

我的OS是ubuntu22.04請問我要怎樣知道，我的系統上現在有沒有安裝docker？版本為何？有沒有啟動呢？

要在 Ubuntu 22.04 上確認你的系統是否有安裝 Docker、版本為何，以及服務有沒有啟動，可以依照以下步驟操作：

1. 檢查 Docker 是否已安裝

打開終端機，輸入：

```
docker --version
```

如果有安裝，會顯示類似 `Docker version 24.0.6, build ed223bc` 這樣的版本資訊。如果沒安裝，會顯示指令找不到或提示你安裝 Docker^[1]。

2. 查詢 Docker 版本與詳細資訊

輸入：

```
docker version
```

這個指令會顯示 Client 和 Server 端的詳細版本資訊。你可以在 Server 欄位看到安裝的 Docker Engine 版本^{[2] [3] [4]}。

3. 檢查 Docker 服務是否啟動

輸入：

```
sudo systemctl status docker
```

你會看到類似以下的輸出：

- `docker.service - Docker Application Container Engine`
Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
Active: active (running) since ...

只要看到 `Active: active (running)`，就代表 Docker 服務目前正在運行^{[5] [6] [7] [1] [4]}。

如果顯示 `inactive` 或 `failed`，代表 Docker 沒有啟動。這時你可以用以下指令啟動：

```
sudo systemctl start docker
```

4. 進一步確認 Docker 狀態

你也可以用：

```
docker info
```

這個指令會顯示更多關於 Docker 的運作狀態、容器數量、映像檔數量等資訊。如果 Docker 沒有啟動，這個指令會出現錯誤訊息 [\[5\]](#) [\[3\]](#) [\[1\]](#)。

常用指令總結

功能	指令	說明
檢查是否安裝	<code>docker --version</code>	有安裝會顯示版本
查詢詳細版本	<code>docker version</code>	顯示 Client/Server 版本資訊
檢查服務狀態	<code>sudo systemctl status docker</code>	顯示 Docker 服務是否啟動
查看詳細狀態	<code>docker info</code>	顯示運作狀態與詳細資訊

只要這幾個指令，你就能快速確認 Docker 是否安裝、版本為何、有沒有啟動。

✻

コマンド 'docker' が見つかりません。次の方法でインストールできます:

```
sudo snap install docker # version 28.1.1, or
```

```
sudo snap install docker # version 27.5.1
```

```
sudo apt install docker.io # version 26.1.3-0ubuntu1~24.04.1
```

```
sudo apt install podman-docker # version 4.9.3+ds1-1ubuntu0.2
```

他のバージョンについては 'snap info <snapname>' を確認してください。

根據你提供的訊息「コマンド 'docker' が見つかりません」，這表示你的 Ubuntu 22.04 系統目前尚未安裝 Docker 指令，或是安裝後沒有正確設定 PATH。因此：

- 你的系統「尚未安裝」Docker，或安裝不完全，否則執行 docker 指令時會有版本資訊或其他回應 [\[8\]](#) [\[9\]](#) [\[10\]](#) [\[11\]](#)。
- 既然沒有安裝，當然也查不到版本號與啟動狀態。

安裝 Docker 的建議步驟

你可以依照以下步驟安裝 Docker：

1. 更新系統套件庫

```
sudo apt update
```

2. 安裝 Docker (推薦用 Ubuntu 官方套件庫的 [docker.io](https://docs.docker.com/install/linux/docker-ce/ubuntu/) 套件)

```
sudo apt install docker.io
```

或者根據提示，也可以選擇用 snap 安裝：

```
sudo snap install docker
```

3. 啟動並設為開機自動啟動

```
sudo systemctl start docker
sudo systemctl enable docker
```

4. 確認 Docker 是否安裝成功與版本

```
docker --version
```

或

```
docker version
```

5. 檢查 Docker 服務狀態

```
sudo systemctl status docker
```

補充說明

- 如果你想用 apt 安裝最新版 Docker，可以參考官方文件，先加上 Docker 官方套件庫再安裝^[8]^[9]^[10]^[11]。
- 安裝後，建議將自己的帳號加入 docker 群組，這樣執行 docker 指令時就不用每次都加 sudo^[8]^[10]^[11]。

總結：目前你的系統沒有安裝 Docker，也查不到版本與啟動狀態。請依上述步驟安裝後再查詢即可。

✻

\$ sudo systemctl start docker

Failed to start docker.service: Unit docker.service not found.

