TECH TIP — BEGINNER/INTERMEDIATE

How to Set Up SSH on Windows

(And Actually Connect to Something)

From Nolan Law Firm — nemolegal.com/tools

What SSH Is (One Paragraph)

SSH stands for Secure Shell. It lets you connect to another computer over a network and control it from a command line. When someone says "just SSH in and check the logs," they mean: open a terminal, type one command, and you're in. It's the backbone of server management — and until recently, Windows made it needlessly painful to set up.

The Two Things Windows SSH Involves

Client — the thing you type commands into to connect *to* another machine. Already installed on Windows 10/11. Open PowerShell and type `ssh`. If you get a usage message instead of an error, you have it.

Server — the thing that lets other machines connect *to you*. This is what you need to install. That's what this guide covers.

Step 1 — Install OpenSSH Server (The Right Way)

Go through Settings. Do not use PowerShell commands you found online — half of them are outdated.

Settings → System → Optional Features → Add a Feature → search for **OpenSSH Server** → Install

If you're on Windows 11: Settings → System → Optional Features → View Features → search OpenSSH Server → Next → Install

> **Warning:** Do NOT install OpenSSH Client — it's probably already there. You want OpenSSH Server specifically. They are listed separately.

Step 2 — Start the Service

After installing, open PowerShell as Administrator and run:

```
Start-Service sshd

Set-Service -Name sshd -StartupType Automatic
```

The second line makes SSH start automatically when the computer reboots. Skip it and you'll wonder why SSH stopped working next Monday.

Step 3 — Open the Firewall (The Step Everyone Misses)

Windows Firewall blocks SSH by default even after you install it. Run this in PowerShell as Administrator:

```
New-NetFirewallRule -Name sshd -DisplayName "OpenSSH Server (sshd)"

-Enabled True -Direction Inbound -Protocol TCP -Action Allow -LocalPort
22
```

If the rule already exists, you'll get an error saying so. That's fine — it means it's already open.

Step 4 — Test It Locally

In PowerShell, connect to yourself:

```
ssh localhost
```

It will ask if you want to trust the host fingerprint — type `yes`. Then it will ask for your Windows password. Type it. If you get a prompt back, SSH is working.

Step 5 — Connect From Another Machine

Find your IP address:

```
ipconfig
```

Look for **IPv4 Address** under your active adapter (usually Ethernet or Wi-Fi). It will look like 192.168.1.xxx.

From the other machine:

```
ssh YourWindowsUsername@192.168.1.xxx
```

> **Warning:** WinRM and CIM errors are a completely different system — Windows Remote Management, used for PowerShell remoting and enterprise management tools. If you see errors mentioning WinRM, CIM, or WSMan, you're in the wrong place. Close that window. OpenSSH Server has nothing to do with WinRM. They just happen to live near each other in Windows settings and documentation, which is why this costs people hours.

Quick Reference

Task	Command (PowerShell as Admin)
Install	Settings → Optional Features → OpenSSH Server

Start service	Start-Service sshd
Auto-start on boot	Set-Service -Name sshd -StartupType Automatic
Open firewall	New-NetFirewallRule ... -LocalPort 22 (see above)
Test locally	ssh localhost
Find your IP	ipconfig → IPv4 Address
Connect from another machine	ssh Username@IP-Address