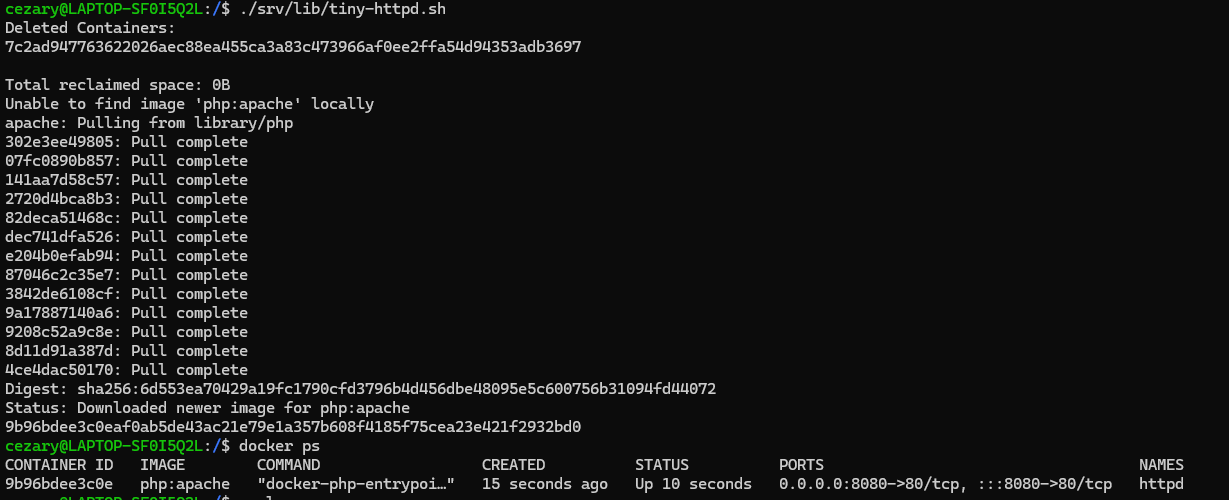
|  |  |  |  |
| --- | --- | --- | --- |
|  | Politechnika Bydgoska im. J. J. Śniadeckich  Wydział Telekomunikacji,  Informatyki i Elektrotechniki  **Zakład Systemów Teleinformatycznych** | |  |
| **Przedmiot** | Podstawy inżynierii danych | | |
| **Prowadzący** | prof. dr hab. inż. prof. PBŚ Piotr Cofta | | |
| **Temat** | *Laboratorium 1* | | |
| **Student** | Cezary Tytko | | |
| **Ocena** |  | **Data oddania spr.** | 10.10.2024 |

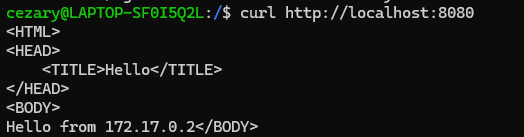
Etap 1. – Postawienie systemu.

Zdecydowałem się na rozwiązanie inne niż zaproponowane, będę wykorzystywał WSL z Ubuntu. Taki system miałem już zainstalowany, musiałem tylko doinstalować dockera, aby muc tworzyć i uruchamiać kontenery.

Etap 2. – Sprawdzenie skryptu

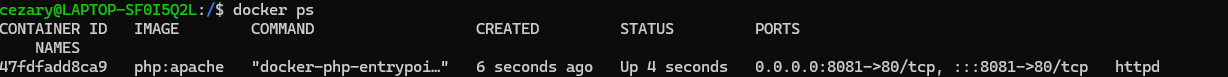
Umieściłem skrypt w odpowiednim katalogu i uruchomiłem skrypt w nie zmienionej formie, następnie odpytałem stronę curlem





Etap 3. – Modyfikacja skryptu

Zmodyfikowałem skrypt zgodnie z poleceniem, zmieniając port jak i katalog. Następnie uruchomiłem skrypt strona działa już pod nowo wskazanym portem.



Zmianę portu widać po wylistowaniu kontenerów, oraz można sprawdzić że został utworzony nowy katalog.

Etap 4. – Dwa kontenery.

Zmodyfikowałem skrypt tak aby tworzył on dwa kontenery z różnymi portami, katalogami i inną treścią strony.

1. #! /bin/bash

2. # build tiny configuration with Apache httpd with php support

3. # version 2024.03

4.

5. # set all parameters as variables

6. LIBDIR="/srv/lib"

7. WEBDIR1="/srv/www1"

8. WEBDIR2="/srv/www2"

9. CNAME1="httpd1"

10. CNAME2="httpd2"

11.

12. # make sure all required software is present

13. sudo apt-get install -y ntp docker.io >/dev/null 2>/dev/null

14.

