SDN Fundamentals & Techniques

Chapter 5 - Demo 2 - Using ONOS RESTful API to filter, mirror, and forward networking traffic based on OpenFlow capabilities

Jing Yan (yanj3, jing.yan@aalto.fi) Mar 10th, 2020

1 Task 1

1.1 Create a flow rule to mirror (copy) the traffic between the “Red" and the “Blue" namespaces, i.e., consider only traffic from “Red" to “Blue", to also be sent to the “Green" namespace

Before running the code

```
# Check there is no any extra rules for br-3 OVS switch
# ovs-ofctl dump-flows br-3
cookie=0x10000598babfd, duration=1515.327s, table=0, n_packets=489, n_bytes=68460,
priority=40000,dl_type=0x88cc actions=CONTROLLER:65535
cookie=0x100002eb46ec0, duration=1515.327s, table=0, n_packets=3, n_bytes=294,
priority=5,ip actions=CONTROLLER:65535
cookie=0x100006e2643fc, duration=1515.327s, table=0, n_packets=489, n_bytes=68460,
priority=40000,dl_type=0x8942 actions=CONTROLLER:65535
cookie=0x100006b70549c, duration=1515.327s, table=0, n_packets=3, n_bytes=126,
priority=40000,arp actions=CONTROLLER:65535

# In the 1st terminal, trying to capture the pakage on Green namespace
# ip netns exec green tcpdump -i veth-green icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on veth-green, link-type EN10MB (Ethernet), capture size 262144 bytes

# In the 2nd terminal, Ping blue namespace from red namespace
# ip netns exec red ping -c1 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=14.4 ms

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 14.429/14.429/14.429/0.000 ms


# Check the 1st terminal, trying to capture the pakage on Green namespace
# ip netns exec green tcpdump -i veth-green icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on veth-green, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
```

