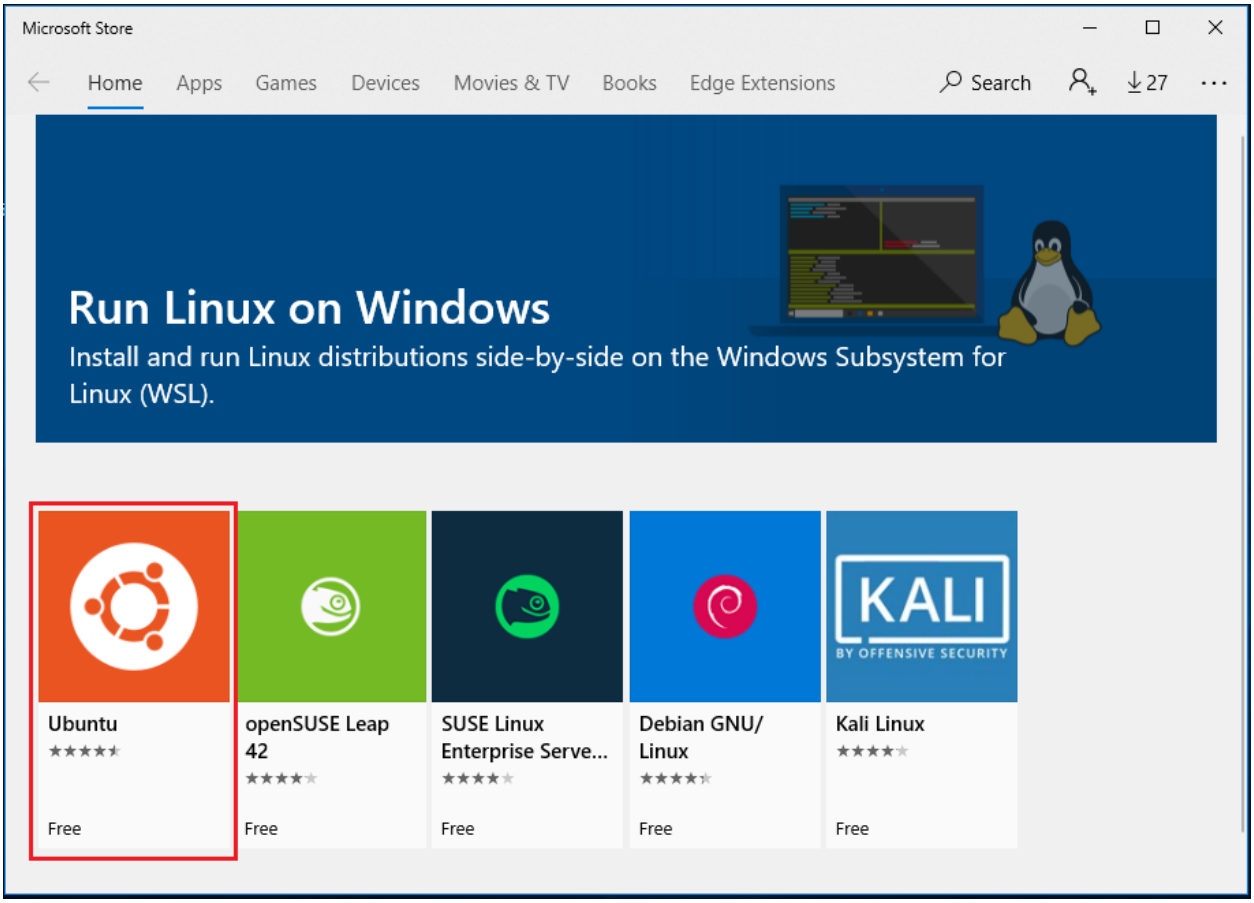
2. Once you're updated, Run the 'appwiz.cpl' command from the Run dialog box (Windows Key + R) which will open up your 'Programs and Features'. Open 'Turn Windows features on or off' which is located on the left side of the explorer window. Find 'Windows Subsystem for Linux' and install it.



