How to connect to AWS GUI using SSH tunneling

1 What is SSH ?

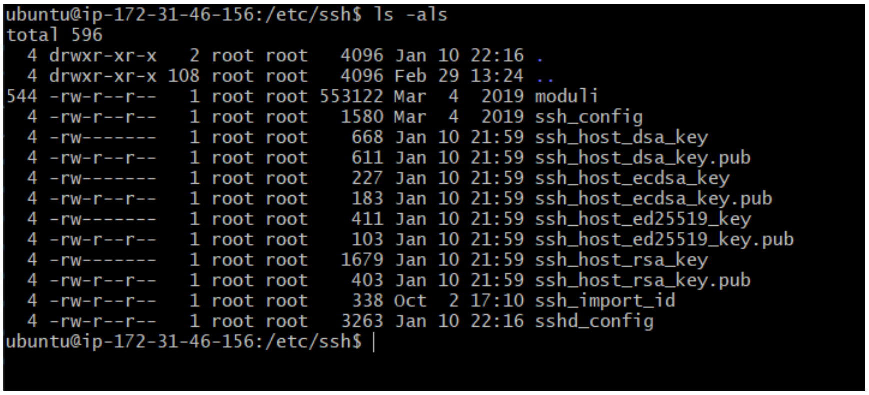
2 Linux sshd daemon ?

2.1 checking the status of the sshd

2.2 controlling the behavior of the sshd on the server

In /etc/ssh, there are some configuration files which change the behavior of the server and the client as well

- ssh_config file is affecting the client.
- sshd_config file is affecting the server.



```
ubuntu@ip-172-31-46-156:/etc/ssh$ ls -als
total 596
  4 drwxr-xr-x   2 root root   4096 Jan 10 22:16 .
  4 drwxr-xr-x 108 root root   4096 Feb 29 13:24 ..
544 -rw-r--r--   1 root root 553122 Mar  4  2019 moduli
  4 -rw-r--r--   1 root root   1580 Mar  4  2019 ssh_config
  4 -rw-------   1 root root    668 Jan 10 21:59 ssh_host_dsa_key
  4 -rw-r--r--   1 root root    611 Jan 10 21:59 ssh_host_dsa_key.pub
  4 -rw-------   1 root root    227 Jan 10 21:59 ssh_host_ecdsa_key
  4 -rw-r--r--   1 root root    183 Jan 10 21:59 ssh_host_ecdsa_key.pub
  4 -rw-------   1 root root    411 Jan 10 21:59 ssh_host_ed25519_key
  4 -rw-r--r--   1 root root    103 Jan 10 21:59 ssh_host_ed25519_key.pub
  4 -rw-------   1 root root   1679 Jan 10 21:59 ssh_host_rsa_key
  4 -rw-r--r--   1 root root    403 Jan 10 21:59 ssh_host_rsa_key.pub
  4 -rw-r--r--   1 root root    338 Oct  2 17:10 ssh_import_id
  4 -rw-r--r--   1 root root   3263 Jan 10 22:16 sshd_config
ubuntu@ip-172-31-46-156:/etc/ssh$ |
```

let's speak about the sshd configuration file, and some of its configuration.

-

2.3 passwordless login

1. Create a new key-pair

Listing 1: Generate a new pair

```
aramadan@CAI1-L11666 MSYS ~
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key
    (/home/aramadan/.ssh/id_rsa): aws-ubuntu
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in aws-ubuntu.
Your public key has been saved in aws-ubuntu.pub.
The key fingerprint is:
SHA256:TiBNm2edovcpRqclZ2MXFBVaiGBnoGZobBjCpwFurRQ
    aramadan@CAI1-L11666
The key's randomart image is:
+---[RSA 3072]----+
|=E.  . +oo.o++. |
|.+o= + = +.o.o |
| +=.* O + o o  |
|o..o + = .  .  |
| .   . S B .   |
|      = X +    |
|       = o     |
|      . .      |
|               |
+----[SHA256]-----+

aramadan@CAI1-L11666 MSYS ~
$ ls -altr | grep aws
-rw-r--r-- 1 aramadan VNET+Group(513) 2610 Feb 29
    16:24 aws-ubuntu
-rw-r--r-- 1 aramadan VNET+Group(513) 574 Feb 29
    16:24 aws-ubuntu.pub
```

