**SFTP**

**Installation RunBook on Centos 7**



**SFTP Installation Steps On Centos 7**

### **Introduction**

[SFTP](https://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server) stands for **S**SH **F**ile **T**ransfer **P**rotocol. As its name suggests, it's a secure way of transferring files to a server using an encrypted SSH connection. Despite the name, it's a completely different protocol than [FTP](https://www.digitalocean.com/community/tutorials/what-is-ftp-and-how-is-it-used)(**F**ile **T**ransfer **P**rotocol), though it's widely supported by modern FTP clients.

SFTP is available by default with no additional configuration on all servers that have SSH access enabled. It's secure and easy to use, but comes with a disadvantage: in a standard configuration, the SSH server grants file transfer access and terminal shell access to all users with an account on the system.

In some cases, you might want only certain users to be allowed file transfers and no SSH access. In this tutorial, we'll set up the SSH daemon to limit SFTP access to one directory with no SSH access allowed on per user basis.

**Prerequisites**

* One CentOS 7 server set up with [this initial server setup tutorial](https://www.digitalocean.com/community/tutorials/initial-server-setup-with-centos-7), including a sudo non-root user.

## **Creating a New User**

First, create a new user who will be granted only file transfer access to the server

sudo useradd <username>

sudo passwd <username>

**Creating a Directory for File Transfers**

In order to restrict SFTP access to one directory, we first have to make sure the directory complies with the SSH server's permissions requirements, which are very particular.

Specifically, the directory itself and all directories above it in the filesystem tree must be owned by **root** and not writable by anyone else. Consequently, it's not possible to simply give restricted access to a user's home directory because home directories are owned by the user, not **root**.

There are a number of ways to work around this ownership issue. In this tutorial, we'll create and use /var/sftp/uploads as the target upload directory. /var/sftp will be owned by **root** and will be unwritable by other users; the subdirectory /var/sftp/uploads will be owned by **sammyfiles**, so that user will be able to upload files to it.

First, create the directories.

sudo mkdir -p /var/sftp/uploads

Set the owner of /var/sftp to **root**.

sudo chown root:root /var/sftp

Give **root** write permissions to the same directory, and give other users only read and execute rights.

sudo chmod 755 /var/sftp

Change the ownership on the uploads directory to user file

sudo chown <username>:<groupname> /var/sftp/uploads

Now that the directory structure is in place, we can configure the SSH server itself.

## **Restricting Access to One Directory**

In this step, we'll modify the SSH server configuration to disallow terminal access fo ruser but allow file transfer access.

sudo vi /etc/ssh/sshd\_config

/etc/ssh/sshd\_config

Match User <username>

ForceCommand internal-sftp

PasswordAuthentication yes

ChrootDirectory /var/sftp

PermitTunnel no

AllowAgentForwarding no

AllowTcpForwarding no

X11Forwarding no

Then save and close the file.

sudo systemctl restart sshd

ssh <username@<ipaddress>

this should not work

sftp <username>@<ipaddress>

SFTP prompt

Connected to localhost.

sftp>

You can list the directory contents using ls in the prompt:

ls

This will show the uploads directory that was created in the previous step and return you to the sftp>prompt.

SFTP file list output

uploads

Copyright AIM Software 2018

All rights reserved. No part of this document may be reproduced or published in any form without written permission from the publisher.