SDN DEMO PREP

Iteration 1

By

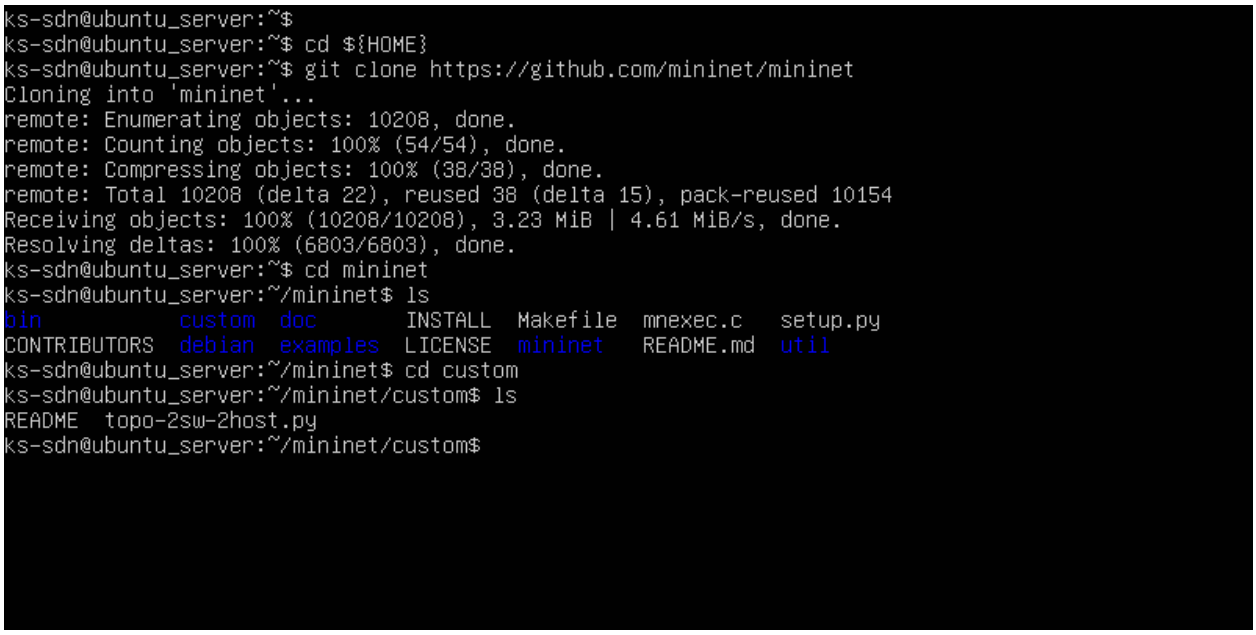
Kabiru Sanusi

Hands-on Python Net Apps

Description: Run, analyze, and interpret Python NetApps
Introduction: Clone the mininet GIT repo to your home dir, then the ${HOME}/mininet/custom directory should hold configuration files for custom mininets. See **topo-2sw-2host.py,** which loads the default minimal topology.

Commands Steps:
$ cd ${HOME}
$ git clone https://github.com/mininet/mininet
$ cd mininet
$l s
$ cd custom
$ ls
$ sudo mn –custom topo-2sw-2host.py –topo mytopo

Output:



```
*** 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.
ks-sdn@ubuntu_server:~/mininet/custom$ ls
README  topo-2sw-2host.py
ks-sdn@ubuntu_server:~/mininet/custom$ sudo fuser -k 6653/tcp
6653/tcp:             1575
ks-sdn@ubuntu_server:~/mininet/custom$
ks-sdn@ubuntu_server:~/mininet/custom$ sudo mn --custom topo-2sw-2host.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> _
```

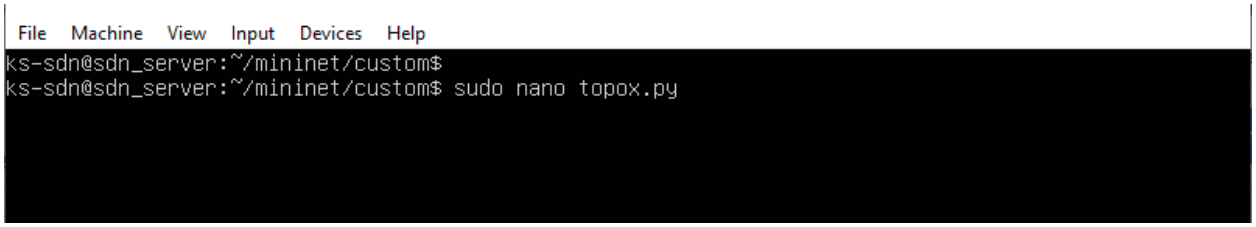
```
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s4-eth2
s3 lo:  s3-eth1:h1-eth0 s3-eth2:s4-eth1
s4 lo:  s4-eth1:s3-eth2 s4-eth2:h2-eth0
c0
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=2521>
<Host h2: h2-eth0:10.0.0.2 pid=2523>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=2528>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=2531>
<OVSController c0: 127.0.0.1:6653 pid=2514>
mininet>
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s3 s4
*** Stopping 2 hosts
h1 h2
*** Done
completed in 147.584 seconds
ks-sdn@ubuntu_server:~/mininet/custom$
```

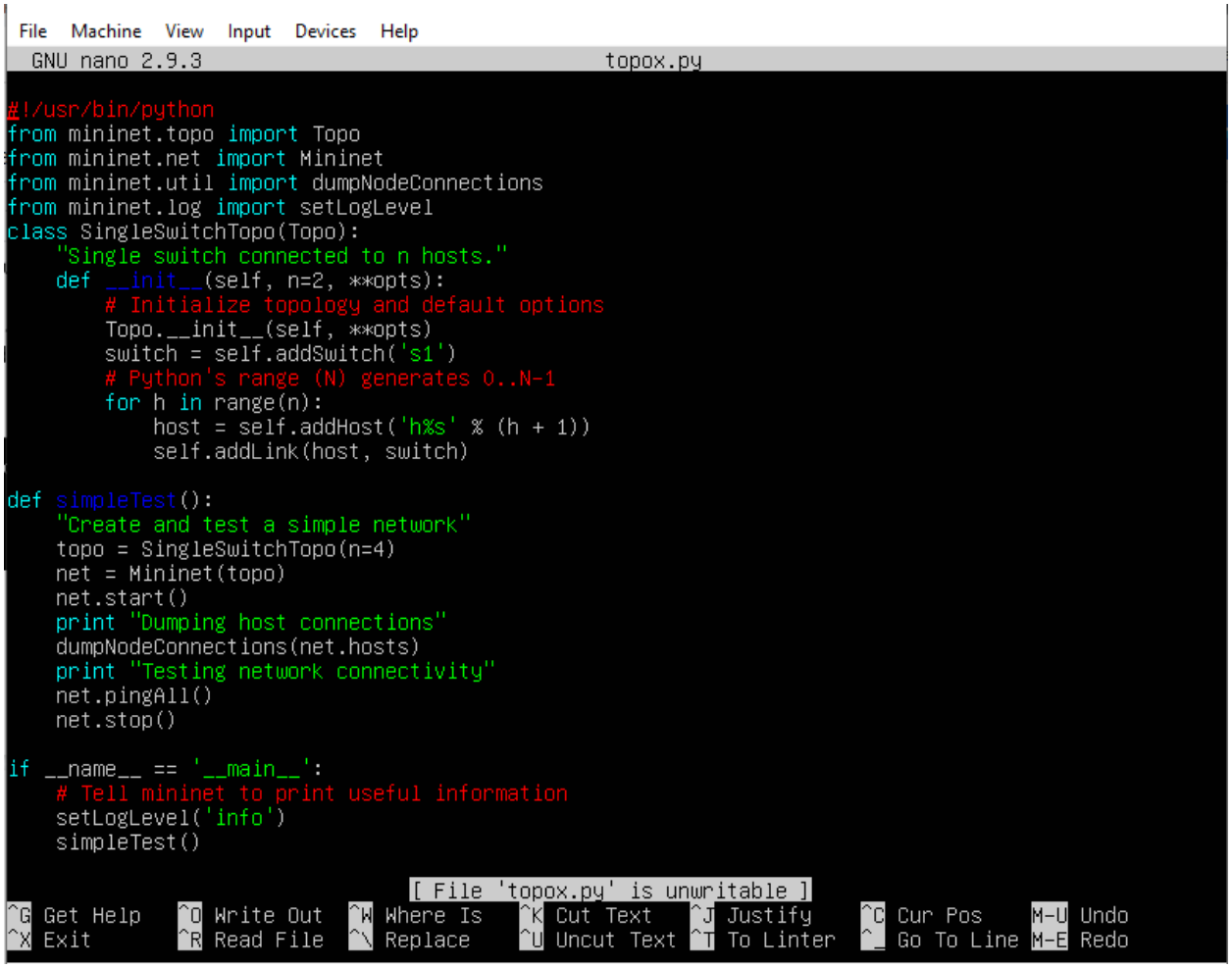
Note: Run the command below to kill tcp pocess running on port 6653 if it fails to create the topology

