

O'REILLY®



Google Compute Engine

安全與可擴展的雲端運算

Marc Cohen & Kathryn Hurley & Paul Newson 著
陳佳新 譯

Google Compute Engine 安全與可擴展的雲端運算

Google Compute Engine

Marc Cohen, Kathryn Hurley, and Paul Newson 著

陳佳新 譯

Google Compute Engine | 安全與可擴展的雲端運算

作者：Marc Cohen / Kathryn Hurley / Paul Newson

譯者：陳佳新

企劃編輯：莊吳行世

文字編輯：王雅雯

設計裝幀：陶相騰

發行人：廖文良

發行所：碁峰資訊股份有限公司

地址：台北市南港區三重路 66 號 7 樓之 6

電話：(02)2788-2408

傳真：(02)8192-4433

網站：www.gotop.com.tw

書號：A432

版次：2015 年 08 月初版

建議售價：NT\$450

商標聲明：本書所引用之國內外公司各商標、商品名稱、網站畫面，其權利分屬合法註冊公司所有，絕無侵權之意，特此聲明。

版權聲明：本著作物內容僅授權合法持有本書之讀者學習所用，非經本書作者或碁峰資訊股份有限公司正式授權，不得以任何形式複製、抄襲、轉載或透過網路散佈其內容。

版權所有 ● 翻印必究

國家圖書館出版品預行編目資料

Google Compute Engine:安全與可擴展的雲端運算 / Marc Cohen, Kathryn Hurley, Paul Newson 原著；陳佳新譯. -- 初版. -- 臺北市：碁峰資訊, 2015.08

面；公分

譯自：Google Compute Engine

ISBN 978-986-347-718-1(平裝)

1.雲端運算

312.136

104012406

讀者服務

- 感謝您購買碁峰圖書，如果您對本書的內容或表達上有不清楚的地方或其他建議，請至碁峰網站：「聯絡我們」\「圖書問題」留下您所購買之書籍及問題。(請註明購買書籍之書號及書名，以及問題頁數，以便能儘快為您處理)

<http://www.gotop.com.tw>

- 售後服務僅限書籍本身內容，若是軟、硬體問題，請您直接與軟體廠商聯絡。
- 若於購買書籍後發現有破損、缺頁、裝訂錯誤之問題，請直接將書寄回更換，並註明您的姓名、連絡電話及地址，將有專人與您連絡補寄商品。

- 歡迎至碁峰購物網

<http://shopping.gotop.com.tw>
選購所需產品。

© 2015 GOTOP Information, Inc. Authorized Chinese Complex translation of the English edition of Google Compute Engine, ISBN 9781449360887 © 2015 Marc Cohen, Kathryn Hurley and Paul Newson. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

前言

今日大多數的高階運算技術都是存在於包含了大量低成本伺服器陣列的大型建築物裡。大量的運算資源被放置在安全的區域中，由訓練有素的技術人員所維護。圖 P-1 看到的是 Google 位於美國愛荷華州 Council Bluffs 面積 115,000 平方英尺的資料中心，降低了 Google 資料中心網絡之間的互動延遲（請見 <http://www.google.com/about/datacenters/gallery>）。

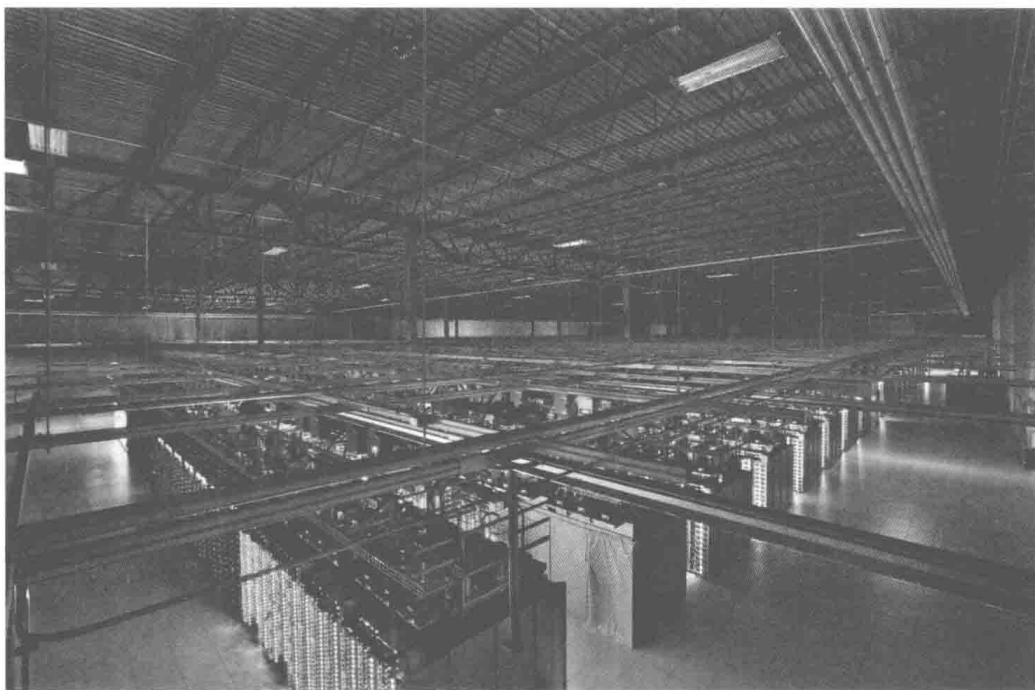


圖 P-1 Google 資料中心（由 Google/Connie Zhou 拍攝）

在大型主機時代，假使你的成長超過了一台電腦所能夠負荷的，那麼你就需要在花幾百萬美元再買一台電腦。現代的資料中心透過跨越大量的商用主機來配置任務，進而實現了資源的可擴展性。在資料中心時代，你可以依照需求配置足夠的廉價伺服器，然後在完成工作時歸還那些資源。

一直到最近，都還是只有少數幸運的工程師得以在新世代的科技公司中使用資料中心資源。但是在過去這些年來，革命已經悄悄發生。如同先前的電腦硬體革命一樣，人們可以取用更大量更小型的電腦，雲端運算允許更大範圍的存取，透過公開的網際網路，從大規模的電腦叢集，到目前最先進的資料中心。一如過去，提昇的存取性是非常大的創新。

在 Google 簡短的發展歷史中，在建構和管理雲端運算服務的領域裡，它一直都是許多技巧和最佳實踐的領導者。從搜尋到 Gmail 到 YouTube 再到 Google Maps，其服務每天為數百萬名用戶提供了安全、可擴展、可靠的雲端運算，每天應付數億次查詢。現在，透過使用 Google Compute Engine 這套基礎建設，任何人都將可以使用那些服務。

Compute Engine 擁有諸多優點：定期且自動升級的先進硬體、隨著企業需求而增減的虛擬無限容量、彈性的計費模型、一群專家負責維護運算和網路資源，以及基於安全和效能考量，能夠將你的資源佈署在全球網路。

本書是 Google Compute Engine 的使用指南，重點放在解決一些實際可能遭遇的問題。從高階角度來看，Google Compute Engine 讓你能夠存取全世界最新進的資料中心網絡——而那些運算資源是由 Google 所提供的。幾乎可以這麼說，這意味著將會提供 API、命令列工具和網頁使用者介面來使用 Google 的運算和網路資源。

在後續的章節中，我們將會詳細介紹產品的功能，搭配一些關於使用 Google Compute Engine 的最佳實踐。我們將會提供許多範例，來描述如何運用 Google 提供的方法來存取 Compute Engine 的服務。儘管本書當中的程式範例皆以 Python 和 JavaScript 撰寫而成，但是底層的 RESTful API 及其概念則是和程式語言無關的。

本書結構

圖 P-2 展示的是將所有的 Compute Engine 元件放在一起時的情形。從高階角度來看，Compute Engine 執行個體（Instance）、網路和儲存都隸屬於 Compute Engine 專案本身。Compute Engine 專案基本上是一個關於你的應用程式的具名資訊集合，並且作為你的 Compute Engine 資源的容器。任何你所建立的 Compute Engine 資源（例如執行個體、網路和磁碟）都會與你的 Compute Engine 專案關聯並且包含在其中。

