

Aim :-

Using SSH Services to connect remote Linux Server with login credentials and without login credentials .



Created By :-

Name :- Chander Mohan Meena

Domain :- Btech cse (cloud computing and full stack development)

College :- Poornima University

Index :-

- Overview of ssh service
- Purpose of connecting remote server on linux
- Procedure to link ssh services on two different systems of linux

Prerequisites :-

Knowledge about the basics of Linux (Kali, Ubuntu, RHEL ,etc) commands, systemctl and sshd service.

SSH Service :-

What is SSH?

- SSH constitutes a cryptographic network protocol designed to enable secure communication between two systems over potentially insecure networks.
- It's widely used for remote access to servers and secure file transfers between computers.

Secure Shell (SSH) is a cryptographic network protocol used for an encrypted connection between a client and a server. The ssh client creates a secure connection to the SSH server on a remote machine. The encrypted connection can be used to execute commands on the server, X11 tunneling, port forwarding, and more

What is an SSH Key?

SSH is an authentication mechanism called public key authentication that is based on cryptographic keys. SSH replaced the insecure `.rhosts` authentication, which was vulnerable to active network-level attacks and improved network security. The basics of SSH are:

- SSH keys have two parts.
 - user keys: The keys used for user authentication .
 - host keys: Used for authenticating hosts.
- One or more public keys may be configured as authorized keys;
- A private key corresponding to an authorized key serves as authentication to the server.
- Both authorized keys and private keys are stored in the `.ssh` directory in a user's home directory.
- These keys function like super strong passwords, but they cannot be stolen from the network.
 - The private key can be encrypted locally

What is Open SSH ?

OpenSSH (also known as OpenBSD Secure Shell) is a suite of secure networking utilities based on the Secure Shell (SSH) protocol. It provides a secure channel over an unsecured network in a client–server architecture .

Connecting to a remote server in Linux serves several essential purposes:

1.Remote Administration:

- System administrators can manage servers, perform maintenance tasks, and troubleshoot issues without being physically present.
- SSH (Secure Shell) provides a secure way to access the server remotely.

2.File Transfer:

- You can transfer files between your local machine and the remote server using tools like `scp` (secure copy) or `rsync`.
- Example:

```
$ scp local-file.txt user@remote-host:/path/to/destination/
```

3.Running Commands Remotely:

- Execute commands on the remote server without logging in via a graphical interface.
- Useful for automation, batch processing, or running scripts.
- Example:

```
$ ssh user@remote-host 'ls -l /var/www'
```

4.Accessing Services:

- Connect to services (e.g., web servers, databases) hosted on the remote server.
- Port forwarding allows you to access services running on the server from your local machine.
- Example:

```
$ ssh -L 8080:localhost:80 user@remote-host
```

5.Secure Communication:

- SSH encrypts data during transmission, ensuring confidentiality and integrity.
- Protects sensitive information (passwords, data) from interception.

Remember, connecting remotely allows efficient management, collaboration, and resource utilization across distributed systems.

Procedure to link ssh services on two different systems of linux :-

1. Here I am using two machines in Vmware.
2. First is kali linux and second one is Ubuntu OS
3. Both are connected with same IP but we can use different IP as well.
4. Now let's start the process for establishing the connectivity between them and we will control the kali system's terminal from our ubuntu system remotely.

