****

**RAMNIRANJAN JHUNJHUNWALA COLLEGE**

**GHATKOPAR (W), MUMBAI - 400 086**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**2021 - 2022**

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

**Software-Defined Networking**

**Name: Soniya Mishra**

**Roll No.: 06**



**CERTIFICATE**

This is to certify that Ms. **Soniya Mishra** with Roll No. **06** has successfully completed the necessary course of experiments in the subject of Natural Language Processing 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 -lII.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner External Examiner

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

Head of Department College Seal

| Practical No. | TOPIC | Date |
| --- | --- | --- |
| 1. | WireShark and bash Script application on ubuntu. | 27/04/2021 |
| 2. | Theory: Hewlett-Packard controller, HP Network Protector, HP Network Visualizer, HP Network Optimizer, Cisco XNC controller | 30/04/2021 |
| 3. | Install ODL(OpenDayLight) controller | 13/03/2021 |
| 4. | Implement OVS on ODL | 17/03/2021 |
| 5. | Implement Mininet. | 12/04/2021 |
| 6. | Install RYU controller with mininet topology. | 16/04/2021 |
| 7. | Install FLOODLIGHT controller with mininet topology. | 19/04/2021 |
| 8. | Install ONOS controller on ubuntu. | 23/04/2021 |

## **Practical No. 1**

## **Wireshark**

**Aim:** Working with Sniffers for monitoring network communication (Ethereal)

Wireshark is a network packet analyzer that intercepts, captures and logs information about packets passing through a network interface. This is useful for analyzing network problems, detecting network intrusions, network misuse, and other security problems, monitor usage and gather statistics, and many other applications.

**Installations and steps for ubuntu**

sudo add-apt-repository universe

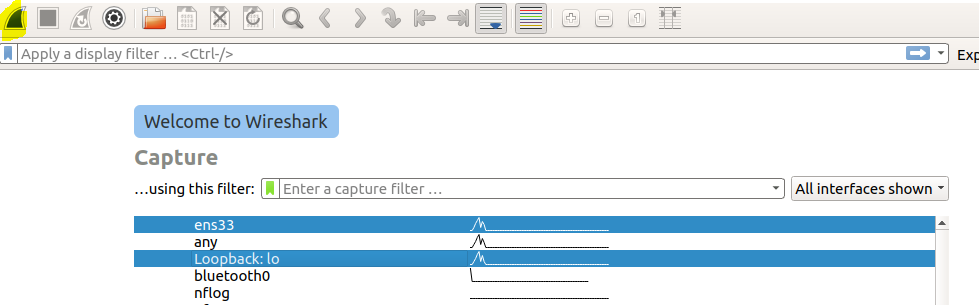
sudo apt install wireshark

If your wireshark start option is not enable do :

sudo dpkg-reconfigure wireshark-common

press the right arrow and enter for yes

sudo chmod +x /usr/bin/dumpcap

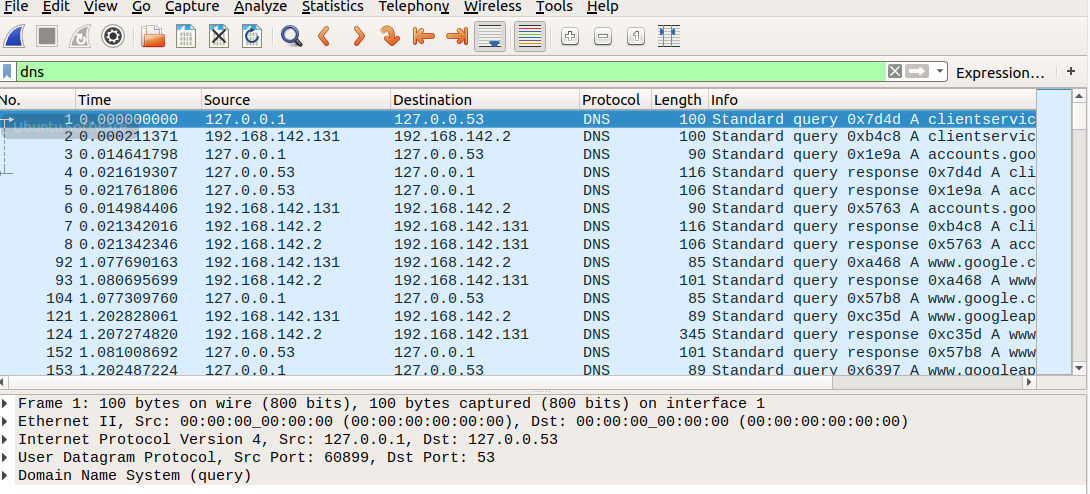


Click on highlighted button by selecting the interfaces.

### **Filtering the Packet List**

Capturing network traffic for a couple minutes could include traffic on many different protocols such as ARP, TCP, UDP, DNS, HTTP, etc.

We may not be interested in all of these, depending on what we are trying to achieve. Fortunately, Wireshark allows us to filter the list based on different criteria using the “Filter” toolbar:

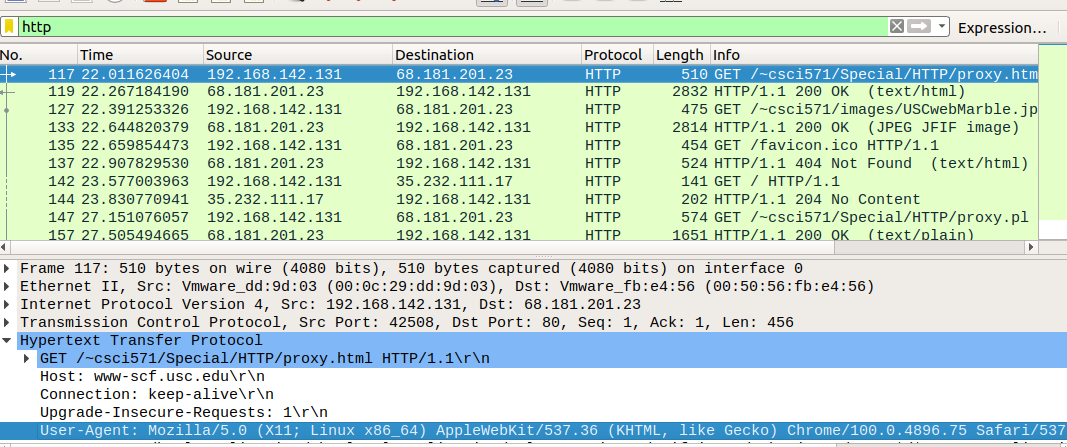


start with the HTTP request sent from your web browser.

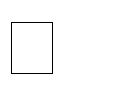
In your web browser, navigate to [some](http://www-scf.usc.edu/%7Ecsci571/Special/HTTP/proxy.html) webpage

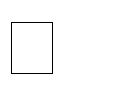
In the middle frame of the Wireshark window, expand the “Hypertext Transfer Protocol” section. Notice the details given for the:

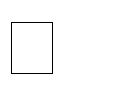
* GET request
* Host
* User-Agent
* Accepts
* cookie
* etc

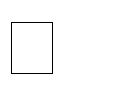
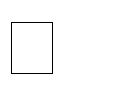


Now, repeat the exercise with the following links and observe the behavior of the corresponding requests and responses:

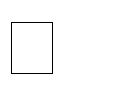
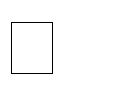
 a file with only text: [http://www-](http://www-scf.usc.edu/%7Ecsci571/Special/HTTP/simple1.html) [scf.usc.edu/~csci571/Special/HTTP/simple1.html](http://www-scf.usc.edu/%7Ecsci571/Special/HTTP/simple1.html)

 a file with text and one gif image: [http://www-](http://www-scf.usc.edu/%7Ecsci571/Special/HTTP/simple2.html) [scf.usc.edu/~csci571/Special/HTTP/simple2.html](http://www-scf.usc.edu/%7Ecsci571/Special/HTTP/simple2.html)

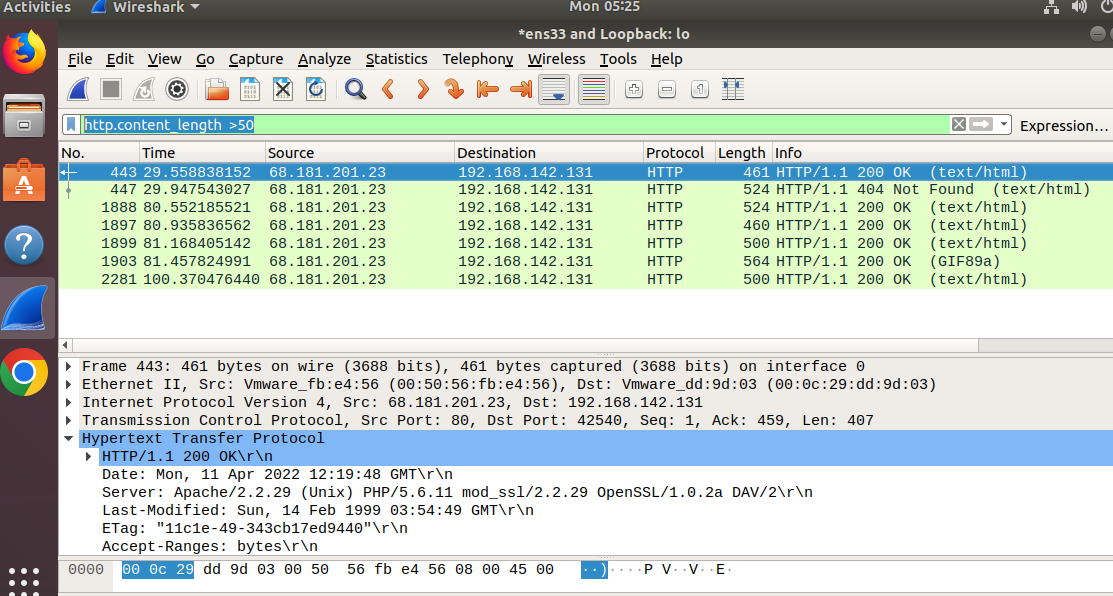
 a file with two frames containing simple1.htm and simple2.html: [http://www-](http://www-scf.usc.edu/%7Ecsci571/Special/HTTP/simple3.html) [scf.usc.edu/~csci571/Special/HTTP/simple3.html](http://www-scf.usc.edu/%7Ecsci571/Special/HTTP/simple3.html)

