# How to Setup Passwordless SSH Login

Updated Feb 19, 2019 • 4 min read



Secure Shell (SSH) is a cryptographic network protocol used for secure connection between a client and a server and supports various authentication mechanisms. The two most popular mechanisms are passwords based authentication and public key based authentication.

In this tutorial, we will show you how to setup an SSH key-based authentication as well how to connect to your Linux server without entering a password.

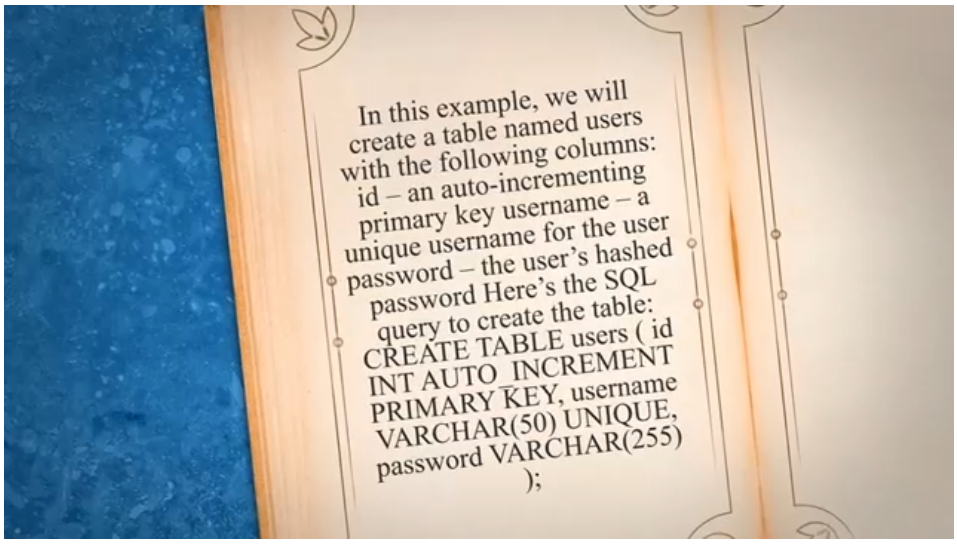
## Setup SSH Passwordless Login

To set up a passwordless SSH login in Linux all you need to do is to generate a public authentication key and append it to the remote hosts `~/.ssh/authorized_keys` file.

The following steps will describe the process for configuring passwordless SSH login:

01. Check for existing SSH key pair.

Before generating a new SSH key pair first check if you already have an SSH key on your client machine because you don't want to overwrite your existing keys.




How To Log In Php

Run the following ls command to see if existing SSH keys are present:

```
$ ls -al ~/.ssh/id_*.pub
```

If there are existing keys, you can either use those and skip the next step or backup up the old keys and generate a new one.

If you see `No such file or directory` or `no matches found` it means that you do not have an SSH key and you can proceed with the next step and generate a new one.

02. Generate a new SSH key pair.

The following command will generate a new 4096 bits SSH key pair with your email address as a comment:

```
$ ssh-keygen -t rsa -b 4096 -C "your_email@domain.com"
```