Code

```
# cat chapter5_demo2_task1_1.1.py
#!/usr/bin/env python3
"""
# ID for br-3 is of:00006a8feafb3346
# ID for br-ovs32 port 2
# ID for veth-blue-br port 3
# ID for veth-green-br port 4
# ovs-ofctl add-flow br-3
"in_port=br-ovs32,nw_src=10.0.0.2,nw_dst=10.0.0.3,action=output:veth-green-br,veth-
blue-br"
# cookie=0x0, duration=44.319s, table=0, n_packets=5, n_bytes=490,
in_port="br-ovs32" actions=output:"veth-green-br",output:"veth-blue-br"
# cookie=0x7a0000ae0f2963, duration=35.430s, table=0, n_packets=2, n_bytes=196,
priority=10,in_port="veth-blue-br",dl_src=fe:2e:ba:e2:2e:c8,dl_dst=56:70:cf:bc:17:d
8 actions=output:"br-ovs32"
# cookie=0x7a000006564d97, duration=33.503s, table=0, n_packets=1, n_bytes=98,
priority=10,in_port="veth-green-br",dl_src=f2:94:c9:5f:b2:29,dl_dst=56:70:cf:bc:17:
d8 actions=output:"br-ovs32"

curl -u onos:rocks -X POST --header "Content-Type: application/json" --header
"Accept: application/json" -d
'{"appId":"org.onosproject.core","priority":40000,"timeout":0,"isPermanent":true,"d
eviceId":"of:00006a8feafb3346","tableId":0,"tableName":"0","treatment":{"instructio
ns":[{"type":"OUTPUT","port":"3"},
{"type":"OUTPUT","port":"4"}]},"selector":{"criteria":[{"type":"IN_PORT","port":"2"
}]}}' http://100.109.0.1:8181/onos/v1/flows/of:00006a8feafb3346
curl -u onos:rocks -X POST --header "Content-Type: application/json" --header
"Accept: application/json" -d '{"appId":"org.onosproject.core",
"priority":40000,"timeout":0,"isPermanent":true,"deviceId":"of:00006a8feafb3346","t
ableId":0,"tableName":"0",
"treatment":{"instructions":[{"type":"OUTPUT","port":"2"}]},"selector":{"criteria":
[{"type":"IN_PORT","port":"3"}, {"type": "ETH_DST","mac": "56:70:cf:bc:17:d8"},
{"type": "ETH_SRC","mac": "fe:2e:ba:e2:2e:c8"}]}}'
http://100.109.0.1:8181/onos/v1/flows/of:00006a8feafb3346
curl -u onos:rocks -X POST --header "Content-Type: application/json" --header
"Accept: application/json" -d '{"appId":"org.onosproject.core",
"priority":40000,"timeout":0,"isPermanent":true,"deviceId":"of:00006a8feafb3346","t
ableId":0,"tableName":"0",
"treatment":{"instructions":[{"type":"OUTPUT","port":"2"}]},"selector":{"criteria":
[{"type":"IN_PORT","port":"4"}, {"type": "ETH_DST","mac": "56:70:cf:bc:17:d8"},
{"type": "ETH_SRC","mac": "f2:94:c9:5f:b2:29"}]}}'
http://100.109.0.1:8181/onos/v1/flows/of:00006a8feafb3346
"""
import requests, json
from requests.auth import HTTPBasicAuth

if __name__ == "__main__":
   headers = {"Content-type": "application/json", "Accept": "application/json"}
   rule0 = {"appId":"org.onosproject.core", "priority":40000,
"timeout":0,"isPermanent":"true","deviceId":"of:00006a8feafb3346","tableId":0,"tabl
eName":"0","treatment":{"instructions":[{"type":"OUTPUT","port":"3"},
{"type":"OUTPUT","port":"4"}]},"selector":{"criteria":[{"type":"IN_PORT","port":"2"
}]}}
   rule1 = {"appId":"org.onosproject.core","priority":40000,
"timeout":0,"isPermanent":"true","deviceId":"of:00006a8feafb3346","tableId":0,"tabl
eName":"0",
"treatment":{"instructions":[{"type":"OUTPUT","port":"2"}]},"selector":{"criteria":
[{"type":"IN_PORT","port":"3"}, {"type": "ETH_DST","mac": "56:70:cf:bc:17:d8"},
{"type": "ETH_SRC","mac": "fe:2e:ba:e2:2e:c8"}]}}
   rule2 = {"appId":"org.onosproject.core","priority":40000,
"timeout":0,"isPermanent":"true","deviceId":"of:00006a8feafb3346","tableId":0,"tabl
eName":"0",
"treatment":{"instructions":[{"type":"OUTPUT","port":"2"}]},"selector":{"criteria":
[{"type":"IN_PORT","port":"4"}, {"type": "ETH_DST","mac": "56:70:cf:bc:17:d8"},
{"type": "ETH_SRC","mac": "f2:94:c9:5f:b2:29"}]}}
   res0 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:00006a8feafb3346",
json.dumps(rule0), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))
   res1 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:00006a8feafb3346",
json.dumps(rule1), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))
   res2 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:00006a8feafb3346",
json.dumps(rule2), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))
```

Result

```
# python chapter5_demo2_task1_1.1.py

# ovs-ofctl dump-flows br-3
...
cookie=0x10000634fbc0b, duration=20.557s, table=0, n_packets=3, n_bytes=247,
priority=40000,in_port="br-ovs32"
actions=output:"veth-blue-br",output:"veth-green-br"
cookie=0x10000bbef4245, duration=20.550s, table=0, n_packets=0, n_bytes=0,
priority=40000,in_port="veth-blue-br",dl_src=fe:2e:ba:e2:2e:c8,dl_dst=56:70:cf:bc:1
7:d8 actions=output:"br-ovs32"
cookie=0x10000a7f498da, duration=20.544s, table=0, n_packets=0, n_bytes=0,
priority=40000,in_port="veth-green-br",dl_src=f2:94:c9:5f:b2:29,dl_dst=56:70:cf:bc:
17:d8 actions=output:"br-ovs32"

# On the 1st terminal, trying to capture the packte on Green namespace
# ip netns exec green tcpdump -i veth-green icmp

# On the 2nd terminal, ping blue namespace from red namespace
# ip netns exec red ping -c1 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=11.9 ms

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 11.969/11.969/11.969/0.000 ms

# Check the 1st terminal
# ip netns exec green tcpdump -i veth-green icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on veth-green, link-type EN10MB (Ethernet), capture size 262144 bytes
^C14:13:39.886137 IP 10.0.0.2 > 10.0.0.3: ICMP echo request, id 4653, seq 1, length
64

1 packet captured
1 packet received by filter
0 packets dropped by kernel
```

