STREAM SERIES

devnator

# SSH remote access with password or encryption keys

SERGIO CABRAL

sergiocabral.com

# Topics

## Practical purpose of this demonstration

Establish a connection via SSH, either using a password, or using a pair of encryption keys.

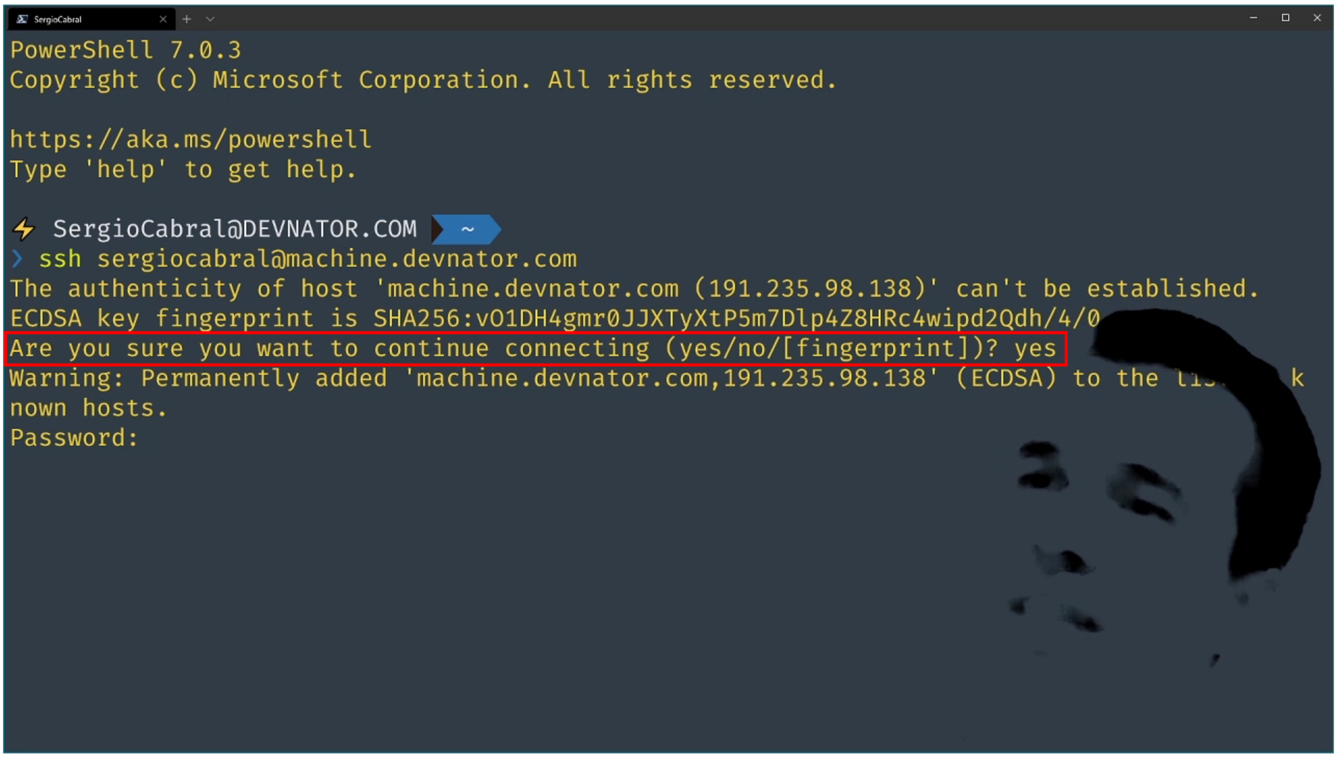
# 1. Connection with password

To connect to a server with the native Windows 10 client, type:

```
ssh username@server-address
```

There are other famous SSH clients as indicated in section Alternatives for SSH clients. But the native Windows 10 client will probably be enough for what you need to do.