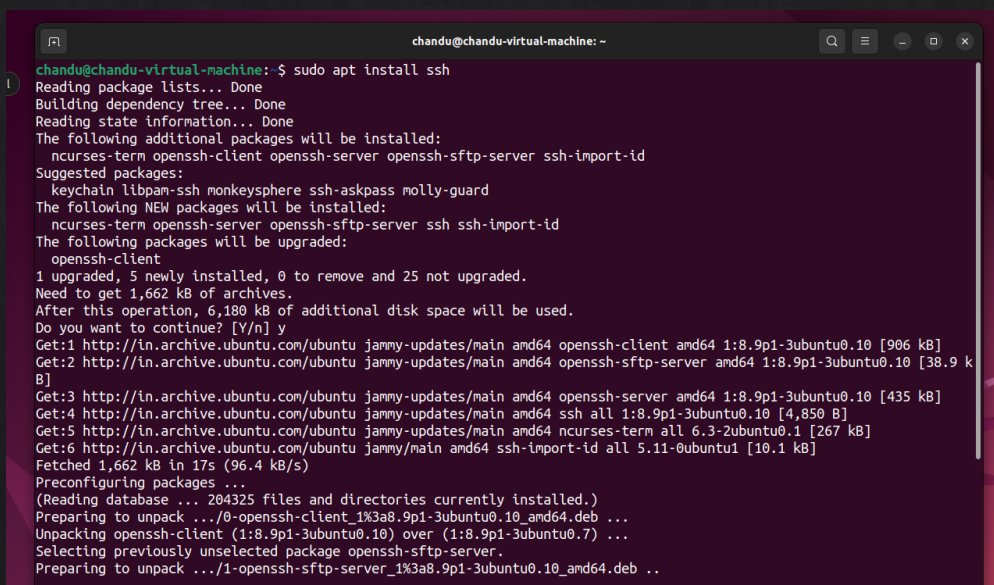
Step 1 :- Here I am installing the SSH services in both the system with the help of <<< **sudo apt install ssh** >>> command

For starting the services we can use the <<< **sudo systemctl start ssh** >>> command.

For stoping the services we can use the <<< **sudo systemctl stop ssh** >>> command.

For enabling the services we can use the <<< **sudo systemctl enable ssh** >>> similarly for disabling <<< **sudo systemctl disable ssh** >>>

So we have started the SSH services in both of the system here the screenshot of executing commands :



```
chandu@chandu-virtual-machine: ~  
chandu@chandu-virtual-machine:~$ sudo apt install ssh  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  ncurses-term openssh-client openssh-server openssh-sftp-server ssh-import-id  
Suggested packages:  
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard  
The following NEW packages will be installed:  
  ncurses-term openssh-server openssh-sftp-server ssh ssh-import-id  
The following packages will be upgraded:  
  openssh-client  
1 upgraded, 5 newly installed, 0 to remove and 25 not upgraded.  
Need to get 1,662 kB of archives.  
After this operation, 6,180 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-client amd64 1:8.9p1-3ubuntu0.10 [906 kB]  
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.10 [38.9 kB]  
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-3ubuntu0.10 [435 kB]  
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ssh all 1:8.9p1-3ubuntu0.10 [4,850 B]  
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term all 6.3-2ubuntu0.1 [267 kB]  
Get:6 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.11-0ubuntu1 [10.1 kB]  
Fetched 1,662 kB in 17s (96.4 kB/s)  
Preconfiguring packages ...  
(Reading database ... 204325 files and directories currently installed.)  
Preparing to unpack .../0-openssh-client_1%3a8.9p1-3ubuntu0.10_amd64.deb ...  
Unpacking openssh-client (1:8.9p1-3ubuntu0.10) over (1:8.9p1-3ubuntu0.7) ...  
Selecting previously unselected package openssh-sftp-server.  
Preparing to unpack .../1-openssh-sftp-server_1%3a8.9p1-3ubuntu0.10_amd64.deb ..
```

After installation and starting / enabling the services now we are ready to perform other steps

Step 2:- Firstly , let's check the status of our services with the <<< **systemctl status sshd** >>> command. where sshd is a server (like a web server serving https) and SSH is a client (think of a web browser).

We can also use <<< **systemctl status ssh** >>>.

