



This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use. [Learn more](#)

Server & Tools Blogs > Server & Management Blogs > PowerShell Team Blog

[Sign in](#)

PowerShell Team Blog

Automating the world one-liner at a time...

Using the OpenSSH Beta in Windows 10 Fall Creators Update and Windows Server 1709



December 15, 2017 by [Joey Aiello \[MSFT\]](#) // [54 Comments](#)

Share 93

0

0

I'm thrilled to share that a Beta OpenSSH client and server daemon are available as a Feature-on-Demand in Windows 10 Fall Creators Update and [Windows Server 1709](https://blogs.technet.microsoft.com/windowsserver/2017/09/25/faster-releases-for-a-faster-world-first-semi-annual-release-of-windows-server/) (<https://blogs.technet.microsoft.com/windowsserver/2017/09/25/faster-releases-for-a-faster-world-first-semi-annual-release-of-windows-server/>). Since [our last update blog](https://blogs.msdn.microsoft.com/powershell/2015/10/19/openssh-for-windows-update/) (<https://blogs.msdn.microsoft.com/powershell/2015/10/19/openssh-for-windows-update/>), we've been working hard on a [Win32 port of OpenSSH](https://github.com/powershell/Win32-OpenSSH) (<https://github.com/powershell/Win32-OpenSSH>) and working closely with members of the OpenSSH Portable and OpenBSD projects with the eventual goal of bringing Win32 support upstream into OpenSSH Portable.

Until then, you should expect OpenSSH support in Windows to continue to improve in future updates of Windows, including upcoming Windows Insider builds. You can track our progress [on GitHub](https://github.com/powershell/Win32-OpenSSH) (<https://github.com/powershell/Win32-OpenSSH>) where you can find [our wiki](https://github.com/powershell/Win32-OpenSSH/wiki) (<https://github.com/powershell/Win32-OpenSSH/wiki>) and the [latest builds](https://github.com/powershell/Win32-OpenSSH) (<https://github.com/powershell/Win32-OpenSSH>) that include tons of fixes and support for operating systems downlevel to Windows 7 and Server 2008 R2.

Overview

OpenSSH (<https://www.openssh.com/>) is a collection of client/server utilities that enable secure remote login, remote file transfer, and public/private key pair management. It's an extremely powerful tool that originated as part of the OpenBSD project, and has been used for many years across the BSD, Linux, macOS, and Unix ecosystems.

Note: The OpenSSH client and server are still very much in Beta, so we do not recommend using them in production environments.

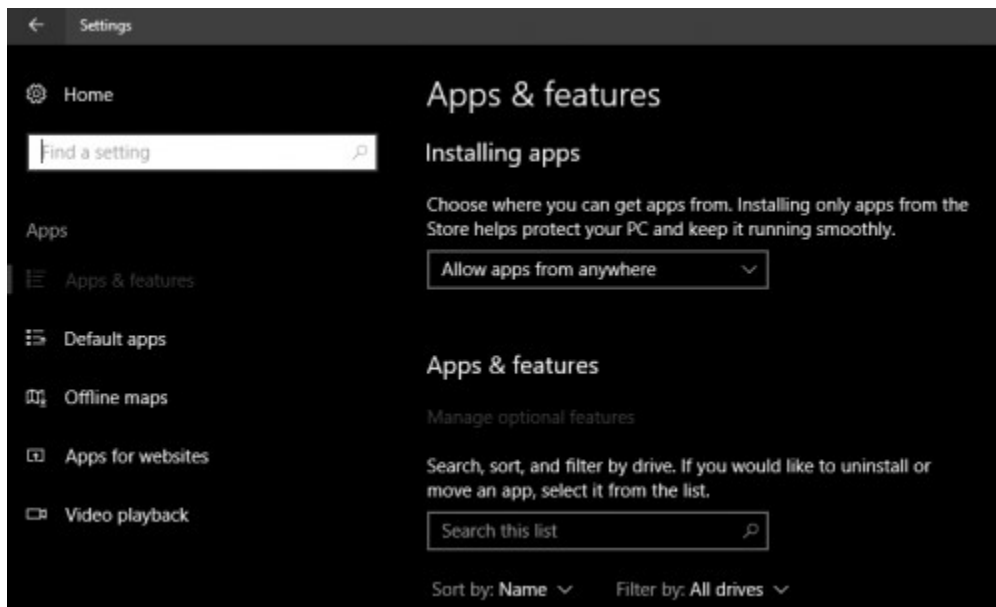
Installation

Great! So how do I install the bits?

