Lab1 - ONOS + Mininet

0866007 胡孝德

Part 1: Answer Questions

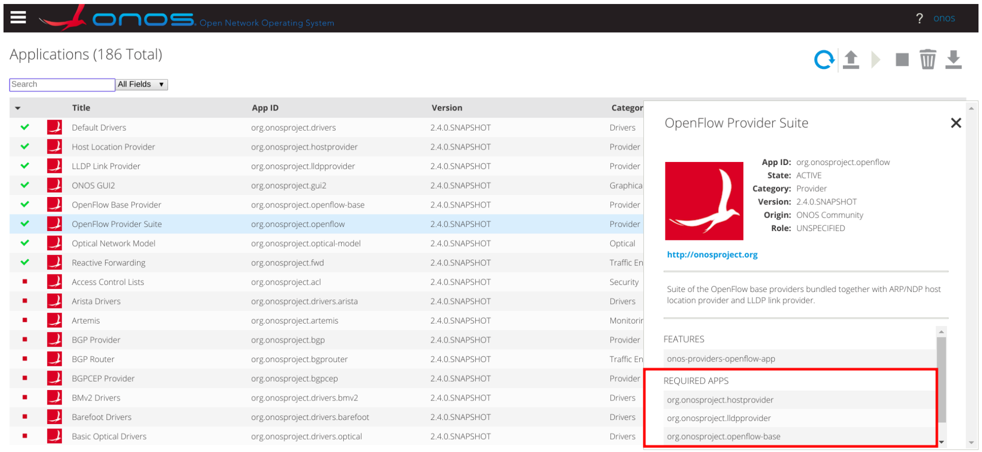
1. Activate ONOS APPS

1-1. When activating “org.nonsproject.openflow”, what are the APPs which also be activated?

- org.onosproject.hostprovider
- org.onosproject.lldpprovider
- org.onosproject.openflow-base

答案由以下節圖得知：

- GUI



Applications (186 Total)

Search All Fields ▾

		Title	App ID	Version	Categor
✓		Default Drivers	org.onosproject.drivers	2.4.0.SNAPSHOT	Drivers
✓		Host Location Provider	org.onosproject.hostprovider	2.4.0.SNAPSHOT	Provider
✓		LLDP Link Provider	org.onosproject.lldpprovider	2.4.0.SNAPSHOT	Provider
✓		ONOS GUI2	org.onosproject.gui2	2.4.0.SNAPSHOT	Graphica
✓		OpenFlow Base Provider	org.onosproject.openflow-base	2.4.0.SNAPSHOT	Provider
✓		OpenFlow Provider Suite	org.onosproject.openflow	2.4.0.SNAPSHOT	Provider
✓		Optical Network Model	org.onosproject.optical-model	2.4.0.SNAPSHOT	Optical
✓		Reactive Forwarding	org.onosproject.fwd	2.4.0.SNAPSHOT	Traffic En
■		Access Control Lists	org.onosproject.acl	2.4.0.SNAPSHOT	Security
■		Arista Drivers	org.onosproject.drivers.arista	2.4.0.SNAPSHOT	Drivers
■		Artemis	org.onosproject.artemis	2.4.0.SNAPSHOT	Monitorir
■		BGP Provider	org.onosproject.bgp	2.4.0.SNAPSHOT	Provider
■		BGP Router	org.onosproject.bgprouter	2.4.0.SNAPSHOT	Traffic En
■		BGPCEP Provider	org.onosproject.bgpcep	2.4.0.SNAPSHOT	Provider
■		BMv2 Drivers	org.onosproject.drivers.bmv2	2.4.0.SNAPSHOT	Drivers
■		Barefoot Drivers	org.onosproject.drivers.barefoot	2.4.0.SNAPSHOT	Drivers
■		Basic Optical Drivers	org.onosproject.drivers.optical	2.4.0.SNAPSHOT	Drivers

OpenFlow Provider Suite ✕

App ID: org.onosproject.openflow
State: ACTIVE
Category: Provider
Version: 2.4.0.SNAPSHOT
Origin: ONOS Community
Role: UNSPECIFIED

http://onosproject.org

Suite of the OpenFlow base providers bundled together with ARP/NDP host location provider and LLDP link provider.