Here , all services are active and enabled.

```
chandu@chandu-virtual-machine:~$ systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-07-04 21:34:57 IST; 2min 15s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 9949 (sshd)
    Tasks: 1 (limit: 4554)
   Memory: 1.7M
      CPU: 35ms
   CGroup: /system.slice/ssh.service
           └─9949 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jul 04 21:34:57 chandu-virtual-machine systemd[1]: Starting OpenBSD Secure Shell server...
Jul 04 21:34:57 chandu-virtual-machine sshd[9949]: Server listening on 0.0.0.0 port 22.
Jul 04 21:34:57 chandu-virtual-machine sshd[9949]: Server listening on :: port 22.
Jul 04 21:34:57 chandu-virtual-machine systemd[1]: Started OpenBSD Secure Shell server.
chandu@chandu-virtual-machine:~$
```

Step 3:- Now we have to take the IP from our kali system with the <<< **ifconfig** >>> command

```
chandu@chandu-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.217.128 netmask 255.255.255.0 broadcast 192.168.217.255
    inet6 fe80::9450:af2:b2f9:f668 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:84:73:d2 txqueuelen 1000 (Ethernet)
    RX packets 55821 bytes 74157742 (74.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22532 bytes 1547538 (1.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

or

```
(kali㉿kali)-[~]  
$ ifconfig  
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
    ether 02:42:5a:c8:4d:34 txqueuelen 0 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.217.129 netmask 255.255.255.0 broadcast 192.168.217.255  
    inet6 fe80::7f33:c71c:26fe:7664 prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:bb:83:ed txqueuelen 1000 (Ethernet)  
    RX packets 2958 bytes 1991516 (1.8 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 1883 bytes 339289 (331.3 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 27 bytes 4214 (4.1 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 27 bytes 4214 (4.1 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
(kali㉿kali)-[~]  
$
```

After taking the IP now we can establish the connection .

Step 4:- now we will login into the system using `ssh user@server-name/ IP` <<< `ssh kali@192.168.***.***` >>> command .

```
chandu@chandu-virtual-machine:~$ ssh kali@192.168.217.129  
The authenticity of host '192.168.217.129 (192.168.217.129)' can't be established.  
ED25519 key fingerprint is SHA256:dv9lYngB5uFnsc+y02ddbGLL2A0BRI7Q50x7WTwfgwk.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Then type << yes >>

After entering yes , then give the system password of kali
And the connection will be established and we can now use/control the whole kali system in Ubuntu system.

Here for proving I simply used <<< **ls** >>> to show the directory's and files in the system and we can do modifications in it also .

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.217.129' (ED25519) to the list of known hosts.
kali@192.168.217.129's password:
Linux kali 6.3.0-kali1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.3.7-1kali1 (2023-06-29) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jun 17 12:01:22 2024 from 192.168.213.130
kali
(kali@kali)-[~]
$ ls
alphanat.txt  data      Documents  hardlink-to-new  new_project1  project  t1      test2      Videos
backup        demo     Downloads  msg1.sh          new.txt       Public   Templates test3      workshop
class.txt     Desktop  g          Music            Pictures      softlink-new  test  today.txt
```

This the the list content of the main kali system which we are using as server remotely.

```
(kali@kali)-[~]
$ ls
alphanat.txt  class.txt  demo      Documents  g          msg1.sh  new_project1  Pictures  Public  t1
backup        data      Desktop   Downloads  hardlink-to-new  Music    new.txt       project  softlink-new  Templ
```

For disconnecting the server simply we can use <<< **exit** >>> command .

```
(kali@kali)-[~]
$ exit
Connection to 192.168.217.129 closed.
chandu@chandu-virtual-machine:~$
```


- Here the connection was done but if we want to connect regular bases so it will become Inconvenient for entering security keys .
- So we can simply set up the system for login without any credentials.
- Now let's set up the system with following commands.

Step 5 :- Here we have to generate the ssh key by using <<< **ssh-keygen** >>>

```
chandu@chandu-virtual-machine:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/chandu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/chandu/.ssh/id_rsa
Your public key has been saved in /home/chandu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:br8zceo/zUozLIrT0iBxuD6s3y5ixw5eoRw8Bhs5f/c chandu@chandu-virtual-machine
The key's randomart image is:
+----[RSA 3072]-----+
|
|  .
| =.
| .Bo .
| .o*+o S
| .o.=++ ....
| .++..oo E+=o
| .+o*+.o+o.oO
| ..=*+.+=oo
|
+-----[SHA256]-----+
chandu@chandu-virtual-machine:~$
```

- After generating the public/private RSA key pair.
- The file was saved in `.ssh` directory in Ubuntu system.

Step 6 :- So we have to list all the files by using <<< **ls -all** >>> command

