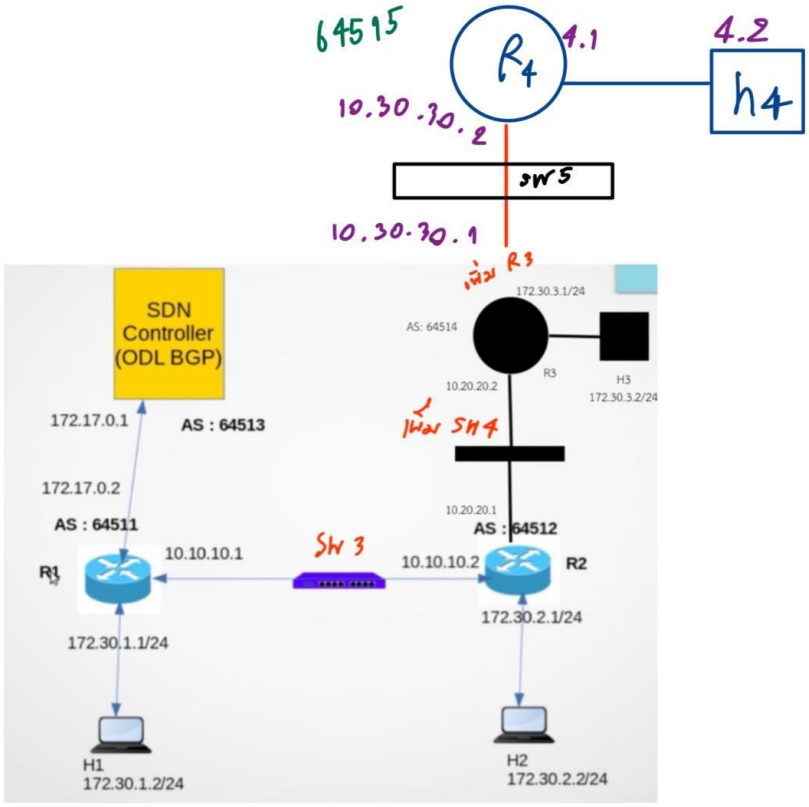
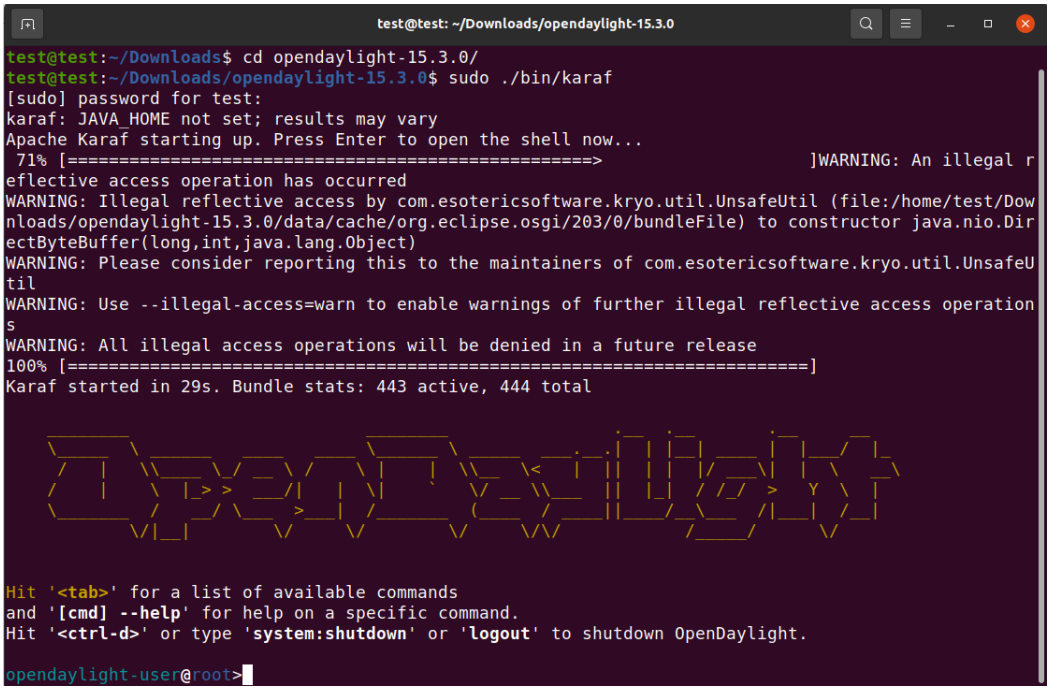
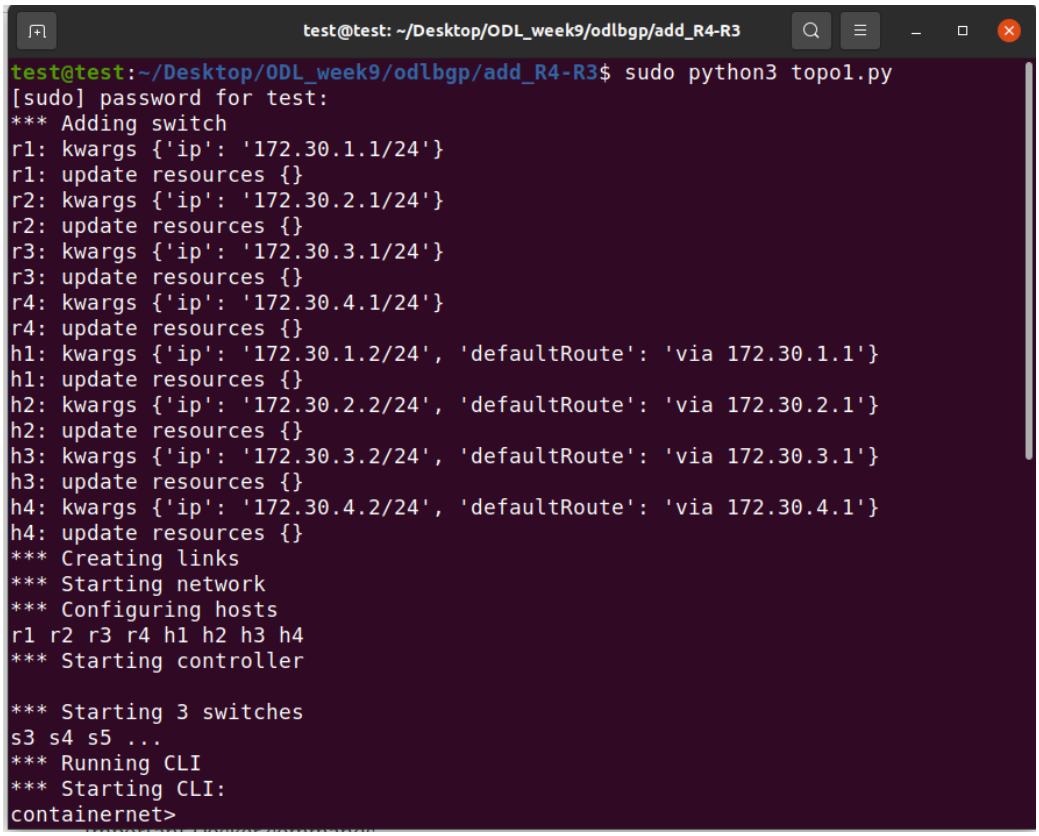
เพิ่ม R4 ต่อ จาก R3 ข้างบน



