

SOFTWARE DEFINED NETWORKING

MSc IT- Part 2 Semester 4

Roll no. 03

Ajay Chaurasiya

E-Journal



Hindi Vidya Prachar Samiti's
**RAMNIRANJAN
JHUNJHUNWALA COLLEGE
(AUTONOMOUS)**

Opposite Ghatkopar Railway Station, Ghatkopar West, Mumbai-400086



CERTIFICATE

This is to certify that **Mr. Ajay Chaurasiya** with Seat No.**03** has successfully completed the necessary course of experiments in the subject of **Software Defined Networking** during the academic year **2020 – 2021** complying with the requirements of **RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ARTS, SCIENCE AND COMMERCE**, for the course of M.Sc. (IT) semester -IV.

Internal Examiner

Date: _____

Head of Department
Examiner

College Seal

External

INDEX

Sr	Practical	Date
1	Packet capturing with Wireshark and programming with bash Script on Ubuntu	27/4/2021
2	Theory: HP controller, HP Network Protector, HP Network Visualizer, HP Network Optimizer, Cisco XNC Controller	30/4/2021
3	Install ODL (Open Daylight) controller	13/3/2021
4	Implement OVS (Open vSwitch) on ODL	17/3/2021
5	Implement mininet on ODL	12/4/2021
6	Install RYU controller with mininet topology	16/4/2021
7	Install FLOODLIGHT controller with mininet topology	19/4/2021
8	Install ONOS controller on Ubuntu	23/4/2021

PRACTICAL NO: 01

Wireshark and bash Script

Installing Wireshark on Ubuntu 16.04 using terminal

```
root@ubuntu-KVM: ~  
root@ubuntu-KVM:~# #sudo add-apt-repository ppa:wireshark-dev/stable  
root@ubuntu-KVM:~# #sudo apt update  
root@ubuntu-KVM:~# #sudo apt install wireshark -y  
root@ubuntu-KVM:~#
```

```
root@ubuntu-KVM: ~  
root@ubuntu-KVM:~# sudo add-apt-repository ppa:wireshark-dev/stable  
Latest stable Wireshark releases back-ported from Debian package versions.  
  
Back-porting script is available at https://github.com/rbalint/pkg-wireshark-ubuntu-ppa  
  
From Ubuntu 16.04 you also need to enable "universe" repository, see:  
http://askubuntu.com/questions/148638/how-do-i-enable-the-universe-repository  
  
The packaging repository for Debian and Ubuntu is at: https://salsa.debian.org/debian/wireshark  
More info: https://launchpad.net/~wireshark-dev/+archive/ubuntu/stable  
Press [ENTER] to continue or ctrl-c to cancel adding it  
  
gpg: keyring `/tmp/tmpfn7l8em_/secreing.gpg' created  
gpg: keyring `/tmp/tmpfn7l8em_/pubring.gpg' created  
gpg: requesting key 14ECA0F0 from hkp server keyserver.ubuntu.com  
gpg: /tmp/tmpfn7l8em_/trustdb.gpg: trustdb created  
gpg: key 14ECA0F0: public key "Launchpad PPA for Wireshark Developers" imported  
gpg: Total number processed: 1  
gpg: imported: 1 (RSA: 1)  
OK  
root@ubuntu-KVM:~# sudo apt update  
Hit:1 http://in.archive.ubuntu.com/ubuntu xenial InRelease
```

```

root@ubuntu-KVM: ~
root@ubuntu-KVM:~# sudo apt install wireshark -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libproxychains3 snapd-login-service
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libbrotli1 libc-ares2 libmaxminddb0 libminizip1 libnghttp2-14
  libnl-route-3-200 libqgsttools-p1 libqt5multimedia5-plugins
  libqt5multimediawidgets5 libsmi2ldbl libsnappy1v5 libspandsp2
  libssh-gcrypt-4 libwireshark-data libwireshark14 libwiretap11 libwsutil12
  wireshark-common wireshark-qt
Suggested packages:
  mmdns-bin snmp-mibs-downloader geopyupdate geopy-database-extra libjs-leaflet
  libjs-leaflet.markercluster wireshark-doc
The following NEW packages will be installed:
  libbrotli1 libc-ares2 libmaxminddb0 libminizip1 libnghttp2-14
  libnl-route-3-200 libqgsttools-p1 libqt5multimedia5-plugins
  libqt5multimediawidgets5 libsmi2ldbl libsnappy1v5 libspandsp2
  libssh-gcrypt-4 libwireshark-data libwireshark14 libwiretap11 libwsutil12
  wireshark wireshark-common wireshark-qt
0 upgraded, 20 newly installed, 0 to remove and 0 not upgraded.
Need to get 23.3 MB of archives.

```

Starting Wireshark

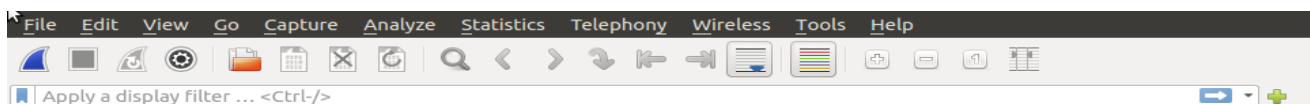
Launching a Wireshark application using CLI

Type wireshark in CLI and press enter

```

root@ubuntu-KVM: ~
root@ubuntu-KVM:~# wireshark

```



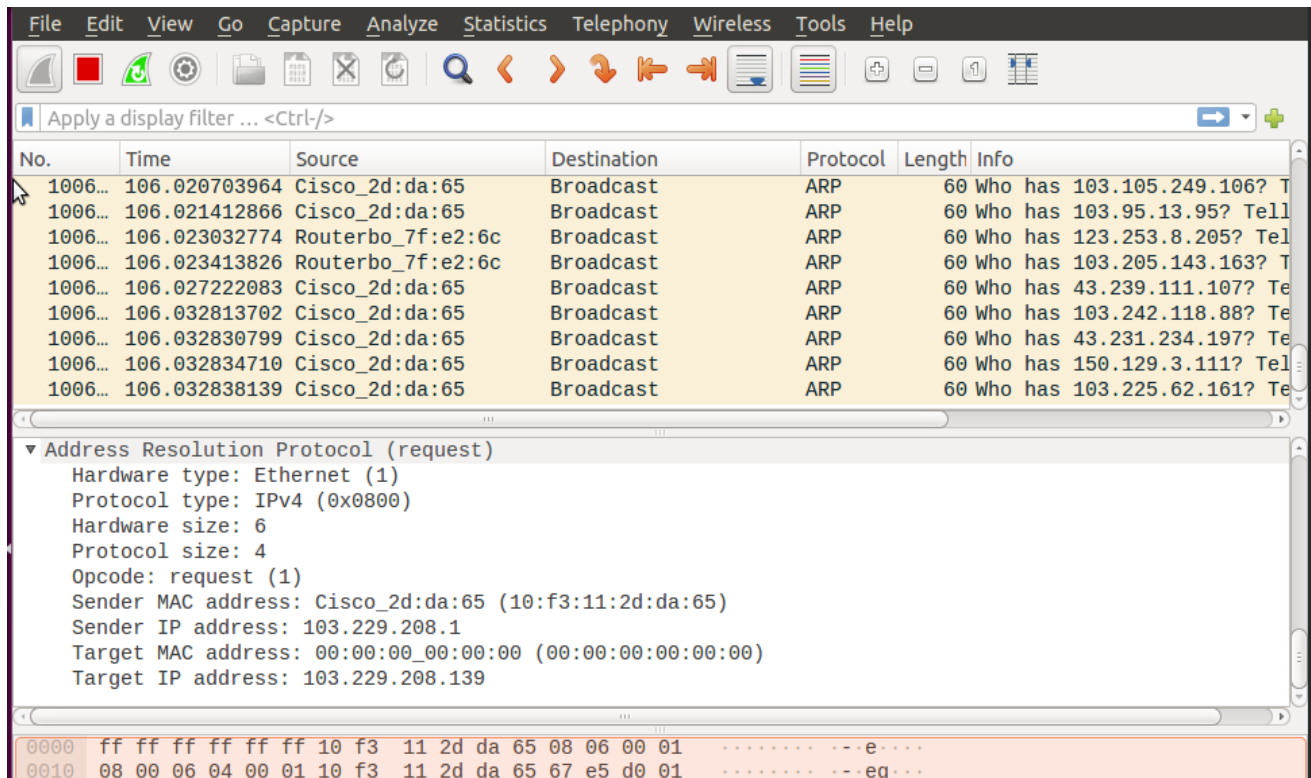
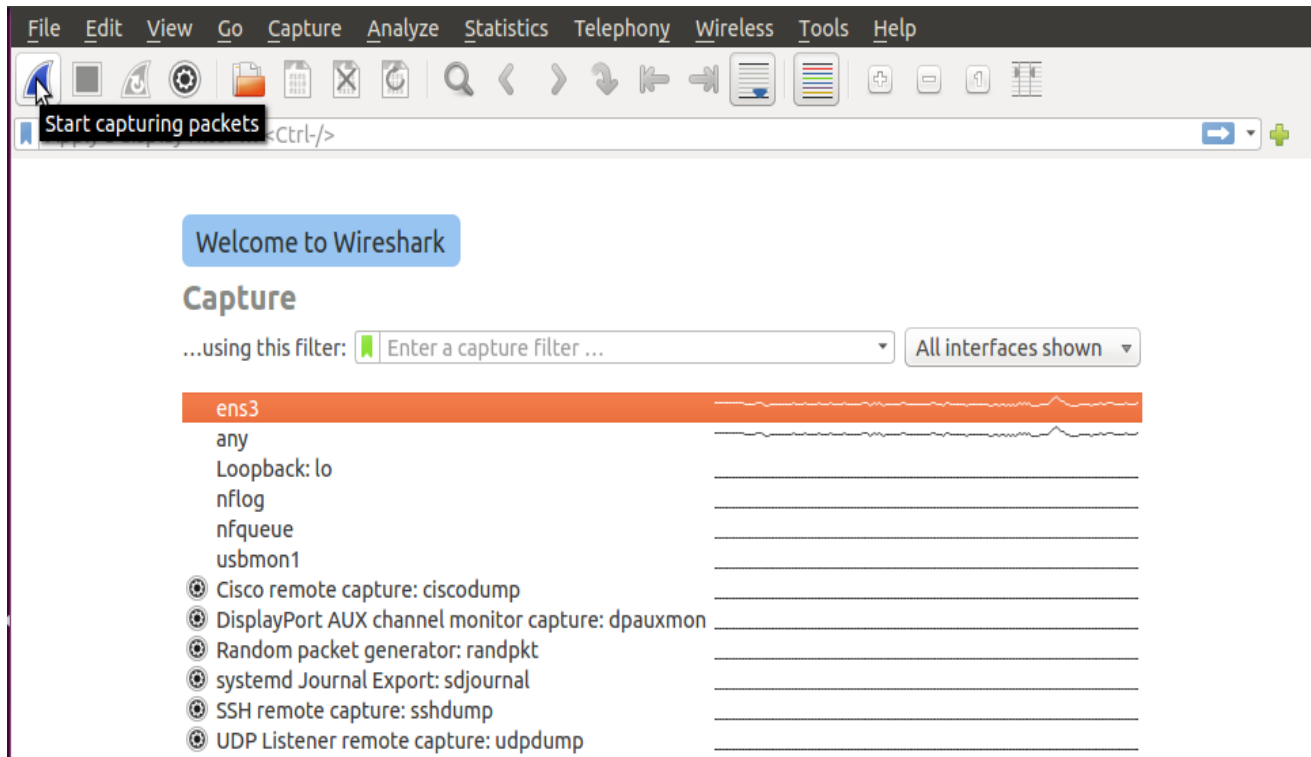
Welcome to Wireshark

Capture

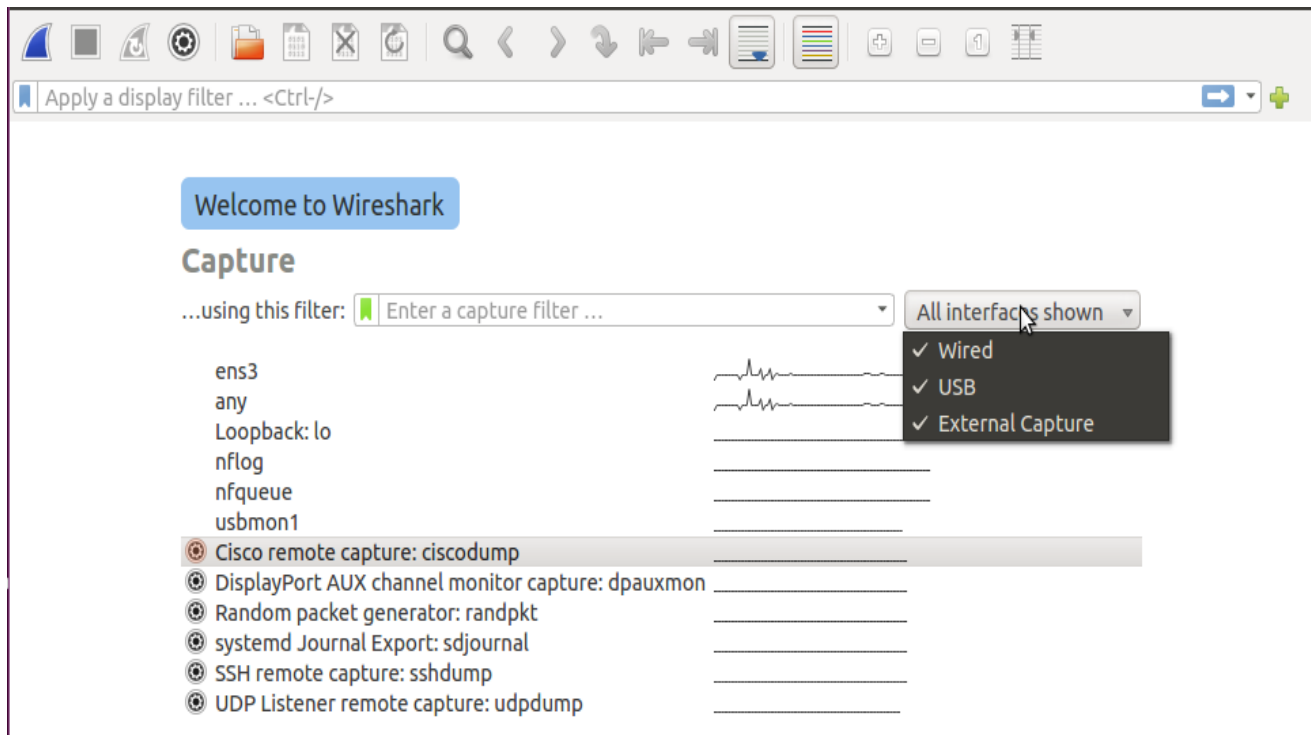
...using this filter: All interfaces shown

ens3	N
any	N
Loopback: lo	
nflag	No addresses
nqueue	No capture filter
usbmon1	
⊗ Cisco remote capture: ciscodump	
⊗ DisplayPort AUX channel monitor capture: dpauxmon	
⊗ Random packet generator: randpkt	
⊗ systemd Journal Export: sdjournal	
⊗ SSH remote capture: sshdump	
⊗ UDP Listener remote capture: udpdump	

