SSH Linux | Linux ssh command

In Linux, **ssh** is a protocol, which stands for S**ecure Shell** or S**ecure Socket Shell.** The secure shell is useful for security while connecting to a remote server. The **ssh command uses a ssh protocol**, which is a secure protocol, as the data transfer between the client and the host takes place in encrypted form. It transfers the input through the client to the host and returns the output transferred by the host. It executes through TCP/IP port 22.

The encrypted connection is also used to run the commands on a **Linux server**, **port forwarding**, **tunnelling**, and more.

There are lots of SSH clients that are available for both commercial and free. The OpenSSH is its most widely used client. It is available for all the most used platforms such as Windows, Linux, macOS, OpenBSD, and more.

Syntax:

1. ssh user_name@host(IP/Domain_name)

Components of ssh command

The ssh command consists of three different types of components:

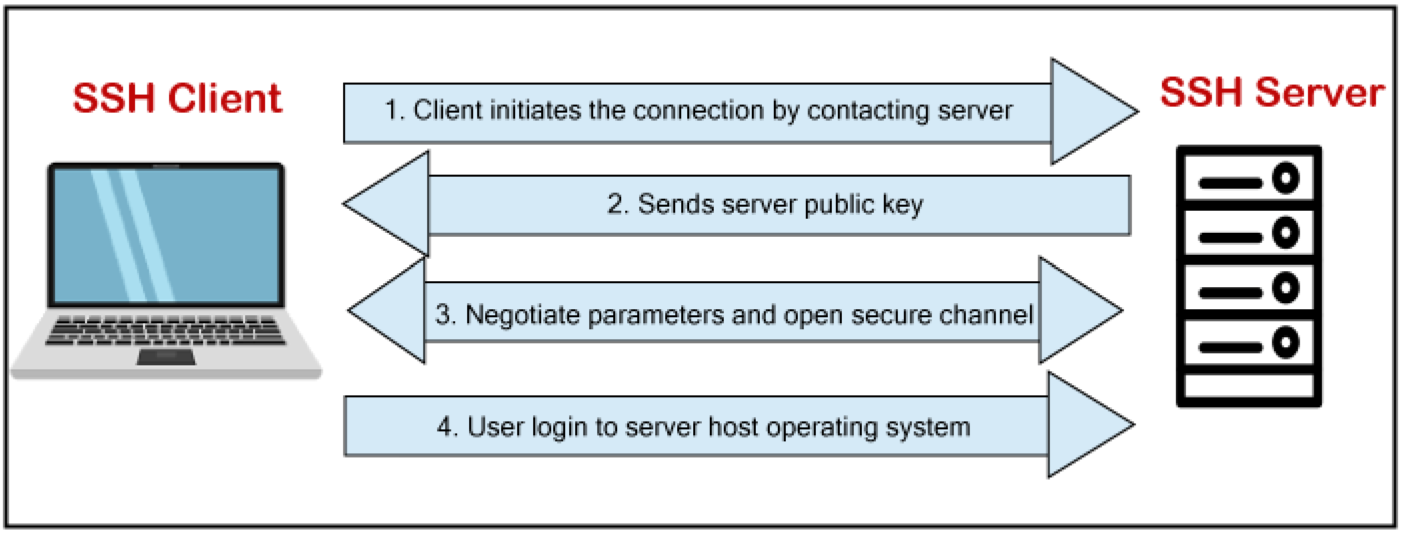
❍**ssh command:** It instructs the machine to create a secure encrypted connection with the host system.

❍**User name:** User name is the name of the Linux user, which is being accessed by the host machine.

❍**Host:** A host is a machine that is being accessed by the user, such as a computer or a router. A domain name or an IP address also refers as Host.

How SSH works?

To establish an SSH connection, we need two primary components; a client and a host, which can be a server, domain name, IP address, and more. Also, we require a ssh client to connect with another computer or server. The client uses the specified host information to establish the connection; if the provided credential verified, it will establish an encrypted connection.


SSH Client
1. Client initiates the connection by contacting server
SSH Server
2. Sends server public key
3. Negotiate parameters and open secure channel
4. User login to server host operating system

The server (Host) contains an SSH process that is ready to take a request for the client connection through a TCP/IP port. Once the client initiates a connection, the host responds with the necessary information and exchanges the credentials.

If the provided information is verified, the SSH protocol establishes a new connection for the available environment.

