

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА СИСТЕМНОГО ПРОЕКТУВАННЯ

**«Встановлення та налаштування платформи Docker для
створення, розгортання і запуску додатків на прикладі
Python Flask application»**

Виконав: ст. гр. ДА-52с
Мельничук М.М
Керівник: доц., к. т. н.
Харченко К.В

КИЇВ
2017

Ціль роботи

- Розробити методику розгортання Flask додатку за допомогою Docker платформи
- Створити Flask застосунок та перевірити його роботу в контейнері Docker

Для досягнення даних цілей потрібно:

- Проаналізувати роботу Flask фреймворку та Docker платформи
- Створити тестовий Flask додаток
- Налаштувати систему Docker для роботи з додатком
- Розгорнути та перевірити роботу додатку на тестовому сервері

Актуальність роботи

Система Docker широко використовується в світі та надає наступні переваги:

- **Проста передача продукту клієнту** за рахунок простоти запуску сформованих образів
- **Тиражування на велику кількість серверів** займає малу кількість часу
- **Дешевше розгортання додатків** – в порівнянні з віртуальними машинами екземпляр контейнера займає набагато менше місця на диску

Порівняння підходів КОНТЕЙНЕРИЗАЦІЇ та ВІРТУАЛІЗАЦІЇ

	Контейнеризація (Docker)	Віртуалізація (Vagrant)
Гарантовані апаратні ресурси	НІ	ТАК
Підтримуючі ОС	Linux	Linux, Unix, Windows
Час запуску	Декілька секунд	Декілька хвилин
Рівень ізоляції	Часткова	Повна
Вага створеної віртуальної системи	Легкі	Важкі

Вибір контейнерної системи для запуску Flask додатку

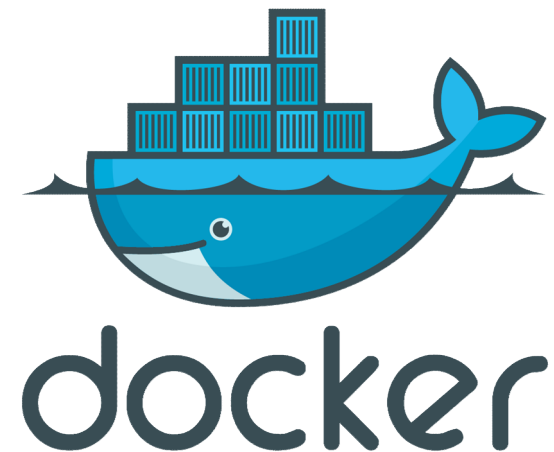
- **LXD lightervisor** - маніпулює контейнерами з повноцінною Linux системою всередині так само, як ви б працювали з системою, на фізично му обладнанні або в віртуальній машині.
- **Docker** - це відкрита платформа для розробки, доставки і експлуатації додатків. Docker розроблений для більш швидкого викладання додатків.
- **rkt** - позиціонується як більш безпечна, переносима і адаптована для серверного застосування альтернатива інструментарію Docker. Не надто популярна та не має розвинутої інфраструктури.



Архітектура системи контейнеризації Docker

Основні компоненти

- Образ (Image) – шаблон контейнера
- Реєстр (Registry) – сховище образів
- Контейнер – середовище запуску



Мікрофреймворк для розробки веб-застосунків Flask

Flask – веб мікрофреймворк Python з великою кількістю розширень.



Переваги:

- Швидкий старт – встановлення та налаштування базового додатку легко та швидко
- Легка вага – додаток не містить невикористаних бібліотек та займає невелику кількість пам`яті
- Гнучкість – містить багато плагінів які можна легко додавати в образ Docker за допомогою requirements.txt

Тестовий додаток Flask для встановлення на платформу Docker

Додаток містить:

- Змінна *container_number* передається додатку під час його запуску для ідентифікації контейнера запуску

REST api:

- / - привітання, використовується для перевірки роботи додатку
- /connect – надсилає get запит до іншого додатку із вказуванням *container_number*, використовується для з'єднання додатків між собою в контексті контейнерів
- /hello – відповідь на /connect з додаванням власного *container_number*, для перевірки з'єднання двох різних контейнерів

