**國立暨南國際大學**

附件一

**獎勵大學部學生參與專題研究計畫申請書**

一、綜合資料: 申請日期： 111 年 6 月 日

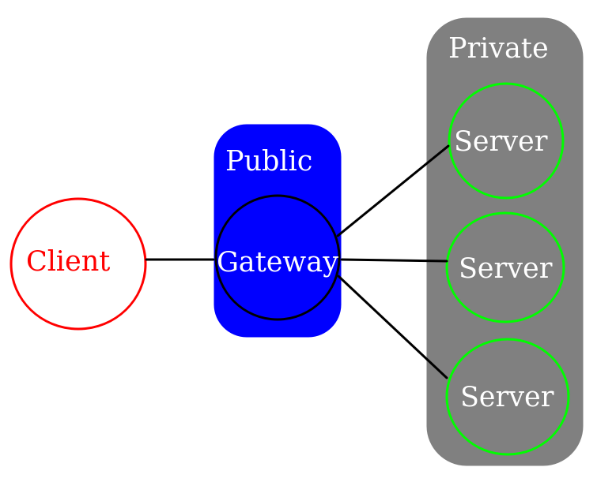
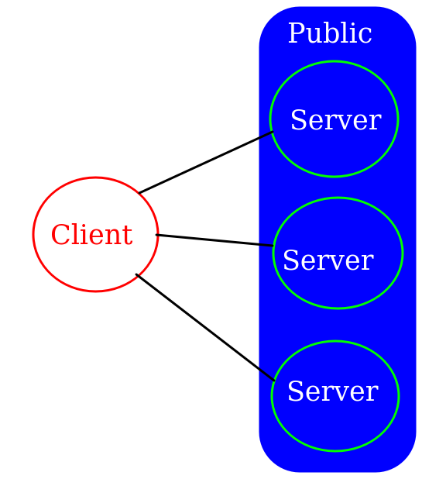
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 申 請 人  │學 生 | 姓名 | | 洪胤勛 | | 系(學程)年級 | | | 資工系大三 | |
| 學號 | | 108321019 | | 聯絡電話及e-mail | | | 0965598275  s108321019@mail1.ncnu.edu.tw | |
| 學生研究計畫名稱 | | 可延展式與容錯性WebAPI安全管制機制之設計 | | | | | | |
| 研究期間 | | 自\_\_111\_\_\_年8月1日至\_\_112\_\_\_年1月31日止，計6個月 | | | | | | |
| 計畫歸屬司別 | | □自然司■工程司 □生科司 □人文司(含科學教育領域) | | | | | | |
| 已申請且未通過本年度科技部大專學生研究計畫條碼編號 | | E4003 計算機網路與網際網路 | | | | | | |
| 曾申請科技部大專學生研究計畫 | | □通過  計畫編號： (不敷使用請自行增加)  ■未通過  條碼編號： E4003 (不敷使用請自行增加)  □未曾申請 | | | | | | |
| 指  導  教  授 | 姓名 | | 吳坤熹 | | | | | | |
| 系(所) | | 資訊工程學系 | | | | | | |
| 職稱 | | 副教授 | | | | 電話 | | (049)2910960  Exit 4736 |
| 每位學生每月補助研究獎學金2,000元，研究期間6個月，共計12,000元 | | | | | | | | | |
| 身分證字號 | | N126356072 | | 戶籍通訊地址 | | 彰化縣伸港鄉七嘉村崙頂街38巷9弄6號 | | | |
| 郵局(銀行)局號 | | 0081181 | | 銀行(含分行)名稱 | |  | | | |
| 郵局(銀行)帳號 | | 0361742 | | | | | | | |
| * 以上資料為本研究獎學金申請獲通過補助後使用，請於填寫資料後，確實核對個人基本資料及郵局(銀行)局號、帳號，以利出納組順利將獎學金轉帳至帳戶內。 * 茲同意將本人獲通過「獎勵大學部學生參與專題研究計畫」研究獎學金轉帳至個人帳戶內，並已確認以上資料正確無誤。   申請人：  (簽章) | | | | | | | | | |

二、研究計畫摘要—請對研究計畫之問題依下列各項述明：

(1)摘要

在一個分工的時代，工程師也要懂得如何分工。在軟體工程裡面，有個專有名詞叫做 API (Application Programming Interface)。他是一個溝通的介面，旨在讓工程師們專心在他們的任務，而不需要管理其他層面的問題。比如說一個人用電腦時，他只要知道鍵盤、滑鼠可以輸入資料，螢幕可以輸出結果，但他並不需要去知道電腦是如何運作的。如果是網頁的開發，通常會分為前端和後端，前端負責設計使用者介面，後端負責資料的儲存，他們溝通的介面，又可稱為 WebAPI。