Installing with the Settings UI

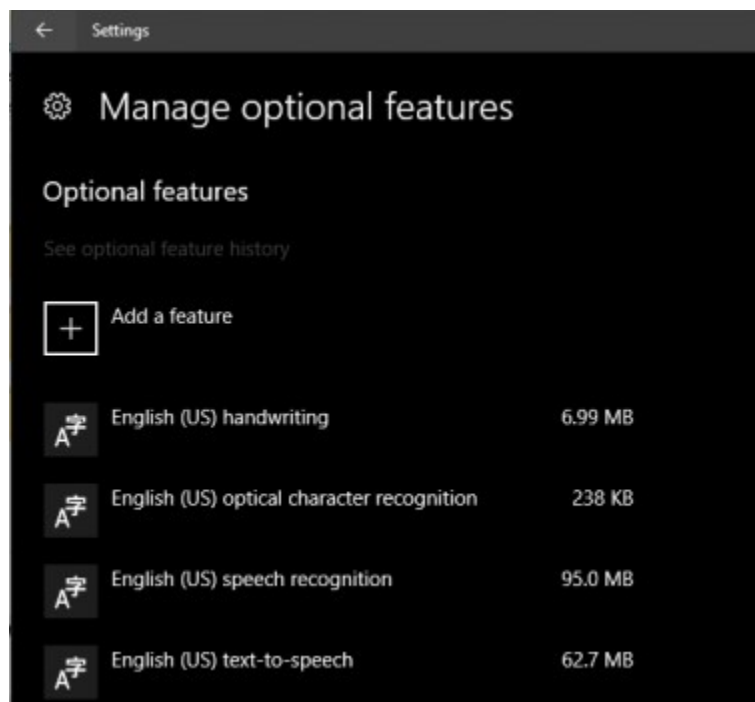
To install it using the Settings UI, go to Apps -> Apps and Features -> Manage optional features -> Add a feature:

(https://msdnshared.blob.core.windows.net/media/2017/12/apps_and_features_dark.png)



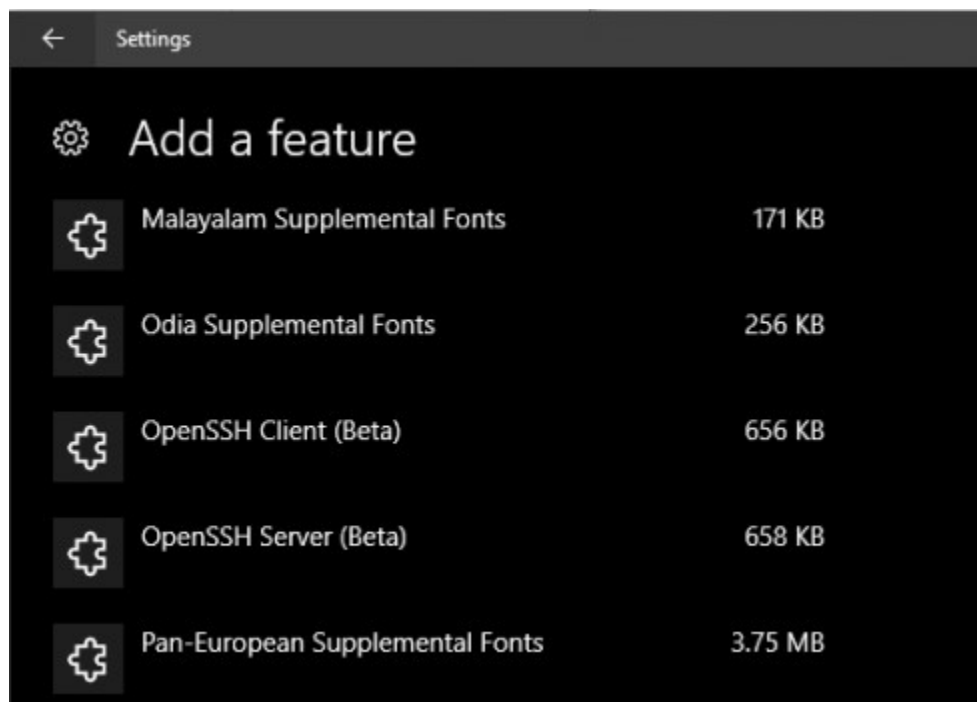
(https://msdnshared.blob.core.windows.net/media/2017/12/apps_and_features_dark.png)

tures_dark.png)



(https://msdnshared.blob.core.windows.net/media/2017/12/manage_optional_features_dark.png)

Then select OpenSSH Client (Beta) or OpenSSH Server (Beta) and Install:



https://msdnshared.blob.core.windows.net/media/2017/12/add_a_feature_dark.png
(/path/to/image)

Installing with PowerShell

To install OpenSSH using PowerShell, first launch PowerShell as an Administrator.

To make sure that the OpenSSH features are available for install:

```
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
```

Then, install the server and/or client features:

```
# 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
```

Both of these should return the following output:

```
Path      :
Online    : True
RestartNeeded : False
```

Installing with DISM.exe

To install OpenSSH with DISM.exe, first open CMD as an Administrator.

