

CONTENTS

Prerequisites

Step 1 — Installing SSHFS

Step 2 — Mounting the Remote Filesystem

Step 3 — Permanently Mounting the Remote Filesystem

Conclusion

// Tutorial //

How To Use SSHFS to Mount Remote File Systems Over SSH

Published on December 23, 2013 · Updated on April 27, 2022

Linux Basics



By Paul White and [Alex Garnett](#)



Introduction

Transferring files over an SSH connection, by using either [SFTP](#) or SCP, is a popular method of moving small amounts of data between servers. In some cases, however, it may be necessary to share entire directories, or entire filesystems, between two remote environments. While this can be accomplished by configuring an [SMB](#) or [NFS](#) mount, both of these require additional dependencies and can introduce security concerns or other overhead.

As an alternative, you can install *SSHFS* to mount a remote directory by using SSH alone. This has the significant advantage of requiring no additional configuration, and inheriting permissions from the SSH user on the remote system. SSHFS is particularly useful when you need to read from a large set of files interactively on an individual basis.

Prerequisites

- Two Linux servers which are configured to allow SSH access between them. One of these can be a local machine rather than a cloud server. You can accomplish this by following our [Initial Server Setup](#) Guide, by connecting directly from one machine to the other.

Step 1 – Installing SSHFS

SSHFS is available for most Linux distributions. On Ubuntu, you can install it using `apt`.

First, use `apt update` to refresh your package sources:

```
$ sudo apt update
```

[Copy](#)

Then, use `apt install` to install the `sshfs` package.

```
$ sudo apt install sshfs
```

[Copy](#)

Note: SSHFS can be installed on Mac or Windows through the use of filesystem libraries called *FUSE*, which provide interoperability with Linux environments. They will use identical concepts and connection details to this tutorial, but may require you to use different configuration interfaces or install third-party libraries. This tutorial will cover SSHFS on Linux only, but you should be able to adapt these steps to Mac or Windows FUSE implementations.

You can install SSHFS for Windows from the project's [GitHub Repository](#).

You can install SSHFS for Mac from the [macFUSE Project](#).

Step 2 – Mounting the Remote Filesystem

Whenever you are mounting a remote filesystem in a Linux environment, you first need an empty directory to mount it in. Most Linux environments include a directory called `/mnt` that you can create subdirectories within for this purpose.

Note: On Windows, remote filesystems are sometimes mounted with their own drive letter like `G:`, and on Mac, they are usually mounted in the `/Volumes` directory.

Create a subdirectory within `/mnt` called `droplet` using the `mkdir` command:

```
$ sudo mkdir /mnt/droplet
```

[Copy](#)

You can now mount a remote directory using `sshfs`.

```
$ sudo sshfs -o allow_other,default_permissions sammy@your_other_server: Copy t
```

The options to this command behave as follows:

- `-o` precedes miscellaneous mount options (this is the same as when running the `mount` command normally for non-SSH disk mounts). In this case, you are using `allow_other` to allow other users to have access to this mount (so that it behaves like a normal disk mount, as `sshfs` prevents this by default), and `default_permissions` (so that it otherwise uses regular filesystem permissions).
- `sammy@your_other_server:~/` provides the full path to the remote directory, including the remote username, `sammy`, the remote server, `your_other_server`, and the path, in this case `~/` for the remote user's home directory. This uses the same syntax as SSH or SCP.
- `/mnt/droplet` is the path to the local directory being used as a mount point.

If you receive a `Connection reset by peer` message, make sure that you have [copied your SSH key to the remote system](#). `sshfs` uses an ordinary SSH connection in the background, and if it is your first time connecting to the remote system over SSH, you may be prompted to accept the remote host's key fingerprint.

