WSL2 UBUNTU 18.04 Installation and Setup

AUTOSAR Adaptive Community

Exported on 10/26/2023

Table of Contents

1 WSL2 UBUNTU 18.04 INSTALLATION

Pre-requistics

> Before proceed further , Please make sure that VT-X option should be enabled in BIOS if its not enabled already . Refer below screen shot to check its state enabled or not

> It was observed that , after modifing the settings , bitlocker prompts the users to enter the key - Please contact IT team for the support.

- It will be look like below and refer more info on how-to-enable-virtualization[1]

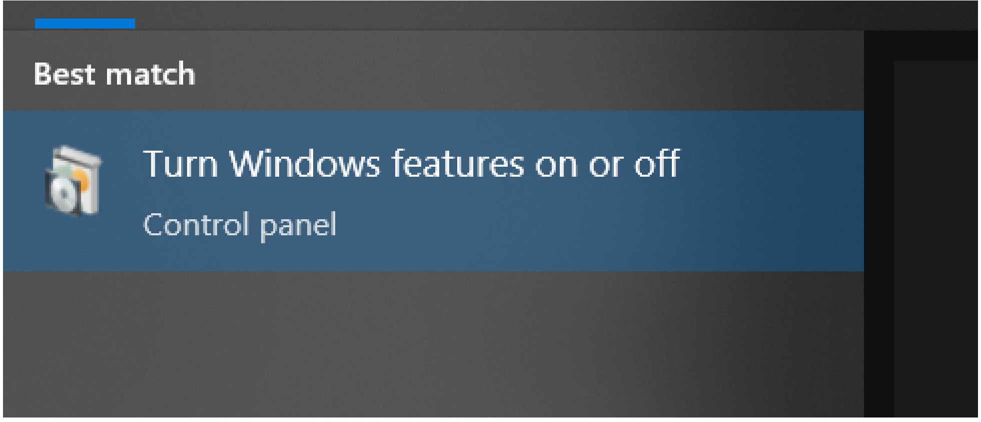
Install Ubuntu on WSL2

- Pre-requisites :
  - Before proceed further , Please make sure that user has an admin rights - if User doesnt have - Request via serivce Portal Link[2]
- Steps
  - Follow the link instructions https://docs.microsoft.com/en-us/windows/wsl/install-manual **Or** Below are the steps to follow to enable wsl in Windows 10 or Windows 11
  - First, press the Start key on the keyboard to **open the Start menu**.

[1] https://www.intel.in/content/www/in/en/support/articles/000005486/processors.html

[2] https://vitescoprod.service-now.com/sp?id=sc_category

- Type “**Turn Windows Features on or off**” and press Enter.

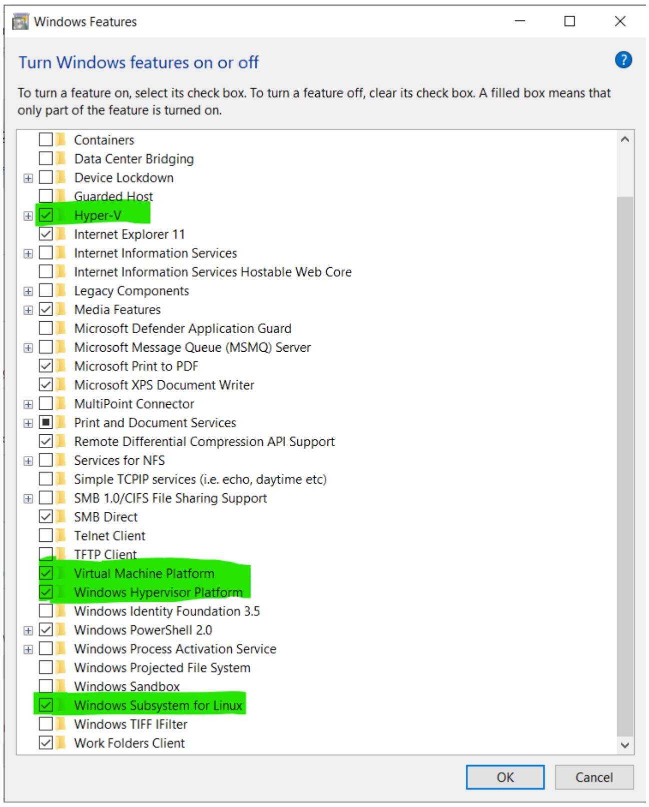


Best match

Turn Windows features on or off

Control panel

- Select the “**Highlighted**” checkbox in the Windows Features box.
- Click “**Ok**“.



