

112-1 雲端架構實務

Cloud Architecture Practice.

期中報告

Infrastructure as a Service

授課老師：潘得龍 老師

學 生：B11123206 陳冠欣

中 華 民 國 1 1 2 年 1 1 月 7 日

目錄

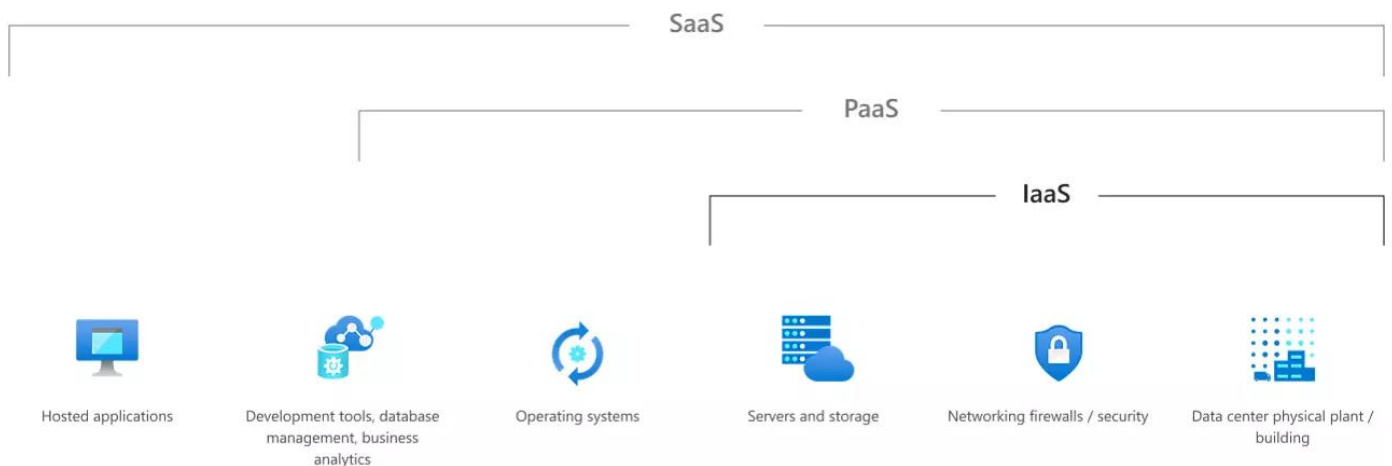
目錄	2
壹、 什麼是 IaaS ?	3
貳、 IaaS 服務提供商比較	4
參、 IaaS 實作	6
肆、 參考資料	16

壹、 什麼是 IaaS ？

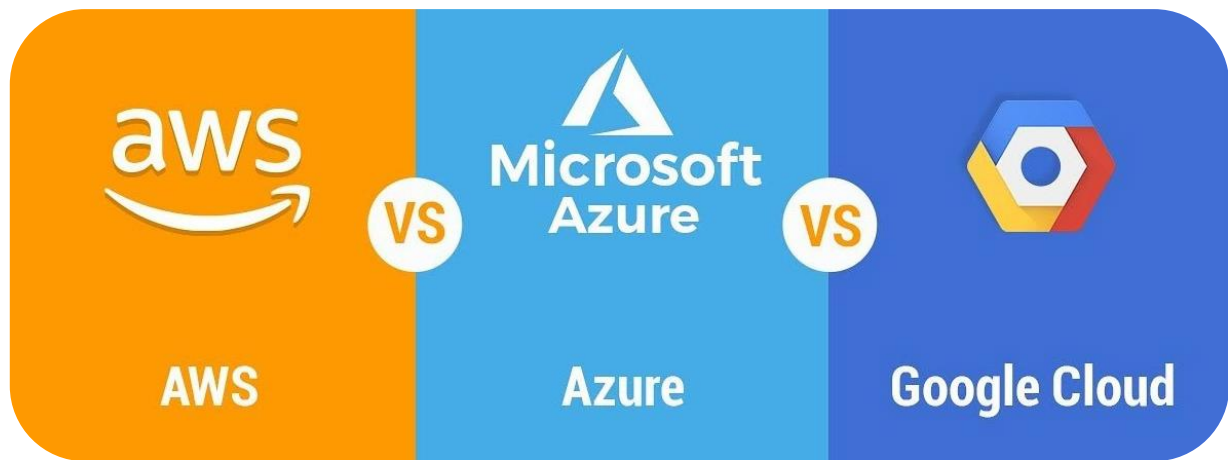
基礎結構即服務 IaaS 是一種雲端運算服務類型，以隨用隨付的方式，依需求提供基本的計算、儲存體及網路資源。IaaS 解決方案可讓您彈性地隨著需求的變化，對應增減所需的 IT 資源，此外也可以協助您快速佈建新的應用程式，並提升基礎結構的可靠性。