1. รัน Apache karaf

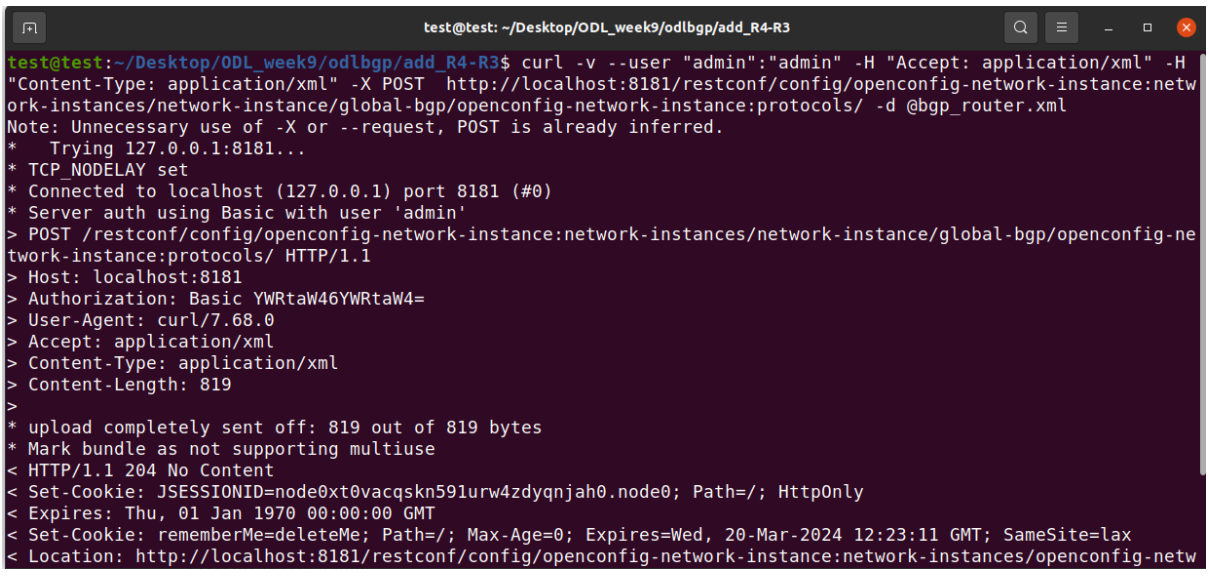


2.Topo1.py



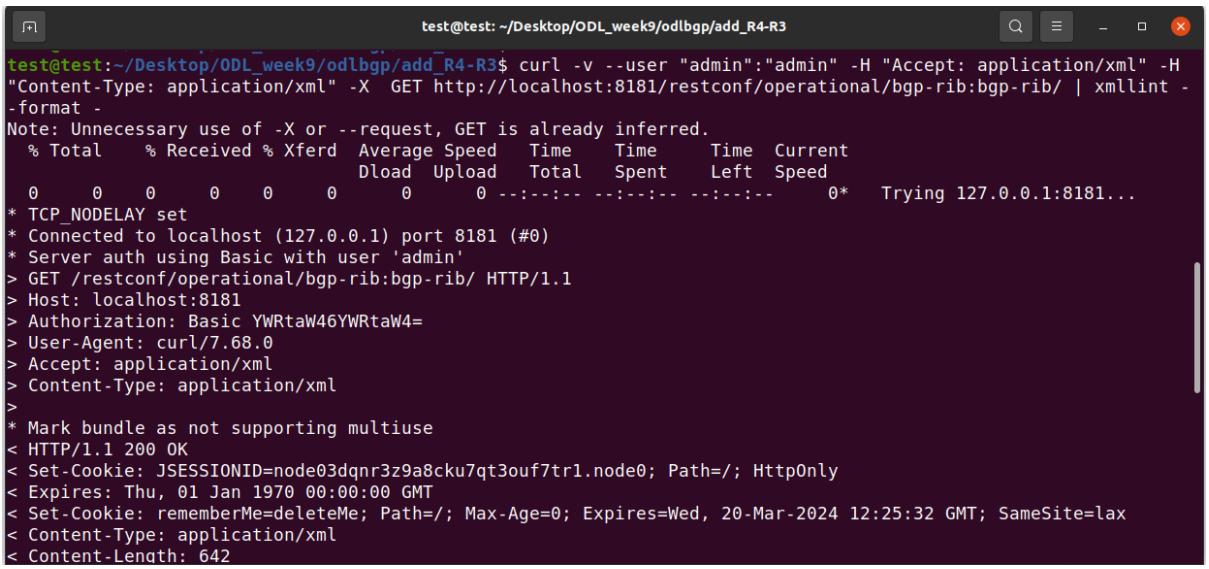
3. Create the ODL BGP Instance

ใช้คำสั่ง curl -v -u “admin”:”admin” -H “Accept: application/xml” -H “Content-Type: application/xml” -X POST http://localhost:8181/restconf/config/openconfig-network-instance:networkinstances/network-instance/global-bgp/openconfig-network-instance:protocols/ -d @bgp_router.xm



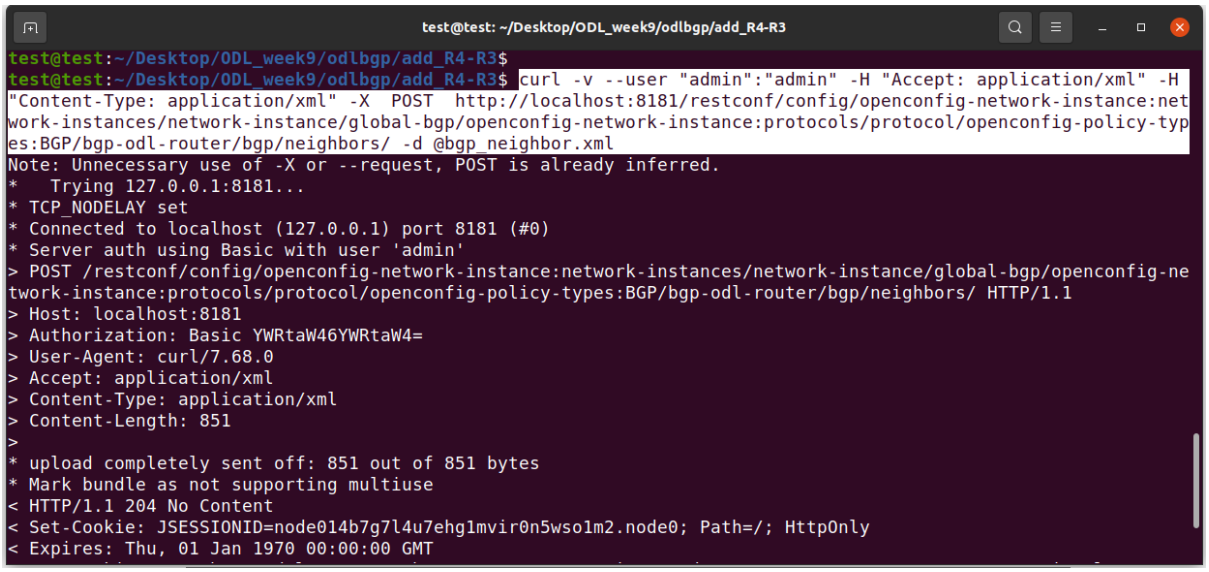
4. verification

ใช้คำสั่ง curl -v -u "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X GET http://localhost:8181/restconf/operational/bgp-rib:bgp-rib/ | xmllint --format –



