SDN DEMO PREP

Iteration 4

By

Kabiru Sanusi

Getting Started with ONOS CLI

- **Setup Java Environment (first time only)**

- **Spin UP ONOS**
  - Connect ONOS CLI : **ssh -l karaf localhost -p 8101**
  - Password: **karaf**
  - Check current installed features in ONOS:
    - **karaf@root> apps -s -a**
  - Installed required features:
    - **karaf@root> app activate org.onosproject.fwd**
    - **karaf@root> app activate org.onosproject.openflow**
  - Check current installed features again:
    - **karaf@root> app -s -a**
  - See figure below:

```
File  Machine  View  Input  Devices  Help
ks-sdn@sdn_server:~$ ssh -l karaf localhost -p 8101
Password authentication
Password:
Welcome to Open Network Operating System (ONOS)!
    ____  _  ______  ____
   / __ \/ |/ / __ \/ __/
  / /_/ /    / /_/ /\ \
  \____/_/|_/\____/___/

Documentation: wiki.onosproject.org
Tutorials:     tutorials.onosproject.org
Mailing lists: lists.onosproject.org

Come help out! Find out how at: contribute.onosproject.org

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'logout' to exit ONOS session.

karaf@root > apps -s -a
*  55 org.onosproject.drivers             2.2.3    Default Drivers
* 182 org.onosproject.gui2                2.2.3    ONOS GUI2
karaf@root > app activate org.onosproject.fwd
Activated org.onosproject.fwd
karaf@root > app activate org.onosproject.openflow
Activated org.onosproject.openflow
karaf@root > apps -s -a
*  15 org.onosproject.optical-model       2.2.3    Optical Network Model
*  42 org.onosproject.hostprovider        2.2.3    Host Location Provider
*  43 org.onosproject.lldpprovider        2.2.3    LLDP Link Provider
*  44 org.onosproject.openflow-base       2.2.3    OpenFlow Base Provider
*  45 org.onosproject.openflow            2.2.3    OpenFlow Provider Suite
*  55 org.onosproject.drivers             2.2.3    Default Drivers
* 176 org.onosproject.fwd                 2.2.3    Reactive Forwarding
* 182 org.onosproject.gui2                2.2.3    ONOS GUI2
karaf@root >
```

- **Setup Mininet topology & test**
  - **$ sudo mn –topo torus,3,3 –mac –controller=remote,ip=192.168.1.54,port=6633 –switch ovs,protocols=OpenFlow13**
  - Note: ip address might differ in another VM
  - Perform dump command:
    - **mininet> dump**
  - Perform pingall command:
    - **mininet> pingall**
  - Figures below show the procedure:

```
File  Machine  View  Input  Devices  Help
ks-sdn@sdn_server:~$ sudo mn --topo torus,3,3 --mac --controller=remote,ip=192.168.1.54,port=6633 --
switch ovs,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1x1 h1x2 h1x3 h2x1 h2x2 h2x3 h3x1 h3x2 h3x3
*** Adding switches:
s1x1 s1x2 s1x3 s2x1 s2x2 s2x3 s3x1 s3x2 s3x3
*** Adding links:
(h1x1, s1x1) (h1x2, s1x2) (h1x3, s1x3) (h2x1, s2x1) (h2x2, s2x2) (h2x3, s2x3) (h3x1, s3x1) (h3x2, s3
x2) (h3x3, s3x3) (s1x1, s1x2) (s1x1, s2x1) (s1x2, s1x3) (s1x2, s2x2) (s1x3, s1x1) (s1x3, s2x3) (s2x1
, s2x2) (s2x1, s3x1) (s2x2, s2x3) (s2x2, s3x2) (s2x3, s2x1) (s2x3, s3x3) (s3x1, s1x1) (s3x1, s3x2) (
s3x2, s1x2) (s3x2, s3x3) (s3x3, s1x3) (s3x3, s3x1)
*** Configuring hosts
h1x1 h1x2 h1x3 h2x1 h2x2 h2x3 h3x1 h3x2 h3x3
*** Starting controller
c0
*** Starting 9 switches
s1x1 s1x2 s1x3 s2x1 s2x2 s2x3 s3x1 s3x2 s3x3 ...
*** Starting CLI:
mininet>
```

```
mininet> dump
<Host h1x1: h1x1-eth0:10.0.0.1 pid=4161>
<Host h1x2: h1x2-eth0:10.0.0.2 pid=4163>
<Host h1x3: h1x3-eth0:10.0.0.3 pid=4165>
<Host h2x1: h2x1-eth0:10.0.0.4 pid=4167>
<Host h2x2: h2x2-eth0:10.0.0.5 pid=4169>
<Host h2x3: h2x3-eth0:10.0.0.6 pid=4171>
<Host h3x1: h3x1-eth0:10.0.0.7 pid=4173>
<Host h3x2: h3x2-eth0:10.0.0.8 pid=4175>
<Host h3x3: h3x3-eth0:10.0.0.9 pid=4177>
<OVSSwitch{'protocols': 'OpenFlow13'} s1x1: lo:127.0.0.1,s1x1-eth1:None,s1x1-eth2:None,s1x1-eth3:Non
e,s1x1-eth4:None,s1x1-eth5:None pid=4182>
<OVSSwitch{'protocols': 'OpenFlow13'} s1x2: lo:127.0.0.1,s1x2-eth1:None,s1x2-eth2:None,s1x2-eth3:Non
e,s1x2-eth4:None,s1x2-eth5:None pid=4185>
<OVSSwitch{'protocols': 'OpenFlow13'} s1x3: lo:127.0.0.1,s1x3-eth1:None,s1x3-eth2:None,s1x3-eth3:Non
e,s1x3-eth4:None,s1x3-eth5:None pid=4188>
<OVSSwitch{'protocols': 'OpenFlow13'} s2x1: lo:127.0.0.1,s2x1-eth1:None,s2x1-eth2:None,s2x1-eth3:Non
e,s2x1-eth4:None,s2x1-eth5:None pid=4191>
<OVSSwitch{'protocols': 'OpenFlow13'} s2x2: lo:127.0.0.1,s2x2-eth1:None,s2x2-eth2:None,s2x2-eth3:Non
e,s2x2-eth4:None,s2x2-eth5:None pid=4194>
<OVSSwitch{'protocols': 'OpenFlow13'} s2x3: lo:127.0.0.1,s2x3-eth1:None,s2x3-eth2:None,s2x3-eth3:Non
e,s2x3-eth4:None,s2x3-eth5:None pid=4197>
<OVSSwitch{'protocols': 'OpenFlow13'} s3x1: lo:127.0.0.1,s3x1-eth1:None,s3x1-eth2:None,s3x1-eth3:Non
e,s3x1-eth4:None,s3x1-eth5:None pid=4200>
<OVSSwitch{'protocols': 'OpenFlow13'} s3x2: lo:127.0.0.1,s3x2-eth1:None,s3x2-eth2:None,s3x2-eth3:Non
e,s3x2-eth4:None,s3x2-eth5:None pid=4203>
<OVSSwitch{'protocols': 'OpenFlow13'} s3x3: lo:127.0.0.1,s3x3-eth1:None,s3x3-eth2:None,s3x3-eth3:Non
e,s3x3-eth4:None,s3x3-eth5:None pid=4206>
<RemoteController{'ip': '192.168.1.54', 'port': 6633} c0: 192.168.1.54:6633 pid=4155>
mininet> _
```

```
e,s3x2-eth4:None,s3x2-eth5:None pid=4203>
<OVSSwitch{'protocols': 'OpenFlow13'} s3x3: lo:127.0.0.1,s3x3-eth1:None,s3x3-eth2:None,s3x3-eth3:Non
e,s3x3-eth4:None,s3x3-eth5:None pid=4206>
<RemoteController{'ip': '192.168.1.54', 'port': 6633} c0: 192.168.1.54:6633 pid=4155>
mininet> pingall
*** Ping: testing ping reachability
h1x1 -> h1x2 h1x3 h2x1 h2x2 h2x3 h3x1 h3x2 h3x3
h1x2 -> h1x1 h1x3 h2x1 h2x2 h2x3 h3x1 h3x2 h3x3
h1x3 -> h1x1 h1x2 h2x1 h2x2 h2x3 h3x1 h3x2 h3x3
h2x1 -> h1x1 h1x2 h1x3 h2x2 h2x3 h3x1 h3x2 h3x3
h2x2 -> h1x1 h1x2 h1x3 h2x1 h2x3 h3x1 h3x2 h3x3
h2x3 -> h1x1 h1x2 h1x3 h2x1 h2x2 h3x1 h3x2 h3x3
h3x1 -> h1x1 h1x2 h1x3 h2x1 h2x2 h2x3 h3x2 h3x3
h3x2 -> h1x1 h1x2 h1x3 h2x1 h2x2 h2x3 h3x1 h3x3
h3x3 -> h1x1 h1x2 h1x3 h2x1 h2x2 h2x3 h3x1 h3x2
*** Results: 0% dropped (72/72 received)
mininet>
```