The default SSH protocol version for SSH server and SSH client communication is version 2.

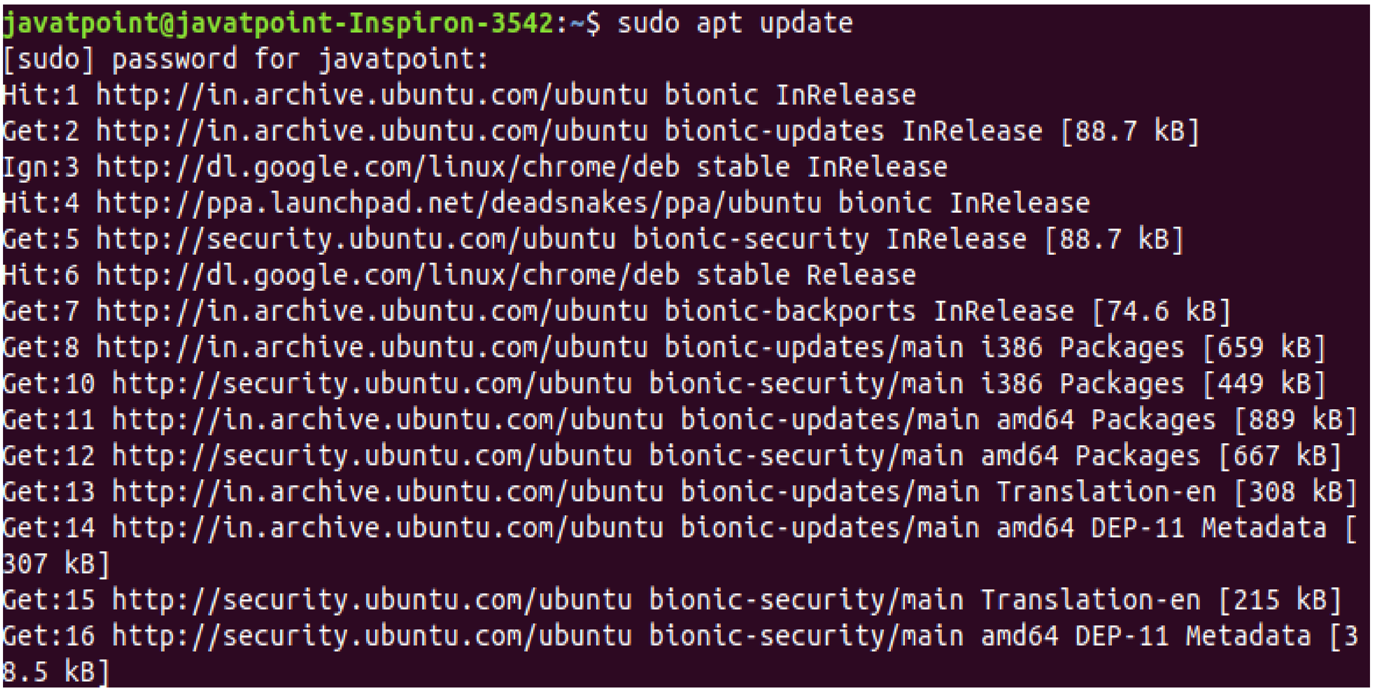
Install OpenSSH client on Linux (Ubuntu)

The OpenSSH client is a connectivity tool for the systems to connect two systems with the ssh protocol. It is also called as ssh and can be invoked from the Linux terminal. This client package contains other SSH utilities like sftp, scp, and ssh that are installed by default with the ssh command. It performs remote operations using these ssh utilities.

The OpenSSH client comes preinstalled with most Linux distributions. If any Linux system does not have the ssh client, we can install it manually by using the package manager.