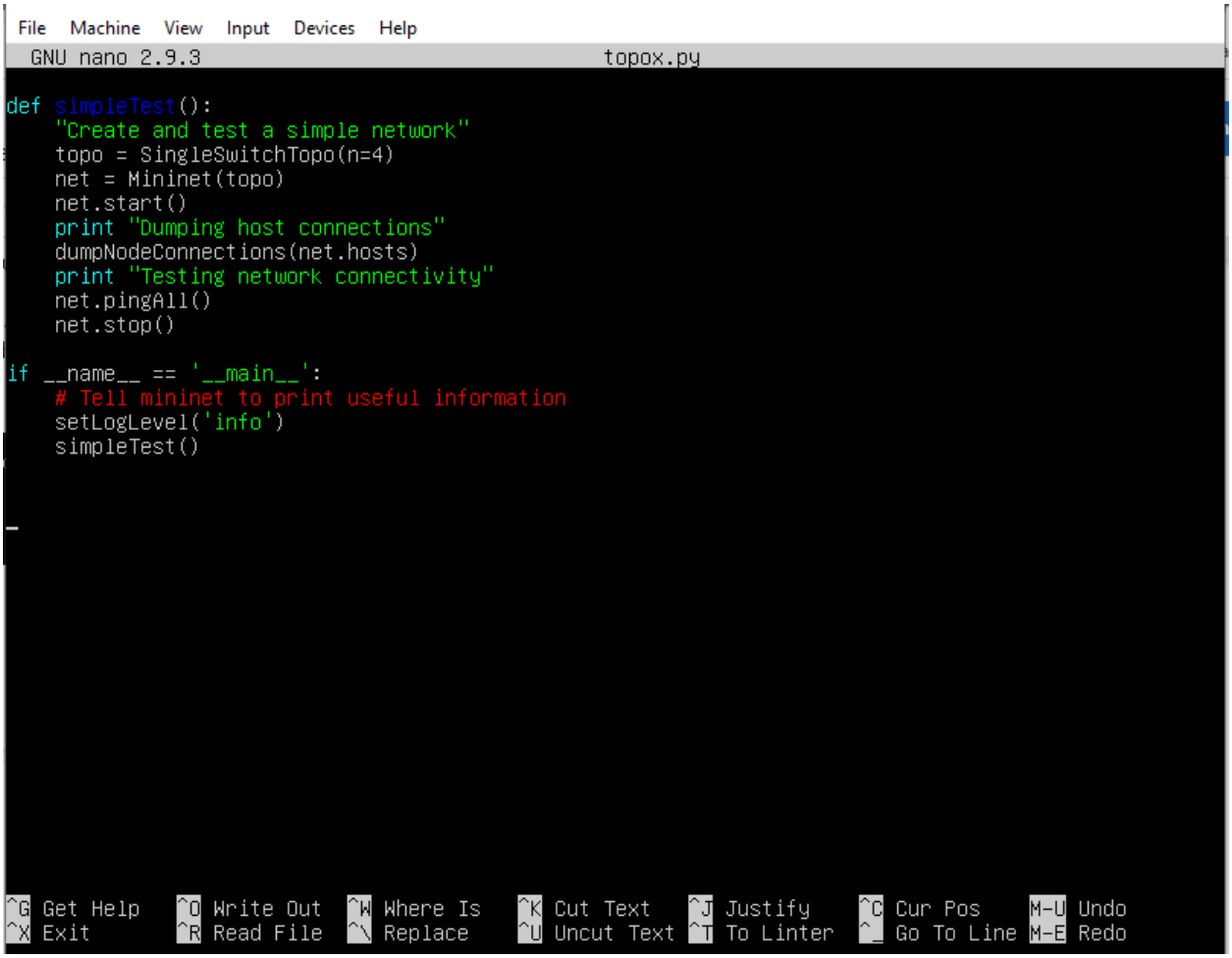
$ sudo fuser -k 6653/tcp

Python Programming for Mininet

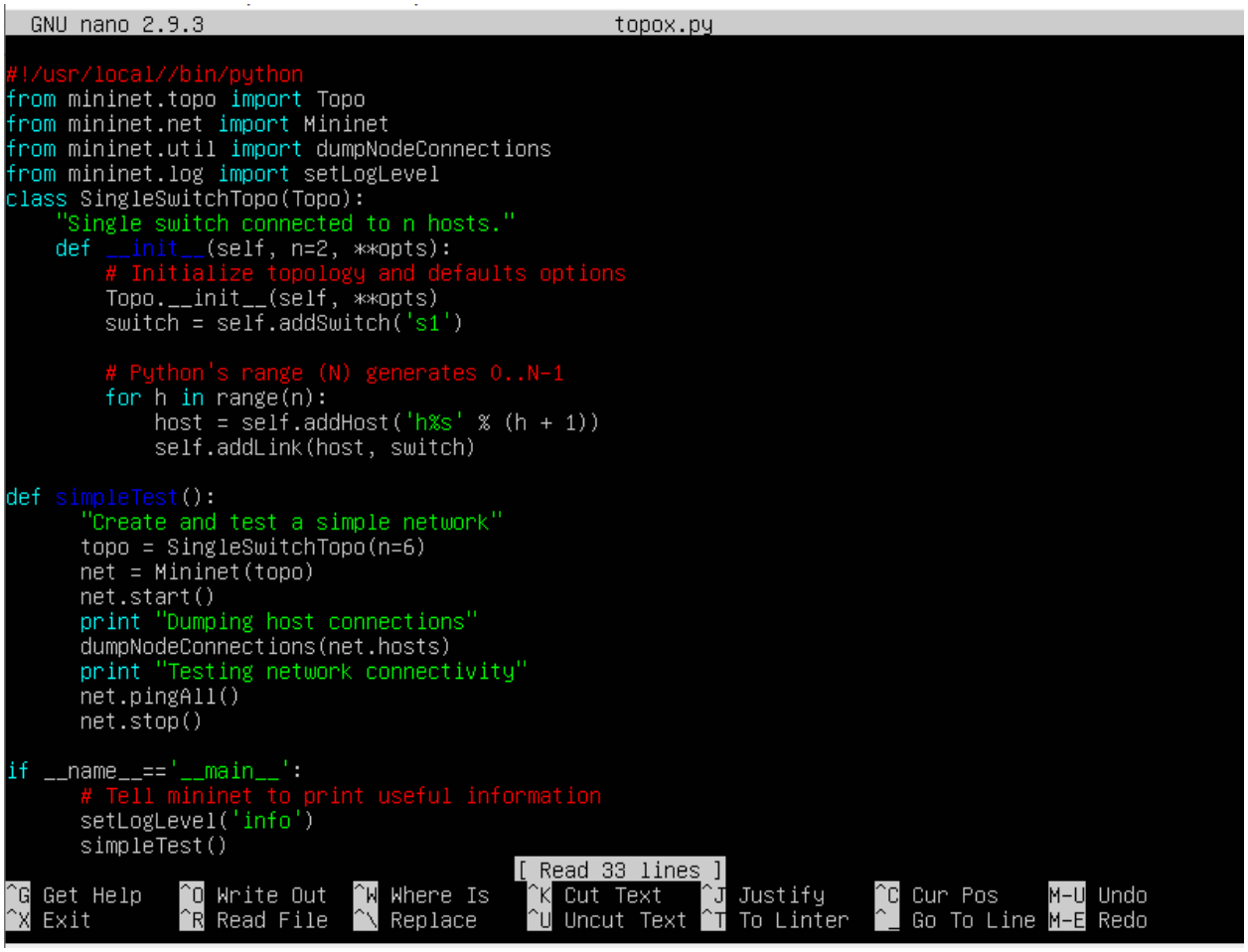
- Using an editor to enter Python statements:
- For the example, nano was used and the file saved as topox.py
- Save in the directory /home/<username>/mininet/custom/
- Run command :
    - **$ sudo python topox.py**
- The image below shows the detail

