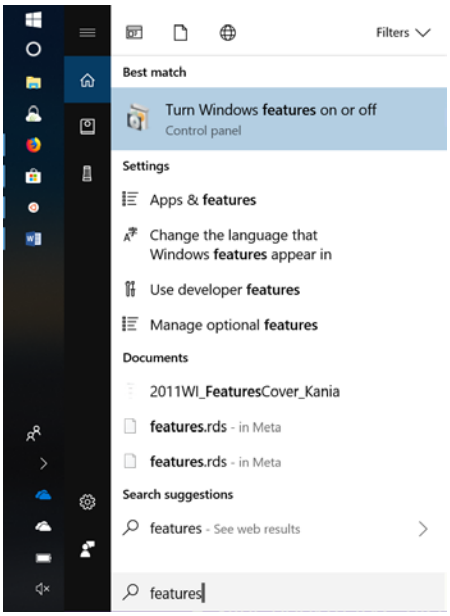
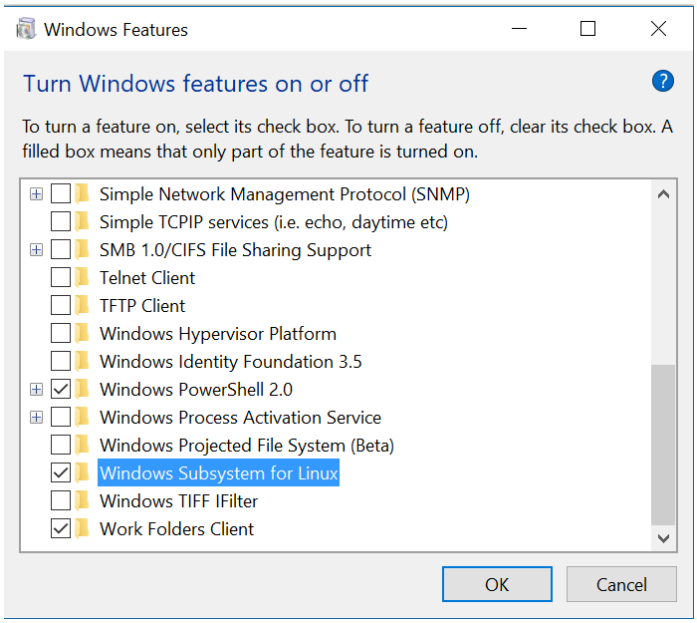
Installing Linux OS UBUNTU on Windows 10

Inspired by “How to Run Linux/Bash on Windows 10 Using the Built-In…” by Corey Schafer, YouTube

1. Search on “Features” and select “Turn Windows features on or off”. Scroll to bottom of features, select “Windows Subsystem for Linux”, and click ok.

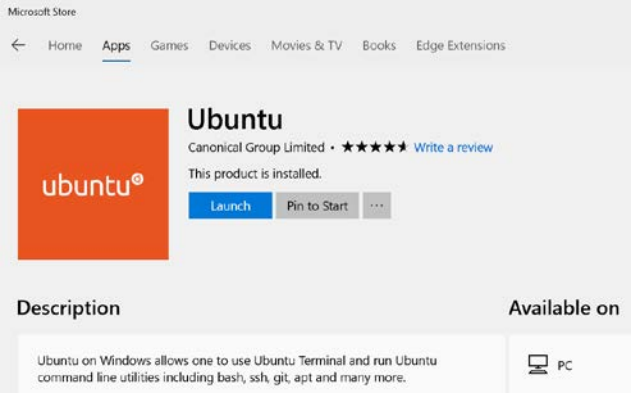




Your computer will restart.

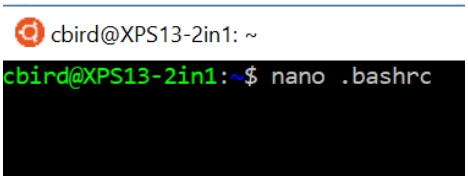
2. Goto the Microsoft Store and search on “Linux”. Select “Get the apps” button