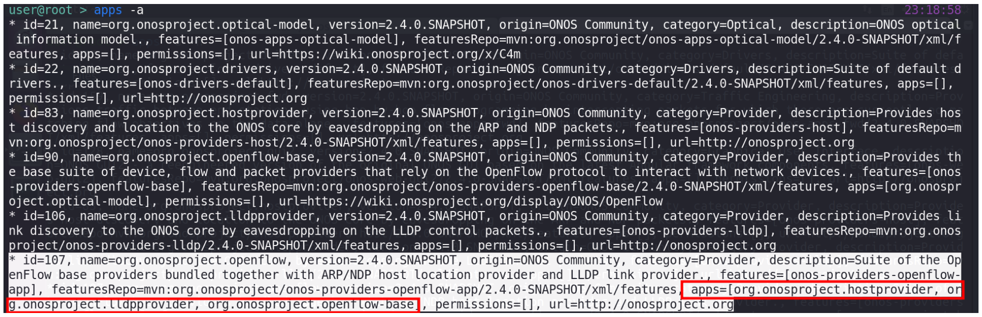
FEATURES

onos-providers-openflow-app

REQUIRED APPS

org.onosproject.hostprovider
org.onosproject.lldpprovider
org.onosproject.openflow-base

- CLI



```
user@root > apps -a                                                                                    23:18:58
* id=21, name=org.onosproject.optical-model, version=2.4.0.SNAPSHOT, origin=ONOS Community, category=Optical, description=ONOS optical
 information model., features=[onos-apps-optical-model], featuresRepo=mvn:org.onosproject/onos-apps-optical-model/2.4.0-SNAPSHOT/xml/f
eatures, apps=[], permissions=[], url=https://wiki.onosproject.org/x/C4m
* id=22, name=org.onosproject.drivers, version=2.4.0.SNAPSHOT, origin=ONOS Community, category=Drivers, description=Suite of default d
rivers., features=[onos-drivers-default], featuresRepo=mvn:org.onosproject/onos-drivers-default/2.4.0-SNAPSHOT/xml/features, apps=[],
permissions=[], url=http://onosproject.org
* id=83, name=org.onosproject.hostprovider, version=2.4.0.SNAPSHOT, origin=ONOS Community, category=Provider, description=Provides hos
t discovery and location to the ONOS core by eavesdropping on the ARP and NDP packets., features=[onos-providers-host], featuresRepo=m
vn:org.onosproject/onos-providers-host/2.4.0-SNAPSHOT/xml/features, apps=[], permissions=[], url=http://onosproject.org
* id=90, name=org.onosproject.openflow-base, version=2.4.0.SNAPSHOT, origin=ONOS Community, category=Provider, description=Provides th
e base suite of device, flow and packet providers that rely on the OpenFlow protocol to interact with network devices., features=[onos
-providers-openflow-base], featuresRepo=mvn:org.onosproject/onos-providers-openflow-base/2.4.0-SNAPSHOT/xml/features, apps=[org.onospr
oject.optical-model], permissions=[], url=https://wiki.onosproject.org/display/ONOS/OpenFlow
* id=106, name=org.onosproject.lldpprovider, version=2.4.0.SNAPSHOT, origin=ONOS Community, category=Provider, description=Provides li
nk discovery to the ONOS core by eavesdropping on the LLDP control packets., features=[onos-providers-lldp], featuresRepo=mvn:org.onos
project/onos-providers-lldp/2.4.0-SNAPSHOT/xml/features, apps=[], permissions=[], url=http://onosproject.org
* id=107, name=org.onosproject.openflow, version=2.4.0.SNAPSHOT, origin=ONOS Community, category=Provider, description=Suite of the Op
enFlow base providers bundled together with ARP/NDP host location provider and LLDP link provider., features=[onos-providers-openflow-
app], featuresRepo=mvn:org.onosproject/onos-providers-openflow-app/2.4.0-SNAPSHOT/xml/features, apps=[org.onosproject.hostprovider, or
g.onosproject.lldpprovider, org.onosproject.openflow-base], permissions=[], url=http://onosproject.org
```

