

# 從 VSCode 將 Python AI Linebot 部署到 AWS

完整部署步驟與雲端部署的重要性

# 簡介

- 目標：將 Python 撰寫的 AI Linebot 部署至 AWS Lambda
- 工具：VSCode、AWS CLI、Zappa、PostgreSQL、Line Messaging API

# 步驟總覽

- 1. 建立資料庫與資料表
- 2. 安裝與設定 AWS CLI
- 3. 使用 Zappa 將程式部署到 Lambda
- 4. 產生 Lambda URL 並整合到 Linebot

# 步驟一：建立資料庫

- 使用 AWS RDS 建立 PostgreSQL 資料庫
- 設定實例名稱、使用者帳號與安全性群組

# 步驟一：建立資料表

- 根據遊戲邏輯設計資料表（如使用者、對話紀錄等）
- 使用 SQL 建立資料表並加上索引提升查詢效率

## 步驟二：安裝 AWS CLI

- 從 AWS 官網下載 CLI 工具
- 可使用 pip、brew 或執行安裝檔進行安裝

## 步驟二：設定 AWS CLI

- 執行 `aws configure` 指令
- 輸入 Access Key、Secret Key、Region 與預設格式

## 步驟三：安裝 Zappa

- 執行 `pip install zappa`
- 可快速將 Python 應用部署到 AWS Lambda

## 步驟三：初始化 Zappa

- 執行 `zappa init` 建立設定檔
- 依照提示設定項目名稱、S3 存儲桶、Python 環境等

## 步驟三：部署至 AWS Lambda

- 執行 `zappa deploy production` 進行首次部署
- 可使用 `zappa update production` 更新程式

## 步驟四：取得 Lambda URL

- Zappa 部署完成後會產生 HTTPS 的 URL
- 此 URL 可作為 Webhook 使用於 Line Messaging API

## 步驟四：整合到 Linebot

- 登入 Line Developers 管理後台
- 在 Messaging API 的 Webhook 欄位填入 Lambda URL

# 雲端部署的重要性（一）

- 隨時可擴展：根據使用者需求自動調整資源
- 不需自行管理伺服器，節省維運成本

# 雲端部署的重要性（二）

- 提高可靠性與可用性
- 支援持續整合與快速迭代部署流程

# 部署後的維護建議

- 使用 CloudWatch 監控 Lambda 執行狀況
- 定期備份資料庫與檢查資安設定

# 結語

- 結合 VSCode 與 AWS 工具可快速部署 AI Linebot
- 善用 Zappa 簡化部署流程，實現雲端運算的彈性與效能