- SDN - הוראות הרצה לפרויקט

שלב הכנת הפרויקט

הקבצים שצריך בשביל להריץ את הפרויקט שלי (המצורפים בקובץ zip. המטלה):



SimpleBalancerController.py | mininerLoadBalancer.py | run_controller | run_mn | run

:אז ככה

לאחר התקנת ה mininet & pox באופן הבא:

תכנס לתיקיית הhome שלך, ותפתח שמה את ה terminal ותריץ את הפקודה הבאה:

git clone git://github.com/mininet/mininet

זה הוריד לך את תיקיית הmininet בתיקיית ה home, לאחר ההתקנה בטרמינל הנוכחי, תקליד:

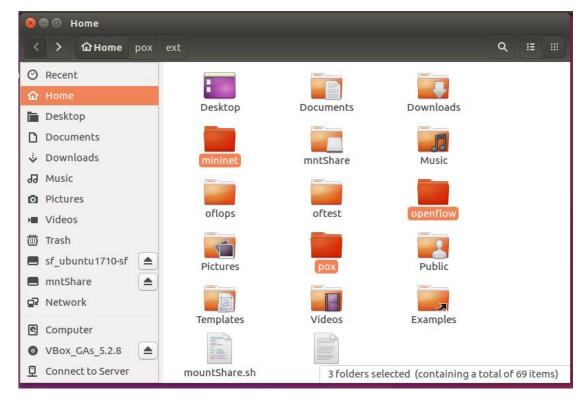
mininet/util/install.sh [options]

sudo apt-get install mininet

וכן את הפקודה:

עבור Path על מנת שיתקין את ה

וזה יתקין את כל מה שצריך, כולל את תיקיית הpox (ותיקיית) בתיקיית הhome , זה יראה דבר כזה



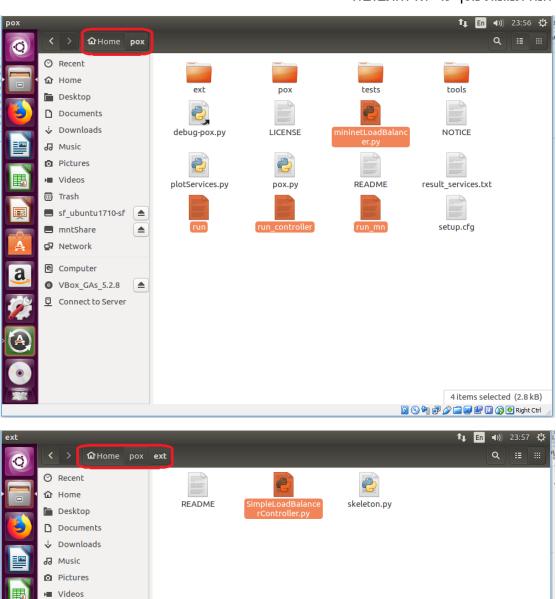
http://mininet.org/download/ יש להיכנס לאתר : mininet והאסק יש להיכנס לאתר

שלב הרצת הפרויקט

יש למקם את הקבצים הבאים בתיקיות הבאות:

+ home
| + pox
| | run
| | run_mn
| | run_controller
| | mininetLoadBalancer.py
| | + ext
| | | SimpleLoadBalancerController.py

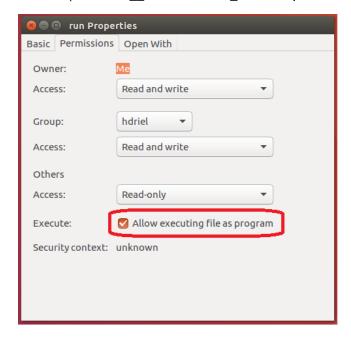
הנה תמונות מסך כדי לא להתבלבל:



(בתיקיית הxo) run בתיקייה של הקובץ רun בתיקיית של הקובץ

ולהקליד:

כדי לוודא שזה עובד, תוודא במצב properties>Permission>Allow executing file as program נמצא במצב (run , run_controller, run_mn ; לבדוק את זה בקבצים: check



ניתן להיכנס לזה בלחיצה ימנית עם העכבר על הקובץ, ואז בתחתית האפשרויות יש properties: לאחר שתריצו את הפקודה run/. ייפתחו 2 חלונות טרמינל לפרויקט, וזה יראה דבר כזה:

: ניתן לראות עבור הרצת הפרויקט שלנו, אנחנו נכתוב את הפקודות של

```
$ c1 ping 10.1.2.3 -c 1
```

\$ c2 ping 10.1.2.3 -c 1

\$ c3 ping 10.1.2.3 -c 1

\$ c4 ping 10.1.2.3 -c 1

כל שורה כזאת פותחת request בין ci לשרת , והשרת בוחר לכל service request רנדומלי, ומתקשר ביניהם, ובכך מפחית את העומס על שרת כלהו, ומוכח שהשיטה הרנדומלית מפזרת את החלוקה בצורה יחסית שווה במספר גדול יחסית, למשל n=10,000 פקטות .

הערה: לכל request יש 10 שניות של שידור, ולכן אחרי ששלחנו ping, נמתין 10 שניות ואז נוכל request לשלוח שוב, ונקבל request חדש servicel רנדומלי חדש.

בסוף הפרויקט אחרי שנסיים , נלחץ על Ctrl+C בארן , נכתבים בקובץ , נכתבים בקובץ , result_services.txt

הערות נוספות

הערה: בשביל להשתמש בספריית matplotlib יש להתקין אותה בצורה הבאה:

\$ sudo apt install python-numpy

\$ sudo apt install python-matplotlib

לבסוף, כאשר ניצא מה controller (לפני שיוצאים מהmininet) נלחץ על Ctrl+C, ואז נראה את הפלט הבסוף, כאשר ניצא מה

```
packet.ARP_TYPE
arp_packet.protosts: 10.0.0.1 (c1)
arp_packet.protosts: 10.1.2.3 (mainServer)
arp_REQUEST

The amount of requests that handel each service of Load Balancer

packet.IP_TYPE
str(packet.next.srcip): 10.0.0.4
str(packet.next.srcip): 00:00:00:00:00:00
str(packet.next.dst)p): 10.1.2.3
str(packet.next.dst)p): 10.1.2.3
str(packet.next.dst)p): 10.1.2.3
str(packet.next.dst)p): 00:00:00:00:00:00:01
flag_sclinetSendToCiserver: True
flag_serverSendToCitent: False
send packet from 'CLIENT' to 'SERVER' A
ddd new maphins: {
        "00:00:00:00:00:00:00:00:00:00:00:00:00
}
Server selected 00:00:00:00:00:00:00:00:00
}
Server selected 00:00:00:00:00:00:00:00

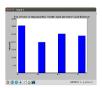
**CINFO:core:Going down...
INFO:openFlow.of 01:[00-00-00-00-00-12] disconnected
connectionDown event fired - init setup switch s 1 !
Anount Handleing services:
st: 1
s2: 2
s3: 5
s4: 2

write the result to file 'result_services.txt'

**Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the 'result_services.txt' **Total Content of the
```

<u>הערה:</u> במידה והגרף נתקע או לא עובד דרך הכתונים , הנתונים נכתבים לקובץ zip אותו דרך, י~/pox/plotService.py, הנמצא בקובץ, י*result_services.txt של המטלה, יש להריץ את הקובץ הזה כשהוא בתוך התיקייה POX, בעזרת הפקודה:

\$ sudo python plotServices.txt



ונקבל את הפלט הבא למשל: