Github Action

Github Actions 的基本元素

要了解 Github Actions 的基本概念，有幾個元素（或是術語）是必須要知道的，範圍由大至小分別為：

```
Workflow -> Job -> Step -> Action
```

- Workflow : CI/CD 一次要運行的整個過程，就稱作 Workflow，一個 Workflow 裡會涵蓋多個 Job、Step、 Action。
- Job : 意義跟翻成中文差不多，代表「任務」。一個 Workflow 由多個 Job 組成，這也意謂著 ***一個 Workflow 可以完成多個任務***。
- Step : 代表一個個「步驟」，一個 Job 由多個 Step 組成，意謂著一個 Job 是一個步驟一個步驟來完成的。
- Action :「命令」或「動作」，每個 Step 可以依序執行多個命令（動作）。

Github Actions Workflow Config

要啟用 Github Actions，不需要什麼複雜的設定，只需要在專案的根目錄新增 .github/workflow 的路徑，將專案推到 Github 上後 Github 就會自動執行放在該路徑裡的 .yml config 檔 (Workflow document 採用 YAML 格式)。.yml 檔的檔名可以隨意取，也可以創建多個 .yml 檔，Github 會執行 .github/workflow 路徑下的所有 YAML 文件。

以下是python flask linebot範例:

```yaml
# action名稱
name: line-bot-Gemini
#監聽 當*發生時
on:
  push:
    branches: [ "self-use" ]
  pull_request:
    branches: [ "self-use" ]
#做以下的事情
jobs:
  #做的事情名稱
  build_and_deployment:
  # 跑在自己的伺服器上 下面範例是用github action runner 綁定帳戶的
    runs-on: self-hosted  # windows server 2018
```

```
    #步驟
    steps:
    - uses: actions/checkout@v2
    - name: Set up Python 3.11
    - #建立python環境 確保環境一致
      uses: actions/setup-python@v2  # 將 uses 更改為 actions/setup-python@v2
      with:
        python-version: "3.11"
    - name: Install dependencies
    - # 安裝所需套件
      run: |
          python -m pip install --upgrade pip
          if (Test-Path requirements.txt) { pip install -r requirements.txt }
```

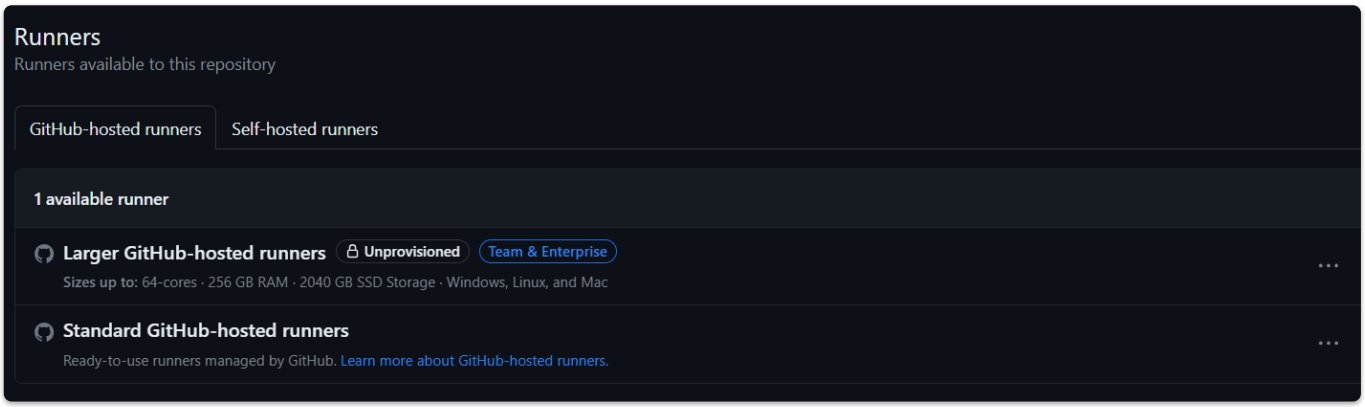
以下是dotnet framework 部屬至IIS網頁伺服器:

```
name: .NET framework 4.7.6 Deployment

# 監聽以下事件發生
on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]
# 做以下事情
jobs:
  build:
  # 自己的伺服器
    runs-on: self-hosted
    steps:
    # 環境建立測試
    - uses: actions/checkout@v3
    - name: Setup .NET 4.7.2
      uses: actions/setup-dotnet@v2
      with:
        dotnet-version: 4.7.2
        # 還原依賴
    - name: Restore dependencies
      run: dotnet restore
        # 構建程式
    - name: Build
      run: dotnet build --no-restore
       # 發布程式
    - name: Publish
      run: dotnet publish -c Release -o published
```

```
    # 部屬 IIS 伺服器
  # deployment:
  #   needs: build
  #   runs-on: self-hosted
    #部屬iis
    # - name: Deploy to IIS
    #   run: |
    #     # 使用 xcopy 將文件複製到 IIS 目錄
    #     xcopy /Y /S .\ISHAudit C:\inetpub\wwwroot\ishapublish\
          #            原目錄      目標目錄
    #     # 如果需要停止和啟動應用程序，可以使用以下命令
    #     # iisreset
```

