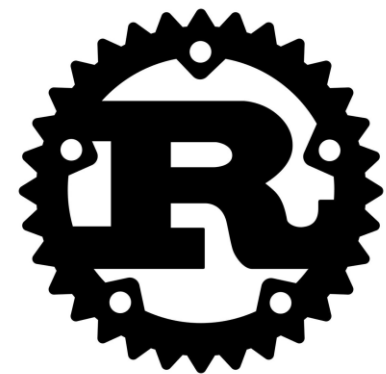








部署 Heroku



# 部署Heroku

傳統上，當想到要架設網站時，第一個想法就是要建置一台網路伺服器，或是到雲端平台上購買一台虛擬主機，接著在主機上安裝Apache、MySQL等等，之後再想辦法把自己的網站放在主機的空間中以啟用網站的功能。這種以主機為環境的雲端平台，叫做IaaS(Infrastructure as a Service)。它的好處是有一台主機可以自由地設定及安裝所有需要的伺服軟體，但缺點就是，要管理的事情非常多。

在部署網站時，其實還有一種跳過建立主機的方式，而是把自己的網站以APP的形式運行在特定提供PaaS(Platform as a Service)服務的主機商上，只要遵循它們的介面協定，使用它們支援的程式語言製作的網站，均可以推到它們的主機變成網站，Heroku一開始就是提供此種服務的主機商，另外一個著名的PaaS則是Google的APP Engine。

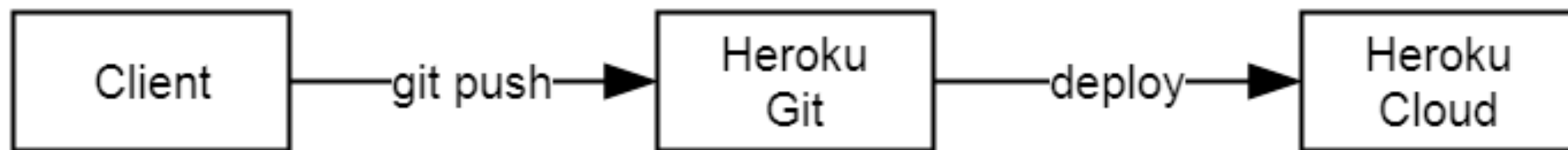
Dyno Types						
	 Free \$0	 Hobby \$7/dyno per month	 Standard 1x \$25/dyno per month	 Standard 2x \$50/dyno per month	 Performance M \$250/dyno per month	 Performance L \$500/dyno per month
	PROFESSIONAL					
What is it good for?	Ideal for experimenting with cloud applications in a limited sandbox.	Perfect for small scale personal projects and hobby apps.	Enhanced visibility, performance, and availability for powering your production applications.		Superior performance when it's most critical for your super scale, high traffic apps.	
RAM	512MB	512MB	512MB	1GB	2.5GB	14GB
Deploy from Git	•	•	•	•	•	•
Automated OS patching	•	•	•	•	•	•
Unified logs	•	•	•	•	•	•
Number of process types	2	10	Unlimited	Unlimited	Unlimited	Unlimited
Always on	Sleeps after 30 mins of inactivity, otherwise always on depending on your remaining monthly free dyno hours.	•	•	•	•	•
Custom domains	•	•	•	•	•	•

# 部署 Heroku

Heroku 是一套支援多程式個語言的雲平台即服務 (platform as a service, PaaS)，只要將程式碼上傳到該服務中，便可上頭自動編譯執行，建造自己的網路應用程式，且該服務提供免費試用方案 (有限制的)，又有支援 SSL (https)。