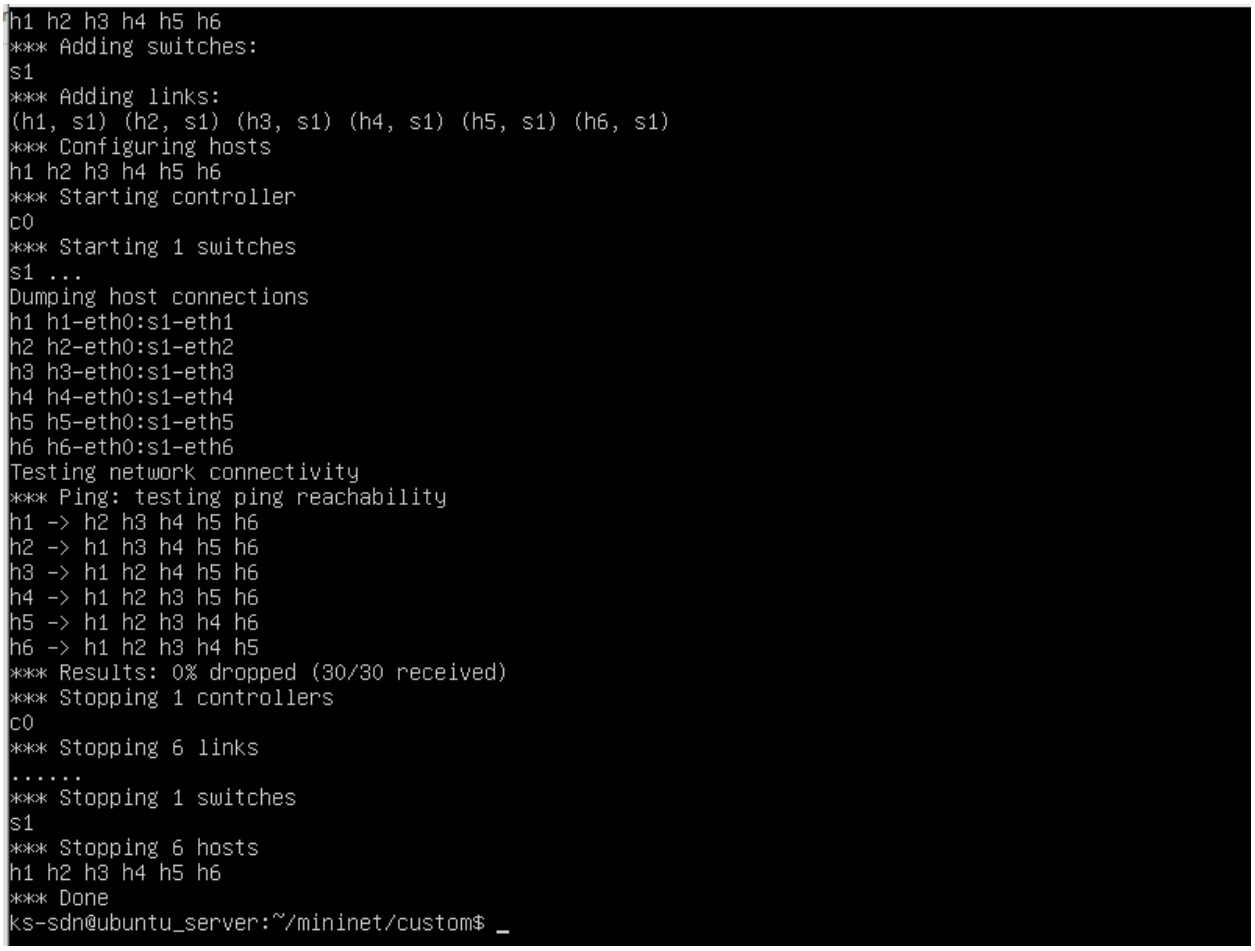



File Machine View Input Devices Help
GNU nano 2.9.3 topox.py
#!/usr/bin/python
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel
class SingleSwitchTopo(Topo):
"Single switch connected to n hosts."
def __init__(self, n=2, **opts):
# Initialize topology and default options
Topo.__init__(self, **opts)
switch = self.addSwitch('s1')
# Python's range (N) generates 0..N-1
for h in range(n):
host = self.addHost('h%s' % (h + 1))
self.addLink(host, switch)
def simpleTest():
"Create and test a simple network"
topo = SingleSwitchTopo(n=4)
net = Mininet(topo)
net.start()
print "Dumping host connections"
dumpNodeConnections(net.hosts)
print "Testing network connectivity"
net.pingAll()
net.stop()
if __name__ == '__main__':
# Tell mininet to print useful information
setLogLevel('info')
simpleTest()
[File 'topox.py' is unwritable]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line M-E Redo


File Machine View Input Devices Help
GNU nano 2.9.3 topox.py
def simpleTest():
"Create and test a simple network"
topo = SingleSwitchTopo(n=4)
net = Mininet(topo)
net.start()
print "Dumping host connections"
dumpNodeConnections(net.hosts)
print "Testing network connectivity"
net.pingAll()
net.stop()
if __name__ == '__main__':
# Tell mininet to print useful information
setLogLevel('info')
simpleTest()
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line M-E Redo

```
ks-sdn@ubuntu_server:~/mininet/custom$ sudo python topox.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
Dumping host connections
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s1-eth3
h4 h4-eth0:s1-eth4
Testing network connectivity
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
*** Stopping 1 controllers
c0
*** Stopping 4 links
....
*** Stopping 1 switches
s1
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
ks-sdn@ubuntu_server:~/mininet/custom$ _
```


GNU nano 2.9.3
topox.py
#!/usr/local//bin/python
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel
class SingleSwitchTopo(Topo):
"Single switch connected to n hosts."
def __init__(self, n=2, **opts):
# Initialize topology and defaults options
Topo.__init__(self, **opts)
switch = self.addSwitch('s1')
# Python's range (N) generates 0..N-1
for h in range(n):
host = self.addHost('h%s' % (h + 1))
self.addLink(host, switch)
def simpleTest():
"Create and test a simple network"
topo = SingleSwitchTopo(n=6)
net = Mininet(topo)
net.start()
print "Dumping host connections"
dumpNodeConnections(net.hosts)
print "Testing network connectivity"
net.pingAll()
net.stop()
if __name__=='__main__':
# Tell mininet to print useful information
setLogLevel('info')
simpleTest()
[Read 33 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line M-E Redo



Simple Python-based controller

- Installed RYU
    - The directory /ryu was removed completed with the linux command
    - **rm -rf ryu # this remove package installed in the previous iteration**
    - Then installed RYU from source code located at Github repo:
    - $ cd ~
    - $ git clone https://github.com/faucetsdn/ryu.git
    - $ cd ryu
    - $ python2 -m virtualenv v-ryu
    - $ source v-ryu/bin/activate
    - (v-ryu)$ pip install ovs==2.6.0 tinyrpc==0.9 paramiko
    - (v-ryu)$ pip install .
- Start RYU with:
    - $PYTHONPATH=. ./bin/ryu-manager –verbose ryu/app/simple_switch.py