15. # stop and remove the old version of this container as well as unused ones

16. docker stop "$CNAME1" 2>/dev/null

17. docker rm "$CNAME1" 2>/dev/null

18.

19. docker stop "$CNAME2" 2>/dev/null

20. docker rm "$CNAME2" 2>/dev/null

21.

22. docker container prune -f 2>/dev/null

23.

24. # make sure web directory exist

25. sudo rm -r -f  $WEBDIR1 $WEBDIR2

26. sudo mkdir -p $WEBDIR1 $WEBDIR2

27. sudo chmod 777 $WEBDIR1

28. sudo chmod 777 $WEBDIR2

29.

30. # copy web content

31. cat >"$WEBDIR1/index.php" <<EOF

32. <HTML>

33. <HEAD>

34.     <TITLE>Hello from Instance 1</TITLE>

35. </HEAD>

36. <BODY>

37. <?php

38. \$ip\_server = \$\_SERVER['SERVER\_ADDR'];

39. echo "Hello from \$ip\_server - Instance 1";

40. ?>

41. </BODY>

42. EOF

43.

44. sudo chmod 744 $WEBDIR1/\*.{html,php} 2>/dev/null

45.

46. cat >"$WEBDIR2/index.php" <<EOF

47. <HTML>

48. <HEAD>

49.     <TITLE>Hello from Instance 2</TITLE>

50. </HEAD>

51. <BODY>

52. <?php

53. \$ip\_server = \$\_SERVER['SERVER\_ADDR'];

54. echo "Hello from \$ip\_server - Instance 2";

55. ?>

56. </BODY>

57. EOF

58.

59. sudo chmod 744 $WEBDIR2/\*.{html,php} 2>/dev/null

60.

61. # start the container

62. docker run \

63.   --hostname "$CNAME1" \

64.   --publish "8081:80" \

65.   --volume "$WEBDIR1:/var/www/html" \

66.   --name "$CNAME1" \

67.   --detach \

68.   php:apache

69.

70. # start the second container

71. docker run \

72.   --hostname "$CNAME2" \

73.   --publish "8082:80" \

74.   --volume "$WEBDIR2:/var/www/html" \

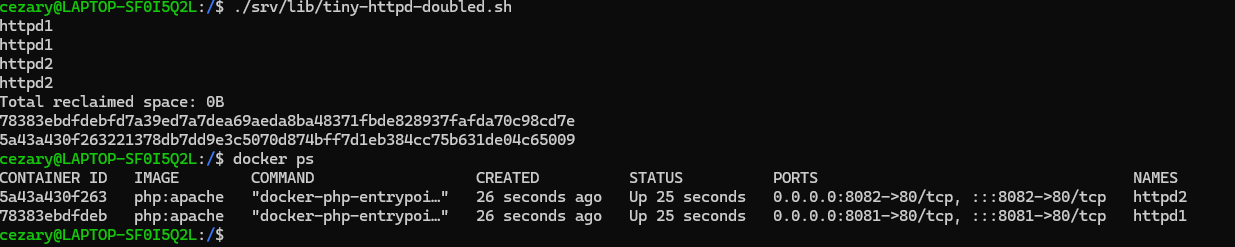
75.   --name "$CNAME2" \

76.   --detach \

77.   php:apache

78.

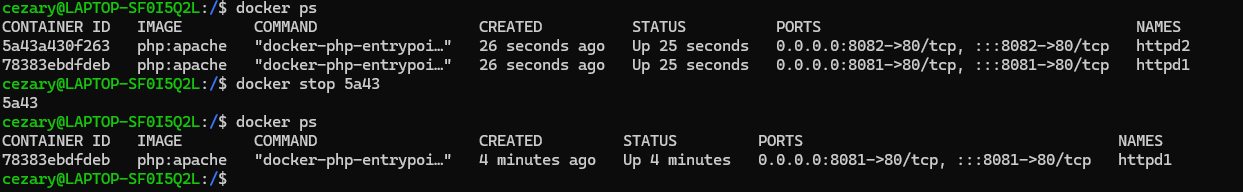
Uruchomienie skryptu utworzyło dwa kontenery, i obie strony działały jednocześnie







Zatrzymałem tylko jeden z kontenerów, co skutkowało brakiem dostępu do tej strony, kiedy druga cały czas działała



Wnioski:

Ćwiczenie było dla mnie proste ponieważ już wcześniej korzystałem z konteneryzacji w pracy jak i na studiach. Użycie WSLa zamiast maszyny wirtualnej uważam za dobrą decyzję, uprościło to cały proces bez zbędnych komplikacji związanych z wirtualizacją.