Project/Lab 2

Submitted by : Nandini Krupa Krishnamurthy

Task:

To launch a DoS attack on the SDN data plane and explain the attack consequences.

Experiment:

All the steps are explained along with the screenshots.

Step1 : Starting the experiment by creating a profile.
Figures starting from Fig1 to Fig6 show the steps needed to start this experiment. It is essential to choose DosServer to demonstrate DoS attack as shown in the figures below.

Note: As the topology used is default to DosServer, no changes to topology have been made in this experiment.

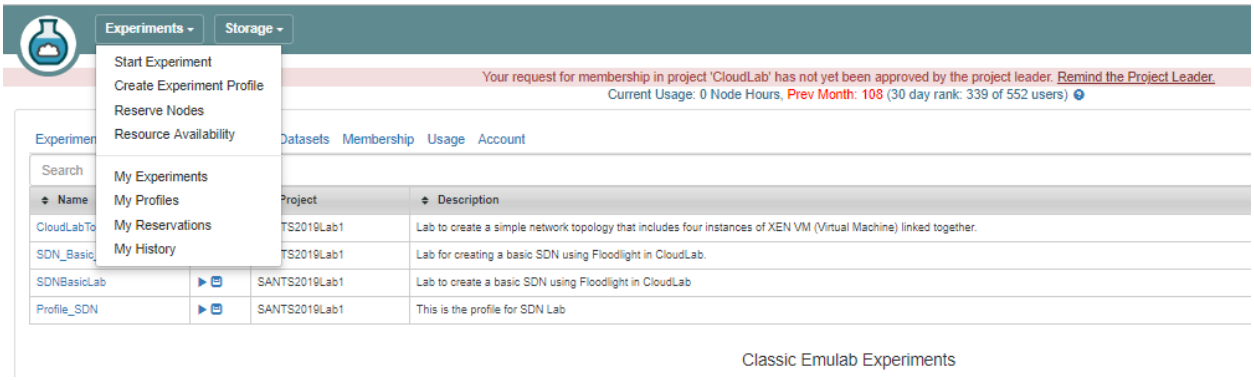

Fig 1 – Starting an Experiment.

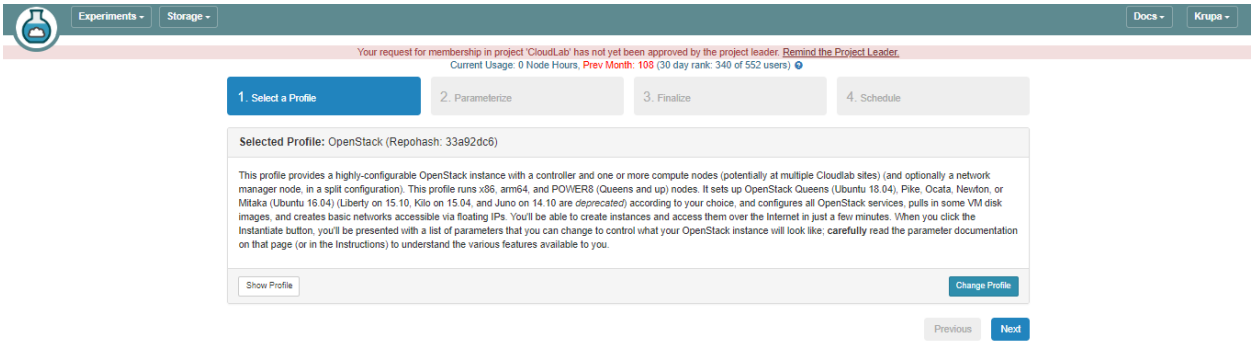

Fig 2 – Changing a profile.

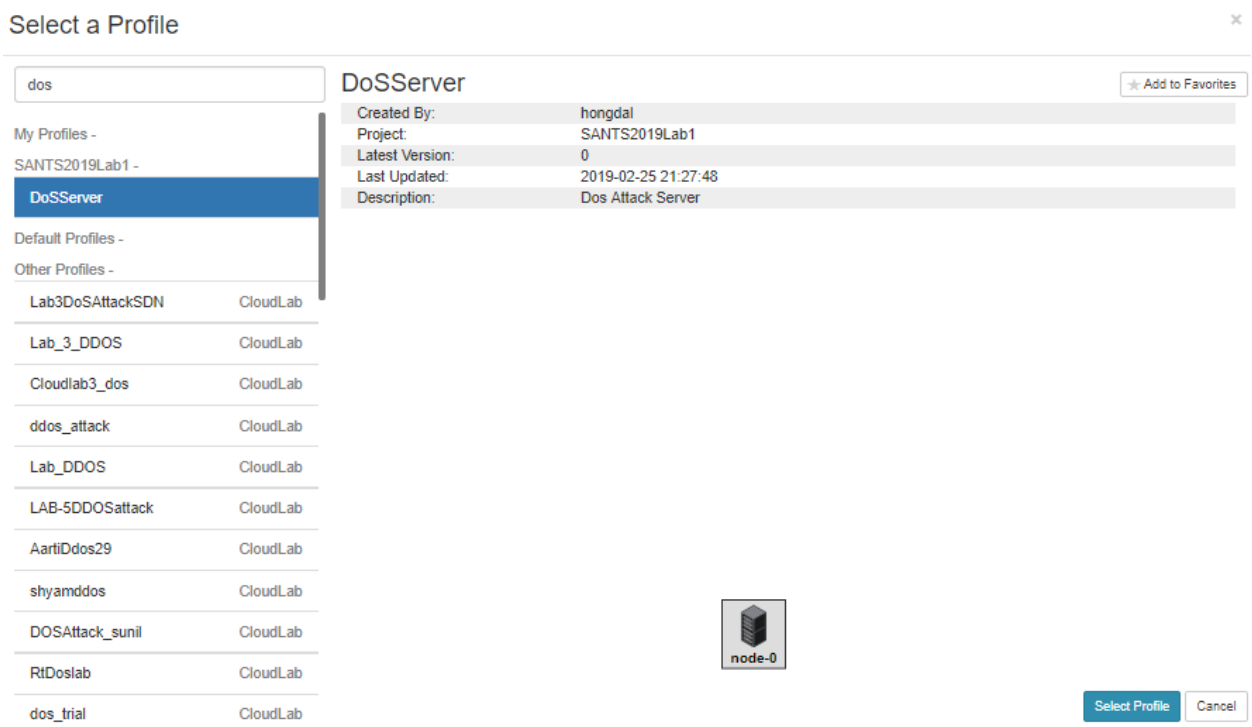

Fig 3 – Selecting a DOS Server.

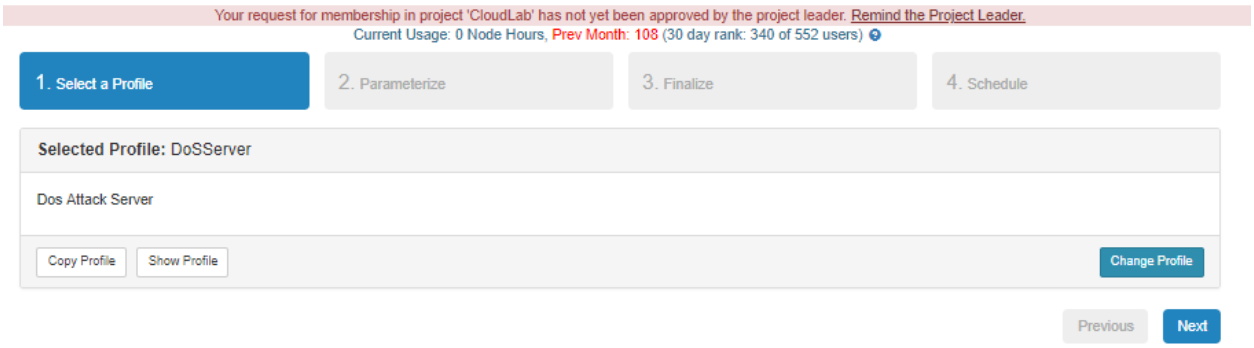

Fig 4 – Selecting and changing a profile.

