ADMINISTRATING REMOTE SYSTEM

Remote shell Access using SSH

What Is SSH?

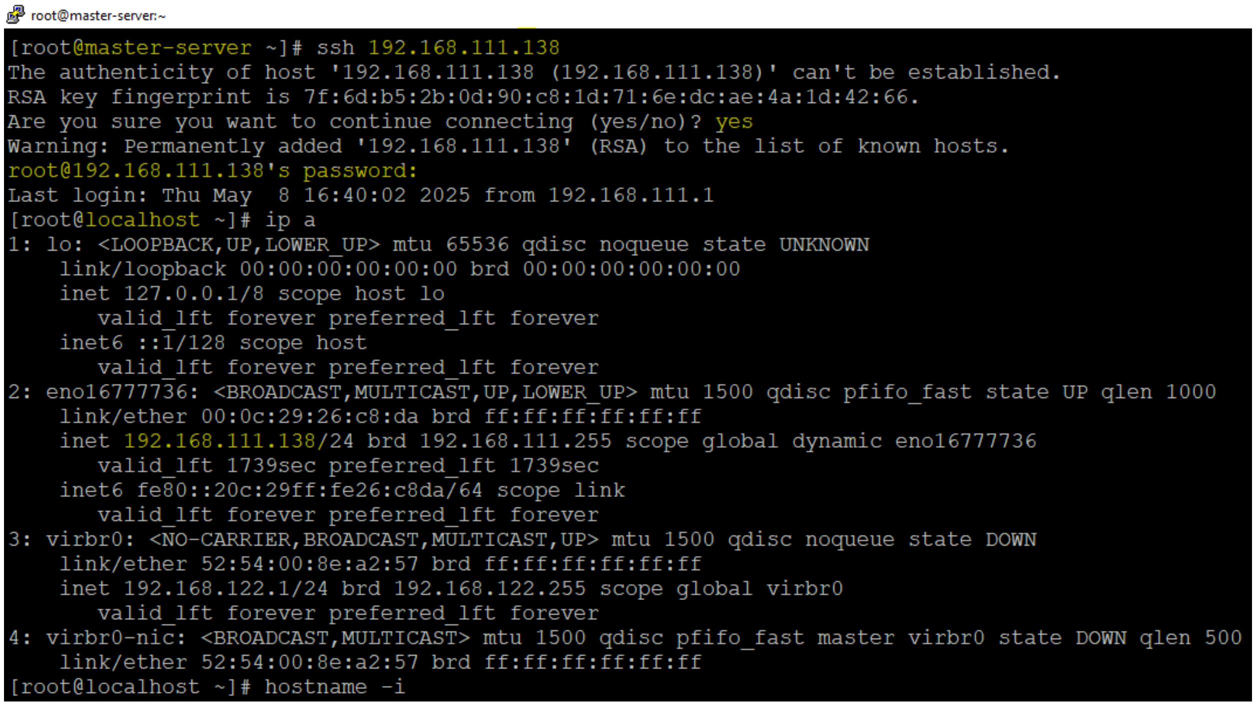
SSH stands for **Secure Shell**. It's a network protocol used to securely connect to a remote computer or server over an unsecured network.

Important configuration files are :

