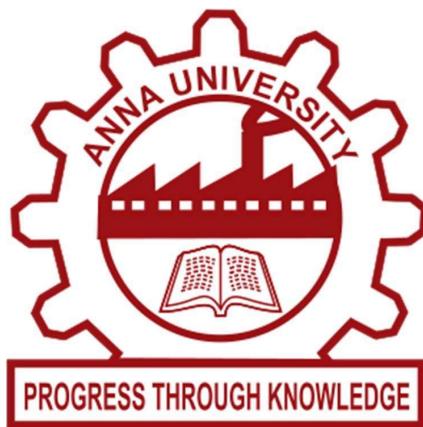


**UNIVERSITY COLLEGE OF
ENGINEERING NAGERCOIL**
(ANNA UNIVERSITY CONSTITUENT COLLEGE)
KONAM, NAGERCOIL – 629 004



RECORD NOTE BOOK

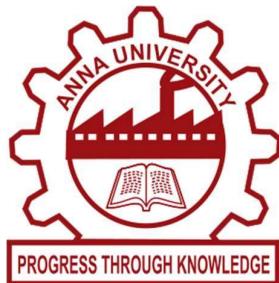
CCS365-SOFTWARE DEFINED NETWORKS

REGISTER NO:

UNIVERSITY COLLEGE OF ENGINEERING NAGERCOIL

(ANNA UNIVERSITY CONSTITUENT COLLEGE)

KONAM, NAGERCOIL – 629 004



Register No:

*Certified that, this is the bonafide record of work done by
Mr./Ms. of VI
Semester in Computer Science and Engineering of this college, in
the **CCS365-SOFTWARE DEFINED NETWORKS** during academic
year 2023-2024 in partial fulfillment of the requirements of the B.E
Degree course of the Anna University Chennai.*

Staff-in-charge

Head of the Department

This record is submitted for the University Practical Examination held on

.....

Internal Examiner

External Examiner

INDEX

Exp No	Date	Title	Page	Sign
1.	01/02/2024	Setup your own virtual SDN lab	1	
2.	11/03/2024	Create a simple mininet topology with SDN controller and use Wireshark to capture and visualize the OpenFlow messages such as OpenFlow FLOW MOD, PACKET IN, PACKET OUT etc.	8	
3.	16/03/2024	Create a SDN application that uses the Northbound API to program flow table rules on the switch for various use cases like L2 learning switch, Traffic Engineering, Firewall etc..	12	
4.	21/03/2024	Create a simple end-to-end network service with two VNFs using vim-emu.	15	
5.	28/03/2024	Install OSM and onboard and orchestrate network service.	18	

Ex No:1

Set Up Your Own Virtual SDN Lab

Date:01/02/2024

AIM:

To set up a virtual SDN Lab by Installing mininet, kathara, GNS3.

PROCEDURE:

I. Installation of Mininet:

Step 1: Start your Ubuntu terminal

```
$ sudo apt-get update
```

Step 2: Install Mininet

```
$ sudo apt-get install mininet
```

```
seeker@sarsys:~$ sudo apt-get install mininet
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cgroup-tools iperf libcgroup1 libpython2.7-stdlib libpython2.7-minimal libpython2.7-stdlib libunbound8 net-tools openvswitch-common
  openvswitch-switch python-pkg-resources python2 python2-minimal python2.7 python2.7-minimal python3-openvswitch python3-sortedcontainers socat
Suggested packages:
  openvswitch-doc python-setuptools python2-doc python-tk python2.7-doc binfmt-support python-sortedcontainers-doc
The following NEW packages will be installed:
  cgroup-tools iperf libcgroup1 libpython2.7-stdlib libpython2.7-minimal libunbound8 mininet net-tools openvswitch-common
  openvswitch-switch python-pkg-resources python2 python2-minimal python2.7 python2.7-minimal python3-openvswitch python3-sortedcontainers socat
0 upgraded, 19 newly installed, 0 to remove and 153 not upgraded.
Need to get 6,894 kB/7,942 kB of archives.
After this operation, 35.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libpython2.7-minimal amd64 2.7.18-1-20.04.4 [335 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python2.7-minimal amd64 2.7.18-1-20.04.4 [1,280 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libpython2.7-stdlib amd64 2.7.18-1-20.04.4 [1,887 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python2.7 amd64 2.7.18-1-20.04.4 [248 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libunbound8 amd64 1.9.4-2ubuntu1.5 [353 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 openvswitch-common amd64 2.13.8-0ubuntu1.4 [1,156 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-openvswitch all 2.13.8-0ubuntu1.4 [94.9 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 openvswitch-switch amd64 2.13.8-0ubuntu1.4 [1,539 kB]
Fetched 6,894 kB in 3min (38.1 kB/s)
Selecting previously unselected package libpython2.7-minimal:amd64.
(Reading database ... 185625 files and directories currently installed.)
Preparing to unpack .../0-libpython2.7-minimal_2.7.18-1-20.04.4_amd64.deb ...

```

Step 3: Check the mininet version

```
$ mn --version
```

```
update-alternatives: using /usr/lib/openvswitch-switch/ovs-vswitchd to provide /usr/sbin/ovs-vswitchd (ovs-vswitchd) in auto mode
Created symlink /etc/systemd/system/multi-user.target.wants/openvswitch-switch.service → /lib/systemd/system/openvswitch-switch.service.
Created symlink /etc/systemd/system/openvswitch-switch.service.requires/ovs-record-hostname.service → /lib/systemd/system/ovs-rec...
Setting up python2.7 (2.7.18-1-20.04.4) ...
Setting up libpython2.7-stdlib:amd64 (2.7.17-2ubuntu4) ...
Setting up cgroup-tools (0.41-10) ...
Setting up python2 (2.7.17-2ubuntu4) ...
Setting up python-pkg-resources (44.0.0-2ubuntu0.1) ...
Setting up mininet (2.2.2-Subuntu1) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.12) ...
Processing triggers for systemd (245.4-4ubuntu3.22) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
seeker@sarsys:~$ mn --version
2.2.2
```

Step 4: To test multiple switches and Openflow controllers.

```
$ sudo mn --switch ovsbr --test pingall
```

```

seeker@sarsys:~$ sudo mn --switch ovsbr --test pingball
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
-----
Caught exception. Cleaning up...

```

Step 5: If mininet complaints that open vswitch isn't working, make sure it is installed and running ,

\$ sudo apt-get install openvswitch - -switch

```

cse@cse-Veriton-M200-B365:~$ sudo mn --switch ovsbr --test pingall
[sudo] password for cse:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 0 controllers

*** Stopping 2 links
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 0.565 seconds
cse@cse-Veriton-M200-B365:~$ sudo apt-get install openvswitch --switch

```

Step 6: Now start the mininet VM by

\$ sudo mn

```

pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
seeker@sarsys:~$ sudo mn
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ...
*** Starting CLI:

```

Step 6: To display the available nodes

mininet > nodes

mininet >net

```
mininet >dump
```

```
mininet> nodes
available nodes are:
h1 h2 s1
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=23647>
<Host h2: h2-eth0:10.0.0.2 pid=23649>
<OVSBridge s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=23654>
mininet>
```

II. Installation of Kathara:

Step 1: Add the Kathara repository to your repository.

```
$ sudo add-apt-repository ppa:katharaframework/kathara
```

```
seeker@sarsys:~$ sudo add-apt-repository ppa:katharaframework/kathara
[sudo] password for seeker:
Kathará is a lightweight network emulation system based on Docker containers. It can be really helpful in showing interactive demos/lessons, testing production networks in a sandbox environment, or developing new network protocols.

Kathará is the spiritual successor of the notorious Netkit, hence it is cross-compatible, and inherits its language and features.
More info: https://launchpad.net/~katharaframework/+archive/ubuntu/kathara
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:4 http://ppa.launchpad.net/katharaframework/kathara/ubuntu focal InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2,810 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [951 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3,187 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [726 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [425 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [508 kB]
Fetched 8,834 kB in 8s (1,130 kB/s)
Reading package lists... Done
[sudo] password for seeker:
```

Step 2: Update your apt cache by running.

```
$ sudo apt update
```

```
seeker@sarsys:~$ sudo apt update
Hit:1 http://ppa.launchpad.net/katharaframework/kathara/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
153 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Step 3: Now install Kathara by

```
$ sudo apt install kathara
```

```

seeker@sarsys:~$ sudo apt install kathara
[sudo] password for seeker:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  kathara
0 upgraded, 1 newly installed, 0 to remove and 153 not upgraded.
Need to get 25.7 MB of archives.
After this operation, 146 MB of additional disk space will be used.
Get:1 http://ppa.launchpad.net/katharaframework/kathara/ubuntu focal/main amd64 kathara amd64 3.7.3-1focal [25.7 MB]
Fetched 25.7 MB in 32s (807 kB/s)
Selecting previously unselected package kathara.
(Reading database ... 188180 files and directories currently installed.)
Preparing to unpack .../kathara_3.7.3-1focal_amd64.deb ...
Unpacking kathara (3.7.3-1focal) ...
Setting up kathara (3.7.3-1focal) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.12) ...

```

Step 4: By default,kathara depends on Docker ,you can install docker by

```
$ sudo apt install docker.io
```

```

seeker@sarsys:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroups-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvinit
  git-dbg git-el git-email git-gui gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 153 not upgraded.
Need to get 68.4 MB of archives.
After this operation, 305 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [38.5 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.1.7-0ubuntu1-20.04.2 [3,836 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd amd64 1.7.2-0ubuntu1-20.04.1 [32.5 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 docker.io amd64 24.0.5-0ubuntu1-20.04.1 [26.4 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ubuntu3.11 [887 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.11 [4,605 kB]

```

Step 5: Install xterm terminal emulator is suggested.

```
$ sudo apt install xterm
```

```

seeker@sarsys:~$ sudo apt install xterm
[sudo] password for seeker:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  xfonts-cyrillic
The following NEW packages will be installed:
  xterm
0 upgraded, 1 newly installed, 0 to remove and 153 not upgraded.
Need to get 765 kB of archives.
After this operation, 2,329 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 xterm amd64 353-1ubuntu1.20.04.2 [765 kB]
Fetched 765 kB in 2s (490 kB/s)
Selecting previously unselected package xterm.
(Reading database ... 188204 files and directories currently installed.)
Preparing to unpack .../xterm_353-1ubuntu1.20.04.2_amd64.deb ...
Unpacking xterm (353-1ubuntu1.20.04.2) ...
Setting up xterm (353-1ubuntu1.20.04.2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menu (3.36.0-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
seeker@sarsys:~$ 

```

Step 6: Checking the Kathara version by

```
$ kathara --version
```

```

seeker@sarsys:~$ kathara --version
Current version: 3.7.3
seeker@sarsys:~$ docker --version
Docker version 24.0.5, build 24.0.5-0ubuntu1~20.04.1
seeker@sarsys:~$ 

```

Step 7: Starting kathara can be done by

```
$ sudo kathara vstart -n name
```

III. Installation of GNS3

Step 1: Add the gns3 repository to your repository.

