

Методології та технології розробки програмного заюзезпечення

Лабораторна робота №3

Виконав: Корнійчук Олег, група ІМ-13

1. Python

Перша версія Dockerfile:

[Commit](#)

```
GNU nano 7.2
FROM python:slim-bullseye
WORKDIR /app
COPY . .
RUN pip install -r requirements/backend.in
CMD uvicorn spaceship.main:app --host=0.0.0.0 --port=8080
```

Збирання зайняло 23 секунди:

`sudo docker build . -t spaceship1`

Розмір образу – 193 мб.

```
Removing intermediate container a748ca990988
--> 0043ed7e95cd
Step 5/5 : CMD uvicorn spaceship.main:app --host=0.0.0.0 --port=8080
--> Running in 5c229565e76d
Removing intermediate container 5c229565e76d
--> 44173206e364
Successfully built 44173206e364
Successfully tagged spaceship1:latest
~/Learning/MTRPZlabs/MTRPZ-Lab3/python main 71 23s
```

Запуск застосунку:

```
Removing intermediate container a748ca990988
--> 0043ed7e95cd
Step 5/5 : CMD uvicorn spaceship.main:app --host=0.0.0.0 --port=8080
--> Running in 5c229565e76d
Removing intermediate container 5c229565e76d
--> 44173206e364
Successfully built 44173206e364
Successfully tagged spaceship1:latest
~/Learning/MTRPZlabs/MTRPZ-Lab3/python main 71 23s
```

Зміна файлу index.html (додано рядок `<h2>AMOGUS!!!</h2>`):

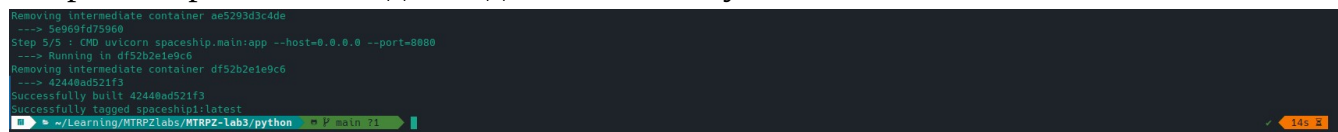
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Spaceship</title>
  </head>
  <body>
    <h1>Spaceship</h1>
    <p>Laboratory assignment 03 starter project for software engineering course</p>
    <ul>
      <li><a href='docs'>Swagger-UI documentation</a></li>
      <li><a href='redoc'>Redoc documentation</a></li>
    </ul>
    <h2>AMOGUS!!!</h2>
  </body>
</html>
```

Збирання зайняло 14 секунд:

`sudo docker build . -t spaceship2`

Розмір образу – 193 мб.

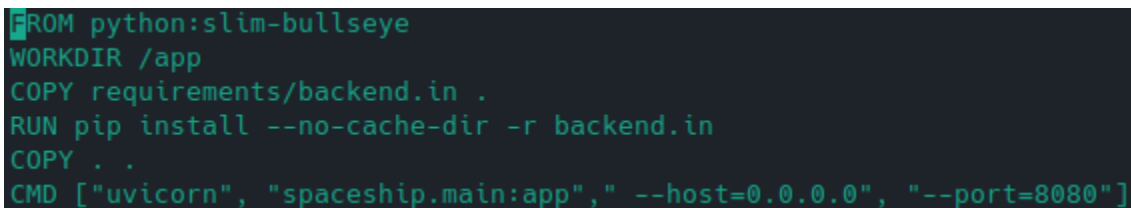
Збирання пройшло швидше, адже slim-bullseye вже завантажено.

A terminal window showing the Docker build process. It starts with removing an intermediate container, then runs 'Step 5/5 : CMD uvicorn spaceship.main:app --host=0.0.0.0 --port=8080'. It then removes another intermediate container and successfully builds and tags the image 'spaceship1:latest'. The terminal path is '~/Learning/MTRP2-Lab3/python' and the prompt is 'P: main 21'.

Оптимізація Dockerfile:

Commit

Було додано кешування залежностей. Таким чином, якщо при повторному збиранні залежності не зміняться – вони не завантажуватимуться заново.

A terminal window showing Dockerfile commands. It starts with 'FROM python:slim-bullseye', then 'WORKDIR /app', 'COPY requirements/backend.in .', 'RUN pip install --no-cache-dir -r backend.in', 'COPY . .', and finally 'CMD ["uvicorn", "spaceship.main:app", "--host=0.0.0.0", "--port=8080"]'.

Повторне збирання після внесення змін зайняло менше 2 секунд:

`sudo docker build . -t spaceship3.2`

Розмір образу – 182 мб.