API 提供了透過程式設計來與 Compute Engine 元件和資源進行互動的方式。

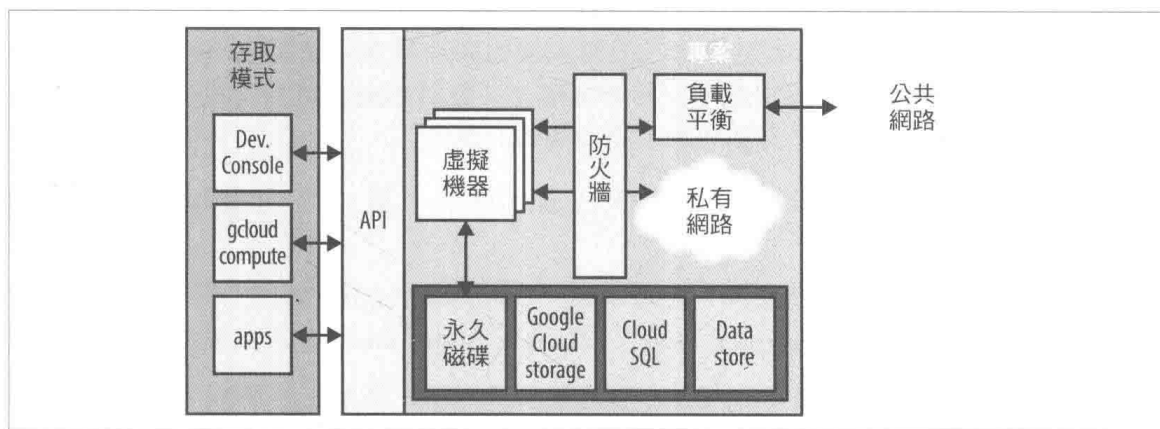


圖 P-2 Google Compute Engine 的元件總覽

我們將會探索下列的 Google Compute Engine 元件和資源：

- 專案、存取模式和 API（第 1 章）
- 虛擬機器（第 2 章和第 7 章）
- 永久磁碟（第 3 章）
- Google Cloud Storage（第 4 章）
- Cloud SQL 和 Cloud Datastore（第 5 章）
- 防火牆、負載平衡和私有網路（第 6 章）
- 完整的應用程式（第 8 章）

章節內容

第 1 章：開始使用

我們將會看一下如何開始使用 Compute Engine。我們將從建立一個使用了 Google Developers Console 網頁介面的 Compute Engine 專案開始。接著會探索各種存取和管理 Compute Engine 資源的方法，包括透過 Developers Console、透過 gcloud compute 這套命令列介面，以及透過 RESTful 介面形式的 Compute Engine API。

第 2 章：執行個體

執行個體是可自訂的 Linux 機器，代表 Google Compute Engine 的核心。你擁有 root 權限可以存取任何你所建立的執行個體，讓你可以下載及安裝套件，並且自訂磁碟、硬體，或是其他的設定選項。第 2 章涵蓋了使用 Compute Engine 執行個體來完成工作的基礎知識，並且詳細解釋了執行個體的屬性。

第 3 章：儲存永久磁碟；第 4 章：儲存 Cloud Storage；第 5 章：儲存 Cloud SQL 與 Cloud Datastore

大部分的應用程式都需要空間來存放資料。Compute Engine 提供了許多種形式的儲存，包括永久磁碟（Persistent Disk）、Google Cloud Storage、Cloud SQL 和 Cloud Datastore。

如同其名稱所揭示的，永久磁碟所存放的資料不會因為執行個體失效而喪失。Cloud Storage 可以讓你在 Google 的基礎建設中存放、存取和管理任何規模的物件。Cloud Storage 提供很棒的選擇，可以存放 highly durable、high availability data storage。透過各種機制和工具（包括 Developers Console、gsutil 命令，以及 Cloud Storage API），你的資料可以在 Compute Engine 的範圍之內和之外被存取得到。

Cloud SQL 提供了雲端上由 Google 管理的 MySQL 服務，而 Cloud Datastore 則提供了 Google 等級的 NoSQL 資料服務。這 2 種服務都可以在 Compute Engine 的範圍之內和之外使用。

第 3 章、第 4 章和第 5 章詳細介紹了這些儲存方案，並提供豐富的範例來說明如何運用所有支援的存取方法。

第 6 章：網路

每個專案搭配整合式 DNS 服務、編碼規劃和路由邏輯，都擁有自己的私有網路。一個專案可以擁有多個網路，每個網路也可以包含多個執行個體。可以將防火牆規則套用至網路，以便允許或禁止流量進入執行個體和網際網路。負載平衡提供了進階的響應式可擴展流量分散。第 6 章提供了 TCP/IP 網路的簡介，並涵蓋了 Compute Engine 的詳細進階網路功能。

第 7 章：進階主題

在第 7 章中，我們介紹了各式各樣的進階主題，提供額外的方法讓你可以自訂自己的 Compute Engine 資源。這些議題包括自訂圖片、啟動腳本，以及中繼資料（Metadata）伺服器。