優點	描述
降低成本	- 耗費了大筆資金建造和維護自己的資料中心。 - 沒有硬體或軟體更新/更換的費用。
提高效率	- 快速建置和配置所需的基礎設施，無需等待硬體/軟體安裝或配置。
增加彈性	- 根據隨時業務需求或市場變化擴展或縮減所需資源，避免資源浪費或短缺。
支持創新	- 利用IaaS提供的最新技術和服務（如人工智慧、機器學習、高效能運算）進行創新和產品開發。
提升安全性	- 利用IaaS提供的安全措施和功能（如加密、防火牆、備份、復原）來保護資料和應用程式。

除了 IaaS 以外，還有軟體即服務 SaaS 和平台即服務 PaaS 這兩種雲端服務類型。這三種模式的主要差異，在於由第三方雲端服務供應商所代管的雲端運算服務項目而有所不同。



貳、 IaaS 服務提供商比較



1. 亞馬遜網路服務 (AWS)

特點	描述
廣泛的服務	提供多種IaaS服務，包括運算、儲存、網路、資料庫、安全性等。
全球性覆蓋	全球多地資料中心，提供多地域部署，提高效能和可用性。
彈性和可擴展性	快速擴展流量或縮減資源，自動縮放觀察波動和需求變化。
安全性和合規性	提供身分和存取管理、加密、防火牆等強大的安全性工具。
支持創新	提供最新技術和服務，如人工智慧、機器學習，支援新產品開發。
付費方式	採用付費模型，無前期成本，使用者只需支付實際使用的資源。

2. 微軟 Azure

特點	描述
整合性	與微軟產品緊密整合，適合與微軟相關的企業。
多種作業系統支援	支援多種作業系統，包括Windows和各種Linux發行版本。
靈活	提供各種虛擬機器規格和定價選項，適應不同的需求，包括運算、儲存等。
全球存在	全球資料中心支持，低延遲地部署應用程式，適合全球用戶。
混合雲支援	支援雲端部署，整合本地基礎設施與混合雲端資源。

3. Google Cloud Platform (GCP)

特點	描述
數據分析與人工智慧	提供大數據處理、機器學習、深度學習工具，適合數據密集應用。
容器化	支援Kubernetes，容器編排和管理，適合容器化應用部署。
網路效能	擁有全球高效能網路，低延遲和高吞吐量，滿足高可用性需求。
安全性和合規性	提供強大的安全措施和合規性支持，包括多層次網路安全、身分認證、加密等。
可伸縮性	快速擴展或縮減資源，適應不同的工作負載需求。

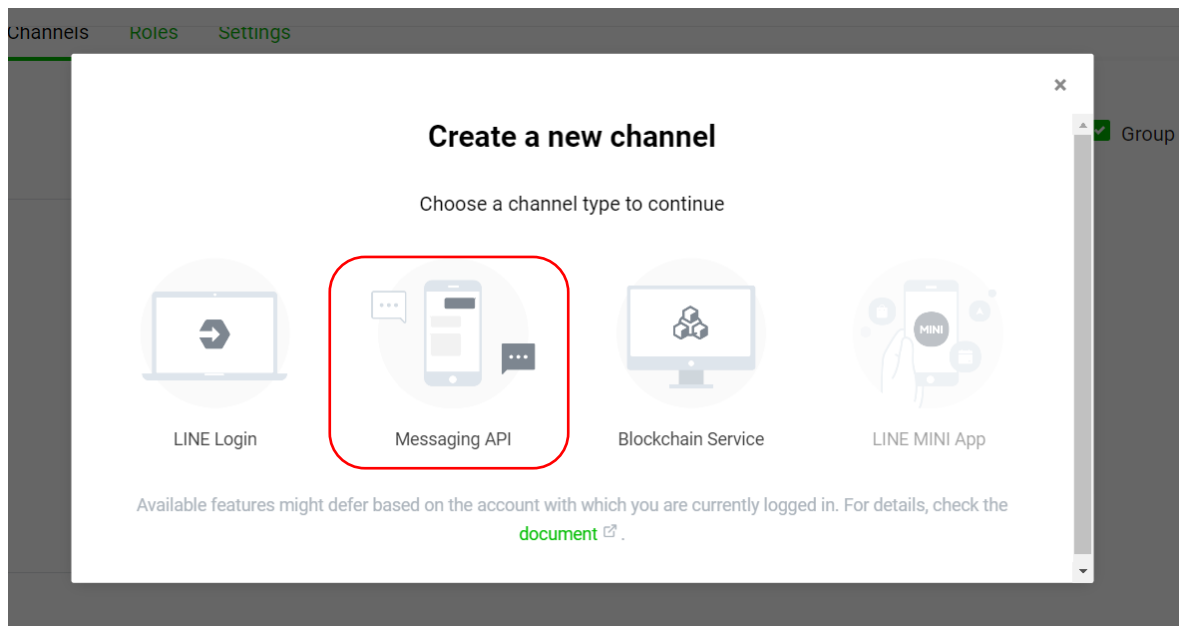
AWS、Azure 和 GCP 都是同類的雲端運算服務典範，它們的 IaaS 服務模式都有一些共同的特色和優勢，在一些方面又各有各的重點和優勢，用戶可以根據其具體需求和背景選擇最適合的服務。

