You probably know what a server is and that you can access it from anywhere just by typing commands on another machine. SSH is a common and secure method of connecting with a server. Let's explore how we can set up an SSH server and access it by looking at an example where we create an SSH connection between devices; like your desktop computer and laptop, another desktop, or a smartphone. We will also look at some practical applications of SSH, like copying files over an SSH connection.

What is SSH and how to set SSH server and client?

SSH stands for secure shell, secure because all the information is encrypted. There are a few programs for SSH connection, the most popular is OpenSSH made by the OpenBSD project. This tool is included in most Unix-like systems and even Windows. Almost any package manager has it and in many distributions, it is installed by default.

SSH uses a client-server architecture, which means for connecting two computers SSH must be installed on both of them. To set up a server you need to install its server component and to get into the server you need an installed client on the machine you are using.

First, let's install the server component of SSH on our home desktop. If you have Debian-based distributions use this command

```
$ sudo apt install openssh-server
```

Commands on other distributions will be similar, also the package name can slightly differ.

To enable and start SSH-server on systems with SystemD run

```
$ sudo systemctl enable ssh  # enable ssh such that it will start every time the system boots up

$ sudo systemctl start ssh   # start it now


$ sudo systemctl status ssh  # check its status
```

In systems with other initialization systems, commands look similar, the general idea — is to first enable and then start.

You probably also need to configure a firewall. Uncomplicated firewall (UFW) will be dealt with in this way

```
$ sudo ufw allow OpenSSH
```

On your second device, you need to have an SSH-client program. To install it with APT, run

```
$ sudo apt install openssh-client
```

How to get into a server with SSH

Now you have your SSH server and client, so you can try to make a connection. For this, type the

`ssh` command, then your username on that server and the IP address of the server. Here is an example of the `ssh` command:

```
$ ssh mdukuzi@104.21.85.36
```

If you do not specify a username you will be asked for the root password.

Also, you can write this IP address and username in the configuration file which you create in this address *~/.ssh/config*. For example:

```
Host my-ssh-server
 hostname 104.21.85.36

 user mdukuzi
```

Here we gave the server the name `my-ssh-server`, but you can choose any other name for the host. With this configuration file, you can use the following command to access the server.

```
$ ssh my-ssh-server
```

After you successfully typed the right password you should be asked for confirmation, as you log in for the first time. Your IP address and key fingerprint will be added to the file *~/.ssh/known_hosts*.

Now you can work on the server! To close the connection just type the command: `exit`.

Secure way There is an alternative and more secure way of connecting — using private and public keys.

The first step is to generate a public and private key pair. When you're not yet logged in to the server type this command

```
$ ssh-keygen
```

You can use this command to generate keys for other websites such as GitHub

After that, you can specify the location and name for your key or leave default settings that's to say *~/.ssh* directory and *id_rsa* name. Also, you will be asked to choose a passphrase, if you don't want to choose don't type anything.

Finally, you should have 2 keys in the chosen directory: private and public, the one with *.pub* extension.

Then add your public key to the server's directory *~/.ssh* to the file named *authorized_keys*. Easier to do it by using

`ssh-copy-id` command which will create a *.ssh* directory and *authorized_keys* file automatically if they didn't exist.

```
$ ssh-copy-id -i ~/.ssh/id-key.pub my-host-server
```

Now when you log in to the server you will be asked for a passphrase to use your keys or you will not if you skipped the step with a passphrase before.

Usually this should work, however, if `ssh` is still asking for your password not the passphrase for the private key then add this line to your SSH-config file

`IdentityFile id-rsa` here, `id-rsa` is the name of your private key

How to copy files from the server

Now let's do something through the SSH. For example, copy some files from the server and vice versa. For this, we will use the `scp` command. Remember you don't need to enter the SSH server with `ssh` command. To copy the file *manifest* from the server to our local machine you run`$ scp my-host-server:~/Documents/manifest ~/Project/`

After `scp`, we type the name of the server from the config file (or username@ip-address of the server), the `:` sign, the path to the file on the server space, and after that the path on the local machine.

When we want to copy files from the local machine, we type the path on the local machine and then the name of the server, `:` sign, and the path on the server.

Conclusion

Let's highlight the main points we've discussed. Now you know how to:

- set up an SSH server by installing `openssh-server` program;
- connect with the server using `ssh` command;
- set up passwordless authentication using commands
- `ssh-keygen` and `ssh-copy-id` transfer files using `scp` command.

This is only a basic overview of SSH but it's enough for you to start working with a server from any place through the internet.

Public key

Choose the right command to add your public key to the server's directory *.ssh*.

```
ssh-copy-id -i ~/.ssh/id-key.pub ssh-server
```

Which command do you need for generating public and private key pair?
`Ssh-keygen`

Let's say that you have a working SSH-server at home. You went traveling and forgot to copy the file warp-engine to the ~/Documents directory on your laptop. Which command will help you get that file?
`scp home-server:~/Documents/warp-engine ~/Documents/`

What does SSH stands for?
Secure Shell

Type the command to log in to the server for the user `mole` and IP address of the server `104.5.0.6`.
ssh mole@104.5.0.6

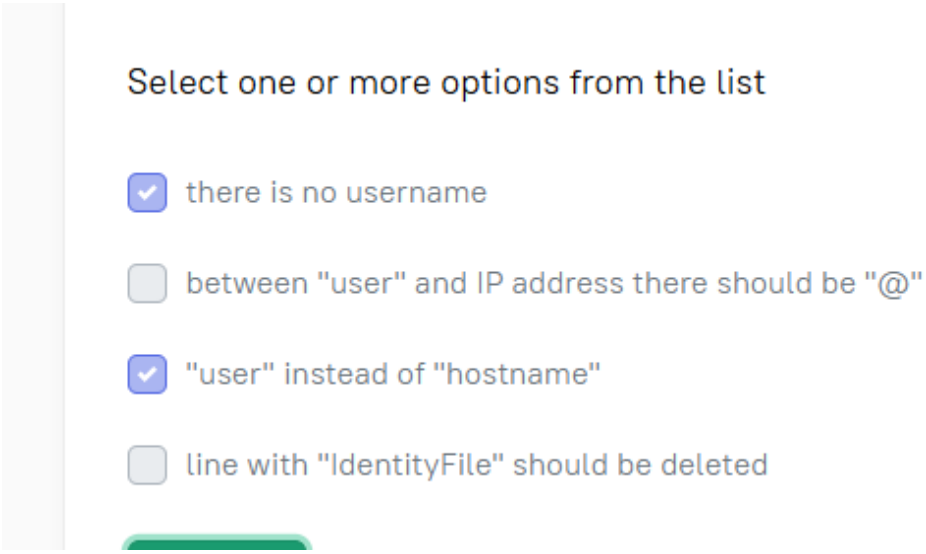
A new communication station was built on the moon. But the people there don't know how to set up an SSH-server on the station's computer. You have been tasked with the mission of setting up an SSH-server, so that people from Earth can use the station. The operating system on the computer uses APT package manager, SystemD as system manager and uncomplicated firewall (ufw). Sort the commands to set up an SSH-server there.

```
sudo apt install OpenSSH-server
sudo systemctl enable ssh
sudo systemctl start ssh
sudo ufw allow OpenSSH
```

Find mistakes in this configuration file for SSH-connection in *~/.ssh/config*

```
Host happy-server
 user 104.21.85.36

 IdentityFile id-rsa
```



Corrected config:

```
Host happy-server
  HostName 104.21.85.36
  User myusername
  IdentityFile ~/.ssh/id_rsa
```