(2)研究動機與研究問題



圖一、傳統Web Server 圖二、WebAPI Gateway

研究動機

在使用網頁時，通常使用者 (client) 會需要去WebAPI服務端提取資料 (也就是server端)。當使用者需要多筆資料，而那些資料可能是由不同的服務端所提供的，那他形成的架構就會像(圖一)。但這架構會造成一個大問題，每個服務端都必須要有一個公用的IP地址，也就表示所有人都可以直接連到服務端。尤其在IoT的佈建當中，許多的感測器是由外包廠商負責建置，再透過公開的4G網路傳回伺服器。IoT設備由4G取得的是浮動的IP位址，因此實務上伺服端都必須擁有固定的公開IP位址，以讓IoT設備傳回資料。如果有心人士想竊取伺服器的資料，或是想讓伺服器癱瘓，那這種架構極有可能會讓有心人士達成他們的目的。

於是後來有了Gateway (圖二) 這架構。Gateway 扮演了使用者連到服務端間的中繼站，使用者需要的資料都間接透過Gateway取得，服務端要給使用者的資料也都透過Gateway傳送。公用的 IP地址只需要Gateway這台電腦擁有，真正在服務的服務端可以只使用私用IP地址。這解決了剛剛所提到的問題，服務端不再需要暴露在所有人都可以直接連線的狀況。除了上述的好處之外，Gateway還能讓服務端的管理者清楚知道當前有哪些電腦正在對外提供服務，並把沒在提供服務的電腦給關機，減少了被入侵的機會。

研究問題

雖然上述的架構解決了把服務端暴露在公用IP的環境，但相對地，也讓風險全部都在Gateway上，只要Gateway停止運作，那麼使用者也沒辦法正常使用我們的服務。為了解決此問題，此研究將使用Gateway與Kubernetes搭配，讓停止運作的Gateway再次重啟。

現在開源的Gateway也有很多，此研究也將對各種Gateway進行效能上的分析與比較。

(3)文獻回顧與探討

**Express Gateway**

Express是一個開源的項目，使用Node.js來編寫程式。它提供了動態且集中式的管理系統，還提供了使用者和憑證管理 [1]。

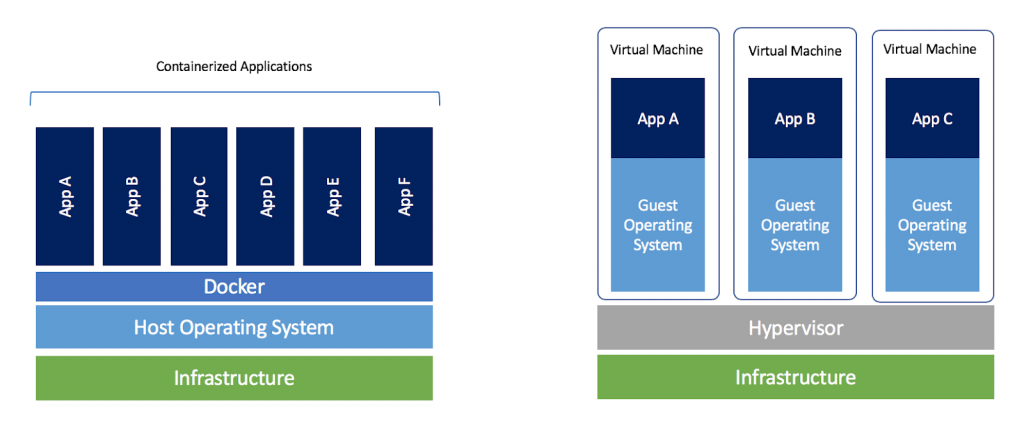
**API Umbrella**

API Umbrella是一個開源的軟體，使用ruby來編寫程式。他讓API創建者和使用者更容易去使用。創建者可以專心在開發API上，而其他的管理(像是限速、訪問控制)則交給API Umbrella。而使用者可以只透過單一端點訪問到服務端所有的API，且標準化了訪問的流程 [2]。

**Kong API Gateway**

Kong API Gateway使用lua編寫程式，其軟體又分為開源版本和企業版本。在本節所提及的三個Gateway中，它最為社群所信賴 。Kong提供了大量的插件，包含了使用限速、用戶端的認證，以及客製化的log形式。建立者還可以透過lua來編寫log所要輸出的結果[3]。