- **Perfom some verification commands in onos CLI**
  - **karaf@root> devices**
  - **karaf@root> links**
  - **karaf@root> flows**

```
karaf@root > devices
id=of:0000000000000101, available=true, local-status=connected 3m34s ago, role=MASTER, type=SWITCH,
mfr=Nicira, Inc., hw=Open vSwitch, sw=2.9.8, serial=None, chassis=101, driver=ovs, channelId=192.168
.1.54:49100, managementAddress=192.168.1.54, protocol=OF_13
id=of:0000000000000102, available=true, local-status=connected 3m34s ago, role=MASTER, type=SWITCH,
mfr=Nicira, Inc., hw=Open vSwitch, sw=2.9.8, serial=None, chassis=102, driver=ovs, channelId=192.168
.1.54:49102, managementAddress=192.168.1.54, protocol=OF_13
id=of:0000000000000103, available=true, local-status=connected 3m33s ago, role=MASTER, type=SWITCH,
mfr=Nicira, Inc., hw=Open vSwitch, sw=2.9.8, serial=None, chassis=103, driver=ovs, channelId=192.168
.1.54:49108, managementAddress=192.168.1.54, protocol=OF_13
id=of:0000000000000201, available=true, local-status=connected 3m34s ago, role=MASTER, type=SWITCH,
mfr=Nicira, Inc., hw=Open vSwitch, sw=2.9.8, serial=None, chassis=201, driver=ovs, channelId=192.168
.1.54:49098, managementAddress=192.168.1.54, protocol=OF_13
id=of:0000000000000202, available=true, local-status=connected 3m34s ago, role=MASTER, type=SWITCH,
mfr=Nicira, Inc., hw=Open vSwitch, sw=2.9.8, serial=None, chassis=202, driver=ovs, channelId=192.168
.1.54:49094, managementAddress=192.168.1.54, protocol=OF_13
id=of:0000000000000203, available=true, local-status=connected 3m34s ago, role=MASTER, type=SWITCH,
mfr=Nicira, Inc., hw=Open vSwitch, sw=2.9.8, serial=None, chassis=203, driver=ovs, channelId=192.168
.1.54:49104, managementAddress=192.168.1.54, protocol=OF_13
id=of:0000000000000301, available=true, local-status=connected 3m34s ago, role=MASTER, type=SWITCH,
mfr=Nicira, Inc., hw=Open vSwitch, sw=2.9.8, serial=None, chassis=301, driver=ovs, channelId=192.168
.1.54:49096, managementAddress=192.168.1.54, protocol=OF_13
id=of:0000000000000302, available=true, local-status=connected 3m33s ago, role=MASTER, type=SWITCH,
mfr=Nicira, Inc., hw=Open vSwitch, sw=2.9.8, serial=None, chassis=302, driver=ovs, channelId=192.168
.1.54:49110, managementAddress=192.168.1.54, protocol=OF_13
id=of:0000000000000303, available=true, local-status=connected 3m33s ago, role=MASTER, type=SWITCH,
mfr=Nicira, Inc., hw=Open vSwitch, sw=2.9.8, serial=None, chassis=303, driver=ovs, channelId=192.168
.1.54:49106, managementAddress=192.168.1.54, protocol=OF_13
karaf@root >
```

```
karaf@root >
karaf@root > links
```