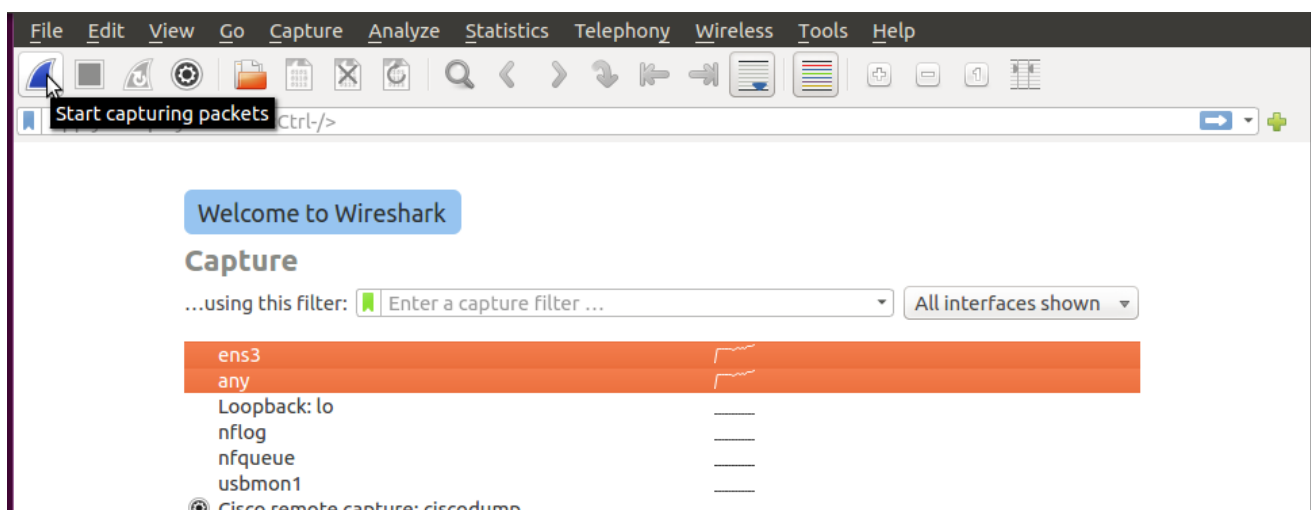
To test packets capturing, select interface(ens3) to use and click “**Start capturing packets**” button.



There are many types of interfaces you can monitor using Wireshark, for example, Wired, Wireless, USB and many external devices. You can choose to show specific types of interfaces in the welcome screen from the marked section of the screenshot below.



You can also capture packets to and from multiple interfaces at the same time. Just press and hold <Ctrl> and click on the interfaces that you want to capture packets to and from and then click on the **Start capturing packets** icon as marked in the screenshot below.



Now you can click on a packet to select it. Selecting a packet would show many information about that packet. As you can see, information about different layers of TCP/IP Protocol is listed.

The image shows the Wireshark network protocol analyzer interface. The top toolbar contains various icons for file operations, navigation, and analysis. Below the toolbar is a filter bar with the text "Apply a display filter ... <Ctrl-/>". The main packet list table displays several ARP requests from a Cisco_2d:da:65 source to a broadcast destination. Packet 2661 is selected, and its details are shown in the lower pane. The details pane for packet 2661 shows the following information:

- ▼ Frame 2661: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface ens3, id 0
- ▶ Interface id: 0 (ens3)
- Encapsulation type: Ethernet (1)
- Arrival Time: Apr 25, 2021 19:42:23.942057940 IST
- [Time shift for this packet: 0.000000000 seconds]
- Epoch Time: 1619359943.942057940 seconds
- [Time delta from previous captured frame: 0.000022009 seconds]
- [Time delta from previous displayed frame: 0.000022009 seconds]
- [Time since reference or first frame: 3.189140430 seconds]
- Frame Number: 2661
- Frame Length: 60 bytes (480 bits)

Below the details pane, the raw packet data is shown in hexadecimal and ASCII format. The hexadecimal data is displayed in a table with columns for offset, hex bytes, and ASCII characters. The ASCII column shows the characters "e", "g", and "g" corresponding to the hex bytes 0x0e, 0x01, and 0x00 respectively.

Offset	Hex	ASCII
0000	ff ff ff ff ff ff 10 f3 11 2d da 65 08 06 00 01-e....
0010	08 00 06 04 00 01 10 f3 11 2d da 65 67 5f 0e 01eg....
0020	00 00 00 00 00 00 67 5f 0e 9e 00 00 00 00 00 00g_....
0030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Bash Scripts

open text editor and save file with .sh; run in terminal

Bash script to print current date and time and Hostname .

```
#!/bin/bash
```

```
var="Hello World"
```

```
# Run date and hostname command and store output to shell variables
```

```
now="$(date)"
```

```
computer_name="$(hostname)"
```

```
# print it or use the variable
```

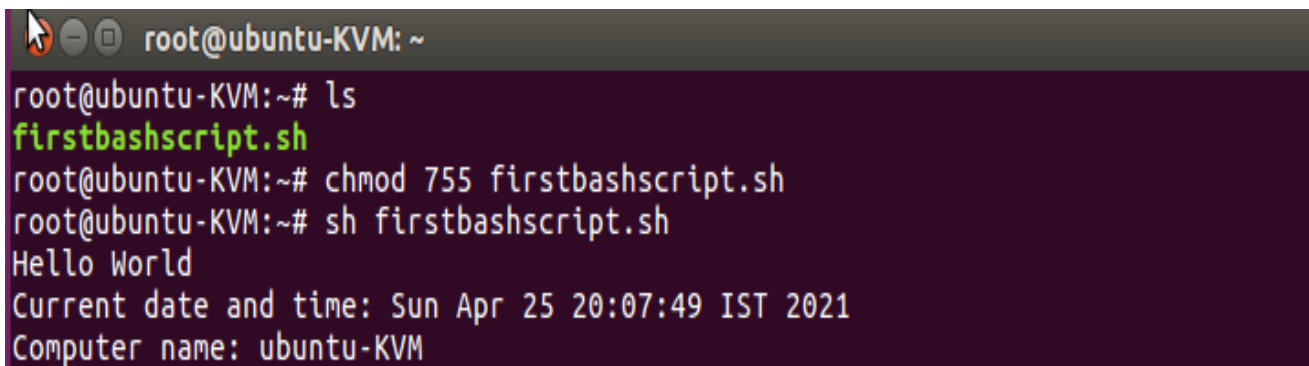
```
echo "$var"
```

```
echo "Current date and time: " $now
```

```
echo "Computer name: " $computer_name
```

After creating the script give execute permission to the file/script.

```
#chmod 755 firstbashscript.sh
```

A terminal window titled 'root@ubuntu-KVM: ~' with a dark background. It shows the execution of a script: 'ls' lists 'firstbashscript.sh', 'chmod 755 firstbashscript.sh' sets permissions, and 'sh firstbashscript.sh' runs the script. The script's output is displayed: 'Hello World', 'Current date and time: Sun Apr 25 20:07:49 IST 2021', and 'Computer name: ubuntu-KVM'.

```
root@ubuntu-KVM:~# ls
firstbashscript.sh
root@ubuntu-KVM:~# chmod 755 firstbashscript.sh
root@ubuntu-KVM:~# sh firstbashscript.sh
Hello World
Current date and time: Sun Apr 25 20:07:49 IST 2021
Computer name: ubuntu-KVM
```

PRACTICAL NO: 02**Theory: HP controller, HP Network Protector, HP Network Visualizer, HP Network Optimizer, Cisco XNC Controller****Hewlett-Packard**

HP is the first to offer SDN technologies for all three critical layers—infrastructure, control software and application—to simplify networks and improve agility across the enterprise. These layers create a complete, open SDN hardware and software solution that provides a single point of control for the entire network.

The infrastructure layer delivers open programmable access through OpenFlow, a networking protocol that automates hardware configurations.

- New SDN functionality in the infrastructure layer enables clients to simplify network configuration. HP today announced nine additional switch models providing OpenFlow-enabled support for HP Flex Network architecture, offering clients a flexible and programmable standards-based interface. The addition of the new HP 3800 switch series to the 16 existing models in the product portfolio reaffirms HP as the only major networking vendor with more than 15 million installed OpenFlow-enabled ports. (1)

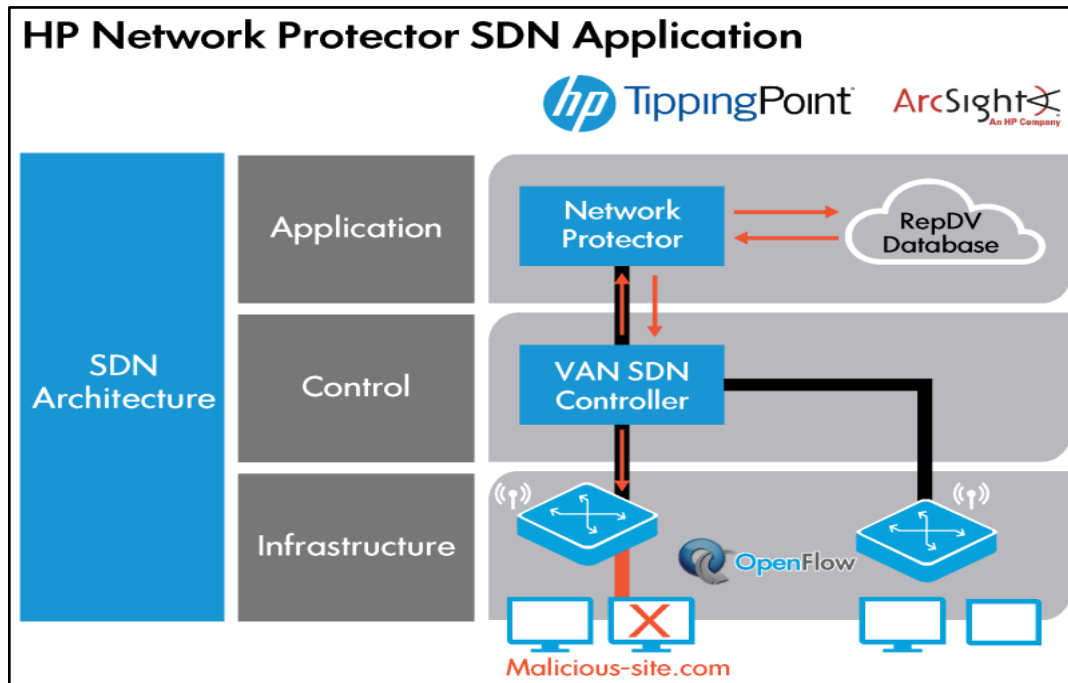
The control-software layer creates a centralized view of the network.

- Within the control layer, the new HP Virtual Application Networks SDN Controller abstracts the physical hardware from the logical deployment, providing a centralized view and automating network configuration of all devices in the infrastructure. By eliminating thousands of manual CLI entries, the controller enables network administrators to easily and flexibly program and scale their network environment for single-touch automated applications. It also provides application program interfaces (APIs) to third-party developers to integrate custom enterprise applications.
“We recognized the need to scale our perimeter firewall capacity to cope with the increase of internet traffic,” said Jean-Michel Juanito, communication systems group leader, IT Department, CERN. “In the framework of the CERN open lab R&D collaboration, we are developing with HP’s software defined network technologies a load-balancing application for the Virtual Application Networks SDN controller to distribute network traffic across multiple devices including firewalls and servers, increasing simplicity while reducing cost and bandwidth bottlenecks.”

The application layer delivers open programmable interfaces to automate applications across the network.

- New HP Virtual Cloud Networks software enables cloud providers to deliver automated and scalable public cloud services to enterprises. Using this software, enterprises can create an isolated virtual cloud network environment through a self-service public cloud infrastructure, providing them complete control for introducing new services and applications to their users.

Also at the application layer, the new HP Sentinel Security software application automates network access control and intrusion prevention security for enterprise campus networks with existing OpenFlow-enabled switch hardware through HP's controller. As a result, clients can eliminate the complexity and expense of dedicated networking hardware appliances, while achieving scalable security needed for bring-your-own-device (BYOD) initiatives.



HP Network Protector

The HP Network Protector SDN Application leverages HP Networking, Tipping Point, and ArcSight products to deliver a converged solution that addresses security threats in a completely new way by leveraging the network itself. The HP Network Protector enables network intelligence on network infrastructure devices. The application uses the HP VAN SDN (Virtual Application Network Software-Defined Networking) Controller and OpenFlow to program the network infrastructure with security intelligence from the HP Tipping Point Reputation Digital Vaccine (RepDV). This turns the entire network infrastructure into security-enforcement devices, providing visibility and threat protection against more than one million malicious botnets, malware, and spyware sites.

Some of the key features of the solution include:

- Runs on HP VAN SDN Controller
- Complimentary to Tipping Point IPS solutions
- Delivers real-time threat characterization with HP Tipping Point DVLabs Database
- Protects from over 1,000,000+ botnet, malware, spyware, and malicious sites
- OpenFlow enabled switches gain ability to detect malware, botnets, and other threats
- Ability to create custom whitelist and blacklist
- Improves visibility and accuracy with ArcSight Integration

- Dynamic switch learning with HPN OpenFlow enabled switches distributes detection into switch infrastructure

HP Network Visualizer

The HP Network Visualizer SDN Application by utilizing HP VAN SDN Controller provides dynamic traffic capture with real-time detailed network monitoring allowing for fast network diagnosis and verification, rapid transition from incident to fix.

Features

1. Real-time Visibility and Diagnosis

Network Visualizer provides dynamic traffic capture to diagnose the root cause of the network. It proactively monitors the network to reduce the number of help desk issues.

2. Low Cost, Simple and Automated Troubleshooting

Network Visualizer allows for simple troubleshooting that requires high level network detail. The application eliminates the need for any expensive manual network tapping tools for troubleshooting.

3. Fast Transition from Incident to Fix

Network Visualizer solves network issues in a matter of seconds versus minutes.

Network Optimizer

HP Network Optimizer SDN Application for Microsoft Lync enables automated provisioning of network policy and quality of service to provide an enhanced user experience. The Network Optimizer Application dynamically provisions the end-to-end network path and Quality of Service (QoS) policy via the HP Virtual Application Networks (VAN)SDN Controller, reducing the need for manual, device-by-device configuration via the CLI, which greatly simplifies policy deployment and reduces the likelihood of human errors.

Network Optimizer provides:

- Traffic classification – Identify application traffic requiring preferential treatment
- Policy enforcement – Prioritize time sensitive traffic based QoS profiles, supports up to 10K users within a single SDN domain
- Visibility into per call SDN operation in terms of DSCP marking applied
- Granular sorting and display filtering – by IP address, media type, call quality and jitter
- History records management
- Configurable global template and phone trust

Cisco XNC Controller

Cisco created the Cisco XNC Controller, in order to keep up with the changing software-defined networking (SDN) environments. Its support of OpenFlow, the most widely used SDN communications standard, helps it integrate into varied SDN deployments to enable organizations to better control and scale their networks. As of mid-2015, Cisco has retired the Cisco XNC Controller — see [What the Cisco XNC Controller Tells Us About OpenDaylight](#) to learn about XNC's demise.