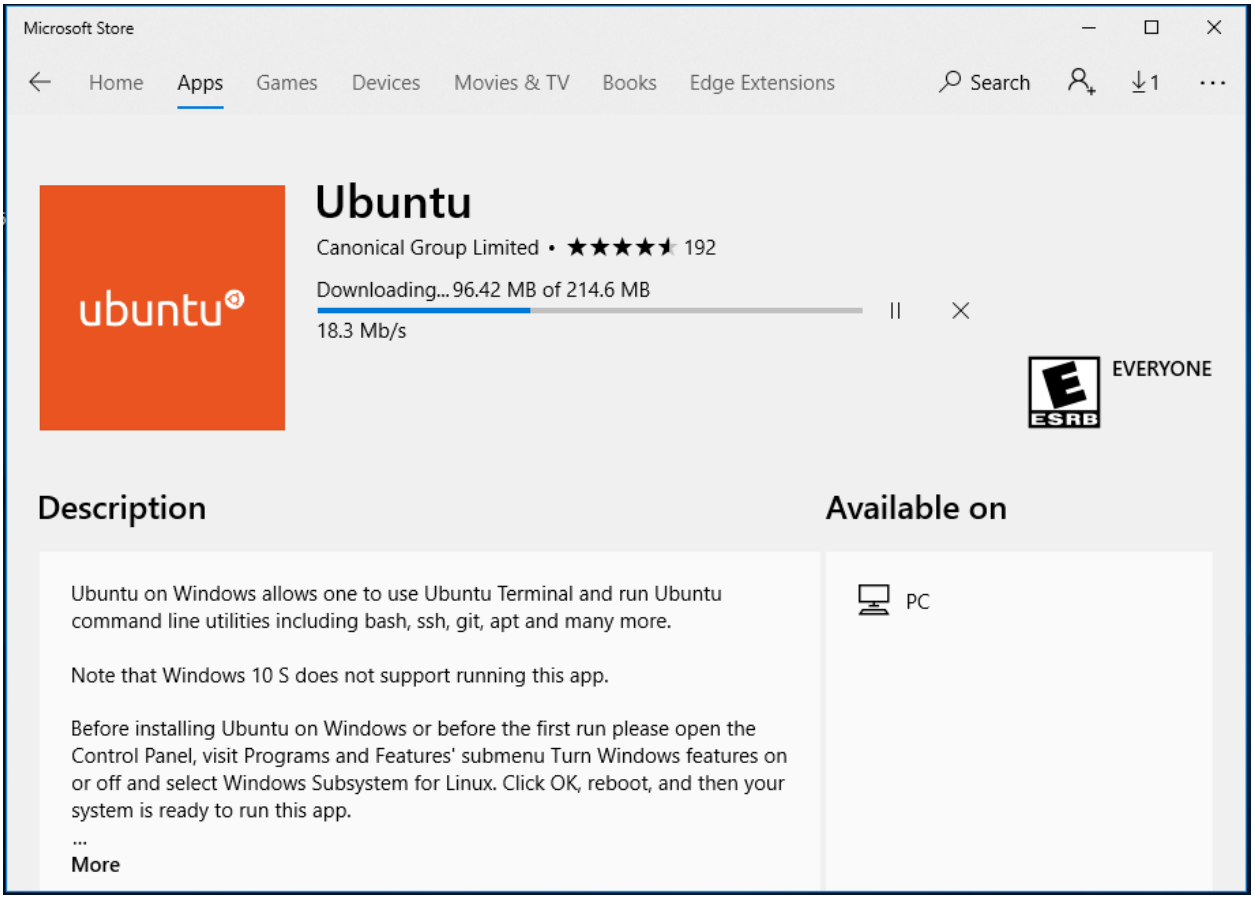
3. Restart the system to finish setting up the Linux Subsystem installation. Run the ‘bash’ command from the Run dialog box (Windows Key + R).



