

Politechnika Lubelska

Wydział Elektrotechniki i Informatyki

Kierunek Informatyka



Programowanie full-stack w chmurze obliczeniowej

Student

Adrian Duwer

nr. albumu: 90902

grupa lab: I2N 1.1/1

Prowadzący

dr. inż. Sławomir Przyłucki

Lublin 2023

Spis treści

Zadanie 1	3
-----------------	---

Zadanie 1

CZĘŚĆ OBOWIĄZKOWA

1. (max. 100%)

Wykorzystując opracowaną aplikację (kod + Dockerfile) z zadania nr1 należy:

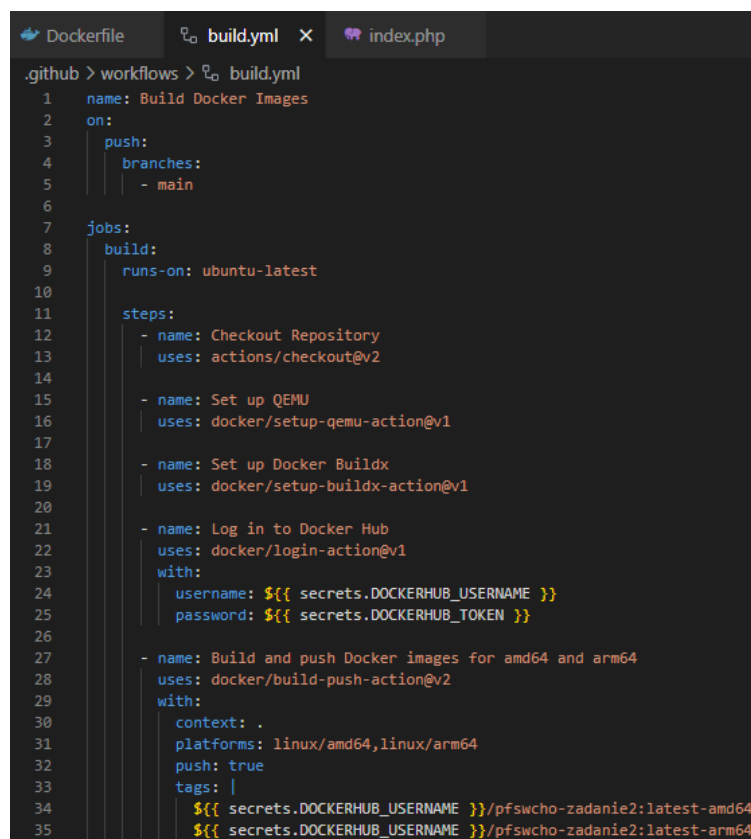
- zbudować, uruchomić i potwierdzić poprawność działania łańcucha Github Actions, który zbuduje obrazy kontenera z tą aplikacją na architektury: **linux/arm64/v8** oraz **linux/amd64** wykorzystując QEMU

W celu realizacji zadania opisanego powyżej wykorzystany zostanie kod opracowany w ramach pierwszego ze sprawozdań, który dostępny jest pod adresem: <https://github.com/aduwer/PFSwChO-zadanie1>

Kluczowe pliki, które zostały tam zamieszczone to index.php oraz Dockerfile.

W celu zbudowania, uruchomienia i potwierdzenia poprawności działania łańcucha Github Actions, który zbuduje obrazy kontenera z przygotowaną aplikacją utworzony został katalog .github/workflows. Katalog ten jest standardowym miejscem, w którym przechowuje się pliki konfiguracyjne dla workflow GitHub Actions. Umieszczenie pliku build.yml w tym katalogu umożliwi automatyczne wykonywanie działań w odpowiednim momencie, na przykład po pushu do repozytorium. Jest to rekomendowane miejsce dla plików konfiguracyjnych workflow GitHub Actions, ponieważ jest to standardowa praktyka i ułatwia zrozumienie struktury repozytorium dla innych osób współpracujących nad projektem.

