

Міністерство освіти і науки України Харківський національний
університет радіоелектроніки

Кафедра «Штучного інтелекту»

Звіт з

практичного проєкту №1

з дисципліни «Адміністрування unіх-подібних операційних систем»

Виконала:

Ст.гр ІТШ-22-2

Отман А.Ш

Перевірив:

Ст. викладач Бібічков І. Є.

Харків - 2025

Мета роботи:

Ознайомитись з основами контейнеризації застосунків у середовищі Docker.
Навчитись створювати, запускати та автоматизувати контейнеризацію простого API-додатку за допомогою bash-скрипту.

Хід роботи

Базовий рівень

1. Створення проекту:

Реалізовано простий API-додаток на базі Flask (Python), який при зверненні до головної сторінки (/) повертає повідомлення "hiiii :)".

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return 'hiiii :)'

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

- Створено файл `requirements.txt` який включає список потрібних бібліотек:

```
flask
```

2. Запуск API локально:

```
hiiii :)
```

3. Створення Dockerfile:

```
FROM python:3.10
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt
CMD ["python", "main.py"]
```

- Файл `Dockerfile` створено в корені проекту.
- Збірка контейнера командою:

```
docker build -t lab1_docker_api .
```

4. Запуск контейнера:

```
docker run -d -p 5000:5000 lab1_docker_api
```

- У браузері: `http://localhost:5000` — додаток успішно працює в контейнері.



```
hiii :)
```

5. Автоматизація за допомогою bash-скрипта:

- Створено файл `deploy.sh`:

```
docker build -t lab1_docker_api .
```

```
docker rm -f lab1_api_container 2>/dev/null
```

```
docker run -d -p 5000:5000 --name lab1_api_container lab1_docker_api
```

- Команда для запуску:

```
./deploy.sh
```

Висновок:

У ході виконання лабораторної роботи було створено простий клієнт-серверний API-додаток на Flask. Застосунок було контейнеризовано за допомогою Docker, автоматизовано запуск через bash-скрипт `deploy.sh`. Отримані навички дозволяють створювати переносимі, масштабовані та зручні у розгортанні сервіси.