

Should Django be hosted on PythonAnywhere?

PythonAnywhere is suitable for Django development but is it easy to set up? Let's find out together!



Django is a popular web framework that simplifies web application development, while PythonAnywhere is a cloud-based platform that provides an online environment to run and host Python applications.

If you're considering hosting your Django project on PythonAnywhere, the article tells you the benefits and limitations of the platform, along with step-by-step instructions for setting up a Django project.

Benefits of hosting Django projects on PythonAnywhere

Signing up for PythonAnywhere is easy, and the platform offers a number of useful features for hosting Django projects. Its friendly interface allows users to easily deploy and manage the application, while the pre-configured Python environment simplifies the development process.

It provides a convenient web-based console, SSH access and scheduled tasks, allowing users to take full advantage of the hosting service. In addition, PythonAnywhere provides powerful data archiving, complete with backup and restore services to keep user data safe.

Limitations when hosting Django projects on PythonAnywhere

In addition to the benefits, PythonAnywhere also has some challenges and limitations that programmers should consider before using the platform:

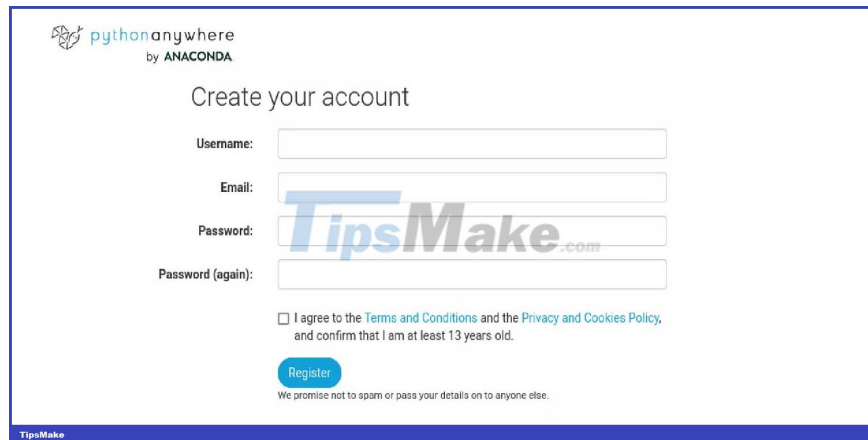
1. **Limited resources** . that means it may not be suitable for large and complex Django projects for the free plan. This can slow down application performance, especially during peak traffic periods.

2. **Limited server control** . Developers cannot configure the server according to their needs. This can limit the flexibility of the application. This can be a big problem when developing complex projects that require custom configuration.
3. Performance issues during peak access times, such as slow response times, can affect the user experience.

How to host a Django project on PythonAnywhere

Sign up for a PythonAnywhere account

To get started, you need to sign up for a PythonAnywhere account with your email and set up a password.



The screenshot shows the PythonAnywhere account creation page. At the top, it says 'pythonanywhere by ANACONDA'. Below that is the heading 'Create your account'. There are four input fields: 'Username:', 'Email:', 'Password:', and 'Password (again:'. A 'Register' button is located below the password fields. There is also a checkbox for agreeing to terms and conditions, and a small text at the bottom: 'We promise not to spam or pass your details on to anyone else.'

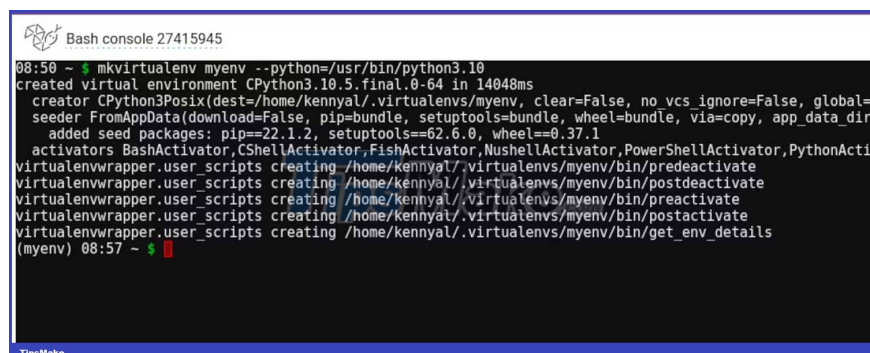
Create new virtual environment for Django project

Once logged in to PythonAnywhere, the first step is to create a new virtual environment for the Django project. This ensures isolation of dependencies for the project from other projects on the platform.

To do the above, navigate to the Consoles tab and select Bash. In the console, enter the following command:

```
mkvirtualenv myenv --python=/usr/bin/python3.10
```

By running this command, you can create a new virtual environment, named 'myenv' (running Python 3.10). If you want to assign a different name to your environment, simply replace 'myenv' with the name of your choice.



```
Bash console 27415945
08:50 ~ $ mkvirtualenv myenv --python=/usr/bin/python3.10
created virtual environment CPython3.10.5.final.0-64 in 14048ms
creator CPython3Posix(dest=/home/kennyal/.virtualenvs/myenv, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=
added seed packages: pip==22.1.2, setuptools==62.6.0, wheel==0.37.1
activators BashActivator,CShellActivator,FishActivator,MyShellActivator,PowerShellActivator,PythonActiv
virtualenvwrapper.user_scripts creating /home/kennyal/.virtualenvs/myenv/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/kennyal/.virtualenvs/myenv/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/kennyal/.virtualenvs/myenv/bin/preactivate
virtualenvwrapper.user_scripts creating /home/kennyal/.virtualenvs/myenv/bin/postactivate
virtualenvwrapper.user_scripts creating /home/kennyal/.virtualenvs/myenv/bin/get_env_details
(myenv) 08:57 ~ $
```

