Install VSFTPD on Ubuntu 20.04

Updated Thursday, March 9, 2023, by Cameron Laird

What is VSFTPD?

VSFTPD (very secure FTP daemon) is an open-source FTP (File Transfer Protocol) server that is the default FTP server for several prominent Linux distributions. VSFTPD is widely believed to be as secure as any competitive FTP server. VSFTPD supports TLS (Transport-Layer Security), FTPS (File Transfer Protocol Secure), and IPv6.

VSFTPD is important because several prominent platforms, including the WordPress content manager, rely on FTP for crucial workflows. It is widely used in "vertical markets" like accounting, architecture, construction, medicine, and transcription to move, share, and archive large files. FTP allows a remote computer to connect to a server, examine parts of the server's filesystem, retrieve files, and upload files. While more modern protocols offer advantages in security, performance, and convenience, FTP at its best is a fast and well-established file-sharing platform.

In this Guide

This guide demonstrates:

- How to install VSFTPD on Ubuntu 20.04
- Where to adjust the configuration options that VSFTPD makes available, along with some initial recommended options
- How to download files over FTP
- How to upload files over FTP

Before You Begin

This guide assumes that you have access to a server running Ubuntu 20.04 that you can install the FTP server on and upload files to. To create a server on Linode, follow the Creating a Compute Instance and Setting Up and Securing a Compute Instance guides. Be sure to add a limited Linux user to issue the commands in this guide from.

> **Note**
>
> This guide is written for a non-root user. Commands that require elevated privileges are prefixed with `sudo`. If

VSFTPD Installation Steps

Install VSFPTD on Ubuntu 20.04, along with some supporting packages:

2. Update your system's packages:

```
sudo apt update
```

3. Install the VSFTPD server, the FTP command line client, and the UFW firewall. The FTP command line client is used in this guide to issue local test connections to the VSFTPD server:

```
sudo apt install vsftpd ftp ufw -y
```

4. Set VSFTPD to start whenever your server boots:

```
sudo systemctl enable vsftpd
```

5. Launch VSFPTD:

```
sudo systemctl start vsftpd
```

6. Verify that VSFTPD is running properly after this installation:

```
sudo systemctl status vsftpd
```

You should see output similar to:

```
vsftpd.service - vsftpd FTP server
    Loaded: loaded (/usr/lib/systemd/system/vsftpd.service, enabled)
    Active: active (running)
```

Create an FTP User

To see VSFTPD in action–a kind of "Hello, world" for FTP–create a special-purpose user on your server:

2. Create a Linux user named `ftp_client`:

```
sudo useradd -m ftp_client
```

3. Set the password for your new user:

```
sudo passwd ftp_client
```

4. Create an example text file under the home directory of the new `ftp_client` user:

```
sudo -u ftp_client sh -c 'echo "This is the content in the file." > /home/ftp_client/
```

5. Open an FTP connection to the VSFTPD server running on localhost. This syntax is similar to connections you would make from remote systems, which is demonstrated later in this guide:

```
ftp localhost
```

6. You are prompted for your FTP username ('ftp_client'), and then prompted for this user's password (set in step 2 of this section). After entering this information successfully, an `ftp>` command prompt appears:

```
ftp localhost
Connected to localhost.
220 (vsFTPd 3.0.3)
Name (localhost:linode_user): ftp_client
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

7. Verify that the sample file `testfile.txt` in the `ftp_client` user's home directory is visible from the FTP connection:

```
ls /home/ftp_client
```

8. The output resembles:

```
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
-rw-rw-r--    1 1002     1002            33 Aug 05 16:39 testfile.txt
226 Directory send OK.
```

9. Close the ftp client with the `exit` or `quit` commands:

```
exit
```

```
221 Goodbye.
```

You have verified that your VSFTPD accepts connections. The next sections show where you can configure more sophisticated account management, encryption, and security restrictions:

How to Restart VSFTPD

VSFTPD is restarted via systemctl:

```
sudo systemctl restart vsftpd
```

When VSFTPD starts or restarts, it reads from the current configuration files for the service, which are detailed in the next section.

VSFTPD's Configuration File

In Ubuntu and other common distributions, VSFTPD's configuration is located in `/etc/vsftpd.conf`. When updating the configuration file, follow these steps:

2. *(Recommended)* Back up the current configuration by making a copy:

    ```
    sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bak
    ```

3. Edit the `/etc/vsftpd.conf` configuration file in your preferred text editor.

4. Restart VSFTPD to activate the changes:

    ```
    sudo systemctl restart vsftpd
    ```

As its name promises, one of VSFTPD's goals is security. It offers a wide range of settings to help match a range of security and business requirements. The official manual page outlines all configuration options available. The next section introduces some relevant permissions.

VSFTPD User Permissions

To make VSFTPD useful for some real-world use-cases, you can adjust some of the default options set in `vsftpd.conf`:

2. Open `/etc/vsftpd.conf` in your text editor.

3. Locate the following recommended options within the file. Some may be commented out. If an option is commented out, remove the comment (by removing the `#` prefix at the beginning of the line). Some options may not be listed in the file. For these options, add a new line with the option. When finished making changes, save the file.

    - `anonymous_enable`: Set this option to `NO` (`anonymous_enable=NO`). This blocks anonymous logins to the FTP server.

    - `local_enable`: Set this option to `YES` (`local_enable=YES`). This allows you to log in as the users specified in your system's `/etc/passwd` file.

    - `write_enable`: Set this option to `YES` (`write_enable=YES`). This allows you to make changes to the filesystem via FTP, including uploading files.

4. Restart VSFTPD to activate these changes:

    ```
    systemctl restart vsftpd
    ```

> **Note**
>
> A common strategy for securing user accounts is to use VSFTPD's `userlist_enable`, `userlist_file`, and `userlist_deny` attributes. These can be configured to only allow a selected subset of local accounts to establish FTP sessions.

> **Important**
>
> By default, FTP connections are communicated in clear text and not encrypted. Investigate the SSL options available to VSFTPD to set up encryption.

VSFTPD Log File

VSFTPD logs its actions. The default location of the log file is `/var/log/vsftpd.log`. The configuration attribute `xferlog_file` controls this location. View its content from time to time to understand the information the logfile preserves:

```
sudo more /var/log/vsftpd.log
```

Downloading with VSFTPD

2. On the server, open an FTP connection to `localhost`:

```
ftp localhost
```

3. Enter the `ftp_client` username and password when prompted.

4. At the FTP command prompt, change directory to the `ftp_client` home directory:

```
cd /home/ftp_client
```

```
250 Directory successfully changed.
```

5. Use the `get` command to retrieve the test file that was created in the Create an FTP User section:

```
get testfile.txt
```

```
local: testfile.txt remote: testfile.txt
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for testfile.txt (33 bytes).
226 Transfer complete.
33 bytes received in 0.00 secs (947.8400 kB/s)
```

6. Exit the FTP session:

```
exit
```

7. Observe that the file is now present in your original user's home directory:

```
ls -l
```

```
total 4
-rw-rw-r-- 1 linode_user linode_user 33 Aug  5 16:59 testfile.txt
```

Uploading with VSFTPD

2. Create a text file in your system's `/tmp` directory:

```
cd /tmp
echo "This is sample content for uploading through FTP." > testfile2.txt
```

3. Open an FTP connection to `localhost`. Enter the `ftp_client` username and password when prompted:

```
ftp localhost
```

4. Within the FTP session, upload the file created in step 1 by using the `put` command:

```
put testfile2.txt
```

```
200 EPRT command successful. Consider using EPSV.
150 Ok to send data.
226 Transfer complete.
50 bytes sent in 0.00 secs (2.3842 MB/s)
```

> **Note**
>
> The `write_enable` option for VSFTPD must be set to `YES` for this file upload operation to succeed. Review the VSFTPD's Configuration File section for help with setting this option.

5. Exit the FTP session:

```
quit
```

6. Verify that the sample file `testfile2.txt` was uploaded to the `ftp_client` home directory via FTP:

```
ls /home/ftp_client
```

7. The output should resemble:

```
testfile.txt testfile2.txt
```

Connect to Your Server using VSFTPD

This section shows how to allow connections from remote clients to VSFTPD by configuring the UFW firewall. The UFW firewall was installed as part of the VSFTPD Installation Steps section.

2. Before enabling VSFTPD connections, make sure SSH connections are also allowed:

```
sudo ufw allow ssh
```

3. Allow VSFTPD traffic on ports 20 and 21:

```
sudo ufw allow from any to any port 20,21 proto tcp
```

4. Enable the UFW firewall:

```
sudo ufw enable
```

5. Use any convenient FTP client on your desktop to connect to the VSFTPD server. When connecting, specify the `ftp_client` user and the IP address of the server (e.g. `ftp://ftp_client@ip_address`).

More Information

You may wish to consult the following resources for additional information on this topic. While these are provided in the hope that they will be useful, please note that we cannot vouch for the accuracy or timeliness of externally hosted materials.

- VSFTPD.CONF man page

This page was originally published on Tuesday, December 14, 2021.

UBUNTU

Your Feedback Is Important

Let us know if this guide was helpful to you.

Provide Feedback

Join the conversation.

Read other comments or post your own below. Comments must be respectful, constructive, and relevant to the topic of

the guide. Do not post external links or advertisements. Before posting, consider if your comment would be better addressed by contacting our Support team or asking on our Community Site.
The Disqus commenting system for Linode Docs requires the acceptance of Functional Cookies, which allow us to analyze site usage so we can measure and improve performance. To view and create comments for this article, please update your Cookie Preferences on this website and refresh this web page. Please note: You must have JavaScript enabled in your browser.

Site Map

Support

Partners

Careers

Press Center

Legal Center

System Status