

**RAMNIRANJAN JHUNJHUNWALA  
COLLEGE GHATKOPAR (W),  
MUMBAI - 400 086**

**DEPARTMENT OF INFORMATION  
TECHNOLOGY 2021 - 2022**

**M.Sc. ( I.T. ) SEM IV**

**Software Defined Networking**

**Name: Sneha Ramchandra Pawar**

**Roll No.: 18**



Hindi Vidya Prachar Samiti's

## RAMNIRANJAN

## JHUNJHUNWALA COLLEGE

(AUTONOMOUS)

Opposite Ghatkopar Railway Station, Ghatkopar West, Mumbai-400086



## CERTIFICATE

This is to certify that Ms. **Pawar Sneha Ramchandra** with Roll No. **18** has successfully completed the necessary course of experiments in the subject of **Software Defined Networking** during the academic year **2021 – 2022** complying with the requirements of **RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ARTS, SCIENCE AND COMMERCE**, for the course of **M.Sc. (IT)** semester -IV.

---

Internal Examiner

---

External Examiner

---

Head of Department

---

College Seal

## INDEX

<b>Practical No.</b>	<b>Topic</b>	<b>Date</b>
1.	WireShark and Bash Script application on Ubuntu	12/03/2022
2.	Theory: Hewlett-Packard Controller, HP Network Protector, HP Network Visualizer	12/03/2022
3.	Install ODL (Open DayLight Controller)	12/03/2022
4.	Implement OVS on ODL	19/03/2022
5.	Implement Mininet	19/03/2022
6.	Install RYU Controller with Mininet Topology	01/04/2022
7.	Install Floodlight Controller with Mininet Topology	01/04/2022
8.	Install ONOS Controller on Ubuntu	01/04/2022

## **Practical No. 01**

### **Aim: Wireshark and Bash Script application on Ubuntu**

## **Practical No. 01**

### **Aim: Wireshark and bash script.**

#### **What is wireshark?**

Wireshark is the world's foremost and widely-used network protocol analyzer. It lets you see what's happening on your network at a microscopic level and is the de facto (and often de jure) standard across many commercial and non-profit enterprises, government agencies, and educational institutions. Wireshark development thrives thanks to the volunteer contributions of networking experts around the globe and is the continuation of a project started by Gerald Combs in 1998.

Wireshark has a rich feature set which includes the following:

**Deep inspection of hundreds of protocols, with more being added all the time**

**Live capture and offline analysis**

**Standard three-pane packet browser**

**Multi-platform: Runs on Windows, Linux, macOS, Solaris, FreeBSD, NetBSD, and many others**

**Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility**

**The most powerful display filters in the industry**

**Rich VoIP analysis**

**Read/write many different capture file formats: tcpdump (libpcap), Pcap NG, Catapult**

**DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network General Sniffer® (compressed and uncompressed), Sniffer® Pro, and NetXray®, Network Instruments**

**Observer, NetScreen snoop, Novell LANalyzer, RADCOM WAN/LAN Analyzer,**

**Shomiti/Finisar Surveyor, Tektronix K12xx, Visual Networks Visual UpTime, WildPackets**

**EtherPeek TokenName/AiroPeek, and many others**

**Capture files compressed with gzip can be decompressed on the fly**

**Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform)**

**Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2**

**Coloring rules can be applied to the packet list for quick, intuitive analysis**

**Output can be exported to XML, PostScript®, CSV, or plain text**

#### **Steps:**

**Installing wireshark on Ubuntu.**

**Update the system. Use following command to update.**

**→ Sudo apt update**

```
sdn@sdn-VirtualBox:~$ sudo -i
[sudo] password for sdn:
root@sdn-VirtualBox:~# sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu bionic InRelease [21
.3 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu bionic/main i386 Pac
kages [3,712 B]
Get:6 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2,
170 kB]
Get:8 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu bionic/main amd64 Pa
ckages [3,720 B]
Get:9 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2
,517 kB]
Get:10 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu bionic/main Transla
tion-en [1,820 B]
Get:11 http://in.archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [1
,454 kB]
```

## → Sudo apt upgrade

```
root@sdn-VirtualBox:~# sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  efibootmgr gir1.2-geocodeglib-1.0 libegl1-mesa libfwupd liblvm8
  libwayland-egl1-mesa libwireshark11 libwiredtap8 libwscodecs2 libwsutil9
  ubuntu-web-launchers
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  distro-info fwupd-signed liblvm10 libminizip1 libnetplan0 libwireshark14
  libwiredtap11 libwsutil12 libxmlb1 linux-headers-5.4.0-107-generic
```

## → Sudo apt-get install --reinstall ca-certificates

```
root@sdn-VirtualBox:~# sudo apt-get install --reinstall ca-certificates
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  ca-certificates
1 upgraded, 0 newly installed, 0 to remove and 481 not upgraded.
Need to get 0 B/145 kB of archives.
After this operation, 13.3 kB disk space will be freed.
Preconfiguring packages ...
(Reading database ... 145272 files and directories currently installed.)
Preparing to unpack .../ca-certificates_20210119~18.04.2_all.deb... Activate Windows
Unpacking ca-certificates (20210119~18.04.2) over (20180409)... Go to Settings to activate Windows.
```

## → Sudo apt install software-properties-common -y

```
root@sdn-VirtualBox:~# sudo apt install software-properties-common -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties software-properties-gtk
The following packages will be upgraded:
  python3-software-properties software-properties-common
  software-properties-gtk
3 upgraded, 0 newly installed, 0 to remove and 478 not upgraded.
Need to get 0 B/98.7 kB of archives.
After this operation, 17.4 kB of additional disk space will be used.
(Reading database ... 145267 files and directories currently installed.)
Preparing to unpack .../software-properties-common_0.96.24.32.18_all.deb ...
Unpacking software-properties-common (0.96.24.32.18) over (0.96.24.32.9) ...
Preparing to unpack .../software-properties-gtk_0.96.24.32.18_all.deb ...
Unpacking software-properties-gtk (0.96.24.32.18) over (0.96.24.32.9) ...
Preparing to unpack .../python3-software-properties_0.96.24.32.18_all.deb ...
Unpacking python3-software-properties (0.96.24.32.18) over (0.96.24.32.9) ...
Processing triggers for mime-support (3.60ubuntu1) ... Activate Windows
Go to Settings to activate Windows.
Progress: [ 50%] [#####.....]
```

## → Sudo add-apt-repository ppa:wireshark-dev/stable

```

sdn@sdn-VirtualBox:~$ sudo -i
[sudo] password for sdn:
root@sdn-VirtualBox:~# sudo app-apr-repository ppa:wireshark-dev/stable
sudo: app-apr-repository: command not found
root@sdn-VirtualBox:~# sudo add-apr-repository ppa:wireshark-dev/stable
sudo: add-apr-repository: command not found
root@sdn-VirtualBox:~# sudo add-apt-repository ppa:wireshark-dev/stable
Latest stable Wireshark releases back-ported from Debian package versions.

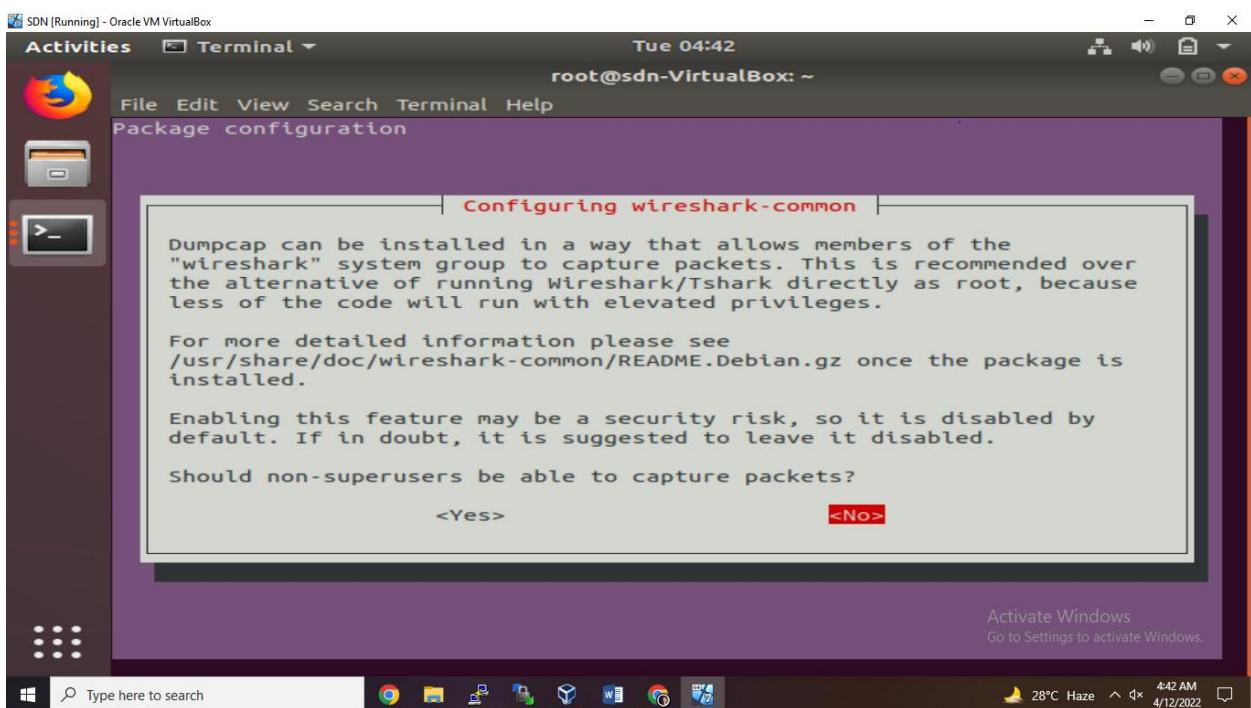
Back-porting script is available at https://github.com/rbalint/pkg-wireshark-ubuntu-ppa

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.

```

## → Sudo apt install wireshark –y



## Starting wireshark:

**Launching a Wireshark application using CLI.  
Type wireshark in CLI and press enter.**

SDN [Running] - Oracle VM VirtualBox

Activities Terminal Tue 04:46 root@sdn-VirtualBox: ~

```
File Edit View Search Terminal Help
root@sdn-VirtualBox:~# wireshark
04:45:03.542 Main Warn QStandardPaths: XDG_RUNTIME_DIR not set, defaulting
to '/tmp/runtime-root'
04:45:08.998 Warn Could not compile "of" in colorfilters file "/root/.wireshark/colorfilters".
"of" is neither a field nor a protocol name.
04:45:08.999 Warn Could not compile "Checksum Errors" in colorfilters
file "/root/.wireshark/colorfilters".
Neither "cdp.checksum_bad" nor "1" are field or protocol names.
```

Activate Windows  
Go to Settings to activate Windows.

Type here to search

SDN [Running] - Oracle VM VirtualBox

Activities Wireshark Tue 04:45

The Wireshark Network Analyzer

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

Welcome to Wireshark

Capture

...using this filter: Enter a capture filter ... All interfaces shown

enp0s3

- any
- Loopback: lo
- nflog
- nfqueue
- 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

Learn

User's Guide · Wiki · Questions and Answers · Mailing Lists

You are running Wireshark 3.4.8 (Git v3.4.8 packaged as 3.4.8-1~ubuntu18.04.0+wiresharkdevstable1).

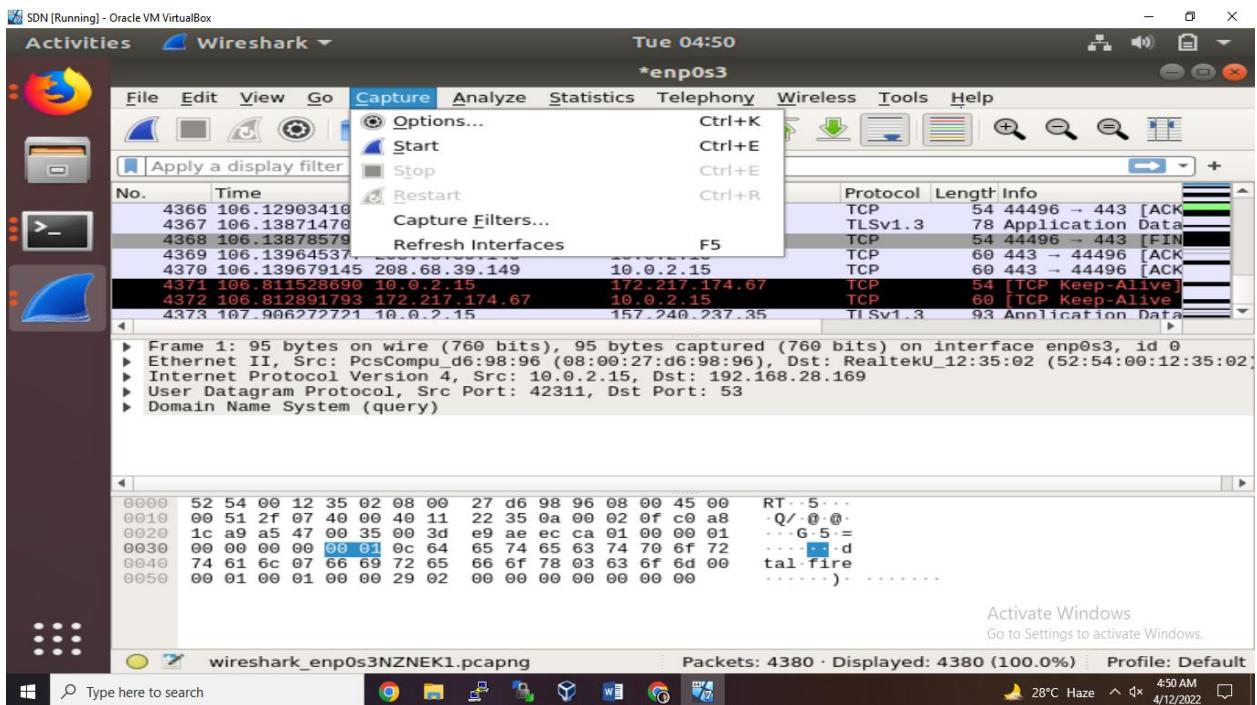
Activate Windows  
Go to Settings to activate Windows.

Type here to search

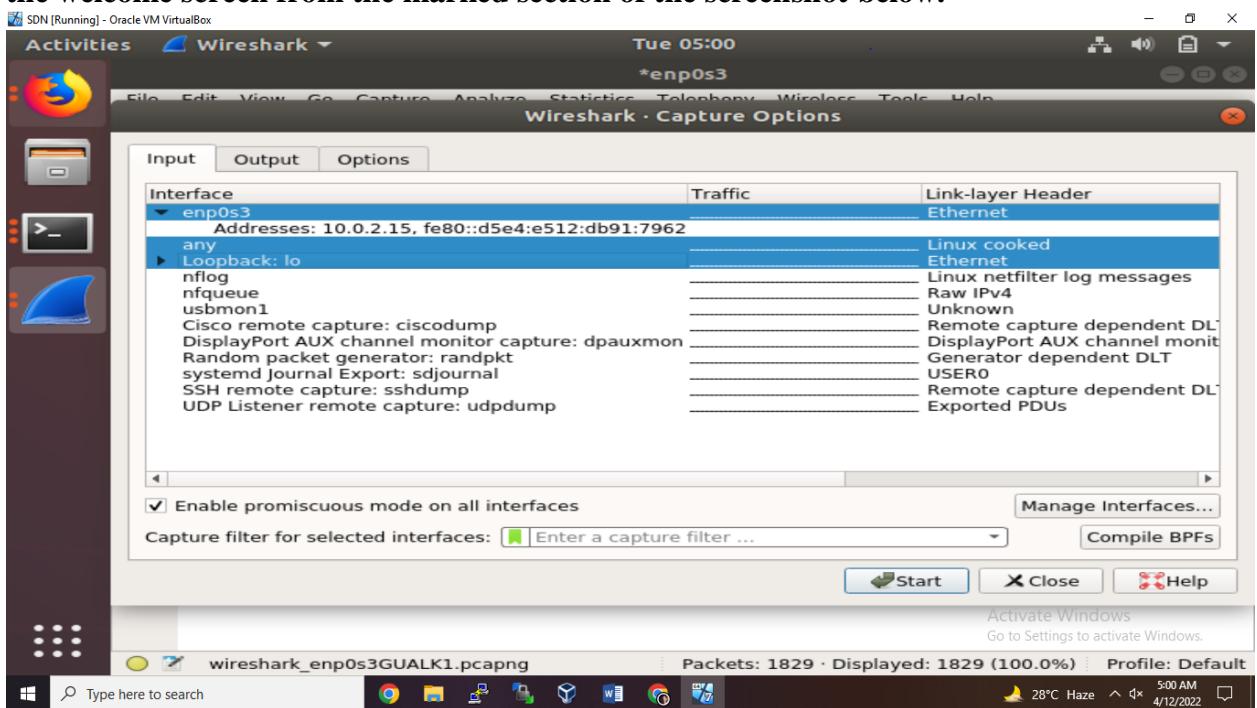
No Packets Profile: Default

28°C Haze 4:45 AM 4/12/2022

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

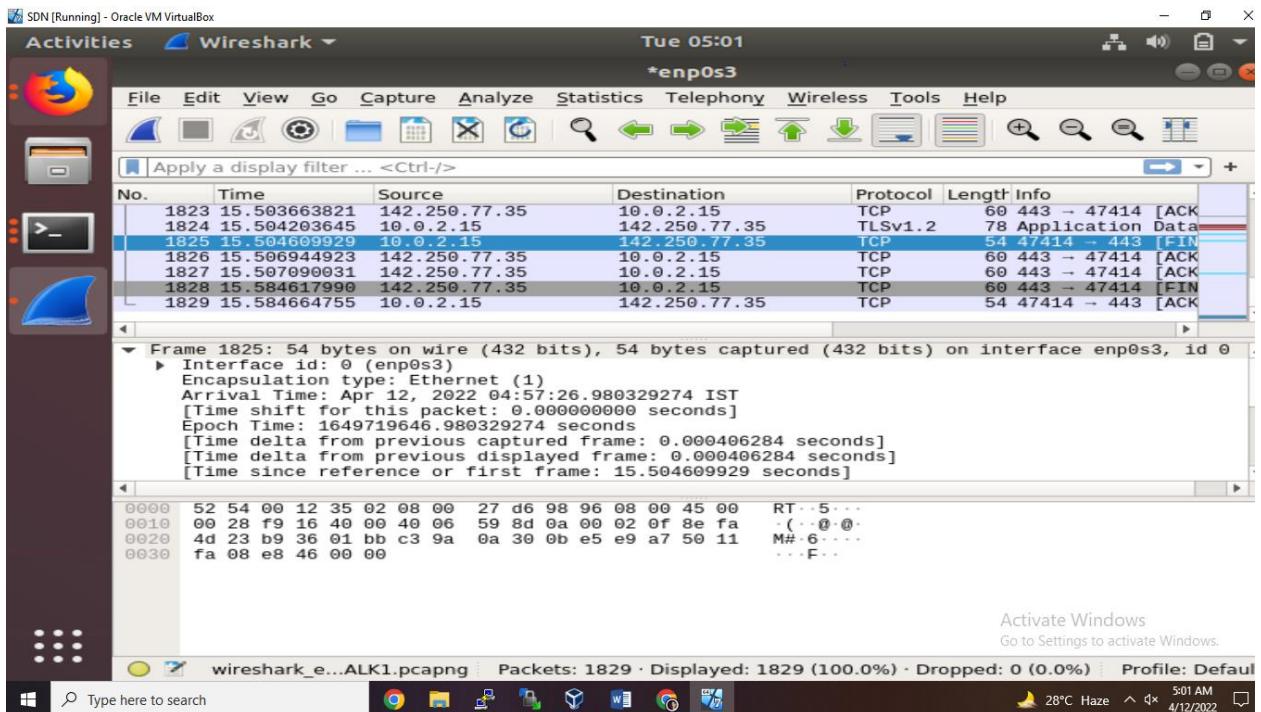


There are many type of interfaces you can monitor using wireshark, for example, wired, wireless, USB and many external devices. You can choose to show specific types of interfaces in the welcome screen from the marked section of the screenshot below.