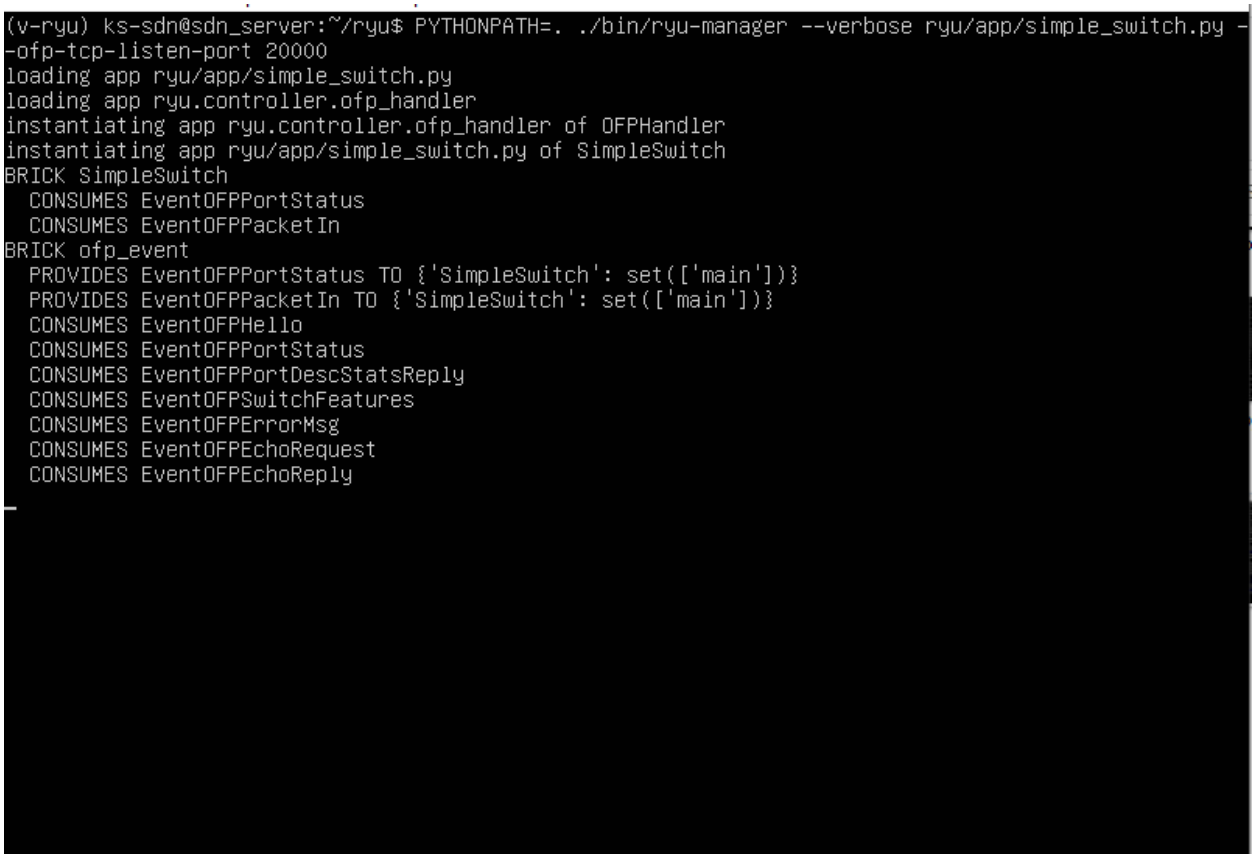
```
  Building wheel for ryu (setup.py) ... done
  Created wheel for ryu: filename=ryu-4.34-py2-none-any.whl size=2203449 sha256=2f7eb7981d77160be5b4
32f816918a35478a294699ac12f06cd2599040bec04b
  Stored in directory: /tmp/pip-ephem-wheel-cache-q0yHAG/wheels/0e/72/70/a20908bfdd7b14e5f93e58acca5
87d31e94b0a4cedbce29c30
Successfully built ryu
Installing collected packages: monotonic, greenlet, dnspython, eventlet, msgpack, contextlib2, typin
g, singledispatch, zipp, scandir, pathlib2, importlib-resources, netaddr, pbr, pytz, Babel, oslo.i18
n, stevedore, wrapt, funcsigs, debtcollector, rfc3986, certifi, urllib3, chardet, idna, requests, Py
YAML, oslo.config, pyparsing, packaging, repoze.lru, routes, tinyrpc, webob, ryu
  Attempting uninstall: tinyrpc
    Found existing installation: tinyrpc 0.9
    Uninstalling tinyrpc-0.9:
      Successfully uninstalled tinyrpc-0.9
Successfully installed Babel-2.9.1 PyYAML-5.4.1 certifi-2021.10.8 chardet-4.0.0 contextlib2-0.6.0.po
st1 debtcollector-1.22.0 dnspython-1.16.0 eventlet-0.31.1 funcsigs-1.0.2 greenlet-1.1.2 idna-2.10 im
portlib-resources-3.3.1 monotonic-1.6 msgpack-1.0.4 netaddr-0.8.0 oslo.config-7.0.0 oslo.i18n-3.25.1
 packaging-20.9 pathlib2-2.3.7.post1 pbr-5.9.0 pyparsing-2.4.7 pytz-2022.1 repoze.lru-0.7 requests-2
.27.1 rfc3986-1.5.0 routes-2.5.1 ryu-4.34 scandir-1.10.0 singledispatch-3.7.0 stevedore-1.32.0 tinyr
pc-1.0.4 typing-3.10.0.0 urllib3-1.26.9 webob-1.8.7 wrapt-1.14.1 zipp-1.2.0
(v-ryu) ks-sdn@sdn_server:~/ryu$ PYTHONPATH=. ./bin/ryu-manager --verbose ryu/app/simple_switch.py
Traceback (most recent call last):
  File "./bin/ryu-manager", line 18, in <module>
    from ryu.cmd.manager import main
  File "/home/ks-sdn/ryu/ryu/cmd/manager.py", line 33, in <module>
    from ryu.app import wsgi
  File "/home/ks-sdn/ryu/ryu/app/wsgi.py", line 23, in <module>
    from tinyrpc.server import RPCServer
  File "/home/ks-sdn/ryu/v-ryu/local/lib/python2.7/site-packages/tinyrpc/__init__.py", line 4, in <m
odule>
    from .protocols import *
  File "/home/ks-sdn/ryu/v-ryu/local/lib/python2.7/site-packages/tinyrpc/protocols/__init__.py", lin
e 15
    def __init__(self) -> None:
                       ^
SyntaxError: invalid syntax
(v-ryu) ks-sdn@sdn_server:~/ryu$ _
```

- **Resolving the syntax error:**
  - $ sudo pip uninstall tinyrpc && sudo pip install tinyrpc==0.9.4
  - Then Start RYU

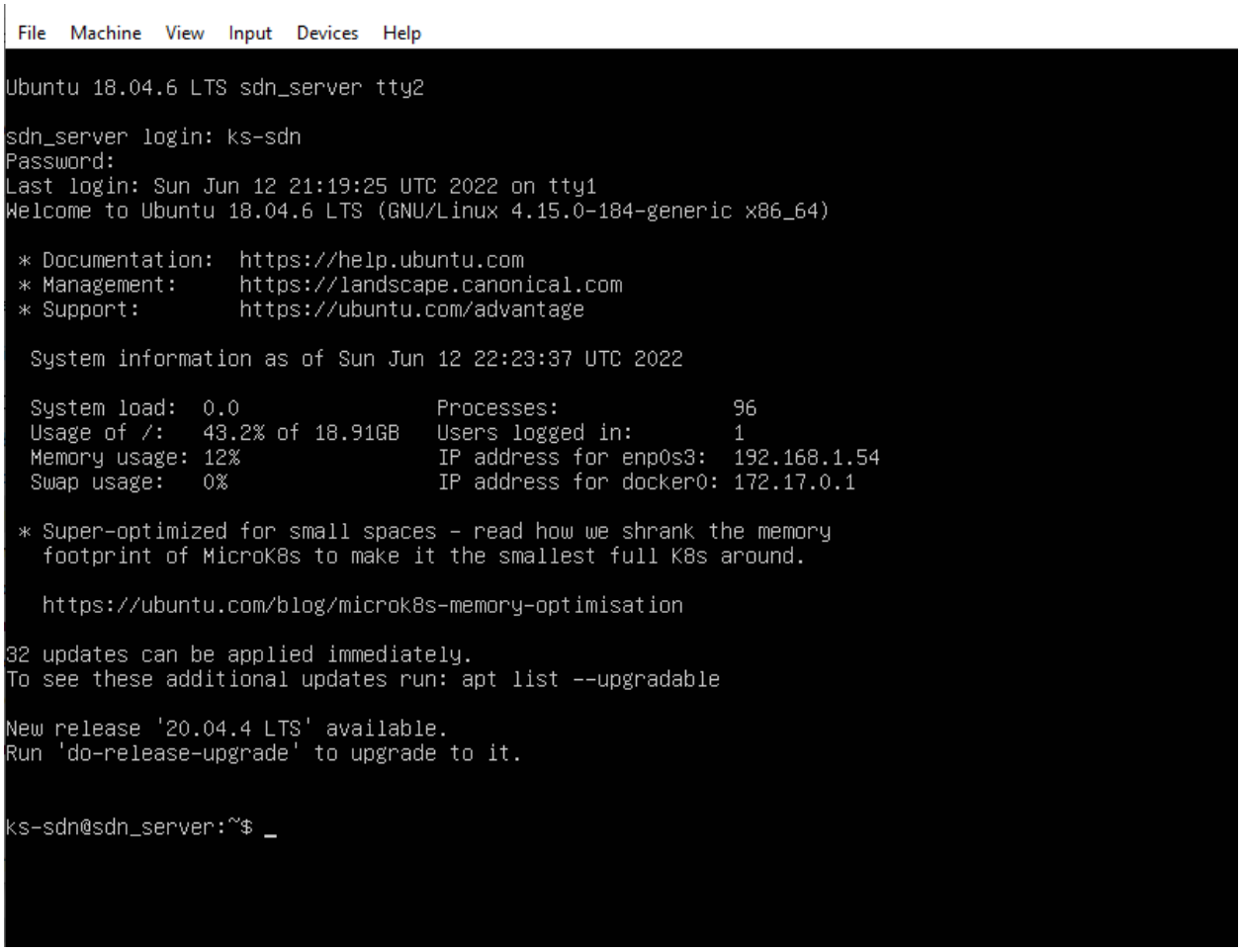
```
Installing setuptools, pkg_resources, pip, wheel...sourcedone.
ks-sdn@sdn_server:~/ryu$ source v-ryu/bin/activate
(v-ryu) ks-sdn@sdn_server:~/ryu$ PYTHONPATH=. ./bin/ryu-manager --verbose ryu/app/simple_switch.py
loading app ryu/app/simple_switch.py
loading app ryu.controller.ofp_handler
instantiating app ryu.controller.ofp_handler of OFPHandler
instantiating app ryu/app/simple_switch.py of SimpleSwitch
BRICK SimpleSwitch
  CONSUMES EventOFPPacketIn
  CONSUMES EventOFPPortStatus
BRICK ofp_event
  PROVIDES EventOFPPacketIn TO {'SimpleSwitch': set(['main'])}
  PROVIDES EventOFPPortStatus TO {'SimpleSwitch': set(['main'])}
  CONSUMES EventOFPPortDescStatsReply
  CONSUMES EventOFPHello
  CONSUMES EventOFPErrorMsg
  CONSUMES EventOFPEchoRequest
  CONSUMES EventOFPPortStatus
  CONSUMES EventOFPEchoReply
  CONSUMES EventOFPSwitchFeatures
hub: uncaught exception: Traceback (most recent call last):
  File "/home/ks-sdn/ryu/ryu/lib/hub.py", line 60, in _launch
    return func(*args, **kwargs)
  File "/home/ks-sdn/ryu/ryu/controller/controller.py", line 154, in __call__
    self.ofp_ssl_listen_port)
  File "/home/ks-sdn/ryu/ryu/controller/controller.py", line 206, in server_loop
    datapath_connection_factory)
  File "/home/ks-sdn/ryu/ryu/lib/hub.py", line 127, in __init__
    self.server = eventlet.listen(listen_info)
  File "/home/ks-sdn/ryu/v-ryu/local/lib/python2.7/site-packages/eventlet/convenience.py", line 78,
in listen
    sock.bind(addr)
  File "/usr/lib/python2.7/socket.py", line 228, in meth
    return getattr(self._sock,name)(*args)
error: [Errno 98] Address already in use

(v-ryu) ks-sdn@sdn_server:~/ryu$ _
```

