

# CH03 Render 雲端服務

# Render 簡介

- 一個 PaaS (Platform as a Service) 雲端服務
  - 程式碼部署到 Render 後，不用管 Linux、Firewall、nginx 的設定
- 能部署 Web Services、Static Sites
- 支援主流的語言與框架
- 亦支援 Dockerfile
  - 理論上可以跑任何的 Docker Container

# Render 簡介 (Cont.)

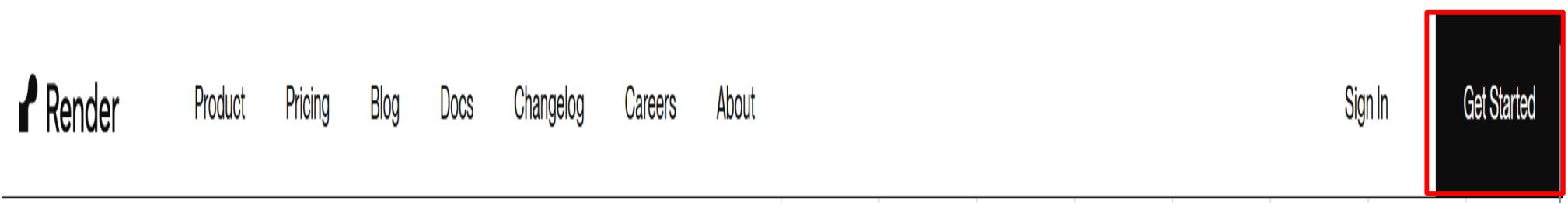
- 所有服務皆有支援 HTTP/2 協定
- 有 Cloudflare 進階防 DDoS
- 全球 CDN
- 可以直接跟 Github 連接
  - 只要有新的 Commit 會自動 Trigger Deploy

# Render 適用情境

- 部署靜態網站 (Static Website)
  - 前 100 GB 流量都免費
- 部署 Django/Flask/Laravel/Node/Rails 等網路服務
  - 前 7 天免費
- 部署 Docker Container
- 需要 PostgreSQL 資料庫
- 需要架論壇/部落格/其他網站

# 註冊 Render 帳號

- <https://render.com/>



# 註冊 Render 帳號 (Cont.)



## Create an account



GitHub



GitLab



Google

or

Email

your@email.com

Password

correct horse battery staple

Create Account

By signing up you agree to our [terms of service](#).

Already have an account? [Sign in](#)

This site is protected by [hCaptcha](#). Its [Privacy Policy](#) and [Terms of Service](#) apply.

# 註冊 Render 帳號 (Cont.)

Almost there!

We've sent you an email at **fordlin@mail.ncut.edu.tw**.

Please follow the instructions in the email.

[Resend Verification Email](#)

# 註冊 Render 帳號 (Cont.)

## Tell us about yourself

**What should we call you?**

Ford Lin

**How will you primarily use Render?**

For personal use



**What type of project are you building?**

Web app



**Where is your project hosted? \***

Project running locally



**Continue to Render**

# Render Dashboard

[Dashboard](#)[Blueprints](#)[Env Groups](#)[Docs](#)[Community](#)[Help](#)[New +](#)

Ford Lin



## Get started in minutes

### Static Sites

Static Sites are automatically served over a global CDN. Add a custom domain and get free, fully-managed SSL.

[New Static Site](#)

### Web Services

Web Services include zero-downtime deploys, persistent storage and PR previews. Scale up and down with ease.

[New Web Service](#)

### Private Services

Private Services are only accessible within your Render network and can speak any protocol.

[New Private Service](#)

### Background Workers

Background Workers are suitable for long running processes like consumers for queues and streaming.

[New Worker](#)

### Cron Jobs

With Cron Jobs, you can schedule any command or script to run on a regular interval.

[New Cron Job](#)

### PostgreSQL

Fully-managed hosted PostgreSQL with internal and external connectivity, and automated daily backups.

[New PostgreSQL](#)

### Redis

A cloud based in-memory key value datastore. Render offers fully managed hosted Redis instances.

[New Redis](#)

### Blueprints

A Blueprint specifies your Infrastructure as Code in a single file. Use it to set up all your services at once.

[New Blueprint](#)

# 靜態網站部署

- 建立 GitHub 帳號
- 創建 Repository
- 新增資料夾
- 設定 index.html
- 上傳檔案
- 登入 Render
- 新增靜態網站

# 靜態網站部署 (Cont.)

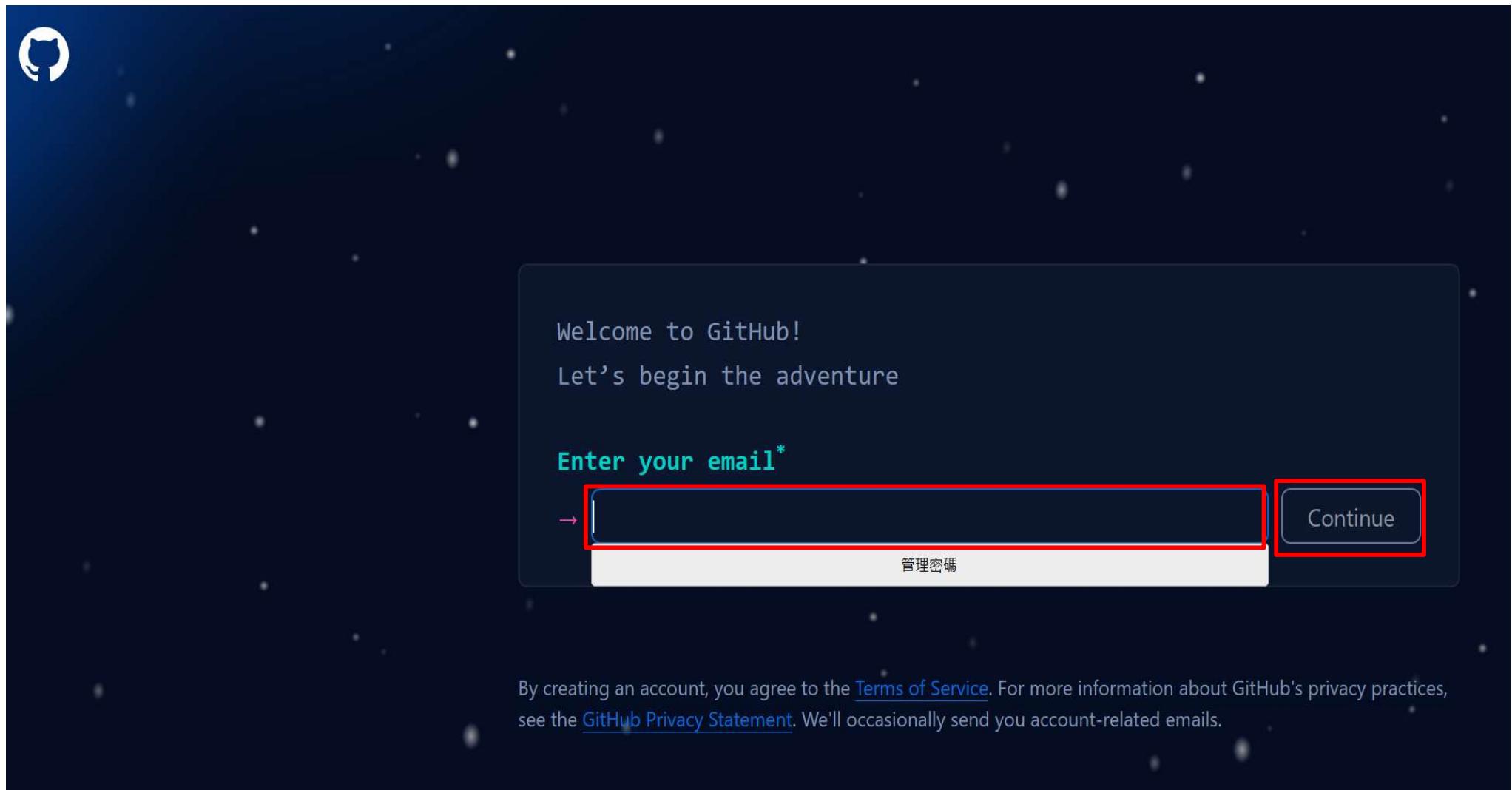
- 授權 Render
- 安裝 Render
- 連接 Repository
- 部署靜態網站
- 新增 Rewrites 規則
- 檢視靜態網站

