WSL Dev Environment

Wednesday, January 2, 2019 6:26 PM

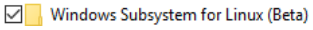
1. Install WSL on Windows
   a. From Powershell with Admin rights, execute:

Command	`Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux`

   Restart computer when prompted.

   b. Enable Developer Mode

   c. Turn Windows features for Windows Subsystem on
      i. Control, Programs, Turn Windows features on or off

   d. Get Ubuntu from Microsoft store
      i. Proceed with installation, when asked, specify Unix account name and password:

UID:	`nick`
PWD:	`zaika code`

2. Update your Ubuntu System

Commands	`sudo apt-get update` `sudo apt-get dist-upgrade`

3. Accessing Files between systems
   a. From Windows to Bash - DO NOT USE THIS DIR to MAKE EDITS.

Dir	`C:\Users\nick\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\LocalState\rootfs`

   b. From Bash To Windows

Bash	Windows
`/mnt/c/Users/nick`	`%userprofile%`

   c. Bash to Windows Directories

Bash	Windows
`~` `/home/nick`	`C:\Users\nick\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\LocalState\rootfs\home\nick`
`/etc`	`C:\Users\nick\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\LocalState\rootfs\etc`
`/mnt/c/Users/nick/Downloads`	`C:\Users\nick\Downloads`

4. Share files between Linux and Windows
   i. Create a new work directory from Windows, i.e.:

Dir	`%userprofile%\WslFiles`

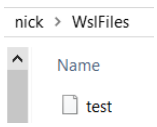
   ii. Create a soft link in the home dir

Command	`ln -s /mnt/c/Users/nick/WslFiles work`

   iii. Test
      i. In Bash, from home dir, change to work dir and create a file

Commands	`cd work` `touch test`

      ii. Observe file `test` in Windows

      iii. Delete file `test` from Windows and note that it is gone from Linux

5. Bash into you home directory and execute `.profile`

Command	`bash.exe ~ --login`

6. Make changes to `.profile`, add:

`.profile`

```
export ENV_OS=$(uname)
export HOSTNAME=`hostname -s`

PS1=$'\[\e[31m\]'`whoami`$'\[\e[0m\]@'$'\[\e[0;36m\]'$HOSTNAME$'\[\e[0m\]:'$'\[\e[32m\]'`pwd`$'\[\e[0m\]> '
```

```
mycd()
{
  unalias cd
  cd $1
  PS1=$'\[\e[31m\]'`whoami`$'\[\e[0m\]@'$'\[\e[0;36m\]'$HOSTNAME$'\[\e[0m\]:'$'\[\e[32m\]'`pwd`$'\[\e[0m\]> '
  alias cd=mycd
}

stty erase ^?

set -o vi

alias cd=mycd
```

a. Append to `.profile`. If the text indents, type:

Command	`:set paste`

7. Set up `vi` as editor for
   a. `git`

Command	`git config --global core.editor "vim"`

   b. for `v` command from `less`

Command:	`sudo update-alternatives --config editor`

Expected Response:	`  Selection    Path                Priority   Status`
	`------------------------------------------------------------`
	`  0            /bin/nano            40        auto mode`
	`  1            /bin/ed             -100       manual mode`
	`  2            /bin/nano            40        manual mode`
	`* 3            /usr/bin/vim.basic   30        manual mode`
	`  4            /usr/bin/vim.tiny    10        manual mode`

Install Ruby

Monday, January 28, 2019 8:40 AM

1. Install prerequisites for Ruby and Rails

Command	`sudo apt-get install -y gcc g++ libxml2 make ruby ruby-dev ruby-bundler`

2. Install RVM and Ruby
   a. Use the `gpg` command to contact a public key server and request the RVM project's key which is used to sign each RVM release.

Command:	`sudo gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB`

Response:	`gpg: requesting key D39DC0E3 from hkp server keys.gnupg.net`
	`gpg: requesting key 39499BDB from hkp server keys.gnupg.net`
	`gpg: /home/nickguner/.gnupg/trustdb.gpg: trustdb created`
	`gpg: key D39DC0E3: public key "Michal Papis (RVM signing) <mpapis@gmail.com>" imported`
	`gpg: key 39499BDB: public key "Piotr Kuczynski <piotr.kuczynski@gmail.com>" imported`
	`gpg: no ultimately trusted keys found`
	`gpg: Total number processed: 2`
	`gpg: imported: 2 (RSA: 2)`

   b. Install `gnupg2` package

Command:	`sudo apt-get install gnupg2`
PWD:	`zaika code`

   c. Use the `curl` command to download the RVM installation script from the project's website. Make sure to be disconnected from BNY Network to run this command.