When it is the first access you need to inform that you trust the remote computer's identity by answering `yes`, as indicated in *Figure 1*.



```
PowerShell 7.0.3
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/powershell
Type 'help' to get help.

⚡ SergioCabral@DEVNATOR.COM  ~
❯ ssh sergiocabral@machine.devnator.com
The authenticity of host 'machine.devnator.com (191.235.98.138)' can't be established.
ECDSA key fingerprint is SHA256:vO1DH4gmr0JJXTyXtP5m7Dlp4Z8HRc4wipd2Qdh/4/0
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'machine.devnator.com,191.235.98.138' (ECDSA) to the list of k
nown hosts.
Password:
```

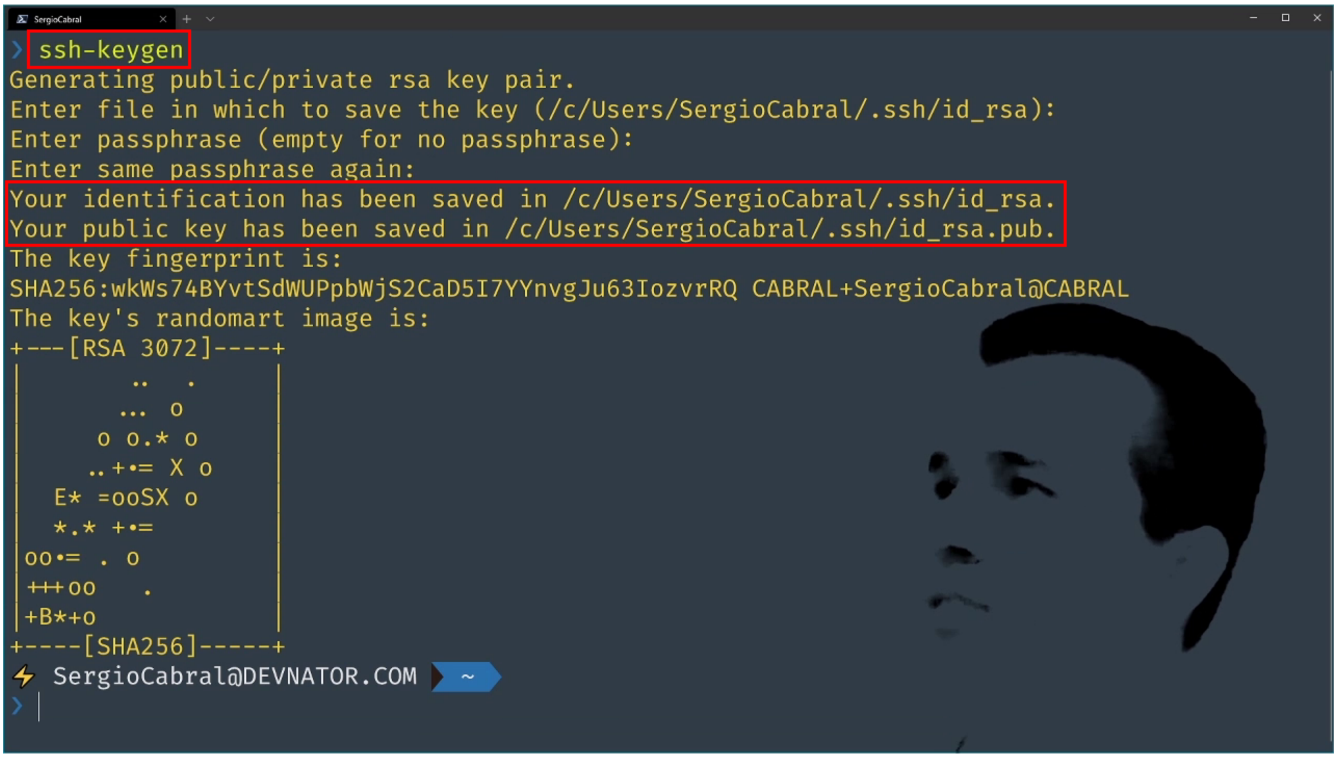
*Figure 1. Confirmation on first access to remote computer*

> By doing this, the public computer's encryption key is stored in the `~/.ssh/know_hosts` file at the local computer.
>
> If this file is deleted or the line containing the remote computer's public key is removed, then the confirmation message will be displayed again.

Then you enter your password, and that's it. Connected!

# 2. Connection with encryption keys

Another form of authentication is to let the target computer know the source computer. In this case we create a pair of encryption keys on the source computer with the command `ssh-keygen`. Run and press `Enter` until done, as shown in *Figure 2*.