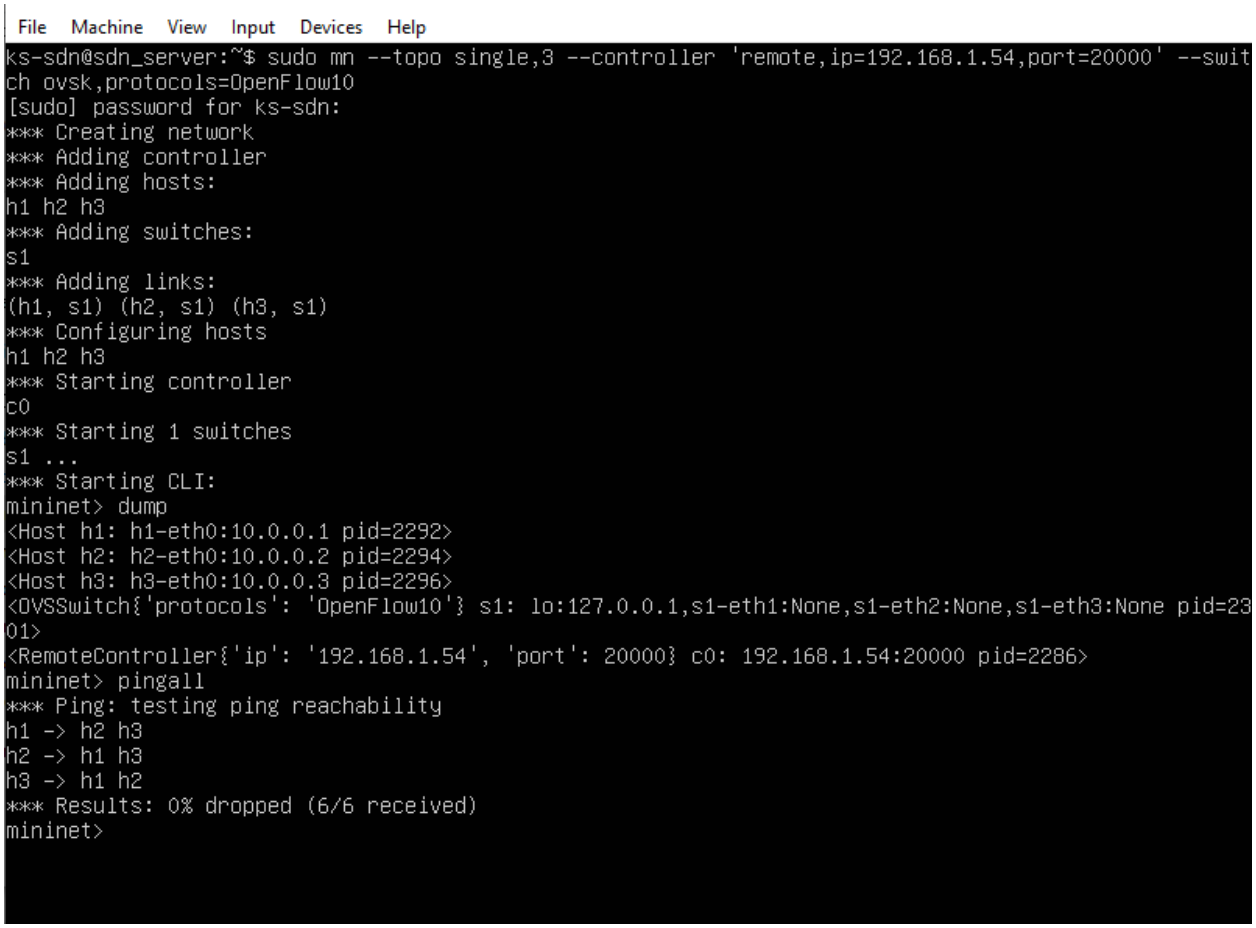
- Resolving the Address already in use error above:
    - Run the modified statement with port 20000

```
File  Machine  View  Input  Devices  Help
(v-ryu) ks-sdn@sdn_server:~/ryu$ PYTHONPATH=. ./bin/ryu-manager --verbose ryu/app/simple_switch.py -
-ofp-tcp-listen-port 20000
```