1.2 Create a flow rule to block the ICMP traffic from the “Blue" namespace to the “Red" namespace

Before running the code

```
# ip netns exec blue ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=3.79 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.792/3.792/3.792/0.000 ms
```

Code

```
# cat chapter5_demo2_task1_1.2.py
#!/usr/bin/env python3
"""
# ID for br-3 is of:00006a8feafb3346
ovs-ofctl add-flow br-3
"priority=11,icmp,nw_src=10.0.0.3,nw_dst=10.0.0.2,action=drop"
cookie=0x0, duration=9.208s, table=0, n_packets=0, n_bytes=0, idle_age=9,
icmp,nw_src=10.0.0.3,nw_dst=10.0.0.2 actions=drop
"""
```

```
import requests, json
from requests.auth import HTTPBasicAuth

if __name__ == "__main__":
   headers = {"Content-type": "application/json", "Accept": "application/json"}
   rule0 = {"appId":"org.onosproject.core", "priority":50000,
"timeout":0,"isPermanent":"true","deviceId":"of:00006a8feafb3346","tableId":0,"tabl
eName":"0","treatment":{},"selector":{"criteria":[{"type": "ETH_TYPE","ethType":
"0x800"}, {"type": "ETH_SRC","mac": "fe:2e:ba:e2:2e:c8"}, {"type": "ETH_DST","mac":
"56:70:cf:bc:17:d8"}]}}
   res0 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:00006a8feafb3346",
json.dumps(rule0), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))
```

Result

```
# python chapter5_demo2_task1_1.2.py
# ip netns exec blue ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

1.3 Create a flow rule to block the traffic to the “Red" namespace

Before running the code

```
# ip netns exec green ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=25.0 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 25.086/25.086/25.086/0.000 ms
```

Code

```
# cat chapter5_demo2_task1_1.3.py
#!/usr/bin/env python3
"""
# ID for br-1 is of:000082f382360b43
# ovs-ofctl add-flow br-1 "priority=11,arp,nw_dst=10.0.0.2 actions=drop"
# ovs-ofctl add-flow br-1 "priority=11,ip,nw_dst=10.0.0.2 actions=drop"
cookie=0x0, duration=24.271s, table=0, n_packets=0, n_bytes=0, idle_age=24,
priority=11,arp,arp_tpa=10.0.0.2 actions=drop
cookie=0x0, duration=17.343s, table=0, n_packets=1, n_bytes=98, idle_age=12,
priority=11,ip,nw_dst=10.0.0.2 actions=drop
"""
import requests, json
from requests.auth import HTTPBasicAuth

if __name__ == "__main__":
   headers = {"Content-type": "application/json", "Accept": "application/json"}
   rule0 = {"appId":"org.onosproject.core", "priority":50000,
"timeout":0,"isPermanent":"true","deviceId":"of:000082f382360b43","tableId":0,"tabl
eName":"0","treatment":{},"selector":{"criteria":[{"type": "ETH_TYPE","ethType":
"0x800"}, {"type": "ETH_DST","mac": "fa:49:73:27:d4:4d"}]}}
   res0 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:000082f382360b43",
json.dumps(rule0), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))
   rule1 = {"appId":"org.onosproject.core", "priority":50000,
"timeout":0,"isPermanent":"true","deviceId":"of:000082f382360b43","tableId":0,"tabl
eName":"0","treatment":{},"selector":{"criteria":[{"type": "ETH_TYPE","ethType":
"0x806"}, {"type": "ETH_DST","mac": "fa:49:73:27:d4:4d"}]}}
```

```
   res1 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:000082f382360b43",
json.dumps(rule1), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))
```

Result

```
# python chapter5_demo2_task1_1.3.py
# ovs-ofctl dump-flows br-1
cookie=0x10000a854c8be, duration=2.439s, table=0, n_packets=0, n_bytes=0,
priority=50000,ip,dl_dst=fa:49:73:27:d4:4d actions=drop
cookie=0x10000d7a82d05, duration=2.432s, table=0, n_packets=0, n_bytes=0,
priority=50000,arp,dl_dst=fa:49:73:27:d4:4d actions=drop
...
# ip netns exec green ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
^C
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

1.4 Create a flow rule to allow total access to the “Red" namespace from the “Green" and the “Blue" namespaces

Before running the code

```
# ip netns exec green ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
^C
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

# ip netns exec green ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
^X^C
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

Code

