# Foundations of Information Assurance

*Lab Assignment 1 Report*

**Team 20**

**Sonam Ghatode**

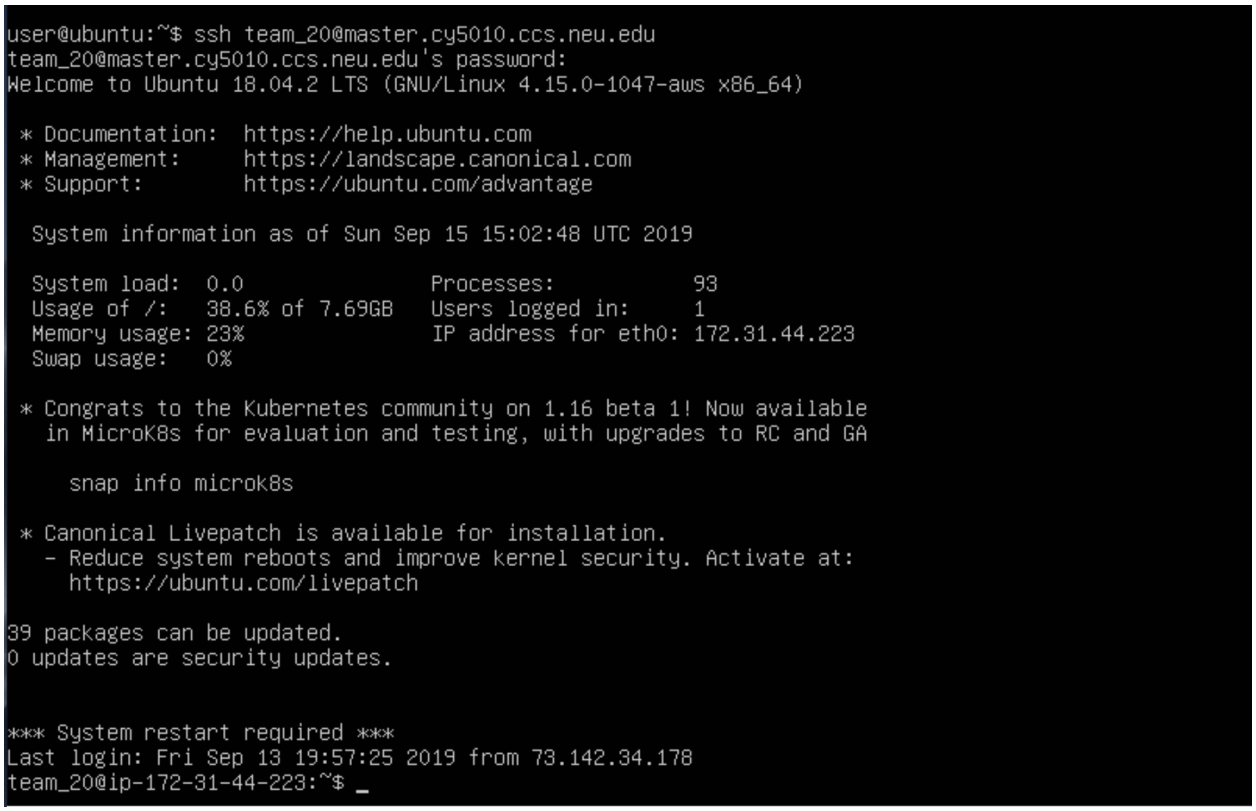
**Vishal Maurya**

09.15.2019
CY5010

# Index

## Part 1 - Password Authentication

- In this part we basically use password authentication to perform a successful remote ssh login. Here is the screenshot for the same:



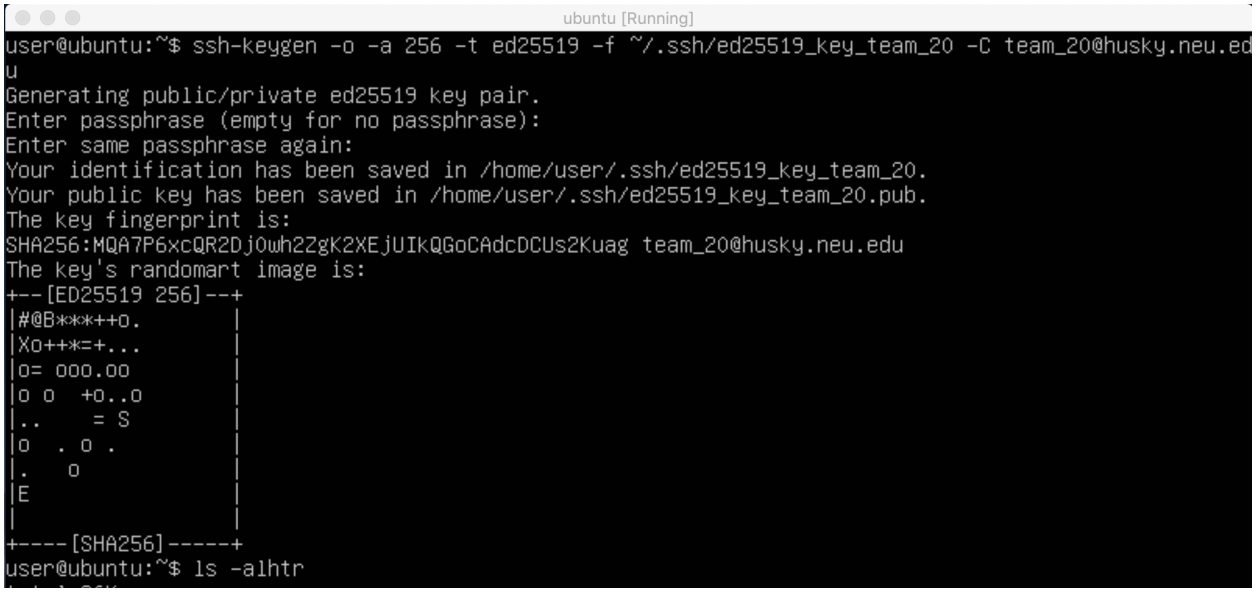

## Part 2 - Public Key Authentication

- Public Key authentication involves authenticating a user's remote ssh login attempt from source machine to the destination machine(master server : AWS EC2 in our case) using key authentication by matching public key and private key.
- On the destination machine, this public key is copied into a special file called ***'authorized_keys'*** inside the ***'.ssh'*** directory of the user, if this file already exist then the public key is appended in that file along with the keys already present.
- Hence this way multiple users on multiple machines can login into a destination

machine as the same user with same privileges.

Steps to perform this operation :

1. Generate private-public key pair :




Attribute List :

| Attribute | Description |
|---|---|
| -o | This option saves the generated private-key using the new OpenSSH format instead of PEM format. |
| -a | Used for providing the numbers of KDF (Key Derivation Function) rounds for increasing the resistance to a brute-force password cracking attack in case the private-key is stolen. |
| -t | To provide which encryption algorithm to use in order to create the key pair. |
| -f | Used to give the resultant file name in which the private key should be saved. |
| -C | To provide any comment. It does not play any role in the encryption or strength of they key. |

2. Update the */home/user/.ssh/config* file and add host entry for easier ssh command entry with an alias instead of writing whole server IP/domain_name :

Command to enter in terminal to open */home/user/.ssh/config* file :

$ vi */home/user/.ssh/config*

Add the following content in it :

```
Host cy5010_master
        Hostname cy5010.ccs.neu.edu
        port 22
        User team_20
        IdentityFile /home/user/.ssh/ed25519_key_team_20
```

3. Copy the newly generated public key to the master machine using ***ssh-copy-id*** command:

   In order to copy your public key to the destination, ***you need to use password authentication for security, this is a one time operation*** as once the public key is copied successfully we will be using public-key authentication.

   *Command format : ssh-copy-id -i <path/to/publick_key> destination_username@destination_hostname*

```
user@ubuntu:~$ ssh-copy-id -i ~/.ssh/ed25519_key_team_20.pub team_20@master.cy5010.ccs.neu.edu
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/user/.ssh/ed25519_key_team_20.p
ub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
team_20@master.cy5010.ccs.neu.edu's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'team_20@master.cy5010.ccs.neu.edu'"
and check to make sure that only the key(s) you wanted were added.

user@ubuntu:~$ _
```

   Attribute List :

| Attribute | Description |
|---|---|
| -i | Used for providing path to the public key that needs to be copied to destination user on destination machine. |

   **Note** : On the destination machine, this public key is copied into a special file called ***'authorized_keys'*** inside the ***'.ssh'*** directory of the user, if this file already exist then the public key is appended in that file along with the keys already present otherwise this file is created by ssh-copy-id command.