As an SDN Controller, which is the “brains” of the network, Cisco XNC uses OpenFlow to communicate information “down” to the forwarding plane (switches and routers), with southbound APIs, and “up” to the applications and business logic, with northbound APIs. It enables organizations to deploy and even develop a variety of network services, using representational state transfer application program interfaces (REST APIs), as well as Java APIs.

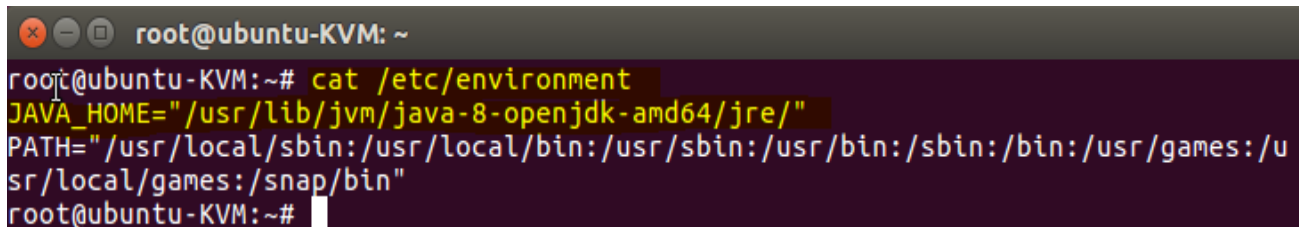
The XNC is Cisco's implementation of the OpenDaylight stack. Cisco is a contributor to the OpenDaylight initiative, which is focused on developing open standards for SDN that promote innovation and interoperability. Cisco XNC is designed to deliver the cutting edge OpenDaylight technologies as commercial, enterprise-ready solutions.

PRACTICAL NO: 03**Install ODL(Open Daylight) controller****STEP 1: Install java and Set JAVA_HOME**

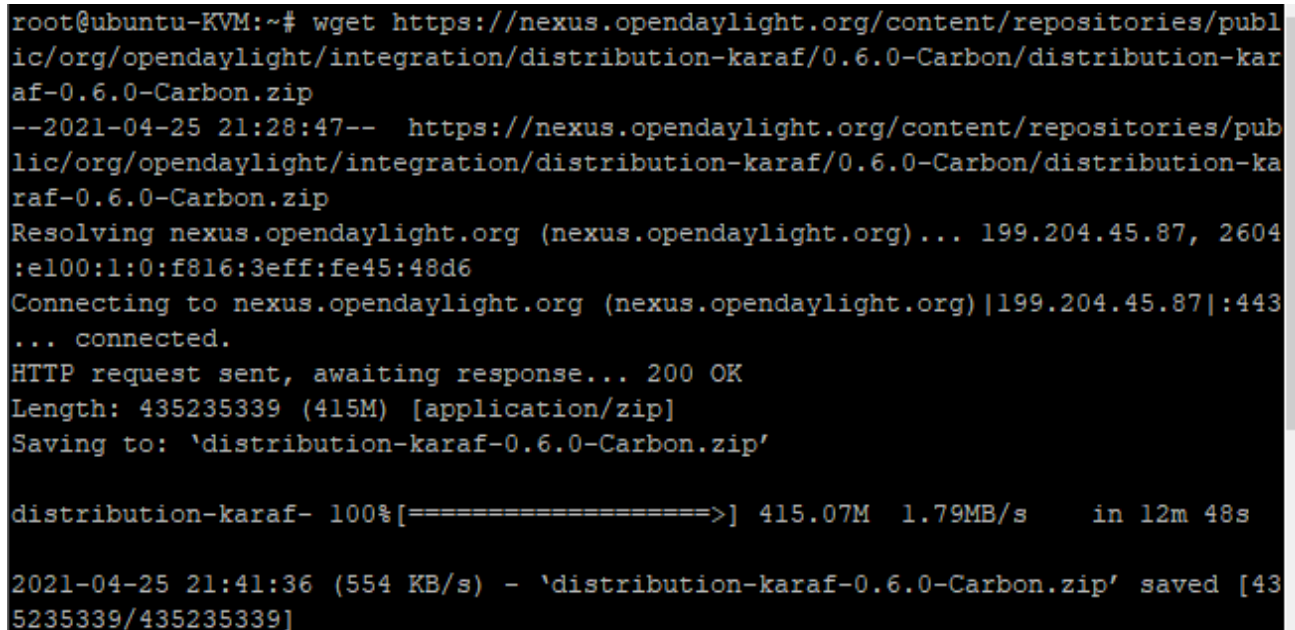
```
#sudo apt install openjdk-8-jre -y
```

Add below jre path in /etc/environment to set java path

```
#sudo nano /etc/environment
```

A terminal window titled 'root@ubuntu-KVM: ~' showing the command 'cat /etc/environment' and its output. The output sets 'JAVA_HOME' to '/usr/lib/jvm/java-8-openjdk-amd64/jre/' and 'PATH' to '/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin'.

```
root@ubuntu-KVM:~# cat /etc/environment
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre/"
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
root@ubuntu-KVM:~#
```

STEP 2: Download distribution-karaf-0.6.0-Carbon.zip and unzip it Move to the directory distribution-karaf-0.6.0-Carbon and type: ./bin/karafA terminal window showing the command 'wget https://nexus.opendaylight.org/content/repositories/public/org/opendaylight/integration/distribution-karaf/0.6.0-Carbon/distribution-karaf-0.6.0-Carbon.zip' and its output. The output shows the file being downloaded from nexus.opendaylight.org, with a size of 415M and a speed of 1.79MB/s. The file is saved as 'distribution-karaf-0.6.0-Carbon.zip'.

```
root@ubuntu-KVM:~# wget https://nexus.opendaylight.org/content/repositories/public/org/opendaylight/integration/distribution-karaf/0.6.0-Carbon/distribution-karaf-0.6.0-Carbon.zip
--2021-04-25 21:28:47-- https://nexus.opendaylight.org/content/repositories/public/org/opendaylight/integration/distribution-karaf/0.6.0-Carbon/distribution-karaf-0.6.0-Carbon.zip
Resolving nexus.opendaylight.org (nexus.opendaylight.org)... 199.204.45.87, 2604:el00:1:0:f816:3eff:fe45:48d6
Connecting to nexus.opendaylight.org (nexus.opendaylight.org)|199.204.45.87|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 435235339 (415M) [application/zip]
Saving to: 'distribution-karaf-0.6.0-Carbon.zip'

distribution-karaf- 100%[=====>] 415.07M  1.79MB/s   in 12m 48s

2021-04-25 21:41:36 (554 KB/s) - 'distribution-karaf-0.6.0-Carbon.zip' saved [435235339/435235339]
```

[illegible]

STEP 3: Install features

following ARE ODL-DLUX-ALL FEATURES

```
opendaylight-user@root> feature:install odl-dlux-core
```

```
opendaylight-user@root>feature:install odl-dluxapps-nodes
```

```
opendaylight-user@root>feature:install odl-dluxapps-topology
```

```
opendaylight-user@root>feature:install odl-dluxapps-yangui
```

```
opendaylight-user@root>feature:install odl-dluxapps-yangvisualizer
```

```
opendaylight-user@root>feature:install odl-dluxapps-yangman
```

```
opendaylight-user@root>feature:install odl-dlux-all
```

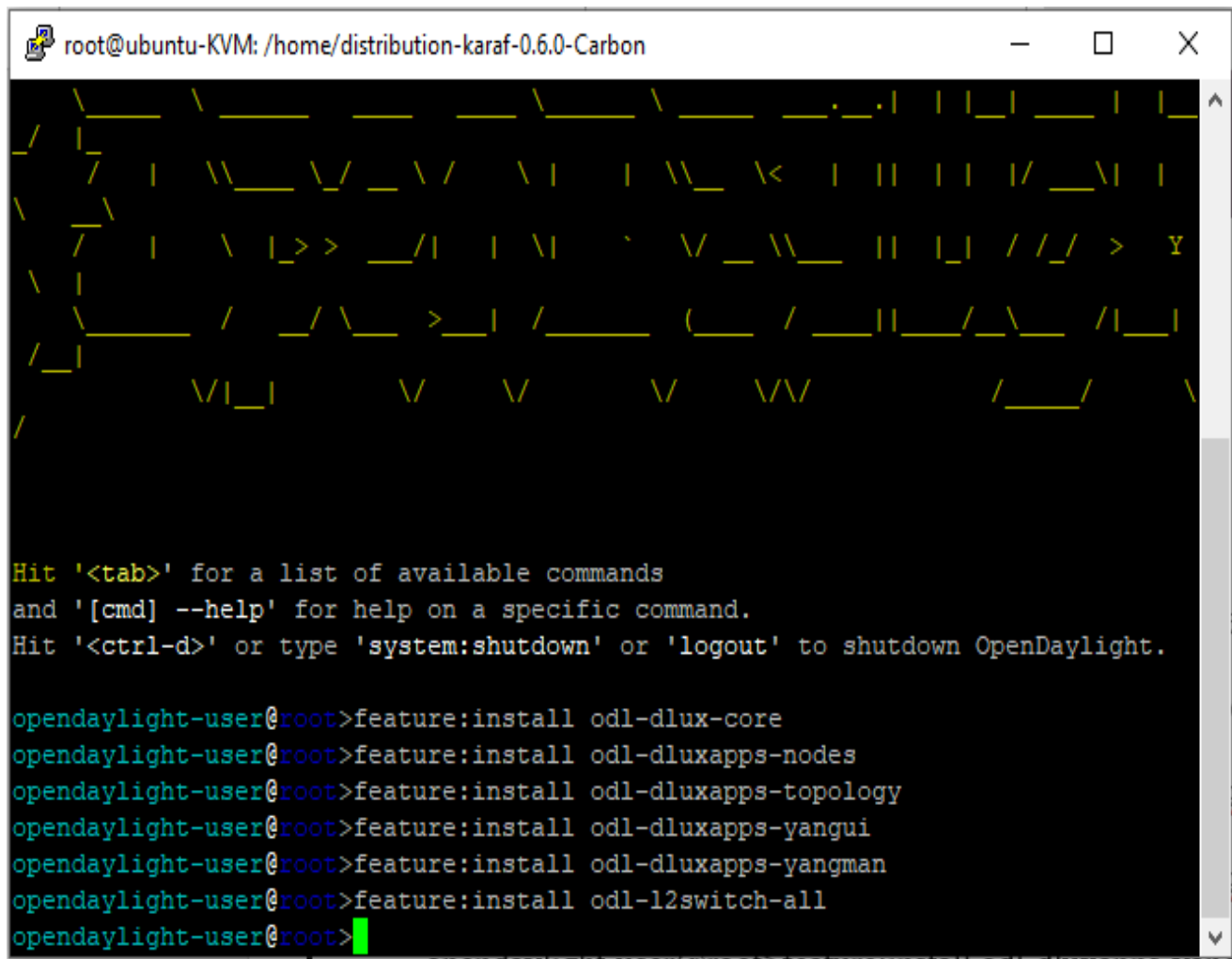
```
opendaylight-user@root> feature:install odl-restconf odl-l2switch-switch odl-  
mdsal-apidocs
```

```
opendaylight-user@root> feature:install odl-l2switch-all
```

```
opendaylight-user@root> feature:install odl-vtn-manager-neutron odl-neutron-  
service odl-neutron-hostconfig-ovs
```

```
opendaylight-user@root>feature:install odl-ovsdb-library odl-restconf-all odl-  
ovsdb-southbound-api odl-ovsdb-southbound-impl odl-ovsdb-southbound-impl-  
rest
```

```
opendaylight-user@root>feature:install odl-netvirt-openstack
```