To install the OpenSSH client, execute the below command:

1. sudo apt update
   The above sudo command will update the package of the Linux system. Consider the below snap of the output:

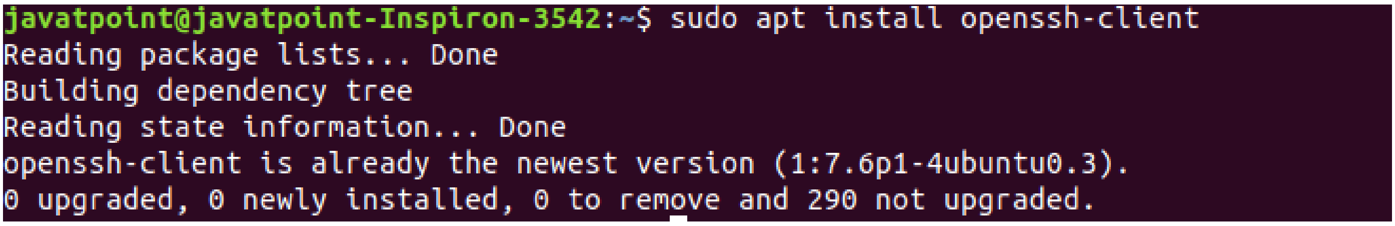


```
javatpoint@javatpoint-Inspiron-3542:~$ sudo apt update
[sudo] password for javatpoint:
Hit:1 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Ign:3 http://dl.google.com/linux/chrome/deb stable InRelease
Hit:4 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:6 http://dl.google.com/linux/chrome/deb stable Release
Get:7 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [659 kB]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/main i386 Packages [449 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [889 kB]
Get:12 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [667 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [308 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Metadata [
307 kB]
Get:15 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [215 kB]
Get:16 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [3
8.5 kB]
```

After updating the Linux system, execute the below command to install the OpenSSH client:

1. sudo apt install OpenSSH-client

The above command will install the latest package of the OpenSSH client. Consider the below output:



```
javatpoint@javatpoint-Inspiron-3542:~$ sudo apt install openssh-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-client is already the newest version (1:7.6p1-4ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 290 not upgraded.
```

As we can see from the above output, a daemon process is running to install the OpenSSH client. As in our machine, OpenSSH client is already installed, so it has displayed the message 'openssh-client is already the newest version.'

Note: The macOS carries the Openssh client by default.

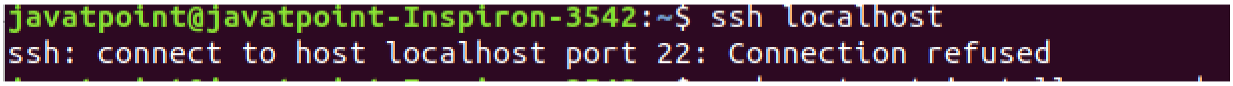
Install OpenSSH server on Linux(Ubuntu)

To make an SSH connection, we need to have the server-side part of the SSH software in our machine.

To check the installation status of the server, open the terminal and execute the below command:

1. ssh localhost

If our machine does not have the server tool kit of the OpenSSH client, then it will display the output as follows:

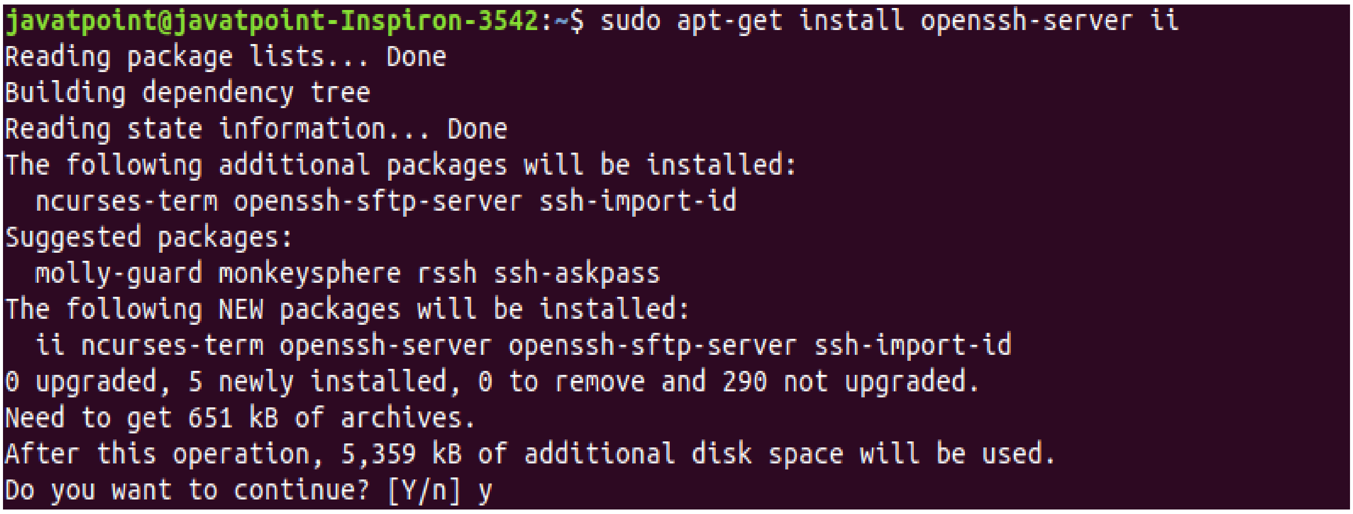


```
javatpoint@javatpoint-Inspiron-3542:~$ ssh localhost
ssh: connect to host localhost port 22: Connection refused
```