你執行 `sudo systemctl start docker` 時出現錯誤「Failed to start docker.service: Unit docker.service not found.」表示系統找不到名為 `docker.service` 的服務單元，原因可能是 Docker 並非以傳統 apt 套件安裝，而是透過 snap 安裝，或是 Docker 尚未正確安裝。

根據相關解決方案，請依照以下步驟排查與處理：

1. 確認 Docker 是否透過 snap 安裝

執行：

```
sudo snap list docker
```

如果有安裝，會顯示版本資訊。

接著檢查 snap 版 Docker 服務狀態：

```
sudo snap services docker
```

如果看到類似 `docker dockerd enabled active`，代表 snap 版 Docker 服務正在運行。

若服務未啟動，可用：

```
sudo snap start docker
```

來啟動服務。

2. 如果是 snap 版 Docker，啟動服務的 systemd 單元名稱會是：

```
snap.docker.dockerd.service
```

你可以用以下指令啟動或重啟：

```
sudo systemctl restart snap.docker.dockerd.service
```

3. 如果沒有安裝 Docker，或想改用 apt 安裝，請先移除 snap 版：

```
sudo snap remove docker
```

然後用 apt 安裝：

```
sudo apt update  
sudo apt install docker.io
```

安裝完成後，啟動 Docker 服務：

```
sudo systemctl start docker  
sudo systemctl enable docker
```

並確認狀態：

```
sudo systemctl status docker
```

4. 另一種快速安裝官方最新版 Docker 的方法是使用官方安裝腳本：

```
curl -sSL https://get.docker.com/ | sh
```

此腳本會自動安裝並設定 Docker，安裝完成後即可使用 systemctl 管理 docker.service。

總結

- 「Unit docker.service not found」通常是因為 Docker 是用 snap 安裝，服務名稱不同，或 Docker 尚未安裝。
- 若是 snap 版 Docker，請用 `sudo snap start docker` 或 `sudo systemctl restart snap.docker.dockerd.service` 啟動服務。
- 若想用傳統 apt 安裝，請先移除 snap 版，再用 `sudo apt install docker.io` 安裝，並用 systemctl 管理服務。

這樣即可解決找不到 docker.service 單元的問題，並順利啟動 Docker 服務^{[12] [13]}。

✻

我已經在 dockerhub 註冊帳號可以登入了？請問這對使用docker有什麼好處？有何幫助？

你已經在 Docker Hub 註冊並登入帳號，這對使用 Docker 有以下幾個重要好處與幫助：

1. 容器映像的存取與管理

- 你可以將自己建置的 Docker 映像 (Image) 推送 (push) 到 Docker Hub，方便保存和備份^{[14] [15]}。
- 也能從 Docker Hub 拉取 (pull) 其他人或官方公開的映像，快速取得需要的應用環境，節省自行建置時間^{[14] [16]}。

2. 映像的分享與協作

- Docker Hub 提供公開和私有儲存庫，可以選擇將映像公開給所有人使用，或限制只有授權用戶能存取，方便團隊合作與版本管理^{[14] [16] [17]}。
- 團隊成員可以共用映像，確保開發、測試與生產環境一致，減少環境差異帶來的問題^[18]。

3. 自動化與整合

- Docker Hub 支援 Webhooks，可在映像推送時觸發自動化流程，如持續整合 (CI) 和持續部署 (CD) ^[14]。
- 你可以將 Docker Hub 與其他工具 (如 GitHub、Jenkins) 整合，簡化開發到部署的工作流程 ^[14]。

4. 安全性與版本控制

- Docker Hub 會對映像進行安全掃描，幫助你了解映像中可能存在的安全漏洞 ^[14]。
- 支援映像的版本管理，可以隨時回滾到先前版本，方便維護和更新 ^{[18] [14]}。

總結

註冊並登入 Docker Hub，讓你能輕鬆管理和分享 Docker 映像，提升開發效率，確保環境一致性，並且方便與團隊協作及自動化部署，是使用 Docker 的重要一環 ^{[18] [14] [16]}。

✻

我目前在這ubuntu機上有安裝virtualbox以運行windows11，請問我可以運用docker來達到使用windows11的效果嗎？

你目前在 Ubuntu 上用 VirtualBox 運行 Windows 11，想知道是否能用 Docker 來達到使用 Windows 11 的效果，答案是：

Docker 與虛擬機 (VirtualBox) 本質不同

- VirtualBox 是完整的虛擬機管理程式，可以模擬整套作業系統 (例如 Windows 11)，讓你能在 Ubuntu 上完整執行 Windows 11 環境。
- Docker 是容器技術，主要用來封裝應用程式及其執行環境，基於作業系統層級的虛擬化，並非模擬整個作業系統。