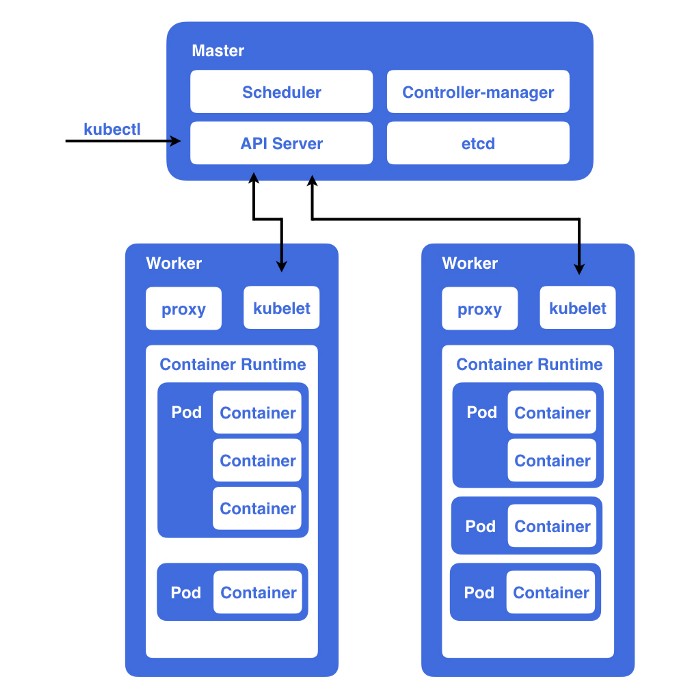
**Docker**



圖三、Docker vs. VM [4]

Docker使用GO語言開發，為開源軟體。Docker的基本單位為container，其原理與虛擬機（VM）類似，但卻比虛擬機更為輕巧，更能夠快速地建立起來。傳統的虛擬機著重在將硬體設備給虛擬化，而container則著重在將作業系統給虛擬化。如(圖三)所示，左邊為container的架構，右邊則為傳統的虛擬機。我們可以看到，右邊的虛擬機可以把一台電腦的硬體支援分配給多台虛擬機，但每台虛擬機都需要有作業系統在上面執行。左邊的container一樣是把硬體資源分配給container，但值得注意的是，每個container並不需要再架起一個作業系統，這也是為甚麼container比起虛擬機更為輕量，更能夠快速建立起來的原因 [5]。

**Kubernetes**



圖四、Kubernetes架構 [6]

Kubernetes又稱為K8s，使用GO語言開發，且為開源軟體。由Google設計出來，現在已屬於Cloud Native Computing Foundation。K8s中，一個基本單位為Pod，Pod裡面可以有一個或多個container，但通常只會有一個。在運行K8s時，有兩個主要腳色，如(圖四)所示，為Master和Worker，Master和一個或多個Worker組合起來就稱為Cluster。

使用者會在Master端下達指令，告訴K8s我們要架起服務的需求共要幾個Pod。Master則會根據Worker的狀態，來決此Worker 要分配幾個Pod。K8s中還有一個物件為Deployment，他的工作是保持Pod 的數量，當Pod若因意外故障而少於預期的數量，Deployment則會啟動新的Pod，讓Pod維持在一定的數量[6][7][8]。

(4)研究方法及步驟

在建立完K8s Cluster且確認Master與Worker能正常運作後，把Gateway透過Master來佈署到我們的Cluster上。

此步驟將會嘗試多的開源的API Gateway，包含Express Gateway、API Umbrella、Kong API Gateway。

在佈署完成後，針對不同的API Gateway，確認他們能否照著需求去開創。

(5)預期結果

(一) 此計畫將會搭建出API Gateway，並讓使用者的連線都須透過此Gateway。

(二) 使用Gateway與K8s做搭配，並使用多台電腦，做到動態的負載，且最大限度地提升各電腦的使用效率。

(三) 此計畫將測量各開源的Gateway，比較他們之間在電腦效能上的使用、對於大流量的表現。

(6)參考文獻

|  |  |
| --- | --- |
| [1] | "github," ExpressGateway, [Online]. Available: https://github.com/ExpressGateway/express-gateway. |
| [2] | "github," National Renewable Energy Laboratory, [Online]. Available: https://github.com/NREL/api-umbrella. |
| [3] | "github," Kong, [Online]. Available: https://github.com/Kong/kong. |
| [4] | alfredkosasihh, "ithelp," October 2020. [Online]. Available: https://ithelp.ithome.com.tw/articles/10250841. |
| [5] | "wiki," 12 10 2013. [Online]. Available: https://en.wikipedia.org/wiki/Docker. |
| [6] | C.-W. Hu, "kubernetes-basic-concept-tutorial," July 2019. [Online]. Available: https://cwhu.medium.com/kubernetes-basic-concept-tutorial-e033e3504ec0. |
| [7] | "Kubernetes", June 2014. [Online]. Available: https://en.wikipedia.org/wiki/Kubernetes. |
| [8] | M. Lukša, "Kubernetes in Action”, 2nd Edition, Manning, February 2020. [Online]. Available: https://www.manning.com/books/kubernetes-in-action-second-edition. |

(7)需要指導教授指導內容

吳坤熹老師是本校網路通訊的專家，在IPv4與IPv6的轉移機制提供了許多的貢獻及論文。吳老師長年與埔里基督教醫院進行多年的合作計畫，協助醫院進行網路安全的架構提升，以及API Gateway的建置與容錯機制設計，對於醫院在實務上的網路需求有足夠的解析。此計畫，教授將提供實驗的背景理論以及實驗器材。

(如篇幅不足，另紙繕付)