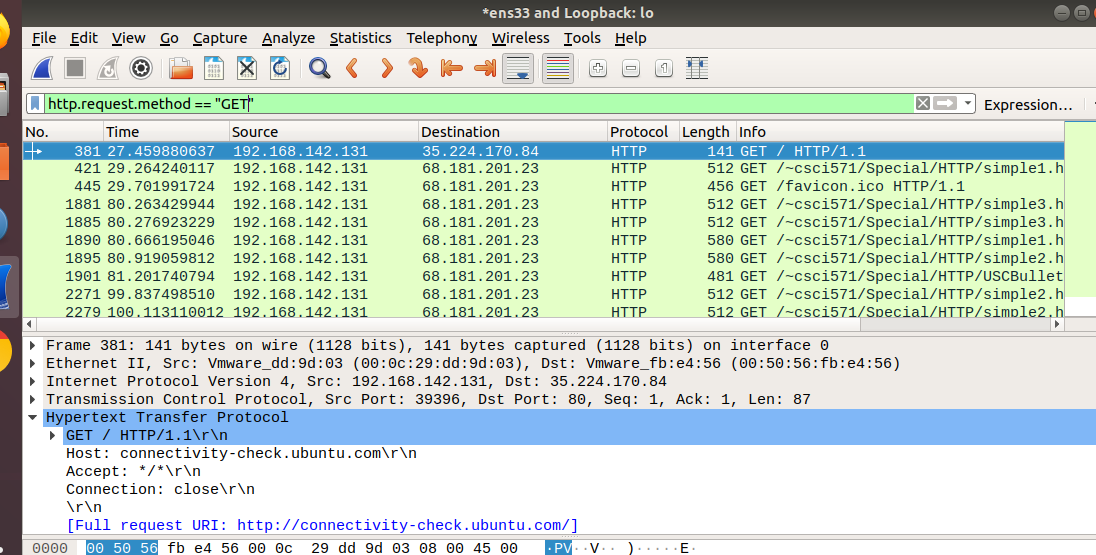
an executable script: http://nunki.usc.edu:8088/cgi-bin/test-cgi

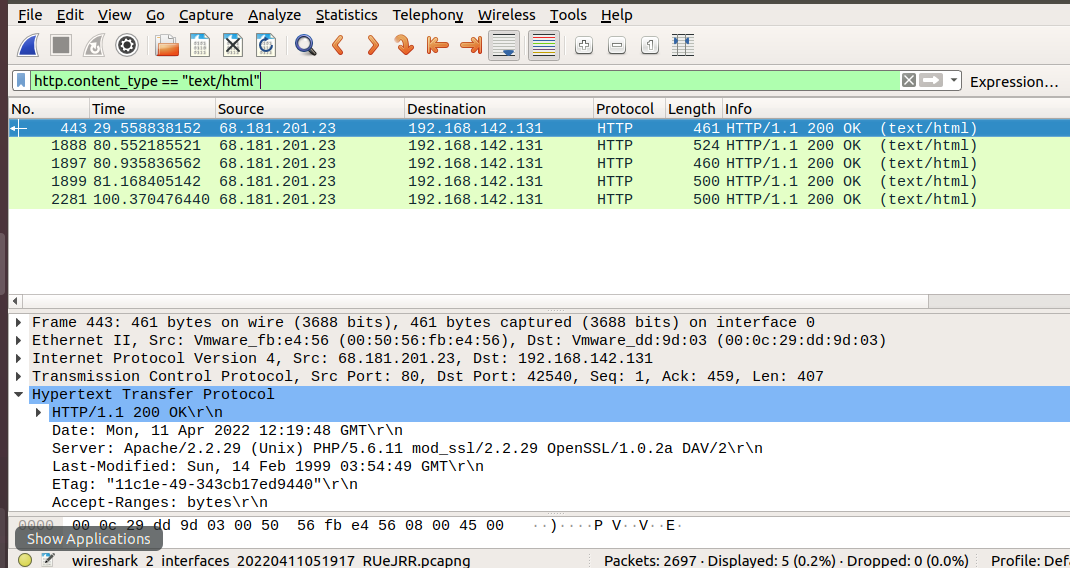
a page that uses the POST method: http://nunki.usc.edu:8088/birthday.html

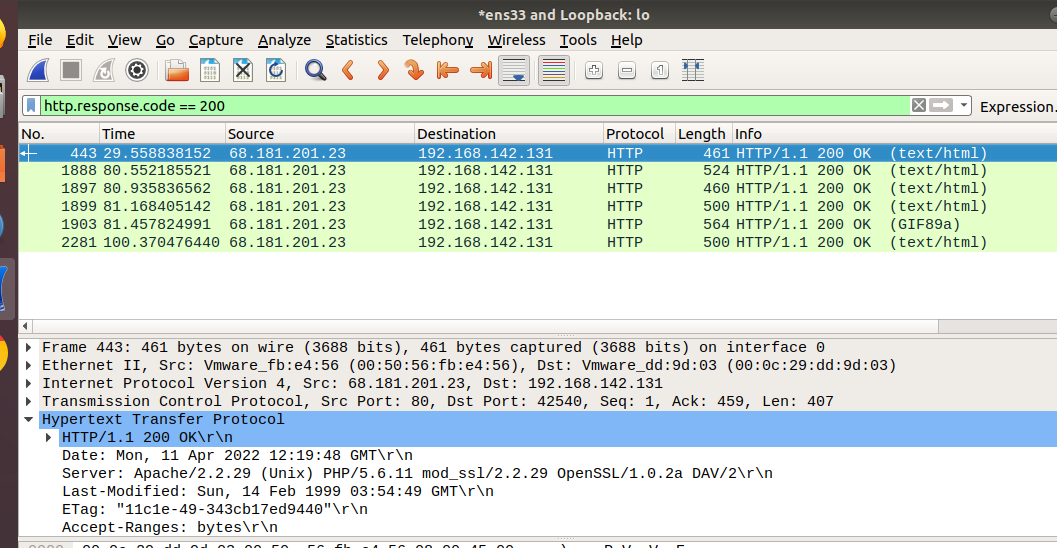
- fill and submit the form, and examine the request and response. a missing file: [http://www-scf.usc.edu/~csci571/missingfile.html](http://www-scf.usc.edu/%7Ecsci571/missingfile.html)

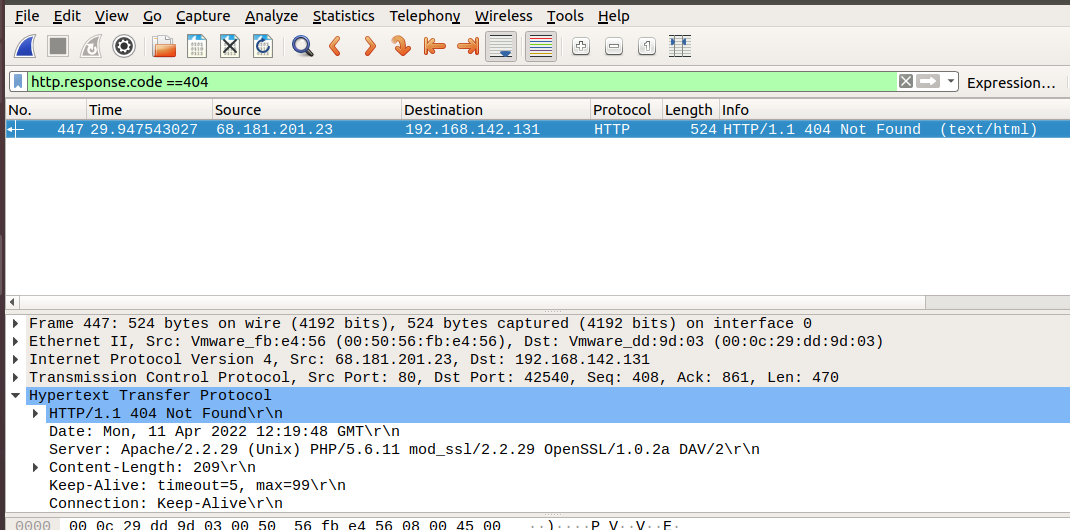
a secure directory: http://nunki.usc.edu:8088/cgi-bin/secure/test-cgi