# 註冊 GitHub 帳號

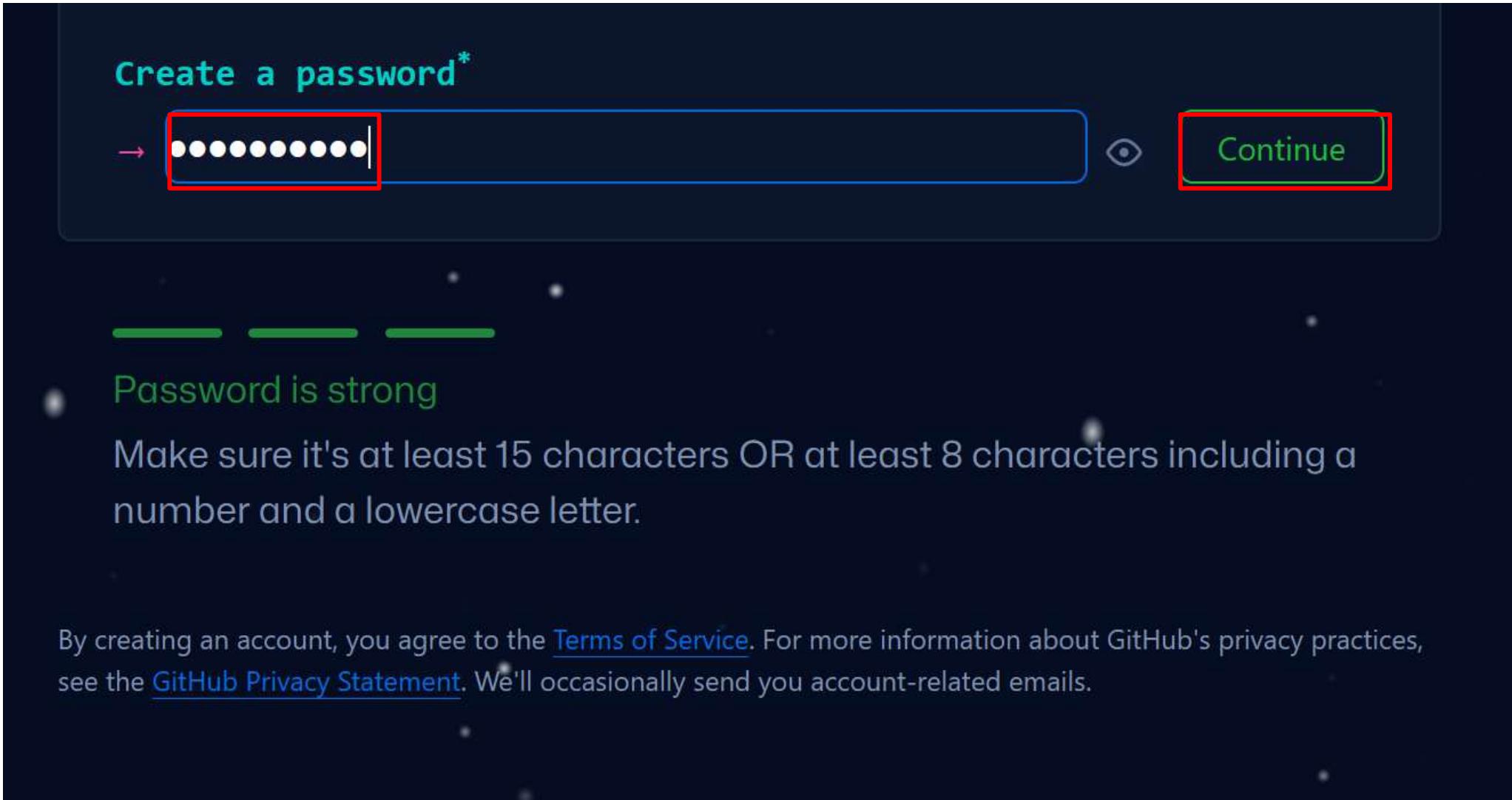
- <https://github.com/>



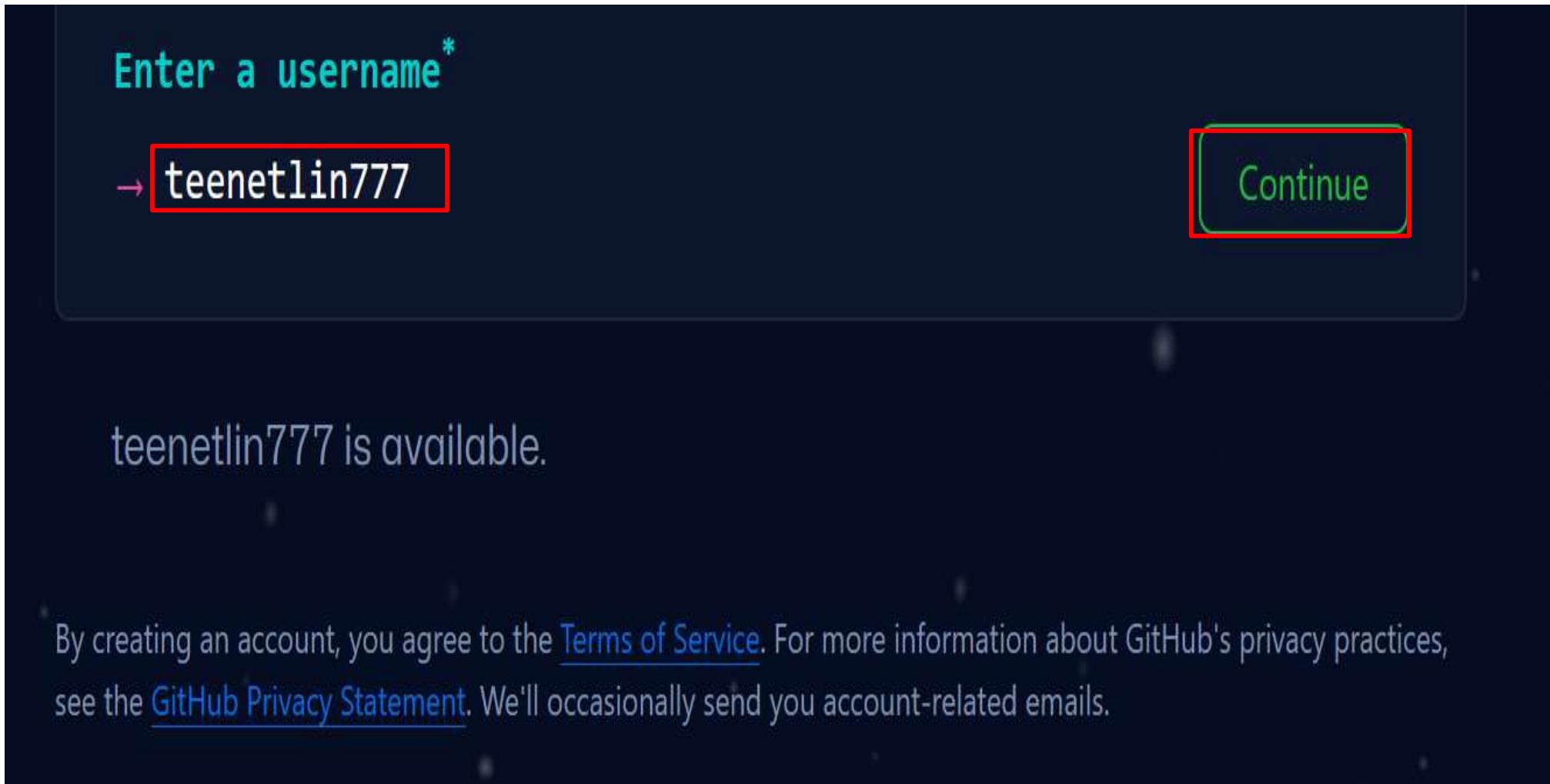
# 註冊 GitHub 帳號 (Cont.)



# 註冊 GitHub 帳號 (Cont.)



# 註冊 GitHub 帳號 (Cont.)



# 註冊 GitHub 帳號 (Cont.)

## Email preferences

Receive occasional product updates and announcements.

