

# **EECE.7290 Selected Topics on Software Defined Networking**

## **Lab 2:** Programming OpenDayLight.

Name : Jigar Makwana

ID : 0171137

Email : [jigar\\_makwana@student.uml.edu](mailto:jigar_makwana@student.uml.edu)

## Purpose:

To learn the development of controller applications on OpenDayLight.

## Procedure:

- Download and run the Virtual Machine and add SDN\_tutorial.vm to Virtual Machine. I have used VM Ware Workstation player.
- Open SDN Tutorial OS in VM and update the tutorial code. By opening terminal in SDNHub\_Openaylight\_Tutorial directory. And use “Pull” command to get updated version of code from github repository.

**\$ git pull --rebase**

Also run

**\$ wget -q -O - https://raw.githubusercontent.com/opendaylight/odlparent/master/settings.xml**  
**> ~/.m2/settings.xml**

- Now that you update code you can use “**mvn**” command to build the tutorial code. It compiles code based on the pom.xml file in that directory. “install” is essential for compilation. It also accepts an optional argument “clean” if you wish to clean the temporary build files.
  - Go to the same directory and run the following command in terminal in that directory  
**\$ sudo mvn install -nsu**
  - It will take so much time depending your computer and it will show “**BUILD SUCCESS**”. If it gets build error and build is failed that means some files are corrupted get files from github again update and build mvn again.

- Now to run our controller go to the following directory.

**\$ SDNHub\_Openaylight\_Tutorial/distribution/opendaylight-karaf/target/assembly/bin**

- Now run karaf by  
**./karaf**
  - Running karaf starts all the Java bundles installed as jar files in the OSGi environment.
  - Karaf is main portal for managing all applications and the Java bundles.
  - “feature:list” and “bundle:list -s” are commands used to find active features and bundles in the Karaf runtime environment.(<feature:list -j> for only installed features, <bundle:list -s> to see loaded bundle,)
  - To install new feature in karaf
    - ◆ **Feature:install <feature name>**
      - You need to know the feature name which can be find using “feature:list”.
      - Once you install a feature it always stays in controller. So to remove the features go to the “**SDNHub\_Openaylight\_Tutorial/distribution**” directory and delete data,snapshots and journal folder and start the controller again.

- ◆ Now you need to install some features.
  - “**odi-dlux-all**”. This command will install all the dlux features that are required for GUI like **core** for topology application, **node** for node inventory applications, **YangUI** for browsing Yang RESTCONF interface, Yangvisualizer to browsing Yand models.
  - GUI can be open at <http://localhost:8181/>
  - User ID/Password : admin/admin
  - You can see all the features install and any topology running and Yang model and so many other things.

- Now keep controller running and start another terminal and start mininet
  - `$ sudo mn --topo single,3 --mac --switch ovsk,protocols=OpenFlow13 --controller remote`
- Now try to ping in mininet

The screenshot shows a VMware Workstation 12 Player window titled "SDN\_tutorial - VMware Workstation 12 Player (Non-commercial use only)". Inside the player, there are two terminal windows.

The left terminal window, titled "Terminal - ubuntu@sdnhubvm: ~", shows the following output:

```
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h1 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
From 10.0.0.1 icmp_seq=4 Destination Host Unreachable
From 10.0.0.1 icmp_seq=5 Destination Host Unreachable
From 10.0.0.1 icmp_seq=6 Destination Host Unreachable
^C
--- 10.0.0.3 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7034ms
pipe 3
mininet>
```