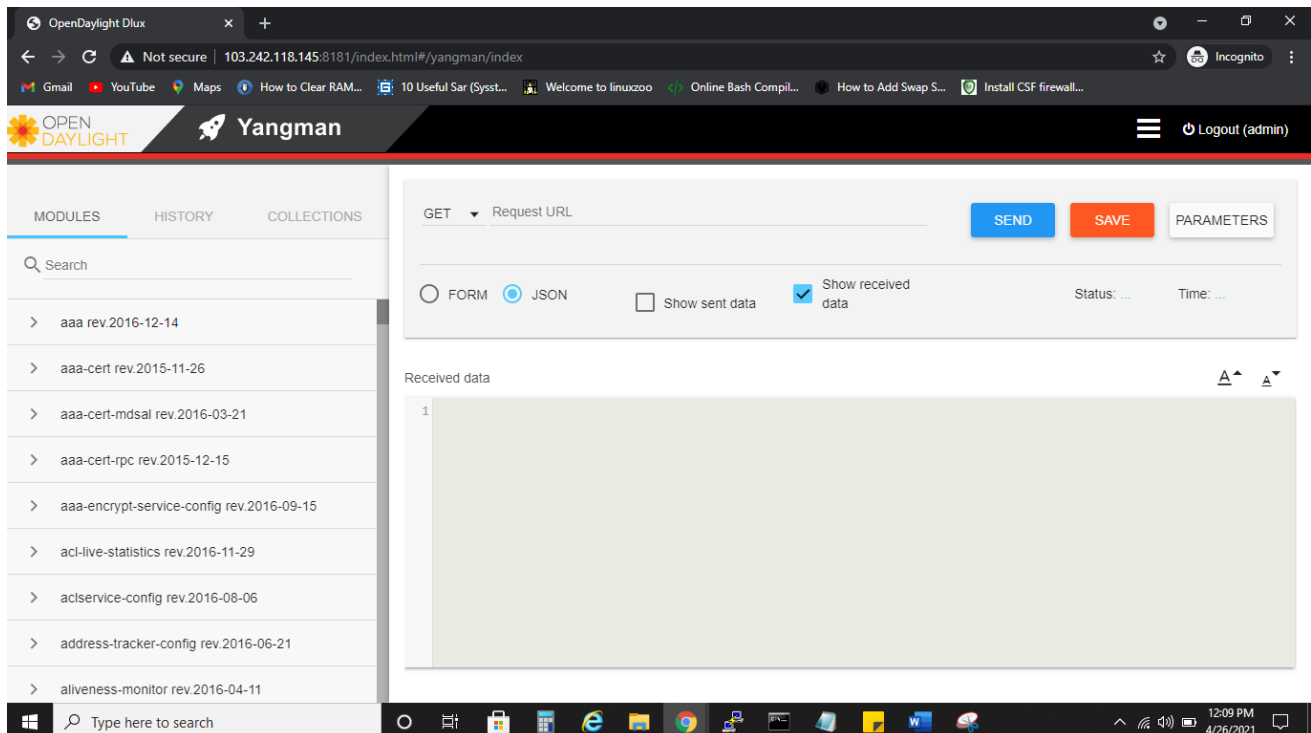
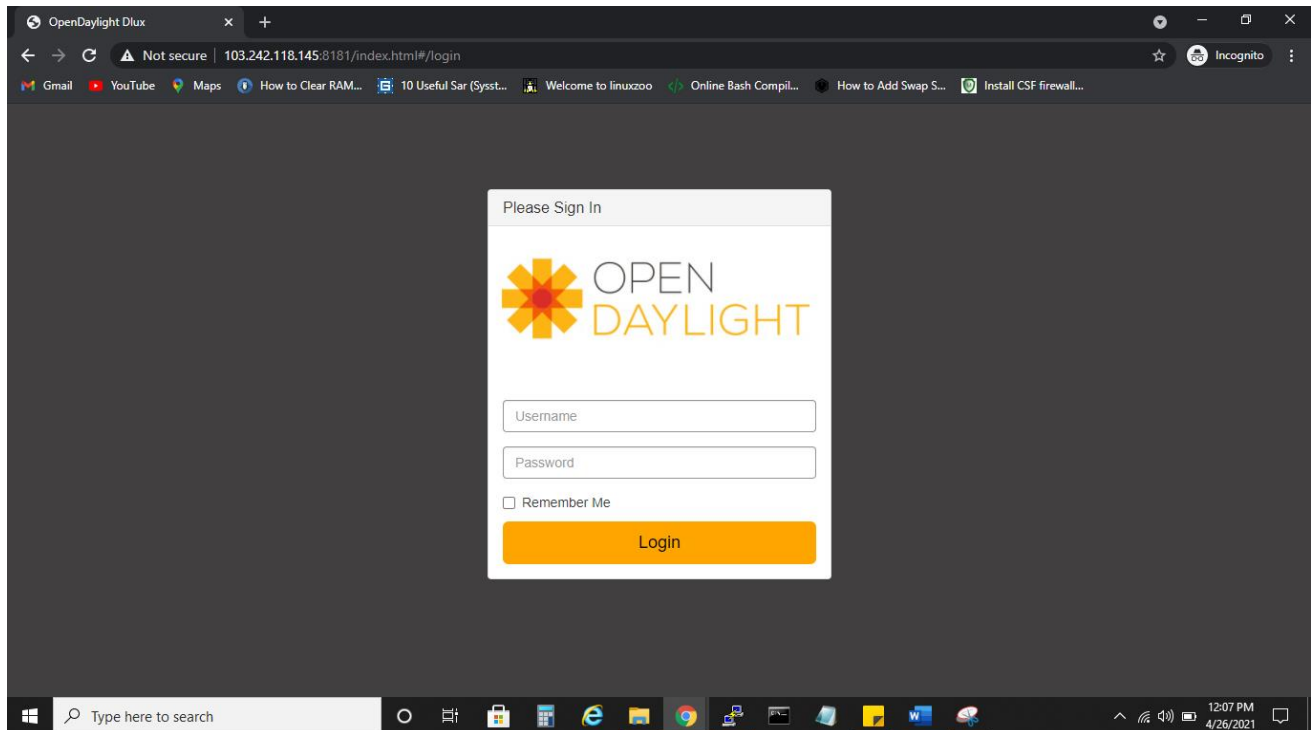
The screenshot shows a terminal window titled "root@ubuntu-KVM: /home/distribution-karaf-0.6.0-Carbon". The terminal displays a large ASCII art logo for OpenDaylight at the top. Below the logo, there are instructions: "Hit '<tab>' for a list of available commands", "and '[cmd] --help' for help on a specific command.", and "Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown OpenDaylight." Following these instructions, a series of "feature:install" commands are entered, installing various OpenDaylight features: odl-dlux-core, odl-dluxapps-nodes, odl-dluxapps-topology, odl-dluxapps-yangui, odl-dluxapps-yangman, and odl-l2switch-all. The terminal ends with the prompt "opendaylight-user@root>" and a green cursor.

```
root@ubuntu-KVM: /home/distribution-karaf-0.6.0-Carbon

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

opendaylight-user@root>feature:install odl-dlux-core
opendaylight-user@root>feature:install odl-dluxapps-nodes
opendaylight-user@root>feature:install odl-dluxapps-topology
opendaylight-user@root>feature:install odl-dluxapps-yangui
opendaylight-user@root>feature:install odl-dluxapps-yangman
opendaylight-user@root>feature:install odl-l2switch-all
opendaylight-user@root>
```


Even ODL has dashboard DLUX feature which gives GUI web interface to ODL controller at URL **<http://localhost:8181/index.html>** OR **<http://ip:8181/index.html>** on which ODL user can log in (by default username and password is 'admin') and check network topology on the browser of the system where ODL is running.

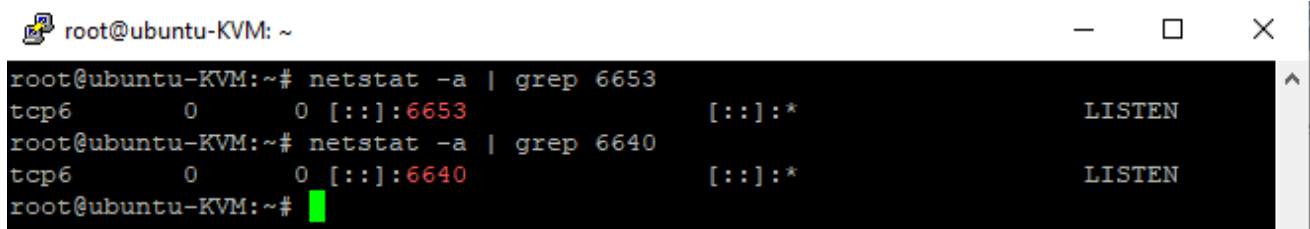


PRACTICAL NO: 04**Implement OVS**

Once OpenFlow and OVSDDB plugins are installed, ODL starts listening for OpenFlow and OVSDDB connections at 6653 and 6640 port respectively. This can be checked at controller by running the following commands:

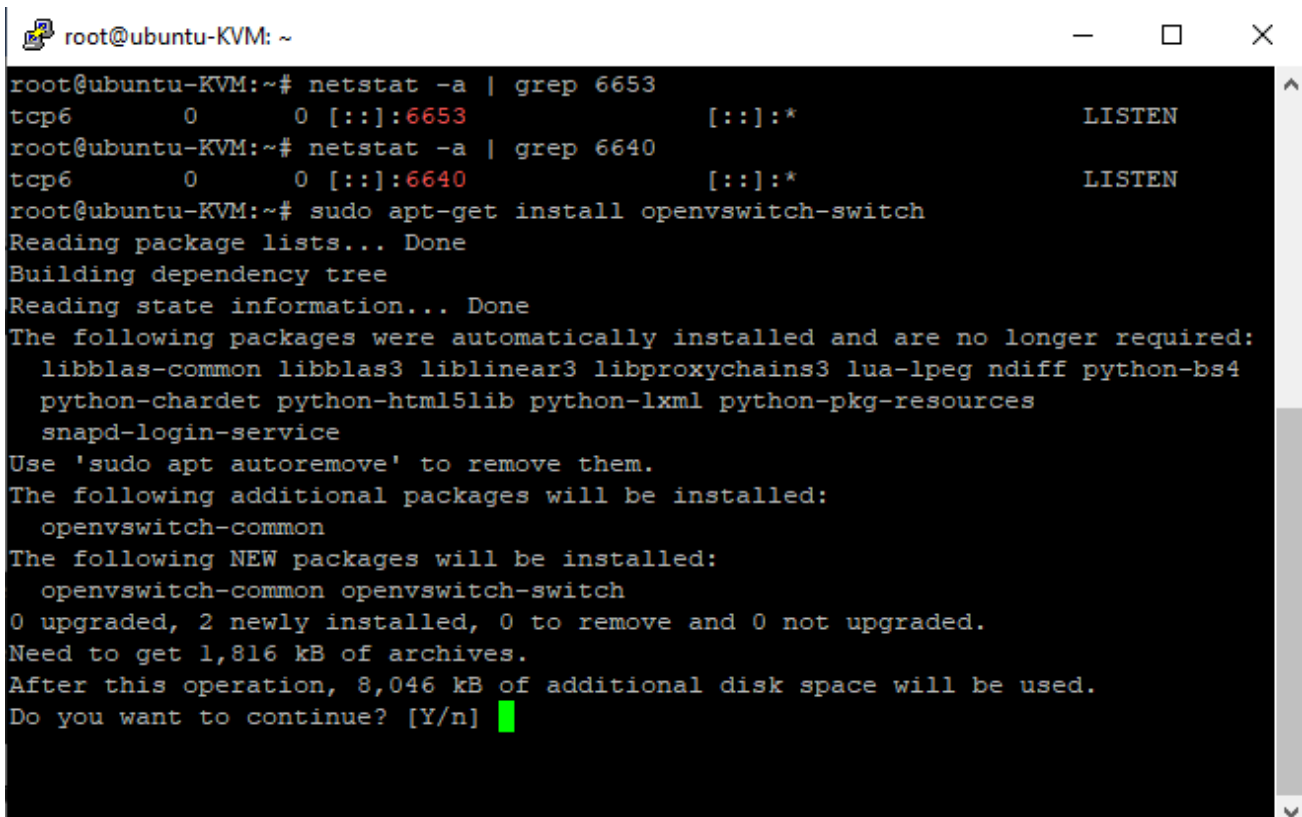
netstat -a | grep 6653

netstat -a | grep 6640



```
root@ubuntu-KVM: ~  
root@ubuntu-KVM:~# netstat -a | grep 6653  
tcp6      0      0 [::]:6653          [::]:*              LISTEN  
root@ubuntu-KVM:~# netstat -a | grep 6640  
tcp6      0      0 [::]:6640          [::]:*              LISTEN  
root@ubuntu-KVM:~#
```

sudo apt-get install openvswitch-switch

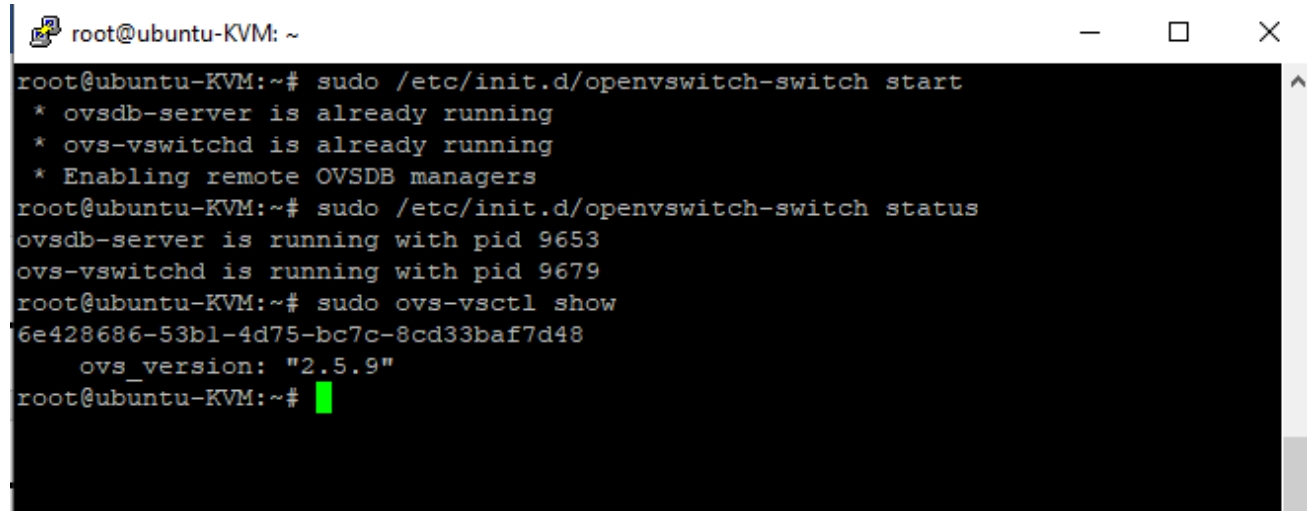


```
root@ubuntu-KVM: ~  
root@ubuntu-KVM:~# netstat -a | grep 6653  
tcp6      0      0 [::]:6653          [::]:*              LISTEN  
root@ubuntu-KVM:~# netstat -a | grep 6640  
tcp6      0      0 [::]:6640          [::]:*              LISTEN  
root@ubuntu-KVM:~# sudo apt-get install openvswitch-switch  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  libblas-common libblas3 liblinear3 libproxychains3 lua-lpeg ndiff python-bs4  
  python-chardet python-html5lib python-lxml python-pkg-resources  
  snapd-login-service  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  openvswitch-common  
The following NEW packages will be installed:  
  openvswitch-common openvswitch-switch  
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.  
Need to get 1,816 kB of archives.  
After this operation, 8,046 kB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

```
sudo /etc/init.d/openvswitch-switch start
```

```
sudo /etc/init.d/openvswitch-switch status
```

```
sudo ovs-vsctl show
```



```

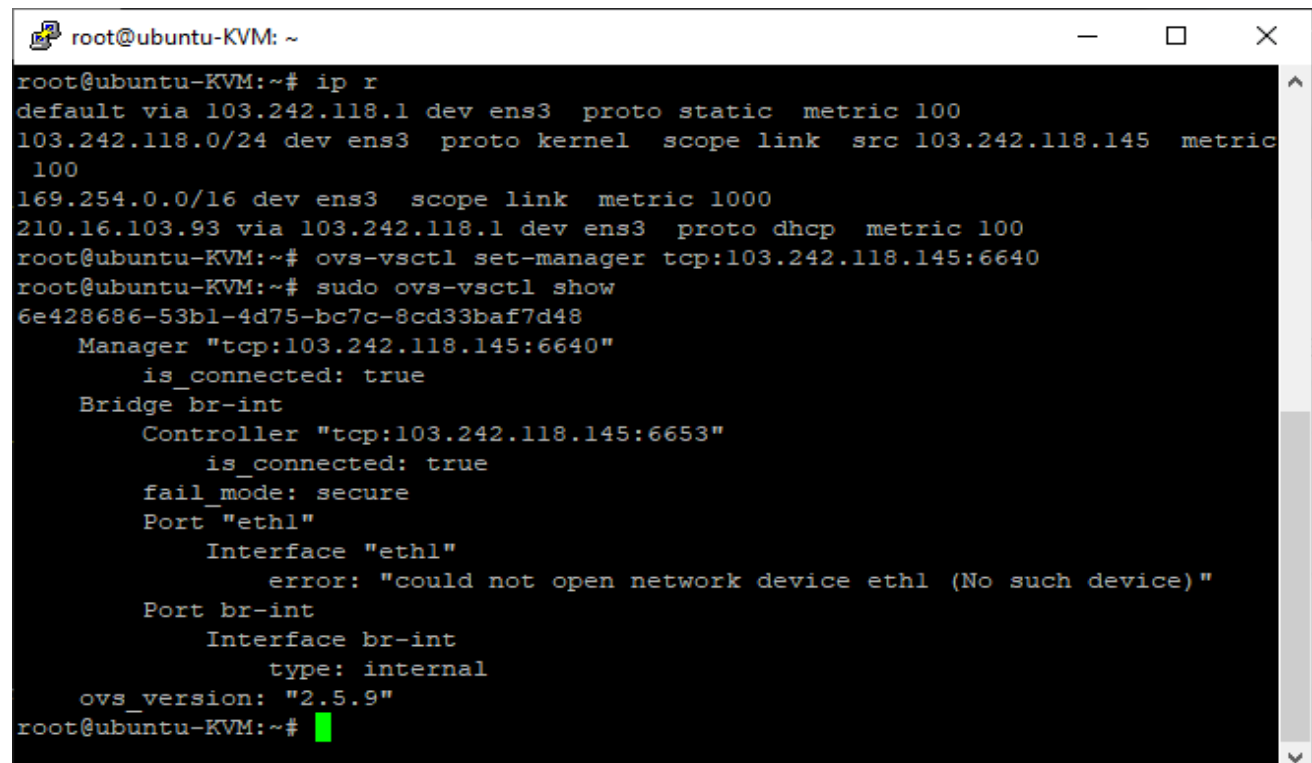
root@ubuntu-KVM: ~
root@ubuntu-KVM:~# sudo /etc/init.d/openvswitch-switch start
* ovsdb-server is already running
* ovs-vswitchd is already running
* Enabling remote OVSDb managers
root@ubuntu-KVM:~# sudo /etc/init.d/openvswitch-switch status
ovsdb-server is running with pid 9653
ovs-vswitchd is running with pid 9679
root@ubuntu-KVM:~# sudo ovs-vsctl show
6e428686-53b1-4d75-bc7c-8cd33baf7d48
    ovs_version: "2.5.9"
root@ubuntu-KVM:~#

```

At OVS side, below command is run to initiate OVSDb connection with controller:

```
ovs-vsctl set-manager tcp:<controller-IP>:6640
```

To confirm that the connection is established, below command can be run which shows 'is_connected' flag is set to true in the output.



