Chapter. 1 環境概要

首先說明在本書中使用的環境與技術。
本書使用的是 **Linux** 和 **AWS**。
若你對本章有興趣，那麼可繼續詳讀內容，若你是「總之想先動手試試」的讀者，也可跳過本章，直接從下一章開始實際動手做，待稍有進度後再回頭閱讀本章。

1.1 Linux 是什麼？

Linux 和 Windows 及 Mac 一樣，是主要的 **OS（作業系統）**種類之一。若不講究的話，一般都發音為「李那克斯」。
包括一般常見的智慧型手機及路由器、網頁、商業應用程式等各式各樣的裝置，Linux 是一種應用範圍極廣的 OS。

而在詳細解釋 Linux 之前，我們必須先針對開放原始碼做一些說明。

1.1.1 OSS（Open Source Software，開放原始碼軟體）

所謂的開放原始碼軟體，是指將原始碼公開，好讓大家可使用、修改、重複利用的軟體。這類軟體主要是由社群（而非企業）所開發，不過也有企業利用此支援獲取報酬的商業形式存在。
由於是由社群開發，可供大眾自由使用並修改，故能更有效地反映需求、衍生出不同的新軟體，更快速地解決世上的許多問題亦是其主要特色。
Linux 可說就是全世界的工程師們用於解決他們自己遇到的問題、彼此共享回饋意見，並合作開發而成的作業系統。

1.1.2 Linux 的種類

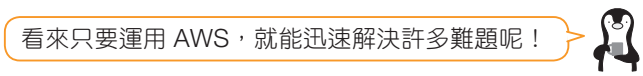
就如前述，由於是以開放原始碼的形式持續進化，因此雖然都通稱 Linux，但實際上並不只有一種，而是有好多種 Linux 存在。
這樣的 Linux 種類叫做發行版（Distribution）。
目前有 **Debian 家族**、**Redhat 家族**及 **SUSE 家族**等不同系列的發行版。
而本書主要操作的是源自 Redhat 家族的 **Amazon Linux 2**。

1.1.3 Linux 的優點

- 可依需求只選擇最基本的功能，達成最精簡的系統建構。
- 可透過指令操作，易於自動化處理。
- 有些完全不需要軟體授權費。
- 支援的軟體眾多。

1.2 AWS 是什麼？

AWS（Amazon Web Service）本是為解決 Amazon 公司內部問題而生，但後來為了向全世界提供這種系統架構，於是在 2006 年做為一種 IT 基礎設施服務，開始對外提供。



1.2.1 AWS 所解決的問題

所謂需要解決的問題，也就是如下這些傳統的內部部署式系統架構的一些限制。

- 不需要硬體時仍必須擁有硬體。
- 硬體的採購很花時間。
- 即使制訂了縝密的計畫，仍無法應付需求的變化。
- 必須配置人員來應付磁碟故障等實體維護工作。
- 無法應付急遽增長的存取量。

Amazon Web Service（AWS）的誕生正是為了解決這類問題，AWS 就是一種於時代的進化及不斷改變的需求中，持續解決新的挑戰的服務。

1.2.2 AWS 的優點

AWS 具有如下的優點特色。

- 不必擁有基礎設施，但卻能在需要時使用需要的量。
- 用多少付多少，不用就不會產生費用。
- 只要短短幾分鐘便能建立起新的伺服器。
- 在需求發生變化時，也能彈性應對、靈活地重新製作。
- 可集中力量於服務的提供，而不必多花力氣在磁碟管理等硬體的效能維護上。
- 可直接回應存取量等需求變化。

Chapter. 3 啟動 AWS 上的 Linux 伺服器

3.1 EC2 是什麼？

本書是使用 **Amazon Elastic Compute Cloud（EC2）**來建構 Linux 伺服器。

首先來簡介一下 EC2 到底是什麼樣的服務。

3.1.1 資料儲存不受限

- 可在必要時啟動所需要的伺服器。
- 不需要時可隨時丟棄。
- 僅針對使用期間收費。
- 幾分鐘內就可啟動。
- 可依據各種用途、規模來選擇性能。
- 可從 AMI（Amazon Machine Image）啟動多個具相同配置的伺服器。
- 可搭配 AWS 的其他服務一同運用，易於管理。
- 可啟動於全世界各個地區。