參、IaaS 實作

建立一個通訊機器人輔助日常生活應用，是一件很便利的事情，這裡將展現如何在 GCP 上建立一個 Line Bot 伺服器，並開發一個簡易而實用的聊天機器人。

一、Line Developer 申請

選擇 Messaging API



在建立完畢後，會看到 Channel 的基本頁面，這裡有一些基本資訊可以進行修改。製作機器人最關鍵的兩個資訊，Channel secret 和 Channel access token，都可以在此頁面取得。您可以在 Basic setting 的最下方找到 Channel secret，而 Channel access token 則位於 Messaging API 的最下方，點擊 Issue 即可生成。

LINE 開發商

關於

訊息

產品

文件

常見問題

詞彙表

更多的

🌐

🔔

👤

⚙️

控制台首頁

供應商

搜尋...

行政

GPT

GX

頂部

建立新頻道

通道類型

Messaging API

✓不要將此留空

提供者

GX

✓不要將此留空

公司或所有者所在國家或地區 ①

台灣

公司應選擇其公司所在的國家或地區，個人應選擇其商號或居住地在的國家或地區。

✓不要將此留空

頻道圖示 選擇的

🖼️

登記

✓檔案類型必須是以下其中一：PNG、JPG、JPEG、GIF、BMP

✓文件不得大於 3 MB

頻道名稱

GPT

注意：頻道名稱7天內不能更改。

✓不要將此留空

✓不要使用特殊字元（4 位元組 Unicode）

✓輸入不得大於20個字元

頻道說明

聊天機器人

✓不要將此留空

✓不要使用特殊字元（4 位元組 Unicode）

✓輸入不得大於500個字元

類別

其他媒體

✓不要將此留空

子類別

媒體(其他)

✓不要將此留空

電子郵件地址 ①

p0972931647@gmail.com

✓不要將此留空

✓輸入一個有效的電子郵件地址

✓輸入不得大於100個字元

隱私權政策網址 選擇的

輸入隱私權政策 URL

✓輸入有效的 HTTPS URL

✓輸入不得大於500個字元

使用條款網址 選擇的

輸入使用條款 URL

✓輸入有效的 HTTPS URL

✓輸入不得大於500個字元

☐ 我已閱讀並同意LINE 官方帳號使用條款 ②

☐ 我已閱讀並同意LINE 官方帳戶 API 使用條款 ②

✖️閱讀相關文件後領取認證

創建

工具

支援

點選右上角的 Settings > Response settings > Detailed settings，把 Auto-response 關掉，Webhooks 打開，這樣我們就可以用 API 來操控機器人回覆，不使用 line 自己的自動回覆後台。

Detailed settings

Auto-response ☐ Enabled

☒ Disabled

Auto-response message settings

Webhooks ☒ Enabled

☐ Disabled

Messaging API settings

二、 建立開發環境

本次使用 GCP 的 VM 作為開發機器，GCP 提供 US300 的試用額度及 90 天的試用時間，是做為測試的好選擇。以下是我建立 VM 的資訊。

The screenshot shows the Google Cloud Platform console for a project named 'My First Project'. The left sidebar lists various services, with 'Compute Engine' selected. Under 'Compute Engine', 'VM instances' is chosen. The main panel displays the details for a VM instance named 'gpt'. The instance is in the 'Running' state, created on October 31, 2023, at 11:09:01 UTC+08:00. It is located in the 'asia-east1-b' region. The instance has 1 vCPU and 0 GB of memory. The machine type is 'n1-standard-4'. The CPU platform is 'Intel Broadwell'. The OS is 'Ubuntu 22.04-LTS'. The instance is connected to the 'default' network and has a public IP address. The console also shows the 'SSH' button and a 'Logs' section.