```
# cat chapter5_demo2_task1_1.4.py
#!/usr/bin/env python3
"""
# ID for br-3 is of:00001a984bba324d
# ID for br-1 is of:000082f382360b43
# ovs-ofctl add-flow br-1 "priority=12,arp,nw_dst=10.0.0.2,actions=normal"
# ovs-ofctl add-flow br-1 "priority=12,ip,nw_dst=10.0.0.2,actions=normal"
# ovs-ofctl add-flow br-3
"priority=12,icmp,nw_src=10.0.0.3,nw_dst=10.0.0.2,action=normal"
"""
import requests, json
from requests.auth import HTTPBasicAuth

if __name__ == "__main__":
   headers = {"Content-type": "application/json", "Accept": "application/json"}
   rule0 = {"appId":"org.onosproject.core", "priority":60000,
"timeout":0,"isPermanent":"true","deviceId":"of:00001a984bba324d","tableId":0,"tabl
eName":"0","treatment":{"instructions": [{"type":
"OUTPUT","port":"NORMAL"}]},"selector":{"criteria":[{"type": "ETH_TYPE","ethType":
"0x800"}, {"type": "ETH_SRC","mac": "1e:cc:92:a1:d4:ac"}, {"type": "ETH_DST","mac":
"fa:49:73:27:d4:4d"}]}}
   res0 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:00001a984bba324d",
json.dumps(rule0), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))

   rule1 = {"appId":"org.onosproject.core", "priority":60000,
"timeout":0,"isPermanent":"true","deviceId":"of:000082f382360b43","tableId":0,"tabl
eName":"0","treatment":{"instructions": [{"type":
"OUTPUT","port":"NORMAL"}]},"selector":{"criteria":[{"type": "ETH_TYPE","ethType":
"0x800"}, {"type": "ETH_DST","mac": "fa:49:73:27:d4:4d"}]}}
   res1 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:000082f382360b43",
json.dumps(rule0), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))
   rule2 = {"appId":"org.onosproject.core", "priority":60000,
"timeout":0,"isPermanent":"true","deviceId":"of:000082f382360b43","tableId":0,"tabl
eName":"0","treatment":{"instructions": [{"type":
"OUTPUT","port":"NORMAL"}]},"selector":{"criteria":[{"type": "ETH_TYPE","ethType":
"0x806"}, {"type": "ETH_DST","mac": "fa:49:73:27:d4:4d"}]}}
   res2 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:000082f382360b43",
json.dumps(rule1), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))
```

Result

```
# python chapter5_demo2_task1_1.4.py
# ovs-ofctl dump-flows br-3
cookie=0x100002368fb1a, duration=13.154s, table=0, n_packets=0, n_bytes=0,
priority=60000,ip,dl_src=1e:cc:92:a1:d4:ac,dl_dst=fa:49:73:27:d4:4d actions=NORMAL
...
# ovs-ofctl dump-flows br-1
cookie=0x100009c6a8320, duration=15.927s, table=0, n_packets=0, n_bytes=0,
priority=60000,ip,dl_dst=fa:49:73:27:d4:4d actions=NORMAL
...
# ip netns exec blue ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=16.1 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 16.113/16.113/16.113/0.000 ms

# ip netns exec green ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=6.37 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 6.372/6.372/6.372/0.000 ms
```

1.5 Do you need to delete the previously created flow rules to activate flow rule number 3?

No need, I can just let the traffic pass by setting high priority to “Normal” action (priority=60000) than “Drop” action(priority=50000).

1.6 Create a flow rule to allow only HTTP and HTTPS traffic to the “Red" namespace

In this question, I deleted all the rules for all OVS devices and at the same time created a flow rule again to block the traffic to the “red“ namespace.

```
# curl -u onos:rocks -X DELETE
http://100.109.0.1:8181/onos/v1/flows/of:00001a984bba324d/281475570793242
...
# curl -u onos:rocks -X DELETE
http://100.109.0.1:8181/onos/v1/flows/of:000082f382360b43/281478594833669
...
```

Before running the code

```
# python chapter5_demo2_task1_1.3.py
# ip netns exec blue telnet 10.0.0.2 80
Trying 10.0.0.2...
^C
# ip netns exec blue telnet 10.0.0.2 443
Trying 10.0.0.2...
^C
```

Code

```
#!/usr/bin/env python3
import requests, json
from requests.auth import HTTPBasicAuth

