1.3.22

**הנדסת פתרונות תוכנה מתקדמים- המשך DOCKER**

פקודות נוספות שמריצים בתוך wsl: (דוגמאות קודמות נמצאות בקובץ EASS-Lec02-Natalie)

* ~echo print\_something

פקודה שמבצעת הדפסה

* ~cat filename

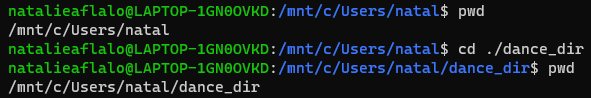
פקודה בעלת 4 שימושים נפוצים:

1. הצגת תוכן של קובץ
2. שילוב של כמה קבצים
3. הצגת טקסט
4. יצירת קובץ חדש

* ~pwd

מציג את המיקום שאני נמצא בו בתיקיות.

* ~cd /myDir/file.txt

Cd= change directory, לעבור למיקום של תיקייה או קובץ אחר

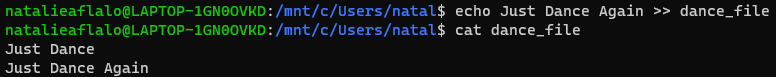
* ~<proc1>|<proc2>

לוקח STDOUT של proc1 ומזריק אותו לSTDIN של proc2.

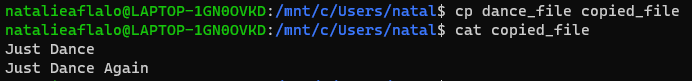
* ~command > file-disc/file

לוקח את ה-STDOUT של הפקודה ומכניס לקובץ file.

* ~command >> file-disc/file

מוסיף את ה-STDOUT של הפקודה בסוף תוכן הקובץ file, כלומר מבצע append.

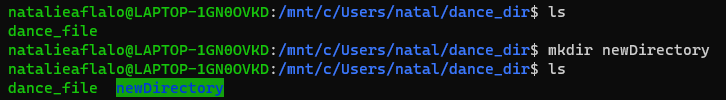
* ~cp oldFile newFile

מעתיק תוכן קובץ אחד לקובץ אחר (יוצר את הקובץ המועתק אם לא קיים)

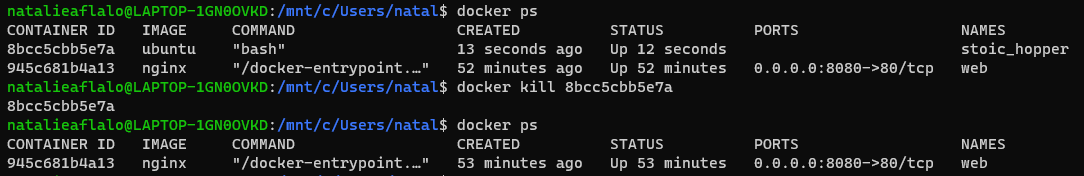
* ~mv fileName newFileName \ ./myDir

משנה שם של קובץ או משנה את המיקום שלו.

* ~mkdir directoryName

יצירת תיקייה חדשה, make directory.

* ~docker kill ContainerID

"הורג" את ה-container.

* ~docker run -ti --rm -d -v/home/username/filename : /from\_host --name hi ubuntu

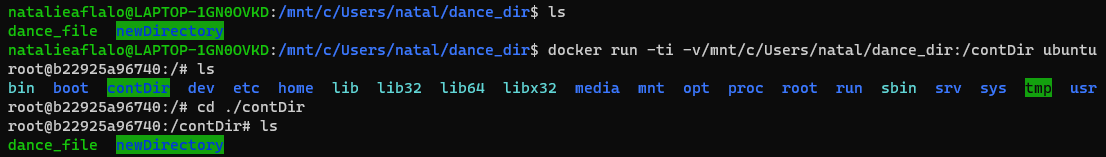
-ti טרמינל אינטרקטיבי, מאפשר לנו לתת פקודות לcontainer.