```
$ sudo add-apt-repository ppa:gns3/ppa
```

```
cse@cse-OptiPlex-9010:~$ sudo add-apt-repository ppa:gns3/ppa
[sudo] password for cse:
12 PPA for GNS3 and Supporting Packages. Please see http://www.gns3.com for more details
More info: https://launchpad.net/~gns3/+archive/ubuntu/ppa
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ppa.launchpad.net/gns3/ppa/ubuntu focal InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://ppa.launchpad.net/katharaframework/kathara/ubuntu focal InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:6 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu focal InRelease
Hit:7 https://download.docker.com/linux/ubuntu focal InRelease
Hit:8 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
cse@cse-OptiPlex-9010:~$ 
```

Step 2: Update the version installed

```
$ sudo apt update
```

```
Reading package lists... done
cse@cse-OptiPlex-9010:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://ppa.launchpad.net/gns3/ppa/ubuntu focal InRelease
Hit:6 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:7 http://ppa.launchpad.net/katharaframework/kathara/ubuntu focal InRelease
Hit:8 http://ppa.launchpad.net/wireshark-dev/stable/ubuntu focal InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
89 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Step 3:Install GNS3

```
$sudo apt install gns3-gui gns3-server
```

```
cse@cse-OptiPlex-9010:~$ sudo apt install gns3-gui gns3-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
gns3-gui is already the newest version (2.2.46~focal1).
gns3-server is already the newest version (2.2.46~focal1).
The following packages were automatically installed and are no longer required:
  bridge-utils gir1.2-goa-1.0 linux-headers-5.15.0-87-generic
  linux-hwe-5.15-headers-5.15.0-87 linux-image-5.15.0-87-generic
  linux-modules-5.15.0-87-generic linux-modules-extra-5.15.0-87-generic
  ubuntu-fan
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 89 not upgraded.
cse@cse-OptiPlex-9010:~$
```

Step 12: To install docker ce

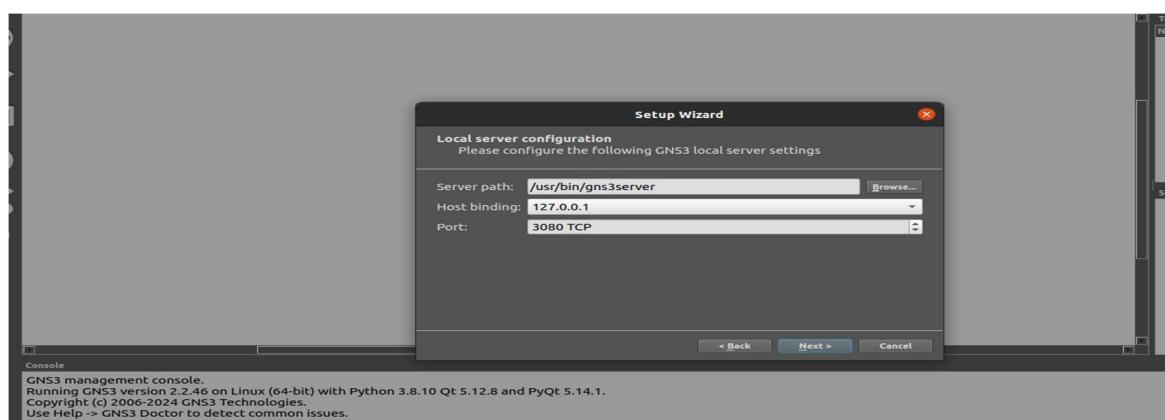
```
$ sudo apt install docker-ce
```

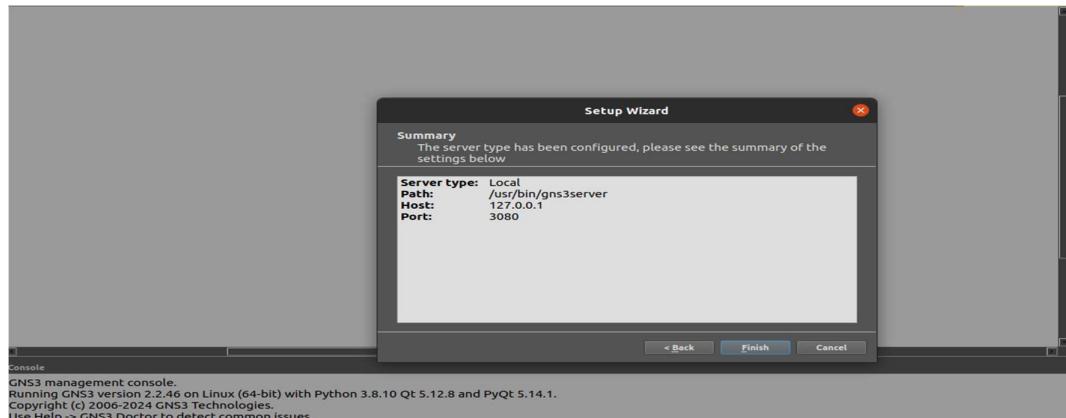
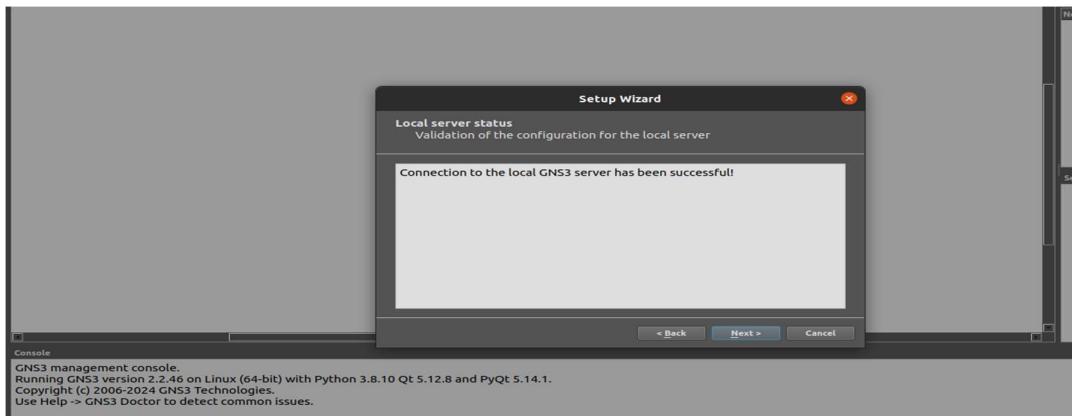
```
cse@cse-OptiPlex-9010:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker-ce is already the newest version (5:25.0.3-1~ubuntu.20.04~focal).
The following packages were automatically installed and are no longer required:
  bridge-utils gir1.2-goa-1.0 linux-headers-5.15.0-87-generic
  linux-hwe-5.15-headers-5.15.0-87 linux-image-5.15.0-87-generic
  linux-modules-5.15.0-87-generic linux-modules-extra-5.15.0-87-generic
  ubuntu-fan
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 89 not upgraded.
cse@cse-OptiPlex-9010:~$
```

