How to setup and use FTP Server in Ubuntu Linux

Contents[Hide]

1. Introduction

FTP is a service known to almost anyone who works with Internet on daily basis. This guide will describe in detail how to setup an FTP server on Ubuntu Linux in simple to follow steps. We will show how to setup an FTP server in normal and stand-alone mode. We will also provide some security tips for your new FTP setup. This guide will use lightweight and efficient FTP server **vsFTPd,** which is designed to withstand on servers with high load.

2. Conventions

Occasionally in this article we will refer to **vsFTPd** simply as FTP server.

3. What is FTP

For those who are not familiar with FTP here is a short description of what this service offers. FTP stand for **F**ile **T**ransfer **P**rotocol. As the name suggest this network protocol allows you to transfer files or directories from one host to another over the network whether it is your LAN or Internet.

Main features of **vsFTPd** are: Virtual IP configurations, Virtual users, Standalone or inetd operation, Powerful per-user configurability, Bandwidth throttling, Per-source-IP configurability, Per-source-IP limits, IPv6 andEncryption support through SSL integration.

4. Installation of FTP server in Ubuntu

As in any other Ubuntu like Linux systems, installation of the FTP server comes down to a single command. Open up your terminal and enter the command:

```
$ sudo apt-get install vsftpd
```

The above command will install and start the ftp server on your Linux system.

```
Setting up vsftpd (2.3.5-1ubuntu2) ...
vsftpd start/running, process 1891
```

5. Stand-alone vs Normal FTP mode

5.1. Stand-alone mode

By default vsftpd starts for stand-alone mode. In stand-alone mode the service you run on your server has its own startup script called daemon. In case of vsftpd it is a **/etc/init.d/vsftpd** . This stand-alone daemon will take control of the FTP service the moment it gets started. The vsftpd daemon provides the administrator with a couple of commands to manage the vsftpd FTP server:

- **start** or **stop** - used to start and stop the ftp server
- **status** - provides more information about the current status of your FTP server
- **restart** - this is an alternative to consequent execution of **stop** and **start** commands. If the FTP server is down, the **restart** command will start it.
- **reload** - this will instruct the FTP server to reload and apply all new configurations. The difference between **reload** and **restart** is that with **reload** you are not shutting down your FTP but rather only reloading it with new configuration settings and at the same time ensure continuous uptime.

To start, restart, reload or get the status of our FTP server we can use the **service** utility:

```
$ sudo service vsftpd start
```

The above syntax applies to all other commands so simply replace start with a command you wish to execute.

5.2. Normal mode

The different approach is to start the vsftpd FTP service in a normal mode, which means that the **xinetd** superserver will be responsible for keeping up your FTP service alive. To start the vsftpd FTP server in the normal mode we first need to install the **xinetd** superserver:

```
$ sudo apt-get install xinetd
```

The above command will install and start the **xinetd** superserver on your system. The chances are that you already have **xinetd** installed on your system. In that case you can omit the above installation command.

Next, create a file called vsftpd in /etc/xinetd.d/ with the following content:

```
service ftp
{
        disable                 = no
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/vsftpd
        per_source              = 5
        instances               = 200
        no_access               = 10.1.1.10
        banner_fail             = /etc/vsftpd.busy
        log_on_success          += PID HOST DURATION
        log_on_failure          += HOST
}
```

and at the same time alter any options to match your situation. The options you need to keep an eye on are:

- **server** - type "$ which vsftpd" on a command line to get a correct path
- **no_access** - this will block any hosts with IP address defined by this directive
- **banner_fail** - this can be a path to any text file with a text to show to any blocked IP address

In the next step we need to edit the FTP server's configuration file **/etc/vsftpd.conf** and change the following line:

```
listen=YES
```

to

```
listen=NO
```

This will instruct the FTP server not to open any ports and leave it entirely to the **xinetd** superserver. Before you start the FPT server in the normal mode using **xinetd** make sure to shutdown the vsftpd daemon first with:

```
$ sudo service vsftpd stop
```

Now, that we are ready we can start the FTP server in the normal mode with:

```
$ sudo service xinetd restart
```

To confirm that your FTP server started normally test and open port 21 with the **netstat** command:

```
$ netstat -ant | grep 21
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
```

6. Making first FTP connection

Whether you start your FTP server in a stand-alone or normal mode you should be able to make your first local ftp connection. The vsftpd FTP server by default allows anonymous access. This is the user we are going to employ to make our first FTP connection to test the FTP server installation. To do that, simply execute the ftp command with a localhost as an argument:

```
$ ftp localhost
Connected to localhost.
220 (vsFTPd 2.3.5)
Name (localhost:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
ftp> quit
221 Goodbye.
```

The above only confirms that your FTP server is setup correctly on the local system. Before you go deeper into the FTP server configuration you should test your new FTP server installation using some remote hosts.