先到 官方網站註冊帳號，根據官方手冊安裝必須要的 Heroku Cli與版本控制軟體 Git，完成準備的前置作業。

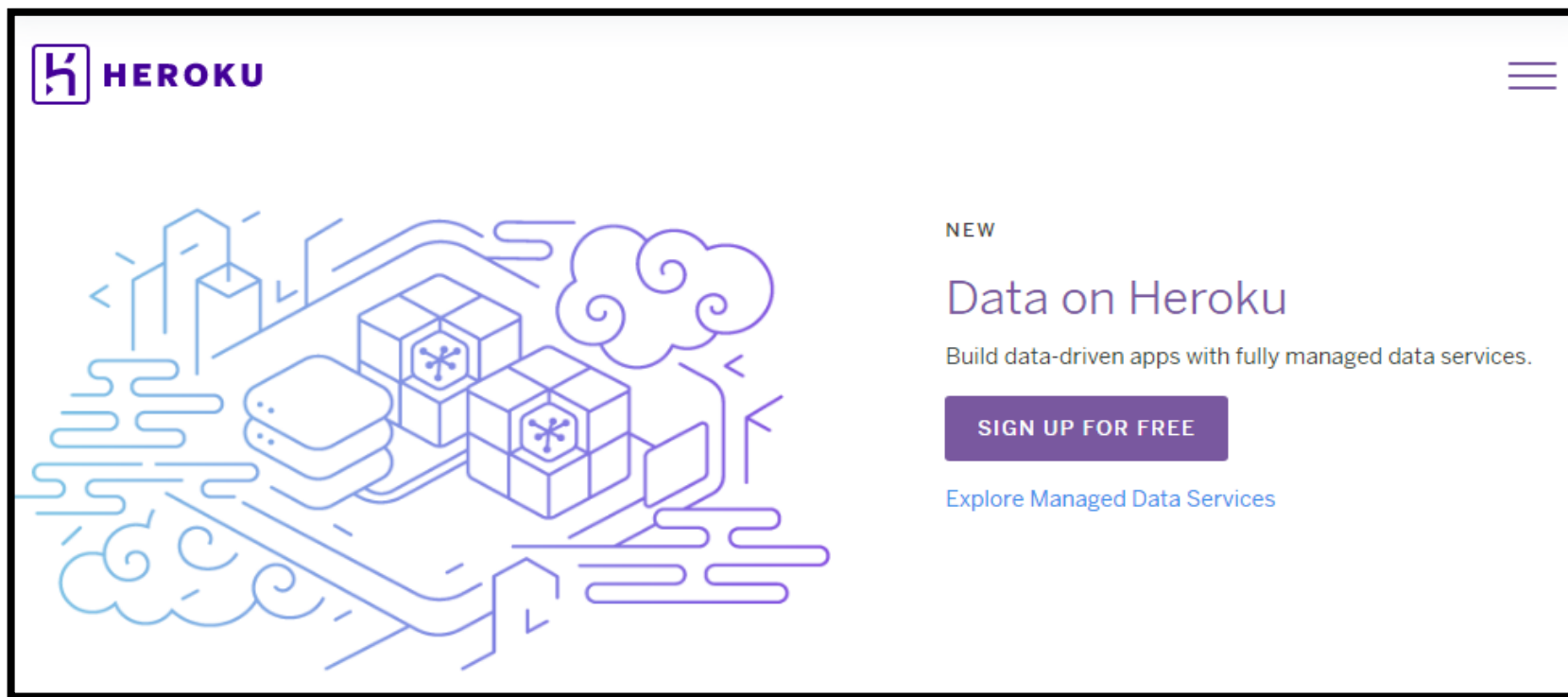
理解 Heroku 更新運作概念後，覺得其機制相當直覺與方便，每個運作在 Heroku 的 WebApp 都有一份自己的 Git repo. (repository)，在本機端初始 git repo. 後，利用 git commit 將程式碼加入本機 repo. 控管，然後進行 git push 將資料送到遠端的 Heroku git repo.，Heroku 再收到資料後，便開始展開編譯建置，然後部署在其伺服器上。