```

root@ubuntu-KVM: ~
root@ubuntu-KVM:~# ip r
default via 103.242.118.1 dev ens3 proto static metric 100
103.242.118.0/24 dev ens3 proto kernel scope link src 103.242.118.145 metric 100
169.254.0.0/16 dev ens3 scope link metric 1000
210.16.103.93 via 103.242.118.1 dev ens3 proto dhcp metric 100
root@ubuntu-KVM:~# ovs-vsctl set-manager tcp:103.242.118.145:6640
root@ubuntu-KVM:~# sudo ovs-vsctl show
6e428686-53b1-4d75-bc7c-8cd33baf7d48
    Manager "tcp:103.242.118.145:6640"
        is_connected: true
    Bridge br-int
        Controller "tcp:103.242.118.145:6653"
            is_connected: true
        fail_mode: secure
        Port "eth1"
            Interface "eth1"
                error: "could not open network device eth1 (No such device)"
        Port br-int
            Interface br-int
                type: internal
    ovs_version: "2.5.9"
root@ubuntu-KVM:~#

```

OpenFlow connection is made on bridge, so either you can create bridge on OVS or ODL can also create bridge on OVS by sending configuration to OVS via OVSDDB connection. Over the bridge in OVS, below command can be run to connect bridge to ODL, which establishes establishing OpenFlow connection between ODL and OVS.

ovs-vsctl set-controller <bridge name> tcp:<controller-IP>:6653

ovs-vsctl set-controller br-int tcp: 103.242.118.145:6653

```

root@ubuntu-KVM: ~
root@ubuntu-KVM:~# sudo ovs-vsctl set-controller br-int tcp:103.242.118.145:6653
root@ubuntu-KVM:~# sudo ovs-vsctl show
6e428686-53b1-4d75-bc7c-8cd33baf7d48
    Manager "tcp:103.242.118.145:6640"
        is_connected: true
    Bridge br-int
        Controller "tcp:103.242.118.145:6653"
            is_connected: true
        fail_mode: secure
        Port "eth1"
            Interface "eth1"
                error: "could not open network device eth1 (No such device)"
        Port br-int
            Interface br-int
                type: internal
        ovs_version: "2.5.9"
root@ubuntu-KVM:~#

```

At the OVS side, the following command can be run to show details of bridge.

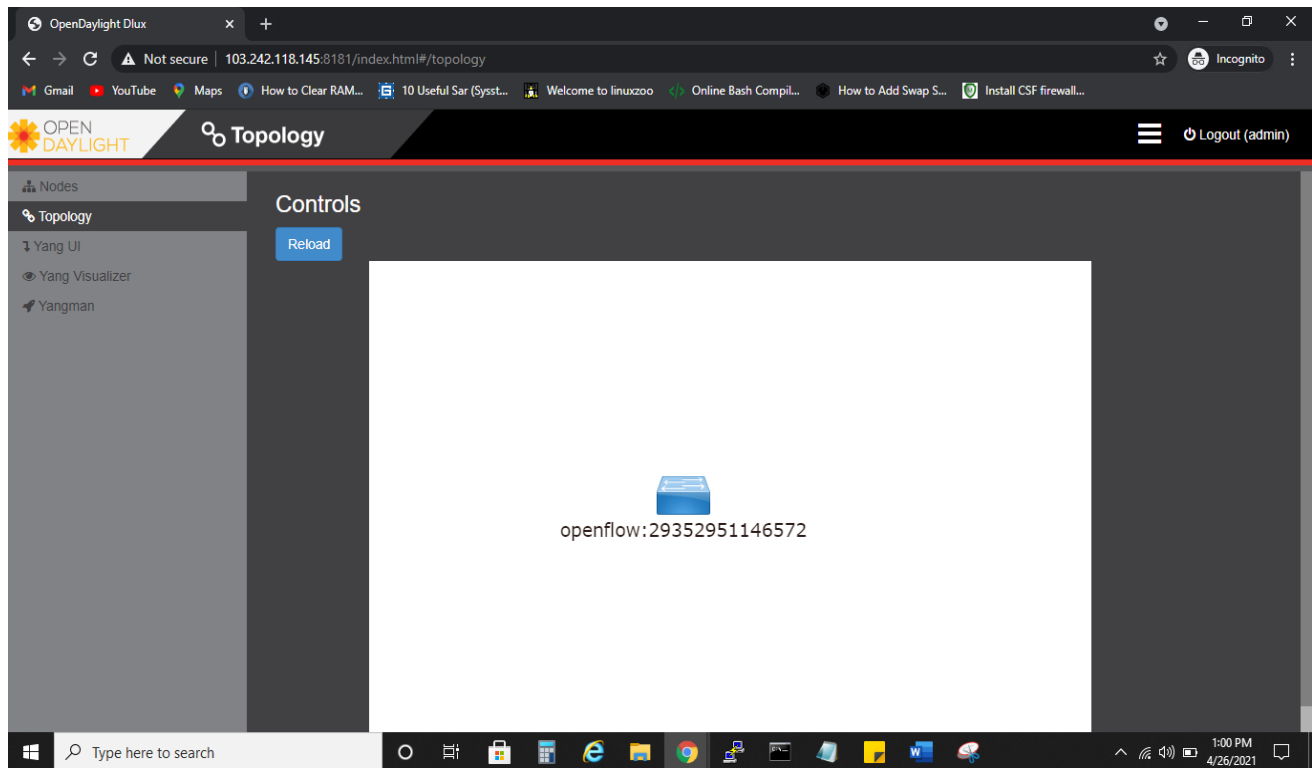
sudo ovs-ofctl show br-int -OOpenFlow13

```

root@ubuntu-KVM: ~
root@ubuntu-KVM:~# sudo ovs-ofctl show br-int -OOpenFlow13
OFPT_FEATURES_REPLY (OF1.3) (xid=0x2): dpid:00001ab24439e84c
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS GROUP_STATS QUEUE_STATS
OFPT_PORT_DESC reply (OF1.3) (xid=0x3):
    LOCAL(br-int): addr:1a:b2:44:39:e8:4c
        config:      PORT_DOWN
        state:        LINK_DOWN
        speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (OF1.3) (xid=0x5): frags=normal miss_send_len=0
root@ubuntu-KVM:~#

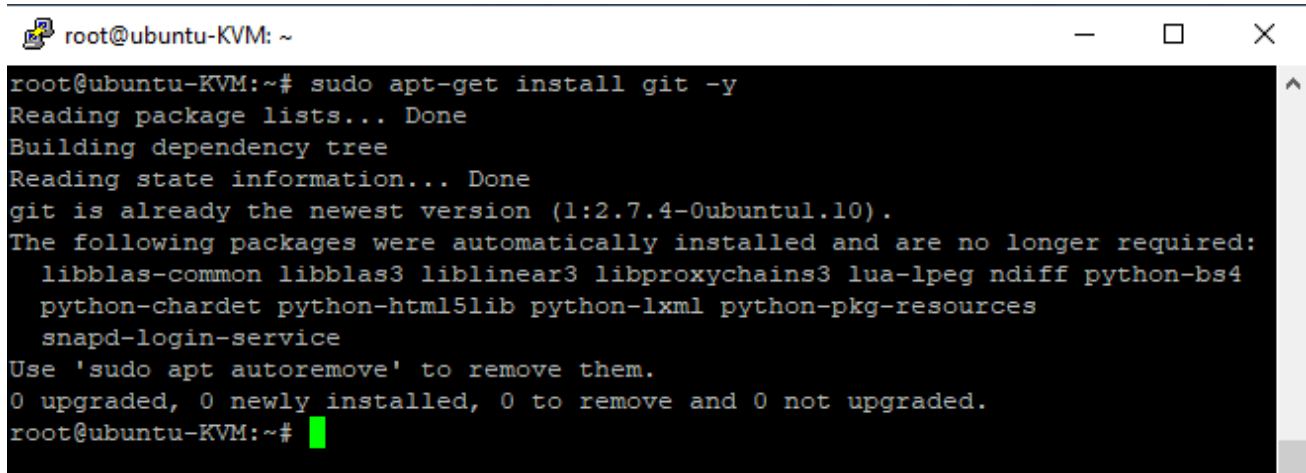
```

Restart odl and login to view OpenFlow switch



PRACTICAL NO: 05**Implement Mininet on ODL**

```
sudo apt-get install git
```

A terminal window titled 'root@ubuntu-KVM: ~' with standard window controls. The terminal output shows the command 'sudo apt-get install git -y' being executed. It reports that git is already the newest version (1:2.7.4-0ubuntu1.10) and lists several packages that were automatically installed and are no longer required: libblas-common, libblas3, liblinear3, libproxychains3, lua-lpeg, ndiff, python-bs4, python-chardet, python-html5lib, python-lxml, python-pkg-resources, and snapd-login-service. It suggests using 'sudo apt autoremove' to remove them and shows that 0 packages were upgraded, 0 newly installed, 0 to be removed, and 0 not upgraded. The prompt returns to 'root@ubuntu-KVM:~#'.

```
root@ubuntu-KVM:~# sudo apt-get install git -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.7.4-0ubuntu1.10).
The following packages were automatically installed and are no longer required:
  libblas-common libblas3 liblinear3 libproxychains3 lua-lpeg ndiff python-bs4
  python-chardet python-html5lib python-lxml python-pkg-resources
  snapd-login-service
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ubuntu-KVM:~#
```

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

A terminal window titled 'root@ubuntu-KVM: ~' with standard window controls. The terminal output shows the command 'git clone git://github.com/mininet/mininet' being executed. It shows the progress of cloning into a folder named 'mininet', including enumerating 10165 objects, counting 100%, compressing 100%, and receiving 10165 objects (3.19 MiB) at 2.11 MiB/s. It also shows resolving deltas and checking connectivity. The prompt returns to 'root@ubuntu-KVM:~#'.

```
root@ubuntu-KVM:~# git clone git://github.com/mininet/mininet
Cloning into 'mininet'...
remote: Enumerating objects: 10165, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10165 (delta 2), reused 7 (delta 2), pack-reused 10154
Receiving objects: 100% (10165/10165), 3.19 MiB | 2.11 MiB/s, done.
Resolving deltas: 100% (6783/6783), done.
Checking connectivity... done.
root@ubuntu-KVM:~#
```

This creates a folder in the home directory named mininet that contains the project file structure.

To find the latest beta version of Mininet, list all tagged releases in the Mininet project.

cd /root/mininet

git tag

```
root@ubuntu-KVM: ~/mininet
root@ubuntu-KVM:~# ls
distribution-karaf-0.6.0-Carbon.zip  firstbashscript.sh  mininet
root@ubuntu-KVM:~# cd mininet/
root@ubuntu-KVM:~/mininet# git tag
1.0.0
2.0.0
2.1.0
2.1.0p1
2.1.0p2
2.2.0
2.2.1
2.2.2
2.3.0
2.3.0b1
2.3.0b2
2.3.0d3
2.3.0d4
2.3.0d5
2.3.0d6
2.3.0rc1
2.3.0rc2
cs244-spring-2012-final
root@ubuntu-KVM:~/mininet#
```

The Mininet project provides an install script. Run the script.

git checkout -b cs244-spring-2012-final

Sudo ~/mininet/util/install.sh -a

```

root@ubuntu-KVM:~/mininet/util# pwd
/root/mininet/util
root@ubuntu-KVM:~/mininet/util# ls
build-ovs-packages.sh  install.sh  openflow-patches  unpep8
colorfilters           kbuild     sch_htb-ofbuf      vm
doxify.py              nox-patches sysctl_addon
root@ubuntu-KVM:~/mininet/util# ./install.sh -a
Detected Linux distribution: Ubuntu 16.04 xenial amd64
Running all commands...
Install Mininet-compatible kernel if necessary
Hit:1 http://in.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Hit:4 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu xenial InRelease
Get:5 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu xenial-updates/main amd64 DEP-11 Metad
ata [326 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 DEP-11 M
etadate [281 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 DEP-11
Metadata [5,980 B]
Get:9 http://in.archive.ubuntu.com/ubuntu xenial-backports/main amd64 DEP-11 Met
adata [3,328 B]
Get:10 http://in.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 DEP-1
1 Metadata [6,616 B]
Get:11 http://security.ubuntu.com/ubuntu xenial-security/main amd64 DEP-11 Metad
ata [93.7 kB]
Get:12 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 M
etadate [130 kB]
Get:13 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 DEP-11
Metadata [2,464 B]
Fetched 1,175 kB in 2s (447 kB/s)

```

```

endfile --log-file /var/log/openvswitch/ovs-vswitchd.log --pidfile /var/run/openvswitch/ovs-v
root      24914  0.0  0.0 14224 1084 pts/1    S+   14:04   0:00 grep --color=auto -i ovs
root@ubuntu-KVM:~/mininet# kill -9 24492 24493 24503 24504 24914

```

sudo service openvswitch-switch stop

Close Opendaylight by ctrl + D

service openvswitch-switch start


```

root@ubuntu-KVM: ~/mininet
root@ubuntu-KVM:~/mininet# service openvswitch-switch status
● openvswitch-switch.service - Open vSwitch
   Loaded: loaded (/lib/systemd/system/openvswitch-switch.service; enabled; vendor preset: enabled)
   Active: active (exited) since Wed 2021-04-28 14:05:47 IST; 5min ago
     Process: 24963 ExecStop=/bin/true (code=exited, status=0/SUCCESS)
     Process: 25073 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 25073 (code=exited, status=0/SUCCESS)

Apr 28 14:05:47 ubuntu-KVM systemd[1]: Starting Open vSwitch...
Apr 28 14:05:47 ubuntu-KVM systemd[1]: Started Open vSwitch.
root@ubuntu-KVM:~/mininet# █

```

Sudo mn

 root@ubuntu-KVM: /home/mininet

— □ ×