Windows Features

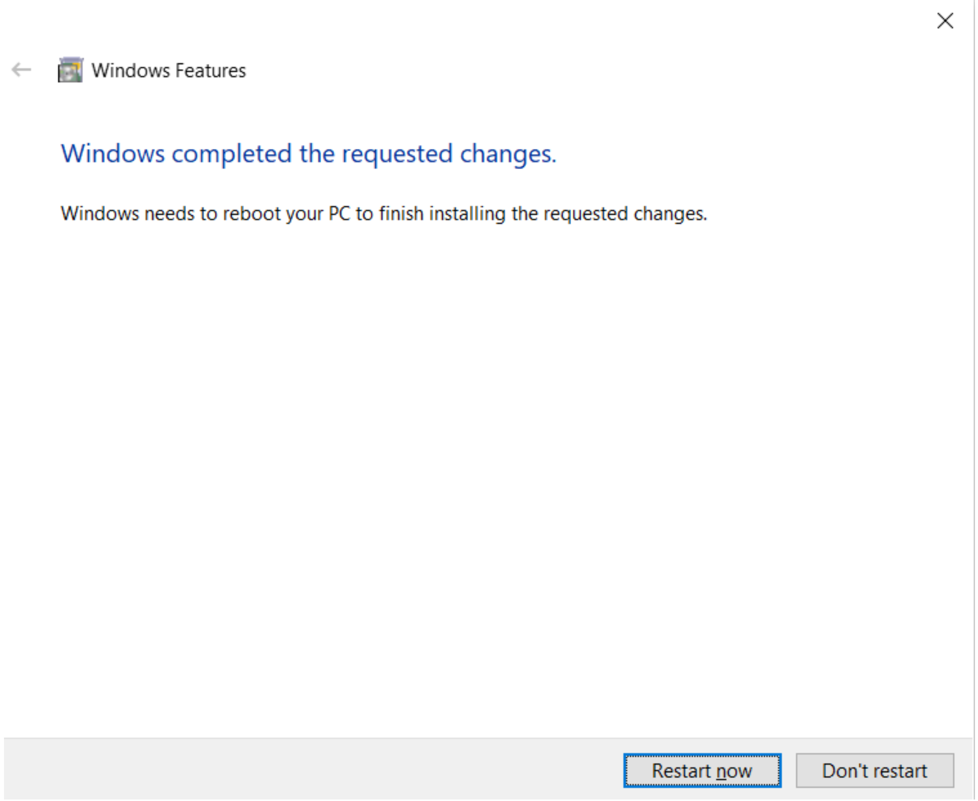
Turn Windows features on or off

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.

☐ Containers
☐ Data Center Bridging
☐ Device Lockdown
☐ Guarded Host
☑ Hyper-V
☑ Internet Explorer 11
☐ Internet Information Services
☐ Internet Information Services Hostable Web Core
☐ Legacy Components
☑ Media Features
☐ Microsoft Defender Application Guard
☐ Microsoft Message Queue (MSMQ) Server
☑ Microsoft Print to PDF
☑ Microsoft XPS Document Writer
☐ MultiPoint Connector
■ Print and Document Services
☑ Remote Differential Compression API Support
☐ Services for NFS
☐ Simple TCPIP services (i.e. echo, daytime etc)
☐ SMB 1.0/CIFS File Sharing Support
☑ SMB Direct
☐ Telnet Client
☐ TFTP Client
☑ Virtual Machine Platform
☑ Windows Hypervisor Platform
☐ Windows Identity Foundation 3.5
☑ Windows PowerShell 2.0
☐ Windows Process Activation Service
☐ Windows Projected File System
☐ Windows Sandbox
☑ Windows Subsystem for Linux
☐ Windows TIFF IFilter
☑ Work Folders Client

- As soon as you click Ok, Windows will download and install any required files and **enables the feature**.

- Finally, click the “**Restart now**” button to apply the changes.



Windows Features

Windows completed the requested changes.

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

- Download and install(execute) the Linux Kernel Package: WSL2 Linux kernel update package for x64 machines[3]
- Set WSL 2 as your default version (in normal command shell or PowerShell):

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

- In this example, ubuntu-18.04 installed . we can also install manually in cmd terminal

```
wsl --install -d Ubuntu-18.04
```

- For the first time installation, user will be prompted to enter "username" and password"
- username should be same as windows - eg. "uivxxxxx"
- Password can be set as per users wish
- Now we are done with the installation
- Please check "sudo su" and enter passoword whether you have sudo permissions. if yes, you will be navigate to root login.

