iKala Cloud

Google Cloud Platform
成本優化與最佳實踐白皮書

iKala Cloud

目錄

iKala Cloud

前言

企業的 Google Cloud Platform（以下簡稱 GCP）雲端成本優化是個多面向且需持續不斷精進的一套流程。GCP 的服務眾多，以 Google Compute Engine 為首的 IaaS 與不同代管程度的 PaaS、SaaS 和 FaaS (Function as a Service) 等等，不同的服務各自有不同的成本優化與最佳實踐方式。

一般而言，雲端成本優化首先要瞭解您的**成本結構與資源使用**狀況，並且需做到**可視化**，以便隨時掌控現在與近期的成本與資源使用。有這些可視化資訊，就可以往後進行**基礎成本優化**，這個階段通常不會對架構有調整，主要的精神是找出沒在使用的資源來減少浪費，並且使用優惠方案等等。基礎成本優化完成後就可以進行**基礎架構現代化**，主要目的是使用成本更合算的代管服務或者用新技術讓資源使用率更高來減少所需資源。最後就是會**採用混合雲的架構**，讓資源放置在成本最合算的位置。以上方法在後面都會有更詳盡的描述。

iKala Cloud

一、成本結構與資源使用可視化

成本優化前要先瞭解成本是怎麼發生的，在 GCP 裡面可以透過內建的帳單報表[1] 來瞭解是哪個專案、哪個產品服務、或者是哪個 SKU 是費用主要的來源。如果想要做較顯著的成本優化，通常都是挑最大的成本來源來節省，舉例來說常見的成本主要來自：

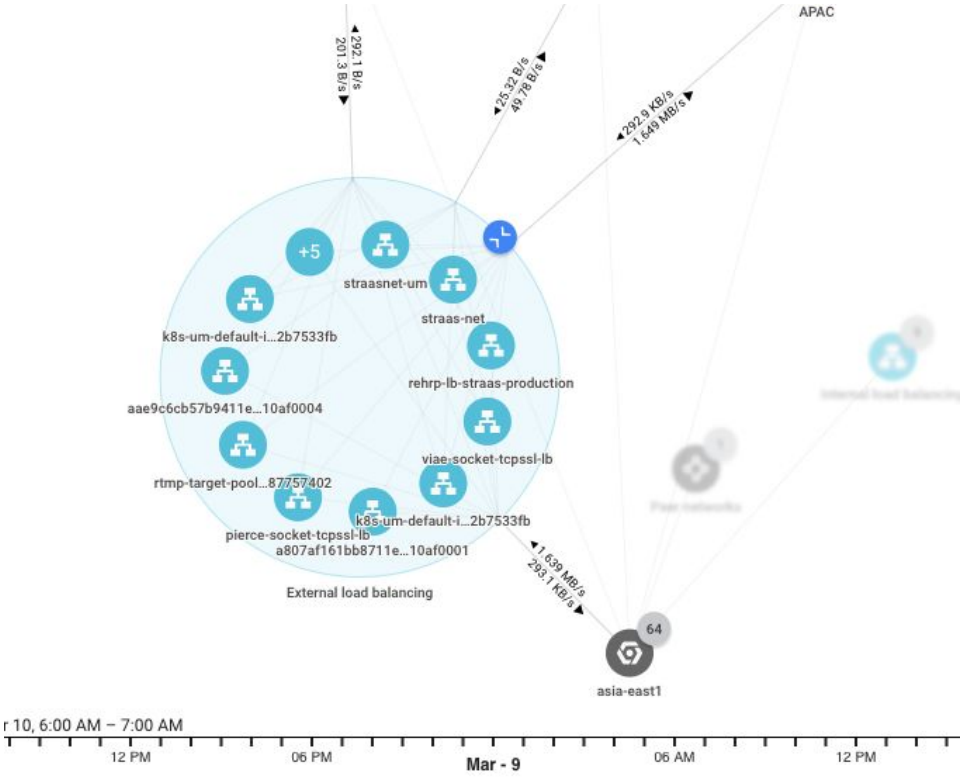
1. Windows License Fee
2. Compute Engine Virtual Machine (vCPU/RAM/Disk)
3. Network Egress

這在後面兩個章節就會討論如何優化，除了內建帳務報表 (billing dashboard) 外，也可以將帳單資料匯出至 BigQuery[2] 裡面搭配 Google Data Studio[3] 等視覺化工具來客製更符合企業內部需求的報表，讓 IT 單位能更快更容易瞭解成本與隨時審查成本。