In the above case, we have to install the OpenSSH server.

To install the SSH server, execute the below command:

1. sudo apt-get install openssh-server ii
   The above command will prompt for the system administrative password, type the password, and press **ENTER** key to start the installation process. Next, it will confirm the installation type **'y'** key and press ENTER key. Consider the below output:

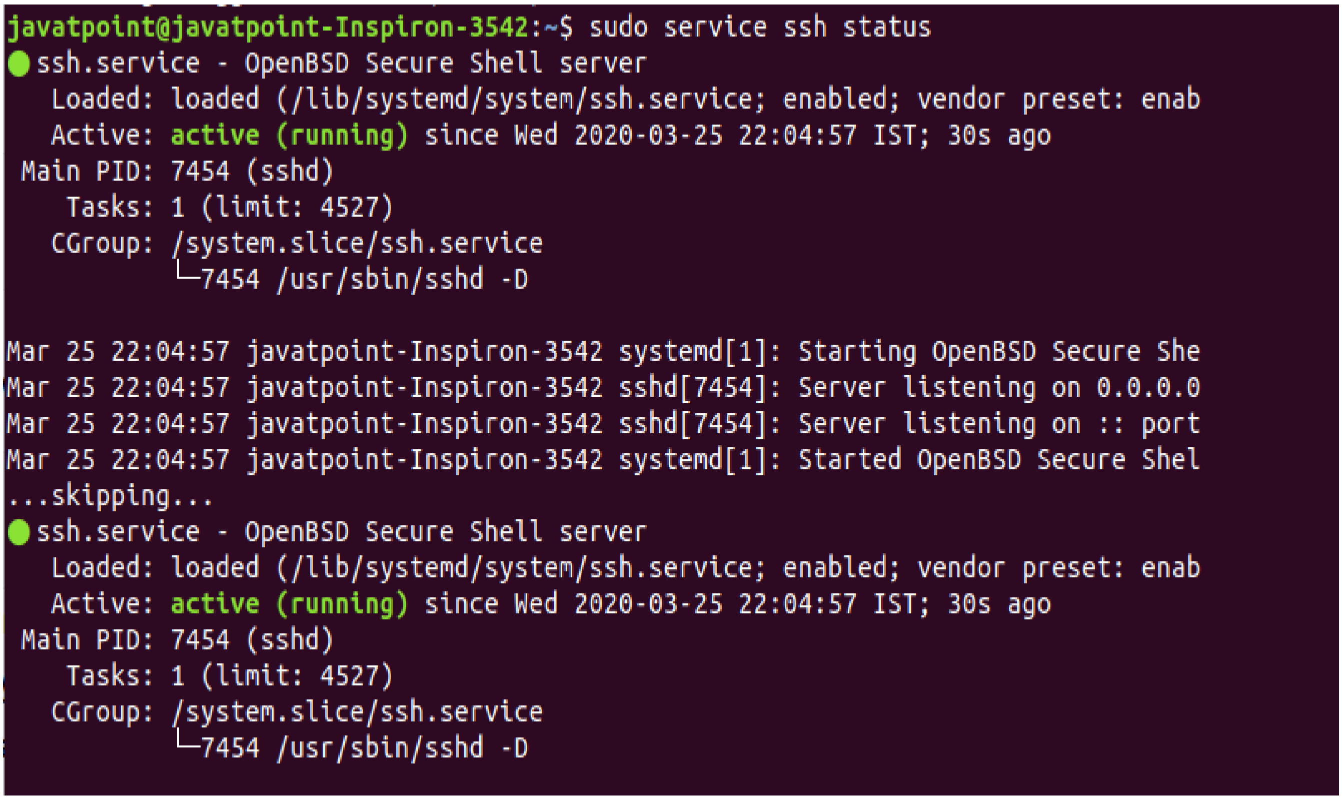


```
javatpoint@javatpoint-Inspiron-3542:~$ sudo apt-get install openssh-server ii
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere rssh ssh-askpass
The following NEW packages will be installed:
  ii ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 5 newly installed, 0 to remove and 290 not upgraded.
