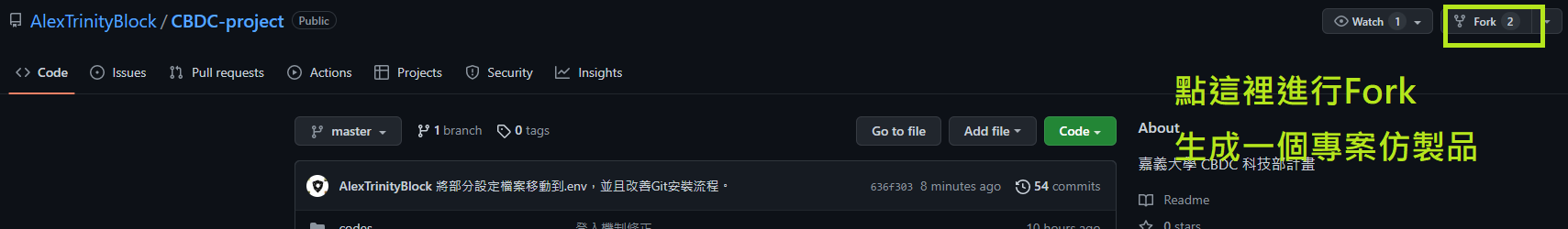


|  |
| --- |
| 啟動專案 |
|  |
| 10月30日  開啟CBDC的容器群 |

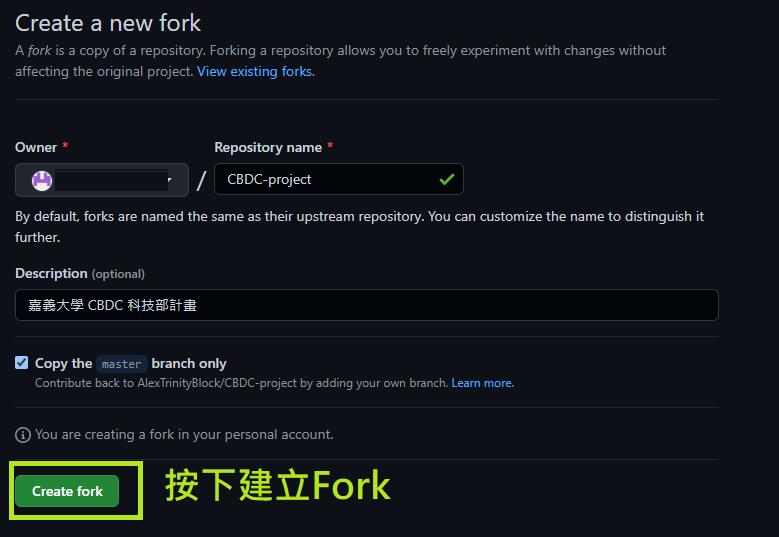
# Fork 專案

登入Github然後到以下網址，對實驗室的專案進行Fork，也就是建立專屬於你的專案備份。

<https://github.com/AlexTrinityBlock/CBDC-project>



不用改動任何選項，按下綠色按鈕。



# Git 下載專案

從Github 上對專案進行clone。



開啟 powershell 終端輸入指令進行Clone

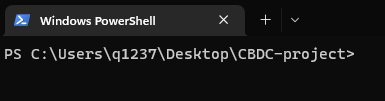
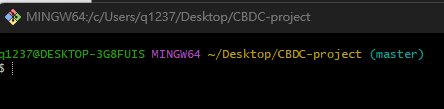
cd 你想放專案的路徑

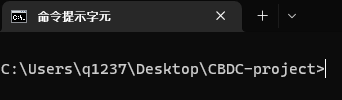
git clone git@github.com:你的帳號/CBDC-project.git

這樣專案就會被下載了。

# 設置你的Fork專案與原始專案的聯繫

用任意一種Shell 命令行切換到專案目錄



然後執行以下三行指令讓您本地撰寫的程式可以達到順利提交(Pull Request)的目標:

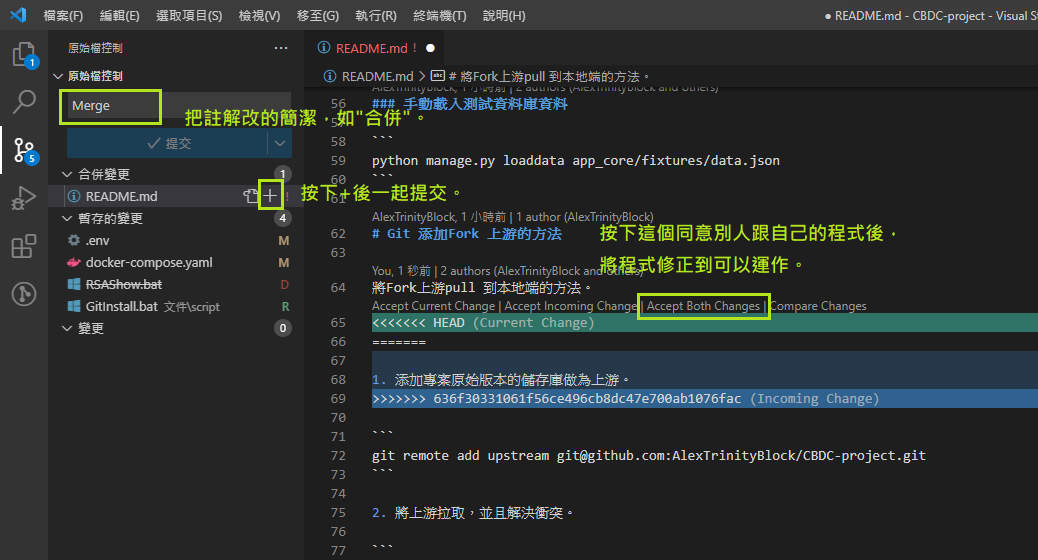
1. 將原始的儲存庫稱為 upstream 加以記錄:

git remote add upstream git@github.com:AlexTrinityBlock/CBDC-project.git

1. 從遠端儲存庫下載最新的開發進度，並且和本地端合併。

git pull upstream master

1. 打開 Visual Studio Code 查看是否有衝突，如果遠端跟本地有衝突，代表有人也寫了些程式，這個時候，最好選擇你的和她/他的程式都加以保留的按鈕。