除了 billing，另外要視覺化的是使用率。同樣 1 顆 vCPU，使用率 90% 和 5% 可優化的空間就有很大的差異。在 GCP 內可以使用 Cloud Monitoring (Stackdriver)[4] 搭建各種資源或群組的使用率 dashboard，除了內建的 vCPU 使用率監控項目外，網路流量和各種代管服務的使用量也是很常見的監控項目。Virtual Machine 建議內部都要安裝 Cloud Monitoring agent[5] 來收集例如 memory 或者是 disk 的使用率。網路流量方面，還可以使用 GCP 提供的 Network Intelligence Center[6] 以連線拓樸與時間軸兩個面向來分析網路用量流向與流量，進而找出異常或不合理的流量。

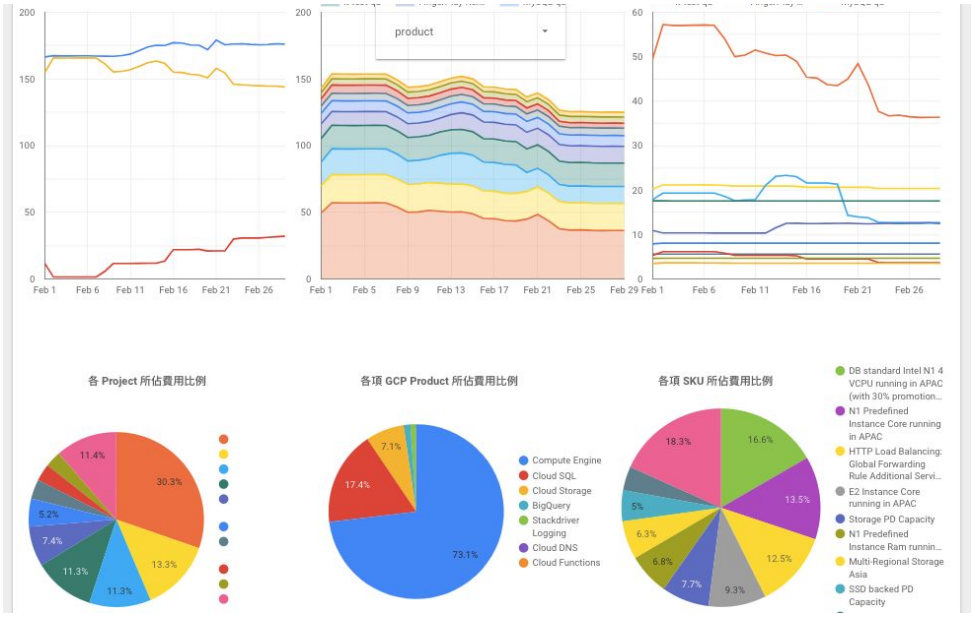
---

[1] Google Cloud Platform 帳單
[2] Export cloud billing data to BigQuery
[3] Google Data Studio
[4] Cloud Monitoring (Stackdriver)
[5] Installing the Cloud Monitoring agent
[6] Network Intelligence Center



Network Intelligence 範例

有了可視化之後，建立針對帳單或者是使用率的快訊或告警系統也是很重要的；大多雲端服務都是以使用量來計費的，像是網路流出量。如發生誤用或者意外遭入侵，造成的費用往往是很驚人的，因此打造告警、及早發現異常的費用或者異常的使用項目是必須的。告警可以使用 GCP 內建的 Cloud billing budget alerts 或者是 iKala Cloud 為客戶設定的 billing alert，如果 billing 有匯出到 BigQuery 也可以分析 BigQuery 的資料來打造快訊或告警。



Data Studio dashboard 範例

二、基礎成本優化

將雲端使用量可視化後，便可著手進行基礎成本的優化。這個階段通常不需大幅度調整架構，可以直接透過**降低規格**或者**找出非必要資源**，然後再搭配 **BYOL（Bring Your Own License，自有授權）與承諾使用**等方法來降低成本。