```
root@ubuntu-KVM:/home/mininet# sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> █
```

sudo mn --test pingall

```
root@ubuntu-KVM: /home/mininet
root@ubuntu-KVM:/home/mininet# sudo mn --test pingall
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.879 seconds
root@ubuntu-KVM:/home/mininet#
```

root@ubuntu-KVM: ~

```
root@ubuntu-KVM:~# mn --controller=remote,ip=103.242.117.180 --topo=tree,1,3 --mac
*** Creating network
*** Adding controller
Unable to contact the remote controller at 103.242.117.180:6653
Unable to contact the remote controller at 103.242.117.180:6633
Setting remote controller to 103.242.117.180:6653
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(s1, h1) (s1, h2) (s1, h3)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

sudo mn --controller=remote,ip=103.242.117.180 --mac --arp

root@ubuntu-KVM: /home/mininet

```
root@ubuntu-KVM:/home/mininet# mn --controller=remote,ip=103.242.117.180 --mac --arp
*** Creating network
*** Adding controller
Unable to contact the remote controller at 103.242.117.180:6653
Unable to contact the remote controller at 103.242.117.180:6633
Setting remote controller to 103.242.117.180:6653
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

```
mininet> s1 ovs-ofctl r-flow "s1"  
priority=1,ip,nw_dst=10.0.0.1,actions=mod_dl_dst:00:00:00:00:00:01,output=1
```

```
mininet> s1 ovs-ofctl add-flow "s1"  
priority=1,ip,nw_dst=10.0.0.2,actions=mod_dl_dst:00:00:00:00:00:02,output=2
```

```
mininet> sh ovs-ofctl add-flow s1 dl_type=0x806,nw_proto=1,actions=flood
```

```
mininet> h1 ping -c2 h2
```

```
root@ubuntu-KVM: /home/mininet  
root@ubuntu-KVM:/home/mininet# mn --controller=remote,ip=103.242.117.180 --mac --arp  
*** Creating network  
*** Adding controller  
Unable to contact the remote controller at 103.242.117.180:6653  
Unable to contact the remote controller at 103.242.117.180:6633  
Setting remote controller to 103.242.117.180:6653  
*** Adding hosts:  
h1 h2  
*** Adding switches:  
s1  
*** Adding links:  
(h1, s1) (h2, s1)  
*** Configuring hosts  
h1 h2  
*** Starting controller  
c0  
*** Starting 1 switches  
s1 ...  
*** Starting CLI:  
mininet> nodes  
available nodes are:  
c0 h1 h2 s1  
mininet> s1 ovs-ofctl add-flow "s1" priority=1,ip,nw_dst=10.0.0.1,actions=mod_dl_dst:00:00:00:00:00:01,output=1  
mininet> s1 ovs-ofctl add-flow "s1" priority=1,ip,nw_dst=10.0.0.2,actions=mod_dl_dst:00:00:00:00:00:02,output=2  
mininet> sh ovs-ofctl add-flow s1 dl_type=0x806,nw_proto=1,actions=flood  
mininet> h1 ping -c2 h2  
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.  
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.738 ms  
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.081 ms  
  
--- 10.0.0.2 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1019ms  
rtt min/avg/max/mdev = 0.081/0.409/0.738/0.329 ms  
mininet>
```

Start the.opendaylight(odl)

```

at org.eclipse.osgi.framework.internal.core.Framework.notifyEventHooksPrivileged(Framework.java:1602)
at org.eclipse.osgi.framework.internal.core.Framework.publishBundleEventPrivileged(Framework.java:1557)
at org.eclipse.osgi.framework.internal.core.Framework.publishBundleEvent(Framework.java:1504)
at org.eclipse.osgi.framework.internal.core.Framework.publishBundleEvent(Framework.java:1499)
at org.eclipse.osgi.framework.internal.core.BundleHost.startWorker(BundleHost.java:391)
at org.eclipse.osgi.framework.internal.core.AbstractBundle.resume(AbstractBundle.java:390)
at org.eclipse.osgi.framework.internal.core.Framework.resumeBundle(Framework.java:1176)
at org.eclipse.osgi.framework.internal.core.StartLevelManager.resumeBundles(StartLevelManager.java:559)
at org.eclipse.osgi.framework.internal.core.StartLevelManager.resumeBundles(StartLevelManager.java:544)
at org.eclipse.osgi.framework.internal.core.StartLevelManager.incFWSL(StartLevelManager.java:457)
at org.eclipse.osgi.framework.internal.core.StartLevelManager.doSetStartLevel(StartLevelManager.java:243)
at org.eclipse.osgi.framework.internal.core.StartLevelManager.dispatchEvent(StartLevelManager.java:438)
at org.eclipse.osgi.framework.eventmgr.EventManager.dispatchEvent(EventManager.java:230)
at org.eclipse.osgi.framework.eventmgr.EventManager$EventThread.run(EventManager.java:340)

100% [=====]

Karaf started in 55s. Bundle stats: 414 active, 414 total

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

opendaylight-user@root>

```

OpenDaylight Nodes

Search Nodes

Node Id	Node Name	Node Connectors	Statistics
openflow:156898680233537	None	1	Flows Node Connectors

Logout (admin)

PRACTICAL NO: 06

Install RYU controller with mininet topology

Step 1. Install tools

```
sudo apt-get -y install git python-pip python-dev
```

```
root@ubuntu-KVM: /home
root@ubuntu-KVM:/home# apt-get -y install git python-pip python-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-dev is already the newest version (2.7.15~rc1-1).
python-dev set to manually installed.
git is already the newest version (1:2.17.1-1ubuntu0.8).
python-pip is already the newest version (9.0.1-2.3~ubuntu18.04.4).
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol girl1.2-gst-plugins-base-1.0 girl1.2-gstreamer-1.0 girl1.2-gudev-1.0
 libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1 li
 libcmis-0.5-5v5 libcolamd2 libdazzle-1.0-0 libe-book-0.1-1 libedataserverui-1.2-2 libeot0 libepubgen
 libfreerdp-client2-2 libfreerdp2-2 libgcl2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpo
 liblua5.3-0 libmediaart-2.0-0 libmtp-0.1-1 libodfgen-0.1-1 libqgwing2v5 libraw16 libvenge-0.0-0
 libwinpr2-2 libxapian3 libxmlsec1 libxmlsec1-nss lp-solve media-player-info python3-mako python3-ma
 usb-creator-common
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
1 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Setting up openvswitch-testcontroller (2.9.8-0ubuntu0.18.04.2) ...
```

Step 2. Install python packages

```
sudo apt-get -y install python-eventlet python-routes python-webob python-paramiko
```

```
root@ubuntu-KVM: /home
root@ubuntu-KVM:/home# sudo apt-get -y install python-eventlet python-routes python-webob python-paramiko
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol girl1.2-gst-plugins-base-1.0 girl1.2-gstreamer-1.0 girl1.2-gudev-1.0
 libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1 li
 libcmis-0.5-5v5 libcolamd2 libdazzle-1.0-0 libe-book-0.1-1 libedataserverui-1.2-2 libeot0 libepubgen
 libfreerdp-client2-2 libfreerdp2-2 libgcl2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpo
 liblua5.3-0 libmediaart-2.0-0 libmtp-0.1-1 libodfgen-0.1-1 libqgwing2v5 libraw16 libvenge-0.0-0
 libwinpr2-2 libxapian3 libxmlsec1 libxmlsec1-nss lp-solve media-player-info python3-mako python3-ma
 usb-creator-common
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 python-greenlet python-pyasn1 python-repoze.lru python-simplejson
Suggested packages:
 python-eventlet-doc python-greenlet-doc python-greenlet-dev python-greenlet-dbg python-gssapi python
The following NEW packages will be installed:
 python-eventlet python-greenlet python-paramiko python-pyasn1 python-repoze.lru python-routes python
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 625 kB of archives.
After this operation, 3,717 kB of additional disk space will be used.
```

Step 3. Clone RYU git Repo

```
git clone --depth=1 https://github.com/osrg/ryu.git
```

```
root@ubuntu-KVM: /home#
root@ubuntu-KVM:/home# git clone --depth=1 https://github.com/osrg/ryu.git
Cloning into 'ryu'...
remote: Enumerating objects: 1536, done.
remote: Counting objects: 100% (1536/1536), done.
remote: Compressing objects: 100% (1037/1037), done.
remote: Total 1536 (delta 488), reused 1016 (delta 307), pack-reused 0
Receiving objects: 100% (1536/1536), 1.43 MiB | 1.98 MiB/s, done.
Resolving deltas: 100% (488/488), done.
root@ubuntu-KVM:/home# ls
distribution-karaf-0.6.0-Carbon  distribution-karaf-0.6.0-Carbon.zip  mininet  openflow  pox  ryu  ubuntu
root@ubuntu-KVM:/home#
```

Step 4. Install RYU

```
sudo pip install setuptools --upgrade
```

```
root@ubuntu-KVM:/home# sudo pip install setuptools --upgrade
Collecting setuptools
  Downloading https://files.pythonhosted.org/packages/e1/b7/182161210a13158cd3ccc41eel9aad
  100% | ████████████████████████████████████████████████████████████████████████████████ | 583kB 519kB/s
Installing collected packages: setuptools
  Found existing installation: setuptools 39.0.1
  Not uninstalling setuptools at /usr/lib/python2.7/dist-packages, outside environment /
Successfully installed setuptools-44.1.1
root@ubuntu-KVM:/home# cd ryu/
root@ubuntu-KVM:/home/ryu#
```

```
cd ryu
```

```
sudo python ./setup.py install
```

```
root@ubuntu-KVM: /home/ryu#
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/lib/test_addrconv.py -> /usr/local/lib/python2.7/dist-packages/ryu/tes
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/lib/test_pcaplib.py -> /usr/local/lib/python2.7/dist-packages/ryu/tes
creating /usr/local/lib/python2.7/dist-packages/ryu/tests/unit/lib/ovs
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/lib/ovs/test_vsetl.py -> /usr/local/lib/python2.7/dist-packages/ryu/t
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/lib/ovs/__init__.py -> /usr/local/lib/python2.7/dist-packages/ryu/tes
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/lib/test_ofctl_action_match.py -> /usr/local/lib/python2.7/dist-packa
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/lib/test_ofctl_vl_3.py -> /usr/local/lib/python2.7/dist-packages/ryu/
creating /usr/local/lib/python2.7/dist-packages/ryu/tests/unit/ofproto
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/ofproto/test_parser_vl0.py -> /usr/local/lib/python2.7/dist-packages/
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/ofproto/test_ether.py -> /usr/local/lib/python2.7/dist-packages/ryu/t
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/ofproto/test_ofproto_parser.py -> /usr/local/lib/python2.7/dist-packa
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/ofproto/test_ofproto_vl2.py -> /usr/local/lib/python2.7/dist-packages
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/ofproto/test_parser_vl2.py -> /usr/local/lib/python2.7/dist-packages/
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/ofproto/test_parser_compat.py -> /usr/local/lib/python2.7/dist-packag
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/ofproto/test_oxm.py -> /usr/local/lib/python2.7/dist-packages/ryu/tes
creating /usr/local/lib/python2.7/dist-packages/ryu/tests/unit/ofproto/json
creating /usr/local/lib/python2.7/dist-packages/ryu/tests/unit/ofproto/json/of14
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/ofproto/json/of14/5-6-ofp_features_reply.packet.json -> /usr/local/li
json/of14
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/ofproto/json/of14/5-48-ofp_meter_stats_reply.packet.json -> /usr/loca
oto/json/of14
copying build/lib.linux-x86_64-2.7/ryu/tests/unit/ofproto/json/of14/5-21-ofp_group_mod.packet.json -> /usr/local/lib/py
/of14
```


Step 5. Install and Update python packages

```
sudo pip install six --upgrade
```

```
sudo pip install oslo.config msgpack-python
```

```
sudo pip install eventlet --upgrade
```

```
root@ubuntu-KVM:/home/ryu# sudo pip install six --upgrade
Collecting six
  Downloading https://files.pythonhosted.org/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1...
Installing collected packages: six
  Found existing installation: six 1.11.0
    Not uninstalling six at /usr/lib/python2.7/dist-packages, outside environment /usr
Successfully installed six-1.15.0
root@ubuntu-KVM:/home/ryu# sudo pip install oslo.config msgpack-python
Collecting oslo.config
  Downloading https://files.pythonhosted.org/packages/5c/45/53255cdbabefabc4ac8ae36671ecbd8a92a6d0293...
    100% |██████████| 133kB 4.1MB/s
Collecting msgpack-python
  Downloading https://files.pythonhosted.org/packages/8a/20/6eca772d1a5830336f84acald8198e5a3f4715cdl...
    100% |██████████| 143kB 1.2MB/s
Requirement already satisfied: enum34>=1.0.4; python_version == "2.7" or python_version == "2.6" or p...
lo.config)
Collecting stevedore>=1.20.0 (from oslo.config)
  Downloading https://files.pythonhosted.org/packages/e6/49/a35dd566626892d577e426dbe5ea424dd7fbe1064...
    100% |██████████| 51kB 6.2MB/s
Collecting PyYAML>=3.12 (from oslo.config)
  Downloading https://files.pythonhosted.org/packages/ba/d4/3cf562876e0cda0405e65d351b835077ab13990e5...
(574kB)
    100% |██████████| 583kB 514kB/s
Collecting requests>=2.18.0 (from oslo.config)
  Downloading https://files.pythonhosted.org/packages/29/cl/24814557fld22c56d50280771a17307e6bf87b707...
    100% |██████████| 61kB 913kB/s
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python2.7/dist-packages (from oslo.conf...
Collecting oslo.i18n>=3.15.3 (from oslo.config)
  Downloading https://files.pythonhosted.org/packages/4e/a4/e2c71ea714266097b2edc5188462cfc8f0fffd277...
    100% |██████████| 51kB 5.1MB/s
Collecting netaddr>=0.7.18 (from oslo.config)
  Downloading https://files.pythonhosted.org/packages/ff/cd/9cdf8a8fc45c56680b798db6a55fa60a22e2d3d3c...
    100% |██████████| 1.9MB 167kB/s
Collecting debtcollector>=1.2.0 (from oslo.config)
```


Step 6. Test ryu-manager