The right terminal window, titled "Terminal - ubuntu@sdnhubvm: ~/SDNHub\_OpenDaylight\_Tutorial/dist", shows the following output:

```
File Edit View Terminal Tabs Help
ubuntu@sdnhubvm:~$ cd distribution/opendaylight-karaf/target/assembly/karaf
bash: cd: distribution/opendaylight-karaf/target/assembly/karaf: No such file or directory
ubuntu@sdnhubvm:~$ cd SDNHub_OpenDaylight_Tutorial/distribution/opendaylight-karaf/target/assembly/karaf
bash: cd: SDNHub_OpenDaylight_Tutorial/distribution/opendaylight-karaf/target/assembly/karaf: No such file or directory
ubuntu@sdnhubvm:~$ cd SDNHub_OpenDaylight_Tutorial/distribution/opendaylight-karaf/target/assembly
ubuntu@sdnhubvm:~/SDNHub_OpenDaylight_Tutorial/distribution/opendaylight-karaf/target/assembly$ ls
in configuration data deploy etc lib LICENSE system version.properties
ubuntu@sdnhubvm:~/SDNHub_OpenDaylight_Tutorial/distribution/opendaylight-karaf/target/assembly$ cd bin
ubuntu@sdnhubvm:~/SDNHub_OpenDaylight_Tutorial/distribution/opendaylight-karaf/target/assembly/bin$ ./karaf
karaf: Enabling Java debug options: -Xdebug -Xnoagent -Djava.compiler=NONE -Xrun
dmp:transport=dt_socket,server=y,suspend=n,address=5005
java HotSpot(TM) Server VM warning: ignoring option MaxPermSize=512m; support was
removed in 8.0
listening for transport dt_socket at address: 5005

SDNHub

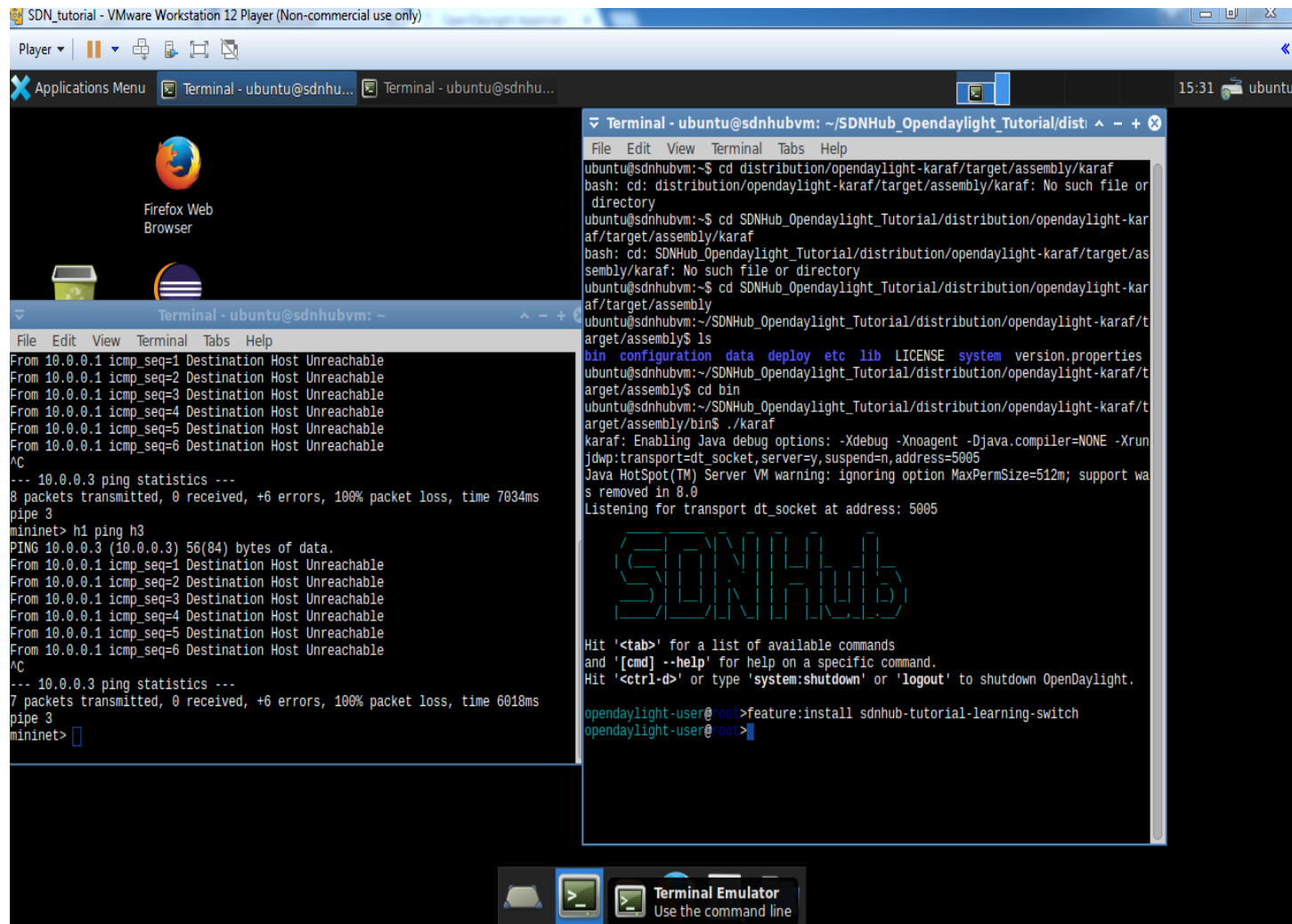
it '<tab>' for a list of available commands
and '<cmd>' --help' for help on a specific command.
it '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown OpenDaylight.

opendaylight-user@root>
```

Since there is no controller ping won't reach other hosts.