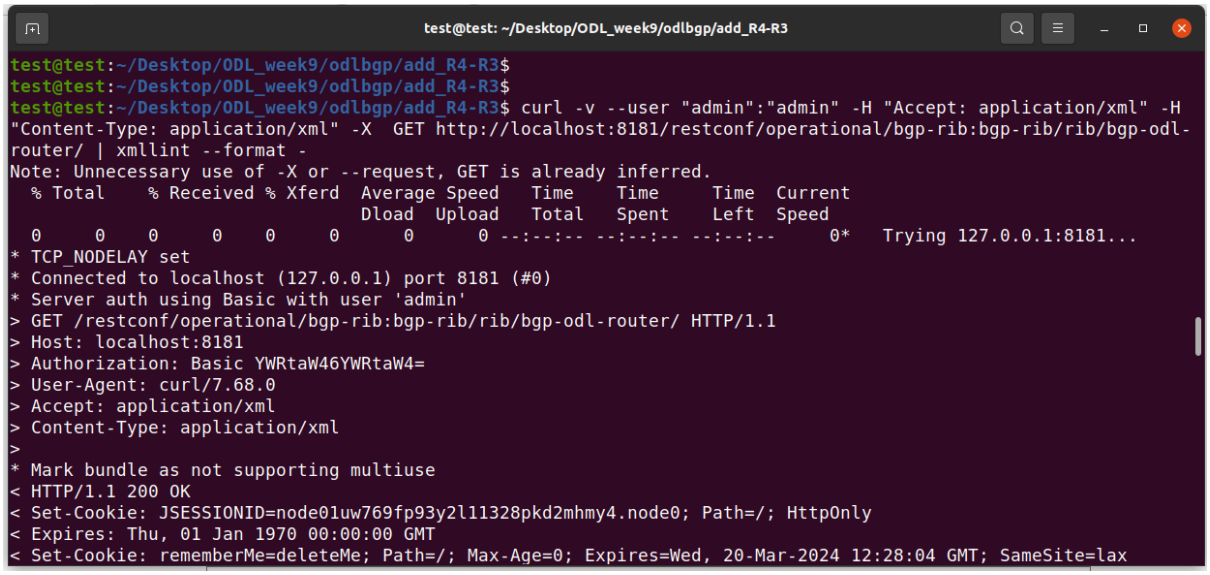
5. create bgp neighbor

ใช้คำสั่ง curl -v -u "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X POST http://localhost:8181/restconf/config/openconfig-network-instance:networkinstances/network-instance/global-bgp/openconfig-network-instance:protocols/protocol/openconfigpolicy-types:BGP/bgp-odl-router/bgp/neighbors/ -d @bgp_neighbor.xml



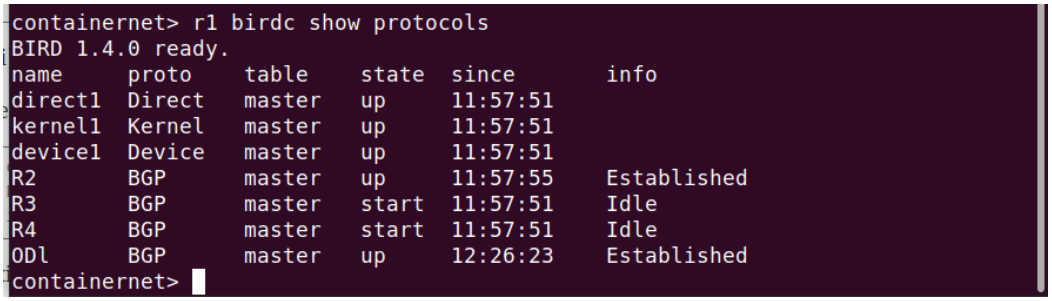
6.verify bgp neighbor/route details

ใช้คำสั่ง curl -v -u "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X GET http://localhost:8181/restconf/operational/bgp-rib:bgp-rib/bgp-odl-router/ | xmllint --format -



