SDN-NFV

0516045 張凱翔

Part1.

1.

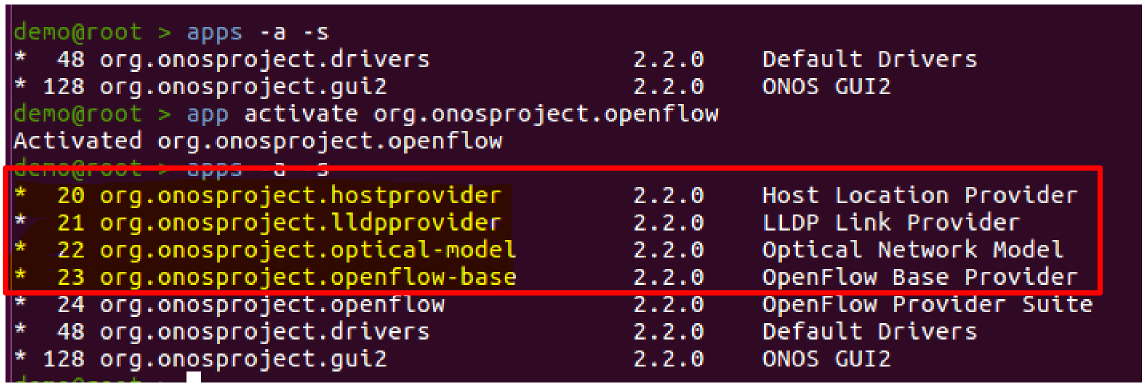
1.

org.onosproject.hostprovider

org.onosproject.lldpprovider

org.onosproject.optical-model

org.onosproject.openflow-base



```
demo@root > apps -a -s
*  48 org.onosproject.drivers                2.2.0    Default Drivers
* 128 org.onosproject.gui2                   2.2.0    ONOS GUI2
demo@root > app activate org.onosproject.openflow
Activated org.onosproject.openflow
demo@root > apps -a -s
*  20 org.onosproject.hostprovider           2.2.0    Host Location Provider
*  21 org.onosproject.lldpprovider           2.2.0    LLDP Link Provider
*  22 org.onosproject.optical-model          2.2.0    Optical Network Model
*  23 org.onosproject.openflow-base          2.2.0    OpenFlow Base Provider
*  24 org.onosproject.openflow               2.2.0    OpenFlow Provider Suite
*  48 org.onosproject.drivers                2.2.0    Default Drivers
* 128 org.onosproject.gui2                   2.2.0    ONOS GUI2
```

2.

org.onosproject.fwd

2.

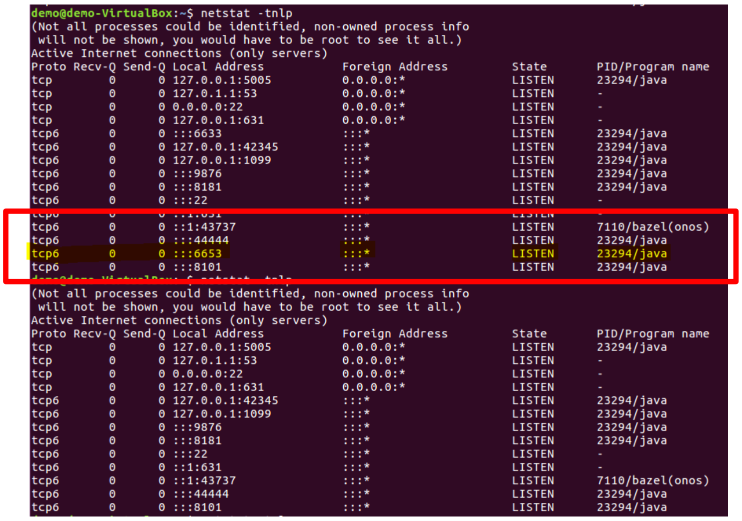
1.