- Try Windows Terminal Preview if you want - Has lots of nice features for terminal access - Download link[4]

Back to top(see page 3)

[3] https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi

[4] https://github.com/microsoft/terminal/releases

2 WSL2 UBUNTU-18.04 MIGRATION TO D DRIVE FOR YOCTO

Please make sure that your D: drive having atleast 200Gb free space before proceed further below steps - its highly recommended that if we have minimum 500Gb in D: drive

- Move the Ubuntu to a non-system disk Do not install ubuntu to disk C, and then download yocto on another disk,
- There will be a lot of file system access errors, because WSL2 file system permissions do not fully control the Windows disk. Therefore, the installed WSL needs to be moved to a non-system disk.
- The export distribution is a tar file to a d disk

```
wsl --export Ubuntu-18.04 d:\wsl-ubuntu18.04.tar
```

- Sign out of the current distribution

```
wsl --unregister Ubuntu-18.04
```

- Re-import and install WSL on the D disk

```
wsl --import Ubuntu-18.04 d:\wsl-ubuntu18.04 d:\wsl-ubuntu18.04.tar --version 2
```

- Set the default landing user to be the user name at the time of installation

```
ubuntu1804 config --default-user <USERNAME>
```

- Delete the wsl-ubuntu18.04 .tar

```
del d:\wsl-ubuntu18.04.tar
```

- You can enter \\wsl$\Ubuntu-18.04 by entering the windows resource management bar, or copy it directly to the Windows file system under CTRL-C.

Back to top(see page 3)

3 WSL2 SET USERNAME PASSWORD FOR IMPORTED MACHINE

- When you import a WSL2 distro manually, running that distro on the new machine will end up logging you in as root.
- There's a lot of ways to fix this that involve the registry or passing in arguments to wsl, but just want it to work when I run " `wsl` " or " `wsl -d distroname` ."
- step 1.
    - Set default username :
        - Run your distro, and then edit /etc/wsl.conf and add a [user] section like this: if that file not available , create and update it
        - e.g. user is "nxp"
        - after setting up the username - shutdown the wsl machine - "wsl --shutdown -d <distroname>"

```
[user]
default=nxp
```

- step 2.
- Open the terminal of the imported machine
    - Set password for the username :

```
sudo passwd nxp
New password: <Enter your password>
Retype new password:<Enter your password>
passwd: password updated successfully
```

4 WSL2 VPNKIT ENABLE

> ⓘ The below instrctions tested with vpn connected setup e.g. wfh , yet to be tested in - wfo

After successful installation of ubuntu-18.04, now the time to activate the internet connection available under our network to install the packages.

- Pre-requisites
  - Install any Linux distribution like ubuntu-18.04 – in this example, ubuntu-18.04 installed
- Deactivate the Hyper-V option if already activated
- Download and store the file wsl-vpnkit.tar.gz from the link[5]and install it as wsl2 version -
  - Open CMD in where wsl-vpnkit folder and import the distribution (cmd)

```
wsl --import wsl-vpnkit wsl-vpnkit wsl-vpnkit.tar.gz
```

- Run the wsl-vpnkit before use any linux distribution

```
wsl -d wsl-vpnkit
```

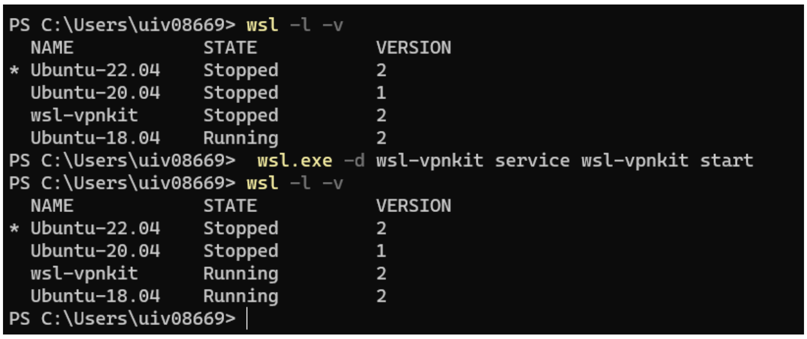
- Open CMD and start the wsl-vpnkit

```
wsl.exe -d wsl-vpnkit service wsl-vpnkit start
```

For autostart the wsl-vpnkit while booting the wsl. Put the above CMD in the ~/.bashrc

> ⓘ **How CMD commands in WSL?**
>
> Commands from CMD can be used across WSL and PowerShell. More details click here[6]

- Status will be like below - wsl-vpnkit should be in running state

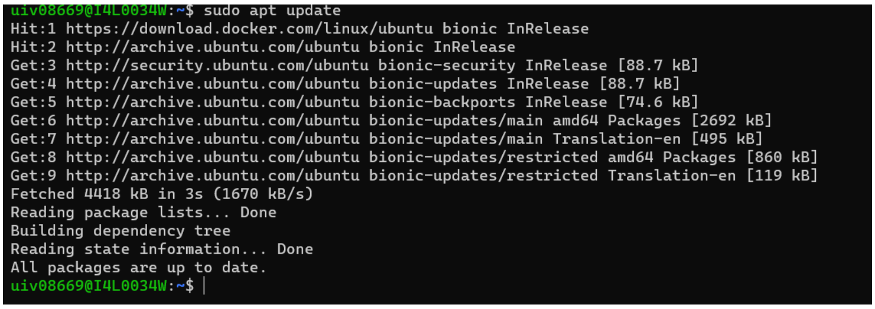


```
PS C:\Users\uiv08669> wsl -l -v
  NAME            STATE           VERSION
* Ubuntu-22.04    Stopped         2
  Ubuntu-20.04    Stopped         1
  wsl-vpnkit      Stopped         2
  Ubuntu-18.04    Running         2
PS C:\Users\uiv08669>  wsl.exe -d wsl-vpnkit service wsl-vpnkit start
PS C:\Users\uiv08669> wsl -l -v
  NAME            STATE           VERSION
* Ubuntu-22.04    Stopped         2
  Ubuntu-20.04    Stopped         1
  wsl-vpnkit      Running         2
  Ubuntu-18.04    Running         2
PS C:\Users\uiv08669> |
```

- Start WSL. Now try to update the distribution – sudo apt update

[5] https://github.com/sakai135/wsl-vpnkit/releases/tag/v0.3.2

[6] https://learn.microsoft.com/en-us/windows/wsl/filesystems#interoperability-between-windows-and-linux-commands



```
uiv08669@I4L0034W:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2692 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [495 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [860 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [119 kB]
Fetched 4418 kB in 3s (1670 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
uiv08669@I4L0034W:~$
```

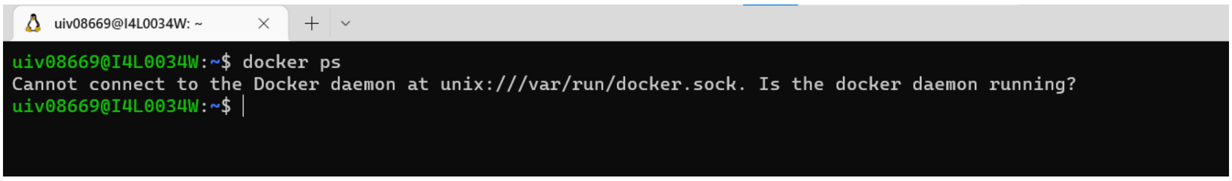
Back to top(see page 3)

5 WSL2 USE DOCKER WITH UBUNTU

- Follow the below link instructions to install the docker on ubuntu

https://docs.docker.com/engine/install/ubuntu/

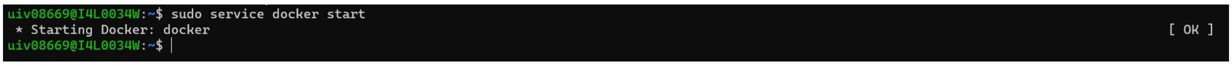
- After installation done – If try to check the status -Cmd - sudo docker ps - you will get the error like this



```
uiv08669@I4L0034W:~$ docker ps
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
uiv08669@I4L0034W:~$ |
```

