A Guide to Launching an SDN-Cockpit Application

This guide will assume the installation and setup of the environment as depicted in 1_SDNCockpitInstallManual.pdf was followed.

Step 1: Open Git Bash and change the directory as depicted in figure 1.

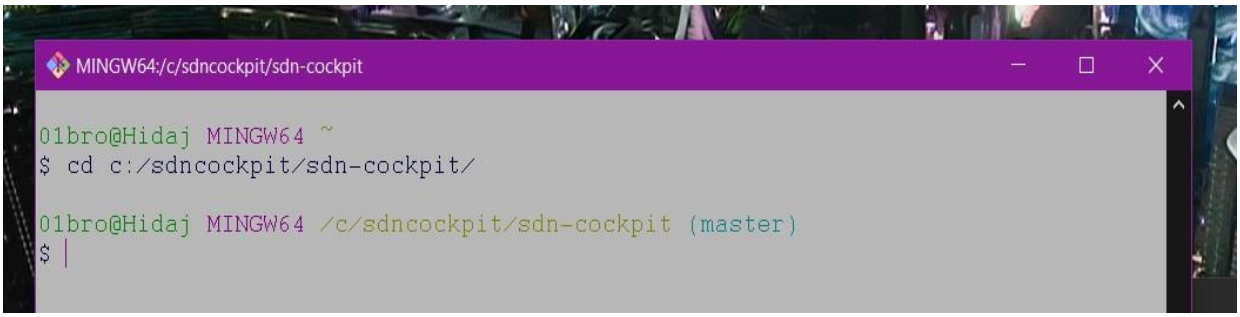

Figure 1. Depicts git bash terminal with the directory change instigated.

Step 2: Utilise the command vagrant up to instantiate a virtual machine (vm).

Figure 2. Depicts the vagrant up command about to be run.

Step 3: When vagrant up has finished instantiating the virtual environment use the vagrant ssh command to connect to the vm.

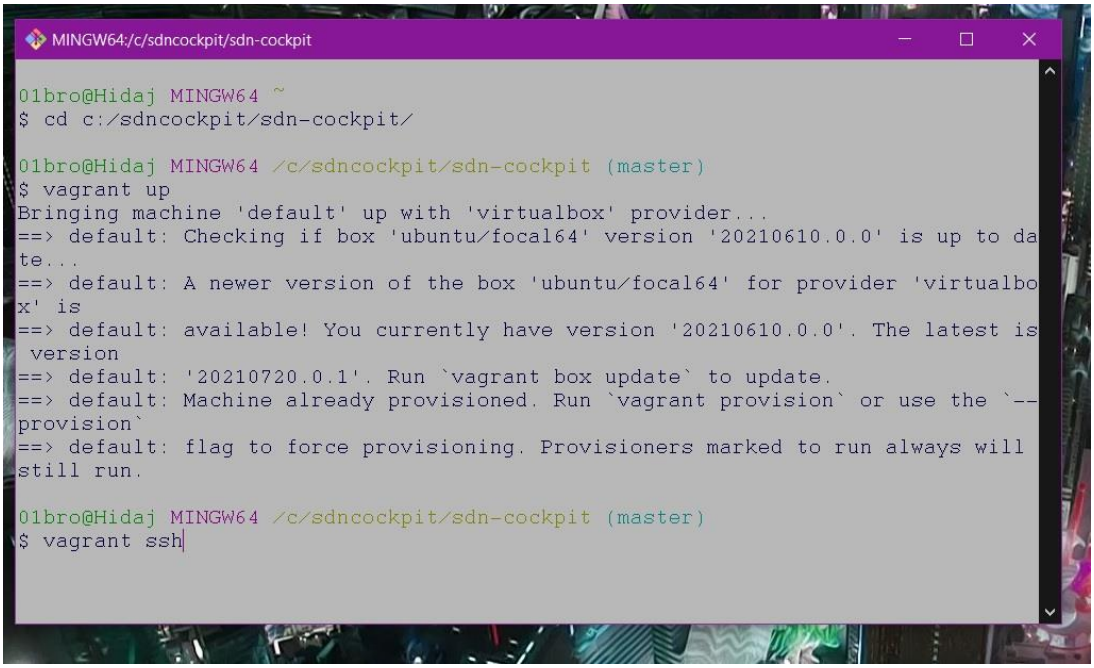

Figure 3. Depicts completed vm instantiation and vagrant ssh command execution.

Step 4: Once a connection to the vm has been made use the command bash run.sh to launch the SDN-Cockpit environment.

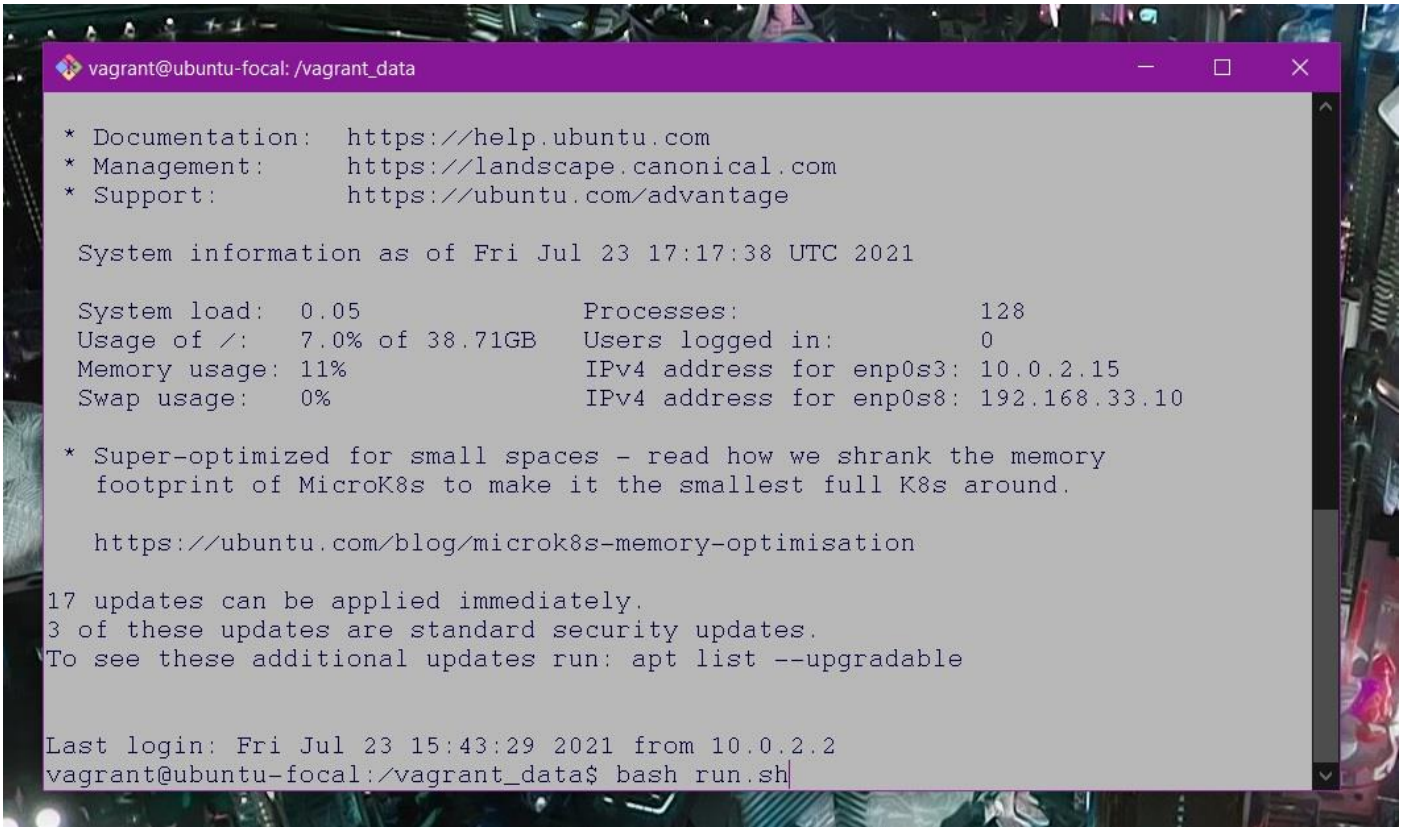

Figure 4. Depicts the completion of the vagrant ssh command and the command to launch SDN-Cockpit.

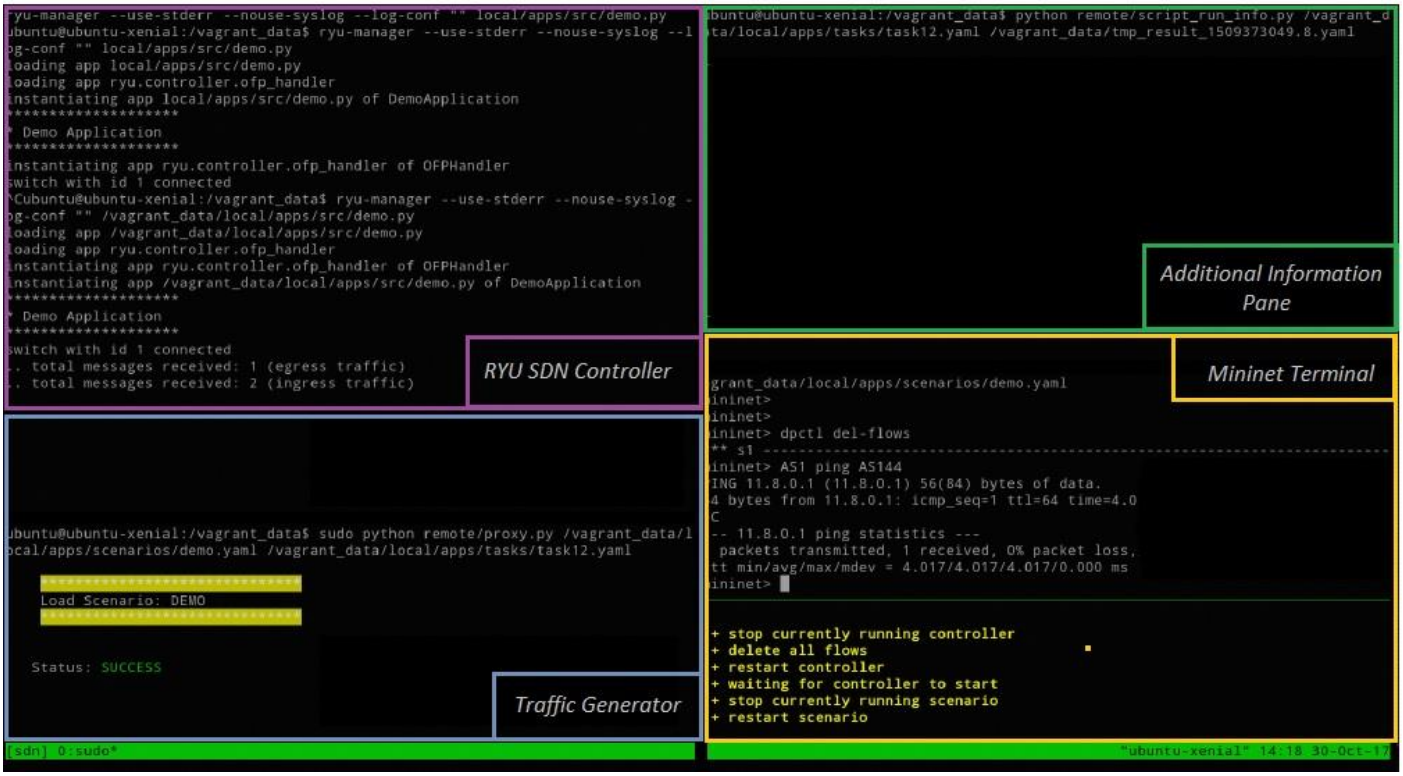

Figure 4. Depicts the SDN-Cockpit environment and its constituent components have been highlighted.
By default SDN Cockpit loads a demo configuration.

Step 5: Replace the Demo.py application in the RYU SDN controller pane with unconfigured.py to view the unmodified application, my solution is named configured.py.

```
vagrant@ubuntu-focal: /vagrant_data
loading app local/apps/src/configured.py
loading app ryu.controller.ofp_handler
instantiating app local/apps/src/configured.py of SecureGateway
instantiating app ryu.controller.ofp_handler of OFPHandler
switch with id 1 connected


/// Switch connected. ID: 1
/// [Switch 1]: PACKET-IN (#1) on port: 3
^Cvagrant@ubuntu-focal:/vagrant_data$ python3 -m py_compile local/apps/src/conf
igured.py && ryu-manager --use-stderr --nouse-syslog --log-conf "" local/apps/s
rc/configured.py
loading app local/apps/src/configured.py
loading app ryu.controller.ofp_handler
instantiating app local/apps/src/configured.py of SecureGateway
instantiating app ryu.controller.ofp_handler of OFPHandler
switch with id 1 connected


/// Switch connected. ID: 1
/// [Switch 1]: PACKET-IN (#1) on port: 4
/// [Switch 1]: PACKET-IN (#2) on port: 3
/// [Switch 1]: PACKET-IN (#3) on port: 1
/// [Switch 1]: PACKET-IN (#4) on port: 2
-----------------------------------------------------------------------------
vagrant@ubuntu-focal:/vagrant_data$ sudo python3 remote/proxy.py /vagrant_data/
local/apps/scenarios/demo.yaml

    ******************** ERROR ********************
    Scenario file not found: /vagrant_data/local/apps/scenarios/demo.yaml
    Make sure to specify an existing scenario!
    ***********************************************
^CTraceback (most recent call last):
  File "remote/proxy.py", line 786, in <module>
    time.sleep(1)
KeyboardInterrupt

vagrant@ubuntu-focal:/vagrant_data$ sudo python3 remote/proxy.py /vagrant_data/
local/apps/scenarios/firewall.yaml
```

```
Task: Secure Medical IOT Gateway Topology:
Scenario: firewall
Description:                                  +----+
Implement the following rules:                | L1 |
1. Patient Network (L1) may                   +----+
exchange packets with Doctor                    |
Network (H1) 2. Patient                         |
Records (R1) may exchange                       |
packets with Doctor Network      +----+     +----+     +----+
(H1) 3. Noone from these three   | R1 | --- | S1 | --- | H1 |
networks (L1,R1,H1) may send     +----+     +----+     +----+
packets to Public Network                       |
Access (M1). 4. Patient                         |
Network (L1) and Patient                        |
Records (R1) may not exchange                 +----+
packets.                                      | M1 |
Result: SUCCESS                               +----+

-----------------------------------------------------------------------------

    ******************** ERROR ********************
    Scenario file /vagrant_data/local/apps/scenarios/demo.yaml invalid (could
not be parsed): [Errno 2] No such file or directory: '/vagrant_data/local/apps
/scenarios/demo.yaml'
    ***********************************************
vagrant@ubuntu-focal:/vagrant_data$ sudo python3 remote/script_run_mininet.py
/vagrant_data/local/apps/scenarios/firewall.yaml
Unable to contact the remote controller at 127.0.0.1:6633
mininet> exit
vagrant@ubuntu-focal:/vagrant_data$ sudo python3 remote/script_run_mininet.py
/vagrant_data/local/apps/scenarios/firewall.yaml
Unable to contact the remote controller at 127.0.0.1:6633
mininet>
-----------------------------------------------------------------------------
vagrant@ubuntu-focal:/vagrant_data$ python3 remote/script_run_watch.py
```

Figure 5. Depicts the configured firewall application being loaded into the controller.

Step 6: Load the firewall.yaml file from the tasks folder in the additional information pane by replacing the None with /vagrant_data/local/apps/tasks/firewall.yaml.

```
vagrant@ubuntu-focal: /vagrant_data
  File "/usr/lib/python3.8/py_compile.py", line 215, in <module>
    sys.exit(main())
  File "/usr/lib/python3.8/py_compile.py", line 207, in main
    compile(filename, doraise=True)
  File "/usr/lib/python3.8/py_compile.py", line 142, in compile
    source_bytes = loader.get_data(file)
  File "<frozen importlib._bootstrap_external>", line 1037, in get_data
FileNotFoundError: [Errno 2] No such file or directory: 'local/apps/src/demo.py
'
vagrant@ubuntu-focal:/vagrant_data$ ^C
vagrant@ubuntu-focal:/vagrant_data$ python3 -m py_compile local/apps/src/config
ured.py && ryu-manager --use-stderr --nouse-syslog --log-conf "" local/apps/src
/configured.py
loading app local/apps/src/configured.py
loading app ryu.controller.ofp_handler
instantiating app local/apps/src/configured.py of SecureGateway
instantiating app ryu.controller.ofp_handler of OFPHandler
switch with id 1 connected


/// Switch connected. ID: 1
/// [Switch 1]: PACKET-IN (#1) on port: 3
^Cvagrant@ubuntu-focal:/vagrant_data$ python3 -m py_compile local/apps/src/conf
igured.py && ryu-manager --use-stderr --nouse-syslog --log-conf "" local/apps/s
rc/configured.py
-----------------------------------------------------------------------------
vagrant@ubuntu-focal:/vagrant_data$ sudo python3 remote/proxy.py /vagrant_data/
local/apps/scenarios/demo.yaml

    ******************** ERROR ********************
    Scenario file not found: /vagrant_data/local/apps/scenarios/demo.yaml
    Make sure to specify an existing scenario!
    ***********************************************
^CTraceback (most recent call last):
  File "remote/proxy.py", line 786, in <module>
    time.sleep(1)
KeyboardInterrupt

vagrant@ubuntu-focal:/vagrant_data$ sudo python3 remote/proxy.py /vagrant_data/
local/apps/scenarios/firewall.yaml
```

```
vagrant@ubuntu-focal:/vagrant_data$ python3 remote/script_run_info.py BootDumm
y

    ******************************
    There is currently no task selected
    ******************************

    Select an application (or create a new one) and add the
    [#tm task=taskname] makro to start task. Please contact your
    supervisor if there are any problems!
^CTraceback (most recent call last):
  File "remote/script_run_info.py", line 54, in <module>
    time.sleep(10) # don't return
KeyboardInterrupt

vagrant@ubuntu-focal:/vagrant_data$ python3 remote/script_run_info.py /vagrant
_data/local/apps/tasks/firewall.yaml /vagrant_data/tmp_result_1627058395.80421
04.yaml
-----------------------------------------------------------------------------
vagrant@ubuntu-focal:/vagrant_data$ sudo python3 remote/script_run_mininet.py
/vagrant_data/local/apps/scenarios/demo.yaml

    ******************** ERROR ********************
    Scenario file /vagrant_data/local/apps/scenarios/demo.yaml invalid (could
not be parsed): [Errno 2] No such file or directory: '/vagrant_data/local/apps
/scenarios/demo.yaml'
    ***********************************************
vagrant@ubuntu-focal:/vagrant_data$ sudo python3 remote/script_run_mininet.py
/vagrant_data/local/apps/scenarios/firewall.yaml
Unable to contact the remote controller at 127.0.0.1:6633
mininet> exit
vagrant@ubuntu-focal:/vagrant_data$ sudo python3 remote/script_run_mininet.py
/vagrant_data/local/apps/scenarios/firewall.yaml
-----------------------------------------------------------------------------
vagrant@ubuntu-focal:/vagrant_data$ python3 remote/script_run_watch.py
```

Figure 6. The firewall task file being loaded into the additional information pane.

Hint: Sometimes the vagrant_data/tmp_result_1627058395.80421.yaml does not sync correctly. If so change the temporary file in the pane to point to the temporary result file on your system.

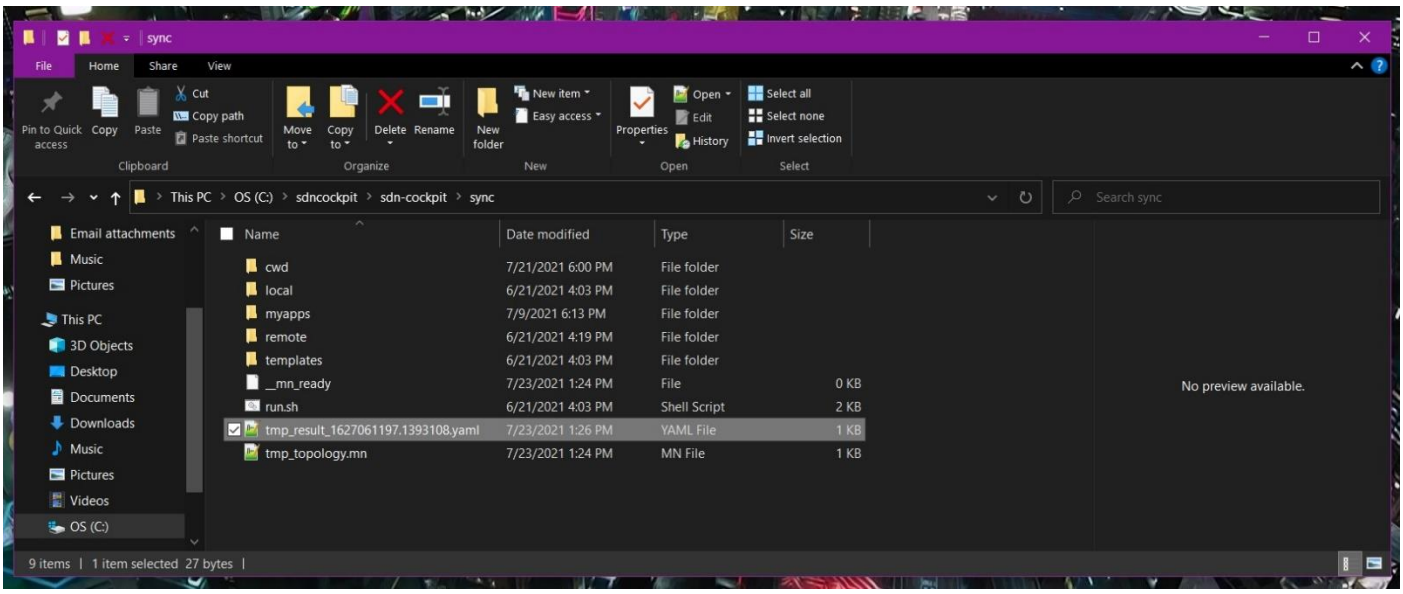

Figure 7. Shows the location of the temporary results file, if there is an incompatibility change the location in the additional information pane.

Step 7: Enter the command exit in the mininet pane then replace the demo.yaml file with firewall.yaml from the scenario folder to configure the network topology depicted in the additional information pane.

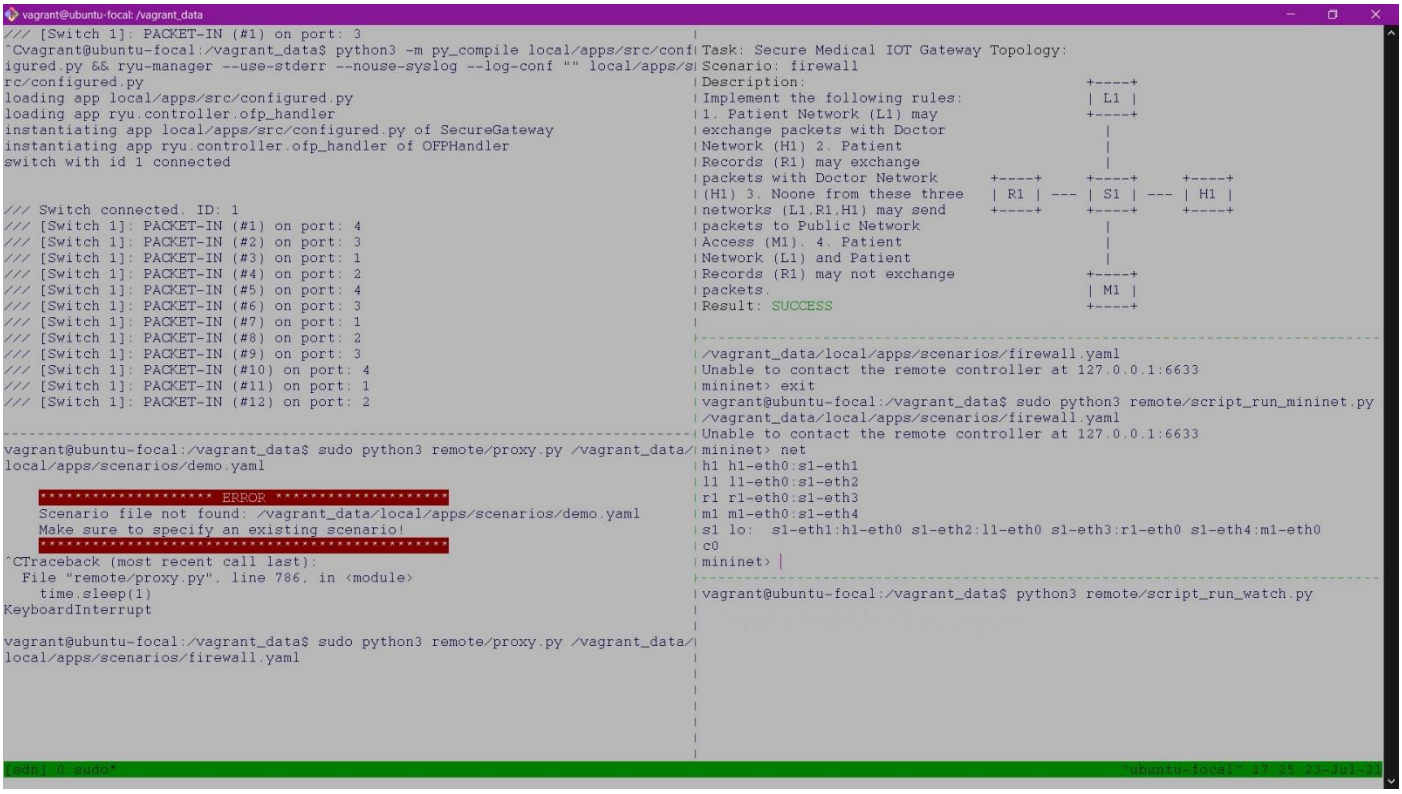

Figure 8. Depicts the successfully loaded task and *network topology in the mininet pane.*

Step 8: In the traffic generator pane replace the scenarios/demo.yaml file with the scenarios/firewall.yaml file to generate traffic in the configured network topology.

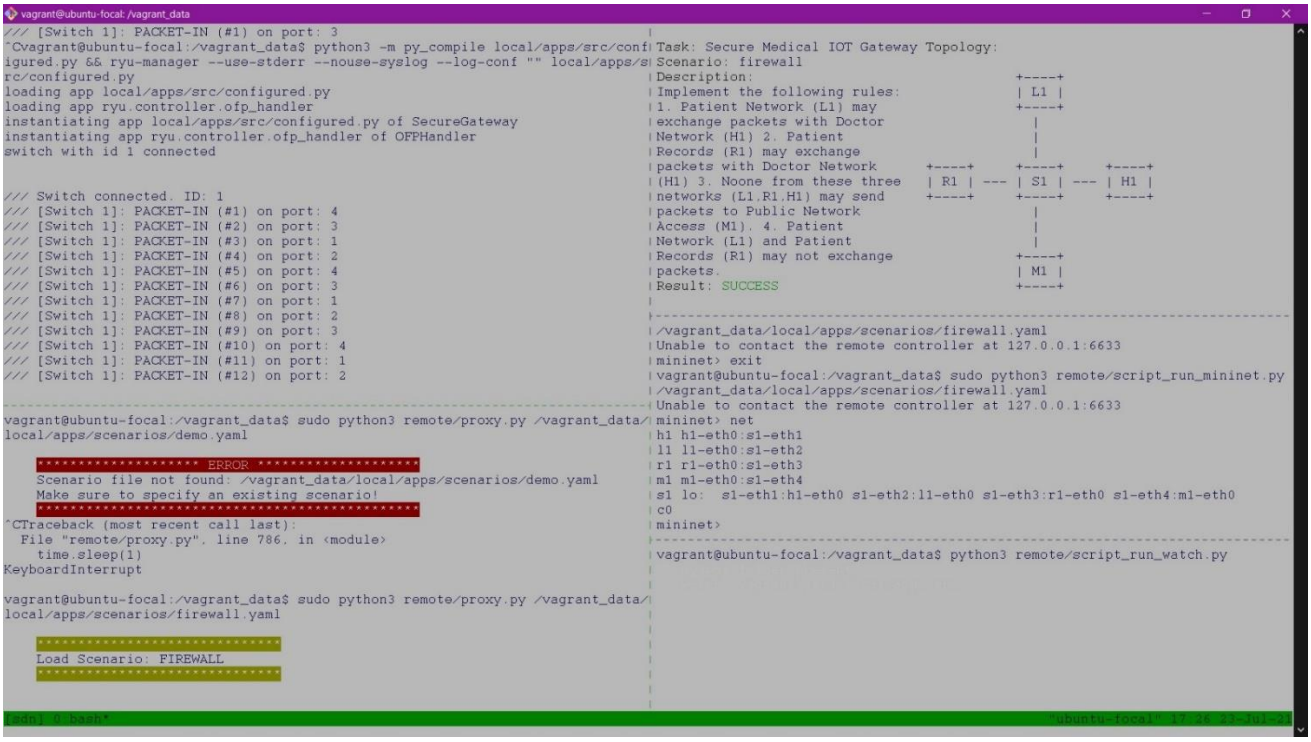

Figure 9. Depicts the firewall scenario traffic generator loaded to test the functionality of the configured controller.

Step 9: The traffic generated will then be filtered, if successful it will be indicated in the additional information panes result field and traffic generator pane status field as a 'SUCCESS' elsewise the result will be listed as a 'FAILURE' in the additional information pane and the traffic generator will describe the number of packets expected through the switch to each network.

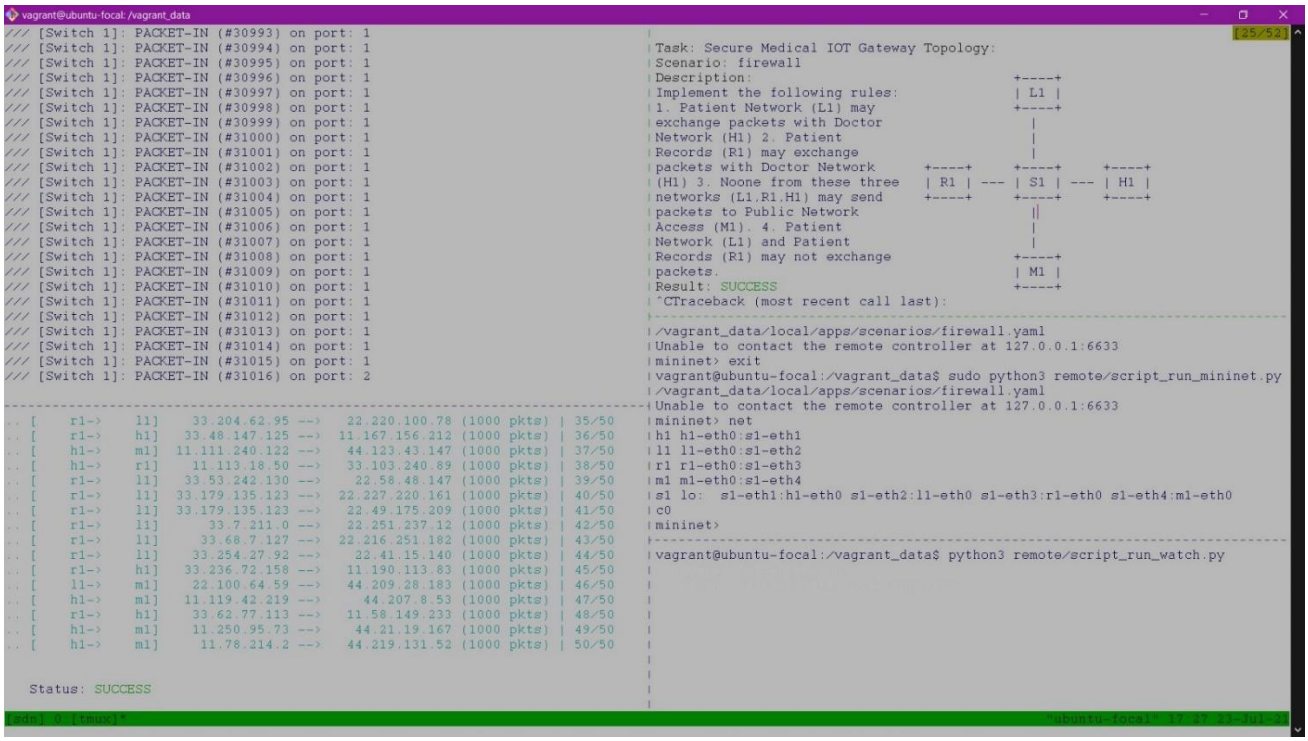

Figure 10. Depicts the configured firewall application correctly routing packages to their respective networks.

Step 10: To exit the SDN Cockpit environment input ' ctrl+b d ' and vagrant halt to close the connection to the vm.