Output

```
The authenticity of host '164.90.133.64 (164.90.133.64)' can't be established.
ED25519 key fingerprint is SHA256:05SYuIMxeTDWFZtf3/ruDDm/3mmHkiTfAr+67FBC0+Q.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Note: If you need to mount a remote directory using SSHFS without requiring `sudo` permissions, you can create a user group called `fuse` on your local machine, by using `sudo groupadd fuse`, and then adding your local user to that group, by using `sudo usermod -a -G fuse sammy`.

You can use `ls` to list the files in the mounted directory to see if they match the contents of the remote directory:

[Developer Support Plan] Get response times wit... We're hiring Blog Docs Get Support Contact Sales



Tutorials Questions Learning Paths For Businesses Product Docs Social Impact

Now you can work with files on your remote server as if it were a physical device attached to your local machine. For instance, if you create a file in the `/mnt/droplet` directory, the file will appear on your virtual server. Likewise, you can copy files into or out of the

`/mnt/droplet` folder and they will be uploaded to or from your remote server in the background.

It is important to note that the `mount` command only mounts a remote disk for your current session. If the virtual server or local machine is powered off or restarted, you will need to use the same process to mount it again.

If you no longer need this mount, you can unmount it with the `umount` command:

```
$ sudo umount /mnt/droplet
```

[Copy](#)

In the last step, you'll walk through an example of configuring a permanent mount.

Step 3 – Permanently Mounting the Remote Filesystem

As with other types of disk and network mounts, you can configure a permanent mount using SSHFS. To do this, you'll need to add a configuration entry to a file named `/etc/fstab`, which handles Linux filesystem mounts at startup.

Using `nano` or your favorite text editor, open `/etc/fstab`:

```
$ sudo nano /etc/fstab
```

[Copy](#)

At the end of the file, add an entry like this:

```
...  
sammy@your_other_server:~/ /mnt/droplet fuse.sshfs noauto,x-systemd.automount,_n
```

Permanent mounts often require a number of different options like this to ensure they behave as expected. They work as follows:

- `sammy@your_other_server:~/` is the remote path again, just as before.
- `/mnt/droplet` is the local path again.
- `fuse.sshfs` specifies the driver being used to mount this remote directory.
- `noauto,x-systemd.automount,_netdev,reconnect` are a set of options that work together to ensure that permanent mounts to network drives behave gracefully in case the network connection drops from the local machine or the remote machine.

- `identityfile=/home/sammy/.ssh/id_rsa` specifies a path to a local SSH key so that the remote directory can be mounted automatically. Note that this example assumes that both your local and your remote username are `sammy` – this refers to the local path. It is necessary to specify this because `/etc/fstab` effectively runs as root, and would not otherwise know which username's SSH configurations to check for a key that is trusted by the remote server.
- `allow_other,default_permissions` use the same permissions from the `mount` command above.
- `0 0` signifies that the remote filesystem should never be dumped or validated by the local machine in case of errors. These options may be different when mounting a local disk.

Save and close the file. If you are using `nano`, press `Ctrl+X`, then when prompted, `Y` and then `ENTER`. You can then test the `/etc/fstab` configuration by restarting your local machine, for example by using `sudo reboot now`, and verifying that the mount is recreated automatically.

It should be noted that permanent SSHFS mounts are not necessarily popular. The nature of SSH connections and SSHFS means that it is usually better suited to temporary, one-off solutions, when you don't need to commit to an SMB or NFS mount which can be configured with greater redundancy and other options. That said, SSHFS is very flexible, and more importantly, acts as a full-fledged filesystem driver, which allows you to configure it in `/etc/fstab` like any other disk mount and use it as much as needed. Be careful that you do not accidentally expose more of the remote filesystem over SSH than you intend.

Conclusion

In this tutorial, you configured an SSHFS mount from one Linux environment to another. Although it is not the most scalable or performant solution for a production deployment, SSHFS can be very useful with minimal configuration.

Next, you may want to learn about working with [object storage](#) which can be mounted concurrently across multiple servers.

We've made it super easy to add SSH Keys to your new or existing DigitalOcean virtual machines.

[Learn more here](#) →

About the authors



Paul White Author



[Alex Garnett](#) Author
Senior DevOps Technical Writer

Still looking for an answer?

Ask a question

Search for more help

Was this helpful?

Yes

No



Comments

10 Comments

B *I* U ~~S~~ H₁ H₂ H₃ “”



This textbox defaults to using **Markdown** to format your answer.

You can type **!ref** in this text area to quickly search our full set of tutorials, documentation & marketplace offerings and insert the link!

[Sign In or Sign Up to Comment](#)

SamiF • January 21, 2015 ^

I followed these instructions exactly and still had connection problems.

I was trying to connect my Ubuntu 14.04 laptop to an Ubuntu 14.04 droplet that I have setup SSH access without password and without a password for the key.

I was getting “permission denied” at the mount folder, even with sudo.

What fixed it is adding my account to the fuse group:

```
sudo gpasswd -a USERNAME fuse
```

And also editing **/etc/fuse.conf** on my laptop and enabling a flag by uncommenting the second line here:

```
# Allow non-root users to specify the allow_other or allow_root mount options
user_allow_other
```

Lastly the mount command needs to be modified with an extra option:

```
sudo sshfs -o allow_other -o IdentityFile=~/.ssh/id_rsa root@xxx.xxx.xxx /
```

[Show replies](#) ▾ [Reply](#)

[Kim DongGyu](#) • September 23, 2017 ^

defer_permissions to

```
default_permissions
```

defer_permission is deprecated

[Reply](#)

[jeirosz](#) • March 15, 2017 ^

Hi,

Thanks for the useful tutorial. I'm working on a macOS Sierra v 10.12.13, and I've got 3 different Linux based servers I want to mount. I can manually mount them using my ssh keys as suggested. I've created an `/etc/fstab` file and added the following lines:

```
sshfs#je714@host1:/work/je714/ /mnt/HPC
sshfs#je714@host1:/box/je714/ /mnt/box
sshfs#je714@host2:/ /mnt/titanx2
```

And made sure that the directories do exist under the `/mnt/` directory. But when I reboot my machine, nothing shows up in the folders. Does anyone know if using `/etc/fstab` needs some extra action to work on OSX?

Thanks,

Juan

[Show replies](#) ▼ [Reply](#)

[Joshuabcave](#) • October 10, 2014 ^

I reckon this is the best option.

