How to manage remote Linux server using SSH

Managing the server is still a necessary and sometimes heavy task. Fortunately, Secure Shell (SSH) is available - a network protocol that allows services to run on an unsecured network.

Managing the server is still a necessary and sometimes heavy task. It is especially difficult for remote servers or headless servers. Thankfully, Secure Shell (SSH) is available. Secure Shell is a network protocol that allows services to run on an unsecured network.

SSH boasts a lot of functions. One of them is to manage the remote server. Let's learn how to remotely manage Linux server via SSH, including connection to software installation and file transfer, via the following article.

How to remotely manage a Linux server using SSH

What is SSH?

SSH stands for Secure Shell. It is an encrypted network protocol. For more details, please refer to the article: Understanding SSH.

Set up server to accept SSH

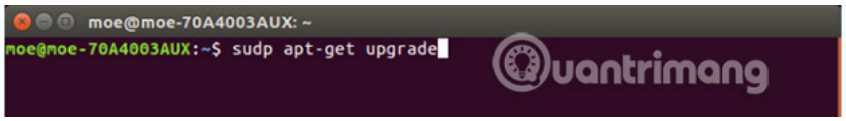
Before you start managing your Linux server via SSH, you will have to set up your server, so that it allows SSH connections. Suppose we have a dedicated headless Plex media server. Since there are no screens and peripherals, we will use SSH to manage the server. When installing or updating software and transferring files, simply install SSH to the server. The exact setting is a ThinkServer TS140 running Ubuntu 16.04 LTS. Depending on your hardware and Linux distribution, setting up for SSH may vary slightly.

To set up your Linux server to accept incoming connections, you will need to install the remote login tool for the SSH protocol. One of the most popular tools is OpenSSH. On Debian-based distributions, OpenSSH is available through the main repositories. Open a new terminal (**Ctrl + Alt + T**) and enter the following command:

```
sudo apt-get update
```

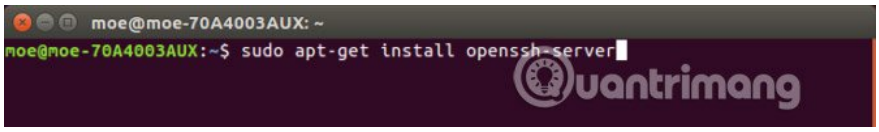
This performs updates and ensures you have the latest archives. Before proceeding to install OpenSSH, check all updates. In the terminal, run the command:

```
sudo apt-get upgrade
```



Once you have updated and upgraded, open a new command line and enter:

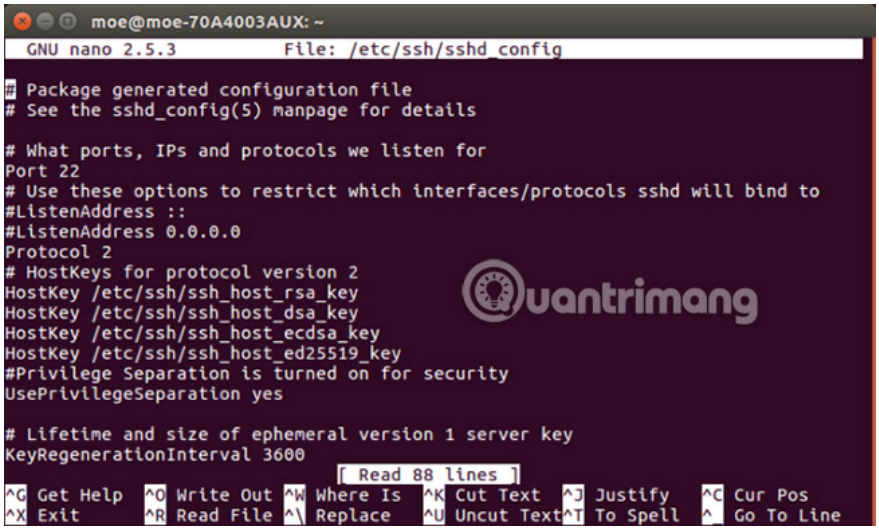
```
sudo apt-get install openssh-server
```



Configure SSH settings for the server

After OpenSSH has been installed on the server side, you can edit basic configuration information. Open a new terminal and enter the following string to open the SSH configuration file:

```
sudo nano / etc / ssh / sshd_config
```