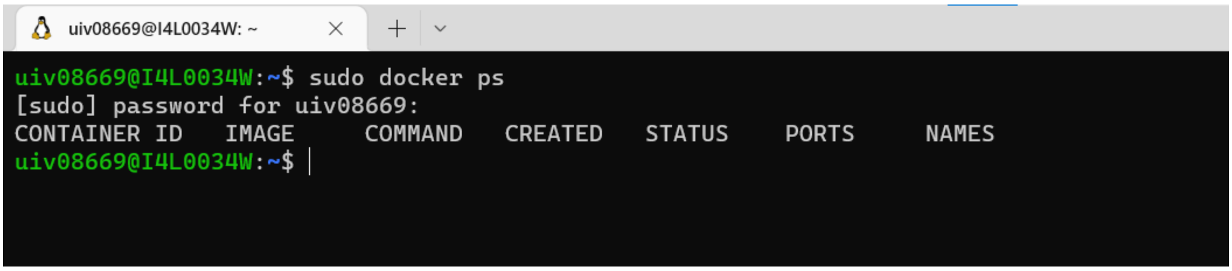
- Fix Docker in Debian WSL 2 (shell) - In order to activate the docker daemon inside the ubuntu distribution , Please do the following in terminal .

```
sudo touch /etc/fstab
sudo update-alternatives --set iptables /usr/sbin/iptables-legacy
sudo update-alternatives --set ip6tables /usr/sbin/ip6tables-legacy
sudo service docker start
```



```
uiv08669@I4L0034W:~$ sudo service docker start
 * Starting Docker: docker                                                                                  [ OK ]
uiv08669@I4L0034W:~$ |
```

- After enabling the above settings, we could able to query the docker status and you will get like below.



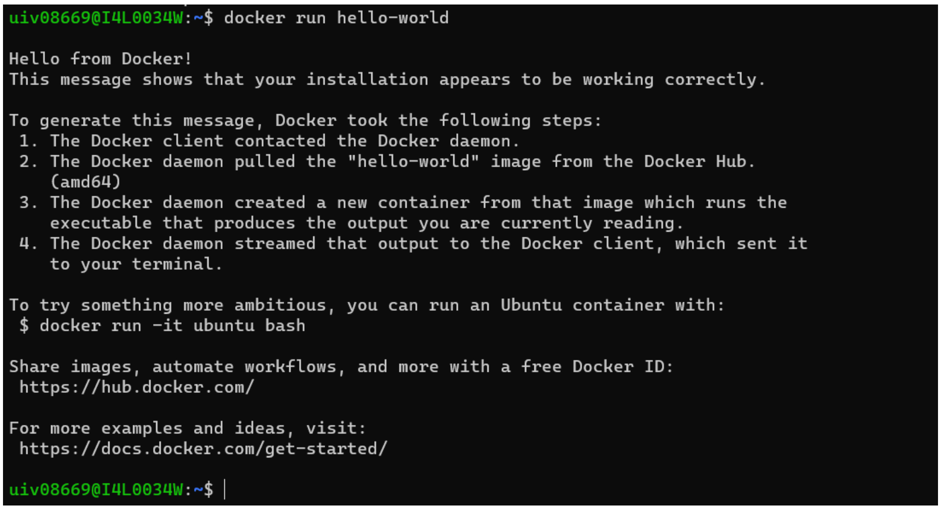
```
uiv08669@I4L0034W:~$ sudo docker ps
[sudo] password for uiv08669:
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
uiv08669@I4L0034W:~$ |
```

- **Run Docker without sudo** (shell)
- Open the ubuntu-18.04 terminal

```
sudo groupadd docker
sudo usermod -aG docker $USER
newgrp docker
```

- **Hello world with docker**
- Try the docker hello-world – the below command will pull the hello-world image and print the hello world inside docker container. Refer the screenshot

```
docker run hello-world
```



```
uiv08669@I4L0034W:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

uiv08669@I4L0034W:~$
```

6 WSL2 IP ADDRESSES

At every restart of Windows the WSL2 ethernet adapters (i. e. 'eth0' in Ubuntu as well as 'vEthernet (WSL)' in Windows) get new / different IP addresses.