```
sudo pip install -r tools/pip-requires
```

```
root@ubuntu-KVM: /home/ryu
root@ubuntu-KVM:/home/ryu# sudo pip install -r tools/pip-requires
Collecting pip==20.3.4 (from -r tools/./pip-requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/27/79/8a850fe3496446ff0d584327ae44e7500daf
  100% |████████████████████████████████████████| 1.5MB 204kB/s
Collecting eventlet==0.30.1 (from -r tools/pip-requires (line 5))
  Downloading https://files.pythonhosted.org/packages/95/31/8daea2c6d5b6a9c567da07e976f8b4112dd1
  100% |████████████████████████████████████████| 225kB 1.2MB/s
Collecting msgpack>=0.4.0 (from -r tools/pip-requires (line 6))
  Downloading https://files.pythonhosted.org/packages/59/04/87fc6708659c2ed3b0b6d4954f270b6e931d
  100% |████████████████████████████████████████| 133kB 1.3MB/s
Requirement already satisfied: netaddr in /usr/local/lib/python2.7/dist-packages (from -r tools/
Requirement already satisfied: oslo.config>=2.5.0 in /usr/local/lib/python2.7/dist-packages (fro
Collecting ovs>=2.6.0 (from -r tools/pip-requires (line 9))
  Downloading https://files.pythonhosted.org/packages/7f/77/237a1fa124f05d5965e70325e7d085cf5bbe
  100% |████████████████████████████████████████| 112kB 6.1MB/s
Requirement already satisfied: routes in /usr/lib/python2.7/dist-packages (from -r tools/pip-req
Requirement already satisfied: six>=1.4.0 in /usr/local/lib/python2.7/dist-packages (from -r too
Collecting tinyrpc==0.9.4 (from -r tools/pip-requires (line 12))
  Downloading https://files.pythonhosted.org/packages/ba/c0/837060e3612636d4c93d3455983b9b610fb2
Requirement already satisfied: webob>=1.2 in /usr/lib/python2.7/dist-packages (from -r tools/pip
Collecting monotonic>=1.4; python_version < "3.5" (from eventlet==0.30.1->-r tools/pip-requires
  Downloading https://files.pythonhosted.org/packages/9a/67/7e8406a29b6c45be7af7740456f7f37025f0
```

```
sudo python setup.py install
```

```
root@ubuntu-KVM: /home/ryu
root@ubuntu-KVM:/home/ryu# sudo python setup.py install
running install
[pbr] Writing ChangeLog
[pbr] Generating ChangeLog
[pbr] ChangeLog complete (0.1s)
[pbr] Generating AUTHORS
[pbr] AUTHORS complete (0.0s)
running build
running build_py
running egg_info
writing requirements to ryu.egg-info/requirements.txt
writing ryu.egg-info/PKG-INFO
writing top-level names to ryu.egg-info/top_level.txt
writing dependency links to ryu.egg-info/dependency_links.txt
writing entry points to ryu.egg-info/entry_points.txt
writing pbr to ryu.egg-info/pbr.json
[pbr] Processing SOURCES.txt
[pbr] In git context, generating filelist from git
warning: no previously-included files matching '.gitreview'
warning: no previously-included files matching '*.pyc' found anywhere in distribution
reading manifest template 'MANIFEST.in'
warning: no previously-included files matching '*' found under directory 'doc/build'
warning: no previously-included files matching '*~' found anywhere in distribution
warning: no previously-included files matching '*.pyc' found anywhere in distribution
warning: no previously-included files matching '.gitignore' found anywhere in distribution
writing manifest file 'ryu.egg-info/SOURCES.txt'
running install_lib
running install_data
running install_egg_info
removing '/usr/local/lib/python2.7/dist-packages/ryu-4.34-py2.7.egg-info' (and everything under it)
Copying ryu.egg-info to /usr/local/lib/python2.7/dist-packages/ryu-4.34-py2.7.egg-info
running install_scripts
/usr/local/lib/python2.7/dist-packages/setuptools/command/easy_install.py:2088: EasyInstallDeprecationWarning: Use get_args
  warnings.warn("Use get_args", EasyInstallDeprecationWarning)
/usr/local/lib/python2.7/dist-packages/setuptools/command/easy_install.py:2090: EasyInstallDeprecationWarning: Use get_header
  header = cls.get_script_header("", executable, wininst)
Installing ryu-manager script to /usr/local/bin
Installing ryu script to /usr/local/bin
root@ubuntu-KVM:/home/ryu#
```

```
ryu-manager --version
```

```
root@ubuntu-KVM: /home/ryu
root@ubuntu-KVM:/home/ryu# ryu-manager --version
ryu-manager 4.34
root@ubuntu-KVM:/home/ryu#
```

Topology Viewer

Run mininet (or join your real environment):

```
sudo mn --controller remote --topo tree,depth=3
```

```
root@ubuntu-KVM: /home/mininet
root@ubuntu-KVM:/home/mininet# mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

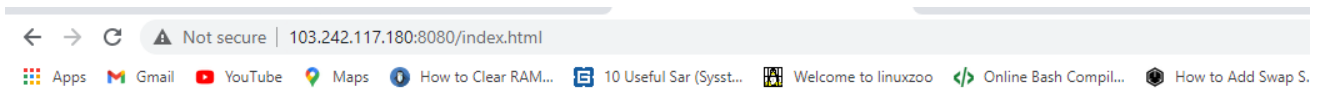
```
PYTHONPATH=. ./bin/ryu run --observe-links
ryu/app/gui_topology/gui_topology.py
```

```

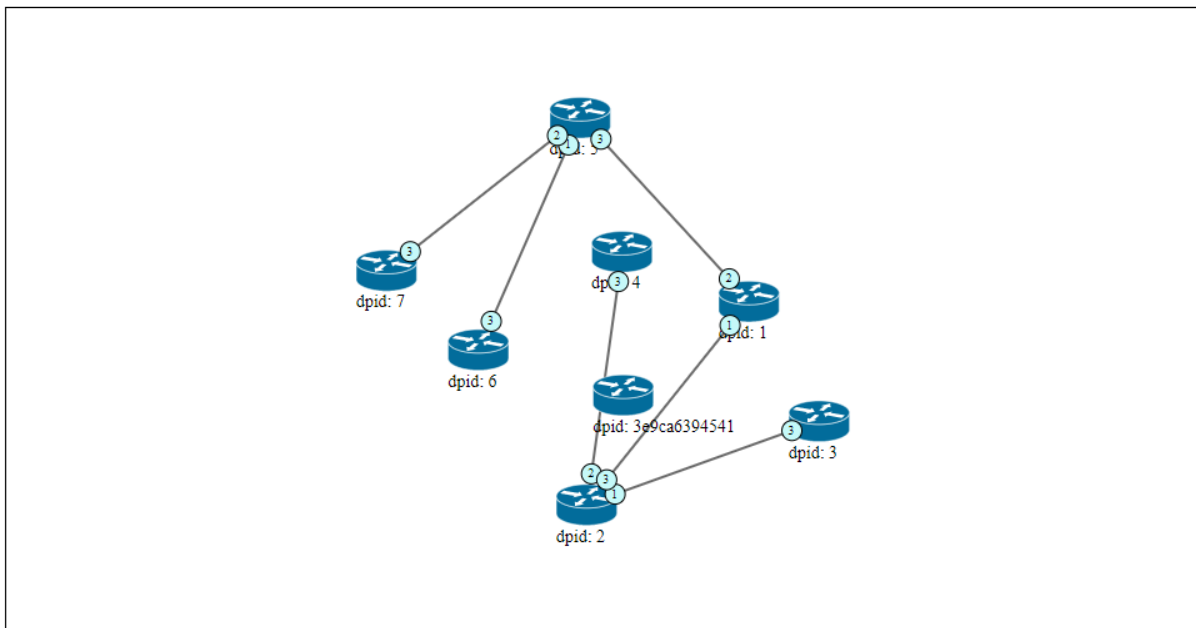
root@ubuntu-KVM: /home/ryu
root@ubuntu-KVM:/home/ryu# PYTHONPATH=. ./bin/ryu run --observe-links ryu/app/gui_topology/gui_topology.py
loading app ryu/app/gui_topology/gui_topology.py
loading app ryu.app.rest_topology
loading app ryu.app.ws_topology
loading app ryu.app.ofctl_rest
loading app ryu.controller.ofp_handler
instantiating app None of Switches
creating context switches
instantiating app None of DPSet
creating context dpset
creating context wsgi
instantiating app ryu.app.rest_topology of TopologyAPI
instantiating app ryu.app.ws_topology of WebSocketTopology
instantiating app ryu.app.ofctl_rest of RestStatsApi
instantiating app ryu.controller.ofp_handler of OFPHandler
instantiating app ryu/app/gui_topology/gui_topology.py of GUIServerApp
(16504) wsgi starting up on http://0.0.0.0:8080

```

output:



Ryu Topology Viewer



PRACTICAL NO: 07**Install floodlight on Ubuntu****Floodlight Controller:**

The Floodlight Open SDN Controller is an enterprise-class, Apache-licensed, Java-based OpenFlow Controller and intended to run with standard JDK tools and ant.

Highlights:


- Easy to set up with minimal dependencies
- Supports a broad range of virtual and physical OpenFlow switches
- Can handle mixed OpenFlow and non- OpenFlow networks.
- It can manage multiple “islands” of OpenFlow hardware switches
- Designed to be high-performance

Installation:**Prerequisites:**

- Java development kit
 - JDK 8 for Floodlight master and above
 - JDK 7 for Floodlight v1.2 and below
- Ant to build
- Python development package

- To download dependencies for Floodlight master and above:

`sudo apt-get install build-essential ant maven python-dev`

 root@ubuntu-KVM: /home

```
root@ubuntu-KVM:/home# sudo apt-get install build-essential ant maven python-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.4ubuntu1).
build-essential set to manually installed.
python-dev is already the newest version (2.7.15-rc1-1).
The following packages were automatically installed and are no longer required:
 fonts-liberation2 fonts-opensymbol girl.2-gst-plugins-base-1.0 girl.2-gstreamer-1.0 girl.2-gudev-1.0 girl.2-udisks-2
 libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1 libcdr-0.1-1 libcl
 libcmis-0.5-5v5 libcolamd2 libdazzle-1.0-0 libe-book-0.1-1 libedataserverui-1.2-2 libeot0 libepubgen-0.1-1 libetonye
 libfreerdp-client2-2 libfreerdp2-2 libgcl2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpod-common libgpod
 liblua5.3-0 libmediaart-2.0-0 libmtp-0.1-1 libodfgen-0.1-1 libqwing2v5 libraw16 librevenge-0.0-0 libsgutils2-2 li
 libwinpr2-2 libxapian3 libxmlsec1 libxmlsec1-nss lp-solve media-player-info python3-mako python3-markupsafe syslinu
 usb-creator-common
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 ant-optional libaopalliance-java libapache-pom-java libatinject-jsr330-api-java libcdi-api-java libcommons-cli-java
```

```
git clone git://github.com/floodlight/floodlight.git
```

```
root@ubuntu-KVM: /home
root@ubuntu-KVM:/home# git clone git://github.com/floodlight/floodlight.git
Cloning into 'floodlight'...
remote: Enumerating objects: 52975, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 52975 (delta 12), reused 10 (delta 10), pack-reused 52957
Receiving objects: 100% (52975/52975), 384.02 MiB | 10.59 MiB/s, done.
Resolving deltas: 100% (34324/34324), done.
Checking out files: 100% (835/835), done.
root@ubuntu-KVM:/home#
```

```
cd floodlight
```

```
git submodule init
```

```
git submodule update
```

```
root@ubuntu-KVM: /home/floodlight
root@ubuntu-KVM:/home# ls
distribution-karaf-0.6.0-Carbon    floodlight    openflow    ryu
distribution-karaf-0.6.0-Carbon.zip mininet       pox         ubuntu
root@ubuntu-KVM:/home# cd floodlight/
root@ubuntu-KVM:/home/floodlight# ls
apps                floodlight_style_settings.xml  server.crt
azure-pipelines.yml installOracleJdk8.sh          server.csr
build.xml           keystore.jks                  server.key
debian              lib                           server.keystore
docs                LICENSE.txt                   server.pl2
example             NOTICE.txt                   setup-eclipse.sh
findbugs-exclude.xml pom.xml                       src
floodlight.sh       README.md                     Vagrantfile
root@ubuntu-KVM:/home/floodlight# git submodule init
Submodule 'src/main/resources/web' (https://github.com/floodlight/floodlight-web
ui) registered for path 'src/main/resources/web'
root@ubuntu-KVM:/home/floodlight# git submodule update
Cloning into '/home/floodlight/src/main/resources/web'...
Submodule path 'src/main/resources/web': checked out '580bf06fd86bb7ff270019447f
023f9d98e431d9'
root@ubuntu-KVM:/home/floodlight#
```

ant

```
root@ubuntu-KVM: /home/floodlight
root@ubuntu-KVM:/home/floodlight# ant
Buildfile: /home/floodlight/build.xml
[taskdef] Could not load definitions from resource tasks.properties. It could
not be found.

init:
[mkdir] Created dir: /home/floodlight/target/lib
[mkdir] Created dir: /home/floodlight/target/test

compile:

compile-test:

dist:
[echo] Setting Floodlight version: 1.2-SNAPSHOT
[echo] Setting Floodlight name: floodlight
[jar] Building jar: /home/floodlight/target/floodlight.jar
[jar] Building jar: /home/floodlight/target/floodlight-test.jar

BUILD SUCCESSFUL
Total time: 1 minute 9 seconds
root@ubuntu-KVM:/home/floodlight#
```

