

Day 27

使用GCP部署機器學習API

[全民瘋AI系列]



第12屆 iT邦幫忙 鐵人賽

Day 27 學習目標

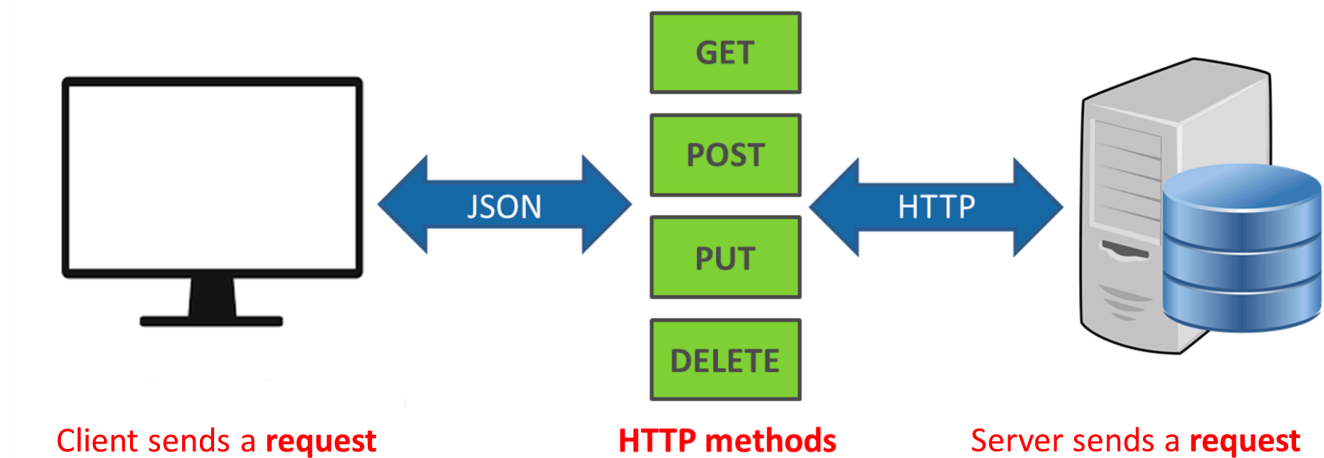
01

動手部署自己的機器學習 API


使用 GCP 免費雲端平台部署應用程式

Python Flask API




- Flask 是一個使用 Python 撰寫的輕量級 Web 應用框架。





//// Fork 專案到自己的 GitHub 帳號中




[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)


 [1010code](#) / [Flask-API-example-with-ML-model-GCP](#)

 Unwatch ▼

1


 Star


0


 Fork

0


[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

 main ▼






 1 branch

 0 tags

[Go to file](#) [Add file ▼](#) [Code ▼](#)


 **andy6804tw** commit

e7e4b76 3 minutes ago ⌚ 9 commits

 app	commit	3 minutes ago
 .gitignore	commit	11 hours ago
 README.md	commit	3 minutes ago
 requirements.txt	commit	3 minutes ago
 run.py	commit	3 minutes ago

About

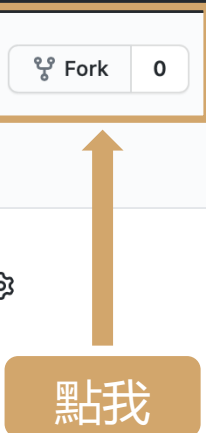
No description, website, or topics provided.

 [Readme](#)

Releases

No releases published

[Create a new release](#)



//// GCP 設定

建立一個虛擬機器

每個Google帳號都有免費一年300美金額度的試用，啟用後首先一開始點選 Compute Enging 並新增建立 VM 執行個體。



Google Cloud Platform My First Project

Compute Engine VM 執行個體 [瞭解詳情](#)

VM 執行個體

執行個體群組

執行個體範本

單一用戶群節點

機器映像檔

磁碟

快照

映像檔

TPU

Compute Engine VM 執行個體

Compute Engine 可讓您使用在 Google 基礎架構上執行的虛擬機器，可供建立的執行個體類型包括微型 VM 以及搭載 Debian、Windows 或其他標準映像檔的大型執行個體。歡迎立即建立您的第一個 VM 執行個體、利用遷移服務匯入 VM 執行個體，或嘗試透過快速入門課程建立範例應用程式。

[建立](#) 或 [匯入](#) 或 [進入快速入門導覽課程](#)



