

Звіт
З практичної роботи №1
Студента групи МІТ-31
Добровольського Арсенія Михайловича
Варіант №9

Тема роботи: Основи Docker

Мета роботи: розглянути принципи контейнеризації програмного забезпечення; навчитися використовувати Docker як один із засобів контейнеризації.

Основні опції команди **docker run** включають:

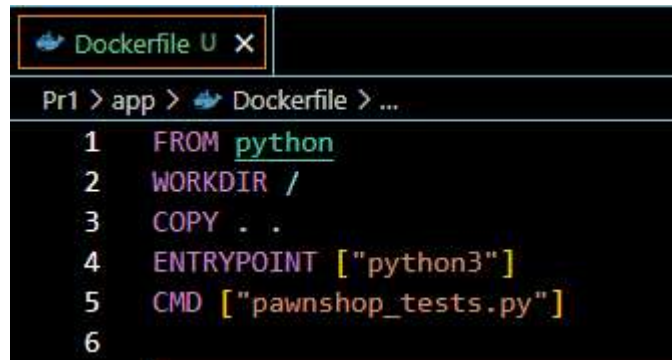
1. **-d**: Запуск контейнера у фоновому режимі (демон) без призначення терміналу. Це дозволяє продовжувати використовувати командний рядок без блокування терміналу контейнера.
2. **-it**: Запуск контейнера в інтерактивному режимі з призначенням терміналу. Це корисно, коли потрібно взаємодіяти з контейнером, наприклад, виконувати команди всередині нього.
3. **--rm**: Видалення контейнера після завершення його роботи.
4. **--name**: Надання імені контейнеру.
5. **-p**: Перенаправлення портів з контейнера на хостову систему. Наприклад, **-p 8080:80** перенаправить порт 80 контейнера на порт 8080 хостової системи.
6. **-v**: Підключення томів (папок або файлів) з хостової системи до контейнера. Це дозволяє обмінюватися даними між контейнером і хостом.
7. **--env**: Визначення змінних середовища для контейнера.
8. **--network**: Підключення контейнера до мережі Docker. Це дозволяє контейнерам спілкуватися один з одним або із зовнішніми сервісами.
9. **--link**: Підключення контейнера до іншого контейнера, що надає доступ до служб і сервісів іншого контейнера.
10. **--user**: Визначення користувача, в якому виконується процес в контейнері.

11. `--privileged`: Виконання контейнера в привілейованому режимі, що надає доступ до всіх можливостей ядра хостової системи.
12. `--entrypoint`: Визначення команди або файлу, який виконується при запуску контейнера, замість того, який заданий в образі.
13. `--restart`: Визначення політики перезапуску контейнера в разі аварійного завершення роботи.

Для отримання повного переліку опцій та подробиць щодо їх використання можна скористатися командою **docker run --help**.

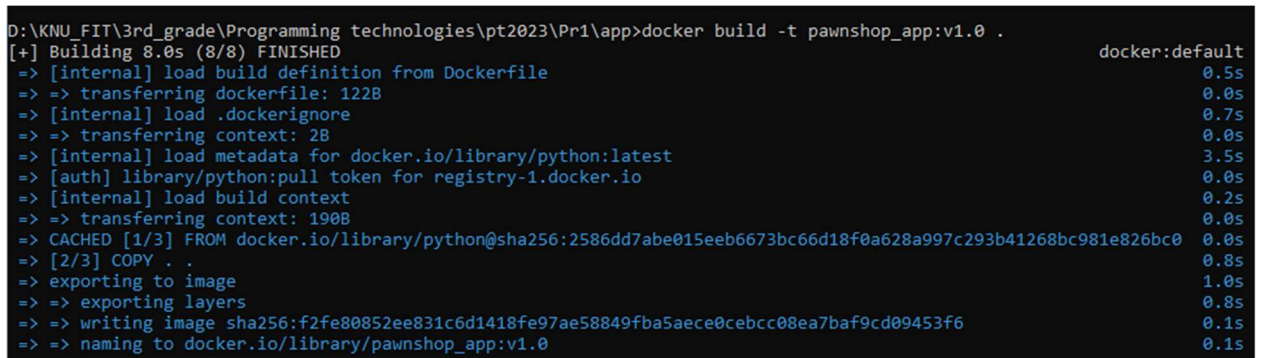
Завдання для самостійного виконання

- Вміст Dockerfile:

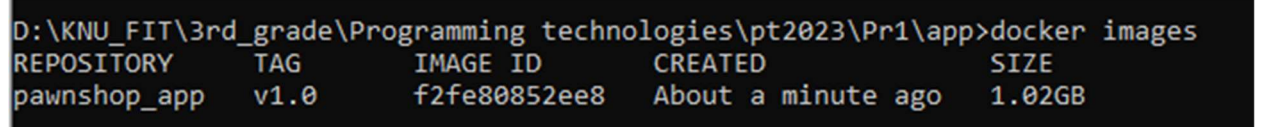


```
Dockerfile U X
Pr1 > app > Dockerfile > ...
1 FROM python
2 WORKDIR /
3 COPY . .
4 ENTRYPOINT ["python3"]
5 CMD ["pawns_shop_tests.py"]
6
```

- Створення образу з назвою `pawns_shop_app:v1.0`:



```
D:\KNU_FIT\3rd_grade\Programming technologies\pt2023\Pr1\app>docker build -t pawns_shop_app:v1.0 .
[+] Building 8.0s (8/8) FINISHED
=> [internal] load build definition from Dockerfile                                docker:default 0.5s
=> => transferring dockerfile: 122B                                              0.0s
=> [internal] load .dockerignore                                                 0.7s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/python:latest                3.5s
=> [auth] library/python:pull token for registry-1.docker.io                  0.0s
=> [internal] load build context                                                0.2s
=> => transferring context: 190B                                                0.0s
=> CACHED [1/3] FROM docker.io/library/python@sha256:2586dd7abe015eeb6673bc66d18f0a628a997c293b41268bc981e826bc0 0.0s
=> [2/3] COPY . .                                                              0.8s
=> exporting to image                                                         1.0s
=> => exporting layers                                                         0.8s
=> => writing image sha256:f2fe80852ee831c6d1418fe97ae58849fba5aece0cebcc08ea7baf9cd09453f6 0.1s
=> => naming to docker.io/library/pawns_shop_app:v1.0                         0.1s
```



```
D:\KNU_FIT\3rd_grade\Programming technologies\pt2023\Pr1\app>docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
pawns_shop_app      v1.0         f2fe80852ee8     About a minute ago  1.02GB
```

- Створення і запуск контейнера з назвою pawnshop_container в інтерактивному режимі через опцію -it (для того, щоб результат роботи програми відобразився у вікні терміналу):

```
D:\KNU_FIT\3rd_grade\Programming technologies\pt2023\Pr1\app>docker run -it --name pawnshop_container pawnshop_app:v1.0
.X....
-----
Ran 6 tests in 0.002s
OK (expected failures=1)
```

Результат запуску контейнера свідчить про те, що програма працює належним чином (тобто, файл з тестами виконується успішно).

- Видалення контейнера та образу для звільнення ресурсів комп'ютера:

```
D:\KNU_FIT\3rd_grade\Programming technologies\pt2023\Pr1\app>docker rm pawnshop_container
pawnshop_container
```

```
D:\KNU_FIT\3rd_grade\Programming technologies\pt2023\Pr1>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
```

```
D:\KNU_FIT\3rd_grade\Programming technologies\pt2023\Pr1\app>docker rmi pawnshop_app:v1.0
Untagged: pawnshop_app:v1.0
Deleted: sha256:f2fe80852ee831c6d1418fe97ae58849fba5aece0cebcc08ea7baf9cd09453f6
```

```
D:\KNU_FIT\3rd_grade\Programming technologies\pt2023\Pr1\app>docker images
REPOSITORY    TAG          IMAGE ID   CREATED    SIZE
```

Висновок: в ході виконання практичної роботи було розглянуто принципи контейнеризації програмного забезпечення. Також ми детально ознайомилися із програмою Docker та використали надані нею можливості для розгортання власного контейнера.