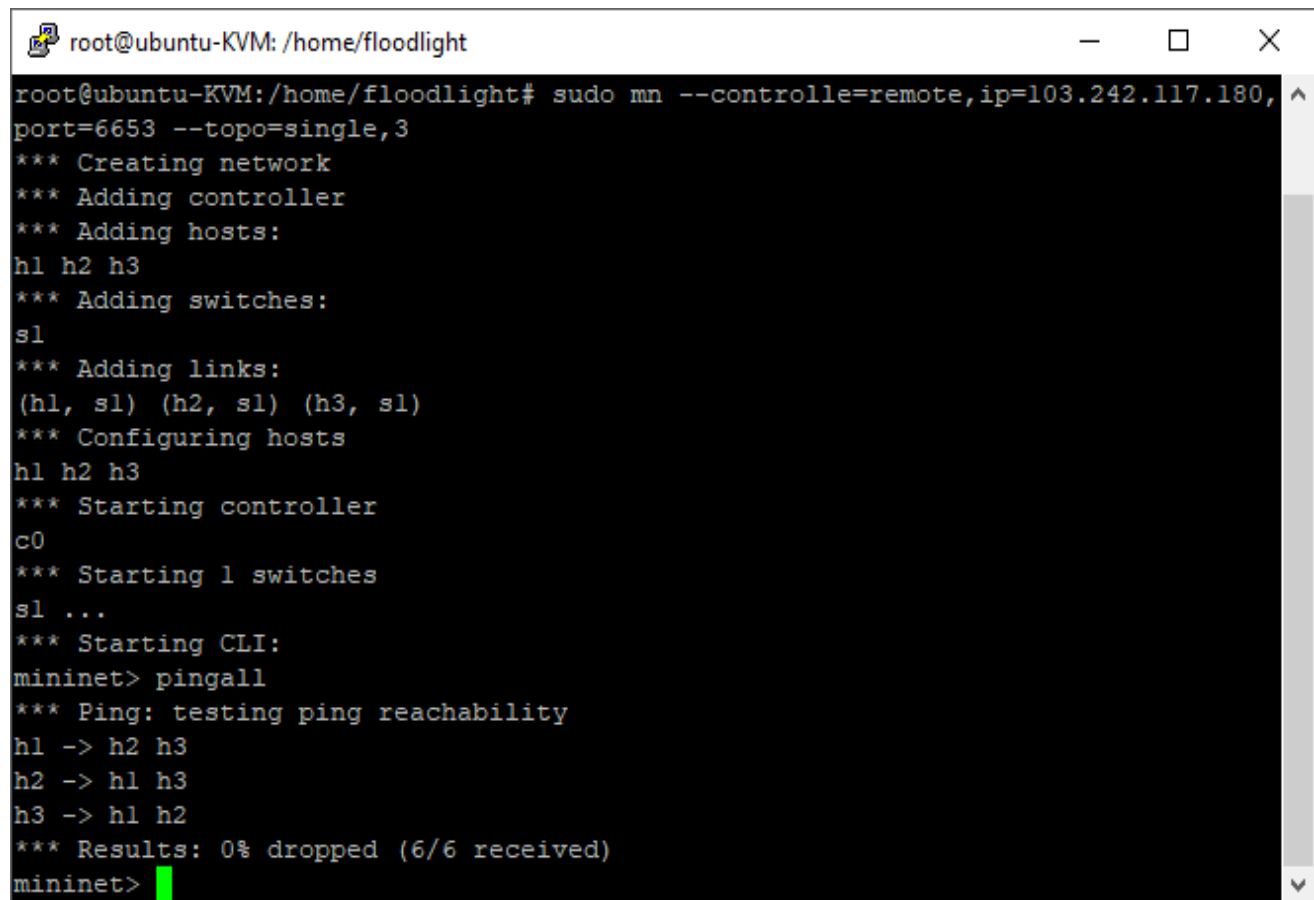
sudo mkdir /var/lib/floodlight

sudo chmod 777 /var/lib/floodlight

cd /home/floodlight/

java -jar target/floodlight.jar

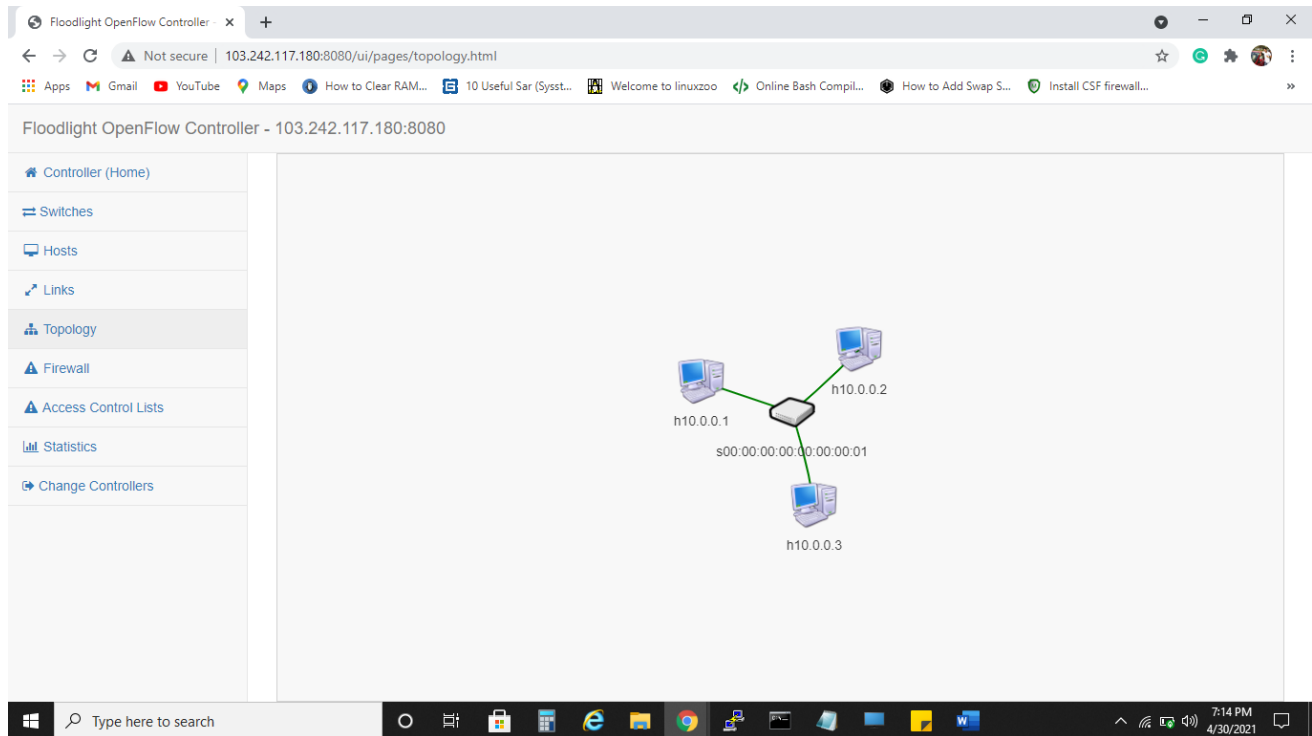
sudo mn --controlle=remote,ip=103.242.117.180,port=6653 --topo=single,3



```
root@ubuntu-KVM: /home/floodlight
root@ubuntu-KVM:/home/floodlight# sudo mn --controlle=remote,ip=103.242.117.180,
port=6653 --topo=single,3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** 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> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
mininet>
```

We created a single topology with 3 host 1 switch and one controller. This can be view in the following URL. Go to web browser and type,

<http://103.242.117.180:8080/ui/index.html>



PRACTICAL NO: 08**Install ONOS controller on Ubuntu**

Check java version and set it to java 8

Java -version

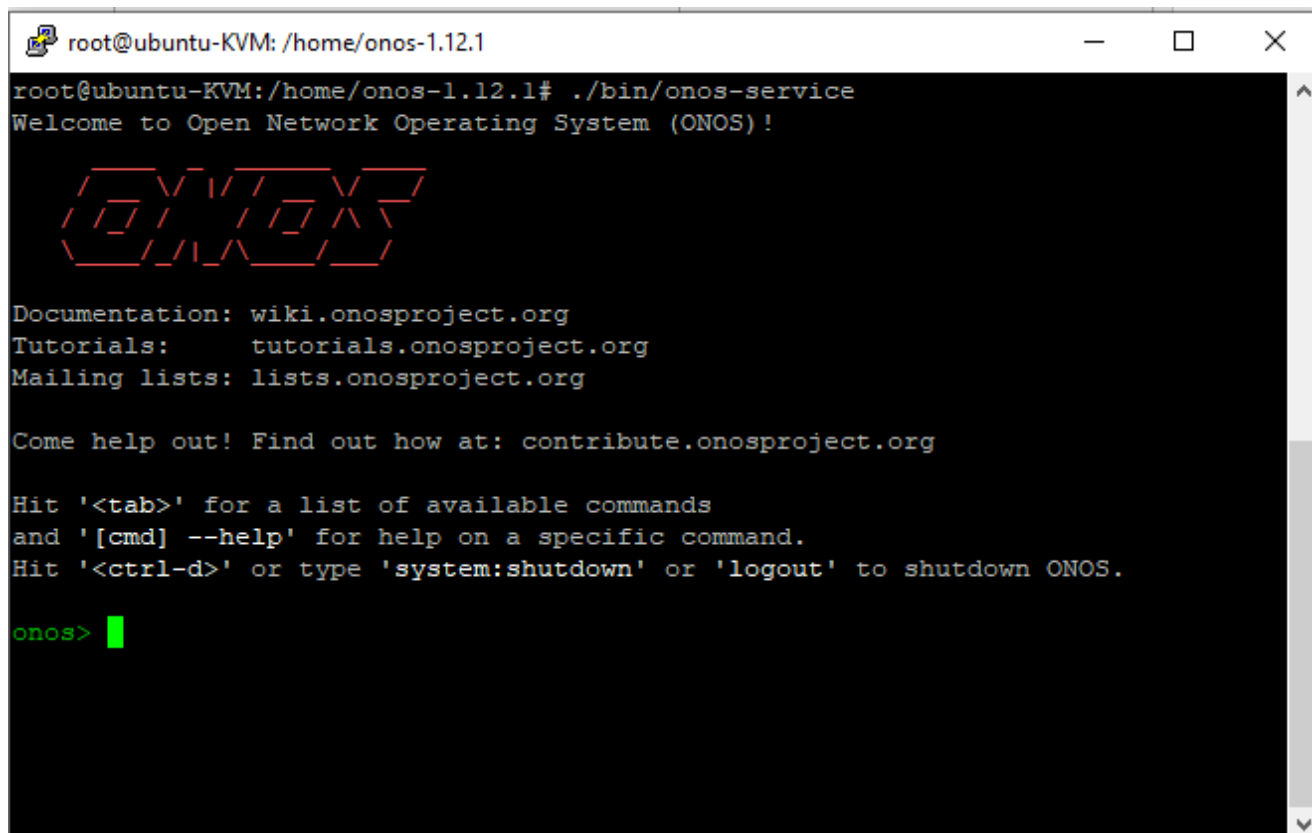
sudo apt-get install curl

Download the onos file from :

<https://wiki.onosproject.org/display/ONOS/Downloads>

wget https://repo1.maven.org/maven2/org/onosproject/onos-releases/1.12.0/onos-1.12.0.tar.gz

Run onos-service file in bin placed directory of onos

A screenshot of a terminal window titled 'root@ubuntu-KVM: /home/onos-1.12.1'. The terminal shows the command './bin/onos-service' being executed. The output includes a welcome message 'Welcome to Open Network Operating System (ONOS)!', a large red 'ONOS' logo, and links for documentation, tutorials, mailing lists, and contributions. It also provides instructions on how to use the command-line interface, such as using tab for command completion and ctrl-d to shutdown. The prompt 'onos>' is visible at the bottom.

```
root@ubuntu-KVM: /home/onos-1.12.1
root@ubuntu-KVM:/home/onos-1.12.1# ./bin/onos-service
Welcome to Open Network Operating System (ONOS)!

  ONOS

Documentation: wiki.onosproject.org
Tutorials:    tutorials.onosproject.org
Mailing lists: lists.onosproject.org

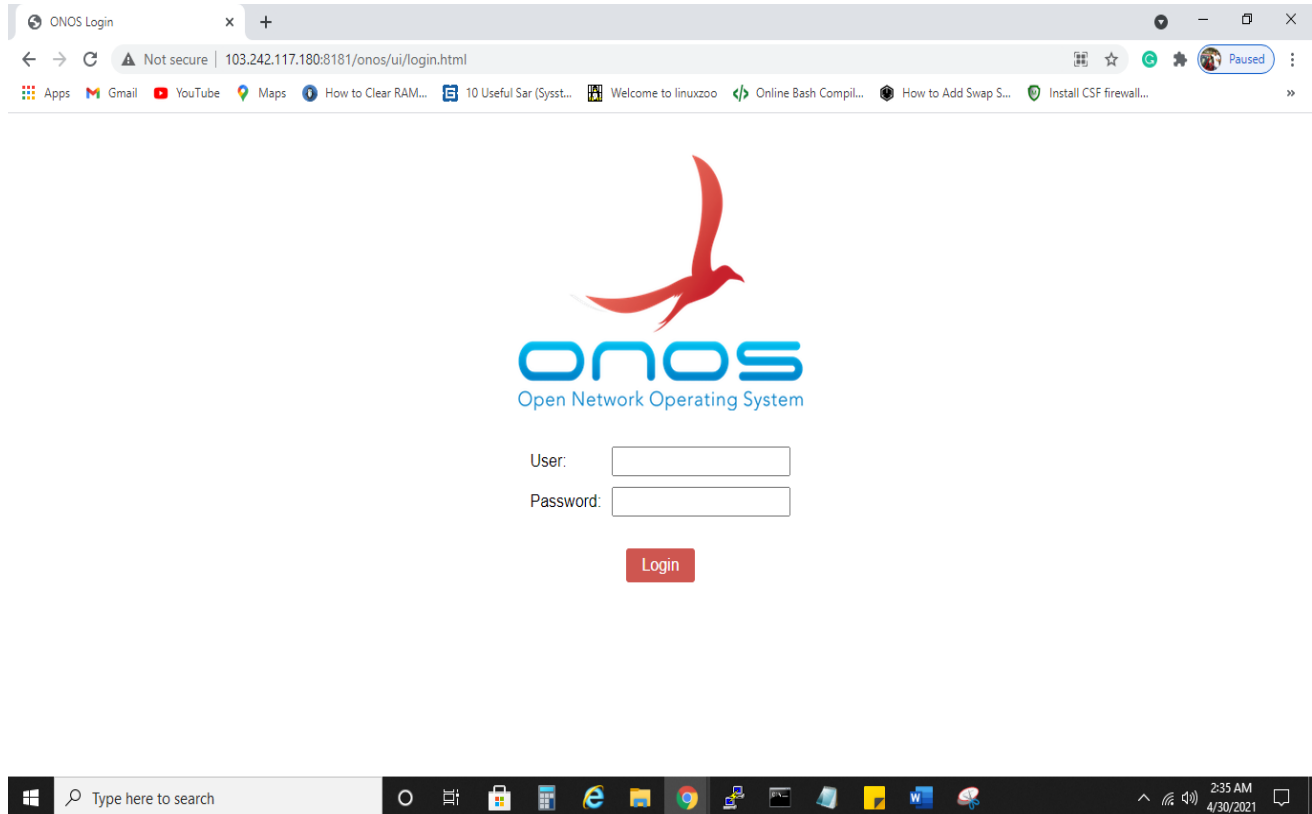
Come help out! Find out how at: contribute.onosproject.org

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

onos>
```

<http://103.242.117.180:8181/onos/ui/login.html>

Login on browser username: onos & password: rocks



Check the menu on onos console

The screenshot shows the ONOS console web interface. The browser address bar displays the URL `103.242.117.180:8181/onos/ui/index.html#/app`. The interface features a dark sidebar menu on the left with the following sections:

- PLATFORM**
 - Applications
 - Settings
 - Cluster Nodes
 - Packet Processors
 - Partitions
- NETWORK**
 - Topology
 - Topology 2
 - Devices
 - Links
 - Hosts
 - Intents
 - Tunnels

The main content area displays a table of applications. The table has columns for APP ID, VERSION, CATEGORY, and ORIGIN. The table shows 17 total applications, with the first 10 visible in the screenshot. The applications are all version 1.12.1 and originate from the ONOS Community.

APP ID	VERSION	CATEGORY	ORIGIN
org.onosproject.drivers	1.12.1	Drivers	ONOS Community
org.onosproject.acl	1.12.1	Security	ONOS Community
org.onosproject.drivers.arista	1.12.1	Drivers	ONOS Community
org.onosproject.artemis	1.12.1	Monitoring	ONOS Community
org.onosproject.bgp	1.12.1	Provider	ONOS Community
org.onosproject.bgprouter	1.12.1	Traffic Steering	ONOS Community
org.onosproject.bgpcep	1.12.1	Provider	ONOS Community
org.onosproject.drivers.bmv2	1.12.1	Drivers	ONOS Community
org.onosproject.drivers.barefoot	1.12.1	Drivers	ONOS Community
org.onosproject.drivers.optical	1.12.1	Drivers	ONOS Community
org.onosproject.pipelines.basic	1.12.1	Pipeline	ONOS Community
org.onosproject.cord-support	1.12.1	Utility	ONOS Community
org.onosproject.castor	1.12.1	Utility	ONOS Community

The Windows taskbar at the bottom shows the time as 2:37 AM on 4/30/2021.