Port 6653

2.

org.onosproject.openflow-base

有開跟沒開的差別



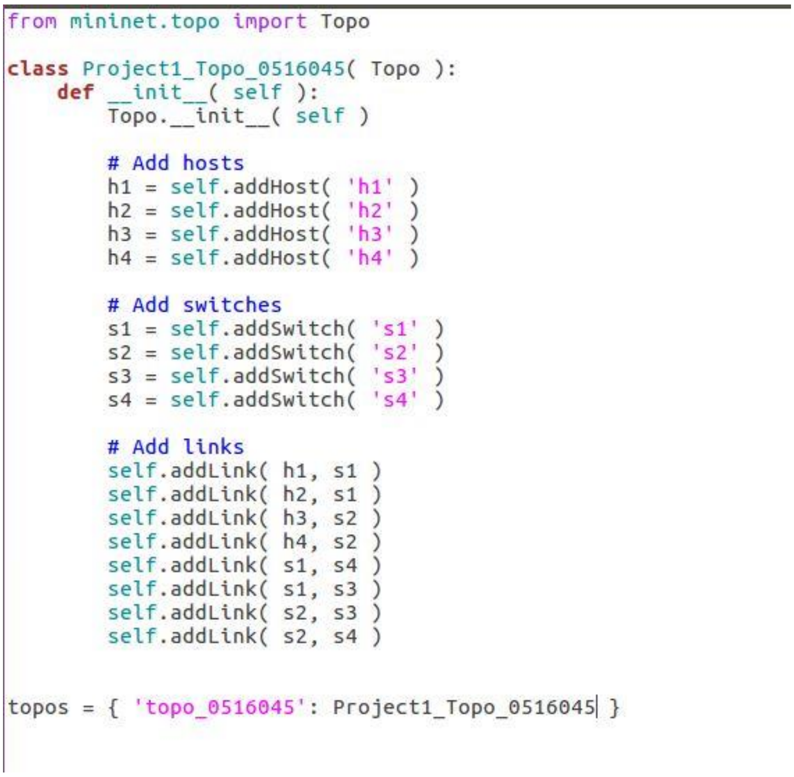
```
demo@demo-VirtualBox:~$ netstat -tnlp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:5005          0.0.0.0:*               LISTEN      23294/java
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 :::6633                 :::*                    LISTEN      23294/java
tcp6       0      0 127.0.0.1:42345         :::*                    LISTEN      23294/java
tcp6       0      0 127.0.0.1:1099          :::*                    LISTEN      23294/java
tcp6       0      0 :::9876                 :::*                    LISTEN      23294/java
tcp6       0      0 :::8181                 :::*                    LISTEN      23294/java
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 ::1:43737               :::*                    LISTEN      7110/bazel(onos)
tcp6       0      0 :::44444                :::*                    LISTEN      23294/java
tcp6       0      0 :::6653                 :::*                    LISTEN      23294/java
tcp6       0      0 :::8101                 :::*                    LISTEN      23294/java
demo@demo-VirtualBox:~$ netstat -tnlp
(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      0 127.0.0.1:5005          0.0.0.0:*               LISTEN      23294/java
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      0 127.0.0.1:42345         :::*                    LISTEN      23294/java
tcp6       0      0 127.0.0.1:1099          :::*                    LISTEN      23294/java
tcp6       0      0 :::9876                 :::*                    LISTEN      23294/java
tcp6       0      0 :::8181                 :::*                    LISTEN      23294/java
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 ::1:43737               :::*                    LISTEN      7110/bazel(onos)
tcp6       0      0 :::44444                :::*                    LISTEN      23294/java
tcp6       0      0 :::8101                 :::*                    LISTEN      23294/java
```

Part2.