The top screenshot shows the 'Capture Options' dialog in Wireshark. The 'Input' tab is selected, showing a list of interfaces and their types. 'enp0s3' is selected. The 'Link-layer Header' column lists various options like Ethernet, Linux cooked, and Raw IPv4. A checkbox for 'Enable promiscuous mode on all interfaces' is checked. The bottom screenshot shows the main Wireshark window with the title 'Capturing from enp0s3'. It displays a list of captured TCP keep-alive packets between two hosts. The packet details pane at the bottom shows the raw hex and ASCII data for the selected packet.

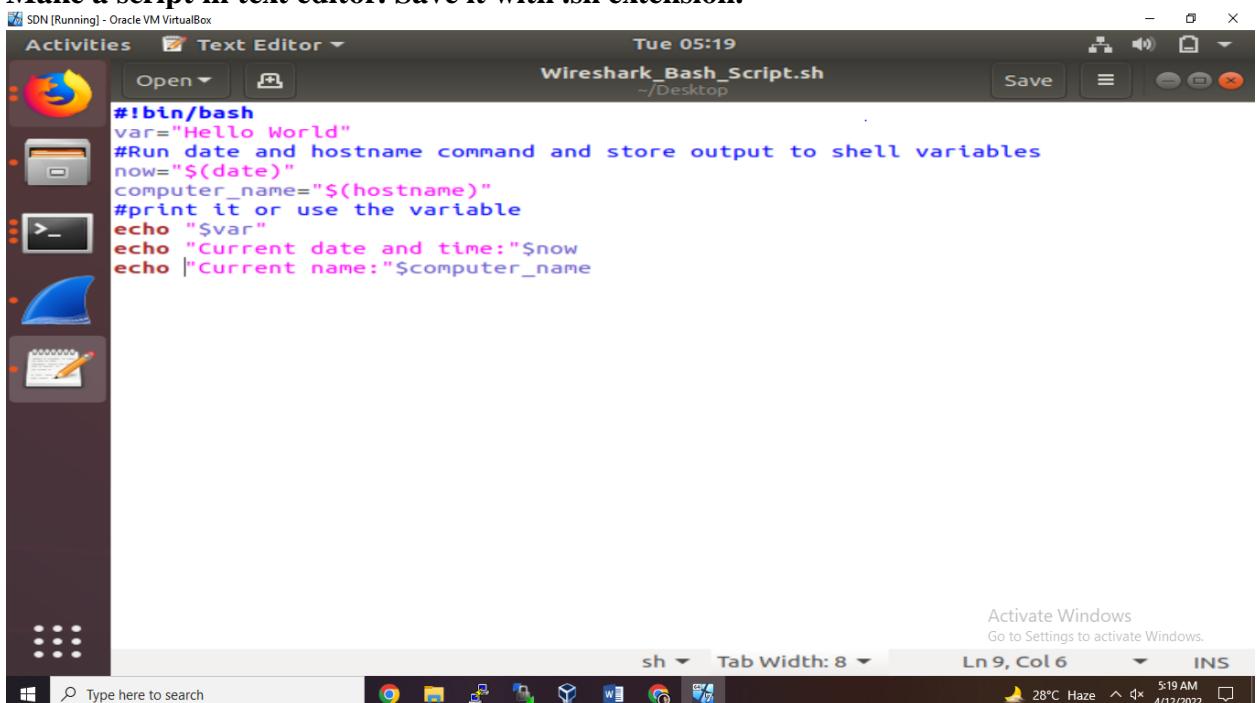
Now you can click on a packet to select it. Selecting a packet would show many information about that packet. As you can see, information about different layers of TCP/IP protocol is listed.



## Bash Script

Save the file with .sh extension.

Make a script in text editor. Save it with .sh extension.



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

`chmod 755 Wireshark_Bash_Script.sh`

```

root@sdn-VirtualBox:/home/sdn/Desktop# chmod 755 Wireshark_Bash_Script.sh
root@sdn-VirtualBox:/home/sdn/Desktop# sh Wireshark_Bash_Script.sh
Hello World
Current date and time:Tue Apr 12 05:19:35 IST 2022
Current name:sdn-VirtualBox
root@sdn-VirtualBox:/home/sdn/Desktop#

```

## **Practical No. 02**

### **Theory: HP controller, HP Network Protector, HP Network Visualizer, HPNetwork Optimizer, Cisco XNC Controller**

#### **HP Controller**

- The HPE Virtual Application Networks (VAN) SDN Controller provides a unified control point in an SDN-enabled network, simplifying management, provisioning, and orchestration.
- This enables delivery of a new generation of application-based network services. It also provides open application programming interfaces (APIs) to allow developers to create innovative solutions to dynamically link business requirements to network infrastructure via either custom Java programs or general-purpose RESTful control interfaces.
- The VAN SDN Controller is designed to operate in campus, data center, or service provider environments. The HPE VAN SDN Controller provides a unified control point in an OpenFlow-enabled network, simplifying management, provisioning, and orchestration and enabling delivery of a new generation of application-based network services.
- In the Hewlett Packard Enterprise Software Defined Networking (SDN) architecture, the control and data planes of the network are decoupled from each other, centralizing network intelligence and abstracting the underlying network infrastructure from applications. Controller software directly provisions physical and virtual switches under its control via the industry-standard OpenFlow protocol. Network ports, links, and topologies are all directly visible, enabling centralized policy administration and more effective path selection based on a dynamic, global view of the network.
- This dramatically simplifies the orchestration of multi-tenant environments and the enforcement of network policy for both mobile clients and servers. The HPE VAN SDN Controller is designed to operate in a variety of computing environments, including campus, data center, service provider, private cloud, and public cloud.

#### **The HP SDN controller delivers:**

Open programmable interfaces. Our SDN controller delivers tight integration between the network and business system, with open,programmable interfaces that enable the orchestration of applications and automation of network functions.

Your developers can use the language of their choice and leverage the extensible RESTful API for the creation of SDN applications that will unleash new levels of innovation. Centralized, resilient control. Our controller provides centralized, resilient control of the SDN network, including functions such as discovery of the network topology and shortest path forwarding through the network.

Highly available and scalable. We designed our SDN controller to meet high availability and scaling requirements through a scale-out teaming model. Our SDN controller can be clustered, so that if any one controller in the network fails, another in the cluster will take over.

Robust security. Security is an important factor of our controller. It uses robust authentication and authorization methods so that SDN application scan interact with the controller while preventing unauthorized applications from gaining network access. The southbound connections between the OpenFlow switches and the HP SDN controller are also secured and encrypted.

Full Integration with HP Intelligent Management Center (IMC). Administrators can use the HP Intelligent Management Center (IMC) SDN Manager for full fault, configuration, accounting, performance, and security management for the HP SDN controller and OpenFlow infrastructure.

HP SDN Manager leverages flow monitoring, topology mapping, and troubleshooting to provide full SDN management through the same interface as the wired, physical, and virtual network. IMC provides full controller application life cycle management and monitoring, enhanced reporting, and SDN network visualization.

## **Product features**

### **Proactive flow processing :**

Enables highly scalable, centrally orchestrated SDN networks

### **Reactive flow processing :**

Enables dynamic monitoring of new flows or endpoints

### **Graphical user interface (GUI) :**

Facilitates controller administration and API documentation

## **Northbound APIs :**

Leverage the controller's extensible RESTful HTTPS interface; provide an abstract representation of the underlying OpenFlow network and allow external applications running above the controller to exert deliberative, business-level control over the network. Provide the services necessary to support a full management platform such as HPE Intelligent Management Center (IMC)

## **Native APIs :**

Allow Java applications to run within the controller as a collection of OSGi bundles that enable high performance event and packet processing

## **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

## **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

Aim: Install ODL (OpenDayLight) controller on Ubuntu.

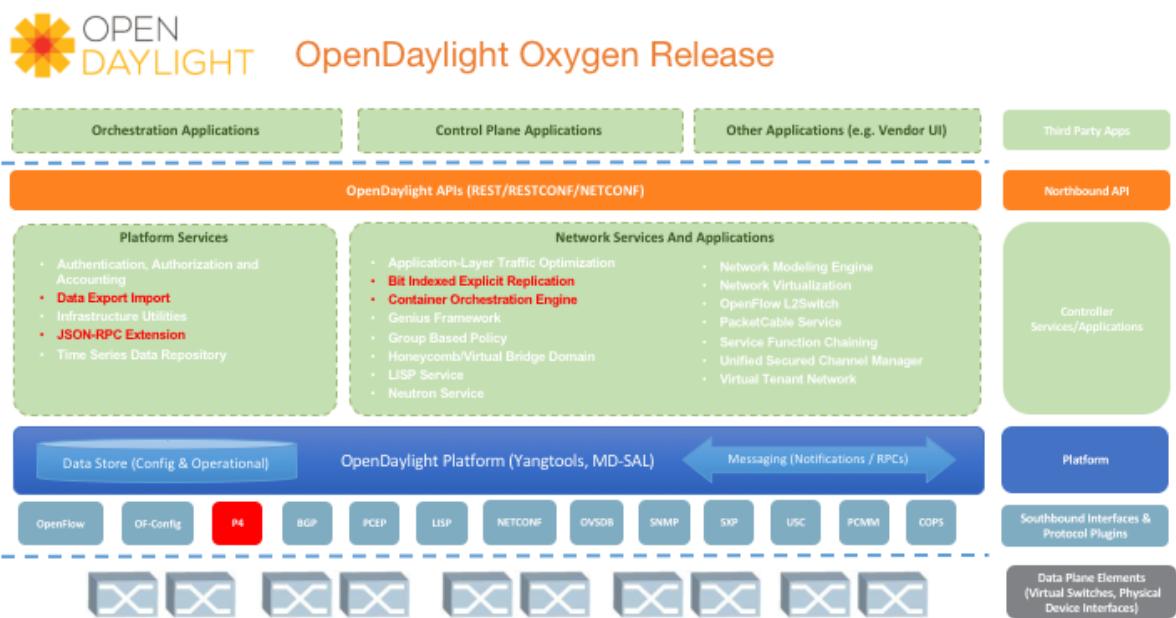
### What is ODL?

**OpenDaylight** is a modular open platform for customising and automating networks of any size and scale. The OpenDaylight Project arose out of the SDN movement, with a clear focus on network programmability. It was designed from the outset as a foundation for commercial solutions that address a variety of use cases in existing network environments.

### Architecture

ODL consists of 3 layers:

- Southbound plugins to communicate with the network devices
- Core Services that can be used by means of Service Abstraction Layer (SAL) which is based on OSGi to help components going in and out of the controller while the controller is running
- Northbound interfaces (e.g. REST/NETCONF) that allow operators to apply high-level policies to network devices or integration of ODL with other platforms



### Steps:

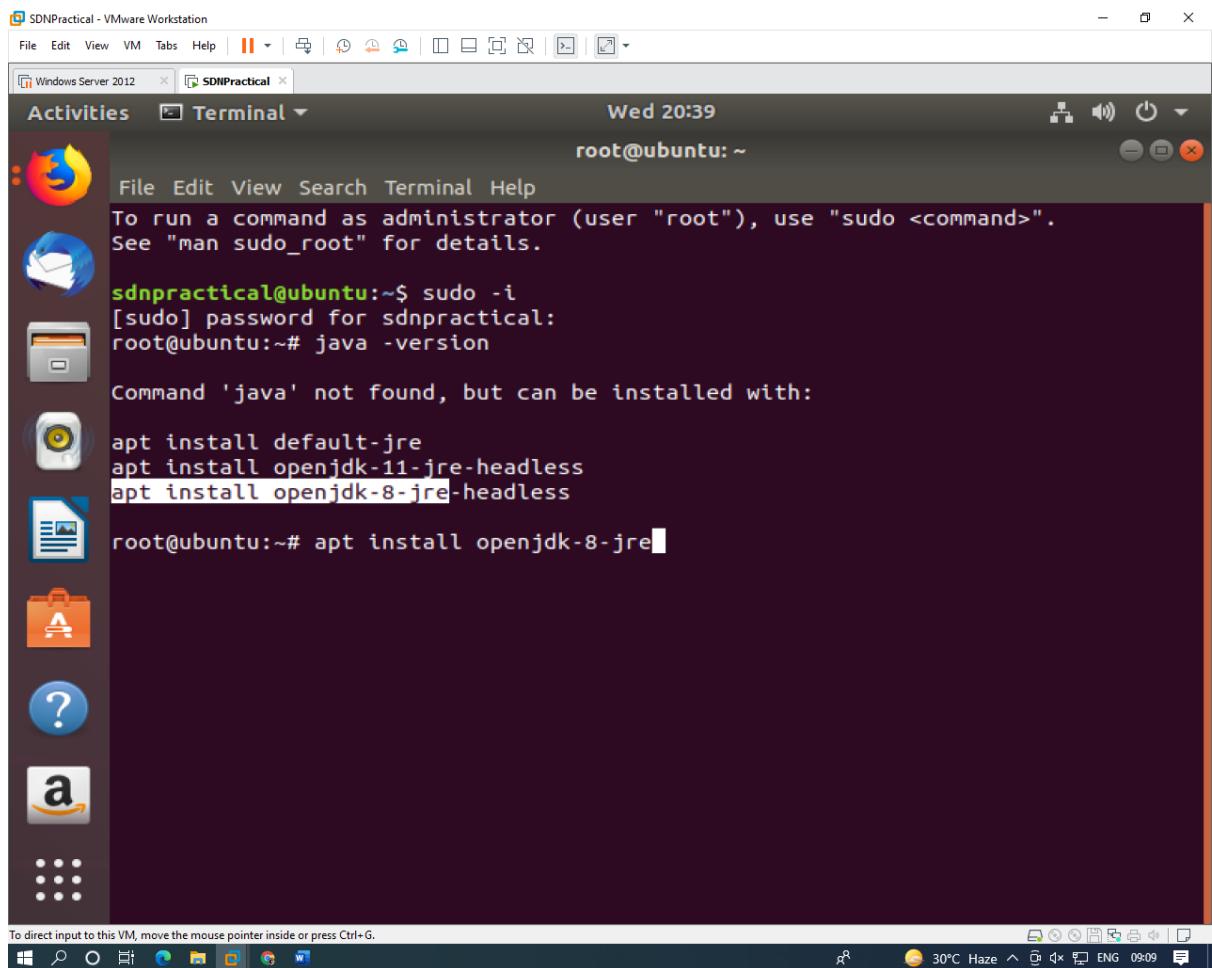
Open the terminal. Switch to the root user.

→ sudo -i

Enter the password for the root user.

Install the java.

