TECH TIP — BEGINNER

Why SSH Setup on Windows Took Three Hours

(And How to Do It in 20 Minutes)

From Nolan Law Firm — nemolegal.com/tools

The Real Story

Setting up SSH on Windows should take 20 minutes. For us it took three hours, required two people, and still left things not quite right. This guide documents exactly what went wrong — so you don't have to learn it the same way.

The short version: Windows has two completely separate remote-access systems that live near each other in the documentation and Settings menus. Most guides on the internet mix them up. If you hit the wrong one, you'll spend hours in the wrong rabbit hole.

The Two Systems (And Why They Get Confused)

System	What It Is	What It's For
OpenSSH Server	Standard SSH — same as Linux	Connect via ssh command from any machine
WinRM / CIM / WSMan	Windows Remote Management	PowerShell remoting, enterprise IT management

You want **OpenSSH Server**. Full stop. WinRM is for corporate IT departments managing fleets of Windows machines. It has nothing to do with what you're trying to do.

The Rabbit Holes We Fell Into

Rabbit Hole #1 — WinRM errors. After searching for SSH setup help, we hit guides that mentioned enabling remote access via PowerShell. That enabled WinRM, not SSH. WinRM uses different ports, different authentication, different everything. We spent 45 minutes trying to make it work before realizing it was the wrong tool entirely.

Rabbit Hole #2 — CIM/DCOM errors. WinRM depends on a Windows component called CIM (Common Information Model). When WinRM has issues, it throws CIM errors. Searching for those leads to even deeper Windows enterprise documentation. None of it was relevant. The fix was to abandon WinRM entirely and use OpenSSH instead.

Rabbit Hole #3 — Wrong Optional Feature installed. Windows lists both 'OpenSSH Client' and 'OpenSSH Server' in Optional Features. The Client lets you connect OUT. The Server lets others connect IN. Installing Client when you need Server means SSH appears to be set up but nothing works remotely.

> **Rabbit Hole #4 — Firewall.** Even after correctly installing OpenSSH Server and starting the service, connections from other machines timed out. Windows Firewall was blocking port 22. This is not mentioned in most quick-start guides. It requires a separate firewall rule.

The 20-Minute Version (Now That You Know the Traps)

1. Install the right thing: Settings → System → Optional Features → Add a Feature → **OpenSSH Server**

2. Start the service (PowerShell as Admin):

```
Start-Service sshd

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

3. Open the firewall (PowerShell as Admin):

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

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

4. Test it:

```
ssh localhost
```

5. Connect from another machine:

```
ssh YourWindowsUsername@your-ip-address
```

> **If you see WinRM, CIM, WSMan, or DCOM in any error message:** Stop. Close the window. You're in the wrong place. Those systems have nothing to do with OpenSSH. Go back to Step 1 and make sure you installed OpenSSH Server specifically.