Continue

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

# 註冊 GitHub 帳號 (Cont.)

Verify your account

保護您的帳戶

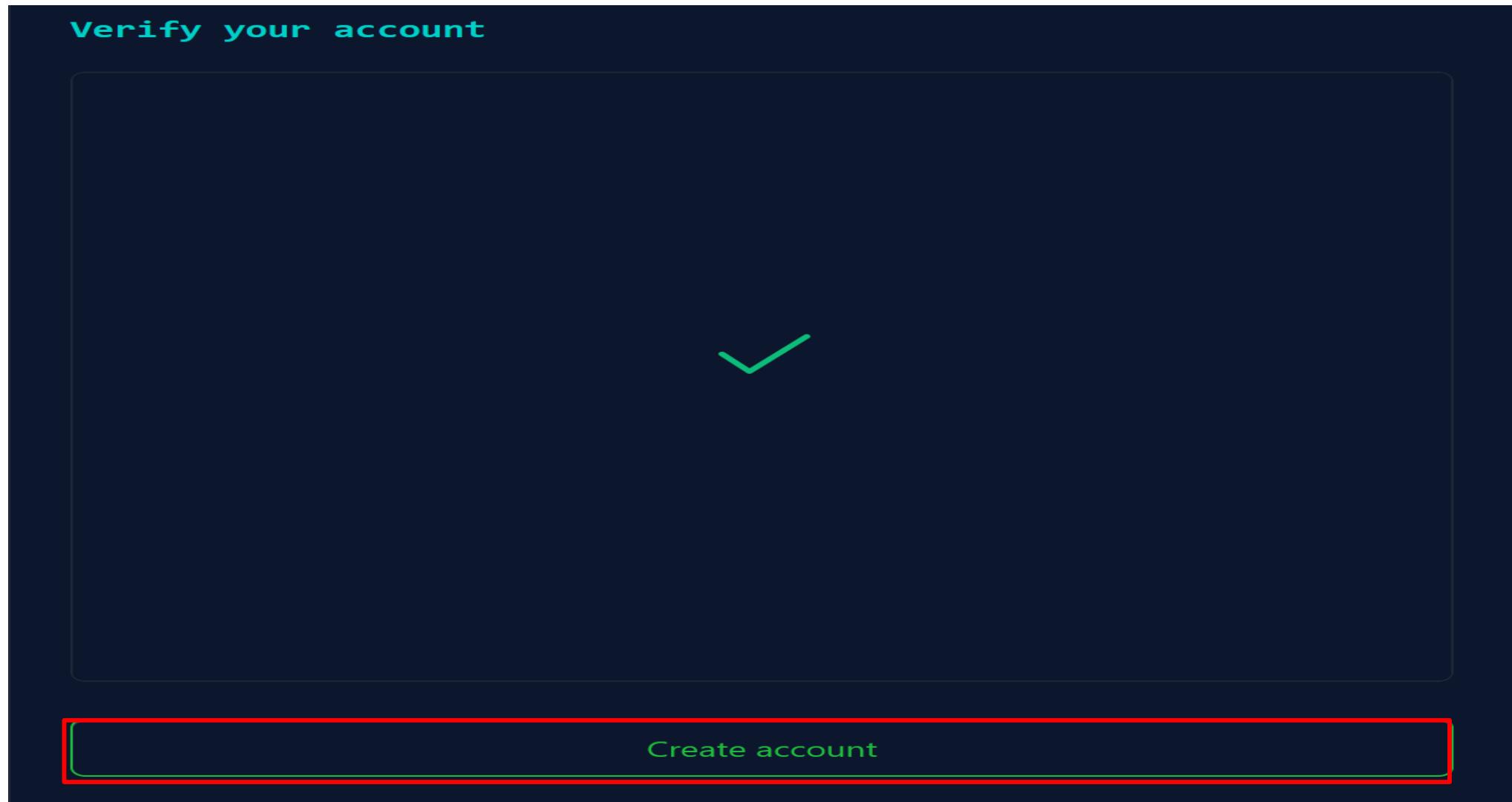
請解答這個謎題 讓我們知道您是真人

驗證

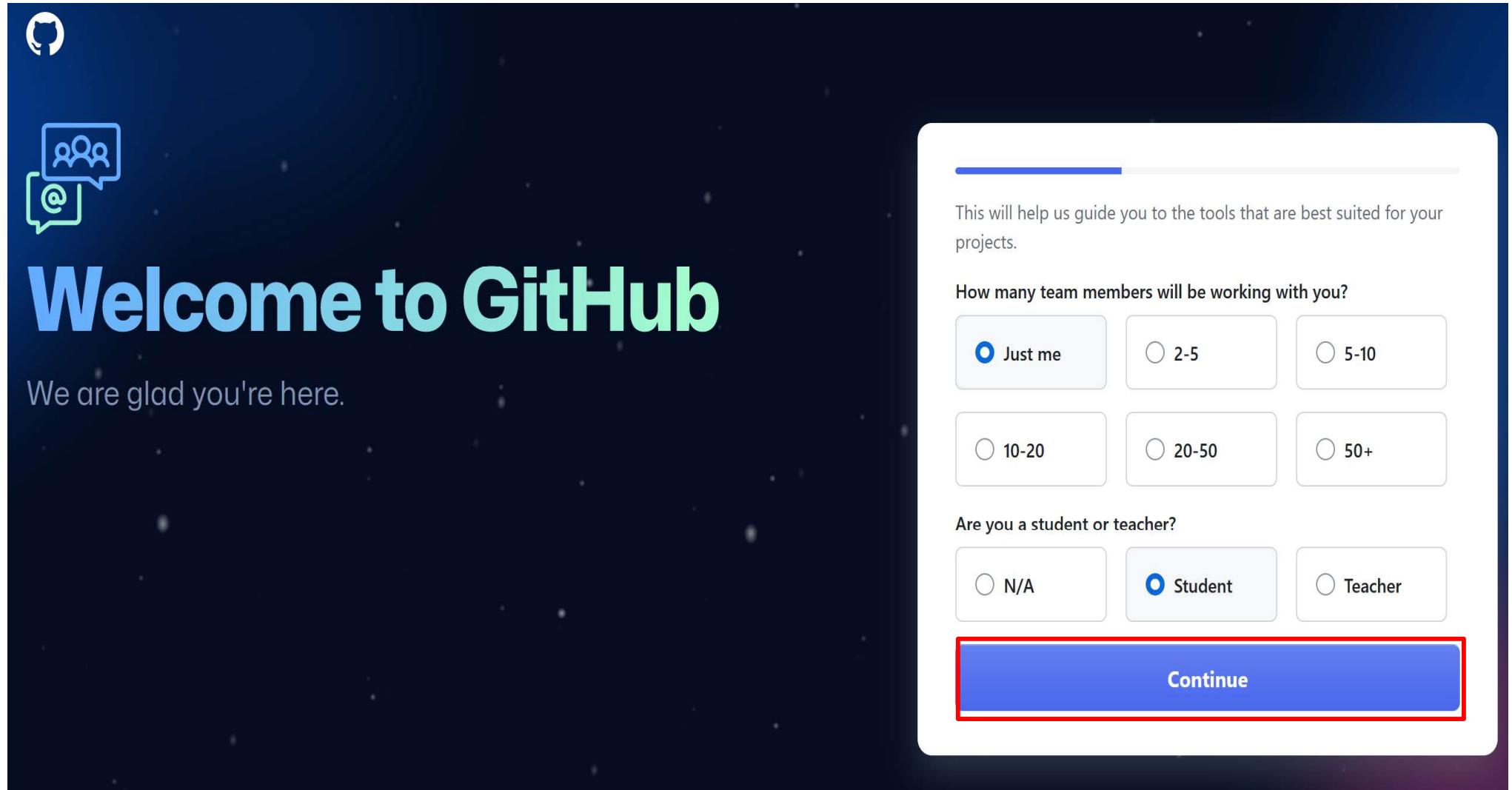


聽覺

# 註冊 GitHub 帳號 (Cont.)



# 註冊 GitHub 帳號 (Cont.)



# 註冊 GitHub 帳號 (Cont.)

## What specific features are you interested in using?

Select all that apply so we can point you to the right GitHub plan.

 Collaborative coding

Codespaces, Pull requests, Notifications, Code review, Code review assignments, Code owners, Draft pull requests, Protected branches, and more.

 Automation and CI/CD

Actions, Packages, APIs, GitHub Pages, GitHub Marketplace, Webhooks, Hosted runners, Self-hosted runners, Secrets management, and more.

 Security

Private repos, 2FA, Required reviews, Required status checks, Code scanning, Secret scanning, Dependency graph, Dependabot alerts, and more.

 Client Apps

GitHub Mobile, GitHub CLI, and GitHub Desktop.

# 註冊 GitHub 帳號 (Cont.)



## Project Management

Projects, Labels, Milestones, Issues, Unified Contribution Graph, Org activity graph, Org dependency insights, Repo insights, Wikis, and GitHub Insights.



## Team Administration

Organizations, Invitations, Team sync, Custom roles, Domain verification, Audit Log API, Repo creation restriction, and Notification restriction.



## Community

GitHub Marketplace, GitHub Sponsors, GitHub Skills, and Electron.

**Continue**

# 註冊 GitHub 帳號 (Cont.)

The screenshot shows a dark-themed registration page for a GitHub account. At the top left, the word "Free" is displayed in white. Below it is a list of six features, each preceded by a right-pointing arrow:

- > Unlimited public/private repositories
- > 2,000 CI/CD minutes/month  
Free for public repositories
- > 500MB of Packages storage  
Free for public repositories
- > 120 core-hours of Codespaces compute
- > 15GB of Codespaces storage
- > Community support

At the bottom of the page is a red rectangular button with the text "Continue for free" in black.

# 創建 Repository

The screenshot shows the GitHub Dashboard. At the top, there is a navigation bar with a menu icon, the GitHub logo, and the word "Dashboard". Below this, a large call-to-action section features the text "Create your first project" and a description: "Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it." Two buttons are present: a green "Create repository" button with a red border, and a blue "Import repository" button. Below this section, there is a "Recent activity" area with a dashed border containing the text: "When you take actions across GitHub, we'll provide links to that activity here."

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repository Import repository

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

# 創建 Repository (Cont.)

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*

/ myweb

✓ myweb is available.

Great repository names are short and memorable. Need inspiration? How about **solid-octo-computing-machine** ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

# 創建 Repository (Cont.)

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

---

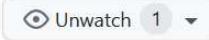
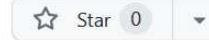
ⓘ You are creating a private repository in your personal account.

---

Create repository

# myweb Repository

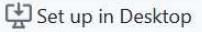
 myweb Private

 **Set up GitHub Copilot**  
Use GitHub's AI pair programmer to autocomplete suggestions as you code.  
[Get started with GitHub Copilot](#)

 **Add collaborators to this repository**  
Search for people using their GitHub username or email address.  
[Invite collaborators](#)

**Quick setup — if you've done this kind of thing before**

 Set up in Desktop    or    [HTTPS](#)    [SSH](#)    <https://github.com/teenetlin777/myweb.git> 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# myweb" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/teenetlin777/myweb.git
git push -u origin main
```

# myweb Repository (Cont.)

...or push an existing repository from the command line

```
git remote add origin https://github.com/teenetlin777/myweb.git  
git branch -M main  
git push -u origin main
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

💡 ProTip! Use the URL for this page when adding GitHub as a remote.

# 新增資料夾

- 假設專案資料夾為「D:\Projects\Web\myweb」
- 新增資料夾「build」
- 在「build」資料夾新增「README」檔案

# 設定 index.html

- 假設專案資料為「D:\Projects\Web\myweb」
- 使用 Visual Studio Code 新增 index.html

```
1 <html>
2 <head>
3   <meta charset="utf-8">
4   <title>My Web</title>
5 </head>
6 <body>
7   <h1>Hello, My Web.</h1>
8 </body>
9 </html>
```

# 上傳檔案

- 假設專案資料為「D:\Projects\Web\myweb」
- 開啟 Git Bash
- \$ cd /d/Projects/Web/myweb
- \$ ls
- \$ git init

```
Initialized empty Git repository in D:/Projects/Web/myweb/.git/
```

# 上傳檔案 (Cont.)

- \$ git config --global user.email "lin777@gmail.com"
- \$ git config --global user.name "teenetlin777"
- \$ git add .
- \$ git commit -m "First Commit"
- \$ git branch -M main
- \$ git remote add origin <https://github.com/teenetlin777/myweb.git>

# 上傳檔案 (Cont.)

- \$ git push -u origin main

```
* [new branch]          main -> main  
branch 'main' set up to track 'origin/main'.
```

# 登入 Render

- <https://dashboard.render.com/>

## Sign In to Render

 GitHub

 GitLab

 Google

or

Email

your@email.com

Password

correct horse battery staple

Sign in

Need an account? [Sign up](#)

Forgot your password? [Reset it](#)

# 新增靜態網站

The screenshot shows the Render dashboard interface. At the top, there's a navigation bar with the Render logo, a redboxed 'Dashboard' button, 'Blueprints', 'Env Groups', 'Docs', 'Community', 'Help', a 'New +' button, a user profile for 'Ford Lin', and a dropdown menu. Below the navigation is a section titled 'Get started in minutes' containing eight service categories, each with a descriptive paragraph and a 'New [Service Type]' button:

- Static Sites**: Describes automatic global CDN serving. A red box highlights the 'New Static Site' button.
- Web Services**: Describes zero-downtime deploys, persistent storage, and PR previews. Includes a 'New Web Service' button.
- Private Services**: Describes services accessible within the Render network. Includes a 'New Private Service' button.
- Background Workers**: Describes long-running processes like consumers for queues and streaming. Includes a 'New Worker' button.
- Cron Jobs**: Describes scheduling commands or scripts. Includes a 'New Cron Job' button.
- PostgreSQL**: Describes fully-managed hosted PostgreSQL with connectivity and backups. Includes a 'New PostgreSQL' button.
- Redis**: Describes a cloud-based in-memory key-value datastore. Includes a 'New Redis' button.
- Blueprints**: Describes infrastructure as code. Includes a 'New Blueprint' button.

# 新增靜態網站 (Cont.)

## Create a new Static Site

Connect your Git repository or use an existing public repository URL.

**Connect a repository**

Connect your Render account to GitHub, GitLab, or Bitbucket to begin using your existing repos for new services.

