

מהו דוקר ?

בתכלס זהו פרויקט קוד פתוח הכתוב בשפת לינוקס המאפשר התקנה והרצת ישומים בתוך מכולות אשר תכונתם הוא בבידוד המשאבים. כל קונטיינר מוגדר בקובץ הגדרות. הדוקר לוקח קובץ זה ויוצר קונטיינר שהיא למעשה מכונה וירטואלית עם מה שאנחנו רוצים ומוגדרת במערכת קבצים שמתממשקת אלינו, פורט משל עצמה וכו'. כל זה נעשה בקלות ובמהירות.

היתרון של קונטיינר מול מכונה וירטואלית - קונטיינר עולה תוך שניות בעוד שמכונה וירטואלית היא איטית הרבה יותר לפעולה. קונטיינר מוגדרת בקלות באמצעות קובץ הגדרות בעוד שמכונה וירטואלית לא.

מס' פקודות חשובות בדוקר

ראשית נעלה את ה-power shell או ה-cmd וזאת לאחר שהתקנו את הדוקר במחשב (יש לשים לב שכדי להשתמש בתוכנה יש להפעיל את ה-HyperV בחלונות ולהשתמש בגרסה Pro/Enterprise 10 בגרסת Home לא ניתן להשתמש בדוקר).

```
$ docker version
```

תציג את הגרסה של הדוקר המותקן במחשב -הן לשרת והן לקליינט

```
$ docker images
```

תציג את רשימת קבצי ההגדרות במחשב

```
$ docker pull image_name
```

מוריד למחשב שלנו קובץ קיים ודיפולטיבי מהמאגר של Docker

במחשב שלי החלטתי להוריד את ה-images הבאים:

- selenium/node-chrome-debug (connected to a Grid Hub)
Grid Node with Chrome installed and runs a VNC server, needs to be)

- selenium/node-firefox-debug (connected to a Grid Hub)
Grid Node with Firefox installed and runs a VNC server, needs to be)

- selenium/node-firefox (Grid Node with Firefox installed, needs to be connected to a Grid Hub)
- selenium/hub (image for running a Selenium Grid Hub)

כדי לראות את ה-UI של הדפדפנים דרך שרת ה-VLC התקנתי את התוכנה VNC Viewer.

```
$ docker run -d --name selenium-hub selenium/hub
```

מריץ את קובץ ההתקנה וברירת המחדל פורט 4444

```
$ docker ps
```

מראה לנו את המכולות שמורצות ברקע

```
$ docker logs my_container_id
```

מראה לנו את כל הפרטים אודות הקונטיינר

```
$ docker stop my_container_id
```

מפסיק את פעולת הקונטיינר

```
$ docker rm my_container_id
```

מסיר את הקונטיינר מהמחשב

בפקודה הראשונה שבה הרצנו את קובץ ההתקנה יצרנו קונטיינר עם פורט פנימי שקיבל את ברירת המחדל 4444 אבל כאשר ברצוננו לחשוף פורט חיצוני כדי לבצע טסטים דרך קוד שבמחשב שלנו אנו צריכים להגדיר גם פורט חיצוני. ולכן נשתמש בפקודה הבאה:

```
$ docker run -d -p 4446:4444 --name selenium-hub -P selenium/hub
```

כעת אנו נוכל להשתמש ביישום שלנו המקומי כדי להתחבר לHUB.
קודם לכן הורדתי למחשב המקומי 3 images הכוללים NODE ולשם הדוגמא אשתמש ב2 מתוכם :
selenium/node-chrome-debug & selenium/node-firefox-debug

נשתמש בפקודות הבאה כדי לקשר בין Nodesn לבין ה-HUB:

```
$ docker run -d -P --link selenium-hub:hub selenium/node-firefox-debug  
$ docker run -d -P --link selenium-hub:hub selenium/node-chrome-debug
```

לאחר שקישרנו 2 קונטיינרים לHUB ניכנס לViewer VNC ונכניס את הפורט 127.0.0.0:32768
את התוספת "32768" לקחנו מהקונטיינר selenium/node-chrome-debug לאחר שהשתמשנו בפקודה

```
$ docker ps
```

לאחר הכנסת הפורט אנו נכניס סיסמה בשם "secret" שהיא למעשה ברירת המחדל
שלאחריו יופיע ה-UI של הקונטיינר וכרגע יציג לוגו של מערכת ההפעלה אובונטו

נוריד את הקוד שעשיתי ונמצא בGit בכתובת: https://github.com/amiryiz/Daka90_Automation_Maven

בקוד יצרתי 2 טסטים שרצים במקביל באותה מכונה אבל ב2 טרידים שונים.
את הקוד אשנה כך שעדיין אריץ את שני הטסטים במקביל אבל בשני מכונות שונות (קונטיינרים שונים) דרך
ההתחברות לHUB. כל קונטיינר יבצע את הטסט שבקוד בשני דפדפנים שונים בכרום ופיירפוקס.

שים לב לפעמים קורה המכולות במחשב המרוחק לא מגיבות לכן מומלץ לבצע restart דרך Desktop Docker
לאשר את ההודעה שאתחול תעצור את כל הקונטיינרים אשר רצים.

ולהשתמש בפקודה: במקרה שלנו לכל שלושת המכולות

```
$ docker restart my_container_id
```