Docker說明

[1. 前言 2](#_Toc182941793)

[1.1. 映像檔與容器介紹 2](#_Toc182941794)

# 前言

https://hackmd.io/@whYPD8MBSHWRZV6y-ymFwQ/rynv0a7CT

使用 Docker，能簡化從下載軟體到運行起來的過程，達到輕鬆部署的效果。且每個軟體皆有自己專屬的執行環境，因此就算換到另一台電腦，只要有裝 Docker，都能正常地運行。

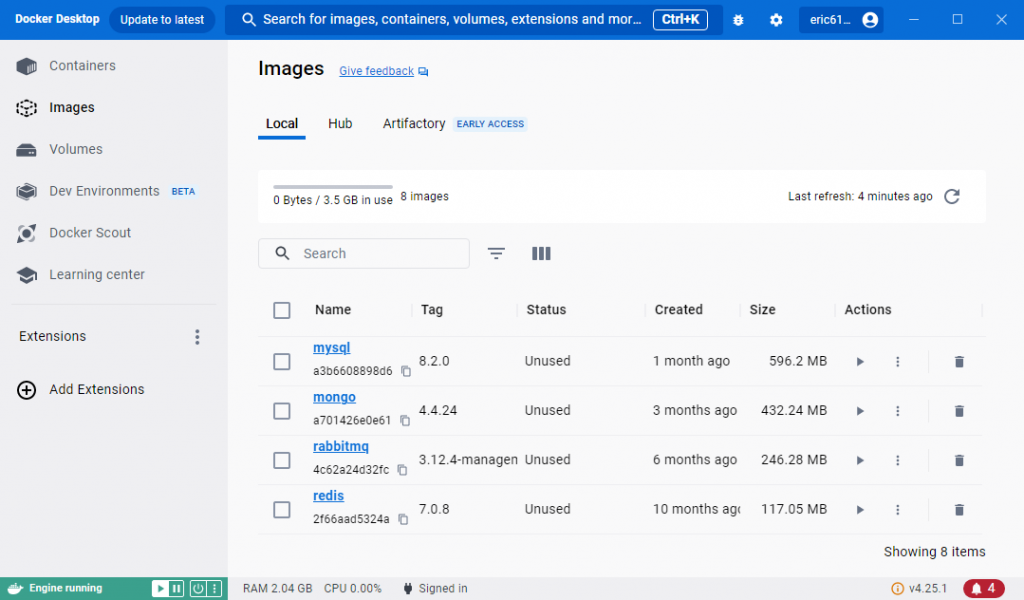
## 映像檔與容器介紹

Docker 會將軟體程式以及運行時所需要的環境，打包在一個叫「映像檔」（image）的東西裡。軟體程式就像是前端 React 的網頁專案，或是後端 Spring Boot 的 JAR 檔。而對應的環境則為 Node.js 或 Java。讀者姑且可將映像檔想像成安裝程式。

使用映像檔，可建立出「容器」（container）。容器是軟體所運行的地方，且與原本的電腦環境是隔離的。舉例來說，假設我們在電腦上安裝了 Java 8，則在容器 A 中仍可運行要求 Java 11 的軟體。當然也能在容器 B 準備 Java 17 環境，並讓它們都在同一台電腦上跑。

看到這裡，讀者或許有個想法：「那一開始便在電腦上安裝 Java 17，這樣所有軟體不就能直接在電腦上運行了嗎？」關於這問題，請假想有軟體是使用低版本的 Java 8 語言所寫成，但有些用到的 library 到了較高版本的 Java 11 或 17，已經被棄用（deprecated）甚至移除了，那麼該軟體可能是無法運行。此為容器環境彼此隔離的好處！

以下是 Docker Desktop 的映像檔清單，筆者曾經下載過數個映像檔，畫面供讀者參考。

docker-image-list-demo.png  


筆者的電腦是 Windows 系統，到這邊為止，就算是完成 Docker 的安裝。下一篇文章開始，會開始介紹映像檔與容器的操作。