```

Sending build context to Docker daemon  20.99kB
Step 1/6 : FROM python:slim-bullseye
--> 4d2191666712
Step 2/6 : WORKDIR /app
--> Using cache
--> 7bb193d9cd9f
Step 3/6 : COPY requirements/backend.in .
--> Using cache
--> 58c79e8d9f8b
Step 4/6 : RUN pip install --no-cache-dir -r backend.in
--> Using cache
--> b6e1af3d0faa
Step 5/6 : COPY . .
--> 462fda413f57
Step 6/6 : CMD ["uvicorn", "spaceship.main:app", "--host=0.0.0.0", "--port=8080"]
--> Running in d90f079a8c3c
Removing intermediate container d90f079a8c3c
--> 34967d0cbacf
Successfully built 34967d0cbacf
Successfully tagged spaceship3.2:latest
~/Learning/MTRPZlabs/MTRPZ-lab3/python  P main 11
```

Зміна образу:

[Commit](#)

У Dockerfile змінив образ slim-bullseye на alpine:

```

FROM python:alpine
WORKDIR /app
COPY requirements/backend.in .
RUN pip install --no-cache-dir -r backend.in
COPY . .
CMD ["uvicorn", "spaceship.main:app", "--host=0.0.0.0", "--port=8080"]
```

Збирання зайняло 28 секунд, адже необхідно завантажити образ alpine:

`sudo docker build . -t spaceship4`

Розмір образу – 112 мб через менший розмір alpine.

```

Removing intermediate container 604209ed9121
--> dbe236a78e9f
Step 5/6 : COPY . .
--> 96a9d9f7c5e3
Step 6/6 : CMD ["uvicorn", "spaceship.main:app", "--host=0.0.0.0", "--port=8080"]
--> Running in 3bccf84a8d7e
Removing intermediate container 3bccf84a8d7e
--> be348cf6b22e
Successfully built be348cf6b22e
Successfully tagged spaceship4:latest
~/Learning/MTRPZlabs/MTRPZ-lab3/python  P main 11 28s
```

Додавання нових залежностей:

Commit

У файлі `api.py` я додав новий роут для перемноження двох випадкових матриць. Цей код залежить від бібліотеки `numpy`, відповідну залежність було додано до файлу `backend.in`

```
@router.get(['/matrix'])
def multiplyMatrixes() -> dict:
    firstMatrix = np.random.rand(10, 10)
    secondMatrix = np.random.rand(10, 10)

    resultMatrix = bp.dot(firstMatrix, secondMatrix)

    response = {
        "matrix_a": firstMatrix.tolist(),
        "matrix_b": secondMatrix.tolist(),
        "product": resultMatrix.tolist()
    }

    return response
```

Збирання зайняло 17 секунд. Для збирання використовувався `slim-bullseye`, бо `alpine` має проблеми з підтримкою `numpy`:

`sudo docker build . -t spaceship5`

Розмір образу – 256 мб.

```
Removing intermediate container 8d8f364ba51b
--> 17cbd66b25c0
Step 5/6 : COPY . .
--> 72853615c38f
Step 6/6 : CMD ["uvicorn", "spaceship.main:app", "--host=0.0.0.0", "--port=8080"]
--> Running in 0eca7b6fc11f
Removing intermediate container 0eca7b6fc11f
--> 4bd4ef985b80
Successfully built 4bd4ef985b80
Successfully tagged spaceship5:latest
```

Спробую використати `alpine`. Для цього потрібно прописати завантаження додаткових залежностей у `Dockerfile`.

Commit

```
FROM python:alpine
WORKDIR /app
COPY requirements/backend.in .
RUN apk --no-cache add musl-dev linux-headers g++
RUN pip install --no-cache-dir -r backend.in
COPY . .
CMD ["uvicorn", "spaceship.main:app", "--host=0.0.0.0", "--port=8080"]
```

Через необхідність їх завантаження збирання образу зайняло майже 6 хвилин, розмір образу склав 443 мб. Таким чином, використання легшого образу не має сенсу, адже його переваги нівелюються необхідністю встановлення додаткових залежностей.

