Windows Subsystem for Linux - Debian 10.3

Osiris, Hera, Vulcan, Heracles, Poseidon, Iapetus

This guide has been tested under Windows 10 / Debian 10 and Windows 11 / Debian 13.

Installation

1. Enable “Developer mode” in *Settings –> System –> Advanced* (Win 11) or *Settings –> Update & Security –> For developers* (Win 10)
2. Press Windows key + R and run **optionalfeatures.exe**. Enable *Windows Subsystem for Linux*. Restart the computer.
3. Open Windows Power Shell as Administrator, then run the following line and reboot:

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

4. If you want to use WSL on Windows 11 Home you need to first install Hyper-V (see How to Install and Enable Hyper V in Windows 11 Home). Create the following Windows batch file and run it as administrator:

   ```
   pushd "%~dp0"
   dir /b %SystemRoot%\servicing\Packages\*Hyper-V*.mum >hyper-v.txt
   for /f %%i in ('findstr /i . hyper-v.txt 2^>nul') do dism /online /norestart /add-package:"%SystemRoot%\servicing\Packages\%%i"
   del hyper-v.txt
   Dism /online /enable-feature /featurename:Microsoft-Hyper-V -All /LimitAccess /ALL
   pause
   ```

5. Restart your computer
6. Open the Microsoft Store and search for “Linux”. Choose and get Debian, as per 2020-03-10: Debian 10 Buster, 2025-11-7: Debian 13.0 Trixie.
7. Open *Debian* from the start menu and complete installation.
8. Upgrade all packages:

   ```
   sudo apt update
   sudo apt upgrade
   ```

9. Install packages:

   ```
   sudo apt install vim
   sudo apt install wget
   sudo apt install zip
   sudo apt install rsync
   ```

10. Create /run/sshd folder:

```
sudo mkdir /run/sshd
```

11. Create a shortcut to the executable on your desktop:

```
C:\Users\user\AppData\Local\Microsoft\WindowsApps\debian.exe
```

12. The root file system can be found at a path similar to:

```
C:\Users\bco\AppData\Local\Packages\TheDebianProject.DebianGNULinux_76v4gfsz19hv4\LocalState\rootfs
```

13. Add ☒ and ☒ icons to the shortcuts

Files

- WSL2 image file is located at `C:\Users\<username>\AppData\Local\Packages\<distribution>\LocalState\ext4.vhdx` (replace 'ext4' with the filesystem of your distribution)
- You can copy file `ext4.vhdx` to another location and open it with 7zip.
- To read files from an ext4.vhdx file, which is commonly used by Windows Subsystem for Linux (WSL) 2, you can utilize the wsl –mount command. This process involves mounting the VHDX file within Windows and then accessing its contents from within your WSL environment. See following steps to read files from ext4.vhdx using WSL.
- Terminate WSL Instances: Ensure all running WSL instances are shut down to prevent conflicts with the VHDX file.

```
wsl.exe --shutdown
```

- Mount the VHDX in Windows (with Administrator privileges): Open PowerShell or Command Prompt as an administrator and use Mount-VHD to make the VHDX file accessible to Windows. Replace <pathToVHD> with the actual path to your ext4.vhdx file.

```
Mount-VHD -Path "<pathToVHD>"
```

- Alternatively, you can use diskpart to mount the VHDX:

```
diskpart
    select vdisk file="<pathToVHD>"
    attach vdisk
```

- Identify the Disk Path in Windows: After mounting, you need to find the disk path assigned by Windows. You can use Get-Disk in PowerShell or list disk in diskpart to identify the newly mounted disk.
- Mount the Disk in WSL: Open your WSL distribution (e.g., Ubuntu) and mount the disk path identified in the previous step. Replace <diskPath> with the actual disk path (e.g., /dev/sdb) and <mountPoint> with your desired mount point within WSL (e.g., /mnt/ext4).

```
sudo wsl --mount <diskPath> --bare
sudo mkdir -p <mountPoint>
sudo mount <diskPath> <mountPoint>
```