To make sure that OpenSSH features are available for install:

```
dism /Online /Get-Capabilities | findstr OpenSSH
```

This should return the following output:

```
Capability Identity : OpenSSH.Client~~~~0.0.1.0  
Capability Identity : OpenSSH.Server~~~~0.0.1.0
```

Then, install the server and/or client features:

```
dism /Online /Add-Capability /CapabilityName:OpenSSH.Client~~~~0.0.1.0  
dism /Online /Add-Capability /CapabilityName:OpenSSH.Server~~~~0.0.1.0
```

Configuration

Great! You've installed OpenSSH. What now?

Configuring the SSH Client (ssh.exe)

Password-based authentication

If you want to use the SSH client with password authentication, no configuration is necessary. Just pop open PowerShell or cmd, and use `ssh` to connect to your SSH server:

```
ssh user1@contoso.com  
  
# You can also use domain accounts to login  
  
# UPN syntax works...  
ssh user1@domain1@contoso.com  
# ...as does NetBIOS syntax  
ssh user1\domain1@contoso.com
```

Key-based authentication

If you want to use key-based authentication, you first need to generate some public/private key pairs for your client. From PowerShell or cmd, use `ssh-keygen` to generate some key files.

```
cd ~\.ssh\  
ssh-keygen
```

This should output something like:

```
Generating public/private ed25519 key pair.  
Enter file in which to save the key (C:\Users\user1\.ssh\id_ed25519):
```

You can hit Enter to accept the default or specify a path where you'd like your keys to be generated. At this point, you'll be prompted to use a passphrase to encrypt your private key files.

```
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in C:\Users\user1\.ssh\id_ed25519.  
Your public key has been saved in C:\Users\user1\.ssh\id_ed25519.pub.  
The key fingerprint is:  
SHA256:0Izc1yE7joL2Bzy8/gS0j8eGK7bYaH1FmF3sDuMeSj8 user1@CONTOSO@LOCAL-HOSTNAME  
The key's randomart image is:  
+--[ED25519 256]--+  
|      .      |  
|      o      |  
|    . + + .   |  
|   o B * = .  |  
|  o= B S .    |  
|   . = B O o   |  
|  + = + % o    |  
| *oo.O.E      |  
|+.o+=o. .     |  
+-----[SHA256]-----+
```

Now you have a public/private ED25519 key pair
(the .pub files are public keys and the rest are private keys):

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	11/8/2017 11:09 AM	1679	id_ed25519
-a----	11/8/2017 11:09 AM	414	id_ed25519.pub

Your private key files are the equivalent of a password. You should protect them under any and all circumstances. If someone acquires your private key, they can log in to any SSH server as an identity that authorizes the corresponding public key to log in.

For that reason, we should take advantage of `ssh-agent` to securely store the private keys within a Windows security context. To do that, we simply start the `ssh-agent` service (as Administrator) and use `ssh-add` to store our private key. Then, whenever a private key is needed for authentication, `ssh-agent` will automatically retrieve your local user's private key and pass it to your SSH client.

```
# Make sure you're running as an Administrator
Start-Service ssh-agent

# This should return a status of Running
Get-Service ssh-agent

# Now load your key files into ssh-agent
ssh-add ~\.ssh\id_ed25519

# Now that it's loaded into ssh-agent,
# we don't have to keep the key file anymore
Remove-Item ~\.ssh\id_ed25519
```

Move the contents of your public key (`~\.ssh\id_ed25519.pub`) into a text file called `authorized_keys` in `~\.ssh\` on your server/host.

Note: these directions assume your `sshd` server is a Windows-based machine using our OpenSSH-based server, and that you've properly configured it based on the instructions below (including the installation of the `OpenSSHUtils` PowerShell module). If you're using a non-Windows machine, you should replace all remote instances of `C:\users\user1` with something like `/home/user1`. Additionally, the ACL line should be unnecessary that uses PowerShell should be unnecessary.

```
# Make sure that the .ssh directory exists in your server's home folder
ssh user1@domain1@contoso.com mkdir C:\users\user1\.ssh\

# Copy your public key file to authorized_keys on your server
scp C:\Users\user1\.ssh\id_ed25519.pub
user1@domain1@contoso.com:C:\Users\user1\.ssh\authorized_keys

# Appropriately ACL the authorized_keys file on your server
ssh --% user1@domain1@contoso.com powershell -c $ConfirmPreference = 'None';
Repair-AuthorizedKeyPermission C:\Users\user1\.ssh\authorized_keys
```

Congrats! You should no longer need a password when authenticating as `User1` against `contoso.com`.

Configuring the OpenSSH Server (sshd)

First, it's worth noting again that this OpenSSH for Windows is still very much in beta form. It should only be used in safe, testing environments.