→ apt install openjdk-8-jre



Activities Terminal ▾ Wed 20:41  
root@ubuntu:~

```
sdnpractical@ubuntu:~$ sudo -i
[sudo] password for sdnpractical:
root@ubuntu:~# java -version
Command 'java' not found, but can be installed with:

apt install default-jre
apt install openjdk-11-jre-headless
apt install openjdk-8-jre-headless

root@ubuntu:~# apt install openjdk-8-jre
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
    libatk-wrapper-java-jni libgif7 openjdk-8-jre-headless
Suggested packages:
  default-jre icedtea-8-plugin fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
    libatk-wrapper-java-jni libgif7 openjdk-8-jre openjdk-8-jre-headless
0 upgraded, 8 newly installed, 0 to remove and 497 not upgraded.
Need to get 30.4 MB of archives.
After this operation, 112 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

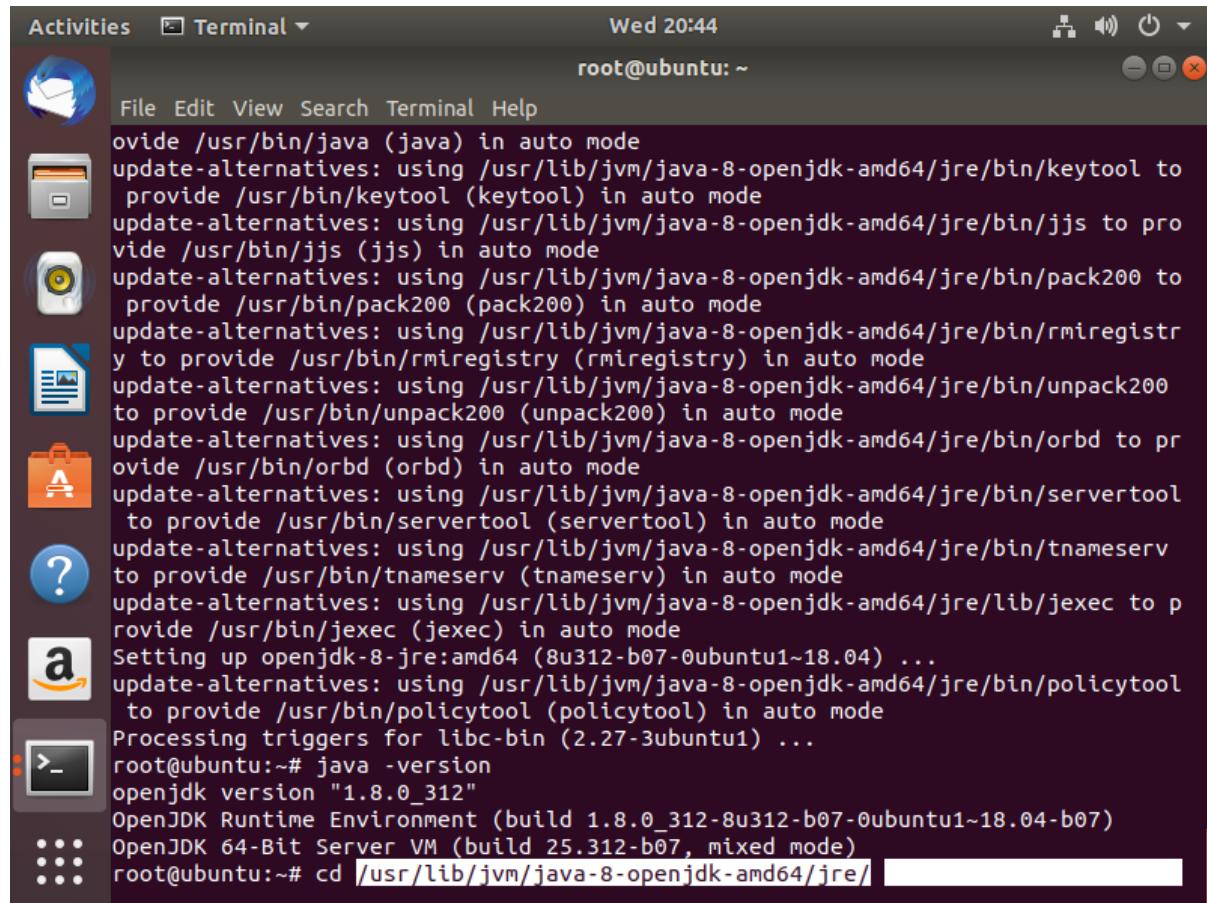
Activities Terminal ▾ Wed 20:42  
root@ubuntu:~

```
File Edit View Search Terminal Help
Provide /usr/bin/java (java) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/keytool to
  provide /usr/bin/keytool (keytool) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/jjs to pro
  vide /usr/bin/jjs (jjs) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/pack200 to
  provide /usr/bin/pack200 (pack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/rmiregistr
  y to provide /usr/bin/rmiregistry (rmiregistry) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/unpack200
  to provide /usr/bin/unpack200 (unpack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/orbd to pr
  ovide /usr/bin/orbd (orbd) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/serv
  ertool to provide /usr/bin/serv
  ertool (serv
  ertool) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/tnameserv
  to provide /usr/bin/tnameserv (tnameserv) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/jexec to p
  rovide /usr/bin/jexec (jexec) in auto mode
Setting up openjdk-8-jre:amd64 (8u312-b07-0ubuntu1~18.04) ...
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/policytool
  to provide /usr/bin/policytool (policytool) in auto mode
Processing triggers for libc-bin (2.27-3ubuntu1) ...
root@ubuntu:~# java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1~18.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
root@ubuntu:~#
```

**Java will get installed on the following path:**

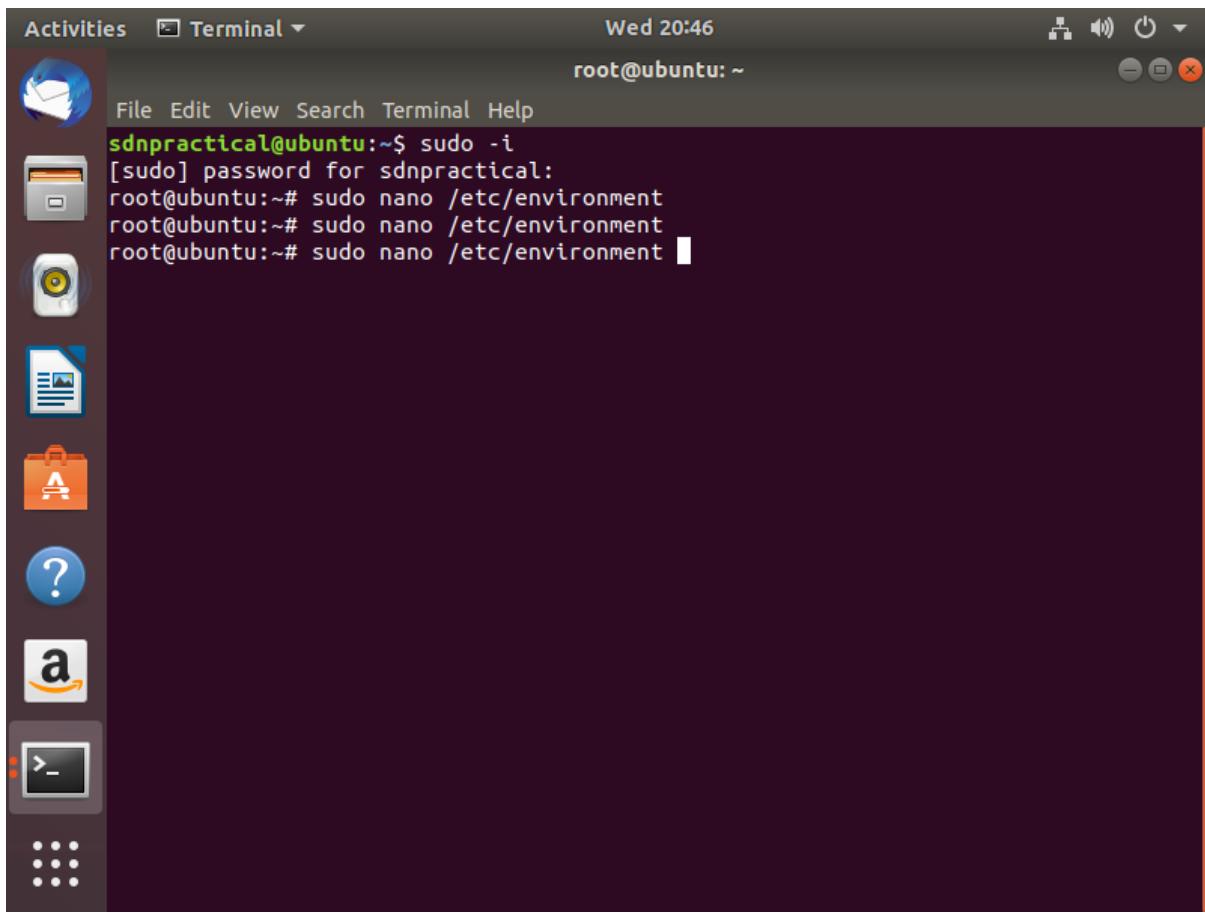
**/usr/lib/jvm/java-8-openjdk-amd64/jre/**

**Copy this path and set JAVA\_HOME path in /etc/environment file.**

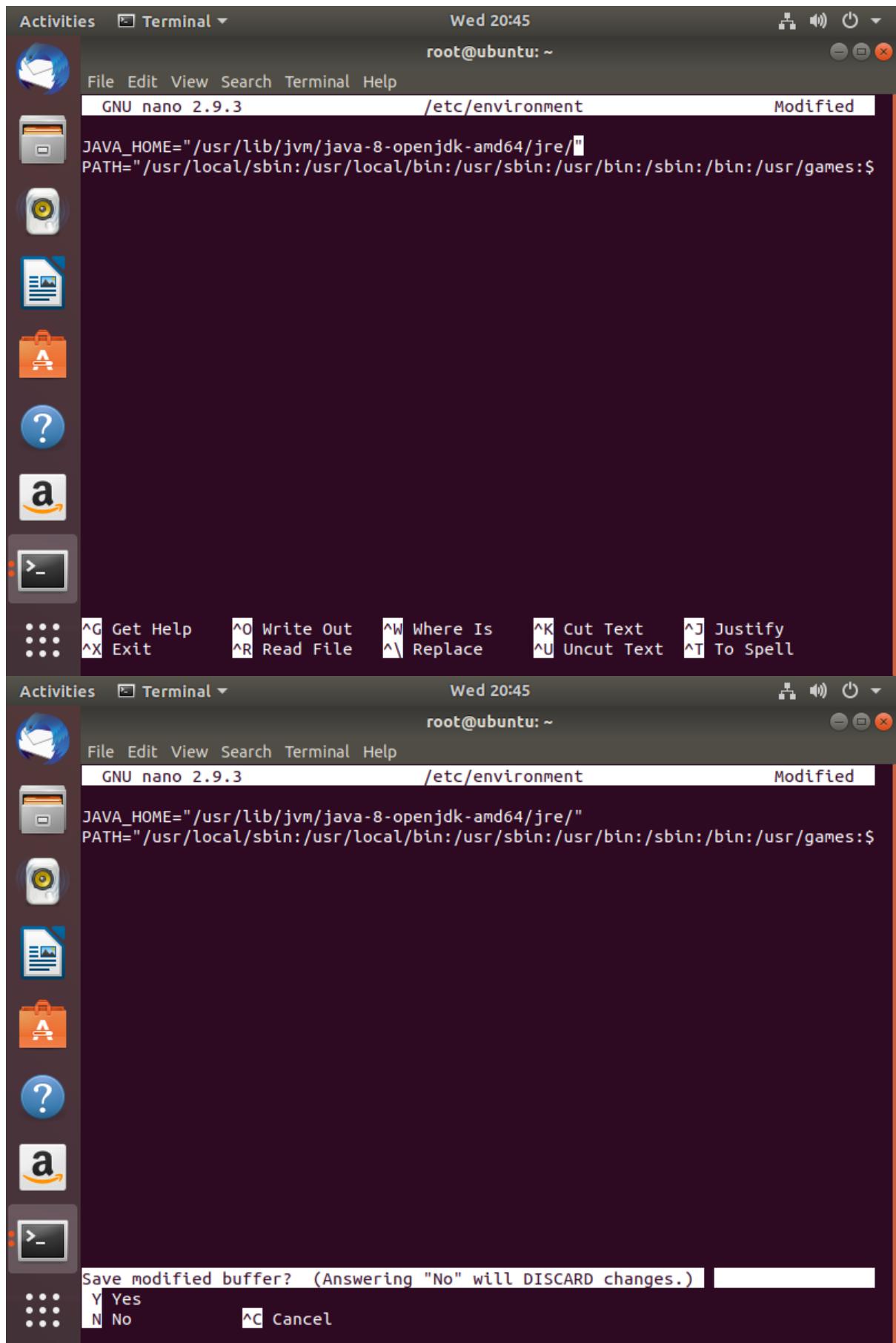


A screenshot of an Ubuntu desktop environment. On the left is the Unity Dash icon bar with various application icons. In the center is a terminal window titled "Terminal". The terminal shows the root user's command-line session. The user has run several commands to install Java 8, including "update-alternatives" and "java -version". The output indicates the Java 8 OpenJDK runtime environment is now installed and active. The terminal window has a dark theme and is located at the top of the screen above the Dash.

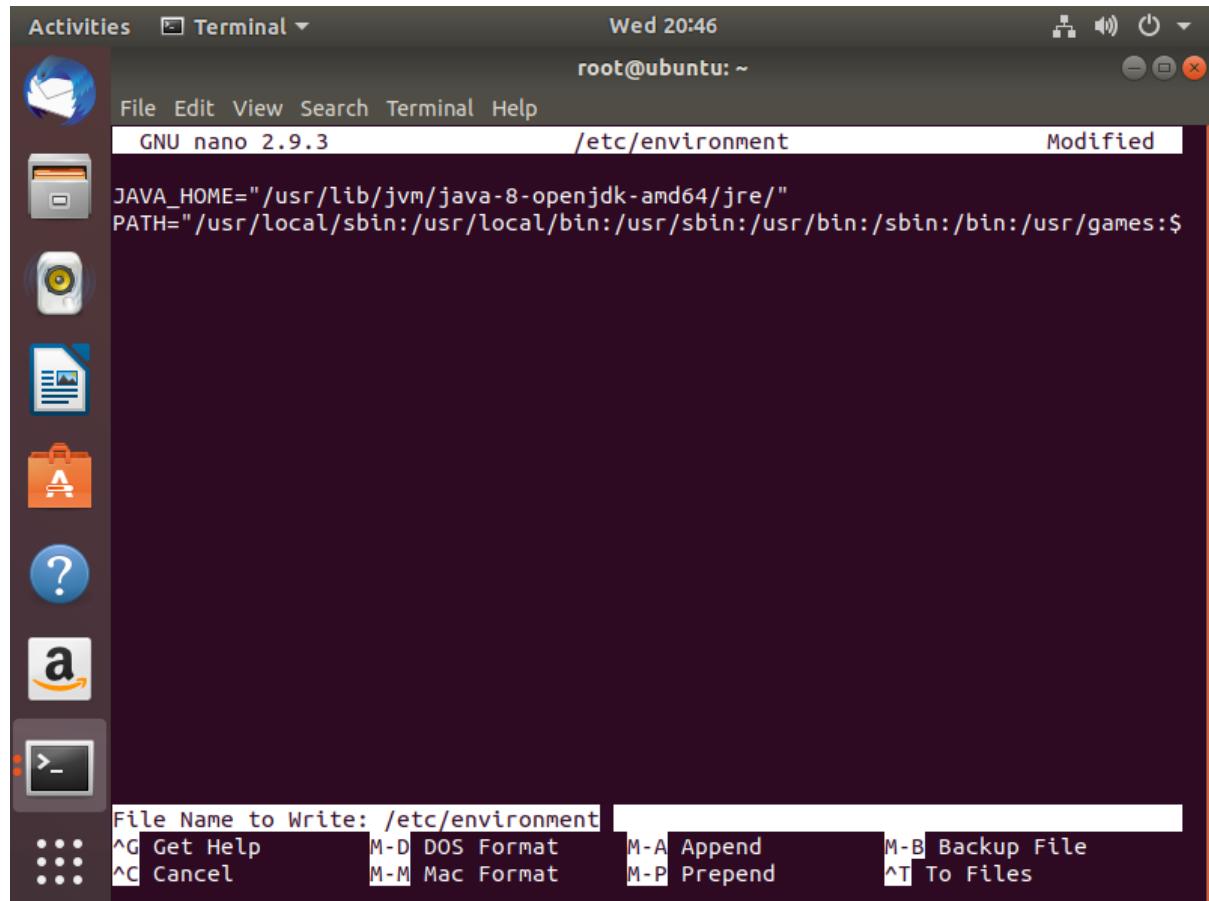
```
File Edit View Search Terminal Help
provide /usr/bin/java (java) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/keytool to
    provide /usr/bin/keytool (keytool) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/jjs to pro
    vide /usr/bin/jjs (jjs) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/pack200 to
    provide /usr/bin/pack200 (pack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/rmiregistr
    y to provide /usr/bin/rmiregistry (rmiregistry) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/unpack200
    to provide /usr/bin/unpack200 (unpack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/orbd to pr
    ovide /usr/bin/orbd (orbd) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/servertool
    to provide /usr/bin/servertool (servertool) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/tnameserv
    to provide /usr/bin/tnameserv (tnameserv) in auto mode
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/jexec to p
    rovide /usr/bin/jexec (jexec) in auto mode
Setting up openjdk-8-jre:amd64 (8u312-b07-0ubuntu1~18.04) ...
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/policytool
    to provide /usr/bin/policytool (policytool) in auto mode
Processing triggers for libc-bin (2.27-3ubuntu1) ...
root@ubuntu:~# java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1~18.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
root@ubuntu:~# cd /usr/lib/jvm/java-8-openjdk-amd64/jre/
```



**Press CTRL+X & type Y.**

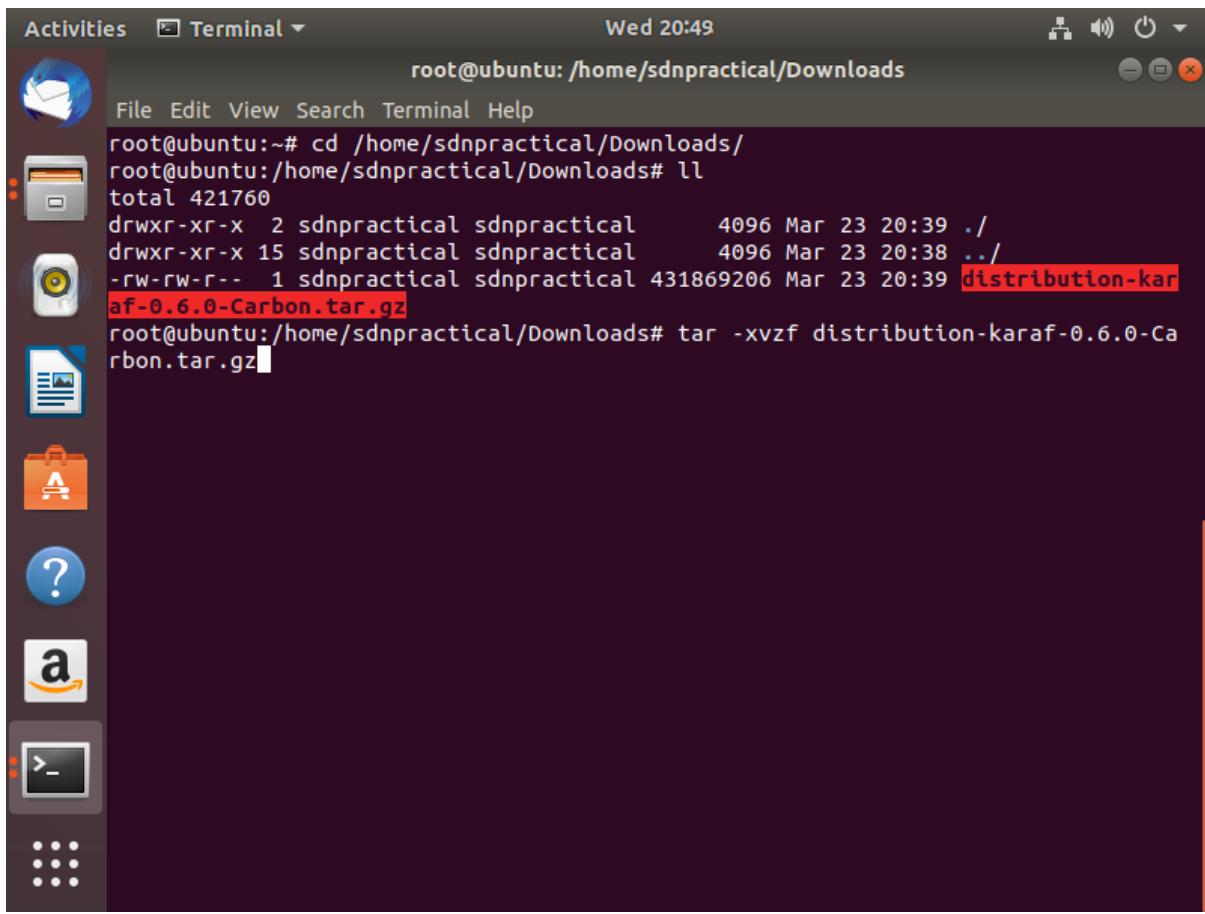


**Press Enter.**



Go to the path where you have downloaded the ODL controller and extract the file by directly right clicking on it or using the following command on the terminal.

→ **tar -xvzf distribution-karaf-0.6.0-Carbon.tar.gz**

A screenshot of an Ubuntu desktop environment. On the left is a vertical dock with icons for Activities, Terminal, Dash, Home, Applications, Help, and a terminal icon. The main window is a terminal window titled "Terminal" with the command line "root@ubuntu: /home/sdnpractical/Downloads". The terminal shows the following commands and output:

```
root@ubuntu:~# cd /home/sdnpractical/Downloads/
root@ubuntu:/home/sdnpractical/Downloads# ll
total 421760
drwxr-xr-x  2 sdnpractical sdnpractical      4096 Mar 23 20:39 .
drwxr-xr-x 15 sdnpractical sdnpractical      4096 Mar 23 20:38 ../
-rw-rw-r--  1 sdnpractical sdnpractical 431869206 Mar 23 20:39 distribution-karaf-0.6.0-Carbon.tar.gz
root@ubuntu:/home/sdnpractical/Downloads# tar -xvzf distribution-karaf-0.6.0-Carbon.tar.gz
```

**Then execute the karaf script file which is present in the bin folder of extracted file of ODL controller.**

```
Activities Terminal Wed 20:51
root@ubuntu: /home/sdnpractical/Downloads/distribution-karaf-0.6.0-Carbon/bin
File Edit View Search Terminal Help
xml
distribution-karaf-0.6.0-Carbon/taglist.log
distribution-karaf-0.6.0-Carbon/bin/
distribution-karaf-0.6.0-Carbon/bin/aaa-cli-jar.jar
distribution-karaf-0.6.0-Carbon/bin/client
distribution-karaf-0.6.0-Carbon/bin/configure-cluster-ipdetect.sh
distribution-karaf-0.6.0-Carbon/bin/configure_cluster.sh
distribution-karaf-0.6.0-Carbon/bin/custom_shard_config.txt
distribution-karaf-0.6.0-Carbon/bin/instance
distribution-karaf-0.6.0-Carbon/bin/karaf
distribution-karaf-0.6.0-Carbon/bin/set_persistence.sh
distribution-karaf-0.6.0-Carbon/bin/setenv
distribution-karaf-0.6.0-Carbon/bin/shell
distribution-karaf-0.6.0-Carbon/bin/start
distribution-karaf-0.6.0-Carbon/bin/status
distribution-karaf-0.6.0-Carbon/bin/stop
distribution-karaf-0.6.0-Carbon/bin/client.bat
distribution-karaf-0.6.0-Carbon/bin/instance.bat
distribution-karaf-0.6.0-Carbon/bin/karaf.bat
distribution-karaf-0.6.0-Carbon/bin/setenv.bat
distribution-karaf-0.6.0-Carbon/bin/shell.bat
distribution-karaf-0.6.0-Carbon/bin/start.bat
distribution-karaf-0.6.0-Carbon/bin/status.bat
distribution-karaf-0.6.0-Carbon/bin/stop.bat
root@ubuntu:/home/sdnpractical/Downloads# cd distribution-karaf-0.6.0-Carbon/bi
n/
root@ubuntu:/home/sdnpractical/Downloads/distribution-karaf-0.6.0-Carbon/bin# .
/karaf
```

After that OpenDayLight terminal will open.

```
Activities Terminal Wed 20:52
root@ubuntu: /home/sdnpractical/Downloads/distribution-karaf-0.6.0-Carbon/bin
File Edit View Search Terminal Help
at org.eclipse.osgi.framework.eventmgr.EventManager$EventThread.run(EventManager.java:340)
100% [=====]
Karaf started in 4s. Bundle stats: 64 active, 64 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>
```

**Install the below features:**

- ➔ feature:install odl-dlux-core
- ➔ feature:install odl-dluxapps-nodes
- ➔ feature:install odl-dluxapps-topology
- ➔ feature:install odl-dluxapps-yangui
- ➔ feature:install odl-dluxapps-yangvisualizer
- ➔ feature:install odl-dluxapps-yangman

```
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
```

- ➔ feature:install odl-restconf odl-l2switch-switch odl-mdsal-apidocs
- ➔ feature:install odl-l2switch-all
- ➔ feature:install odl-vtn-manager-neutron odl-neutron-service odl-neutron-hostconfig-ovs
- ➔ feature:install odl-ovsdb-library odl-restconf-all odl-ovsdb-southbound-api odl-ovsdb-southbound-impl odl-ovsdb-southbound-impl-rest
- ➔ feature:install odl-netvirt-openstack

```
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
Refreshing bundles org.eclipse.persistence.core (156)
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
opendaylight-user@root>
```

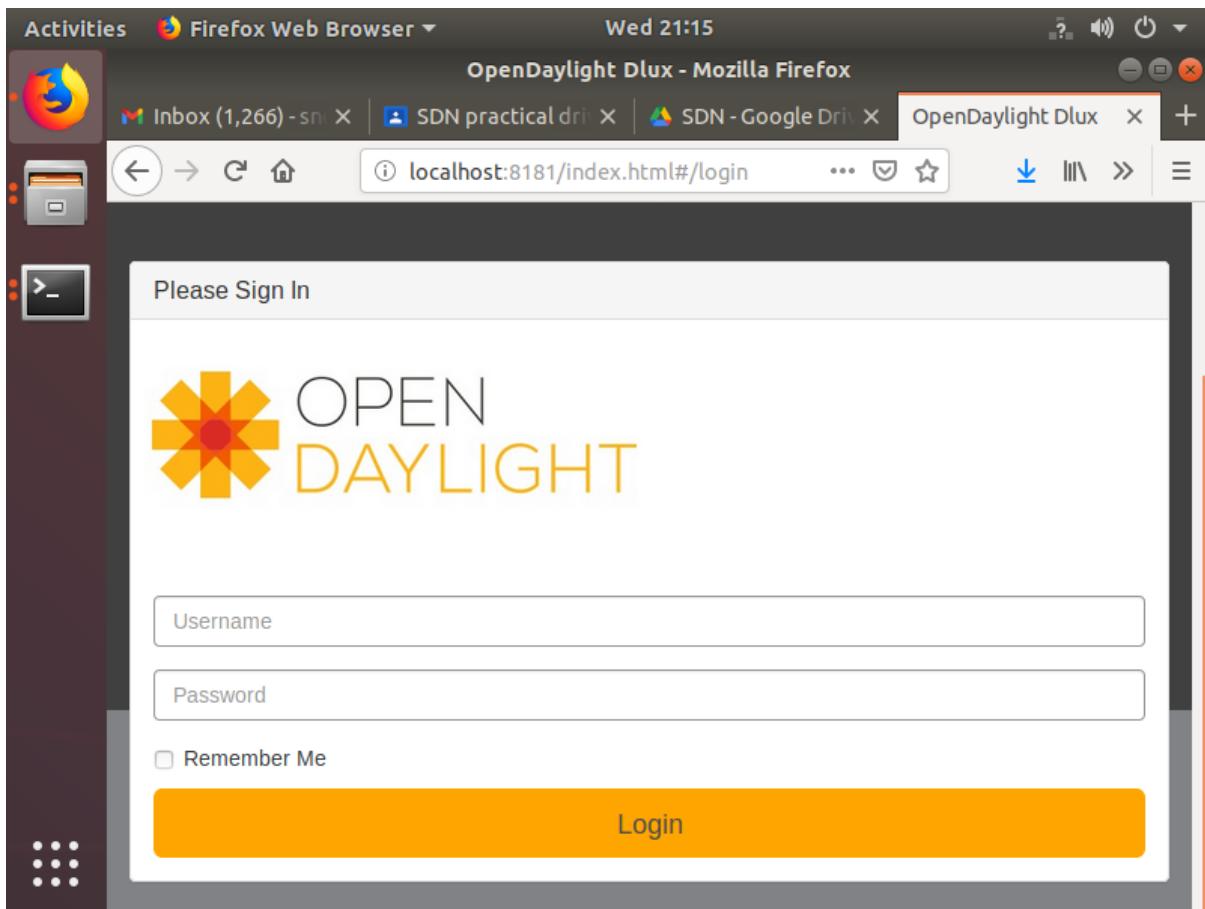
**After installing this features the Open Daylight dashboard will be accessible to us at localhost.**

**We can access it by using the following URL.**

**<http://localhost:8181/index.html#/login>**

**Username: admin**

**Password : admin**



Activities Firefox Web Browser ▾ Wed 21:17

OpenDaylight Dlux - Mozilla Firefox

Inbox (1,266) X SDN practical X SDN - Google Dr X OpenDaylight Dlux X New Tab X +

localhost:8181/index.html 67%

Yangman

Logout (admin)

Nodes

Topology

Yang UI

Yang Visualizer

Yangman

MODULES HISTORY COLLECTIONS

aaa rev.2016-12-14

aaa-cert rev.2015-11-26

aaa-cert-mdsal rev.2016-03-21

aaa-cert-rpc rev.2015-12-15

aaa-encrypt-service-config rev.2016-09-15

acl-live-statistics rev.2016-11-29

aclservice-config rev.2016-08-06

address-tracker-config rev.2016-06-21

aliveness-monitor rev.2016-04-11

arp-handler-config rev.2014-05-28

Received data

1

## Practical No. 04

### Aim: Implement OVS (Open vSwitch) on ODL

This practical is based on OpenFlow and OVSDB connection between SDN controller and Openflow based switches and connection lifecycle.

We will use open source SDN controller i.e. Opendaylight (ODL) Carbon release and Openflow virtual switch i.e. OpenVswitch (OVS) version 2.6.0 to describe about OVSDB and Openflow connection.

SDN controller manages underlying Openflow switches through two types of connection:

- OVSDB
- Openflow

#### OVSDB and Openflow connection

OVSDB is OVS database management protocol which defines schema for OVS database and specification for communication between controller and switch. SDN controller uses this protocol to control OVS switch by populating its database with the intended configuration.

Typically, SDN controller runs in passive mode to listen for OVSDB and Openflow connections from OVS on ports 6640 and 6653 respectively.

So first, let us start ODL controller, you can download it from here. It is a prerequisite to set JAVA\_HOME environment variable in your system to latest Java version. Running ODL is so easy, just do below steps:

**Download ODL distribution from here.**

**Unzip the ODL distribution.**

Enter the ‘bin’ sub-directory located under the top-level directory of the downloaded package.

Run the following command to get your controller up: karaf

Below is the image for above steps performed:

- ☒ Cd /home/snehapawar/Downloads/distribution-karaf-0.6.0-Carbon/bin
- ☒ ./karaf

```
root@snehapawar-VirtualBox:/# cd /home/snehapawar/Downloads/distribution-karaf-0.6.0-Carbon/bin
root@snehapawar-VirtualBox:/home/snehapawar/Downloads/distribution-karaf-0.6.0-Carbon/bin# ./karaf
```



```

opendaylight-user@root>feature:list -i | grep openflow
odl-openflowplugin-nxm-extensions           | 0.4.0-Carbon   | x  |
[openflowplugin-extension-0.4.0-Carbon]    | OpenDaylight :: Openflow Plugin :: N
icira Extension
odl-openflowplugin-nxm-extensions           | 0.4.0-Carbon   | x  |
[openflowplugin-extension-0.4.0-Carbon]    | OpenDaylight :: Openflow Plugin :: N
icira Extension
odl-openflowplugin-flow-services            | 0.4.0-Carbon   | x  |
[openflowplugin-0.4.0-Carbon]              | OpenDaylight :: Openflow Plugin :: F
low Services
odl-openflowplugin-southbound               | 0.4.0-Carbon   | x  |
[openflowplugin-0.4.0-Carbon]              | OpenDaylight :: Openflow Plugin :: L
i southbound A
odl-openflowplugin-nsf-model                | 0.4.0-Carbon   | x  |
[openflowplugin-0.4.0-Carbon]              | OpenDaylight :: OpenflowPlugin :: NS
F :: Model
odl-openflowplugin-app-config-pusher        | 0.4.0-Carbon   | x  |
[openflowplugin-0.4.0-Carbon]              | OpenDaylight :: Openflow Plugin :: A
pplication - d
odl-openflowplugin-app-topology             | 0.4.0-Carbon   | x  |
[openflowplugin-0.4.0-Carbon]              | OpenDaylight :: Openflow Plugin :: A
pplication - t
odl-openflowplugin-app-forwardingrules-manager | 0.4.0-Carbon | x  |
[openflowplugin-0.4.0-Carbon]              | OpenDaylight :: Openflow Plugin :: A
pplication - F
odl-openflowjava-protocol                  | 0.9.0-Carbon   | x  |
[odl-openflowjava-0.9.0-Carbon]            | OpenDaylight :: Openflow Java :: Pro
tocol
opendaylight-user@root>

```

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

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

```

snehapawar@snehapawar-VirtualBox:~$ sudo -i
[sudo] password for snehapawar:
root@snehapawar-VirtualBox:~# netstat -a | grep 6653
tcp6      0      0 [::]:6653          [::]:*                      LISTEN
root@snehapawar-VirtualBox:~# netstat -a | grep 6640
tcp6      0      0 [::]:6640          [::]:*                      LISTEN
root@snehapawar-VirtualBox:~#

```

It is noted that OVS switch should already be running before executing any OVS command. To run OVS on Ubuntu, run the following command with root permissions:

It is available as service, so you can check its status as well.