Istnieje możliwość umieszczenia pliku build.yml w innym miejscu, jednak należy pamiętać o zmianie ścieżki w deklaracji uses dla każdego kroku w workflow.



```
.github > workflows > build.yml
1  name: Build Docker Images
2  on:
3    push:
4      branches:
5        - main
6
7  jobs:
8    build:
9      runs-on: ubuntu-latest
10
11     steps:
12       - name: Checkout Repository
13         uses: actions/checkout@v2
14
15       - name: Set up QEMU
16         uses: docker/setup-qemu-action@v1
17
18       - name: Set up Docker Buildx
19         uses: docker/setup-buildx-action@v1
20
21       - name: Log in to Docker Hub
22         uses: docker/login-action@v1
23         with:
24           username: ${ secrets.DOCKERHUB_USERNAME }
25           password: ${ secrets.DOCKERHUB_TOKEN }
26
27       - name: Build and push Docker images for amd64 and arm64
28         uses: docker/build-push-action@v2
29         with:
30           context: .
31           platforms: linux/amd64,linux/arm64
32           push: true
33           tags: |
34             ${ secrets.DOCKERHUB_USERNAME }/pfswocho-zadanie2:latest-amd64
35             ${ secrets.DOCKERHUB_USERNAME }/pfswocho-zadanie2:latest-arm64
```

Rysunek 1.1. Kod źródłowy dla pliku build.yml

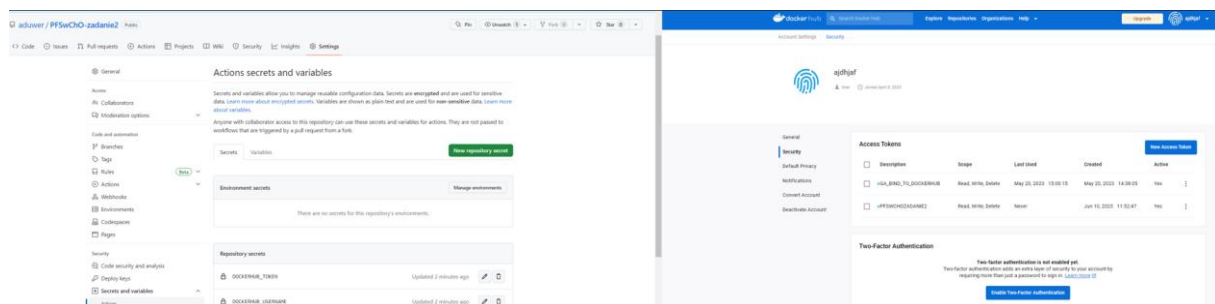
Kod źródłowy wykorzystany w pliku build.yml przedstawiony na rysunku 1.1. umożliwia budowanie obrazów na podstawie pliku Dockerfile. Poniżej przedstawię skrócony opis jego zawartości:

Krok "Checkout Repository" - pobiera zawartość repozytorium, na którym został uruchomiony skrypt.

Krok "Set up QEMU" - używa akcji docker/setup-qemu-action@v1 do konfiguracji QEMU na środowisku uruchomieniowym, co jest konieczne do budowania obrazów dla architektury arm64 na platformie x86_64.

Krok "Set up Docker Buildx" - używa akcji docker/setup-buildx-action@v1 do konfiguracji narzędzia Docker Buildx. Buildx jest narzędziem wieloplatformowym, które umożliwia budowanie obrazów Docker dla różnych architektur.

Krok "Log in to Docker Hub" - używa akcji docker/login-action@v1 do zalogowania się do Docker Hub. Wymaga podania nazwy użytkownika i hasła do konta Docker Hub. W naszym przypadku hasłem jest wygenerowany przez na platformie Docker Hub specjalny token. Nazwa użytkownika oraz token pobierane są z tzw. „Repository secret”, które umieszczone zostały w ustawieniach repozytorium na platformie GitHub o nazwach DOCKERHUB_USERNAME oraz DOCKERHUB_TOKEN. Wynik utworzenia wspomnianych wartości przedstawiony jest na rysunku 1.2. znajdującym się poniżej:



Rysunek 1.2. Utworzenie zmiennych w repozytorium GitHub oraz tokenu dostępu na platformie Docker Hub

Krok "Build and push Docker images for amd64 and arm64" - używa akcji docker/build-push-action@v2 do budowania i wysyłania obrazów Docker. Ten krok ma następujące ustawienia:

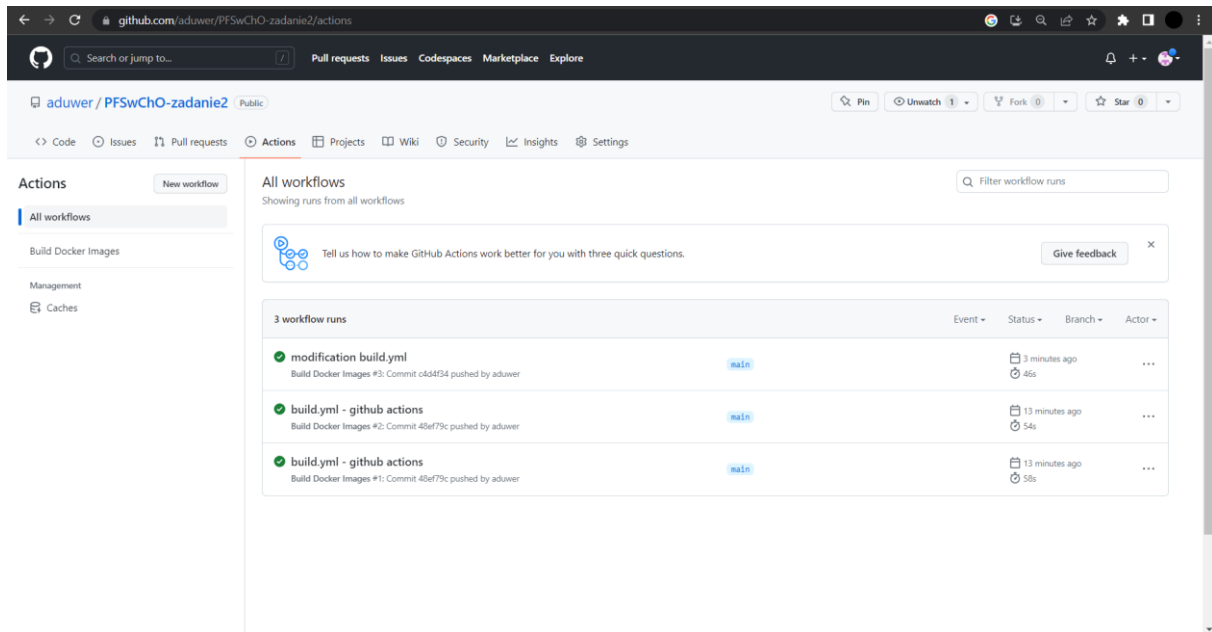
"context: ." wskazuje, że kontekst budowy (czyli katalog, w którym znajduje się plik Dockerfile oraz inne potrzebne pliki) jest aktualnym katalogiem.

"platforms: linux/amd64,linux/arm64" definiuje, dla jakich platform mają być zbudowane obrazy kontenera. W tym przypadku są to amd64 i arm64.

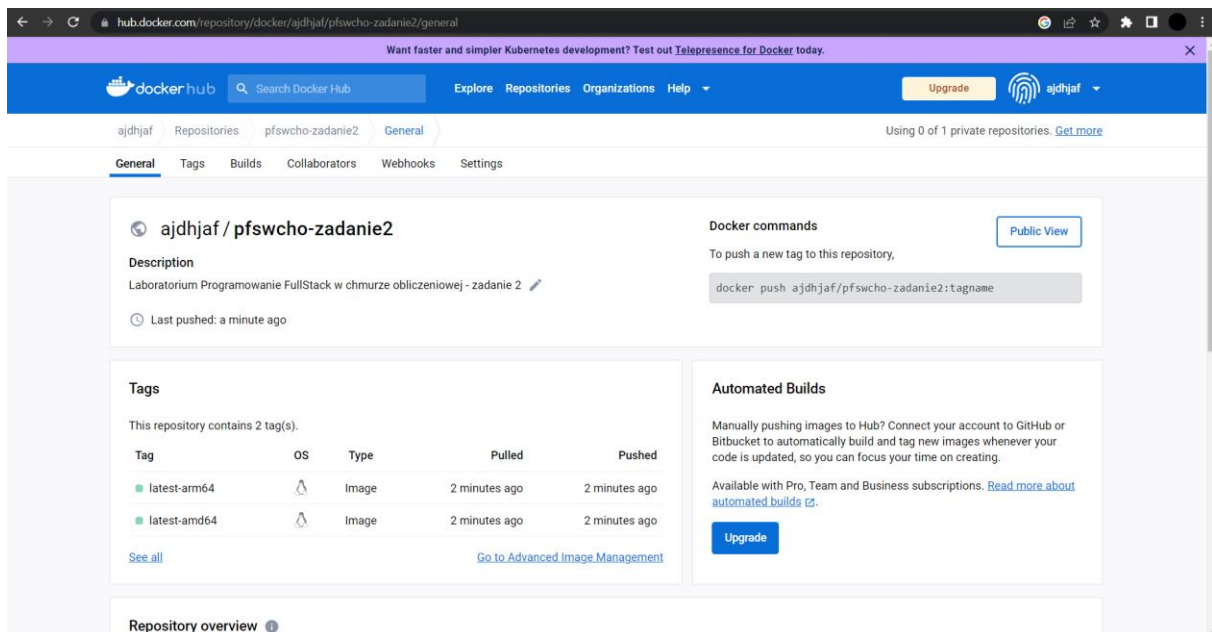
"push: true" określa, że po zbudowaniu obrazów należy je wypchnąć na zewnętrzny rejestr kontenerów (w tym przypadku Docker Hub).