4. Ssh to the master server using alias/hostname mentioned in the ssh config instead of domain_name/IP :

   To give an alias name for server, ifconfig is edited accordingly to facilitate ssh:

```
Host cy5010_master
        Hostname cy5010.ccs.neu.edu
        port 22
        User team_20
        IdentityFile /home/user/.ssh/ed25519_key_team_20
```

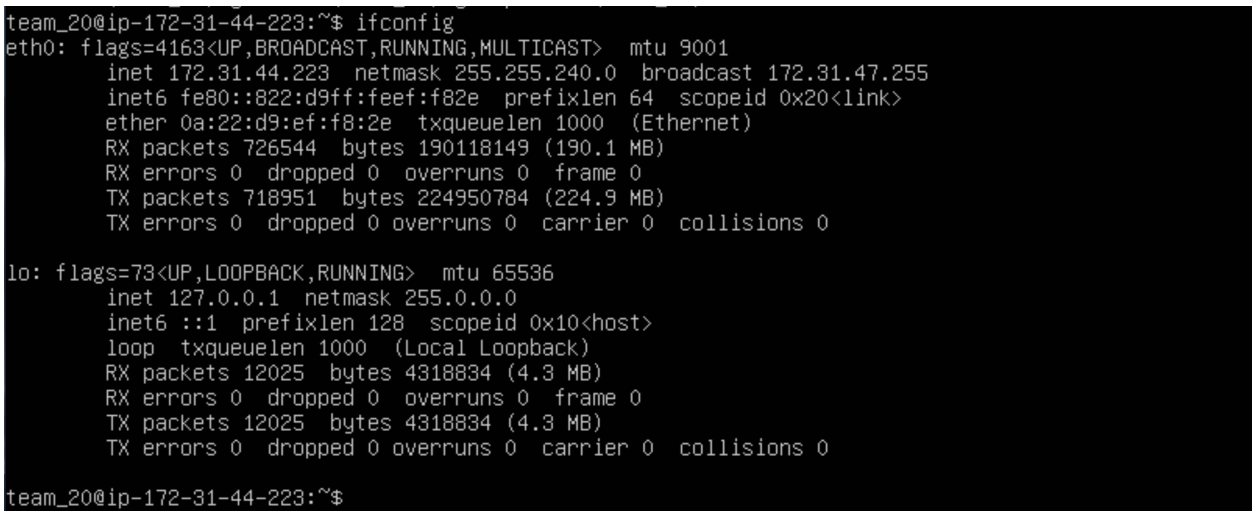
   This is followed by doing ssh using alias/hostname:

```
user@ubuntu:~$ ssh cy5010_master
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1047-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Sep 13 19:30:35 UTC 2019

  System load:  0.0               Processes:           108
  Usage of /:   37.2% of 7.69GB   Users logged in:     3
  Memory usage: 21%               IP address for eth0: 172.31.44.223
  Swap usage:   0%

 * Congrats to the Kubernetes community on 1.16 beta 1! Now available
   in MicroK8s for evaluation and testing, with upgrades to RC and GA

     snap info microk8s

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

39 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Fri Sep 13 19:11:46 2019 from 73.142.34.178
team_20@ip-172-31-44-223:~$ _
```

***Output of id command on master server:***

```
team_20@ip-172-31-44-223:~$ id
uid=1031(team_20) gid=1031(team_20) groups=1031(team_20)
team_20@ip-172-31-44-223:~$
```

*Output of ifconfig command on master server:*




## How can you debug on your SSH connection ?

To debug a SSH connection we can use the -v attribute of ssh program, it increases the verbosity of the command and gives the description of each step while authenticating a ssh connection. To increase the debugging level we can introduce more ‘v’s in the command, max allowed number of vs is three.