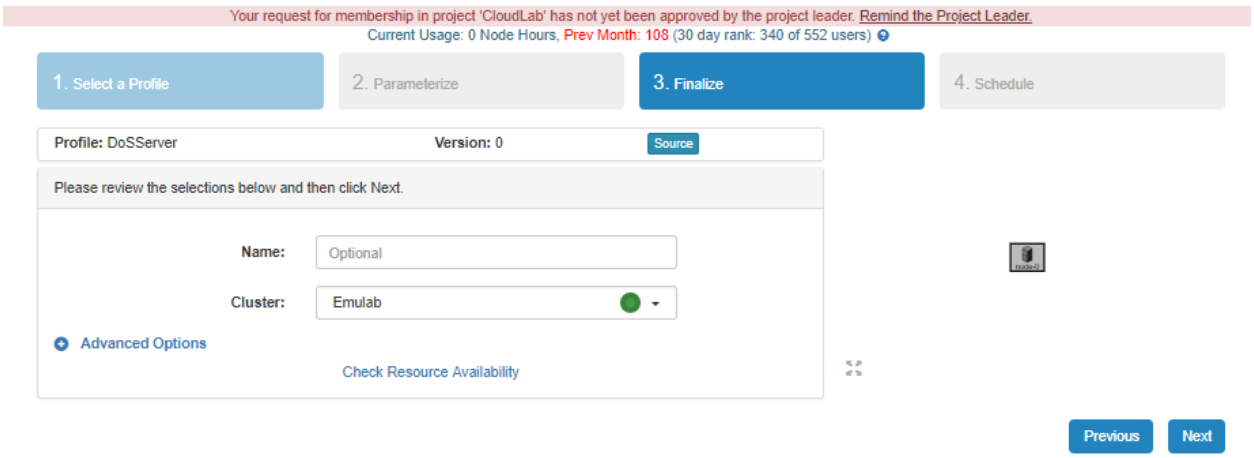

Fig 5 – Selecting a Cluster (the one chosen here is Emulab).

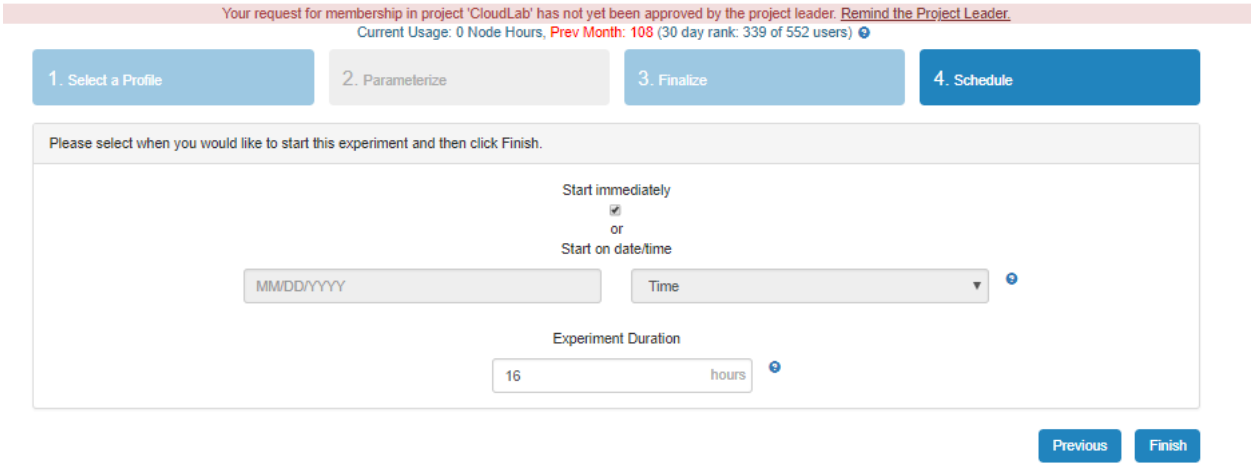

Fig 6 – Final Step – to select the duration of the experiment.

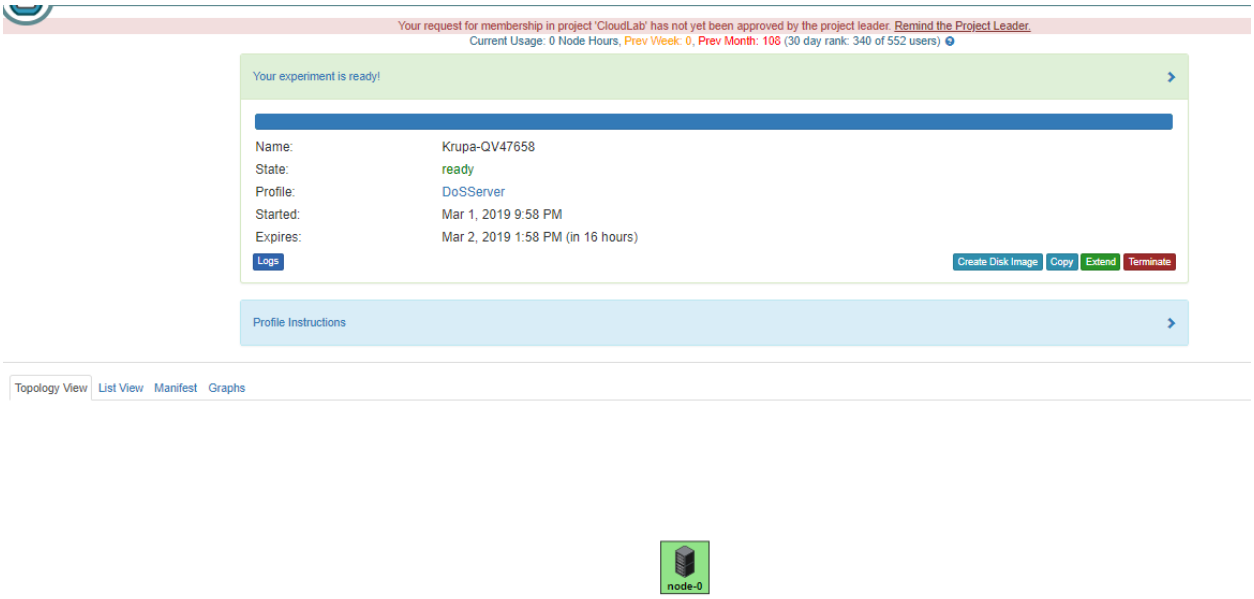

Fig 7 – The console when instantiated.

Step2: Conducting the Experiment.

Section 2.1 : Installing Dependencies

The figures from Fig 8 to Fig 37 demonstrate the installation of dependencies for carrying out the experiment. Further, there are three dependencies created. They are:

1. Floodlight – Fig 8 to Fig 27
2. Mininet – Fig 28 to Fig 35
3. Hping - Fig 36 and Fig 37

Installing Floodlight

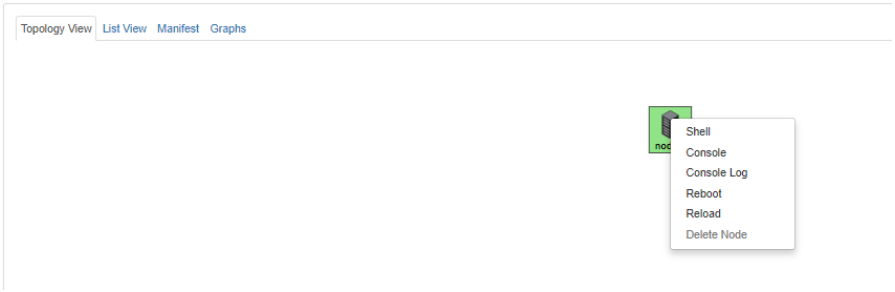

Fig 8 – Opening a Shell.