Docker 無法直接在 Ubuntu 上運行 Windows 11 桌面環境

- Docker 容器需要與宿主機的核心 (kernel) 共用，Linux 上的 Docker 容器是 Linux 容器，無法直接運行 Windows 桌面系統或完整 Windows 11 作業系統。
- Windows 容器必須在 Windows 主機上運行，且通常需要 Windows 10/11 專業版或企業版，並啟用 Hyper-V 和 Docker Desktop 等支援 ^{[19] [20]}。

Windows 容器的限制與需求

- Windows 容器主要用於執行 Windows 應用程式的服務容器，不是完整桌面系統。
- Windows 容器在虛擬機 (如 VirtualBox) 中運行會有支援限制，官方建議虛擬機主機使用 Windows Server 來支援容器 ^[19]。

結論

- 若你想在 Ubuntu 上完整使用 Windows 11 桌面環境，VirtualBox 是正確選擇，因為它能模擬整個作業系統。
- Docker 無法取代虛擬機來執行完整 Windows 11 桌面系統，只適合用來執行應用程式容器，且 Windows 容器必須在 Windows 主機上運行。
- 若你想在 Windows 11 上使用 Docker，則可安裝 Docker Desktop，並運行 Windows 容器或 Linux 容器，但這與在 Ubuntu 上使用 Docker 來「使用 Windows 11」是不同的概念^{[19] [20] [21]}。

總結：你無法用 Docker 在 Ubuntu 上達到像 VirtualBox 那樣完整運行 Windows 11 的效果，Docker 是應用程式容器技術，不是虛擬機，無法取代完整的 Windows 桌面環境。



請問，我使用virtualbox以運行windows11主要是要執行Eviews8.1的軟體，請問我可以運用docker來達到使用Eviews8.1的軟體的效果嗎？

你目前用 VirtualBox 在 Ubuntu 上運行 Windows 11 來使用 EViews 8.1，想知道是否能用 Docker 來達到同樣效果，以下是專業分析：

Docker 運行 Windows 桌面環境的現況

- 雖然有社群專案（如 dockurr/windows）能在 Linux 主機上透過 Docker 容器搭配 KVM 虛擬化技術，運行 Windows 11 桌面環境^{[22] [23] [24] [25]}，但這並非 Docker 傳統意義上的純容器，而是結合了虛擬機技術（KVM）來模擬完整 Windows 系統。
- 這類方案仍屬實驗性質，設定較複雜，且效能與穩定性可能不如直接使用 VirtualBox 這類成熟虛擬機軟體。
- Docker 容器本身不支援完整 Windows 桌面 GUI，且 Windows 容器主要用於執行 Windows Server 或應用程式服務，不適合直接執行像 EViews 這類需要完整桌面環境的軟體^[26]。

EViews 8.1 的需求與 Docker 適用性

- EViews 是 Windows 平台的統計分析軟體，需完整 Windows 作業系統支援才能執行。
- Docker 容器無法直接執行 Windows 桌面應用，除非在 Windows 容器中，但 Windows 容器多用於服務型應用，且 Windows 11 官方並無正式的 Docker 映像可用^[26]。
- 目前沒有成熟的 Docker 映像能直接支援執行 EViews 8.1 這類 Windows 桌面應用。

建議方案

- 繼續使用 VirtualBox 來虛擬 Windows 11，這是目前最穩定且支援完整桌面應用的方式。
- 若想嘗試用 Docker 運行 Windows 11 桌面，可參考社群專案 dockurr/windows 等，但須準備好較複雜的設定與可能的效能問題^{[22] [24] [25]}。

- 若只是想執行 EViews，虛擬機方案仍是最佳選擇。

總結

項目	VirtualBox (虛擬機)	Docker 運行 Windows 桌面
運行完整 Windows 11	支援，穩定	可行但需結合 KVM，較複雜且不穩定
執行 EViews 8.1	完整支援	不直接支援，無成熟映像
設定難度	低至中	高，需額外設定 KVM 與網路等
效能	較佳	可能較差，且不保證穩定

因此，若目標是執行 EViews 8.1，建議繼續使用 VirtualBox 虛擬 Windows 11，Docker 目前無法直接替代此用途。



那我安裝docker還有加入Dockerhub可以做些什麼有趣或實用的工作嗎？

安裝 Docker 並加入 Docker Hub 帳號後，你可以做很多有趣且實用的工作，主要包括：

1. 快速取得與運行各種應用容器

- 從 Docker Hub 下載數百萬個公開鏡像 (Image)，例如 Ubuntu、MySQL、Nginx、Python、Node.js 等，快速搭建開發、測試或生產環境 [27] [28]。
- 只需一條指令就能拉取鏡像並啟動容器，省去繁瑣的環境設定。