Need to get 651 kB of archives.
After this operation, 5,359 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

After confirming the installation, a daemon process will begin and install the OpenSSH server on your machine.

To verify the installation, execute the below command:

1. sudo service ssh status
   The above command will display the status of the installation. If the installation is successful, it will display the output as follows:



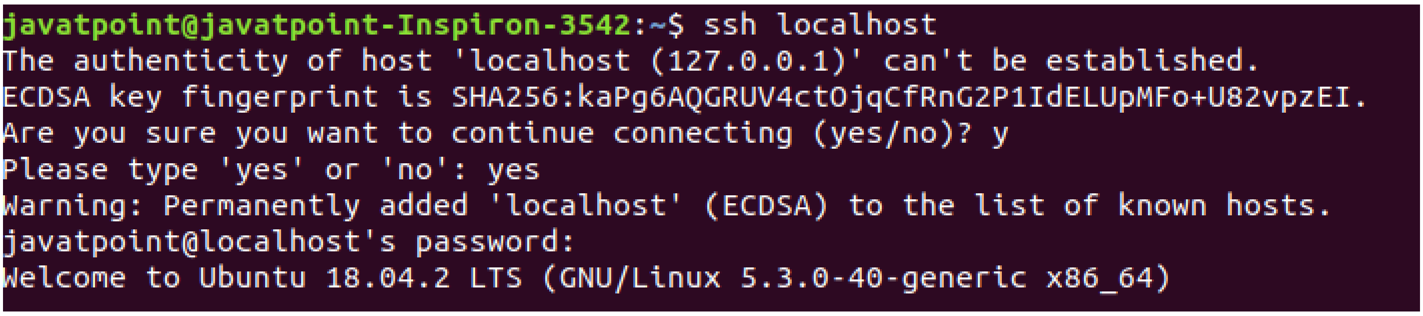
```
javatpoint@javatpoint-Inspiron-3542:~$ sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enab
   Active: active (running) since Wed 2020-03-25 22:04:57 IST; 30s ago
 Main PID: 7454 (sshd)
    Tasks: 1 (limit: 4527)
   CGroup: /system.slice/ssh.service
           └─7454 /usr/sbin/sshd -D