```

root@snehapawar-VirtualBox:~# sudo /etc/init.d/openvswitch-switch start
Starting openvswitch-switch (via systemctl): openvswitch-switch.service.
root@snehapawar-VirtualBox:~# sudo /etc/init.d/openvswitch-switch status
● openvswitch-switch.service - Open vSwitch
   Loaded: loaded (/lib/systemd/system/openvswitch-switch.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sun 2022-03-20 18:43:01 IST; 24min ago
     Main PID: 3374 (code=exited, status=0/SUCCESS)
       Tasks: 0 (limit: 5839)
      Memory: 0B
     CGroup: /system.slice/openvswitch-switch.service

Mar 20 18:43:01 snehapawar-VirtualBox systemd[1]: Starting Open vSwitch...
Mar 20 18:43:01 snehapawar-VirtualBox systemd[1]: Finished Open vSwitch.
root@snehapawar-VirtualBox:~#

```

Once, OVS is running, you can execute its CLI commands also. For example,

Below is the image for above command execution output:

- ip r
- ovs-vsctl set-manager tcp:10.0.2.15:6640
- sudo ovs-vsctl show

```
root@snehapawar-VirtualBox:~# sudo ovs-vsctl show
adba7ab1-ac54-431c-b8cc-47eb6cd8eeae
    ovs_version: "2.13.3"
root@snehapawar-VirtualBox:~#
```

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

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@snehapawar-VirtualBox:~# ip r
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
root@snehapawar-VirtualBox:~# ovs-vsctl set-manager tcp:<controller-IP>:6640
-bash: controller-IP: No such file or directory
root@snehapawar-VirtualBox:~# ovs-vsctl set-manager tcp:<10.0.2.15>:6640
-bash: 10.0.2.15: No such file or directory
root@snehapawar-VirtualBox:~# ovs-vsctl set-manager tcp:10.0.2.15:6640
root@snehapawar-VirtualBox:~# sudo ovd-vdctl show
sudo: ovd-vdctl: command not found
root@snehapawar-VirtualBox:~# sudo ovs-vsctl show
adba7ab1-ac54-431c-b8cc-47eb6cd8eeae
    Manager "tcp:10.0.2.15:6640"
        is_connected: true
    Bridge br-int
        Controller "tcp:10.0.2.15:6653"
            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: "2.13.3"
root@snehapawar-VirtualBox:~#
```

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 OVSDB 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 br-int tcp:10.0.2.15:6653
- sudo ovs-vsctl show

```
root@snehapawar-VirtualBox:~# ovs-vsctl set-controller br-int  tcp:10.0.2.15:6653
root@snehapawar-VirtualBox:~# sudo ovs-vsctl show
adba7ab1-ac54-431c-b8cc-47eb6cd8eeae
    Manager "tcp:10.0.2.15:6653"
    Bridge br-int
        Controller "tcp:10.0.2.15:6653"
            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: "2.13.3"
root@snehapawar-VirtualBox:~#
```

As we have seen from OVS side that OVSDB and Openflow connections have been

made and same can be verified at ODL side by running the following commands. Check below image for the output of the commands which confirms connection ESTABLISHED state:

☒ apt install net-tools

```
root@snehapawar-VirtualBox:~# apt install net-tools

root@snehapawar-VirtualBox:~# apt install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libvba-wayland2
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 383 not upgraded.
Need to get 196 kB of archives.
After this operation, 864 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+git20180626.aebd88e-1ubuntu1 [196 kB]
Fetched 196 kB in 0s (1,093 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 184188 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb .
..
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
root@snehapawar-VirtualBox:~#
```

**netstat -a | grep 6653**

**netstat -a | grep 6640**

```
root@snehapawar-VirtualBox:~# netstat -a | grep 6653
tcp      0      0 snehapawar-Virtua:53648 snehapawar-Virtual:6653 TIME_WAIT
tcp      0      0 snehapawar-Virtua:53656 snehapawar-Virtual:6653 TIME_WAIT
tcp      0      0 snehapawar-Virtua:53566 snehapawar-Virtual:6653 ESTABLISHED
tcp      0      0 snehapawar-Virtua:53662 snehapawar-Virtual:6653 TIME_WAIT
tcp      0      0 snehapawar-Virtua:53654 snehapawar-Virtual:6653 TIME_WAIT
tcp      0      0 snehapawar-Virtua:53650 snehapawar-Virtual:6653 TIME_WAIT
tcp      0      0 snehapawar-Virtua:53658 snehapawar-Virtual:6653 TIME_WAIT
tcp      0      0 snehapawar-Virtua:53652 snehapawar-Virtual:6653 TIME_WAIT
tcp      0      0 snehapawar-Virtua:53660 snehapawar-Virtual:6653 TIME_WAIT
tcp6     0      0 [::]:6653          [::]:*              LISTEN
tcp6     0      0 snehapawar-Virtual:6653 snehapawar-Virtua:53566 ESTABLISHED
root@snehapawar-VirtualBox:~# netstat -a | grep 6640
tcp6     0      0 [::]:6640          [::]:*              LISTEN
root@snehapawar-VirtualBox:~#
```

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

Check below image for the output of the command:

☒ sudo ovs-ofctl show br-int -OOpenFlow13

```
root@snehapawar-VirtualBox:~# sudo ovs-ofctl show br-int -OOpenFlow13
OFPT_FEATURES_REPLY (OF1.3) (xid=0x2): dpid:00009e72e2029a49
n_tables:254, n_buffers:0
capabilities: FLOW_STATS TABLE_STATS PORT_STATS GROUP_STATS QUEUE_STATS
OFPST_PORT_DESC reply (OF1.3) (xid=0x3):
  LOCAL(br-int): addr:9e:72:e2:02:9a:49
    config:      PORT_DOWN
    state:       LINK_DOWN
    speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (OF1.3) (xid=0x9): frags=normal miss_send_len=0
root@snehapawar-VirtualBox:~#
```

Here, you can see FEATURE\_REPLY shows the datapath ID (in hex format) of the bridge on OVS.

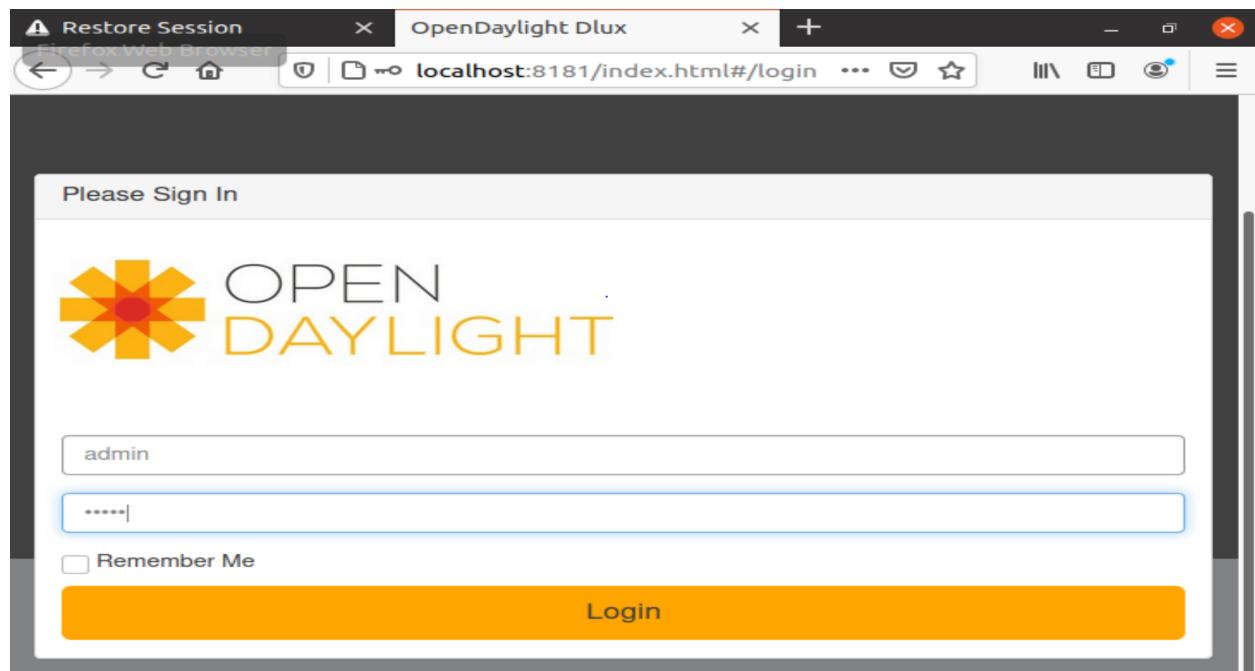
Even ODL has dashboard DLUX feature which gives GUI web interface to ODL controller at url <http://localhost:8181/index.html> on which ODL user can log in (by default username and password is ‘admin’) and check network topology on the browser of the system where ODL is running. Check below images for same.

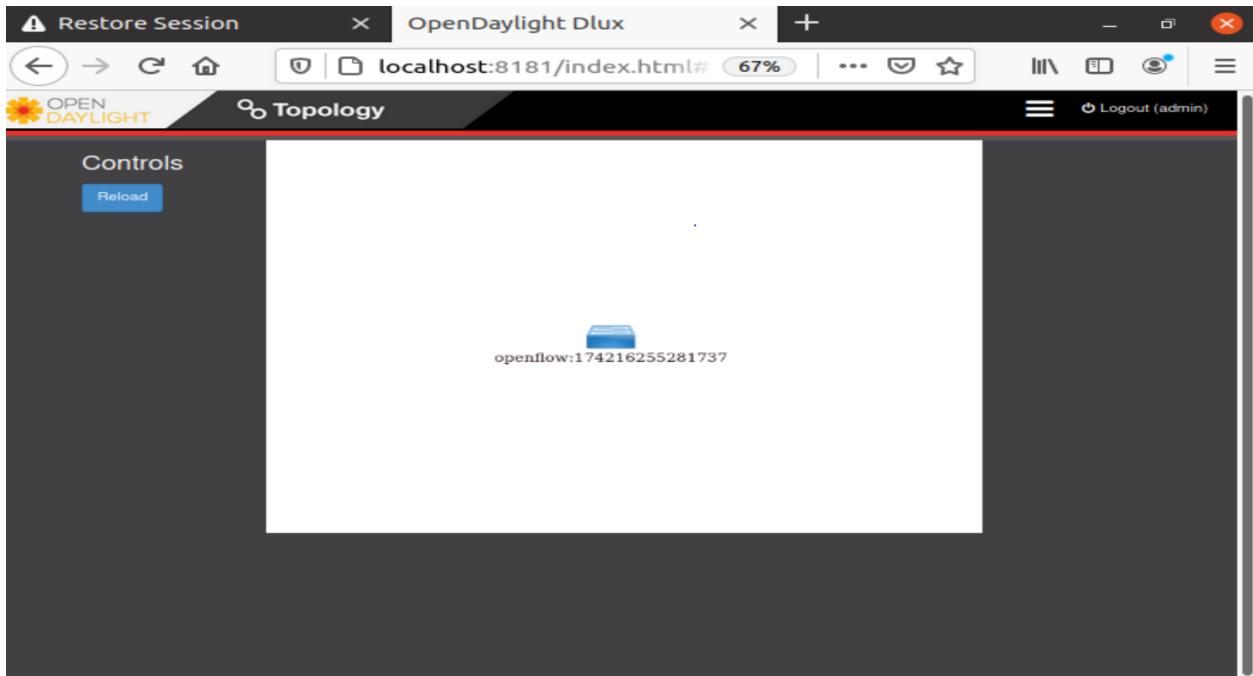
<http://localhost:8181/index.html#/login>

Username: admin

Password:admin

<http://localhost:8181/index.html#/topology>





In the above image, you can see the OVS switch is shown in the network topology being managed by ODL controller. Since, we have connected one switch only, that is why only one Openflow node is shown. Here, OVS switch is shown with its datapath ID in decimal format.

## Practical No. 05

**Aim: Implement Mininet on ODL (OpenDayLight) Controller.**

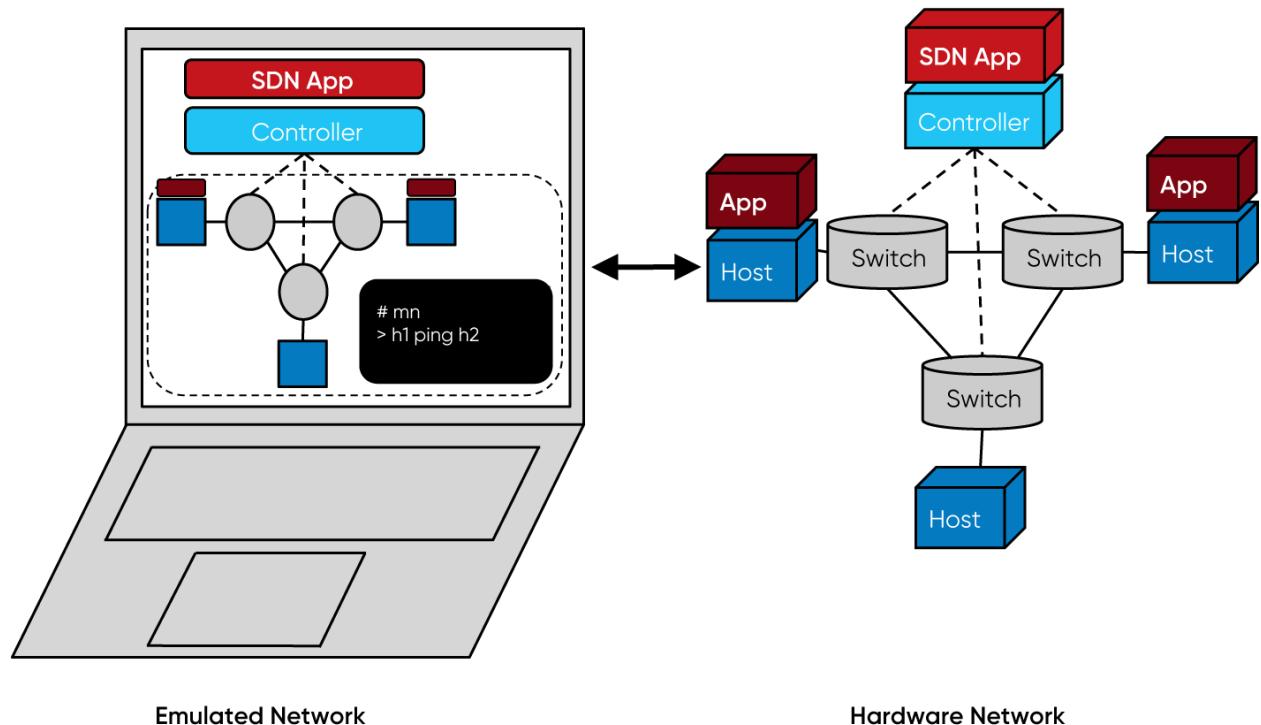
### What is Mininet?

Mininet provides a virtual test bed and development environment for software-defined networks (SDN). Mininet enables SDN development on any laptop or PC, and SDN designs can move seamlessly between Mininet (allowing inexpensive and streamlined development), and the real hardware running at line rate in live deployments. Mininet enables

- Rapid prototyping of software-defined networks
- Complex topology testing without the need to wire up a physical network
- Multiple concurrent developers to work independently on the same topology

Mininet networks run real code including standard Unix/Linux network applications as well as the real Linux kernel and network stack.

Mininet provides an extensible Python API for network creation and experimentations. It is released under a permissive BSD Open Source license and is actively developed and supported by community of networking and SDN enthusiasts.



**Steps:**

**Switch to the root user.**

To check which version of mininet is installed, use the command below:

→ mn --version

```
sdn@sdn-VirtualBox:~$ sudo -i
[sudo] password for sdn:
root@sdn-VirtualBox:~# mn --version

Command 'mn' not found, but can be installed with:

apt install mininet

root@sdn-VirtualBox:~#
```

Install the mininet by using the following command.

→ apt install mininet

```
root@sdn-VirtualBox:~# apt install mininet
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cgroup-bin cgroup-tools iperf libcgroup1 libpython-stdlib libpython2.7 libpython2.7-minimal libpython2.7-stdlib net-tools
  openvswitch-common openvswitch-switch python python-minimal python-pkg-resources python-six python2.7 python2.7-minimal socat
Suggested packages:
  ethtool openvswitch-doc python-doc python-tk python-setuptools python2.7-doc binfmt-support
The following NEW packages will be installed:
  cgroup-bin cgroup-tools iperf libcgroup1 libpython-stdlib mininet net-tools openvswitch-common openvswitch-switch python python-minimal
  python-pkg-resources python-six python2.7 python2.7-minimal socat
The following packages will be upgraded:
  libpython2.7 libpython2.7-minimal libpython2.7-stdlib
3 upgraded, 16 newly installed, 0 to remove and 495 not upgraded.
Need to get 8,346 kB of archives.
After this operation, 19.0 MB of additional disk space will be used.
```

→ mn --version

You can install it by using the following way as well.

To clone the mininet from github, type the following command on terminal.

Before that install the git by using the following command:

→ apt install git

```
root@sdn-VirtualBox:~# git clone https://github.com/mininet/mininet
Command 'git' not found, but can be installed with:

apt install git

root@sdn-VirtualBox:~# apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 498 not upgraded.
Need to get 4,750 kB of archives.
After this operation, 34.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Activate V
Go to Setting

→ git clone <https://github.com/mininet/mininet>

```
root@sdn-VirtualBox:~# git clone https://github.com/mininet/mininet
Cloning into 'mininet'...
remote: Enumerating objects: 10208, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 10208 (delta 21), reused 34 (delta 12), pack-reused 10154
Receiving objects: 100% (10208/10208), 3.23 MiB | 3.50 MiB/s, done.
Resolving deltas: 100% (6803/6803), done.
root@sdn-VirtualBox:~#
```

After that run the `install.sh` script under the `/mininet/util` directory.

→ `./install.sh`  
→ `mn --version`

```
root@sdn-VirtualBox:~/mininet/util# ll
total 100
drwxr-xr-x 7 root root 4096 Apr  6 12:22 .
drwxr-xr-x 11 root root 4096 Apr  6 12:22 ..
-rwxr-xr-x 1 root root 2932 Apr  6 12:22 build-ovs-packages.sh*
-rwxr-xr-x 1 root root 5422 Apr  6 12:22 clustersetup.sh*
-rw-r--r-- 1 root root 1425 Apr  6 12:22 colorfilters
-rwxr-xr-x 1 root root 2001 Apr  6 12:22 doxify.py*
-rwxr-xr-x 1 root root 29180 Apr  6 12:22 install.sh*
drwxr-xr-x 2 root root 4096 Apr  6 12:22 kbuild/
-rwxr-xr-x 1 root root 816 Apr  6 12:22 m*
drwxr-xr-x 2 root root 4096 Apr  6 12:22 nox-patches/
drwxr-xr-x 2 root root 4096 Apr  6 12:22 openflow-patches/
drwxr-xr-x 2 root root 4096 Apr  6 12:22 sch_htb-ofbuf/
-rw-r--r-- 1 root root 507 Apr  6 12:22 sysctl_addon
-rwxr-xr-x 1 root root 6949 Apr  6 12:22 unpep8*
-rwxr-xr-x 1 root root 855 Apr  6 12:22 versioncheck.py*
drwxr-xr-x 2 root root 4096 Apr  6 12:22 vm/
root@sdn-VirtualBox:~/mininet/util# ./install.sh
```

```
root@sdn-VirtualBox:~/mininet/util# mn --version
2.3.1b1
root@sdn-VirtualBox:~/mininet/util#
```

Then run the `./karaf` script, means start the ODL (OpenDayLight) controller.

→ `./karaf`

```
root@sdn-VirtualBox:/home/sdn/Downloads# cd distribution-karaf-0.6.0-Carbon/bin/
root@sdn-VirtualBox:/home/sdn/Downloads/distribution-karaf-0.6.0-Carbon/bin# ./karaf
[INFO] [karaf@sdn-VirtualBox ~] 1>
```

Install the following features.