 [Connect GitHub](#)

 [Connect GitLab](#)

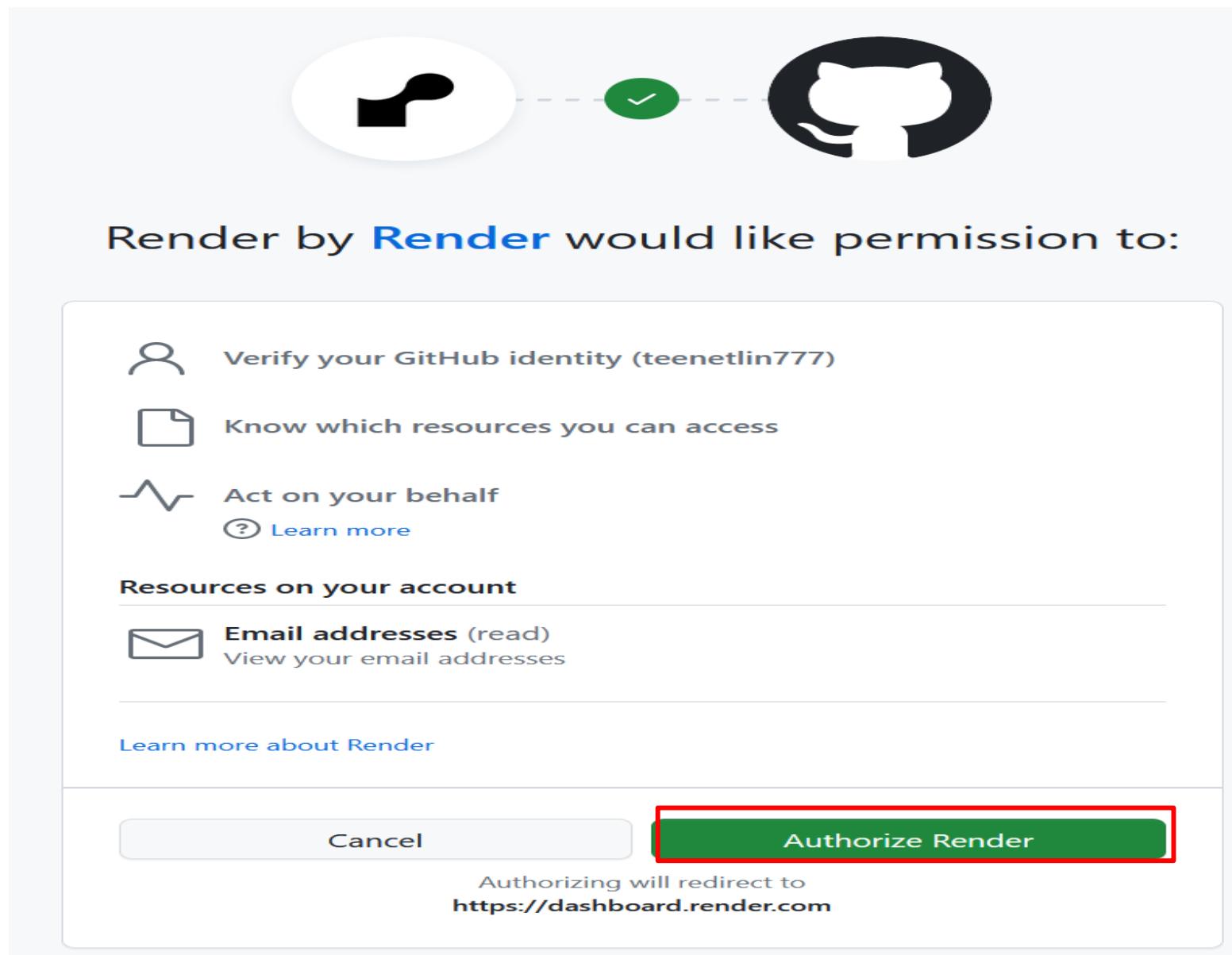


 [Connect account](#)



 [Connect account](#)

# 授權 Render



# 安裝 Render

**Install Render**

Install on your personal account teenetlin777 

for these repositories:

**All repositories**  
This applies to all future repositories owned by the resource owner.  
Also includes public repositories (read-only).

with these permissions:

- ✓ **Read** access to Dependabot alerts, administration, code, and metadata
- ✓ **Read and write** access to actions, checks, commit statuses, deployments, environments, issues, pull requests, repository hooks, and workflows

**User permissions**  
Render can also request users' permission to the following resources. These permissions will be requested and authorized on an individual-user basis.

- ✓ **Read** access to email addresses

Install [Cancel](#)

Next: you'll be directed to the GitHub App's site to complete setup.

# 連接 Repository

The screenshot shows the Render dashboard interface. At the top, there is a navigation bar with the Render logo, a 'Dashboard' button (which is highlighted with a red box), 'Blueprints', 'Env Groups', and links for 'Docs', 'Community', and 'Help'. Below the navigation bar, the main content area has a heading 'Create a new Static Site' and a sub-instruction 'Connect your Git repository or use an existing public repository URL.' A large callout box covers the central part of the page. Inside this box, the title 'Connect a repository' is displayed above a search bar with a placeholder 'Search...'. Below the search bar, a list item is shown: a GitHub icon followed by the text 'teenetlin777 / myweb' (with the entire text block highlighted by a red box), a lock icon, and the timestamp '• 4 minutes ago'. To the right of this list item is a blue 'Connect' button (also highlighted by a red box). The bottom left corner of the callout box contains the number '38'.

# 部署靜態網站

You are deploying a static site for [teenetlin777/myweb](#).

## Name

A unique name for your static site.

## Branch

The repository branch used for your static site.

## Root Directory Optional

Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.

## Build Command

This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

# 部署靜態網站 (Cont.)

Publish directory

The [relative](#) path of the directory containing built assets to publish. Examples: [./](#), [./build](#), [dist](#) and [frontend/build](#).

**Environment Variables** Optional

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more](#).

NAME_OF_VARIABLE	value	Generate	trash

[+ Add Environment Variable](#) [Add from .env](#)

Advanced ▾

Create Static Site

# 新增 Rewrites 規則

STATIC SITE

myweb

Manual Deploy ▾

teenetlin777 / myweb ➔ main

<https://myweb-dkt9.onrender.com> 🌐

Events

Environment

Redirects/Rewrites

Headers

Previews

Metrics

Settings

Add Redirect or Rewrite Rules to modify requests to your site without writing code. You can use URL parameters to capture path segments, and wildcards to redirect everything under a given path.

Add Rule

# 新增 Rewrites 規則 (Cont.)

STATIC SITE  
myweb Manual Deploy

teenetlin777 / myweb main  
<https://myweb-dkt9.onrender.com>

Events  
Environment  
**Redirects/Rewrites**  
Headers  
Previews  
Metrics  
Settings

### Redirect and Rewrite Rules

Add Redirect or Rewrite Rules to modify requests to your site without writing code. You can use URL parameters to capture path segments, and wildcards to redirect everything under a given path.

Source	Destination	Action
/*	/index.html	Rewrite

**Add Rule** **Save Changes**

# 新增 Rewrites 規則 (Cont.)

STATIC SITE

myweb

teenetlin777 / myweb main

<https://myweb-dkt9.onrender.com>

Manual Deploy

Events

Environment

Redirects/Rewrites

Headers

Previews

Metrics

Settings

Redirect and Rewrite Rules

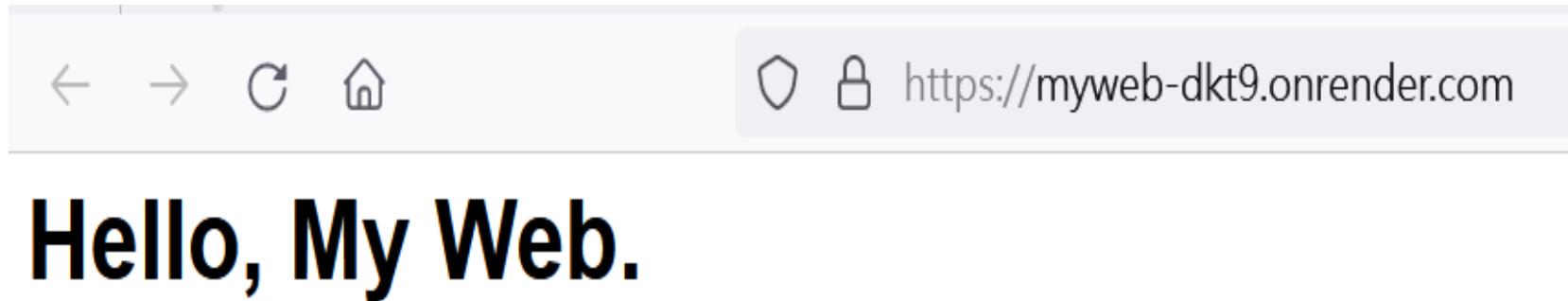
Add Redirect or Rewrite Rules to modify requests to your site without writing code. You can use URL parameters to capture path segments, and wildcards to redirect everything under a given path.

Source	Destination	Action
/*	/index.html	Rewrite

Add Rule Save Changes

# 檢視靜態網站

- <https://myweb-dkt9.onrender.com/>



# 動態網站部署

- Python Flask App 部署
- Python LINE Bot 部署
- Node Express App 部署

# Python Flask App 部署

- 創建 Repository
- 建立專案
- 設定 requirements.txt
- 設定 app.py
- 上傳檔案
- 登入 Render
- 新增 Web Service

# Python Flask App 部署 (Cont.)

- 連接 Repository
- 部署 Flask App
- 檢視 Flask App
- 新增路由 /hello
- 上傳檔案
- 檢視 hello 網頁
- 拜訪未註冊 View 函式的 URL

# 創建 Repository

The screenshot shows the GitHub Dashboard interface. At the top, there is a navigation bar with a menu icon, the GitHub logo, and the word "Dashboard". Below this, a section titled "Top Repositories" contains a search bar with the placeholder "Find a repository...". A green button labeled "New" is highlighted with a red border. Below the search bar, a repository card for "teenetlin777/myweb" is shown, featuring a profile picture and the repository name. Further down, a section titled "Recent activity" contains a message stating: "When you take actions across GitHub, we'll provide links to that activity here." The entire dashboard has a light gray background.

# 創建 Repository (Cont.)

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*

A text input field containing the repository name 'flaskhome', which is highlighted with a red border.

 flaskhome is available.

Great repository names are short and memorable. Need inspiration? How about [potential-umbrella](#) ?