To enable authentication into an SSH server on Windows, you first have to generate host keys. As an Administrator:

```
Start-Service ssh-agent

cd C:\Windows\System32\OpenSSH
.\ssh-keygen -A
# C:\Windows\System32\OpenSSH\ssh-keygen.exe: generating new host keys: ED25519
.\ssh-add ssh_host_ed25519_key
# Identity added: .\ssh_host_ed25519_key (User1@CONTOSO@LOCAL-HOSTNAME)
```

Due to certain security requirements, you will also have to install our `OpenSSHUtils` helper module to appropriately ACL your host keys. As an Administrator:


```
Install-Module -Force OpenSSHUtils
```

```
Repair-SshdHostKeyPermission -FilePath
```

```
C:\Windows\System32\OpenSSH\ssh_host_ed25519_key
```

```
# Use A or Y as your response to the prompts to set file owners
```

Then you can start sshd and your server is ready to go:

```
Start-Service sshd
```

```
# This should return a Status of Running
```

```
Get-Service sshd
```

Note: currently only the built-in ED25519 authentication key type is supported. In the future, we plan to add support for LibreSSL which will enable additional authentication key types. In the meantime, you can experiment with LibreSSL builds [on GitHub \(https://github.com/powershell/openssh/releases\)](https://github.com/powershell/openssh/releases).

You may also need to add a firewall rule like this one that allows traffic on port 22 (though your requirements may vary based on your environment, e.g. Domain might be Private):

```
New-NetFirewallRule -Name sshd -DisplayName 'OpenSSH Server (sshd)' -Service  
sshd -Enabled True -Direction Inbound -Protocol TCP -Action Allow -Profile  
Domain
```

Stay tuned!

Enjoy playing with OpenSSH on Windows, and keep your eyes peeled on the PowerShell blog for upcoming news.

Search MSDN with Bing



☐ Search this blog ☒ Search all blogs



[ScottGu's Blog](#)

[Brad Anderson's "In the Cloud" Blog](#)

[Brian Harry's Blog](#)

[Steve "Guggs" Guggenheimer's Blog](#)

Recent Posts

[Parsing Text with PowerShell \(1/3\)](#) January 18, 2019

[Windows Security change affecting PowerShell](#) January 10, 2019

[DSC Resource Kit Release January 2019](#) January 9, 2019

[UPDATE: The PowerShell-Docs repo is moving ... soon](#) January 7, 2019

Tags

[Advanced Functions](#) [CMDLET](#) [Community](#) [CTP3](#) [Desired State](#)
[Configuration](#) [DOCUMENTATION](#) [DOTNET](#) [DSC](#) [DSC Resource](#)
[Kit](#) [DSC Resources](#) [FAQ](#) [Get-Help](#) [Help](#) [Help file](#) [HowTo](#) [INTERNAL](#) [Jeffrey Snover](#)
[LANGUAGE](#) [Module](#) [Modules](#) [News](#) [PHILOSOPHY](#) [PowerGUI](#) [PowerShell](#)
[PowerShell 4.0](#) [PowerShell 5.0](#) [PowerShell Deep Dive](#) [PowerShell ISE](#)
[PowerShell Release](#) [PowerShell V2](#) [Presentation](#) [Quest](#) [Release/Download](#) [Remoting](#)
[Resources](#) [Sapien](#) [TechED](#) [Video](#) [Windows Management Framework](#)
[Windows PowerShell 4.0](#) [Windows PowerShell 5.0](#) [Windows](#)
[PowerShell Desired State Configuration](#) [Windows Server](#)
[2012](#) [WMI](#) [WPF](#)

Archives

[January 2019](#) (4)

[All of 2019](#) (4)

[All of 2018](#) (23)

[All of 2017](#) (44)
[All of 2016](#) (48)
[All of 2015](#) (69)
[All of 2014](#) (54)
[All of 2013](#) (29)
[All of 2012](#) (33)
[All of 2011](#) (29)
[All of 2010](#) (44)
[All of 2009](#) (169)
[All of 2008](#) (193)
[All of 2007](#) (208)
[All of 2006](#) (210)



GTTGd

8 months ago

I want this to work with git ssh keys. I've configured the ssh-agent as stated and I can see my GitHub key in ssh-add.exe -l. However when I do a git pull it checks the host configured under ~/.ssh/config and looks for the IdentityFile={KeyFile}. If I specify my {KeyFile} here and the file exists everything works fine. However, if the file doesn't exist (since this article says we can remove the key file as it has been cached by ssh-agent) then it does work.

What am I doing wrong and how does git know to look up the file name in ssh-agent?

Thank you!



Collin Chaffin

8 months ago

Hey Joey,

You may want to take a look at this issue I just opened. It shows a full reproduction screengrab demo of OpenSSH-Win32 every version going back years to when Docker was first released and how ONLY using OpenSSH is what has been responsible for MANY of these "mystery" 255/SSH/you-name-it Docker on Windows issues..

