Remote Terminal

SSH – or Secure Shell – allows us to connect to our server over our local network to run commands remotely.

This is very helpful for running an always-on server that doesn't have a display attached, often called a "headless" system. During the Debian install process, we opted to enable SSH by support.

When connecting to a new host, SSH will create a digital fingerprint for it. When we connect to this server in the future, it will compare the server's fingerprint to the one currently on file. In the event they don't match in the future, SSH will warn us that there is a mismatch and someone might be impersonating your server.

There are many different ways to connect to your server using SSH. We will highlight some common methods for connecting to our server and controlling it remotely.

Linux

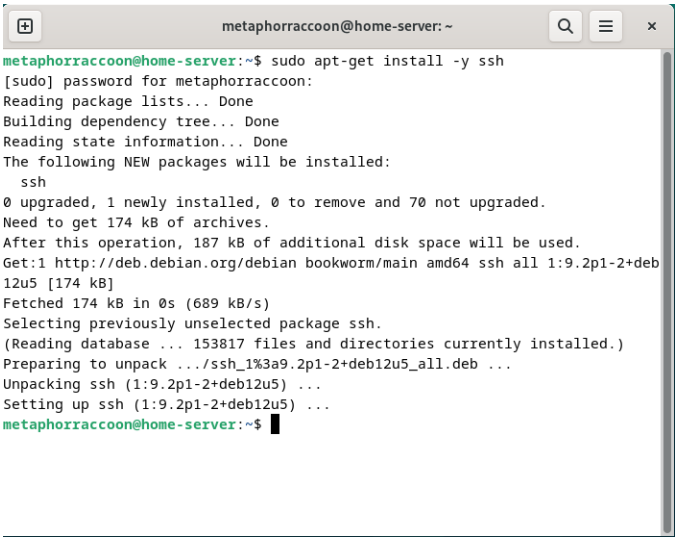
You can use the terminal available on most Distributions of Linux – such as Debian, Ubuntu, and Linux Mint – to remotely connect to your server.

> You will need to know your username set during the installation process and the IP Address you set during network configuration.

You can open a terminal and ensure that SSH is installed by running the command:

```
sudo apt-get install -y ssh
```



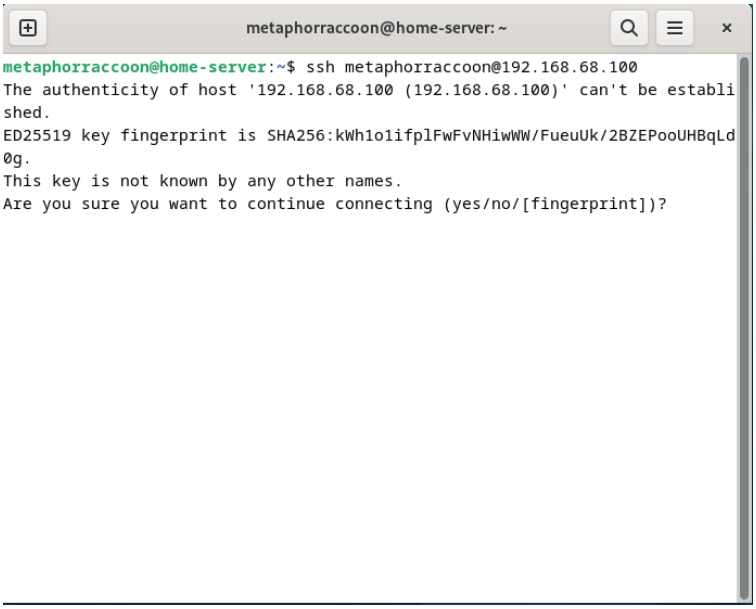


Once we are sure this is installed, we can run connect to our server through the terminal using the *ssh* command. We will be connecting to our user account what is at our server's IP address:

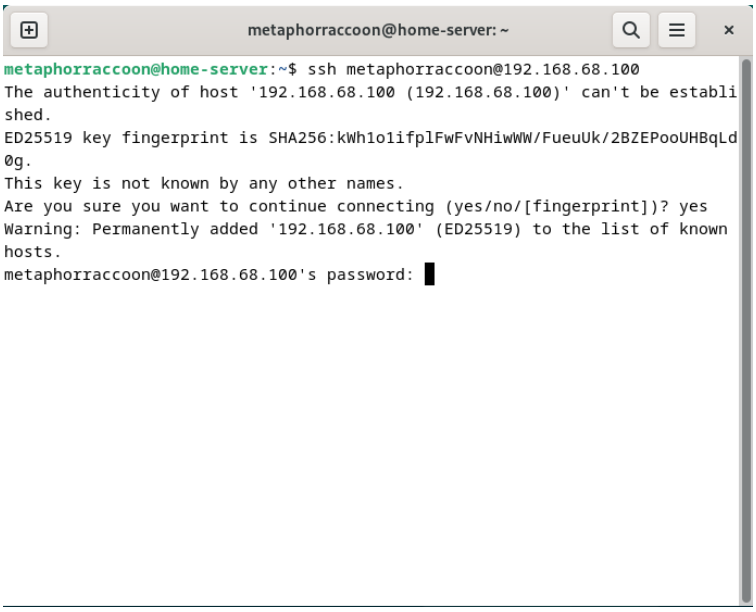
```
ssh username@192.168.68.100
```



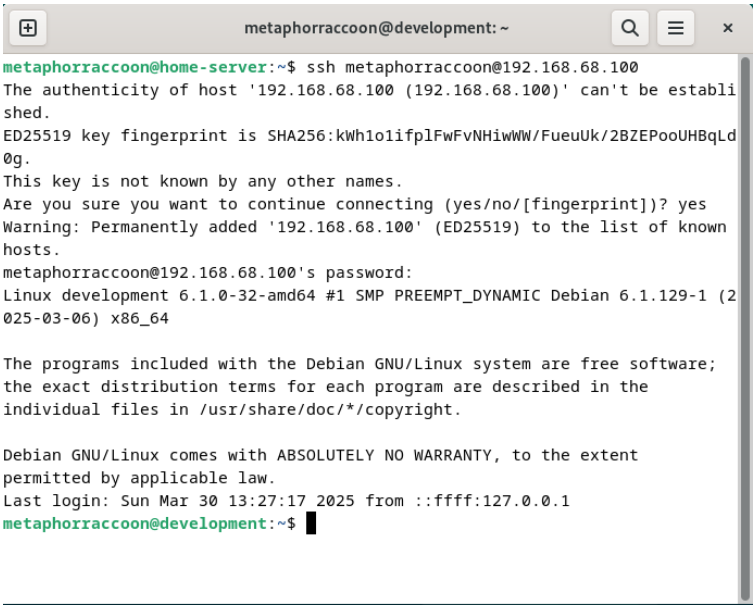
You will be alerted that the authenticity of the host server could not be verified. This happens because we have never connected to the server before and it can't ensure the server's fingerprint.



We can type 'yes' then hit 'enter' to accept the newly generated host fingerprint and remember it for future use.



Now, you will be prompted for your password. After hitting 'enter', you will be connected to your server and can run commands.



You can disconnect from the SSH connection with the following command:

```
exit
```

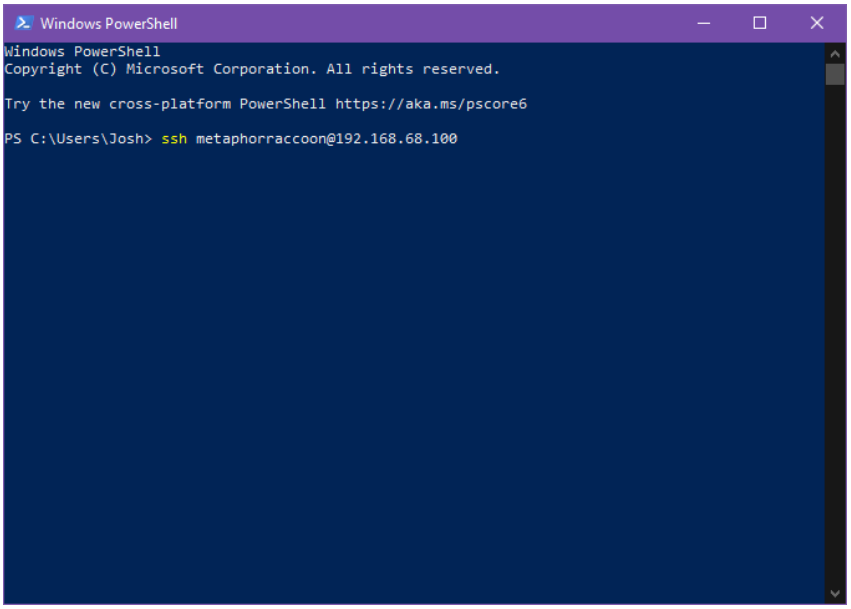
Windows

Windows 10 and Windows 11 come with a client pre-installed so you can use SSH to connect to your server remotely.