```
opendaylight-user@root>feature:install odl-restconf odl-l2switch-switch odl-mdsal-apidocs
opendaylight-user@root>feature:install odl-vtn-manager-neutron odl-neutron-service odl-neutron-hostconfig-ovs
Refreshing bundles org.eclipse.persistence.core (156)
```

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

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

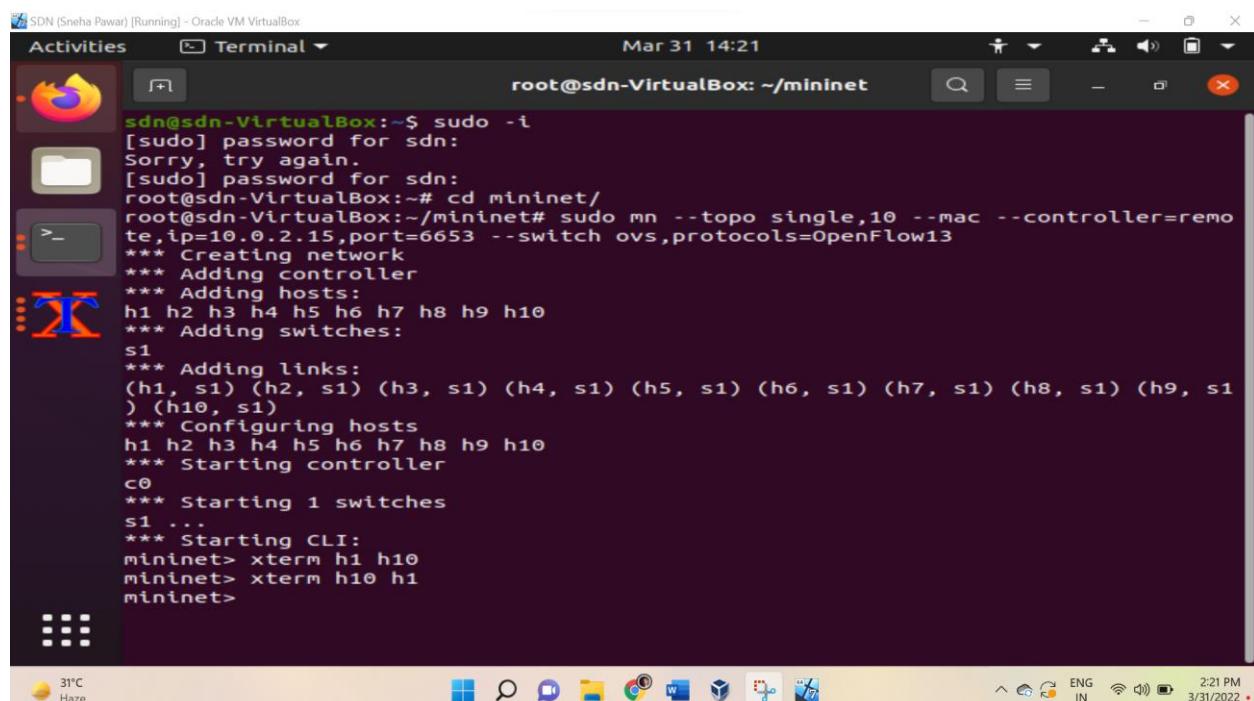
```
opendaylight-user@root>feature:install odl-l2switch-all  
opendaylight-user@root>
```

Then go to mininet directory.

→ cd mininet/

sudo mn – command helps to create a topology.

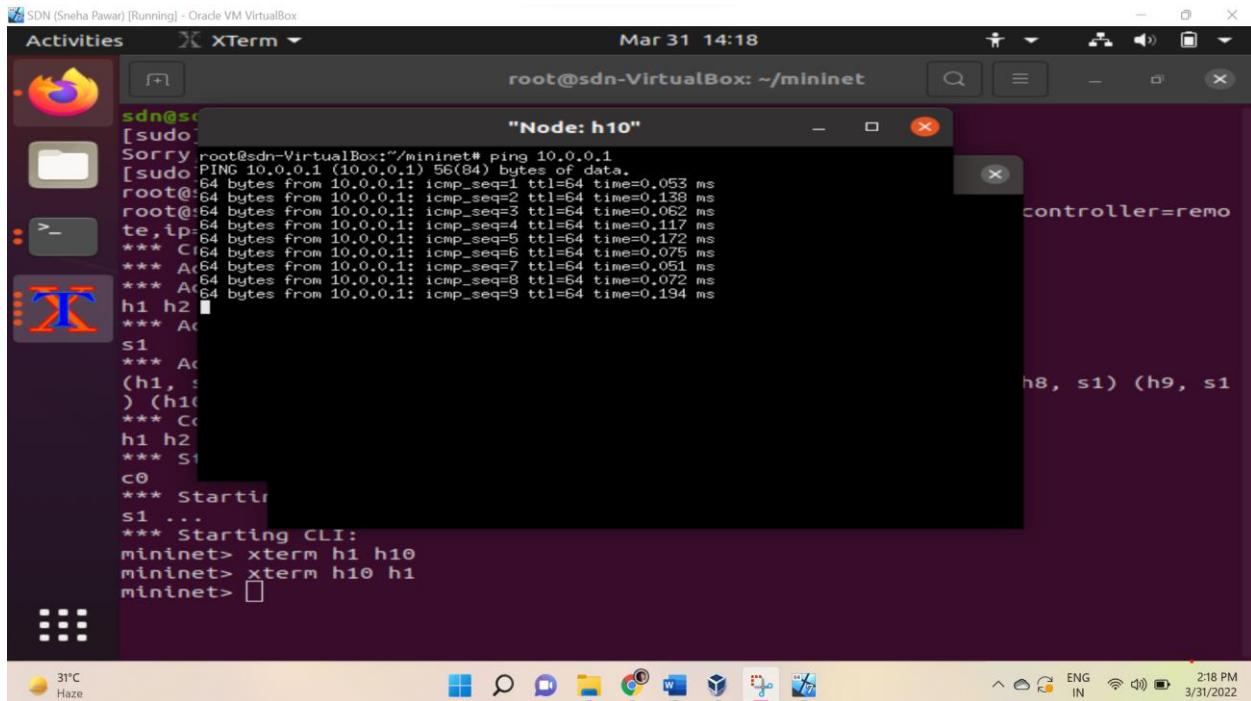
→ Sudo mn –topo single,10 –mac –controller=remote,ip=10.0.2.15,port=6653 –switch ovs,protocol=OpenFlow13



```
sdn@sdn-VirtualBox:~$ sudo -i
[sudo] password for sdn:
Sorry, try again.
[sudo] password for sdn:
root@sdn-VirtualBox:~# cd mininet/
root@sdn-VirtualBox:~/mininet# sudo mn --topo single,10 --mac --controller=remote,ip=10.0.2.15,port=6653 --switch ovs,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1) (h7, s1) (h8, s1) (h9, s1)
(h10, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> xterm h1 h10
mininet> xterm h10 h1
mininet>
```

→ xterm h1 h10

→ xterm h10 h1



For cleanup you can use the following command.

→ Sudo mn -c

```
root@sdn-VirtualBox:~/mininet# sudo mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_coreltnox_core ovs-openflowd
ovs-controllerovs-testcontroller udpbwtest mnexec ivs ryu-manager 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_coreltnox_core ovs-openflowd
ovs-controllerovs-testcontroller udpbwtest mnexec ivs ryu-manager 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+` | sed 's/dp/nl:/
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([-_.[:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
root@sdn-VirtualBox:~/mininet#
```

Then visit the following site on localhost:

<http://localhost:8181/index.html#/login>

**Username: admin**

**Password:admin**

SDN (Sneha Pawar) [Running] - Oracle VM VirtualBox

Activities Firefox Web Browser Mar 31 14:19

Inbox SDN pr Aditi Re Aditi Re OpenDa +

localhost:8181/index.html# 67% ... ☰ ⚡

Nodes

Nodes

Topology Yang UI Yang Visualizer Yangman

Search Nodes

Node Id	IP Address	Node Name	Node Connectors	Statistics
openflow:1	192.168.56.1	s1	11	Flows   Node Connectors

Logout (admin)

31°C Haze 2:19 PM 3/31/2022

This screenshot shows the 'Nodes' section of the OpenDaylight management interface. It displays a table with one row for 'openflow:1'. The columns are labeled 'Node Id', 'IP Address', 'Node Name', 'Node Connectors', and 'Statistics'. The 'Node Name' column contains 's1'. The 'Node Connectors' column contains '11'. A link to 'Flows | Node Connectors' is provided in the 'Statistics' column.

SDN (Sneha Pawar) [Running] - Oracle VM VirtualBox

Activities Firefox Web Browser Mar 31 14:20

Inbox SDN pr Aditi Re Aditi Re OpenDa +

localhost:8181/index.html# 67% ... ☰ ⚡

Topology

Nodes Topology

Yang UI Yang Visualizer Yangman

Controls Reload

This screenshot shows the 'Topology' section of the OpenDaylight management interface. It displays a network diagram with three nodes: 'host:00:00:00:00:00:0a' (top), 'openflow:1' (middle), and 'host:00:00:00:00:00:01' (bottom). An arrow points from the top host to the switch, and another arrow points from the switch to the bottom host.

# Practical No. 6

## Install RYU Controller with Mininet Topology

<https://rpoernama.wordpress.com/2018/02/04/ryu-sdn-controller-installation-on-ubuntu-16-04/>

Ryu requires latest version of pip and python so please install and use python3 and pip3 instead of python and pip (basically replace pip and python with pip3 and python3 in every command)

### 1. install Mininet and Ryu controller in Lubuntu

### 2. 3- Easy steps to check working of Mininet and RYU Controller

### 3. <https://ernie55ernie.github.io/sdn/2019/03/25/install-mininet-and-ryu-controller.html> 4. <https://github.com/sdnds-tw/ryuInstallHelper>

question can be asked like implement ryu with mininet or only implement ryu for practicaljournal you have to implement ryu with mininet

To install pip3: <https://linuxize.com/post/how-to-install-pip-on-ubuntu-18.04/>

Step 1. Install tools

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

```
File Edit View Search Terminal Help
sdn@sdn-VirtualBox:~$ sudo -i
[sudo] password for sdn:
root@sdn-VirtualBox:~# sudo apt-get -y install git python3-pip python3-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.9).
The following additional packages will be installed:
  dh-python libpython3-dev libpython3.6 libpython3.6-dev libpython3.6-minimal
  libpython3.6-stdlib python3-distutils python3-lib2to3 python3-setuptools
  python3-wheel python3.6 python3.6-dev python3.6-minimal
Suggested packages:
  python-setuptools-doc python3.6-venv python3.6-doc binfmt-support
The following NEW packages will be installed:
  dh-python libpython3-dev libpython3.6-dev python3-dev python3-distutils
  python3-lib2to3 python3-pip python3-setuptools python3-wheel python3.6-dev
The following packages will be upgraded:
  libpython3.6 libpython3.6-minimal libpython3.6-stdlib python3.6
  python3.6-minimal
5 upgraded, 10 newly installed, 0 to remove and 476 not upgraded.
Need to get 51.6 MB of archives.
After this operation, 82.2 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython3.6 amd64 3.6.9-1~18.04ubuntu1.7 [1,414 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3.6 amd64 3.6.9-1~18.04ubuntu1.7 [203 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython3.6-stdlib python3.6-dev amd64 3.6.7-1~18.04 [1,000 kB] Activate Windows
Setting up libpython3.6-stdlib (3.6.7-1~18.04) ...
Setting up python3-dev (3.6.7-1~18.04) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
root@sdn-VirtualBox:~# python --version
Python 2.7.17
root@sdn-VirtualBox:~#
```

Step 2. Install python packages

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

```

Processing triggers for libtbb0_2.2-3ubuntu1 ... 
root@sdn-VirtualBox:~# python --version
Python 2.7.17
root@sdn-VirtualBox:~# pip --version
pip 9.0.1 from /usr/lib/python2.7/dist-packages (python 2.7)
root@sdn-VirtualBox:~# sudo apt-get -y install python3-eventlet python3-routes
python3-webob python3-paramiko
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-greenlet python3-pyasn1 python3-repoze.lru
Suggested packages:
  python-eventlet-doc python-greenlet-doc python-greenlet-dev
  python3-greenlet-dbg python3-gssapi python3-paste python-webob-doc
The following NEW packages will be installed:
root@sdn-VirtualBox:~# sudo mkdir ~/project
root@sdn-VirtualBox:~# cd project
root@sdn-VirtualBox:~/project#

```

### Step 3. Clone RYU git Repo

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

```

root@sdn-VirtualBox:~# sudo mkdir ~/project
root@sdn-VirtualBox:~# cd project
root@sdn-VirtualBox:~/project# sudo 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 1021 (delta 307), pack-reused 0
Receiving objects: 100% (1536/1536), 1.44 MiB | 2.53 MiB/s, done.
Resolving deltas: 100% (488/488), done.

```

```
cd ryu
```

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

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

```
sudo apt-get install pip
```

```

Resolving deltas: 100% (488/488), done.
root@sdn-VirtualBox:~/project# ls
ryu
root@sdn-VirtualBox:~/project# sudo pip install setuptools --upgrade
Collecting setuptools
  Downloading https://files.pythonhosted.org/packages/e1/b7/182161210a13158cd3c
cc41ee19aade54496b74f2817cc147006ec932b4/setuptools-44.1.1-py2.py3-none-any.whl
(583kB)
  100% |██████████| 583kB 241kB/s
Installing collected packages: setuptools
  Found existing installation: setuptools 39.0.1
    Not uninstalling setuptools at /usr/lib/python2.7/dist-packages, outside en
vironment /usr
Successfully installed setuptools-44.1.1
root@sdn-VirtualBox:~/project# 

```

```
sudo apt-get install python-pip
```

```

Successfully installed setuptools-44.1.1
root@sdn-VirtualBox:~/project# sudo apt-get install pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package pip
root@sdn-VirtualBox:~/project# sudo apt-get install python-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-pip is already the newest version (9.0.1-2.3-ubuntu1.18.04.5).
0 upgraded, 0 newly installed, 0 to remove and 476 not upgraded.
root@sdn-VirtualBox:~/project# 

```

### 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@sdn-VirtualBox:~/project# sudo pip install setuptools --upgrade
Requirement already up-to-date: setuptools in /usr/local/lib/python2.7/dist-packages
root@sdn-VirtualBox:~/project# pip --version
pip 9.0.1 from /usr/lib/python2.7/dist-packages (python 2.7)
root@sdn-VirtualBox:~/project# sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (9.0.1-2.3~ubuntu1.18.04.5).
0 upgraded, 0 newly installed, 0 to remove and 476 not upgraded.
root@sdn-VirtualBox:~/project# 
0 upgraded, 0 newly installed, 0 to remove and 476 not upgraded.
root@sdn-VirtualBox:~/project# pip3 --version
sudo pip3 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)
root@sdn-VirtualBox:~/project# sudo pip3 install setuptools --upgrade
Collecting setuptools
  Downloading https://files.pythonhosted.org/packages/b0/3a/88b210db68e56854d0b
  cf4b38e165e03be377e13907746f825790f3df5bf/setuptools-59.6.0-py3-none-any.whl (9
  52kB)
    100% |██████████| 962kB 238kB/s
Installing collected packages: setuptools
  Found existing installation: setuptools 39.0.1
    Not uninstalling setuptools at /usr/lib/python3/dist-packages, outside environment /usr
Successfully installed setuptools-59.6.0
root@sdn-VirtualBox:~/project#

```

## Step 6. Test ryu-manager

```

sudo: puip3: command not found
root@sdn-VirtualBox:~/project/ryu# sudo pip3 install six --upgrade
Collecting six
  Downloading https://files.pythonhosted.org/packages/d9/5a/e7c31adbe875f2abbb9
  1bd84cf2dc52d792b5a01506781dbcfc25c91daf11/six-1.16.0-py2.py3-none-any.whl
Installing collected packages: six
  Found existing installation: six 1.11.0
    Not uninstalling six at /usr/lib/python3/dist-packages, outside environment /usr
Successfully installed six-1.16.0
root@sdn-VirtualBox:~/project/ryu#

```

## sudo python setup.py install

```

sudo: puip3: command not found
root@sdn-VirtualBox:~/project/ryu# sudo pip3 install six --upgrade
Collecting six
  Downloading https://files.pythonhosted.org/packages/d9/5a/e7c31adbe875f2abbb9
  1bd84cf2dc52d792b5a01506781dbcfc25c91daf11/six-1.16.0-py2.py3-none-any.whl
Installing collected packages: six
  Found existing installation: six 1.11.0
    Not uninstalling six at /usr/lib/python3/dist-packages, outside environment /usr
Successfully installed six-1.16.0
root@sdn-VirtualBox:~/project/ryu# sudo pip3 install oslo.config msgpack-python
3
Collecting oslo.config
  Downloading https://files.pythonhosted.org/packages/c5/4c/6cf9274f3fb665f276
  ec41bd7c98fce20d5ec903bac7b4cd2c30f5832fe/oslo.config-8.8.0-py3-none-any.whl (1
  28kB)
    100% |██████████| 133kB 324kB/s
Collecting msgpack-python
  wrapt-1.14.0 zippp-3.6.0
root@sdn-VirtualBox:~/project/ryu# sudo pip3 install oslo.config msgpack-python

```

## sudo pip3 install eventlet --upgrade

```

wrapt-1.14.0 zippp-3.6.0
root@sdn-VirtualBox:~/project/ryu# sudo pip3 install eventlet

```

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

```

root@sdn-VirtualBox:~/project/ryu# sudo pip3 install -r tools/pip-requires