Setting up a Django project on PythonAnywhere using the web interface

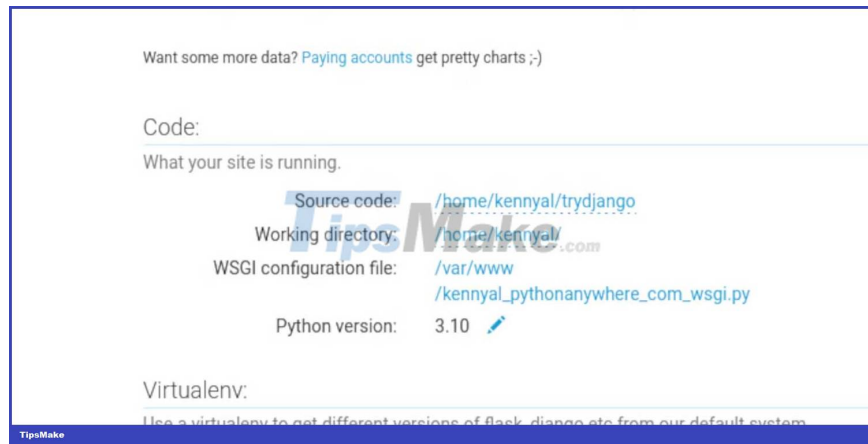
PythonAnywhere provides a web interface for the programmer to set up the project. It includes creating a new web app, installing the active directory, and configuring the virtual environment. Here's how to set up a new web app on PythonAnywhere:

1. **Log in to your PythonAnywhere** account .
2. Navigate to the **Dashboard** tab .
3. Click the **Create a new web app** button .
4. Choose a domain name for the web application. PythonAnywhere provides a free subdomain that you can use for your web application. If you want to use your domain name, you need to sign up for a paid plan. You can buy a domain name on sites like NameCheap, GoDaddy and configure it to point to the PythonAnywhere web app.
5. Select the web framework you want to use. PythonAnywhere has several frameworks available, including Django, Flask, web2py, Bottle and manual configuration. Select **manual configuration** from this list and set up the web app and configure it for your Django project.
6. Select the Python version you want to use for the project.
7. Click the **Next** button . PythonAnyWhere will create a new web app for you and take you to the web app dashboard.
8. **Go to the Virtualenv** section , on the web tab, enter the path to the virtual environment and click **OK** .

Active Directory Setup

Once you've created a new web app on PythonAnywhere, you need to set up the working directory for the Django project's root directory. Detailed steps are as follows:

1. Go to the **Files** tab in the web app dashboard.
2. Click the **New directory** button to create a new directory.
3. Enter a name for the folder, for example 'myproject'.
4. Click the **Create** button to create the folder.
5. Navigate to the newly created folder by clicking on its name in the file browser.
6. Click the **Upload** button to upload the Django project file to that folder.
7. After you have uploaded the project file, click the web tab to change the WSGI configuration file. This system uses the WSGI file in your file.
8. In the **Code** section , click the link to the WSGI configuration file to edit it.



9. Uncomment the Django section and remove all other code.

10. Update the path in the respective variable line to include it in the project directory. For example:

```
path = '/home/username/myproject'
```

11. Save changes to the WSGI configuration file.

Virtual environment configuration

After setting up the working directory, you need to configure the virtual environment to include the necessary packages for the Django project. Follow these steps:

1. Navigate to the Consoles tab in the web app dashboard.
2. Click the Bash console button to open the Bash console.
3. Activate the virtual environment by running the following command:

```
source /path/to/virtual/environment/bin/activate
```

Replace **"/path/to/virtual/environment"** with the path to your virtual environment.

4. After activating the virtual environment, the next step is to install Django and any other required packages using pip. This can be done from the command line on PythonAnywhere. In this console, enter the following command:

```
pip install django
```

Act on installing the latest version of Django. If you want to install a specific version of Django, you can replace Django with "Django==xyz", where xyz is the version number.

5. Install the required packages for your Django project by running the following command:

```
pip install -r requirements.txt
```

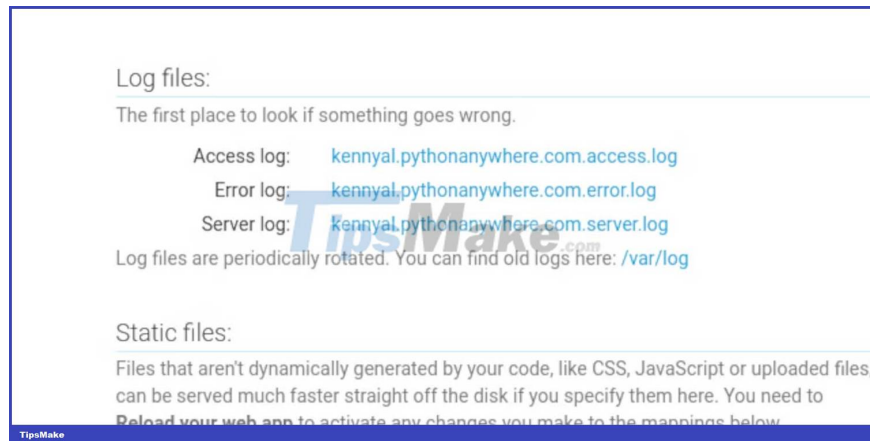
Replace "requirements.txt" with the name of the request file.

6. Wait for the installation of packages to complete.

7. Test the Django project by reloading the web application.

8. Visit your website and check the results.

If you get an error, you can use the error log or the relevant server in the web tab.



Here's what you need to know when hosting Django on PythonAnywhere. Hope the article is useful to you.

You finished reading the article "**Should Django be hosted on PythonAnywhere?**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.