nixCraft
Linux and Unix tutorials for new and seasoned sysadmin

How To Set up SSH Keys on a Linux / Unix System

last updated August 7, 2019 **in CentOS, Cryptography, Debian / Ubuntu, Linux, OpenBSD, RedHat and Friends, UNIX**

I recently read that SSH keys provide a secure way of logging into a Linux and Unix-based server. How do I set up SSH keys on a Linux or Unix based systems? In SSH for Linux/Unix, how do I set up public key authentication?

This page explains a public key and shows you how to set up SSH keys on a Linux or Unix-like server. I am assuming that you are using Linux or Unix-like server and client with the following software:

- OpenSSH SSHD server
- OpenSSH ssh client and friends on Linux (Ubuntu, Debian, {Free,Open,Net}BSD, RHEL, CentOS, MacOS/OSX, AIX, HP-UX and co).

What is a public key authentication?

OpenSSH server supports various authentication schema. The two most popular are as follows:

1. Passwords based authentication
2. Public key based authentication. It is an alternative security method to using passwords. This method is recommended on a VPS, cloud, dedicated or even home based server.

How to set up SSH keys

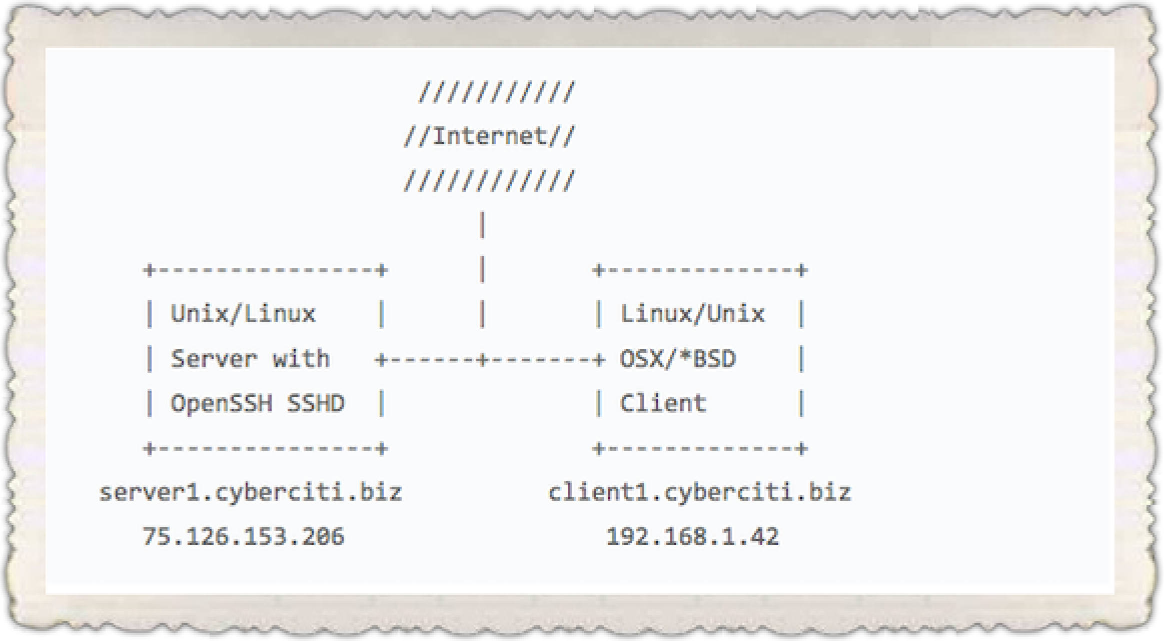
Steps to setup secure ssh keys:

1. Create the ssh key pair using **`ssh-keygen`** command.
2. Copy and install the public ssh key using **`ssh-copy-id`** command on a Linux or Unix server.
3. Add yourself to sudo or wheel group admin account.
4. Disable the password login for root account.
5. Test your password less ssh keys login using **`ssh user@server-name`** command.

Let us see all steps in details.

How do I set up public key authentication?

