

Before Public Deployment

Make sure your dashboard is fully operational, since the public deployment is just making your local deployment onto a public server.

Instruction

During the whole process, I will use my code as an example.

1. Create a repo(can be private) on Github and upload files

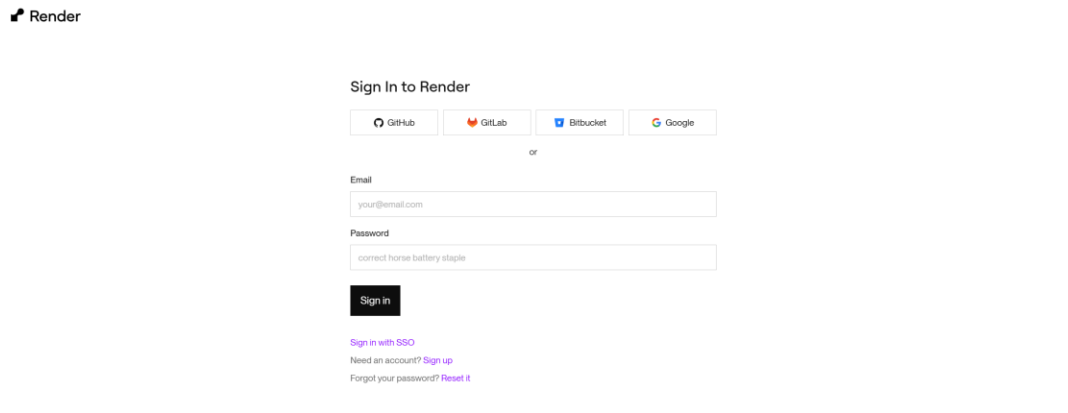
The screenshot displays a GitHub repository interface for a repository named "Dashboard" (Private). The repository has 1 branch (main) and 0 tags. The file list shows several files uploaded via upload, including a "data" folder and files like "data.py", "lineChart.py", "requirements.txt", "spiderChart.py", and "tabs.py". Below this, the "data" folder is expanded, showing a list of CSV files: "ders2_merged.csv", "ders_merged.csv", "gad_merged.csv", "phq_merged.csv", "ptsd_merged.csv", and "who_merged.csv". The "requirements.txt" file is selected, showing its content: "dash==3.0.0", "dash_bootstrap_components==2.0.0", "pandas==2.2.3", "plotly==6.0.1", "scikit_learn==1.6.1", "gunicorn", "dash-tools", "dash_core_components", and "dash_html_components".

Name	Last commit message	Last commit date
..		
ders2_merged.csv	Add files via upload	3 minutes ago
ders_merged.csv	Add files via upload	3 minutes ago
gad_merged.csv	Add files via upload	3 minutes ago
phq_merged.csv	Add files via upload	3 minutes ago
ptsd_merged.csv	Add files via upload	3 minutes ago
who_merged.csv	Add files via upload	3 minutes ago

```
1 dash==3.0.0
2 dash_bootstrap_components==2.0.0
3 pandas==2.2.3
4 plotly==6.0.1
5 scikit_learn==1.6.1
6 gunicorn
7 dash-tools
8 dash_core_components
9 dash_html_components
```

