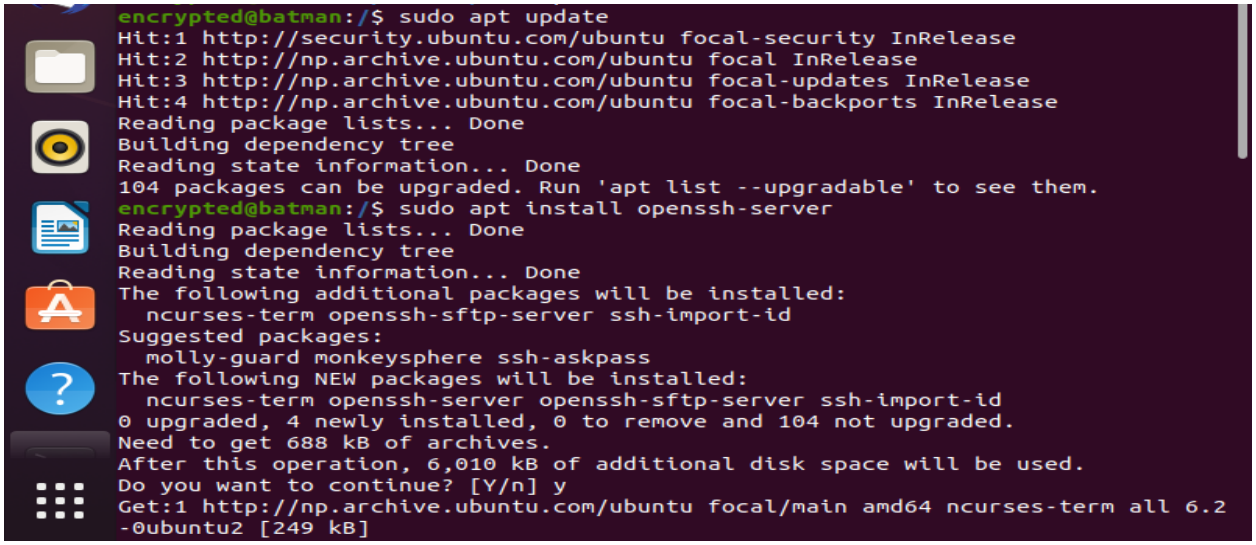
Q1: Install SSH server in your operating system installed previously.

Ans. In ubuntu 20.04, ssh is installed by using following command:
'***Sudo apt install openssh-client***'
'***Sudo apt install openssh-server***'



```
encrypted@batman:/$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://np.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://np.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://np.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
104 packages can be upgraded. Run 'apt list --upgradable' to see them.
encrypted@batman:/$ sudo apt install openssh-server
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 ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 104 not upgraded.
Need to get 688 kB of archives.
After this operation, 6,010 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://np.archive.ubuntu.com/ubuntu focal/main amd64 ncurses-term all 6.2
-0ubuntu2 [249 kB]
```

To start ssh, this command can be used,
'***Systemctl start ssh***'
And to check the status of ssh:
'***Systemctl status ssh'***

QN2 Change ssh port from 22 to 8080

Ans: The default port of ssh is 22. It can be seen by using command
'***Sudo telnet -tulnp | grep ssh***'
When above command is entered, this screen is shown:

```
encrypted@batman:/$ sudo netstat -tulnp | grep ssh
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
 4792/sshd: /usr/sbi
tcp6       0      0 :::22                   :::*                    LISTEN
 4792/sshd: /usr/sbi
encrypted@batman:/$ sudo systemctl restart ssh
```

Which shows that ssh is running on port 22.

To change port address, we should edit the ssh_config file, which can be done by this command:
'***Sudo nano /etc/ssh/sshd_config***'
Which takes us to the editor for the sshd_config document.
At the document we have to find the line "#port 22" and replace it with "port 8080" as shown below:

```
Include /etc/ssh/sshd_config.d/*.conf

Port 8080
#AddressFamily any
#ListenAddress 0.0.0.0
```

After every changes to the sshd_config file, the ssh server must be restarted in order to implement the changes, which is done using command:
'***Systemctl restart ssh***'