Category	Property	Value
Basic Information	Name	gpt
	Instance ID	5139333319513461027
	Status	Running
	Machine Type	n1-standard-4
	Region	asia-east1-b
	Zone	asia-east1-b
	Creation Time	10月31, 2023, 11:09:01 上午 UTC+08:00
	OS Image	Ubuntu 22.04-LTS
	Boot Disk	100 GB
	Network	default
Machine Settings	Machine Type	n1-standard-4
	CPU Platform	Intel Broadwell
	Number of CPUs	1
	Memory	0 GB
	vCPU	1
	Number of Cores	1
	Operating System	Ubuntu 22.04-LTS
	GPU	None
	Network	default
	Public IP	35.244.172.100

公開 DNS PTR 記錄

無

網路輸出緩衝層級

—

NIC 類型

—

→ 在 NETWORK TOPOLOGY 中查看

防火牆

HTTP 流量

已啟用

HTTPS 流量

已啟用

允許負載平衡器健康狀態檢查

已停用

網路標記

gpt

http-server

https-server

網路介面

名稱	↑	網路	子網路	主要內部 IP 位址	別名 IP 範圍	IP 地址類型	外部 IP 位址	網路類別	IP 轉寄
nic0		default	default	10.140.0.2		IPv4	gpt (35.194.210.79)	端點	已停用

儲存空間

開機磁碟

名稱	↑	映像檔	介面類型	大小 (GB)	裝置名稱	類型	架構	加密	模式	刪除執行個體時
instance-1		ubuntu-2004-focal-v20231025	SCSI	50	instance-1	SSD 永久磁碟	x86/64	Google 代管	可讀取/寫入的開機磁碟	刪除磁碟

本機磁碟

無

新增磁碟

無

安全性與存取權

受防護的 VM

安全啟動

已停用

vTPM

已啟用

完整性監控

已啟用

安全殼層金鑰

安全殼層金鑰

使用者名稱	鍵
p0972931647	ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHh0NTYAAAIbmldHh0NTYAAABBBjYU6V1Gg5TuZFZQkT1SUr9tFHJjo3...
p0972931647	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACN++X/iVmG4+BwLQg7MkBL2K2K5inFj1WZ3mLV1x/F0WSYXIRiB/xQxbj22...

封鎖專案層級的安全殼層金鑰

已停用

API 與身分識別管理

服務帳戶

58513115332-compute@developer.gserviceaccount.com

Cloud API 存取權範圍

允許所有 Cloud API 的完整存取權

管理

資料加密

金鑰 ID

—

金鑰名稱

—

可用性政策

VM 佈建模型

標準

時間長度上限

無

先占功能

已停用 (推薦)

在 VM 終止時

—

在主機縮減時間

遷移 VM 執行個體 (推薦)

自動重新啟動

已啟用 (推薦)

客戶自行管理的加密金鑰 (CMEK) 策略

不執行任何操作

單一用戶群

相依性標籤

無

CPU 超額配置

已停用

自訂中繼資料

無

對等 REST

已啟動執行個體



SSH 進去後先安裝一些基本工具：

```
sudo apt update
```

```
p0972931647@gpt:~$ sudo apt update
Hit:1 http://asia-east1.gce.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://asia-east1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://asia-east1.gce.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
```

```
sudo apt install python3 python3-dev python3-venv python3-pip
```

```
p0972931647@gpt:~$ sudo apt install python3 python3-dev python3-venv python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.8.2-0ubuntu2).
python3-dev is already the newest version (3.8.2-0ubuntu2).
python3-venv is already the newest version (3.8.2-0ubuntu2).
python3-pip is already the newest version (20.0.2-5ubuntu1.9).
The following packages were automatically installed and are no longer required:
  libatasmart4 libblockdev-fs2 libblockdev-loop2 libblockdev-part-err2
  libblockdev-part2 libblockdev-swap2 libblockdev-utils2 libblockdev2
  libmbim-glib4 libmbim-proxy libmm-glib0 libnspr4 libnss3 libnuma1
  libparted-fs-resize0 libqmi-glib5 libqmi-proxy libudisks2-0 libxmlb2
  usb-modeswitch usb-modeswitch-data
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
# pip 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)
```

```
pip3 --version
```

```
p0972931647@gpt:~$ pip3 --version
pip 20.0.2 from /usr/lib/python3/dist-packages/pip (python 3.8)
```

接下來建立專案資料夾，我取名叫 line_bot

```
mkdir stock_bot
```

```
cd stock_bot
```

```
p0972931647@gpt:~$ mkdir line_bot
p0972931647@gpt:~$ cd line_bot
p0972931647@gpt:~/line_bot$
```

然後建立專屬虛擬環境，以利後續開發：

```
python3 -m venv venv
```

```
source venv/bin/activate
```

```
p0972931647@gpt:~/line_bot$ python3 -m venv venv
p0972931647@gpt:~/line_bot$ source venv/bin/activate
(venv) p0972931647@gpt:~/line_bot$
```