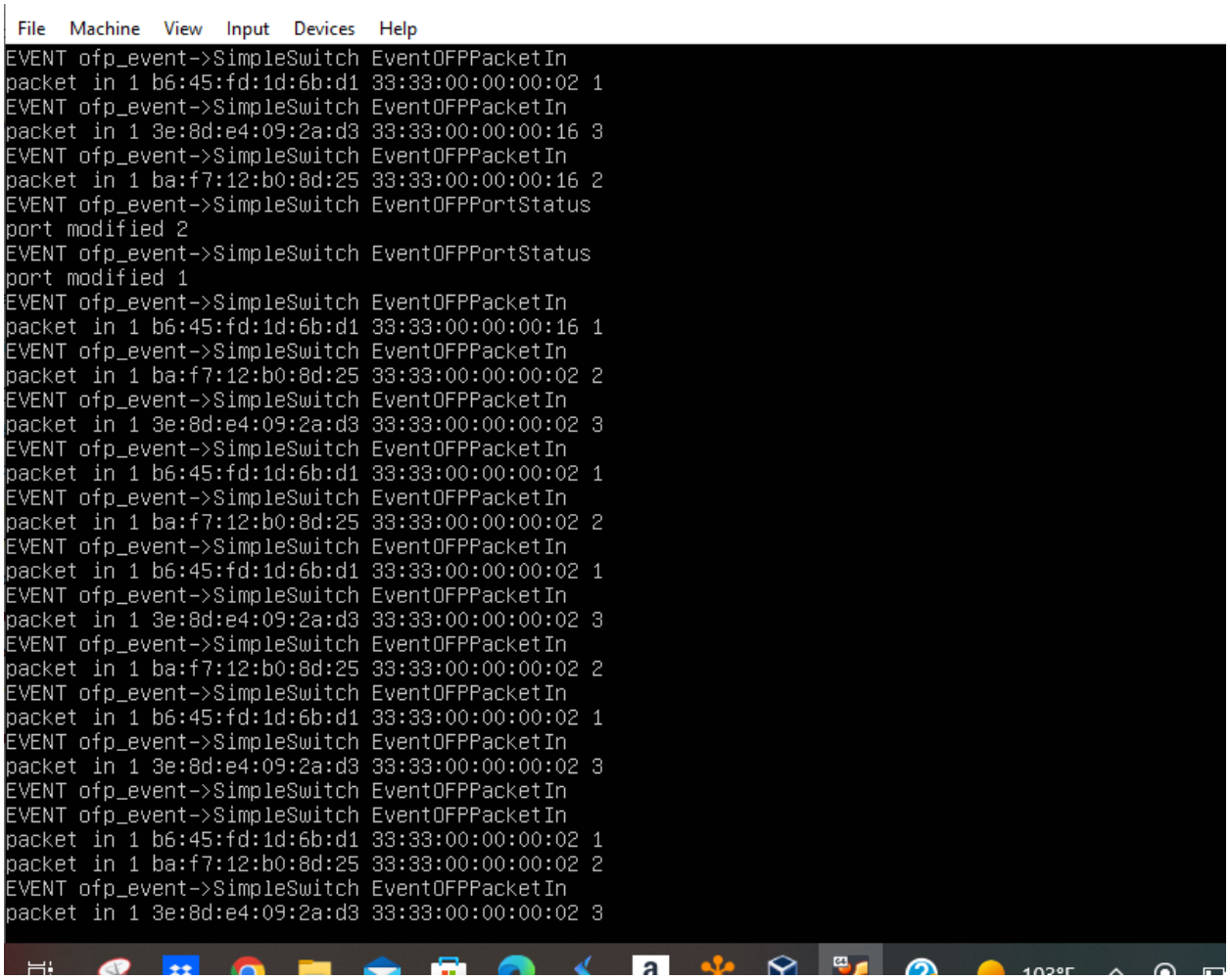


- Open Terminal 2:
    - Start Mininet network and test

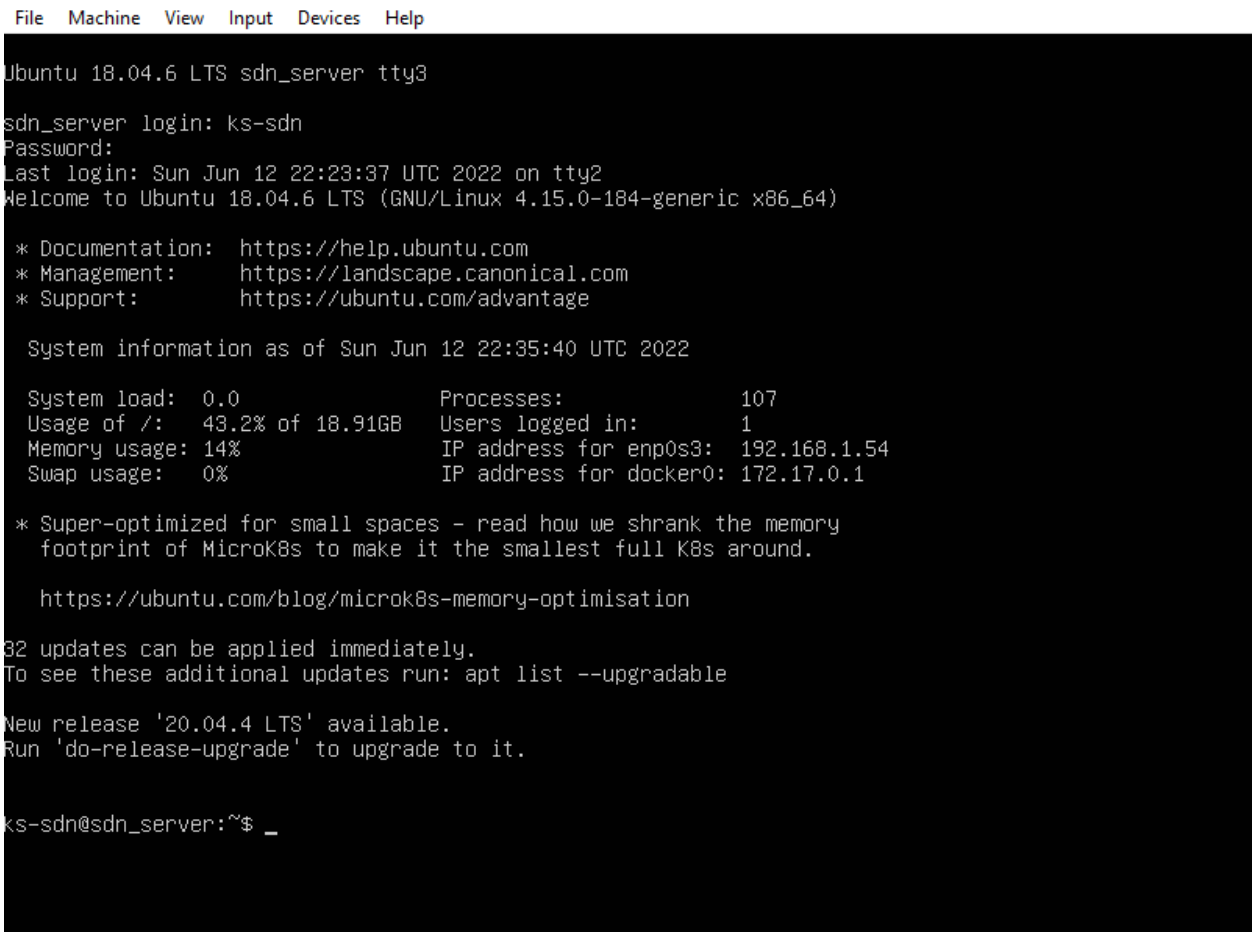

File Machine View Input Devices Help
Ubuntu 18.04.6 LTS sdn_server tty2
sdn_server login: ks-sdn
Password:
Last login: Sun Jun 12 21:19:25 UTC 2022 on tty1
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-184-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information as of Sun Jun 12 22:23:37 UTC 2022
System load: 0.0 Processes: 96
Usage of /: 43.2% of 18.91GB Users logged in: 1
Memory usage: 12% IP address for enp0s3: 192.168.1.54
Swap usage: 0% IP address for docker0: 172.17.0.1
* Super-optimized for small spaces - read how we shrank the memory
footprint of MicroK8s to make it the smallest full K8s around.
https://ubuntu.com/blog/microk8s-memory-optimisation
32 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
ks-sdn@sdn_server:~$

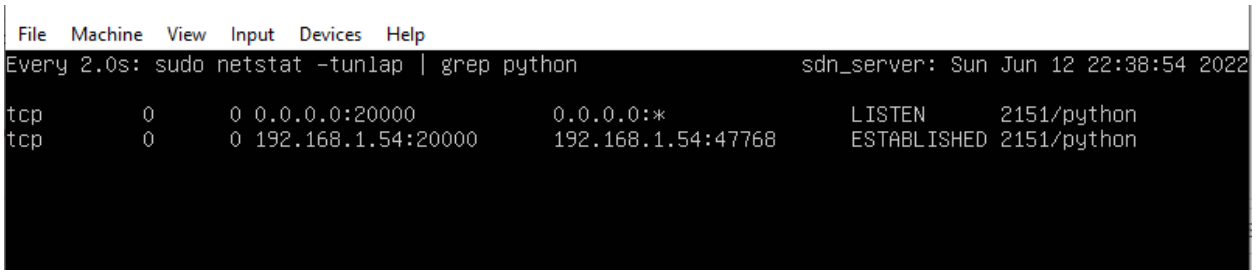


- See the entries information on terminal 1:

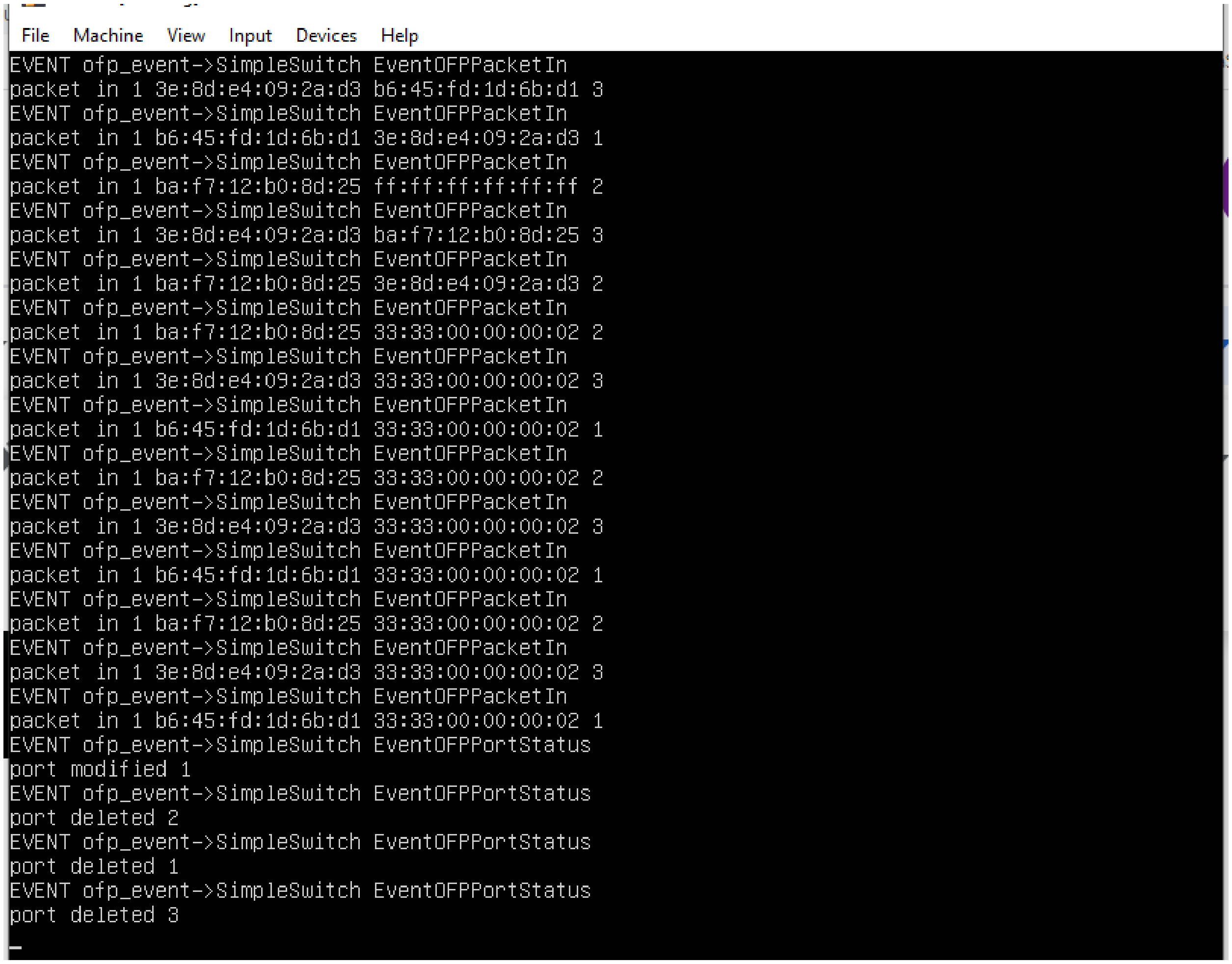


- Open Terminal 3:
    - To monitor RYU SDN controller OpenFlow connections





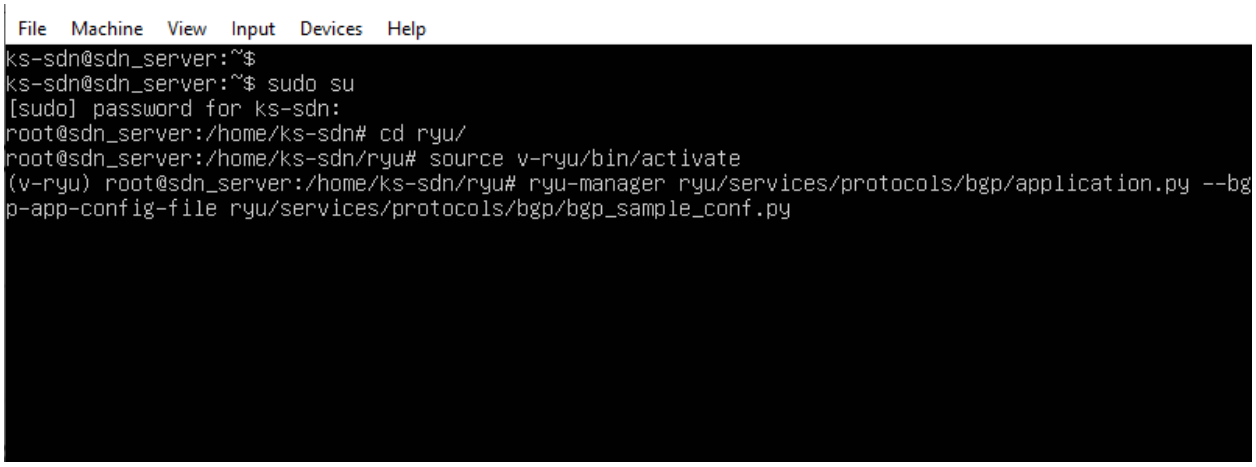
- See the entries information after stopping and cleanup of mininet network:

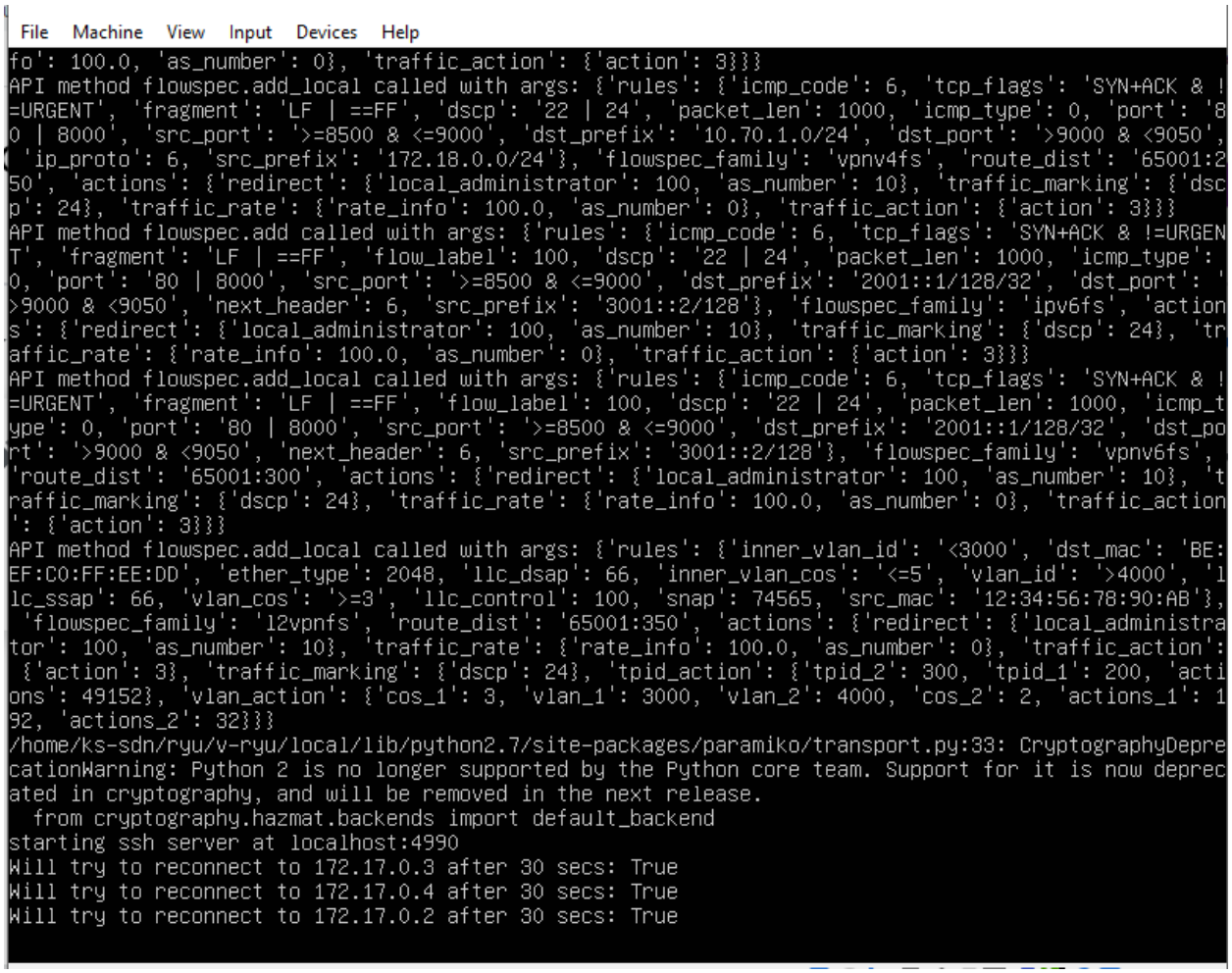



```
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 3e:8d:e4:09:2a:d3 b6:45:fd:1d:6b:d1 3
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 b6:45:fd:1d:6b:d1 3e:8d:e4:09:2a:d3 1
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 ba:f7:12:b0:8d:25 ff:ff:ff:ff:ff:ff 2
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 3e:8d:e4:09:2a:d3 ba:f7:12:b0:8d:25 3
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 ba:f7:12:b0:8d:25 3e:8d:e4:09:2a:d3 2
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 ba:f7:12:b0:8d:25 33:33:00:00:00:02 2
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 3e:8d:e4:09:2a:d3 33:33:00:00:00:02 3
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 b6:45:fd:1d:6b:d1 33:33:00:00:00:02 1
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 ba:f7:12:b0:8d:25 33:33:00:00:00:02 2
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 3e:8d:e4:09:2a:d3 33:33:00:00:00:02 3
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 b6:45:fd:1d:6b:d1 33:33:00:00:00:02 1
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 ba:f7:12:b0:8d:25 33:33:00:00:00:02 2
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 3e:8d:e4:09:2a:d3 33:33:00:00:00:02 3
EVENT ofp_event->SimpleSwitch EventOFPPacketIn
packet in 1 b6:45:fd:1d:6b:d1 33:33:00:00:00:02 1
EVENT ofp_event->SimpleSwitch EventOFPPortStatus
port modified 1
EVENT ofp_event->SimpleSwitch EventOFPPortStatus
port deleted 2
EVENT ofp_event->SimpleSwitch EventOFPPortStatus
port deleted 1
EVENT ofp_event->SimpleSwitch EventOFPPortStatus
port deleted 3
_
```

Simple RYU SDN BGP Topology

- This app is ran as root user
- $ sudo su -
- # cd ryu
- # source v-ryu/bin/activate
- (v-ryu)# ryu-manager ryu/services/protocols/bgp/application.py –bgp-app-config-file ryu/services/protocols/bgp/bgp_sample_conf.py
- The image shown below:

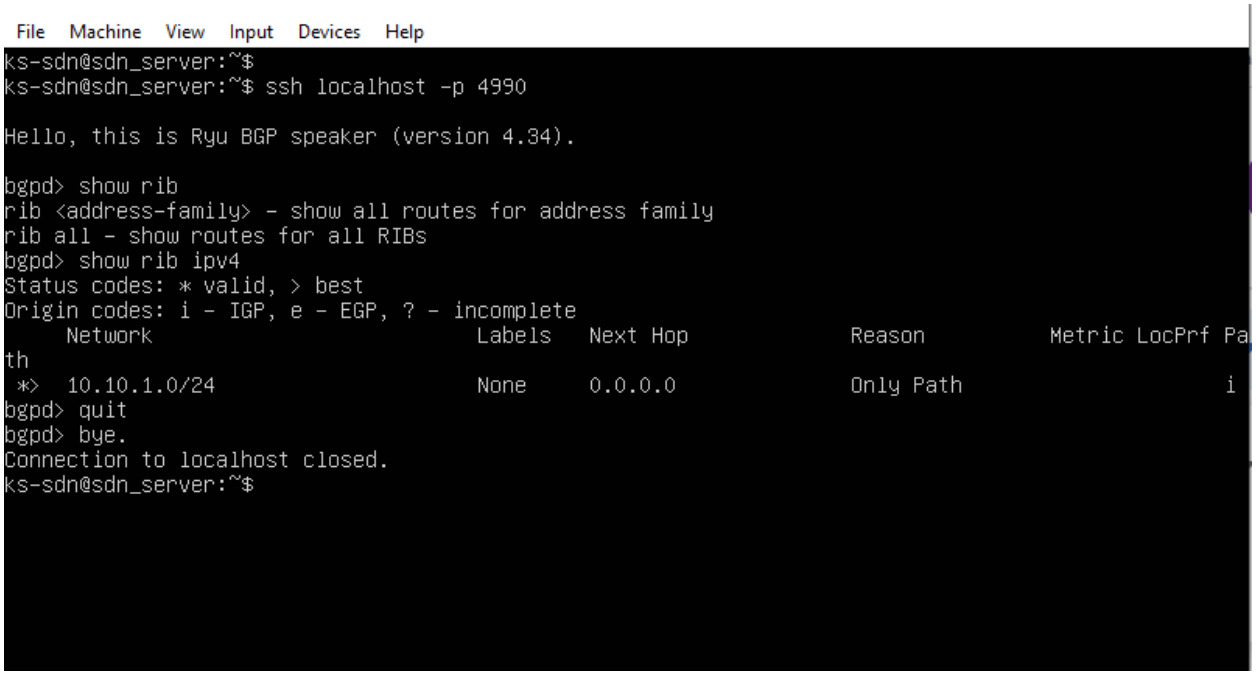




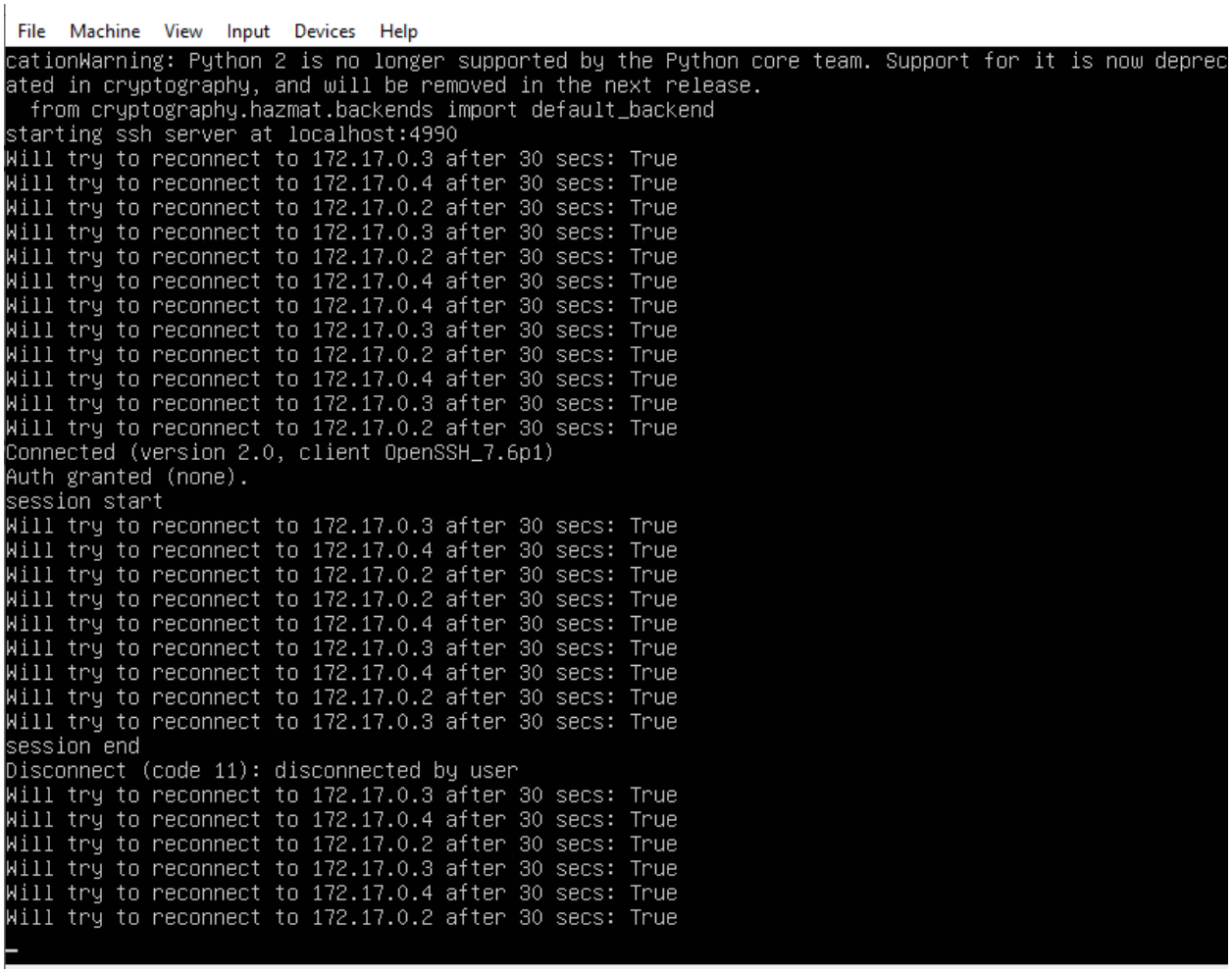
- Open Terminal 2:
- Log (ssh) in at the RYU BGP instance:
    - $ ssh localhost -p 4990
        - bgpd>show rib ?

- bgpd>show rib ipv4
- bgpd>quit

The image shows detail below:



Finally, the image below shows the entries information of terminal 2 in terminal 1:


File Machine View Input Devices Help
cationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprec
ated in cryptography, and will be removed in the next release.
from cryptography.hazmat.backends import default_backend
starting ssh server at localhost:4990
Will try to reconnect to 172.17.0.3 after 30 secs: True
Will try to reconnect to 172.17.0.4 after 30 secs: True
Will try to reconnect to 172.17.0.2 after 30 secs: True
Will try to reconnect to 172.17.0.3 after 30 secs: True
Will try to reconnect to 172.17.0.2 after 30 secs: True
Will try to reconnect to 172.17.0.4 after 30 secs: True
Will try to reconnect to 172.17.0.4 after 30 secs: True
Will try to reconnect to 172.17.0.3 after 30 secs: True
Will try to reconnect to 172.17.0.2 after 30 secs: True
Will try to reconnect to 172.17.0.4 after 30 secs: True
Will try to reconnect to 172.17.0.3 after 30 secs: True
Will try to reconnect to 172.17.0.2 after 30 secs: True
Connected (version 2.0, client OpenSSH_7.6p1)
Auth granted (none).
session start
Will try to reconnect to 172.17.0.3 after 30 secs: True
Will try to reconnect to 172.17.0.4 after 30 secs: True
Will try to reconnect to 172.17.0.2 after 30 secs: True
Will try to reconnect to 172.17.0.2 after 30 secs: True
Will try to reconnect to 172.17.0.4 after 30 secs: True
Will try to reconnect to 172.17.0.3 after 30 secs: True
Will try to reconnect to 172.17.0.4 after 30 secs: True
Will try to reconnect to 172.17.0.2 after 30 secs: True
Will try to reconnect to 172.17.0.3 after 30 secs: True
session end
Disconnect (code 11): disconnected by user
Will try to reconnect to 172.17.0.3 after 30 secs: True
Will try to reconnect to 172.17.0.4 after 30 secs: True
Will try to reconnect to 172.17.0.2 after 30 secs: True
Will try to reconnect to 172.17.0.3 after 30 secs: True
Will try to reconnect to 172.17.0.4 after 30 secs: True
Will try to reconnect to 172.17.0.2 after 30 secs: True