Since we have activated the firewall, we need to allow the new ssh port for the firewall and must be reloaded using following commands:
'***ufw allow 8080/tcp***'
'ufw reload'

```
tom@batman:/$ sudo netstat -pnltu | grep 8080
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN
 3051/sshd: /usr/sbi
tcp6       0      0 :::8080                 :::*                    LISTEN
 3051/sshd: /usr/sbi
tom@batman:/$ sudo ufw allow 8080/tcp
Rule added
Rule added (v6)
tom@batman:/$ sudo ufw reload
Firewall reloaded
tom@batman:/$
```

QN 3. Create 3 ssh-key for Tom, Hary and Encrypted

Ans: ssh key can be generated by using command :
'***Ssh-keygen***'
We can enter this above command for every user once to generate an ssh-key pair.

```
tom@batman:/$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/tom/.ssh/id_rsa):
Created directory '/home/tom/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/tom/.ssh/id_rsa
Your public key has been saved in /home/tom/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:92fFd4YaXGn20AL/z3s4vXHECqtWVTPmjbpBChEZ8Y0 tom@batman
The key's randomart image is:
+---[RSA 3072]----+
|        +=  .    |
|        o. o o *.|
|         .E . @o=|
|        .  ..+oX.|
|        S..o+o. @|
|         ...+= *+|
|           .+o+++|
|           ...oo *|
|          ..    +o|
+----[SHA256]-----+
tom@batman:/$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
```

And then we can switch the user to hary using the command: '***Su hary'***, and entering the password for hary and repeating the same command: '***ssh-keygen***'

```
hary@batman:/$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hary/.ssh/id_rsa):
Created directory '/home/hary/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hary/.ssh/id_rsa
Your public key has been saved in /home/hary/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:qZ/ohusKK1vTT6pVp5vkxDOFxV8ppuQPe8hw5LyJCsk hary@batman
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|       .     .   |
|        = o o    |
|       O = o     |
|      + S .      |
| . o o X B       |
|. E o.@ * o      |
|.+ +.O.B o       |
|+.o+=+B o        |
+----[SHA256]-----+
hary@batman:/$ su encrypted
Password:
encrypted@batman:/$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/encrypted/.ssh/id_rsa
Enter passphrase (empty for no passphrase):
```

And again for encrypted,
We first switch the user and enter the same command:

```
encrypted@batman:/$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/encrypted/.ssh/id_rsa):
/home/encrypted/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/encrypted/.ssh/id_rsa
Your public key has been saved in /home/encrypted/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5qZcKQReunjPG4sakWsBM5QX7AmHX8RZAoLWaFN/Rek encrypted@batman
The key's randomart image is:
+---[RSA 3072]----+
|.oB+=oo..o.      |
|oO.=.+. ..       |
|* B +....        |
| + * +.  E       |
|  + o . S        |
|   = o o .       |
|  = o o =        |
| . o = B         |
|  ... B.         |
+----[SHA256]-----+
encrypted@batman:/$
```

We can go to ~/.ssh directory to see the created public and private keys.

```
encrypted@batman:/$ cd ~
encrypted@batman:~$ cd .ssh
encrypted@batman:~/.ssh$ ls
id_rsa  id_rsa.pub  known_hosts
encrypted@batman:~/.ssh$ cd /
```

Here, id_rsa file contains private key and id_rsa.pub file contains public key,
Which can be viewed by command ‘***cat id_rsa.pub***’

QN 4 Disable Password authentication in ssh.

Ans:
For this, we should edit the sshd_config file using command:
'***Nano /etc/ssh/sshd_config***'
And find the line which contains "#PasswordAuthentication yes"
And change that "yes" into "no" as shown below:

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

This file is saved and ssh is reloaded for the recent change to be effective using command:
'***Systemctl reload ssh***'

QN 5. Configure sshd_config so that Tom and hary can ssh and modify the server and Encrypted can access only sftp via public key only. Note: disable password based authentication in ssh.

Ans:
First of all, a file named authorized_keys is made on the server which should contain the public keys of all the users which can access the server using ssh.