7. verify the bgp portss and neighbor establishmen

7.1 r1 birdc show protocol



```
containernet> r2 birdc show protocols
BIRD 1.4.0 ready.
name     proto    table    state  since       info
direct1  Direct   master   up     11:57:51
kernel1  Kernel   master   up     11:57:51
device1  Device   master   up     11:57:51
R1       BGP      master   up     11:57:55    Established
R3       BGP      master   up     11:57:55    Established
R4       BGP      master   start  11:57:51    Idle
containernet>
containernet> r3 birdc show protocols
BIRD 1.4.0 ready.
name     proto    table    state  since       info
direct1  Direct   master   up     11:57:51
kernel1  Kernel   master   up     11:57:51
device1  Device   master   up     11:57:51
R2       BGP      master   up     11:57:55    Established
R1       BGP      master   start  11:57:51    Idle
R4       BGP      master   up     11:57:55    Established
containernet>
containernet> r4 birdc show protocols
BIRD 1.4.0 ready.
name     proto    table    state  since       info
direct1  Direct   master   up     11:57:51
kernel1  Kernel   master   up     11:57:51
device1  Device   master   up     11:57:51
R2       BGP      master   start  11:57:51    Idle
R1       BGP      master   start  11:57:51    Idle
R3       BGP      master   up     11:57:55    Established
containernet>
```

7.2 r1 ip route

```
containernet> r1 ip route
default via 172.17.0.1 dev eth0
10.10.10.0/24 dev r1-eth1  proto kernel  scope link  src 10.10.10.1
10.20.20.0/24 via 10.10.10.2 dev r1-eth1  proto bird
10.30.30.0/24 via 10.10.10.2 dev r1-eth1  proto bird
172.17.0.0/16 dev eth0  proto kernel  scope link  src 172.17.0.2
172.30.1.0/24 dev r1-eth0  proto kernel  scope link  src 172.30.1.1
172.30.2.0/24 via 10.10.10.2 dev r1-eth1  proto bird
172.30.3.0/24 via 10.10.10.2 dev r1-eth1  proto bird
172.30.4.0/24 via 10.10.10.2 dev r1-eth1  proto bird
containernet>
```

```
containernet> r2 ip route
default via 172.17.0.1 dev eth0
10.10.10.0/24 dev r2-eth1  proto kernel  scope link  src 10.10.10.2
10.20.20.0/24 dev r2-eth2  proto kernel  scope link  src 10.20.20.1
10.30.30.0/24 via 10.20.20.2 dev r2-eth2  proto bird
172.17.0.0/16 dev eth0  proto kernel  scope link  src 172.17.0.3
172.30.1.0/24 via 10.10.10.1 dev r2-eth1  proto bird
172.30.2.0/24 dev r2-eth0  proto kernel  scope link  src 172.30.2.1
172.30.3.0/24 via 10.20.20.2 dev r2-eth2  proto bird
172.30.4.0/24 via 10.20.20.2 dev r2-eth2  proto bird
containernet>
```

```
containernet> r3 ip route
default via 172.17.0.1 dev eth0
10.10.10.0/24 via 10.20.20.1 dev r3-eth1  proto bird
10.20.20.0/24 dev r3-eth1  proto kernel  scope link  src 10.20.20.2
10.30.30.0/24 dev r3-eth2  proto kernel  scope link  src 10.30.30.1
172.17.0.0/16 dev eth0  proto kernel  scope link  src 172.17.0.4
172.30.1.0/24 via 10.20.20.1 dev r3-eth1  proto bird
172.30.2.0/24 via 10.20.20.1 dev r3-eth1  proto bird
172.30.3.0/24 dev r3-eth0  proto kernel  scope link  src 172.30.3.1
172.30.4.0/24 via 10.30.30.2 dev r3-eth2  proto bird
containernet>
```

```
containernet> r4 ip route
default via 172.17.0.1 dev eth0
10.10.10.0/24 via 10.30.30.1 dev r4-eth1  proto bird
10.20.20.0/24 via 10.30.30.1 dev r4-eth1  proto bird
10.30.30.0/24 dev r4-eth1  proto kernel  scope link  src 10.30.30.2
172.17.0.0/16 dev eth0  proto kernel  scope link  src 172.17.0.5
172.30.1.0/24 via 10.30.30.1 dev r4-eth1  proto bird
172.30.2.0/24 via 10.30.30.1 dev r4-eth1  proto bird
172.30.3.0/24 via 10.30.30.1 dev r4-eth1  proto bird
172.30.4.0/24 dev r4-eth0  proto kernel  scope link  src 172.30.4.1
containernet>
```

