

Szkolenie

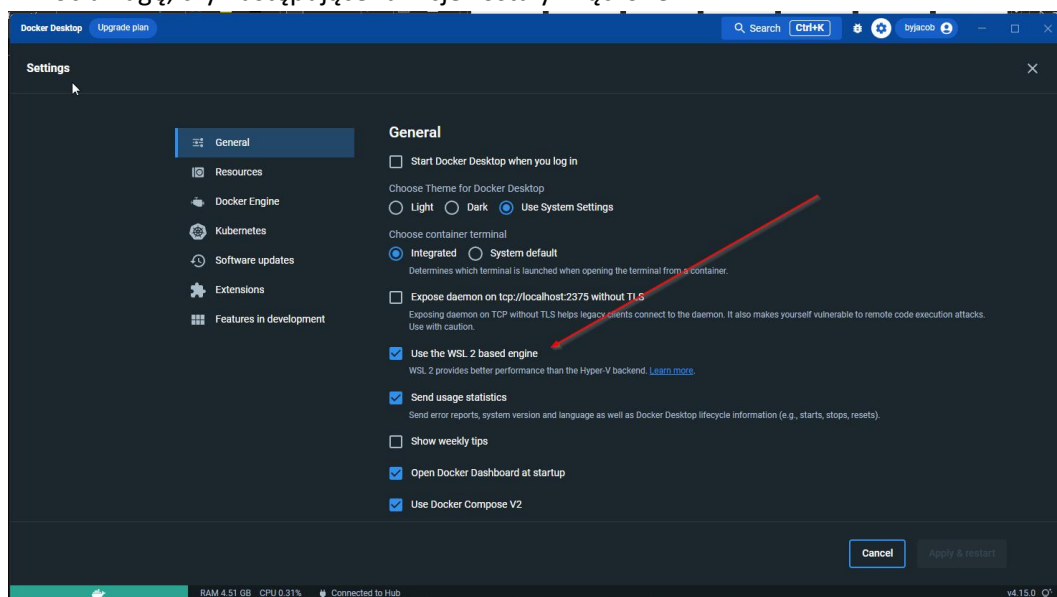
„Jak skutecznie zaprojektować i wytworzyć dobry obraz dockerowy” przygotowanie środowiska

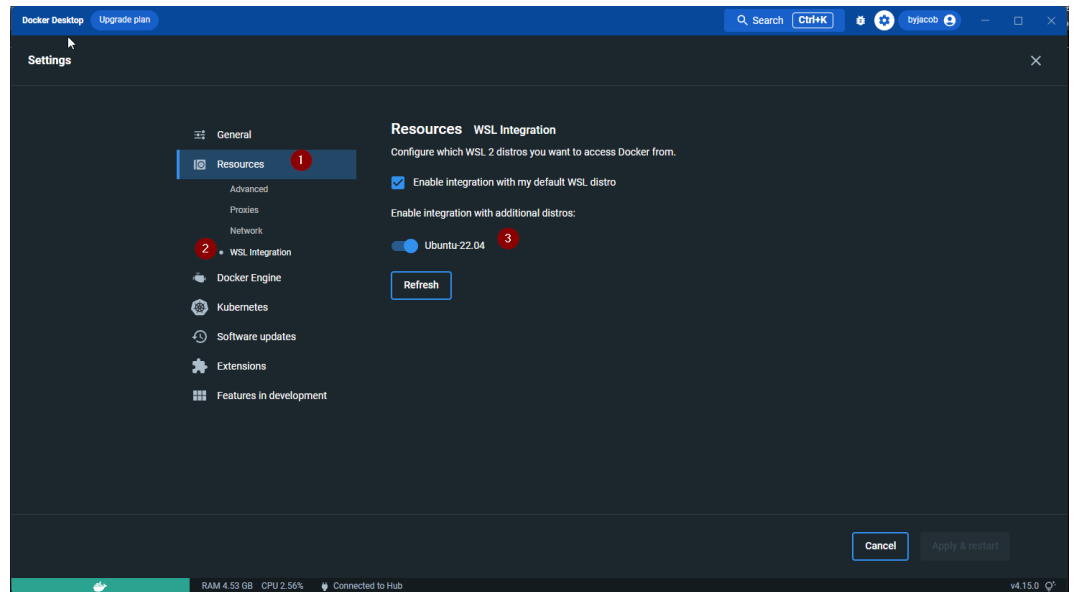
1. Środowisko pracy

- a. Windows – Uruchomienie i praca na WSL2 - <https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-11-with-gui-support> (pierwsze 3 punkty)
 - i. Następne komendy należy uruchamiać poprzez konsolę zainstalowanej dystrybucji, np. Ubuntu
- b. Linux: w terminalu