-d מאפשר להתהליך לרוץ ברקע

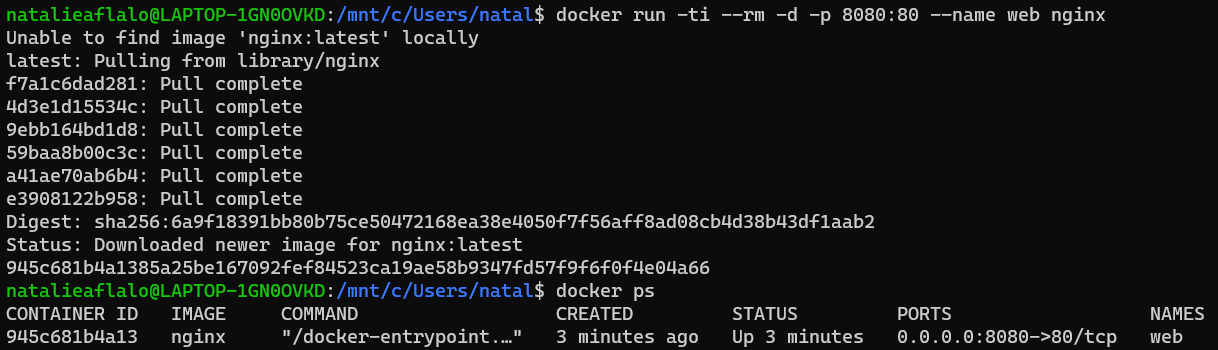
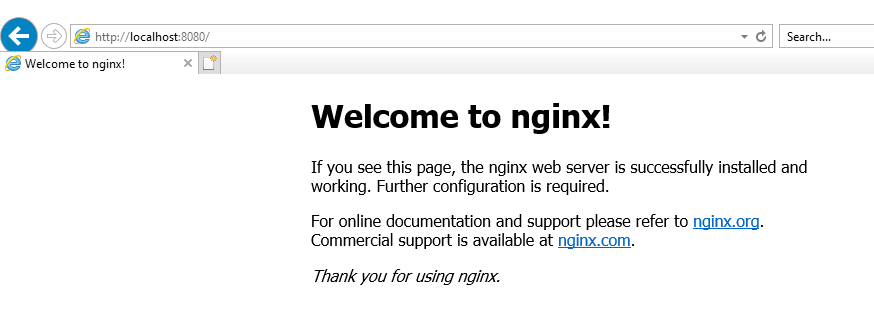
--rm מסיר את ה-container אחרי שיוצאים ממנו אוטומטית.

--name נותן שם לcontainer.

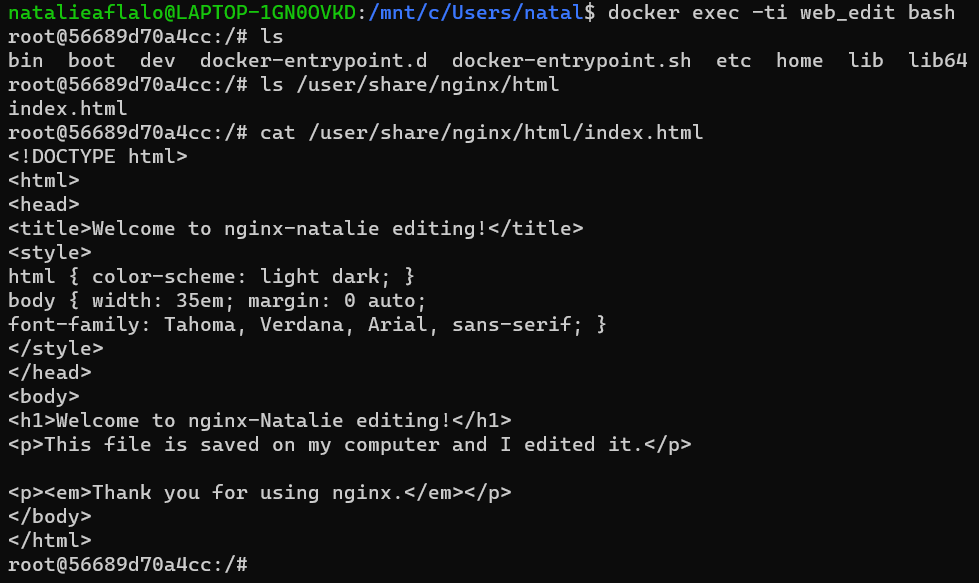
-p port, נקודת גישה למחשב

-v מבצע מיפוי דו כיווני בין תיקייה של host (המחשב הלוקלי) לבין תיקייה של ה-container. קיצור שלVolume .