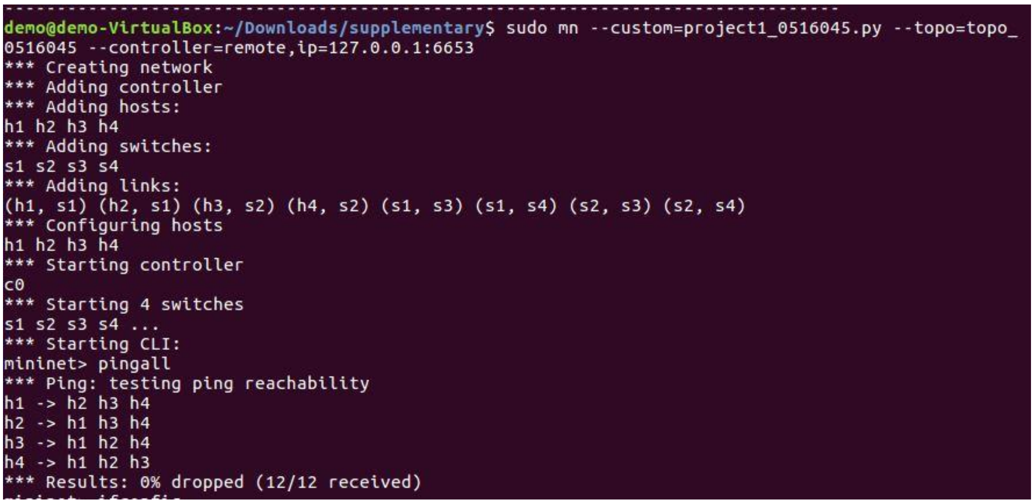
下圖是用來建 topo 的 python 檔

依照作業要求加上 host 4 個 switch 4 個 和之間的連線

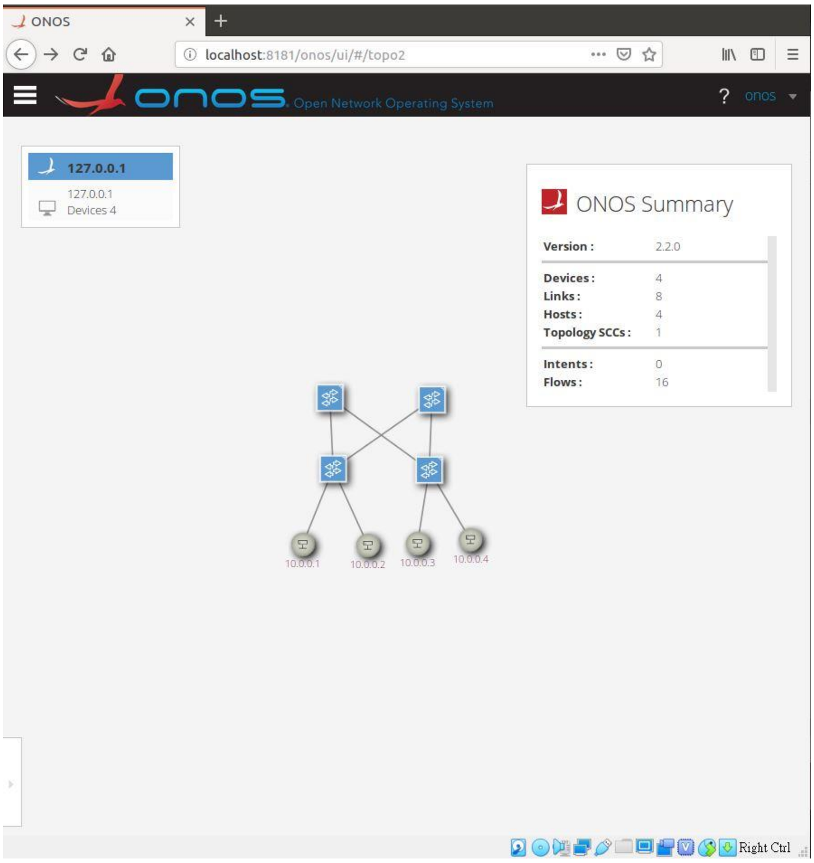
在 mininet 下 pingall 指令結果如 onos 網頁上的拓樸圖



```python
from mininet.topo import Topo

class Project1_Topo_0516045( 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, s4 )
        self.addLink( s1, s3 )
        self.addLink( s2, s3 )
        self.addLink( s2, s4 )


topos = { 'topo_0516045': Project1_Topo_0516045 }
```



```
demo@demo-VirtualBox:~/Downloads/supplementary$ sudo mn --custom=project1_0516045.py --topo=topo_
0516045 --controller=remote,ip=127.0.0.1:6653
*** 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> 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)
```



