

ПРАКТИЧНА РОБОТА №3

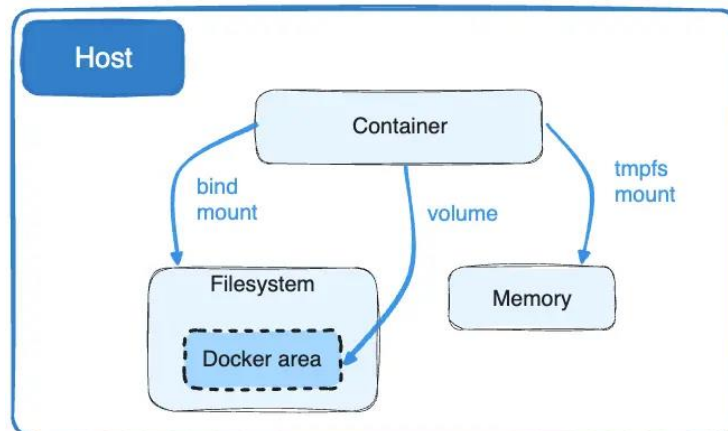
РОБОТА З DOCKER VOLUMES

Мета: Ознайомитися з механізмом використання Docker volumes та bind mounts для зберігання та обробки даних з зовнішніх джерел. Навчитися встановлювати додатки у контейнері Docker.

Стислі теоретичні відомості

Volumes

Volumes (томи) є зручним механізмом для зберігання даних, які генеруються і використовуються Docker контейнерами. У той час як прив'язка до папок залежить від структури директорій і ОС хост-машини, томи повністю управляються Docker.



Томи мають кілька переваг порівняно з прив'язкою до папок:

- Томи легше резервно копіювати або мігрувати, ніж фізичні директорії.
- Ви можете керувати томами за допомогою команд Docker CLI або Docker API.
- Томи працюють як на Linux, так і на Windows контейнерах.
- Томи можна безпечніше спільно використовувати між кількома контейнерами.
- Драйвери томів дозволяють зберігати томи на віддалених хостах або у хмарних провайдерів, шифрувати вміст томів або додавати інші функції.

- Нові томи можуть мати свій вміст, що автоматично заповнюється контейнером.
- Томи на Docker Desktop працюють набагато швидше, ніж прив'язки до папок з Mac і Windows хостів.
- Крім того, томи часто є кращим вибором, ніж зберігання даних у змінному шарі контейнера, оскільки том не збільшує розмір контейнерів, що його використовують, і вміст тому існує незалежно від життєвого циклу даного контейнера.

Bind mounts

Bind mounts (монтування прив'язкою) існують з самого початку Docker. Вони мають обмежену функціональність порівняно з томами. Коли ви використовуєте bind mount, файл або директорія на хост-машині монтується у контейнер. Файл або директорія вказуються за їхнім абсолютним шляхом на хост-машині. Натомість, коли ви використовуєте том, створюється нова директорія у директорії зберігання Docker на хост-машині, і Docker керує вмістом цієї директорії.

Файл або директорія не повинні вже існувати на хості Docker. Вони створюються за потреби, якщо ще не існують. Bind mounts дуже продуктивні, але вони покладаються на те, що файлова система хост-машини має певну структуру директорій. Якщо ви розробляєте нові додатки Docker, розгляньте можливість використання іменованих томів замість цього. Ви не можете використовувати команди Docker CLI для безпосереднього керування директоріями прив'язаними за допомогою Bind Mount.

Хід роботи

Задача 1. Створити та дослідити volume

На відміну від bind mount, томи можна створювати та керувати ними за межами будь-якого контейнера.

1. Створіть "docker volume"
`docker volume create my-vol`
2. Виведіть список доступних томів:
`docker volume ls`
3. Вивести параметри створеного тому:
`docker volume inspect my-vol`
4. Видалити створений том:
`docker volume rm my-vol`

Задача 2. Стиснення зображення за допомогою ffmpeg

1. Створити Dockerfile для контейнера на основі ubuntu

```
# Використовуйте офіційний образ Ubuntu як базовий  
FROM ubuntu:latest
```

2. Додати команду для встановлення утиліти "ffmpeg"

```
# Встановіть FFmpeg  
RUN apt-get update && \  
    apt-get install -y ffmpeg && \  
    rm -rf /var/lib/apt/lists/*
```

3. Встановіть робочу директорію

```
|  
# Створення директорії для вихідних даних  
RUN mkdir /data  
# Встановити поточну папку для команди ffmpeg  
WORKDIR /data
```

4. Встановити команду для стиснення зображення як Entry point для docker контейнера.

```
CMD ffmpeg -i /sources/image.jpg -vf "scale=iw*0.5:ih*0.5" output.jpg
```

5. Зібрати контейнер за допомогою команди docker build
docker build . -t <container_name>

6. Запустити контейнер
docker run -v /c/images:/sources <container_name>

7. Скопіювати результат з зупиненого контейнеру
docker cp <container_id>:/data/output.jpg /images