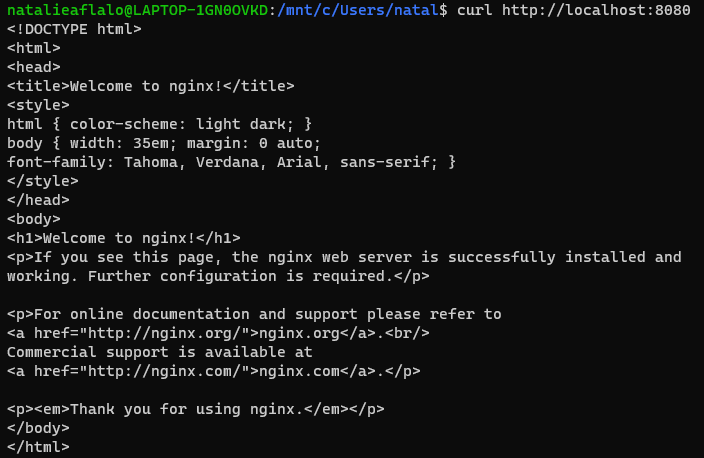
nginx

מדובר על סוג image. כאשר docker מריץ אותו ניתן לראות בפורט שרשמנו את דף ה-html שנמצא בimage. נראה כך:

כדי שנוכל לתת פקודות בתוך ה-container של nginx יש להשתמש בפקודת exec בפורמט bash. נראה כך:

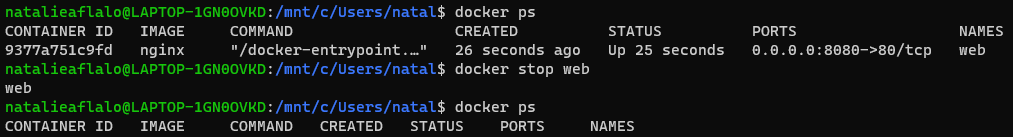


פקודת curl מקבלת כתובת url ומציגה לנו מה נמצא בקובץ ה-html . נראה כך:



קובץ הhtml הזה נקרא index.html והוא חלק מהimage שנקרא nginx.

אם נשתמש במקום בפקודת wget יווצר גם עותק של index.html לוקלית במחשב.

במקום docker kill ניתן להשתמש בdocker stop ולתת את השם של הcontainer.

אם port מסוים כבר נבחר לשימוש בcontainer מסוים, לא ניתן להשתמש בו בcontainer אחר במקביל. אם ננסה נקבל שגיאה מה-demon.

עריכת קבצים דרך WSL:

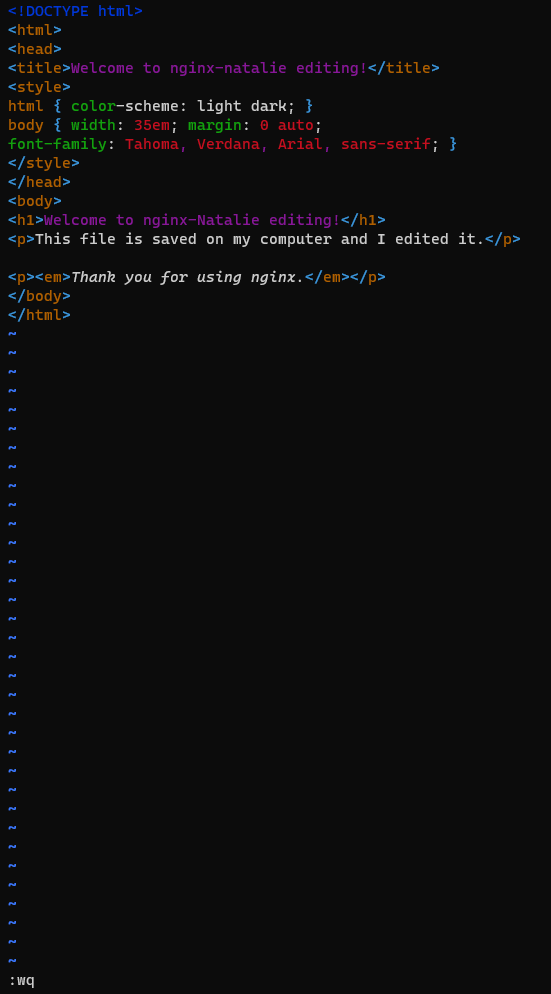
בהמשך לפקודת wget, לאחר שיצרנו את הקובץ אצלנו (ניתן להעביר אותו לתיקייה מסוימת) ניתן לערוך אותו. פקודת vi מאפשרת הצגת תוכן הקובץ ושינוי שלו.