```

```

ist-packages (from stevedore>=1.20.0->oslo.config>=2.5.0->-r tools/pip-requires
(line 8))
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8"
  in /usr/local/lib/python3.6/dist-packages (from importlib-metadata>=1.7.0; pyt
hon_version < "3.8"->oslo.config>=2.5.0->-r tools/pip-requires (line 8))
Building wheels for collected packages: msgpack, ovs, tinyrpc
  Running setup.py bdist_wheel for msgpack ... done
  Stored in directory: /root/.cache/pip/wheels/b4/58/67/1a6b3c87c4b15456c801d68
297a8d6e9040b1e95f3293a82cf
  Running setup.py bdist_wheel for ovs ... done
  Stored in directory: /root/.cache/pip/wheels/68/82/aa/6ac87403fd4507d4fcc55ee
5e0b6b2fea98c27f570f6246210
  Running setup.py bdist_wheel for tinyrpc ... done
  Stored in directory: /root/.cache/pip/wheels/b9/dd/89/740804dc7cd0f0b68f1b41f
f1022de61b2dd2024aec09471e9
Successfully built msgpack ovs tinyrpc
Installing collected packages: pip, dnspython, eventlet, msgpack, sortedcontain
ers, ovs, pyparsing, packaging, tinyrpc
  Found existing installation: pip 9.0.1
    Not uninstalling pip at /usr/lib/python3/dist-packages, outside environment
/usr
  Found existing installation: eventlet 0.20.0
    Not uninstalling eventlet at /usr/lib/python3/dist-packages, outside enviro
nment /usr
Successfully installed dnspython-1.16.0 eventlet-0.31.1 msgpack-1.0.3 ovs-2.16.
0 packaging-20.9 pip-20.3.4 pyparsing-3.0.7 sortedcontainers-2.4.0 tinyrpc-1.0.
4
root@sdn-VirtualBox:~/project/ryu# █

```

```

File Edit View Search Terminal Help
warning: no previously-included files matching '*' found under directory 'doc/b
uild'
warning: no previously-included files matching '*~' found anywhere in distribut
ion
warning: no previously-included files matching '*.pyc' found anywhere in distri
bution
warning: no previously-included files matching '.gitignore' found anywhere in d
istribution
adding license file 'LICENSE'
adding license file 'AUTHORS'
writing manifest file 'ryu.egg-info/SOURCES.txt'
running install_lib
running install_data
running install_egg_info
removing '/usr/local/lib/python3.6/dist-packages/ryu-4.34-py3.6.egg-info' (and
everything under it)
Copying ryu.egg-info to /usr/local/lib/python3.6/dist-packages/ryu-4.34-py3.6.e
gg-info
running install_scripts
/usr/local/lib/python3.6/dist-packages/setuptools/command/easy_install.py:2091:
EasyInstallDeprecationWarning: Use get_args
  warnings.warn("Use get_args", EasyInstallDeprecationWarning)
/usr/local/lib/python3.6/dist-packages/setuptools/command/easy_install.py:2093:
EasyInstallDeprecationWarning: Use get_header
  header = cls.get_script_header("", executable, wininst)
Installing ryu script to /usr/local/bin
Installing ryu-manager script to /usr/local/bin
Activate Windows
Go to Settings to activate Windows.
root@sdn-VirtualBox:~/project/ryu# █

```

ryu-manager --version

```

PYTHONPATH=../bin/ryu run --observe-links ryu/app/gui_topology/gui_topology.py
Installing Ryu-Manager script to /usr/local/bin
root@sdn-VirtualBox:~/project/ryu# ryu-manager --version
ryu-manager 4.34
root@sdn-VirtualBox:~/project/ryu# PYTHONPATH=. ./bin/ryu run --observe-links r
yu/app/gui_topology/gui_topology.py
  Import oslo_config.cfg
ImportError: No module named oslo_config.cfg
root@sdn-VirtualBox:~/project/ryu# ls
AUTHORS  CONTRIBUTING.rst  LICENSE          run_tests.sh  setup.py
bin      debian            MANIFEST.in       ryu           tools
build    doc               pip-requirements.txt ryu.egg-info tox.ini
ChangeLog etc              README.rst       setup.cfg
root@sdn-VirtualBox:~/project/ryu# █

```

Cd ryu

```

ImportError: No module named oslo_config.cfg
root@sdn-VirtualBox:~/project/ryu# ls
AUTHORS    CONTRIBUTING.rst    LICENSE
bin        debian             MANIFEST.in
build      doc                pip-requirements.txt
ChangeLog  etc                README.rst
root@sdn-VirtualBox:~/project/ryu# cd ryu
root@sdn-VirtualBox:~/project/ryu/ryu# ls
app        cfg.pyc controller hooks.py     lib      __pycache__ topology
base        cmd    exception.py __init__.py log.py   services   utils.py
cfg.py     contrib flags.py     __init__.pyc ofproto tests
root@sdn-VirtualBox:~/project/ryu/ryu# cd app
root@sdn-VirtualBox:~/project/ryu/ryu/app# ls
bmpstation.py      rest_router.py      simple_switch_lacp.py
cbench.py         rest_topology.py    simple_switch.py
conf_switch_key.py rest_vtep.py       simple_switch_rest_13.py
example_switch_13.py simple_monitor_13.py simple_switch_snort.py
gui_topology      simple_switch_12.py  simple_switch_stp_13.py
__init__.py        simple_switch_13.py simple_switch_stp.py
ofctl            simple_switch_14.py  simple_switch_websocket_13.py
ofctl_rest.py     simple_switch_15.py wsgi.py
rest_conf_switch.py simple_switch_igmp_13.py ws_topology.py
rest_firewall.py  simple_switch_igmp.py
rest_qos.py       simple_switch_lacp_13.py
root@sdn-VirtualBox:~/project/ryu/ryu/app# █

```

Activate Windows  
Go to Settings to activate Windows.

ryu-manager –version

PYTHONPATH= ./bin/ryu run --observe-links ryu/app/gui\_topology/gui\_topology.py

```

root@sdn-VirtualBox:~# cd --
cd--: command not found
root@sdn-VirtualBox:~# cd --
root@sdn-VirtualBox:~# PYTHONPATH=. ./bin/ryu run --observe-links ryu/app/gui_topo
logy/gui_topology.py █

```

Activate Windows  
Go to Settings to activate Windows.

Topology Viewer

Run mininet (or join your real environment):

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

```

root@sdn-VirtualBox:~/project/ryu# sudo mn --controller remote --topo tree,depth
=3
Usage: mn [options]
(type mn -h for details)

mn: error: no such option: -o
root@sdn-VirtualBox:~/project/ryu# sudo mn --controller remote --topo tree,dept
h=3
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7
*** Adding links:
(s1, s2) (s1, s5) (s2, s3) (s2, s4) (s3, h1) (s3, h2) (s4, h3) (s4, h4) (s5, s6
) (s5, s7) (s6, h5) (s6, h6) (s7, h7) (s7, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
*** Starting CLI:
mininet> █

```

Activate Windows  
Go to Settings to activate Windows.

output:

<https://ryu.readthedocs.io/en/latest/gui.html>

Firefox Web Browser — Fri 21:57

## Topology Viewer — Ryu 4.34 documentation - Mozilla Firefox

Topology Viewer — Ryu [Edit on GitHub](#)

Docs » Configuration » Topology Viewer

# Topology Viewer

ryu.app.gui\_topology.gui\_topology provides topology visualization.

This depends on following ryu applications.

ryu.app.rest_topology	Get node and link data.
ryu.app.ws_topology	Being notified change of link up/down.
ryu.app.ofctl_rest	Get flows of datapaths.

## Usage

Run mininet (or join your real environment):

### Screenshot

Mozilla Firefox (Private Browsing) <http://192.168.31.202:8080/>

## Ryu Topology Viewer

```
graph TD; S1[dpid: 1] --- S2[dpid: 2]; S1 --- S3[dpid: 3]; S2 --- S4[dpid: 4]; S2 --- S5[dpid: 5]; S3 --- S6[dpid: 6];
```

Activate Windows  
Go to Settings to activate Windows.

## Practical No. 07

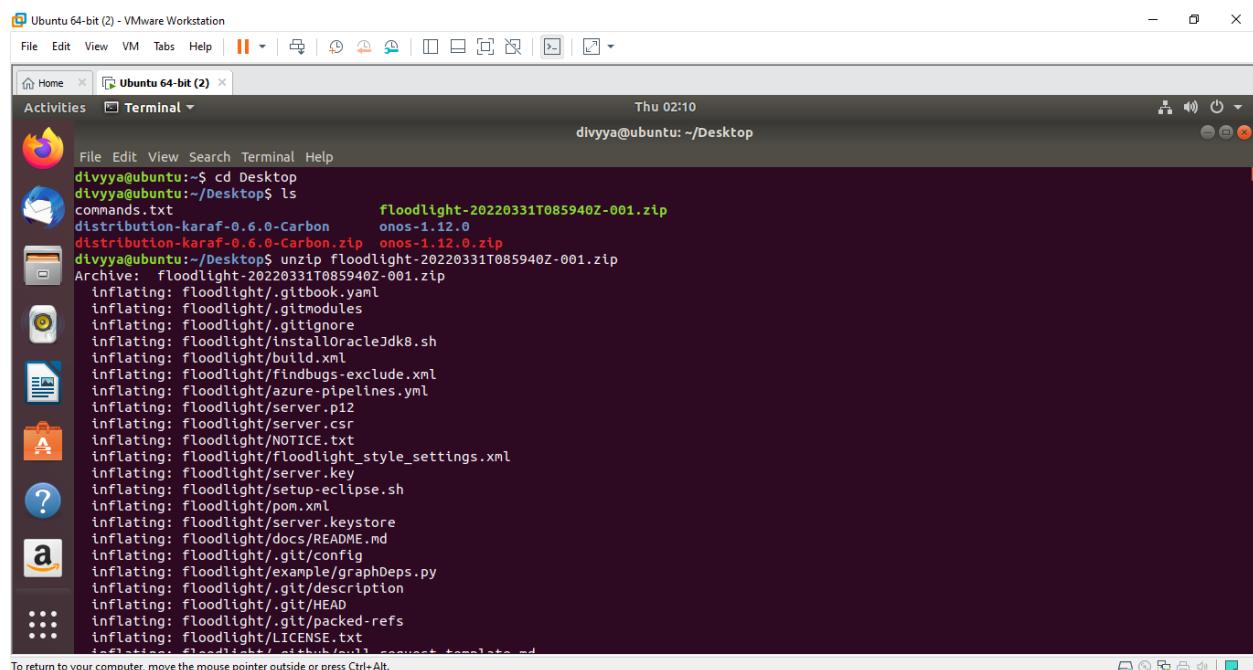
### Install floodlight controller with mininet topology.

#### What is Floodlight Controller?

**Floodlight Controller** is an SDN Controller developed by an open community of developers, many of which from Big Switch Networks, that uses with the OpenFlow protocol to orchestrate traffic flows in a software-defined networking (SDN) environment. OpenFlow is one of the first and most widely used SDN standards; it defines the open communications protocol in an SDN environment that allows the SDN Controller (brains of the network) to speak to the forwarding plane (switches, routers, etc.) to make changes to the network.

The SDN Controller is responsible for maintaining all of the network rules and providing the necessary instructions to the underlying infrastructure on how traffic should be handled. This enables businesses to better adapt to their changing needs and have better control over their networks.

Download the floodlight zip file and unzip it.



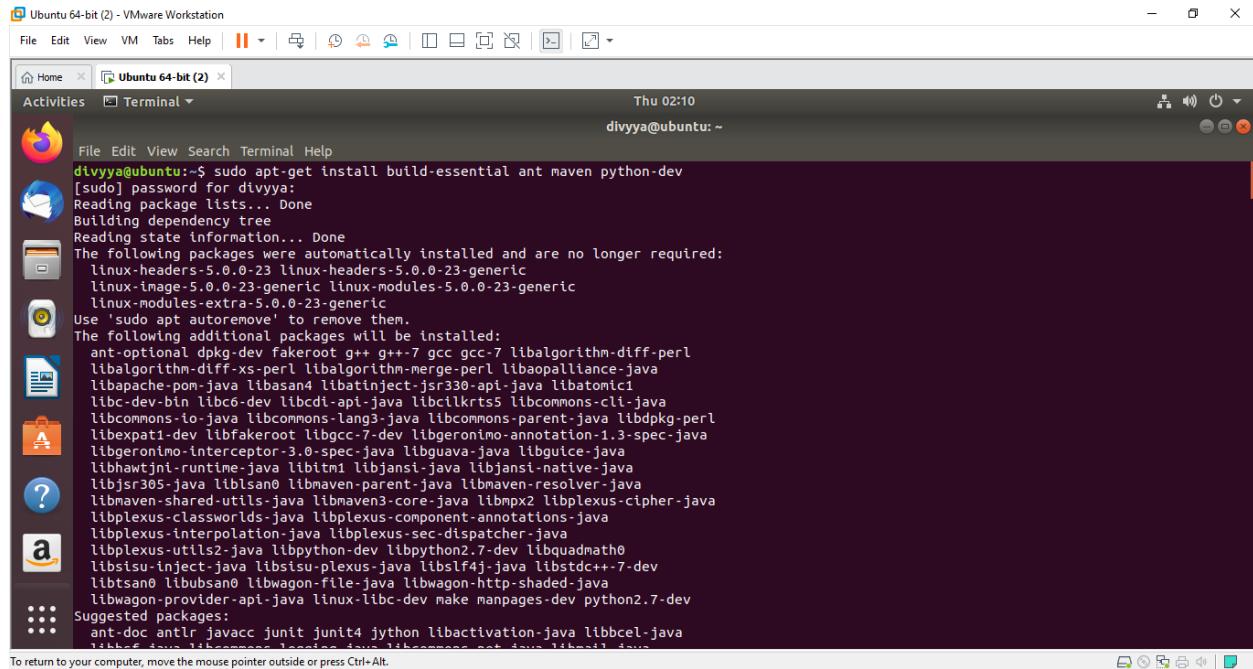
A screenshot of a terminal window titled "Ubuntu 64-bit (2) - VMware Workstation". The window shows a terminal session with the following command history:

```
divyya@ubuntu:~$ cd Desktop
divyya@ubuntu:~/Desktop$ ls
commands.txt          floodlight-20220331T085940Z-001.zip
distribution-karaf-0.6.0-Carbon    onos-1.12.0
distribution-karaf-0.6.0-Carbon.zip onos-1.12.0.zip
divyya@ubuntu:~/Desktop$ unzip floodlight-20220331T085940Z-001.zip
Archive: floodlight-20220331T085940Z-001.zip
  inflating: floodlight/.gitbook.yaml
  inflating: floodlight/.gitmodules
  inflating: floodlight/.gitignore
  inflating: floodlight/installOracleJdk8.sh
  inflating: floodlight/build.xml
  inflating: floodlight/findbugs-exclude.xml
  inflating: floodlight/azure-pipelines.yml
  inflating: floodlight/server.p12
  inflating: floodlight/server.csr
  inflating: floodlight/NOTICE.txt
  inflating: floodlight/floodlight_style_settings.xml
  inflating: floodlight/server.key
  inflating: floodlight/setup-eclipse.sh
  inflating: floodlight/pom.xml
  inflating: floodlight/server.keystore
  inflating: floodlight/docs/README.md
  inflating: floodlight/.git/config
  inflating: floodlight/example/graphDeps.py
  inflating: floodlight/.git/description
  inflating: floodlight/.git/HEAD
  inflating: floodlight/.git/packed-refs
  inflating: floodlight/LICENSE.txt
  inflating: floodlight/.github/pull_request_template.md
```

The terminal window has a dark theme and includes standard Linux desktop icons in the sidebar.

Then install build-essential ant maven python-dev tool.

→ Sudo apt-get install build-essential ant maven python-dev

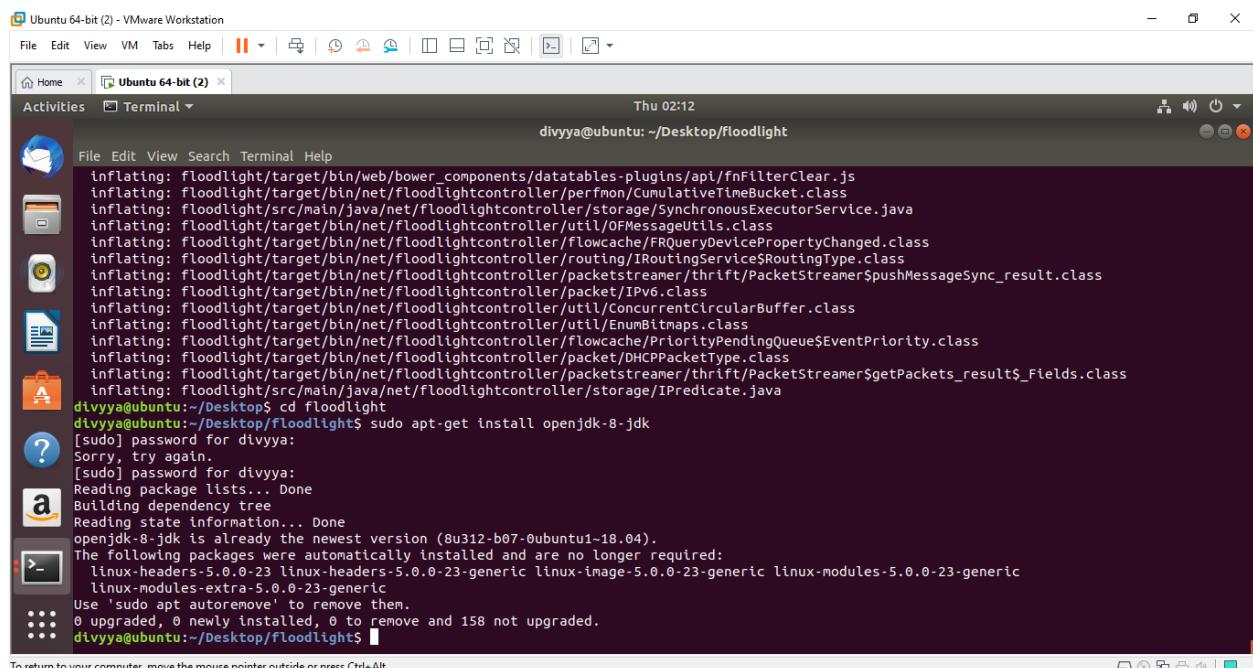


Ubuntu 64-bit (2) - VMware Workstation

File Edit View VM Tabs Help | | Home Ubuntu 64-bit (2) | Activities Terminal Thu 02:10 divyya@ubuntu: ~

```
divyya@ubuntu:~$ sudo apt-get install build-essential ant maven python-dev
[sudo] password for divyya:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.0.0-23 linux-headers-5.0.0-23-generic
  linux-image-5.0.0-23-generic linux-modules-5.0.0-23-generic
  linux-modules-extra-5.0.0-23-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ant-optimal dpkg-dev fakeroot g++ g++-7 gcc gcc-7 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libapaliance-java
  libapache-pom-java libasan4 libatinject-jsr330-api-java libatomic1
  libc-dev-bin libc6-dev libcdi-api-java libclkrts5 libcommons-clt-java
  libcommons-io-java libcommons-lang3-java libcommons-parent-java libdpkg-perl
  libexpat1-dev libfakeroot libgcc-7-dev libgeronimo-annotation-1.3-spec-java
  libgeronimo-interceptor-3.0-spec-java libguava-java libguice-java
  libhawtjni-runtime java libitm1 libjansi-java libjansi-native-java
  libjsr305-java liblsan libmaven-parent-java libmaven-resolver-java
  libmaven-shared-utils-java libmaven3-core-java libmpx2 libplexus-cipher-java
  libplexus-classworlds-java libplexus-component-annotations-java
  libplexus-interpolation-java libplexus-sec-dispatcher-java
  libplexus-util2s-java libpython-dev libpython2.7-dev libquadmath0
  libsisu-inject-java libsisu-plexus-java libslf4j-java libstdc++-7-dev
  libwagon-provider-api-java linux-libc-dev make manpages-dev python2.7-dev
Suggested packages:
  ant-doc antlr javacc junit junit4 jython libactivation-java libbcel-java
  libltsv-java libcommons-logging-java libcommons-net-java libmail-java
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.
```