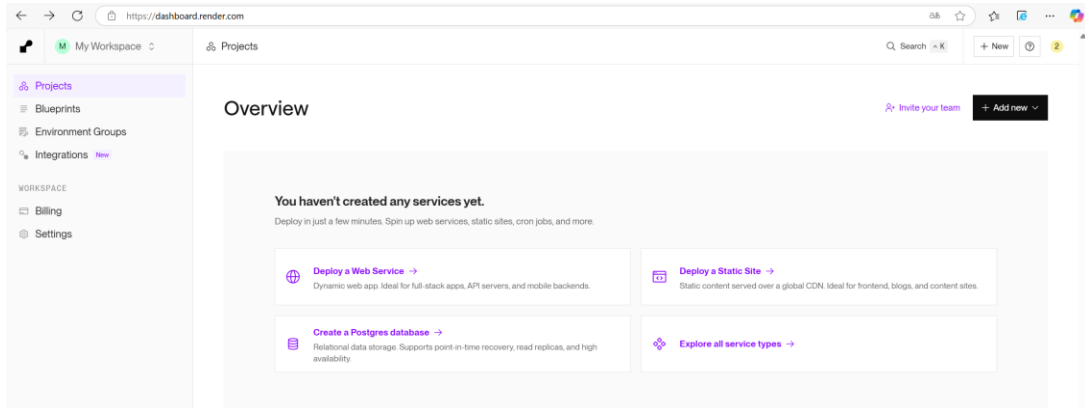
Make sure you have these files: your python files for the local dashboard, csv files for your code and a txt file with packages you import in your code. If you do not know the exact version then just type the name of the package.

2. Sign up an account in Render and create a project



The image shows the 'Sign In to Render' page. At the top left is the Render logo. The main heading is 'Sign In to Render'. Below it are four social login buttons: GitHub, GitLab, Bitbucket, and Google. A small 'or' is centered below these buttons. There are two input fields: 'Email' with the placeholder 'your@email.com' and 'Password' with the placeholder 'correct horse battery staple'. A black 'Sign In' button is below the password field. At the bottom, there are three links: 'Sign in with SSO', 'Need an account? Sign up', and 'Forgot your password? Reset it'.

Choose one you like, I prefer to use Github to sign in because we need our repo



After sign in, go to <https://dashboard.render.com/>. On 'Projects' page chooses 'Deploy a Web Service', follow the guide to deploy

Root Directory Optional	
If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of this directory do not trigger an auto-deploy. Most commonly used with a monorepo .	<div></div> <div>Edit</div>
Build Command	
Render runs this command to build your app before each deploy.	<div>\$ pip install -r requirements.txt</div> <div>Edit</div>
Pre-Deploy Command Optional	
Render runs this command before the start command. Useful for database migrations and static asset uploads.	<div>\$</div> <div>Edit</div>
Start Command	
Render runs this command to start your app with each deploy.	<div>\$ gunicorn tabs:server</div> <div>Edit</div>

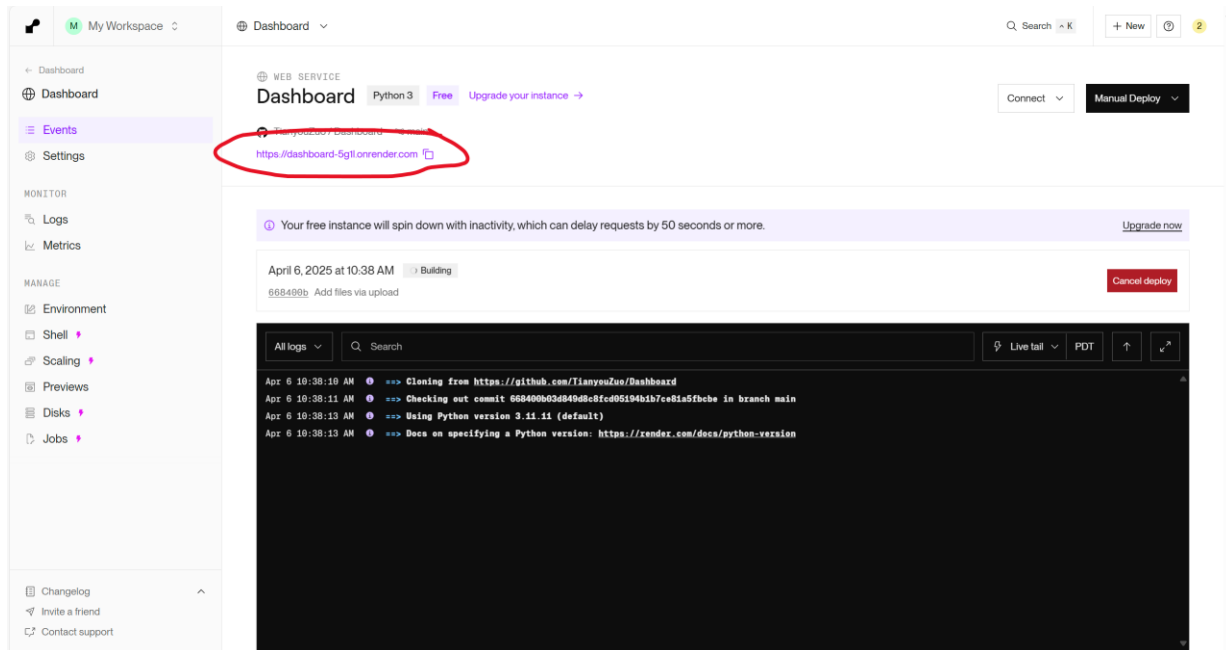
Only need to pay attention to these three options. Root Directory, Build Command and Start Command

