

Florida International University
School of Computing and Information Sciences

Software Engineering Focus

Installation Guide

AmLight - Learning OpenFlow/SDN Network Research and
Experimentation

Team Members: Cindy Perez, Jose Mercardo, Kenia Chang He, and Yuyang Leng

Product Owner: Vasilka Chergarova

Mentor: Jeronimo Bezerra

Instructor: Masoud Sadjadi

This section contains all the necessary steps in order to successfully run Kytos for the first time on any clean Ubuntu virtual machine.

Step 1: Enter the command in the terminal to install the latest version of Python which is python3.

```
ubuntu@ip-172-31-42-4:~$ sudo apt-get install python3
```

Step 2: Enter the command in the terminal to download the package lists.

```
ubuntu@ip-172-31-42-4:~$ sudo apt-get update
```

Step 3: Enter the command in the terminal to upgrade the packages from step 2.

```
ubuntu@ip-172-31-42-4:~$ sudo apt-get upgrade
```

Step 4: Enter the command in the terminal to install the dependency package.

```
ubuntu@ip-172-31-42-4:~$ sudo apt-get install libpython3-dev
```

Step 5: Enter the command in the terminal to install the python virtual environment package.

```
ubuntu@ip-172-31-42-4:~$ sudo apt-get install python3-venv
```

Step 6: Enter the command in the terminal to install the curl package.

```
ubuntu@ip-172-31-42-4:~$ sudo apt-get install curl
```

Step 7: Enter the command in the terminal to install the python3 pip tool.

```
ubuntu@ip-172-31-42-4:~$ sudo apt-get install python3-pip
```

Step 8: Enter the command in the terminal to make a directory called kytostest.

```
ubuntu@ip-172-31-42-4:~$ mkdir kytostest
```

Step 9: Enter the command in the terminal to upgrade the necessary python packages.

```
ubuntu@ip-172-31-42-4:~$ pip3 install --upgrade pip setuptools wheel
```

Step 10: Enter the command in the terminal to go to the kytostest directory.

```
ubuntu@ip-172-31-42-4:~$ cd kytostest
```

Step 11: Enter the command in the terminal to download Kytos onto the machine.

```
ubuntu@ip-172-31-42-4:~/kytostest$ for repo in python-openflow kytos-utils kytos; do  
> sudo git clone https://github.com/kytos/${repo}  
> cd ${repo}  
> sudo python3 setup.py develop  
> cd ..  
> done
```

Step 12: Enter the command in the terminal to start running Kytos in background mode.

```
ubuntu@ip-172-31-42-4:~/kytostest$ sudo kytosd
```

Step 13: Enter the command in the terminal to install the necessary Kytos NApps (you may need to run the command again if the installation fails).

```
ubuntu@ip-172-31-42-4:~/kytostest$ sudo kytos napps install kytos/of_core \  
> kytos/flow_manager \  
> kytos/of_l2ls \  
> kytos/of_lldp \  
> kytos/topology
```

Step 14: Enter the command in the terminal to check the status of the Kytos NApps installed.

```
ubuntu@ip-172-31-42-4:~/kytostest$ kytos napps list
```

Here are the results that should be given after running the command above.

```
ubuntu@ip-172-31-42-4:~/kytostest$ kytos napps list  


| Status | NApp ID                | Description                                                                  |
|--------|------------------------|------------------------------------------------------------------------------|
| [ie]   | kytos/flow_manager:2.3 | Manage switches' flows through a REST API.                                   |
| [ie]   | kytos/of_core:1.4.1    | OpenFlow Core of Kytos Controller, responsible for main OpenFlow operations. |
| [ie]   | kytos/of_l2ls:1.1.1    | A L2 learning switch application for OpenFlow switches.                      |
| [ie]   | kytos/of_lldp:0.1.4    | Discover network-to-network interfaces (NNIs) using the LLDP protocol.       |
| [ie]   | kytos/storehouse:1.3.0 | Persistence NApp with support for multiple backends                          |
| [ie]   | kytos/topology:3.6.1   | Manage the network topology.                                                 |

  
Status: (i)nstalled, (e)nabled
```

Step 15: Enter the command in the terminal to install Mininet.

```
ubuntu@ip-172-31-42-4:~/kytostest$ sudo apt-get install mininet
```

Step 16: Enter the command in the terminal to test Mininet.

```
ubuntu@ip-172-31-42-4:~/kytostest$ sudo mn --test pingall
```

Here is the result of testing Mininet.

```
ubuntu@ip-172-31-42-4:~/kystotest$ sudo mn --test pingall
*** 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 ...
*** 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 0.236 seconds
```

Step 17: Enter the command in the terminal to check the process running in the machine.

```
ubuntu@ip-172-31-42-4:~/kystotest$ sudo ss -tunlp
```

Here are the results of the currently running processes command. Take note of the pid of “kystod” as this can change from system to system.

```
ubuntu@ip-172-31-42-4:~/kytostest$ sudo ss -tunlp
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port
udp UNCONN 0 0 127.0.0.53%lo:53 0.0.0.0:* users:(("systemd-resolve",pid=716,fd=12))
udp UNCONN 0 0 172.31.42.4%eth0:68 0.0.0.0:* users:(("systemd-network",pid=695,fd=15))
tcp LISTEN 0 128 0.0.0.0:8181 0.0.0.0:* users:(("kytosd",pid=10158,fd=11))
tcp LISTEN 0 128 127.0.0.53%lo:53 0.0.0.0:* users:(("systemd-resolve",pid=716,fd=13))
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:* users:(("sshd",pid=1008,fd=3))
tcp LISTEN 0 100 0.0.0.0:6653 0.0.0.0:* users:(("kytosd",pid=10158,fd=10))
tcp LISTEN 0 128 [::]:22 [::]:* users:(("sshd",pid=1008,fd=4))
```

Step 18: Enter the command in the terminal to stop the Kytos process running in the background, replace the number “10158” with the pid of Kytos process observed from step 17.

```
ubuntu@ip-172-31-42-4:~/kytostest$ sudo kill -9 10158
```

Step 19: Enter the command in the terminal to check the currently running processes, and the process “kytosd” should be gone and not show up in the currently running processes list.

```
ubuntu@ip-172-31-42-4:~/kytostest$ sudo ss -tunlp
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port
udp UNCONN 0 0 127.0.0.53%lo:53 0.0.0.0:* users:(("systemd-resolve",pid=716,fd=12))
udp UNCONN 0 0 172.31.42.4%eth0:68 0.0.0.0:* users:(("systemd-network",pid=695,fd=15))
tcp LISTEN 0 128 127.0.0.53%lo:53 0.0.0.0:* users:(("systemd-resolve",pid=716,fd=13))
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:* users:(("sshd",pid=1008,fd=3))
tcp LISTEN 0 128 [::]:22 [::]:* users:(("sshd",pid=1008,fd=4))
```