Step 13: Start GNS3

```
$ gns3
```

```
cse@cse-OptiPlex-9010:~$ gns3
your locale en_IN.ISO8859-1 encoding is not UTF-8, switching to the UTF-8 version...
2024-03-04 10:30:12 INFO root:126 Log level: INFO
2024-03-04 10:30:12 INFO main:265 GNS3 GUI version 2.2.46
2024-03-04 10:30:12 INFO main:266 Copyright (c) 2007-2024 GNS3 Technologies Inc.
2024-03-04 10:30:12 INFO main:267 Application started with /usr/bin/gns3
```





RESULT:

Thus the virtual box environment for SDN Lab was successfully completed.

Ex No: 2 Date:11/03/2024	Create a simple mininet topology with SDN Controller and use Wireshark to capture and visualize the Openflow messages such as Openflow FLOW MODE,PACKET IN,PACKET OUT etc.
---	---

AIM:

1. Create a simple Mininet network with an SDN controller.
2. Capture OpenFlow messages, including FLOW_MOD, PACKET_IN, and PACKET_OUT, using Wireshark.
3. Visualize the captured OpenFlow messages for analysis.

PROCEDURE:

1. Install pox controller:

Install the pox controller in to your system.

```
$ sudo apt-get install git git clone https://github.com/noxrepo/pox.git
```

```
cse@cse-Veriton-M200-B365:~$ git clone https://github.com/noxrepo/pox.git
Cloning into '\github.com\noxrepo\pox'...
ssh: Could not resolve hostname https: Name or service not known
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
cse@cse-Veriton-M200-B365:~$ cd pox
cse@cse-Veriton-M200-B365:~/pox$ cd pox
```

2. Start POX Controller:

Navigate to the POX directory and start the POX controller. You can use the ‘pox.py’ script along with a specific module. For example, you can use the ‘forwarding.l2_learning’ module.

```
$ cd pox
```

```
./pox.py forwarding.l2_learning
```

```
desrtile      Karan-0.16.1 oflops          pox      templates
cse@cse-OptiPlex-9010:~$ cd pox
cse@cse-OptiPlex-9010:~/pox$ ./pox.py forwarding.l2_learning
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
WARNING:version:Support for Python 3 is experimental.
INFO:core:POX 0.7.0 (gar) is up.
INFO:openflow.of_01:[00-00-00-00-00-01 2] connected
```

4. Install Wireshark:

Install Wireshark using the following command.

```
$ sudo apt-get install wireshark
```

```

seeker@sarsys:~/pox$ sudo apt-get install wireshark
[sudo] password for seeker:
Reading package lists... done
Building dependency tree
Reading state information... done
The following additional packages will be installed:
libc-ares2 libdouble-conversion3 liblbus5-2-0 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins
libqt5multimedialddgets5 libqt5multimedialddgets5 libqt5networks libqt5opengl5 libqt5printsupport5 libqt5svg5 libqt5widgts5 libsmi2lidl
libsnapy1v5 libspandsp2 libssh-gcrypt-4 libwireshark-data libwireshark13 libwireshark13 libwretapi0 libwsutll1 libxcb-xinerama0 libxcb-xinput0
qts-gtk-platformtheme qtstranslations-lion wireshark-common wireshark-qt
Suggested packages:
qts-image-formats-plugins qtwaylands snmp-mibs-downloader geotupdate geolp-database geolp-database-extra libjs-leaflet
libjs-leaflet.markercluster wireshark-doc
The following NEW packages will be installed:
libc-ares2 libdouble-conversion3 liblbus5-2-0 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins
libqt5multimedialddgets5 libqt5multimedialddgets5 libqt5networks libqt5opengl5 libqt5printsupport5 libqt5svg5 libqt5widgts5 libsmi2lidl
libsnapy1v5 libspandsp2 libssh-gcrypt-4 libwireshark-data libwireshark13 libwretapi0 libwsutll1 libxcb-xinerama0 libxcb-xinput0
qts-gtk-platformtheme qtstranslations-lion wireshark-common wireshark-qt
0 upgraded, 31 newly installed, 0 to remove and 153 not upgraded.
Need to get 0/832.9 MB of archives.
After this operation, 164 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Extracting templates from packages: 100%
Preconfiguring packages ...
Selecting previously unselected package libdouble-conversion3:amd64.
(Reading database ... 1888 packages and 0 direct dependencies are currently installed.)
Preparing to unpack .../00-libdouble-conversion3_3.1.5-4ubuntu1_amd64.deb ...
Unpacking libdouble-conversion3:amd64 (3.1.5-4ubuntu1) ...
Selecting previously unselected package libpcre2-16-0:amd64.

