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 config, який конфігурації nginx для налаштування всього веб-сайту.

default.conf:

server {

    listen 80;

    server\_name localhost;

    location /json\_server/posts {

        proxy\_pass http://localhost:8000/json\_server/posts;

    }

    location /json\_server/comments {

        proxy\_pass http://localhost:8000/json\_server/comments;

    }

    location /json\_server/profile {

        proxy\_pass http://localhost:8000/json\_server/profile;

    }

    location / {

        proxy\_pass http://localhost:3000/lite\_server;

    }

}

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

Результат виконання команди docker-compose up

[+] Running 3/0

- Container task\_4-json\_server-1 Created 0.0s

- Container task\_4-lite\_server-1 Created 0.0s

- Container task\_4-nginx-1 Created 0.0s

Attaching to task\_4-json\_server-1, task\_4-lite\_server-1, task\_4-nginx-1

task\_4-nginx-1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration

task\_4-nginx-1 | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/

task\_4-nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh

task\_4-nginx-1 | 10-listen-on-ipv6-by-default.sh: info: IPv6 listen already enabled

task\_4-nginx-1 | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh

task\_4-nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh

task\_4-nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh

task\_4-nginx-1 | /docker-entrypoint.sh: Configuration complete; ready for start up

task\_4-nginx-1 | 2023/11/30 10:06:48 [notice] 1#1: using the "epoll" event method

task\_4-nginx-1 | 2023/11/30 10:06:48 [notice] 1#1: nginx/1.25.3

task\_4-nginx-1 | 2023/11/30 10:06:48 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)

task\_4-nginx-1 | 2023/11/30 10:06:48 [notice] 1#1: OS: Linux 5.10.102.1-microsoft-standard-WSL2

task\_4-nginx-1 | 2023/11/30 10:06:48 [notice] 1#1: getrlimit(RLIMIT\_NOFILE): 1048576:1048576

task\_4-nginx-1 | 2023/11/30 10:06:48 [notice] 1#1: start worker processes

task\_4-nginx-1 | 2023/11/30 10:06:48 [notice] 1#1: start worker process 25

task\_4-nginx-1 | 2023/11/30 10:06:48 [notice] 1#1: start worker process 26

task\_4-nginx-1 | 2023/11/30 10:06:48 [notice] 1#1: start worker process 27

task\_4-nginx-1 | 2023/11/30 10:06:48 [notice] 1#1: start worker process 28

task\_4-json\_server-1 |

task\_4-json\_server-1 | changed 116 packages in 6s

task\_4-json\_server-1 |

task\_4-json\_server-1 | 15 packages are looking for funding

task\_4-json\_server-1 | run `npm fund` for details

task\_4-json\_server-1 |

task\_4-json\_server-1 | \{^\_^}/ hi!

task\_4-json\_server-1 |

task\_4-json\_server-1 | Loading db.json

task\_4-json\_server-1 | Done

task\_4-json\_server-1 |

task\_4-json\_server-1 | Resources

task\_4-json\_server-1 | http://0.0.0.0:8000/posts

task\_4-json\_server-1 | http://0.0.0.0:8000/comments

task\_4-json\_server-1 | http://0.0.0.0:8000/profile

task\_4-json\_server-1 |

task\_4-json\_server-1 | Home

task\_4-json\_server-1 | http://0.0.0.0:8000

task\_4-json\_server-1 |

task\_4-json\_server-1 | Type s + enter at any time to create a snapshot of the database

task\_4-json\_server-1 | Watching...

task\_4-json\_server-1 |

task\_4-json\_server-1 | GET /posts 200 16.844 ms - 77

task\_4-lite\_server-1 |

task\_4-lite\_server-1 | changed 157 packages in 11s

task\_4-lite\_server-1 |

task\_4-lite\_server-1 | 8 packages are looking for funding

task\_4-lite\_server-1 | run `npm fund` for details

task\_4-lite\_server-1 | Did not detect a `bs-config.json` or `bs-config.js` override file. Using lite-server defaults...

task\_4-lite\_server-1 | \*\* browser-sync config \*\*

task\_4-lite\_server-1 | {

task\_4-lite\_server-1 | injectChanges: false,

task\_4-lite\_server-1 | files: [ './\*\*/\*.{html,htm,css,js}' ],

task\_4-lite\_server-1 | watchOptions: { ignored: 'node\_modules' },

task\_4-lite\_server-1 | server: {

task\_4-lite\_server-1 | baseDir: './',

task\_4-lite\_server-1 | middleware: [ [Function (anonymous)], [Function (anonymous)] ]

task\_4-lite\_server-1 | }

task\_4-lite\_server-1 | }

task\_4-lite\_server-1 | [Browsersync] Access URLs:

task\_4-lite\_server-1 | -----------------------------------

task\_4-lite\_server-1 | Local: http://localhost:3000

task\_4-lite\_server-1 | External: http://172.29.0.2:3000

task\_4-lite\_server-1 | -----------------------------------

task\_4-lite\_server-1 | UI: http://localhost:3001

task\_4-lite\_server-1 | UI External: http://localhost:3001

task\_4-lite\_server-1 | -----------------------------------

task\_4-lite\_server-1 | [Browsersync] Serving files from: ./

task\_4-lite\_server-1 | [Browsersync] Watching files...

task\_4-lite\_server-1 | [Browsersync] Couldn't open browser (if you are using BrowserSync in a headless environment, you might want to set the open option to false)

task\_4-lite\_server-1 | 23.11.30 10:07:06 200 GET /index.html

**Висновки**

В результаті виконання лабораторної роботи я опанував навчики створення контейнерів для своїх сервісів в докер. Це дає змогу автоматизувати роботу із сервісами. Також налаштував nginx. Це HTTP і зворотний проксі-сервер, поштовий проксі-сервер і загальний проксі-сервер TCP/UDP.