```
sshfs -o IdentityFile=~/.ssh/id_rsa,auto_cache,reconnect,defer_permission
```

I like the folder ~/Deploy could be anything you want though, just change the volume name as well.

[Reply](#)

iconcoast hero • December 25, 2022 ^

So since using nonstandard ports is extremely useful in protecting an ssh server, you need to add a -p ##### after sshfs if you use something other than 22.

[Reply](#)

HooHost • March 15, 2022 ^

SSH key pair is the only available method, there has to be a way to do this on Windows...

[Reply](#)

lemyskaman • August 2, 2020 ^

Guys i've came out with a very handy tool to ease the use of sshfs, is a console one but it also use zenity for gui inputs.

<https://github.com/lemyskaman/kmountssh>

[Reply](#)

altjx • March 28, 2020 ^

Does anyone have any suggestions on speeding this up or an alternative solution? Sometimes sublime doesn't load newly created files in the sidebar until you reopen it. Also, if you open your laptop and are no longer connected to the network, the computer just hangs up for a few minutes until fuse decides to catch up and realize the mount should be disconnected.

[Reply](#)

[mltechi](#) • January 6, 2020 ^

Under: **Mounting the Remote File System** with the syntax: `sudo sshfs -o allow_other,defer_permissions root@xxx.xxx.xxx.xxx: /mnt/droplet`

The `defer_permissions` is incorrect, and should be `default_permissions` Please update and fix. <3

[Show replies](#) ▾ [Reply](#)

[Stephanie Robogeisha](#) • December 17, 2019 ^

TIP: On MacOSX I had to sudo chown as `_${USER}` to get this to mount otherwise I got `mount_osxfuse: failed to mount: Operation not permitted.`

[Reply](#)

Load More Comments



This work is licensed under a Creative Commons Attribution-NonCommercial- ShareAlike 4.0 International License.

Try DigitalOcean for free

Click below to sign up and get **\$200 of credit** to try our products over 60 days!

[Sign up](#)

Popular Topics

[Ubuntu](#)

[Linux Basics](#)

[JavaScript](#)

[Python](#)

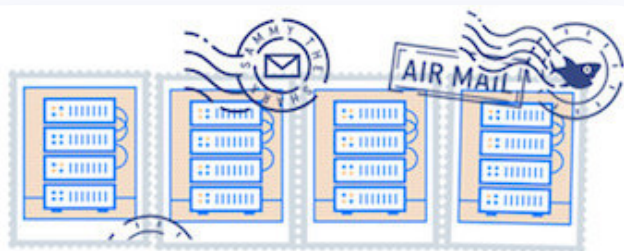
[MySQL](#)

[Docker](#)

[Kubernetes](#)

[All tutorials →](#)

[Talk to an expert →](#)



Get our biweekly newsletter

Sign up for Infrastructure as a Newsletter.

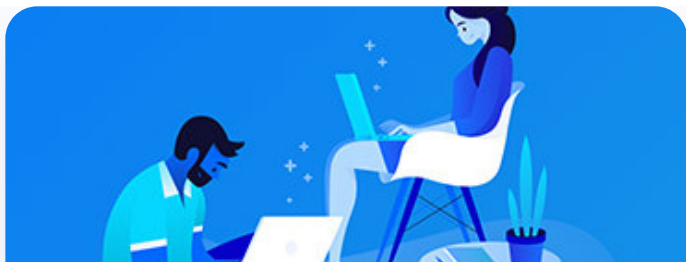
[Sign up →](#)



Hollie's Hub for Good

Working on improving health and education, reducing inequality, and spurring economic growth? We'd like to help.

[Learn more →](#)



Become a contributor

You get paid; we donate to tech nonprofits.

[Learn more →](#)

Featured on Community

- [Kubernetes Course](#)
- [Learn Python 3](#)
- [Machine Learning in Python](#)
- [Getting started with Go](#)
- [Intro to Kubernetes](#)

DigitalOcean Products

- [Cloudways](#)
- [Virtual Machines](#)
- [Managed Databases](#)
- [Managed Kubernetes](#)
- [Block Storage](#)
- [Object Storage](#)
- [Marketplace](#)
- [VPC](#)
- [Load Balancers](#)

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you’re running one virtual machine or ten thousand.

Learn more →

The screenshot shows the DigitalOcean dashboard for a project named 'jonsmith'. The left sidebar contains navigation options: PROJECTS (jonsmith, New Project), MANAGE (Apps, Droplets, Functions, Kubernetes, Volumes, Databases, Spaces, Container Registry, Images, Networking, Monitoring, Add-Ons), Billing, Support, Settings, and API. The main content area has a search bar, a 'Create' button, and a user profile for 'jonsmith' with an estimated cost of \$0.00. Below this, there are tabs for Resources, Activity, and Settings. The Resources tab is active, showing a list of 'SPACES (3)': 'jon-db' (https://jon-db.nyc3.digitaloceanspaces.com), 'jon-website' (https://jon-website.nyc3.digitaloceanspaces.com), and 'jon-backups' (https://jon-backups.nyc3.digitaloceanspaces.com). At the bottom, there are sections for 'Create something new' (Create a Managed Database, Start using Spaces, Spin up a Load Balancer) and 'Learn more' (Product Docs, Tutorials, API & CLI Docs, Ask a question).

Get started for free

Enter your email to get \$200 in credit for your first 60 days with DigitalOcean.

Email address

Send My  mo

New accounts only. By submitting your email you agree to our [Privacy Policy](#).

Company



Products



Community



Solutions



Contact



© 2023 DigitalOcean, LLC.