Since it contains the public keys of all users,
Lets first copy the public key of user encrypted using command
'***cat id_rsa.pub >> authorized_keys***'

```
encrypted@batman:~/.ssh$ cat id_rsa.pub >> authorized_keys
encrypted@batman:~/.ssh$ nano authorized_keys
encrypted@batman:~/.ssh$
```

Now we have to copy the public key of both tom and hary.
First we switch to user tom using command
'***su tom***'
To view content of public key of user tom,
'***cat ~/.ssh/id_rsa.pub***'
The public key is copied and pasted in the authorized_keys file.

```
encrypted@batman:~/.ssh$ su tom
Password:
tom@batman:/home/encrypted/.ssh$ cat ~/.ssh/id_rsa.pub >> au
bash: authorized_keys: Permission denied
tom@batman:/home/encrypted/.ssh$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCZG7HgW1qJdBEq4tpDpata
UUb6QE9HjOTDGDOz1f1lAXyyQpxPng0Q0ygvf08HxT/lkrw2OfRK2HF71CjS
kZY1MfbhYCFGFNH7QiLKZFF6sFkRg5lFh0g6NB0mYQTldDBwdn8Ypz+rzULv
LX+rOyTf5r53iRIPbch+49+zPjbWTDWcyD3qPQFEXMGkQnW3J1DCJf4jKF7A
6/HdtZo+inaS+BkuITCHxu6OXFZheLvshqw4R2ZZmE9Wzha1G1uO3XqvefOJ
+V+J6LiRxgEcz7+dac7MthDOfnfLQh077Undc44dmrf1IjZvzQolMY7nwTYx
CiOwXsm93Wy5iLKH/ivOAswJGRNb1UH0+ExzrKeZc1hyUVMEREiyOsdsjfi+
tom@batman
tom@batman:/home/encrypted/.ssh$
```

Copy
Copy as HTML
Paste
Read-Only
Preferences
New Window
New Tab
Show Menubar

Similarly for user hary also, same step is repeated as shown in figure:

```
encrypted@batman:~/.ssh$ su hary
Password:
hary@batman:/home/encrypted/.ssh$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDhKoPRGBa3hlt3nmtR/Xoyf8patw62fbTek0/mM4T
iS0MNNRvOSqubWrfi/iTMuET+1cPdfXeLupxfSYs97WTutfvyF3byVfCJogV64Y3DcwnsYZUR9CjL3Z
+PPejkFu902cv2/ySF1enTJrQHhuP7FXoD50eZGMqHDjEpiOsZjQ5b8PouHfXndE8CbivrMaHBiSbJd
LodM8di7DoSR4rQbVJwIy9jKnttAQGXl7/ba58BELS0FLMlXCTttHuCyYXnzCDd40CnDUWWe9fbzpHS
ZLvTA31OzPgYsWjTl0wfYW7Y0zUR9sLI0Kik2aikAqYfPGCms4QhXF/SdHt5BuoYaBN1NnOjzIFI6BH
XfE9US7oV+qK1WusqqNtNxkgcmWFm16IlPEGzS6AKGfBda9ewz+ehlM4mt0BydNX6lfQw42sJUIYb4t
AaYuI3kKloAHgXFL+s/Wk83mtVHK6UHr/MxSo/cGM4zSGKDq6gvfRrJUNkTBTbyTSce9nHUWhLW4M=
hary@batman
```

The “authorized_keys” file finally looks like this:

```
  GNU nano 4.8                    .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDCz2OEAuz8yJypSI9Utn5d6zhiOG3Lyi+Nucl8o0>
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCZG7HgW1qJdBEq4tpDpata08lmWVNj76+EBtVDOC>
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDhKoPRGBa3hlt3nmtR/Xoyf8patw62fbTek0/mM4>
```

For tom and hary to modify the server and use ssh, we can group them together and apply rule for the group. Let the group name be “sshgroup”.
The group is created with the command
‘***Sudo groupadd -r sshgroup***’
And we can use the command ‘***sudo tail /etc/group***’ to view the latest groups added in the system. Newly created group “sshgroup” is shown at the last of the file.

Now the users tom and hary should be added to this group using command:
‘***Sudo usermod -aG sshgroup tom***’ for tom and
‘***Sudo usermod -aG sshgroup hary***’ for hary.

```
tom@batman:~/Desktop$ sudo groupadd -r sshgroup
[sudo] password for tom:
tom@batman:~/Desktop$ sudo tail /etc/group
pulse-access:x:129:
gdm:x:130:
sssd:x:131:
lxd:x:132:tom
tom:x:1000:
sambashare:x:133:tom
systemd-coredump:x:999:
hary:x:1001:
encrypted:x:1002:
sshgroup:x:998:
tom@batman:~/Desktop$ sudo usermod -aG sshgroup tom
tom@batman:~/Desktop$ sudo usermod -aG sshgroup hary
tom@batman:~/Desktop$
```

Now same is done for user encrypted with new groupname “sftpgroup”.
The used commands are:

‘***Sudo groupadd sftpgroup***’ for group creation and
‘Sudo usermod -aG sftpgroup encrypted’ to add encrypted to group sftpgroup.