I am having trouble debugging much further without DEVs from both MS and Docker involved at which time I'm certain it can be mitigated very quickly – but that is VERY overdue.

I'm guessing it will make you scratch your head just as much as it did mine when you watch it. In my thirty years it's at the top of the list of weirdest ones I've come across but

still a bit surprising to me that nobody uncovered it until now as popular as Docker has become.

Here's the issue link:

<https://github.com/PowerShell/Win32-OpenSSH/issues/1155>

As you (in this post) and most others are more and more pointing users to OpenSSH as the go-to SSH client on Windows, I'm sure you agree how important it is to minimize major compatibility issues like this so hopefully this is helpful information for you.

-Collin



Ben Hastings72

10 months ago

It works! Here's some issue I've noticed:

- a) If the computer is connected to WSUS, it won't find OpenSSH. There seems to be no way to import/approve and I can't find it in the Update Catalog. Workaround > Temporarily disconnect computer from WSUS
- b) I had to add the user to Replace a process level token like Ken mentioned (secpol.msc)
- c) Putty as client cannot connect to openSSH server sshd. This may be by design tough.



Ayan Mullick

11 months ago

There should be a powershell cmdlet for the ssh-agent.



Scott Jibben

1 year ago

I've installed the OpenSSH client on Windows 10 Pro:

```
C:\Users\sjibben> $PSVersionTable.PSVersion
```

Major Minor Build Revision

5 1 16299 98

ssh only works when using an elevated PowerShell for me. Shouldn't it work for standard PowerShell sessions?



Joey Aiello [MSFT]

1 year ago

Yup! It may just be that your PATH didn't get updated on non-elevated sessions. Sometimes restarting explorer.exe does it for me, other times I need to reboot. On a per-session basis, you should just be able to run this:

```
$env:PATH += 'C:\Windows\system32\OpenSSH'
```



Ken

1 year ago

In regard to setting up the SSH server, I had to manually add the "Replace a process level token" privilege to the NT Service\sshd user in order for the sshd service to start. Was that supposed to happen automatically? I tried rebooting after following the above steps, but it didn't make a difference.



Joey Aiello [MSFT]

1 year ago

I haven't heard of anyone having to do that, no. What prompted you to figure out that was the problem?



kenhiatt

11 months ago

Just had the same experience. Luckily the other Ken had the solution. Added the right and sshd started.



Yauheni

1 year ago

Can I use ssh key generated by ssh-keygen on Ubuntu? I have the following error "Load key "file_name": invalid format".



Joey Aiello [MSFT]

1 year ago

> Note: currently only the built-in ED25519 authentication key type is supported. In the future, we plan to add support for LibreSSL which will enable additional au-

thentication key types. In the meantime, you can experiment with LibreSSL builds on GitHub.



pramod

1 year ago

I've switched from Linux to Windows because of the Linux Subsystem feature. When will Microsoft roll out the stable release of OpenSSH client and server? Is the current release of openssh safe to use on a laptop which is connected to the internet most of the time?



Jason Fossen

1 year ago

On Windows 10, I was able to get the sshd service to start when I explicitly granted the "Replace a process level token" privilege (SeAssignPrimaryTokenPrivilege) to the "NT Service\sshd" identity in the local GPO. This is the only required priv listed in the registry for the sshd service.

Also, if you're experimenting with other user accounts, see the new 'SSH Users' group too.



Richard.Siddaway

1 year ago

Just installed OpenSSH beta on Windows 10 VM – wanted to use it for SSH based remot-ing with PowerShell v6. OpenSSH is installed in C:\Windows\System32 which makes con-figuration changes – i.e. adding the subsystem so OpenSSH can find pwsh.exe very diffi-cult at best and impossible at worst.

I'm backing out the install and will re-install with the rather messy install using the github download.



Joey Aiello [MSFT]

1 year ago

Sorry you hit that, Richard. What kind of problems did you run into? Editing sshd_config as non-Admin, I presume?

Also, it maybe wasn't clear in this blog, but if you already have the GitHub bits, you're way further ahead on the development train than what we have in Win-dows.



Richard.Siddaway

1 year ago

It comes down to the permissions set on the config file because its in C:\windows\System32. Editing any file in C:\windows* is problematic at best on the later Windows systems. The drive to Windows doing more for the users to protect people who don't know what they're doing makes life difficult for people who do know what should be changed



Joey Aiello [MSFT]

1 year ago

Agreed, we're working on a solution there to fix that problem. We'll keep you posted.



Richard.Siddaway

1 year ago

Am I missing something because OpenSSH beta doesn't seem to be available on the Server 1709 VM I built from the iso file.
using Get-WindowsCapability

```
PS> Get-WindowsCapability -Online
Get-WindowsCapability : Get-WindowsCapability failed. Error code = 0x800f0954
At line:1 char:1
+ Get-WindowsCapability -Online
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [Get-WindowsCapability], COMException
+ FullyQualifiedErrorId : Microsoft.Dism.Commands.GetWindowsCapabilityCommand
```

generates a COM error and
using DISM returns nothing



Joey Aiello [MSFT]

1 year ago

Looking into this, appears to be some kind of issue with the FOD system. It may require you to be domain-joined or have a linked MSA in order to access features-on-demand, but I'm not sure yet.



