

Docker Compose

安裝

[Install Docker Compose](#) | [Docker Documentation](#)

```
# linux
# 到這裡找版號 https://github.com/docker/compose/releases
# sudo curl -L "https://github.com/docker/compose/releases/download/v2.2.3/docker-compose-linux-x86_64" -o /usr/local/bin/docker-compose
$ sudo curl -L "https://github.com/docker/compose/releases/download/v2.2.3/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c

# 找不到命令的可以建立軟連結
$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

熟悉python的朋友，可以使用pip去安裝docker-Compose

```
$ pip install docker-compose
```

YML 語法結構

[compose-spec/spec.md at master · compose-spec/compose-spec · GitHub](#)

```
version: "3.8"

services: # 容器
  servicename: # 服務名字，這個名字也是內部 bridge網絡可以使用的 DNS name
  container_name: # 自定義container 名稱，不使用會用 servicename
  # build: ./buildfolder
  build: # 加此參數會使用docker file 建立
  # 如果要指定 dockerfile 使用 context + dockerfile
  #build:
  # context: ./buildfolder
  # dockerfile: Dockerfile.dev
  image: # 鏡像的名字(如果使用build，此選項會變成image名稱，如果沒有build，會從網路拉取該名稱image)
  command: # 可選，如果設置，則會覆蓋默認鏡像裡的 CMD命令， ex: /bin/sh -c "while true;do sleep 3600 ;done"
  environment: # 可選，相當於 docker run裡的 --env
  volumes: # 可選，相當於docker run裡的 -v
  networks: # 可選，相當於 docker run裡的 --network
  ports: # 可選，相當於 docker run裡的 -p

  servicename2: #第二個服務

volumes: # 可選，相當於 docker volume create

networks: # 可選，相當於 docker network create
```

範例

```
version: "3.8"

services:
  flask-demo:
    image: flask-demo:latest
    environment:
      - REDIS_HOST=redis-server
      - TZ=Asia/Taipei #設定時區
    networks:
```

```
- demo-network
ports:
  - 8080:5000
volumes:
  - ./config/logstash/config/logstash.yml:/usr/share/logstash/config/logstash.yml:ro,Z
  - ./config/logstash/pipeline:/usr/share/logstash/pipeline:ro,Z #ro: read only, Z: 在SELinux 下唯讀(啟用SELinux才有用)

redis-server:
  image: redis:latest
  networks:
    - demo-network

networks:
  demo-network:
```

docker-compose 與 docker 版本對應，語法說明

<https://docs.docker.com/compose/compose-file/>

docker run 轉換 docker-compose 語法工具

<https://www.composerize.com/>

常用指令

```
# 前景執行
docker-compose up
# 背景執行
docker-compose up -d
# 使用自定義前綴字(預設是用資料夾名稱)
docker-compose -p my_project up -d

# 停止docker-compose
docker-compose stop

# 移除docker-compose
docker-compose down
# 移除docker-compose 同時移除volumn
docker-comopose down -v

# 列出container
docker-compose ps

# 列出log
docker-compose log
docker-compose log -f

# 在webapp 內執行sh
docker-compose exec webapp sh

# 使用top監控
docker-compose top

# 抓取(build)image
docker-compose pull
```

更新服務

當程式有異動，想要重新build image

```
# 重新build image
docker-compose build
# 啟動時重新build
docker-compose up --build
```

```
# 如果有服務被移除，使用 --remove-orphans 移除相關資源
docker-compose up --remove-orphans
# 如果有變更掛載，需要重啟服務
docker-compose restart
```

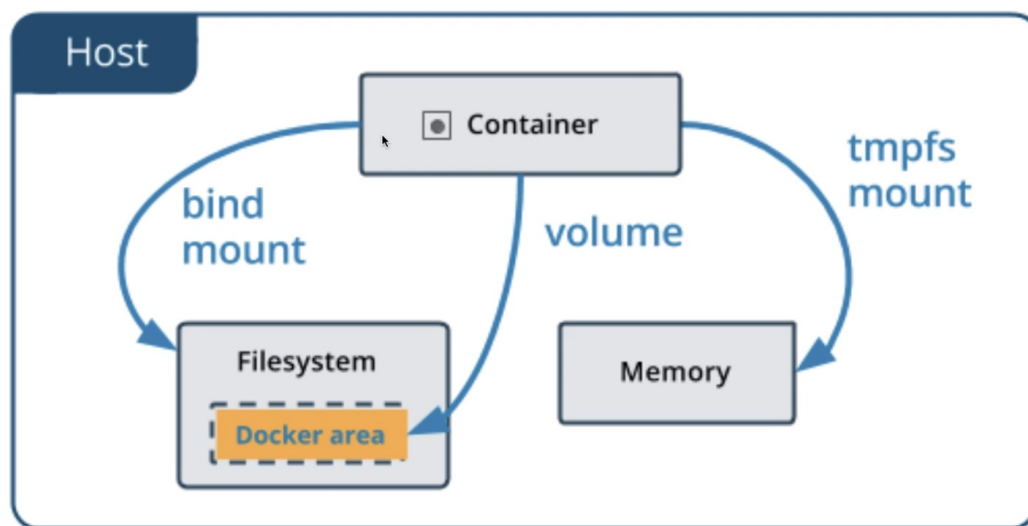
網路配置

```
networks:
  mynetwork1:
    ipam:
      driver: default
      config:
        - subnet: 172.28.0.0/16
  mynetwork2:
    ipam:
      driver: default
      config:
        - subnet: 172.29.0.0/16
```

清除環境

```
docker-compose stop
docker-compose rm
docker system prune -f
```

Docker Compose 磁碟掛載



volumes on the Docker host

docker-compose volume 寫法

volumes:

- type: volume
source: mydata
target: /data
volume:
nocopy: true
- type: bind
source: ./static
target: /opt/app/static

db:

image: postgres:latest

volumes:

- "/var/run/postgres/postgres.sock:/var/run/postgres/postgres.sock"
- "dbdata:/var/lib/postgresql/data"

volumes:

mydata:

dbdata:

docker compose yml 範例

```
version: "3"
services:
  webapp:
    #指定image
    image: httpd:alpine
    ports:
      - "80:80"
    volumes:
      - "/data"
```

```
version: "3"
services:
  webapp:
    #指定image
```

```
build:
context: .
  dockerfile: Dockerfile-alternate
args:
buildno: 1
```

相關連結

- [Docker 教學課程-第8部分：使用 Docker Compose | Microsoft Docs](#)
- [Compose file version 3 reference | Docker Documentation](#)
- [Overview of docker-compose CLI | Docker Documentation](#)
- 官網文件 <https://docs.docker.com/compose/>

🕒 修訂版本 #25

★ 由 treeman 建立於 18 🕒@🕒🕒 2022 22:28:26

✎ 由 treeman 更新於 12 🕒🕒🕒 2023 14:57:40