第 8 章：完整的應用程式

在第 8 章中，我們展示的是應用程式的建構指南，這個應用程式匯集了好幾個本書稍早談論過的主題。透過使用範例應用程式，本章將會一步一步建構出簡單卻又不失其豐富性的雲端運算應用程式。

本書編排慣例

本書使用下列的印刷字體：

斜體字 (*Italic*)

表示檔名、副檔名、路徑、網址和電子郵件。對於初次提到或重要的詞彙，中文以楷體字呈現，其對應的英文則以斜體表示。

定寬字 (*Constant width*)

用於顯示程式碼列表，以及內文段落中參照到的程式元素，舉凡變數或函式名稱、資料庫、資料型別、環境變數、陳述式、關鍵字。

定寬粗體字 (*Constant width bold*)

顯示命令或其他應該由使用者逐字輸入的文字。

定寬斜體字 (*Constant width italic*)

顯示應該以使用者提供的數值來取代，或者應該以由上下文所決定的數值來取代的文字。



表示提示或建議。



表示一般的注意事項。



表示警告或提醒。

使用範例程式

本書的目的為協助你完成工作。一般而言，你可以在自己的程式或文件中使用本書的範例程式碼，除非重製了程式碼中的重要部分，否則無須聯絡我們。例如，為了撰寫程式而使用了本書中的數個程式碼區塊，這樣無須取得授權，但是將書中的範例製作成光碟並銷售或散佈，則需要取得授權。此外，在回覆問題時引用了本書的內容或程式碼，同樣無須取得授權，但是把書中大量範例程式放到你自己的產品文件中，就必須要取得授權。

雖然沒有強制要求，但如果你在引用時能標明出處，我們會非常感激。出處一般包含書名、作者、出版社和 ISBN。例如：「*Google Compute Engine* by Marc Cohen, Kathryn Hurley and Paul Newson (O'Reilly). Copyright 2015 Marc Cohen, Kathryn Hurley and Paul Newson, 978-1-449-36088-7.」

假如你不確定自己使用範例程式的程度是否會導致侵權，歡迎隨時聯絡我們：
permissions@oreilly.com。

致謝

Marc、Kathryn 和 Paul 要感謝他們的專業編輯 Rachel Roumeliotis，感謝她在本書專案期間的耐心和指導。此外，他們還要感謝諸多協助的審閱者和同事提供的許多寶貴的校正回饋：Andrew Kadatch、Ankur Parikh、Benson Kalahar、Dan Miller、Danielle Aronstam、Dave Barth、Elizabeth Markman、Eric Johnson、Greg DeMichillie、Ian Barber、Jay Judkowitz、Joe Beda、Joe Faith、Johan Euphrosine、Jonathan Burns、Julia Ferraioli、Laurence Moroney、Martin Maly、Mike Trinh、Nathan Herring、Nathan Parker、Phun Lang、Rebecca Ward、Renny Hwang、Scott Van Woudenberg、Simon Newton、Sunil James。

Marc 要感謝他那些專業、有才華、全力支援的共同作者群。此外 Marc 也要感謝家人的愛與支持，包括太太 Kimberly 和女兒 Maya，感謝他們忍受他花了許多時間在本書的撰寫上。

Kathryn 要感謝她的老公 James 和女兒 Violet 的愛與支持，以及優秀的共同作者們 Marc 和 Paul。

Paul 要感謝 Marc 和 Kathryn 提供本書專案的合作機會，雖然他是後來才加入的。

譯者序

從使用者的角度來看，網路和雲端早已融入我們的日常生活當中。舉例而言，許多人已經很習慣將檔案和照片存放在 Google Drive 或 Dropbox，並且與其他人分享。而從服務提供者的角度來看，雲端基礎建設提供的穩定性、擴展性和可用性，不僅使得服務的架設變得簡單，同時也確保了服務的品質，為服務提供者與使用者創造了雙贏的局面。

假使你也是 Gmail 等 Google 雲端服務的重度使用者，相信將會非常樂見 Google 將同樣等級的雲端平台釋放出來供開發者付費使用。儘管 Google Compute Engine 與 Amazon Web Services 和 Microsoft Azure 三強鼎立，而且在運算、頻寬和儲存方面的收費又差不了多少，不過仍然各有其所長之處，例如管理介面的易用性以及對於手動自訂建立（透過 API 或命令列）的支援程度。因此開發者依然能夠依據本身的喜好與習慣，以及適合放置該服務的地點（例如主打歐美或亞洲）來決定如何選擇這些雲端平台業者。關於這個問題，我自己也被詢問過好多次，所以在此推薦 Quora 上的一篇問與答：<http://bit.ly/gce-azure-aws>。

