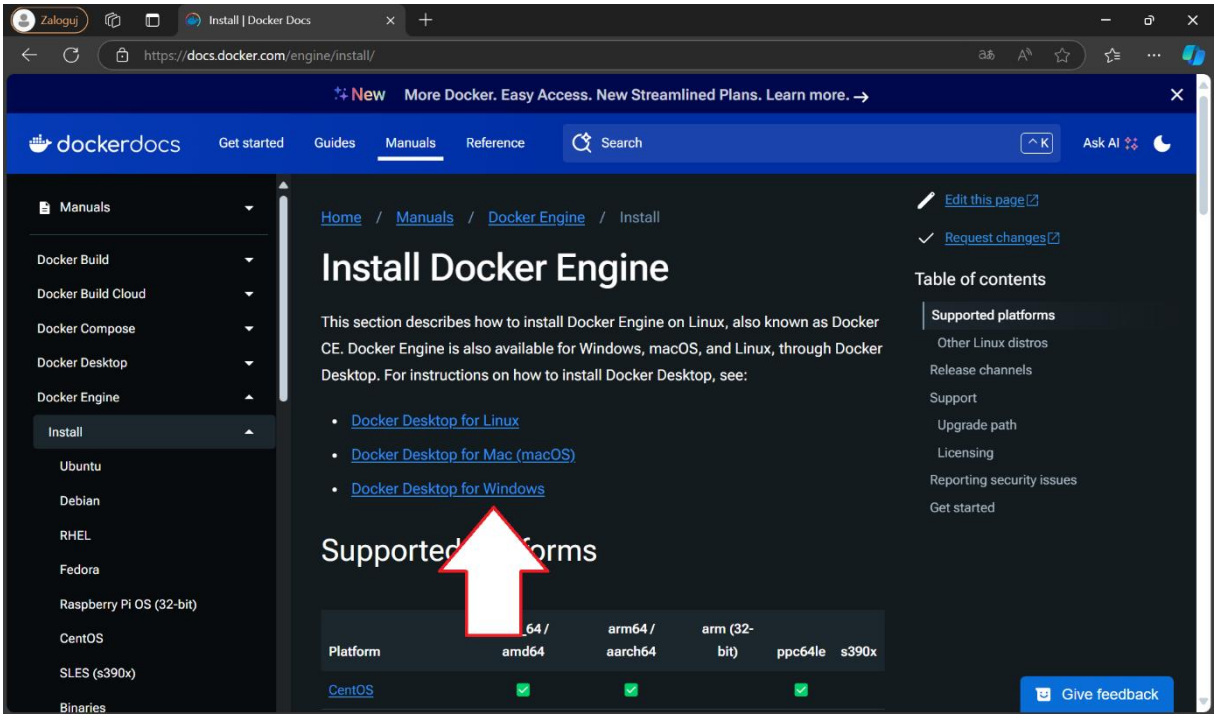
Docker

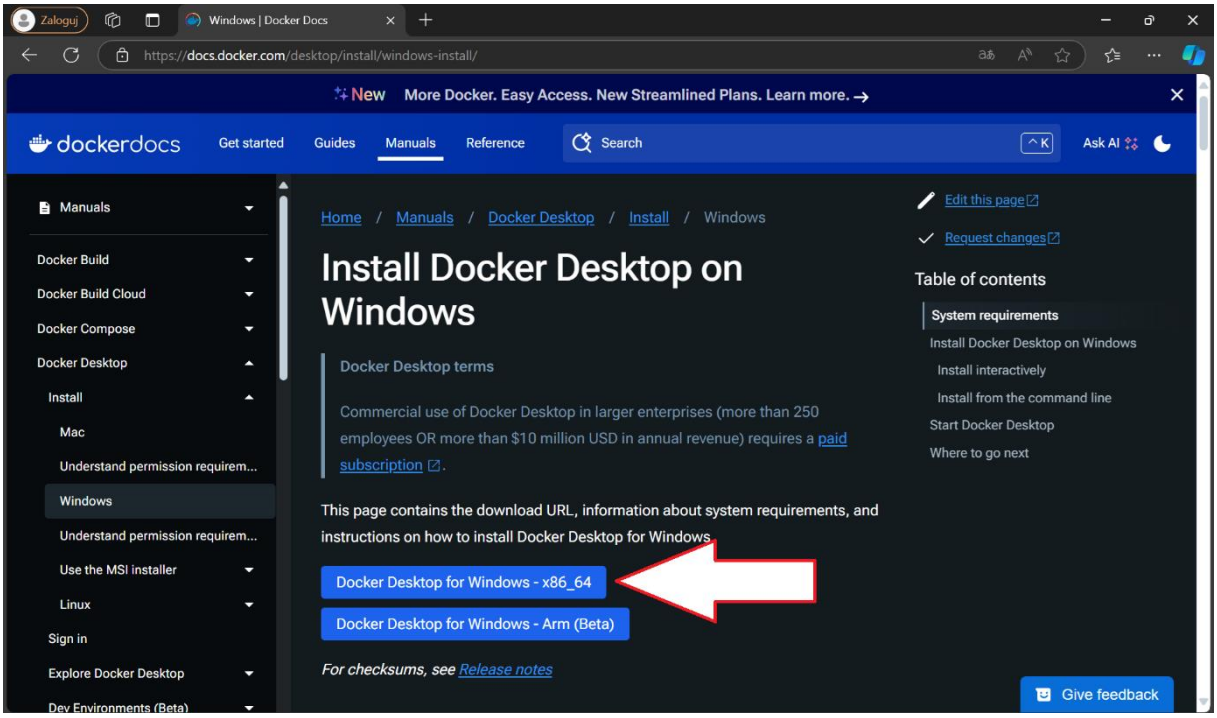
jak go właściwie skonfigurować?