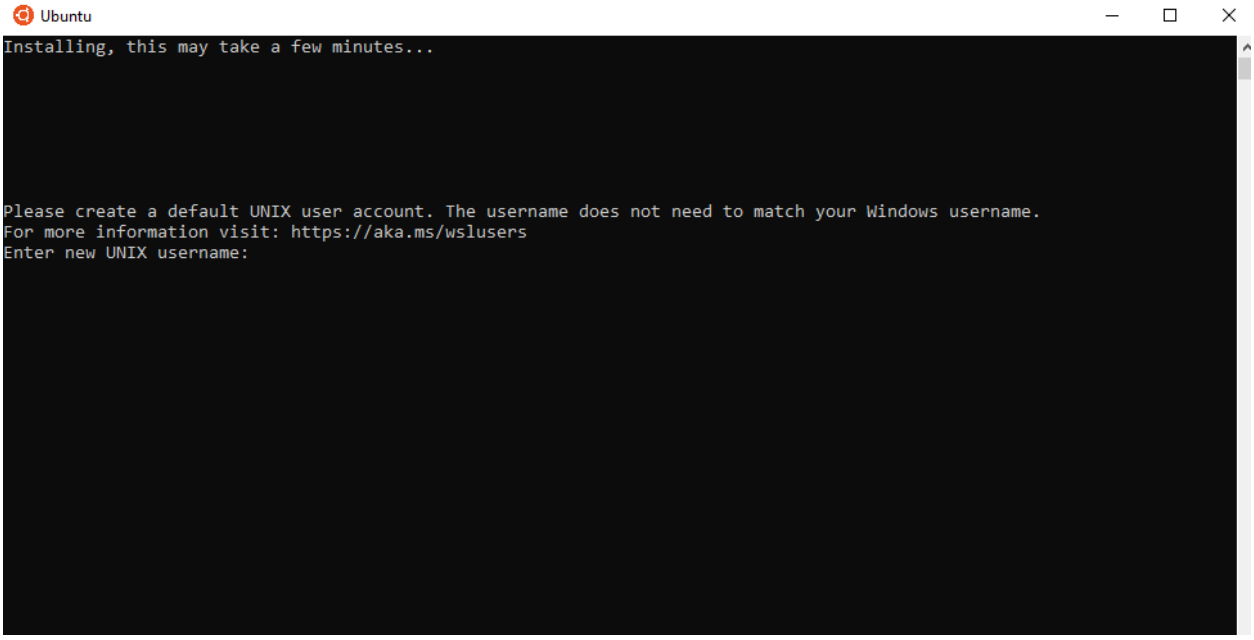
4. As you can see, the subsystem package is not actually installed. You'll need to open the app store to finish installing the package.