1-2. Which APP enables hosts to ping each other?

- org.onosproject.fwd

當封包進入交換器時，因為沒有任何匹配的 flow，因此被阻擋通過。透過 Reactive Forwarding app，動態地將 flow 新增到 Flow table 當中，因此主機之間才有辦法 ping 對方。

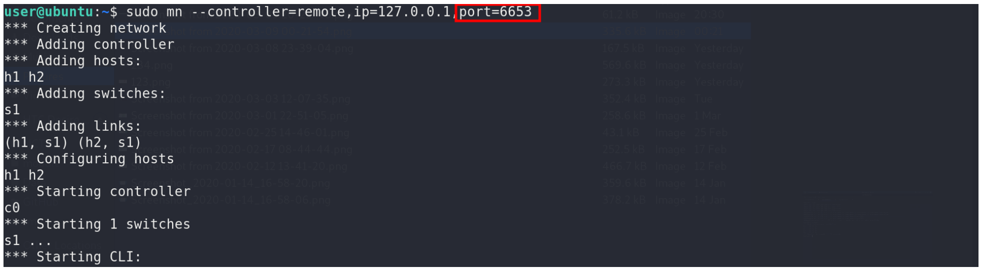
2. Observe listening port

2-1. OpenFlow protocol defines the TCP port for connection between controller and switch. What is the number of this port?

- 6653

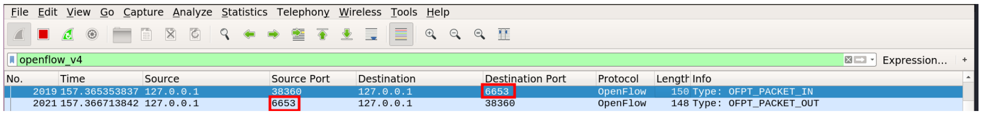
答案可由 Mininet 的 controller 參數得知，或者是透過 wireshark 封包蒐集得知。

- Mininet



```
user@ubuntu:~$ sudo mn --controller=remote,ip=127.0.0.1,port=6653
*** 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:
```

- Wireshark



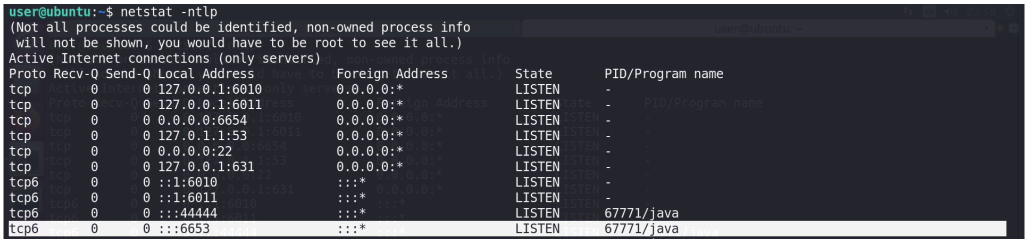
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
openflow_v4
Expression...
+

```
No.   Time            Source     Source Port  Destination  Destination Port  Protocol  Length Info
 2019 157.365353837   127.0.0.1  38360        127.0.0.1    6653              OpenFlow     150 Type: OFPT_PACKET_IN
 2021 157.366713842   127.0.0.1  6653         127.0.0.1    38360             OpenFlow     148 Type: OFPT_PACKET_OUT
```

2-2. Regarding to the previous question, which APP enables that TCP port be listening?