# 部署Heroku

## 註冊Heroku帳號

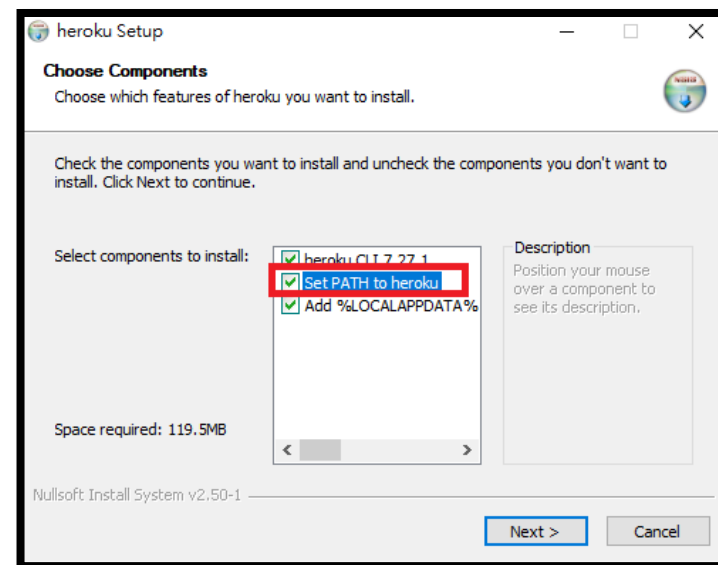
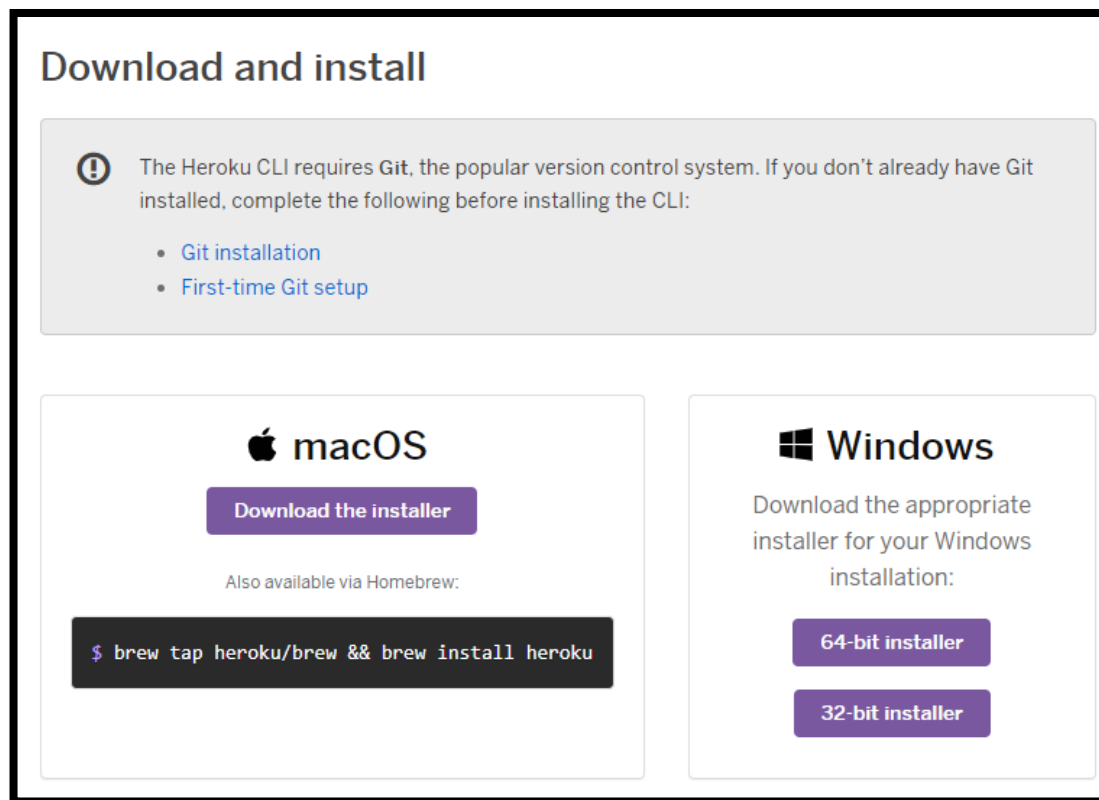
- 首先前往Heroku官網，點擊「SIGN UP FOR FREE」按鈕，依照指示來註冊使用者帳號，如下圖：