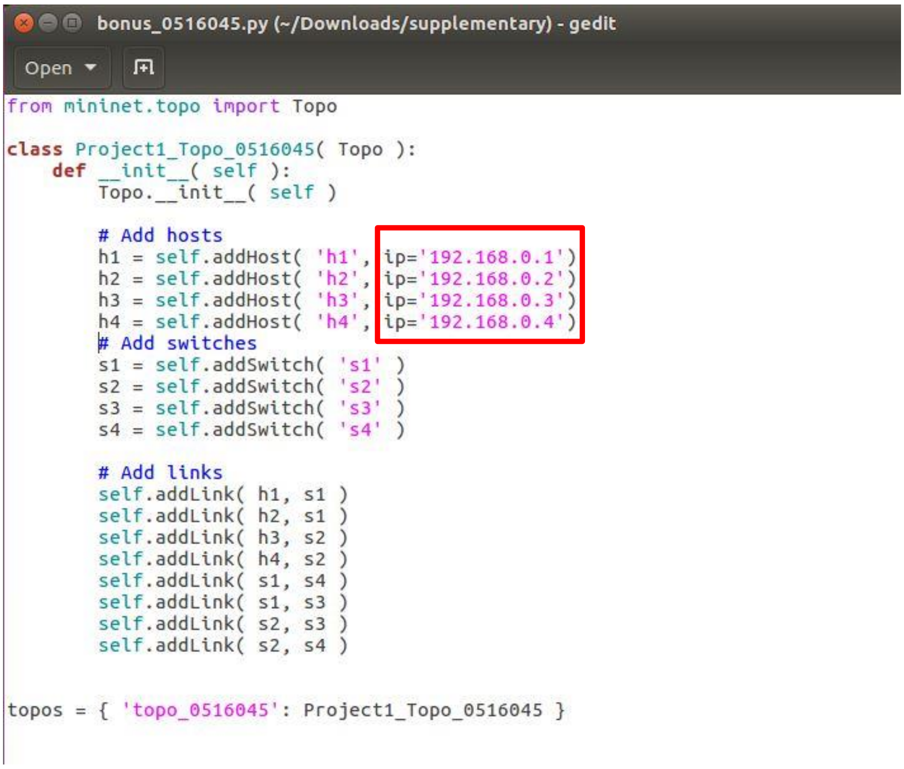
ONOS Summary

Version :	2.2.0
Devices :	4
Links :	8
Hosts :	4
Topology SCCs :	1
Intents :	0
Flows :	16

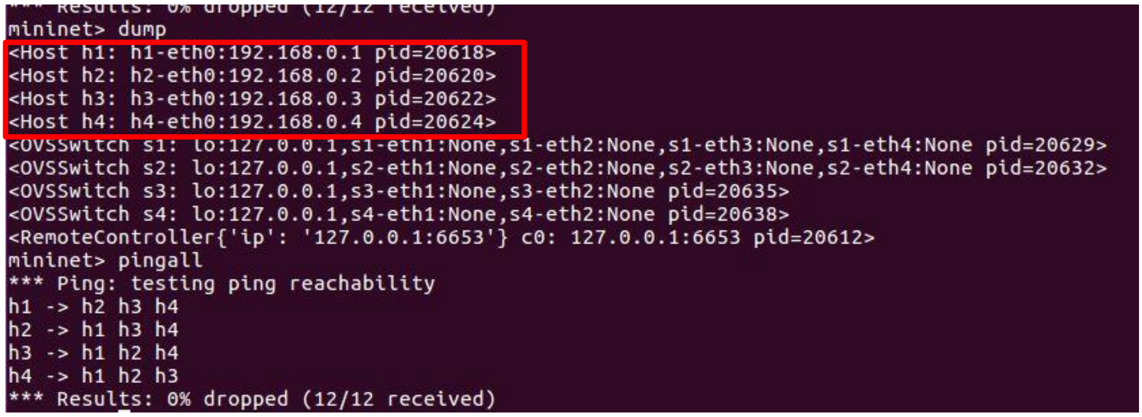
Bonus.