8. Check แต่ละ Host ว่า ping หากันเจอ

```
containernet> h1 ping h2
PING 172.30.2.2 (172.30.2.2) 56(84) bytes of data.
64 bytes from 172.30.2.2: icmp_seq=1 ttl=62 time=0.147 ms
64 bytes from 172.30.2.2: icmp_seq=2 ttl=62 time=0.053 ms
^C
--- 172.30.2.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1024ms
rtt min/avg/max/mdev = 0.053/0.100/0.147/0.047 ms
containernet>
```

```
containernet> h1 ping h3
PING 172.30.3.2 (172.30.3.2) 56(84) bytes of data.
64 bytes from 172.30.3.2: icmp_seq=1 ttl=61 time=0.327 ms
64 bytes from 172.30.3.2: icmp_seq=2 ttl=61 time=0.062 ms
^C
--- 172.30.3.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1015ms
rtt min/avg/max/mdev = 0.062/0.194/0.327/0.133 ms
containernet>
```

```
containernet> h1 ping h4
PING 172.30.4.2 (172.30.4.2) 56(84) bytes of data.
64 bytes from 172.30.4.2: icmp_seq=1 ttl=60 time=0.248 ms
64 bytes from 172.30.4.2: icmp_seq=2 ttl=60 time=0.064 ms
^C
--- 172.30.4.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1019ms
rtt min/avg/max/mdev = 0.064/0.156/0.248/0.092 ms
containernet>
```

สามารถปิงไปที่ 127.17.0.1

```
containernet>
containernet> h4 ping 127.17.0.1
PING 127.17.0.1 (127.17.0.1) 56(84) bytes of data.
64 bytes from 127.17.0.1: icmp_seq=1 ttl=64 time=0.055 ms
64 bytes from 127.17.0.1: icmp_seq=2 ttl=64 time=0.026 ms
^C
--- 127.17.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1004ms
rtt min/avg/max/mdev = 0.026/0.040/0.055/0.015 ms
containernet>
```

Check traceroute

```
containernet>
containernet> h1 traceroute h2
traceroute to 172.30.2.2 (172.30.2.2), 64 hops max
  1   172.30.1.1  0.002ms  0.002ms  0.001ms
  2   10.10.10.2  0.516ms  0.001ms  0.000ms
  3   172.30.2.2  0.002ms  0.000ms  0.002ms
containernet>
```

```
containernet> h1 traceroute h3
traceroute to 172.30.3.2 (172.30.3.2), 64 hops max
  1   172.30.1.1  0.003ms  0.001ms  0.001ms
  2   10.10.10.2  0.180ms  0.002ms  0.001ms
  3   10.20.20.2  0.002ms  0.001ms  0.001ms
  4   172.30.3.2  0.001ms  0.001ms  0.001ms
containernet>
```

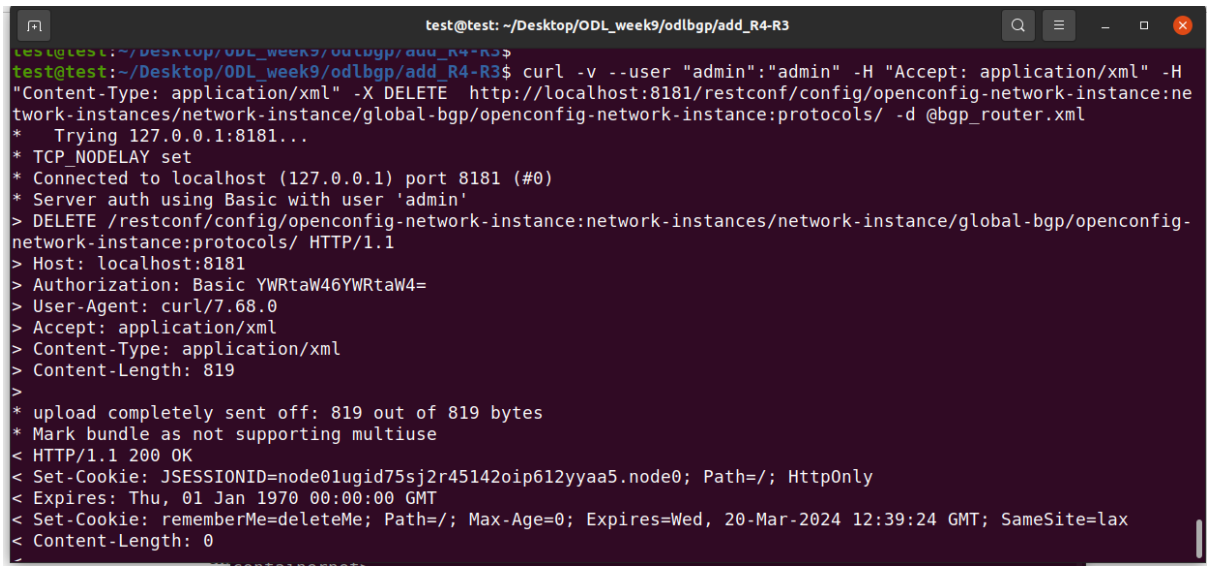
```
containernet> h1 traceroute h4
traceroute to 172.30.4.2 (172.30.4.2), 64 hops max
  1   172.30.1.1  0.002ms  0.001ms  0.002ms
  2   10.10.10.2  0.002ms  0.002ms  0.002ms
  3   10.20.20.2  0.002ms  0.000ms  0.002ms
  4   10.30.30.2  0.001ms  0.001ms  0.002ms
  5   172.30.4.2  0.002ms  0.002ms  0.002ms
containernet>
```

```
containernet>
containernet> h1 traceroute 127.17.0.1
traceroute to 127.17.0.1 (127.17.0.1), 64 hops max
  1   127.17.0.1  0.002ms  0.001ms  0.001ms
containernet>
containernet> h2 traceroute 127.17.0.1
traceroute to 127.17.0.1 (127.17.0.1), 64 hops max
  1   127.17.0.1  0.001ms  0.001ms  0.001ms
containernet>
containernet> h3 traceroute 127.17.0.1
traceroute to 127.17.0.1 (127.17.0.1), 64 hops max
  1   127.17.0.1  0.002ms  0.001ms  0.001ms
containernet>
containernet> h4 traceroute 127.17.0.1
traceroute to 127.17.0.1 (127.17.0.1), 64 hops max
  1   127.17.0.1  0.002ms  0.001ms  0.000ms
containernet>
containernet>
```

```
containernet> links
h1-eth0<->r1-eth0 (OK OK)
h2-eth0<->r2-eth0 (OK OK)
h3-eth0<->r3-eth0 (OK OK)
h4-eth0<->r4-eth0 (OK OK)
r1-eth1<->s3-eth1 (OK OK)
r2-eth1<->s3-eth2 (OK OK)
r2-eth2<->s4-eth1 (OK OK)
r3-eth1<->s4-eth2 (OK OK)
r3-eth2<->s5-eth1 (OK OK)
r4-eth1<->s5-eth2 (OK OK)
containernet>
```