7. Configuring FTP server

In the following section we list some basic and simple to configure options.

Important: When making any changes to the FTP server configuration make sure to restart / reload in order to apply all changes you've made.

7.1. Configuring user access

With the vsftpd FTP server you have the option to leave the FTP service authentication for only anonymous access or you can allow users , defined in /etc/passwd or in relevant access list, to login.

7.1.1. Anonymous FTP access

By default the vsftpd FTP server is configured for an anonymous access only. If this is what your intention is for the FTP server to be used for, you can make anonymous access even easier by disabling the password requirement. The most secure option for the FTP server is not to allow users authenticate with the password in a simple clear text format. To disable the requirement for an anonymous password simply edit the FTP server's configuration file /etc/vsftpd.conf and set directive no_anon_password to NO:

```
no_anon_password=YES
```

7.1.2. Local users FTP access

By now your server should be set to disallow any access except for an anonymous user. Set local_enable directive in the configuration file /etc/vsftpd.conf to allow login all users defined in /etc/passwd. The default is NO.

```
local_enable=YES
```

Now any user defined in /etc/passwd will be able to login with their relevant passwords.

```
$ ftp localhost
Connected to localhost.
220 (vsFTPd 2.3.5)
Name (localhost:root): lubos
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> quit
221 Goodbye.
```

7.1.3. User Access list

First create an access list to be used with the vsFTPd FTP server. Normally, you would define such list in /etc/ directory. Use some arbitrary file name and list all user names in this file, which you wish to allow or deny access. For example, let's define a new list with a single user "lubos":

```
echo lubos > /etc/vsftpd.userlist
```

Next, define a new user list in the configuration file /etc/vsftpd.conf and enable userlist_enable directive:

```
userlist_file=/etc/vsftpd.userlist
userlist_enable=YES
userlist_deny=NO
```

Simply, just by defining and enabling a user list any users listed in /etc/vsftpd.userlist will have access denied to your FTP server.

To reverse this option and only allow users listed in /etc/vsftpd.userlist set userlist_deny configuration directive to NO. By doing so you only allow users defined in /etc/vsftpd.userlist to access your system. **The format of /etc/vsftpd.userlist file is one user per line.**

```
$ ftp localhost
Connected to localhost.
220 (vsFTPd 2.3.5)
Name (localhost:root): lubos
530 Permission denied.
Login failed.
ftp>
```

7.2. Changing default port number

By default any FTP server listens on standard port 21 for user authentication and port 20 for data transfer. The vsFTPd FTP server is no exception. To change default listening port simply change your /etc/vsftpd.conf configuration file and alter directive listen_port. Thus, to change your FTP server to listen to on port 2121 simply add the following directive to you configuration file:

```
listen_port=2121
```

followed by the FTP server restart.

However, this only applies if you run vsFPTd in stand-alone mode. In case that you run your FTP server in a normal mode using the **xinetd** superserver and you wish to change listening port to 2121, edit line FTP line /etc/services file and change it from 21 to 2121 and restart **xinetd.**

```
$ sudo service xinetd restart
```

Now, you should be able to see your FTP server to listen on port 2121:

```
$ netstat -ant | grep 2121
tcp        0      0 0.0.0.0:2121            0.0.0.0:*               LISTEN
```

7.3. Other Configuration Options

The vsFTPd server comes with a number of configuration options to suit your needs. Here we list some other worth to mention configuration options:

- **max_clients** - This option sets the maximum number of users allowed to use your FTP server at the same time. 0 = unlimited.
- **max_per_ip** - set the maximum number of users from the same IP address
- **download_enable** - if set to NO any download request will be denied

8. Conclusion

Currently vsFTPd comes with around 125 configuration options. This makes this FTP server to be extremely versatile and at the same time very easy to use and configure. Whether you are setting up the FTP server for local home use, within a company or on some remote server, be sure that vsFTPd will accommodate you. The vsFTPd FTP server can also be turned to sftp, but this we will discuss in the next tutorial, so do not forget to subscribe to our RSS.