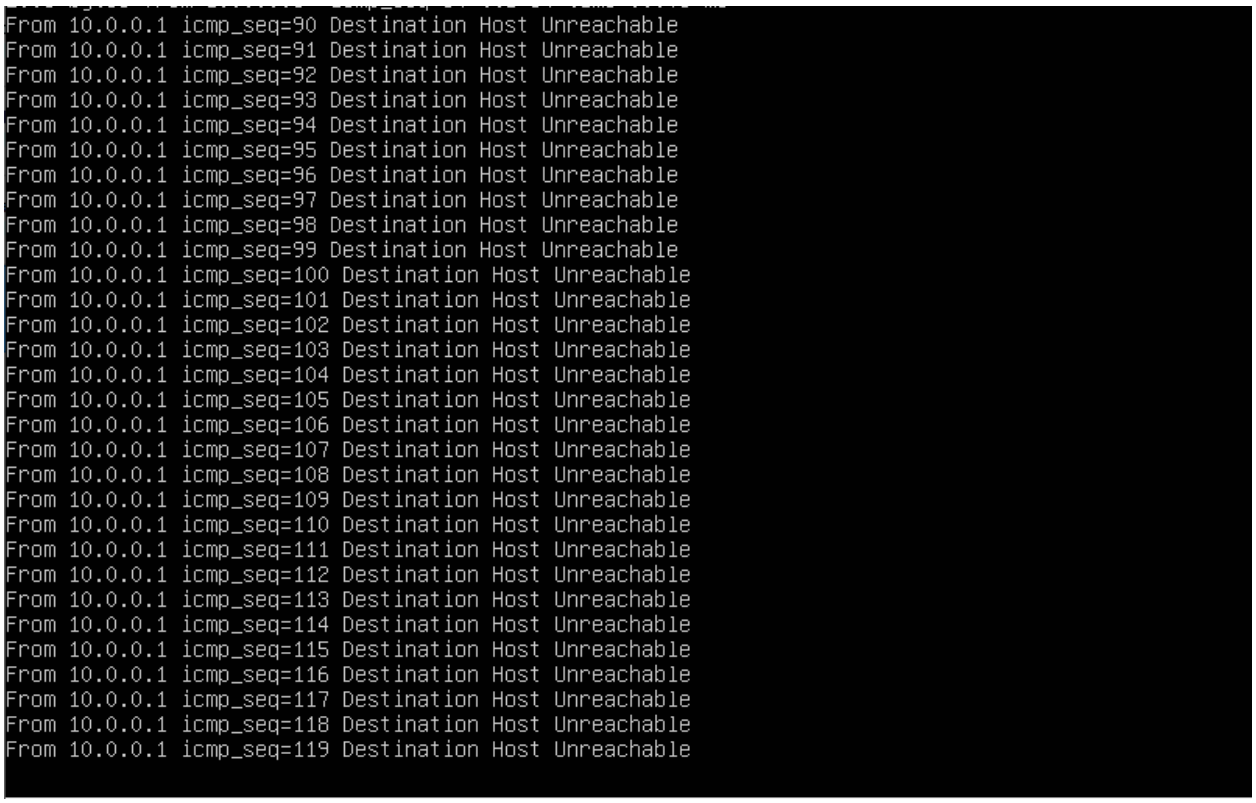
```
src=of:0000000000000101/2, dst=of:0000000000000102/2, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000101/3, dst=of:0000000000000201/2, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000101/4, dst=of:0000000000000103/3, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000101/5, dst=of:0000000000000301/4, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000102/2, dst=of:0000000000000101/2, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000102/3, dst=of:0000000000000103/2, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000102/4, dst=of:0000000000000202/2, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000102/5, dst=of:0000000000000302/5, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000103/2, dst=of:0000000000000102/3, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000103/3, dst=of:0000000000000101/4, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000103/4, dst=of:0000000000000203/2, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000103/5, dst=of:0000000000000303/5, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000201/2, dst=of:0000000000000101/3, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000201/3, dst=of:0000000000000202/3, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000201/4, dst=of:0000000000000301/2, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000201/5, dst=of:0000000000000203/4, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000202/2, dst=of:0000000000000102/4, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000202/3, dst=of:0000000000000201/3, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000202/4, dst=of:0000000000000203/3, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000202/5, dst=of:0000000000000302/2, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000203/2, dst=of:0000000000000103/4, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000203/3, dst=of:0000000000000202/4, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000203/4, dst=of:0000000000000201/5, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000203/5, dst=of:0000000000000303/2, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000301/2, dst=of:0000000000000201/4, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000301/3, dst=of:0000000000000302/3, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000301/4, dst=of:0000000000000101/5, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000301/5, dst=of:0000000000000303/4, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000302/2, dst=of:0000000000000202/5, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000302/3, dst=of:0000000000000301/3, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000302/4, dst=of:0000000000000303/3, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000302/5, dst=of:0000000000000102/5, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000303/2, dst=of:0000000000000203/5, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000303/3, dst=of:0000000000000302/4, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000303/4, dst=of:0000000000000301/5, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000303/5, dst=of:0000000000000103/5, type=DIRECT, state=ACTIVE, expected=false
karaf@root > _
```