降低規格

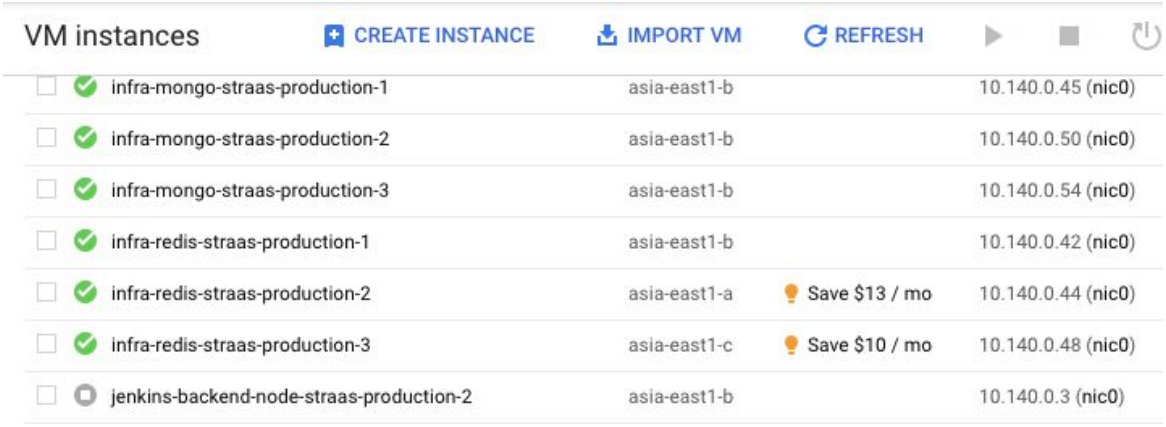
降低規格可以再細分為幾個方式：

1. Google Compute Engine Sizing Recommender
2. 自行降低規格
3. 使用 N2/E2/C2/M2 類型 CPU
4. 使用先佔型的 Virtual Machine
5. 使用自動擴展

以下，將針對 5 種方式分別進行說明。

Google Compute Engine Sizing Recommender

如果整體的 CPU 使用率不高，在機器類型方面就會有一些優化的空間。Google Compute Engine 的操作介面列表中，會有 Rightsizing 的建議，可以根據建議對 Virtual Machine 規格進行調整。



Sizing Recommender 範例

自行降低規格

另外，因為已建立了使用量可視化，也可以自行根據使用量進行調整規格，尤其是 RAM 與 Disk utilization 這兩項 Sizing Recommender 沒有提供建議的。像是高單價的 extended memory 如從用量觀察沒有在使用，又或 disk IOPS 較低，可以從 SSD 換成一般硬碟與降低大小。

使用 N2/C2/M2/E2 類型 CPU

GCP 現在多了新的機器類型 (N2/C2/M2/E2)，這些機器類型可為各種工作負載提供最佳性價比。以新型的 E2 類型來說，擁有和過往的 N1 類型同等的效能，但比 N1 費率最多便宜 31%，是目前 GCP 最具價格優勢的機型。又以 C2 類型舉例來說，C2 機器類型提供更多的運算效能、在更新的平台上執行，適用於 CPU-bound 的情境。對於使用情境的說明，可以參考 iKala Cloud 技術部落格的文章[7]。

[7] GCP 四大 VM 類型 (N2/E2/C2/M2) 應用場景及價格介紹

使用先佔型的 Virtual Machine

先佔型 Virtual Machine 是 GCP 將多餘的運算資源給客戶使用，價格可能比一般執行便宜多達 80%，但須注意的是，GCP 隨時有可能把這份資源給收回去。因此對於像是轉檔或跑數據等隨時可以中斷再重啟的工作，就很適合用先佔型 Virtual Machine 來降低成本。