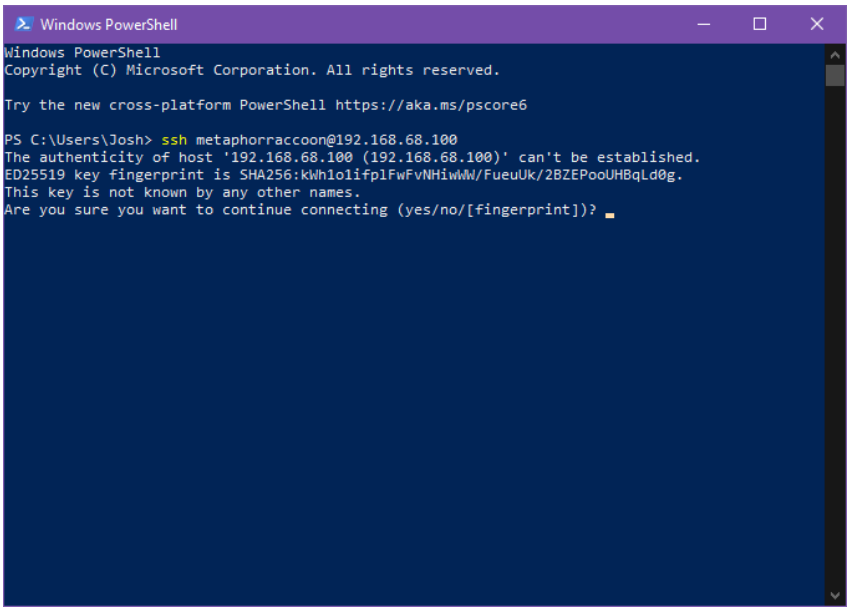
> You will need to know your username set during the installation process and the IP Address you set during network configuration.

You can open a terminal or PowerShell and run the following command:

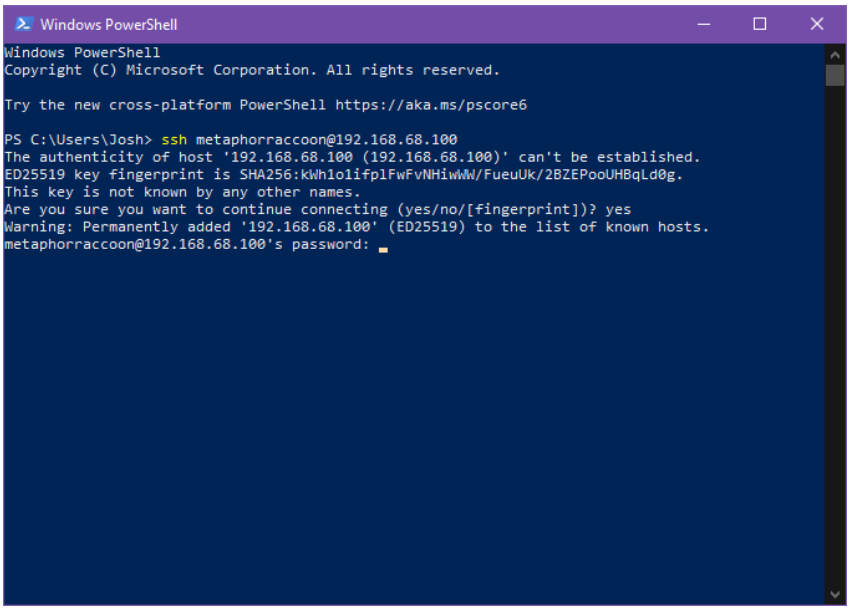
```
ssh username@192.168.68.100
```



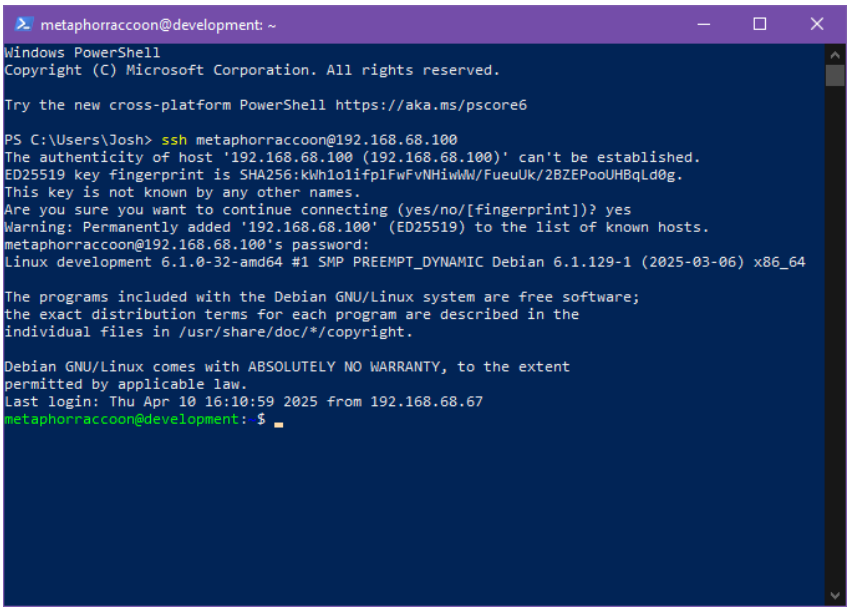
You will be alerted that the authenticity of the host server could not be verified. This happens because we have never connected to the server before and it can't ensure the server's fingerprint.