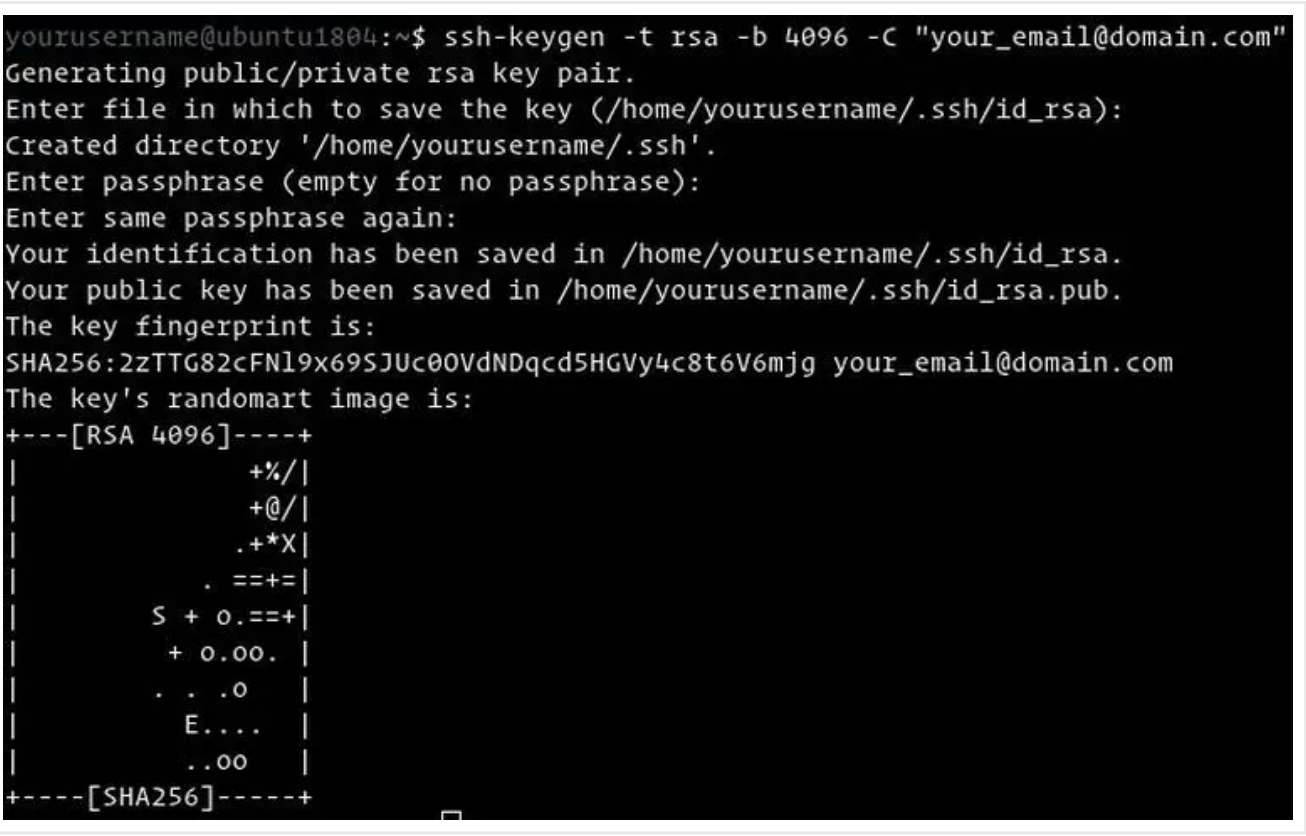
Press `Enter` to accept the default file location and file name:

```
Output
Enter file in which to save the key (/home/yourusername/.ssh/id_rsa):
```

Next, the `ssh-keygen` tool will ask you to type a secure passphrase. Whether you want to use passphrase it's up to you, if you choose to use passphrase you will get an extra layer of security. In most cases, developers and system administrators use SSH without a passphrase because they are useful for fully automated processes. If you don't want to use a passphrase just press `Enter` .

```
Output
Enter passphrase (empty for no passphrase):
```

The whole interaction looks like this:




To be sure that the SSH keys are generated you can list your new private and public keys with:

```
$ ls ~/.ssh/id_*
```

```
Output
/home/yourusername/.ssh/id_rsa /home/yourusername/.ssh/id_rsa.pub
```

03. Copy the public key

Now that you have generated an SSH key pair, in order to be able to login to your server without a password you need to copy the public key to the server you want to manage.

The easiest way to copy your public key to your server is to use a command called `ssh-copy-id`. On your local machine terminal type:

```
$ ssh-copy-id remote_username@server_ip_address
```

You will be prompted to enter the `remote_username` password:

```
Output
remote_username@server_ip_address's password:
```

Once the user is authenticated, the public key will be appended to the remote user `authorized_keys` file and connection will be closed.

If by some reason the `ssh-copy-id` utility is not available on your local computer you can use the following command to copy the public key:

04. Login to your server using SSH keys

    After completing the steps above you should be able log in to the remote server without being prompted for a password.

    To test it just try to login to your server via SSH:

    ```
    $ ssh remote_username@server_ip_address
    ```

    If everything went well, you will be logged in immediately.

## Disabling SSH Password Authentication

To add an extra layer of security to your server you can disable the password authentication for SSH.

Before disabling the SSH password authentication make sure you can log in to your server without a password and the user you are logging in with has sudo privileges.

The following tutorials describe how to configure sudo access:

- How to create sudo user on Ubuntu
- How to create sudo user on CentOS
- How to create sudo user on Debian

01. Log into your remote server with SSH keys, either as a user with sudo privileges or root:

    ```
    $ ssh sudo_user@server_ip_address
    ```

02. Open the SSH configuration file `/etc/ssh/sshd_config` , search for the following directives and modify as it follows:

/etc/ssh/sshd_config

```
PasswordAuthentication no
ChallengeResponseAuthentication no
UsePAM no
```

Once you are done save the file and restart the SSH service.

On Ubuntu or Debian servers, run the following command:

```
$ sudo systemctl restart ssh
```

On CentOS or Fedora servers, run the following command:

```
$ sudo systemctl restart sshd
```

## Conclusion

In this tutorial you have learned how to set up an SSH key-based authentication, allowing you to login to your remote server without providing a user password. You can add the same key to multiple remote serves.

We have also shown you how to disable SSH password authentication and add an extra layer of security to your server.

If you have any questions or feedback, feel free to leave a comment.

ssh security

## Related Articles

OCT 29, 2018

### How to Set up SSH SOCKS Tunnel for Private Browsing


OCT 22, 2018

### How to Set Up SSH Keys on Debian 9

OCT 7, 2018

How to Set Up SSH Keys on CentOS 7


Show comments (7)