1. Instalacja dla systemu Windows

Pierwszy krok:



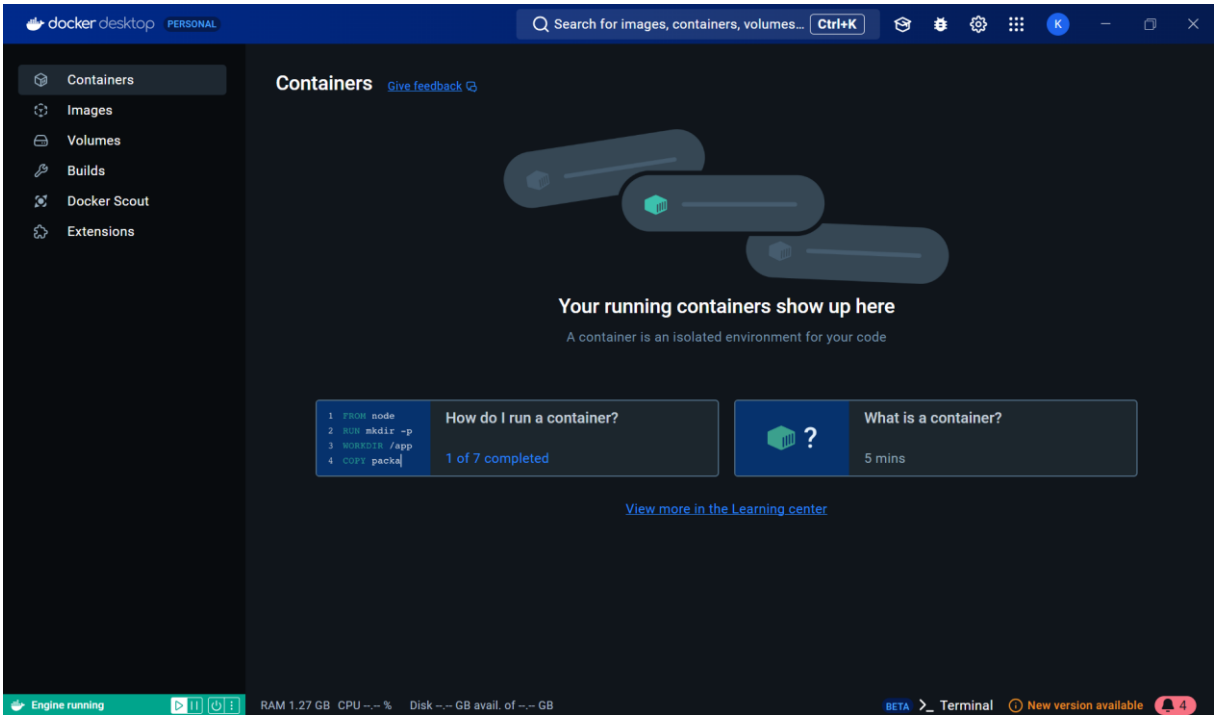
Drugi krok:



Autor: Kamil Pawłowski

Po tych krokach powinna rozpocząć się instalacja. Po zainstalowaniu instalatora postępować zgodnie z intuicją :)

Po zainstalowaniu nasz Docker będzie wyglądać mniej więcej tak:



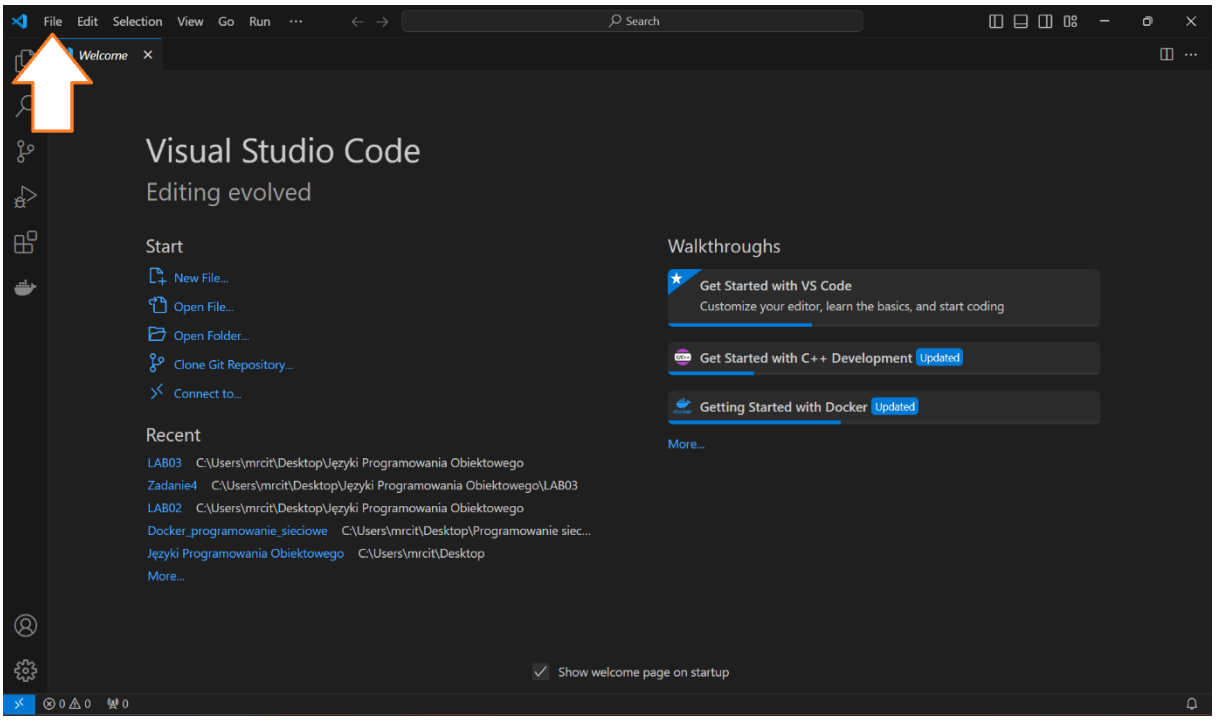
2. Pobieramy: docker_pliki.zip

Po pobraniu polecam przerzucić sobie docker_pliki.zip na pulpit, następnie otworzyć go i przerzucić folder „Docker” na pulpit. Folder będzie zawierać 2 pliki:

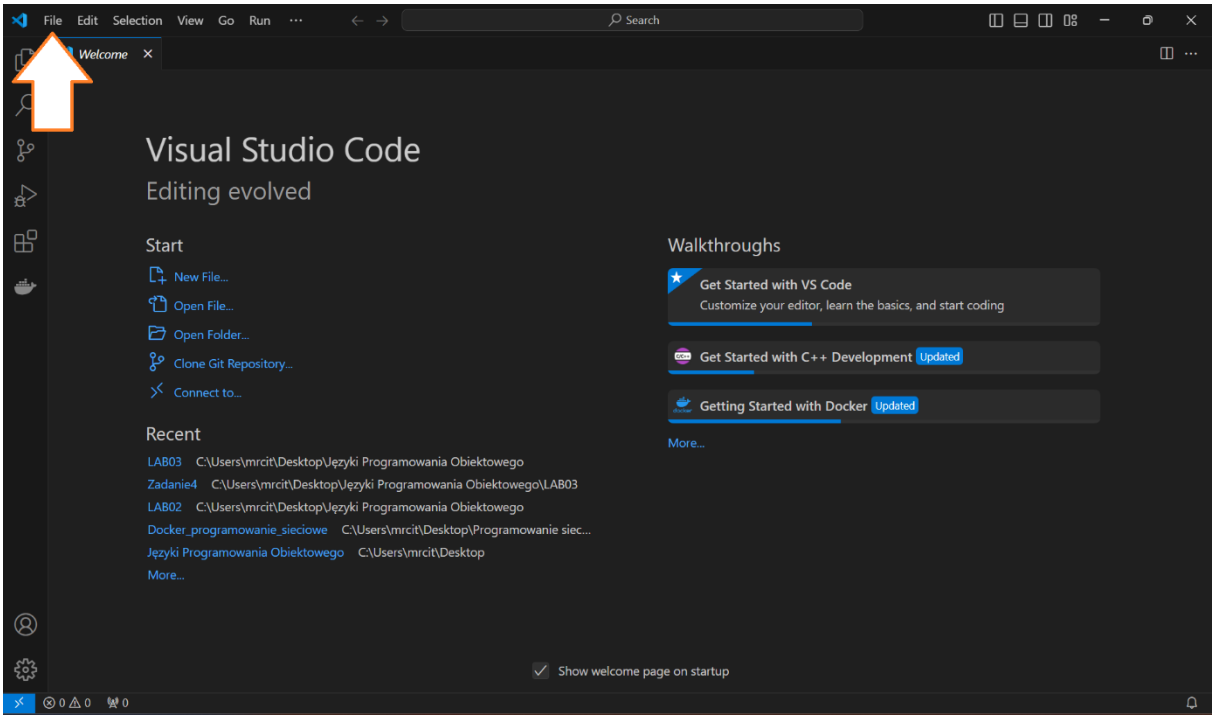
- Dockerfile
- Docker-compose.yml

3. Otwieramy dowolne środowisko programistyczne. W moim przypadku będzie to Visual Studio Code (w skrócie VSC).

Klikamy „File”

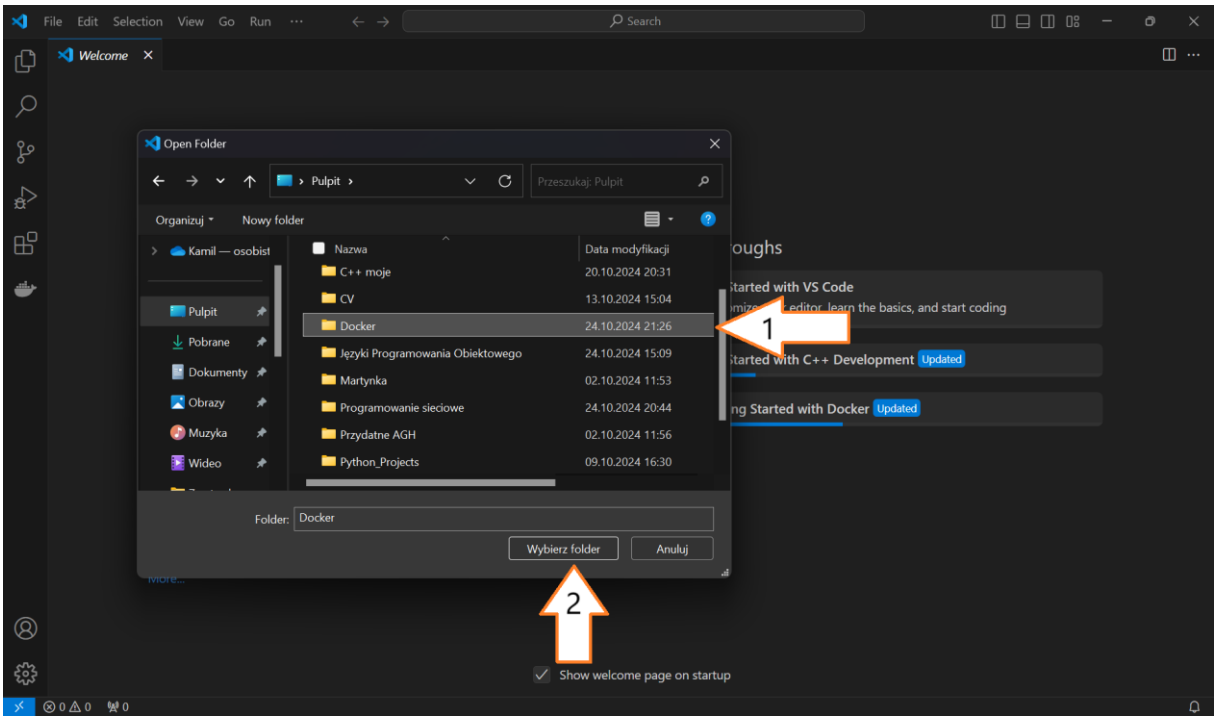


Następnie „Open Folder...”