Example :

ssh -v My_Local_Macbook

ssh -vv My_Local_Macbook

ssh -vvv My_Local_Macbook

## Other Use Cases for SSH :

**1. scp**: scp is a command that is used to copy file or files across the server. Some of the options available to be used in scp are:

Attribute List :

| Attribute | Description |
|---|---|
| -v | verbose command prints the debugging messages to the screen to facilitate user to view actual debugging process. |
| -3 | to be used when a file needs to be copied from a source system to the destination system that are different from the current system without leaving the file on the mediater system. |
| -r | copies entire directory recursively. |
| -q | used to disable the progress meter. |

**2. sftp**: SSH File Transfer Protocol is a network protocol that is used to transfer files or file management. Following are the commands to transfer files:

Attribute List :

| Attribute | Description |
|---|---|
| get | used to pull a file from the destination remote server.<br>Syntax :<br>sftp > get file/to/be/pulled |
| put | used to push a file to the destination remote server. |

| | |
|---|---|
| | Syntax :<br>sftp> put file/to/be/pushed |

**3. rsync**: rsync command is used to synchronize files across remote server directory and host server directory.

**4. ssh-agent:** very often you would want to jump from a NAT or a mediator virtual machine / system to another system without having to physically copy your private key to the mediator system, this feature is provided by ssh agent which allows ssh key forwarding with -A attribute.

**5. Proxy Jump** : same as the above case when you want to ssh into a machine using a mediator machine as a hop, you can do this by providing ProxyCommand attribute in hostname entry of ~/.ssh/config file.

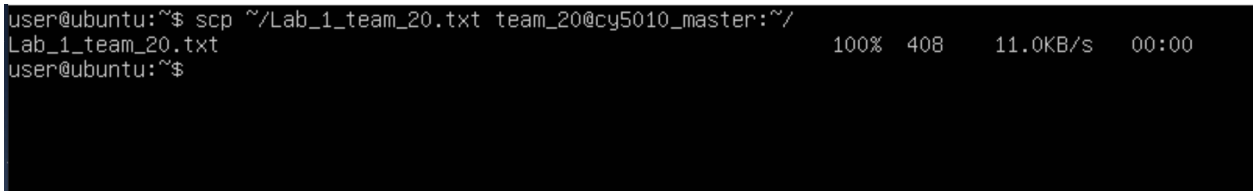
Example Host Entry on local machine to use VM as mediator :

```
Host cy5010_master
HostName  cy5010.ccs.neu.edu
User user
Port 22
ProxyCommand ssh team_20@10.0.2.12 -W %h:%p
```

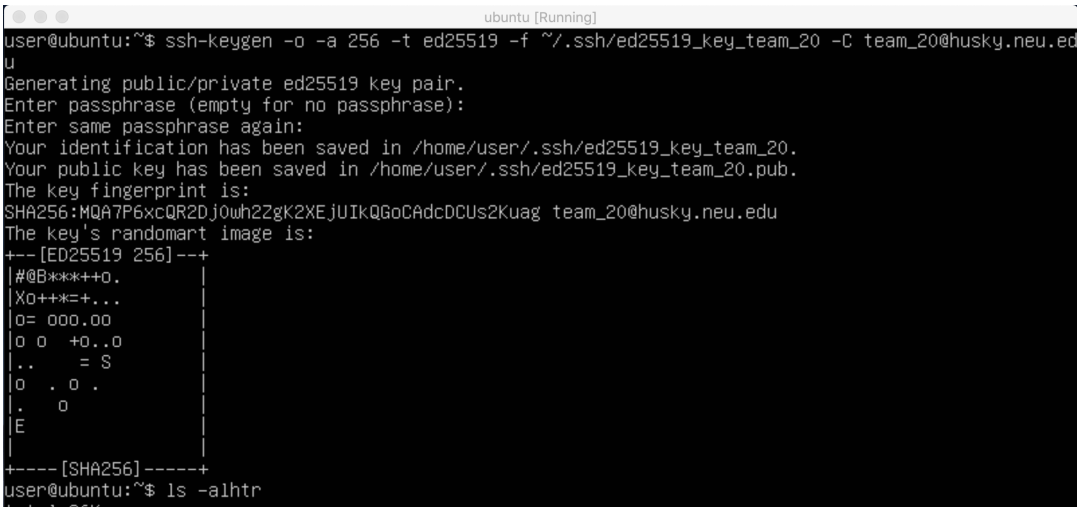
## How to secure copy file from VM to master server :

To copy a file securely from VM to master server, we used command scp. We created a file named Lab_1_team_20.txt with team member goals on VM and copied it to master server.

