

Design Document

架構

設計

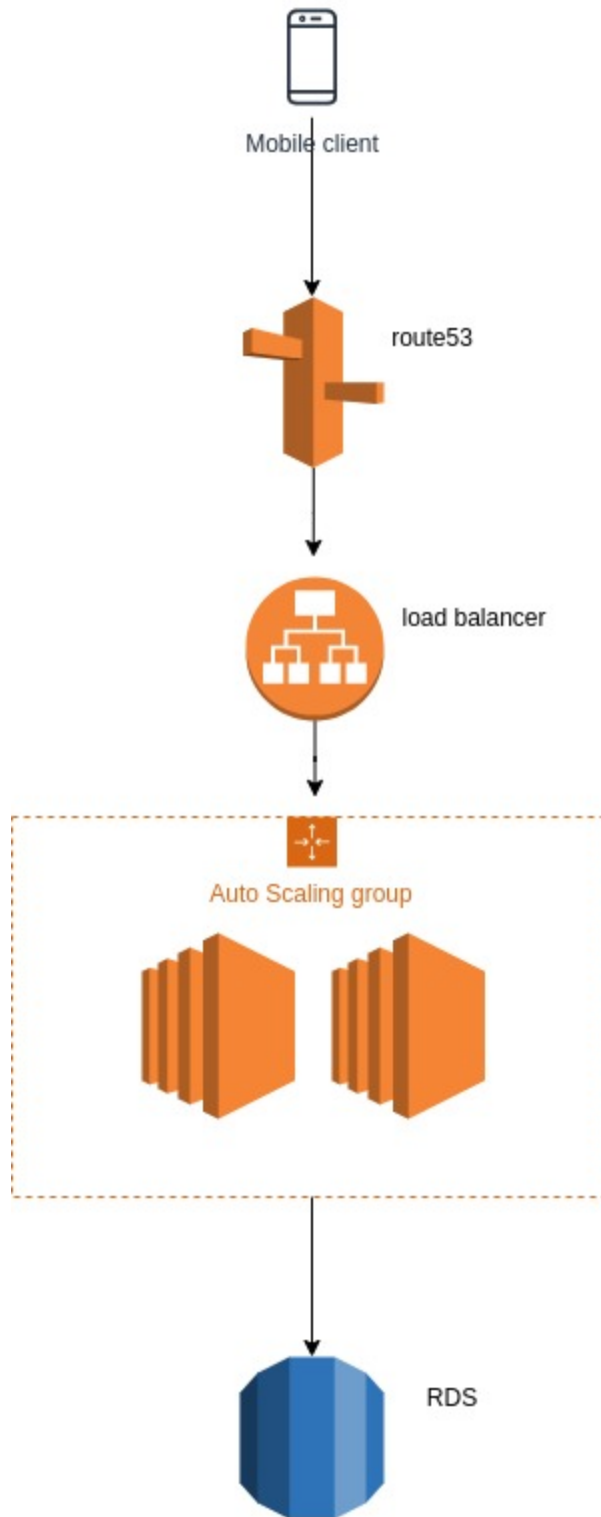
- 將 TODO list 分為兩種，一種為一次性的任務，一種為週期性的任務，並分別存在不同的 table 中，分別為 tasks 以及 routine_task_infos
- routine_task_infos 其中一個欄位 period 存這個週期性任務的週期多長，目前支援年、月、週、日，而欄位 routine_time 則為這個週期內的何時，例如 year 的起始時間是每年 1/1 00:00，從起始時間開始到欲重複時間所經過的秒數。假設此重複性任務是每年的 1/1 3:00，則此欄位的值為 10800
- 週期性任務會有多個子任務存在 routine_tasks 這個 table 內，其中會有一個是下一次的任務，其他都是已經超過 due_date 的任務
- 使用者在取的 TODO list 時會在後端檢查此任務是否已經過期，如果過期且還沒被設為 completion，則將其設為 delay
- 「弱聯網」的支援可以透過 client 端在記下沒有網路前最後一次與 server 端獲取的資訊，並在連上網路後比對該資訊與之後使用者經過操作修改後的結果的差異，並將兩者的差異一一透過原本的 API 更新到 server 端
- 目前提供更新任務的 API 一次只能更新一個屬性的值，無法批次更新多個任務或是一次更新某個任務的多個屬性，如此可以較容易避免同時有多個 app 操作造成的衝突，但更新效率可能較低，使用者體驗會較差

問題

- 目前週期性任務在此週期內只能發生一次，無法發生多次以上 (如每年兩次、每月兩次)
- 週期性任務在 period 為月的情況下並不會特別處理每個月天數不同的問題
- 當使用者同時使用兩個裝置，且其中一個裝置處於未連網的狀態下更新任務，另外一個裝置在此裝置未連網的時候更新 server 端的資料，當未連網裝置重新連上網後有可能會覆蓋掉另一裝置更新的結果

Infrastructure

- 以 AWS 的元件為例



Database Schema