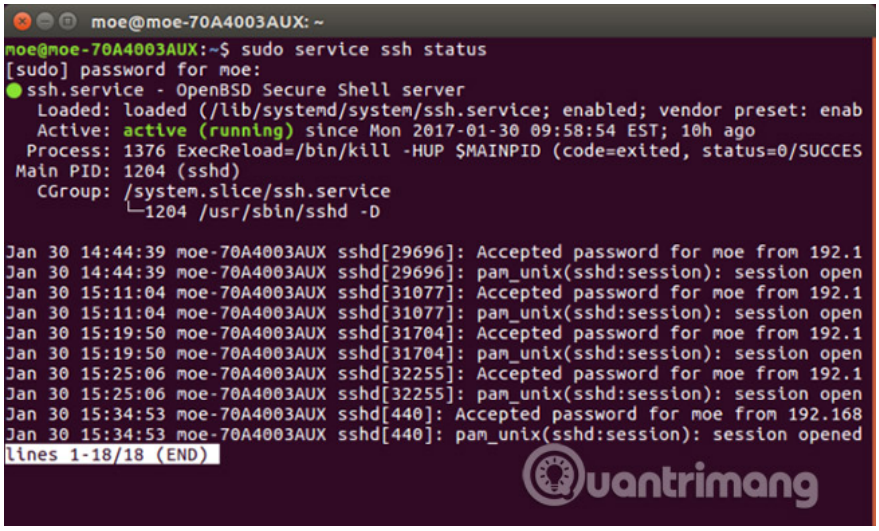


Here you can specify different settings. By default, your SSH server will work on port 22. So, for example, you can change from port 22 to the port of your choice. In addition, you can enhance security by entering the maximum login number. In **Port,** find the **MaxAuthTries** line **.** You can enter any number here. So, to choose the maximum number of logins to four, enter:

```
MaxAuthTries 4
```

After installing OpenSSH, the SSH server will run. But to test, just open a terminal and run:

```
sudo service ssh status
```



This brings up a message that SSH has been activated. To start SSH, open a command line and enter:

```
sudo service ssh start
```

And to prevent SSH from running, enter:

```
sudo service ssh stop
```

Remote access to a Linux server via SSH

Now that SSH is installed and running, you can connect remotely via it. If you are logged into the remote Linux server, you will need the server's IP address. It will have the **192.168.0.x** format. After you have the IP address of the Linux server, you also need to have a login facility via SSH from another computer. There are several methods for remote login using SSH.

Use SSH on Unix-based operating systems

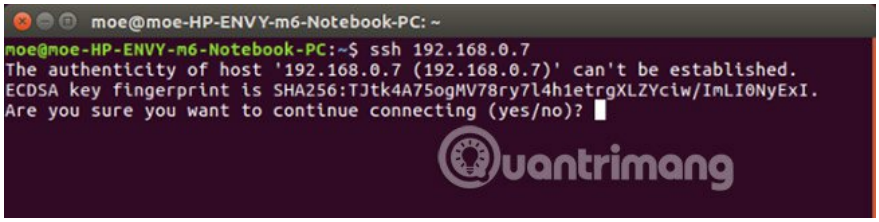
If you are using a Unix-based system like Linux, macOS or FreeBSD, SSH is available in the command line. In a terminal, run:

```
ssh [remote host]
```

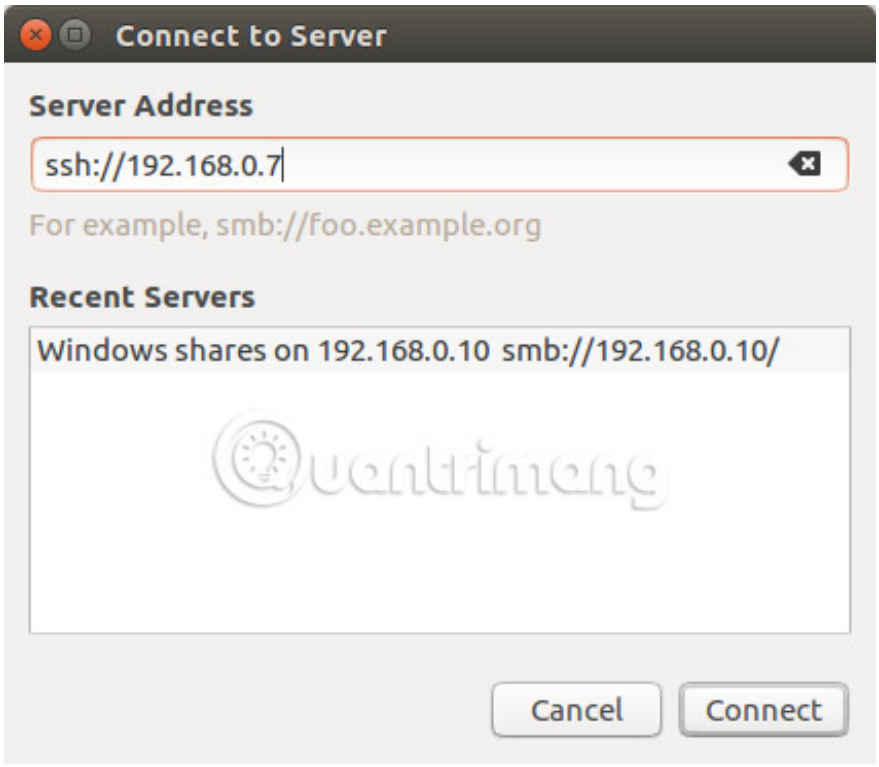
In it, **[remote host]** is the IP address you are accessing. If your username is different from the remote system, you can specify the correct username by typing:

```
ssh [remote username] @ [remote host]
```

