

雲端氣象建議系統

以 AWS 為基礎，設計並部署一個[簡易雲端氣象查詢與建議系統](#)。使用者可在網頁上輸入英文城市名稱，系統即時顯示當地天氣資訊與當日外出建議。



系統說明

- 前端 (React + Vite)

建立響應式使用者介面，提供輸入與查詢功能（輸入城市名稱：如：Taipei, London, Toronto ... etc）。經建置後部署於 AWS S3，並透過 CloudFront 做內容分發。

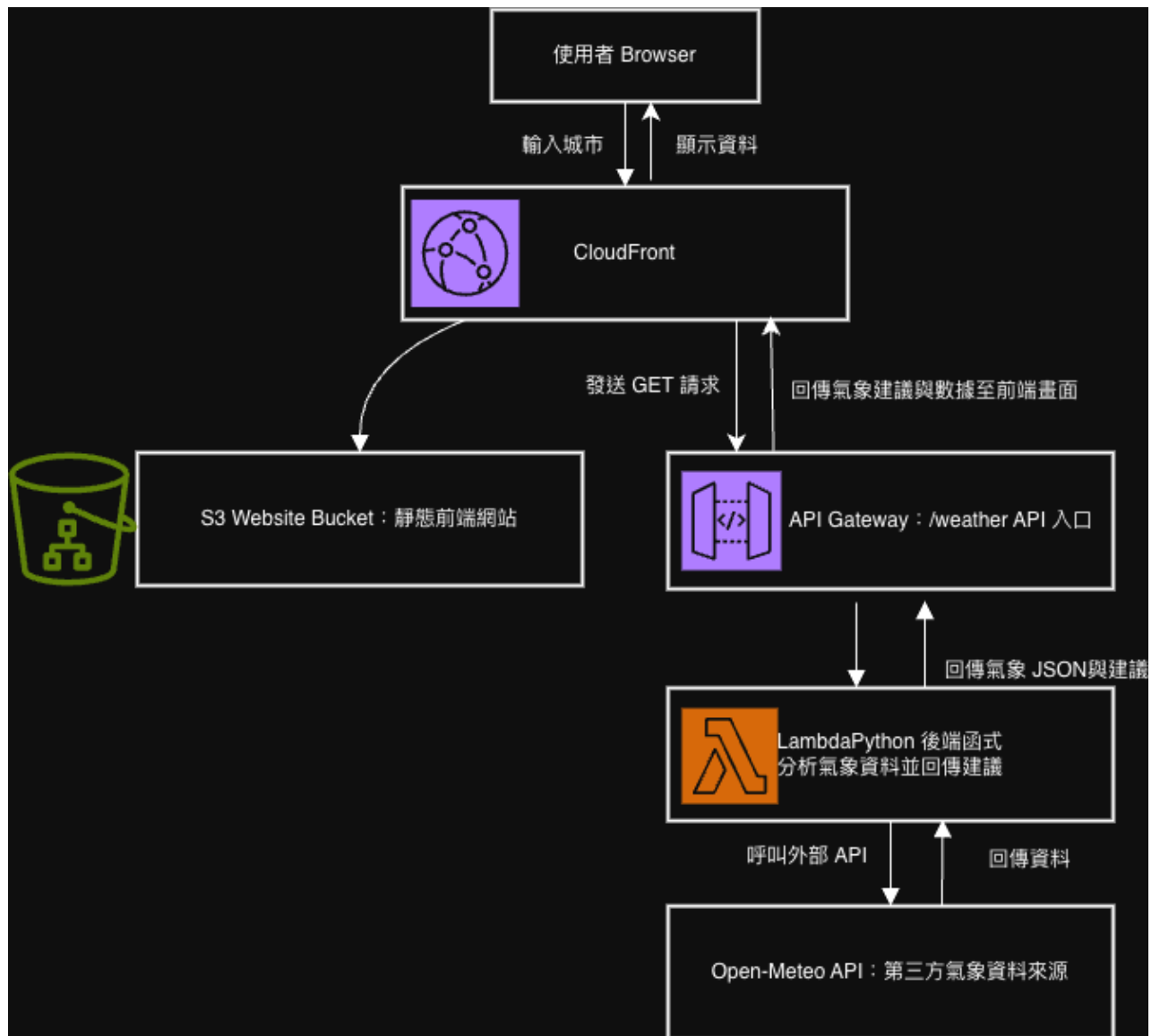
- 後端 (AWS Lambda + API Gateway)

Lambda 以 Python 撰寫，接收前端 GET 請求後呼叫 **Open-Meteo API** 取得氣象資料，經分析後回傳氣溫、風速與外出建議等 JSON 結果。