- SSH configuration file is /etc/ssh/sshd_config
- SSH demon or service is sshd

Accessing the remote machine using ssh

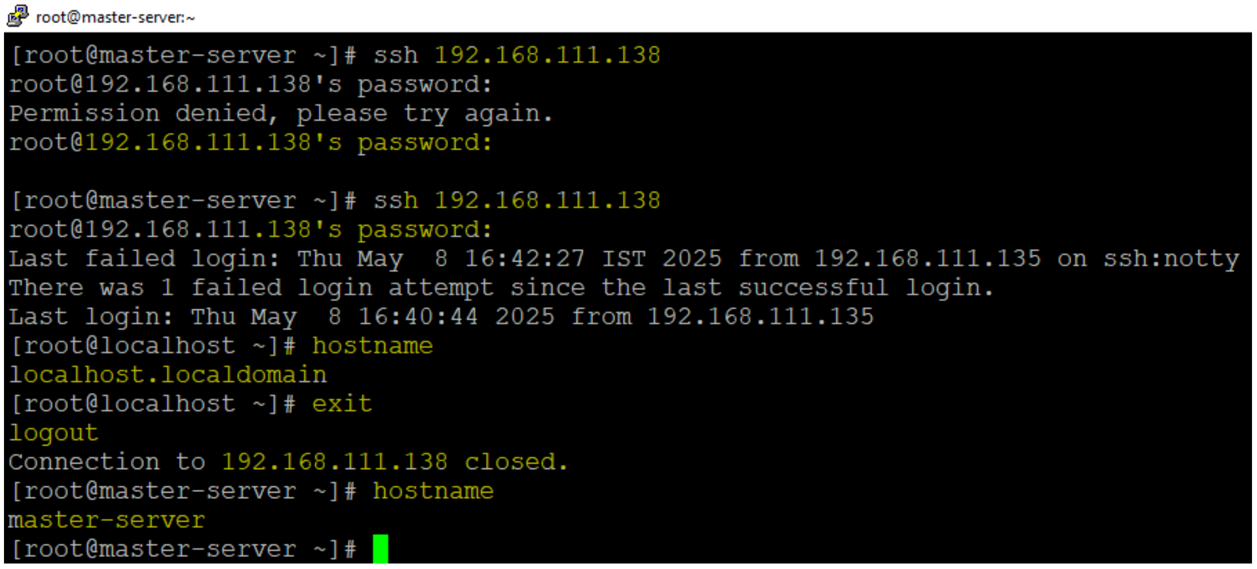
- To access the remote machine using ssh, the syntax is
- #ssh <ipaddress/hostname of the remote machine>
- Note: hostname can only be used when the hostname is saved in /etc/hosts file or, if DNS is configured.
- Ssh 192.168.111.138



```
root@master-server:~
[root@master-server ~]# ssh 192.168.111.138
The authenticity of host '192.168.111.138 (192.168.111.138)' can't be established.
RSA key fingerprint is 7f:6d:b5:2b:0d:90:c8:1d:71:6e:dc:ae:4a:1d:42:66.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.111.138' (RSA) to the list of known hosts.
root@192.168.111.138's password:
Last login: Thu May  8 16:40:02 2025 from 192.168.111.1
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:26:c8:da brd ff:ff:ff:ff:ff:ff
    inet 192.168.111.138/24 brd 192.168.111.255 scope global dynamic eno16777736
       valid_lft 1739sec preferred_lft 1739sec
    inet6 fe80::20c:29ff:fe26:c8da/64 scope link
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 52:54:00:8e:a2:57 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN qlen 500
    link/ether 52:54:00:8e:a2:57 brd ff:ff:ff:ff:ff:ff
[root@localhost ~]# hostname -i
```

- The first time around it will ask you if you wish to add the remote host to a list of known_hosts, go ahead and say yes.
- Enter the password of the remote system correctly, once logged in check hostname and ip address to confirm login.

- ➔ To leave the session, just type exit or logout command and you will be back to your own machine through which you are logged in.


root@master-server:~

```
[root@master-server ~]# ssh 192.168.111.138
root@192.168.111.138's password:
Permission denied, please try again.
root@192.168.111.138's password:

[root@master-server ~]# ssh 192.168.111.138
root@192.168.111.138's password:
Last failed login: Thu May  8 16:42:27 IST 2025 from 192.168.111.135 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Thu May  8 16:40:44 2025 from 192.168.111.135
[root@localhost ~]# hostname
localhost.localdomain
[root@localhost ~]# exit
logout
Connection to 192.168.111.138 closed.
[root@master-server ~]# hostname
master-server
[root@master-server ~]#
```

