УДК 004.738.5:004.722 DOI: https://doi.org/10.30837/IYF.IRTTPI.2024.77

СТВОРЕННЯ DOCKER ОБРАЗІВ IOT СИСТЕМ У PYCHARM ТА VISUAL STUDIO CODE

Бойко В.С.
Науковий керівник – к.т.н., проф. Зубков О.В.
Харківський національний університет радіоелектроніки, каф. МІРЕС,
м. Харків, Україна
e-mail: valerii.boiko@nure.ua

The work analyzed the capabilities of the PyCharm and Visual Studio Code environments for creating Docker images and containers. A test application for a room heating control system with a web interface was created in Python. Created Docker images and containers in PyCharm and Visual Studio Code. The complexity of creating containers and debugging them in both environments was analyzed. Conclusions are drawn on the advantages of using PyCharm when creating and debugging Docker containers.

За останні роки Інтернет речей (IoT) став неабияк важливою та широко використовуваною технологією, що перетворює наше життя та робочі процеси. Він дозволяє підключати до мережі різні пристрої - від домашніх побутових до промислових, та забезпечувати їхню взаємодію та обмін даними в реальному часі. Проте зі зростанням застосування IoT виникає потреба в ефективному управлінні та масштабуванні цих складних систем.

В цьому контексті Docker виходить на передній план як потужний інструмент для контейнеризації, що дозволяє ефективно керувати та управляти розподіленими мережами пристроїв IoT. Docker забезпечує ізольоване середовище для виконання додатків та компонентів, що робить розгортання, управління та моніторинг систем IoT більш простими та ефективними. Технологія Docker передбачає створення образу та контейнерів на базі операційної системи Linux, але створювати їх можна не тільки у Linux, а і у Windows. Найбільш поширеними середовищами написання програм на Python та створення Docker є: PyCharm та Visual Studio Code. PyCharm розроблений спеціально для створення програм мовою Python і має розширення для компіляції образів Docker. PyCharm надає багато функцій, таких як автодоповнення коду, відлагодження, контекстна довідка та інші, що полегшують розробку та налагодження додатків.

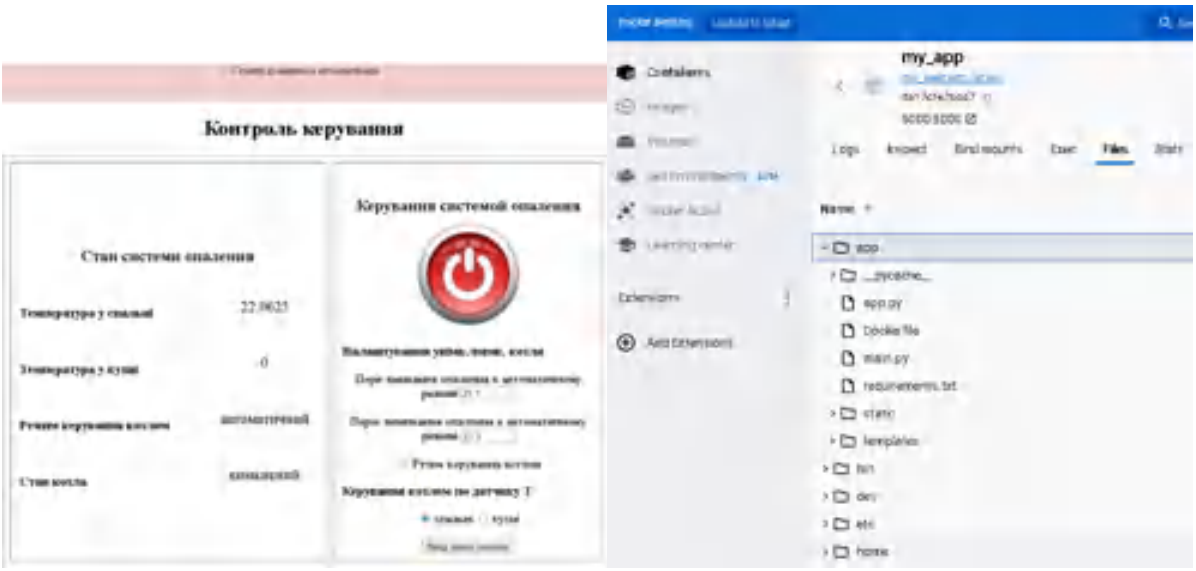
Visual Studio Code безпосередньо не призначено для створення додатків на Python, але має велику кількість розширень, включаючи ті, які дозволяють працювати з Docker, Python та іншими мовами, що робить його дуже гнучким і можливим для налаштування під конкретні потреби користувача.

