AWS, EC2

SSH between two AWS EC2 Ubuntu 16.04 instances

AUGUST 14, 2018 | ANUSHA SHARMA | LEAVE A COMMENT

This blog will enumerate the steps required to setup a password less ssh connection between two AWS EC2 Ubuntu 16.04 instances.

Now a days, the trend is to host the webserver (like Apache2, tomcat, rail etc) on cloud (like Amazon Web Services). Thus setting up the ssh connection between the servers on the cloud could be a bit challenging and brain scratching task for the ones who are new to the AWS platform. We will try to help you by provisioning easy and detailed steps to achieve this task.

Pre-requisite:

- Before connection could be setup, two ubuntu instances are required to be installed and configures with necessary security groups. Click **here** to follow the steps to launch the instances, if in case you have never done it before.
- Connect to the instances (e.g. Putty if a windows user)

SSH between two AWS EC2 Ubuntu 16.04 instances:

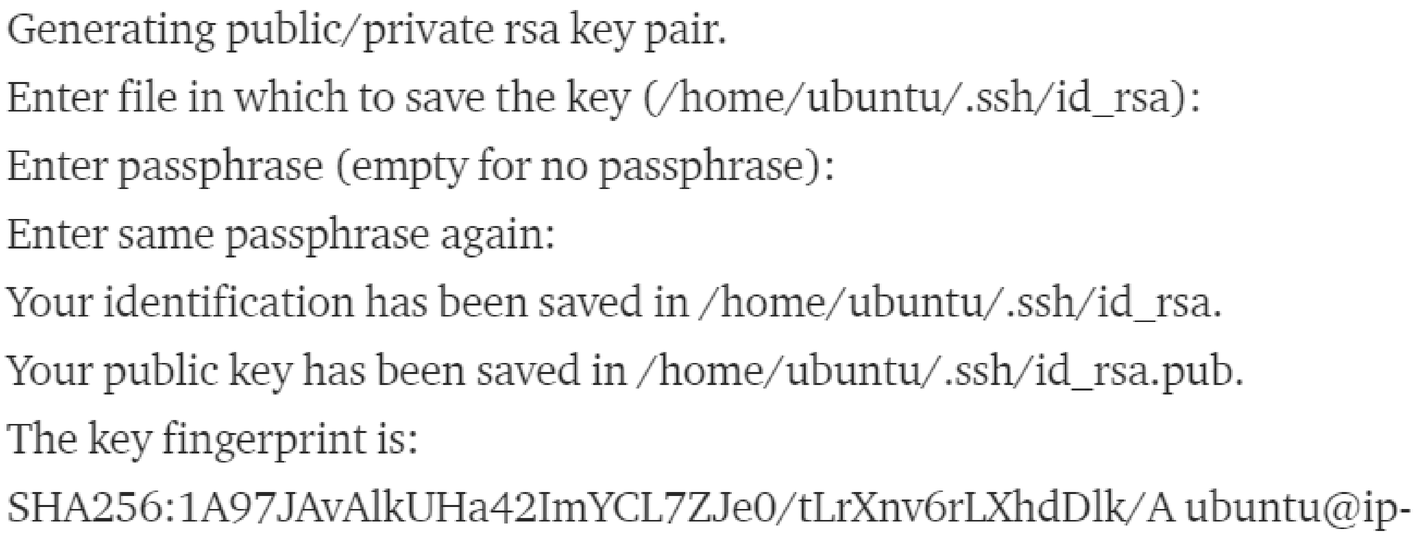
Let's consider a use case wherein; we have two ubuntu instances, namely 'A' and 'B'. We would want to copy (ssh, scp, sftp, rsync) files from instance A (source) to instance B(target). Suppose Apache2 has been hosted by instance B and we would want to copy .html files from server A to /var/www/html folder of Apache2.

Following steps are required to be followed:

Steps to be performed on instance A (source instance):

Step 1. Generate a public/private keypair

```
[ubuntu@ip-xxx-xx-xx-xx]$ ssh-keygen -t rsa
```



```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:1A97JAvAlkUHa42ImYCL7ZJe0/tLrXnv6rLXhdDlk/A ubuntu@ip-
```

- It will ask you the name of file to save the key in. Press enter to keep it default (i.e. id_rsa).
- Also do not enter any passphrase to avoid asking password everytime.
- The keypair file is generated in

/home/ubuntu/.ssh/id_rsa

Step 2. Copy the public key (id_rsa.pub) and not the private key (id_rsa)