- 資料流

使用者 → CloudFront → S3 (前端畫面)

→ API Gateway → Lambda → Open-Meteo API → Lambda 回傳 → 前端顯示。



CI/CD 架構

CI(持續整合)階段

1. 開發者修改程式碼後 **Push** 至 **GitHub Repo**

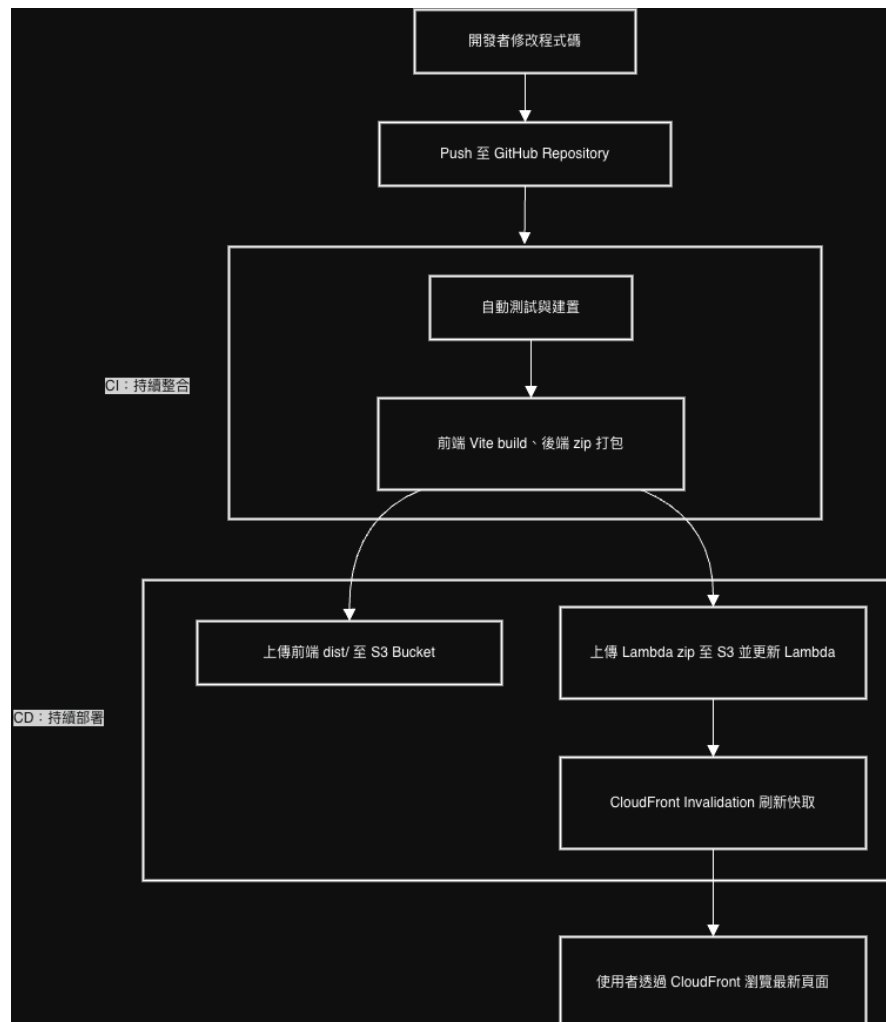
在本地開發環境中更新前端或後端邏輯。模擬自動化觸發點，未來可連接 GitHub Actions 或 AWS CodePipeline。

2. 打包程式與建置

前端: `npm run build`, 後端: `zip -r weather_api.zip .`

CD(持續部署)階段

1. 將最新網站部署至 **S3** (上傳前端 **dist/** 至 **S3 Bucket**)
 - `aws s3 sync dist/ s3://frontend-infra-frontend-bucket`
2. 上傳後端 **Lambda zip** 至 **S3** 並更新部署
 - `aws s3 cp weather_api.zip`
 - `s3://frontend-infra-frontend-bucket/weather_api.zip`
3. 透過 **CloudFormation** 自動更新 **Lambda**
 - `aws cloudformation deploy --template-file template.yml --stack-name weather-backend`
4. 清除舊版 **CloudFront** 快取刷新
 - `aws cloudfront create-invalidation`



多人協作的 **Git** 流程圖

- **main**: 觸發更新
- **dev**: 整合開發分支, 收納所有已完成的功能分支
- **feature/frontend**: 前端開發網頁、API 對應之畫面
- **feature/backend**: 後端開發: AWS Lambda、API Gateway、資料分析
- **test**: 確認所有模組能正確協作

