Windows SSH Server

Native Feature of Windows

Microsoft has adopted openssh as Feature / Capability since Windows version 10 / Windows Server 2019. The openssh functionality is delivered in two independent features - client and server. The main benefit for the user is a complex implementation covering the PowerShell module or the Firewall rule. Other important benefit is delivering updates directly over the system update system.

In previous version you could achieve the SSH access by installing the independent build of openssh but this scenario is not covered in this page.

Installation / enabling the service

You can use GUI to enable the feature but there is not one place where all the stuff would be done. Enabling the Feature (Windows 10) or Capability (Windows 2019) is a first step but there is also a need of enabling and starting the system service. Optionally the firewall rule could be checked.

PowerShell

Primary way to configure the system is to use the PowerShell. Adopting openssh as the native feature / capability also covers the implementation of the management of the service into the powershell environment - the PowerShell module is created. It offers the possibility to do all the necessary things from one console at the same time.

To process following commands the PowerShell console should be run with elevated permission (as administrator).

Check the version

```
Get-WindowsCapability -Online | ? Name -like 'OpenSSH*'

# This should return the following output:

Name  : OpenSSH.Client~~~~0.0.1.0
State : NotPresent
Name  : OpenSSH.Server~~~~0.0.1.0
State : NotPresent
```

OpenSSH - server

Install OpenSSH - server

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

# enable the service (so far not starting just to set the autostart option)
Set-Service -Name sshd -StartupType 'Automatic'
Set-Service -Name ssh-agent -StartupType Automatic

# start the service now
Start-Service sshd
Start-Service ssh-agent

# Optionally you can check the firewall rule
Get-NetFirewallRule -Name *ssh*
# There should be a firewall rule named "OpenSSH-Server-In-TCP", which should be enabled

# Just in case the rule is not there you can add it running
New-NetFirewallRule -Name sshd -DisplayName 'OpenSSH Server (sshd)' -Enabled True -Direction Inbound -Protocol
TCP -Action Allow -LocalPort 22
```

The default shell is **cmd.exe** but it is possible to change it to the PowerShell.

OpenSSH - PowerShell as default shell

```
# Set PowerShell as default shell after the login
New-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell -Value "C:
\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -PropertyType String -Force
```

Remove OpenSSH - server

```
# Uninstall the OpenSSH Server
Remove-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```

OpenSSH - client

install OpenSSH - client

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

Remove OpenSSH - client

```
Remove-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

OpenSSH config

Common configuration

The global system configuration (in *nix system ususally located /etc/ssh) can be found %programdata%/ssh/ (c:\ProgramData\ssh\ ). There is located configuration file and also the keys (used for the secure communication on server side)

- sshd_config
- *_key

For our purpose we don't need to cover all the options available for openssh. As the build has been customized for the purpose of the integration into the windows system there are some options which can't be used in sshd_config the same way as in the linux system. To see more details please see Microsoft Docs page.

User keys (key authentication)

Default location is in user's home directory in the .ssh folder (%HOME%\.ssh\authorized_keys).

> **administrator access (SSH Keys)**
>
> In case the user is a member of the administrator group the key should be placed in the common location instead of user home directory. In this case the location is %programdata%\ssh\administrators_authorized_keys (C:\ProgramData\ssh\administrators_authorized_keys).

To set the proper permission for the file you can use following PowerShell script.

Permission for authorized_keys

```
#get the ACL object for the file
$acl = Get-Acl C:\ProgramData\ssh\administrators_authorized_keys

#set the proper permissions
$acl.SetAccessRuleProtection($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)

#process the setting
$acl | Set-Acl
```

External Link

- OpenSSH
- OpenSSH @ Microsoft Docs

See Also

- SSH Connector