Root Directory: If your files on Github is in a folder, then you need to enter the folder name

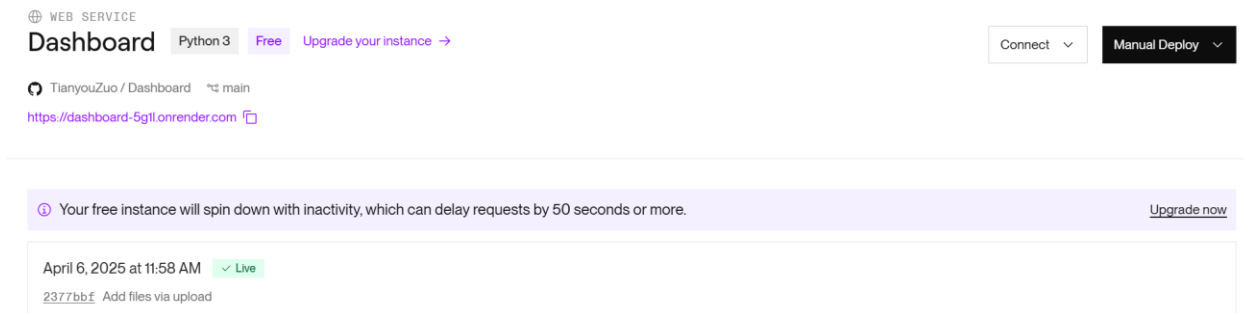
Two Command: make sure the name is same with your files.

Make sure is 'unicorn your_file_name:server'

After deploy, it will start building and deploying








The link in red circle is your public dashboard, you can share your link and let them see the dashboard.



If deployment is successful, it will say 'Live'. Check the dashboard by the link in red circle.

Upgrade your code(you can use it to upload for your initially deployment)

Create a folder on your own PC with all your files you want to put on Github

data	2025/4/8 13:57	文件夹	
 data.py	2025/4/2 21:08	JetBrains PyCharm	3 KB
 lineChart.py	2025/4/6 10:58	JetBrains PyCharm	7 KB
 requirements.txt	2025/4/8 13:56	文本文档	1 KB
 spiderChart.py	2025/4/6 10:53	JetBrains PyCharm	5 KB
 tabs.py	2025/4/3 13:18	JetBrains PyCharm	2 KB

Open your cmd prompt and cd to your folder: `cd your_folder_name`

Following with codes: `git init` #Initialized empty Git repository

`git add ./your_file_name` #Stage all changes made within your project directory

`git commit -m "Initial commit"` # Add a message describing those changes

`git remote add origin YOUR_REMOTE_URL.git` # Add your remote URL here,
you can find it on your Github repository

`git push -u origin main` # Upload the local changes in your Git repository to a
remote repository(basically in main branch)

Finally, on Render, choose 'Deploy latest commit' under Manual Deploy