完成後就開始來寫 python code，先安裝幾個必要的套件，我們使用的 API 框架是 flask，所以先安裝下面的套件：

pip install flask

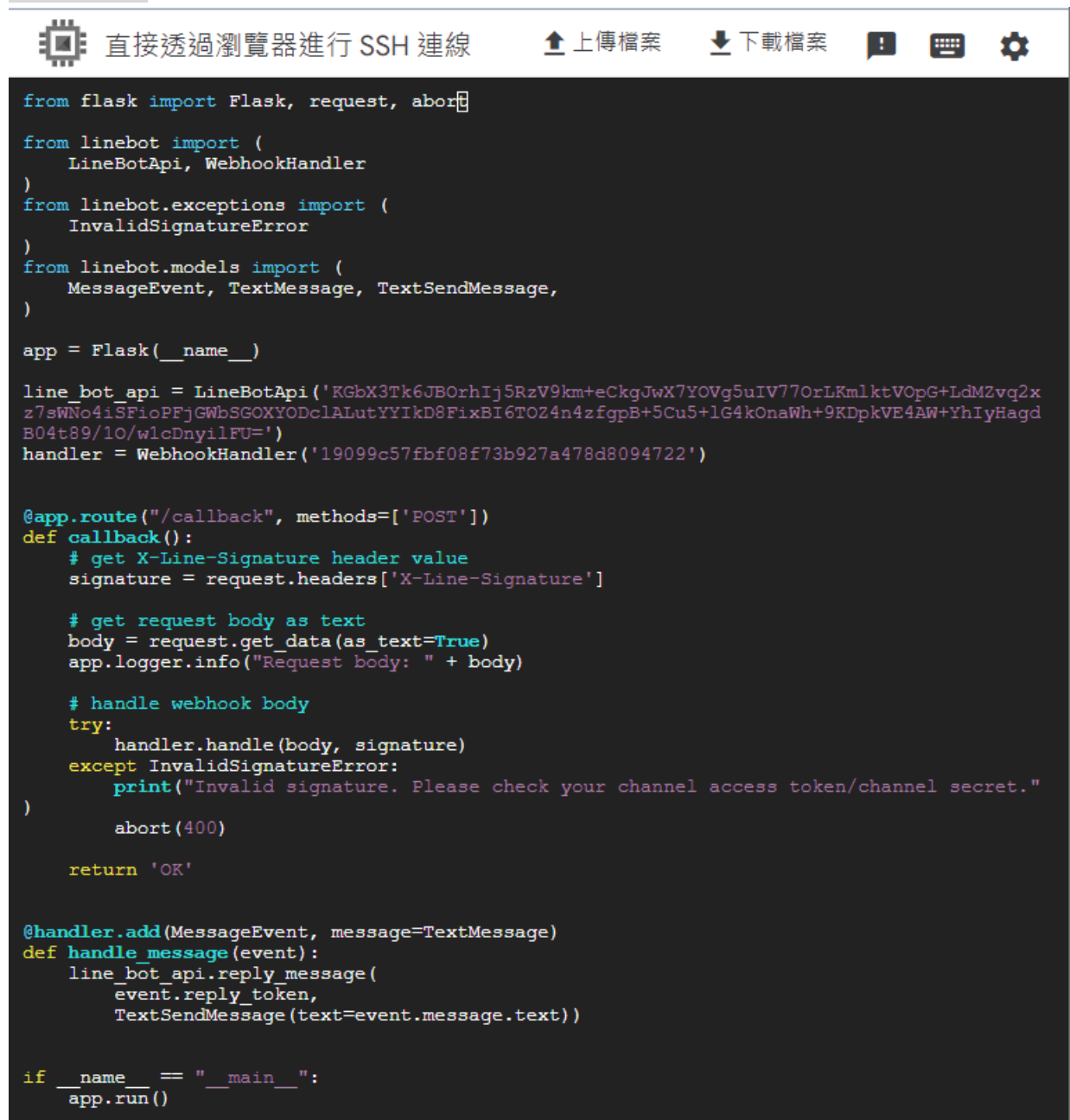
```
(venv) p0972931647@gpt:~/line_bot$ pip install flask
Collecting flask
  Using cached flask-3.0.0-py3-none-any.whl (99 kB)
Collecting Jinja2>=3.1.2
  Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
Collecting blinker>=1.6.2
  Downloading blinker-1.7.0-py3-none-any.whl (13 kB)
Collecting itsdangerous>=2.1.2
  Using cached itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click>=8.1.3
  Using cached click-8.1.7-py3-none-any.whl (97 kB)
Collecting importlib-metadata>=3.6.0; python_version < "3.10"
  Using cached importlib_metadata-6.8.0-py3-none-any.whl (22 kB)
Collecting Werkzeug>=3.0.0
  Using cached werkzeug-3.0.1-py3-none-any.whl (226 kB)
Collecting MarkupSafe>=2.0
  Using cached MarkupSafe-2.1.3-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Collecting zipp>=0.5
  Using cached zipp-3.17.0-py3-none-any.whl (7.4 kB)
Installing collected packages: MarkupSafe, Jinja2, blinker, itsdangerous, click, zipp, importlib-metadata, Werkzeug, flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.3 Werkzeug-3.0.1 blinker-1.7.0 click-8.1.7 flask-3.0.0 importlib-metadata-6.8.0 itsdangerous-2.1.2 zipp-3.17.0
```