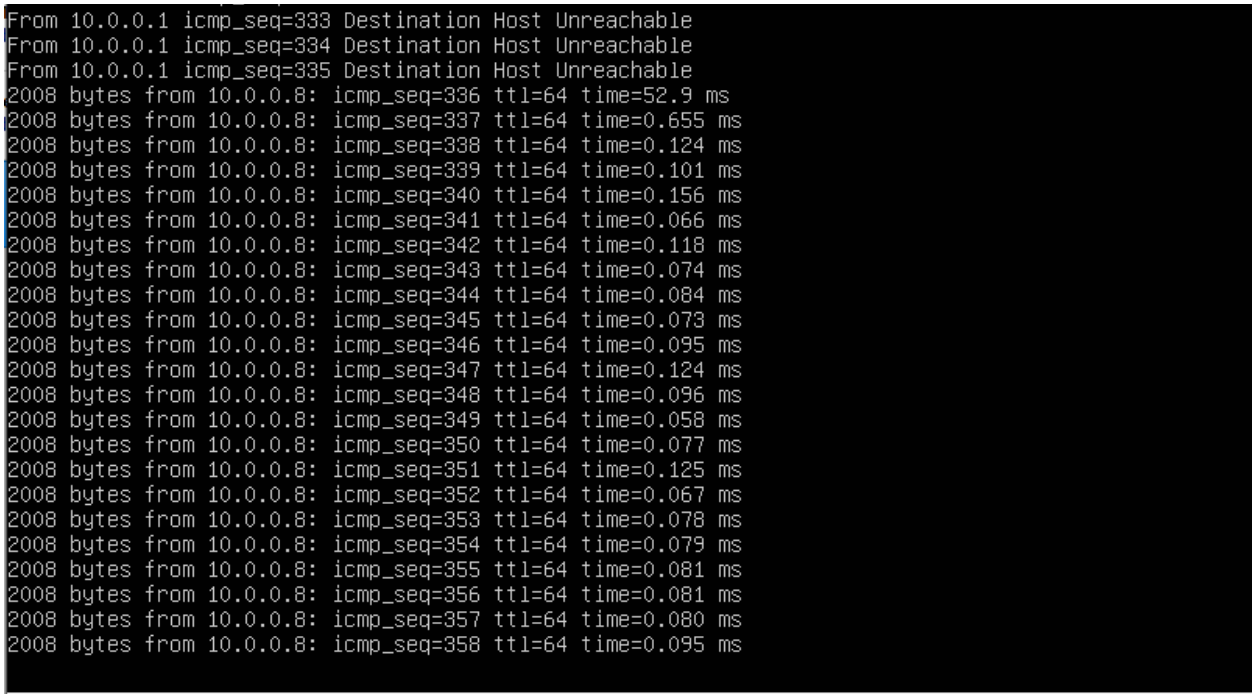
```
karaf@root >
karaf@root > flows_
```

```
t{immediate=[OUTPUT:CONTROLLER], deferred=[], transition=None, meter=[], cleared=true, StatTrigger=n
ull, metadata=null}
deviceId=of:0000000000000302, flowRuleCount=4
    id=10000201b6630, state=ADDED, bytes=7182, packets=171, duration=450, liveType=UNKNOWN, priority
=40000, tableId=0, appId=org.onosproject.core, selector=[ETH_TYPE:arp], treatment=DefaultTrafficTrea
tment{immediate=[OUTPUT:CONTROLLER], deferred=[], transition=None, meter=[], cleared=true, StatTrigg
er=null, metadata=null}
    id=100005615d045, state=ADDED, bytes=82062, packets=582, duration=450, liveType=UNKNOWN, priorit
y=40000, tableId=0, appId=org.onosproject.core, selector=[ETH_TYPE:bddp], treatment=DefaultTrafficTr
eatment{immediate=[OUTPUT:CONTROLLER], deferred=[], transition=None, meter=[], cleared=true, StatTri
gger=null, metadata=null}
    id=100006aee3185, state=ADDED, bytes=82062, packets=582, duration=450, liveType=UNKNOWN, priorit
y=40000, tableId=0, appId=org.onosproject.core, selector=[ETH_TYPE:lldp], treatment=DefaultTrafficTr
eatment{immediate=[OUTPUT:CONTROLLER], deferred=[], transition=None, meter=[], cleared=true, StatTri
gger=null, metadata=null}
    id=10000f1e9ed95, state=ADDED, bytes=2156, packets=22, duration=450, liveType=UNKNOWN, priority=
5, tableId=0, appId=org.onosproject.core, selector=[ETH_TYPE:ipv4], treatment=DefaultTrafficTreatmen
t{immediate=[OUTPUT:CONTROLLER], deferred=[], transition=None, meter=[], cleared=true, StatTrigger=n
ull, metadata=null}
deviceId=of:0000000000000303, flowRuleCount=4
    id=100002d7d00ba, state=ADDED, bytes=81780, packets=580, duration=450, liveType=UNKNOWN, priorit
y=40000, tableId=0, appId=org.onosproject.core, selector=[ETH_TYPE:bddp], treatment=DefaultTrafficTr
eatment{immediate=[OUTPUT:CONTROLLER], deferred=[], transition=None, meter=[], cleared=true, StatTri
gger=null, metadata=null}
    id=100007a562a38, state=ADDED, bytes=7182, packets=171, duration=450, liveType=UNKNOWN, priority
=40000, tableId=0, appId=org.onosproject.core, selector=[ETH_TYPE:arp], treatment=DefaultTrafficTrea
tment{immediate=[OUTPUT:CONTROLLER], deferred=[], transition=None, meter=[], cleared=true, StatTrigg
er=null, metadata=null}
    id=100008d5d7c74, state=ADDED, bytes=81780, packets=580, duration=450, liveType=UNKNOWN, priorit
y=40000, tableId=0, appId=org.onosproject.core, selector=[ETH_TYPE:lldp], treatment=DefaultTrafficTr
eatment{immediate=[OUTPUT:CONTROLLER], deferred=[], transition=None, meter=[], cleared=true, StatTri
gger=null, metadata=null}
    id=10000b0cb3683, state=ADDED, bytes=2352, packets=24, duration=450, liveType=UNKNOWN, priority=
5, tableId=0, appId=org.onosproject.core, selector=[ETH_TYPE:ipv4], treatment=DefaultTrafficTreatmen
t{immediate=[OUTPUT:CONTROLLER], deferred=[], transition=None, meter=[], cleared=true, StatTrigger=n
ull, metadata=null}
karaf@root > _
```

```
karaf@root > hosts
id=00:00:00:00:00:01/None, mac=00:00:00:00:00:01, locations=[of:0000000000000101/1], auxLocations=nu
ll, vlan=None, ip(s)=[10.0.0.1], innerVlan=None, outerTPID=unknown, provider=of:org.onosproject.prov
ider.host, configured=false
id=00:00:00:00:00:02/None, mac=00:00:00:00:00:02, locations=[of:0000000000000102/1], auxLocations=nu
ll, vlan=None, ip(s)=[10.0.0.2], innerVlan=None, outerTPID=unknown, provider=of:org.onosproject.prov
ider.host, configured=false
id=00:00:00:00:00:03/None, mac=00:00:00:00:00:03, locations=[of:0000000000000103/1], auxLocations=nu
ll, vlan=None, ip(s)=[10.0.0.3], innerVlan=None, outerTPID=unknown, provider=of:org.onosproject.prov
ider.host, configured=false
id=00:00:00:00:00:04/None, mac=00:00:00:00:00:04, locations=[of:0000000000000201/1], auxLocations=nu
ll, vlan=None, ip(s)=[10.0.0.4], innerVlan=None, outerTPID=unknown, provider=of:org.onosproject.prov
ider.host, configured=false
id=00:00:00:00:00:05/None, mac=00:00:00:00:00:05, locations=[of:0000000000000202/1], auxLocations=nu
ll, vlan=None, ip(s)=[10.0.0.5], innerVlan=None, outerTPID=unknown, provider=of:org.onosproject.prov
ider.host, configured=false
id=00:00:00:00:00:06/None, mac=00:00:00:00:00:06, locations=[of:0000000000000203/1], auxLocations=nu
ll, vlan=None, ip(s)=[10.0.0.6], innerVlan=None, outerTPID=unknown, provider=of:org.onosproject.prov
ider.host, configured=false
id=00:00:00:00:00:07/None, mac=00:00:00:00:00:07, locations=[of:0000000000000301/1], auxLocations=nu
ll, vlan=None, ip(s)=[10.0.0.7], innerVlan=None, outerTPID=unknown, provider=of:org.onosproject.prov
ider.host, configured=false
id=00:00:00:00:00:08/None, mac=00:00:00:00:00:08, locations=[of:0000000000000302/1], auxLocations=nu
ll, vlan=None, ip(s)=[10.0.0.8], innerVlan=None, outerTPID=unknown, provider=of:org.onosproject.prov
ider.host, configured=false
id=00:00:00:00:00:09/None, mac=00:00:00:00:00:09, locations=[of:0000000000000303/1], auxLocations=nu
ll, vlan=None, ip(s)=[10.0.0.9], innerVlan=None, outerTPID=unknown, provider=of:org.onosproject.prov
ider.host, configured=false
karaf@root > _
```

