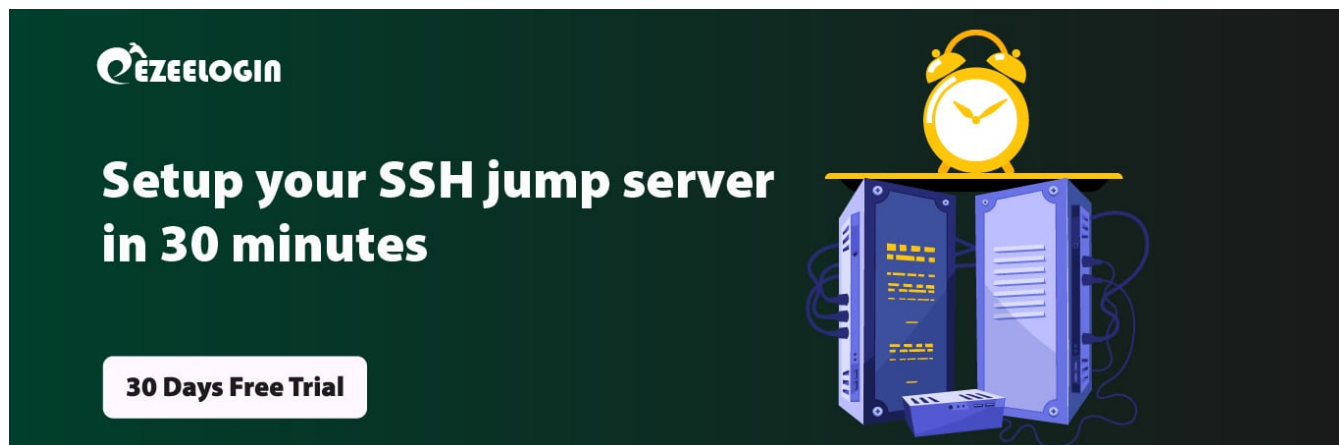


# Configure ssh certificate based authentication

298 Manu Chacko August 28, 2024 [Security Features](#), [Technical](#), [Tweaks & Configuration](#)  
27368



The banner features the EZEELOGIN logo in the top left. The main text reads "Setup your SSH jump server in 30 minutes" in a large, bold, white font. Below this, a white button contains the text "30 Days Free Trial". On the right side, there is an illustration of a server rack with a yellow alarm clock on top, symbolizing speed and reliability.

## Configure Certificate-Based SSH User Authentication

---

**Overview:** This article describes configuring certificate-based SSH user authentication with OpenSSH. It details creating a CA key, signing user SSH keys, and updating server settings to trust the CA. Additionally, it explains issuing short-lived certificates for temporary access, including setting validity periods. This setup provides secure, password-less SSH access by configuring both the server and client machines.

---

Support for certificate authentication of users and hosts using the new OpenSSH certificate format was introduced in Red Hat Enterprise Linux 6.5, in the `openssh-5.3p1-94.el6` package. If required, to ensure the latest OpenSSH package is installed, enter the following command as root :

```
~#
```

## Step 1: Setting Up Certificate Authority Infrastructure

- *Generate the CA key (cert\_ca) for signing user ssh keys with the following command*

```
:~# ssh-keygen -f cert_ca

Generating public/private RSA key pair.
Enter passphrase (empty for no passphrase):
Enter the same passphrase again:
Your identification has been saved in cert_ca.
Your public key has been saved in cert_ca.pub.
The key fingerprint is:
b3:af:e8:ef:c4:5d:90:f8:be:16:99:74:f2:39:3a:3e root@server
The key's randomart image is:
+---[RSA 2048]-----+
|          |
|   ..   |
|  .o   |
|  .o..  |
| S.*..  |
| . =+.+ |
|  +oo . |
|  o .E. |
| .oo+++o |
+-----+

:~# ls
cert_ca cert_ca.pub
```

## Step 2: Copy the keys to /etc/ssh/

```
:~# cp -pr cert_ca* /etc/ssh/
```

### Step 3: Add **CA public key** (cert\_ca.pub) as **Trusted Key** in the ssh server machines

```
vi /etc/ssh/sshd_config
```

(Add the following lines)

```
TrustedUserCAKeys /etc/ssh/cert_ca.pub
```

### Step 4: Restart SSH service

```
service sshd restart (For centos / rhel)
```

```
service ssh restart (For ubuntu / debian)
```

