# SSH

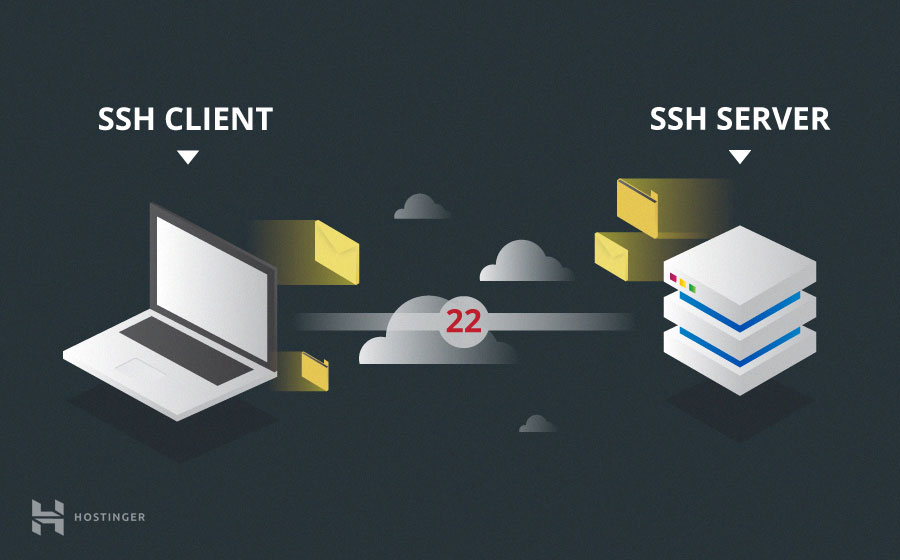
## What Is SSH?

SSH (Secure Shell) is a network communication protocol that enables two or more computers to communicate and share data with each other over the Internet.

## How Does SSH Work?

SSH works by making use of a **client-server model** to allow for authentication of two remote systems and encryption of the data that passes between them.

By default, SSH operates on port 22. The host (server) listens on port 22 (or any other) for incoming connections. It organizes the secure connection by authenticating the client and opening the correct shell environment if the verification is successful.

**[](https://www.hostinger.com/tutorials/wp-content/uploads/sites/2/2017/07/ssh-client-and-server.jpg)**

The client must start the SSH connection by initiating the TCP handshake with the server, ensuring a secured symmetric connection, verifying whether the identity displayed by the server match previous records (typically recorded in an RSA key store file), and presenting the required user credentials to authenticate the connection.

There are two stages to establishing a connection:

* First, both the systems must agree upon encryption standards to protect future communications.
* Second, the user must authenticate themselves. If the credentials match, then the user is granted access.

## SSH Command

The SSH command consists of 4 distinct parts:

$ ssh [options] <user-name>@<host-name-or-host-IP>

Where:

* user-name: The account you want to access. For example, you may want to access the root user, which is basically synonymous for system administrator with complete rights to modify anything on the system.
* host-name-or-host-IP: The computer you want to access. This can be an IP Address (e.g., 244.235.23.19) or a domain name (e.g., [www.xyzdomain.com](http://www.xyzdomain.com)).
* options: Additional settings you add to the command. For example, option for port (-p <port-id>) is used to select the TCP port.

Once you run the command, you will be prompted to enter the password for the requested account. If your password is correct, you will be greeted with a remote terminal window.

## Tips

### SSH between Windows Host and Linux Guest which run on VMWare Player / Virtual Box on another computer

It's very easy to creating a SSH connection **between two computers**, or **between a host and a virtual machine running on this host**. Just execute the SSH command and done!

However, thing becomes **complicated when SSH-ing between a host and virtual machine guest which run on another computer (we can call it a 'remote VM guest')**.

In this example, we'll connect from a Windows host to a Linux guest which run on VMWare Player / Virtual Box on another computer via SSH.

Note: We'll use NAT network instead of Bridge.

**I - VMWare Player / VMWare Workstation**

**1. Modify NAT config**

Open and edit C:\ProgramData\VMware\vmnetnat.conf with Administrator permission as followings:

In [incomingtcp] section, add a line:

2244 = <vm-guest-linux-ip>:22

Note: Besides port 2244, you can choose any port you want.

After saving the file, run Windows command to restart VMWare NAT service (so the config can be reloaded) by running following commands:

$ net stop "VMWare NAT Service"

$ net start "VMWare NAT Service"

Note: If these commands won't work, you can restart the service with *Services* app on Windows.

To make sure things work, try SSH into your Linux guest from your Windows host (though it's not our purpose) with the command:

$ ssh -p 2244 <linux-username>@localhost

**2. Modify Windows Firewall config**

Trying to SSH to the guest from another host won't be allowed because of the Windows Firewall rules.

To make it possible, you have two ways:

1. Disable Windows Firewall
2. Add new rule by:

Open Control Panel > Windows Defender Firewall > Advanced settings. New an 'Inbound Rules'.

Select 'Port' as rule type.

Select TCP and set the port '2244'

Save

**3. SSH to your Linux VM guest from remote Windows host**

Simply run this command:

$ ssh -p 2244 <linux-username>@<windows-hostname-or-ip>

Note: windows-hostname-or-ip is of the local Windows host which the virtual machine run on, NOT of the remote host we're typing on.

**II - Virtual Box**

**1. Modify NAT config**

Settings > Network > Adapter[n] , click 'Port Forwarding'. Configure a NAT rule, for instance:

* Name: External SSH incoming
* protocol: TCP
* Host IP: Your Windows host public IP
* Host port: The port the remote SSH connects to (e.g., port 2244)
* Guest IP: Your virtual machine Linux guest IP
* Guest port: it’s TCP 22 for SSH

**2. SSH to your Linux VM guest from another Windows host**

Simply run this command:

$ ssh -p 2244 <linux-username>@<windows-hostname-or-ip>

Note: windows-hostname-or-ip is of the local Windows host which the virtual machine run on, NOT of the remote host we're typing on.

Ref: <https://slmeng.medium.com/how-to-ssh-into-vmware-player-virtual-box-guest-linux-os-remotely-host-os-is-window-10-40cb348c996f>

### Open GUI app with SSH

**Warning:**

This method only works with **Linux or Mac GUI app**. This means it cannot help you open a Windows GUI app (e.g., Notepad) via SSH-ing to a Windows guest.

Reason [here](https://superuser.com/a/1297214) and [here](https://superuser.com/a/1168000).

SSH-ing with GUI sounds hard to achieve, but it's actually easy and there are many ways to do. The simplest could be:

Step 1: Install a X server. For example, [Xming](https://sourceforge.net/projects/xming/) on Windows.

Step 2: Run the X server. Usually, it will run on background.

Step 3: SSH to the Linux/Mac guest machine with the X forwarding option:

* If using command line – run: ssh -X <guest-username>:<guest-hostname-or-ip>
* If using Putty: From the main windows, go to Connection > SSH > X11, and tick on Enable X11 forwarding. Then make a SSH connection to the guest machine.

Step 4: Once you accessed the guest, try opening an GUI app (e.g., gedit on Linux). You'll see the app opened and displayed with a full GUI window.

