

How to Set Up Passwordless SSH Login?

written by sysadmin | 1 January 2025

As a sysadmin, remote to a Linux server is a daily job to perform various checks on a Linux server. By default, if a sysadmin accesses a server, the sysadmin must enter a username and password. However, when the sysadmin has many servers, it is sometimes difficult for the sysadmin to enter the password for each server, especially if each server has a different password. Therefore, it needs to be made so that SSH does not need to enter a password when accessing a Linux server via SSH.

Problem

How to set up passwordless SSH Login?

Solution

There are 3 steps to setting up passwordless SSH:

1. Generate a key pair

Use `ssh-keygen` to generate a key pair consisting of a public key and a private key on the client computer:

```
ssh-keygen -t rsa
```

The **-t rsa** option specifies that the type of the key should be the RSA algorithm. Hit **Enter** to accept the default.

```
sysadmin@ubuntu2404:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sysadmin/.ssh/id_rsa): Hit the Enter key
Enter passphrase (empty for no passphrase): Hit the Enter key
Enter same passphrase again: Hit the Enter key
Your identification has been saved in /home/sysadmin/.ssh/id_rsa
Your public key has been saved in /home/sysadmin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:gg7kX2UI7jdcr7AuArZ8ATu3zWGQ0N4cP6XIIfHJSmKM sysadmin@ubuntu2404
The key's randomart image is:
+---[RSA 3072]-----+
|
| . =
| + 0 + o
| . X X = S
| .E & X = .
| o.= & 0 .
| =.+.= + .
| +o.oo .
+---[SHA256]-----+
sysadmin@ubuntu2404:~$
```



Running the ssh-keygen command

2. Upload the public key to the remote server

Use **ssh-copy-id** to propagate the public key to the server:

```
ssh-copy-id remote_username@remote_server_ip_address
```

For example, if you want to upload it to the server 192.168.56.2 with the username sysadmin, then use the command below:

```
ssh-copy-id sysadmin@192.168.56.2
```

Type **yes** when prompted and type the password for the remote server.

```

sysadmin@ubuntu2404:~$ ssh-copy-id sysadmin@192.168.56.2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sysadmin/.ssh/id_rsa.pub"
The authenticity of host '192.168.56.2 (192.168.56.2)' can't be established.
ED25519 key fingerprint is SHA256:q/E0kK5y9mMkVxtz3FbvMk1MEWmpW6HKZo0+fhBr8AE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
sysadmin@192.168.56.2's password: type the password

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'sysadmin@192.168.56.2'"
and check to make sure that only the key(s) you wanted were added.

sysadmin@ubuntu2404:~$

```

Running the ssh-copy-id command

For your information, the **id_rsa.pub** file will be saved in the **.ssh/authorized_keys** file on the remote server, like in the image below:

```

[sysadmin@RockyLinux9 ~]$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDRKJjeJQovqFYFLcascZiz370x5qBCSTTYNkfmcenllmCg7P2DvFri+2uH+1PJP1HNTqFCTVY2HcbLkxn1KAqZBYbhk74euIpSHND14DB9gWYCzEYDr605FgfwhXtpBXeVKc6MqVCK7LkedqGw1UQx48RU
AIz4WIAxc5m7Zq3ghv7BsIX3fjZG31jGSQhEkCq1/nl5T/eEMH8zXQqtV4ADhHGz9M/Yq2JK3q1v15tRMUotDc5zRtiJyHLDjs/yET+UwhbxLLRdNF7m9yyg5M2scmadMs4R0BBQ8AthKe5agy9NN8SEzS1x8LP5qVHsPGMQXkNj7XXT46GeAFhjF06e94D
YdvzNJfFh+scXePQF643CKn+d8vcmQYDJKcALF4r3d71K42q9z1EIdhyuYz0VZ53B/pqJFC0p0B1w/UsKtML0MONS54ly8IZ9KJLLp9RXdlmEq120E3UHxUNjbdcpvA59PchvFCkG14VUkrdZdNVoTr7bqZeAELPeDTyAs- sysadmin@ubuntu2404
[sysadmin@RockyLinux9 ~]$

```

The authorized_keys file

3. Test login via SSH

Try to connect to the server using SSH, you should be able to directly access the server without entering the password first. For example, I have 2 Linux servers, each of which uses Ubuntu OS with IP 192.168.56.100 and RockyLinux OS with IP 192.168.56.2. I want to access the RockyLinux server from the Ubuntu server without entering a password. I ran the three steps above to set up passwordless SSH on an Ubuntu server, and the results are as in the image below:

```

sysadmin@ubuntu2404:~$ ssh sysadmin@192.168.56.2
Last login: Mon Dec 30 05:38:18 2024 from 192.168.56.100
[sysadmin@RockyLinux9 ~]$

```

Access the server without entering a username and password

From the image above, you can see that I can directly access the server without entering the server password.

Note

By default, the system will generate a 2048-bit key in the first step when you run the `ssh-keygen` command. However, if you want to be more secure, you can use 4096-bit encryption by using the command below:

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

Besides RSA, you can also use several other public key algorithms, such as ECDSA or ED25519. Elliptic Curve Digital Signature Algorithm, or ECDSA, is one of the more complex public key cryptography encryption algorithms that supports three key sizes: 256, 384, and 521 bits. You can use the command below when using ECDSA:

```
ssh-keygen -t ecdsa -b 521
```

Ed25519 is an elliptic curve signing algorithm using EdDSA and Curve25519, and this is a new algorithm added in OpenSSH. You can use the command below when using ed25519:

```
ssh-keygen -t ed25519
```

Unfortunately, support for this among clients is not yet universal. Therefore, its use in general-purpose applications may not be advisable.

References

strongdm.com
phoenixnap.com
ssh.com
encryptionconsulting.com
cryptography.io