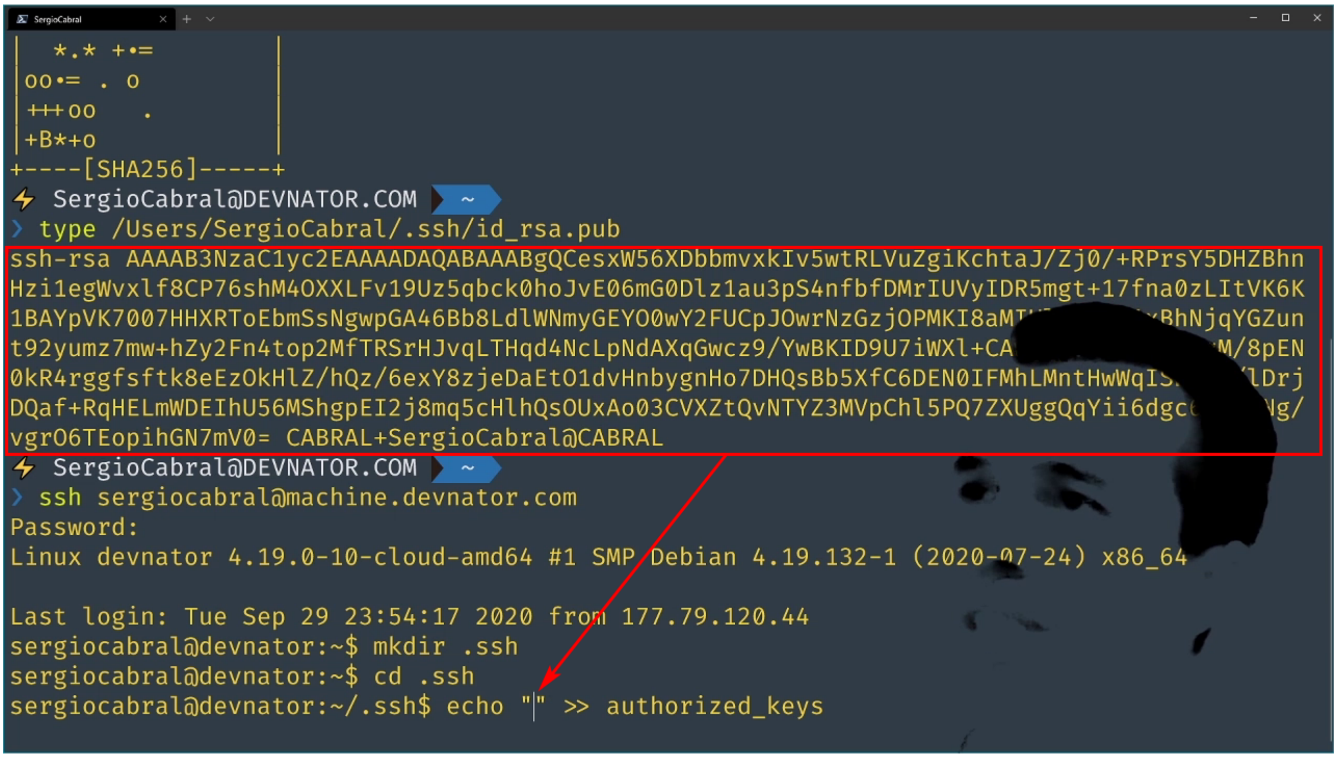


```
> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/SergioCabral/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/SergioCabral/.ssh/id_rsa.
Your public key has been saved in /c/Users/SergioCabral/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:wkWs74BYvtSdWUPpbWjS2CaD5I7YYnvgJu63IozvrRQ CABRAL+SergioCabral@CABRAL
The key's randomart image is:
+---[RSA 3072]----+
|        ..  .    |
|       ...  o    |
|     o o.* o     |
|    ..+•= X o    |
|  E* =ooSX o     |
|  *.* +•=        |
|oo•= . o         |
|+++oo   .        |
|+B*+o            |
+----[SHA256]-----+
⚡ SergioCabral@DEVNATOR.COM  ~
> |
```

*Figure 2. Standard execution of the "ssh-keygen" command*

If you run the `ssh-keygen` command again, by default (by pressing `Enter` until the end) it does not overwrite previously generated keys. But if you specify that you want to do this, you will lose any access to services that depended on those keys. It is an irreversible operation.