You must generate both a public and a private key pair. For example:



```
                      ///////////
                     //Internet//
                     ////////////
                          |
   +---------------+      |       +-------------+
   | Unix/Linux    |      |       | Linux/Unix  |
   | Server with   +------+-------+ OSX/*BSD    |
   | OpenSSH SSHD  |              | Client      |
   +---------------+              +-------------+
server1.cyberciti.biz          client1.cyberciti.biz
   75.126.153.206                  192.168.1.42
```

Fig.01: Our sample setup

Where,

- server1.cyberciti.biz – You store your public key on the remote hosts and you have an accounts on this Linux/Unix based server.
- client1.cyberciti.biz – Your private key stays on the desktop/laptop/ computer (or local server) you use to connect to server1.cyberciti.biz server. Do not share or give your private file to anyone.

In public key based method you can log into remote hosts and server, and transfer files to them, without using your account passwords. Feel free to replace server1.cyberciti.biz and client1.cyberciti.biz names with your actual setup. Enough talk, let's set up public key authentication. Open the Terminal and type following commands if .ssh directory does not exists:

```
mkdir -p $HOME/.ssh
chmod 0700 $HOME/.ssh
```

1: Create the key pair

On the computer (such as client1.cyberciti.biz), generate a key pair for the protocol.

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

Sample outputs:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/vivek/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/vivek/.ssh/id_rsa.
Your public key has been saved in /Users/vivek/.ssh/id_rsa.pub.
The key fingerprint is:
80:5f:25:7c:f4:90:aa:e1:f4:a0:01:43:4e:e8:bc:f5 vivek@desktop01
The key's randomart image is:
+--[ RSA 2048]----+
| oo    ...+.     |
|.oo  .  .ooo     |
|o .o. . .o  .    |
| o ...+o.        |
|  o .=.=S        |
| .  .Eo .        |
    |                 |
    |                 |
    |                 |
    +-----------------+
```

You need to set the Key Pair location and name. I recommend you use the default location if you do not yet have another key there, for example: $HOME/.ssh/id_rsa. You will be prompted to supply a passphrase (password) for your private key. I suggest that you setup a passphrase when prompted. You should see two new files in $HOME/.ssh/ directory:

1. `$HOME/.ssh/id_rsa`– contains your private key.
2. `$HOME/.ssh/id_rsa.pub` – contain your public key.

Optional syntax for advance users

The following syntax specifies the 4096 of bits in the RSA key to creation (default 2048):

```
$ ssh-keygen -t rsa -b 4096 -f ~/.ssh/vps-cloud.web-server.key -C "My web-server key"
```

Where,

- **`-t rsa`** : Specifies the type of key to create. The possible values are "rsa1" for protocol version 1 and "dsa", "ecdsa", "ed25519", or "rsa" for protocol version 2.
- **`-b 4096`** : Specifies the number of bits in the key to create
- **`-f ~/.ssh/vps-cloud.web-server.key`** : Specifies the filename of the key file.
- **`-C "My web-server key"`** : Set a new comment.

2: Install the public key in remote server

Use scp or ssh-copy-id command to copy your public key file (e.g., $HOME/.ssh/id_rsa.pub) to your account on the remote server/host (e.g., nixcraft@server1.cyberciti.biz). To do so, enter the following command on your client1.cyberciti.biz:

```
ssh-copy-id -i $HOME/.ssh/id_rsa.pub user@server1.cyberciti.biz
```

OR just copy the public key in remote server as authorized_keys in ~/.ssh/ directory:

```
scp $HOME/.ssh/id_rsa.pub user@server1.cyberciti.biz:~/.ssh/authorized_keys
```

A note about appending the public key in remote server

On some system `ssh-copy-id` command may not be installed, so use the following commands (when prompted provide the password for remote user account called vivek) to install and append the public key:

```
## First create .ssh directory on server ##
ssh vivek@server1.cyberciti.biz "umask 077; test -d .ssh || mkdir .ssh"

## cat local id.rsa.pub file and pipe over ssh to append the public key in remote server
cat $HOME/.ssh/id_rsa.pub | ssh vivek@server1.cyberciti.biz "cat >> .ssh/authorized_keys
```

3: Test it (type command on client1.cyberciti.biz)

The syntax is as follows for the ssh command:

```
ssh user@server1.cyberciti.biz
ssh user@your-server-ip-address
ssh -i ~/.ssh/your-key user@your-server-ip-address
```

Or copy a text file called foo.txt:

```
scp foo.txt user@server1.cyberciti.biz:/tmp/
```

You will be prompted for a passphrase. To get rid of passphrase whenever you log in the remote host, try ssh-agent and ssh-add commands.

What are ssh-agent and ssh-add, and how do I use them?

To get rid of a passphrase for the current session, add a passphrase to ssh-agent and you will not be prompted for it when using ssh or scp/sftp/rsync to connect to hosts with your public key. The syntax is as follows:

```
eval $(ssh-agent)
```

Type the `ssh-add` command to prompt the user for a private key passphrase and adds it to the list maintained by ssh-agent command:

```
ssh-add
```

Enter your private key passphrase. Now try again to log into user@server1.cyberciti.biz and you will not be prompted for a password:

```
ssh user@server1.cyberciti.biz
```

One can list public key parameters of all identities with the `-L` option:

```
ssh-add -L
```

Deleting all private keys from the ssh-agent can be done with the `-D` option as follows:

```
ssh-add -D
```

When you log out kill the ssh agent, run:

```
kill $SSH_AGENT_PID
```

You can also add something like the below to your shell startup to kill ssh-agent at logout:

```
trap "kill $SSH_AGENT_PID" 0
```

4: Disable the password based login on a server

Login to your server, type:

```
## client commands ##
eval $(ssh-agent)
ssh-add
ssh user@server1.cyberciti.biz
```

Edit /etc/ssh/sshd_config on server1.cyberciti.biz using a text editor such as nano or vim:

⚠ **Warning**: Make sure you add yourself to sudoers files. Otherwise you will not able to login as root later on. See "How To Add, Delete, and Grant Sudo Privileges to Users on a FreeBSD Server" for more info.

```
$ sudo vim /etc/ssh/sshd_config
```

OR directly jump to PermitRootLogin line using a vim text editor:

```
$ sudo vim +/PermitRootLogin /etc/ssh/sshd_config
```

Find PermitRootLogin and set it as follows:

```
PermitRootLogin no
```

Save and close the file. I am going to add a user named vivek to sudoers on Ubuntu Linux:

```
# adduser vivek
```

Finally, reload/restart the sshd server, type command as per your Linux/Unix version:

```
## CentOS/RHEL/Fedora (older version) Linux server reload sshd ##
sudo service sshd reload

## CentOS/RHEL/Fedora (latest version i.e. systemd based) Linux server reload sshd ##
sudo systemctl reload sshd

## Debian/Ubuntu Linux (older version) server reload sshd ##
sudo /etc/init.d/ssh reload

## Debian/Ubuntu Linux (systemd based latest) server reload sshd ##
sudo systemctl reload ssh

## Generic Unix method to reload sshd ##
sudo kill -HUP `cat /var/run/sshd.pid`
OR
sudo kill -HUP $(cat /var/run/sshd.pid)
```

5: How to add or replace a passphrase for an existing private key?

To to change your passphrase type the following command:

```
ssh-keygen -p
```

6: How do I backup an existing private/public key?

Just copy files to your backup server or external USB pen/hard drive:

```
## Copy files to  home based nas server ##
rsync -avr $HOME/.ssh user@home.nas-server:/path/to/encrpted/nas/partition/

## Copy files to  usb pen drive mounted at /mnt/usb ##
cp -avr $HOME/.ssh/ /mnt/usb/backups/
```

How do I protect my ssh keys?

1. Always use a strong passphrase.
2. Do not share your private keys anywhere online or store in insecure cloud storage.
3. Restrict privileges of the account.

How do I create and setup an OpenSSH config file to create shortcuts for servers I frequently access?

See how to create and use an OpenSSH ssh_config file for more info.

Conclusion

This page explained how to set up ssh keys for authentication purposes. For more info see the following resources:

- keychain: Set Up Secure Passwordless SSH Access For Backup Scripts
- Ubuntu / Debian Linux Server Install Keychain SSH Key Manager For OpenSSH
- Man pages – ssh-keygen(1)
- OpenSSH project homepage here.

And, there you have it, ssh set up with public key based authentication for Linux or Unix-like systems.

SHARE ON Facebook Twitter

Posted by: Vivek Gite

The author is the creator of nixCraft and a seasoned sysadmin, DevOps engineer, and a trainer for the Linux operating system/Unix shell scripting. Get the **latest tutorials on SysAdmin, Linux/Unix and open source topics via [RSS/XML feed](#)** or [weekly email newsletter](#).

Historical Comment Archive

18 comment

Casper March 9, 2014 at 9:26 pm

Things might have changed, but did you not forget to setup “authorized_keys” ?

> **Nix Craft** March 10, 2014 at 4:26 am
>
> Thanks for the heads up! The faq has been updated.

Topper March 10, 2014 at 6:28 am

Some typo error:
cp -avr $HONE/.ssh/ /mnt/usb/backups/
has to be
cp -avr $HOME/.ssh/ /mnt/usb/backups/

> **Nix Craft** March 10, 2014 at 11:34 am
>
> Typo is fixed. I appreciate your feedback.

Terry March 13, 2014 at 11:34 am

To add to the story, I do this often with keys setup from my office desktop.