- org.onosproject.openflow-base

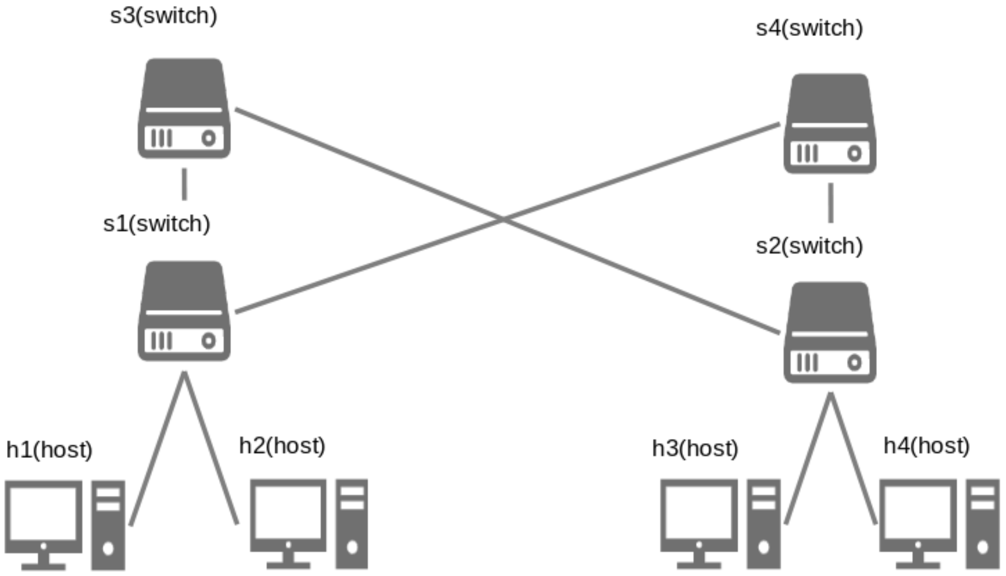
當 openflow-base app 被啟動後，可由 netstat 指令發現 port 6653 進入 LISTEN 狀態。



```
user@ubuntu:~$ netstat -ntlp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:6010          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6011          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:6654            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp6       0      0 ::1:6010                :::*                    LISTEN      -
tcp6       0      0 ::1:6011                :::*                    LISTEN      -
tcp6       0      0 :::44444                :::*                    LISTEN      67771/java
tcp6       0      0 :::6653                 :::*                    LISTEN      67771/java
```

Part 2: Naming Conventions & Command

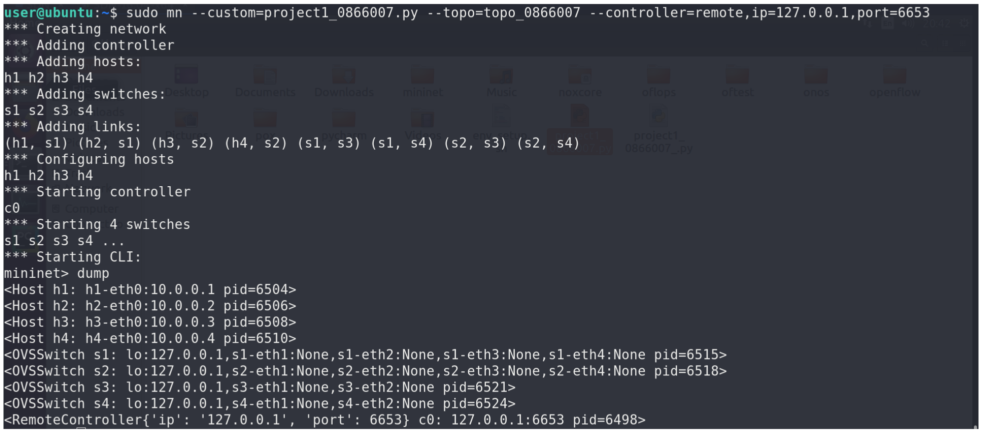
1. Edit a Python script to build the following topology:



```
from mininet.topo import Topo


class Project1_Topo_0866007(Topo):
    def __init__(self):
        Topo.__init__(self)

        # Add hosts
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        h3 = self.addHost('h3')
        h4 = self.addHost('h4')

        # Add switches
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        s4 = self.addSwitch('s4')

        # Add links
        self.addLink(h1, s1)
        self.addLink(h2, s1)
        self.addLink(h3, s2)
        self.addLink(h4, s2)
        self.addLink(s1, s3)
        self.addLink(s1, s4)
        self.addLink(s2, s3)
        self.addLink(s2, s4)


topos = {'topo_0866007': Project1_Topo_0866007}
```

