

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

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ echo 1233
1233
```

- ~cat filename

פקודה בעלת 4 שימושים נפוצים:

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

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ cat a.txt
123
```

- ~pwd

מציג את המיקום שאני נמצא בו בתיקיות.

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ pwd
/mnt/c/Users/natal
```

- ~cd /myDir/file.txt

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

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ pwd
/mnt/c/Users/natal
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ cd ./dance_dir
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal/dance_dir$ pwd
/mnt/c/Users/natal/dance_dir
```

- ~<proc1>|<proc2>

לוקח STDOUT של proc1 ומזריק אותו לSTDIN של proc2. נקרא pipe

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ echo 123 | cat > a.txt
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ cat a.txt
123
```

- ~command > file-disc/file

לוקח את ה-STDOUT של הפקודה ומכניס לקובץ file.

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ echo Just Dance > dance_file
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ cat dance_file
Just Dance
```

- ~command >> file-disc/file

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

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ echo Just Dance Again >> dance_file
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ cat dance_file
Just Dance
Just Dance Again
```

- `~cp oldFile newFile`

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

```
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ cp dance_file copied_file
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ cat copied_file
Just Dance
Just Dance Again
```

- `~mv fileName newFileName \ ./myDir`

משנה שם של קובץ או משנה את המיקום שלו.

```
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ mv dance_file ./dance_dir
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ ls ./dance_dir
dance_file
```

- `~mkdir directoryName`

יצירת תיקייה חדשה, make directory.

```
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal/dance_dir$ ls
dance_file
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal/dance_dir$ mkdir newDirectory
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal/dance_dir$ ls
dance_file  newDirectory
```

- `~docker kill ContainerID`

"הורג" את ה-container.

```
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
8bcc5cbb5e7a   ubuntu   "bash"                  13 seconds ago Up 12 seconds                stoic_hopper
945c681b4a13   nginx    "/docker-entrypoint..." 52 minutes ago Up 52 minutes    0.0.0.0:8080->80/tcp      web
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker kill 8bcc5cbb5e7a
8bcc5cbb5e7a
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
945c681b4a13   nginx    "/docker-entrypoint..." 53 minutes ago Up 53 minutes    0.0.0.0:8080->80/tcp      web
```

- `~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 של Volume.

```
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal/dance_dir$ ls
dance_file  newDirectory
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal/dance_dir$ docker run -ti -v/mnt/c/Users/natal/dance_dir:/contDir ubuntu
root@b22925a96740:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr
root@b22925a96740:/# cd ./contDir
root@b22925a96740:/contDir# ls
dance_file  newDirectory
```

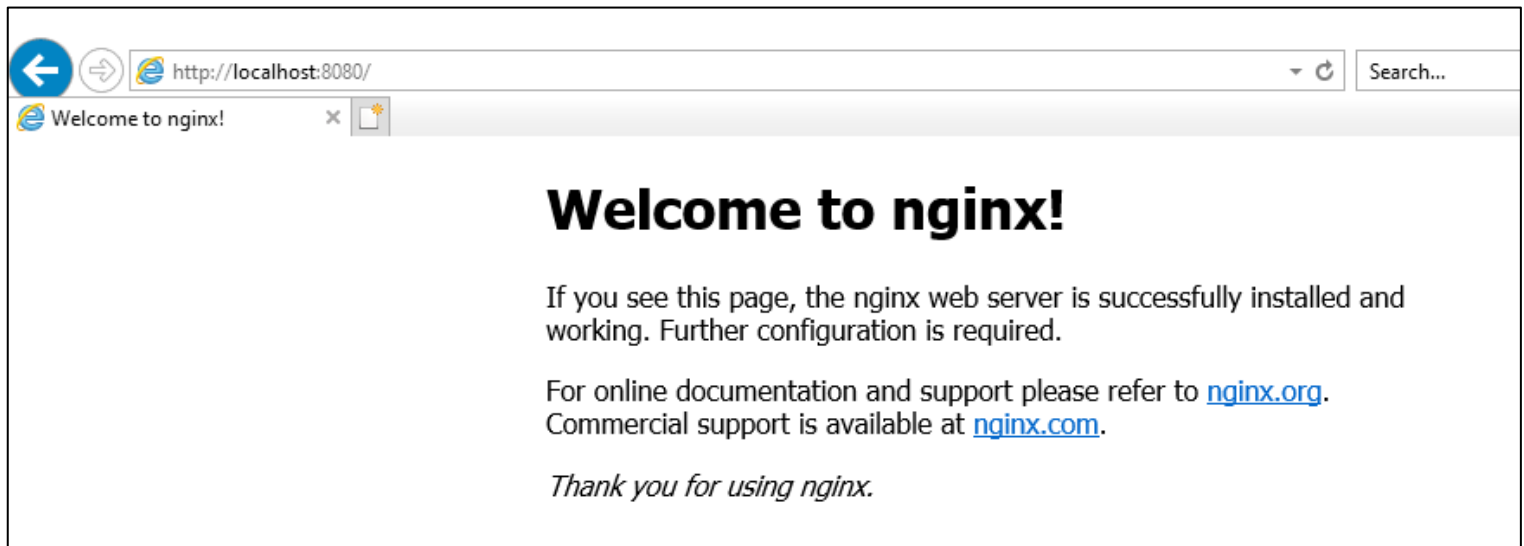
- `~docker logs`

מציג פירוט לוגים של container. אמצעי לזיהוי שגיאות (debugging)

nginx

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