תחילה נקליד את הפקודה ונלחץ enter.

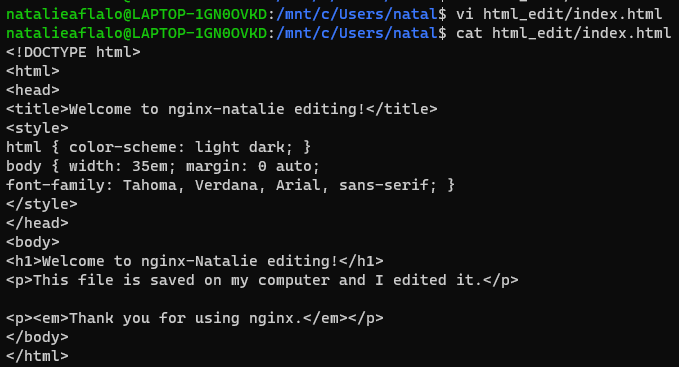


לאחר מכן ייפתח לנו תוכן הקובץ לתצוגה. כדי שנוכל לערוך, יש ללחוץ על מקש i במקלדת (כך שיופיע בתחתית הדף – INSERT --

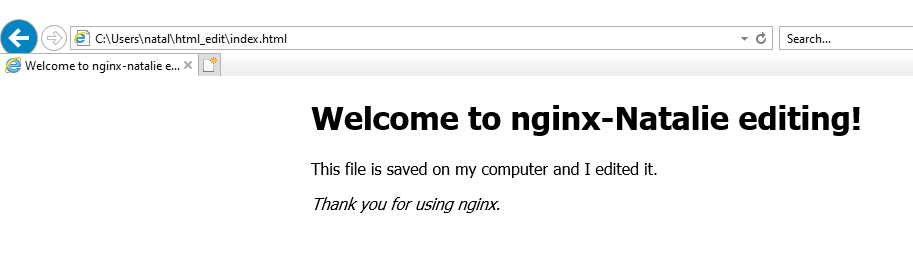
נבצע את השינויים שנרצה. בסוף העריכה נלחץ על מקש esc במקלדת. כדי לשמור את השינויים ולצאת מתצוגת הקובץ חזרה לwsl יש לרשום :wq ואז ללחוץ enter. נראה כך:



לאחר העריכה:



נראה כך בדפדפן:



שינוי בImage קיים:

יש באפשרותינו לשנות image קיים, במקרה זה אנחנו יכולים להחליף את קובץ index.html המקורי בקובץ שערכנו (ששמור בתיקייה לוקלית במחשב). כדי לבצע זאת נשתמש ב -v . נראה כך:

כעת כשנריץ את localhost:8080/ בדפדפן נקבל את הדף שערכנו (ולא את המקורי).

יצירת Image חדש:

באמצעות Dockerfile אנו יכולים לכתוב image. הוא יכול להתבסס על image קיים, ניתן להתקין בו תוכנות נוספות, להוסיף לו קבצים ועוד.

נתחחיל בעריכת dockerfile באמצעות הפקודה vi Dockerfile. לאחר לחיצה על enter נכנס למסך תצוגה. יש ללחוץ על i במקלדת כדי להיכנס למצב עריכה. כעת נכניס את תוכן האימג' :

FROM נספק לdockerfile אימג' קיים שממנו הוא יורש. חייב כאן לרשום את הגרסה של האימג'.