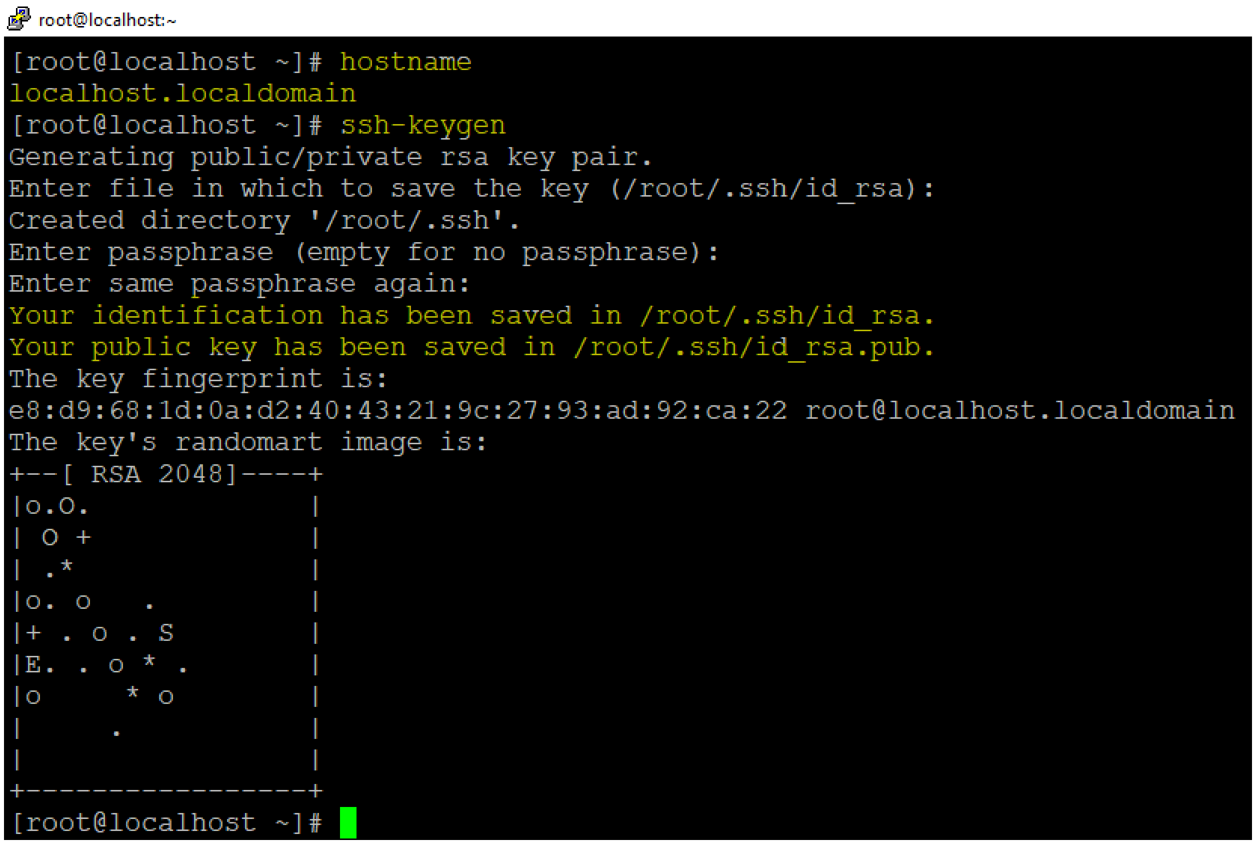
Password less login using SSH keys:

With password less authentication we can login to other server without giving the password of the server. To achieve this we need to place the public key each other on the systems.

So below is the process to do password less login

Generating SSH key pair:

- ➔ To generate the SSH key pair, the syntax is
- ➔ #ssh-keygen



root@localhost:~

```
[root@localhost ~]# hostname
localhost.localdomain
[root@localhost ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
e8:d9:68:1d:0a:d2:40:43:21:9c:27:93:ad:92:ca:22 root@localhost.localdomain
The key's randomart image is:
+--[ RSA 2048]----+
|o.O.             |
| O +             |
| .*              |
|o. o   .         |
|+ . o . S        |
|E. . o * .       |
|o     * o        |
|     .           |
|                 |
+-----------------+
[root@localhost ~]#
```

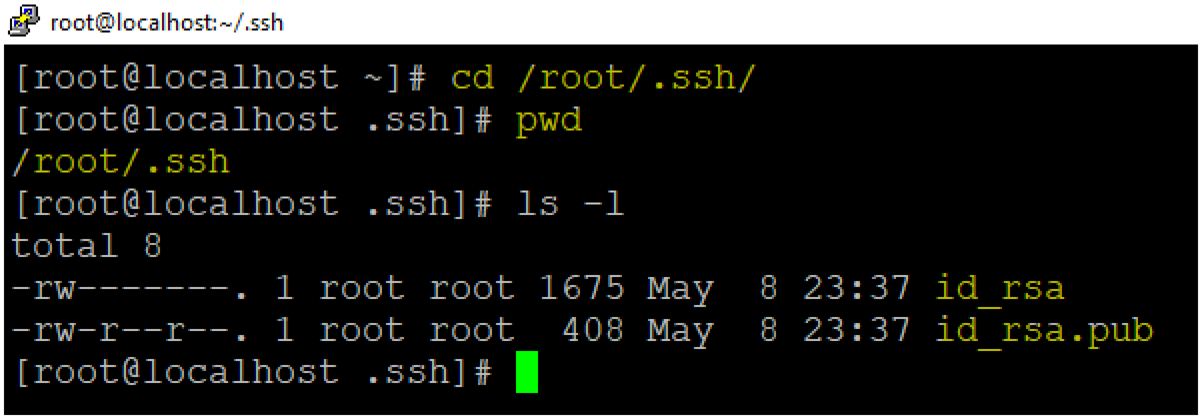
It will prompt above to mention the file where these keys shoud be stored, to keep its default directory just press “Enter”. The default location will be /root/.ssh/ directory

