

# 以docker container開啟nginx網路伺服器

Web Server（網路伺服器）是一台存放網站檔案的電腦，或是處理這些檔案請求的軟體。硬體伺服器（可以是虛擬機）存放 HTML、CSS、JavaScript 等文件，並與網際網路相連。軟體部分（如 HTTP 伺服器）理解 URL 和 HTTP 協議，負責處理用戶端的請求。當用戶端請求網頁時，伺服器找到相應文件並將其傳送回用戶端。

## 設置 Nginx 網頁伺服器

### 1. 準備好你的 `index.html`

在虛擬機的家目錄，準備好要在 Nginx 伺服器上提供的 `index.html` 文件。

### 2. 創建 Nginx 配置文件

在目錄下創建一個名為 `default.conf` 的配置文件：

```
server {  
    listen 80;  
    server_name localhost;  
  
    location / {  
        root /usr/share/nginx/html;  
        index index.html;  
    }  
}
```

## 步驟 1：啟動 Nginx 容器

首先，啟動一個 Nginx 容器。使用官方的 Nginx 映像來創建一個新的 Docker 容器：

```
docker run --name my-nginx -d -p 80:80 nginx
```

- `-name my-nginx`：設置容器的名稱為 `my-nginx`。
- `d`：以分離模式運行容器。
- `p 80:80`：將本地主機的 80 端口映射到容器的 80 端口。
- `nginx`：使用官方的 Nginx 映像。

## 步驟 2：將 HTML 文件複製到 Nginx 容器中

假設你有一個名為 `index.html` 的 HTML 文件存放在本地的 家目錄，使用 `docker cp` 命令將其複製到容器內的 Nginx 預設目錄 `/usr/share/nginx/html` 中：

```
docker cp index.html my-nginx:/usr/share/nginx/html/
```

## 步驟 3：將 Nginx 配置文件複製到容器中

同樣，假設你有一個自定義的 Nginx 配置文件 `default.conf` 存放在本地的 家目錄中，使用 `docker cp` 命令將其複製到容器內的 `/etc/nginx/conf.d/` 目錄中：

```
docker cp default.conf my-nginx:/etc/nginx/conf.d/
```

## 步驟 4：重新加載 Nginx 配置

Nginx 容器運行後，你需要重新加載配置才能使新的配置生效。你可以使用 `docker exec` 進入容器並重新加載 Nginx 配置：

```
docker exec my-nginx nginx -s reload
```

這將在容器內執行 `nginx -s reload` 命令來重新加載 Nginx 的配置文件。

## 步驟 5：檢查 Nginx 容器狀態並訪問你的網站

你可以使用以下命令來檢查 Nginx 容器的運行狀態：

```
docker container ls -a
```

在瀏覽器中訪問 `http://你虛擬機的ip位址or local host`，你應該能夠看到新的 `index.html` 網頁內容。