2. Docker

- a. Linux: <https://docs.docker.com/engine/install/>
 - i. Sprawdź czy działa komenda ``docker compose version`` - jeżeli nie sprawdź instrukcję <https://docs.docker.com/compose/install/linux/>
- b. Windows: <https://docs.docker.com/desktop/windows/wsl/>
 - i. Zwróć uwagę, czy następujące funkcje zostały włączone





- c. Komenda do weryfikacji działania:

`docker run hello-world`

Wynik powinien być podobny do poniższego:

```
jaros@JROSA-DOM:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:94ebc7edf3401f299cd3376a1669bc0a49aef92d6d2669005f9bc5ef028dc333
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

jaros@JROSA-DOM:~$ |
```

`docker compose version`

```
jrosa@homelab:~[1]▶ docker compose version
Docker Compose version v2.17.2
```

3. Hadolint

- Instrukcja: <https://github.com/hadolint/hadolint#install>
- Szybkie komendy(dla linuxa x64 i WSL2):
`sudo wget -O /bin/hadolint`
`https://github.com/hadolint/hadolint/releases/download/v2.12.0/hadolint-Linux-x86_64;`
`sudo chmod a+x /bin/hadolint;`
- Komendy do weryfikacja działania:
`echo -e "FROM hello-world\nRUN apt-get update && apt-get install vim" > Dockerfile;`
`hadolint Dockerfile;`
Wynik powinien być podobny do poniższego(warningi i info):

```
jaros@JROSA-DOM:~$  
jaros@JROSA-DOM:~$ echo -e "FROM hello-world\nRUN apt-get update && apt-get install vim" > Dockerfile  
jaros@JROSA-DOM:~$ hadolint Dockerfile  
Dockerfile:1 DL3006 warning: Always tag the version of an image explicitly  
Dockerfile:2 DL3008 warning: Pin versions in apt get install. Instead of 'apt-get install <package>' use 'apt-get install <package>=<version>'  
Dockerfile:2 DL3015 info: Avoid additional packages by specifying '--no-install-recommends'  
Dockerfile:2 DL3009 info: Delete the apt-get lists after installing something  
Dockerfile:2 DL3014 warning: Use the '-y' switch to avoid manual input 'apt-get -y install <package>'  
jaros@JROSA-DOM:~$
```

- Utworzony plik Dockerfile można usunąć: `rm Dockerfile`

4. Dive

- Instrukcja: <https://github.com/wagoodman/dive#installation>
- Szybkie komendy(dla linuxa x64 i WSL2):
`wget`
`https://github.com/wagoodman/dive/releases/download/v0.9.2/dive_0.9.2_linux_a`
`md64.deb;`
`sudo apt install ./dive_0.9.2_linux_amd64.deb;`
- Komenda do weryfikacja działania:
`dive wordpress:php8.1-fpm;`

Na początku powinno pojawić się okno pobierania obrazu dockera, a następnie powinno wyświetlić się okno podobne do tego poniżej(CTRL+C żeby wyjść):