After entering this information, you will be asked if you want to continue connecting. You will then be prompted to enter your username and password.



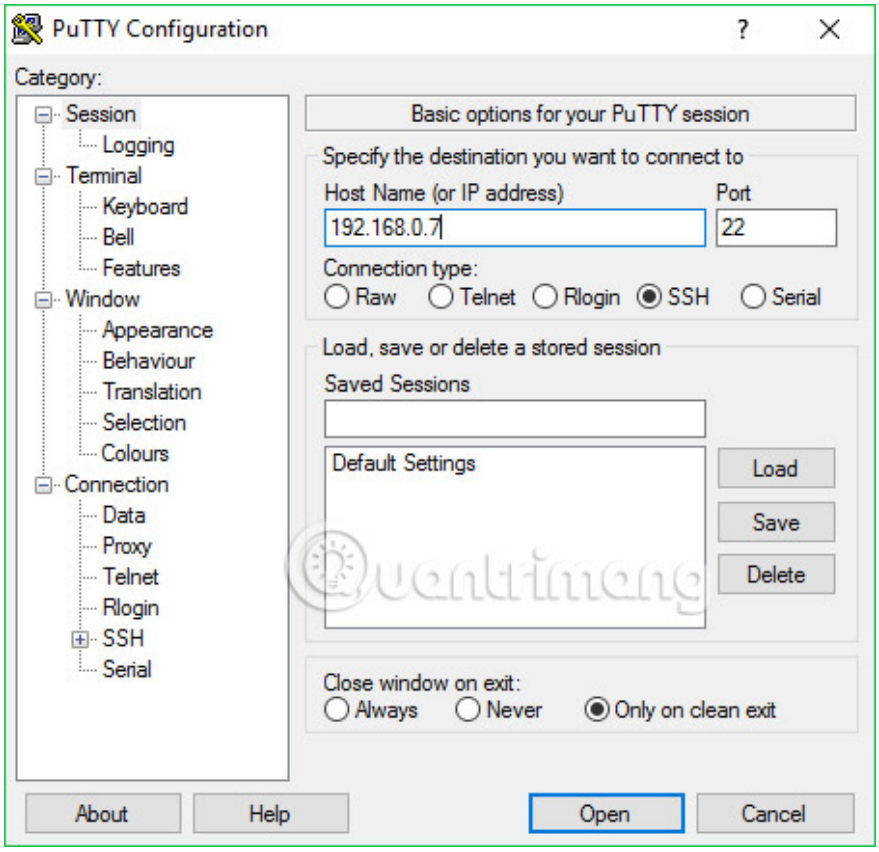
Also, if you want to skip the entire command line, you can log in to your Linux server over the network. On a Linux machine, navigate to the **Connect to Server** and enter **ssh: // [IP address].** You will be asked to provide your username and password.



The main advantage of this method is that you will have complete graphical directory navigation. This is called **SSH File Transfer Protocol** (SFTP) or SFTP. This makes file transfer much easier. Because Linux server in the example is a dedicated Plex server, unless the update is done, SFTP is often used.