Description (optional)

Flask App for Home Automation



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

# 創建 Repository (Cont.)

Initialize this repository with:

**Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

---

ⓘ You are creating a private repository in your personal account.

---

Create repository

# flaskhome Repository

 **flaskhome** Private

 **Set up GitHub Copilot**  
Use GitHub's AI pair programmer to autocomplete suggestions as you code.  
[Get started with GitHub Copilot](#)

 **Add collaborators to this repository**  
Search for people using their GitHub username or email address.  
[Invite collaborators](#)

**Quick setup — if you've done this kind of thing before**

 Set up in Desktop    or     HTTPS     SSH    <https://github.com/teenetlin777/flaskhome.git> 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# flaskhome" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/teenetlin777/flaskhome.git
git push -u origin main
```



# 建立專案

- 假設專案資料夾為「D:\Projects\Flask」
- 開啟「命令提示字元」
- C:\Users\hadoop>d:
- D:\>mkdir Projects\Flask\flaskhome
- D:\>cd Projects\Flask\flaskhome
- D:\Projects\Flask\flaskhome>dir

```
D:\Projects\Flask\flaskhome 的目錄
```

```
2024/03/10 下午 08:54    <DIR> .
2024/03/10 下午 08:54    <DIR> ..
0 個檔案          0 位元組
2 個目錄      302,437,527,552 位元組可用
```

# 設定 requirements.txt

- 使用 Visual Studio Code 在「flaskhome」資料夾下新增 requirements.txt 檔案
- 新增以下套件名稱
  - flask
  - gunicorn

```
1 flask  
2 gunicorn
```

# 設定 app.py

- 使用 Visual Studio Code 在「flaskhome」資料夾下新增 app.py 檔案

```
1  from flask import Flask  
2  
3  app = Flask(__name__)  
4  
5  @app.route('/')  
6  def index():  
7      return 'Hello World'  
8
```

# app.py 說明

- 處理連線請求與回應的程式，稱為路由 (Route)
- 呼叫 @app.route() 定義 URL，並利用 Python 裝飾器 (Decorator) @ 註冊 (Register) 一個回應處理函式 (稱為 View 函式)，即將 URL 與 View 函式綁定在一起
- 程式中的裝飾器 @app.route('/') 在為 URL '/' (根目錄) 註冊一個 View 回應處理函式，即 index() 函式

# app.py 說明 (Cont.)

- Flask 使用 request 類別處理請求訊息

# 上傳檔案

- 假設專案資料為「D:\Projects\Flask\flaskhome」
- 開啟 Git Bash
- \$ cd /d/Projects/Flask/flaskhome
- \$ ls
- \$ git init

```
Initialized empty Git repository in D:/Projects/Flask/flaskhome/.git/
```

# 上傳檔案 (Cont.)

- \$ git add .
- \$ git commit -m "First Commit"
- \$ git branch -M main
- \$ git remote add origin  
<https://github.com/teenetlin777/flaskhome.git>
- \$ git push -u origin main

```
Everything up-to-date
branch 'main' set up to track 'origin/main'.
```

# 登入 Render

- <https://dashboard.render.com/>

## Sign In to Render

 GitHub

 GitLab

 Google

or

Email

your@email.com

Password

correct horse battery staple

Sign in

Need an account? [Sign up](#)

Forgot your password? [Reset it](#)

# 新增 Web Service

The screenshot shows the Render dashboard interface. At the top, there is a navigation bar with links: Render (logo), Dashboard (highlighted with a red box), Blueprints, Env Groups, Docs, Community, Help, a New + button (highlighted with a red box), and a user profile for Ford Lin.

The main area is titled "Overview" and features a search bar labeled "Search services". Below the search bar is a table with columns: SERVICE NAME, STATUS, and TYPE. One row in the table shows a service named "myweb" with a status of "Deployed" and a type of "Static Site".

In the center, there is a detailed view for "Web Service". It includes a description: "Web services are kept up and running at all times, with native SSL and HTTP/2 support. Add a persistent disk or custom domain. Scale up and down with ease." There is also a "Learn more." link. To the right of this description is a sidebar with various service types: Static Site, Web Service (highlighted with a red box), Private Service, Background Worker, Cron Job, PostgreSQL, Redis, and Blueprint. At the bottom of the sidebar, there are filters for Active 1, Suspended 0, and All 1, along with a "LAST DEPLOYED" dropdown set to "LAST DEPLOYED ↓".

# 新增 Web Service (Cont.)

## Create a new Web Service

Connect a Git repository, or use an existing image.

How would you like to deploy your web service?

- Build and deploy from a Git repository

Connect a GitHub or GitLab repository.

- Deploy an existing image from a registry ADVANCED

Pull a public image from any registry or a private image from Docker Hub, GitHub, or GitLab.

Next

# 連接 Repository

## Create a new Web Service

Connect your Git repository or use an existing public repository URL.

Connect a repository

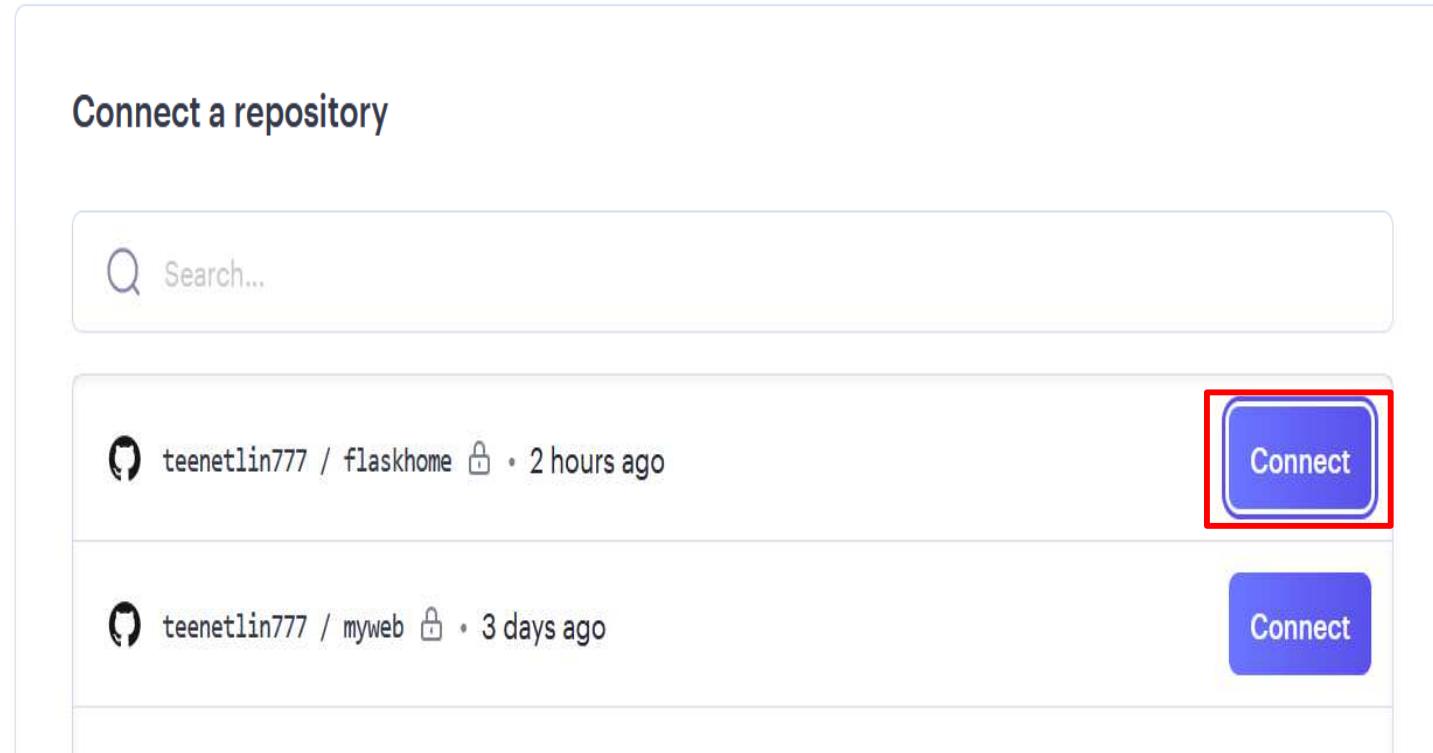
Search...

teenetlin777 / flaskhome · 2 hours ago

teenetlin777 / myweb · 3 days ago

Connect

Connect



@teenetlin777 · 2 repos

Configure account



Connect account

# 部署 Flask App

You are deploying a web service for [teenetlin777/flaskhome](#).

## Name

A unique name for your web service.

## Region

The [region](#) where your web service runs.

## Branch

The repository branch used for your web service.

## Root Directory Optional

Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.

# 部署 Flask App (Cont.)

## Runtime

The runtime for your web service.

Python 3

## Build Command

This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

\$ pip install -r requirements.txt

## Start Command

This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.

\$ gunicorn app:app

# 部署 Flask App (Cont.)

## Instance Type

For hobby projects	Free	512 MB (RAM) 0.1 CPU	⚠️ Upgrade to enable more features Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.
For professional use  For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support: <ul style="list-style-type: none"><li>• Zero Downtime</li><li>• SSH Access</li><li>• Scaling</li><li>• One-off jobs</li><li>• Support for persistent disks</li></ul>	Starter \$7 / month	512 MB (RAM) 0.5 CPU	Standard \$25 / month  2 GB (RAM) 1 CPU
	Pro \$85 / month	4 GB (RAM) 2 CPU	Pro Plus \$175 / month  8 GB (RAM) 4 CPU
	Pro Max \$225 / month	16 GB (RAM) 4 CPU	Pro Ultra \$450 / month  32 GB (RAM) 8 CPU

Need a [custom instance type](#)? We support up to 512 GB RAM and 64 CPUs.

# 部署 Flask App (Cont.)

## Environment Variables Optional

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)

NAME_OF_VARIABLE	value	<a href="#">Generate</a>	<a href="#"></a>
<a href="#">+ Add Environment Variable</a>		<a href="#"></a> <a href="#">Add from .env</a>	

Advanced ▾

[Create Web Service](#)

# 部署 Flask App (Cont.)

Events

Logs  

Disks

Environment

Shell

Previews

Jobs

Metrics

Scaling

Settings

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. [Upgrade now.](#)

March 10, 2024 at 11:47 PM Live

4e29546 Add requirements.txt

The screenshot shows a deployment logs interface. At the top, there's a message about a free instance spinning down. Below it, a timestamp and a commit ID are shown. The main area displays a log stream with the following entries:

- Mar 10 11:48:11 PM → Build successful 🎉
- Mar 10 11:48:14 PM → Deploying...
- Mar 10 11:48:40 PM → Detected service running on port 10000
- Mar 10 11:48:41 PM → Docs on specifying a port: <https://render.com/docs/web-services#port-detection>
- Mar 10 11:48:48 PM → Using Node version 20.11.1 (default)
- Mar 10 11:48:48 PM → Docs on specifying a Node version: <https://render.com/docs/node-version>
- Mar 10 11:48:52 PM → Running 'unicorn app:app'
- Mar 10 11:48:53 PM → [2024-03-10 15:48:53 +0000] [40] [INFO] Starting unicorn 21.2.0
- Mar 10 11:48:53 PM → [2024-03-10 15:48:53 +0000] [40] [INFO] Listening at: <http://0.0.0.0:10000> (40)
- Mar 10 11:48:53 PM → [2024-03-10 15:48:53 +0000] [40] [INFO] Using worker: sync
- Mar 10 11:48:53 PM → [2024-03-10 15:48:53 +0000] [41] [INFO] Booting worker with pid: 41
- Mar 10 11:48:55 PM → Your service is live 🎉

# 檢視 Flask App

- <https://flaskhome.onrender.com/>



# 新增路由 /hello

- 使用 Visual Studio Code 編輯 app.py