- **Perform testing via Mininet CLI & ONOS CLI**
    - Perform ping command in mininet cli:
        - **mininet> h1x1 ping -s 2000 h3x2**
        - observe the responses
    - At ONOS CLI deactivate the fwd:
        - **karaf@root> app deactivate fwd**
        - Observe response
    - At ONOS CLI activate the fwd app
        - **karaf@root> app activate fwd**
        - Observe the changes in response
    - See figures below:



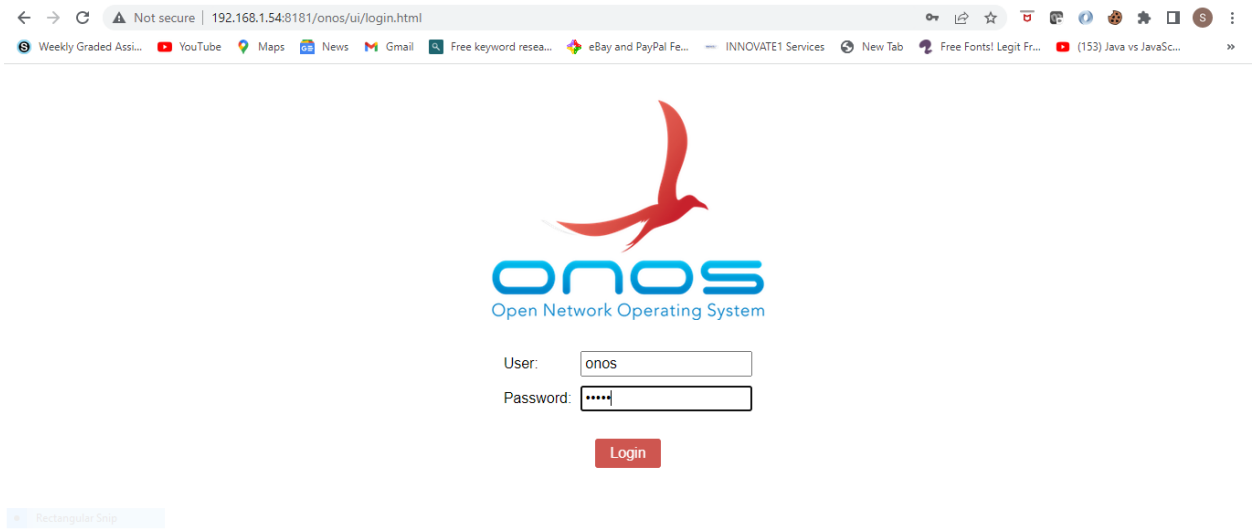


- **Admin Operations**
  - Perform current configuration backup
  - Saves as backup TAR file at **/opt/onos-data.tar.gz**
    - **$ sudo /opt/onos/bin/onos-backup**
  - Restore a backup file to current configuration
    - **$ sudo /opt/onos/bin/onos-restore /opt/onos-data.tar.gz**
  - See figures below:

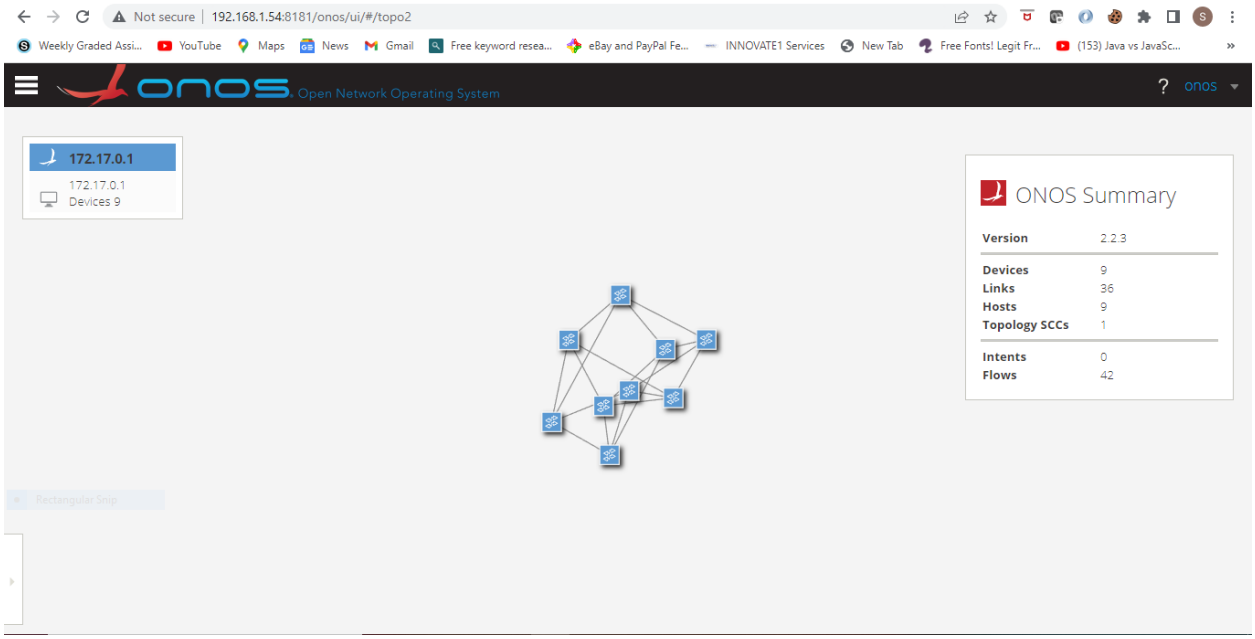
```
File  Machine  View  Input  Devices  Help
ks-sdn@sdn_server:~$ sudo /opt/onos/bin/onos-backup
[sudo] password for ks-sdn:
tar: data/db/partitions/data/partitions/1/raft-partition-1-1.log: file changed as we read it
tar: data/db/partitions/system/partitions/1/system-partition-1-1.log: file changed as we read it
ks-sdn@sdn_server:~$ ls /opt/
containerd  onos  onos-2.2.3  onos-data.tar.gz
ks-sdn@sdn_server:~$ sudo /opt/onos/bin/onos-restore /opt/onos-data.tar.gz
ks-sdn@sdn_server:~$ ls /opt/
containerd  onos  onos-2.2.3  onos-data.tar.gz
ks-sdn@sdn_server:~$
```

