How To Set Up SSH Keys

Authored by: ***ASPHostServer Administrator*** *[asphostserver@gmail.com]*
Saved From: *http://faq.asphosthelpdesk.com/article.php?id=174*

SSH keys provide a more secure way of logging into a virtual private server with SSH than using a password alone. While a password can eventually be cracked with a brute force attack, SSH keys are nearly impossible to decipher by brute force alone. Generating a key pair provides you with two long string of characters: a public and a private key. You can place the public key on any server, and then unlock it by connecting to it with a client that already has the private key. When the two match up, the system unlocks without the need for a password. You can increase security even more by protecting the private key with a passphrase.

Step One—Create the RSA Key Pair

The first step is to create the key pair on the client machine (there is a good chance that this will just be your computer):

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

Step Two—Store the Keys and Passphrase

Once you have entered the Gen Key command, you will get a few more questions:

```
Enter file in which to save the key (/home/demo/.ssh/id_rsa):
```

You can press enter here, saving the file to the user home (in this case, my example user is called demo).

```
Enter passphrase (empty for no passphrase):
```

It's up to you whether you want to use a passphrase.

Entering a passphrase does have its benefits: the security of a key, no matter how encrypted, still depends on the fact that it is not visible to anyone else. Should a passphrase-protected private key fall into an unauthorized users possession, they will be unable to log in to its associated accounts until they figure out the passphrase, buying the hacked user some extra time. The only downside, of course, to having a passphrase, is then having to type it in each time you use the Key Pair.

The entire key generation process looks like this:

```
ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/demo/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/demo/.ssh/id_rsa.
```

```
Your public key has been saved in /home/demo/.ssh/id_rsa.pub.
The key fingerprint is:
4a:dd:0a:c6:35:4e:3f:ed:27:38:8c:74:44:4d:93:67 demo@a
The key's randomart image is:
+--[ RSA 2048]----+
|          .oo.   |
|         .  o.E  |
|        + .  o   |
|     . = = .     |
|      = S = .    |
|     o + = +     |
|      . o + o .  |
|           . o   |
|                 |
+-----------------+
```

The public key is now located in /home/demo/.ssh/id_rsa.pub

The private key (identification) is now located in /home/demo/.ssh/id_rsa

Step Three—Copy the Public Key

Once the key pair is generated, it's time to place the public key on the virtual server that we want to use.

You can copy the public key into the new machine's authorized_keys file with the ssh-copy-id command. Make sure to replace the example username and IP address below.

```
ssh-copy-id user@123.45.56.78
```

Alternatively, you can paste in the keys using SSH:

```
cat ~/.ssh/id_rsa.pub | ssh user@123.45.56.78 "cat >> ~/.ssh/authorized_keys"
```

No matter which command you chose, you should see something like:

```
The authenticity of host '12.34.56.78 (12.34.56.78)' can't be established.
RSA key fingerprint is b1:2d:33:67:ce:35:4d:5f:f3:a8:cd:c0:c4:48:86:12.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '12.34.56.78' (RSA) to the list of known hosts.
user@12.34.56.78's password:
Now try logging into the machine, with "ssh 'user@12.34.56.78'", and check in:
  ~/.ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.
```

Now you can go ahead and log intouser@12.34.56.78and you will not be prompted for a password. However, if you set a passphrase, you will be asked to enter the passphrase at that time (and whenever else you log in in the future).

Optional Step Four—Disable the Password for Root Login

Once you have copied your SSH keys unto your server and **ensured that you can log in with the SSH keys alone**, you can go ahead and restrict the root login to only be permitted via SSH keys.

In order to do this, open up the SSH config file:

```
sudo nano /etc/ssh/sshd_config
```

Within that file, find the line that includes `PermitRootLogin` and modify it to ensure that users can only connect with their SSH key:

```
PermitRootLogin without-password
```

Put the changes into effect:

```
reload ssh
```