# 部署Heroku

## 安裝Heroku CLI

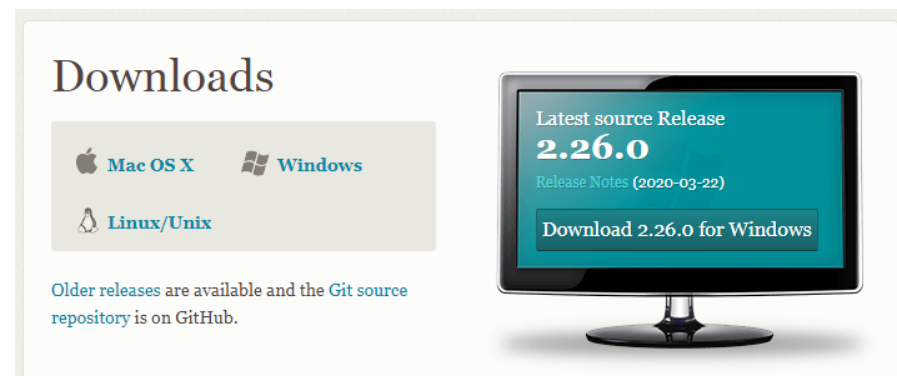
- 接著，要在自己的電腦安裝 **Git** 以及 **Heroku CLI** (<https://devcenter.heroku.com/articles/heroku-cli>)，安裝檔的下載頁面如下：



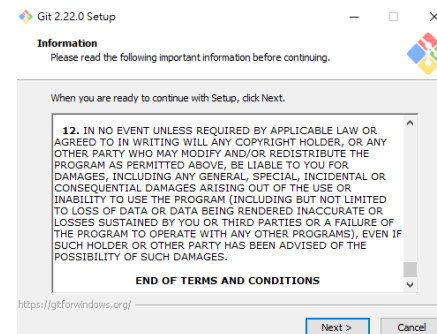
# 部署Heroku

## 安裝Git

- Git 是一個分散式的版本控制系統，除了可以協助開發人員進程式碼的版本控制外，也能夠將程式碼，推送至Heroku雲端平台的Git Repository，來進行部署。
- 前往Git官網的下載頁面，依照作業系統進行安裝，如下圖：



- 安裝完成後，開啟命令提示字元視窗，輸入以下指令：`git --version`
- 如果有顯示版本號，即代表Git安裝成功。



# 部署 Heroku

## 如何將 Rocket 應用程式部署到 Heroku

- Rocket是Rust程式語言的 Web 應用程式框架。Heroku是一家平台即服務提供商，它使託管 Web 應用程式變得簡單，並且在(低流量)業餘愛好級別上是免費的。部署就像推送到特殊的 git 遠端儲存庫一樣簡單。
- 因此首先需要安裝heroku命令列介面 (CLI) 並登入：**heroku login**
- 從已經撰寫的應用程式的根目錄(即其 git 儲存庫的根目錄)中，例如：使用Rust buildpack建立 Heroku 應用程序，並將其命名為todo-backend-rocket-rust：

```
heroku create todo-backend-rocket-rust --buildpack https://github.com/emk/heroku-buildpack-rust.git
```

# 部署 Heroku

## 如何將 Rocket 應用程式部署到 Heroku

- 接下來新增了兩個新檔案：Procfile 和 RustConfig。預設情況下，Rust buildpack 將使用最新的穩定 Rust，但 Rocket 需要nightly Rust，指定 VERSION=nightly在 RustConfig 中指定覆蓋 buildpack的預設值；Procfile 告訴 Heroku 如何執行應用程式。
- 在 Procfile 中增加了一項非常態的內容，以使 Rocket 在 Heroku 指定的連接埠上運行。Rocket 將尋找ROCKET\_PORT環境變數來覆寫其預設或Rocket.toml配置的連接埠。
- 由於Heroku連接埠既不可配置、不可預測也不一致，無法配置它Rocket.toml或依賴 Rocket 的預設值，**需設定ROCKET\_PORT=\$PORT在 Procfile 中進行分配：**

```
web: ROCKET_PORT=$PORT target/release/todo-backend-rocket-rust
```

