
Hands-on Lab
Server Application Requires a Secure SSH Connection
Linux Academy
Cloud Assessments

Contents

In this lab, we will create a service account for the local system that will not have a password, but an SSH key that will make it so that only a user connected to the local host server can access it.

Create a User

First, you will need to become the root user for this lab. As soon as you are, we will create a new user with the name client. You will need to remember the password you make for this user later:

```
[root@linuxacademy ~]# useradd client
[root@linuxacademy ~]# passwd client
```

Once the user is created, we want to test the connection to that user. When prompted, enter `yes` to connect.

```
[root@linuxacademy ~]# ssh client@localhost
```

After we've have checked that connection, we'll go back to root, and then become the client user:

```
[root@linuxacademy ~]# su - client
```

Once we're the client user, we'll need to check to make sure we are in the correct directory:

```
[client@linuxacademy ~]$ cd
[client@linuxacademy ~]$ pwd
/home/client
```

Generate a Key

Now that we're in our home directory, it's time for us to generate the SSH key that will allow us into this server whenever we need to without having to worry about a password:

```
[client@linuxacademy ~]$ ssh-keygen
```

We'll be prompted to change some options; each time this happens, hit enter without adding anything to apply the default settings to the SSH key. Also, make sure NOT to enter a passphrase. We only want to have the connection for this to involve an ssh key; adding a passphrase will change this and require a password to be entered.

With the keys created, let's check them out in the `.ssh` directory:

```
[client@linuxacademy ~]$ cd .ssh
[client@linuxacademy .ssh]$ ll
total 8
-rw------- 1 client client 1675 $DATE $TIME id_rsa
-rw------- 1 client client 401 $DATE $TIME id_rsa.pub
```

Note that both keys are only owned by the user.

Copy the SSH Key

While in this directory, we can add the SSH key over the localhost. When prompted, enter `yes`:

```
[client@linuxacademy .ssh]$ ssh-copy-id client@localhost
```

You will be asked for the password you created at the beginning of the lab for the client user. As soon you enter it, the key will be copied over to the localhost.

Check your keys by running `ll` in the `.shh` directory.

```
[client@linuxacademy .ssh]$ ll
total 16
-rw------- 1 client client 401 $DATE $TIME authorized_keys
-rw------- 1 client client 1675 $DATE $TIME id_rsa
-rw------- 1 client client 401 $DATE $TIME id_rsa.pub
-rw-r--r-- 1 client client 391 $DATE $TIME known_hosts
```

With our keys available, `cat` out and check the authorized key:

```
[root@linuxacademy ~]# cat authorized_keys
```

The public key will appear. With that key owned by our user, we can now log in in to the local server as long as we are the client user, and we won't need a password or passcode to do so:

```
[client@linuxacademy ~]# ssh client@localhost
```

To test to make sure the only the client user can do this, let's exit out to the root user and try to log in to the server:

```
[root@linuxacademy ~]# ssh client@localhost
client@locahost's password:
```

We're prompted for the password because the SSH key is only connected to the client, not root. We would need to have the client user's password to proceed.

Let's go back to being client, and check that we're in our home directory:

```
[root@linuxacademy ~]# su - client
[client@linuxacademy ~]# cd
[client@linuxacademy ~]# pwd
/home/client
```

While in the client directory, let's connect through SSH to localhost again:

```
[client@linuxacademy ~]# ssh localhost
```

We'll automatically sign in without being prompted for a password or passphrase.

Review

Now that you have completed this lab, you can now create a secure connection with a specific user to the localhost without using a password or passphrase, with also making sure no one else can access the localhost passwordless as well. Congratulations on completing this lab!