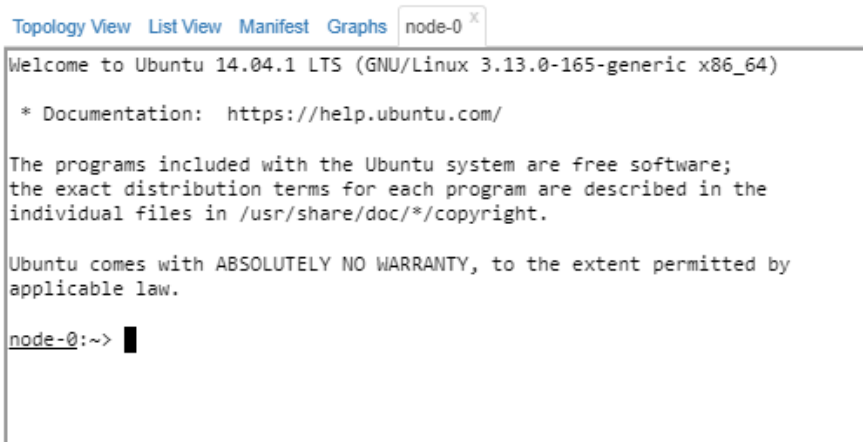

Fig 9 - Shell

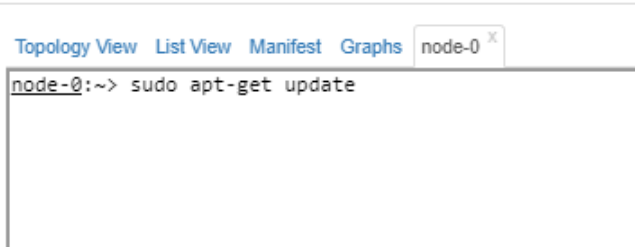

Fig 10 – Update apt using "apt-get update".

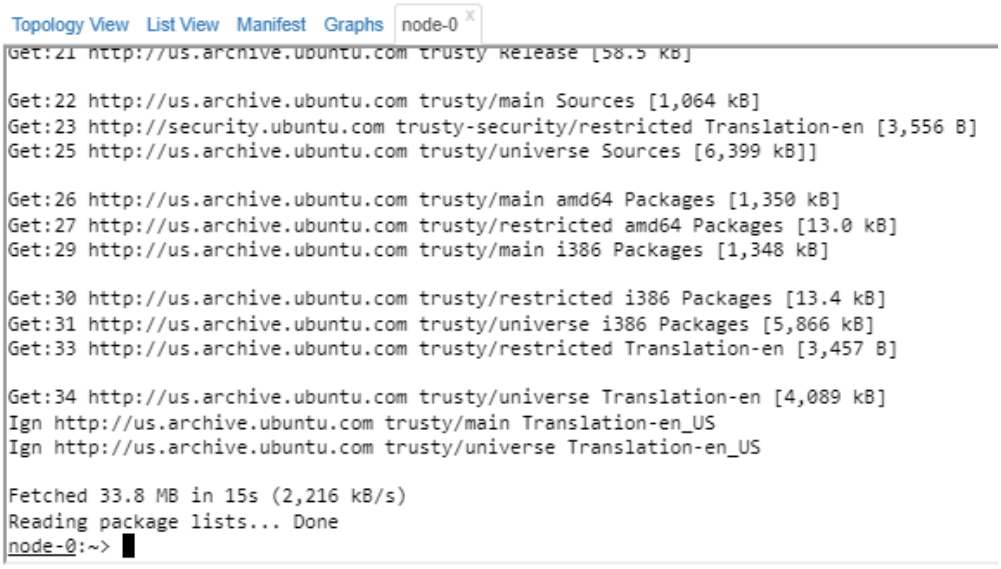

Fig 11 – Apt updating done.

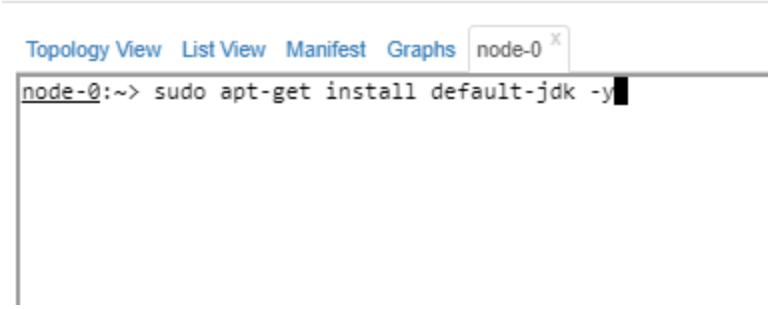

Fig 12 – install jdk using "sudo apt-get install default-jdk -y"

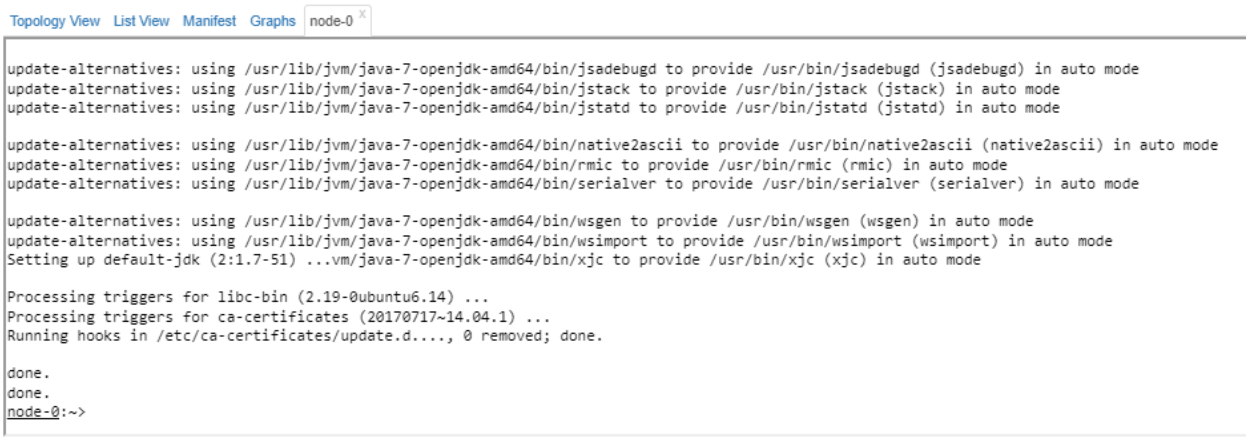

Fig 13 – jdk is installed.

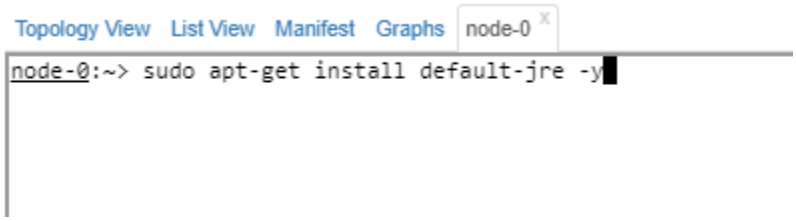

Fig 14 – Install jre using "sudo apt-get install default-jre -y"

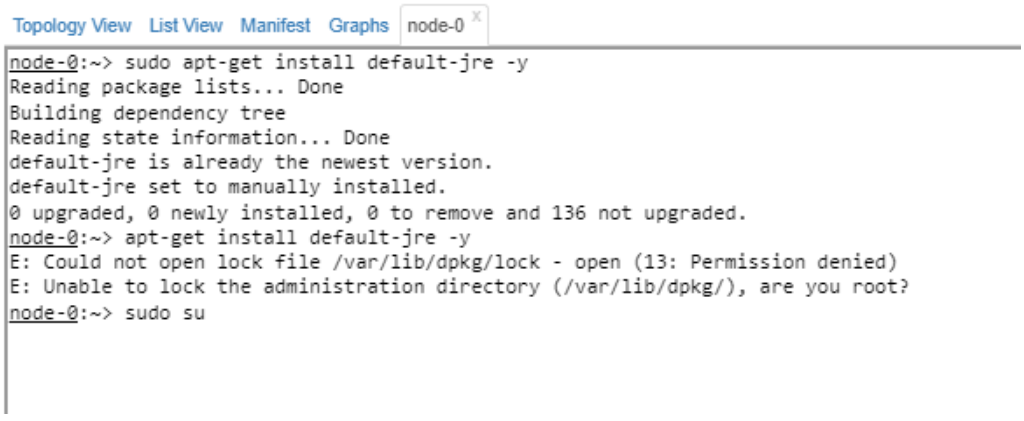

Fig 15 – get SU.