```
Removing intermediate container 2a4d89aafca7
--> ad9f8ade237c
Step 6/7 : COPY . .
--> c8d894bf5e42
Step 7/7 : CMD ["uvicorn", "spaceship.main:app", "--host=0.0.0.0", "--port=8080"]
--> Running in e15c8c757e1c
Removing intermediate container e15c8c757e1c
--> 02867d64424a
Successfully built 02867d64424a
Successfully tagged spaceship5.1:latest
~/Learning/MTRZ-Labs/MTRZ-Lab3/python main 11 21 sudo docker run -p 8080:8080 --rm spaceship5.1
```

Результат запиту *localhost:8080/api/matrix* :

`{"matrix_a":`

```
[[0.21462829145090723,0.7372052303155723,0.5447513287110481,0.5879522110936
927,0.5181617774643216,0.3785896216668272,0.2510766685312892,0.371564028506
8815,0.5348177025279426,0.4106137931631235],
[0.03274315009528628,0.6873703539103554,0.24614535040058716,0.7697062445614
633,0.14169633380117763,0.8486164023893144,0.7112072194580307,0.95065302666
11013,0.5356453919776712,0.10099217757638301],
[0.5647719405452843,0.6638046453582722,0.047536422333877004,0.0593536973198
5751,0.37500108183014147,0.48636590484825204,0.5278159620293512,0.655733181
0491881,0.8159002093818478,0.1296122260535525],
[0.8584975708294977,0.9696386840039758,0.1479021294257945,0.643720248501475
7,0.09367843463500336,0.5159690741918894,0.35335142301488187,0.242387500671
16333,0.6192284741287858,0.1041342621004151],
[0.44935889870503287,0.30045416156529603,0.26797191346075877,0.645351837034
841,0.41962284455709986,0.8642048873226177,0.14760823454243877,0.7543722643
8074,0.3459223183184964,0.8616004990548749],
[0.9832712640400696,0.7235508959209532,0.7831033074041109,0.032721650492488
21,0.4033168469499485,0.7700032835143752,0.9925033215870854,0.2297302893721
5778,0.519165930380672,0.7542540583347123],
[0.851346999632135,0.7141818050009524,0.7339415498884874,0.9201516976390238
,0.8587292259823218,0.9671731404732772,0.7550580071586793,0.536174677236537
1,0.5148517286702528,0.8121213517056471],
[0.0880111231146612,0.9250439023369142,0.8914234549502811,0.953329669957580
5,0.7006342768663752,0.02494828187416598,0.08476173708915014,0.393128110732
6635,0.8977338732253711,0.041700084962601025],
[0.7694834163705997,0.35679226776528383,0.8507206488136793,0.98577927091796
26,0.9767890221098032,0.8907106027570961,0.4448759671624206,0.3666932612517
9624,0.4914691650744376,0.9357665419922528],
[0.9386067106910274,0.14497987894916364,0.20150872634449257,0.8159257889796
```

219,0.9276247723342356,0.905958123327062,0.10964511487778306,0.258041147144
01194,0.9254081773705556,0.29430248795871095]], "matrix_b":
[[0.31384663027965065,0.5699415067950254,0.4050485329798631,0.5823803401413
861,0.2451533037860013,0.3502294195988478,0.023399028352649953,0.2656610128
4955723,0.4630153675843751,0.7646725614696339],
[0.05990267514818415,0.4212644058496684,0.4877577706542465,0.31945072470436
85,0.7095284023925988,0.6767852613612098,0.050221384463785324,0.58978287722
32511,0.13930181421271404,0.4476424690310229],
[0.6499409148873025,0.7310477700748387,0.2008171598019579,0.754721502001720
4,0.20845686506481775,0.40804799162100813,0.6907494475281862,0.964612070342
0383,0.6649742876880202,0.1692412921726496],
[0.9028983543072602,0.15893911553508744,0.10948903460200021,0.5254611937923
801,0.10677548521214475,0.8356776986230768,0.7371852144439893,0.79898953021
82619,0.2301351795553389,0.3578504870419811],
[0.9610732023378616,0.09781385478886095,0.5242302991342446,0.90589706923288
06,0.15343133878821114,0.3682463819306605,0.24514780471950226,0.23864854192
85156,0.7329975629228593,0.24869195818839185],
[0.979969911361611,0.3182449972641008,0.2503698013549236,0.9251538931874479
,0.41004043830522663,0.8754877977914056,0.5436411910513522,0.69987984933363
17,0.5598508691454294,0.75315183364718],
[0.0032724185436628916,0.5417543159904576,0.3252070144503755,0.926414782666
6174,0.5725961165247022,0.0038610945051213985,0.7838452262180743,0.14846027
946371165,0.1402191085614951,0.2726723677659938],
[0.22664686904294618,0.807015007068274,0.7041083364871866,0.235244994531709
33,0.6442413990052033,0.24392361097403215,0.0827530450442896,0.913872242112
2528,0.17841259115118702,0.10363450065001256],
[0.7350545394462135,0.5765138504626756,0.8692585193951283,0.394020914384039
1,0.7659779033352392,0.690441772002745,0.34566054364435606,0.90408560771998
09,0.023772934783721333,0.41581823761827363],
[0.44423164041981755,0.8688028902389848,0.6200546513273408,0.85242380370993
35,0.5866331341150581,0.9360105326869447,0.1157243562896707,0.6920893521589
662,0.7339584236095162,0.7052787682664591]], "product":
[[2.525999299615273,2.1966912768110163,2.049476761205572,2.780988598981616,
2.020440884251135,2.655185794432372,1.6445417495549772,3.020215945025047,1.
7069744730099239,1.8296689406444906],
[2.5305795733566487,2.4434746954903903,2.197874119185603,2.921987808284105,
2.488162445799796,2.7144621184445037,2.101791602068432,3.422791329556596,1.
3868822651692165,1.9106131704179417],
[1.9461852207554298,2.2352914185659234,2.309890416675582,2.4729062175734917
,2.308329567005182,2.1265876065099962,1.2444957012695277,2.5696697823371046
,1.251931672229287,1.8604211054528852],
[2.1580472370028083,2.116077086334856,1.9875737084048561,2.538938553683456,

2.117843503190803,2.6269238006917224,1.4720809257286078,2.7462877501247904,
1.3205382837877593,2.2102069243702034],
[2.974544985405154,2.6339718095426368,2.303481490910139,3.2636453733222566,
2.2077831640575685,3.1503841181557983,1.6565445342469676,3.396041882705522
8,2.1639243965415194,2.3795711945544653],
[2.804645603299634,3.405092893862975,2.7197420784280033,4.310749655923262,2.
855228531967078,3.127908419064568,2.20559377032305,3.4533237908314334,2.557
140823242066,2.942520831120179],
[4.254105385780358,3.704814900418205,3.2078160761859356,5.154846643419308,3.
1435752249109714,4.2622506195288015,2.885555431604275,4.602003681503847,3.1
742409715485715,3.4144833584491585],
[2.9887628456761126,2.236480659073234,2.2543674530373337,2.7385376393595315
,2.097176383280291,2.8522331991234613,1.9664785503875806,3.5875416546200736
,1.6432818203814836,1.6329938465456124],
[4.379004017452535,3.3797443292060927,2.9098558161193537,4.920791406460747,
2.6558680929744143,4.127782403742535,2.731176511312678,4.372715577118551,3.
2395035887358343,3.182262351672263],
[3.820064047707969,2.308917285218793,2.4980600900294436,3.63002394400359,2.0
864157669839782,3.303450262805194,1.9510789243431745,3.3290163250992046,2.2
63122369553476,2.670731881859755]]}]}

2. Golang

Початковий Dockerfile:
[Commit](#)

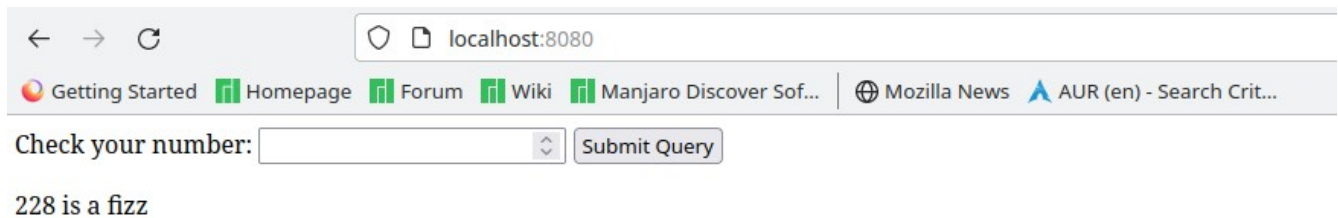
```
FROM golang:latest
WORKDIR /app
COPY go.mod go.sum ./
RUN go mod download
COPY . .
RUN go build -o build/fizzbuzz
EXPOSE 8080
CMD ["/build/fizzbuzz", "serve"]
```

Збирання зайняло 19 секунд:
`sudo docker build . -t fizzbuzz1`

Розмір файлу – 860 мб.