pip install requests

```
(venv) p0972931647@gpt:~/line_bot$ pip install requests
Collecting requests
  Using cached requests-2.31.0-py3-none-any.whl (62 kB)
Collecting idna<4,>=2.5
  Using cached idna-3.4-py3-none-any.whl (61 kB)
Collecting certifi>=2017.4.17
  Using cached certifi-2023.7.22-py3-none-any.whl (158 kB)
Collecting charset-normalizer<4,>=2
  Downloading charset_normalizer-3.3.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (141 kB)
    |████████████████████████████████████████| 141 kB 4.1 MB/s
Collecting urllib3<3,>=1.21.1
  Using cached urllib3-2.0.7-py3-none-any.whl (124 kB)
Installing collected packages: idna, certifi, charset-normalizer, urllib3, requests
Successfully installed certifi-2023.7.22 charset-normalizer-3.3.2 idna-3.4 requests-2.31.0 urllib3-2.0.7
```

建立 app.py 的主腳本，line bot 機器人會將打出來的訊息再重複回覆一次：

vim app.py



```
from flask import Flask, request, abort

from linebot import (
    LineBotApi, WebhookHandler
)
from linebot.exceptions import (
    InvalidSignatureError
)
from linebot.models import (
    MessageEvent, TextMessage, TextSendMessage,
)

app = Flask(__name__)

line_bot_api = LineBotApi('KgbX3Tk6JBOrhIj5RzV9km+eCkgJwX7YOVg5uIV77OrLKmlktVOpG+LdMZvq2x
z7sWNo4iSFioPFjGWbSGOXYODclALutYYIkD8FixBI6TOZ4n4zfpgB+5Cu5+lg4kOnaWh+9KDpkVE4AW+YhIyHagd
B04t89/10/wlcDnyilFU=')
handler = WebhookHandler('19099c57fbf08f73b927a478d8094722')

@app.route("/callback", methods=['POST'])
def callback():
    # get X-Line-Signature header value
    signature = request.headers['X-Line-Signature']

    # get request body as text
    body = request.get_data(as_text=True)
    app.logger.info("Request body: " + body)

    # handle webhook body
    try:
        handler.handle(body, signature)
    except InvalidSignatureError:
        print("Invalid signature. Please check your channel access token/channel secret.")
    abort(400)

    return 'OK'

@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    line_bot_api.reply_message(
        event.reply_token,
        TextSendMessage(text=event.message.text))

if __name__ == "__main__":
    app.run()
```