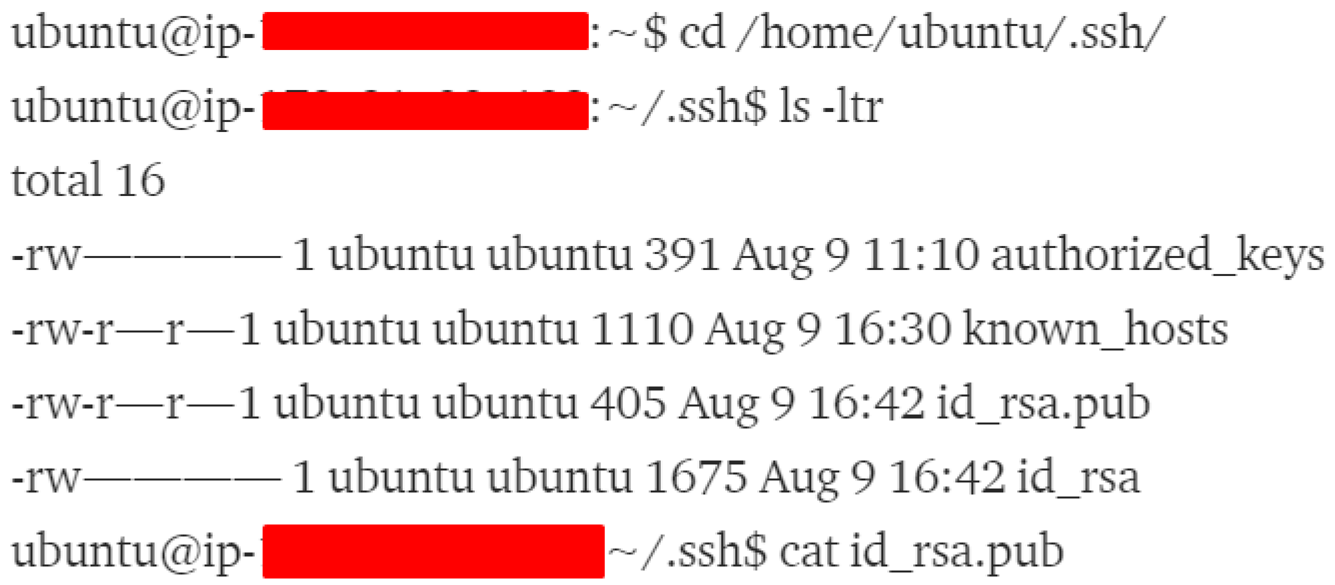
- Browse to the directory containing id_rsa.pub

```
[ubuntu@ip-xxx-xx-xx-xx]$ cd /home/ubuntu/.ssh/
```

- List out all the file and sub-folders under .ssh

```
[ubuntu@ip-xxx-xx-xx-xx]$ ls -ltr
```

This command just list the contents of the current directory in the long listing format (`-l`), sorted by modification time (`-t`) in reverse order (`-r`) of all files and directories beginning with `file*`.



```
ubuntu@ip-[redacted]:~$ cd /home/ubuntu/.ssh/
ubuntu@ip-[redacted]:~/.ssh$ ls -ltr
total 16
-rw———— 1 ubuntu ubuntu 391 Aug 9 11:10 authorized_keys
-rw-r—r—1 ubuntu ubuntu 1110 Aug 9 16:30 known_hosts
-rw-r—r—1 ubuntu ubuntu 405 Aug 9 16:42 id_rsa.pub
-rw———— 1 ubuntu ubuntu 1675 Aug 9 16:42 id_rsa
ubuntu@ip-[redacted]~/.ssh$ cat id_rsa.pub
```

Note: In order to find a particular file within a directory, use:

`find ./ -name file*`: That command searches trough the whole directory structure under the current working directory and all its sub-directories for files and directories beginning with `file*`in their names.

`ls` only applies to the current working directory, while `find`applies to all files and subdirectories starting from the current working directory.

- Use cat command to view the content of id_rsa.pub

```
[ubuntu@ip-xxx-xx-xx-xx]$ cat id_rsa.pub
```

- manually copy the content (which is the generated public key from header to footer)

Step 3. A test.html file created

```
[ubuntu@ip-xxx-xx-xx-xx]$ touch test.html
```

Steps to be performed on instance B (target instance):

Step 1. Edit sshd_config

```
[ubuntu@ip-xxx-xx-xx-xx]$ cd /etc/ssh
[ubuntu@ip-xxx-xx-xx-xx]$ ls -ltr
```

```
[ubuntu@ip-xxx-xx-xx-xx]$ sudo vi sshd_config
```

Uncomment the following two lines:

RSAAuthentcation yes

PubkeyAuthentication yes

Step 2. Browse to the authorized_keys file

```
[ubuntu@ip-xxx-xx-xx-xx]$ cd /ubuntu/.ssh
[ubuntu@ip-xxx-xx-xx-xx]$ ls -ltr
[ubuntu@ip-xxx-xx-xx-xx]$ sudo vi authorized_keys
```

Now, append the copied content from id_rsa.pub file of instance A to the authorized_keys file of instnace B.

vi : command to open the editor

press i to get into the insert mode to paste the content in the end.