Mar 25 22:04:57 javatpoint-Inspiron-3542 systemd[1]: Starting OpenBSD Secure She
Mar 25 22:04:57 javatpoint-Inspiron-3542 sshd[7454]: Server listening on 0.0.0.0
Mar 25 22:04:57 javatpoint-Inspiron-3542 sshd[7454]: Server listening on :: port
Mar 25 22:04:57 javatpoint-Inspiron-3542 systemd[1]: Started OpenBSD Secure Shel
...skipping...
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enab
   Active: active (running) since Wed 2020-03-25 22:04:57 IST; 30s ago
 Main PID: 7454 (sshd)
    Tasks: 1 (limit: 4527)
   CGroup: /system.slice/ssh.service
           └─7454 /usr/sbin/sshd -D
```

There is another way to test the installation by ssh localhost command:

1. ssh localhost

The above command will verify the connectivity type **'yes'** to continue. Consider the below output:



```
javatpoint@javatpoint-Inspiron-3542:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:kaPg6AQGRUV4ctOjqCfRnG2P1IdELUpMFo+U82vpzEI.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
javatpoint@localhost's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 5.3.0-40-generic x86_64)
```

Now, we have successfully installed the OpenSSH server on our machine.

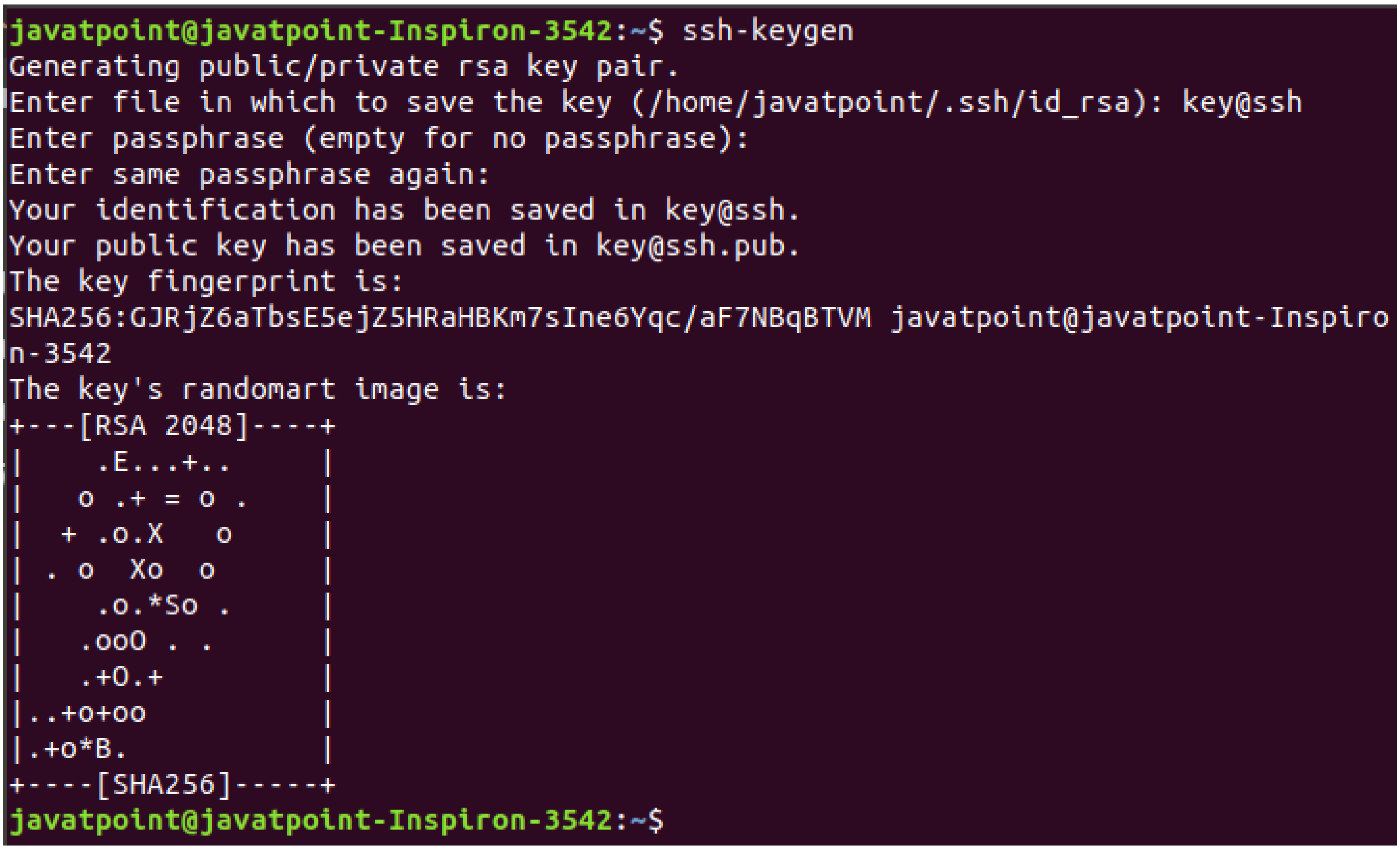
SSH Key Generation

To create a connection with the host client, we need a specific key for an encrypted connection. Logging in to remote host computer by ssh key is more secure than using a password. After logging in the host, computer commands will only work if these commands will be written to the host computer directly.

To generate the ssh key, execute the below command:

1. ssh-keygen

The above command will generate the public and private keys for creating a connection to the host system. Consider the below output:



```
javatpoint@javatpoint-Inspiron-3542:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/javatpoint/.ssh/id_rsa): key@ssh
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in key@ssh.
Your public key has been saved in key@ssh.pub.
The key fingerprint is:
SHA256:GJRjZ6aTbsE5ejZ5HRaHBKm7sIne6Yqc/aF7NBqBTVM javatpoint@javatpoint-Inspiro
n-3542
The key's randomart image is:
+---[RSA 2048]----+
|    .E...+..     |
|   o .+ = o .    |
|  + .o.X   o     |
| . o  Xo  o      |
|    .o.*So .     |
|   .ooO . .      |
|   .+O.+         |
|..+o+oo          |
|.+o*B.           |
+----[SHA256]-----+
javatpoint@javatpoint-Inspiron-3542:~$
```

we can see from the above output, the ssh keys have generated. The **ssh-keygen** command creates two files, **key@ssh,** and **key@ssh.pub,** which contain private and public key, respectively.

It is recommended to hide the private key for security purposes, copy the public key to the remote host. After copying this key to a remote host, we can establish a connection using the SSH key, not by the password.

Techniques of SSH Protocol

The SSH protocol is more secure as compared to other protocols such as **telnet**, and the encryption techniques are quite good than other protocols. There are three major encryption techniques which are used by the SSH. They are as following:

1. **Hashing:** Hashing is an authentication technique that is used to ensure whether the received data is coming from a genuine sender and is unaltered or not. It uses a hash function to generate a hash code from the received data. However, it is not possible to regenerate the data from the hash value. This hash value is verified at the sender's, and receiver's both ends. If it matches, the data is authenticated.

2. **Symmetrical encryption:** This technique generates a single key for encryption as well as decryption. The generated key is distributed among the hosts and clients and creates a secure connection. It is the most basic

encryption technique. It performs its best when the data is encrypted and decrypted on the same machine.

3.**Asymmetric encryption:** The asymmetric encryption technique is considered as more secure than other technologies, as it uses the ssh keys (Public and private keys) for encryption. The public key is distributed to other machines to create a secure connection, while the private key only used by the client machine. The secure connection is established by both the public and private keys.

SSH Commands

The client ssh has many functions for the ssh command, such as **creating a key, configuring a key, opening an SSH server, holding a key for single sign-on, file transfer client,** and more. Some most useful ssh commands are as follows:

- **ssh-keygen:** It is used to create a key pair for establishing a connection and public key authentication.
- **ssh-copy-id:** It is used to configure a public key as a valid user on a server.
- **ssh-agent:** It is used to create an agent to hold private key for single sign-on.
- **ssh-add:** It is a tool to add a key to the agent.
- **scp:** It is a file transfer client that provides an RCP-like command-line interface.
- **sftp:** It is a file transfer command that provides an FTP-like command-line interface.
- **sshd:** It is an OpenSSH server for the Linux system.

- **Options:** There are many command-line options are available to specify the different specification of SSH output. Some useful options are as following:

- **-c:** It is used to specify query class for non-IN data.
- **-C:** It is used to compare SOA records on authoritative nameservers.
- **-d:** This option is considered as equivalent to -v.
- **-i:** It is used for IP6.INT reverse lookups.

- **-l:** It is used to list all hosts in a domain using AXFR.
- **-m:** This option sets the memory debugging flag, such as trace|record|usage.
- **-N:** It is used to change the number of dots allowed before root lookup is done.
- **-r:** It is used to disable recursive processing.
- **-R:** It specifies the number of retries for UDP packets.
- **-s:** It is used for a SERVFAIL response should stop query.
- **-t:** It is used to specify the query type.
- **-T:** It is used to enable the TCP/IP model.
- **-v:** It is used for verbose output.
- **-V:** It is used to print the version number and exit.
- **-w:** It is used to specify the wait forever for a reply.
- **-W:** It is used to specify how long to wait for a reply.
- **-4:** It is used only for IPv4 query transport.
- **-6:** It is used only for IPv6 query transport.

How to connect via SSH

As we have installed the SSH client and server, we can establish a secure connection with other machines. For a secure connection between two machines, they both have ssh client and server installed.

To establish a connection, execute the below command:

1. ssh your_username@host_ip_address

If the user name is verified by the machine that you want to connect, execute the below command:

1. ssh host_ip_address

The above command will ask for the password, type the password, and press ENTER key.

If we are making a connection for the first time, it will ask for the continue connecting; type yes and press Enter. It will add an ECDSA (Elliptical curve Digital Signature Algorithm) key and connect you to a remote server.

You are now eligible to control and manage a remote machine by your terminal. If you face any difficulty in establishing a connection, consider the following points:

- ➢If the provided IP address of the remote machine is valid.
- ➢The port SSH daemon is listening to is not blocked by a firewall or forwarded incorrectly.
- ➢The username and password that you entered are correct.
- ➢SSH software is installed properly.