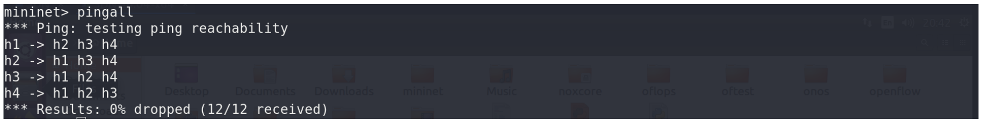
腳本驗證節圖：

- dump



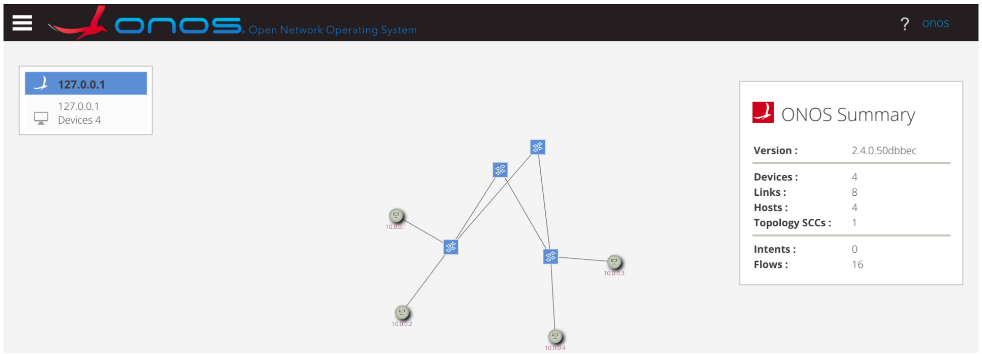
```
user@ubuntu:~$ sudo mn --custom=project1_0866007.py --topo=topo_0866007 --controller=remote,ip=127.0.0.1,port=6653
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s2) (s1, s3) (s1, s4) (s2, s3) (s2, s4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=6504>
<Host h2: h2-eth0:10.0.0.2 pid=6506>
<Host h3: h3-eth0:10.0.0.3 pid=6508>
<Host h4: h4-eth0:10.0.0.4 pid=6510>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None pid=6515>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2-eth4:None pid=6518>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=6521>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=6524>
<RemoteController{'ip': '127.0.0.1', 'port': 6653} c0: 127.0.0.1:6653 pid=6498>
```

- pingall



```
mininet> pingall
*** 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)
```

- ONOS GUI



ONOS Summary

```
Version :          2.4.0.50dbbec
Devices :          4
Links :            8
Hosts :            4
Topology SCCs :    1
Intents :          0
Flows :            16
```