COPY העתקה מה-host (מחשב לוקלי) לתיקייה מסוימת בdockerfile

RUN התקנת תוכנה. לפני ההתקנות חייב לרשום שורה RUN apt-get update

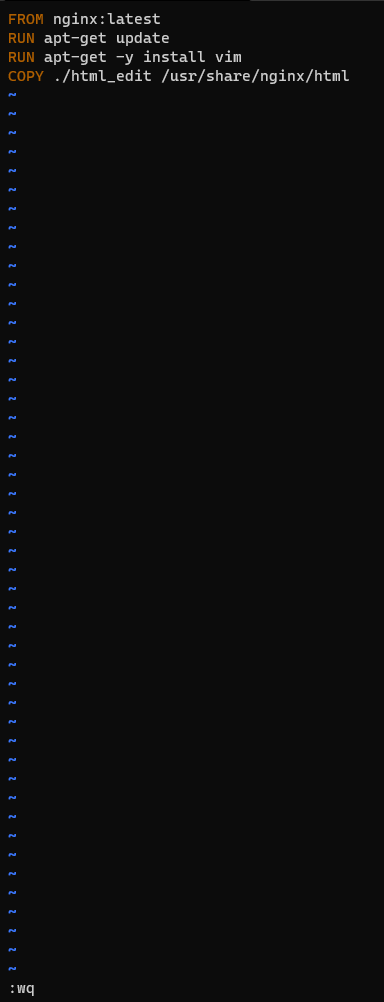
בסוף העריכה נלחץ esc במקלדת וכדי לשמור את השינויים :wq

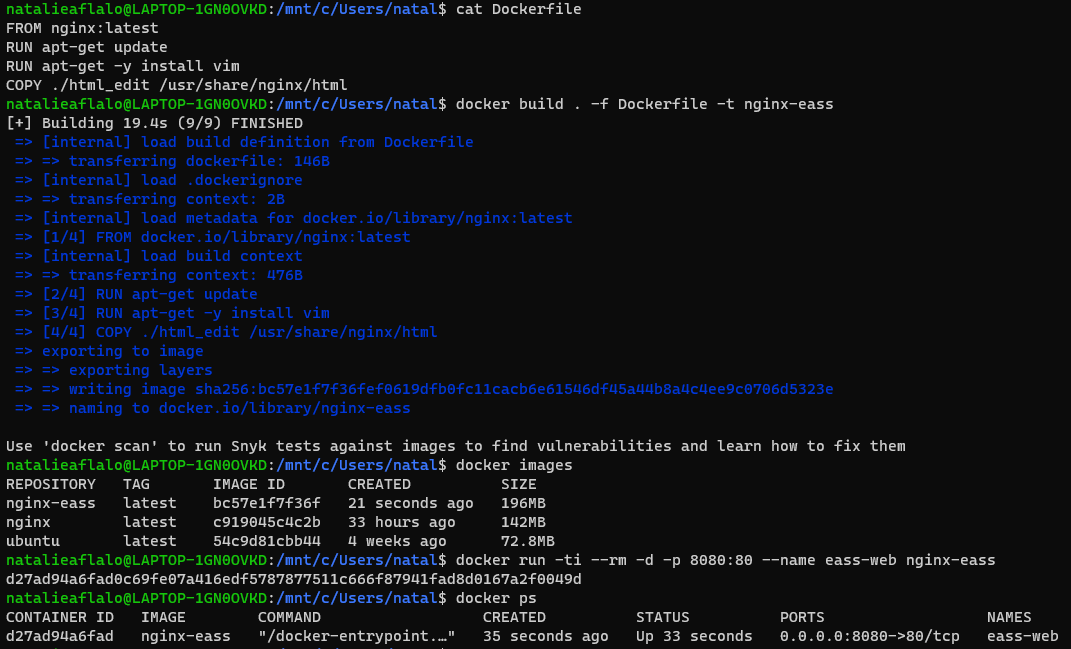
ליצירת האימג' מdockerfile נשתנש בפקודת docker build, חשוב לתת לה את נתיב התיקייה באמצעות נקודה . ולתת לה שם באמצעות -t

לבסוף ניתן ליצור container באמצעות docker run של האימג' שיצרנו.

נראה כך:







הערה חשובה:

בשורה RUN apt-get -y install vim: המשמעות של -y היא שבזמן ההתקנה כל שאלה מקבלת תשובה yes כדי שההתקנה תרוץ עד הסוף בלי תלות במשתמש.