使用自動擴展

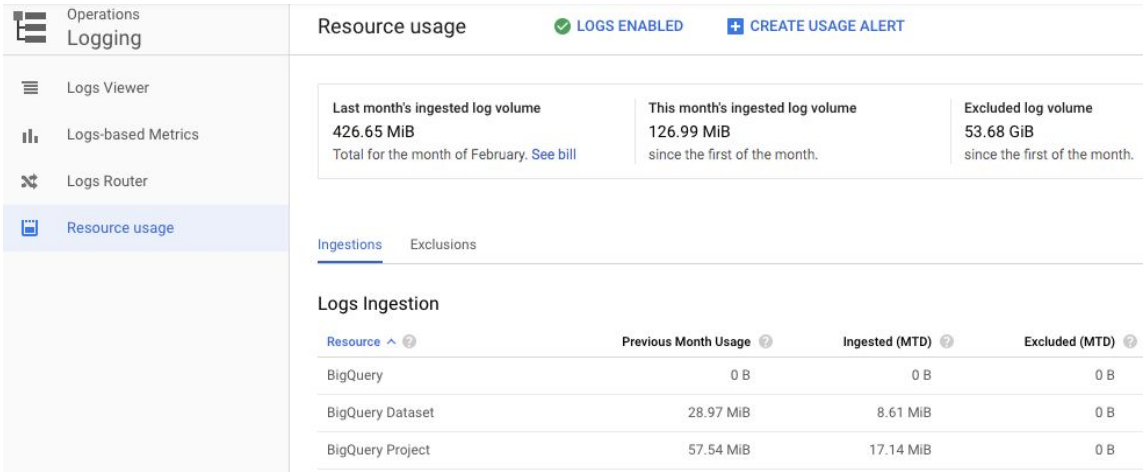
許多工作負載所需要的資源不會一整天都維持一樣，例如：承受流量的 HTTP Server 或一些運算用的 worker。此時，就能使用 GCP 的自動擴展[8] 功能，比如透過 CPU 的使用率或者是使用者連線數等參數去自動增減所需的 Virtual Machine 數量，高峰時迅速開出 Virtual Machine 來承擔流量，離峰的時候透過減少 Virtual Machine 來節約成本。這樣比起為了高峰而固定開出許多 Virtual Machine 可節省很多。

找出非必要資源

許多非必要資源可以透過可視化 billing 報表分析，找出一些費用較高的 SKU 再來評估此成本是否有需要。比較常見的非必要資源如：Cloud Logging、External IP、Network Egress 等項目。

Cloud Logging 是 GCP 全代管的 Log 集中服務，除了 Virtual Machine 的 Log 可以導到 Cloud Logging 之外，代管服務的 Log 也會直接進入 Cloud Logging。當服務一多或者是用量大時，Cloud Logging 的儲存費用自然會水漲船高，這時就可以使用 Cloud Logging usage dashboard[9] 來分析用量的來源定設定告警，不需要的 Log 可以設定排除加上轉存到像是 Cloud Storage 或者是 BigQuery 等較便宜的儲存空間備查。

[8] Autoscaling groups of instances
[9] Viewing usage reports



Cloud Logging resource usage dashboard 範例

從 2020 七月開始，GCP 就會對 Virtual Machine 使用的 External IP 開始收費，許多沒有對外溝通需求的 Virtual Machine 就可以拔除 External IP 來節省部份成本。沒有 External IP 仍然可以透過設定 Private Google Access[10] 來存取大部份的代管服務，短暫要連入如 SSH 或 RDP 也可以使用 IAP TCP Tunneling[11] 等機制。

從 GCP 網路費用列表[12] 來看收取流出 (Egress) 的部份，內網如果有跨 Zone 或跨 Region 會有相對應的費用，流到外網會根據來源與目的有不同的單價。之前在可視化章節提到，使用者可以透過 Network Intelligence 或者是 Stackdriver 來瞭解網路的流向，然後找出可以節省或不正常的流量。如果像是有跨 GCP 專案之間不小的流量，這類跨專案的流量就可以透過 VPC peering[13] 或者是 Shared VPC[14] 等串接 VPC 的方式讓流量無需流出 GCP 進而節省網路 egress 費用。

[10] Configuring Private Google Access
[11] Using IAP for TCP forwarding
[12] 網路費用列表
[13] VPC Network Peering overview
[14] Shared VPC overview

BYOL 與承諾使用

在 GCP 一般的 Virtual Machine 上雖然 Windows 作業系統無法採用 BYOL (Bring Your Own License)，但是 SQL Server 系列有機會可以使用。如果企業已採購大量 SQL Server License，就能夠使用 GCP BYOL 來節省成本。

如果專案本身有穩定長期的 Virtual Machine 使用量，這個時後可以考慮購買 Virtual Machine Committed use discounts[15]，可以購買一年或三年，最多可以節省 57% 的費用。但因購買後無法取消，因此配合可視化很重要，使用者須觀察長期的使用量，抓出最小使用需求量。另外，建議購買前先將用量打折，先購買 60% 至 80% 的用量，避免有些許縮減用量的需求，如果用量確定增加隨時可以加購。如果整體使用量夠高，也可以和 GCP sales 商議整體的承諾使用折扣，這個折扣就能夠套用到 Billing Account 下的所有專案。

[15] Committed use discounts

三、基礎架構現代化

當基礎成本優化完成，使用者可以朝著基礎架構現代化的方向前進；透過**使用代管服務、無伺服器架構**與**容器化**的方式來降低成本。舉個例子來說，不常用的大量檔案可以放置到 Google Cloud Storage[16] 物件儲存空間，這比起直接使用 persistent disk 儲存可能會更節省成本。