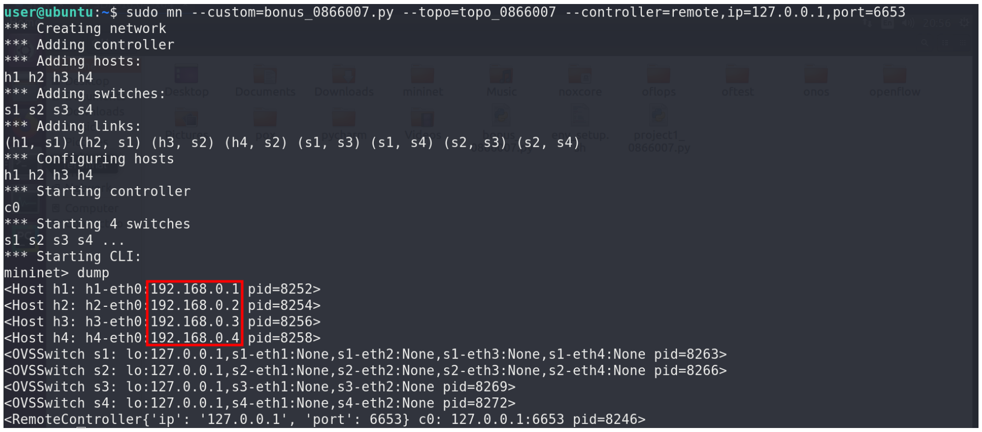
Bonus: Manually Assign Hosts IP Address In Mininet (I)

在 addHost() 加入 ip 參數即可。

```
from mininet.topo import Topo


class Project1_Topo_0866007(Topo):
    def __init__(self):
        Topo.__init__(self)

        # Add hosts
        h1 = self.addHost('h1', ip='192.168.0.1/24')
        h2 = self.addHost('h2', ip='192.168.0.2/24')
        h3 = self.addHost('h3', ip='192.168.0.3/24')
        h4 = self.addHost('h4', ip='192.168.0.4/24')

        # Add switches
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        s4 = self.addSwitch('s4')

        # Add links
        self.addLink(h1, s1)
        self.addLink(h2, s1)
        self.addLink(h3, s2)
        self.addLink(h4, s2)
        self.addLink(s1, s3)
        self.addLink(s1, s4)
        self.addLink(s2, s3)
        self.addLink(s2, s4)


topos = {'topo_0866007': Project1_Topo_0866007}
```

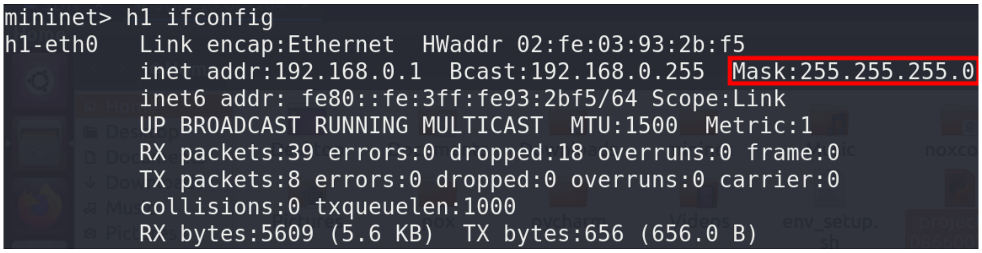
腳本驗證節圖：

- dump



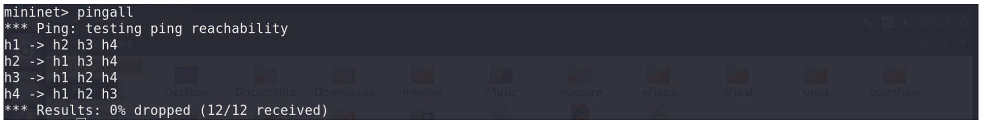
```
user@ubuntu:~$ sudo mn --custom=bonus_0866007.py --topo=topo_0866007 --controller=remote,ip=127.0.0.1,port=6653
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s2) (s1, s3) (s1, s4) (s2, s3) (s2, s4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> dump
<Host h1: h1-eth0:192.168.0.1 pid=8252>
<Host h2: h2-eth0:192.168.0.2 pid=8254>
<Host h3: h3-eth0:192.168.0.3 pid=8256>
<Host h4: h4-eth0:192.168.0.4 pid=8258>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None pid=8263>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2-eth4:None pid=8266>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=8269>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=8272>
<RemoteController{'ip': '127.0.0.1', 'port': 6653} c0: 127.0.0.1:6653 pid=8246>
```

