How to Set Up SSH on Windows

The server side on Windows, with the two gotchas that cost people hours

This is the server side of SSH: turning a Windows machine into one you can connect to. If you only need to connect out to a machine someone else set up, the Windows SSH client is already installed and you can skip this; start with "What SSH Is and Why You'd Use It."

Setting up the SSH server on Windows is genuinely fiddlier than on Linux, not because it is hard, but because two unrelated things sit next to it in the menus and send people down the wrong path for hours. This guide keeps you on the right one.

Quick orientation: the **client** is the part you type into to reach other machines, and Windows 10 and 11 already have it (open PowerShell, type `ssh`, and you should get a usage message). The **server** is the part that lets other machines connect *to* this Windows box, and that is the piece you install below.

Step 1: install OpenSSH Server through Settings

Do this through the Settings app, not with PowerShell commands you found online; half of those are outdated.

Go to **Settings, then System, then Optional Features, then Add a Feature** (on Windows 11, **View Features**), search for **OpenSSH Server**, and install it.

One trap: install **OpenSSH Server**, not OpenSSH Client. The client is probably already there; they are listed separately, and the server is the one you want.

Step 2: start the service

Open PowerShell as Administrator and run:

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

The first line starts SSH now. The second makes it start automatically after a reboot; skip it and you will wonder why SSH stopped working next Monday.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Step 3: open the firewall (the step everyone misses)

Windows Firewall blocks SSH by default even after you install it. Still in an Administrator PowerShell, add the rule:

```
New-NetFirewallRule -Name sshd -DisplayName "OpenSSH Server (sshd)" -Enabled
True -Direction Inbound -Protocol TCP -Action Allow -LocalPort 22
```

If it says the rule already exists, that is fine; it means the port is already open.

Step 4: test it on the machine itself

Before involving another computer, connect to this one:

```
ssh localhost
```

It asks whether to trust the host fingerprint; type `yes`. Then it asks for your Windows password. Type it, and nothing will show as you type, which is normal. If you land at a fresh prompt, the server works.

Step 5: connect from another machine

First find this machine's address:

```
ipconfig
```

Look for the **IPv4 Address** under your active adapter, Ethernet or Wi-Fi, something like `192.168.1.xxx`. Then, from the other computer:

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

The gotcha that costs people hours: WinRM is not this

If you go looking for help and land on anything mentioning **WinRM**, **CIM**, or **WSMan**, close that window. That is Windows Remote Management, a completely different system used for PowerShell remoting and enterprise tooling. It has nothing to do with OpenSSH Server. The two just happen to live near each other in Windows settings and documentation, which is exactly why Windows SSH has a reputation for eating an afternoon.

Stop typing passwords

Password login works, and it is the weak link. An SSH key is more secure and drops the password prompt entirely. That is its own tip in this series: see "How and Why to Use an SSH Key" (part 2), which includes the Windows-specific way to copy your key up, since Windows has no `ssh-copy-id`.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Quick reference

Task	Command (PowerShell as Admin)
Install the server	Settings, Optional Features, OpenSSH Server
Start the service	`Start-Service sshd`
Auto-start on boot	`Set-Service -Name sshd -StartupType Automatic`
Open the firewall	`New-NetFirewallRule ... -LocalPort 22` (see Step 3)
Test locally	`ssh localhost`
Find your IP	`ipconfig` (IPv4 Address)
Connect from another machine	`ssh Username@IP-Address`
Switch to key login	See part 2, How and Why to Use an SSH Key

That is a Windows machine you can reach over SSH, with the two traps that usually derail it stepped around instead of into.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.