Okay now our keys are successfully generated, go to /root/.ssh/ directory and check for the keys.

#cd /root/.ssh

The id_rsa is a private key and id _rsa.pub is the public key which will be used later to make password less login.

Like below

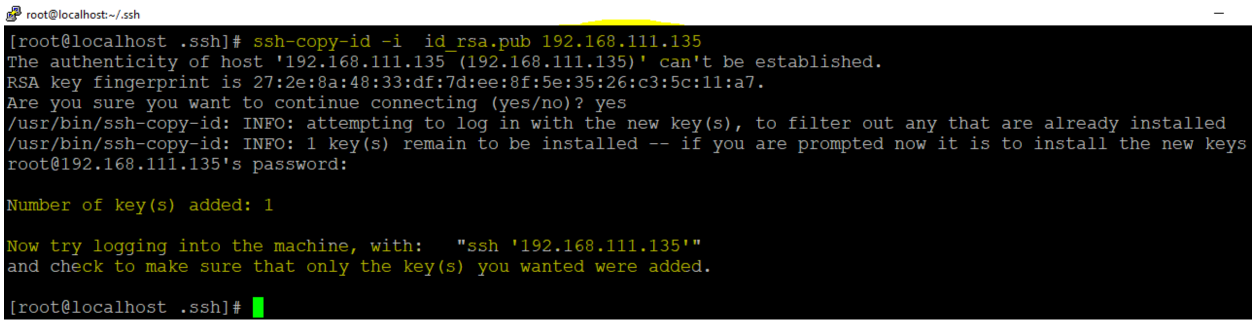


root@localhost:~/.ssh

```
[root@localhost ~]# cd /root/.ssh/
[root@localhost .ssh]# pwd
/root/.ssh
[root@localhost .ssh]# ls -l
total 8
-rw-------. 1 root root 1675 May  8 23:37 id_rsa
-rw-r--r--. 1 root root  408 May  8 23:37 id_rsa.pub
[root@localhost .ssh]#
```

Copying the public key on Client system:

- ➔ To copy the server's public key in client system, the command is
- ➔ #ssh-copy-id –I <pubic key location> <client address> (or user @ client IP)
- ➔ #ssh-copy-id -i /root/.ssh/id_rsa.pub 192.168.111.135



root@localhost:~/.ssh

```
[root@localhost .ssh]# ssh-copy-id -i  id_rsa.pub 192.168.111.135
The authenticity of host '192.168.111.135 (192.168.111.135)' can't be established.
RSA key fingerprint is 27:2e:8a:48:33:df:7d:ee:8f:5e:35:26:c3:5c:11:a7.
Are you sure you want to continue connecting (yes/no)? yes
/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
root@192.168.111.135's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '192.168.111.135'"
and check to make sure that only the key(s) you wanted were added.

[root@localhost .ssh]#
```