按 ESC 然後輸入:wq 退出程式

三、 部署服務

前面完成後回到 CLI 打上面語法來啟動 flask 服務：

```
python app.py
```

```
(venv) p0972931647@gpt:~/line_bot$ pip install line-bot-sdk
Collecting line-bot-sdk
  Using cached line_bot_sdk-3.5.0-py2.py3-none-any.whl (736 kB)
Collecting Deprecated
  Using cached Deprecated-1.2.14-py2.py3-none-any.whl (9.6 kB)
Collecting future
  Using cached future-0.18.3.tar.gz (840 kB)
Collecting python-dateutil<=2.5.3
  Using cached python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
Collecting aenum>=3.1.11
  Using cached aenum-3.1.15-py3-none-any.whl (137 kB)
Collecting aiohttp==3.8.5
  Using cached aiohttp-3.8.5-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.1 MB)
Requirement already satisfied: requests==2.31.0 in ./venv/lib/python3.8/site-packages (from line-bot-sdk) (2.31.0)
Collecting pydantic<3,>=2.0.3
  Using cached pydantic-2.4.2-py3-none-any.whl (395 kB)
Collecting wrapt<2,>=1.10
  Using cached wrapt-1.15.0-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (81 kB)
Collecting six>=1.5
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Collecting yarl<2.0,>=1.0
  Using cached yarl-1.9.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (266 kB)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in ./venv/lib/python3.8/site-packages (from aiohttp==3.8.5->line-bot-sdk) (3.3.2)
Collecting attrs>=17.3.0
  Using cached attrs-23.1.0-py3-none-any.whl (61 kB)
Collecting aiosignal>=1.1.2
  Using cached aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
Collecting multidict<7.0,>=4.5
  Using cached multidict-6.0.4-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (121 kB)
Collecting frozenlist>=1.1.1
  Using cached frozenlist-1.4.0-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (220 kB)
Collecting async-timeout<5.0,>=4.0.0a3
  Using cached async_timeout-4.0.3-py3-none-any.whl (5.7 kB)
Requirement already satisfied: idna<4,>=2.5 in ./venv/lib/python3.8/site-packages (from requests==2.31.0->line-bot-sdk) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in ./venv/lib/python3.8/site-packages (from requests==2.31.0->line-bot-sdk) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in ./venv/lib/python3.8/site-packages (from requests==2.31.0->line-bot-sdk) (2023.7.22)
Collecting pydantic-core==2.10.1
  Using cached pydantic_core-2.10.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.1 MB)
Collecting annotated-types>=0.4.0
  Using cached annotated_types-0.6.0-py3-none-any.whl (12 kB)
Collecting typing-extensions>=4.6.1
  Using cached typing_extensions-4.8.0-py3-none-any.whl (31 kB)
Building wheels for collected packages: future
Building wheel for future (setup.py) ... error
ERROR: Command errored out with exit status 1:
command: /home/p0972931647/line_bot/venv/bin/python3 -u -c 'import sys, setuptools, tokenize;
(venv) p0972931647@gpt:~/line_bot$ python app.py
app.py:15: LineBotSdkDeprecatedIn30: Call to deprecated class LineBotApi. (Use v3 class; linebot.v3.<feature>. See https://github.com/line/line-bot-sdk-python/blob/master/README.rst for more details.) -- Deprecated since version 3.0.0.
  line_bot_api = LineBotApi('KgbX3Tk6JB0rhIj5RzV9km+eCkgJwX7YOVg5uIV770rLKmlktVOPG+LdM2vq2xz7sWN04iSFioPFjGwbSGOXYODclALutYYIkD8FixBI6TOZ4n4zfGPB+5Cu5+lG4kOnaWh+9KDpkVE4AW+YhIyHagdB04t89/1O/wlcDnyilFU=')
app.py:16: LineBotSdkDeprecatedIn30: Call to deprecated class WebhookHandler. (Use 'from linebot.v3.webhook import WebhookHandler' instead. See https://github.com/line/line-bot-sdk-python/blob/master/README.rst for more details.) -- Deprecated since version 3.0.0.
  handler = WebhookHandler('19099c57fbf08f73b927a478d8094722')
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [04/Nov/2023 06:31:52] "POST /callback HTTP/1.1" 200 -
127.0.0.1 - - [04/Nov/2023 06:31:56] "POST /callback HTTP/1.1" 200 -
```

開一個新的 SSH 連線操作，根據的步驟安裝並建立連線

```
wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
```

```
p0972931647@gpt:~$ wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
--2023-11-03 08:38:42-- https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
Resolving bin.equinox.io (bin.equinox.io)... 54.161.241.46, 54.237.133.81, 52.202.168.65, ..
.
Connecting to bin.equinox.io (bin.equinox.io)|54.161.241.46|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13921656 (13M) [application/octet-stream]
Saving to: 'ngrok-stable-linux-amd64.zip'

ngrok-stable-linux-amd 100%[=====>] 13.28M 6.56MB/s in 2.0s

2023-11-03 08:38:45 (6.56 MB/s) - 'ngrok-stable-linux-amd64.zip' saved [13921656/13921656]
```

```
sudo apt install unzip
```

```
p0972931647@gpt:~$ sudo apt install unzip
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-25ubuntu1.1).
The following packages were automatically installed and are no longer required:
  libatasmart4 libblockdev-fs2 libblockdev-loop2 libblockdev-part-err2 libblockdev-part2
  libblockdev-swap2 libblockdev-utils2 libblockdev2 libmbim-glib4 libmbim-proxy
  libmm-glib0 libnspr4 libnss3 libnuma1 libparted-fs-resize0 libqmi-glib5 libqmi-proxy
  libudisks2-0 libxmlb2 usb-modeswitch usb-modeswitch-data
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
unzip ngrok-stable-linux-amd64.zip
```

```
p0972931647@gpt:~$ unzip ngrok-stable-linux-amd64.zip
Archive: ngrok-stable-linux-amd64.zip
replace ngrok? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflating: ngrok
```

```
./ngrok authtoken [your_token]
```

```
p0972931647@gpt:~$ ./ngrok authtoken 2XVpKVldtOElvIsiWzc42emn4Dh_2yzkvjon2ZUSgA4AgLnZh
Authtoken saved to configuration file: /home/p0972931647/.ngrok2/ngrok.yml
```

```
./ngrok http 5000
```

```
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account              GuanXin (Plan: Free)
Version              2.3.41
Region               United States (us)
Web Interface         http://127.0.0.1:4040
Forwarding            http://8073-35-194-210-79.ngrok-free.app -> http://localhost:50
Forwarding            https://8073-35-194-210-79.ngrok-free.app -> http://localhost:50

Connections          ttl     opn     rt1     rt5     p50     p90
                     0       0       0.00    0.00    0.00    0.00
```