As also indicated in *Figure 2*, a pair of files are generated. We are interested in this demonstration in the public file `id_rsa.pub`. The private `id_rsa` file must be kept safe and never shared.



```
|   *.* +•=          |
|oo•= . o            |
| +++oo    .         |
|+B*+o               |
+----[SHA256]-----+
⚡ SergioCabral@DEVNATOR.COM  ~
› type /Users/SergioCabral/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCesxW56XDbbmvxkIv5wtRLVuZgiKchtaJ/Zj0/+RPrsY5DHZBhn
Hzi1egWvxlf8CP76shM4OXXLFv19Uz5qbck0hoJvE06mG0Dlz1au3pS4nfbfDMrIUVyIDR5mgt+17fna0zLItVK6K
1BAYpVK7007HHXRToEbmSsNgwpGA46Bb8LdlWNmyGEYO0wY2FUCpJOwrNzGzjOPMKI8aMI[illegible]BhNjqYGZun
t92yumz7mw+hZy2Fn4top2MfTRSrHJvqLTHqd4NcLpNdAXqGwcz9/YwBKID9U7iWXl+CA[illegible]M/8pEN
0kR4rggfsftk8eEzOkHlZ/hQz/6exY8zjeDaEtO1dvHnbygnHo7DHQsBb5XfC6DEN0IFMhLMntHwWqIS[illegible]lDrj
DQaf+RqHELmWDEIhU56MShgpEI2j8mq5cHlhQsOUxAo03CVXZtQvNTYZ3MVpChl5PQ7ZXUggQqYii6dgc[illegible]Ng/
vgrO6TEopihGN7mV0= CABRAL+SergioCabral@CABRAL
⚡ SergioCabral@DEVNATOR.COM  ~
› ssh sergiocabral@machine.devnator.com
Password:
Linux devnator 4.19.0-10-cloud-amd64 #1 SMP Debian 4.19.132-1 (2020-07-24) x86_64

Last login: Tue Sep 29 23:54:17 2020 from 177.79.120.44
sergiocabral@devnator:~$ mkdir .ssh
sergiocabral@devnator:~$ cd .ssh
sergiocabral@devnator:~/.ssh$ echo "|" >> authorized_keys
```

*Figure 3. Content of the file "id_rsa.pub"*

Then, we send the contents of the file `id_rsa.pub`, as exemplified in *Figure 3*, to the destination computer, the server. This content must be added to the `~/.ssh/authorized_keys` file. If it does not exist, it must be created.

> As shown in *Figure 3*, the content of the public encryption key is short text that you can copy using the mouse and the combination of `type` or `cat` commands to display and `echo` to write.
>
> But if you are without a mouse, you may prefer to copy the file directly with the `scp` command and then add the key to the `authorized_keys` file:
>
> Command on the source computer:
>
> - `scp ~/.ssh/id_rsa.pub username@server-address:~/.ssh`
>
> Command on the destination computer:
>
> - `cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`

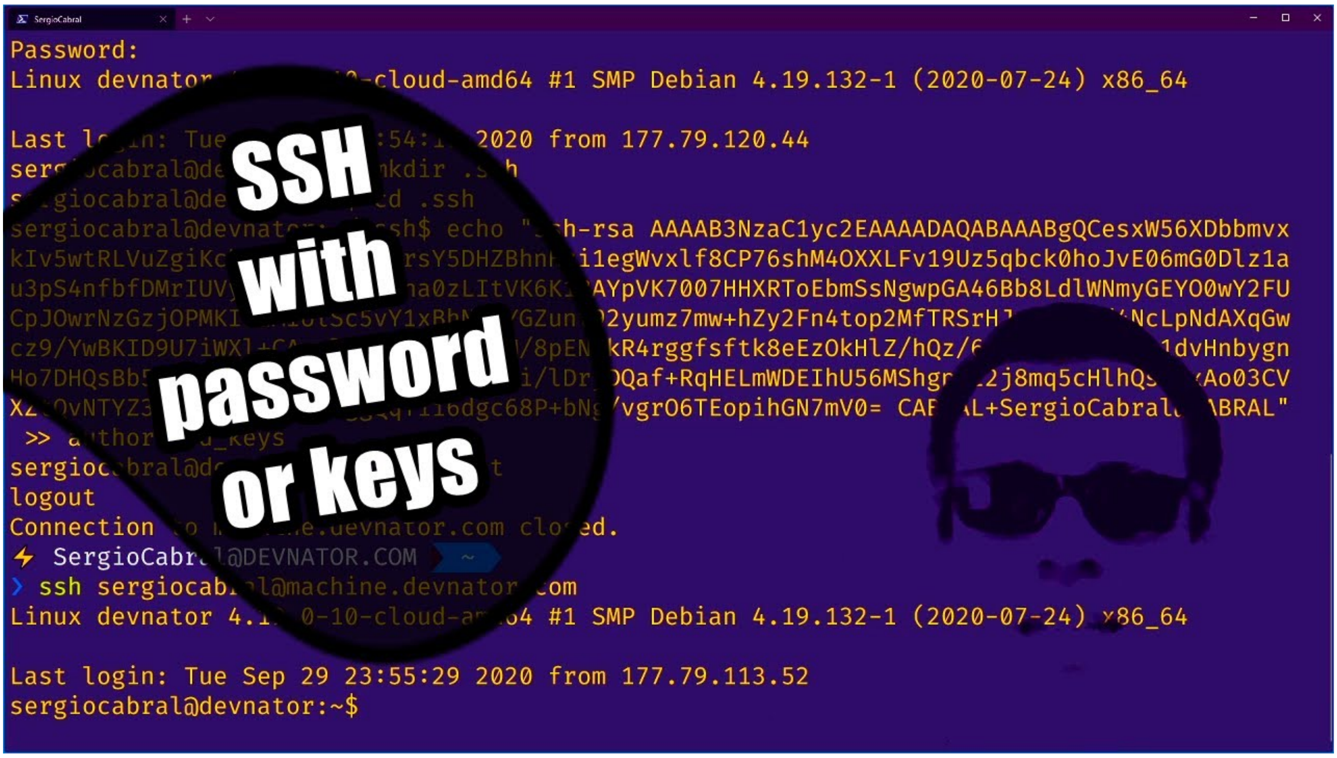
From now on, both your computer knows the server and the server knows your computer. Each computer has the public key of the other.

