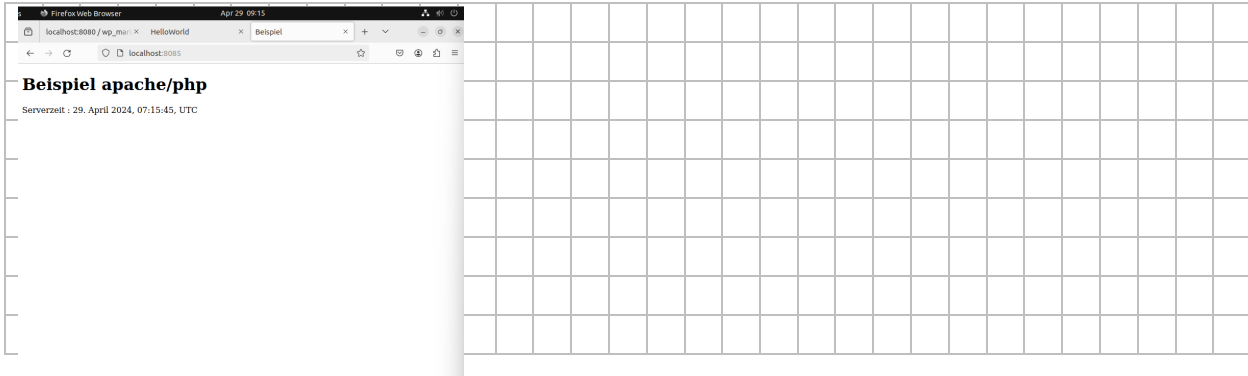


1. Ziele

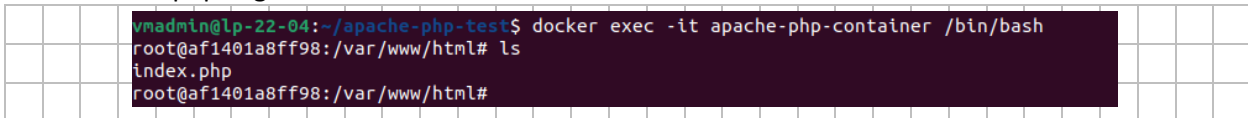
- Sie verstehen die Aufgabe von Dockerfiles und kennen die elementare Syntax

2. Aufgabe 1: Erstellen einer Webseite

- Erstellen Sie ein Dockerfile für ein Image gemäss <https://gbssg.gitlab.io/m347/images-eigene/#erstes-beispiel> und starten Sie daraus einen Container, öffnen Sie die Webseite.

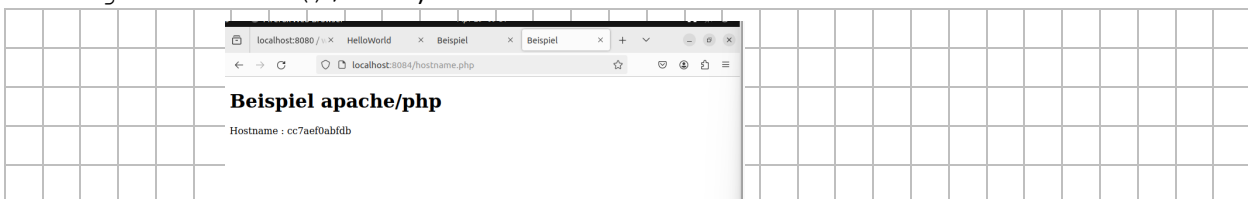


- Öffnen Sie eine interaktive Sitzung in den Container und überprüfen Sie wo die kopierte Datei index.php liegt

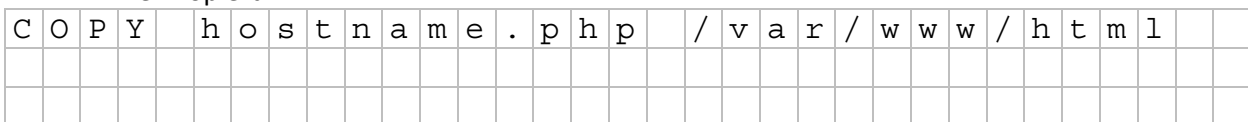


3. Aufgabe 2: Weiterentwicklung der Webseite

- Erstellen Sie in der index.php Datei einen Link auf eine zweite php-Seite (hostname.php). Diese zweite Seite soll den Hostnamen des Containers anzeigen (mit `<?php echo gethostname(); ?>`).



- Fügen im Dockerfile eine Anweisung hinzu, welches die Datei hostname.php ins Image hineinkopiert.



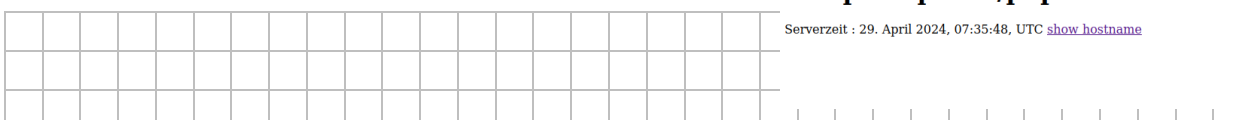
- Nun müssen Sie den Container stoppen und löschen, sowie das Image löschen und neu erstellen.
- Hier ein Bild davon, wie es am Schluss aussehen könnte:



Beispiel apache/php

Serverzeit : 16. March 2023, 09:20:44, UTC

Hostname: [hier klicken](#)



4. Aufgabe 2: COPY vs. ADD, CMD vs. ENTRYPOINT

- Recherchieren Sie den Unterschied zwischen COPY und ADD

add kopiert dateien vom host oder von einem entfernten URL. gezippte dateien werden entzippt
copy kopiert nur dateien vom host.

- Recherchieren Sie den Unterschied zwischen CMD und ENTRYPOINT

(Hinweis: Bei vielen Base-Images ist der ENTRYPOINT nicht definiert)

C M D kann überschrieben werden.

5. Aufgabe 3: Weiteres Beispiel

- Erstellen Sie ein Dockerfile für ein Image gemäss <https://gbssg.gitlab.io/m347/images-eigene/#zweites-beispiel> und starten Sie einen Container daraus

```
vmadmin@ip-22-04:~$ mkdir zweitesbsp
vmadmin@ip-22-04:~$ cd zweitesbsp/
vmadmin@ip-22-04:~/zweitesbsp$ nano Dockerfile
vmadmin@ip-22-04:~/zweitesbsp$ docker build -t meine-webseite-image .
Sending build context to Docker daemon 2.56kB
Step 1/8 : FROM ubuntu:latest
----> 7af9ba4f0a47
Step 2/8 : LABEL maintainer "name@meine-webseite.ch"
----> Running in a1ac35a8b4bc
```

- Warum werden bei RUN die Befehle mit einem && verknüpft und nicht einzeln abgesetzt (siehe <https://gbssg.gitlab.io/m347/images-eigene/#zweites-beispiel>)

mit dem && werden die run befehle verknüpft und so muss man nicht so oft run machen.

6. Hilfsmittel

<https://gbssg.gitlab.io/m347/images-eigene/>
Internet

7. Erwartete Resultate

Zeit: 40 Minuten