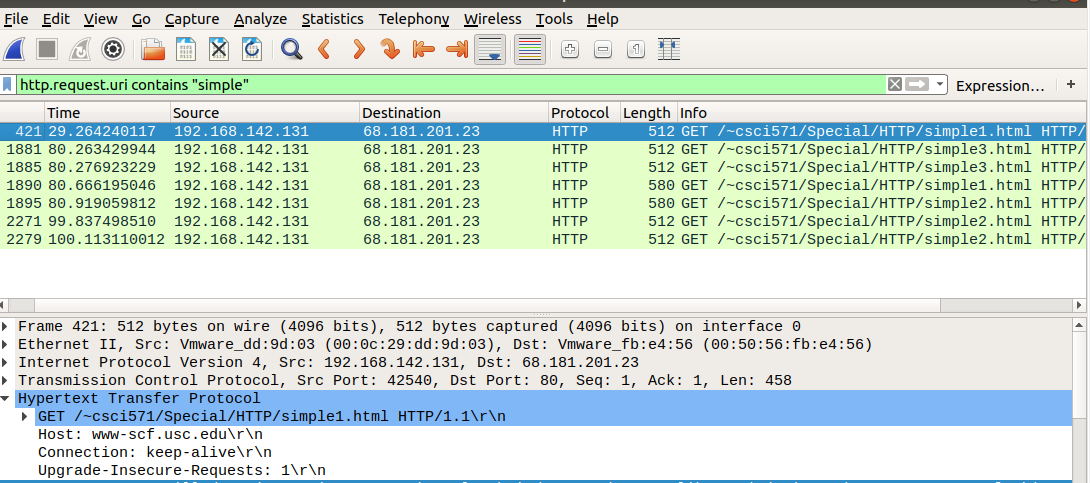


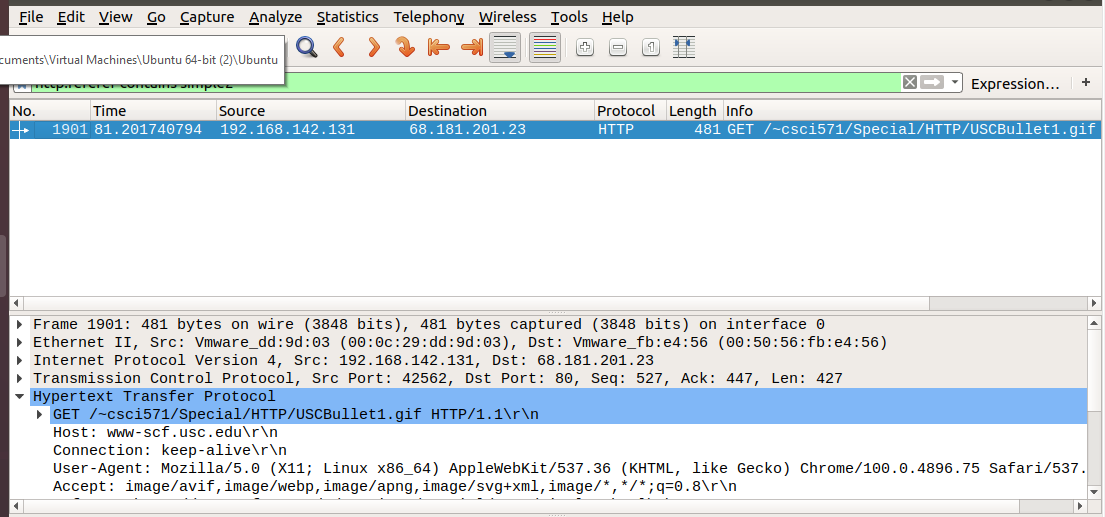


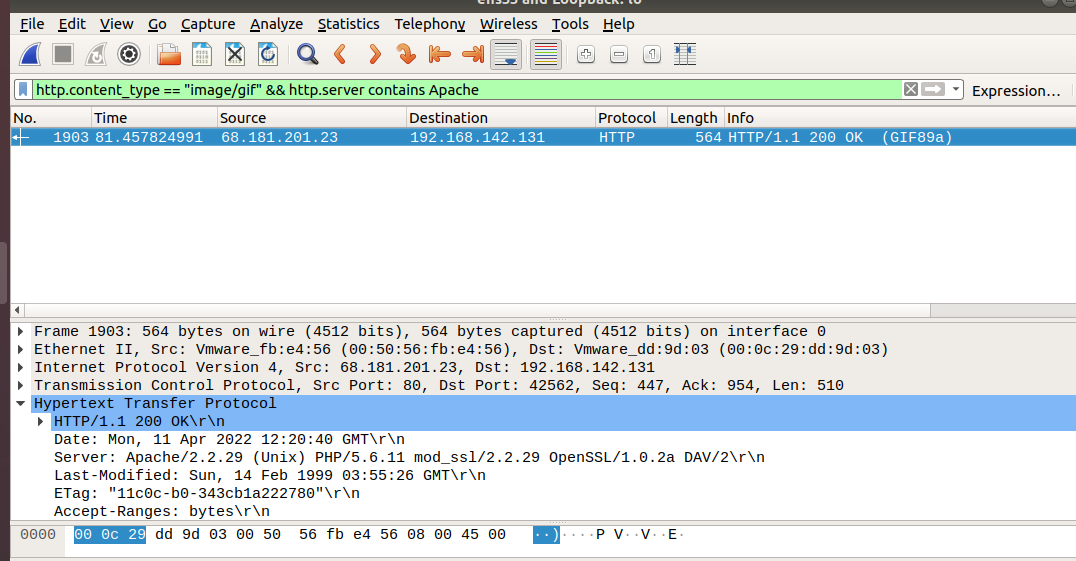












## **Bash Script**

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

Sample program

var=”Hello World”

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

now=”$(date)”

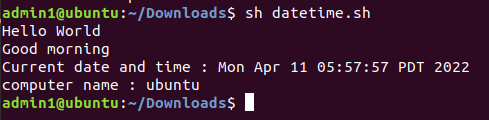
computer\_name=”$(hostname)”

# print it or use the variable

echo “$var”

echo “Current date and time: ” $now

echo “Computer name: ” $computer\_name



Find factorial

echo "Enter a number"

read num

fact=1

while [ $num -gt 1 ]

do

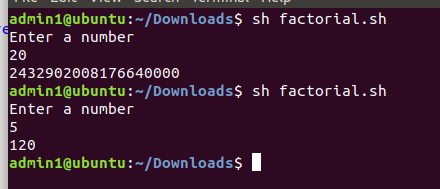
fact=$((fact \* num)) #fact = fact \* num

num=$((num - 1)) #num = num - 1

done

echo $fact

#gt stands for greater than (>).



**Practical No 2**

**Theory: HP controller, HP Network Protector, HP Network Visualizer, HP**

**Network Optimizer, Cisco XNC Controller**

**Hewlett-Packard**

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

The infrastructure layer delivers open programmable access through OpenFlow, a networking

protocol that automates hardware configurations.

New SDN functionality in the infrastructure layer enables clients to simplify network configuration.

HP today announced nine additional switch models providing OpenFlow-enabled support for HP

Flex Network architecture, offering clients a flexible and programmable standards-based interface.

The addition of the new HP 3800 switch series to the 16 existing models in the product portfolio

reaffirms HP as the only major networking vendor with more than 15 million installed OpenFlow-

enabled ports. (1)

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

Within the control layer, the new HP Virtual Application Networks SDN Controller abstracts the

physical hardware from the logical deployment, providing a centralized view and automating

network configuration of all devices in the infrastructure. By eliminating thousands of manual CLI

entries, the controller enables network administrators to easily and flexibly program and scale their

network environment for single-touch automated applications. It also provides application program

interfaces (APIs) to third-party developers to integrate custom enterprise applications.

“We recognized the need to scale our perimeter firewall capacity to cope with the increase of internet

traffic,” said Jean-Michel Juanito, communication systems group leader, IT Department, CERN. “In

the framework of the CERN open lab R&D collaboration, we are developing with HP’s software

defined network technologies a load-balancing application for the Virtual Application Networks SDN

controller to distribute network traffic across multiple devices including firewalls and servers,

increasing simplicity while reducing cost and bandwidth bottlenecks.”

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

network.

New HP Virtual Cloud Networks software enables cloud providers to deliver automated and

scalable public cloud services to enterprises. Using this software, enterprises can create an

isolated virtual cloud network environment through a self-service public cloud infrastructure,

providing them complete control for introducing new services and applications to their users.

Also at the application layer, the new HP Sentinel Security software application automates

network access control and intrusion prevention security for enterprise campus networks with

existing OpenFlow-enabled switch hardware through HP’s controller. As a result, clients can

eliminate the complexity and expense of dedicated networking hardware appliances, while

achieving scalable security needed for bring-your-own-device (BYOD) initiatives.

**HP Network Protector**

The HP Network Protector SDN Application leverages HP Networking, Tipping Point, and ArcSight

products to deliver a converged solution that addresses security threats in a completely new way by

leveraging the network itself. The HP Network Protector enables network intelligence on network

infrastructure devices. The application uses the HP VAN SDN (Virtual Application Network

Software-Defined Networking) Controller and OpenFlow to program the network infrastructure with

security intelligence from the HP Tipping Point Reputation Digital Vaccine (RepDV). This turns the

entire network infrastructure into security-enforcement devices, providing visibility and threat

protection against more than one million malicious botnets, malware, and spyware sites.

Some of the key features of the solution include:

Runs on HP VAN SDN Controller

Complimentary to Tipping Point IPS solutions

Delivers real-time threat characterization with HP Tipping Point DVLabs Database

Protects from over 1,000,000+ botnet, malware, spyware, and malicious sites

OpenFlow enabled switches gain ability to detect malware, botnets, and other threats

Ability to create custom whitelist and blacklist

Improves visibility and accuracy with ArcSight Integration

Dynamic switch learning with HPN OpenFlow enabled switches distributes detection into

switch infrastructure

**HP Network Visualizer**

The HP Network Visualizer SDN Application by utilizing HP VAN SDN Controller provides

dynamic traffic capture with real-time detailed network monitoring allowing for fast network

diagnosis and verification, rapid transition from incident to fix.

