Settting up SFTP for multiple users to access a common SFTP location on Linux

For this tutorial, I am using the lastest Ubuntu image from Ubuntu Docker image repository. I have used Dockers for this tutorial for ease of use. You can still replicate these steps with any Linux environment that supports Debian flavors.

SFTP Creation

Step 1: Create SFTP User Group

Create a new SFTP Users Group. Replace **sftp_users** with your desired group name.

```
$ sudo addgroup sftp_users
```

Step 2: Create a SFTP user

Create a new SFTP User. Replace **acemine**, **syamelux** with your desired user names.

```
$ sudo adduser acemine
```

Add the username's Full Name, Password, if you are trying to follow these steps manually on a Linux box or machine.

Repeat the same for the other user too.

```
$ sudo adduser syamelux
```

Step 3: Add the user to the SFTP group.

Add the users created in **Step 2** to the user group which was created in **Step 1**.

```
$ sudo usermod -G sftp_users acemine
$ sudo usermod -G sftp_users syamelux
```

Step 4: Create the a common directory which needs to be accessed by both users

Taken from: Restrict SSH User Access to Certain Directory Using Chrooted Jail

Change root (chroot) in Unix-like systems such as Linux, is a means of separating specific user operations from the rest of the Linux system; changes the apparent root directory for the current running user process and its child process with new root directory called a chrooted jail.

Create the directories which will be used for SFTP collaboration.

```
$ sudo mkdir -p /opt/pentaho/test/Snapshot_Excelfiles
$ sudo chown root:root /opt/pentaho/test
```

Ensure to change ownership to the directory level above the shared directory, in this case, **Snapshot_Excelfiles** will be excluded.

```
$ sudo chown root:root /opt/pentaho/test
```

Step 5: Assign ownership and permissions to directories

Ensure that ssers and groups have **read** and **write** access to the respective directories.

```
$ sudo chmod -R 755 /opt/pentaho/test/
```

Change ownership on directory **Snapshot_Excelfiles**, assigned to user **syamelux** which was previously created in **Step 2**.

```
$ sudo chown -R syamelux:syamelux /opt/pentaho/test/Snapshot_Excelfiles
```

Step 6: Assign permissions to the group

The below commands will ensure that **sftp_users** can **read, write,** and **execute** within **Snapshot_Excelfiles** directory.

```
$ sudo chgrp sftp_users /opt/pentaho/test/Snapshot_Excelfiles
$ sudo chmod ug+rwX /opt/pentaho/test/Snapshot_Excelfiles
```

Setp 7: Configure SFTP daemon and service

With the sftp group and user accounts created, enable SFTP in the main SSH configuration file.

Using an editor of your choice, open the file /etc/ssh/sshd_config.

```
$ sudo vim /etc/ssh/sshd_config
```

```
# update to only allow sftp and not ssh tunneling to limit the non-necessary activity
Match Group sftp_users
    ForceCommand internal-sftp
    PasswordAuthentication yes
    ChrootDirectory /opt/pentaho/test
    PermitTunnel no
    AllowAgentForwarding no
    AllowTcpForwarding no
    X11Forwarding no
```

Save and close the file.

Below are the functions for each of the above configuration lines:

- Match Group **sftp_users**: Match the user group sftp_users.
- ChrootDirectory **/opt/pentaho/test**: Restrict access to directories within the user's home directory.
- PasswordAuthentication **yes**: Enable password authentication.
- AllowTcpForwarding **no**: Disable TCP forwarding.
- X11Forwarding **no**: Don't permit Graphical displays.
- ForceCommand **internal-sftp**: Enable SFTP only with no shell access.

Also, confirm if SFTP is enabled (it is by default). The line below should be uncommented in **/etc/ssh/sshd_config**:

```
# override default of no subsystems
Subsystem sftp  /usr/lib/openssh/sftp-server
```

Restart the SSH server for changes to take effect.

```
$ sudo systemctl restart sshd
```

Testing SFTP Setup

Open a new terminal window and log in with sftp using a valid user account and password.

```
$ sftp acemine@SERVER-IP
```

OR

```
$ sftp acemine@127.0.01
```

(If running within the same server SSH session) List files within the directory. Your output should be similar to the one below:

```
$ acemine@127.0.0.1's password:

Connected to 127.0.0.1.

sftp> ls

Snapshot_Excelfiles

sftp>
```

Also, try creating a new directory within the subdirectory to test user permissions.

```
sftp> cd Snapshot_Excelfiles
sftp> mkdir uploads
sftp> ls
uploads
```

References:

1. Restrict SSH User Access to Certain Directory Using Chrooted Jail
2. Create SFTP Container using Docker - You may need Medium membership!!
3. Setup SFTP User Accounts on Ubuntu 20.04
4. SSH into Docker Container or Use Docker Exec?
5. Add a User to a Group (or Second Group) on Linux
6. How to Add User to Sudoers in Ubuntu
7. How can I set up SFTP with chrooted groups?