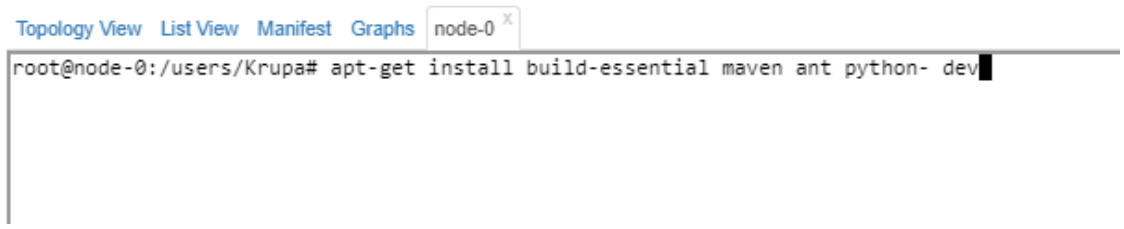

Fig 16 – build essential using "apt-get install build-essential maven ant python- dev"

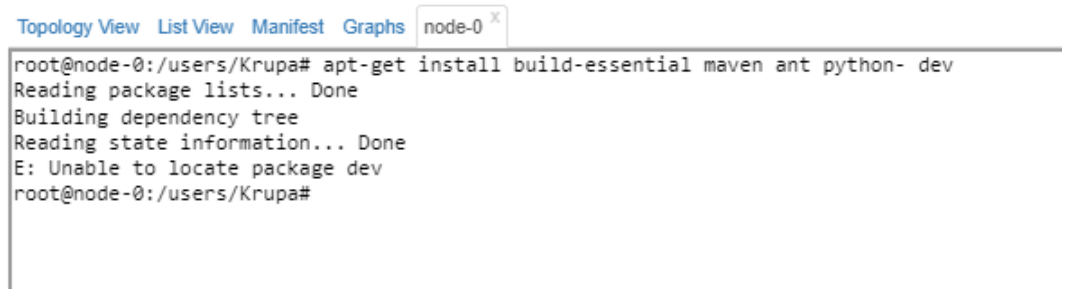

Fig 17 – Error occurred in building.

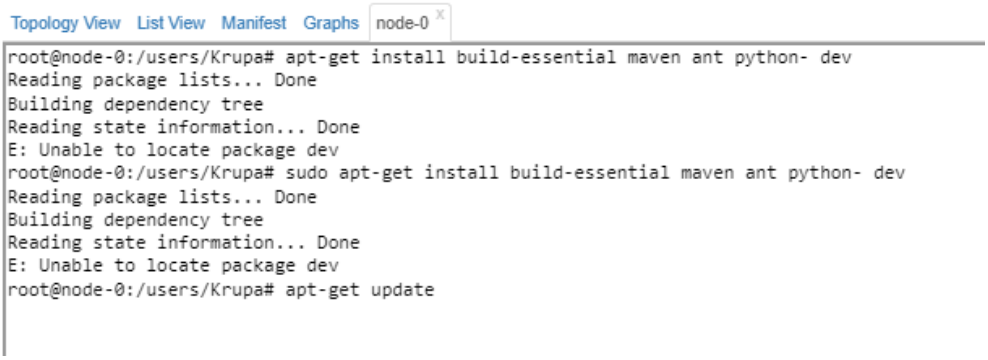

Fig 18 – To overcome error, update apt using "apt-get update".

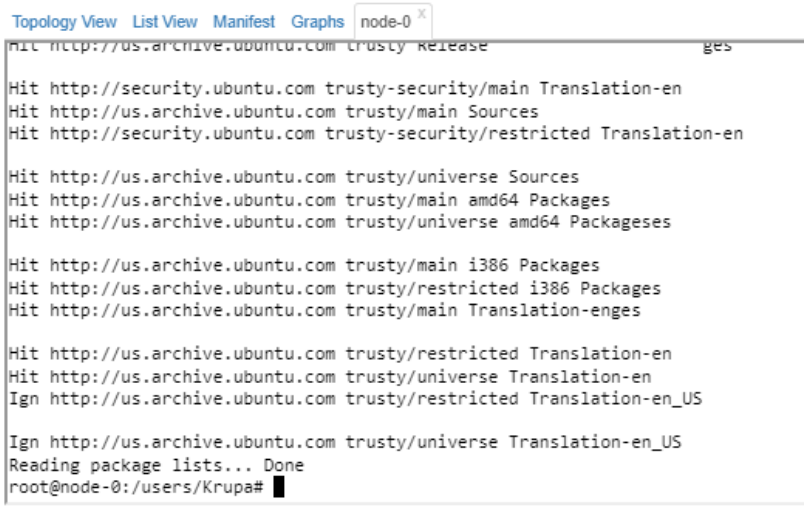

Fig 19 – apt update done.

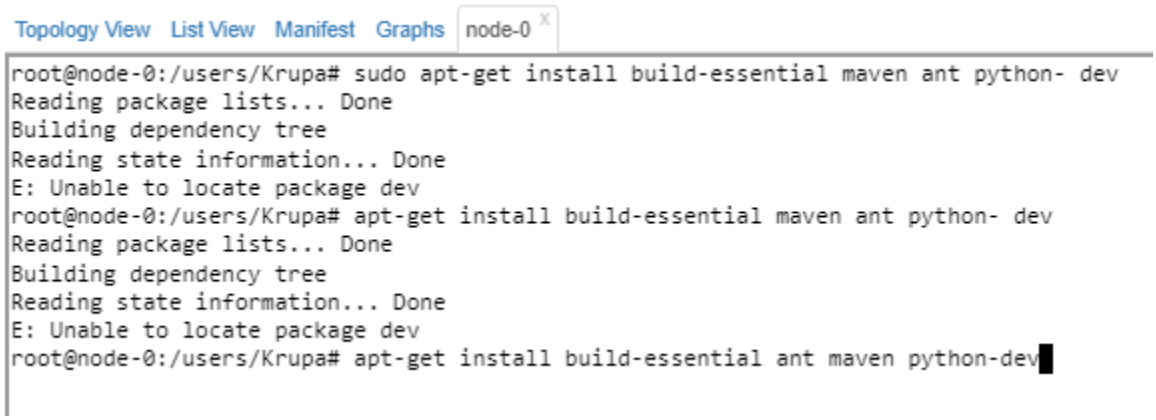

Fig 20 – Now build essential using same command.

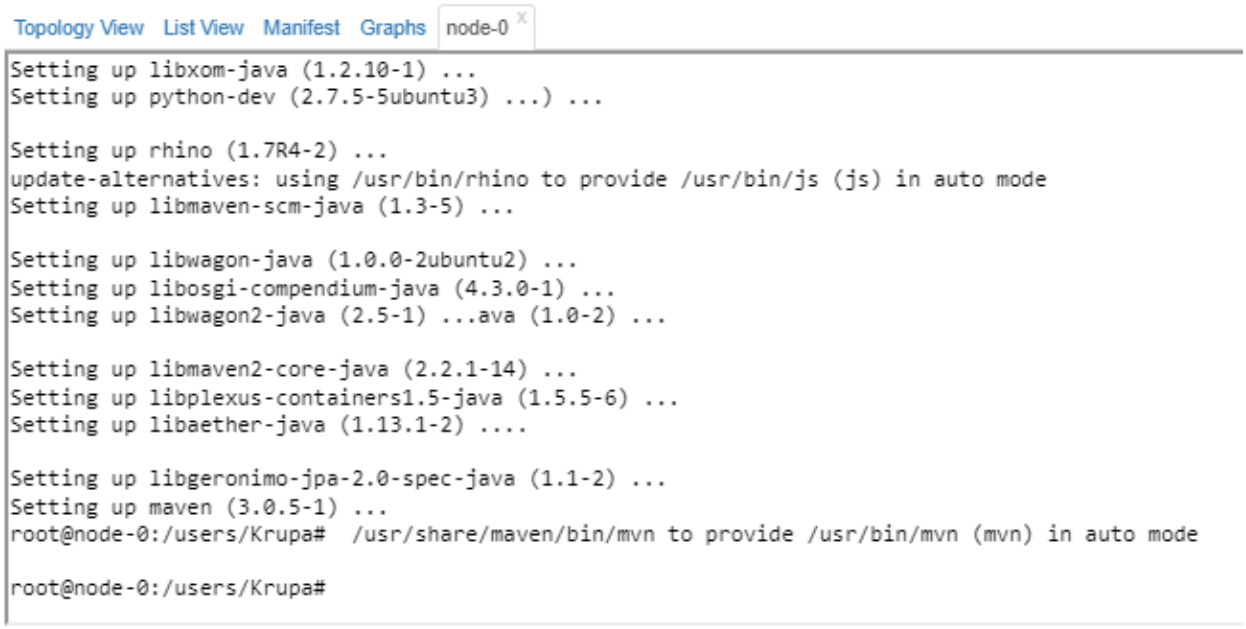

Fig 21 – Maven is set up.