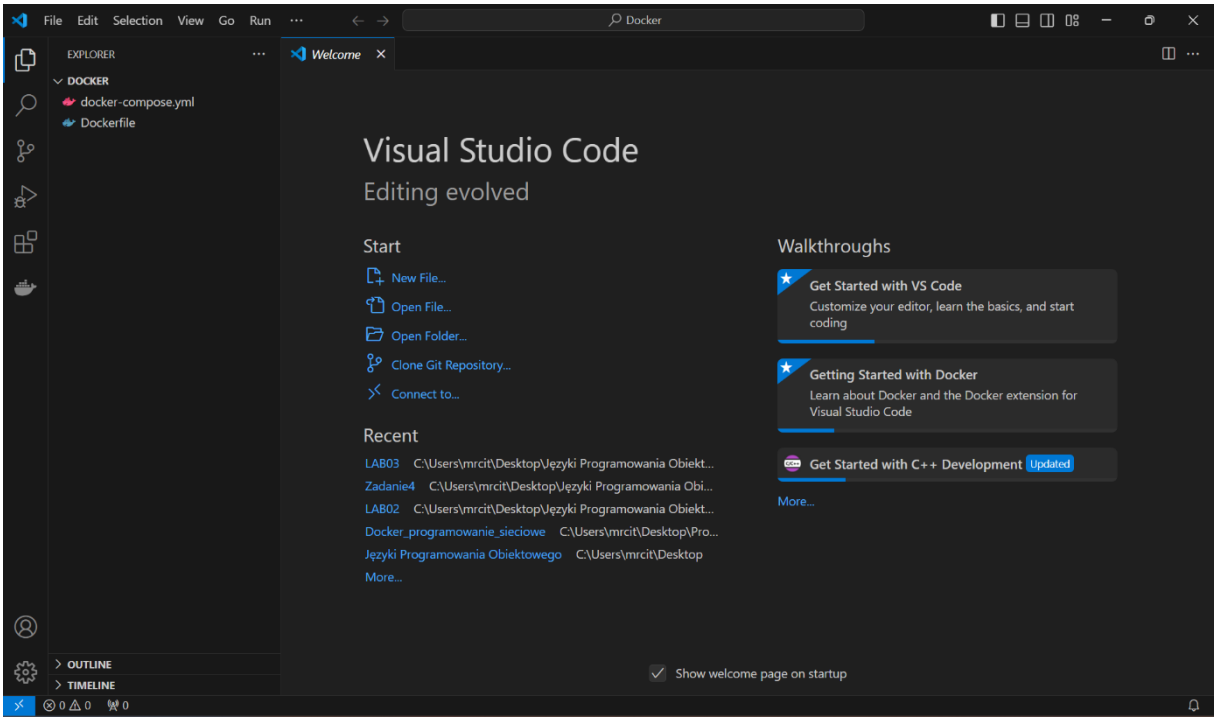


Autor: Kamil Pawłowski

Klikamy „Pulpit” (jeśli zapisaliście na pulpicie), następnie lewym klawiszem myszy raz klikamy folder „Docker” (strzałka nr 1) i dajemy „Wybierz folder” (strzałka nr 2)



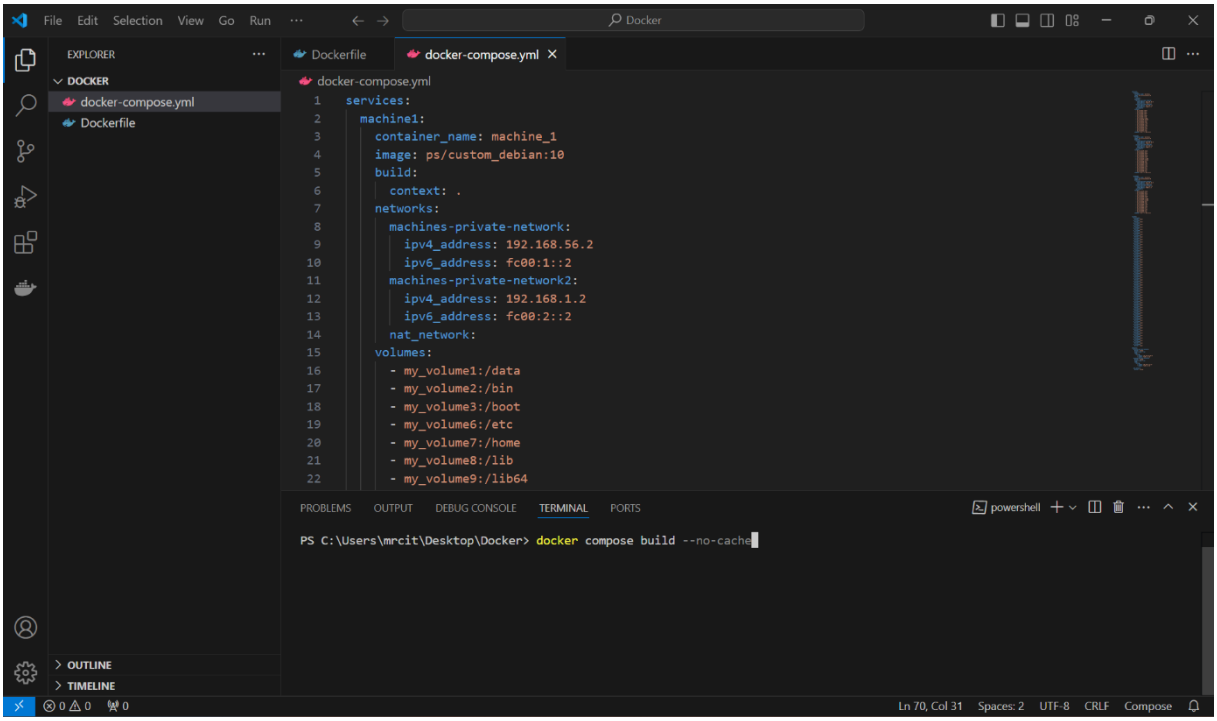
Jeśli wszystko poszło zgodnie z planem, nasze okienko VSC powinno wyglądać tak:



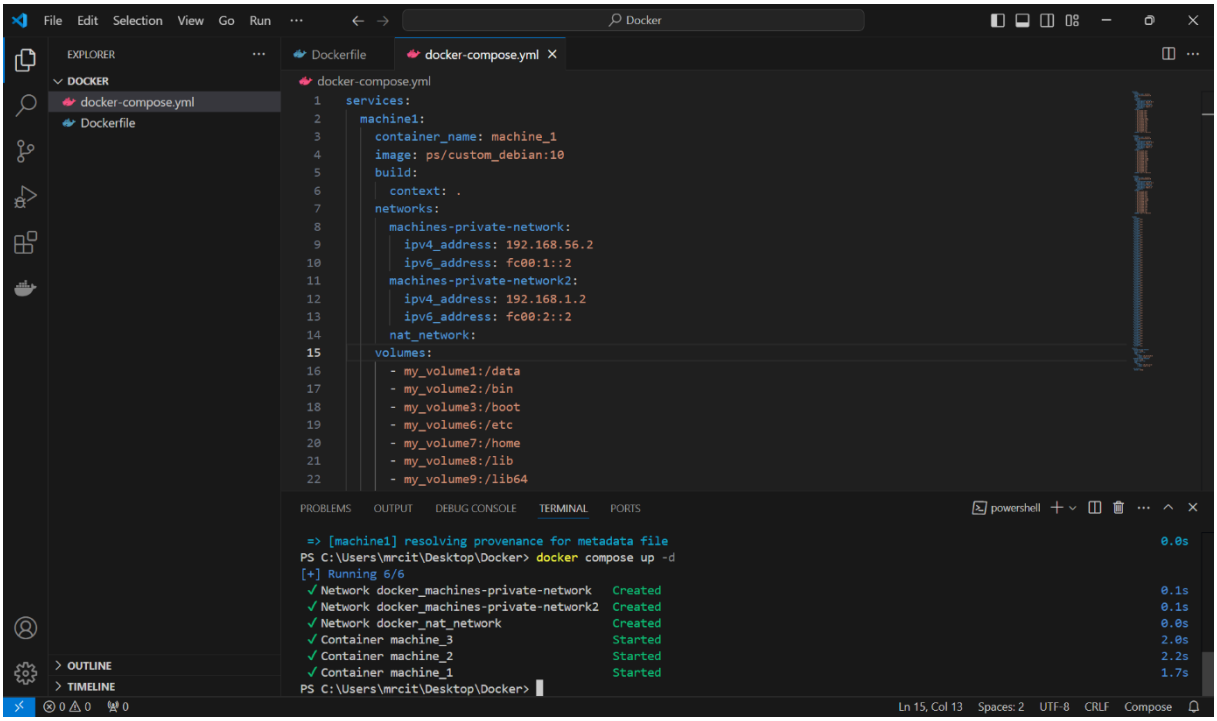
Po lewej stronie powinny pojawić nasze 2 pliki: docker-compose.yml oraz Dockerfile.

Autor: Kamil Pawłowski

Jeśli nie mamy uruchomionej konsoli w Visual Studio Code przyciskamy skrót klawiszowy CTRL+~ (słownie control i tyldę).
Wpisujemy w konsolę: docker compose build --no-cache