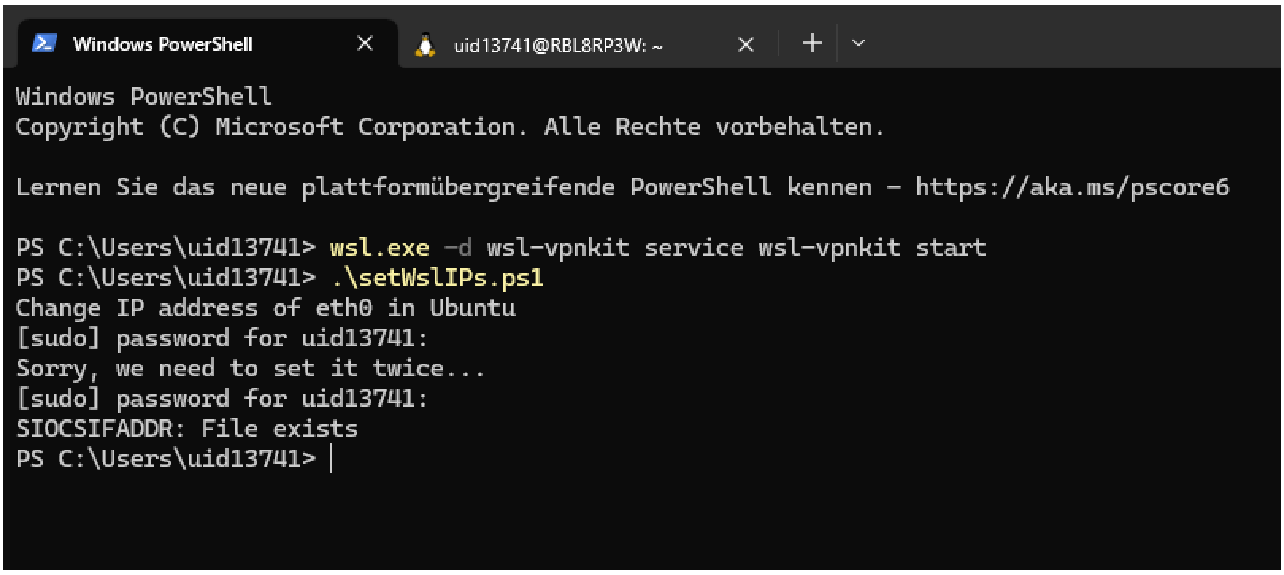
Currently there is no way to configure static addresses, therefore a script has been created to set the IP addresses to the ones that are suggested to be used:

- Windows: 192.168.56.1
- Ubuntu: 192.168.56.25

The Ubuntu IP is the same as for the Vehicle Server, because we use this machine not only for building but also running/testing Adaptive Applications configured for the Vehicle Server.

setWslIPs.ps1

Storing this script in C:\Users\<uidxxxxx>\ makes it most comfortable to run it when opening Terminal-Preview:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Lernen Sie das neue plattformübergreifende PowerShell kennen – https://aka.ms/pscore6

PS C:\Users\uid13741> wsl.exe -d wsl-vpnkit service wsl-vpnkit start
PS C:\Users\uid13741> .\setWslIPs.ps1
Change IP address of eth0 in Ubuntu
[sudo] password for uid13741:
Sorry, we need to set it twice...
[sudo] password for uid13741:
SIOCSIFADDR: File exists
PS C:\Users\uid13741> |
```

Remark: make sure you have the required elevated rights to be able to change IP addresses in Windows and Ubuntu!

7 WSL2 DESKTOP ENVIRONMENT (OPTIONAL)

- If user want to work with GUI based Linux environment.
- Here is the instruction to activate it

```
sudo apt update && sudo apt -y upgrade
sudo apt -y install xfce4
sudo apt-get install xrdp
sudo cp /etc/xrdp/xrdp.ini /etc/xrdp/xrdp.ini.bak
sudo sed -i 's/3389/3390/g' /etc/xrdp/xrdp.ini
sudo sed -i 's/max_bpp=32/#max_bpp=32\nmax_bpp=128/g' /etc/xrdp/xrdp.ini
sudo sed -i 's/xserverbpp=24/#xserverbpp=24\nxserverbpp=128/g' /etc/xrdp/
xrdp.ini
```

- after that edit the below file comment out bellow line(Italic) and add the line mentioned(bold) below sudo nano /etc/xrdp/startwm.sh

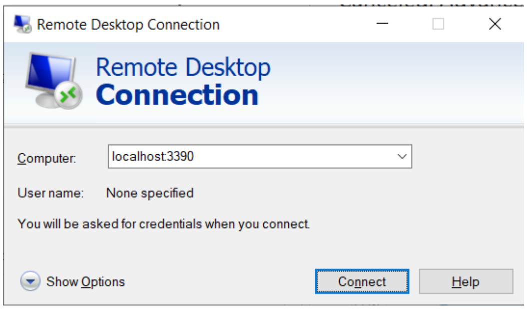
```
comment these lines to:
#test -x /etc/X11/Xsession && exec /etc/X11/Xsession
#exec /bin/sh /etc/X11/Xsession