"""
ID for br-1 is of:000082f382360b43
# ovs-ofctl add-flow br-1 "priority=12,tcp,nw_dst=10.0.0.2
actions=mod_tp_dst:443,normal"
# ovs-ofctl add-flow br-1 "priority=12,tcp,nw_dst=10.0.0.2
actions=mod_tp_dst:80,normal"
cookie=0x0, duration=5s, table=0, n_packets=1, n_bytes=74,
priority=12,tcp,nw_dst=10.0.0.2 actions=mod_tp_dst:443,NORMAL
cookie=0x0, duration=80.515s, table=0, n_packets=1, n_bytes=74,
priority=12,tcp,nw_dst=10.0.0.2 actions=mod_tp_dst:80,NORMAL
"""

if __name__ == "__main__":
   headers = {"Content-type": "application/json", "Accept": "application/json"}
   rule0 = {"appId":"org.onosproject.core", "priority":60000,
"timeout":0,"isPermanent":"true","deviceId":"of:000082f382360b43","tableId":0,"tabl
eName":"0","treatment":{"instructions": [{"type":
"OUTPUT","port":"NORMAL"}]},"selector":{"criteria":[{"type": "IP_PROTO",
"protocol": 80}, {"type": "IPV4_SRC","ip": "10.0.0.2/32"}]}}
   res0 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:000082f382360b43",
json.dumps(rule0), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))
   rule1 = {"appId":"org.onosproject.core", "priority":60000,
"timeout":0,"isPermanent":"true","deviceId":"of:000082f382360b43","tableId":0,"tabl
eName":"0","treatment":{"instructions": [{"type":
"OUTPUT","port":"NORMAL"}]},"selector":{"criteria":[{"type": "IP_PROTO",
"protocol": 443}, {"type": "IPV4_SRC","ip": "10.0.0.2/32"}]}}
   res1 =
requests.post("http://100.109.0.1:8181/onos/v1/flows/of:000082f382360b43",
json.dumps(rule1), headers = headers, auth = HTTPBasicAuth('onos', 'rocks'))
   print(res0.status_code)
```

Result

```
# curl -u onos:rocks http://100.109.0.1:8181/onos/v1/flows/of:000082f382360b43
[{"groupId":0,"state":"PENDING_ADD","life":0,"liveType":"UNKNOWN","lastSeen":161570
3599063,"packets":0,"bytes":0,"id":"281475936926722","appId":"org.onosproject.core"
,"priority":60000,"timeout":0,"isPermanent":true,"deviceId":"of:000082f382360b43","
tableId":0,"tableName":"0","treatment":{"instructions":[{"type":"OUTPUT","port":"2"
}],"deferred":[]},"selector":{"criteria":[{"type":"IP_PROTO","protocol":80},{"type"
:"IPV4_SRC","ip":"10.0.0.2/32"}]}},
{"groupId":0,"state":"PENDING_ADD","life":0,"liveType":"UNKNOWN","lastSeen":1615703
892089,"packets":0,"bytes":0,"id":"281476572516500","appId":"org.onosproject.core",
"priority":60000,"timeout":0,"isPermanent":true,"deviceId":"of:000082f382360b43","t
ableId":0,"tableName":"0","treatment":{"instructions":[{"type":"OUTPUT","port":"NOR
MAL"}],"deferred":[]},"selector":{"criteria":[{"type":"IP_PROTO","protocol":187},{"
type":"IPV4_SRC","ip":"10.0.0.2/32"}]}},

# ip netns exec blue telnet 10.0.0.2 443
Trying 10.0.0.2...
telnet: Unable to connect to remote host: Connection refused
```

```
# ip netns exec blue telnet 10.0.0.2 80
Trying 10.0.0.2...
telnet: Unable to connect to remote host: Connection refused
```

2 Task 2

2.1 Create a python program to list all flow rules per device id, the code must be generic for any given network topology

Code

