# How to Set Up and Use SSH in Linux

Secure Shell, commonly known as SSH, is a network protocol for establishing secure connections between remote clients and servers. It is designed to allow users to securely log on to a variety of computers remotely over the network.

If you're using Linux, you've probably heard of a tool called SSH. Secure Shell, commonly known as SSH, is a network protocol for establishing secure connections between remote clients and servers. It is designed to allow users to securely log on to a variety of computers remotely over the network. Here, the article will show you how to set up and copy SSH keys to the server easily.

## SSH Settings

To get started, you must install an SSH server. You can find and install the **openssh-server** package in the **Software Center** or the package manager. Alternatively, if you're on a server (or just prefer to use Terminal), open a Terminal and enter the following command:

### Ubuntu/Debian

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

### Fedora/CentOS/REHL

```
sudo dnf install openssh-server
```

## Enable SSH in Linux

After the OpenSSH server has been installed on your machine, you need to start and activate **systemd** . To do that, simply type the following command into Terminal:

```
sudo systemctl enable --now ssh
```

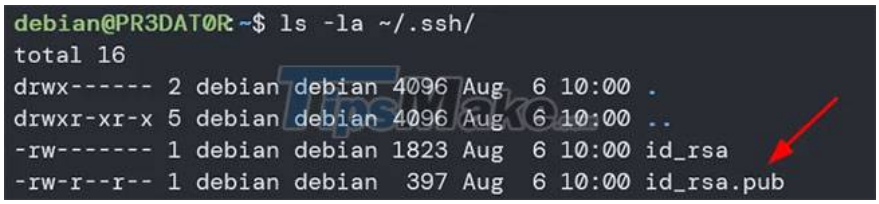
## Generate SSH key

Once the openssh server is installed, you can start generating SSH key pairs. Before continuing, make sure you don't have any existing key pairs, as this process will overwrite an existing key pair.

To check if you currently have a key pair, use the command:

```
ls -la ~/.ssh
```

If you currently have a key pair, the above command will display the files **'id_rsa'** and **'id_rsa.pub'** .



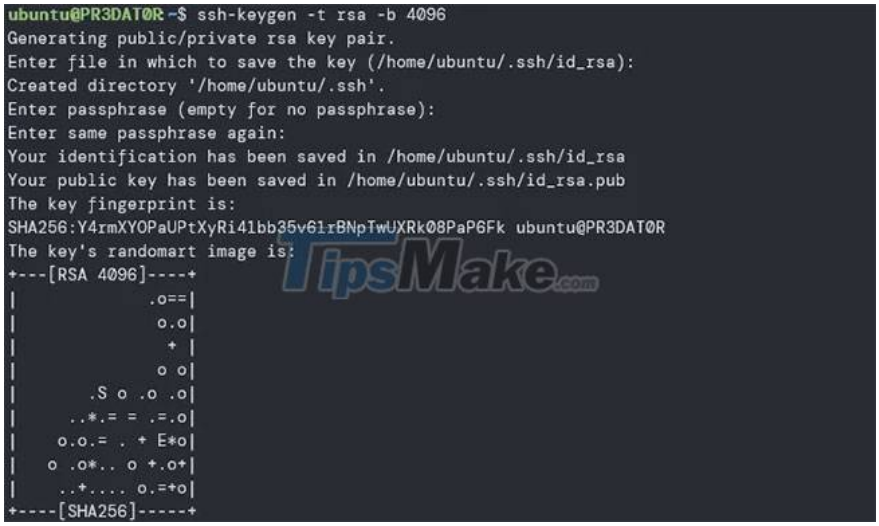

After verifying that you currently do not have an SSH key pair, you can proceed to generate a new one. If not, back up the old keys to avoid losing them.

To generate a new key, use the command:

```
ssh-keygen -t rsa -b 4096
```

The above command calls the **ssh-keygen** utility to interactively generate SSH key pairs. Use the **-t** option to specify the type of key to be generated. In this case, the example will generate an RSA key.

The example also uses the -b option to specify the number of bits in the key. If you use an RSA key, the minimum bit size is 1024. If not specified, it will generate a key of 3072 bits.




You should use the default location to store SSH keys to avoid having to enter the path when connecting to SSH with keys.

If you don't want to encrypt your key with a passphrase, press **Enter** to skip.

# Copy the key to the remote server

Now, the article has created a new SSH key pair and now needs to upload it to the remote computer that you want to manage.

The most efficient way to do this is to use the **ssh-copy-id** command **.** Use the following command:

```
ssh-copy-id [email protected]_IP
```

If you are using a keyfile with its own filename, you can use the following command to specify the path to the keyfile.

```
ssh-copy-id -i ~/.ssh/id_rsa [email protected]_IP
```

If you are logging into the remote machine for the first time, you will need to accept the fingerprint.

Next, enter the SSH password for the remote user.

Once authenticated, the ssh-copy-id command will append the contents of the **id_rsa.pub** key to the **'~/.ssh/authorized_keys' file** on the remote machine and close the connection.


# Log in to the remote machine

Once you have completed all the above steps, you should now be able to login to the remote server without a password.

You can check this using the command:

```
ssh [email protected]_ip
```

Unless you've enabled a passphrase for your key, you'll be automatically signed in.

Hope you are succesful.

You finished reading the article "**How to Set Up and Use SSH in Linux**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.