Richard.Siddaway

1 year ago



Thanks for the update Joey
The Windows Server 1709 VM is domain joined but a Windows Server
17035 VM that isn't domain joined shows the same behaviour



Kaman

1 year ago

I try to use Password-based authentication
I type: ssh [user@domain.com](#), and get

Unable to negotiate with xx.xx.xx.xx port 22: no matching host key type found. Their offer: ssh-rsa,rsa-sha2-512,rsa-sha2-256,ecdsa-sha2-nistp256

Do I have to use Key-based authentication?



Joey Aiello [MSFT]

1 year ago

If the host is a Windows machine where you're using the release outlined here, then you need to follow the instructions above to generate and configure host keys.

If you don't own the host, you need it to have ED25519-based host keys (unless you use our latest LibreSSL-based client from GitHub):

> Note: currently only the built-in ED25519 authentication key type is supported. In the future, we plan to add support for LibreSSL which will enable additional authentication key types. In the meantime, you can experiment with LibreSSL builds on GitHub.



Keith Hill

1 year ago

When I run PowerShell x86, it can't find scp.exe even though %SystemRoot%\System32\OpenSSH is in the path. I have to add %SystemRoot%\sysnative\OpenSSH to the path. That doesn't seem right.



Joey Aiello [MSFT]

1 year ago

You're right, that doesn't seem right. Not sure what the fix is, Keith, let me get back to you on it.

That being said, where are you using x86 Windows PowerShell that you need SSH? (Just out of curiosity... 😊)



Rob Cannon

1 year ago

Where should we log issues? This version is not working as a client. I did have the GitHub PowerShell/Win32-OpenSSH version installed before and it worked. I did remove that version before installing, this. I would like to log the various issues that I am seeing.



WSLUser

1 year ago

Will this increase SSH interop between Windows and WSL? Running the sshd daemon from bash is still rather buggy. Interop of course would at this time only increase for ED25519 key pairs if at all.



Joey Aiello [MSFT]

1 year ago

Not sure how it would help interop, but certainly running sshd will be less buggy (and you can hop into WSL via bash.exe after you've connected to a remote machine via this sshd).



TechPreacher

1 year ago

Very cool! Do you have a pointer for me how to add legacy host key algorithms to c:\users\user\.ssh\config so that OpenSSH for Windows allows me to connect to older hosts?

I tried:

Host xyz

HostName xyz.domain.com

HostKeyAlgorithms ssh-rsa rsa-sha2-512 rsa-sha2-256

but I get back "Bad key types 'ssh-rsa'" from OpenSSH.



Joey Aiello [MSFT]

1 year ago

> Note: currently only the built-in ED25519 authentication key type is supported. In the future, we plan to add support for LibreSSL which will enable additional authentication key types. In the meantime, you can experiment with LibreSSL builds on GitHub.



CJ Harmath

1 year ago

This is great!

Can you please comment on Kerberos support ? (didn't get a chance yet to try)



Ken

1 year ago

OpenSSH is a no show on 10.0.17025.1000. Nothing in settings, PowerShell or dism. It looks like some problem with Get-WindowsCapability.

```
WindowsPowerShell> Get-WindowsCapability -Online -Verbose
```

```
VERBOSE: Target Image Version 10.0.17025.1000
```

```
Get-WindowsCapability : Get-WindowsCapability failed. Error code = 0x800f0950
```

```
At line:1 char:1
```

```
+ Get-WindowsCapability -Online -Verbose
```

```
+ ~~~~~
```

```
+ CategoryInfo          : NotSpecified: (:) [Get-WindowsCapability], COMException
```

```
+ FullyQualifiedErrorId : Microsoft.Dism.Commands.GetWindowsCapabilityCommand
```



Renato Martini

1 year ago

I found a weird problem caused by a space in my WIn10 User Name:

```
PS C:\Users\RM INFORMATICA\.ssh> ssh-keygen.exe -t ed25519
```

```
Generating public/private ed25519 key pair.
```

```
Enter file in which to save the key (C:\Users\RM INFORMATICA\.ssh\id_ed25519): c:\Us-  
ers\RM INFORMATICA\.ssh\id_ed25519
```

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

```
Saving key "id_ed25519" failed: invalid argument
```

(I dont know exactly if there was a problem with the space or the space + the Unix "/"...)

What I did ?I created a new .ssh dir in the C:\ root, I generated the key pair and forced used the c:\.ssh, and finally I tranfer it to the home .ssh:

```
PS C:\> cd .ssh
PS C:\.ssh> ssh-keygen.exe
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\RM INFORMATICA/.ssh/id_ed25519):
c:\.ssh/id_ed25519
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in c:\.ssh/id_ed25519.
Your public key has been saved in c:\.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:ISWjlfRkcuXI8OmF/E9xfnXFgOuqu4D533bY+tyi6BU RM INFORMATI-
CA@DELL@dell
```

There are another solution? Perhaps more elegant...?



Stewart Adam

1 year ago

Followed the instructions but service fails to start:

```
Start-Service : Service 'sshd (sshd)' cannot be started due to the following error: Cannot
start service sshd on
computer ''.
```

```
At line:1 char:1
```

```
+ Start-Service sshd
```

```
+ ~~~~~
```

```
+ CategoryInfo          : OpenError: (System.ServiceProcess.ServiceController:ServiceController)
[Start-Service],
```

```
ServiceCommandException
```

```
+ FullyQualifiedErrorId : CouldNotStartService,Microsoft.PowerShell.Commands.StartServiceCommand
```



ddalu

1 year ago

Thx for the nice article but what can you do when it is does not show up?

I'm running the windows 10 enterprise 1709 but openssh doesn't show up (it is not available for installation).

Version: 10.0.1439

None of the methods above do work as, openssh is not shown as being available.

I tried tools like "sfc /scannow" or "DISM /Online /Cleanup-Image /CheckHealth" but I didn't get any errors or warnings. So I assume everything is ok with the OS.

Any help would be greatly appreciated.
Thanks!
Didier



ddalu

1 year ago

Sorry typo ... correct version is :10.0.16299 (NOT 10.0.1439 ...!)



Joey Aiello [MSFT]

1 year ago

I'm not sure what's going on there. It looks like Paulo is having this problem as well.

Can you confirm that you're using 10.0.16299.15? Running ``$PSVersionTable.BuildVersion.ToString()`` in PowerShell will give you the full OS version.

Also, are you perchance using a non-en-us locale?

Sometimes things get wonky with the FODs getting published to particular branches/builds, just trying to narrow it down.



ddalu

1 year ago

Hi thx for replying.

Let me adjust my first post, and be more specific.

Here is the output of the windows version I'm using:

```
PS C:\Windows\system32> $PSVersionTable.BuildVersion.ToString()
10.0.16299.15
```

I don't see any openssh software available via the GUI "manage optional features", and when I use powershell to install openssh I get an error while trying to install openssh:

```
PS C:\Windows\system32> Add-WindowsCapability -Online -Name
OpenSSH.Client~~~~0.0.1.0
```

```
Add-WindowsCapability : Add-WindowsCapability failed. Error code
= 0x800f0950
```

```
At line:1 char:1
```

```
+ Add-WindowsCapability -Online -Name OpenSSH.Cli-
ent~~~~0.0.1.0
```

```
+
```

```
~~~~~  
~~~~~  
+ CategoryInfo : NotSpecified: (:) [Add-WindowsCapability], COMEx-  
ception  
+ FullyQualifiedErrorId : Microsoft.Dism.Commands.AddWin-  
dowsCapabilityCommand
```

Here is my info:

```
PS C:\Windows\system32> get-culture | fl -property *
```

```
Parent : en  
LCID : 1033  
KeyboardLayoutId : 1033  
Name : en-US  
letfLanguageTag : en-US  
DisplayName : English (United States)  
NativeName : English (United States)  
EnglishName : English (United States)  
TwoLetterISOLanguageName : en  
ThreeLetterISOLanguageName : eng  
ThreeLetterWindowsLanguageName : ENU  
CompareInfo : CompareInfo – en-US  
TextInfo : TextInfo – en-US  
IsNeutralCulture : False  
CultureTypes : SpecificCultures, InstalledWin32Cultures, Framework-  
Cultures  
NumberFormat : System.Globalization.NumberFormatInfo  
DateTimeFormat : System.Globalization.DateTimeFormatInfo  
Calendar : System.Globalization.GregorianCalendar  
OptionalCalendars : {System.Globalization.GregorianCalendar, Sys-  
tem.Globalization.GregorianCalendar}  
UseUserOverride : True  
IsReadOnly : False
```

I have installed the "developer mode" with an Ubuntu Linux a few month ago... and I had huge problems to install the software via the Store. At the time I was using "Luxembourg" in "Region language – Country or Region" and I had different language installed in the "Control Panel\Clock, Language, and Region\Language" feature "add language" , I had English (Luxembourg Keyboard), French (Luxembourg Keyboard), German (Luxembourg Keyboard). I need the languages for the Office spellchecker. But I couldn't launch the MS Store app that was required to install the ubuntu linux os. You can

read my post and the solution on technet here:

<https://social.technet.microsoft.com/Forums/en-US/a5376f94-35a7-4da9-99c8-e69aa683d636/microsoft-store-app-not-working-in-creators-update-something-bad-happened-unknown-layout?forum=win10itprogeneral>

I was able to install access the MS Store after I switched:

- to “United States” in the Country region settings
- and I had to uninstall any additional languages, only keeping US.