```
# cat chapter5_demo2_task2_2.1.py
#!/usr/bin/env python3
import requests, json
from requests.auth import HTTPBasicAuth
if __name__ == "__main__":
  # Lists all infrastructure devices
  res1 = requests.get("http://100.109.0.1:8181/onos/v1/devices", auth =
HTTPBasicAuth('onos', 'rocks'))
  devices_list = res1.json()["devices"]
  for index, value in enumerate(devices_list):
      # Lists the flow rules based on deviceid
      res2 = requests.get("http://100.109.0.1:8181/onos/v1/flows/%s" % value["id"],
auth = HTTPBasicAuth('onos', 'rocks'))
      print("-----------------------------------------------------------")
      print("-----Present the %s device's flow rules as follows: \n%s" %
(value["id"], res2.json()))
```

Result

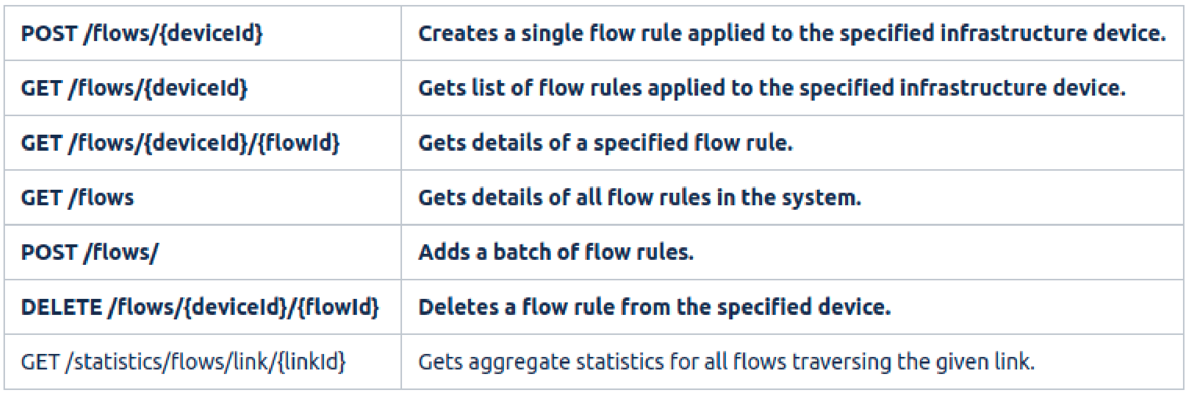
```
# python chapter5_demo2_task2_2.1.py
-----------------------------------------------------------
-----Present the of:0000367f176c9b44 device's flow rules as follows:
{'flows': [{'groupId': 0, 'state': 'ADDED', 'life': 7975, 'liveType': 'UNKNOWN',
'lastSeen': 1615704282000, 'packets': 6, 'bytes': 252, 'id': '281477819964420',
'appId': 'org.onosproject.core', 'priority': 40000, 'timeout': 0, 'isPermanent':
True, 'deviceId': 'of:0000367f176c9b44', 'tableId': 0, 'tableName': '0',
'treatment': {'instructions': [{'type': 'OUTPUT', 'port': 'CONTROLLER'}],
'clearDeferred': True, 'deferred': []}, 'selector': {'criteria': [{'type':
'ETH_TYPE', 'ethType': '0x806'}]}}, {'groupId': 0, 'state': 'ADDED', 'life': 7975,
'liveType': 'UNKNOWN', 'lastSeen': 1615704282000, 'packets': 26, 'bytes': 2548,
'id': '281475053555114', 'appId': 'org.onosproject.core', 'priority': 5, 'timeout':
0, 'isPermanent': True, 'deviceId': 'of:0000367f176c9b44', 'tableId': 0,
'tableName': '0', 'treatment': {'instructions': [{'type': 'OUTPUT', 'port':
'CONTROLLER'}], 'clearDeferred': True, 'deferred': []}, 'selector': {'criteria':
[{'type': 'ETH_TYPE', 'ethType': '0x800'}]}}, {'groupId': 0, 'state': 'ADDED',
'life': 7975, 'liveType': 'UNKNOWN', 'lastSeen': 1615704282000, 'packets': 5144,
'bytes': 720160, 'id': '281475813195846', 'appId': 'org.onosproject.core',
'priority': 40000, 'timeout': 0, 'isPermanent': True, 'deviceId':
'of:0000367f176c9b44', 'tableId': 0, 'tableName': '0', 'treatment':
{'instructions': [{'type': 'OUTPUT', 'port': 'CONTROLLER'}], 'clearDeferred': True,
'deferred': []}, 'selector': {'criteria': [{'type': 'ETH_TYPE', 'ethType':
'0x88cc'}]}}, {'groupId': 0, 'state': 'ADDED', 'life': 7975, 'liveType': 'UNKNOWN',
'lastSeen': 1615704282000, 'packets': 5144, 'bytes': 720160, 'id':
'281476902779985', 'appId': 'org.onosproject.core', 'priority': 40000, 'timeout':
0, 'isPermanent': True, 'deviceId': 'of:0000367f176c9b44', 'tableId': 0,
'tableName': '0', 'treatment': {'instructions': [{'type': 'OUTPUT', 'port':
'CONTROLLER'}], 'clearDeferred': True, 'deferred': []}, 'selector': {'criteria':
[{'type': 'ETH_TYPE', 'ethType': '0x8942'}]}}]}
-----------------------------------------------------------
-----Present the of:000082f382360b43 device's flow rules as follows:
```