Одними з основних вимог до середовищ розробки додатків сьогодні є: швидкість розробки та можливість перевірки роботи у покроковому

режимі. Саме тому метою цієї роботи був аналіз можливостей кожного середовища із розробки та відлагодження Docker образів.

Для досліджень була обрана операційна система Windows 10, у яку встановлено програмне середовище Docker Desctop, що дозволяє запускати та підтримувати образи Docker та вбудовані контейнери. Також було встановлено: Visual Studio Code з розширеннями Python та Docker і PyCharm Community 2023.1 з розширенням Docker. У якості приклада системи для аналіза було обрану систему керування котлом приміщення з дистанційним збором інформації від датчиків температури та візуалізацією стану роботи котла, налаштуваннями та елементами керування. Датчики температури опитувалися за допомогою модулів ESP32, а зібрана інформація передавалась на серверну частину (контейнер Docker) у вигляді JSON запитів. Мовою Python було розроблено web додаток з використанням бібліотеки Falsk 2.3. Спочатку його робота була перевірена в обох середовищах програмування Visual Studio Code та PyCharm Community без компіляції Docker. Для створення Docker образу та контейнеру було створено 3 файла: dockerfile, requirements.txt, docker-compose.yml. Перший файл описує процес створення контейнеру з web додатком. У якості ядра обрана версія Linux – alpine з доданим python версії 3.9 (FROM python:3.9-alpine). Директиви цього файла також забезпечили копіювання вмісту web додатку у контейнер (ADD .. /app), інсталяцію в контейнер додаткових бібліотек (RUN pip install -r requirements.txt), що немає в базовій версії python і перелік яких містить файл requirements.txt. Також його директиви забезпечили запуск додатку у образі. Файл docker-compose.yml містив директиви для створення образу із контейнером web додатку, завдання імен образу та контейнеру, з'єднання внутрішнього порта додатку та зовнішнього порта контейнера.

Компіляція контейнеру та образу у Visual Studio Code виконувалась у вбудованому терміналі через відповідну команду, у якій вказується docker-compose.yml файл і весь процес проходить відповідно до послідовності дій прописаних у раніше розглянутих файлах. В результаті було запущено web додаток у контейнері, доступ до якого перевірено із браузера (рисунок 1.а) та стандартне ядро Linux з доданою папкою app із сукупністю файлів сайту (рисунок 1.б). Компіляція контейнеру у PyCharm значно простіша. Спочатку в налаштуваннях PyCharm було вказано шлях до локального розташування Docker Desctop, хоча можна також вказувати хмарне розташування відповідного сервісу. Далі через стандартний пункт меню RUN було запущено процес створення образу із контейнером web додатку. Основною відмінністю PyCharm від Visual Studio Code є можливість підключатись до працюючої програми у контейнері в режимі налагодження (Debug), встановлювати точки зупинки, передивлятись стан внутрішніх змінних і т.і. (рисунок 2)



а б

Рисунок 1 – Образ із контейнера та робота web додатка

Висновки. Середовище PyCharm Community з розширенням Docker має значні переваги перед Visual Studio Code при створенні Docker контейнерів мовою python. В першу чергу це можливість дистанційного підключення до додатку у контейнері і його налагодження засобами PyCharm, а також дуже спрощений процес створення образу і контейнеру. Розширенням Docker у Visual Studio Code не забезпечує режиму налагодження, але є більш універсальним з точки зору підтримки різних мов програмування. Результати компіляції образу в PyCharm Community та Visual Studio Code показали, що розмір образу, скомпільованого за допомогою PyCharm Community на 1% менший ніж у Visual Studio Code, що пояснюється спеціалізацією PyCharm.

Список використаних джерел:

1. Bernd Oggl, Michael Kofler Docker: Practical Guide for Developers and Devops Teams Rheinwerk, Computing, – 2023, – 492 p.
2. Ruchika M. Evaluation of Docker for IoT Application nternational Journal on Recent and Innovation Trends in Computing and Communication, Volume: 4, Issue: 6, – 2016, pp.624-628
3. Vedat Ozan Oner Developing IoT Projects with ESP32 2nd edition Packt, – 2023, – 578 p.
4. Malhar Lathkar Building Web Apps with Python and Flask: Learn to Develop and Deploy Responsive RESTful Web Applications Using Flask, Framework BPB Online LLP, – 2021, – 421 p.
5. Білоус М. Ю. Аналіз сучасних середовищ розробки програмного забезпечення / М. Ю. Білоус // Automation and Development of Electronic Devices, ADED-2020: збірник студентських наукових статей. – Харків : ХНУРЕ, 2020. – Вип. 2. – С. 13-16.