Features

1. Real-time Visibility and Diagnosis

Network Visualizer provides dynamic traffic capture to diagnose the root cause of the network. It

proactively monitors the network to reduce the number of help desk issues.

2. Low Cost, Simple and Automated Troubleshooting

Network Visualizer allows for simple troubleshooting that requires high level network detail. The

application eliminates the need for any expensive manual network tapping tools for troubleshooting.

3. Fast Transition from Incident to Fix

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

**Network Optimizer**

HP Network Optimizer SDN Application for Microsoft Lync enables automated provisioning of

network policy and quality of service to provide an enhanced user experience. The Network

Optimizer Application dynamically provisions the end-to-end network path and Quality of Service

(QoS) policy via the HP Virtual Application Networks (VAN)SDN Controller, reducing the need for

manual, device-by-device configuration via the CLI, which greatly simplifies policy deployment and

reduces the likelihood of human errors.

Network Optimizer provides:

Traffic classification – Identify application traffic requiring preferential treatment

Policy enforcement – Prioritize time sensitive traffic based QoS profiles, supports up to 10K

users within a single SDN domain

Visibility into per call SDN operation in terms of DSCP marking applied

Granular sorting and display filtering – by IP address, media type, call quality and jitter

History records management

Configurable global template and phone trust

**Cisco XNC Controller**

Cisco created the Cisco XNC Controller, in order to keep up with the changing software-defined

networking (SDN) environments. Its support of OpenFlow, the most widely used SDN

communications standard, helps it integrate into varied SDN deployments to enable organizations to

better control and scale their networks. As of mid-2015, Cisco has retired the Cisco XNC Controller

— see What the Cisco XNC Controller Tells Us About OpenDaylight to learn about XNC’s demise.

As an SDN Controller, which is the “brains” of the network, Cisco XNC uses OpenFlow to

communicate information “down” to the forwarding plane (switches and routers), with southbound

APIs, and “up” to the applications and business logic, with northbound APIs. It enables organizations

to deploy and even develop a variety of network services, using representational state transfer

application program interfaces (REST APIs), as well as Java APIs.

The XNC is Cisco’s implementation of the OpenDaylight stack. Cisco is a contributor to the

OpenDaylight initiative, which is focused on developing open standards for SDN that promote

innovation and interoperability. Cisco XNC is designed to deliver the cutting edge OpenDaylight

technologies as commercial, enterprise-ready solutions.

## **Practical No. 3**

## **ODL**

sudo apt update

**sudo apt install openjdk-8-jdk openjdk-8-jre**

**java -version**

**Setup JAVA\_HOME and JRE\_HOME Variable**

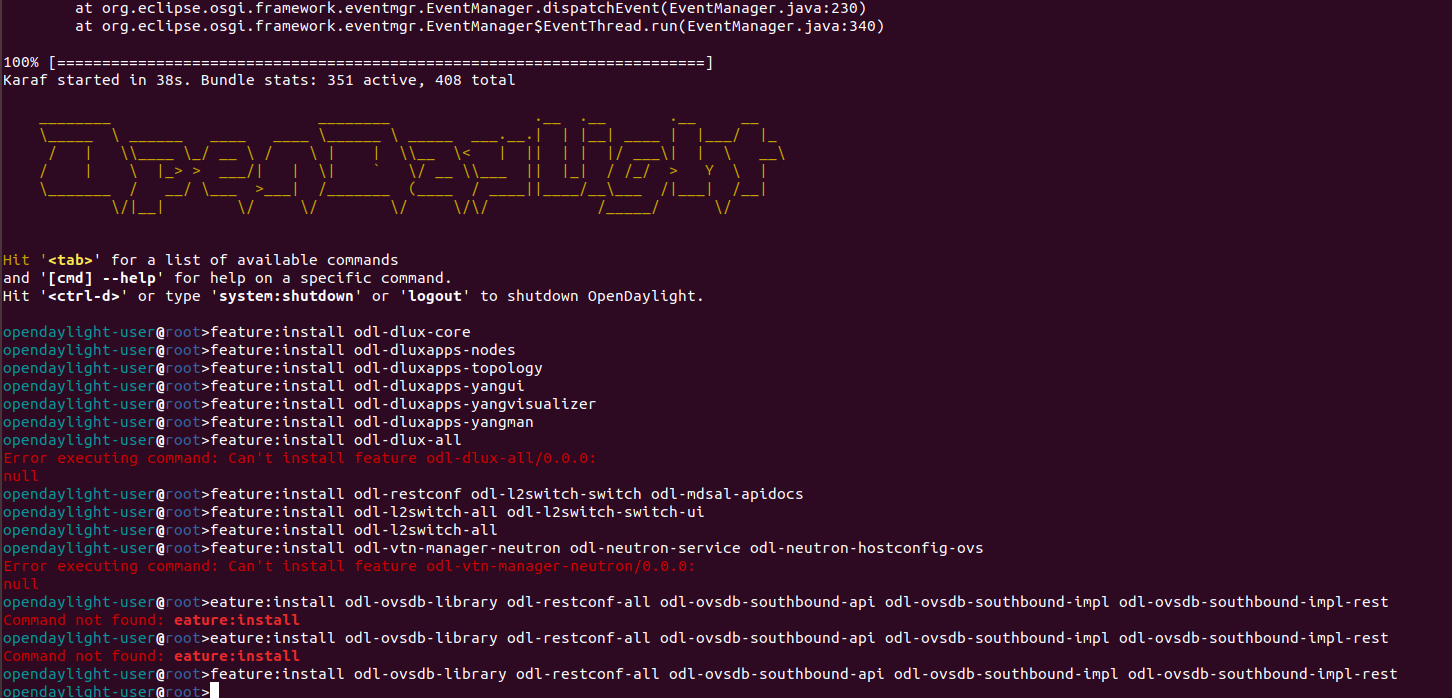
export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export JRE\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre

Step 2 : STEP 2: Download distribution-karaf-0.6.0-Carbon.zip and unzip it Move to the directory distribution-karaf-0.6.0-Carbon and type: ./bin/karaf

<https://docs.onapp.com/agm/6.0/network-settings/sdn-management/install-opendaylight-controller>





STEP 3: Install features

following ARE ODL-DLUX-ALL FEATURES

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

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

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

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

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

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

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

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

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

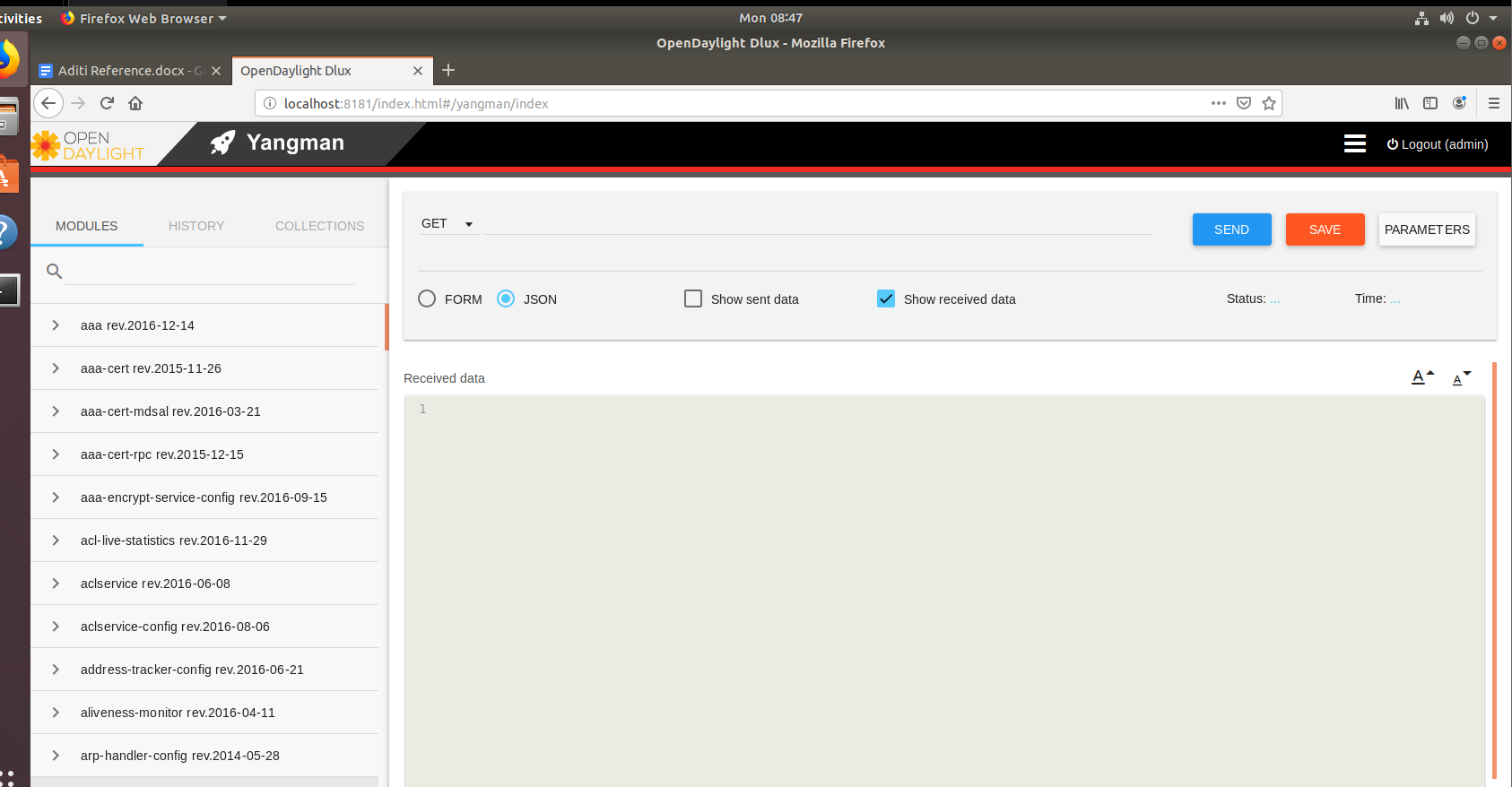
opendaylight-user@root> feature install odl-vtn-manager-neutron odl-neutron-service odl-neutron-hostconfig-ovs