```
{'flows': [{'groupId': 0, 'state': 'PENDING_ADD', 'life': 0, 'liveType': 'UNKNOWN',
'lastSeen': 1615703599063, 'packets': 0, 'bytes': 0, 'id': '281475936926722',
'appId': 'org.onosproject.core', 'priority': 60000, 'timeout': 0, 'isPermanent':
True, 'deviceId': 'of:000082f382360b43', 'tableId': 0, 'tableName': '0',
'treatment': {'instructions': [{'type': 'OUTPUT', 'port': '2'}], 'deferred': []},
'selector': {'criteria': [{'type': 'IP_PROTO', 'protocol': 80}, {'type':
'IPV4_SRC', 'ip': '10.0.0.2/32'}]}}, {'groupId': 0, 'state': 'ADDED', 'life': 5755,
'liveType': 'UNKNOWN', 'lastSeen': 1615704282000, 'packets': 0, 'bytes': 0, 'id':
'281478594833669', 'appId': 'org.onosproject.core', 'priority': 50000, 'timeout':
0, 'isPermanent': True, 'deviceId': 'of:000082f382360b43', 'tableId': 0,
'tableName': '0', 'treatment': {'instructions': [{'type': 'NOACTION'}], 'deferred':
[]}, 'selector': {'criteria': [{'type': 'ETH_DST', 'mac': 'FA:49:73:27:D4:4D'},
{'type': 'ETH_TYPE', 'ethType': '0x806'}]}}, {'groupId': 0, 'state': 'PENDING_ADD',
'life': 0, 'liveType': 'UNKNOWN', 'lastSeen': 1615703892089, 'packets': 0, 'bytes':
0, 'id': '281476572516500', 'appId': 'org.onosproject.core', 'priority': 60000,
'timeout': 0, 'isPermanent': True, 'deviceId': 'of:000082f382360b43', 'tableId': 0,
'tableName': '0', 'treatment': {'instructions': [{'type': 'OUTPUT', 'port':
'NORMAL'}], 'deferred': []}, 'selector': {'criteria': [{'type': 'IP_PROTO',
'protocol': 187}, {'type': 'IPV4_SRC', 'ip': '10.0.0.2/32'}]}}, {'groupId': 0,
'state': 'ADDED', 'life': 5755, 'liveType': 'UNKNOWN', 'lastSeen': 1615704282000,
'packets': 0, 'bytes': 0, 'id': '281477800839358', 'appId': 'org.onosproject.core',
'priority': 50000, 'timeout': 0, 'isPermanent': True, 'deviceId':
'of:000082f382360b43', 'tableId': 0, 'tableName': '0', 'treatment':
{'instructions': [{'type': 'NOACTION'}], 'deferred': []}, 'selector': {'criteria':
[{'type': 'ETH_DST', 'mac': 'FA:49:73:27:D4:4D'}, {'type': 'ETH_TYPE', 'ethType':
'0x800'}]}}, {'groupId': 0, 'state': 'ADDED', 'life': 7975, 'liveType': 'UNKNOWN',
'lastSeen': 1615704282000, 'packets': 22, 'bytes': 2156, 'id': '281477377757939',
'appId': 'org.onosproject.core', 'priority': 5, 'timeout': 0, 'isPermanent': True,
'deviceId': 'of:000082f382360b43', 'tableId': 0, 'tableName': '0', 'treatment':
{'instructions': [{'type': 'OUTPUT', 'port': 'CONTROLLER'}], 'clearDeferred': True,
'deferred': []}, 'selector': {'criteria': [{'type': 'ETH_TYPE', 'ethType':
'0x800'}]}}]}
-----------------------------------------------------------
-----Present the of:00001a984bba324d device's flow rules as follows:
{'flows': [{'groupId': 0, 'state': 'ADDED', 'life': 7975, 'liveType': 'UNKNOWN',
'lastSeen': 1615704282044, 'packets': 2572, 'bytes': 360080, 'id':
'281475335974199', 'appId': 'org.onosproject.core', 'priority': 40000, 'timeout':
0, 'isPermanent': True, 'deviceId': 'of:00001a984bba324d', 'tableId': 0,
'tableName': '0', 'treatment': {'instructions': [{'type': 'OUTPUT', 'port':
'CONTROLLER'}], 'clearDeferred': True, 'deferred': []}, 'selector': {'criteria':
[{'type': 'ETH_TYPE', 'ethType': '0x8942'}]}}, {'groupId': 0, 'state': 'ADDED',
'life': 7975, 'liveType': 'UNKNOWN', 'lastSeen': 1615704282044, 'packets': 25,
'bytes': 2450, 'id': '281477548129622', 'appId': 'org.onosproject.core',
'priority': 5, 'timeout': 0, 'isPermanent': True, 'deviceId':
'of:00001a984bba324d', 'tableId': 0, 'tableName': '0', 'treatment':
{'instructions': [{'type': 'OUTPUT', 'port': 'CONTROLLER'}], 'clearDeferred': True,
'deferred': []}, 'selector': {'criteria': [{'type': 'ETH_TYPE', 'ethType':
'0x800'}]}}]}
```

2.2 Create a python program to list flow rules by application id

Based on the API info provide by https://wiki.onosproject.org/display/ONOS/Appendix+B%3A+REST+API, I can not find any method to get flow rules by application id.

Flow



POST /flows/{deviceId}	**Creates a single flow rule applied to the specified infrastructure device.**
GET /flows/{deviceId}	**Gets list of flow rules applied to the specified infrastructure device.**
GET /flows/{deviceId}/{flowId}	**Gets details of a specified flow rule.**
GET /flows	**Gets details of all flow rules in the system.**
POST /flows/	**Adds a batch of flow rules.**
DELETE /flows/{deviceId}/{flowId}	**Deletes a flow rule from the specified device.**
GET /statistics/flows/link/{linkId}	Gets aggregate statistics for all flows traversing the given link.

2.3 Create a python program to delete flow rules knowing the device id and the flow-id

Code