Fig 22 – Clone GitHub for floodlight.

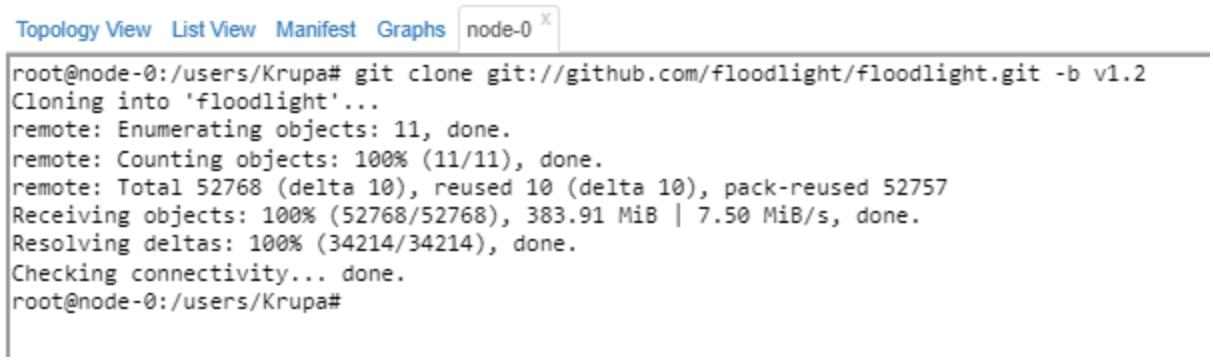

Fig 23 – Cloning done right.

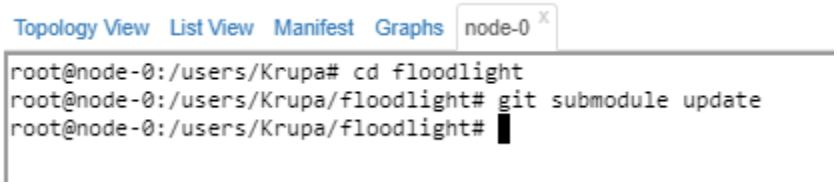

Fig 24 – getting into floodlight directory.

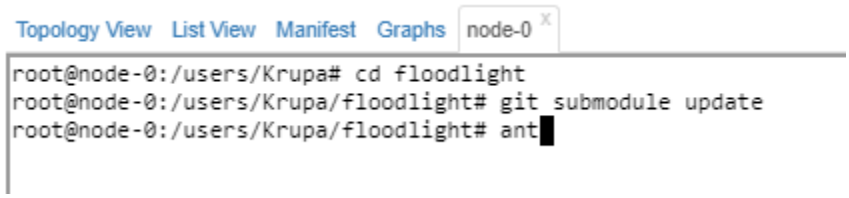

Fig 25 – install ant.

Fig 26 –build is successful.

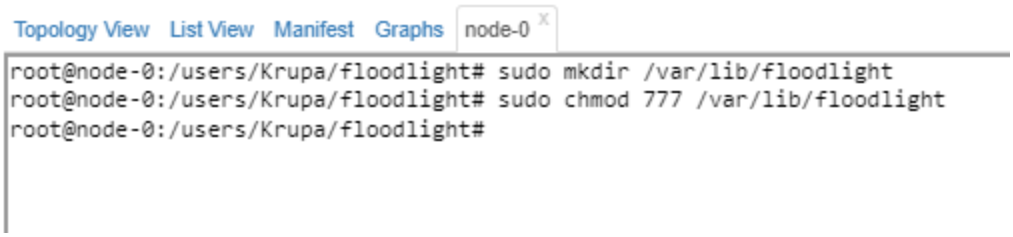

Fig 27 – Create a directory and change permissions.

Installing Mininet

Fig 28 – Clone Github for mininet.

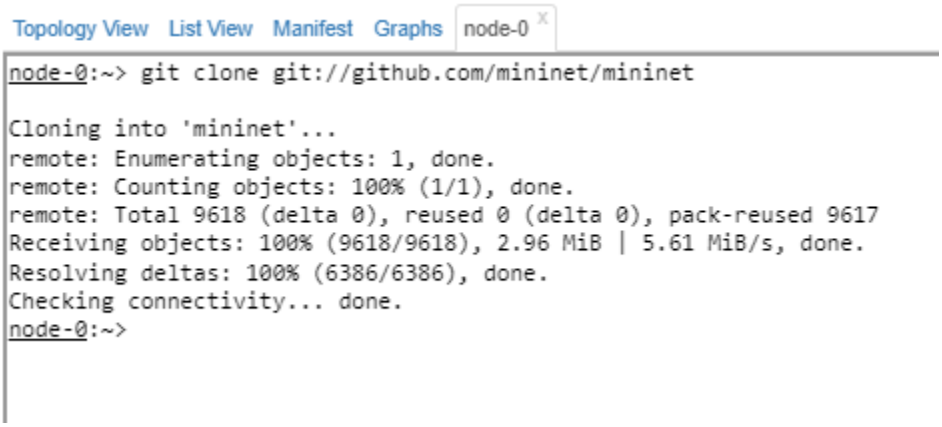

Fig 29 – Cloning done

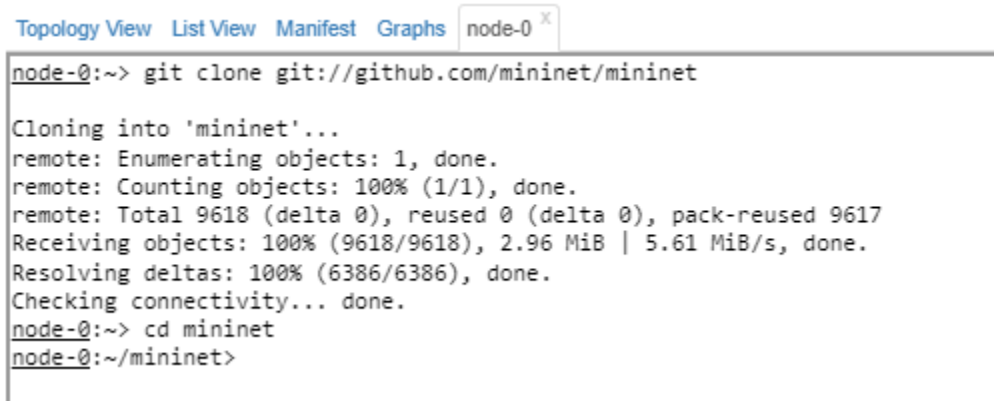

Fig 30 – Moving into mininet.