```
1  from flask import Flask  
2  
3  app = Flask(__name__)  
4  
5  @app.route('/')  
6  def index():  
7      |     return 'Hello World'  
8  
9  @app.route('/hello')  
10 def hello():  
11     |     return '<h1>Hello Flask</h1>'  
12  
13 if __name__ == '__main__':  
14     |     app.run()
```

# 上傳檔案

- \$ git add .
- \$ git commit -m "Second Commit"
- \$ git push -u origin main

```
Enumerating objects: 7, done.  
Counting objects: 100% (7/7), done.  
Delta compression using up to 16 threads  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (4/4), 443 bytes | 443.00 KiB/s, done.  
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To https://github.com/teenetlin777/flaskhome.git  
 4e29546..e9d64ea main -> main  
branch 'main' set up to track 'origin/main'.
```

# 檢視 hello 網頁

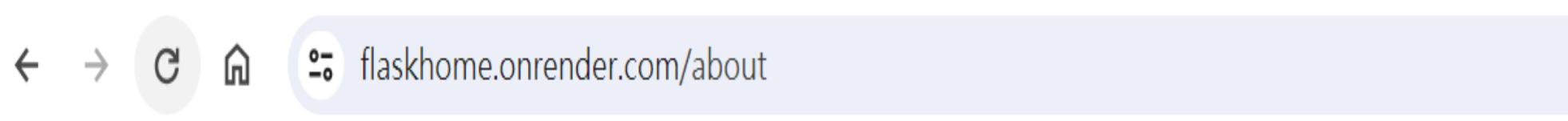
- <https://flaskhome.onrender.com/hello>



Hello Flask

# 拜訪未註冊 view 函式的 URL

- 拜訪未註冊 View 函式的任何 URL，Flask 會回應一個預設的 404 Not found 網頁
- <https://flaskhome.onrender.com/about>



## Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

# 動態路由

- 透過判斷字串的類型或長度決定使用哪個 View 函式或返回 404
- 顯示個人化的歡迎訊息

```
@app.route('/user/<name>')

def user(name):

    return '<h1>Hello, {}!</h1>'.format(name)
```

# 動態路由 (Cont.)

- 顯示多個變數的個人化的歡迎訊息

```
@app.route('/user/<name>/<surname>')

def user(name, surname):

    return '<h1>Hello, {}{}!</h1>'.format(name, surname)
```

# 設定 app.py

- 使用 Visual Studio Code 編輯 app.py

```
1  from flask import Flask  
2  
3  app = Flask(__name__)  
4  
5  @app.route('/')  
6  def index():  
7      |     return 'Hello World'  
8  
9  @app.route('/hello')  
10 def hello():  
11     |     return '<h1>Hello Flask</h1>'  
12
```

# 設定 app.py (Cont.)

```
13  @app.route('/user/<name>')
14  def user(name):
15      return f'<h1>Hello, {name}!</h1>'
16
17  @app.route('/user/<name>/<surname>')
18  def user_surname(name, surname):
19      return f'<h1>Hello, {name} {surname}!</h1>'
20
21  if __name__ == '__main__':
22      app.run()
```

# 上傳檔案

- \$ git add .
- \$ git commit -m "Third Commit"
- \$ git push -u origin main

```
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 443 bytes | 443.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/teenetlin777/flaskhome.git
  4e29546..e9d64ea main -> main
branch 'main' set up to track 'origin/main'.
```

# 檢視 /user/<name> 網頁

- <https://flaskhome.onrender.com/user/Jack>



Hello, Jack!

# 檢視 /user/<name>/<surname> 網頁

- <https://flaskhome.onrender.com/user/Jack/Chen>



**Hello, Jack Chen!**

# Python LINE Bot 部署

- 前置作業
- 創建 Repository
- 建立專案
- 設定 requirements.txt
- 設定 app.py
- 上傳檔案
- 登入 Render

# Python LINE Bot 部署 (Cont.)

- 新增 Web Service
- 連接 Repository
- 部署 Flask App
- 檢視 Flask App 網址
- 測試 LINE Bot
- 檢視測試成果

# 前置作業

- 取得 LINE Bot 的 Channel Access Token
- 取得 LINE Bot 的 Channel Secret

# 創建 Repository (Cont.)

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*

A text input field containing "mybot", which is highlighted with a red border.

 mybot is available.

Great repository names are short and memorable. Need inspiration? How about [crispy-spoon](#) ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

# 創建 Repository (Cont.)

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

---

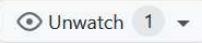
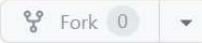
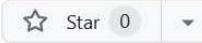
ⓘ You are creating a private repository in your personal account.

---

Create repository

# mybot Repository

 mybot Private

 **Set up GitHub Copilot**  
Use GitHub's AI pair programmer to autocomplete suggestions as you code.  
[Get started with GitHub Copilot](#)

 **Add collaborators to this repository**  
Search for people using their GitHub username or email address.  
[Invite collaborators](#)

**Quick setup — if you've done this kind of thing before**

 Set up in Desktop    or      <https://github.com/teenetlin777/mybot.git> 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# mybot" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/teenetlin777/mybot.git
git push -u origin main
```



# 建立專案

- 假設專案資料夾為「D:\Projects\Flask」
- 開啟「命令提示字元」
- C:\Users\hadoop>d:
- D:\>mkdir Projects\Flask\flaskbot
- D:\>cd Projects\Flask\flaskbot
- D:\Projects\Flask\flaskbot>dir

```
D:\Projects\Flask\flaskbot 的目錄
```

```
2024/03/26 上午 10:54 <DIR> .
2024/03/26 上午 10:54 <DIR> ..
0 個檔案 0 位元組
2 個目錄 309,332,725,760 位元組可用
```

# 設定 requirements.txt

- 使用 Visual Studio Code 在「flaskbot」資料夾下新增 requirements.txt 檔案
- 新增以下套件名稱
  - flask
  - gunicorn
  - line-bot-sdk

```
1  flask
2  gunicorn
3  line-bot-sdk
```

# 設定 bot\_info.py

- 使用 Visual Studio Code 在「flaskbot」資料夾下新增 bot\_info.py 檔案

```
1 # Channel Access Token  
2 line_bot_token = 'Your_Channel_Access_Token'  
3 # Channel Secret  
4 line_bot_secret = 'Your_Channel_Secret'
```

# 設定 app.py

- 使用 Visual Studio Code 在「flaskbot」資料夾下新增 app.py 檔案

```
1  from flask import Flask, request, abort
2  from linebot import LineBotApi, WebhookHandler
3  from linebot.exceptions import InvalidSignatureError
4  from linebot.models import MessageEvent, TextMessage, TextSendMessage
5
6  import bot_info
7
8  app = Flask(__name__)
9
10 # LINE Bot Basic Profile
11 line_bot_api = LineBotApi(bot_info.line_bot_token)
12 handler = WebhookHandler(bot_info.line_bot_secret)
13
```

# 設定 app.py (Cont.)

```
14 # Post Request from LINE Platform
15 @app.route('/callback', methods=['POST'])
16 def callback():
17     signature = request.headers['X-Line-Signature']
18     body = request.get_data(as_text=True)
19     app.logger.info('Request body:' + body)
20
21     try:
22         handler.handle(body, signature)
23     except InvalidSignatureError:
24         abort(400)
25     return 'OK'
26
```

# 設定 app.py (Cont.)

```
27 # Handle Messages  
28 @handler.add(MessageEvent, message=TextMessage)  
29 def echo(event):  
30     message = TextSendMessage(text=event.message.text)  
31     line_bot_api.reply_message(event.reply_token, message)  
32  
33 if __name__ == '__main__':  
34     app.run()
```

# app.py 說明

- signature = request.headers['X-Line-Signature']
  - 訊息驗證的加密簽章
- body = request.get\_data(as\_text=True)
  - 取得文本型態的請求正文
- handler.handle(body, signature)
  - 綁定訊息回傳的相關資訊
- return 'OK'
  - 驗證 Webhook 使用，不能省略

# 上傳檔案

- 假設專案資料為「D:\Projects\Flask\flaskbot」
- 開啟 Git Bash
- \$ cd /d/Projects/Flask/flaskbot
- \$ ls
- \$ git init

```
Initialized empty Git repository in D:/Projects/Flask/flaskbot/.git/
```

# 上傳檔案 (Cont.)

- \$ git add .
- \$ git commit -m "First Commit"

```
[master (root-commit) d0e7f80] First Commit
 3 files changed, 44 insertions(+)
 create mode 100644 app.py
 create mode 100644 bot_info.py
 create mode 100644 requirements.txt
```

