Лаб 1. ІАСА

**Завдання 1 (Task 1)**

Створив файл docker-compose.yaml

services:

    task\_1:

        image: hello-world:latest

Задав image, який буде використовуватися – hello-world останньої версії. Далі запустив сервіси, використавши команду – docker-compose up:

[+] Running 2/2

- task\_1 Pulled 4.3s

- 719385e32844 Pull complete 0.9s

[+] Running 2/2

- Network task\_1\_default Created 0.1s

- Container task\_1-task\_1-1 Created 0.1s

Attaching to task\_1-task\_1-1

task\_1-task\_1-1 |

task\_1-task\_1-1 | Hello from Docker!

task\_1-task\_1-1 | This message shows that your installation appears to be working correctly.

task\_1-task\_1-1 |

task\_1-task\_1-1 | To generate this message, Docker took the following steps:

task\_1-task\_1-1 | 1. The Docker client contacted the Docker daemon.

task\_1-task\_1-1 | 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.

task\_1-task\_1-1 | (amd64)

task\_1-task\_1-1 | 3. The Docker daemon created a new container from that image which runs the

task\_1-task\_1-1 | executable that produces the output you are currently reading.

task\_1-task\_1-1 | 4. The Docker daemon streamed that output to the Docker client, which sent it

task\_1-task\_1-1 | to your terminal.

task\_1-task\_1-1 |

task\_1-task\_1-1 | To try something more ambitious, you can run an Ubuntu container with:

task\_1-task\_1-1 | $ docker run -it ubuntu bash

task\_1-task\_1-1 |

task\_1-task\_1-1 | Share images, automate workflows, and more with a free Docker ID:

task\_1-task\_1-1 | https://hub.docker.com/

task\_1-task\_1-1 |

task\_1-task\_1-1 | For more examples and ideas, visit:

task\_1-task\_1-1 | https://docs.docker.com/get-started/

task\_1-task\_1-1 |

task\_1-task\_1-1 exited with code 0

Цей результат свідчить про коректну роботу.

**Завдання 2 (Task 2)**

Створив docker-compose.yaml

version: '3'

services:

    task\_2:

        image: node:latest

        working\_dir: /lite\_servak

        volumes:

        - ./:/lite\_servak

        ports:

        - 7000:3000

        command: bash script\_pipe.sh

Я також створив index.html, як головну стоірнку сайту. Для використання команд створив файл script\_pipe.sh:

#!/bin/bash

npm install --global lite-server

lite-server

Який буде виконовуватися при кожному виклику команди docker-compose up.

**Завдання 3 (Task 3)**

Створив docker-compose.yaml

services:

    task\_3:

        image: node:latest

        working\_dir: /json\_server

        volumes:

        - ./:/json\_server

        ports:

        - 3000:3000

        command: bash script\_pipe.sh

script\_pipe.sh

#!/bin/bash

npm install -g json-server

json-server -H 0.0.0.0 -p 3000 -w db.json

db.json

{

    "posts": [

        {

            "id": 1,

            "title": "json-server",

            "author": "typicode"

        }

    ],

    "comments": [

        {

            "id": 1,

            "body": "some comment",

            "postId": 1

        }

    ],

    "profile": {

        "name": "typicode"

    }

}

**Завдання 4 (Task 4)**

Створив папки для json-server та lite-server. А також templates, який містить зразок сценарію для автоматичного створення файлу конфігурації nginx для налаштування всього веб-сайту.

Далі запустив контейнер командою docker-compose up --scale json\_server=3 --scale lite\_server=3, де масштабував json-server та lite-server до 3.