```

3. Start Mininet with Openflow support:

Use Mininet's command-line interface to create a network topology. Here a simple topology with one switch and two hosts will be created.

```
$ sudo mn --controller=remote,ip=127.0.0.1,port=6633 --topo single,2
```

```

seeker@sarsys:~/pox$ sudo mn --controller=remote,ip=127.0.0.1,port=6633 --topo single,2
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> 
```

4. Start Mininet:

Ensure you have Mininet installed on your system. You can use a Linux distribution for this task.

```
$ sudo mn
```

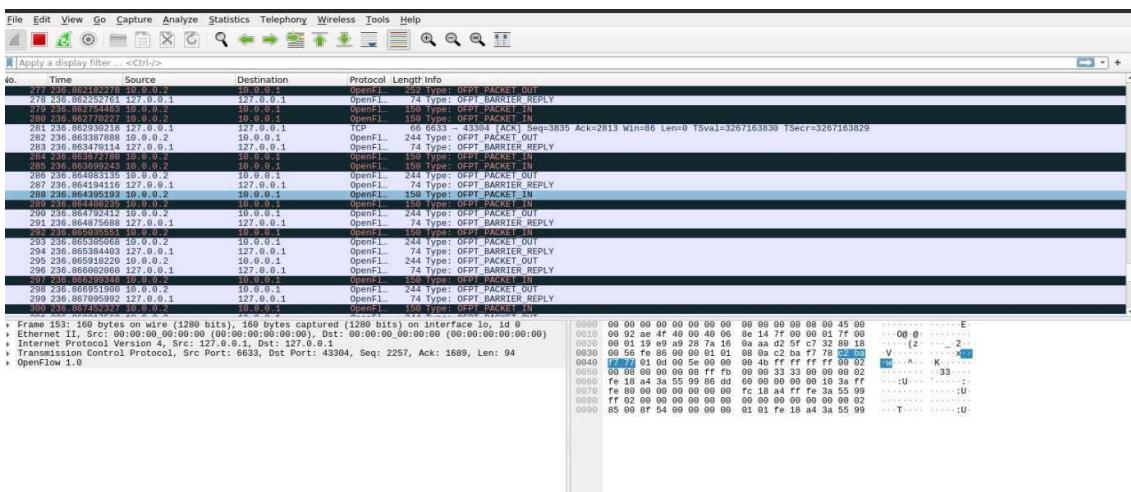
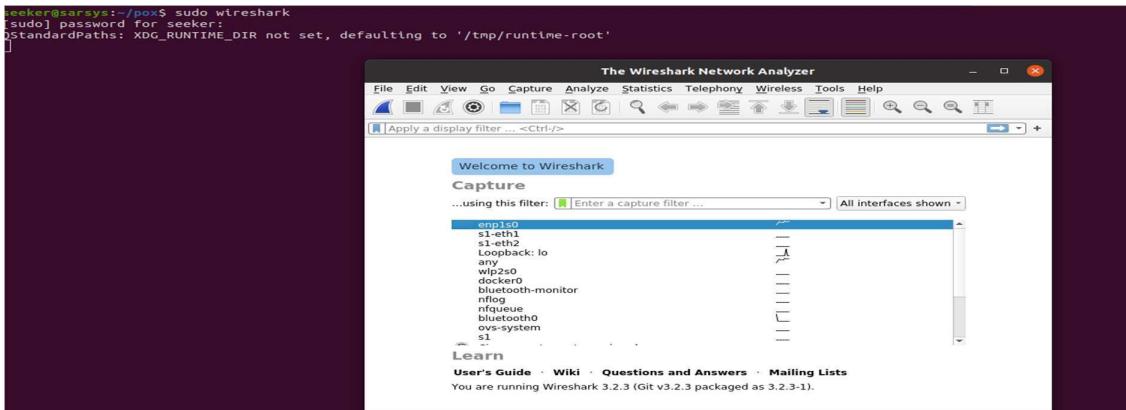
```

[cse@OptiPlex-9010:~/pox$ sudo mn
[sudo] password for cse:
Sorry, try again.
[sudo] password for cse:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['19.9 Gbits/sec', '19.9 Gbits/sec']
mininet> 
```

5. Capture OpenFlow Messages with Wireshark:

- Start Wireshark and select your network interface (e.g., 'eth0').
- Begin capturing packets by clicking the "Start" button in Wireshark.

```
$ sudo wireshark
```



6. Generate OpenFlow Messages:

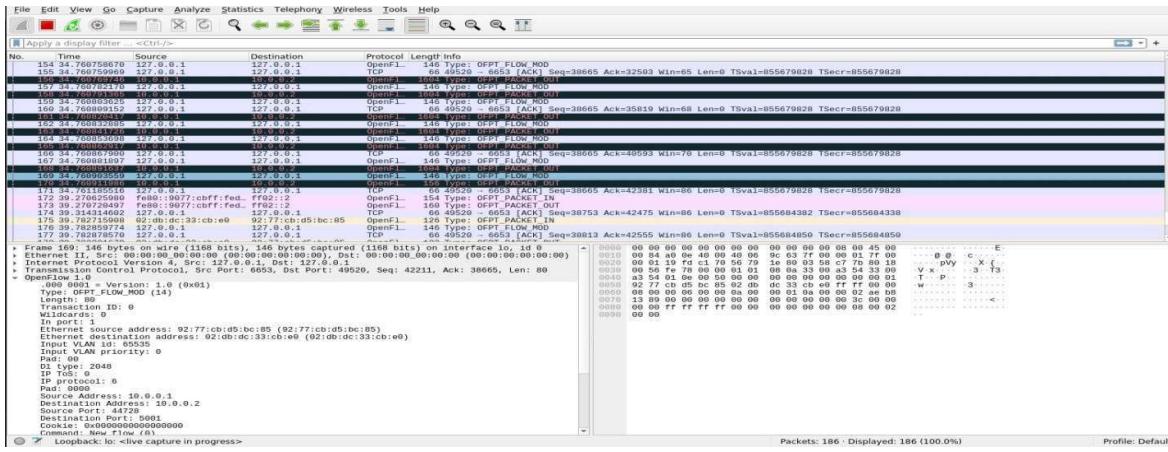
In the Mininet terminal, you can use the Mininet CLI to generate OpenFlow messages. For example, you can add a flow rule (FLOW_MOD) or generate traffic (PACKET_OUT).

```
unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
[mininet]
Interrupt
[mininet]
Interrupt
[mininet]
Interrupt
[mininet]
node
*** Unknown command: node
[mininet]
Available nodes:
@ h1 h2 s1
[mininet]
net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
[mininet] dump
<Host h1: h1-eth0:10.0.0.1 pid=18600>
<Host h2: h2-eth0:10.0.0.2 pid=18602>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=18607>
<RemoteController['ip': '127.0.0.1', 'port': 6633] c0: 127.0.0.1:6633 pid=18594>
[mininet] 
```

```

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
mininet> s1 if config -a
> h1 if config -a
> h1 if config -a
> h1 ps -a
> -----
Caught exception. Cleaning up...
error: (4, 'Interrupted system call')
-----
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofproto.ofdatapath ping nox_core_1t-nox_core ovs-openflow ovs-controller udpbwtest mnexec ivs 2> /dev/null
killall -9 controller ofproto.ofdatapath ping nox_core_1t-nox_core ovs-openflow ovs-controller udpbwtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
rm -rf /tmp/ovs*/vlogs* /tmp/*.* /tmp/*.*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | grep -o 'dp[0-9]*' | sed 's/dp/nl:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
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 del si1-eth1;ip link del si1-eth2 ) 2> /dev/null
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f TunnelEthernet
pkill -9 -f .ssh/mn
rm -rf .ssh/mn/
*** Cleanup complete.
seeker@sarsys:~/pox$ 