- Now install learning\_switch using following command in karaf.

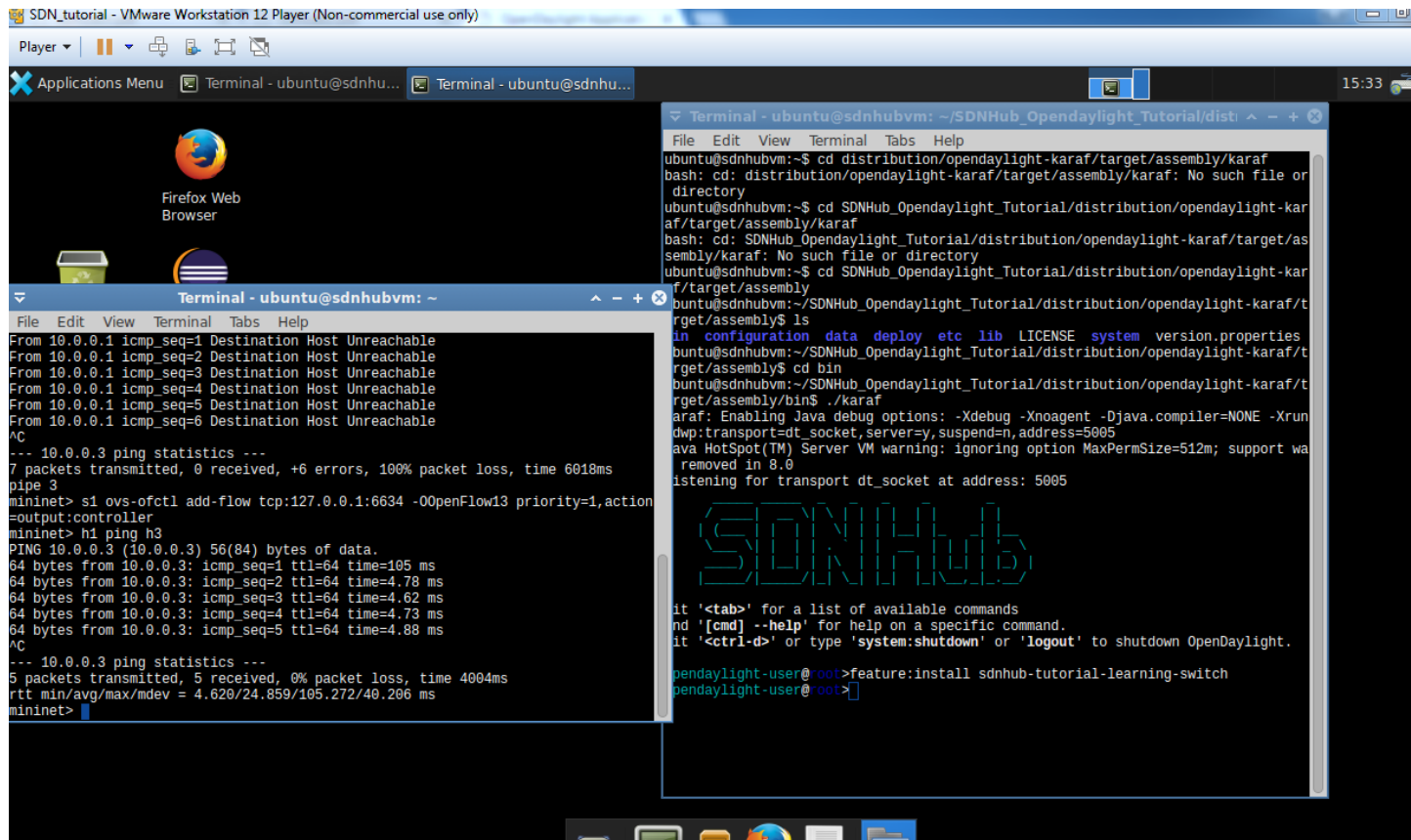
**Root> feature:install sdnhub-tutorial-learning-switch**



- Now try to ping again. It won't work.  
So, use following command in mininet.

**mininet>s1 ovs-ofctl add-flow tcp:127.0.0.1:6634 -OOpenFlow13  
priority=1,action=output:controller**

This will add rule in the switch to send packet-in messages to the controller. And ping will go through.