Command:	`\curl -sSL https://get.rvm.io -o rvm.sh`
Response:	`curl: (77) Problem with the SSL CA cert (path? access rights?)`
Command	`echo insecure >> ~/.curlrc`
Command:	`\curl -sSL get.rvm.io -o rvm.sh`

      i. Update public keys - Note: this step may not be necessary.

Command	`sudo gpg2 --keyserver hkp://pool.sks-keyservers.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB`

Response:	`gpg: key 39499BDB: "Piotr Kuczynski <piotr.kuczynski@gmail.com>" not changed`
	`gpg: key D39DC0E3: "Michal Papis (RVM signing) <mpapis@gmail.com>" not changed`
	`gpg: Total number processed: 2`
	`gpg: unchanged: 2`

      ii. Change permissions on `rvm.sh` and execute

Command:	`chmod 755 rvm.sh` `./rmv.sh`
Response:	`Adding rvm PATH line to /home/nickguner/.profile /home/nickguner/.mkshrc /home/nickguner/.bashrc /home/nickguner/.zshrc.` `Adding rvm loading line to /home/nickguner/.profile /home/nickguner/.bash_profile /home/nickguner/.zlogin.` `Installation of RVM in /home/nickguner/.rvm/ is almost complete:` `* To start using RVM you need to run `source /home/nickguner/.rvm/scripts/rvm` `in all your open shell windows, in rare cases you need to reopen all shell windows.`
Command	`source /home/nickguner/.rvm/scripts/rvm`

   d. Use RVM to install Ruby 2.4.5.

Command	`rvm install ruby-2.4.5`

Response:	`Searching for binary rubies, this might take some time.`
	`Found remote file https://rubies.travis-ci.org/ubuntu/16.04/x86_64/ruby-2.4.5.tar.bz2`
	`Checking requirements for ubuntu.`
	`Requirements installation successful.`
	`ruby-2.4.5 - #configure`
	`ruby-2.4.5 - #download`
	`% Total % Received % Xferd Average Speed Time Time Time Current`
	`Dload Upload Total Spent Left Speed`
	`0 0 0 0 0 0 0 0 --:--:-- 0:00:01 --:--:-- 0`
	`100 16.8M 100 16.8M 0 0 2200k 0 0:00:07 0:00:07 --:--:-- 4348k`
	`No checksum for downloaded archive, recording checksum in user configuration.`
	`ruby-2.4.5 - #validate archive`
	`ruby-2.4.5 - #extract`
	`ruby-2.4.5 - #validate binary`
	`ruby-2.4.5 - #setup`
	`ruby-2.4.5 - #gemset created /home/nickguner/.rvm/gems/ruby-2.4.5@global`
	`ruby-2.4.5 - #importing gemset /home/nickguner/.rvm/gemsets/global.gems.....`
	`ruby-2.4.5 - #generating global wrappers.....`
	`ruby-2.4.5 - #gemset created /home/nickguner/.rvm/gems/ruby-2.4.5`
	`ruby-2.4.5 - #importing gemsetfile /home/nickguner/.rvm/gemsets/default.gems evaluated to empty gem list`
	`ruby-2.4.5 - #generating default wrappers.....`

Command	`rvm use ruby-2.4.5`
Response	`Using /home/nickguner/.rvm/gems/ruby-2.4.5`

      i. Verify

Command:	`ruby --version`
Response:	`ruby 2.4.5p335 (2018-10-18 revision 65137) [x86_64-linux]`

Login as Root user to WSL

Tuesday, March 12, 2019 10:53 AM

1. Execute command to change/set `root` password

Command	`sudo passwd`
Password	`zaika code`

WSL Terminal

Thursday, January 3, 2019 3:44 PM

1. Make sure to have 7Zip installed.
2. Instructions

URL	https://goreliu.github.io/wsl-terminal/

3. From `cmd` console execute:

Command	`cd C:\Users\nick\Linux` `bash -c "wget https://github.com/goreliu/wsl-terminal/releases/download/v0.8.11/wsl-terminal-0.8.11.7z --no-check-certificate && 7z x wsl-terminal-0.8.11.7z"`

   a. This command will create a `wsl-terminal-0.8.11.7z` file under current directory
   b. Extract to directory, i.e.