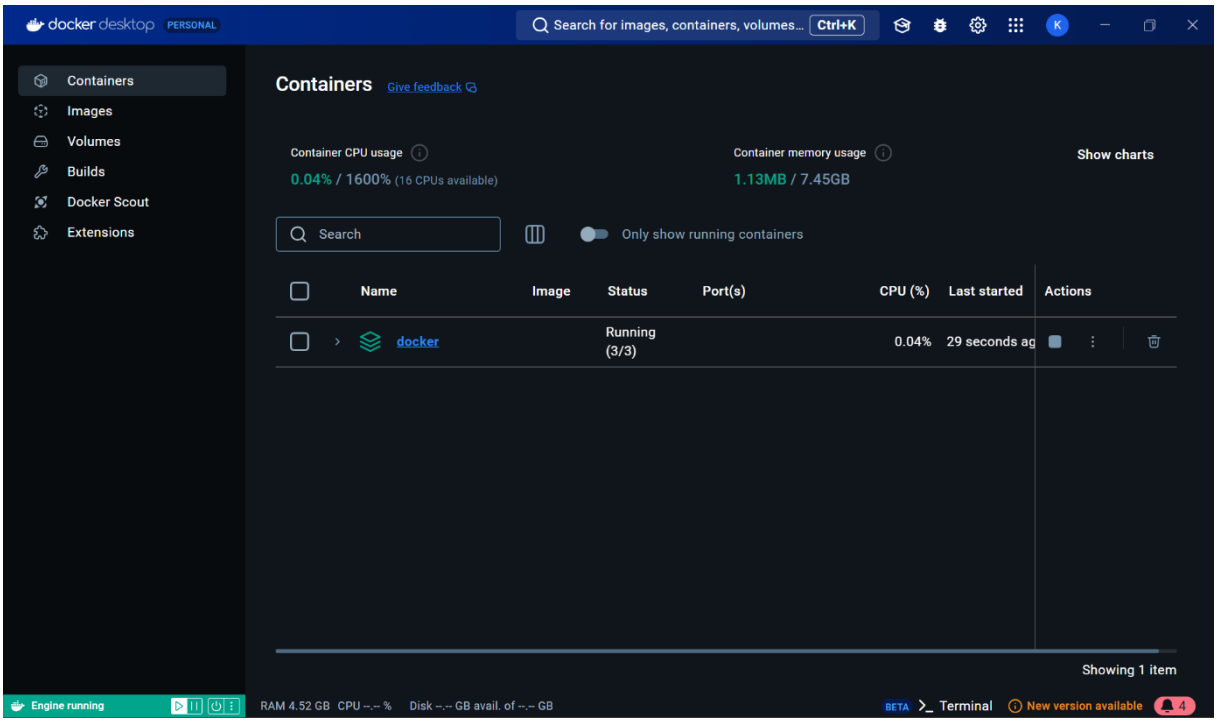


A następnie wpisujemy: docker compose up -d

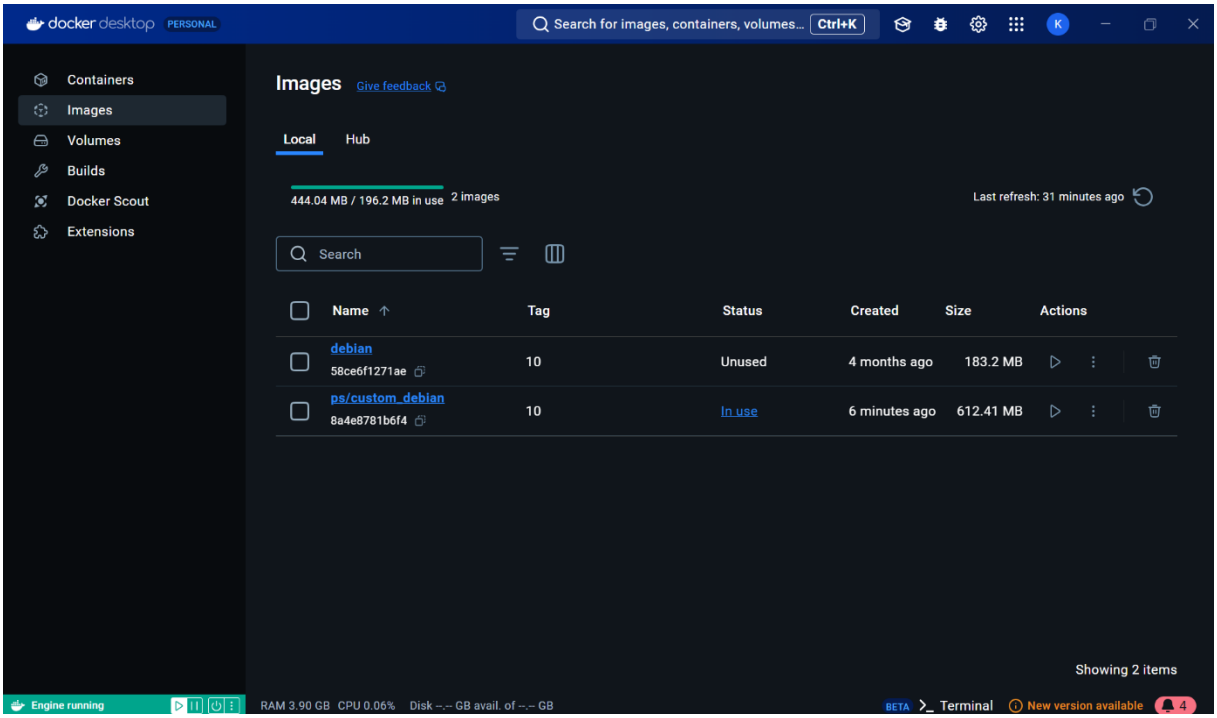


Autor: Kamil Pawłowski

Wchodzimy w program Docker'a (Docker Desktop), w zakładce „Containers" powinien pojawić się nam folder „docker".