- Access Files within WSL: Once mounted, you can navigate to the <mountPoint> within your WSL environment and access the files stored on the ext4.vhdx.

  ```
  cd <mountPoint>
  ls -l
  ```

- Unmount and Detach (after use): When you are finished, unmount the disk within WSL and detach the VHDX from Windows.

  ```
  sudo umount <mountPoint>
  wsl --unmount <diskPath>
  ```

- If you used Mount-VHD:

  ```
  Dismount-VHD -Path "<pathToVHD>"
  ```

- If you used diskpart:

  ```
  diskpart
  select vdisk file="<pathToVHD>"
  detach vdisk
  ```

- Alternative Method (for viewing only): For simply viewing the contents of an ext4.vhdx file without mounting, you can use tools like 7-Zip. 7-Zip can open and browse the contents of various disk image formats, including ext4.vhdx, allowing you to extract files without needing to mount the entire disk.

SSH Server

- Install packages:

  ```
  sudo apt install openssh-server
  ```

- Follow the SSH Client and Server guide. SSH into your host and open port 22 in the Windows Firewall.

Start SSH Server at Windows boot time

- Open a bash command shell and allow <user> to run ssh as root, add the following line after %sudo:

  ```
  sudo visudo
  %sudo ALL=(ALL:ALL) ALL
  <user> ALL=(root:root) NOPASSWD: /etc/init.d/ssh
  ```

- Open Windows Task Scheduler and click *Create Basic Task…*
  1. General
     - Name : Start Bash SSH Server
     - Description: Start the WSL SSH Server via a bash command
     - Run only when user is logged on

    - Run with highest privileges
2. Trigger
    - Begin the task: At log on [OR] On an event
    - (if you mount VeraCrypt volumes during boot it's better to use *On an event* and begin the task when the volumes are mounted)
    - Specific user: <user>
    - Enabled
3. Actions
    - Action: Start a program
    - Program/script: C:\Windows\System32\bash.exe
    - Add arguments (optional): -c “/usr/bin/sudo /etc/init.d/ssh start”
4. Conditions
    - Start the task only if the computer is on AC power
5. Settings
    - *Uncheck* Stop the task if it runs longer than:
6. Alternative Action when starting a non elevated script “startup”
    - Action: Start a program
    - Program/script: C:\Windows\System32\bash.exe
    - Add arguments (optional): -c “/home/user/startup”
    - Script “startup”:

      ```
      #!/bin/bash
      sudo /etc/init.d/ssh start
      ```

LAMP

- LAMP on Debian 12 (Apache 2.4, PHP 8.3, MariaDB 10.11)
- Debian 9 - Apache 2.4 and PHP 7.4
- Debian 9 - MariaDB 10.3

Mounting encrypted drives

- WSL can mount encrypted drives, but they must be mounted at the time wsl or bash runs the first time after boot.
- If you do not execute any tasks when the host boots up, then just make sure your encrypted drives are mounted when you run wsl or bash.
- If you **do** execute WSL tasks when the host boots up, replace the *Task Scheduler* entry for the trigger and make it rund after the *Mount* script has executed:

  ```
  <QueryList>
    <Query Id="0" Path="Microsoft-Windows-TaskScheduler/Operational">
      <Select Path="Microsoft-Windows-TaskScheduler/Operational">*[EventData[@Name='TaskSuccessEvent'][Data[@Name='TaskName']='\Mount']]</Select>
    </Query>
  </QueryList>
  ```

- Remember to “Enable All Task History” under Actions tab. **You might need to do this again**

after a Windows 10 major version upgrade.

Mounting external drives

```
mkdir /mnt/f
mount -t drvfs f: /mnt/f
```

Links

- Windows Subsystem for Linux Installation Guide for Windows 10
- How Does the Windows 10 Subsystem for Linux Work and What Is It For?
- Background Task Support in WSL
- SSH on Windows Subsystem for Linux (WSL)
- Backup with rsync on Windows (WSL)
- Docker Running Seamlessly in Windows Subsystem Linux

From:
https://wiki.condrau.com/ - **Bernard's Wiki**

Permanent link:
https://wiki.condrau.com/win10:wsl

Last update: **2025/11/23 12:01**