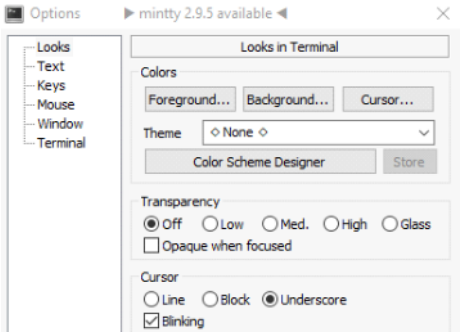
Dir	`C:\Users\nick\wsl-terminal`

   c. Create a Shortcut link on the desktop.
   d. Modify the Shortcut link to change to the home directory and execute login, set:

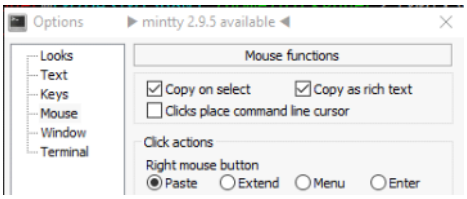
Target	`C:\Users\nick\wsl-terminal\open-wsl.exe -l`

4. Configure
   a. Window Size: 140 X 35

   b. Use underscore for cursor



   c. Use right-mouse button to paste



Using SSH with WSL

Thursday, January 3, 2019 9:57 AM

1. This page covers some of the commands that are available at this link

URL	https://virtualizationreview.com/articles/2018/01/10/hands-on-with-wsl-installation-and-new-features.aspx

2. Check Windows Version

Command	`winver`

3. Packages
   a. See all Ubintu packages

URL	https://packages.ubuntu.com/

   b. Package commands

Description	Command
List packages on your system	`dpkg --list`
files in vim	`dpkg --listfiles vim`
which package installed /usr/bin/vim.basic	`dpkg --search /usr/bin/vim.basic`
Update package repository	`apt update`
	`sudo apt install inetutils-traceroute`

4. Setting up SSH
   a. See Open SSH for Docker
   b. See instructions in:

URL	https://virtualizationreview.com/articles/2018/01/16/hands-on-with-wsl-executing-daily-tasks.aspx

   c. Reinstall OpenSSH

Commands	`sudo apt-get remove openssh-server` `sudo apt-get install openssh-server`

   d. Make changes to `/etc/ssh/sshd_config` file:

Change	`PermitRootLogin no`
	`PasswordAuthentication yes`
	`ListenAddress 0.0.0.0`
	`Port 2200`

Add	`AllowUsers <YourUserName>` `UsePrivilegeSeparation no`

   e. Restart SSH Service

Command	`sudo service ssh --full-restart`

   f. In Windows, allow TCP traffic through on port 2200. From cmd with elevated access execute:

Command	`netsh advfirewall firewall add rule name="Open SSH Port 2200" dir=in action=allow protocol=TCP localport=2200`

   g. SSH from `cmd` console. Note that the SSH service needs to be running.

Command	`ssh -p 2200 nickguner@127.0.0.1` `ssh -p 2200 <YouruserName>@127.0.0.1`

5. Automatically start ssh server on boot on Windows Subsystem for Linux
   a. Link

URL	https://gist.github.com/dentechy/de2be62b55cfd234681921d5a8b6be11

   b. Create new file - `sshd.bat`

```
C:\Users\ADCB8NJ\Documents\sshd.bat
C:\Windows\System32\bash.exe -c "sudo /usr/sbin/sshd -D"
```

c. Create new file - `sshd.vbs`

```
C:\Users\ADCB8NJ\Documents\sshd.vbs
Set WinScriptHost = CreateObject("WScript.Shell")
WinScriptHost.Run Chr(34) & "C:\Users\ADCB8NJ\Documents\sshd.bat" & Chr(34), 0
Set WinScriptHost = Nothing
```

d. From `Start | Run` command execute

Command	`shell:startup`

i. Copy `vbs` file to the Startup folder

e. Configure the `ssh` server to start without requiring password

i. From WSL terminal, execute

Command	`sudo visudo`

ii. Add following line at the end

```
%sudo ALL=NOPASSWD: /usr/sbin/sshd
```

6. SSH Between WSL and Linux Hosts

a. From WSL to Linux

Command	`ssh adcb8nj@r00fwkp0c.bnymellon.net`

b. From Linux Host to Wsl

i. Check Windows IP Address

Command	`ipconfig`
Response	`Connection-specific DNS Suffix . : bnymellon.com` `IPv4 Address. . . . . : 10.246.65.29`

ii. From WSL, Restart SSH Service

iii. From some Linux host

Command	`ssh -p 2200 nickguner@10.246.65.29`

Run Graphical Applications

Thursday, January 3, 2019 5:15 PM

1. Link

URL	https://virtualizationreview.com/articles/2018/01/30/hands-on-with-wsl-graphical-apps.aspx

