

# OpenSSH on Windows

[https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh\\_install\\_firstuse?tabs=gui](https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse?tabs=gui)

## Check if OpenSSH is available

```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'
```

## Install OpenSSH Client and Server

```
# Install the OpenSSH Client
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0

# Install the OpenSSH Server
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```

## Start OpenSSH Server

```
# Start the sshd service
Start-Service sshd
```

## Set OpenSSH Server to start automatically on boot

```
# OPTIONAL but recommended:
Set-Service -Name sshd -StartupType 'Automatic'
```

## Make sure the firewall doesn't block the OpenSSH Server

```
# Confirm the Firewall rule is configured. It should be created automatically by setup. Run
the following to verify
if (!(Get-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -ErrorAction SilentlyContinue |
Select-Object Name, Enabled)) {
    Write-Output "Firewall Rule 'OpenSSH-Server-In-TCP' does not exist, creating it..."
    New-NetFirewallRule -Name 'OpenSSH-Server-In-TCP' -DisplayName 'OpenSSH Server (sshd)' -
Enabled True -Direction Inbound -Protocol TCP -Action Allow -LocalPort 22
} else {
    Write-Output "Firewall rule 'OpenSSH-Server-In-TCP' has been created and exists."
}
```

# Run sshd on an alternate port

Edit the Port line in `C:\ProgramData\ssh\sshd_config`:

```
Port 12322
```

Use the following command to update the Windows Firewall rule to match the port you specified above:

```
Set-NetFirewallRule -Name 'OpenSSH-Server-In-TCP' -LocalPort 12322
```

## Require publickey and password authentication

Add the following line to `C:\ProgramData\ssh\sshd_config`

```
AuthenticationMethods "publickey,password"
```

## publickey authentication for administrators

Public keys for administrators must be put in the file

`%PROGRAMDATA%\ssh\administrators_authorized_keys`. Use the script below to make sure the file has proper permissions.

```
$acl = Get-Acl C:\ProgramData\ssh\administrators_authorized_keys
$acl.SetAccessRuleProtection($true, $false)
$administratorsRule = New-Object
system.security.accesscontrol.filesystemaccessrule("Administrators","FullControl","Allow")
$systemRule = New-Object
system.security.accesscontrol.filesystemaccessrule("SYSTEM","FullControl","Allow")
$acl.SetAccessRule($administratorsRule)
$acl.SetAccessRule($systemRule)
$acl | Set-Acl
```

## Stop and Start the sshd service

```
Stop-Service sshd
```

```
Start-Service sshd
```

```
netstat -an -p TCP | find '"22"'
```

## Example configuration

Port 12322

#PubkeyAuthentication yes

AuthenticationMethods "publickey,password"

AuthorizedKeysFile .ssh/authorized\_keys

#PasswordAuthentication yes

#PermitEmptyPasswords no

# override default of no subsystems

Subsystem sftp sftp-server.exe

Match Group administrators

AuthorizedKeysFile \_\_PROGRAMDATA\_\_/ssh/administrators\_authorized\_keys

## Use PowerShell as default interpreter

# Check current value of registry key

Get-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell

# Legacy PowerShell

New-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell -Value

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -PropertyType String -Force

# PowerShell v7

New-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell -Value "C:\Program  
Files\PowerShell\7\pwsh.exe" -PropertyType String -Force

# Delete registry key to revert to traditional command line

Remove-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell

#end

---

Revision #6

Created 25 June 2023 19:27:22 by bluecrow76

Updated 24 October 2024 22:37:19 by bluecrow76