```



7. Stop Wireshark Capture:

Stop capturing packets in Wireshark when you've generated enough OpenFlow messages.

RESULT:

Thus to create a simple mininet topology with SDN Controller and use wireshark to capture and visualize the openflow messages such as Openflow FLOW MODE,packet in,packet out etc.

Ex No : 3 Date: 16/03/2024	Create a SDN application that uses the Northbound API program flow table rules on the switch for various use cases like L2 learning switch, Traffic Engineering, Firewall etc.
---	---

AIM:

Develop an SDN application that uses the Northbound API to program flow table rules on SDN switches for different use cases: L2 learning switch, Traffic Engineering, and Firewall.

PROCEDURE:

1. Install POX controller:

Install the pox controller in to your system.

```
$ sudo apt-get install git git clone https://github.com/noxrepo/pox.git
```

```
seeker@sarsys:~$ git clone https://github.com/noxrepo/pox.git
Cloning into 'pox'...
remote: Enumerating objects: 13058, done.
remote: Counting objects: 100% (283/283), done.
remote: Compressing objects: 100% (134/134), done.
remote: Total 13058 (delta 174), reused 242 (delta 144), pack-reused 12775
Receiving objects: 100% (13058/13058), 5.02 MiB | 2.78 MiB/s, done.
Resolving deltas: 100% (8423/8423), done.
seeker@sarsys:~$ █
```

2. Start POX Controller:

Navigate to the POX directory and start the POX controller. You can use the ‘pox.py’ script along with a specific module. For example, you can use the ‘forwarding.l2_learning’ module.

```
$ cd pox
```

```
./pox.py forwarding.l2_learning
```

```
desrite      Karan-0.1.0.1 or tops          pox           templates
cse@cse-OptiPlex-9010:~/pox$ ./pox.py forwarding.l2_learning
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
WARNING:version:Support for Python 3 is experimental.
INFO:core:POX 0.7.0 (gar) is up.
INFO:openflow.of_01:[00-00-00-00-00-01 2] connected
█
```

3. Set Up the Development Environment:

The following command is used to create a simple software-defined network (SDN) topology for testing and development purposes using Mininet, a network emulator. It configures a network consisting of one Open vSwitch (OVS) switch connected to three hosts.

```
$ sudo mn --topo single,3 --mac --arp --switch ovsk --controller=remote
```

```

seeker@sarsys:~/pox$ sudo mn --topo single,3 --mac --arp --switch ovsk --controller=remote
[sudo] password for seeker:
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6633
*** 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> █

```

2. Network Testing and Performance Measurement with Mininet:

This sequence of commands in Mininet includes opening an xterm terminal for host h1, conducting a connectivity check between all hosts using pingall, and then measuring network performance between hosts h1 and h2 using iperf.

```

mininet> xterm h1
mininet> pingall
mininet>iperf h1 h2

```

```

cse@cse-OptiPlex-9010:~/pox$ sudo mn --topo single,3 --mac --arp --switch ovsk --controller=remote
[sudo] password for cse:
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
('Rhythmbox', 2, s1) (h3, s1)
* Rhythmbox Ring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> xterm h1
mininet> pingall
* pingall: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*+ Results: 0% dropped (6/6 received)
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
... Results: ['7.6 Gbits/sec', '7.6 Gbits/sec']
mininet> iperf h2 h1
*** Iperf: testing TCP bandwidth between h1 and h2
... Results: ['7.2 Gbits/sec', '7.2 Gbits/sec']
mininet> █

```

4. Start Mininet:

Ensure you have Mininet installed on your system. You can use a Linux distribution for this task.