Command: ***scp source/file/to/be/copied user_name@IP:destination/of/file/to/be/copied***



## SSH Keygen :




We are using EDDSA algorithm of 256 bytes size for generating the encrypted public-private key pair. Usually it is a common norm among users to use RSA keys with 2048 bytes key length, RSA encryption is based on simple mathematics (i.e. finding modulo and multiplication of integers) as compared to EDDSA, short form of Elliptic Curve Digital Signature Algorithm, which is based on graph theory providing much more complexity and security in smaller keys than RSA. Given smaller key sizes of EDDSA are as secure as larger key sizes of RSA, the process of encryption is comparatively faster in EDDSA than in RSA. Because of the faster computation time and smaller secure keys, we chose EDDSA public key encryption algorithm.

## Extra Credit:

1 & 2 :

> We will be using 3 methods to copy the file from master server to local system. To elaborate on this we need to understand one ground rule with virtual box VM configuration. i.e whenever we create a virtual box on a machine. The virtual box/boxes share the same subnet with the host machine in our case. We will leverage this fact and perform the remote file copy operation.

The 3 methods require the newly created ssh public key to be copied from virtual box to our local machine.

In order to copy the public key we first need to **find out the private IP** of the localhost machine. We can do this by using the ***ifconfig*** command on terminal in Localhost machine.

```
Vishals-MacBook-Pro:~ vishal$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
XHC20: flags=0<> mtu 0
XHC0: flags=0<> mtu 0
VHC128: flags=0<> mtu 0
en3: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether ac:de:48:00:11:22
        inet6 fe80::aede:48ff:fe00:1122%en3 prefixlen 64 scopeid 0x7
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (100baseTX <full-duplex>)
        status: active
ap1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
        ether a6:83:e7:7a:91:08
        media: autoselect
        status: inactive
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether a4:83:e7:7a:91:08
        inet6 fe80::1c06:f546:a7c5:1c45%en0 prefixlen 64 secured scopeid 0x9
        inet6 2601:197:700:41cf:3f:f9c4:f2c0:6020 prefixlen 64 autoconf secured
        inet6 2601:197:700:41cf:e93b:e312:39c4:43ac prefixlen 64 autoconf temporary
        inet6 2601:197:700:41cf::b1f7 prefixlen 64 dynamic
        inet 10.0.0.87 netmask 0xffffff00 broadcast 10.0.0.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

According to the above screenshot the Private IP assigned to the laptop connected to Wifi using en0 network interface is 10.0.0.87.

Hence the command to copy the ssh public key to localhost from virtual box is :

```
user@ubuntu:~$ ssh-copy-id -i ~/.ssh/ed25519_key_team_20.pub vishal@10.0.0.87
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/user/.ssh/ed25519_key_team_20.p
ub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
Password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'vishal@10.0.0.87'"
and check to make sure that only the key(s) you wanted were added.

user@ubuntu:~$ _
```

For handy usage and easier running of commands using tab, We have created another host entry for localhost in the ssh config file of virtualbox with hostname My_Local_Macbook. As follows :

```
Host cy5010_master
        Hostname cy5010.ccs.neu.edu
        port 22
        User team_20
        IdentityFile /home/user/.ssh/ed25519_key_team_20