綜觀本書，我個人尤其喜歡最後一章（第 8 章），內容不僅從無到有成功打造了一個完整的應用程式，同時也介紹了許多主流的技術，例如 Node.js、Docker、Web Starter Kit、AngularJS 和 D3.js 等，對於想要迅速建構網站型雲端應用程式的開發者而言，將會非常受用。

翻譯期間庶務繁忙，對於老婆 Viki 和兩個兒子 Linus 與 Morris 難免有怠慢之處，借此版面由衷感謝他們的體諒與支持。

陳佳新

jarsing@chibuapp.com

2015 年 8 月於彰化

關於作者

Marc 負責管理 Google 在倫敦的 Developer Relations Engineering 團隊，協助 EMEA 地區的軟體開發人員取用 Google API 和服務。在這之前，Marc 在 Bell Labs 和 Lucent Technologies 幫忙設計和建構通訊系統。在不工作的閒暇時刻，他喜歡欣賞獨立音樂和電影、寫作、教書，以及下棋。

Kathryn Hurley 是 Google Compute Engine 的開發人員計畫工程師。在職期間，她負責開發範例應用程式，藉此協助開發人員學習如何使用 Compute Engine API。她擁有舊金山大學網頁科學碩士，以及加州大學戴維斯分校學士學位。先前的工作經驗包括行動研究，以及 P2P 運算。

Paul Newson 是 Google 的軟體工程師。目前他關注的重點是幫助開發人員運用 Google 雲端平台的威力來解決他們的大數據難題。在這份工作之前，Paul 是在 Google 雲端儲存擔任工程師，負責建構 Google 的雲端平台。在加入 Google 之前，Paul 共同創辦了一家小型的遊戲技術新創公司，後來賣給了 Microsoft，隨後他歷任 DirectX、Xbox、Xbox Live 和 Forza Motorsport 等方面的職務。之後他也花了一些時間在 Microsoft Research 研究有趣的機器學習領域。工作之餘，Paul 喜歡攀岩、摩托車，以及其他需要非常關心的活動。

出版記事

封面動物是棕腹樹鵲 (*Rufous Treepie*)。這種鳥類分佈於印度和其他東南亞地區，包括泰國、寮國和巴基斯坦。牠們的主食是果實、種子和昆蟲，也會偷吃其他鳥類的蛋和幼鳥。

雄鳥與雌鳥的重量通常都在 90-130 公克之間。牠們的頭部是深色的，看起來幾乎全黑，但是身體則是鮮豔的橘色。眼睛是深紅色。

棕腹樹鵲是一種喜歡發出叫聲的鳥類。各地人士幫這些叫聲取了一些名字，例如 bob-o-link 或 ko-tree。牠們並不怕生，如果附近有人類的話，也會從陌生人身上搶奪食物。只要一有機會，牠們就會叼走食物。

O'Reilly 圖書的許多封面動物都瀕臨絕種；牠們都是重要的世界資產。如欲盡一份心力，請造訪 animals.oreilly.com。

封面圖片選自《Wood's Animate Creation》。

目錄

前言	vii	
譯者序	xiii	
第一章	開始使用	1
	建立 Compute Engine 專案.....	2
	啟動計費	3
	增加團隊成員.....	3
	Compute Engine 資源.....	4
	管理 Compute Engine 資源.....	5
	結語	25
	接下來.....	25
第二章	執行個體.....	27
	使用 Developers Console 建立執行個體.....	28
	使用 Developers Console 存取執行個體.....	32
	使用 Developers Console 刪除執行個體.....	33
	使用 gcloud 建立執行個體	34
	執行個體的屬性	36
	使用 gcloud 存取執行個體	42
	使用 gcloud 刪除執行個體	46
	透過程式建立執行個體.....	47
	使用服務帳戶建立執行個體.....	51
	選擇存取模式.....	52
	清理	53
	結語	53
	接下來.....	53