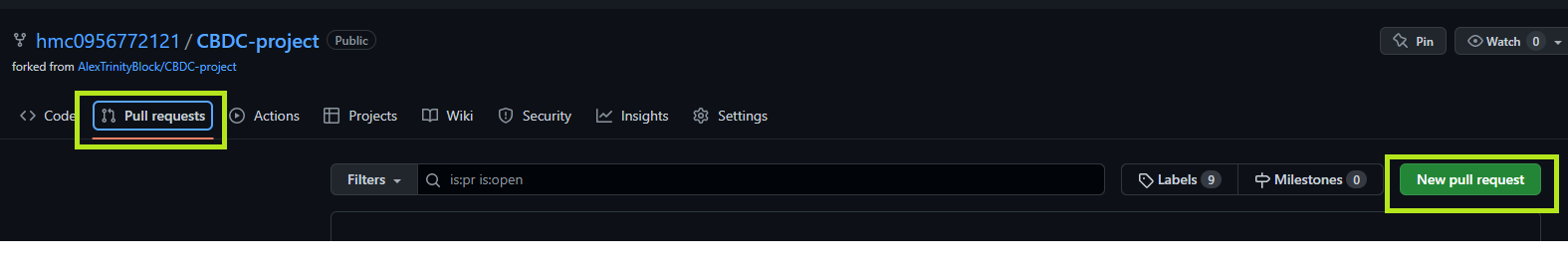


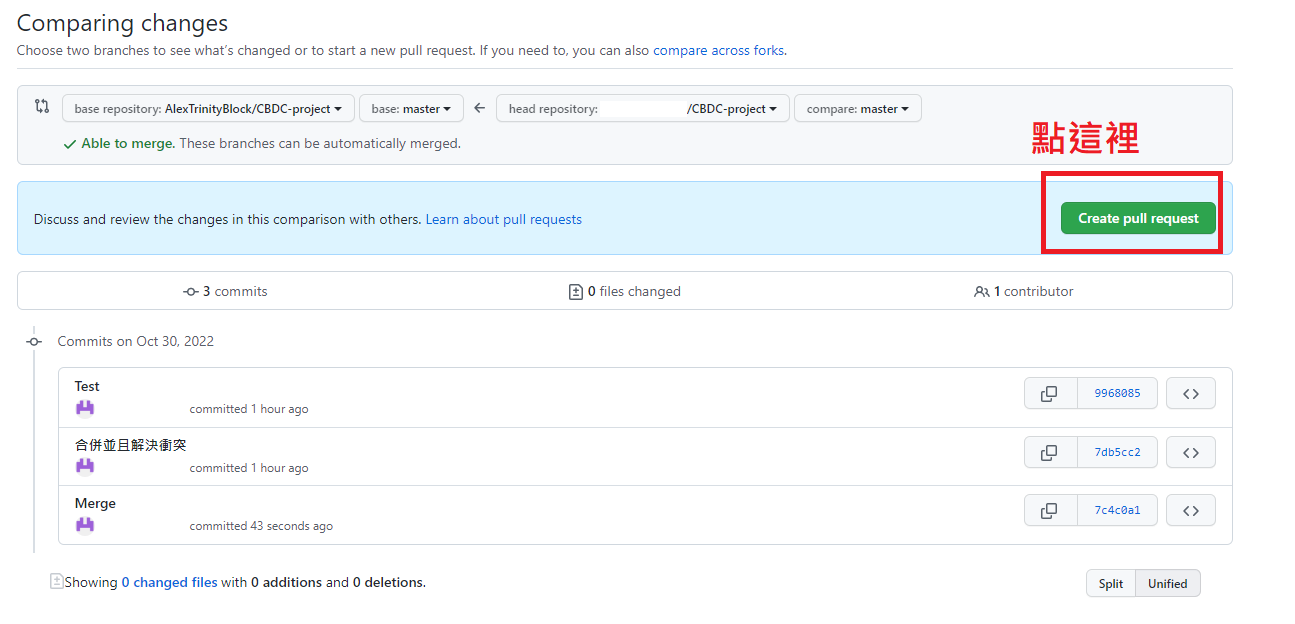
1. 最後推送到你雲端的儲存庫

命令行輸入:

git push origin master

1. 然後發送Pull Request 給負責代碼整合的人員







# 啟動Docker Desktop

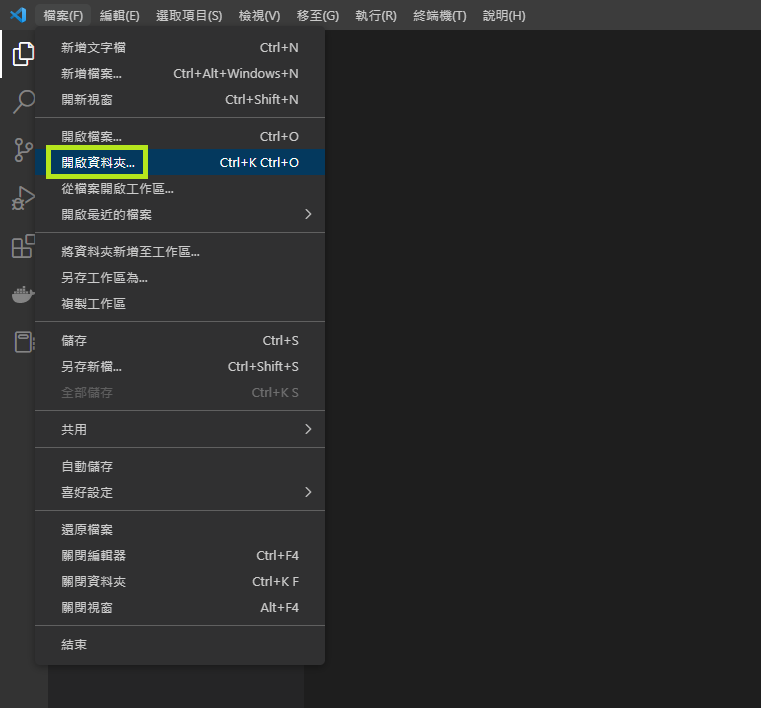
|  |
| --- |
| Docker可以幫我們省去數小時的配置時間 我們已經裝好Docker Desktop 了，左鍵把它點開吧 ! |
|  |

