Configure SSH on Windows 10

Up

Posted by Frank M Kromann
Tags:

Windows 10 Fall Creators Update comes with a hidden gem - ssh client and a less important ssh server, both based on OpenSSH. This is big news as you can now access your favorite server right from the command line without the need for PuTTY or other ssh clients.

This new feature is not installed by default so you will have to go through a few steps to make it available. This is done by following these simple steps:

1. Opening Settings
2. Click on Apps
3. Click on Manage optional features
4. Click on Add a feature
5. Scroll to OpenSSH Client (Beta) and click on it and then click the install button
6. I also had to reboot the computer but that might not be necessary

That's it not you have ssh available on the command line. Typing ssh on the command line looks like this:

If you connect to hosts using userid and password you should be ready to start using the feature. I would recommend using keys to connect, especially if you are in control over the host and can configure the ssh server to disallow PasswordAuthentication by setting "PasswordAuthentication no" in /etc/ssh/sshd_config and restarting the sshd server. Make sure you have tested that you can connect with key before you make this change.

In order to use ssh keys you will have to generate a private and a public key. If you have done this before you are most likely used to the following command:

ssh-keygen -t rsa -b 4096

That will generate a pair of RSA keys typically named id_rsa and id_rsa.pub.

The Windows 10 implementation does not look like it supports RSA keys as the above command returns the error "unknown key type rsa". Changing the command to

ssh-keygen -t ed25519 -b 4096

Does work and it does generate a pair of keys named id_ed25519 and id_ed25519.pub. These will be

placed in a folder called .ssh in the users home directory. I have not tested if it works if the .ssh directory is missing. I had one already from my PuTTY installation.

The next step is to log into the ssh server where you want to use the keys for authentication. When you are logged on to the server you can edit the file ~/.ssh/authorized_keys and add a line where you past the content of id_ed25519.pub. Make sure it is the public file you are copying.

Logout from the server and type

ssh <user>@<hostname>

If the ssh server supports the ed25519 format you should be able to login without using a password. When I tried this on my CentOS 7 based hosts I received errors related to the servers lack of support for the format. It was an easy fix to change that. On CentOS 7 the ssh client and server configuration files are placed in /etc/ssh. In the file called sshd_config there are by default two lines used to define the HostKey values in use for the server. This looks like this:

```
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
```

The file ssh_host_key_ed25519_key is already installed in the folder. WHen you have edited the file you must restart the sshd server with the command:

systemctl restart sshd

That's it, you now have access to ssh on the command line and you can connect to the server without using PuTTY or other ssh clients.

Best of all the **scp** command is also available. You can copy files to and from the server with commands like this:

scp <local file> <user>@<hostnam>:

It is important to remember the trailing : and any directory and filename you would like for the file on the host.

Link to this Post

Get Next 2

Get RSS feed

Recent Blog Posts

Blog Archives

Tags

PHP {2}

C/C++ {0}

Calendar

May 2022

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				