第三章	儲存：永久磁碟.....	55
	Compute Engine 儲存選項概述	55
	永久磁碟	56
	永久磁碟的效能	58
	用 Developers Console 建立永久磁碟	58
	使用 gcloud 建立永久磁碟	61
	連接 / 中斷連接永久磁碟與執行中的虛擬機器	65
	利用程式建立永久磁碟	69
	永久磁碟快照	74
	結語	77
	接下來	77
第四章	儲存：Cloud Storage	79
	了解 BLOB Storage	80
	開始使用	80
	gsutil 簡介	82
	從程式碼中使用 Cloud Storage	84
	設定存取控制	87
	了解 ACL	91
	使用預設物件 ACL	92
	了解物件的不變性	96
	了解強一致性	96
	結語	97
	接下來	97
第五章	儲存：Cloud SQL 和 Cloud Datastore	99
	Cloud SQL	99
	開始使用	100
	建立資料庫與資料表	105
	執行 SQL 查詢	108
	Cloud Datastore	110
	開始使用	112
	透過 Developers Console 建立和檢視實體	112
	利用程式從虛擬機器建立和擷取實體	114
	實現自己的資料庫	118
	結語	119
	接下來	119

第六章	網路	121
	網路概論	121
	網路位址與路由	122
	傳輸控制協定 (TCP)	123
	網域名稱系統 (DNS)	124
	超文本傳輸通訊協定 (HTTP)	126
	負載平衡	127
	防火牆	128
	預設網路	128
	設定防火牆規則	132
	設定負載平衡	135
	保留外部 IP 位址	140
	設定網路組態	141
	了解網路成本	144
	了解路由	145
	選擇存取模式	146
	結語	146
	接下來	147
第七章	進階主題	149
	啟動腳本	149
	gcloud compute	149
	字串值作法	150
	本機檔案作法	151
	Cloud Storage 作法	152
	可以公開取得的作法	153
	API 作法	153
	自訂映像檔	157
	建立自訂的映像檔	157
	使用自訂的映像檔	160
	中繼資料	163
	中繼資料伺服器	163
	中繼資料項目	164
	專案中繼資料	165
	執行個體中繼資料	166
	資料格式	168
	預設 vs. 自訂	169
	專案層級的自訂中繼資料	169

	執行個體層級的自訂中繼資料.....	179
	wait_for_change 網址參數	188
	結語	188
	接下來.....	188
第八章	完整的應用程式.....	189
	應用程式概念.....	189
	需求	189
	應用程式資訊流程	193
	逐步建構我們的應用程式	193
	步驟 0：建立你自己的開發用途虛擬機器，並且複製範例應用程式..	193
	步驟 1：為你的應用程式建立 Docker 映像檔.....	194
	步驟 2：建構 UI 框架	199
	步驟 3：建構使用者介面	201
	步驟 4：實作主要伺服器	207
	步驟 5：實作次要伺服器	214
	步驟 6：加入視覺呈現功能	219
	步驟 7：上線佈署叢集.....	224
	結語	228
	索引	229

開始使用

Google Compute Engine 是一種可以在 Google 架構中提供虛擬機器（Virtual Machine, VM）操作的服務。你可以藉由使用很多項容易取得的作業系統，來打造有各種組態的虛擬機器。執行個體（Instance）的資料可以儲存和保留在永久的儲存空間中，也可以被複寫作為備用救援的資料，而且存活時間比整個虛擬機器的執行時間還要長久。網路的存取可以被設定為允許不同的虛擬機器、網際網路或是你自己的私有網路之間進行對話。

Google Compute Engine 提供很多樣的工具，這些都可以互動及管理你的 Compute Engine 執行個體及其組態；舉例來說，你可以開啟和關閉執行個體、連接磁碟的儲存空間，以及設定這些端點的網路存取方式。這些工具包括：提供了網頁 HTML 格式的使用者介面可以用來建立執行個體並設定其組態的 Google Developers Console（<http://console.developers.google.com>）；可以用作互動性的使用及簡單自動化腳本的 `gcloud compute` 命令列介面；而 Compute Engine API 和 RESTful API 則可以將你自己的程式和雲端管理應用程式進行整合。

在開始操作 Google Compute Engine 之前，首先要做的就是要在 Developers Console 中建立一個 Compute Engine 專案。Compute Engine 專案整合了你的應用程式資訊並且儲存你的 Compute Engine 資源及組態。磁碟、防火牆、網路和執行個體都息息相關，並且涵蓋在一個專案之中。團隊成員可以用指定的權限被加入到專案中，以便存取專案的 Compute Engine 資源。

在本章中，我們會告訴你如何使用 Developers Console 來建立 Compute Engine 專案，以及如何將成員加入到專案中。我們之後會用高階的角度來檢視 Compute Engine 資源。最後，我們會告訴你怎麼開始使用 Compute Engine 工具與 API。