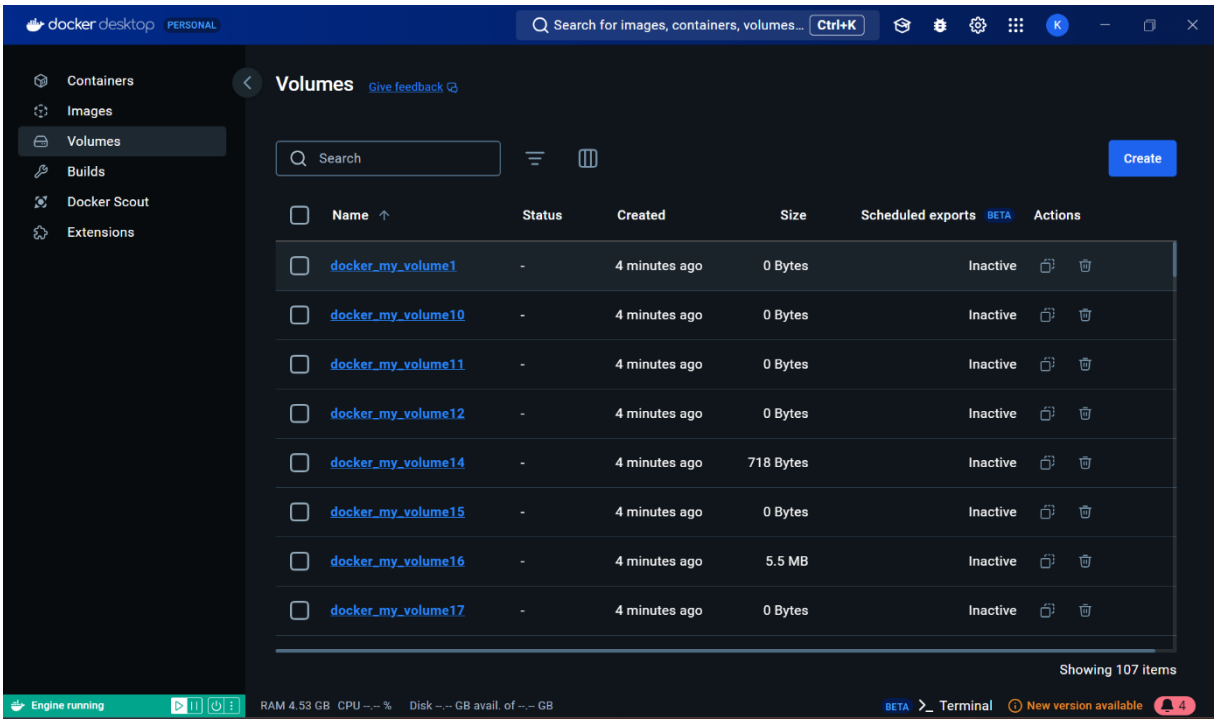


W zakładce „Images" powinien pojawić się obrazek, który stworzyliśmy, czyli „ps/custom_debian" z tagiem „10".

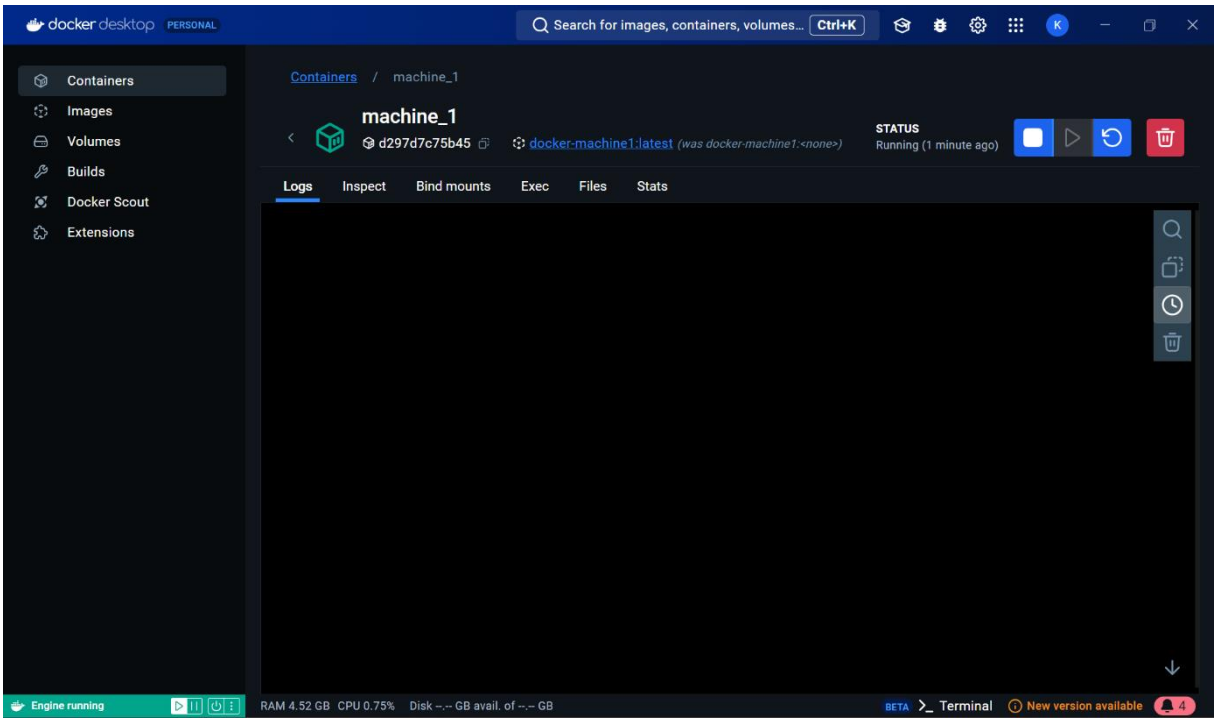


Autor: Kamil Pawłowski

Oraz w zakładce „Volumes” powinno pojawić się kilkanaście folderów jak na obrazku poniżej:

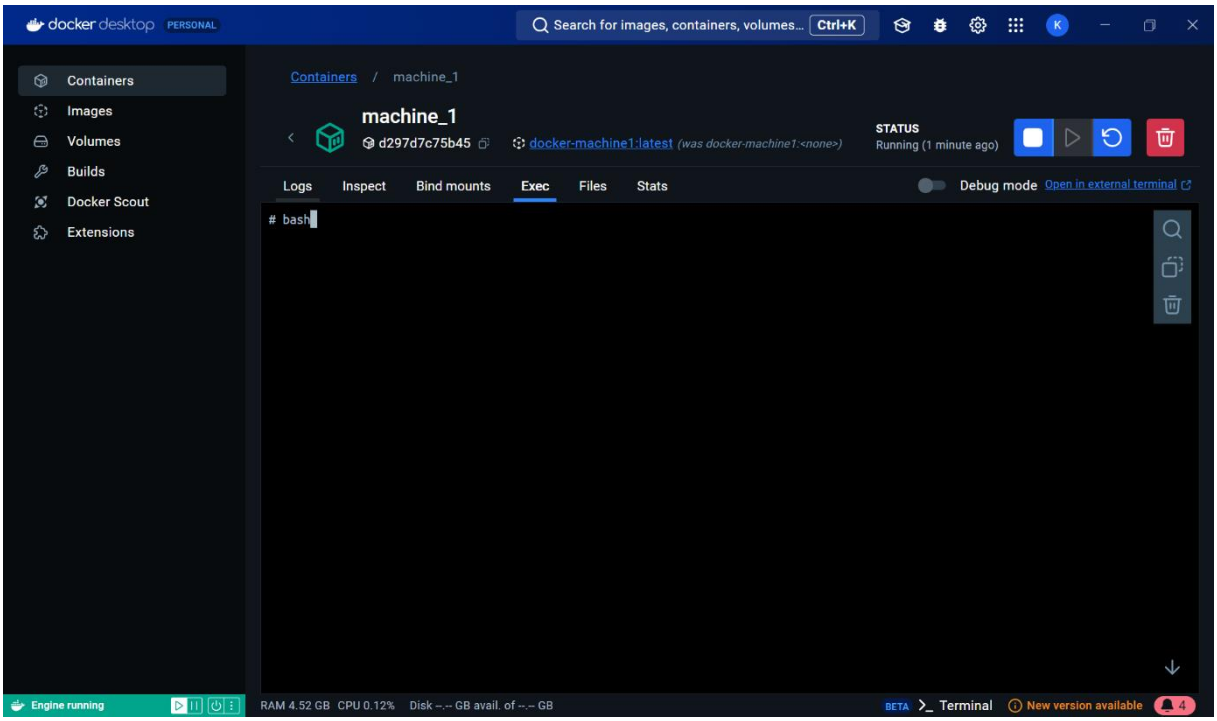


Teraz wchodzimy w zakładkę „Containers”, klikamy w „docker” i klikamy np. na „machine_1”

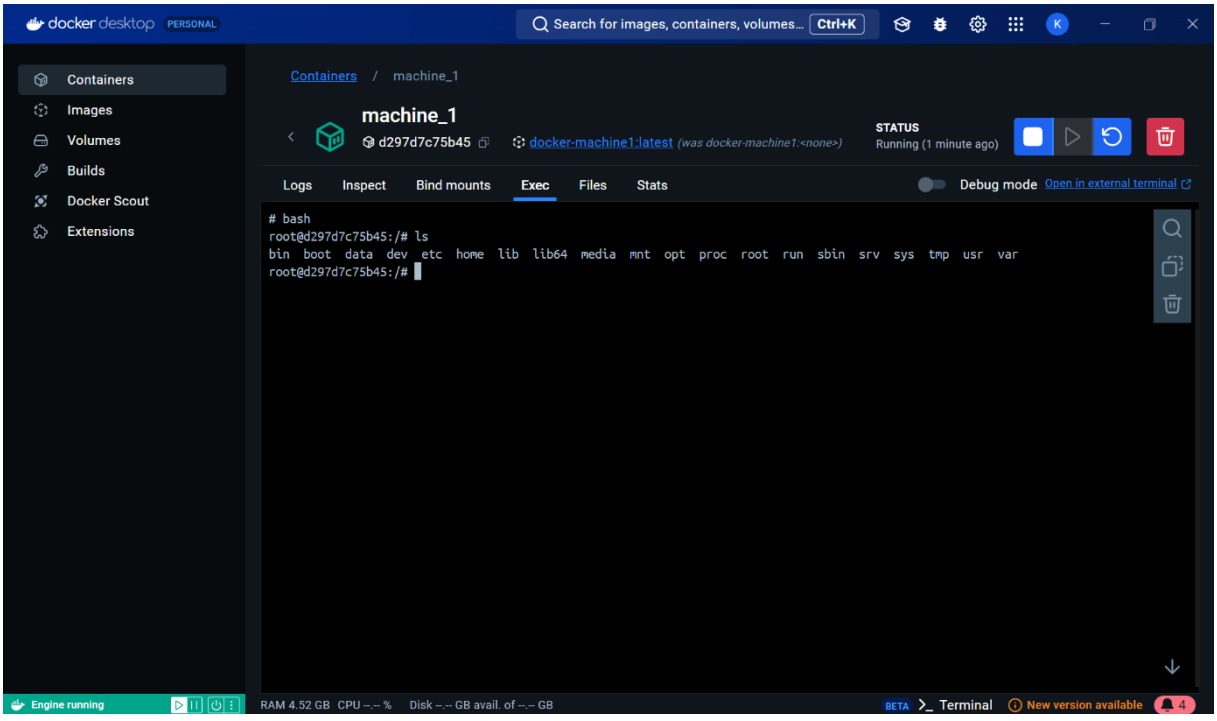


Pojawi się takie okienko jak powyżej, jesteśmy w zakładce konteneru „Logs”, ale chcemy dostać się do konsoli więc klikamy w „Exec” i wpisujemy w nią „bash”.

Autor: Kamil Pawłowski



Jak widać poniżej wszystko ładnie działa. Można korzystać z każdej z maszyn jak z osobnego komputera.



Każdy z kontenerów (maszyn) ma połączenie z internetem, jest w tej samej sieci lokalnej IPv4 oraz IPv6 oraz ma zainstalowane podstawowe narzędzia potrzebne do zajęć.
Jeśli czegoś zabraknie w konsolę konkretnego kontenera wpisujemy:

apt install <nazwa_potrzebnego_narzedzia>

Autor: Kamil Pawłowski

Po skończonej pracy w konsolę np. w Visual Code Studio wpisujemy:

```
docker compose down
```

Co kończy działanie kontenerów.

Za każdym kolejnym razem znajdując się w katalogu gdzie znajduję się nasz plik docker-compose.yml (możemy do tego użyć konsoli jakiejkolwiek, nie musi być ta z VSC), wpisujemy już tylko:

```
docker compose up -d
```

UWAGA: nie wpisujemy jeszcze wcześniej docker compose build --no-cache, to robimy tylko za pierwszym razem!

Autor: Kamil Pawłowski