\$ sudo mn

```

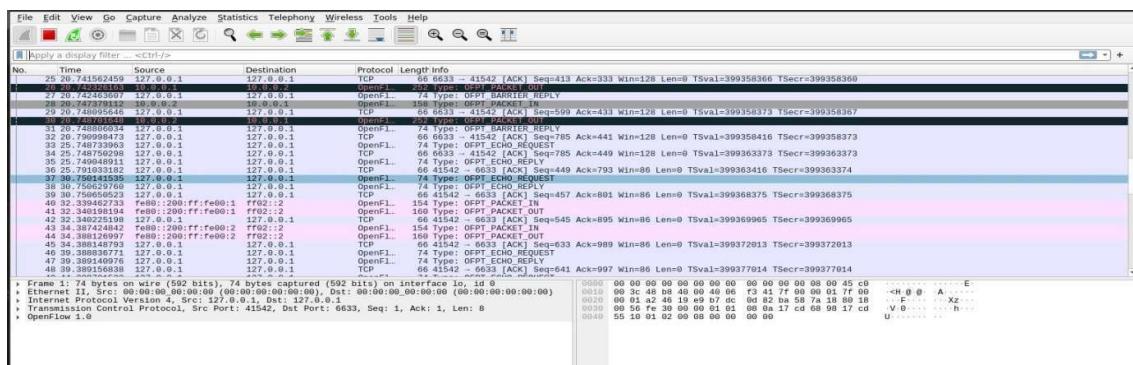
cse@cse-OptiPlex-9010:~/pox$ sudo mn
[sudo] password for cse:
Sorry, try again.
[sudo] password for cse:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
... Results: ['19.9 Gbits/sec', '19.9 Gbits/sec']
mininet> █

```

5. Start Wireshark:

- Start Wireshark and select your network interface (e.g., 'eth0').
- Begin capturing packets by clicking the "Start" button in Wireshark.

```
cse@cse-OptiPlex-9010:~/pox$ sudo wireshark
[sudo] password for cse:
** (wireshark:4576) 09:25:55.976315 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:4576) 09:26:05.718814 [Capture MESSAGE] -- Capture Start ...
** (wireshark:4576) 09:26:05.789521 [Capture MESSAGE] -- Capture started
** (wireshark:4576) 09:26:05.789624 [Capture MESSAGE] -- File: "/tmp/wireshark_loEW4SK2.pcapng"
```



7. Stop Wireshark Capture:

Stop capturing packets in Wireshark when API program flow table rules on the switch for various use cases are created.

RESULT:

Thus to create a SDN application that uses the Northbound API program flow table rule the switch for various use cases like L2 learning switch, Traffic Engineering, Firewall etc. was completed.

Ex NO: 4

Date:21/03/2024

Create a simple end-to-end network service with two VNFs using vim-emu.

AIM:

To Create a simple end-to-end network service with two VNFs using vim-emu.

PROCEDURE:

1. Set Up the Environment:

Make sure you have a Linux system. You can set up a virtual machine or a dedicated system.

Install the required dependencies, including Docker and Docker Compose.

```
$ echo install docker
```

```
$ sudo apt install docker-ce
```

```
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~$ echo install docker
install docker
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following packages will be upgraded:
  docker-ce
1 upgraded, 0 newly installed, 0 to remove and 113 not upgraded.
Need to get 25.3 MB of archives.
After this operation, 621 kB of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd6
4 5:26.1.3-1~ubuntu.22.04~jammy [25.3 MB]
Fetched 25.3 MB in 24s (1,071 kB/s)
(Reading database ... 256137 files and directories currently installed.)
Preparing to unpack .../docker-ce_5%3a26.1.3-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-ce (5:26.1.3-1~ubuntu.22.04~jammy) over (5:26.0.0-1~ubuntu.22.0
4~jammy) ...
Setting up docker-ce (5:26.1.3-1~ubuntu.22.04~jammy) ...
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~$
```

2. To Start the Docker:

Start the installed docker

```
$ echo start docker
```

After installing and starting Docker, it is important to verify that the Docker service is running correctly on your system. On Linux systems, you can use the systemctl command to check the status of the Docker service.

```
$ sudo systemctl status docker
```

```
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~$ echo start docker
start docker
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset=enabled)
   Active: active (running) since Mon 2024-05-20 11:20:56 IST; 44s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
      Main PID: 4706 (dockerd)
        Tasks: 8
       Memory: 27.5M
          CPU: 392ms
        CGroup: /system.slice/docker.service
                  └─4706 /usr/bin/dockerd -H fd:// --containerid=/run/containerd/containerd.sock

May 20 11:20:56 poenixcoder-HP-2000-Notebook-PC dockerd[4706]: time="2024-05-20T11:20:56.000000000+05:30" level=info msg="Starting containerd"
May 20 11:20:56 poenixcoder-HP-2000-Notebook-PC dockerd[4706]: time="2024-05-20T11:20:56.000000000+05:30" level=info msg="Starting containerd"
May 20 11:20:56 poenixcoder-HP-2000-Notebook-PC dockerd[4706]: time="2024-05-20T11:20:56.000000000+05:30" level=info msg="Starting containerd"
May 20 11:20:56 poenixcoder-HP-2000-Notebook-PC dockerd[4706]: time="2024-05-20T11:20:56.000000000+05:30" level=info msg="Starting containerd"
May 20 11:20:56 poenixcoder-HP-2000-Notebook-PC dockerd[4706]: time="2024-05-20T11:20:56.000000000+05:30" level=info msg="Starting containerd"
May 20 11:20:56 poenixcoder-HP-2000-Notebook-PC dockerd[4706]: time="2024-05-20T11:20:56.000000000+05:30" level=info msg="Starting containerd"
May 20 11:20:56 poenixcoder-HP-2000-Notebook-PC dockerd[4706]: time="2024-05-20T11:20:56.000000000+05:30" level=info msg="Starting containerd"
May 20 11:20:56 poenixcoder-HP-2000-Notebook-PC dockerd[4706]: time="2024-05-20T11:20:56.000000000+05:30" level=info msg="Starting containerd"
```

3. Installing vim-emu Using Nested Docker:

vim-emu is an emulator for network functions virtualization (NFV), enabling the simulation of network environments for testing and development. Running vim-emu using nested Docker allows you to create and manage the virtualized networks within a containerized setup, providing flexibility and isolation.

