

Docker Compose

Jag använder mig av docker compose på Ubuntu.

Patching

Patcha en container går väldigt fort och kan ske med minimal nedtid.

Slå först `sudo docker compose pull` för att hämta senaste avbilderna. Sedan `sudo docker container up` för att återskapa containern med senaste avbilden. Klart!

Container name

Ge containern ett specifikt namn med variabeln `container_name`. Ex:

```
grampsweb_redis:  
  container_name: grampsweb_redis
```

User

Specifisera vilken användare som ska användas. Denna måste komma åt filtjänsterna. Specifiserar man inte en användare så används root. Det är alltså bra att specifisera användare. Om man kör numrerar UID/GID, ex 1000:1000, så behöver man inte skapa dem i förväg. Det kan vara bra att ha olika UID/GID per compose-fil.

Ex.

```
services:  
  navidrome:  
    image: deluan/navidrome:latest  
    user: "1000:1000"  
    volumes:  
      - "./data:/data"
```

Restart

Man kan ställa in så att ens container alltid startar när hosten startar genom `restart: always`. Jag gillar `restart: unless-stopped`, vilket gör att containern inte startar automatiskt om den manuellt har stängts av.

Ex:

```
services:
  server:
    restart: unless-stopped
```

Volumes

Går att konfigurera på lika olika sätt. Ex. `config:/etc/dns` placerar volymen i `/var/lib/docker/volumes`. `./data:/data` placerar filerna i en undermapp till mappen som `docker-compose.yml` filen är i. Här är ett exempel på en NFS-monterad mapp:

```
/data/docker/audiobookshelf/audiobooks:/audiobooks
```

`sudo docker volume ls` listar alla volymer som docker känner till. `inspect` ger bra information om en specifik volym, ex:

```
jehrlander@docker-mgmt:~/docker/DnsServer$ sudo docker volume inspect dnsserver_config
[
  {
    "CreatedAt": "2024-04-18T13:54:19Z",
    "Driver": "local",
    "Labels": {
      "com.docker.compose.project": "dnsserver",
      "com.docker.compose.version": "2.20.2",
      "com.docker.compose.volume": "config"
    },
    "Mountpoint": "/var/lib/docker/volumes/dnsserver_config/_data",
    "Name": "dnsserver_config",
    "Options": null,
    "Scope": "local"
```

```
}
```

För att rensa oanvända volymer, som ingen container använder, använd `sudo docker volume prune -a`.

NFS-montera i container

En NFS montering sker från hosten. Man kan antingen montera NFS direkt på hosten via fstab (se den [här artikeln för Ubuntu](#)) eller så kan man montera per compose fil. Gör man det i compose-filen så blir filen mer portabel. Å andra sidan tycker jag att det skalar trevligt med att montera direkt från hosten . Jag vet inte vad som är bäst, säkerheten har jag inte rätt ut ännu.

Montera från host

Följ artikeln för Ubuntu länkad ovan. I exemplet nedanför är då mappen monterad som /data/music. Mer konfiguration behövs ej.

```
services:
  metadata-remote:
    image: ghcr.io/wow-signal-dev/metadata-remote:latest
    container_name: metadata-remote
    ports:
      - "8338:8338"
    volumes:
      - /data/music:/music
    environment:
      - PUID=1000
      - PGID=1000
    restart: unless-stopped
    networks:
      DOCKER_NETWORK:
        ipv4_address: 10.10.0.63

networks:
  DOCKER_NETWORK:
    name: DOCKER_NETWORK
    external: true
```

Ibland kan man få behörighetsproblem:

```
jehrlander@docker-mgmt:~/docker/kiwix$ sudo docker compose up
Attaching to kiwix-serve-1
kiwix-serve-1 | '/data' directory is not writable by 'user:user' (1001:1001). ZIM file(s) can not be written.
kiwix-serve-1 exited with code 1
```

Här syns användaren som används. Då är det lätt att fixa med:

```
sudo chown -R 1001:1001 /data/docker/kiwix
```

Det går även att definiera volymer längst ned i en compose-fil och sedan hänvisa till dem längre upp. Gör man såhär så måste man dock skapa alla mappar själva, det kan inte Docker göra via den här typen av volymer.

```
services:
  # MongoDB: https://hub.docker.com/_/mongo/
  mongodb:
    image: "mongo:7.0"
    restart: "on-failure"
    networks:
      - GRAYLOG_INTERNAL
    volumes:
      - "mongodb_data:/data/db"
      - "mongodb_config:/data/configdb"

volumes:
  mongodb_data:
    driver: local
    driver_opts:
      type: none
      o: bind
      device: /data/docker/graylog/mongodb_data
  mongodb_config:
    driver: local
    driver_opts:
      type: none
      o: bind
      device: /data/docker/graylog/mongodb_config
```

Montera direkt från container

Här är ett exempel på en compose fil (för applikation Kiwix) där monteringen sker via compose-filen. Då är det docker-containern själv som pratar NFS, vilket syns definierat i driver:

```
services:
  kiwix-serve:
    ports:
      - 8080:8080
    image: ghcr.io/kiwix/kiwix-serve:latest
    # uncomment next 4 lines to use it with local zim file in /tmp/zim
    volumes:
      - kiwix-nfs:/data
    command:
      - '*.zim'
    # uncomment next 2 lines to use it with remote zim file
    environment:
      - 'DOWNLOAD=https://download.kiwix.org/zim/wikipedia/wikipedia_en_all_maxi_2025-08.zim'
    networks:
      DOCKER_NETWORK:
        ipv4_address: 10.10.0.66

networks:
  DOCKER_NETWORK:
    name: DOCKER_NETWORK
    external: true

volumes:
  kiwix-nfs:
    driver: local
    driver_opts:
      type: nfs
      o: addr=10.0.7.50,rw,soft,nfsvers=4
      device: ":/volume1/kiwix"
```

Revision #10

Created 5 January 2026 17:00:47 by Jehrlander

Updated 6 February 2026 15:12:48 by Jehrlander