>feature:install odl-ovsdb-library odl-restconf-all odl-ovsdb-southbound-api odl-ovsdb-southbound-impl odl-ovsdb-southbound-impl-rest

Even ODL has dashboard DLUX feature which gives GUI web interface to ODL

controller at URL http://localhost:8181/index.html

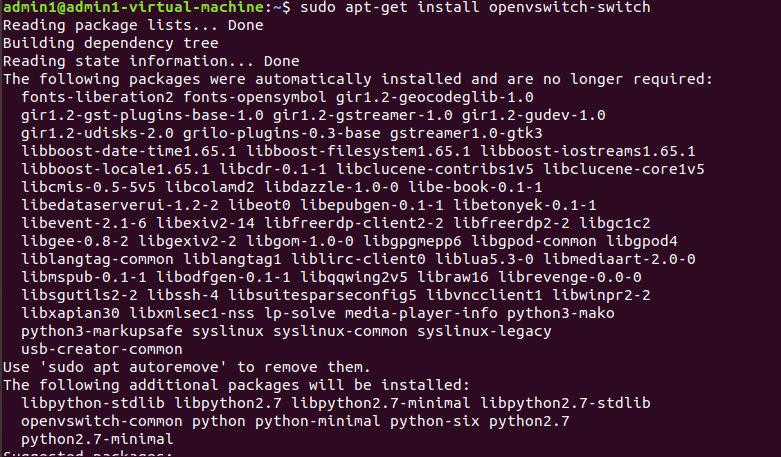
Default Password is admin admin



**Practical 4**

**Implement OVS on ODL**

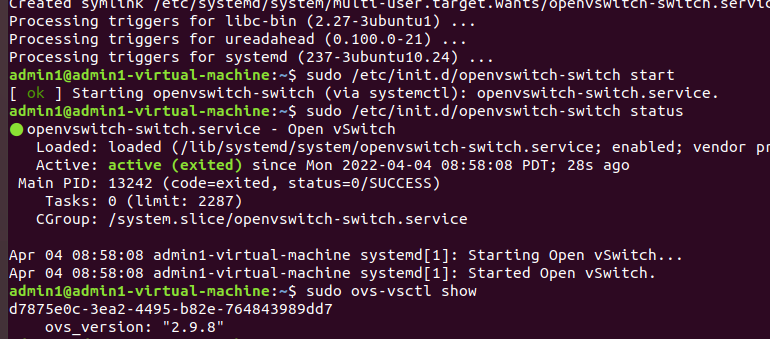
Step 1 sudo apt-get install openvswitch-switch



Step 2 : sudo /etc/init.d/openvswitch-switch start

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

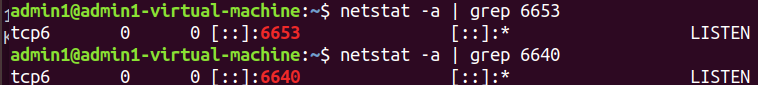
sudo ovs-vsctl show



sudo apt install net-tools

Step 3: netstat -a | grep 6653

netstat -a | grep 6640



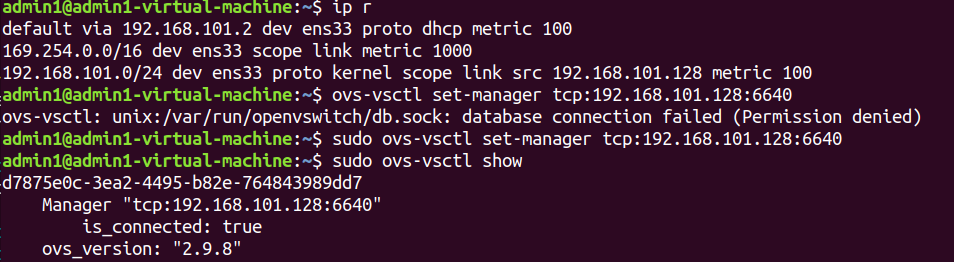
Step 4: Ip r

In my case ip is 192.168.101.128

Step 5 : ovs-vsctl set-manager tcp:<controller-IP>:6640 in this command write your ip instead of controller-IP

