How to SSH into WSL2 on Windows 10 from an external machine

July 31, 2020 [Comment on this post [208]](#) Posted in [Linux](#) | [Win10](#)
Sponsored By

Cool blog post eh? Good title, right?

DO NOT DO THE INSTRUCTIONS IN THIS POST

until you have read the FOLLOW UP [THE EASY WAY how to SSH into Bash and WSL2 on Windows 10 from an external machine](#) and made the right decision for YOU!

[OpenSSH has shipped in Windows for 5 years now](#), so that's cool. You can do lots of things!

- [How to use Windows 10's built-in OpenSSH to automatically SSH into a remote Linux machine](#)
- [How to SSH into a Windows 10 Machine from Linux OR Windows OR anywhere](#)
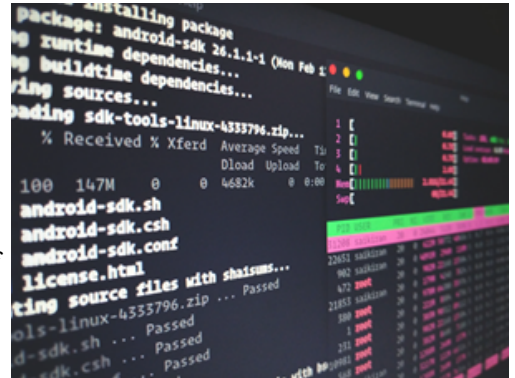
But often folks want to SSH *not* into their Windows 10 machine, but rather, into WSL2 running within/behind their Windows 10 machine. While WSL2 can forward ports from the inside out (for example, localhost:8000 within a WSL2 instance being made available from the local Windows 10 machine) if you want to build a path to a WSL2 port from completely outside a machine, you'll need to be a lot more explicit.

Install OpenSSH-Server in WSL

First, install OpenSSH server inside your Linux Distro:

```
scott@IRONHEART:~$ sudo apt install openssh-server
[sudo] password for scott:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.6p1-4ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Decide on your SSH port number

Next, in WSL2, edit /etc/ssh/sshd_config and uncomment out the Port line.

> I edited it with `sudo nano /etc/ssh/sshd_config`, no shame!

SSH is usually 22, but I like to use something like 2222 so it's less obvious but still easy to remember AND is different from your Window's machine's 22. Note that I told it to listen on 0.0.0.0, so, any adapter. You can also set PasswordAuthentication to "no" if you want to use SSH keys rather than passwords for authentication. Set it to "yes" if you know what you're doing and don't know how to use ssh keys.

/etc/ssh/sshd_config

```
...STUFF ABOVE THIS...
Port 2222
#AddressFamily any
ListenAddress 0.0.0.0
#ListenAddress ::

...STUFF BELOW  THIS...
```

From within WSL2 get your IP address using "ifconfig." Mine is 172.23.129.80, yours will likely be 172.SOMETHINGELSE and **it will change when WSL2 starts up cold**.

You may want to ensure it's running, considering WSL2 has no systemd.

```
service ssh start
```

Forward Ports into WSL2

Now, from an **Administrator Windows prompt** - that can be cmd.exe or powershell.exe, it doesn't matter, use the net shell "netsh" to add a portproxy rule. Again, change connectaddress to YOUR WSL2 ipaddress, which is an internal address to your machine.

```
netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=2222 connectaddress=172.23.129.80 connectport=2222
```

Open the Firewall

Next, from the same Administrator Windows prompt, open an incoming Firewall Port. You can do it from the Advanced Firewall Settings, but even easier you can use netsh again!

```
netsh advfirewall firewall add rule name="Open Port 2222 for WSL2" dir=in action=allow protocol=TCP localport=2222
```

You can list all your portproxy rules like this if you're concerned:

```
netsh interface portproxy show v4tov4
```

You can remove them all if you want with

```
netsh int portproxy reset all
```

A scripted solution?

GitHub user and community member Daehahn is working on a PowerShell Script to automate this process. The comment thread starts here and the gist for the PowerShell script for wsl2-network.ps1 is here. It resets firewall and portproxies, finds your default distro's new IP, and sets you up again. Save this .ps1 somewhere, read it, and run "unblock-file wsl2-network.ps1" on it so you can set up your system quickly for Shushing into your WSL2 instance!

> Note the $Ports variable that likely opens up more than you want or need, remembering that WSL and VS Code will automatically forward ports to localhost when needed for development.

Hope this helps! It would be nice if WSL2 didn't change it's internal IP address every time it starts up so that this could be made even easier and more automated.

To conclude and sum up:

- This blog post - the one you are reading now, has Windows only forwarding ports, and uses WSL2's Linux OpenSSH and authenticates against Linux. Windows is only involved peripherally. The WSL2 IP address changes on reboot and you'll need to maintain your portproxy rules and firewall rules with the script listened at the end of that post.
- This other blog post - over here - uses Windows' OpenSSH and authenticates with Windows *and then runs WSL2.* WSL2 starts up, uses bash, and Windows handles the TCP traffic.

Understand what you want and use the right one for you.

Other cool links:

- Docker Desktop for WSL 2 integrates Windows 10 and Linux even closer
- Remote Debugging a .NET Core Linux app in WSL2 from Visual Studio on Windows
- Cool WSL (Windows Subsystem for Linux) tips and tricks you (or I) didn't know were possible
- Ruby on Rails on Windows is not just possible, it's fabulous using WSL2 and VS Code
- Easily move WSL distributions between Windows 10 machines with import and export!
- What's the difference between a console, a terminal, and a shell?

Hope this helps! Also, please do subscribe to my YouTube channel!

About Scott

Scott Hanselman is a former professor, former Chief Architect in finance, now speaker, consultant, father, diabetic, and Microsoft employee. He is a failed stand-up comic, a cornrower, and a book author.

About   Newsletter
Hosting By