2. Copy the public key to the server using *scp*

Listing 2: exchange the public key with the server

```
$ scp /home/aramadan/aws-ubuntu.pub
    ubuntu@ec2-3-14-88-240.us-east-2.compute.amazonaws.com:~
ubuntu@ec2-3-14-88-240.us-east-2.compute.amazonaws.com's
    password:
aws-ubuntu.pub                           100%  574
    2.7KB/s  00:00
```

3. append the new public key to the **/.ssh/authorized_keys** file

Listing 3: Append it as authorized key

```
ubuntu@ip-172-31-46-156:~$ cat aws-ubuntu.pub >>
    ~/.ssh/authorized_keys

# and make sure that authorized_keys has 755
    permissions
ubuntu@ip-172-31-46-156:~$ ls -la .ssh
total 12
drwx------ 2 ubuntu ubuntu 4096 Jan 10 21:59 .
drwxr-xr-x 18 ubuntu ubuntu 4096 Feb 29 19:06 ..
-rw------- 1 ubuntu ubuntu 965 Feb 29 18:38
    authorized_keys

# Otherwise give them to it, otherwise it will not
    work and try!
ubuntu@ip-172-31-46-156:~$ chmod 755
    .ssh/authorized_keys
```

4. You can copy and append the keys in steps 2, and 3 directly using *ssh-copy-id* command

   You will give it the private-key, and it will login the fist time and automatically append the authorized_keys file for you.

   Listing 4: Configure the sshd

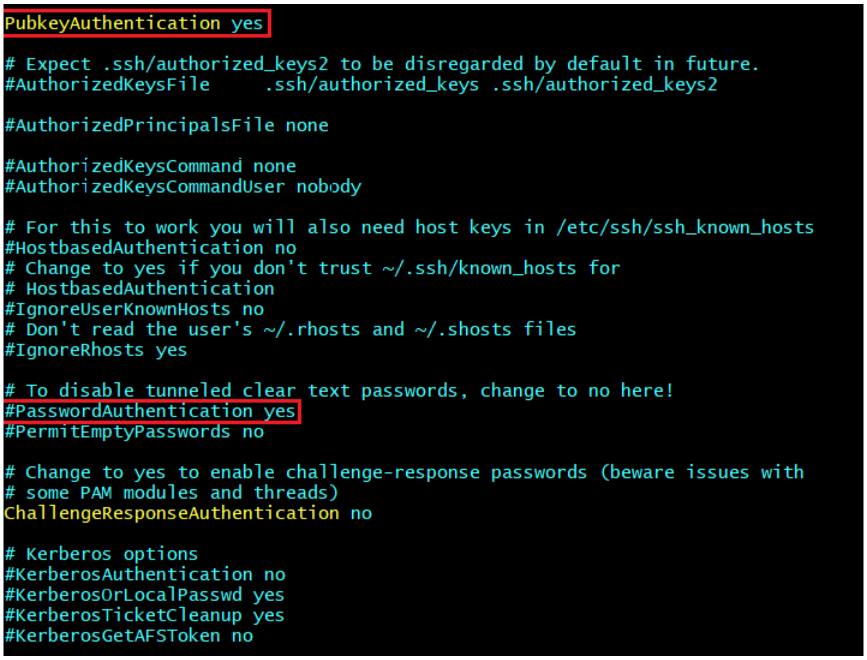
   ```
   $ ssh-copy-id -i aws-ubuntu
       ubuntu@ec2-3-14-88-240.us-east-2.compute.amazonaws.com
   ```

5. edit the *sshd_config* file to the the new key, and restart the sshd daemon to apply the new configuration

   Listing 5: Configure the sshd

   ```
   ubuntu@ip-172-31-46-156:~$ sudo vi
       /etc/ssh/sshd_config
   ```

   Here, you need to comment the *PasswordAuthentication yes*, and uncomment the *PubkeyAuthentication yes*.



```
PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
```

6. communicate to the server using your private key

Listing 6: Start a new communication

```
# communicate using the private key "aws-ubuntu"
$ ssh -i aws-ubuntu
    ubuntu@ec2-3-14-88-240.us-east-2.compute.amazonaws.com
```

Note: you need to make sure the following permissions on the following client-side .ssh is given at least 700, and private-key file is given 600

2.4 ssh-tunnels

1. Create your own AWS EC2 Linux Image (You can have a free tier)

2. Connect to your instance using SSH, You can use putty or any SSH client

3.

Youtube video

3 Working with text processing using grep, sed and awk

grep gnu regular expression parser sed Stream editor awk