### Step 5: Generate **SSH key** for the user

```
:~# ssh-keygen -trsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
6f:d0:27:36:69:80:e4:ad:0c:5c:f9:d9:d8:af:a9:8d root@server
The key's randomart image is:
+---[RSA 2048]-----+
| o. |
| .+.o |
| o o.o= |
| o .+oo. |
```

```
| o S B.. |
| = +. |
| oo |
| +o |
| E.. |
+-----+
```

**Step 6: Sign user ssh public key by CA** with the following command

```
:~# ssh-keygen -s cert_ca -I user_username -n username -V +52w id_rsa.pub
```

```
Signed user key id_rsa-cert.pub: id "user_username" serial 0 for username valid from
2020-04-17T10:33:00 to 2021-04-16T10:34:42
```

**Note:** You should sign with user(username) you want to login to the server machine. For example, If you want to login as user " TED " to the server, You should sign the with the user " TED ". Example : ssh-keygen -s cert\_ca -I user\_ted -n ted -V +52w id\_rsa.pub

**Step 7: Copy the ssh keys to the client machine .ssh directory**

```
:~# scp id_rsa-cert.pub id_rsa admin@client1:/home/admin/.ssh/
```

**Step 8:** Once you copied the ssh keys to the client machine, the user will be able to login into the server with ssh certificate authentication without any password.

```
admin@client1:~# ssh username@server.com  
Last login: Fri Apr 17 11:39:17 2020 from client1  
[username@server~]#
```

### You can also issue short-lived certificates for giving temporary ssh access

**Step 1:** Follow the first three steps in the article.

**Step 2:** You can generate short-lived certificates for a day or a week or for some minutes while Sign user SSH public key by CA with the following command: You can specify the validity interval ( -V )while signing the certificate. We have issued a certificate that is valid for 60 minutes from the generated time. We can specify the Validity after -V option.

```
-V validity_interval
```

Specify a validity interval when signing a certificate. A validity interval may consist of a single time, indicating that the certificate is valid beginning now and expiring at that time, or may consist of two times separated by a colon to indicate an explicit time interval.

The start time may be specified as the string “always” to indicate the certificate has no specified start time, a date in YYYYMMDD format, a time in YYYYMMDDHHMM[SS] format, a relative time (to the current time) consisting of a minus sign followed by an interval in the format described in the TIME FORMATS section of sshd\_config(5).

The end time may be specified as a YYYYMMDD date, a YYYYMMDDHHMM[SS] time, a relative time starting with a plus character or the string “forever” to indicate that the certificate has no expiry date.

For example: “+52w1d” (valid from now to 52 weeks and one day from now), “-4w:+4w” (valid from four weeks ago to four weeks from now), “20100101123000:20110101123000” (valid from 12:30 PM, January 1st, 2010 to 12:30 PM, January 1st, 2011), “-1d:20110101” (valid from yesterday to midnight, January 1st, 2011), “-1m:forever” (valid from one minute ago and never expiring).

```
~# ssh-keygen -s cert_ca -I user_username -n username -V +60m id_rsa.pub
```

```
Signed user key id_rsa-cert.pub: id "user_username" serial 0 for username valid from  
2022-04-17T10:33:00 to 2022-04-17T11:33:00
```

**Note:** You should sign with the user(username) you want to log in to the server machine. For example, If you want to login as user " TED " to the server, You should sign the with the user " TED ". Example :

```
ssh-keygen -s cert_ca -I user_ted -n ted -V +60m id_rsa.pub
```

**Step 3:** Copy the ssh keys to the client machine .ssh directory

```
~# scp id_rsa-cert.pub id_rsa admin@client1:/home/admin/.ssh/
```

**Step 4:** Once you copied the ssh keys to the client machine, the user will be able to login into the server with ssh certificate authentication without any password till **60 minutes** from the certificated generated time. The certificate will expire after 60 minutes, so the user will not be able to ssh to that server with that certificate after 60 minutes.

```
admin@client1:~# ssh username@server.com
```

```
Last login: Fri Apr 17 11:39:17 2020 from client1
```

[username@server~]#

---

## **Related Articles**

[How to install ssl certs in jump server \[secure connection\] ?](#)

[How To Create a Self-Signed SSL Certificate for Nginx on debian](#)

Online URL:

<https://www.ezeelogin.com/kb/article/configure-ssh-certificate-based-authentication-298.html>