sudo ovs-vsctl set-manager tcp:192.168.101.128 :6640

sudo ovs-vsctl show

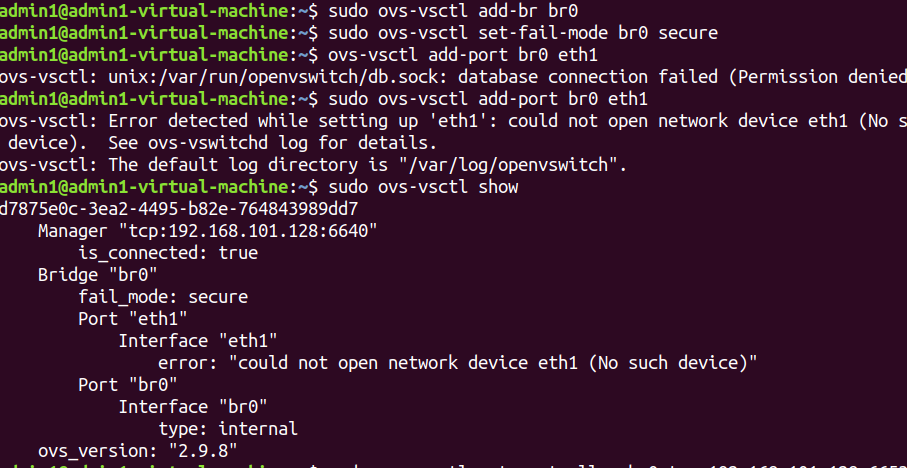


Step 6 : sudo ovs-vsctl add-br br0

sudo ovs-vsctl set-fail-mode br0 secure

sudo ovs-vsctl add-port br0 ens33

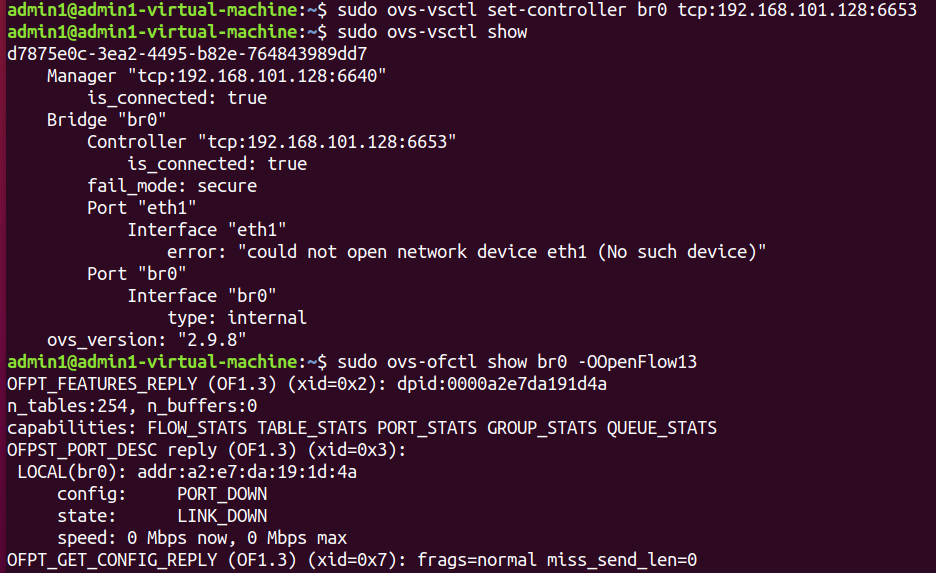
sudo ovs-vsctl show



sudo ovs-vsctl set-controller br0 tcp:192.168.101.128:6653

sudo ovs-vsctl show

sudo ovs-ofctl show br0 -OOpenFlow13

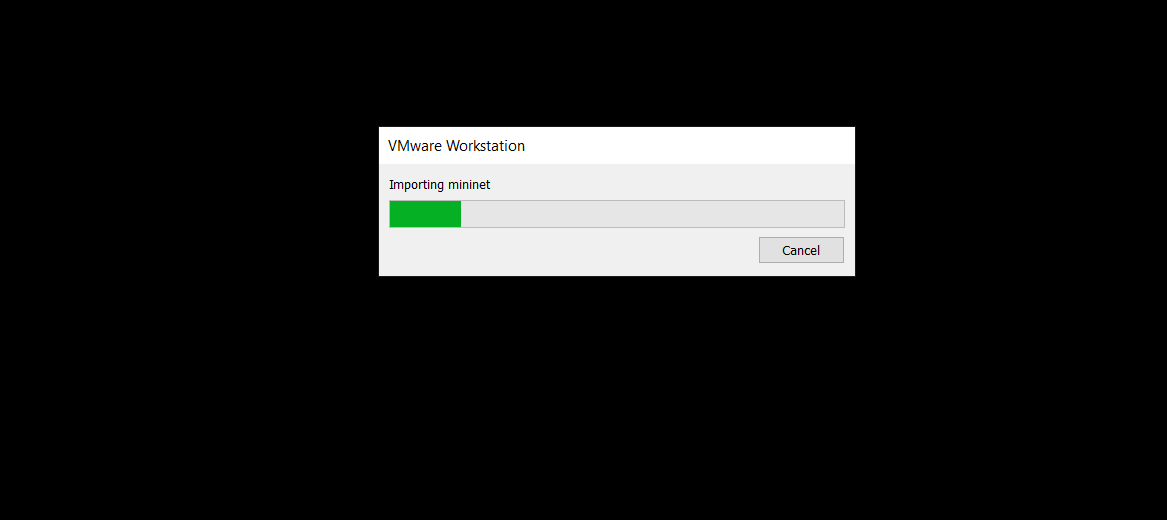


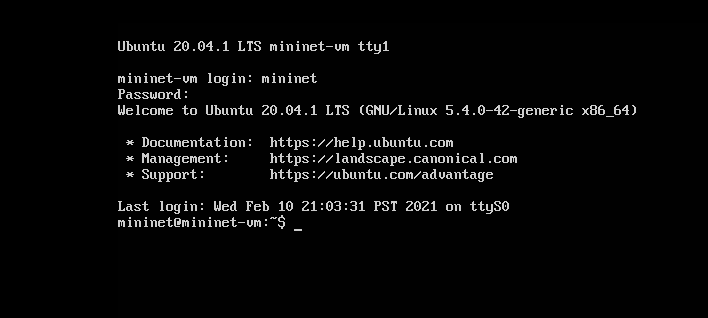
**Practical No 5**

**Implement Mininet.**

**Step 1: download VM** [**https://github.com/mininet/mininet/releases/**](https://github.com/mininet/mininet/releases/)

**Step 2: Open OVF file in the workstation.**





Username is mininet and password is mininet

Do ifconfig to know the ip of the mininet



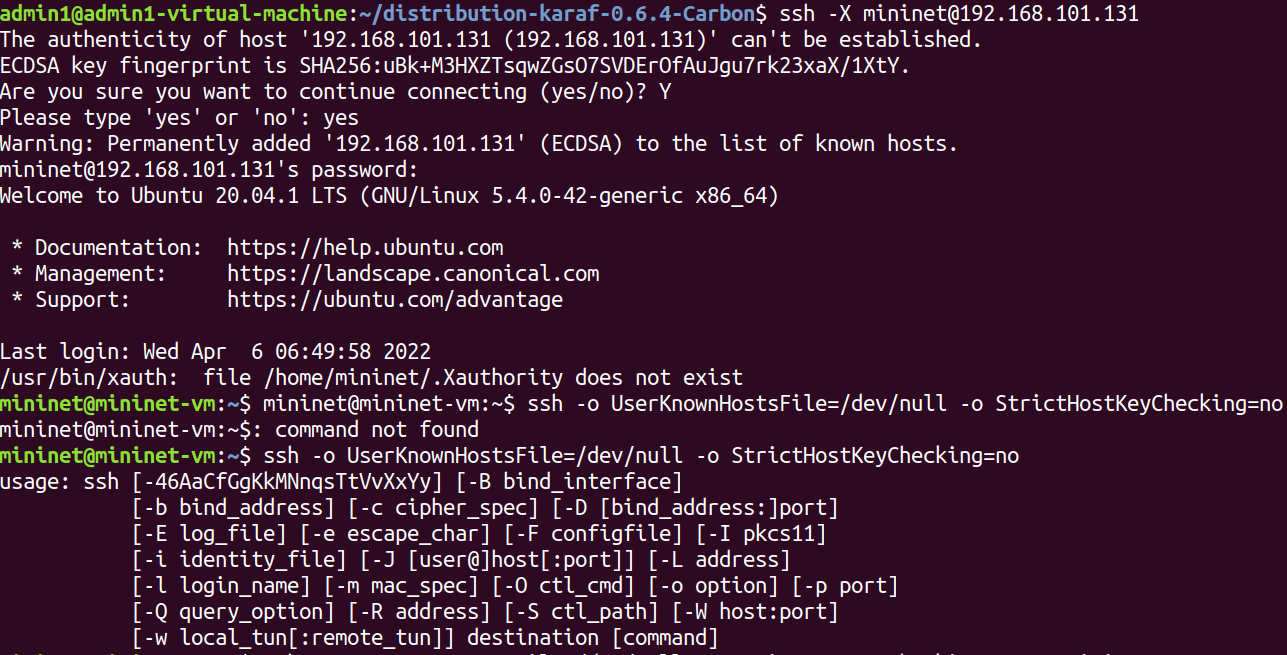
**Open new terminal in ubuntu**

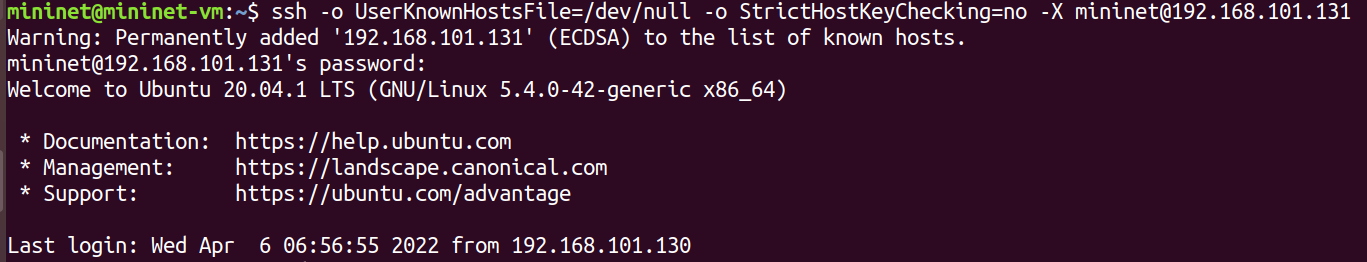
ssh -X mininet@192.168.142.130

Above IP address is of mininet

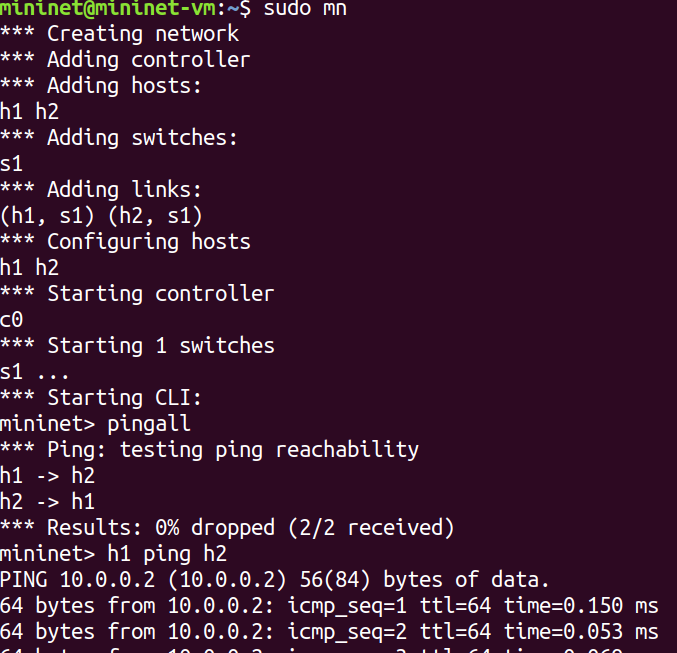
if the authority error exists then type

ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -X mininet@192.168.142.130



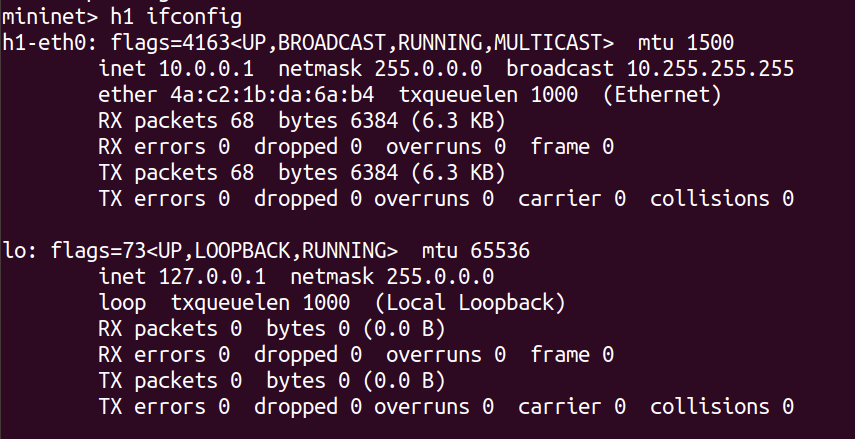


sudo mn



Command that can be performed in mininet CLI:

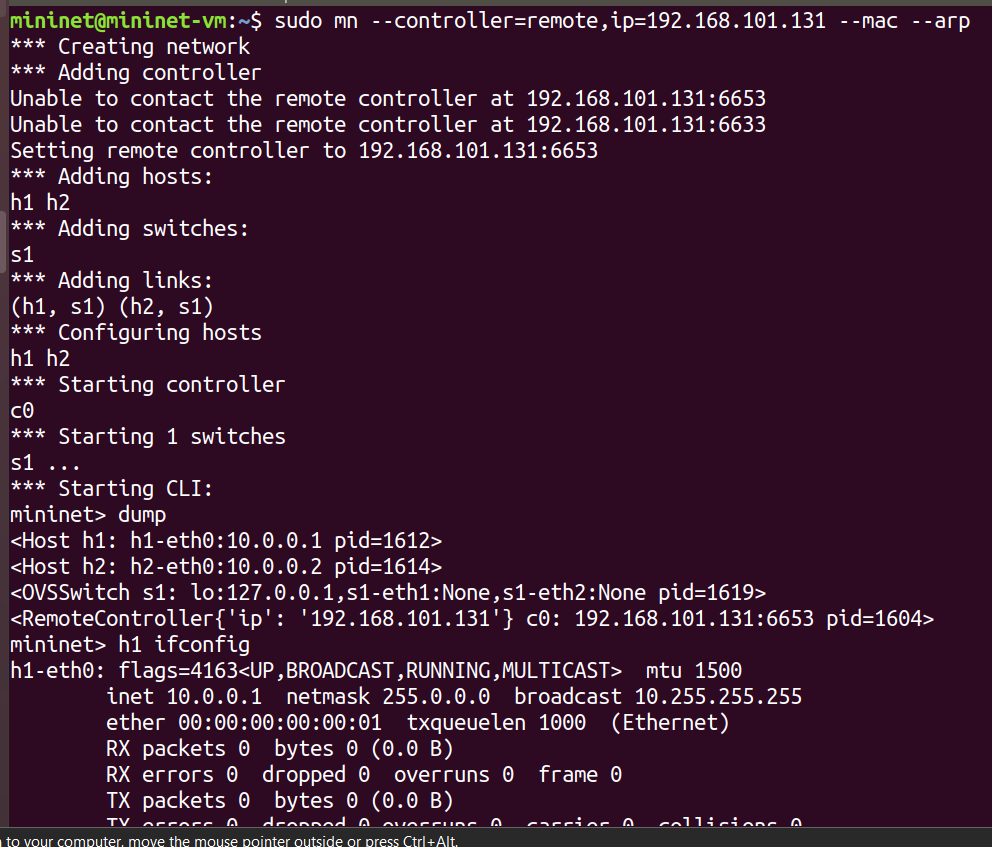
1. pingall
2. h1 ping h2
3. h1 ifconfig
4. dump



**Step 4: in the terminal add mininet topology to controller** type:

sudo mn --controller=remote,ip=192.168.83.134 --mac --arp

dump



**Practical 6**

**Install RYU controller with mininet topology.**

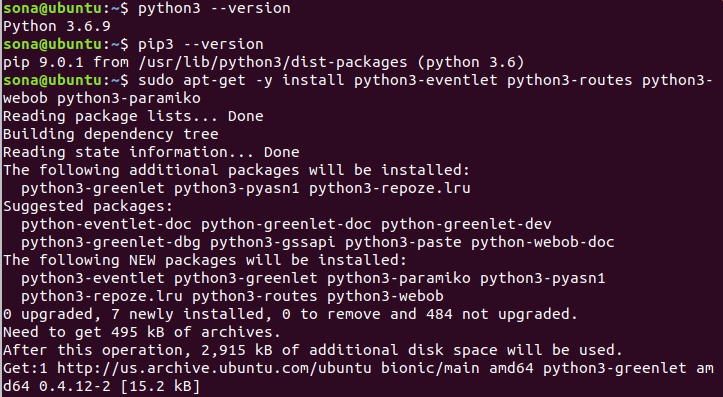
sudo apt-get install git

sudo apt-get -y install git python3-pip python3-dev

python3 --version

pip3 --version

sudo apt-get -y install python3-eventlet python3-routes python3-webob python3-paramiko



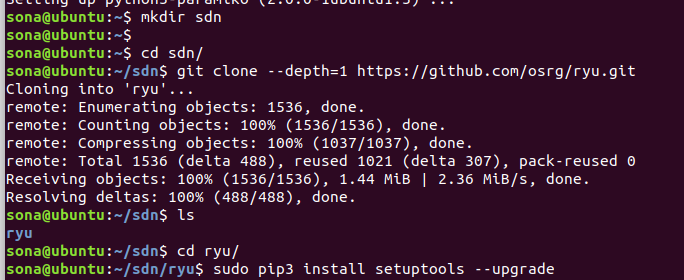
mkdir sdn

cd sdn/

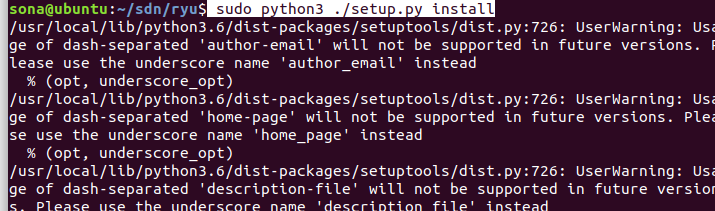
git clone --depth=1 <https://github.com/osrg/ryu.git>

cd ryu/

sudo pip3 install setuptools --upgrade

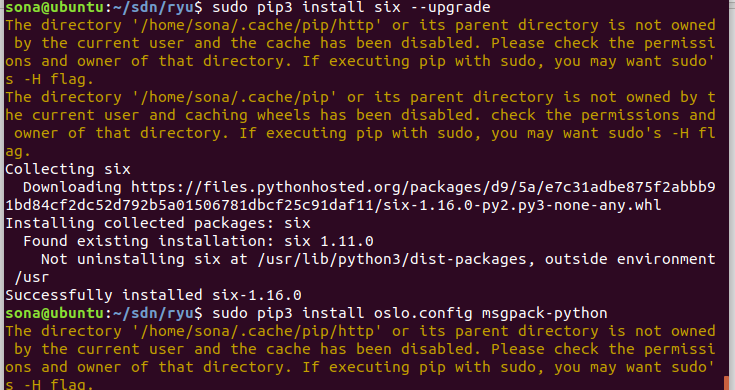


sudo python3 ./setup.py install

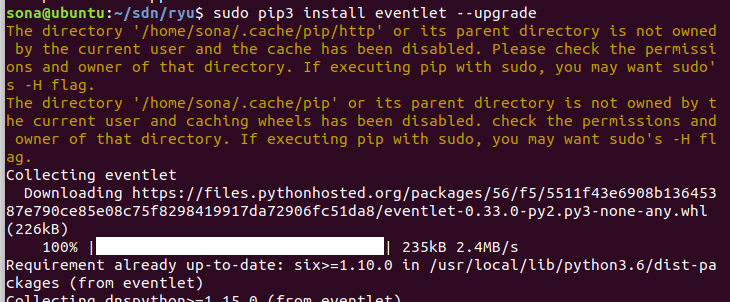


sudo pip3 install six --upgrade

sudo pip3 install oslo.config msgpack-python

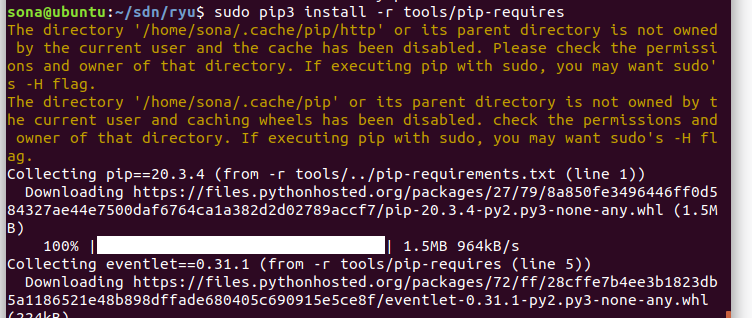


sudo pip3 install eventlet --upgrade



sudo pip3 install -r tools/pip-requires

sudo python3 setup.py install ( Do if you are not able to view ryu-manager)

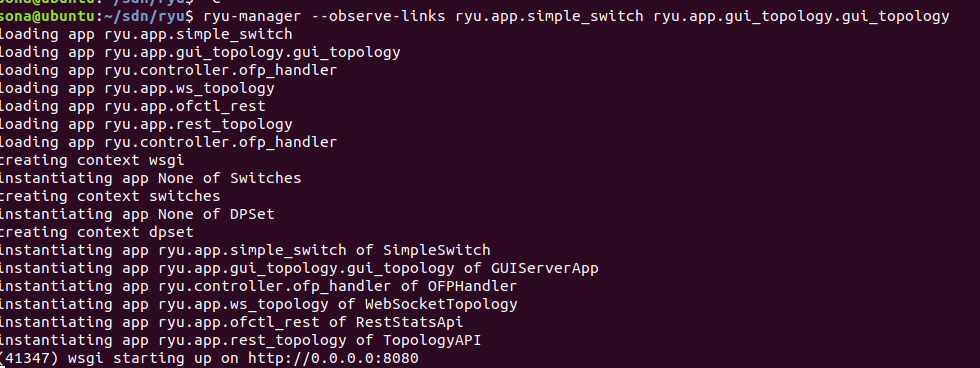


ryu-manager --version



ryu-manager --observe-links ryu.app.simple\_switch ryu.app.gui\_topology.gui\_topology

Ryu is running now dont close the terminal.