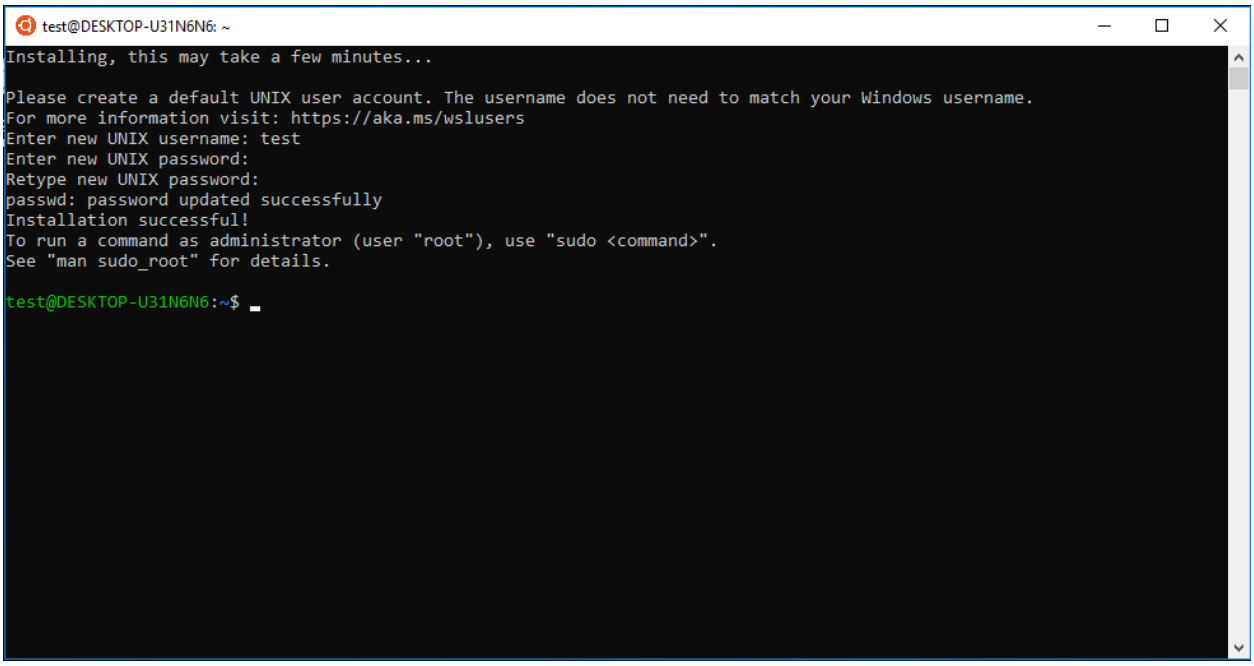


5. We'll focus on Ubuntu Linux in this article, but feel free to select your desired Linux flavor. Select Ubuntu to continue the installation.



6. After installation is complete you'll need to set a username and password. It does not have to be your Windows username. The user created here becomes the default administrator for Ubuntu and will be used for the sudo command.





7. Assuming you have installed and setup your favorite Linux distribution under the Windows 10 WSL, you have to make a couple of tweaks before you can actually SSH into it. In the WSL bash shell, install/reinstall OpenSSH by:

   sudo apt-get purge openssh-server

   sudo apt-get install openssh-server

8. In the WSL bash shell, edit the "/etc/ssh/sshd_config" configuration file:

   sudo nano /etc/ssh/sshd_config

   1) set the port to 2222 (The reason for changing the port is that the default port 22 might be used by windows. Also, as noted in the original guide, you should setup the machine for SSH key-based access, not just for security but also for efficiency.)

   2) disallow root login

   2) allow password authentication

   3) add a line at the end of the file that says: AllowUsers yourusername

   4) add/modify a line to disable privilege separation

```
Include /etc/ssh/sshd_config.d/*.conf

Port 2222
```

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
```

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
```

```
AllowUsers wang109
UsePrivilegeSeparation no
```

9. Open up port 2222 in the windows firewall:

   1) In the Windows Start menu, type "WF.msc".

   2) In the "Windows Firewall with Advanced Security" section, click on "Inbound Rules".

   3) Add a new rule for TCP 2222 and allow the connection:

      "Actions" > "New Rule …"

      "What type of rule would you like to create?": "Port"

      "Does this rule apply to TCP or UDP? "TCP"

      "Does this rule apply to all local ports or specific local ports": "Specific local ports: 2222"

      "What action should be taken when a connection matches the specified conditions": "Allow the connection"

"When does this rule apply" (check all boxes)

10. Restart SSH serer in the WSL bash shell:

    sudo service ssh –full-restart

```
 * Stopping OpenBSD Secure Shell server sshd                                    [ OK ]
 * Starting OpenBSD Secure Shell server sshd
```

11. Get the IP address of the WSL machine by running ifconfig:

    ifconfig

```
% ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 1500
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x0<global>
        loop  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wifi0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.10  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 2600:6c54:7b80:1200:8c11:138f:f2a:5468  prefixlen 64  scopeid 0x0<global>
        inet6 2600:6c54:7b80:1200:ac05:df36:5697:b4b0  prefixlen 128  scopeid 0x0<global>
        inet6 fe80::8c11:138f:f2a:5468  prefixlen 64  scopeid 0x0<global>
        unspec F8-59-71-07-CB-F0-00-00-00-00-00-00-00-00-00-00  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

12. Now you should be able to connect to your Windows subsystem for linux using a ssh client like Putty.