```
# cat chapter5_demo2_task2_2.3.py
#!/usr/bin/env python3
import requests
from requests.auth import HTTPBasicAuth

if __name__ == "__main__":
   device_id = input("Pls input the device id: ")
   flow_id = input("pls input the flow id: ")
   url = "http://100.109.0.1:8181/onos/v1/flows/" + device_id + "/" + flow_id
   print(url)
   # Get the flow rule info based on device_id and flow_id
   res1 = requests.get(url, auth = HTTPBasicAuth('onos', 'rocks'))
   print(res1.json())
   print("-------------------------------")
   # Delete the flow rule info based on device_id and flow_id
   res2 = requests.delete(url, auth = HTTPBasicAuth('onos', 'rocks'))
   # Check whether the flow rule info based on device_id and flow_id is deleted
   res3 = requests.get(url, auth = HTTPBasicAuth('onos', 'rocks'))
   print(res3.json())
```

Result

```
# python chapter5_demo2_task2_2.3.py
Pls input the device id: of:000082f382360b43
pls input the flow id: 281477800839358
http://100.109.0.1:8181/onos/v1/flows/of:000082f382360b43/281477800839358
{'flows': [{'groupId': 0, 'state': 'ADDED', 'life': 7440, 'liveType': 'UNKNOWN',
'lastSeen': 1615705967000, 'packets': 0, 'bytes': 0, 'id': '281477800839358',
'appId': 'org.onosproject.core', 'priority': 50000, 'timeout': 0, 'isPermanent':
True, 'deviceId': 'of:000082f382360b43', 'tableId': 0, 'tableName': '0',
'treatment': {'instructions': [{'type': 'NOACTION'}], 'deferred': []}, 'selector':
{'criteria': [{'type': 'ETH_DST', 'mac': 'FA:49:73:27:D4:4D'}, {'type': 'ETH_TYPE',
'ethType': '0x800'}]}}]}
-------------------------------
{'flows': []}
```