SSH with PuTTY

If you are using a PC or Mac, you will need a SSH client. PuTTY is probably the most famous SSH client. Install PuTTY on your MacOS or Linux computer. Open **PuTTY** and see the **Session** section **.** In the box labeled **Host Name,** enter your IP address. Be sure to specify the correct port. If you use the default, leave this value to **22.**



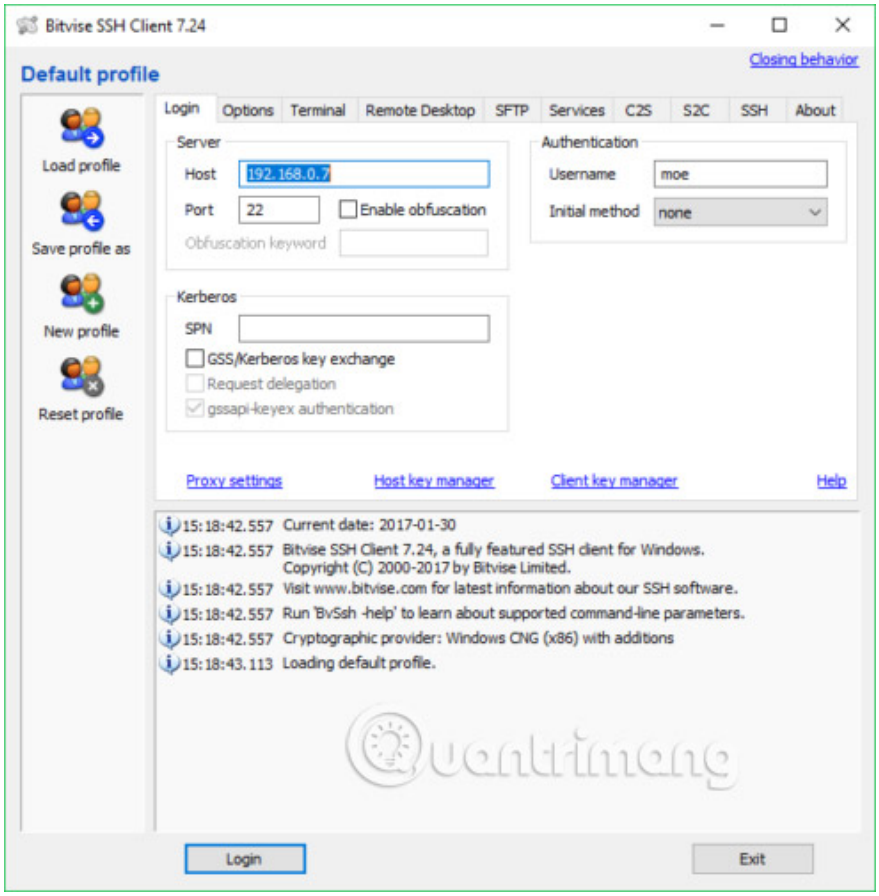
You should now see a terminal with a login prompt. Enter the username for Linux server here.

After entering your username, you will be prompted to enter your password, if you have protected your Linux server with a password. Enter that information and you will see the welcome message, the same information about your system and the command line, just like you would see on your Linux server.