3. Select the UBUNTU installer and click “Install” button. When complete select “Launch”



4. When prompted, type in your username and a password for UBUNTU.

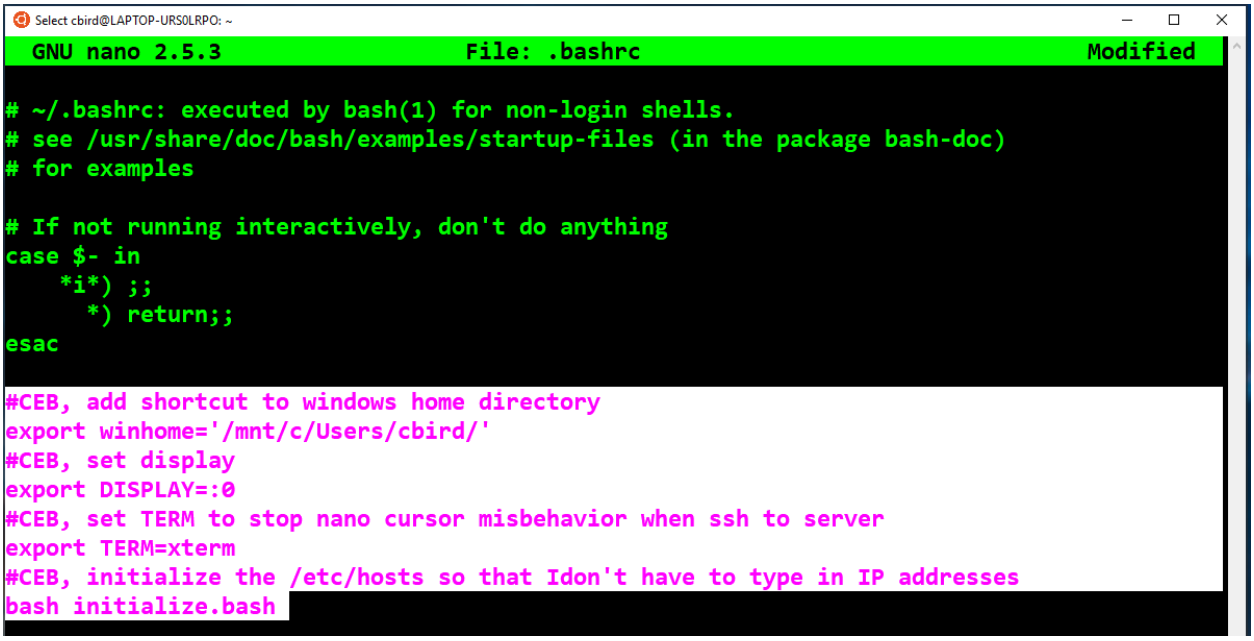
5. Open the .bashrc file using nano
   !!!You can do damage here by accidently changing or deleting text, so only do exactly what is asked in the next step!!!



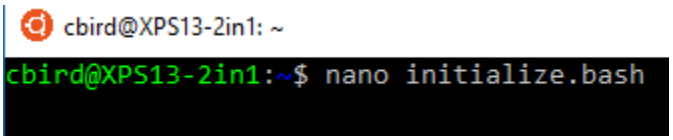
6. Also realize that you cannot use the mouse to move the cursor in nano, you have to use the arrow keys on the keyboard. **Right clicking** the mouse in nano will **paste** in the location of the cursor, not the mouse.

   Copy (ctrl-c) and paste (right click) the following text in the location indicated in the image below to modify the .bashrc file to add a shortcut to your windows home directory, set the DISPLAY so gui apps will run, set the TERM so that nano will work correctly, and run the initialize.bash script fix the domain name bug, then save the changes and exit (ctrl-x) .

```
#CEB, add shortcut to windows home directory

export winhome='/mnt/c/Users/cbird/'

#CEB, set display

export DISPLAY=:0

#CEB, set TERM to stop nano cursor misbehavior when ssh to
server

export TERM=xterm

#CEB, initialize the /etc/hosts so that I don't have to
type in IP addresses

bash initialize.bash
```



7. Create a script file called initialize.bash in your home directory that will allow you to ssh to domain names rather than just ip addresses



8. Copy (ctrl-c) and paste (right click) the following text into initialize.bash then save and close (ctrl-x)

```
#!/bin/bash
#this script initializes my custom settings in UBUNTU on WINDOWS
sudo chmod 777 /etc/hosts
cp ~/hosts /etc
sudo chmod 555 /etc/hosts
```

9. Copy the file /etc/hosts to your home directory with the following command (`sudo cp /etc/hosts .`). We are going to modify this file to attach a name to the TAMUCC supercomputer's ip address.

10. Change the permissions on the hosts file by entering the following command

```
sudo chmod 777 hosts
```

11. Open the hosts file. Be careful to make only the changes that are asked

12. Copy and paste the following line to the hosts file in the location indicated below:

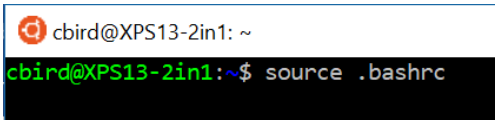
```
10.20.1.189      hpcm.tamucc.edu
```

```
# This file is automatically generated by WSL based on the Windows hosts
 file:
# %WINDIR%\System32\drivers\etc\hosts. Modifications to this file will b
e overwritten.
127.0.0.1       localhost
127.0.1.1       LAPTOP-URS0LRPO.localdomain      LAPTOP-URS0LRPO
10.20.1.189     hpcm.tamucc.edu
172.22.10.254   hobiserver.tamucc.edu
0.0.0.1 mssplus.mcafee.com

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
/etc/hosts (END)
```

