

Raport z projektu

Sieć Jako Usługa

Projekt kontenera developerskiego z konfiguracją przepływu CI/CD

Konfiguracja repozytorium

Kroki realizacji:

- Fork repozytorium bazowego na własne konto
- Projekt założony lokalnie w Visual Studio Code
- Test budowania obrazu Docker: `docker build -t sjuprojekt .`
- Uruchomienie kontenera: `docker run -it --rm -v ./:/home/vscode/workspace sjuprojekt bash`
- Weryfikacja dostępu do plików projektu: `ls /home/vscode/workspace`
- Repozytorium połączone z GitHub przez `git remote add origin`

Komendy git:

Modyfikacja Dockerfile

Instalacja pakietów w środowisku Dockera

- W środowisku **VS Code** skorzystałem z rozszerzenia **Dev Containers**, które umożliwia konfigurację kontenera z poziomu edytora.
- Zmodyfikowałem plik `Dockerfile`, aby instalował niezbędne pakiety:
 - `qiskit`
 - `matplotlib`
 - `pillow`
 - `pycryptodomex`
 - `cryptography`
 - oraz dodatkowo: `ipykernel`, `jupyter`

Konfiguracja kontenera deweloperskiego

Plik `.devcontainer/devcontainer.json`

```
{
  "workspaceMount": "source=${localWorkspaceFolder},target=/home/vscode/workspace,type=bind,consistency=cached",
  "workspaceFolder": "/home/vscode/workspace",
  "name": "Projekt-SJU",
  "image": "ghcr.io/piotrkwia/projekt_sju:latest",
  "customizations": {
    "vscode": {
      "extensions": [
        "ms-python.python",
        "ms-toolsai.jupyter",
        "yzhang.markdown-all-in-one",
        "marp-team.marp-vscode",
        "github.vscode-github-actions"
      ]
    }
  },
  "postCreateCommand": "pip install --no-cache-dir -r requirements.txt && uname -a && python --version && pip --version",
  "remoteUser": "vscode"
}
```

GitHub Actions – CI/CD Pipeline

- Utworzono folder `.github/workflows` w katalogu głównym projektu.
- Dodano plik `docker-build.yml` konfiguracyjny definiujący prosty pipeline CI.
- Pipeline uruchamia się automatycznie przy każdym `push` i `pull_request` do gałęzi `main`.
- Wykonywane kroki w workflow:
 - Ustawienie środowiska (checkout kodu, instalacja Pythona).
 - Instalacja zależności z `requirements.txt`.
 - Walidacja składni i poprawności kodu (lint/test).

```
name: Build, Test and Publish Docker Image
```

```
on:  
  workflow_dispatch:  
  release:  
    types: [created]
```

```
env:  
  REGISTRY: ghcr.io  
  IMAGE_NAME: piotrkwia/projekt_sju
```

Diagram przebiegu pracy

```
graph TD
  A["Pliki źródłowe"] --> B[Repozytorium na GitHubie]
  B --> C[Akcje CI z GitHub Actions]
  C --> D[Budowanie obrazu Dockera]
  D --> E[Uruchomienie testów automatycznych]
  E --> F{Wyniki testów}
  F -->|Sukces| G[Publikacja do ghcr.io]
  F -->|Błąd| H[Błąd kompilacji]
  G --> I[Kontener developerski gotowy]
  I --> J[Uruchomienie Jupyter Notebooks]
  J --> K[Praca w środowisku]
```

Praca z Jupyter Notebooks

W środowisku lokalnym (VSCode)

- Do pracy z notebookami użyto rozszerzenia `Jupyter` dostępnego w edytorze Visual Studio Code.
- Notatniki `.ipynb` były uruchamiane i edytowane lokalnie wewnątrz kontenera developerskiego.
- Środowisko było oparte na wcześniej skonfigurowanym obrazie Dockera oraz pliku `devcontainer.json`.
- Kernel był przypisany do wirtualnego środowiska `.venv`, dzięki czemu wszystkie zależności były dostępne.
- Notebooks wykorzystywały bibliotekę `Qiskit` do wysyłania zadań na komputer kwantowy IBM Quantum.

Wyzwania i doświadczenia

Co było najtrudniejsze:

- Konfiguracja GitHub Actions z odpowiednimi uprawnieniami do publikacji obrazu
- Rozwiązywanie błędów pojawiających się podczas budowy obrazu Dockera
- Utrzymanie spójności wersji obrazu między devcontainerem a rejestrem kontenerów

Czego się nauczyłem:

- Jak praktycznie stosować konteneryzację przy pracy z VSCode i Dockerem
- Jak zautomatyzować procesy CI/CD z użyciem GitHub Actions
- W jaki sposób integrować różne technologie cloud-native w ramach jednego projektu

Podsumowanie i rozwój

Zrealizowane elementy:

- ✓ Skonfigurowane środowisko programistyczne w kontenerze
- ✓ W pełni działający pipeline CI/CD z GitHub Actions
- ✓ Publikacja obrazu do GitHub Container Registry
- ✓ Integracja i uruchamianie notebooków Jupyter w kontenerze
- ✓ Możliwość dalszej rozbudowy środowiska o nowe rozszerzenia VSCode