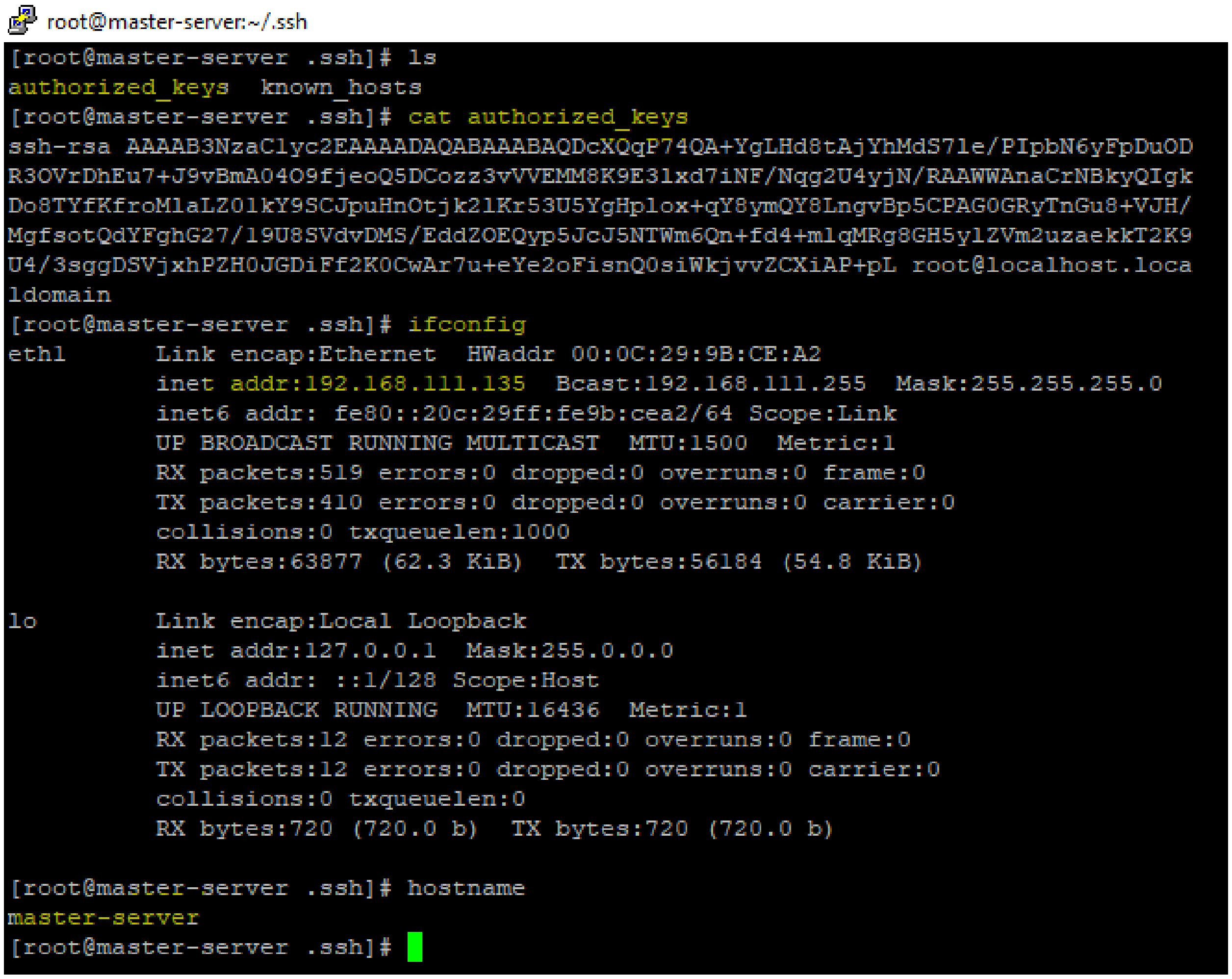
Enter the password of the client to proceed, check it on client side whether it is copied or not Move to client system and check whether the key is copied properly or not

To check the key navigate to /root/.ssh/ directory and check for authorzed_keys file which will hold all the system which are authorized and will not be asked for password..