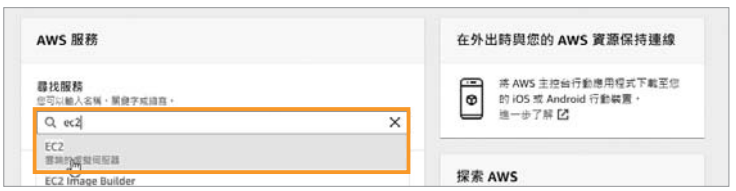
如上所列，EC2 具有各式各樣的優點，尤其是能夠迅速啟動、簡單測試後丟棄這點，在進行如本書這類測試時也非常具吸引力。

接著就讓我們一邊著手啟動 EC2，一邊介紹其主要功能。

3.2 建立 EC2 執行個體

請將所謂的執行個體，想成是以 EC2 建構的虛擬伺服器。

首先請登入管理控制台，並進入 EC2 的儀表板。



不論是在管理控制台的尋找服務欄位搜尋，還是從所有服務清單中尋找，都能很快找到 EC2。

3.4 使用 Session Manager 存取 EC2 執行個體

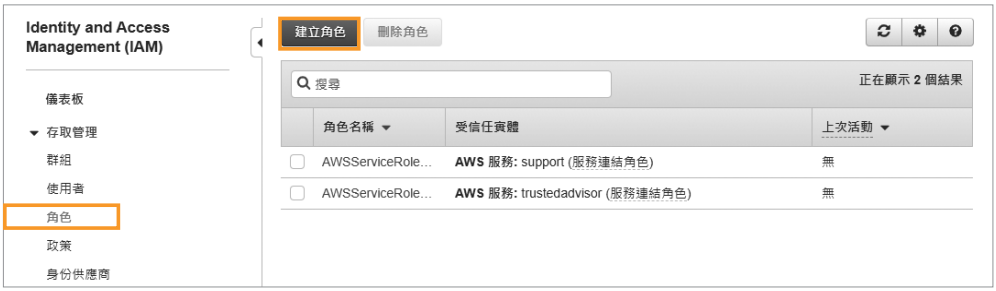
進行 SSH 連線存取時，在操作上會有一些安全性的考量，像是 Windows 必須安裝專用的軟體、一定要儲存私密金鑰、必須在安全群組設定許可 22 號連接埠等。
若是能從管理控制台執行同樣操作，那也不失為一種便利的方式。
為了讓大家知道更多不同做法，前面介紹的是傳統的 SSH 連接方法，不過接下來則要介紹利用相對較新的 Session Manager 來連線的方式。

這個 Session Manager 是 **AWS Systems Manager** 服務的功能之一。
而 Amazon Linux 2 本來就裝有 AWS Systems Manager 代理程式，故只要設定好對 EC2 上 Systems Manager 的存取權限，便能夠加以運用。