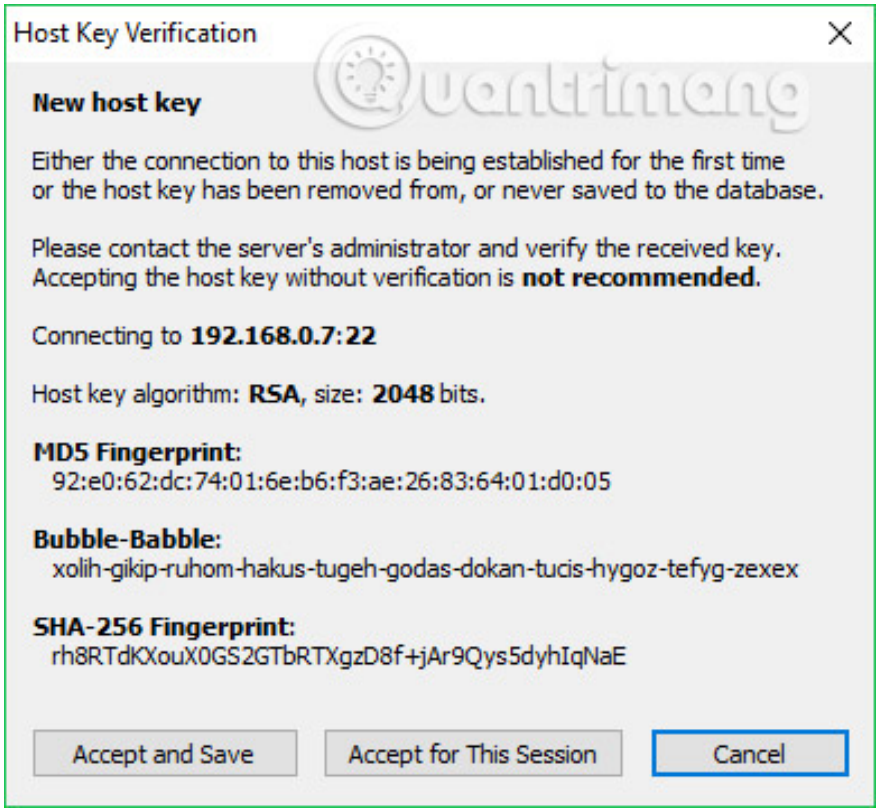
Alternative SSH clients

Although PuTTY is still the most popular SSH client, there are many alternatives. The first choice is Bitvise (Windows only). The reason is that it not only includes the command line interface to remotely manage Linux server via SSH, but also SFTP capability. Therefore, it is perfect for both file transfer and general management. When you just need to update or restart, you can use Bitvise SSH Client to access the command line. But for file transfers, use the graphical interface.

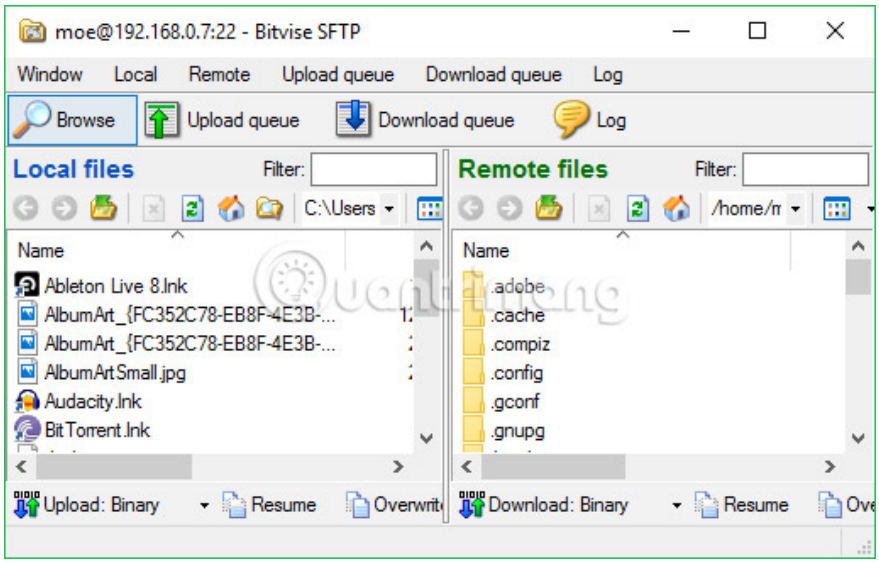
Like PuTTY or start SSH via the command line on Linux, you need to enter your IP address, username and password.



You will receive a prompt asking if you want to accept the session. You can accept only for that session or save for future use. Then you will be asked to enter the password of the Linux server that you are managing remotely.



Then, Bitvise will open both the command line and the SSH graphical window.



SFTP window helps manage really simple file transfer, plus traditional SSH command line for many functions.

How to manage a remote Linux server with SSH

SSH is configured on both the server and the device you will use to manage your server. So what can you really do? The answer is anything you can do with the Linux command line. Some common tasks that you can do first are installing software, updating, restarting, copying files, and even running GUI applications. For example, you

can copy the file using the following commands:

```
scp [remote host]: [local file] [destination directory]
```

```
scp [local file] [remote host]: [destination directory]
```

Installing the software only means entering the correct command. In this example, when installing Plex on headless TS140 via SSH, just enter:

```
sudo apt-get install plexmedia server -y
```

Similarly, to create a WordPress server, you will just need to follow the installation details but through an SSH command line.

What you can do depends on the server and its purpose. For example, when running a dedicated Plex headless server, SSH and SFTP are mostly used to transfer files with occasional software updates. You can also use SSH to access log files and run benchmarks to check CPU performance. If you are running a web server, you may want to back up your site using the SSH command line.

If you are running a Linux server, SSH is a great way to manage it remotely. You retain full command line control and can even run GUI applications and perform file transfers. There is almost limitless possibility for what you can do when managing remote Linux servers via SSH. You can set up file server, media server, Linux game server and more. You can even manage a headless Raspberry Pi server using SSH. Moreover, there are many cross-platform SSH client programs.

How are you using SSH and what server type are you managing? Let us know about that in the comment section below!

See more:

1. How to activate and use SSH commands on Windows 10
2. Protect Internet connection via SSH
3. Install SSH on the Router for secure web access anywhere