"tags" - lista tagów, które zostaną przypisane do zbudowanych obrazów. W tym przypadku są to tagi zależne od nazwy użytkownika w Docker Hub oraz oznaczenia wersji dla obrazów dla architektur amd64 i arm64.

Skrypt ten umożliwi automatyczne budowanie i wysyłanie obrazów Docker do Docker Hub za każdym razem, gdy nastąpi push do gałęzi "main" w repozytorium. Jeśli akcja "Build Docker Images" zakończyła się sukcesem i nie ma żadnych widocznych błędów w logach, można założyć, że Dockerfile został poprawnie zbudowany i obrazy kontenera zostały utworzone zgodnie z oczekiwaniami. Rysunki 1.3. oraz 1.4. potwierdzają prawidłowość zbudowania obrazów.



Rysunek 1.3. Wynik działania na platformie GitHub



Rysunek 1.4. Wynik działania na platformie Docker Hub

Aby sprawdzić, czy obrazy zostały utworzone dla prawidłowych platform i czy aplikacja działa poprawnie, uruchomiona została lokalnie na odpowiednich platformach. W celu weryfikacji poprawności pobrane zostały obrazy kontenera z Docker Hub na lokalne środowisko.

a) platforma amd64

* docker pull ajdhjaf/pfswcho-zadanie2:latest-amd64

```
root@DESKTOP-00TNOHE: ~/PFSwCh0-2
root@DESKTOP-00TNOHE:~/PFSwCh0-2# docker pull ajdhjaf/pfswcho-zadanie2:latest-amd64
latest-amd64: Pulling from ajdhjaf/pfswcho-zadanie2
ca7dd9ec2225: Already exists
1b78b4fe0ca1: Already exists
9d6040f2a28f: Already exists
0e2e66b89284: Already exists
3b1be5f02bec: Already exists
96243f515dda: Already exists
4006f78ca99d: Already exists
3c76b37a5dd7: Already exists
6491a80416fc: Already exists
9dbdbb616fbf: Pull complete
245156a99d96: Pull complete
aaee68455e20: Pull complete
4f4fb700ef54: Pull complete
a3bf958e925c: Pull complete
8a4ce2944fa5: Pull complete
Digest: sha256:50913896d56eb49521ff79cc6b0802d0f6b0c29d97edaee258ee6c5fa1ec8b07
Status: Downloaded newer image for ajdhjaf/pfswcho-zadanie2:latest-amd64
docker.io/ajdhjaf/pfswcho-zadanie2:latest-amd64
root@DESKTOP-00TNOHE:~/PFSwCh0-2#
```

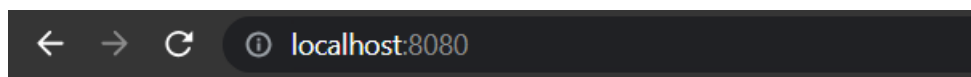
Rysunek 1.5. Pobranie obrazu z Docker Hub dla platformy amd64

* docker run -p 8080:80 ajdhjaf/pfswcho-zadanie2:latest-amd64

```
root@DESKTOP-00TNOHE:~/PFSwCh0-2# docker run -p 8080:80 ajdhjaf/pfswcho-zadanie2:latest-amd64
[Sun Jun 11 20:26:49 2023] PHP 7.4.33 Development Server (http://0.0.0.0:80) started
```

Rysunek 1.6. Uruchomienie aplikacji dla platformy amd64

Wynik potwierdzający prawidłowość działania:



Adres IP klienta: 172.17.0.1

Data i godzina: Sunday, 11-06-2023, 20:28:12

Rysunek 1.6. Rezultat po przejściu na stronę localhost:8080 dla platformy amd64

Wyświetlenie logów:

```
root@DESKTOP-00TNOHE:~/PFSwCh0-2# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
e18cf131b9a1  ajdhjaf/pfswcho-zadanie2:latest-amd64 "docker-php-entrypoi..." 3 minutes ago  Up 3 minutes (healthy)  0.0.0.0:8080->80/tcp, :::8080->80/tcp  priceless_hofstadter

root@DESKTOP-00TNOHE:~/PFSwCh0-2# docker exec priceless_hofstadter cat log.txt
Serwer uruchomiony dnia: Sunday, 11-06-2023, 20:27:49
przez: Adrian Duwer
na porcie: 80

Serwer uruchomiony dnia: Sunday, 11-06-2023, 20:28:11
przez: Adrian Duwer
na porcie: 80

Serwer uruchomiony dnia: Sunday, 11-06-2023, 20:28:50
przez: Adrian Duwer
na porcie: 80

Serwer uruchomiony dnia: Sunday, 11-06-2023, 20:29:50
przez: Adrian Duwer
na porcie: 80

root@DESKTOP-00TNOHE:~/PFSwCh0-2#
```

Rysunek 1.7. Logi serwera dla platformy amd64

b) platforma arm64

* docker pull ajdhjaf/pfswcho-zadanie2:latest-arm64

```
root@DESKTOP-00TNOHE: ~/PFSwChO-2
root@DESKTOP-00TNOHE:~/PFSwChO-2# docker pull ajdhjaf/pfswcho-zadanie2:latest-arm64
latest-arm64: Pulling from ajdhjaf/pfswcho-zadanie2
Digest: sha256:50913896d56eb49521ff79cc6b0802d0f6b0c29d97edaee258ee6c5fa1ec8b07
Status: Downloaded newer image for ajdhjaf/pfswcho-zadanie2:latest-arm64
docker.io/ajdhjaf/pfswcho-zadanie2:latest-arm64
root@DESKTOP-00TNOHE:~/PFSwChO-2#
```

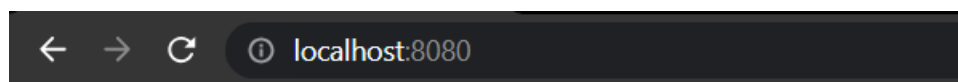
Rysunek 1.8. Pobranie obrazu z Docker Hub dla platformy arm64

* docker run -p 8080:80 ajdhjaf/pfswcho-zadanie2:latest-arm64

```
root@DESKTOP-00TNOHE:~/PFSwChO-2# docker run -p 8080:80 ajdhjaf/pfswcho-zadanie2:latest-arm64
[Sun Jun 11 20:33:21 2023] PHP 7.4.33 Development Server (http://0.0.0.0:80) started
```

Rysunek 1.9. Uruchomienie aplikacji dla platformy arm64

Wynik potwierdzający prawidłowość działania:



Adres IP klienta: 172.17.0.1

Data i godzina: Sunday, 11-06-2023, 20:33:52

Rysunek 1.10. Rezultat po przejściu na stronę localhost:8080 dla platformy arm64

Wyświetlenie logów:

```
root@DESKTOP-00TNOHE:~/PFSwChO-2# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
5ad0b28d82a0   ajdhjaf/pfswcho-zadanie2:latest-arm64 "docker-php-entrypoi..." About a minute ago Up 58 seconds (health: starting) 0.0.0.0:8080->80/tcp, :::8080->80/tcp bold_ride

root@DESKTOP-00TNOHE:~/PFSwChO-2# docker exec bold_ride cat log.txt
Serwer uruchomiony dnia: Sunday, 11-06-2023, 20:33:52
przez: Adrian Duwer
na porcie: 80

Serwer uruchomiony dnia: Sunday, 11-06-2023, 20:34:21
przez: Adrian Duwer
na porcie: 80

root@DESKTOP-00TNOHE:~/PFSwChO-2#
```

Rysunek 1.11. Logi serwera dla platformy arm64

Całość pozwala stwierdzić, że proces budowy, uruchamiania i potwierdzenia poprawności działania łańcucha Github Actions z aplikacją przedstawioną w pierwszym sprawozdaniu na architekturze arm64 oraz amd64 przebiegł prawidłowo.