Now using command ‘sudo tail /etc/group’, we can verify that the groups have been created and required users are included in desired groups:

```
tom@batman:~/Desktop$ sudo groupadd sftpgroup
tom@batman:~/Desktop$ sudo tail /etc/group
gdm:x:130:
sssd:x:131:
lxd:x:132:tom
tom:x:1000:
sambashare:x:133:tom
systemd-coredump:x:999:
hary:x:1001:
encrypted:x:1002:
sshgroup:x:998:tom,hary
sftpgroup:x:1003:
tom@batman:~/Desktop$ sudo usermod -aG sftpgroup encrypted
tom@batman:~/Desktop$ sudo tail /etc/group
gdm:x:130:
sssd:x:131:
lxd:x:132:tom
tom:x:1000:
sambashare:x:133:tom
systemd-coredump:x:999:
hary:x:1001:
encrypted:x:1002:
sshgroup:x:998:tom,hary
sftpgroup:x:1003:encrypted
tom@batman:~/Desktop$
```

We can see that tom and hary belong to sshgroup and encrypted belongs to sftpgroup.

After this, we need to configure the “sshd_config” file,
And the following part is added/appended at the end of the sshd_config file.

```
match group sshgroup

match group sftpgroup
        ChrootDirectory %h
        X11Forwarding no
        AllowTcpForwarding no
        ForceCommand internal-sftp
```

While accessing ssh from hary to encrypted, using command ‘***ssh encrypted@linux -p 8080’*** it generates a message ‘This service allows sftp connections only’

```
hary@linux:~$ sftp -P 8080 encrypted@linux
Connected to linux.
sftp> exit
exit
hary@linux:~$ ssh encrypted@linux -p 8080
This service allows sftp connections only.
Connection to linux closed.
hary@linux:~$
```

For sftp configuration for user encrypted,
Root access was given, and home directory was selected and following commands are used:
‘Chmod 755 /home/encrypted’
‘Cd /home/encrypted’
‘Mkdir docs public_html’
‘Chown root: sftpgroup’

```
encrypted@linux:~/.ssh$ sudo su
[sudo] password for encrypted:
root@linux:/home/encrypted/.ssh# cd
root@linux:~# ls
snap
root@linux:~# cd home
bash: cd: home: No such file or directory
root@linux:~# cd /home
root@linux:/home# ls
encrypted  encrypted.XLms5OgW  hary  lost+found  tom
root@linux:/home# chown root:root /home/encrypted
root@linux:/home# chmod 755 /home/encrypted
root@linux:/home# cd /home/encrypted
root@linux:/home/encrypted# mkdir docs public_html
root@linux:/home/encrypted# cat /etc/groups
```

```
root@linux:/home/encrypted# chown root:sftpgroup *
root@linux:/home/encrypted# exit
exit
encrypted@linux:~/.ssh$ sftp -P 8080 linux
Connected to linux.
sftp>
```

Here we can see that sftp connection was successful from user encrypted.

QN 6: Allow only ssh in the firewall.

Ans: The command '***sudo ufw allow ssh***' is used to allow ssh in the firewall.
The command '***sudo ufw status***' can be used to see the status and all the ports which are allowed in the firewall. We can see that port 8080 is allowed which is a configured port for ssh and also port 22 is allowed for some reason. So we have to delete other allowed ports in the firewall except that of ssh.
The unwanting ports can be deleted by using command:
'***Sudo ufw delete allow 22/tcp***'
And finally only ssh is allowed in the firewall.

```
encrypted@batman:~/.ssh$ ufw allow ssh
ERROR: You need to be root to run this script
encrypted@batman:~/.ssh$ sudo ufw allow ssh
[sudo] password for encrypted:
Rule added
Rule added (v6)
encrypted@batman:~/.ssh$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
8080/tcp                   ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
8080/tcp (v6)              ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)

encrypted@batman:~/.ssh$ sudo ufw delete allow 22/tcp
Rule deleted
Rule deleted (v6)
encrypted@batman:~/.ssh$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
8080/tcp                   ALLOW       Anywhere
8080/tcp (v6)              ALLOW       Anywhere (v6)

encrypted@batman:~/.ssh$
```