3.4.1 設定對 EC2 的存取權限

欲設定對 EC2 上 Systems Manager 的存取權限時，必須使用 **IAM 角色**。
請從管理控制台進入 IAM 的儀表板。

所謂的 IAM 角色，就是一種可用來針對 AWS 資源賦予特定權限的元素。



在 IAM 儀表板的左側點選「角色」項目後，按一下右方內容上端的〔建立角色〕鈕。

在「選擇信任的實體類型」部分，點選「AWS 服務」。

11.2 瞭解 CloudWatch 指標與 Logs 的功能

11.2.1 使用 ps 指令來查看程序

我們可用 ps 指令來查看目前正在執行的程序。而加上 -aux 選項，便能將所有使用者的程序一併詳細列出。

```
$ ps -aux
```

只想查看程序 ID 等較少資訊時，則可用 -ax 選項執行。

```
$ ps -ax
```

```
PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:07 /usr/lib/systemd/systemd --switched-root --system
--deserialize 22
    2 ?        S      0:00 [kthreadd]
    4 ?        I<     0:00 [kworker/0:0H]
    6 ?        I<     0:00 [mm_percpu_wq]
    7 ?        S      0:00 [ksoftirqd/0]
    8 ?        I      0:01 [rcu_sched]
    9 ?        I      0:00 [rcu_bh]
   10 ?        S      0:00 [migration/0]

~ 省略 ~

 3181 ?        S      0:00 qmgr -l -t unix -u
 3227 ?        Ssl    0:16 /usr/bin/amazon-ssm-agent
 3231 ?        Ssl    0:02 /usr/sbin/rsyslogd -n
 3247 ?        Ss     0:00 /usr/sbin/crond -n
 3250 ?        Ss     0:00 /usr/sbin/atd -f
 3299 tty1     Ss+    0:00 /sbin/agetty --noclear tty1 linux
 3300 ttyS0    Ss+    0:00 /sbin/agetty --keep-baud 115200,38400,9600 ttyS0 vt220
 3400 ?        Ss     0:00 /usr/sbin/sshd -D
 3439 ?        Ss     0:00 /usr/sbin/acpid
 5922 ?        S      0:00 pickup -l -t unix -u
 7456 ?        I      0:00 [kworker/u30:0]
 8168 ?        Sl     0:02 /usr/bin/ssm-session-worker yamashita-
065b4f9d9797686fa i-0aae4ecd6e997c545
 8179 pts/0    Ss     0:00 sh
 9363 ?        I      0:00 [kworker/0:0]
```

如上所示，目前有許多程序正在運作。想要分頁檢視或搜尋這些輸出結果時，可將結果以管線（|）傳遞給 less 指令。

指標名稱	內容
VolumeReadBytes	讀取操作所傳送的位元組總數
VolumeWriteBytes	寫入操作所傳送的位元組總數
VolumeReadOps	讀取操作的總數
VolumeWriteOps	寫入操作的總數
VolumeTotalReadTime	讀取操作所耗用時間的總計
VolumeTotalWriteTime	寫入操作所耗用時間的總計
VolumeIdleTime	未進行讀取也未進行寫入操作的時間總計
VolumeQueueLength	待完成的請求總數

關於 CPU 點數

t2、t3 有固定的 CPU 基準使用率，以 t2.micro 來說是 10%。一旦超過基準，就需要 CPU 點數。當 CPU 的使用率低於基準時，便會累積 CPU 點數。而 CPU 點數會在 EC2 執行個體停止時被重設。

另外也有即使無 CPU 點數仍可超出基準使用率的 Unlimited 方案可選。

採取 Unlimited 方案時，若已無剩餘的 CPU 點數，則所花費的 CPU 點數就會被記為負值，並以之後低於基準期間所累積的 CPU 點數來抵用。

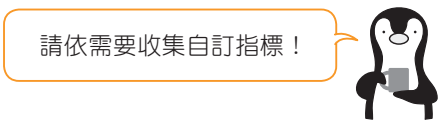
但若 EC2 執行個體在 CPU 點數仍為負值的狀態下就被停止、終止，那麼系統就會針對以 Unlimited 方案花費的 CPU 點數計算費用。

11.2.3 CloudWatch 自訂指標

標準指標並未收集記憶體的使用量及磁碟的剩餘可用空間。這是因為，EC2 執行個體的作業系統屬於使用者所管理的範圍。既然使用者能夠自由地全面控制作業系統，那麼其管理也就必須由使用者自行負責。

而 EC2 執行個體的 OS 的記憶體使用量及磁碟剩餘空間等資訊，也可藉由安裝 CloudWatch 代理程式來收集。像這樣由使用者自行收集而來的指標，就稱為自訂指標。

由於要收集 EC2 執行個體的自訂指標前，必須先在 EC2 執行個體上安裝接下來將介紹的 CloudWatch Logs 和通用的 CloudWatch 代理程式，因此其收集方法將於後續內容中一併解說。