紅色框起的網址就是有 SSL 轉跳後的網址，而該網址會轉向到 flask 服務的 5000 port，該網址後面加上 /callback 就是我們的 webhook URL 了。

貼上去 webhook URL 並開啟下面的 Use webhook 即可

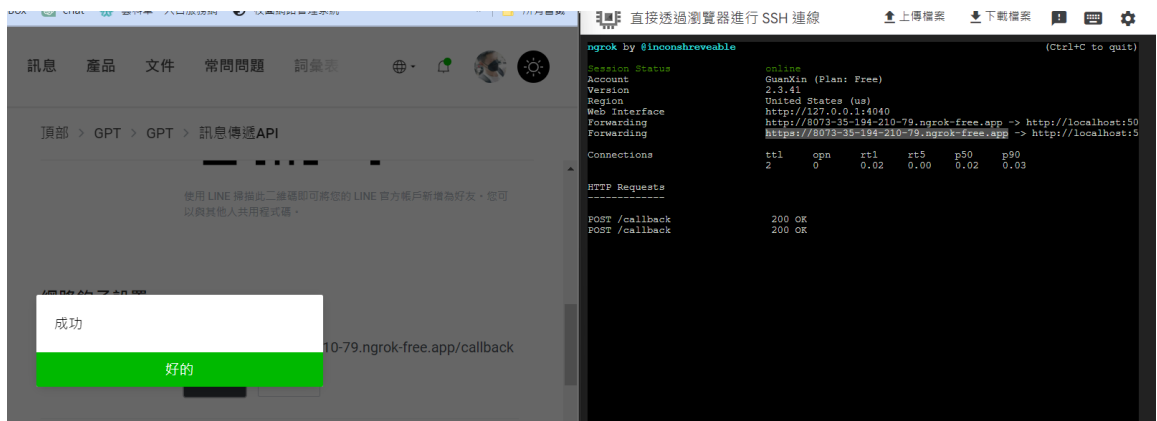
Webhook settings

Webhook URL ? <https://8073-35-194-210-79.ngrok-free.app/callback>

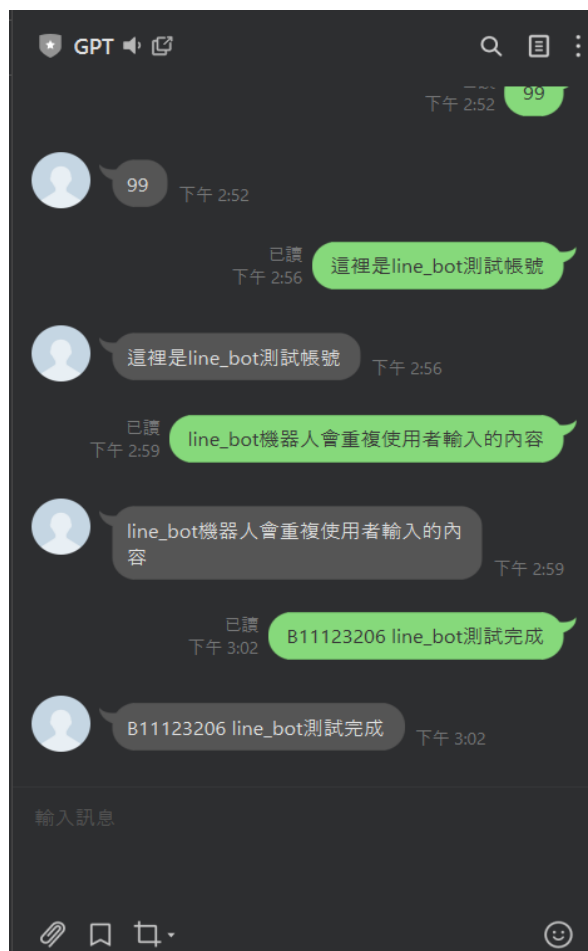
Verify

Edit

Use webhook ?



line bot 測試畫面，line bot 機器人會將打出來的訊息再重複回覆一次



肆、 參考資料

1. 什麼是 IaaS ? 基礎結構即服務 <https://azure.microsoft.com/zh-tw/resources/cloud-computing-dictionary/what-is-iaas/>
2. AWS vs MS Azure vs Google Cloud : 概述、優缺點
<https://novateus.com/blog/aws-vs-ms-azure-vs-google-cloud-overview-pros-and-cons/>
3. Medium
<https://medium.com/@chrislee0728>
4. ngrok
<https://dashboard.ngrok.com/get-started/setup>
5. GCP
<https://console.cloud.google.com/compute/instances?hl=zh-tw&project=dynamic-style-403414>
6. LINE Developers
<https://developers.line.biz/console/>