```
natalieaf1alo@LAPTOP-1GN00VKD:/mnt/c/Users/natal$ docker run -ti --rm -d -p 8080:80 --name web nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
f7a1c6dad281: Pull complete
4d3e1d15534c: Pull complete
9ebb164bd1d8: Pull complete
59baa8b00c3c: Pull complete
a41ae70ab6b4: Pull complete
e3908122b958: Pull complete
Digest: sha256:6a9f18391bb80b75ce50472168ea38e4050f7f56aff8ad08cb4d38b43df1aab2
Status: Downloaded newer image for nginx:latest
945c681b4a1385a25be167092fef84523ca19ae58b9347fd57f9f6f0f4e04a66
natalieaf1alo@LAPTOP-1GN00VKD:/mnt/c/Users/natal$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
945c681b4a13   nginx    "/docker-entrypoint. ..." 3 minutes ago  Up 3 minutes  0.0.0.0:8080->80/tcp   web
```



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

```
natalieaf1alo@LAPTOP-1GN00VKD:/mnt/c/Users/natal$ docker exec -ti web_edit bash
root@56689d70a4cc:/# ls
bin boot dev docker-entrypoint.d docker-entrypoint.sh etc home lib lib64
root@56689d70a4cc:/# ls /user/share/nginx/html
index.html
root@56689d70a4cc:/# cat /user/share/nginx/html/index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx-natalie editing!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx-Natalie editing!</h1>
<p>This file is saved on my computer and I edited it.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@56689d70a4cc:/#
```

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

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ curl http://localhost:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

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

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

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

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
9377a751c9fd   nginx    "/docker-entrypoint...." 26 seconds ago Up 25 seconds  0.0.0.0:8080->80/tcp      web
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker stop web
web
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
```

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.

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ vi html_edit/index.html
```

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 ל'wsl' יש לרשום wq: ואז ללחוץ enter. נראה כך:

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx-natalie editing!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx-Natalie editing!</h1>
<p>This file is saved on my computer and I edited it.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

:wq

לאחר העריכה:

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ vi html_edit/index.html
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ cat html_edit/index.html
<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx-natalie editing!</title>

<style>

html { color-scheme: light dark; }

body { width: 35em; margin: 0 auto;

font-family: Tahoma, Verdana, Arial, sans-serif; }

</style>

</head>

<body>

<h1>Welcome to nginx-Natalie editing!</h1>

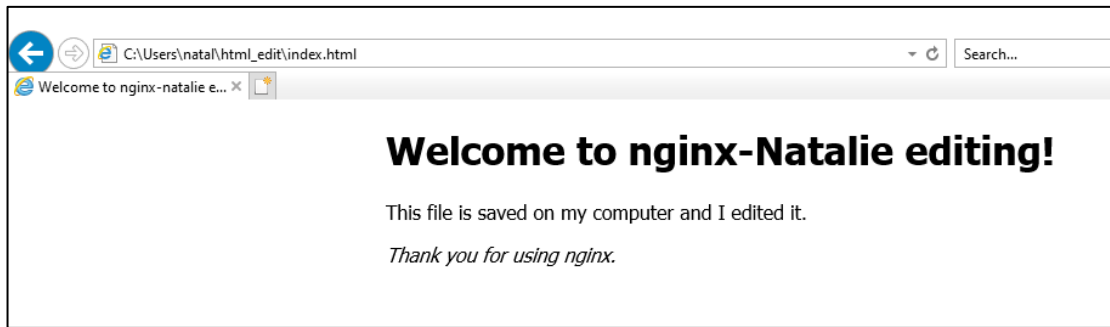
<p>This file is saved on my computer and I edited it.</p>

<p><em>Thank you for using nginx.</em></p>

</body>

</html>
```

נראה כך בדפדפן:



שינוי Image 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:

```
natalieaf1alo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker run -ti --rm -d -p 8080:80 -v /mnt/c/Users/natal/html_edit:/usr/share/nginx/html --name web_edit nginx
05490250d4122ebf612a82eef4eda5268a25cee2d9d549cee046302ab4fc6199
natalieaf1alo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ curl localhost:8080/
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx-natalie editing!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx-Natalie editing!</h1>
<p>This file is saved on my computer and I edited it.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

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.

נתחיל 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 apt-get update

בסוף העריכה נלחץ על esc במקלדת וכדי לשמור את השינויים:wq

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

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

נראה כך:

```
natalieaflalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ vi Dockerfile
```

```
FROM nginx:latest
RUN apt-get update
RUN apt-get -y install vim
COPY ./html_edit /usr/share/nginx/html
```

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

:wq

```

natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ cat Dockerfile
FROM nginx:latest
RUN apt-get update
RUN apt-get -y install vim
COPY ./html_edit /usr/share/nginx/html
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker build . -f Dockerfile -t nginx-eass
[+] Building 19.4s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 146B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [1/4] FROM docker.io/library/nginx:latest
=> [internal] load build context
=> => transferring context: 476B
=> [2/4] RUN apt-get update
=> [3/4] RUN apt-get -y install vim
=> [4/4] COPY ./html_edit /usr/share/nginx/html
=> exporting to image
=> => exporting layers
=> => writing image sha256:bc57e1f7f36fef0619dfb0fc11cacb6e61546df45a44b8a4c4ee9c0706d5323e
=> => naming to docker.io/library/nginx-eass

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx-eass    latest    bc57e1f7f36f   21 seconds ago 196MB
nginx         latest    c919045c4c2b   33 hours ago   142MB
ubuntu        latest    54c9d81cbb44   4 weeks ago    72.8MB
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker run -ti --rm -d -p 8080:80 --name eass-web nginx-eass
d27ad94a6fad0c69fe07a416edf5787877511c666f87941fad8d0167a2f0049d
natalieafalalo@LAPTOP-1GN0OVKD:/mnt/c/Users/natal$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
d27ad94a6fad   nginx-eass  "/docker-entrypoint. ..." 35 seconds ago Up 33 seconds 0.0.0.0:8080->80/tcp      eass-web

```

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.