Chapter. 12 Linux 的安全性設定

為了透過 AWS 來學習 Linux，在先前的章節中便已依需要解說了各種與安全性有關的設定。而本章則是進一步統一整理了保護 EC2 執行個體所需的各項安全性設定。

要保護的範圍

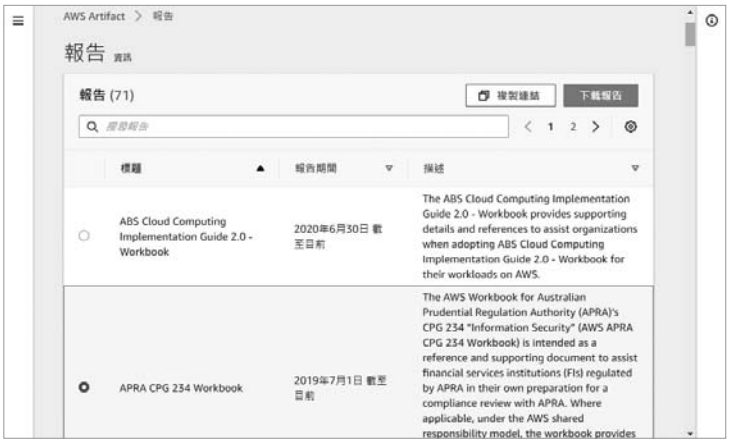
App 使用者	資料…等等
Linux 使用者	軟體 / 應用程式
EC2 Linux	
VPC	安全群組 /NACL
IAM 使用者 /IAM 角色	IAM 政策
根使用者	
AWS 帳戶	

不需保護的範圍

DC、HW、NW 等的使用	網路連線（NW）
硬體（HW）	軟體（SW）
資料中心（DC）	

就 EC2 執行個體的安全性而言，基本上可分為要保護的範圍和不需保護的範圍。所謂不需保護的範圍，其實就是我們無法掌控的部分。這些部分包括了 AWS 為了提供雲端服務所營運並管理的資料中心設施、硬體、網路連線及軟體等。

資料中心的實體保全，還有存放資料之儲存設備的廢棄處理、網路連線的監控與保護、實體與軟體層級的認證等，都屬於 AWS 負責的範圍。不是我們管得到的。



關於 AWS 是如何保護這些資源，又是如何從外部加以審核等資訊，你可從管理控制台進入 **AWS Artifact** 的頁面以取得相關報告，而在《Amazon Web Services: Overview of Security Processes》白皮書等文件中也有相關說明。

身為 AWS 使用者的我們，可瞭解自己所能掌控的保護範圍，並致力於維護其安全性。
雖說在先前各章節中也曾依需要分別做過說明， 不過本章將統一針對 EC2 執行個體上的 Linux 伺服器安全防護，解說其中較具代表性的一些要素。

12.1 AWS 帳戶與根使用者

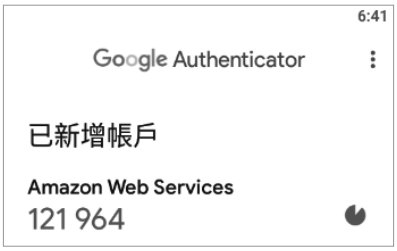
在第 2 章介紹環境建構時便已提到過，AWS 帳戶是依各個使用者區分的獨立環境。因此首先，我們必須保護 AWS 帳戶的安全。

12.1.1 關於根使用者的運用

在此，最重要的要素就是根使用者。根使用者具有 AWS 帳戶的所有權限，而我們無法縮減其權限範圍。
在進行一般的操作處理時，不會使用根使用者。除非是要進行只有根使用者才能進行的操作，否則都不會用到根使用者。而只有根使用者才能進行的操作主要有以下這些。

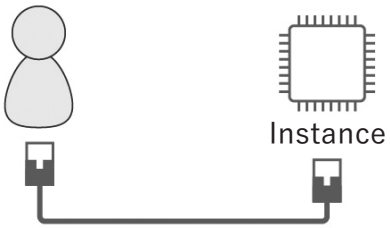
- 變更帳戶名稱、根使用者的電子郵件地址、密碼
- 變更 AWS 的支援計劃
- 在預留執行個體市場上登記掛賣
- 建立 CloudFront 金鑰對
- 啟用 S3 儲存貯體的 MFA Delete
- 關閉 AWS 帳戶