Layers		Current Layer Contents	
Cmp	Size Command	Permission UID:GID	Size Filetree
80 MB	FROM e038596f14d61b3	drwxr-xr-x 0:0	5.3 MB bin
46 B	set -eux; { echo 'Package: php*'; echo 'Pin: release *';	-rwxr-xr-x 0:0	1.2 MB bash
269 MB	set -eux; apt-get update; apt-get install -y --no-install-r	-rwxr-xr-x 0:0	44 kB cat
0 B	set -eux; mkdir -p "\$PHP_INI_DIR/conf.d"; [! -d /var/www/h	-rwxr-xr-x 0:0	73 kB chgrp
13 MB	set -eux; savedAptMark="\$(apt-mark showmanual)"; apt-get u	-rwxr-xr-x 0:0	64 kB chmod
587 B	#(nop) COPY file:ce57c04b70896f77cc11eb2766417d8a1240fcfe5bb	-rwxr-xr-x 0:0	73 kB chown
87 MB	set -eux; savedAptMark="\$(apt-mark showmanual)"; apt-get u	-rwxr-xr-x 0:0	151 kB cp
7.2 kB	#(nop) COPY multi:869bde9dbeae74886a85c9e2107b3e3b4877116db8c	-rwxr-xr-x 0:0	126 kB dash
17 B	docker-php-ext-enable sodium	-rwxr-xr-x 0:0	114 kB date
27 kB	set -eux; cd /usr/local/etc; if [-d php-fpm.d]; then se	-rwxr-xr-x 0:0	81 kB dd
53 MB	set -eux; apt-get update; apt-get install -y --no-install-r	-rwxr-xr-x 0:0	94 kB df
42 MB	set -ex; savedAptMark="\$(apt-mark showmanual)"; apt-get u	-rwxr-xr-x 0:0	147 kB dir
173 B	set -eux; docker-php-ext-enable opcache; { echo 'opcache.	-rwxr-xr-x 0:0	84 kB dmesg
331 B	{ echo 'error_reporting = E_ERROR E_WARNING E_PARSE E	-rwxrwxrwx 0:0	0 B dnsdomainname → hostname
65 MB	set -eux; version='6.1.1'; sha1='80f0f829645dec07c68bcfe0a0	-rwxrwxrwx 0:0	0 B domainname → hostname
5.5 kB	#(nop) COPY --chown=www-data:www-data:file:f95ddeaad9b50dddf2	-rwxr-xr-x 0:0	40 kB echo
3.4 kB	#(nop) COPY file:5be6bcc31206cb827f037769d89fd092037ed61a1e10	-rwxr-xr-x 0:0	28 B egrep
		-rwxr-xr-x 0:0	40 kB false
		-rwxr-xr-x 0:0	28 B fgrep
		-rwxr-xr-x 0:0	69 kB findmnt
		-rwxr-xr-x 0:0	203 kB grep
		-rwxr-xr-x 0:0	2.3 kB gunzip
		-rwxr-xr-x 0:0	6.4 kB gzexe
		-rwxr-xr-x 0:0	98 kB gzip
		-rwxr-xr-x 0:0	23 kB hostname
		-rwxr-xr-x 0:0	73 kB ln
		-rwxr-xr-x 0:0	57 kB login
		-rwxr-xr-x 0:0	147 kB ls
		-rwxr-xr-x 0:0	150 kB lsblk
		-rwxr-xr-x 0:0	85 kB mkdir
		-rwxr-xr-x 0:0	77 kB mkmod
		-rwxr-xr-x 0:0	48 kB mktmp
Total Image size: 610 MB			
*C Quit Tab Switch view *F Filter *L Show layer changes *A Show aggregated changes			

5. Git

- a. Prawdopodobnie jest zainstalowany (sprawdź punkt c), jeżeli jednak nie przejdź do punktu b.
- b. Instrukcja: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
- c. Komendy do weryfikacji:

`git clone https://github.com/quay/clair.git`

Wynik komendy powinien wyglądać podobnie jak na screenie poniżej; możliwe że zapyta aplikacja, czy akceptować nowy klucz; po wszystkim można usunąć stworzony folder clair

```
ja@os@JROSA-DOM:~/Workspace/workshops/projektowanie_obrazow_docker/demo4$ git clone git@github.com:quay/clair.git
Cloning into 'clair'...
remote: Enumerating objects: 17250, done.
remote: Counting objects: 100% (216/216), done.
remote: Compressing objects: 100% (121/121), done.
remote: Total 17250 (delta 124), reused 154 (delta 92), pack-reused 17034
Receiving objects: 100% (17250/17250), 32.66 MiB | 9.77 MiB/s, done.
Resolving deltas: 100% (8165/8165), done.
```

6. Obrazy dockerowe

- a. Jeżeli posiadasz słabsze łącze można wcześniej pobrać odpowiednie obrazy, aby były lokalnie dostępne.
- b. Komendy:

```
docker pull alpine:3
docker pull ubuntu:22.10
docker pull ubuntu:jammy
docker pull docker.io/library/postgres:12
docker pull quay.io/projectquay/golang:1.17
docker pull docker.io/library/traefik:v2.2
docker pull python:3.10-bullseye
docker pull python:3.10-slim-bullseye
docker pull python:3.10-alpine
docker pull php:8.1
docker pull php:7.4
docker pull registry:2
docker pull r.j3ss.co/img
docker pull gcr.io/kaniko-project/executor:v1.9.2
```

- c. Waga obrazów na stan z 05.04.2023 to: **3.531GB**