2. Install `Xming`

3. Install Graphical applications

Command	`sudo apt-get install x11-apps`

4. Start applications
   a. Set display, add to `.profile`

Command	`export DISPLAY=:0`

   b. Run the app, for example

Command	`xeyes`

5. Other graphical apps

Commands	`sudo apt-get install blockout2`
	`sudo apt-get install vlc`
	`sudo apt-get install firefox`

Run Any Windows Apps from WSL

Friday, January 4, 2019 8:23 AM

1. Link

URL	https://www.nextofwindows.com/how-to-invoke-any-windows-executable-inside-wsl

2. Launch cmd shell from WSL

Command	`cmd.exe`

3. Launch Chrome (or any other app)
   a. Create an alias in .profile

```
alias launchchrome="\"/mnt/c/Program Files
(x86)/Google/Chrome/Application/chrome.exe\""
```

   b. Source `.profile` or restart

Command	`. ./.profile`

   c. Launch Chrome

Command	`launchchrome`

Start Linux Services on Windows Login

Tuesday, March 5, 2019 2:41 PM

1. Link

URL	https://dev.to/ironfroggy/wsl-tips-starting-linux-background-services-on-windows-login-3o98

2. Linux services started on start-up

   Start Postgres on Startup

Start Postgress on Startup

Sunday, November 22, 2020 5:37 PM

1. Create a new file under `~/.local/bin`

Command	`echo "service postgresql start" >> ~/.local/bin/start_postgresql.sh`

1. And change permissions:

Command	`chmod +x ~/.local/bin/start_postgresql.sh`

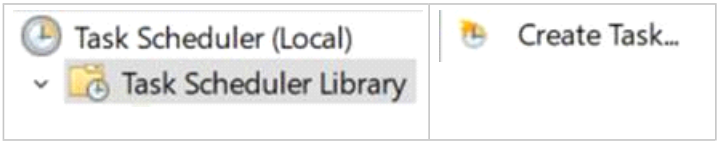
2. Make sure this script does not require `sudo` to run.
   a. Execute

Command	`sudo visudo`

2. Add:

```
nick ALL=(root) NOPASSWD: /home/nick/.local/bin/start_postgresql.sh
```

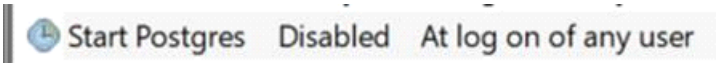
3. Get Windows to run this script in the WSL layer when you login.
   a. Run "Task Scheduler" (`taskschd.msc`).
   b. Select "Task Scheduler Library" on left and "Create Task" on right



3. Values

Tab	Field	Value
General	Name	`Start Postgres`
Triggers	Begin the task	`At Log On`
Actions	Action	`Start a program`
	Program/Script	`C:\Windows\System32\bash.exe`
	Add arguments	`-c "sudo ~/.local/bin/start_postgresql.sh"`

4. To Disable, set state to disabled:



Troubleshoot

Thursday, January 17, 2019 11:28 AM

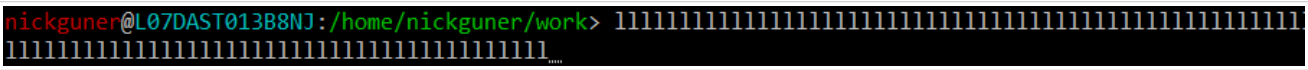
1. Terminal Prompt not wrapping correctly
   a. This problem has to do with color codes used in the PS1 prompt not being escaped properly

URL	https://stackoverflow.com/questions/342093/ps1-line-wrapping-with-colours-problem

   b. Currently, `.profile` uses color codes to specify PS1 prompt. Replace all instances if PS1 in `.profile`

From:	`PS1=$'\E[31m'`whoami`$'\E[0m@'$'\E[0;36m'$HOSTNAME$'\E[0m:'$'\E[32m'`pwd`$'\E[0m> '`
To:	`PS1=$'\[\e[31m\]'`whoami`$'\[\e[0m\]@'$'\[\e[0;36m\]'$HOSTNAME$'\[\e[0m\]:'$'\[\e[32m\]'`pwd`$'\[\e[0m \]> '`

   c. Screen capture:

Before	
After	

Misc, Config

Saturday, February 23, 2019 10:49 AM

1. Set vi tab to two spaces and expand tabs to spaces
   a. Create file `.vimrc` in the home directory

~/.vimrc
set tabstop=2 set expandtab

2. Check Ubuntu WSL Version

Command:	lsb_release -a
Response:	No LSB modules are available. Distributor ID: Ubuntu Description: Ubuntu 18.04.1 LTS Release: 18.04 Codename: bionic