runs-on 參數



github runner 也有免費和租用伺服器提供你選擇
參數填上
windows:

```
windows-latest
```

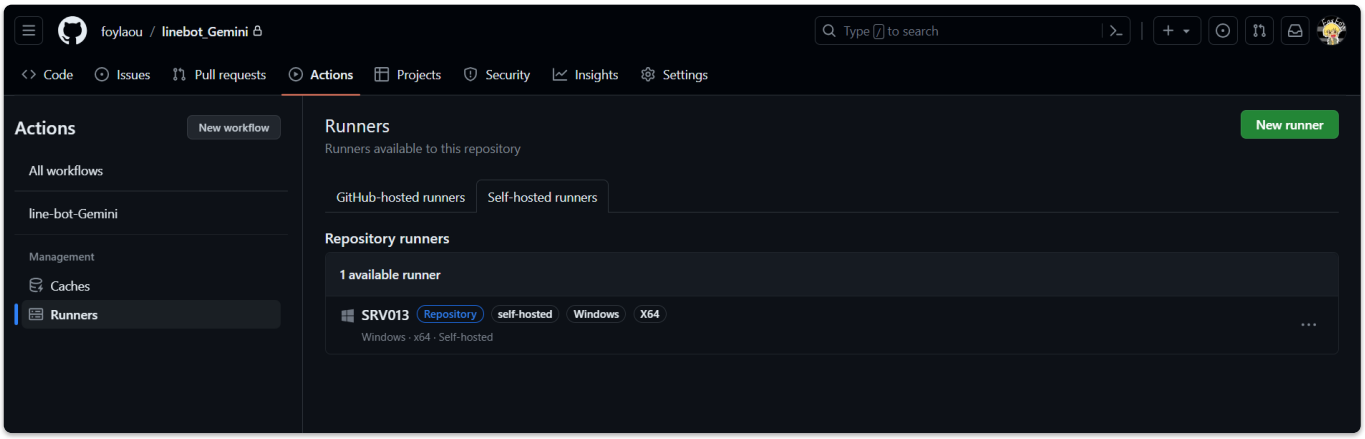
linux:

```
ubuntu-latest
```

MacOs:

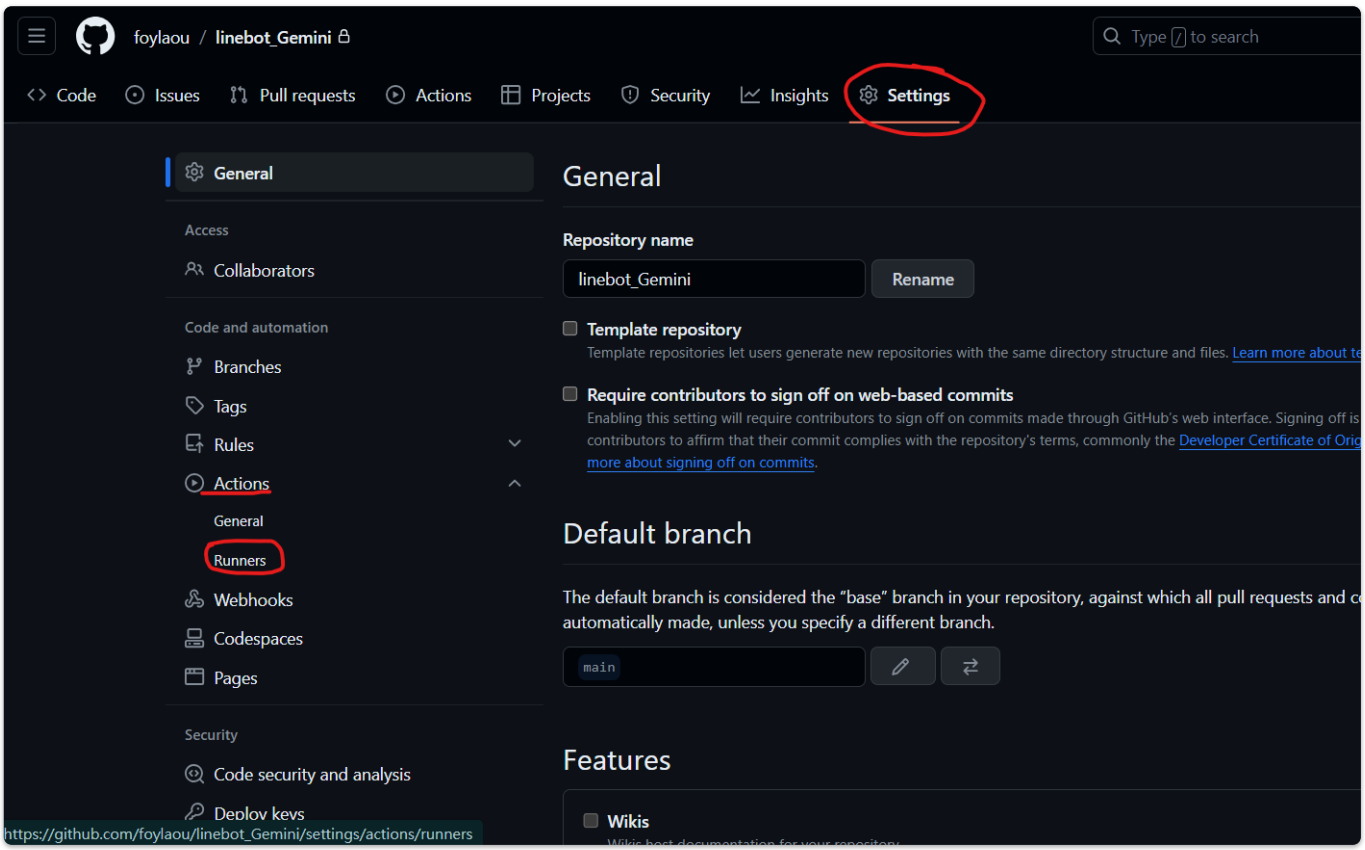
```
macos-latest
```

如何設定自己伺服器當runs-on

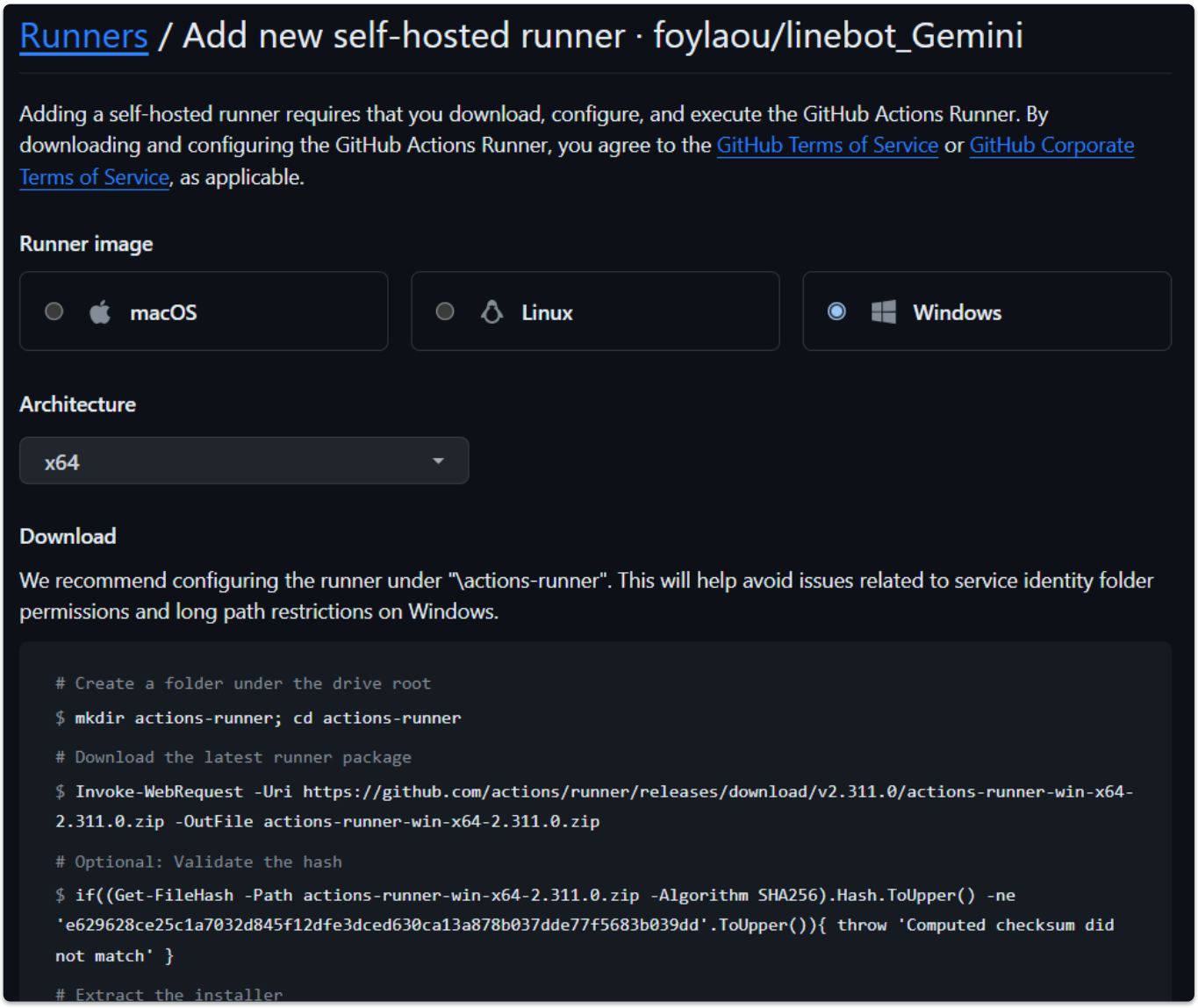


這是設定自己的伺服器

先來到專案設定->Action->Runner



就會看到以下頁面，選擇伺服器上執行的作業系統環境進行安裝



我是用windows server 請使用powershell以系統管理員身分執行

```
# 建立資料夾在裝置的跟目錄下
mkdir actions-runner
cd actions-runner
# 下載最新版github-runner
Invoke-WebRequest -Uri
https://github.com/actions/runner/releases/download/v2.311.0/actions-runner-win-x64-2.311.0.zip -OutFile actions-runner-win-x64-2.311.0.zip
# 選用:驗證資料hash值
if((Get-FileHash -Path actions-runner-win-x64-2.311.0.zip -Algorithm SHA256).Hash.ToUpper() -ne 'e629628ce25c1a7032d845f12dfe3dced630ca13a878b037dde77f5683b039dd'.ToUpper()){ throw 'Computed checksum did not match' }
# 解壓縮並安裝
Add-Type -AssemblyName System.IO.Compression.FileSystem ;
[System.IO.Compression.ZipFile]::ExtractToDirectory("$PWD/actions-runner-win-x64-2.311.0.zip", "$PWD")
```

```
# 建立github-runner程式的設定檔
 ./config.cmd --url https://github.com/foylaou/linebot_Gemini --token
APMS32DGPOD3DBSM62GOSJTFTNORE
# 執行程式
 ./run.cmd
```

這樣你就會看到你的頁面出現server裝置了