**Configure SSH key-based authentication.**

Last updated on [June 3, 2017](https://www.certdepot.net/rhel7-configure-ssh-key-based-authentication/) (36,320 views) - CertDepot — [10 Comments ↓](https://www.certdepot.net/rhel7-configure-ssh-key-based-authentication/#comments)

Share this link

[Twitter0](https://twitter.com/share?original_referer=https%3A%2F%2Fwww.certdepot.net%2F&source=tweetbutton&text=RHEL7%3A+Configure+SSH+key-based+authentication.&url=https%3A%2F%2Fwww.certdepot.net%2Frhel7-configure-ssh-key-based-authentication%2F&via=CertDepot)

[LinkedIn0](https://www.linkedin.com/cws/share?url=https://www.certdepot.net/rhel7-configure-ssh-key-based-authentication/)

Note: This is an [RHCSA 7 exam objective](https://www.certdepot.net/rhel7-rhcsa-exam-objectives/) and an [RHCE 7 exam objective](https://www.certdepot.net/rhel7-rhce-exam-objectives/).

**Presentation**

Instead of connecting through login/password to a remote host, **SSH** allows you to use key-based authentication. To set up key-based authentication, you need two virtual/physical servers that we will call **server1** and **server2**.

**Configuration Procedure**

On the **server1**, create a user **user01** with password **user01**:

# **useradd user01**

# **passwd user01**

Changing password for user user01.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

On the **server2**, create the same user with password **user01**:

# **useradd user01**

# **passwd user01**

Changing password for user user01.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

On the **server1**, connect as this new user:

# **su - user01**

Generate a private/public pair for key-based authentication (here **rsa** key with **2048** bits and **no passphrase**):

[user01@server1 ~]$ **ssh-keygen -b 2048 -t rsa**

Generating public/private rsa key pair.

Enter file in which to save the key (/home/user01/.ssh/id\_rsa): **return**

Created directory '/home/user01/.ssh'.

Enter passphrase (empty for no passphrase): **return**

Enter same passphrase again: **return**

Your identification has been saved in /home/user01/.ssh/id\_rsa.

Your public key has been saved in /home/user01/.ssh/id\_rsa.pub.

The key fingerprint is:

6d:ac:45:32:34:ac:da:4a:3b:4e:f2:83:85:84:5f:d8 user01@server1.example.com

The key's randomart image is:

+--[ RSA 2048]----+

| .o |

| ... |

| . o .o . |

|. o E . \* |

| o o o S = |

| o + . + |

| .+.o . |

| .+= |

| .oo |

+-----------------+

Still on **server1**, copy the public key to **server2**.

[user01@server1 ~]$ **ssh-copy-id -i .ssh/id\_rsa.pub user01@server2.example.com**

The authenticity of host 'server2.example.com (192.168.1.49)' can't be established.

ECDSA key fingerprint is 67:79:67:88:7f:da:31:49:7b:dd:ed:40:af:ae:b6:ae.

Are you sure you want to continue connecting (yes/no)? yes

/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

user01@server2.example.com's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'user01@server2.example.com'"

and check to make sure that only the key(s) you wanted were added.

On the **server2**, edit the **/etc/ssh/sshd\_config** file and set the following options:

PasswordAuthentication no

PubkeyAuthentication yes

Note: Don’t hesitate to [set up a virtual console access](https://www.certdepot.net/rhel7-access-virtual-machines-console/) on **server2**, this will avoid re-installing the physical/virtual server if something goes wrong.

Restart the **sshd** service:

# **systemctl restart sshd**

**Testing Time**

On the **server1** as **user01**, connect to the **server2**:

[user01@server1 ~]$ **ssh server2.example.com**

Note1: This configuration can also be done for the **root** account.  
Note2: Use **-v**, **-vv**, or **-vvv** options to get some debug information