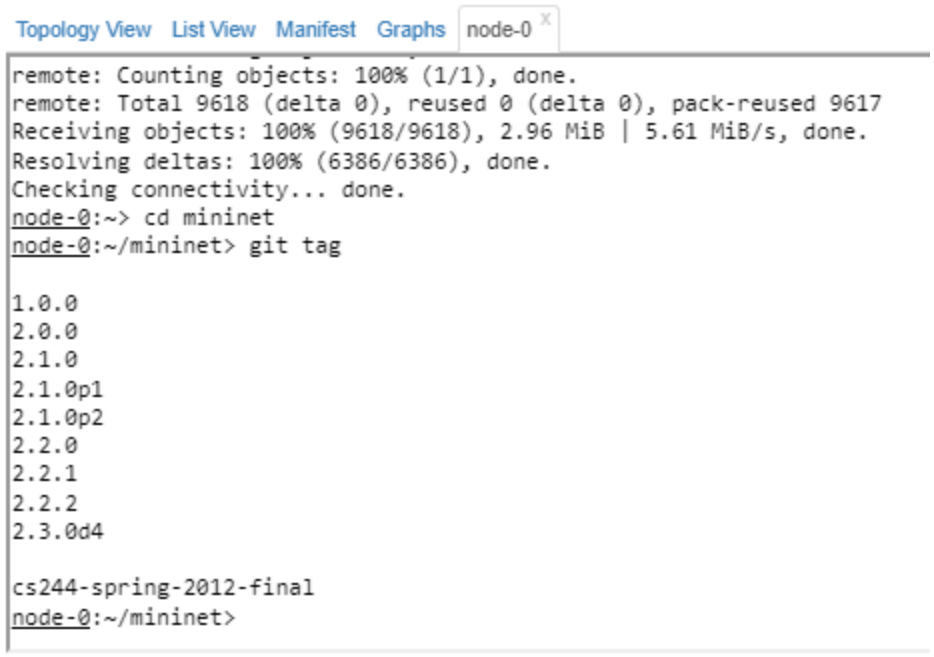

Fig 31 – To see available branches use “git tag”.

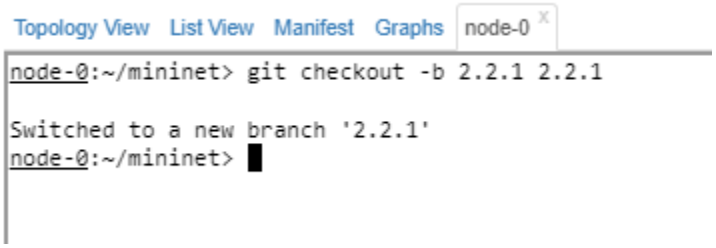

Fig 32 – Install mininet version 2.2.1 using “git checkout -b 2.2.1 2.2.1.

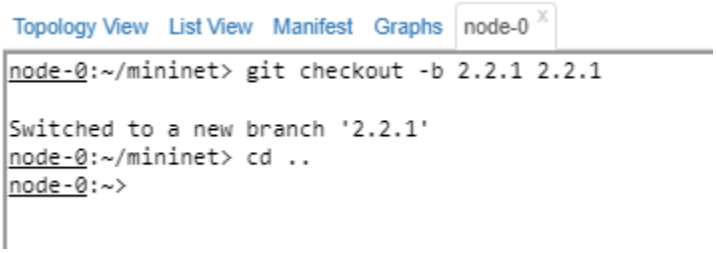

Fig 33 – coming out of mininet directory.

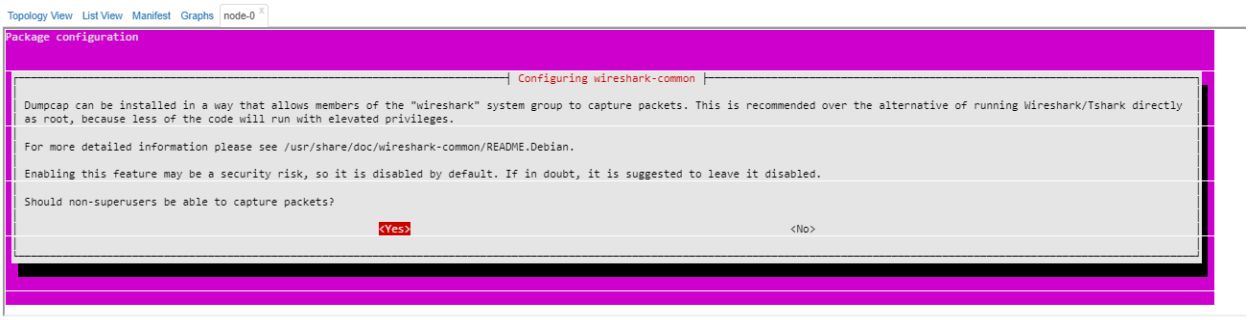

Fig 34 – Installing mininet using "mininet/util/install.sh -a" and Choosing "yes" to continue installation.

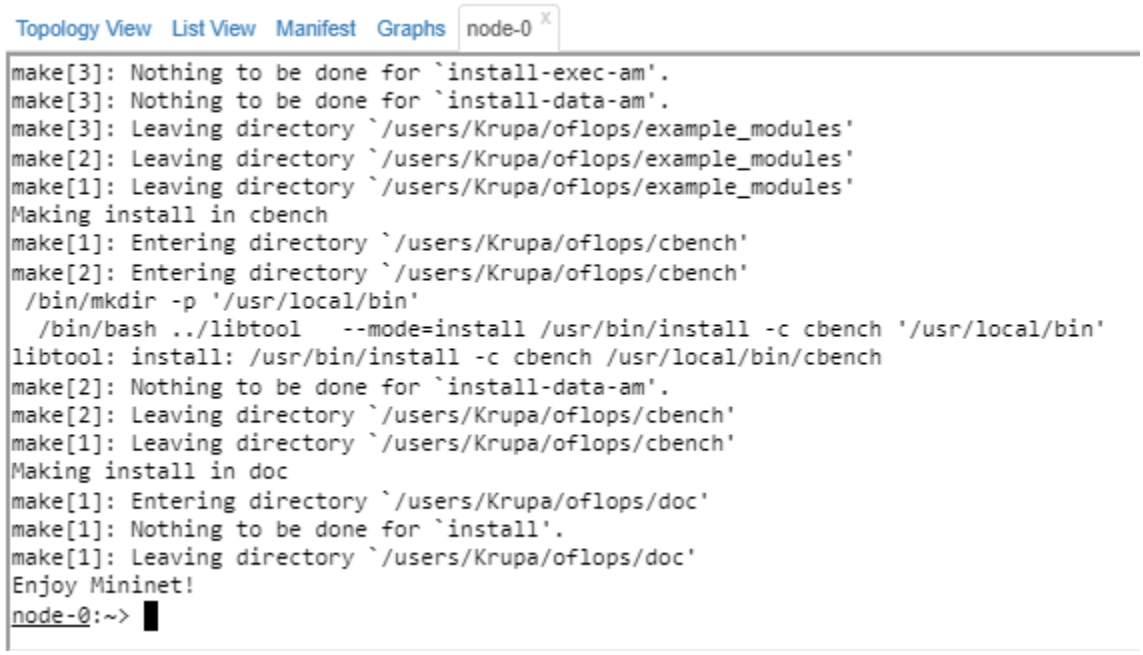

Fig 35 – Mininet installed.

Installing Hping3

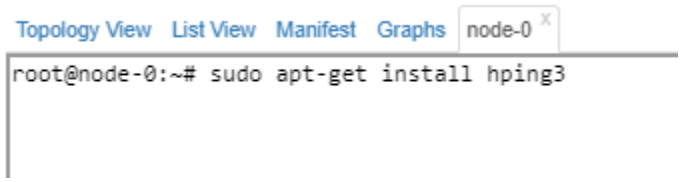

Fig 36 – to install Hping3 use "sudo apt-get install hping3".