Now steps for mininet

**step 1: download VM** [**https://github.com/mininet/mininet/releases/**](https://github.com/mininet/mininet/releases/)

**Step 2: Open OVF file in the workstation.**

**Open new terminal in ubuntu**

ssh -X mininet@192.168.142.130

Above IP address is of mininet

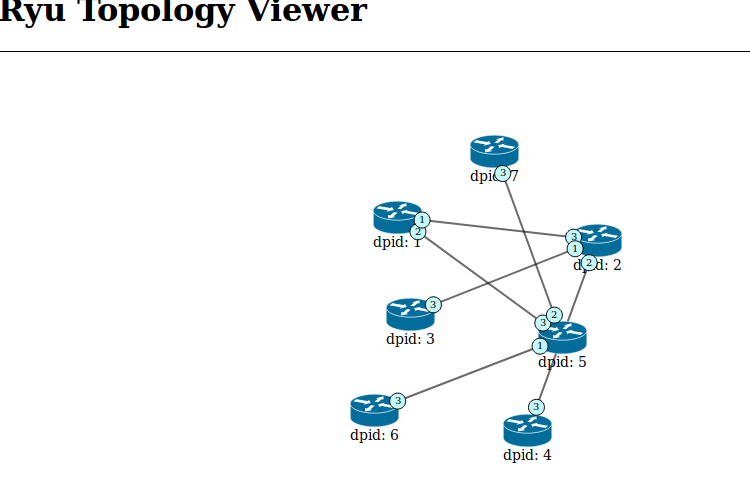
if the authority error exists then type

ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -X mininet@192.168.142.130

sudo mn --controller=remote,ip=192.168.142.131 --topo tree,depth=3

In above command ip is of ubuntu machine you can get by command ip r ( do in new terminal to get the ip)

OPEN HTTP :http://0.0.0.0:8080

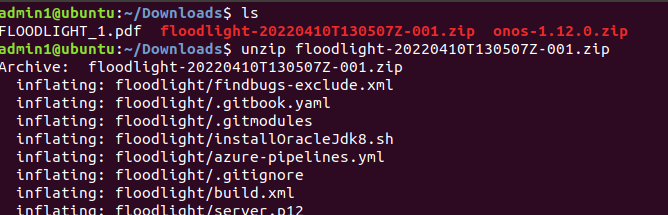


**Practical No 7**

**Install FLOODLIGHT controller with mininet topology.**

ls

unzip floodlight-20220410T130507Z-001.zip



**sudo apt install openjdk-8-jdk openjdk-8-jre**

sudo nano /etc/environment

JAVA\_HOME=”/usr/lib/jvm/java-8-openjdk-amd64”

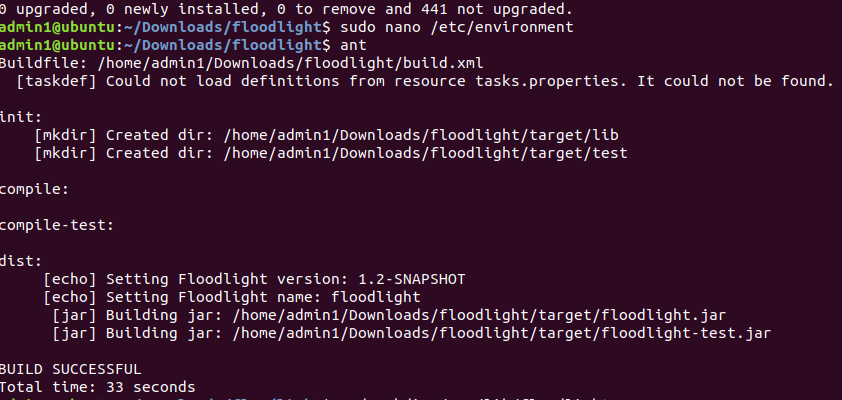
JRE\_HOME=”/usr/lib/jvm/java-8-openjdk-amd64/jre”

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export JRE\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre

sudo apt install ant

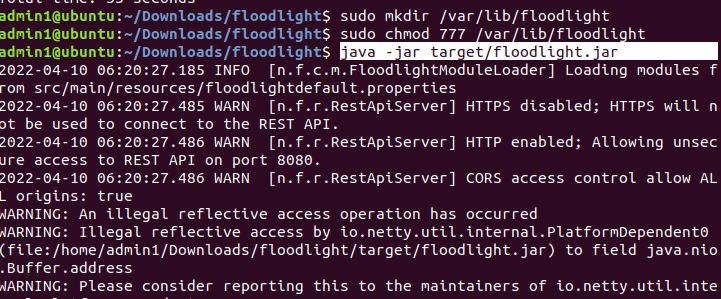
ant



sudo mkdir /var/lib/floodlight

sudo chmod 777 /var/lib/floodlight

java -jar target/floodlight.jar



Don’t close the running of jar

And Open new terminal for mininet

connect to mininet vm.



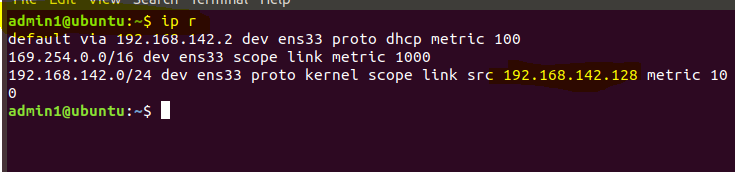
ssh -X mininet@192.168.142.130

sudo mn --controller=remote,ip=192.168.142.128 --topo=single,3

pingall

The ip address of the above remote if your ubuntu machine ip

You have to open a 3rd terminal (new terminal) and wite ip r



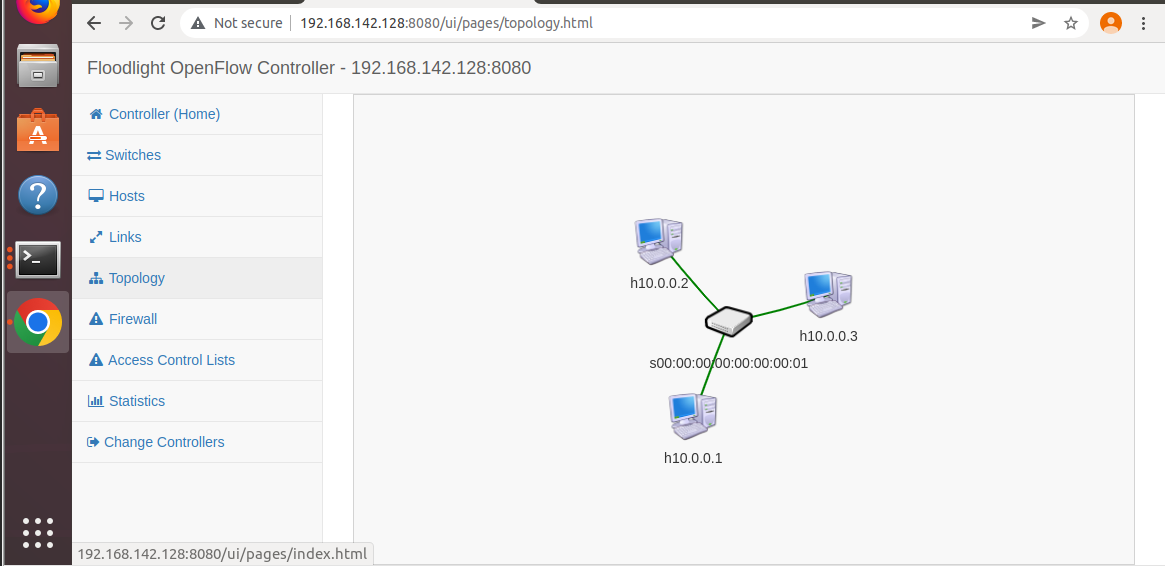


Step 6:

We created a single topology with 3 host 1 switch and one controller. This can be

view in the following URL. Go to web browser and type,

http://localhost:8080/ui/index.html



**Practical No 8**

**Install ONOS controller on ubuntu.**

**sudo apt install openjdk-8-jdk openjdk-8-jre**

**java -version**

**sudo nano /etc/environment**

Type in the first line

JAVA\_HOME=”/usr/lib/jvm/java-8-openjdk-amd64”

JRE\_HOME=”/usr/lib/jvm/java-8-openjdk-amd64/jre”

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export JRE\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre

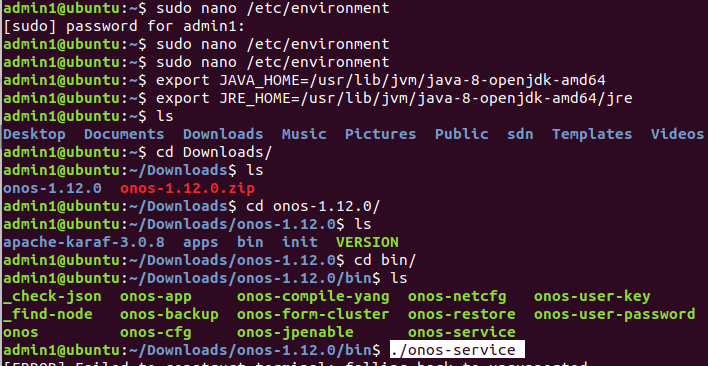
unzip onos-1.12.0.zip

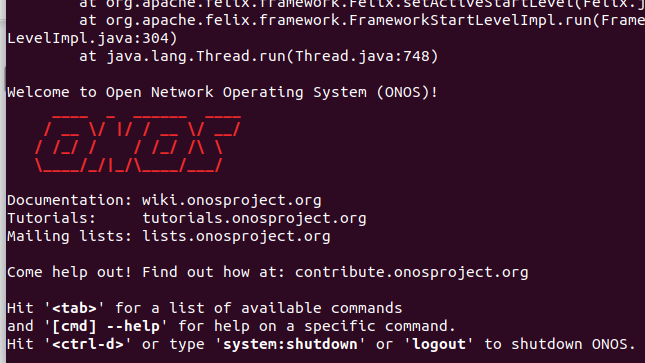
cd onos-1.12.0/

cd bin

ls

./onos-service





Go to browser in localhost:8181/onos/ui/index.html

username - onos

Password - rocks