2. 自訂並分享自己的鏡像

- 你可以基於官方鏡像或自訂環境，製作自己的 Docker 映像，並推送 (push) 到 Docker Hub，方便跨機器或團隊共享 [28] [29]。
- 支援公開或私有倉庫，保護你的專案或商業機密。

3. 自動化建構與持續整合

- Docker Hub 支援自動構建功能，能連結 GitHub 或 Bitbucket，當你推送程式碼時自動建構並更新鏡像 [28] [29]。
- 可設定 webhook，自動觸發部署流程，提升開發效率。

4. 團隊協作與版本管理

- 透過 Docker Hub 的組織與團隊管理功能，方便多人協作、權限控管與版本控制 [28]。
- 讓開發、測試、部署環境保持一致，減少「在我機器上可以跑」的問題。

5. 節省資源與提升效能

- Docker 容器比傳統虛擬機輕量，啟動快，資源佔用少，適合快速部署和擴展^[30]。
- 適合微服務架構、雲端部署等現代應用場景。

6. 實際應用案例示範

- 建立資料庫容器（如 MySQL），並將其狀態保存成鏡像，隨時拉取使用^[31]。
- 部署 Web 伺服器、開發環境、測試工具等，實現一鍵啟動。

總結

功能	說明
拉取與運行鏡像	快速部署各種應用環境
自訂鏡像與分享	製作專屬環境並推送到 Docker Hub
自動化建構	連結 GitHub 自動建構與更新鏡像
團隊協作	多人管理鏡像與權限控制
節省資源	輕量快速，適合現代應用與微服務架構

透過 Docker 和 Docker Hub，你可以大幅簡化開發、測試與部署流程，並且方便與團隊共享與協作，這讓軟體開發和運維工作變得更高效且有趣^{[30] [27] [28]}。

✻

1. <https://tms-outsourcing.com/blog/posts/how-to-check-if-docker-is-running/>
2. <https://www.docker.com/blog/how-to-check-docker-version/>
3. <https://thelinuxcode.com/checking-if-docker-daemon-or-container-is-running/>
4. <https://onebite.dev/snippet/docker/check-if-docker-daemon-is-running>
5. <https://linuxconfig.org/check-if-docker-is-running-linux-commands>
6. <https://stackoverflow.com/questions/43721513/how-to-check-if-the-docker-engine-and-a-docker-container-are-running>
7. <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-22-04>
8. <https://cloudcone.com/docs/article/how-to-install-docker-on-ubuntu-22-04-20-04/>
9. <https://www.cherryservers.com/blog/install-docker-ubuntu-22-04>
10. <https://linuxconfig.org/how-to-install-docker-on-ubuntu-22-04>
11. <https://www.youtube.com/watch?v=yPrhyaNKYfU>
12. <https://stackoverflow.com/questions/43569781/unable-to-start-docker-service-with-error-failed-to-start-docker-service-unit>
13. <https://copyprogramming.com/howto/failed-to-start-docker-service-unit-docker-service-not-found>
14. <https://blog.csdn.net/arthas777/article/details/133623458>
15. <https://www.cnblogs.com/liuyuelinfighting/p/16418769.html>

16. <https://rexford0302.github.io/2021/12/10/20211210/>
17. <https://ithelp.ithome.com.tw/m/articles/10248160>
18. <https://tw.alphacamp.co/blog/docker>
19. <https://learn.microsoft.com/zh-tw/troubleshoot/windows-server/containers/support-for-windows-containers-docker-on-premises-scenarios>
20. <https://docs.docker.net.tw/desktop/setup/install/windows-install/>
21. https://viml.nchc.org.tw/archive_blog_823/
22. <https://github.com/trannguyenhan/windows-docker-container>
23. <https://github.com/isaccanedo/Windows-inside-a-Docker-container/blob/main/readme.md>
24. <https://www.youtube.com/watch?v=xhGYobuG508>
25. <https://linuxcommunity.io/t/run-windows-with-docker-compose-kvm/3965>
26. <https://forums.docker.com/t/windows-11-docker-image/140055>
27. https://yeasy.gitbook.io/docker_practice/repository/dockerhub
28. <https://www.widuu.com/docker/userguide/dockerrepos.html>
29. <https://juejin.cn/post/7327352858118242330>
30. https://blog.csdn.net/baimao_Ch/article/details/141020840
31. <https://www.cnblogs.com/dogleftover/p/15724359.html>