跟 part2 的差別為在宣告 host 的時候加上 ip

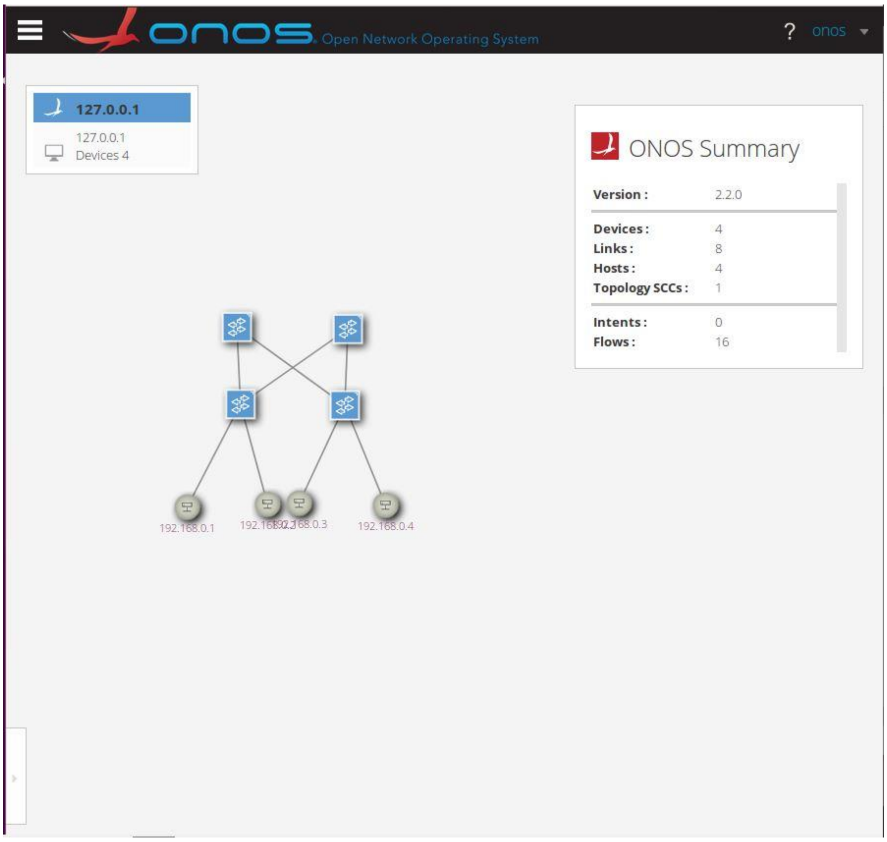
從 CLI 可以看到 ip 為 192.168.0.0/24



```
from mininet.topo import Topo

class Project1_Topo_0516045( Topo ):
    def __init__( self ):
        Topo.__init__( self )

        # Add hosts
        h1 = self.addHost( 'h1', ip='192.168.0.1')
        h2 = self.addHost( 'h2', ip='192.168.0.2')
        h3 = self.addHost( 'h3', ip='192.168.0.3')
        h4 = self.addHost( 'h4', ip='192.168.0.4')
        # 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, s4 )
        self.addLink( s1, s3 )
        self.addLink( s2, s3 )
        self.addLink( s2, s4 )


topos = { 'topo_0516045': Project1_Topo_0516045 }
```



```
*** Results: 0% dropped (12/12 received)
mininet> dump
<Host h1: h1-eth0:192.168.0.1 pid=20618>
<Host h2: h2-eth0:192.168.0.2 pid=20620>
<Host h3: h3-eth0:192.168.0.3 pid=20622>
<Host h4: h4-eth0:192.168.0.4 pid=20624>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None pid=20629>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2-eth4:None pid=20632>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=20635>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=20638>
<RemoteController{'ip': '127.0.0.1:6653'} c0: 127.0.0.1:6653 pid=20612>
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 Summary

Version : 2.2.0

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

Intents : 0
Flows : 16

Learned.

這次作業較困難的地方為安裝虛擬機時，因為電腦沒有辦法開到8GB的RAM，導致ONOS安裝的時候會有錯誤，在試著逐漸加大 RAM 之後終於成功將需要的稱是安裝完成。經過這次作業讓我對 ONOS 和 MININET 的操作有了基礎的認識。