12.1.2 替根使用者新增 MFA



正如第 2 章曾詳細說明過的，請啟用多重驗證（Multi Factor Authentication）功能。

Chapter. 13 瞭解網路



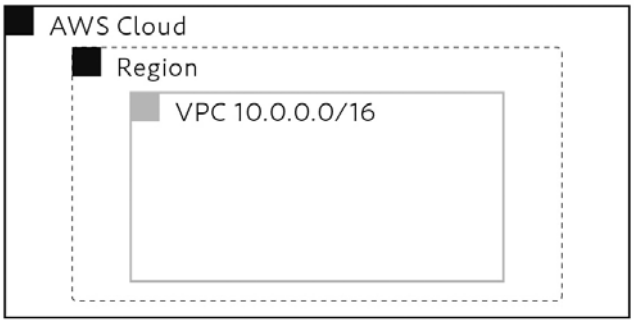
本章將針對啟動 EC2 執行個體的網路及網路相關指令、注意事項、最佳做法等進行解說。在第 3 章中，我們利用 AWS 帳戶預設已備好的 VPC 網路環境，建構了 EC2 執行個體。
這個預設的 VPC，是為了在 AWS 上進行驗證時能夠簡單輕鬆地開始而準備的。
本書便是以預設的 VPC 輕鬆地開始了驗證處理。但在建構正式的營運環境時，必須先以 VPC 建構專用的網路環境，再建立 EC2 執行個體。

因此接下來就要為各位說明 VPC 及 VPC 相關的網路服務。

13.1 設定 VPC 網路環境

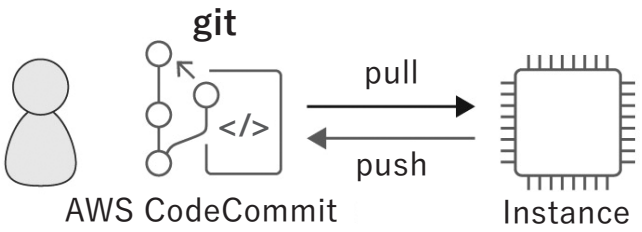
13.1.1 什麼是 VPC 網路？

利用 VPC（Virtual Private Cloud），我們就能在 AWS 上建構隔離的網路環境。



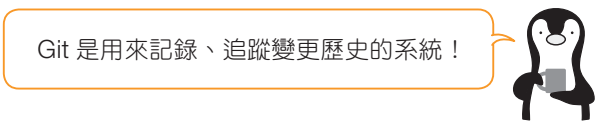
首先選擇在想要的地區，定義想要的私有 IP 位址範圍，以建立 VPC。
其中 IP 位址範圍是以 CIDR 表示法來指定。而在指定 IP 位址時，可指定 IPv4 位址也可指定 IPv6 位址。

Chapter. 14 版本管理也用 AWS



有個版本管理系統叫 Git。這個系統能對開發工程師所更新的程式原始碼等進行版本管理、與軟體組建及部署工具協作以達成自動化、進行多人的團隊開發工作等。
不只是開發人員，其實在營運與建構方面也會用到 Git，所以接著就讓我們來學習一些基本的 Git 指令。

而將 Git 做為一種託管服務來提供的，是 **AWS CodeCommit**。
在本章中，我們就要嘗試透過 EC2 執行個體上的 Git 指令操作來使用 AWS CodeCommit。