```

chandu@chandu-virtual-machine:~$ ls -all
total 96
drwxr-x--- 18 chandu chandu 4096 Jul  4 21:58 .
drwxr-xr-x  3 root   root   4096 Jun 21 12:00 ..
-rw-----  1 chandu chandu 1693 Jul  4 21:33 .bash_history
-rw-r--r--  1 chandu chandu  220 Jun 21 12:00 .bash_logout
-rw-r--r--  1 chandu chandu 3771 Jun 21 12:00 .bashrc
drwx----- 13 chandu chandu 4096 Jun 26 20:12 .cache
drwx----- 12 chandu chandu 4096 Jun 22 22:32 .config
-rw-r--r--  1 root   root   1036 Jun 22 23:05 demo.com
drwxr-xr-x  2 chandu chandu 4096 Jun 21 12:34 Desktop
drwxr-xr-x  2 chandu chandu 4096 Jun 21 12:34 Documents
drwxr-xr-x  2 chandu chandu 4096 Jun 21 12:34 Downloads
-rw-----  1 chandu chandu   20 Jul  4 21:37 .lessht
drwx-----  3 chandu chandu 4096 Jun 21 12:34 .local
drwx-----  3 chandu chandu 4096 Jun 22 22:36 .mozilla
drwxr-xr-x  2 chandu chandu 4096 Jun 21 12:34 Music
drwxr-xr-x  2 chandu chandu 4096 Jun 21 12:34 Pictures
drwxrwxr-x  2 chandu chandu 4096 Jun 23 12:43 prasang
-rw-r--r--  1 chandu chandu  807 Jun 21 12:00 .profile
drwxr-xr-x  2 chandu chandu 4096 Jun 21 12:34 Public
drwx-----  5 chandu chandu 4096 Jun 22 22:14 snap
drwx-----  2 chandu chandu 4096 Jul  4 22:21 .ssh
-rw-r--r--  1 chandu chandu    0 Jun 21 12:38 .sudo_as_admin_successful
drwxr-xr-x  2 chandu chandu 4096 Jun 21 12:34 Templates
drwx-----  6 chandu chandu 4096 Jun 22 22:36 .thunderbird
drwxr-xr-x  2 chandu chandu 4096 Jun 21 12:34 Videos
chandu@chandu-virtual-machine:~$

```

Step 7 :- with the help of <<< **cd .ssh** >>> command we can simply open the .ssh file .

```

chandu@chandu-virtual-machine:~$ cd .ssh
chandu@chandu-virtual-machine:~/ssh$ ls
id_rsa  id_rsa.pub  known_hosts  known_hosts.old

```

Step 8 :- now we have to copy the .ssh file and transfer it to kali system with the help of <<< `ssh-copy-id kali@192.168.***.*29` >>> command .

```
chandu@chandu-virtual-machine:~/ssh$ ssh-copy-id kali@192.168.217.129
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that
are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is
to install the new keys
kali@192.168.217.129's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'kali@192.168.217.129'"
and check to make sure that only the key(s) you wanted were added.
```

Step 9 :- Then enter the credentials of the system .

- Now we are ready to login and logout from the system without any authentication process .

```
chandu@chandu-virtual-machine:~/ssh$ ssh 'kali@192.168.217.129'
Linux kali 6.3.0-kali1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.3.7-1kali1 (2023-06-29) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jul  4 21:58:40 2024 from 192.168.217.128
kali
(kali@kali)-[~]
$ ls
alphabat.txt  demo      g          new_project1  Public    test      Videos
backup        Desktop  hardlink-to-new  new.txt      softlink-new  test2     workshop
class.txt     Documents msg1.sh       Pictures     t1            test3
data          Downloads Music         project      Templates     today.txt
```

These are some basics steps for set-up the ssh in our system .

THANK YOU