9. Destrop the test (Delete the BGP)

ใช้คำสั่ง curl -v -u "admin":"admin" -H "Accept: application/xml" -H "Content-Type: application/xml" -X DELETE http://localhost:8181/restconf/config/openconfig-networkinstance:network-instances/network-instance/global-bgp/openconfig-network-instance:protocols/ -d @bgp_router.xml



10. Check หลังลบ BGP ออก

Interrupt มีการขัดจังหวะหรือหยุดลง อาจจะ

H1 ping h1

Code

Topo1.py

Open — topo1.py — ~/Desktop/ODL_week9/odlbgp/add_R4-R3 — Save

```python
from mininet.net import Containernet
from mininet.node import RemoteController, Docker, OVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Link

# topo diagram
# h1----r1---s3---r2-----h2


def topology():

    "Create a network with some docker containers acting as hosts."

    net = Containernet(controller=RemoteController)

    info('*** Adding switch\n')

    #add r
    r1 = net.addDocker('r1', ip='172.30.1.1/24', dimage="knet/urouter:1.4")
    r2 = net.addDocker('r2', ip='172.30.2.1/24', dimage="knet/urouter:1.4")
    r3 = net.addDocker('r3', ip='172.30.3.1/24', dimage="knet/urouter:1.4")
    r4 = net.addDocker('r4', ip='172.30.4.1/24', dimage="knet/urouter:1.4")

    #add h
    h1 = net.addDocker('h1', ip='172.30.1.2/24', defaultRoute='via 172.30.1.1' ,dimage="knet/host-ubuntu:1-2")
    h2 = net.addDocker('h2', ip='172.30.2.2/24', defaultRoute='via 172.30.2.1' ,dimage="knet/host-ubuntu:1-2")
    h3 = net.addDocker('h3', ip='172.30.3.2/24', defaultRoute='via 172.30.3.1' ,dimage="knet/host-ubuntu:1-2")
    h4 = net.addDocker('h4', ip='172.30.4.2/24', defaultRoute='via 172.30.4.1' ,dimage="knet/host-ubuntu:1-2")


    #add sw
    s3 = net.addSwitch('s3', failMode='standalone')
    s4 = net.addSwitch('s4', failMode='standalone')
    s5 = net.addSwitch('s5', failMode='standalone')

    info('*** Creating links\n')
```

Open — topo1.py — ~/Desktop/ODL_week9/odlb

```python
    info('*** Creating links\n')

    #add h - r
    net.addLink(h1, r1)
    net.addLink(h2, r2)
    net.addLink(h3, r3)
    net.addLink(h4, r4)


    #add r - sw
    net.addLink(r1, s3, params1={"ip": "10.10.10.1/24"})
    net.addLink(r2, s3, params1={"ip": "10.10.10.2/24"})
    net.addLink(r2, s4, params1={"ip": "10.20.20.1/24"})
    net.addLink(r3, s4, params1={"ip": "10.20.20.2/24"})

    net.addLink(r3, s5, params1={"ip": "10.30.30.1/24"})
    net.addLink(r4, s5, params1={"ip": "10.30.30.2/24"})



    info('*** Starting network\n')
    net.start()

    #copy the bird config files
    s3.cmd("sudo docker cp r1.conf mn.r1:/etc/bird.conf")
    s3.cmd("sudo docker cp r2.conf mn.r2:/etc/bird.conf")
    s3.cmd("sudo docker cp r3.conf mn.r3:/etc/bird.conf")
    s3.cmd("sudo docker cp r4.conf mn.r4:/etc/bird.conf")

    s4.cmd("sudo docker cp r1.conf mn.r1:/etc/bird.conf")
    s4.cmd("sudo docker cp r2.conf mn.r2:/etc/bird.conf")
    s4.cmd("sudo docker cp r3.conf mn.r3:/etc/bird.conf")
    s4.cmd("sudo docker cp r4.conf mn.r4:/etc/bird.conf")