#cd /root/.ssh/

#cat authorized_keys

Login to client machine and check weather pub key copied or not

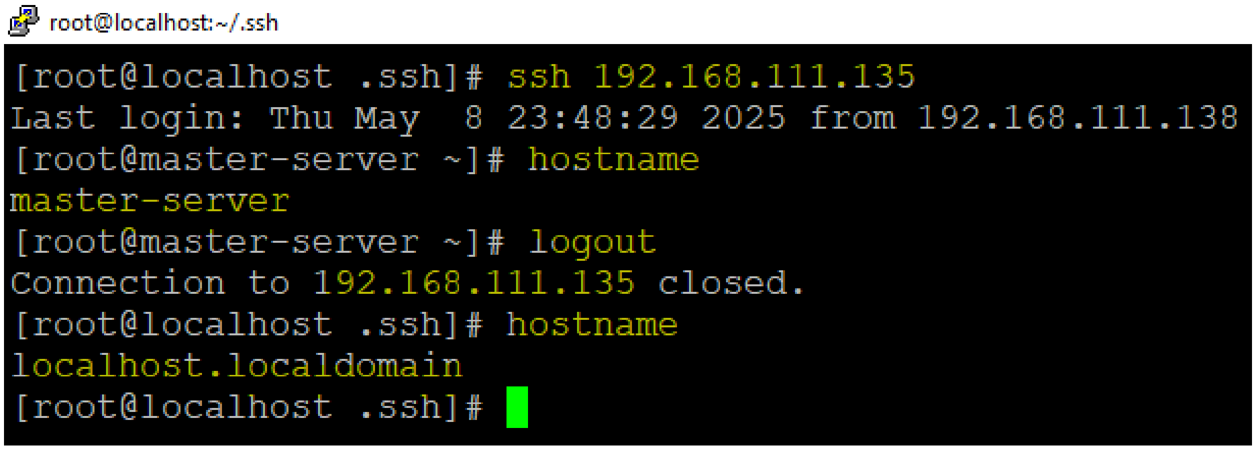


root@master-server:~/.ssh

```
[root@master-server .ssh]# ls
authorized_keys  known_hosts
[root@master-server .ssh]# cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDcXQqP74QA+YgLHd8tAjYhMdS71e/PIpbN6yFpDuOD
R3OVrDhEu7+J9vBmA04O9fjeoQ5DCozz3vVVEMM8K9E31xd7iNF/Nqg2U4yjN/RAAWWAnaCrNBkyQIgk
Do8TYfKfroMlaLZ01kY9SCJpuHnOtjk21Kr53U5YgHplox+qY8ymQY8LngvBp5CPAG0GRyTnGu8+VJH/
MgfsotQdYFghG27/19U8SVdvDMS/EddZOEQyp5JcJ5NTWm6Qn+fd4+mlqMRg8GH5ylZVm2uzaekkT2K9
U4/3sggDSVjxhPZH0JGDiFf2K0CwAr7u+eYe2oFisnQ0siWkjvvZCXiAP+pL root@localhost.loca
ldomain
[root@master-server .ssh]# ifconfig
eth1      Link encap:Ethernet  HWaddr 00:0C:29:9B:CE:A2
          inet addr:192.168.111.135  Bcast:192.168.111.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe9b:cea2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:519 errors:0 dropped:0 overruns:0 frame:0
          TX packets:410 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:63877 (62.3 KiB)  TX bytes:56184 (54.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:720 (720.0 b)  TX bytes:720 (720.0 b)

[root@master-server .ssh]# hostname
master-server
[root@master-server .ssh]#
```



root@localhost:~/.ssh

```
[root@localhost .ssh]# ssh 192.168.111.135
Last login: Thu May  8 23:48:29 2025 from 192.168.111.138
[root@master-server ~]# hostname
master-server
[root@master-server ~]# logout
Connection to 192.168.111.135 closed.
[root@localhost .ssh]# hostname
localhost.localdomain
[root@localhost .ssh]#
```

Now we are able to login to the client server without password.

Note: same way you can do from client to server also. Process is same.

====================END Passwdless authentication===============

Remote file transfer with SCP and RSYNC:

SCP (SECURE COPY):

- ➔ scp stands for secure cp (copy), which means that you can copy files across an ssh connection that will be encrypted, and therefore secured. As scp will be using ssh protocol to transfer the data, hence it is termed as the safest method of transferring data from one location to another.
- ➔ To copy a file using SCP to remote machine from source location

Source IP: 192.168.111.138

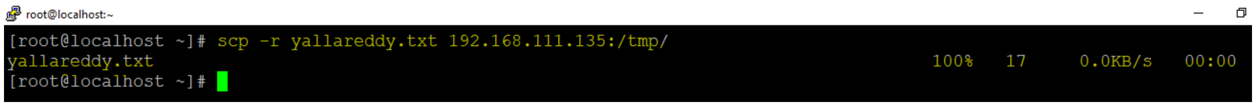
Destination IP: 192.168.111.135

I want to copy a file yallareddy.txt from source to destination. The syntax is below

The syntax for SCP a file from source location.