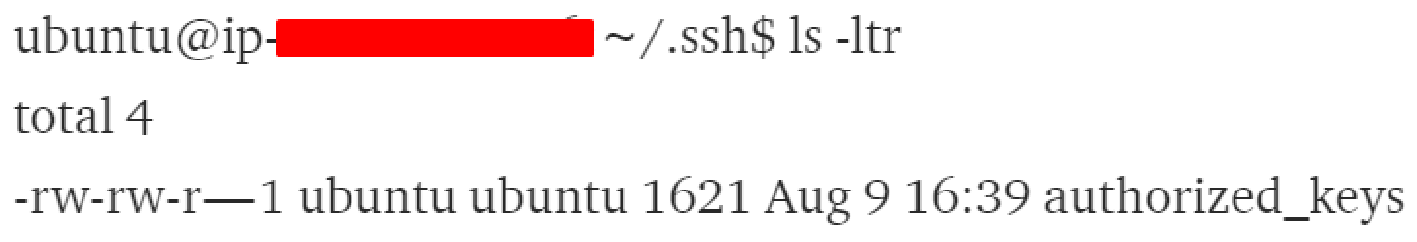
press esc key to exit from the insert mode.

type –

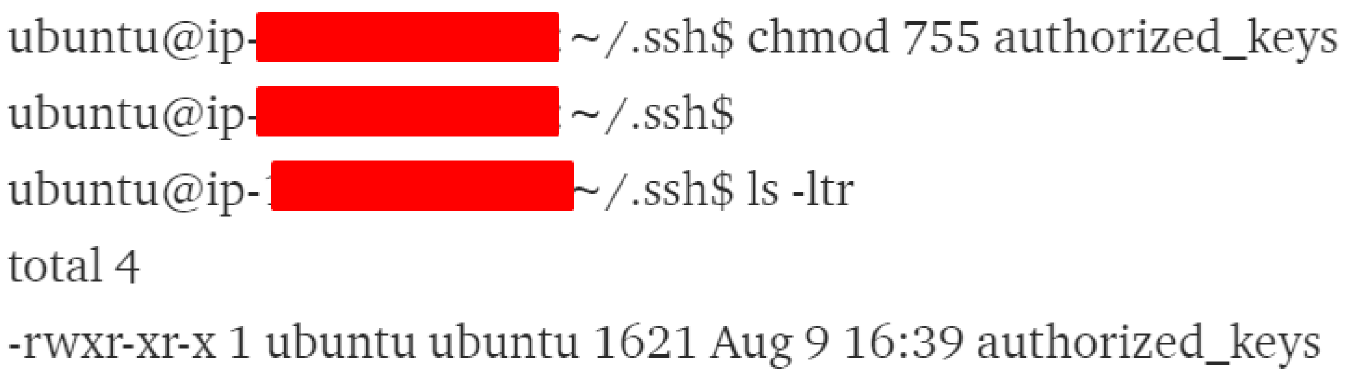
:wq! to save the changes made before exiting.

:q! to exit without saving.

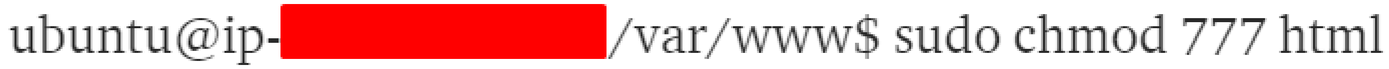
Step 3. Provide permission to the authorized_keys file and /var/www/html folder



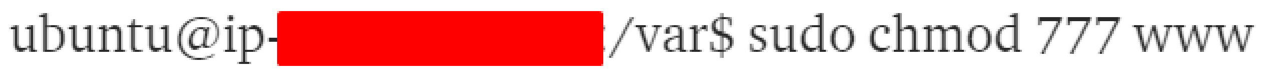
```
ubuntu@ip-[redacted] ~/.ssh$ ls -ltr
total 4
-rw-rw-r—1 ubuntu ubuntu 1621 Aug 9 16:39 authorized_keys
```



```
ubuntu@ip-[redacted]~/.ssh$ chmod 755 authorized_keys
ubuntu@ip-[redacted]~/.ssh$
ubuntu@ip-[redacted]~/.ssh$ ls -ltr
total 4
-rwxr-xr-x 1 ubuntu ubuntu 1621 Aug 9 16:39 authorized_keys
```



```
ubuntu@ip-[redacted]/var/www$ sudo chmod 777 html
```



```
ubuntu@ip-[redacted]/var$ sudo chmod 777 www
```

Similarly, find and give permission to the var folder:

cd/

ls -ltr

chmod -r 777 var

After configuring both the instances, test the connection as follows:

```
[ubuntu@ip-xxx-xx-xx-xx]$ ssh <ip_address_B>
```

Use scp command to copy the test.html from 'A' to /var/www/html of 'B'

scp filename user@<hostname>:/directory

```
[ubuntu@ip-xxx-xx-xx-xx]$ scp test.html ubuntu@ip-xxx-xx-xx-x:/var/www/html
```

Conngratulations!! You have successfully established SSH between two AWS EC2 Ubuntu 16.04 instances. Moreover, now you know to copy files securely between two ec2 instances.

AWS EC2 SCP SSH UBUNTU