If you need any further details, test or whatever please let me know.

Didier



Paulo Morgado

1 year ago

I'm not getting this on Windows 10.0.16299.125 Enterprise.

```
C:\WINDOWS\system32> dism /Online /Add-Capability /CapabilityName:OpenSSH.Cli-ent~~~~0.0.1.0
```

Deployment Image Servicing and Management tool

Version: 10.0.16299.15

Image Version: 10.0.16299.125

Error: 0x800f0954

DISM failed. No operation was performed.

For more information, review the log file.

The DISM log file can be found at C:\WINDOWS\Logs\DISM\dism.log



Brian Wood

1 year ago

Also, the sshd service on the 1709 Server refuses to start, while the SSH-agent does start.



Joey Aiello [MSFT]

1 year ago

You need to follow the directions to above to generate host keys, install OpenSSHUtils, and run Repair-SshdHostKeyPermission on the generated host keys. Until

you do that, sshd will refuse start, and without it running, you can't make a connection.



Napert

1 year ago

i followed all instruction in configuring ssh server (sshd) and when i try Start-Service sshd it just gives me an error "cannot start sshd service on computer" or something



Brian Habel

11 months ago

I had this problem also. Fixed it by checking the REG key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sshd\RequiredPrivileges to find it set to the value of SeAssignPrimaryTokenPrivilege. Then I loaded SecPol.msc and under 'Local Policies' -> 'User Rights Management' -> 'Replace a process level token' added the 'NT SERVICE\sshd' user to this policy. After that, the sshd service started.



Brian Wood

1 year ago

Scenario: Windows 10 1709 Hyper-V VM, with openssh client installed, trying to connect to another Hyper-V VM, server 1709, with openSSH Server installed. Both are in a workgroup, on a Windows 10 1709 host. Connection by password based authentication does not seem to work, is it not supposed to? I keep getting an error that I cannot connect from WIN10 client to Server 1709 due to port 22 connection refused (firewall is turned off on both VMs). Connection using Project Honolulu works fine in this scenario. Wondering what I am doing wrong if it appears to be easy according to this blog.



响叮咚

1 year ago

how can i save password when new-pssession?
from windows ,to linux,use powershell 6.0,use
New-PSSession -HostName ip -UserName xxx



Macfass

1 year ago

- -

Very nice, but if I try I get

Unable to negotiate with 192.168.73.252 port 22: no matching host key type found. Their offer: ssh-rsa,rsa-sha2-512,rsa-sha2-256

and

ssh -oHostKeyAlgorithms=+ssh-rsa [admin@192.168.73.252](#)

doesn't help!

Any ideas?

KR

Macfass



Joey Aiello [MSFT]

1 year ago

Yes, in Fall Creators Update, we only ED25519-based keys because we haven't included LibreSSL yet.

Future builds will have LibreSSL and will therefore support RSA, DSA, and other key exchange algorithms.



C. Scott

1 year ago

Does that mean key based auth isn't possible? I've created a new ED25519 key pair on my ssh client server, added pub key to authorized_keys, etc. I haven't been able to auth using keys though.



jlm-gh

1 year ago

On some linux servers, ed25519 keys are not accepted by default openssh config. If your openssh is 6.5 and over, you can change that behavior adding this in /etc/ssh/sshd_config

HostKey /etc/ssh/ssh_host_ed25519_key

And restart the ssh service



Piyush Aggarwal

1 year ago

it created an SSH_HOST public key. What am I supposed to do with this key ? Please tell me the further steps...



Piyush Aggarwal

1 year ago

Repair-SshdHostKeyPermission -FilePath C:\Windows\System32\OpenSSH\ssh_host_ed25519_key
here, I mean
so this is a public key created by HOST ? what now ?



Joey Aiello [MSFT]

1 year ago

Now you can start sshd, and connect to that machine using an SSH client.

Were my instructions above unclear somehow?



Graham Bloice

1 year ago

When generating keys and attempting to add a passphrase, which is surely best practice, ssh-keygen fails with an error "invalid argument".

Also what's the upgrade path as bugfixes and improvements are made?



William

1 year ago

This happened to me too



Joey Aiello [MSFT]

1 year ago

Looks like using a passphrase with your user keys (which IS DEFINITELY A BEST PRACTICE), is broken in some cases with the Beta version that we shipped to Fall Creators Update.

You can grab a newer version from GitHub for testing, but it's expected that you uninstall this one first. In the future, new builds will show up in Windows via Windows Update.

Comments are closed.

© 2019 Microsoft Corporation.

[Terms of Use](#) | [Trademarks](#) | [Privacy & Cookies](#)