```

```
70
71
72      #add r
73      r1.cmd("bird -c /etc/bird.conf")
74      r2.cmd("bird -c /etc/bird.conf")
75      r3.cmd("bird -c /etc/bird.conf")
76      r4.cmd("bird -c /etc/bird.conf")
77
78
79      info('*** Running CLI\n')
80      CLI(net)
81      info('*** Stopping network')
82      net.stop()
83
84  if __name__ == '__main__':
85      setLogLevel('info')
86      topology()
```

R1.conf

Open

```
1  log "/var/log/bird.log"  all;
2  debug protocols all
3
4  router id 10.10.10.1;
5  protocol direct {
6      interface "*";
7  }
8
9  protocol kernel {
10     learn;
11     scan time 20;
12     export all;
13     import all;
14 }
15
16
17 protocol device {
18     scan time 10;
19 }
20
21
22
23 #BGP Configuration
24
25 protocol bgp R2{
26         export all;
27         import all;
28         local as 64511;
29         neighbor 10.10.10.2 as 64512;
30 }
31
32 protocol bgp R3{
33         export all;
34         import all;
35         local as 64511;
36         neighbor 10.20.20.2 as 64514;
37 }
```

```
37 }
38
39 protocol bgp R4{
40         export all;
41         import all;
42         local as 64511;
43         neighbor 10.30.30.2 as 64515;
44 }
45
46 protocol bgp ODl{
47         export all;
48         import all;
49         local as 64511;
50         neighbor 172.17.0.1 as 64513;
51 }
```

R2.conf

Open

```
 1 log "/var/log/bird.log"  all;
 2 debug protocols all
 3
 4 router id 10.10.10.2;
 5 protocol direct {
 6     interface "*";
 7 }
 8
 9
10 protocol kernel {
11     learn;
12     scan time 20;
13     export all;
14     import all;
15 }
16
17
18 protocol device {
19     scan time 10;
20 }
21
22
23
24 #BGP Configuration
25
26 protocol bgp R1{
27         export all;
28         import all;
29         local as 64512;
30         neighbor 10.10.10.1 as 64511;
31 }
32
33 protocol bgp R3{
34         export all;
35         import all;
36         local as 64512;
37         neighbor 10.20.20.2 as 64514;
```

```
37          neighbor 10.20.20.2 as 64514;
38 }
39
40 protocol bgp R4{
41          export all;
42          import all;
43          local as 64512;
44          neighbor 10.30.30.2 as 64515;
45 }
46
```

R3.conf

```
 1 log "/var/log/bird.log"  all;
 2 debug protocols all
 3
 4 router id 10.20.20.2;
 5 protocol direct {
 6     interface "*";
 7 }
 8
 9
10 protocol kernel {
11     learn;
12     scan time 20;
13     export all;
14     import all;
15 }
16
17
18 protocol device {
19     scan time 10;
20 }
21
22
23
24 #BGP Configuration
25
26 protocol bgp R2{
27          export all;
28          import all;
29          local as 64514;
30          neighbor 10.20.20.1 as 64512;
31 }
32
33 protocol bgp R1{
34          export all;
35          import all;
36          local as 64514;
37          neighbor 10.10.10.1 as 64511;
38 }
39
40 protocol bgp R4{
41          export all;
42          import all;
43          local as 64514;
44          neighbor 10.30.30.2 as 64515;
45 }
```

R4.conf

```
log "/var/log/bird.log"  all;
debug protocols all

router id 10.30.30.2;
protocol direct {
    interface "*";
}


protocol kernel {
    learn;
    scan time 20;
    export all;
    import all;
}


protocol device {
    scan time 10;
}



#BGP Configuration

protocol bgp R2{
        export all;
        import all;
        local as 64515;
        neighbor 10.20.20.1 as 64512;
}

protocol bgp R1{
        export all;
        import all;
        local as 64515;
        neighbor 10.10.10.1 as 64511;
}

protocol bgp R3{
        export all;
        import all;
        local as 64515;
        neighbor 10.30.30.1 as 64514;
}

```