## Install the java jdk 8 and set the JAVA\_HOME path in /etc/environment file.



Ubuntu 64-bit (2) - VMware Workstation

File Edit View VM Tabs Help | | Home Ubuntu 64-bit (2) | Activities Terminal Thu 02:12 divyya@ubuntu: ~/Desktop/floodlight

```
divyya@ubuntu:~/Desktop$ cd floodlight
divyya@ubuntu:~/Desktop/floodlight$ sudo apt-get install openjdk-8-jdk
[sudo] password for divyya:
Sorry, try again.
[sudo] password for divyya:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-8-jdk is already the newest version (8u312-b07-0ubuntu1-18.04).
The following packages were automatically installed and are no longer required:
  linux-headers-5.0.0-23 linux-headers-5.0.0-23-generic linux-image-5.0.0-23-generic linux-modules-5.0.0-23-generic
  linux-modules-extra-5.0.0-23-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 158 not upgraded.
divyya@ubuntu:~/Desktop/floodlight$
```

```
divyya@ubuntu:~/Desktop/floodlight
GNU nano 2.9.3
/etc/environment
JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk-amd64"
JRE_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"
```

## Execute the ant command in floodlight directory.

```
divyya@ubuntu:~/Desktop/floodlight
Sorry, try again.
[sudo] password for divyya:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-8-jdk is already the newest version (8u312-b07-0ubuntu1-18.04).
The following packages were automatically installed and are no longer required:
  linux-headers-5.0.0-23 linux-headers-5.0.0-23-generic linux-image-5.0.0-23-generic linux-modules-5.0.0-23-generic
  linux-modules-extra-5.0.0-23-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 158 not upgraded.
divyya@ubuntu:~/Desktop/floodlight$ sudo nano /etc/environment
divyya@ubuntu:~/Desktop/floodlight$ sudo nano /etc/environment
divyya@ubuntu:~/Desktop/floodlight$ ant
Buildfile: /home/divyya/Desktop/floodlight/build.xml
[taskdef] Could not load definitions from resource tasks.properties. It could not be found.

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

compile:

compile-test:

dist:
[echo] Setting Floodlight version: 1.2-SNAPSHOT
[echo] Setting Floodlight name: floodlight
```

```

Ubuntu 64-bit (2) - VMware Workstation
File Edit View VM Tabs Help | 
Activities Terminal Thu 02:15
divyya@ubuntu: ~/Desktop/floodlight

File Edit View Search Terminal Help
openjdk-8-jdk is already the newest version (8u312-b07-0ubuntu1-18.04).
The following packages were automatically installed and are no longer required:
  linux-headers-5.0.0-23 linux-headers-5.0.0-23-generic linux-image-5.0.0-23-generic
  linux-modules-extra-5.0.0-23-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 158 not upgraded.
divyya@ubuntu:~/Desktop/floodlight$ sudo nano /etc/environment
divyya@ubuntu:~/Desktop/floodlight$ sudo nano /etc/environment
divyya@ubuntu:~/Desktop/floodlight$ ant
Buildfile: /home/divyya/Desktop/floodlight/build.xml
[taskdef] Could not load definitions from resource tasks.properties. It could not be found.

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

compile:

compile-test:

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

BUILD SUCCESSFUL
Total time: 18 seconds
divyya@ubuntu:~/Desktop/floodlight$ 

```

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

**Then make floodlight directory in /var/lib/floodlight.**

→ Sudo mkdir /var/lib/floodlight

→ Sudo chmod 777 /var/lib/floodlight

Then execute the jar file.

→ Java –jar target/floodlight.jar

```

Ubuntu 64-bit (2) - VMware Workstation
File Edit View VM Tabs Help | 
Activities Terminal Thu 02:17
divyya@ubuntu: ~/Desktop/floodlight

File Edit View Search Terminal Help
BUILD SUCCESSFUL
Total time: 18 seconds
divyya@ubuntu:~/Desktop/floodlight$ sudo mkdir /var/lib/floodlight
divyya@ubuntu:~/Desktop/floodlight$ sudo chmod 777 /var/lib/floodlight
divyya@ubuntu:~/Desktop/floodlight$ java -jar target/floodlight.jar
2022-03-31 02:17:10.888 INFO [n.f.c.m.FloodlightModuleLoader] Loading modules from src/main/resources/floodlightdefault.properties
2022-03-31 02:17:11.167 WARN [n.f.r.RestApiServer] HTTPS disabled; HTTPS will not be used to connect to the REST API.
2022-03-31 02:17:11.168 WARN [n.f.r.RestApiServer] HTTP enabled; Allowing unsecure access to REST API on port 8080.
2022-03-31 02:17:11.168 WARN [n.f.r.RestApiServer] CORS access control allow ALL origins: true
2022-03-31 02:17:11.391 WARN [n.f.c.i.OFSwitchManager] SSL disabled. Using unsecure connections between Floodlight and switches.
2022-03-31 02:17:11.396 INFO [n.f.c.i.OFSwitchManager] Clear switch flow tables on initial handshake as master: TRUE
2022-03-31 02:17:11.397 INFO [n.f.c.i.OFSwitchManager] Clear switch flow tables on each transition to master: TRUE
2022-03-31 02:17:11.397 INFO [n.f.c.i.OFSwitchManager] Setup default rules for all tables on switch connect: true
2022-03-31 02:17:11.409 INFO [n.f.c.i.OFSwitchManager] Setting 0x1 as the default max tables to receive table-miss flow
2022-03-31 02:17:11.469 INFO [n.f.c.i.OFSwitchManager] OpenFlow version OF_15 will be advertised to switches. Supported fallback versions [OF_10, OF_11, OF_12, OF_13, OF_14, OF_15]
2022-03-31 02:17:11.470 INFO [n.f.c.i.OFSwitchManager] Listening for OpenFlow switches on [0.0.0.0]:6653
2022-03-31 02:17:11.471 INFO [n.f.c.i.OFSwitchManager] OpenFlow socket config: 1 boss thread(s), 16 worker thread(s), 60000 ms TCP connection timeout, max 1000 connection backlog, 4194304 byte TCP send buffer size
2022-03-31 02:17:11.472 INFO [n.f.c.i.Controller] ControllerId set to 1
2022-03-31 02:17:11.472 INFO [n.f.c.i.Controller] Shutdown when controller transitions to STANDBY HA role: true
2022-03-31 02:17:11.473 WARN [n.f.c.i.Controller] Controller will automatically deserialize all Ethernet packet-in messages. Set 'deserializeEthPacketIns' to 'FALSE' if this feature is not required or when benchmarking core performance
2022-03-31 02:17:11.473 INFO [n.f.c.i.Controller] Controller role set to ACTIVE
2022-03-31 02:17:11.527 INFO [n.f.l.l.LinkDiscoveryManager] Link latency history set to 10 LLDP data points
2022-03-31 02:17:11.534 INFO [n.f.l.l.LinkDiscoveryManager] Latency update threshold set to +/-0.5 (50.0%) of rolling historical average
2022-03-31 02:17:11.536 INFO [n.f.t.TopologyManager] Path metrics set to LATENCY
2022-03-31 02:17:11.546 INFO [n.f.t.TopologyManager] Will compute a max of 3 paths upon topology updates
2022-03-31 02:17:11.546 INFO [o.f.e.DefaultDefaultForwarder] Default forwarder timeout set to configured. Using a

```

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

**Run the mininet VM.**

```

Ubuntu 18.04.5 LTS mininet-vm tty1
mininet-vm login: mininet
Password:
Last login: Tue Mar 29 21:12:52 PDT 2022 from 192.168.219.130 on pts/1
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-112-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
mininet@mininet-vm:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.219.130 netmask 255.255.255.0 broadcast 192.168.219.255
        ether 00:0c:29:a0:62:1e txqueuelen 1000 (Ethernet)
          RX packets 149 bytes 25306 (25.3 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 152 bytes 15381 (15.3 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
mininet@mininet-vm:~$ _

```

## Create the topology.

→ Ssh –X mininet@192.168.219.139

→ -> sudo mn --controller=remote, ip=192.168.219.139 --topo=single,3

```

Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help | X | 
Home Ubuntu 64-bit Miminet
Activities Terminal Sun 10:06
mininet@mininet-vm: ~

dtvya@ubuntu:~$ ssh -X mininet@192.168.219.139
The authenticity of host '192.168.219.139 (192.168.219.130)' can't be established.
ECDSA key fingerprint is SHA256:agQqGXdgcZl2DPEC2TxVAN3Edox1aiL4jkZgReZL8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.219.130' (ECDSA) to the list of known hosts.
mininet@192.168.219.139's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-112-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Apr  3 09:41:00 2022
mininet@mininet-vm:~$ sudo mn --controller=remote,ip=192.168.219.130 --topo=single,3
*** Creating network
*** Adding controller
Unable to contact the remote controller at 192.168.219.130:6653
Unable to contact the remote controller at 192.168.219.130:6633
Setting remote controller to 192.168.219.130:6653
*** 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

```

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

→ pingall

```

Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help Sun 10:07
Activities Terminal Miminet
File Edit View Search Terminal Help
*** Removing OVS datapaths
ovs-vsctl --timeout=1 llist-br
ovs-vsctl --timeout=1 llist-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([_-][:alnum:]+)-eth[[:digit:]]+'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet
*** Shutting down stale tunnels
pkill -9 -f Tunnel-Ethernet
pkill -9 -f .ssh/mn
rm -rf ~/.ssh/mn/*
*** Cleanup complete.
mininet@mininet-vm:~$ sudo mn --controller=remote,ip=192.168.219.132 --topo=single,3
*** Creating network
*** Adding controller
Connecting to remote controller at 192.168.219.132:6653
*** 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>

```

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

## → ifconfig

```

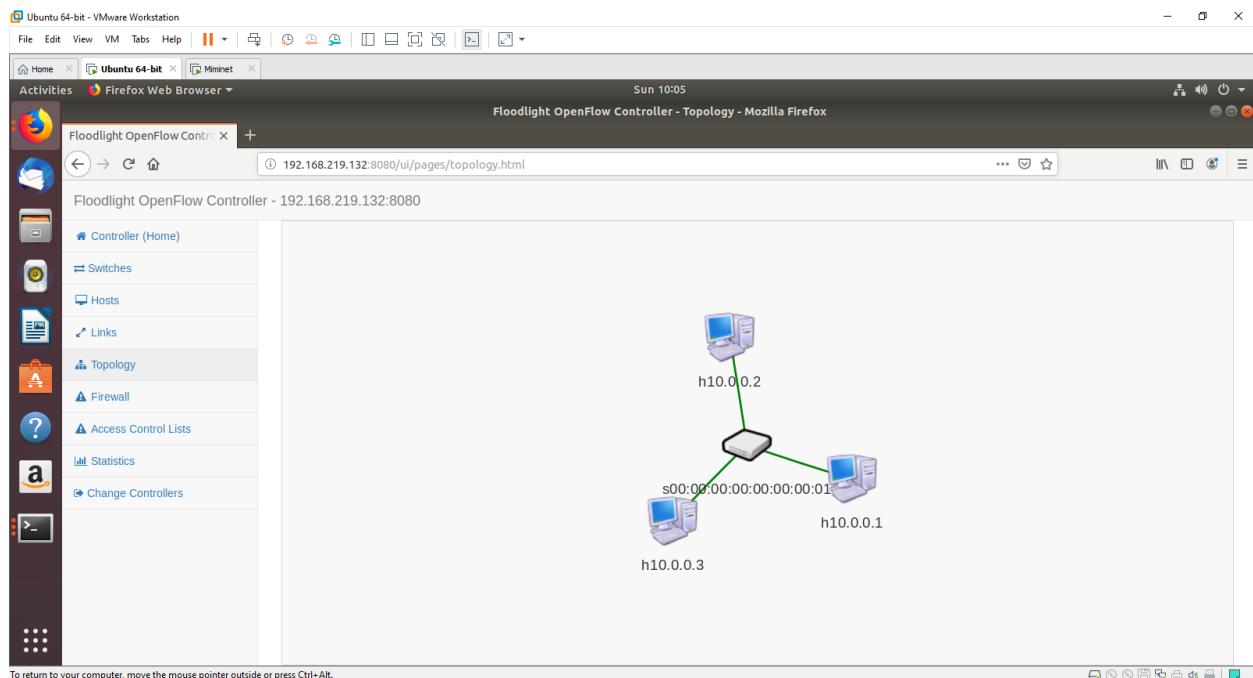
Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help Sun 10:09
Activities Terminal Miminet
File Edit View Search Terminal Help
[sudo] password for divyya:
Reading package lists... Done
Building dependency tree... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 482 not upgraded.
Need to get 194 kB of archives.
After this operation, 803 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 net-tools amd64 1.60+git20161116.90da8a0-1ubuntu1 [194 kB]
Fetched 194 kB in 2s (96.4 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 134886 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
divyya@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.219.132  netmask 255.255.255.0  broadcast 192.168.219.255
              inet 192.168.219.132  netmask 255.255.255.0  broadcast 192.168.219.255
      inet 127.0.0.1  netmask 255.0.0.0
              inet6 ::1  prefixlen 128  scopcid 0x10<host>
      ether 00:0c:29:fb:77:ea  txqueuelen 1000  (Ethernet)
      RX packets 137901  bytes 201674433 (201.6 MB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 48516  bytes 3180922 (3.1 MB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
              inet6 ::1  prefixlen 128  scopcid 0x10<host>
      loop  txqueuelen 1000  (Local Loopback)
      RX packets 751145  bytes 37638241 (37.6 MB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 751145  bytes 37638241 (37.6 MB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
divyya@ubuntu:~$ 

```

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

Then visit the site on browser.

→ <http://192.168.219.132:8080/ui/pages/topology.html>



## Practical no. 08

### Aim: Install ONOS controller on Ubuntu.

What is ONOS(Open Network Operating System)?

ONOS (Open Network Operating System) is an operating system (OS) designed to help network service providers build carrier-grade software-defined networks architected for high scalability, availability and performance. Although specifically designed to address the needs of service providers, ONOS can also act as a software-defined networking (SDN) control plane for enterprise campus local area networks (LANs) and data center networks.

The Open Networking Lab (ON.Lab) released the ONOS source code, written in Java, to the open source community in December 2014. In October 2015, the ONOS project joined the Linux Foundation as a collaborative open source Linux project. New releases of ONOS come out quarterly, in February, May, August and November, and are alphabetically named after a bird, which is also the ONOS logo. The first two releases were named Avocet and Blackbird. As with most open source projects, ONOS has a GitHub page where collaborators can contribute changes to the code.

Among the service providers contributing to the ONOS initiative are AT&T, NTT Communications and SK Telecom. Vendors contributing to ONOS include Cisco, Ericsson, Intel, NEC, Ciena and Huawei. ON.Lab and ONOS partners have found multiple use cases for the operating system. One of the most well-known is ON.Lab's Central Office Re-architected as a Datacenter (CORD). CORD was created to transform telecom carrier central offices into more scalable and agile environments, similar to next-generation data centers. It accomplishes this transformation through virtual network functions and customer premises equipment, for example. ONOS is one of the open source software systems used to manage CORD.

How ONOS works?

The ONOS core is based on a modular architecture, as opposed to an integrated system that blurs the division between its components. This modularity keeps north-south workflows separated from east-west workflows while also permitting easier customization for the entire system. Because service providers require the ability to scale their networks, the ONOS controller can scale out to accommodate a physically distributed system of devices. This allows service providers to add new switches or components without disturbing the rest of the system. Additionally, the distributed architecture reduces network failure, as identical instances can pick up where another fails. This, in turn, results in high availability.

Steps:

- Check the Java version and set it to Java 8.
- `java -version`

```
root@snehapawar-VirtualBox:~# java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1~20.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
root@snehapawar-VirtualBox:~#
```

Then install the curl package.

- **sudo apt-get install curl**

```
root@snehapawar-VirtualBox:~# sudo apt-get install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  libcurl4
1 upgraded, 1 newly installed, 0 to remove and 381 not upgraded.
Need to get 396 kB of archives.
After this operation, 413 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Download the Onos and extract the zip file. And run the onos-service script in bin folder.

- **./bin/onos-service**

```
root@snehapawar-VirtualBox:/home/snehapawar/Downloads# cd onos-1.12.0
root@snehapawar-VirtualBox:/home/snehapawar/Downloads/onos-1.12.0# ./bin/onos-service
Welcome to Open Network Operating System (ONOS)!

[ONOS Logo]

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>
```

Then go to Web Browser:

Login to below link by using the Username & Password below.

**Link: <http://localhost:8181/onos/ui/login.html>**

**Username :- onos**

**Password :- rocks**