\$ echo install vim-emu using nested docker

```
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~$ echo install vim-emu using nested
docker
install vim-emu using nested docker
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~$ git clone https://osm.etsi.org/ge
rrit/osm/vim-emu.git
Cloning into 'vim-emu'...
remote: Counting objects: 167, done
```

4. Clone into vim-emu:

Clone the ‘vim-emu’ repository from ETSI’s Git repository.

\$ git clone <https://osm.etsi.org/gerrit/osm/vim-emu.git>

```
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~$ git clone https://osm.etsi.org/ge
rrit/osm/vim-emu.git
Cloning into 'vim-emu'...
remote: Counting objects: 167, done
remote: Finding sources: 100% (3/3)
remote: Total 6173 (delta 0), reused 6173 (delta 0)
Receiving objects: 100% (6173/6173), 1.72 MiB | 233.00 KiB/s, done.
Resolving deltas: 100% (3622/3622), done.
```

5. Start vim-emu Controller:

Navigate to the vim-emu directory and start the vim-emu controller. After cloning the vim-emu repository, the next step is to build the Docker image for vim-emu. This image will include all the necessary components to run the emulator in a Docker container.

\$ cd ~/vim-emu

\$ docker build -t vim-emu-img

```
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~$ git clone https://osm.etsi.org/ge
rrit/osm/vim-emu.git
Cloning into 'vim-emu'...
remote: Counting objects: 167, done
remote: Finding sources: 100% (3/3)
remote: Total 6173 (delta 0), reused 6173 (delta 0)
Receiving objects: 100% (6173/6173), 1.72 MiB | 233.00 KiB/s, done.
Resolving deltas: 100% (3622/3622), done.
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~/vim-emu$
```

```
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~/vim-emu$ docker build -t vim-emu-i
mg .
```

6. To Start Containernet :

Containernet is an extension of the Mininet network emulator that allows you to create virtualized network topologies using Docker containers.

```
$ echo start the containernet
```

```
$ docker run --name vim-emu -it -rn --privileged --pid='host' -v /var/run/docker.sock:/var/run/docker.sock vim-emu-img/bin/bash
```

```
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~/vim-emu$ echo start the container
start the container
poenixcoder@poenixcoder-HP-2000-Notebook-PC:~/vim-emu$ docker run --name vim-emu -it --rm --privileged --pid='host' -v /var/run/docker.sock:/var/run/docker.sock vim-emu-img /bin/bash
+ exec /containernet/util/docker/entrypoint.sh /bin/bash
* /etc/openvswitch/conf.db does not exist
* Creating empty database /etc/openvswitch/conf.db
* Starting ovsdb-server
* Configuring Open VSwitch system IDs
* Starting ovs-vswitchd
* Enabling remote OVSDB managers
Pulling the "ubuntu:trusty" and "ubuntu:xenial" image for later use...
```

```
openixcoder@openixcoder-HP-2000-Notebook-PC:~/vim-emu$ echo start the container
start the container
openixcoder@openixcoder-HP-2000-Notebook-PC:~/vim-emu$ docker run --name vim-emu -it --rm --privileged --pid='host' -v /var/run/docker.sock:/var/run/docker.sock -v /bin/bash
exec /containernet/util/docker/entrypoint.sh /bin/bash
* /etc/openvswitch/conf.db does not exist
* Creating empty database /etc/openvswitch/conf.db
* Starting ovsdb-server
* Configuring Open vSwitch system IDs
* Starting ovsdb-vtchd
* Enabling remote OVSDB managers
Pulling the "ubuntu:trusty" and "ubuntu:xenial" image for later use...
trusty: Pulling from library/ubuntu
2e6e20c8e2e6: Pull complete
0551a797c01d: Pull complete
512123a864da: Pull complete
Digest: sha256:64483f3496c1373bf55348e88694d1c4d0c9b660dee6bfe5e12f43b9933b30
ubuntu:trusty: Pulling from library/ubuntu
ubuntu:trusty: Image is up to date for ubuntu:trusty
docker.io/library/ubuntu:trusty
xenial: Pulling from library/ubuntu
58690f9b18fc: Pull complete
b51569e7c507: Pull complete
da8ef40b9eca: Pull complete
fb15d46c38dc: Pull complete
Digest: sha256:1f1a2d50de1d604801a9671f301190704c25d604a416f59e03c04f5c6ffee0d
ubuntu: xenial: Pulling newer image for ubuntu:xenial
docker.io/library/ubuntu:xenial
Welcome to ContainerNet running within a Docker container ...
root@3dd04e948a9:~/vim-emu#
```

RESULT:

Thus Creating a simple end-to-end network service with two VNFs using vim-emu was successfully completed.

EX No:5
Date:28/03/2024

Install OSM and onboard and orchestrate network service.

AIM:

To Install OSM and onboard and orchestrate network service.

PROCEDURE:

1. Install OSM:

Follow the instructions on the OSM GitHub repository (<https://osm.etsi.org>) to install OSM, which includes installing the OSM client and server components.

2. Clone OSM Repositories:

Clone the OSM repositories from their GitLab:

```
$ git clone https://osm.etsi.org/gerrit/osm/osm.git
```

3. Onboard the Network Service:

Once your descriptor files are validated, onboard your network service to the OSM platform. This process registers your service with OSM and makes it available for deployment and orchestration.

```
osm ns-create --nsd=my_network_service --ns_name=my_service_instance
```

4. Verify Onboarding:

Once the command is executed successfully, verify that your network service has been onboarded by checking the OSM platform or using the CLI to list the available network services.

```
osm ns-create --nsd=my_network_service --ns_name=my_service_instance
```

Replace '<ns-instance-name>' with a suitable name for your instantiated service.

5. Instantiate the Network services:

Instantiate the network service using OSM

```
osm ns-instantiate <ns-instance-name> <your_network_service_name>
```

Replace '<ns-instance-name>' with a suitable name for your instantiated service.

RESULT:

Thus the Installation of OSM and onboard and orchestrate network service was successfully completed.