- ifconfig



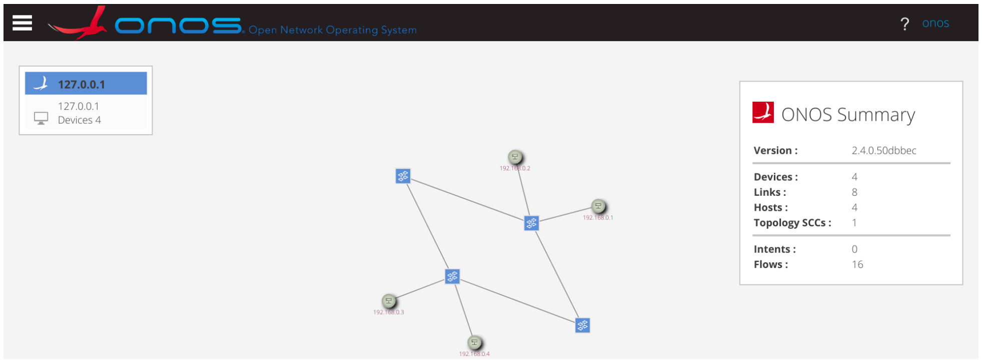
```
mininet> h1 ifconfig
h1-eth0   Link encap:Ethernet  HWaddr 02:fe:03:93:2b:f5
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::fe:3ff:fe93:2bf5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:39 errors:0 dropped:18 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5609 (5.6 KB)  TX bytes:656 (656.0 B)
```

- pingall



```
mininet> pingall
*** 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)
```

- ONOS GUI



ONOS Summary

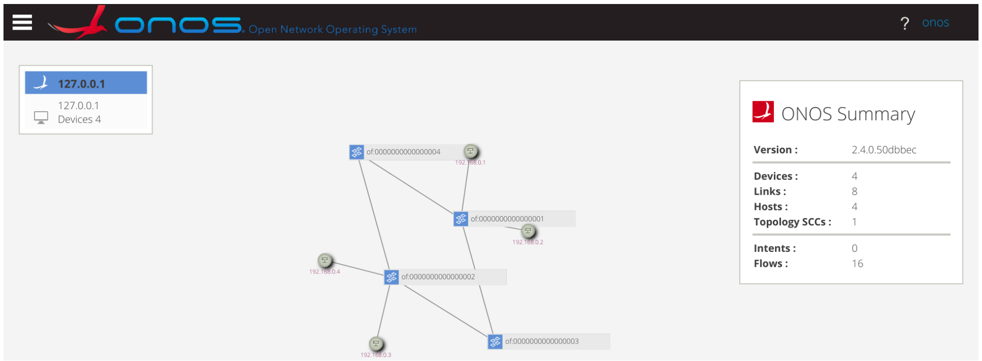
Version : 2.4.0.50dbbec

Devices : 4
Links : 8
Hosts : 4
Topology SCCs : 1

Intents : 0
Flows : 16

What I've learned or solved

1. 在 ONOS GUI topology 當中，可用 **L** 按鍵顯示每台 switch 的 device_ID，幫助識別各個交換器，如下圖所示。



ONOS Summary

Version : 2.4.0.50dbbec

Devices : 4
Links : 8
Hosts : 4
Topology SCCs : 1

Intents : 0
Flows : 16

2. 在 ONOS CLI 要啟動 APP 時，是可以不用打出全名的。舉例來說，以下兩行是等價的。

```
onos> app activate org.onosproject.fwd
onos> app activate fwd
```