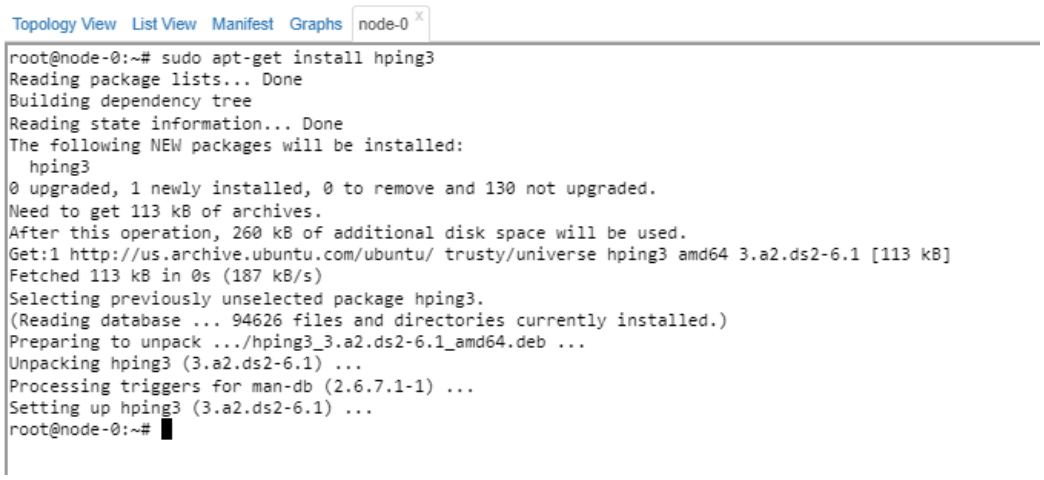

Fig 37 – Hping successfully installed.

Section 2.2 Running floodlight

The figures below show running of floodlight to how to start a flood f packets to demonstrate DoS attack.

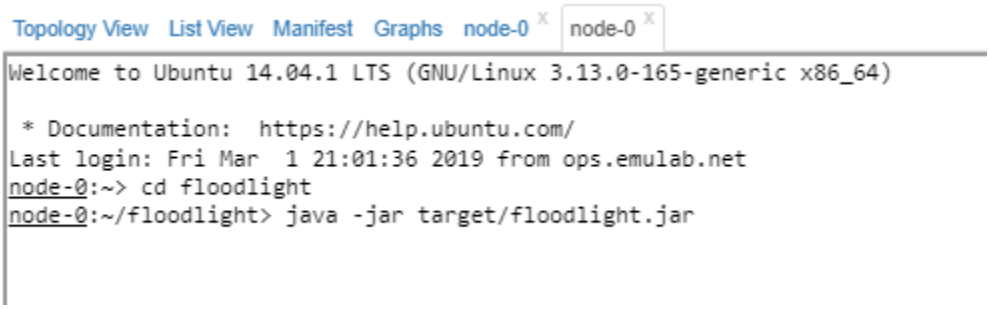

Fig 38 – After changing the directory to floodlight in a new shell, use "java -jar target/floodlight.jar" to run Floodlight Controller.

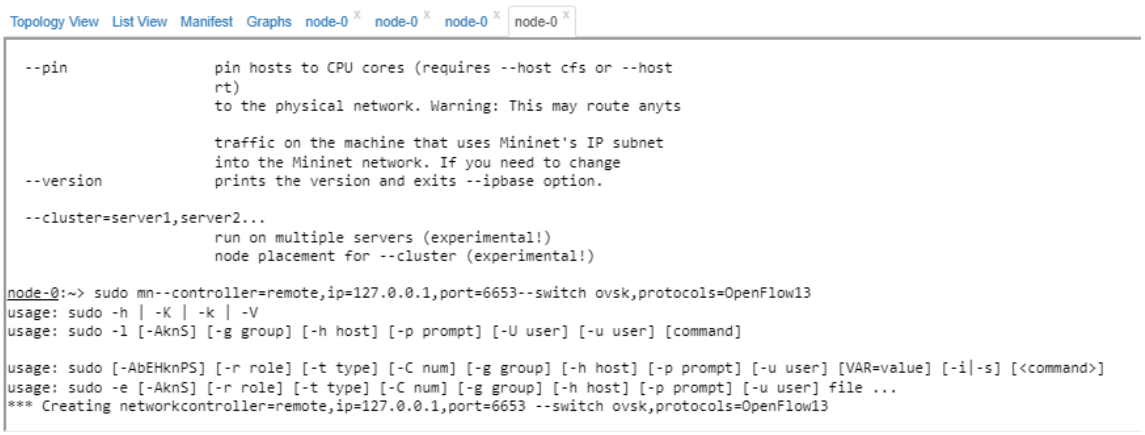

Fig 39 – try running mininet using "sudo mn --controller=remote,ip=127.0.0.1,port=6653 --switch ovsk, protocols=OpenFlow13"

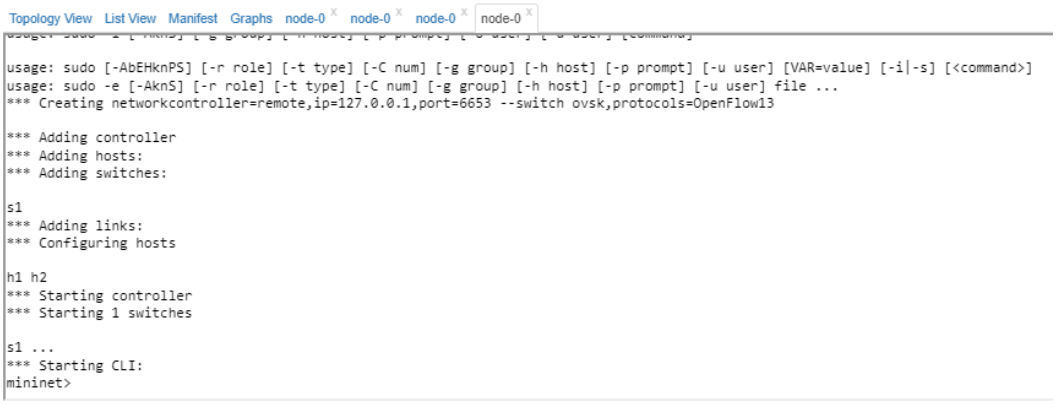

Fig 40 – The result: 2 hosts are created by default and are connected using OVS Bridges.

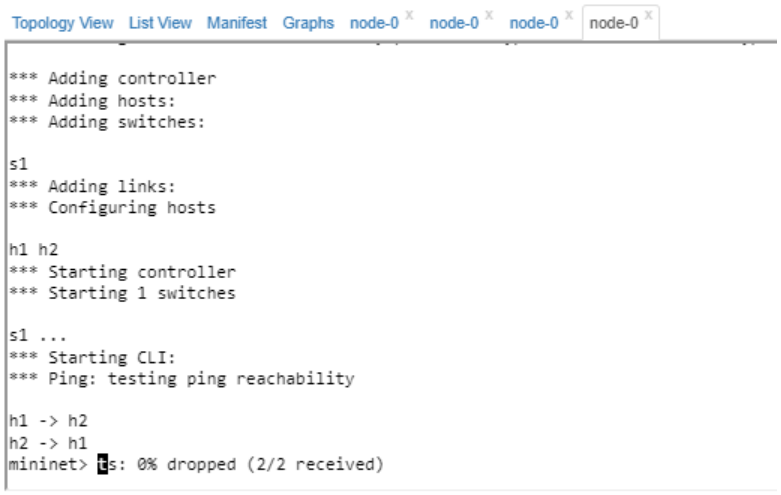

Fig 42 – pingall test to see connectivity.

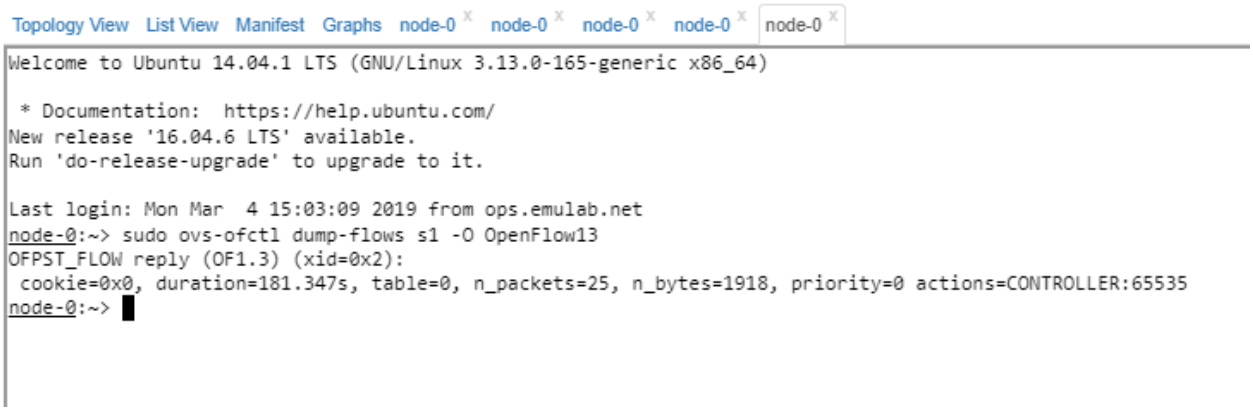

Fig 43 – To know the current flow rules , use "sudo ovs-ofctl dump-flows s1 -O OpenFlow13".

