



ServerPilot



- [Built by](#)
- [Free Trial](#)
- [Features](#)
- [Use Cases](#)
- [Pricing](#)
- [Docs](#)
- [Sign in →](#)



[← ServerPilot Docs](#)

How to Run a Simple FTP Server

This is an advanced tutorial.
ServerPilot cannot provide any support for running an FTP server.

The easiest way to run an FTP server is with a Python script that starts a simple FTP server. This tutorial will show you how to do that.

There are two steps to running an FTP server:

1. Configuring and starting the FTP server
2. Opening up your firewall to allow connections to your FTP server

Configuring and Starting an FTP Server

First, SSH in to your server as root and install the Python `pyftplib` library.

On Ubuntu 20.04, use this command:

```
sudo apt-get install python3-pyftplib
```

On Ubuntu 14.04, 16.04, and 18.04, use this command:

```
sudo apt-get install python-pyftplib
```

Next, log out of your server as root. The rest of your steps should be done while logged in as your app's system user.

Now, SSH into your server as your app's system user and create a file named `ftpserver.py` in the user's home directory. That is, create this file:

```
/srv/users/SYSUSER/ftpserver.py
```

In that file, put the following contents and change the name, password, and directory for the FTP user that are defined near the top of the file:

```
from pyftplib.authorizers import DummyAuthorizer
from pyftplib.handlers import FTPHandler
from pyftplib.servers import FTPServer

# The port the FTP server will listen on.
# This must be greater than 1023 unless you run this script as root.
FTP_PORT = 2121

# The name of the FTP user that can log in.
FTP_USER = "myuser"

# The FTP user's password.
FTP_PASSWORD = "change_this_password"

# The directory the FTP user will have full read/write access to.
FTP_DIRECTORY = "/srv/users/SYSUSER/apps/APPNAME/public/"

def main():
    authorizer = DummyAuthorizer()

    # Define a new user having full r/w permissions.
    authorizer.add_user(FTP_USER, FTP_PASSWORD, FTP_DIRECTORY, perm='elradfmw')

    handler = FTPHandler
    handler.authorizer = authorizer

    # Define a customized banner (string returned when client connects)
    handler.banner = "pyftplib based ftpd ready."

    # Optionally specify range of ports to use for passive connections.
    #handler.passive_ports = range(60000, 65535)

    address = ('', FTP_PORT)
    server = FTPServer(address, handler)

    server.max_cons = 256
    server.max_cons_per_ip = 5

    server.serve_forever()

if __name__ == '__main__':
    main()
```

You can now start the FTP server.

On Ubuntu 20.04, use this command to start the server:

```
python3 /srv/users/SYSUSER/ftpserver.py >>/srv/users/SYSUSER/ftpserver.log 2>&1 &
```

On Ubuntu 14.04, 16.04, and 18.04, use this command to start the server:

```
python /srv/users/SYSUSER/ftpserver.py >>/srv/users/SYSUSER/ftpserver.log 2>&1 &
```

The above command will start the FTP server with a log file named `ftpserver.log`.

For more information on customizing this FTP server, see the [pyftplib documentation](#).

You can verify your FTP server is running with the command:

```
netstat -npl --inet | grep 2121
```

Your output should look like this:

```
tcp        0      0 0.0.0.0:2121          0.0.0.0:*            LISTEN      29972/python
```

which shows your FTP server listening on port 2121.

To stop the FTP server, use the `kill` command with the FTP server processes' PID, which you can find with this command:

```
ps -ef | grep ftpserver | grep -v grep | awk '{print $2}'
```

Configuring Your Firewall to Allow FTP Server Access

Now that you've started your FTP server, you need to customize your server's firewall to open up the port you ran your FTP server on. For example, if you used port 2121, you'd need to open port 2121 in your server's firewall.

Additionally, if you uncommented the passive port range line in the above script to enable passive FTP, you also need to open ports 60000-65535 in your server's firewall.

For information on how to do this, see our article on [customizing your server's firewall](#).

Starting Your FTP Server on Boot

To start your FTP server automatically when your server is rebooted, SSH in as your app's system user and [create a cron job](#) like the following:

```
@reboot python /srv/users/SYSUSER/ftpsrv.py >>/srv/users/SYSUSER/ftpsrv.log 2>&1
```

Last updated: November 3, 2020

Launch your first site in 5 minutes

[Sign Up](#) [Contact Us](#)

Overview

[WordPress Hosting](#)

[Magento Hosting](#)

[Laravel Hosting](#)

[PHP Hosting](#)

[Terms](#)

[Privacy](#)

[GDPR](#)

© Less Bits

[Pricing](#)

[Features](#)

[Company](#)

[Blog](#)

[Security](#)

[API](#)

[AutoSSL](#)

Resources

[Documentation](#)

[Sign Up](#)

[Getting Started](#)

[Contact](#)

[Referral Program](#)

[Logos and Badges](#)

[WordPress Firewall](#)

[Managed Hosting](#)

Articles

[What Is ServerPilot?](#)

[How ServerPilot Works](#)

[WordPress on DigitalOcean](#)

[DigitalOcean Control Panel](#)

[cPanel Alternative](#)

[Plesk Alternative](#)

[Managed Cloud on AWS and DigitalOcean](#)

[English](#)

[Deutsch](#)

[Español](#)

[Français](#)

[Italiano](#)

[日本語](#)

[Português](#)

[Русский язык](#)