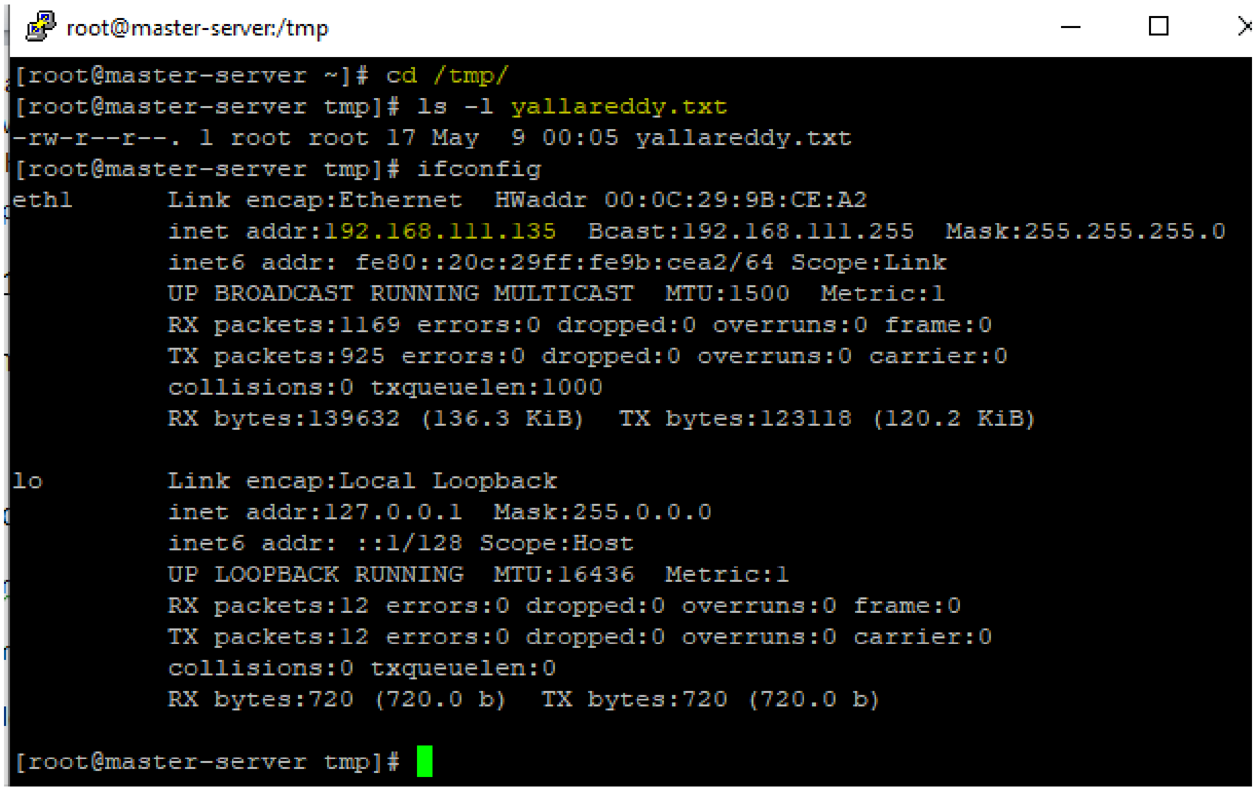
#scp <filename> <remotehost IP>:/<location to copy the file>

scp -r yallareddy.txt 192.168.111.135:/tmp/



```
[root@localhost ~]# scp -r yallareddy.txt 192.168.111.135:/tmp/
yallareddy.txt                                                         100%   17     0.0KB/s   00:00
[root@localhost ~]#
```

- ➔ Now log in to destination system and check whether if the file is there.


root@master-server:/tmp

```
[root@master-server ~]# cd /tmp/
[root@master-server tmp]# ls -l yallareddy.txt
-rw-r--r--. 1 root root 17 May  9 00:05 yallareddy.txt
[root@master-server tmp]# ifconfig
eth1      Link encap:Ethernet  HWaddr 00:0C:29:9B:CE:A2
          inet addr:192.168.111.135  Bcast:192.168.111.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe9b:cea2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1169 errors:0 dropped:0 overruns:0 frame:0
          TX packets:925 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:139632 (136.3 KiB)  TX bytes:123118 (120.2 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:720 (720.0 b)  TX bytes:720 (720.0 b)

[root@master-server tmp]#
```

##############END SCP

RSYNC (REMOTE SYNCHRONIZATION):

if rsync is used it will only copy the updated files/directories rather than copying all files/directories inside main directory, which saves lots of time and speedup the transfer.

Note: rsync the syntax is same like scp only.