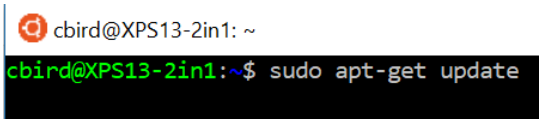
13. Change the permissions on the hosts file by entering the following command

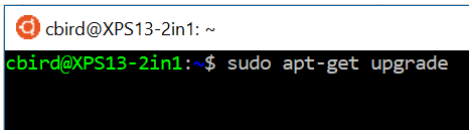
    `sudo chmod 555 hosts`

14. Load the new **`.bashrc`** settings

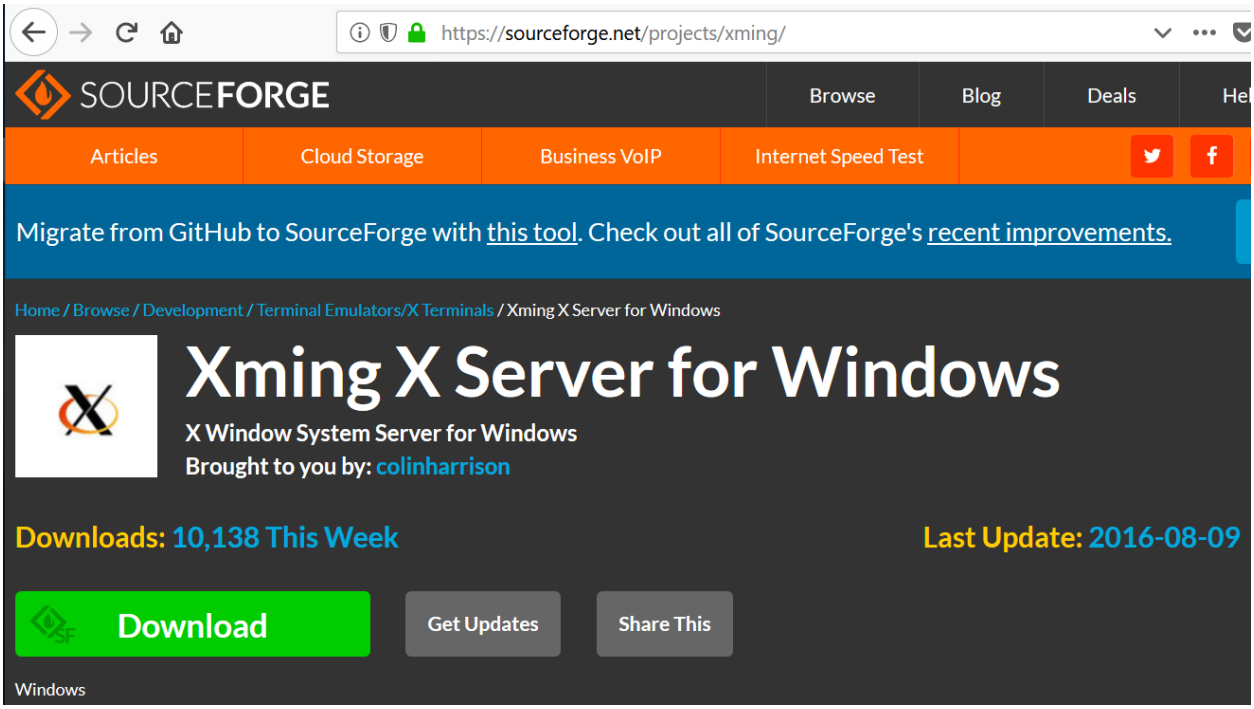


15. Update and upgrade UBUNTU. If prompted, enter a “y” for yes. This updates all of the software to the most recent version





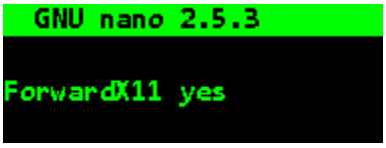
16. You need to install xming x server on windows and run it to run gui apps from UBUNTU



17. If you want to run gui programs through ssh within UBUNTU on Windows, then you need to create a ~/.ssh/config file by copying and pasting the following commands into the terminal:

```
mkdir .ssh
nano .ssh/config
```

18. Type the following into the config file, then save and close the file (ctrl x)



19. Lastly, to make nano behave properly on a remote server, open ~/.profile with nano and copy and paste the following function **to the end of the file**:

```
# This ssh alias function resolves an issue with running nano within an ssh session
function ssh(){
/usr/bin/ssh -t ${@:1} "stty sane; export TERM=linux; bash"
}
```

```
# This ssh alias function resolves a issue running nano within an ssh session
function ssh(){
  /usr/bin/ssh -t ${@:1} "stty sane; export TERM=linux; bash"
}
```

20. Save and close (ctrl x); you are done!!!