Файл налаштування для створення образів Dockerfile

Інструкції dockerfile:

- FROM – базовий образ контейнера (Ubuntu, Linux)
- RUN – запуск команди при створенні образу (apt-get update)
- COPY – копіює файли в контекст образу
- EXPOSE – позначає вихідні порти
- ENV – запис змінної оточення (NUMBER 1)
- WORKDIR – поточна робоча папка (/app)

- CMD – основна команда запуску контейнера (python3 hello.py)

Інструменти роботи з Docker клієнтом

- *docker build* - команда для будування та перебудовування образів. Використовує концепцію слоїв (схожі на git) для забезпечення легкості зміни образів
- *docker run* – команда для запуску контейнерів на основі образу. Визначає режим роботи, параметри мережі контейнера, перезапис змінних оточення образу.
 - *-p* – публікує порт додатку
 - *-e* – перезаписує змінні оточення
 - *-d* – запуск у фоновому режимі
 - *--net* – встановлює підмережу контейнера
- *docker push* – записує образ в реєстр (Docker Hub та ін.)

Приклад розгортання Flask додатку у Docker контейнері

- Створення підмережі.

```
docker network create --subnet=172.18.0.0/16 flasknet
```

- З'єднання хоста з контейнером

```
docker run -t --net flasknet -p 5001:5000 diploma:latest
```



Приклад розгортання Flask додатку у Docker контейнері

- Міжконтейнерне з'єднання
 - Через внутрішню мережу для контейнеру 2 та 3
- ```
docker run -t -e "container_number=2" --net flasknet --ip 172.18.0.2 -p 5002:5000 -d diploma:latest
```



- Через хост

# Приклад розгортання Flask додатку у Docker контейнері

- З'єднання контейнера з хостом

```
docker run -t -e "container_number=2" --net flasknet --ip 172.18.0.2
-p 5002:5000 -d diploma:latest
```



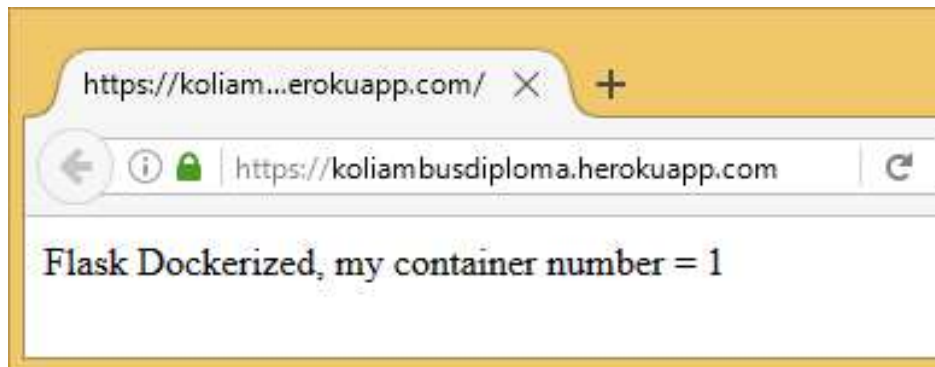
# Розгортання Docker контейнера у хмарний сервер HEROKU

- **Heroku** — хмарна PaaS-платформа, що підтримує Docker контейнери. Використовується як реєстр для розгортання контейнерів.



Створений додаток розгорнутий за адресою:

<http://koliambusdiploma.herokuapp.com/>



# Висновки

В результаті даної роботи було:

- Створено Flask додаток з використанням мережевих технологій
- Налаштована Docker платформа для запуску Flask додатку
- Зібрано образ для швидкого запуску контейнерів
- На прикладі додатку були перевірені можливості мережевого налаштування та командного рядку Docker
- Створений образ був розгорнутий як на локальному, так і віддаленому сервері
- Пояснювальна записка до представленого проекту являє собою детальну інструкцію зі встановлення та налаштування платформи Docker для розгортання і запуску Python Flask додатку яка може бути використана для навчальних цілей.