```
Sending build context to Docker daemon 93.7kB
Step 1/8 : FROM golang:latest
----> 74b93dfc9812
Step 2/8 : WORKDIR /app
----> Using cache
----> 7389e83ffc49
Step 3/8 : COPY go.mod go.sum ./
----> Using cache
----> 5f995eddf63b
Step 4/8 : RUN go mod download
----> Using cache
----> faef5abc1548
Step 5/8 : COPY . .
----> 64323eb834f9
Step 6/8 : RUN go build -o build/fizzbuzz
----> Running in 06d61fd9add4
Removing intermediate container 06d61fd9add4
----> 897d723f5d4
Step 7/8 : EXPOSE 8080
----> Running in b2fb8894a5ae
Removing intermediate container b2fb8894a5ae
----> 076e7526cc7d
Step 8/8 : CMD ["/build/fizzbuzz", "serve"]
----> Running in f78f8f15c8cc
Removing intermediate container f78f8f15c8cc
----> 65ee9c45eb6d
Successfully built 65ee9c45eb6d
Successfully tagged fizzbuzz:latest
~/Learning/MTRPZLabz/MTRPZ-Lab3/golang
```

Запуск застосунку:



Для запуску використовується лише бінарний файл, зібраний командою go build, адже go – компільована мова програмування.

Багатоетапне збирання:

[Commit](#)

Внесемо зміни в Dockerfile для поетапного збирання:

```
FROM golang AS builder
WORKDIR /app
COPY go.mod go.sum ./
RUN go mod download
COPY . .
RUN CGO_ENABLED=0 go build -ldflags "-w -s -extldflags '-static'" -o build/fizzbuzz
FROM scratch
COPY --from=builder /app/build/fizzbuzz /
COPY --from=builder /app/templates/index.html /templates/
EXPOSE 8080
CMD ["/fizzbuzz", "serve"]
```


Збирання образу зайняло 19 секунд:

```
sudo docker build . -t fizzbuzz2
```

Розмір – 6.83 мб

```
Sending build context to Docker daemon 93.7kB
Step 1/11 : FROM golang AS builder
----> 74b93dfc9812
Step 2/11 : WORKDIR /app
----> Using cache
----> 73b5ee5ffc49
Step 3/11 : COPY go.mod go.sum ./
----> Using cache
----> 5f995eddff63b
Step 4/11 : RUN go mod download
----> Using cache
----> Tae5a0c1548
Step 5/11 : COPY . .
----> 04b950a12eeb
Step 6/11 : RUN CGO_ENABLED=0 go build -ldflags "-w -s -extldflags '-static'" -o build/fizzbuzz
----> Running in 81874b491feb
Removing intermediate container 81874b491feb
----> 523cc20471bf
Step 7/11 : FROM scratch
---->
Step 8/11 : COPY --from=builder /app/build/fizzbuzz /
----> f0b4cdf0d62b
Step 9/11 : COPY --from=builder /app/templates/index.html /templates/
----> 5a163c454012
Step 10/11 : EXPOSE 8080
----> Running in 4f0da11fcea
Removing intermediate container 4f0da11fcea
----> 032e32974ee2
Step 11/11 : CMD ["/fizzbuzz", "serve"]
----> Running in 7f93656736c9
Removing intermediate container 7f93656736c9
----> 0c3830145d0d
Successfully built 0c3830145d0d
Successfully tagged fizzbuzz2:latest
```

В результаті отримано образ, який містить лише скомпільований бінарний файл без залежностей та соурс коду. Через свій малий розмір він може бути зручним для швидкої демонстрації роботи застосунку, адже швидко розгортатиметься, але робота з ним може бути менш зручною, ніж із звичайним образом.

Збірка на основі distroless:

[Commit](#)

Вносимо зміни в Dockerfile:

```
FROM golang AS builder
WORKDIR /app
COPY go.mod go.sum ./
RUN go mod download
COPY . .
RUN go build -o build/fizzbuzz
FROM gcr.io/distroless/base
COPY --from=builder /app/build/fizzbuzz /
COPY --from=builder /app/templates/index.html /templates/
EXPOSE 8080
CMD ["/fizzbuzz", "serve"]
```

Основна відмінність від попереднього метода – при компіляції будуть використовуватися динамічні бібліотеки.

Збирання зайняло 24 секнуди:

`sudo docker build . -t fizzbuzz3`

Розмір образу – 30.3 мб

```
Digest: sha256:711a15d0b7fe9c4f7b01477e8e0b451619fcae0bc94fd6109d490b16cea0
Status: Downloaded newer image for gcr.io/distroless/base:latest
--> a955ac8d0d9c
Step 8/11 : COPY --from=builder /app/build/fizzbuzz /
--> c0c9c5d2d682
Step 9/11 : COPY --from=builder /app/templates/index.html /templates/
--> 938fe0857a67
Step 10/11 : EXPOSE 8080
--> Running in 59e77f27e784
Removing intermediate container 59e77f27e784
--> b6bc9e9c2de3
Step 11/11 : CMD ["/fizzbuzz", "serve"]
--> Running in 95dd28d0fde9
Removing intermediate container 95dd28d0fde9
--> e2ee02b9af1a
Successfully built e2ee02b9af1a
Successfully tagged fizzbuzz3:latest
~/Learning/MTRP2labs/MTRP2-lab3/golang > P main *1 11
```

В результаті отримано компактний образ який містить динамічні залежності, але все одно менший за класичний образ через відсутність стандартних для дистрибутивів лінукс утиліт. Він поступається from scratch збірці у компактності та швидкості розгортання, але не потребує встановлення залежностей.

3. Javascript

Commit

Я створив простий express додаток із примітивним функціоналом:

```
const express = require("express");
const app = express();
const PORT = 8080;

app.get('/', (req, res) => {
  res.json('Hello, user!');
})

app.get('/greeting/:name', (req, res) => {
  res.json(`Hello, ${req.params.name}!`);
});

app.listen(PORT, () => {
  console.log('Server listening on port '+PORT)
});
```

Dockerfile:

```
FROM node:9-slim
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . /app
EXPOSE 8080
CMD ["npm", "start"]
```

Залежності знаходяться у файлах `package.json` та `package-lock.json`

Збирання зайняло 7 секунд:

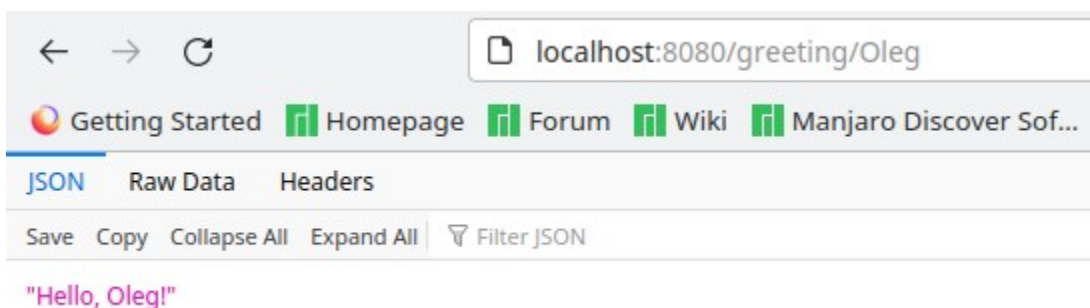
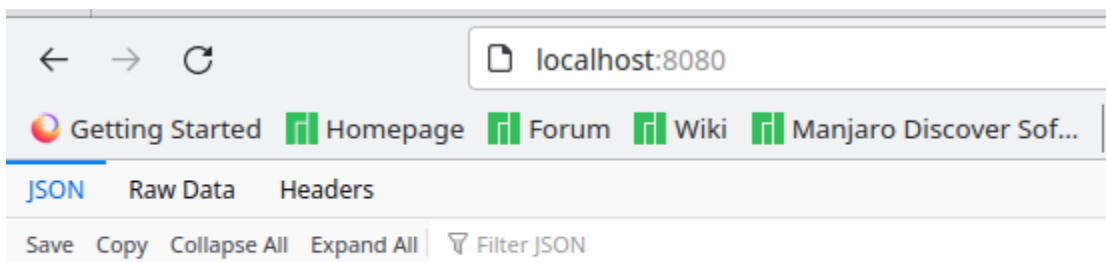
`sudo docker build . -t myapp`

Розмір образу – 185 мб.

```
auded 5/ packages in 1.135s
Removing intermediate container 7bbe2641181f
----> 2bffaaf1d387
Step 5/7 : COPY . /app
----> 944e5428f0e7
Step 6/7 : EXPOSE 8080
----> Running in d9db858b71ab
Removing intermediate container d9db858b71ab
----> 3a8d85c4eb4c
Step 7/7 : CMD ["npm", "start"]
----> Running in 51a88e917344
Removing intermediate container 51a88e917344
----> d0233a0be5c1
Successfully built d0233a0be5c1
Successfully tagged myapp:latest
```

Запуск контейнера:

```
~/Learning/MTRPZlabs/MTRPZ-lab3/js P main ?2 sudo docker run -p 8080:8080 --rm myapp
> mtrpz-lab3@1.0.0 start /app
> node index.js
Server listening on port 8080
```



Висновок:

Docker – це дуже потужний і цікавий інструмент, який значно полегшує розробку та тестування застосунку різними людьми на різних пристроях. Проте користування ним вимагає багато досвіду, адже часом виникають абсолютно непередбачувані помилки. Також потрібно гарне знання лінуксу, адже докер

емулює його дистрибутиви, і помилки, які виникають, часто пов'язані із конфліктом або нестачею залежностей. Цікавий момент, який я помітив – докер краще оптимізований для роботи на лінукс в якості хост-системи, ніж на віндоус – тут він швидше збирає та розгортає образи. В залежності від задачі можна розглядати різні типи образів, такі як `from scratch` і `distroless`. Вони є більш компактними, ніж класичні, але мають менше вбудованих бібліотек: `from scratch` образ взагалі містить лише бінарний файл із статичними бібліотеками, `distroless` має динамічні залежності, але не має звичних інструментів та бібліотек оточення. Ці недоліки компенсуються високою швидкістю розгортання та малим розміром, що може бути важливим для певних задач, хоча у моєму випадку розмір файлів був дуже малим, і переваги цих образів не були відчутні. Також розмір образів можна зменшити, використовуючи більш базовий дистрибутив лінукс, такий як `alpine`. Проте це не завжди доцільно, адже у таких дистрибутивах можуть бути відсутніми необхідні бібліотеки, і їх все одно прийдеться завантажувати, що нівелює перевагу у розмірі та швидкості розгортання.