# 上傳檔案 (Cont.)

- \$ git branch -M main
- \$ git remote add origin  
<https://github.com/teenetlin777/mybot.git>
- \$ git push -u origin main

```
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.03 KiB | 1.03 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/teenetlin777/mybot.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

# 登入 Render

- <https://dashboard.render.com/>

## Sign In to Render

 GitHub

 GitLab

 Google

or

Email

your@email.com

Password

correct horse battery staple

Sign in

Need an account? [Sign up](#)

Forgot your password? [Reset it](#)

# 新增 Web Service

The screenshot shows the Render.com dashboard interface. At the top, there is a navigation bar with links for 'Render', 'Dashboard' (which is highlighted with a red box), 'Blueprints', 'Env Groups', 'Docs', 'Community', 'Help', and a user profile for 'Ford Lin'. A 'New +' button is also highlighted with a red box.

The main area is titled 'Overview' and features a search bar labeled 'Search services'. Below the search bar is a table with columns: 'SERVICE NAME', 'STATUS', and 'TYP'. One row in the table shows a service named 'myweb' with a status of 'Deployed' and a type of 'Stat'.

A large callout box highlights the 'Web Service' option in the sidebar menu. The sidebar also lists other service types: 'Static Site', 'Web Service' (which is selected and highlighted with a red box), 'Private Service', 'Background Worker', 'Cron Job', 'PostgreSQL', 'Redis', and 'Blueprint'.

At the bottom right of the sidebar, there are buttons for 'Active 1', 'Suspended 0', and 'All 1'. Below these buttons, there is a link 'LAST DEPLOYED' with a downward arrow. Further down, there is a link '3 days ago' followed by three dots.

# 新增 Web Service (Cont.)

## Create a new Web Service

Connect a Git repository, or use an existing image.

How would you like to deploy your web service?

- Build and deploy from a Git repository

Connect a GitHub or GitLab repository.

- Deploy an existing image from a registry ADVANCED

Pull a public image from any registry or a private image from Docker Hub, GitHub, or GitLab.

Next

# 連接 Repository

## Create a new Web Service

Connect your Git repository or use an existing public repository URL.

### Connect a repository

 teenetlin777 / mybot  • 44 minutes ago	<button>Connect</button>
 teenetlin777 / smartlife  • 20 days ago	<button>Connect</button>
 teenetlin777 / flaskhome  • 21 days ago	<button>Connect</button>

-  GitHub  
 @teenetlin777  • 4 repos  
[Configure account](#)
-  GitLab  
[+ Connect account](#)
-  Bitbucket

# 部署 Flask App

You are deploying a web service for [teenetlin777/mybot](#).

You seem to be using Flask, so we've autofilled some fields accordingly. Make sure the values look right to you!

## Name

A unique name for your web service.

## Region

The [region](#) where your web service runs. Services must be in the same region to communicate privately and you currently have services running in [Singapore](#).

## Branch

The repository branch used for your web service.

## Root Directory Optional

Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.

# 部署 Flask App (Cont.)

## Runtime

The runtime for your web service.

Python 3

## Build Command

This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

\$ pip install -r requirements.txt

## Start Command

This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.

\$ gunicorn app:app

# 部署 Flask App (Cont.)

## Instance Type

For hobby projects

<b>Free</b>	512 MB (RAM)
\$0 / month	0.1 CPU

### ⚠ Upgrade to enable more features

Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.

For professional use

For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

<b>Starter</b>	512 MB (RAM)
\$7 / month	0.5 CPU

<b>Standard</b>	2 GB (RAM)
\$25 / month	1 CPU

<b>Pro</b>	4 GB (RAM)
\$85 / month	2 CPU

<b>Pro Plus</b>	8 GB (RAM)
\$175 / month	4 CPU

<b>Pro Max</b>	16 GB (RAM)
\$225 / month	4 CPU

<b>Pro Ultra</b>	32 GB (RAM)
\$450 / month	8 CPU

Need a [custom instance type](#)? We support up to 512 GB RAM and 64 CPUs.

# 部署 Flask App (Cont.)

## Environment Variables Optional

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)

NAME_OF_VARIABLE	value	 <a href="#">Generate</a>	
<a href="#">+ Add Environment Variable</a>		<a href="#">Add from .env</a>	

Advanced ▾

Create Web Service

# 部署 Flask App (Cont.)

Events

Logs  

Disks

Environment

Shell

Previews

Jobs

Metrics

Scaling

Settings

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. [Upgrade now.](#)

April 1, 2024 at 1:00 AM Live

d0e7f80 First Commit

All logs   Search   Live tail   GMT+8      

```
Apr 1 01:00:49 AM  ==> Deploying...
Apr 1 01:01:03 AM  ==> Using Node version 20.12.0 (default)
Apr 1 01:01:03 AM  ==> Docs on specifying a Node version: https://render.com/docs/node-version
Apr 1 01:01:06 AM  ==> Running 'gunicorn app:app'
Apr 1 01:01:08 AM  [2024-03-31 17:01:08 +0000] [40] [INFO] Starting gunicorn 21.2.0
Apr 1 01:01:08 AM  [2024-03-31 17:01:08 +0000] [40] [INFO] Listening at: http://0.0.0.0:10000 (40)
Apr 1 01:01:08 AM  [2024-03-31 17:01:08 +0000] [40] [INFO] Using worker: sync
Apr 1 01:01:08 AM  [2024-03-31 17:01:08 +0000] [41] [INFO] Booting worker with pid: 41
Apr 1 01:01:10 AM  ==> Your service is live 🎉
Apr 1 01:01:11 AM  127.0.0.1 - - [31/Mar/2024:17:01:11 +0000] "GET / HTTP/1.1" 404 207 "-" "Go-http-client/2.0"
Apr 1 01:01:13 AM  ==> Detected service running on port 10000
Apr 1 01:01:13 AM  ==> Docs on specifying a port: https://render.com/docs/web-services#port-detection
```

# 檢視 Flask App 網址

- <https://mybot-2phe.onrender.com>



# 測試 LINE Bot

- 掃描 QR Code，將 Bot 添加到 LINE 的朋友中
- 在 LINE 上向 Bot 發送文字訊息，並確認會學使用者說話

TOP > GoGoCloud > MyBot > **Messaging API**

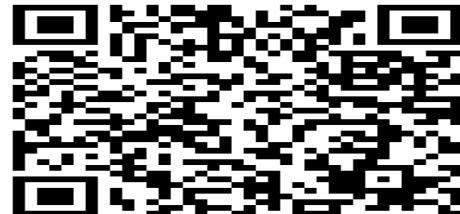
Basic settings    **Messaging API**    LIFF    Security    Statistics    Roles

## Messaging API settings

### Bot information

Bot basic ID    @156vtzbq    

### QR code



Scan this QR code with LINE to add your LINE Official Account as a friend. You can share the code with others.

# 檢視 LINE Bot 測試成果



# Node Express App 部署

- 創建 Repository
- 建立專案
- 產生 package.json
- 設定 package.json
- 設定 smartlife.js
- 上傳檔案
- 登入 Render

# Node Express App 部署 (Cont.)

- 新增 Web Service
- 連接 Repository
- 部署 Express App
- 檢視 Express App

# 創建 Repository

The screenshot shows the GitHub Dashboard. At the top, there is a navigation bar with a menu icon, the GitHub logo, and the word "Dashboard". Below this, a section titled "Top Repositories" displays three repositories owned by the user "teenetlin777": "myweb", "flaskhome", and "mybot". Each repository entry includes a small profile picture icon and the repository name. To the right of the repositories is a search bar with the placeholder text "Find a repository...". Further to the right is a green button with a plus sign icon and the word "New", which is highlighted with a red rectangular border. Below the top repositories section is another section titled "Recent activity", which contains a message stating: "When you take actions across GitHub, we'll provide links to that activity here.".

# 創建 Repository (Cont.)

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*

✓ smartlife is available.

Great repository names are short and memorable. Need inspiration? How about [potential-octo-fishstick](#) ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

# 創建 Repository (Cont.)

Initialize this repository with:

**Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

---

ⓘ You are creating a private repository in your personal account.

---

Create repository

# smartlife Repository

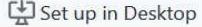
smartlife Private

Unwatch 1 Fork 0 Star 0

 Set up GitHub Copilot  
Use GitHub's AI pair programmer to autocomplete suggestions as you code.  
Get started with GitHub Copilot

 Add collaborators to this repository  
Search for people using their GitHub username or email address.  
Invite collaborators

**Quick setup — if you've done this kind of thing before**

 Set up in Desktop    or    [HTTPS](https://github.com/teenetlin777/smartlife.git)    [SSH](#)    <https://github.com/teenetlin777/smartlife.git> 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# smartlife" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/teenetlin777/smartlife.git
git push -u origin main
```

# 建立專案

- 假設專案資料夾為「D:\Projects\Node」
- 開啟「命令提示字元」
- C:\Users\hadoop>d:
- D:\>mkdir Projects\Node\smartlife
- D:\>cd Projects\Node\smartlife
- D:\Projects\Node\smartlife>dir

# 產生 package.json

- D:\Projects\Node\smartlife>npm init

This utility will walk you through creating a package.json file.  
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields  
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and  
save it as a dependency in the package.json file.