Make a new connection attempt and you're done. Connected using keys without having to enter authentication data.

# 3. Alternatives for SSH clients

| Name | License | Download |
|---|---|---|
| PuTTY | free; open-source | http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html |
| SuperPutty | free; open-source; based on PuTTY | https://github.com/jimradford/superputty |
| PuTTY Tray | free; open-source; based on PuTTY | https://puttytray.goeswhere.com/ |
| KiTTY | free; open-source; based on PuTTY | http://www.9bis.net/kitty/ |
| MobaXterm | free; paid Pro version available | http://mobaxterm.mobatek.net/ |
| SmarTTY | free | http://smartty.sysprogs.com/ |
| Dameware SSH client | free; paid options available | http://www.dameware.com/free-ssh-client-for-windows.aspx |
| mRemoteNG | free; open-source | http://www.mremoteng.org/ |
| Terminals | free; open-source | https://terminals.codeplex.com/ |
| Secure Shell App | free; Chrome Addon | https://chrome.google.com/webstore/detail/pnhechapfaindjhompbnflcldabbghjo |

# 4. Video demonstration



```
Password:
Linux devnator            0-cloud-amd64 #1 SMP Debian 4.19.132-1 (2020-07-24) x86_64

Last lo in: Tue             :54:1  2020 from 177.79.120.44
serg  cabral@de          mkdir .s h
s  giocabral@de           cd .ssh
sergiocabral@devnat      ssh$ echo "  h-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCesxW56XDbbmvx
kIv5wtRLVuZgiKc           rsY5DHZBhnH  i1egWvxlf8CP76shM4OXXLFv19Uz5qbck0hoJvE06mG0Dlz1a
u3pS4nfbfDMrIUVy          a0zLItVK6K  AYpVK7007HHXRToEbmSsNgwpGA46Bb8LdlWNmyGEYO0wY2FU
CpJOwrNzGzjOPMKI    Sc5vY1xBhN  /GZun  2yumz7mw+hZy2Fn4top2MfTRSrH          NcLpNdAXqGw
cz9/YwBKID9U7iWX1          /8pEN  kR4rggfsftk8eEzOkHlZ/hQz/6            1dvHnbygn
Ho7DHQsBb5              i/lDr  DQaf+RqHELmWDEIhU56MShgr  2j8mq5cHlhQs   Ao03CV
XZ  OvNTYZ3          16dgc68P+bNg /vgrO6TEopihGN7mV0= CAF  L+SergioCabral  BRAL"
 >> a thor  d_keys
sergioc bral@d                t
logout
Connection  o m    ne.devnator.com clo ed.
⚡ SergioCabr  l@DEVNATOR.COM  ~
❯ ssh sergiocab  l@machine.devnator com
Linux devnator 4.1  0-10-cloud-a  64 #1 SMP Debian 4.19.132-1 (2020-07-24) x86_64

Last login: Tue Sep 29 23:55:29 2020 from 177.79.113.52
sergiocabral@devnator:~$
```

*https://youtu.be/KeF9I7zMMMw*

Hasta la vista.