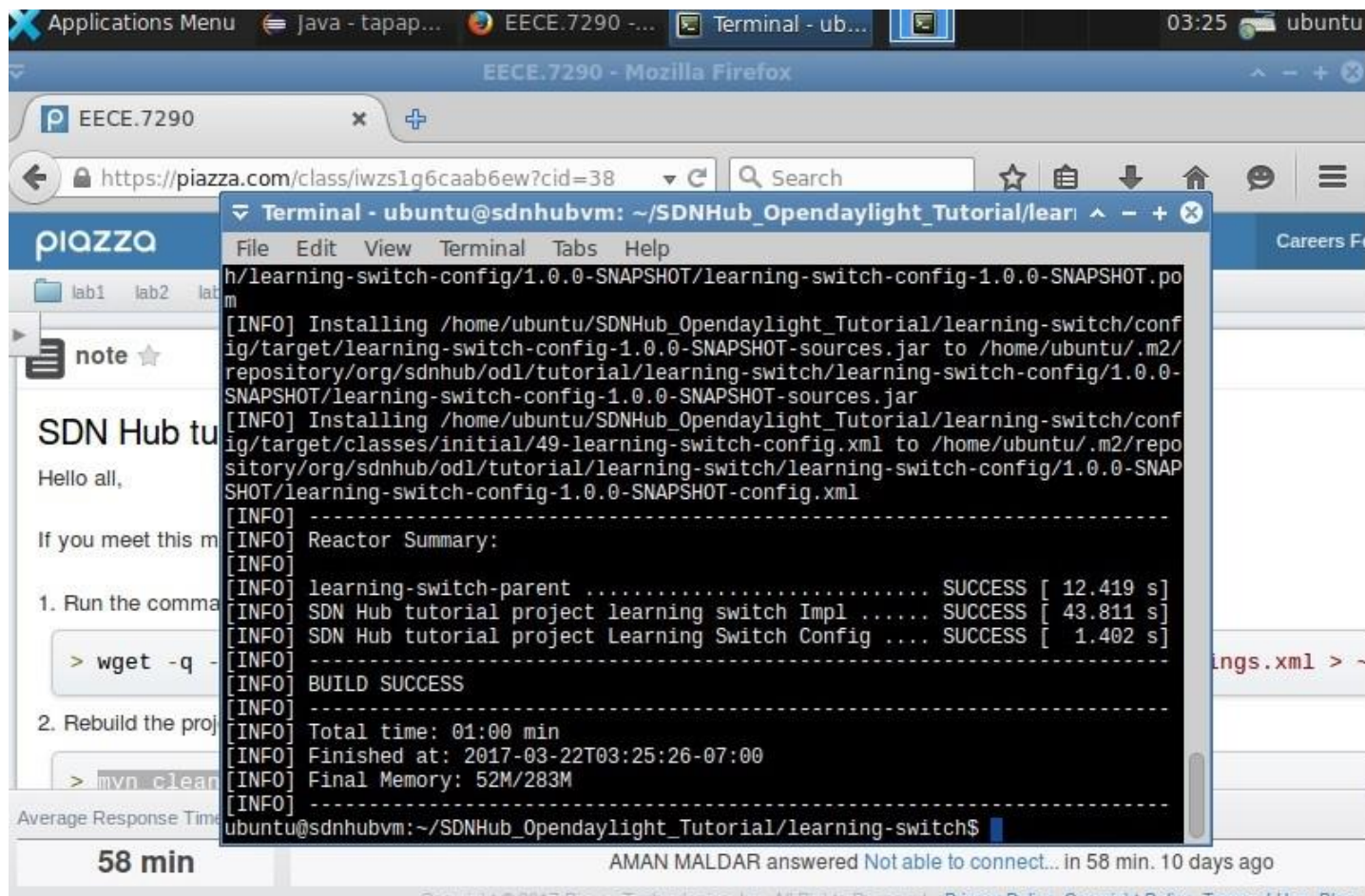


- Now to install the feature you modified in “**TutorialL2Forwarding.java**” for creating L2 learning switch.
  - To modify this file open eclipse that is already installed in SDNHub\_tutorial.vm.
  - In “file” menu chose “Import” then select “Existing Maven project” and “Next” then go to the directory “**SDNHub\_Openaylight\_Tutorial/learning-switch/implementation/src/main/java/org/sdnhub/odl/tutorial/learningswitch/impl/**” and enter.

Or you could go to that directory and open terminal and enter

**\$ nano TutorialL2Forwarding.java**

- Now build Maven.



- Start controller using following command. Go to the following directory.  
**\$ SDNHub\_Opendaylight\_Tutorial/distribution/.opendaylight-karaf/target/assembly/bin**
  - Now run karaf by  
**./karaf**



- Now install the feature we modified. And run the mininet in other terminal and start observing on wireshark for open flow messages.