使用代管服務

使用代管服務最重要的是，前面章節提到的**可視化**一定要先完成，因為代管服務大多是根據用量收費，需要可視化來避免一些誤用而造成高額費用的狀況。另一個重點是要瞭解收費方式，並作**成本試算**。以上述 Google Cloud Storage 來說，主要費用會在儲存與網路兩方面，雖然儲存費用較低，但網路則可能根據使用情境產生較多的費用。每個 GCP 服務都有明確的收費計算說明，可以參考說明或者是計算機 (Google Cloud Pricing Calculator)[17] 進行試算。

代管服務的另一項優勢，是能節省維運成本；基本上，代管服務不用考量如：搭建 High Availability、備份、Log 集中、監控、安全防護和除錯等各種維運問題。在進行成本優化時，也可以將節省的人力成本計算進去。

使用無伺服器 (Serverless) 架構

無伺服器架構目的是讓開發者專注在應用上，而無需擔心底層伺服器的種種問題，且只有在執行的時候才會產生費用。因此，如果是一些較少量的應用情境，如偶爾執行的 cron job 或者是少量的 RESTful API 使用情境，都可以使用如 Cloud Functions[18] 來進行開發。這樣比起持續開一台 Virtual Machine，不論是資源成本或人力成本都可能更加節省。這個部份可以參考 Cloud

[16] Google Cloud Storage
[17] Google Cloud Pricing Calculator
[18] Cloud Functions

Functions[19]、Cloud Run[20] 或 Google App Engine[21] 的價格與 Virtual Machine 價格進行試算比較。

容器化

如果 Windows License Fee 和 Virtual Machine 占了很大的比例，如前一章提到根據使用量降低規格雖然可行，但規格上還是得照尖峰時間需求設定，因此以長期來看還是會有許多閒置資源。一個專案的應用程式很多，但很多時候應用之間的尖峰時間並不相同，這就可以將應用容器化[22]。容器化是應用級別的虛擬化，允許在同一台 Virtual Machine 上有多個獨立的用戶空間，彼此之間不會干擾，因此可以把多個應用程式塞在同一台 Virtual Machine 裡面，讓整體使用率更高。現在 Windows 也支援容器化，GCP 上面的 Windows License Fee 是和使用的 vCPU 數成正比，因此提昇整體使用率來降低 Virtual Machine 使用就能同時降低 Windows License Fee 和 Virtual Machine 費用。

容器化再進一步可以使用 Google Kubernetes Engine (GKE)[23]，是用於自動部署、擴展和管理的容器化全代管叢集。叢集通常會由很多 Virtual Machine 所構成，GKE 對於資源的管理是以整個叢集來作考量，比起各個 Virtual Machine 自行跑容器，可以再提高資源使用率。如果再搭配叢集的自動擴展與應用程式的自動擴展 (Horizontal Pod Autoscaler, HPA)，讓資源的使用可以達到最佳化。

另外，如果 Windows License Fee 也偏高，將 windows-based 的應用程式或服務轉換成為以 linux-based 與 open source solution 也是節省費用的方式。當然，這邊仍需考量轉換的人力成本，以及採用 open source 的維運支援等綜合因素。

[19] Cloud Functions 定價
[20] Cloud Run 定價
[21] Google App Engine 定價
[22] 容器化：應用級別的虛擬化
[23] Google Kubernetes Engine

四、使用混合雲架構

在企業中雖然採用了公有雲，但通常還是會有自建機房或者是租用機房等地端機房 (on-prem) 的情況。有時候這些機房會有一些閒置機器的產生，而地端和雲端可以互相搭配，在透過 Cloud VPN[24] 來串接雲地兩邊的內網環境之後，使用者就能選擇將部分應用程式佈署在地端機房，以平衡雲端服務與地端機房的費用。

適合放在地端的是比較少在異動的 Windows Server，可以同時節約 License Fee 和 Virtual Machine 使用量；適合放在雲端的，則是要自動擴展或者是需要較大頻寬的應用程式。如同上一章所建議的使用代管服務，GCP 的代管服務大多能夠直接被地端機房存取，而連線都有加密，符合資訊安全規範。如有需要，GCP 也有資安政策[25] 可以限制其存取的方式與範圍。

[24] Cloud VPN overview
[25] Introduction to the Organization Policy Service