Host My_Local_Macbook
        Hostname 10.0.0.87
        port 22
        User vishal
        IdentityFile /home/user/.ssh/ed25519_key_team_20
```

Method 1 to copy file from master to localhost:

1. We will be using scp command for secure file copy. In this method we will be directly copying file from source (master server) to localhost ***using scp's -3 attribute***, which copies files from remote source to remote destination without saving the file on local machine. The catch here is that both source and destination should have copied public key of the user on virtual box machine to one of the users ssh authorized_keys file, as scp uses ssh authentication for login.

   Command :

   ***scp -3 remote_source_user@remote_source_domain:</path/to/source/file> remote_destination_user@remote_destination_domain:</path/to/destination/file>***

   As we have already added ssh config entry, we do not need to provide username and IP/domain, we can use the hostname in our ssh config instead, So in our case the command will be :

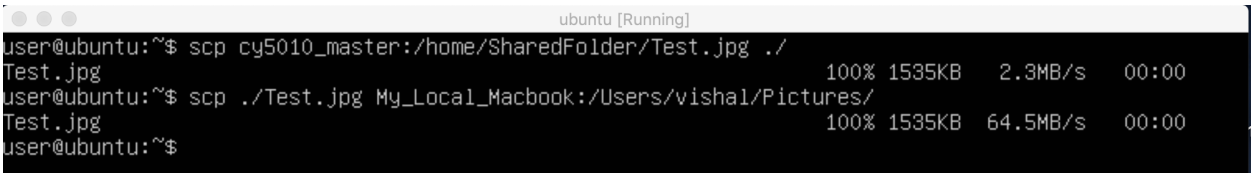
```
user@ubuntu:~$ scp -3 cy5010_master:/home/SharedFolder/Test.jpg My_Local_Macbook:/Users/vishal/Pictu
res/
user@ubuntu:~$ _
```

   Though the file is copied successfully we are not able to see the output of the command in the screenshot as the ***'-3'*** kills the verbosity associated with ssh program.

Method 2 to copy file from master to localhost:

In this method we will be doing 2 hops, **in 1 hop we will copy file from Remote server to Virtual box which is a conventional scp pull**

operation. Additionally to **get the file from virtualbox to local machine we will be using scp push**, i.e. pushing file from virtualbox to localhost machine. As follows :



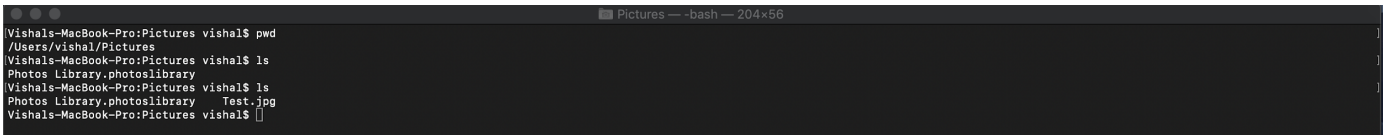

Method 3 to copy file from master to localhost :

Another powerful tool which uses ssh based authentication for file transfer is SFTP (Secure File Transfer Protocol) we will start an interactive sftp console and use combination of SFTP's get and put commands to achieve our goal.

First we will start an interactive sftp console with master server and use ***get*** command to download file from master server, then exit console with ***bye*** command. As follows :


Now we will start an interactive sftp console with Localhost and use ***put*** command to push file from virtualbox to local machine. As follows :


The result of these commands is the same, that we get the file in local machine we can verify this by ***running pwd command in /Users/vishal/Pictures*** directory from terminal before and after copying the file :

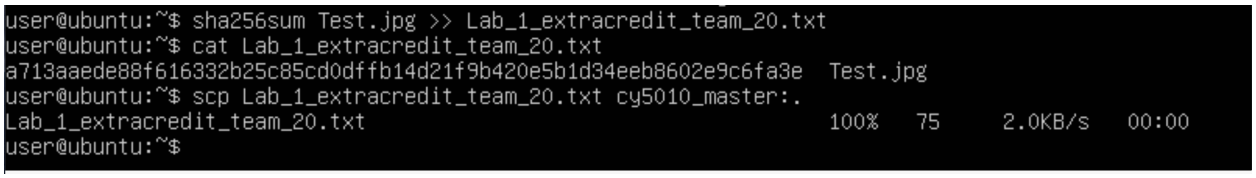



3 & 4 :

For finding out sha value we are using the sha256sum command :

Syntax : $ sha256sum <filename>

In order ***to append the value into a file we are using bash redirection*** >>




## References :

BSD General Commands Manual : https://www.freebsd.org/cgi/man.cgi