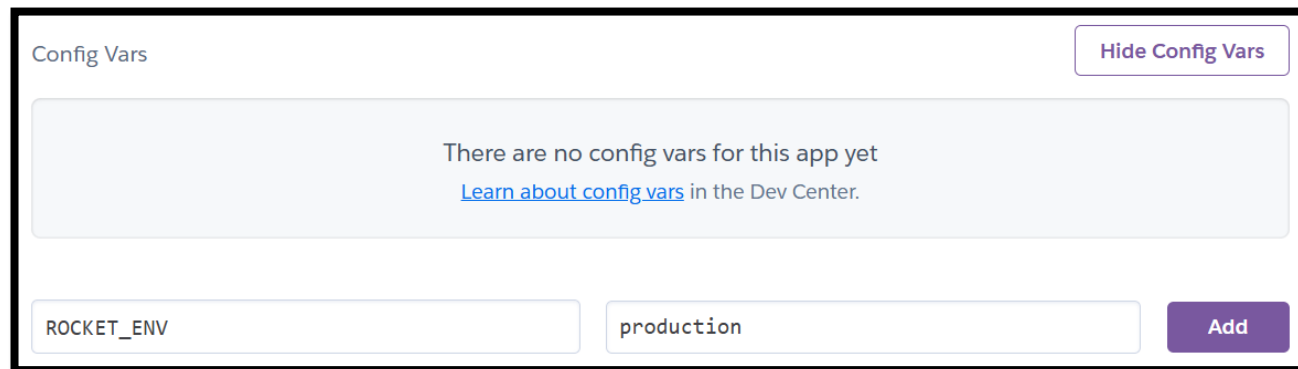


# 部署Heroku

## 如何將 Rocket 應用程式部署到 Heroku

```
web: ROCKET_PORT=$PORT target/release/todo-backend-rocket-rust
```

- 該應用程式在 `https://todo-backend-rocket-rust.herokuapp.com/` 部署後就可用，並且應用程式需要使用其公開基本 URL 進行配置，因此將必要的配置提交到Cargo.toml。
- 然後ROCKET\_ENV為應用程式設定了heroku配置變數，以告訴Rocket載入正確的配置Rocket.toml。Heroku 配置值會作為環境變數公開給應用程式：

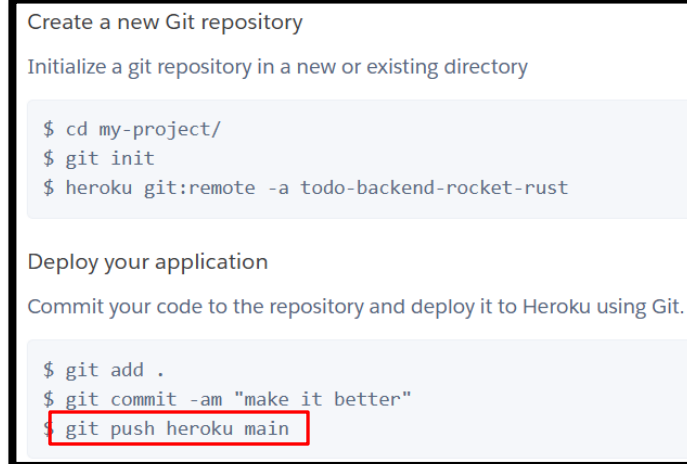


Config Vars Hide Config Vars

There are no config vars for this app yet  
[Learn about config vars](#) in the Dev Center.

ROCKET\_ENV production Add

- 然後就準備部署了：`git push heroku master`



```
Create a new Git repository
Initialize a git repository in a new or existing directory

$ cd my-project/
$ git init
$ heroku git:remote -a todo-backend-rocket-rust

Deploy your application
Commit your code to the repository and deploy it to Heroku using Git.

$ git add .
$ git commit -am "make it better"
$ git push heroku main
```

# Ref

## 學習資源

### 1. [Rust 程式設計語言](#)

- 由 Steve Klabnik 與 Carol Nichols，以及 Rust 社群的貢獻撰寫而成

### 2. [透過例子學 Rust 中文](#)

- 翻譯自 Rust By Example

### 3. [RustPrimer](#)

- 給初學者的Rust中文教學

### 4. [Rust入門祕籍](#)

- 相比官方書籍(The Rust Programming Language)要更詳細、更具系統性