//// 新建一個虛擬機需要注意以下幾個事情：

- 1.主機區域 (通常主機離你越遠相對的費用就會比較便宜，相對的速度會比較慢)
- 2.機器規格設定 (各位可以依據需求配置你的虛擬機)
- 3.系統 (今天的範例使用 Ubuntu18.04 LTS)
- 4.防火牆 (開啟 HTTP 流量，也就是 80 PORT 被允許存取)



← 建立執行個體

如要建立 VM 執行個體，請先選取下列任一選項：

新增 VM 執行個體

從頭建立 VM 執行個體

利用範本建立新的 VM 執行個體

利用現有的範本建立 VM 執行個體

透過機器映像檔建立新 VM 執行個體

透過現有機器映像檔建立一個 VM 執行個體

Marketplace

將可立即使用的解決方案部署至 VM 執行個體

名稱

名稱經設定後即無法變更

demo

標籤

+ 新增標籤

區域

us-central1 (愛荷華州)

區域

us-central1-a

機器設定

機器系列

一般用途 | 記憶體最佳化 | 運算最佳化

適用於一般工作負載的機器類型，已針對費用和彈性進行最佳化

系列

N1

採用 Intel Skylake CPU 平台或其前身

機器類型

f1-micro (1 個 vCPU，614 MB 記憶體)

vCPU

1 個共用核心

記憶體

614 MB

CPU 平台和 GPU

容器

☐ 將容器映像檔部署至這個 VM 執行個體。瞭解詳情

開機磁碟

新的 10 GB 標準永久磁碟

映像檔

Ubuntu 18.04 LTS

變更

身分及 API 存取權

服務帳戶

Compute Engine default service account

存取權範圍

☒ 允許預設存取權

☐ 允許所有 Cloud API 的完整存取權

☐ 針對各個 API 設定存取權

防火牆

您可以新增標記和防火牆規則，允許接受來自網際網路的特定流量

☒ 允許 HTTP 流量

☐ 允許 HTTPS 流量

管理、安全性、磁碟、網路、單獨租用

您必須支付這個執行個體的費用。Compute Engine 定價

建立

取消

對等 REST 或指令列

每月 \$4.28 (預估值)

每小時約為 \$0.006

用多少付多少：無須預繳費用，而且是以秒計費

本月使用「f1-micro」執行個體的前 744 小時完全免費。瞭解詳情

詳細資料

SSH 進入虛擬主機

Google Cloud Platform My First Project 搜尋資源和產品

Compute Engine VM 執行個體 建立執行個體 匯入 VM 重新整理 開始 停止 重設 刪除

VM 執行個體

執行個體群組 執行個體範本 單一用戶群節點

篩選 VM 執行個體 資料欄

<input type="checkbox"/>	名稱 ^	區域	建議	使用者	內部 IP	外部 IP	連接
<input checked="" type="checkbox"/>	demo	us-central1-a			10.128.0.13 (nic0)	35.184.160.222 ↗	SSH

```
m10759011@demo: ~
ssh.cloud.google.com/projects/onyx-harmony-269902/zones/us-central1-a/instances/demo
System load: 0.0      Processes:      109
Usage of /:  30.0% of 9.52GB  Users logged in: 1
Memory usage: 8%      IP address for ens4: 10.128.0.3
Swap usage:  0%

* Kubernetes 1.19 is out! Get it in one command with:

    sudo snap install microk8s --channel=1.19 --classic

https://microk8s.io/ has docs and details.

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

5 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Mon Oct 12 03:11:47 2020 from 35.235.241.16
g10coding_channel@demo:~$
```



//// 安裝 Python

要在 Linux 環境中安裝 Python 3，請安裝相對應的套件。[python3](#)、[python3-dev](#)、[python3-venv](#)。

```
sudo apt update  
sudo apt install python3 python3-dev python3-venv build-essential
```

輸入以下指令安裝 Python 以及 PIP 管理工具。

```
wget https://bootstrap.pypa.io/get-pip.py  
sudo python3 get-pip.py
```



//// 執行 API

你可以直接 Fork 此專案到你自己的 GitHub 帳號中，或是直接 clone 專案到你的 GCP 中。

```
git clone https://github.com/1010code/Flask-API-example-with-ML-model-GCP.git  
cd Flask-API-example-with-ML-model-GCP
```



安裝必要套件

使用 pip3 指令安裝必要的套件。

```
pip3 install -r requirements.txt
```



執行

使用 python3 指令且在 sudo 環境下執行程式，即可監聽 80 PORT。

```
sudo python3 run.py
```

```
gl0coding_channel@demo:~/Flask-API-example-with-ML-model-GCP$ sudo python3 run.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
```



Thanks

PRESENTED BY 10程式中



第12屆 iT邦幫忙 鐵人賽