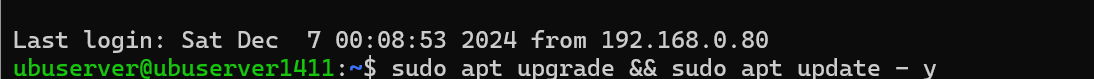
Setting up SFTP server in Ubuntu server VM and sharing files with windows host PC

SSH into your VM first

A screenshot of a computer

Description automatically generated

Run update and upgrade for the system packages



Install openssh-server



Verify SSH service is running if not can use sudo systemctl start ssh and then sudo systemctl enable ssh to keep it running on next reboot

A computer screen with white text

Description automatically generated

Now create a group dedicated for sftp users access and add a new user to the group and set the new user to nologin so they won’t be able to access via ssh. They can transfer files via SFTP but not able to access the server itself

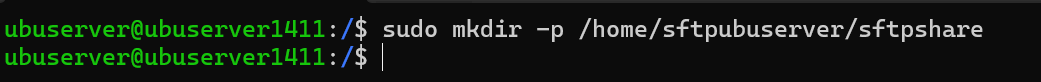


Set password for the SFTP user account

A black screen with white text

Description automatically generated

Create a directory for your sftp uploads



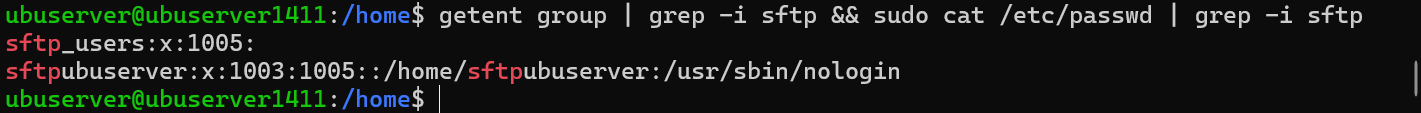
As currently home/sftpubuserver is still on your username and group we need to change the base directory **/home/sftpubuserver** to root:root first to ensure it is not writable by any other user and group

A screenshot of a computer

Description automatically generated

Sets secure permission

Let’s recall the sftp user and the user group we created



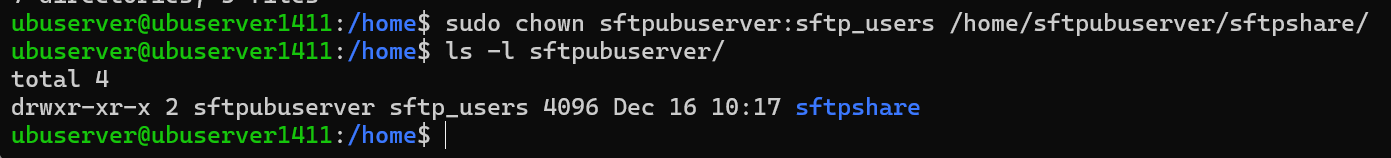
We then set permission for subdirectory of our share folder to be writable by our user and group

\*Side note\*: sometimes I forget chown syntax so tldr came in handy. In this case we have to chown sftpubuserver:sftp\_users for /home/sftpubuserver/sftpshare

A computer screen shot of a program

Description automatically generated

After applying the necessary folder ownership:



However, as we want to let users in sftp\_users to write to this folder as well, we will need to set 775. Execute only allows users to access the files but writing can let them create/delete/edit.

A screen shot of a computer

Description automatically generated

Edit the SSH config file



Add this to the end of the file

A black screen with white text

Description automatically generated

Explanation:

* Match Group sftp\_users: Applies these rules to anyone in the sftp\_users group
* ChrootDirectory /home/%u:
* %u automatically gets replaced with the logging-in user's username so if user "john" logs in, they're chrooted to /home/john
* ForceCommand internal-sftp: Forces SFTP only, no SSH shell access
* AllowTCPForwarding no: Prevents port forwarding
* X11Forwarding no: Prevents X11 GUI forwarding

Restart SSH service for changes of config file to take effect

A black background with white text

Description automatically generated

Now if we open cmd from our host windows pc and sftp in using the sftp user account we created we can see that it is connected.

A screen shot of a computer

Description automatically generated

You can cd into the share folder which users in the sftp\_users group have rwx acccess

A black screen with white text

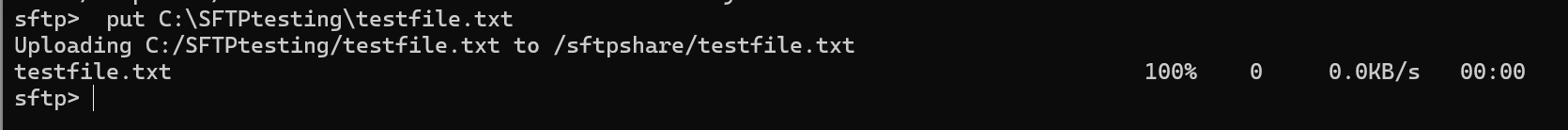
Description automatically generated

Now for example we have a file in our C drive

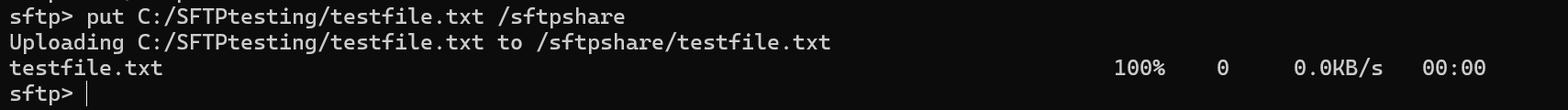
A screenshot of a computer

Description automatically generated

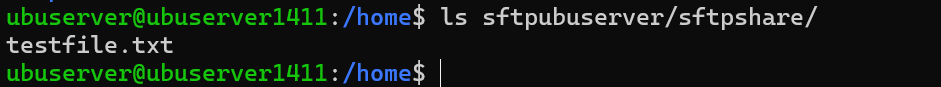
We can try to put the file into the sftp folder (since we are already in the sftpshare sub folder we can do this straight)



However if you are in root directory and wants to upload the testfile.txt from your local host pc to the sftpshare subfolder



As you can see we can now also view the file from our ubuntu server VM



HARDENING – to improve security of our SFTP server

We can disable root login over SSH. As seen in our sshd\_config, we can change the PermitRootLogin to implicit no.

A computer screen with white text

Description automatically generated

First make a copy in case of rollback



Then edit the file



A black background with blue and orange letters

Description automatically generated

Then restart SSH service to apply the changes