We can type 'yes' then hit 'enter' to accept the newly generated host fingerprint and remember it for future use.



Now, you will be prompted for your password. After hitting 'enter', you will be connected to your server and can run commands.



You can disconnect from the SSH connection with the following command:

```
exit
```

MacOS

MacOS, and the legacy OS X, come with SSH already installed so you can remotely connect to your server.

>

> You will need to know yourusername set during the installation process and the IP Address you set during network configuration.

We can connect to our server through the terminal using the *ssh* command. We will be connecting to our user account what is at our server's IP address:

```
ssh username@192.168.68.100
```

You will be alerted that the authenticity of the host server could not be verified. This happens because we have never connected to the server before and it can't ensure the server's fingerprint.

We can type 'yes' then hit 'enter' to accept the newly generated host fingerprint and remember it for future use.

Now, you will be prompted for your password. After hitting 'enter', you will be connected to your server and can run commands.

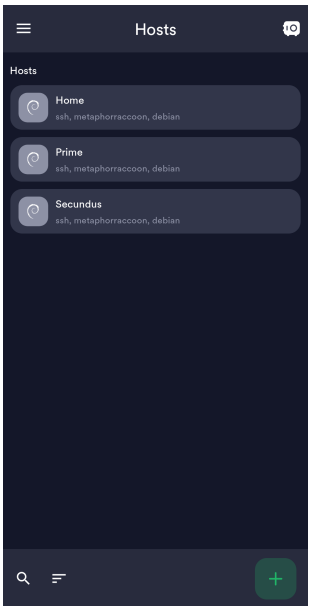
You can disconnect from the SSH connection with the following command:

```
exit
```

Android & iOS

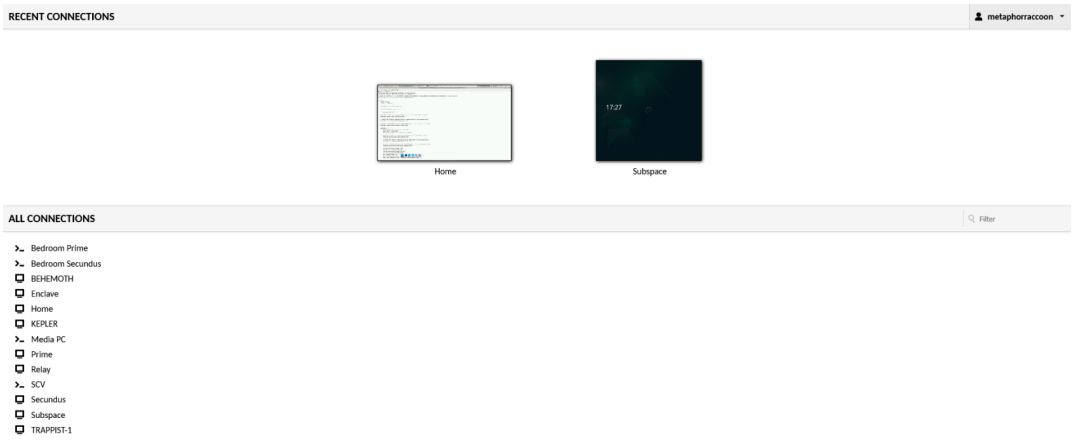
There are apps available through the Google Play Store and Apple App Store that allow you to remotely connect to your server using SSH.

We recommend Termius, a freeware application for Android 7 and iOS 16 or newer. You can create a profile for your server with the credentials pre-saved for quick connection. As a premium paid feature, you can also sync these hosts between the mobile and desktop clients.



Web Access

SSH is a great option for connecting to your server over your local network, but is not available through the world wide web. The easiest way to accomplish this is to use our home server to host Guacamole, a web application that facilitates remote connection to RDP, VNC and SSH through your web browser.



Revision #12
Created 15 February 2025 04:36:24 by metaphorraccoon
Updated 12 May 2025 08:51:40 by metaphorraccoon