```
for i in server1 server2 server3 server4; do ssh mylogin@$i "hostname" ; done
```

This one line will login to the four servers and run the command hostname .

Replace hostname with your hearts desire.
–Terry

Louise August 20, 2014 at 12:30 pm

I have my ssh keys setup with up on one server and connecting with two other servers successfully. I now have to rename and reip all three servers, i noticed with the pub key the server name is listed. When i rename and re-ip my servers is there any way to update the keys or do i have re-create again?

Thank you

rajesh September 23, 2014 at 4:55 pm

i have generate ssh key on A server for communicate server B. then i can able to do ssh with out passwd to server B.

but when i try to login in Server B . ..then trying to do ssh to server A ..it is asking password..why lit happens ?

it should be vise versa ..rt ? server A – server B , Server B- server A

Thanks

rajesh December 2, 2014 at 3:34 am

A to B password less
server A:
ssh-keygen -t rsa
press enter,enter
two keys is create the path /root/.ssh
next go to the cd .ssh
ls -lrt
scp id_pas.pub root@serverB:/home
server B:
check the cd /home
ls
cd /root/.ssh
in server B is not .ssh directory
create the directory
mkdir /root/.ssh
chmod 700 /root/.ssh

cd /root/.ssh
cp /home/id_pas.pub authorized_keys

already authorized_keys in another keys
so append data because over read the data
cat>>authorized_keys</home/id_pas.pub -this command is append the data not over read the old data
next go to server A
ssh root@serverB
login the passwd lesss

milosz October 23, 2014 at 12:33 pm

I believe you forgot the pipe character:

```
cat $HOME/.ssh/id_rsa.pub ssh vivek@server1.cyberciti.biz cat >> .ssh/authorized_keys
```

should be:

```
cat $HOME/.ssh/id_rsa.pub | ssh vivek@server1.cyberciti.biz cat >> .ssh/authorized_keys
```

> **nixCraft** October 24, 2014 at 6:03 am
>
> Thanks for the heads up!

amrx December 12, 2014 at 6:30 am

How can I access this remote server from different network ?

> **Vivek Gite** July 6, 2016 at 4:53 pm
>
> What do you mean?

Jason September 17, 2015 at 11:32 am

Having an issue.
When I am prompted for a key in Ubuntu 14.04, ther terminal will not let me enter one.
I press buttons but nothing appears, and I don't want to set one up without passphrase.

Dan May 3, 2016 at 3:57 pm

Great article, I have found many use full command and scripts to help with setting up a kickstart configuration that loads and configures without interaction using PXEboot.

Do you have an article about a one line to configure ssh public/private key that will run silent and without interaction.

Thank you

volcano May 17, 2016 at 12:41 pm

I did everything as said in article, but there's an error everytime I connect to server: "Server refused our key". I generated key on windows machine with puttygen. May be there are some settings I should change in sshd_config?

Ramphy Rojas July 6, 2016 at 5:09 am

The best explains about SSH SET UP! Thank you!

Flip October 20, 2016 at 1:24 pm

Honestly asking…

Why not use 'ssh-keygen -t rsa -b 4096' to generate the key? Is 2048 perfectly fine in this day n age?

> **Vivek Gite** October 20, 2016 at 7:40 pm
>
> Generally, 2048 bits is considered sufficient. Nevertheless, I added some notes about it :)

Still, have a question? Get help on our forum!

Tagged as: /etc/ssh/sshd_config, ssh-keygen command, Easy

PRIVACY

TERM OF SERVICE

CONTACT/EMAIL

DONATIONS

SEARCH