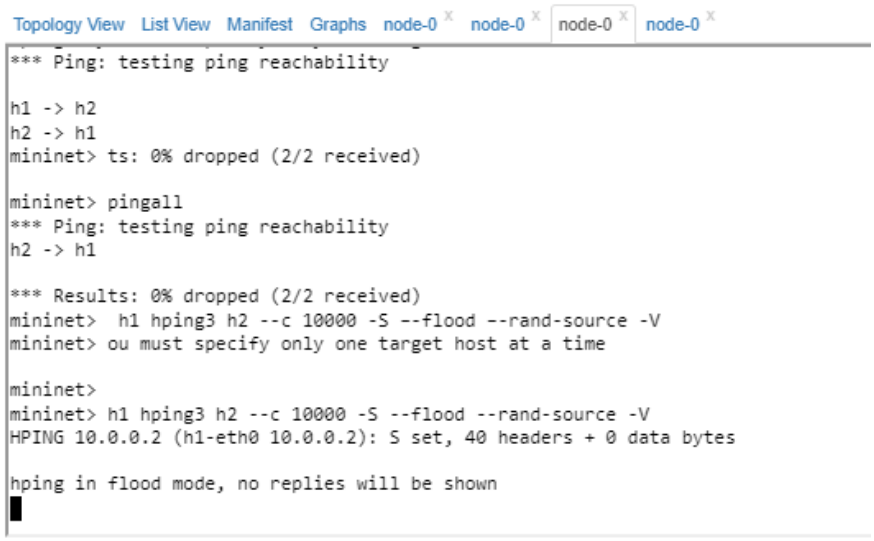

Fig 42 – To flood packets, use "h1 hping3 h2 -c 10000 -S –flood –rand-source -V" which floods packets from from h1 to h3.

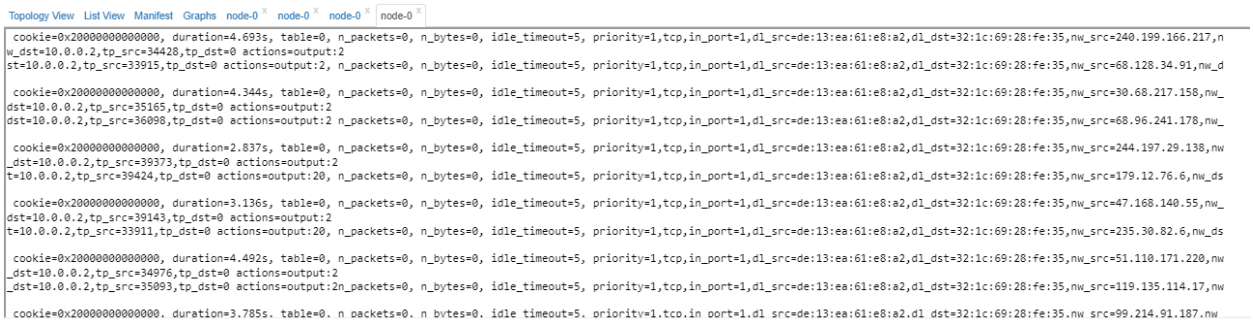

Fig 43 – Open a new controller to see flow entries for switch s1.

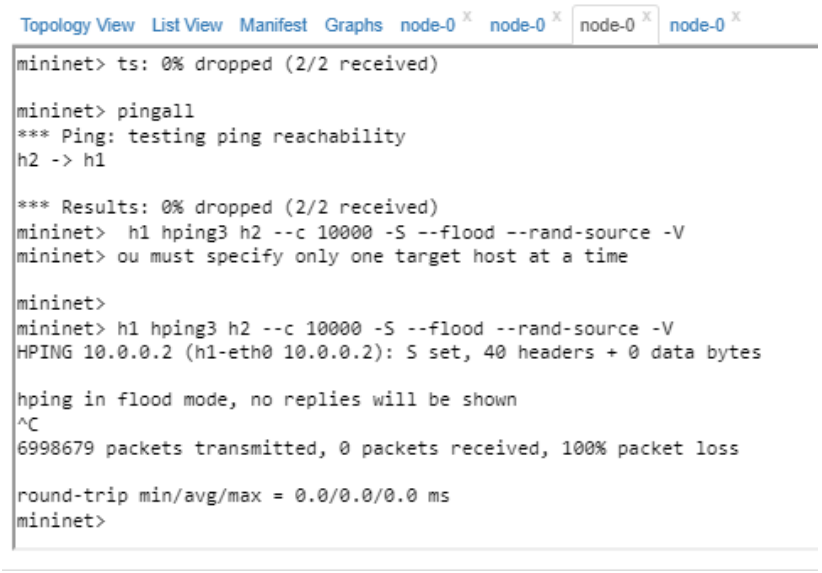

Fig 44 – Stop mininet in its terminal by using ctrl+c and ping h2 from h1.

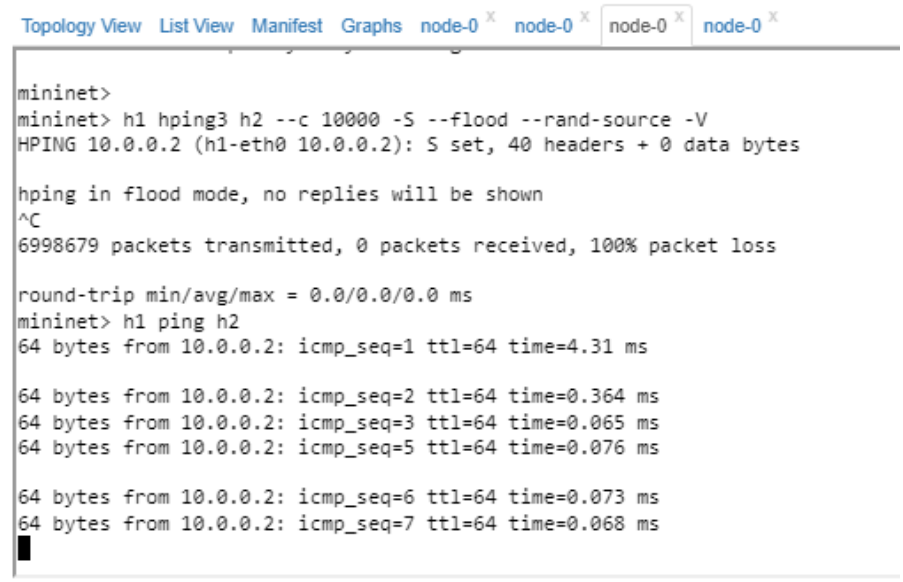

Fig 45 -pinging again (repetitions of previous step)

Fig 46 – Checking flow rules in an empty terminal

Observations and Conclusions:

We come to know how a DoS attack works. A DoS attack doesn't necessarily sniffs out data, but makes the system unreliable by bombarding it with a flood of packets thus making it overloaded and hence making no resources available for it. Since no resources are available, its services get disrupted.

This observation clearly demonstrates the DoS attack. We see that when flooding of packets is started , flow table of OVS switch is full and hence it cannot receive any more number of packets. This is because the space is insufficient and cannot store more packets. This leads to inefficiency as switches are refrained from obtaining data packets. Therefore there is an error here. The switch now send a OFPT_ERROR to the controller saying that the flow table is full by using OFPFMFC_TABLE_FULL. Since it now has no space to store packets, it is forced to drop the incoming packets thus losing data. Furthermore, it also has no resource to update a flowtable entry. This is the demonstration of DoS attack.