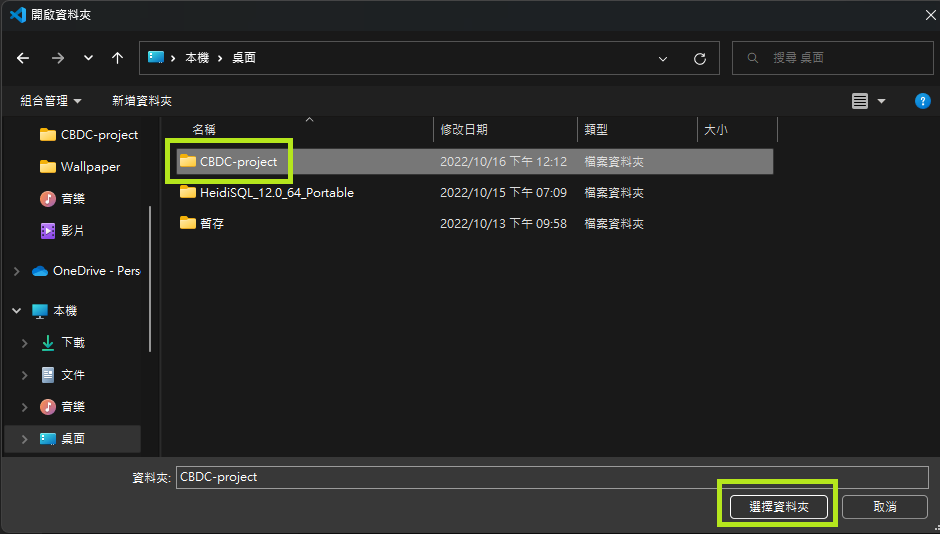
若左下角始終是橘色的，代表 Dokcer 安裝出問題，請直接向團隊中對 Docker 與作業系統了解的人員共同完成安裝。(並且分析問題，然後寫成文檔，以便下個人理解問題所在)

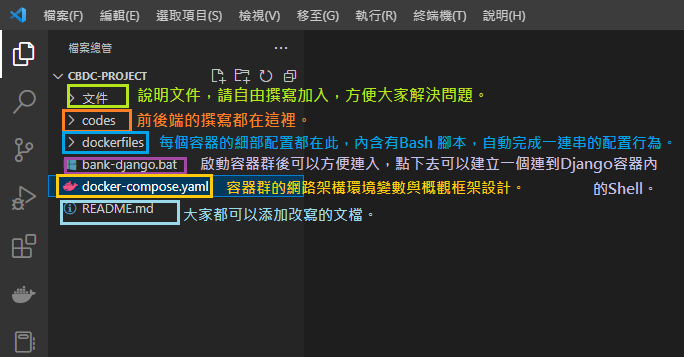
# 打開 Visual Studio Code

Visual Studio Code 內建的終端功能，可以幫我們完成所有我們的命令行任務。

首先開啟專案資料夾。

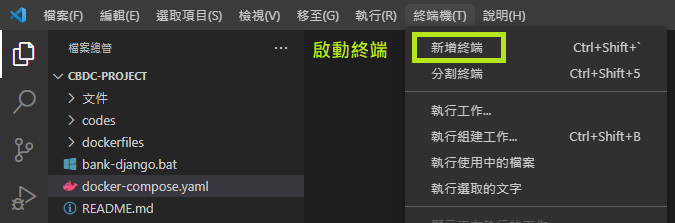






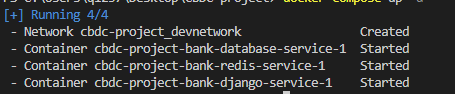
# 啟動Docker 容器群

啟動終端。



輸入指令 docker-compose up -d





輸入 docker-compose ps 在終端或者啟動 Docker desktop 可以看到啟動的數個容器。