add these lines:
# xfce
startxfce4
```

- To start the xrdp desktop environment

```
sudo /etc/init.d/xrdp start
```

- Now in Windows, use Remote Desktop Connection And then connect via RDP localhost:3390 to your desktop.

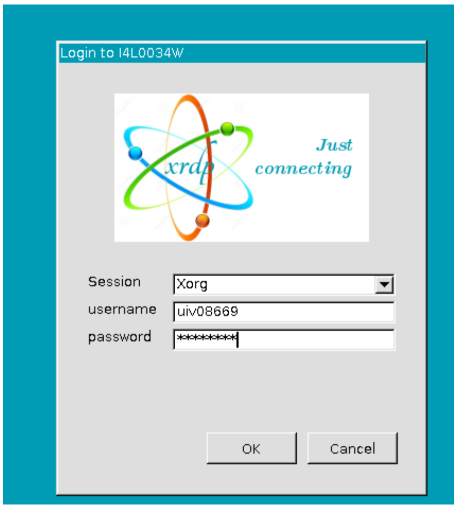

Remote Desktop Connection

Remote Desktop Connection

Computer: localhost:3390

User name: None specified

You will be asked for credentials when you connect.



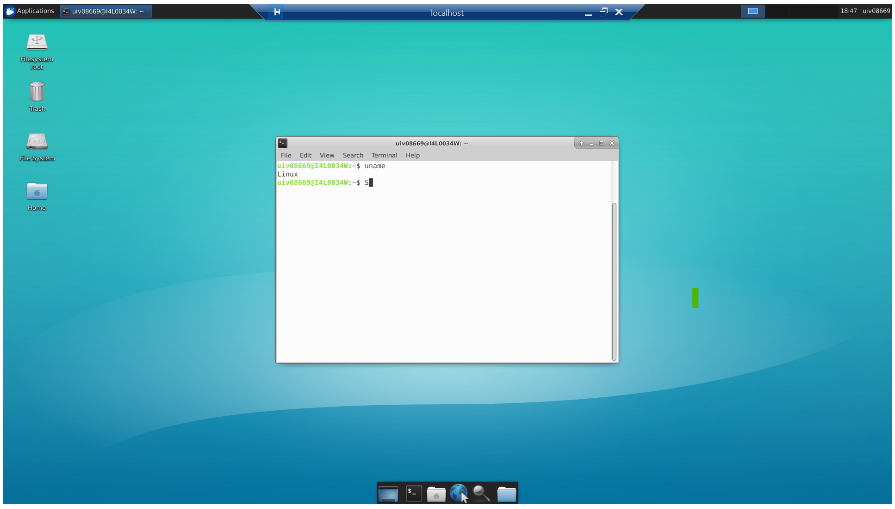
Login to I4L0034W

Session: Xorg

username: uiv08669

password: ********

- Login with your wsl2 username and password

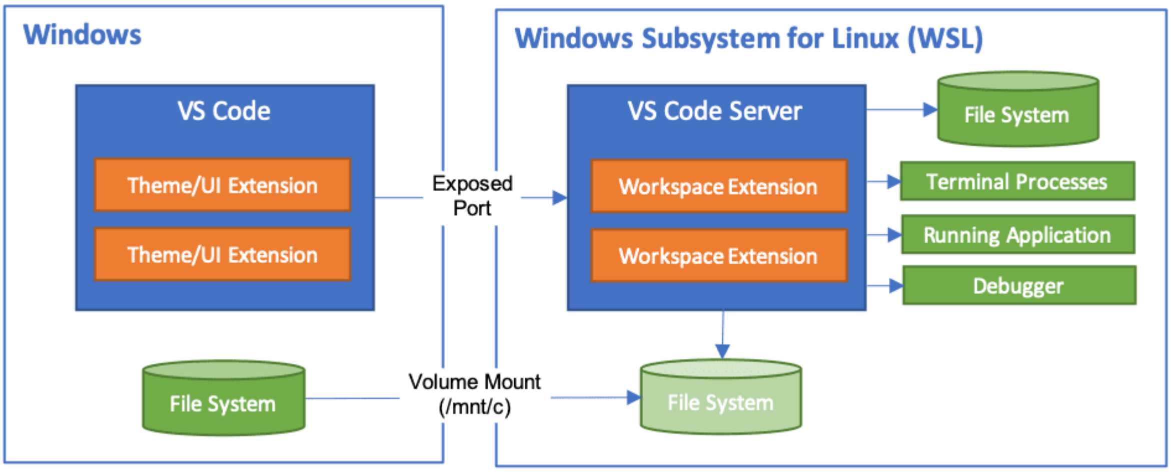


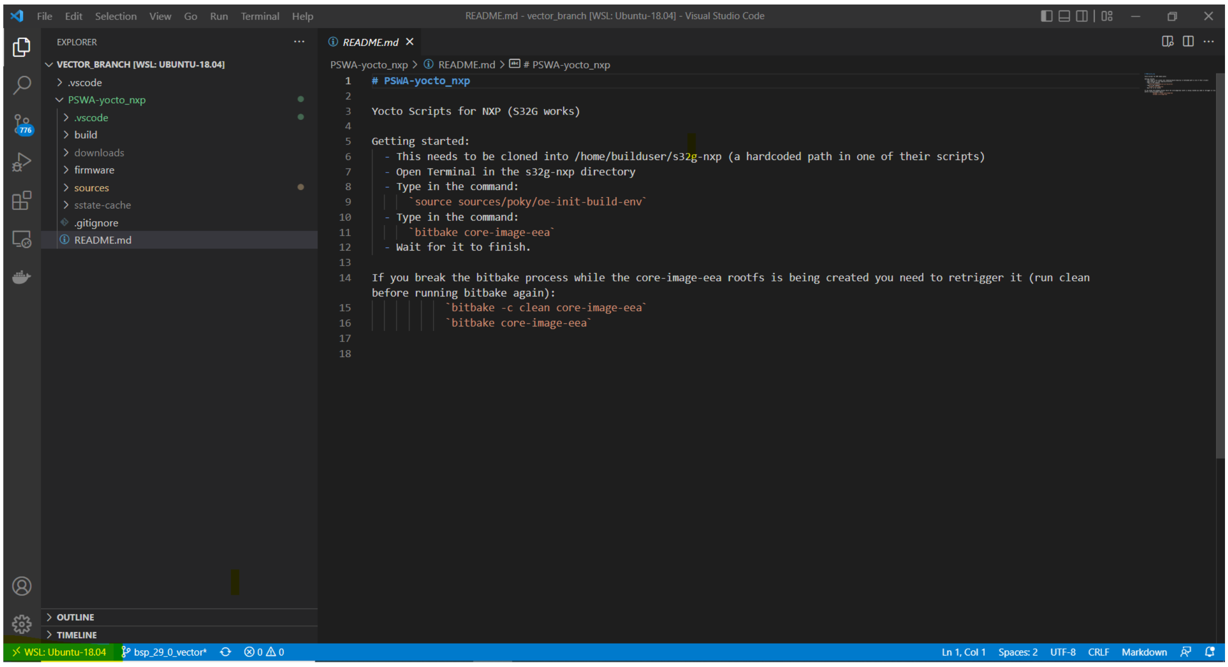
```
uiv08669@I4L0034W:~$ uname
Linux
uiv08669@I4L0034W:~$ S
```

Back to top(see page 3)

8 WSL2 WITH VISUAL STUDIO CODE

- Wsl2 with vscode integration - All of this is made possible by using the Remote - WSL extension[7]. Editing, debugging, and even extensions all work exactly the way you've come to expect in your favorite editor such as VS code.
- Add in the Remote - WSL extension in Visual Studio Code and you have the best of all worlds – Linux and Windows compatibility for your tools with excellent performance and a seamless development experience.
- Follow the instruction to install the ubuntu on wsl2
- https://code.visualstudio.com/blogs/2019/09/03/wsl2





```
# PSWA-yocto_nxp

Yocto Scripts for NXP (S32G works)

Getting started:
  - This needs to be cloned into /home/builduser/s32g-nxp (a hardcoded path in one of their scripts)
  - Open Terminal in the s32g-nxp directory
  - Type in the command:
      `source sources/poky/oe-init-build-env`
  - Type in the command:
      `bitbake core-image-eea`
  - Wait for it to finish.

If you break the bitbake process while the core-image-eea rootfs is being created you need to retrigger it (run clean before running bitbake again):
            `bitbake -c clean core-image-eea`
            `bitbake core-image-eea`
```

Back to top(see page 3)

[7] https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-wsl

Back to top(see page 3)