14.1 安裝 Git

首先要安裝 Git 用戶端。

```
$ sudo yum -y install git
```

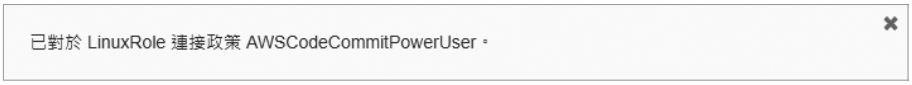
14.2 設定操作 CodeCommit 的權限

14.2.1 附加 IAM 政策以賦予權限

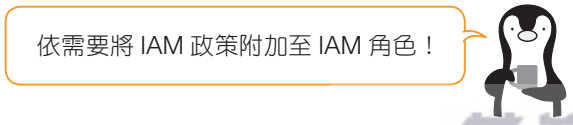
在 EC2 執行個體上從 AWS CLI 操作 CodeCommit 的權限，要透過 IAM 角色來設定。請從管理控制台進入 IAM 的儀表板。

點選左側導覽選單中的「角色」項目，再從右方的 IAM 角色清單中點選「LinuxRole」。接著按一下〔連接政策〕鈕。

於「篩選政策」欄位輸入「codecommit」搜尋，勾選搜尋結果中的「AWSCodeCommit PowerUser」政策後，按一下〔連接政策〕鈕。



畫面中出現已連接的訊息。

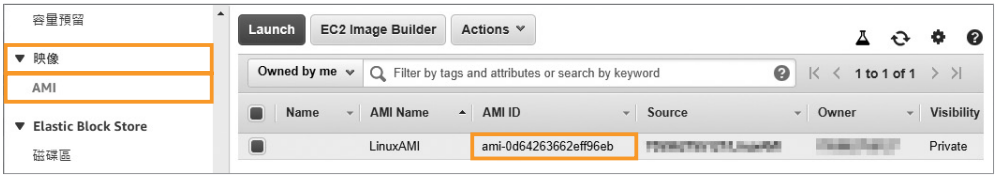


Chapter.20 完成學習課程後記得刪除 AWS 資源

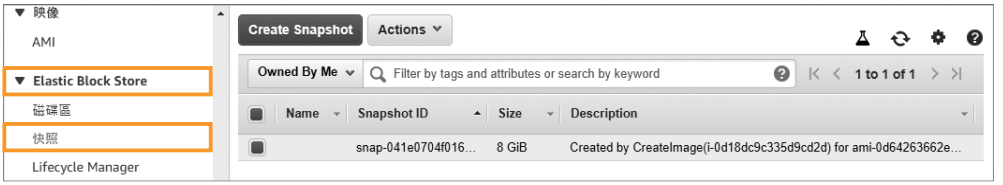
不再需要的 AWS 資源就要刪掉，以免產生費用。本章便為各位列出本書曾處理過的各個 AWS 資源的刪除步驟。

20.1 取消 AMI 的註冊與刪除 EBS 快照

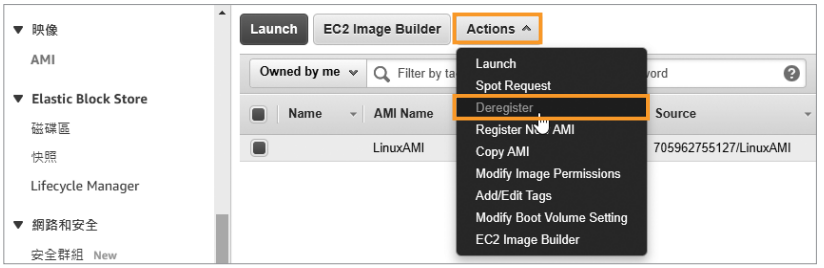
EC2 的 AMI 若不再需要，就要「取消註冊」。
而取消 AMI 的註冊後，別忘了也要刪除與之綁定的 EBS 快照。



在 EC2 儀表板中，點選左側導覽選單「映像」分類下的「AMI」項目，於 AMI 的清單中確認 AMI 的 ID。以此例來說，為「ami-0d64263662eff96eb」。



同樣在 EC2 儀表板中，點選左側導覽選單「Elastic Block Store」分類下的「快照」項目，於 EBS 快照的清單中查看「Description」（說明）欄的內容。
在此可看到寫有「for ami-0d64263662eff96eb」的 EBS 快照。



目標對象都已確認完成，就可動手取消 AMI 的註冊。點選欲取消註冊的 AMI 後，再點選「Actions」（動作）-「Deregister」（取消註冊）。