# 產生 package.json 檔案 (Cont.)

```
Press ^C at any time to quit.  
package name: (smartlife)  
version: (1.0.0)  
description: Smart Life  
entry point: (index.js) smartlife.js  
test command:  
git repository: https://github.com/teenetlin777/smartlife  
keywords:  
author: Ford Lin  
license: (ISC)  
About to write to D:\Projects\Node\smartlife\package.json:
```

# 產生 package.json 檔案 (Cont.)

```
{  
  "name": "smartlife",  
  "version": "1.0.0",  
  "description": "Smart Life",  
  "main": "smartlife.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "repository": {  
    "type": "git",  
    "url": "git+https://github.com/teenetlin777/smartlife.git"  
  },  
  "author": "Ford Lin",  
  "license": "ISC",  
  "bugs": {  
    "url": "https://github.com/teenetlin777/smartlife/issues"  
  },  
  "homepage": "https://github.com/teenetlin777/smartlife#readme"  
}
```

Is this OK? (yes)

# 設定 package.json

- 使用 Visual Studio Code 開啟 package.json 檔案
- 刪除 scripts、bugs、homepage 等 Key 的設定
- 新增 private、scripts、dependencies、engines 等 Key 的設定

# 設定 package.json (Cont.)

```
10  "author": "Ford Lin",
11  "license": "ISC",
12  "private": false,
13  // 偵錯
14  "scripts": {
15    "start": "node smartlife.js"
16  },
17  "dependencies": {
18    "express": "^4.18.2"
19  },
20  "engines": {
21    "node": ">=14"
22 }
```

# 設定 smartlife.js

- 使用 Visual Studio Code 新增 smartlife.js 檔案

```
1 const express = require("express");
2 const app = express();
3 const port = process.env.PORT || 3000;
4
5 app.get("/", (req, res) => {
6   res.type('text/plain');
7   res.send('Smart Life');
8 });
9
10 const server = app.listen(port,
11   () => console.log(`smartlife app listening on port ${port}!`));
12
13 server.keepAliveTimeout = 120 * 1000;
14 server.headersTimeout = 120 * 1000;
```

# 上傳檔案

- 假設專案資料為「D:\Projects\Node\smartlife」
- 開啟 Git Bash
- \$ cd /d/Projects/Node/smartlife
- \$ ls
- \$ git init

```
Initialized empty Git repository in D:/Projects/Node/smartlife/.git/
```

# 上傳檔案 (Cont.)

- \$ git add \*
- \$ git commit -m "Upload app.py"
- \$ git branch -M main
- \$ git remote add origin  
<https://github.com/teenetlin777/smartlife.git>
- \$ git push -u origin main

```
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 723 bytes | 723.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/teenetlin777/smartlife.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

# 登入 Render

- <https://dashboard.render.com/>

## Sign In to Render

 GitHub

 GitLab

 Google

or

Email

your@email.com

Password

correct horse battery staple

Sign in

Need an account? [Sign up](#)

Forgot your password? [Reset it](#)

# 新增 Web Service

The screenshot shows the Render.com dashboard interface. At the top, there is a navigation bar with the Render logo, a 'Dashboard' button (which is highlighted with a red box), 'Blueprints', 'Env Groups', 'Docs', 'Community', 'Help', a 'New +' button (also highlighted with a red box), a user profile for 'Ford Lin', and a dropdown menu.

The main area is titled 'Overview' and features a search bar labeled 'Search services'. Below the search bar is a table with columns: SERVICE NAME, STATUS, TYPE, and ACTION. Two services are listed:

SERVICE NAME	STATUS	TYPE	ACTION
flaskhome	Deployed	Web	...
myweb	Deployed	Static Site	Static

To the right of the table is a 'Web Service' card with the following content:

**Web Service**  
Web services are kept up and running at all times, with native SSL and HTTP/2 support. Add a persistent disk or custom domain. Scale up and down with ease.  
[Learn more.](#)

Below the card is a sidebar with service categories and their counts:

- Static Site: 0
- Web Service**: 1 (highlighted with a blue box)
- Private Service: 0
- Background Worker: 0
- Cron Job: 0
- PostgreSQL: 0
- Redis: 0
- Blueprint: 0

At the bottom of the sidebar, there are buttons for 'Active 2', 'Suspended 0', and 'All 2'.

# 新增 Web Service (Cont.)

## Create a new Web Service

Connect a Git repository, or use an existing image.

How would you like to deploy your web service?

- Build and deploy from a Git repository

Connect a GitHub or GitLab repository.

- Deploy an existing image from a registry ADVANCED

Pull a public image from any registry or a private image from Docker Hub, GitHub, or GitLab.

Next

# 連接 Repository

## Create a new Web Service

Connect your Git repository or use an existing public repository URL.

### Connect a repository

Search...

-  teenetlin777 / smartlife  • 14 days ago Connect
-  teenetlin777 / mybot  • 14 days ago Connect
-  teenetlin777 / flaskhome  • 15 days ago Connect
-  teenetlin777 / myweb  • 18 days ago Connect



@teenetlin777  • 4 repos

 Configure account



 Connect account



 Connect account

# 部署 Express App

You are deploying a web service for [teenetlin777/smartlife](#).

You seem to be using Node, so we've autofilled some fields accordingly. Make sure the values look right to you!

## Name

A unique name for your web service.

## Region

The [region](#) where your web service runs. Services must be in the same region to communicate privately and you currently have services running in [Singapore](#).

## Branch

The repository branch used for your web service.

## Root Directory Optional

Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.

# 部署 Express App (Cont.)

## Runtime

The runtime for your web service.

Node

## Build Command

This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

\$ yarn

## Start Command

This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.

\$ node smartlife.js

# 部署 Express App (Cont.)

## Instance Type

For hobby projects

<b>Free</b>	512 MB (RAM)
\$0 / month	0.1 CPU

### ⚠ Upgrade to enable more features

Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.

For professional use

For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

<b>Starter</b>	512 MB (RAM)
\$7 / month	0.5 CPU

<b>Standard</b>	2 GB (RAM)
\$25 / month	1 CPU

<b>Pro</b>	4 GB (RAM)
\$85 / month	2 CPU

<b>Pro Plus</b>	8 GB (RAM)
\$175 / month	4 CPU

<b>Pro Max</b>	16 GB (RAM)
\$225 / month	4 CPU

<b>Pro Ultra</b>	32 GB (RAM)
\$450 / month	8 CPU

Need a [custom instance type](#)? We support up to 512 GB RAM and 64 CPUs.

# 部署 Express App (Cont.)

## Environment Variables Optional

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)

NAME_OF_VARIABLE	value		
			

Advanced ▼

Create Web Service

# 部署 Express App (Cont.)

Events

Logs  

Disks

Environment

Shell

Previews

Jobs

Metrics

Scaling

Settings

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. [Upgrade now.](#)

March 26, 2024 at 3:30 AM    Building

Oce76d5 Upload smartlife.js Cancel deploy

All logs ▼ Search Q Live tail ▼ GMT+8 ↑ ⤵

```
Mar 26 03:30:40 AM  i  ==> Using Node version 21.7.1 via /opt/render/project/src/package.json
Mar 26 03:30:40 AM  i  ==> Docs on specifying a Node version: https://render.com/docs/node-version
Mar 26 03:30:44 AM  i  ==> Running build command 'yarn'...
Mar 26 03:30:45 AM  i  yarn install v1.22.22
Mar 26 03:30:45 AM  i  info No lockfile found.
Mar 26 03:30:45 AM  i  [1/5] Validating package.json...
Mar 26 03:30:45 AM  i  [2/5] Resolving packages...
Mar 26 03:30:45 AM  i  [3/5] Fetching packages...
Mar 26 03:30:45 AM  i  [4/5] Linking dependencies...
Mar 26 03:30:46 AM  i  [5/5] Building fresh packages...
Mar 26 03:30:46 AM  i  success Saved lockfile.
Mar 26 03:30:46 AM  i  Done in 1.55s.
```

# 檢視 Express App

- <https://smartlife-0b8m.onrender.com>



# 參考文獻

- [Deploy to Render] 什麼是 Render 「雲端運算」
  - <https://ithelp.ithome.com.tw/articles/10255630>
- 將個人網站部署到 Render
  - [https://hackmd.io/@kmx5moqzQWWvI\\_5ozlA2xQ/rJlNYZU6j](https://hackmd.io/@kmx5moqzQWWvI_5ozlA2xQ/rJlNYZU6j)
- Render 提供每月 100GB 免費主機空間，同時還可綁定網域與 SSL 和 CDN 快取
  - <https://www.minwt.com/website/server/22881.html>

# 參考文獻 (Cont.)

- Render | 來試試用來取代 Heroku 的服務吧 - Render 的網路服務部署介紹 !
  - <https://medium.com/starbugs/render-%E4%BE%86%E8%A9%A6%E8%A9%A6%E7%94%A8%E4%BE%86%E5%8F%96%E4%BB%A3-heroku-%E7%9A%84%E6%9C%8D%E5%8B%99%E5%90%A7-render-%E7%9A%84%E7%B6%B2%E8%B7%AF%E6%9C%8D%E5%8B%99%E9%83%A8%E7%BD%B2%E4%BB%8B%E7%B4%B9-b728e86d5716>

# 參考文獻 (Cont.)

- Heroku 終止免費方案，遷移到 Render Cloud 之紀錄
  - <https://medium.com/@bear817005/heroku-%E7%B5%82%E6%AD%A2%E5%85%8D%E8%B2%BB-%E9%81%B7%E7%A7%BB%E5%88%B0-render-cloud-%E4%B9%8B%E7%B4%80%E9%8C%84-a81395495669>
- 關於從 Heroku 跳到 Render 這件事情
  - <https://israynotarray.com/other/20221213/3036227586/>

# 參考文獻 (Cont.)

- [Deploy to Render] 用免費方案部署 LINE Bot
  - <https://ithelp.ithome.com.tw/articles/10283836>
- Heroku 替代方案 Render：部署 LINE Bot
  - <https://sayhitowill.medium.com/heroku-%E6%9B%BF%E4%BB%A3%E6%96%B9%E6%A1%88-render-%E9%83%A8%E7%BD%B2-line-bot-618fdca21427>
- 建立並串接 Webhook
  - <https://steam.oxxostudio.tw/category/python/example/line-webhook.html>

# 參考文獻 (Cont.)

- Line bot + python 建立我的第一個APP(1)-回話  
蟲篇
  - <https://medium.com/@fidhaley/line-bot-python-%E5%BB%BA%E7%AB%8B%E6%88%91%E7%9A%84%E7%AC%AC%E4%B8%80%E5%80%8Bapp-1-ef00fae9a1ae>
- 快速打造Line Chatbot
  - <https://hackmd.io/@littlehsun/linechatbot>

# 參考文獻 (Cont.)

- How to Deploy Static Website for FREE on Render with Custom Domain
  - <https://dev.to/stephengade/how-to-deploy-static-website-for-free-on-render-with-custom-domain-4efo>
- How to upload folders on GitHub
  - <https://stackoverflow.com/questions/40720141/how-to-upload-folders-on-github>

# 參考文獻 (Cont.)

- I uploaded a static website in Render and the website the I uploaded whenever I reloaded any pages aside from the homepage, it renders "not found"
  - <https://stackoverflow.com/questions/76819085/i-uploaded-a-static-website-in-render-and-the-website-the-i-uploaded-whenever-i>
- Deploy a Flask App
  - <https://docs.render.com/deploy-flask>

# 參考文獻 (Cont.)

- How to Deploy Node.js Express Application on Render ?
  - <https://www.geeksforgeeks.org/how-to-deploy-node-js-express-application-on-render/>
- Deploy a Node Express App
  - <https://docs.render.com/deploy-node-express-app>