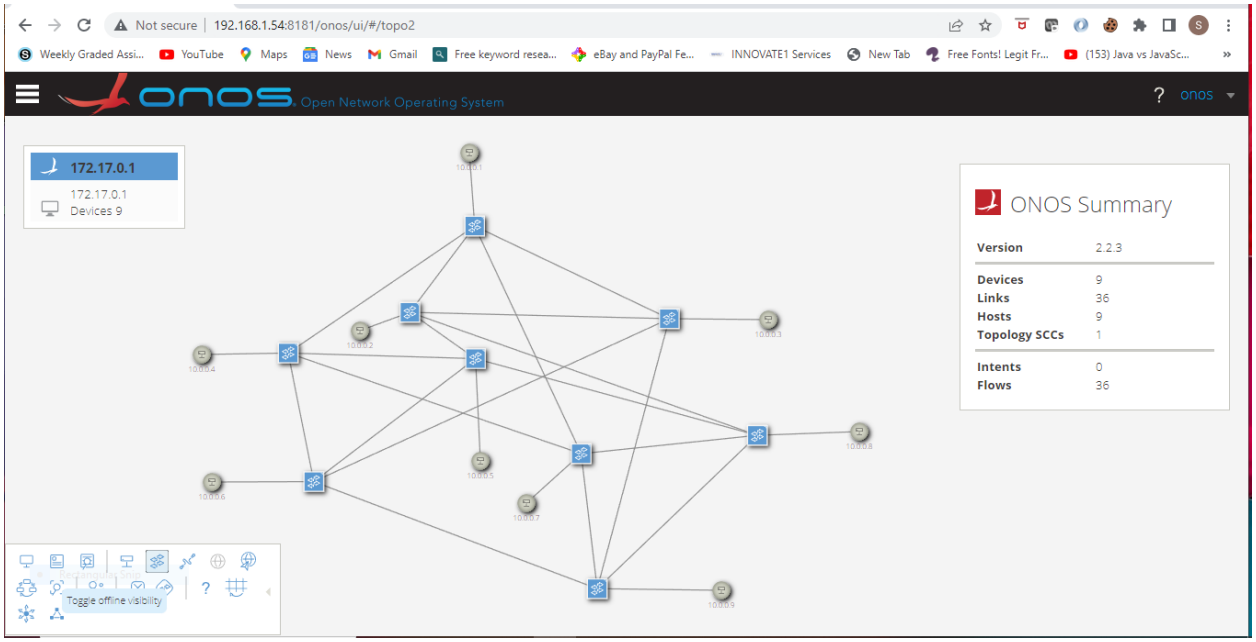
Getting started with ONOS GUI

- **Accessing the GUI**
  - In a web browser, open the ONOS GUI by typing the below url in the address bar
  - **http://<IP-address-onos-server>:8181/onos/ui/login.html**
  - Note: my **<IP-address-onos-server>** is **192.168.1.54, user= onos, password= rocks**
  - See figure below:

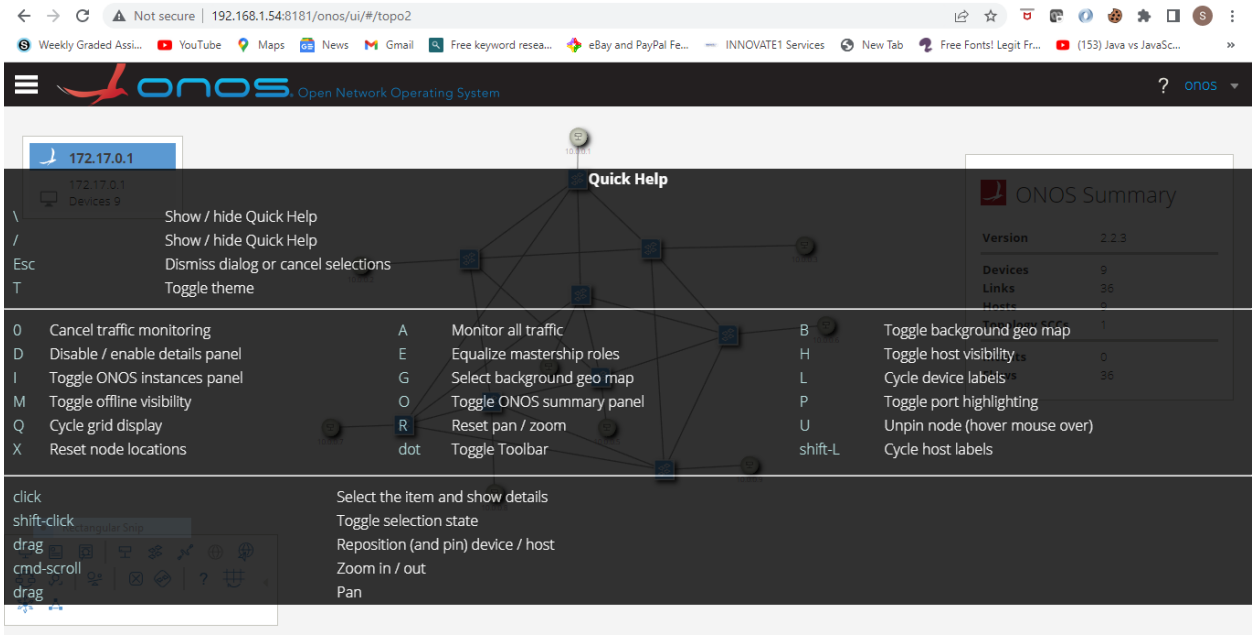


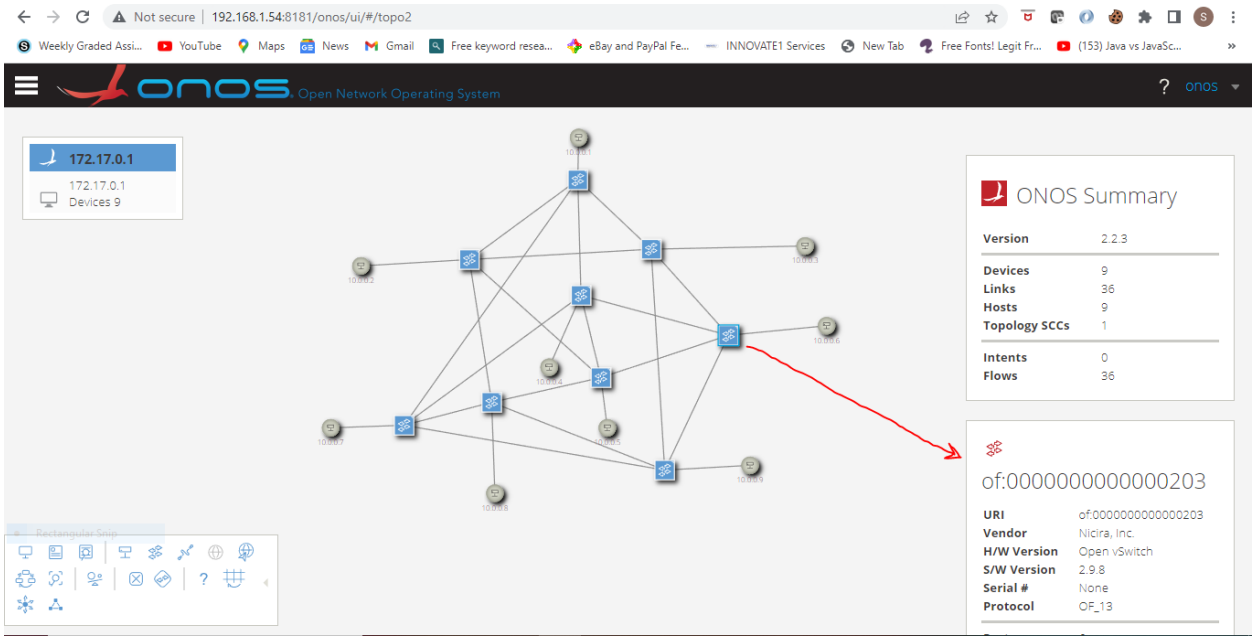
- **Knowing ONOS GUI**

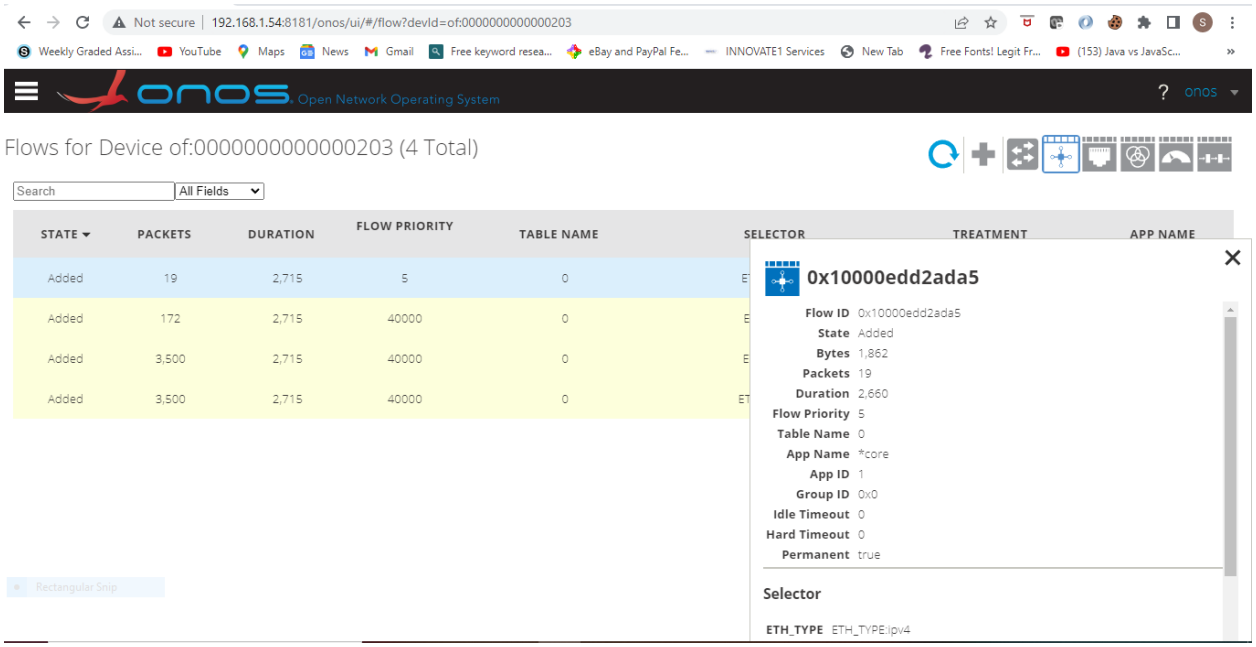


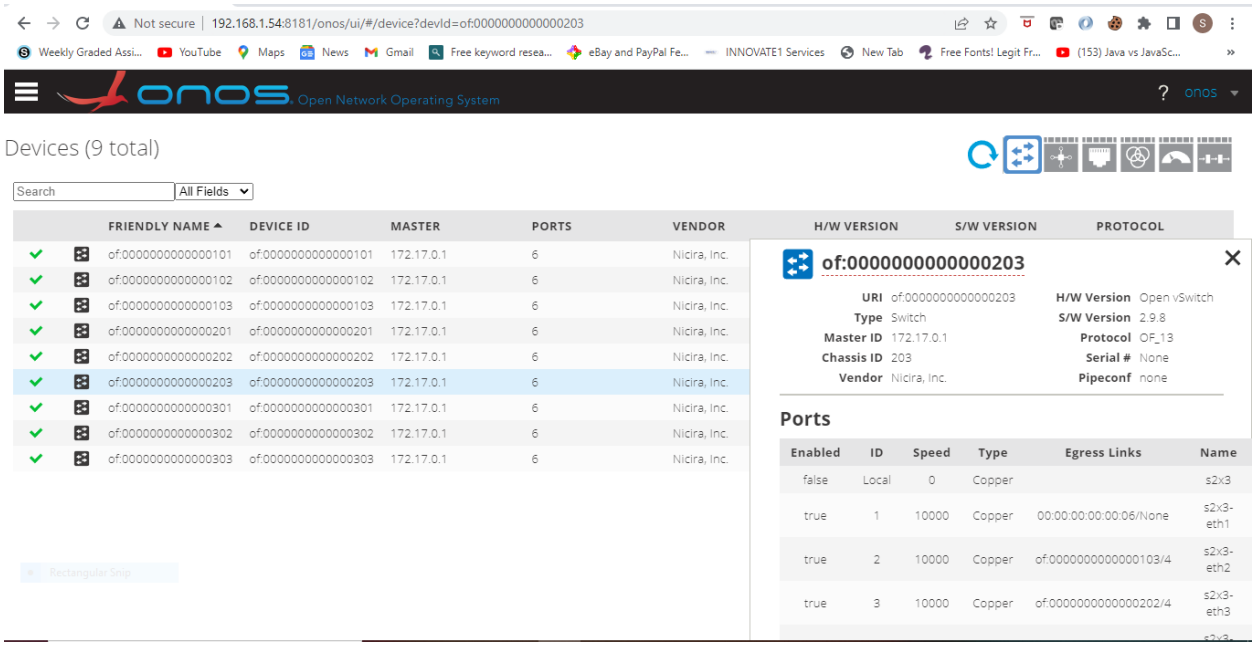


- **Get some details from GUI**



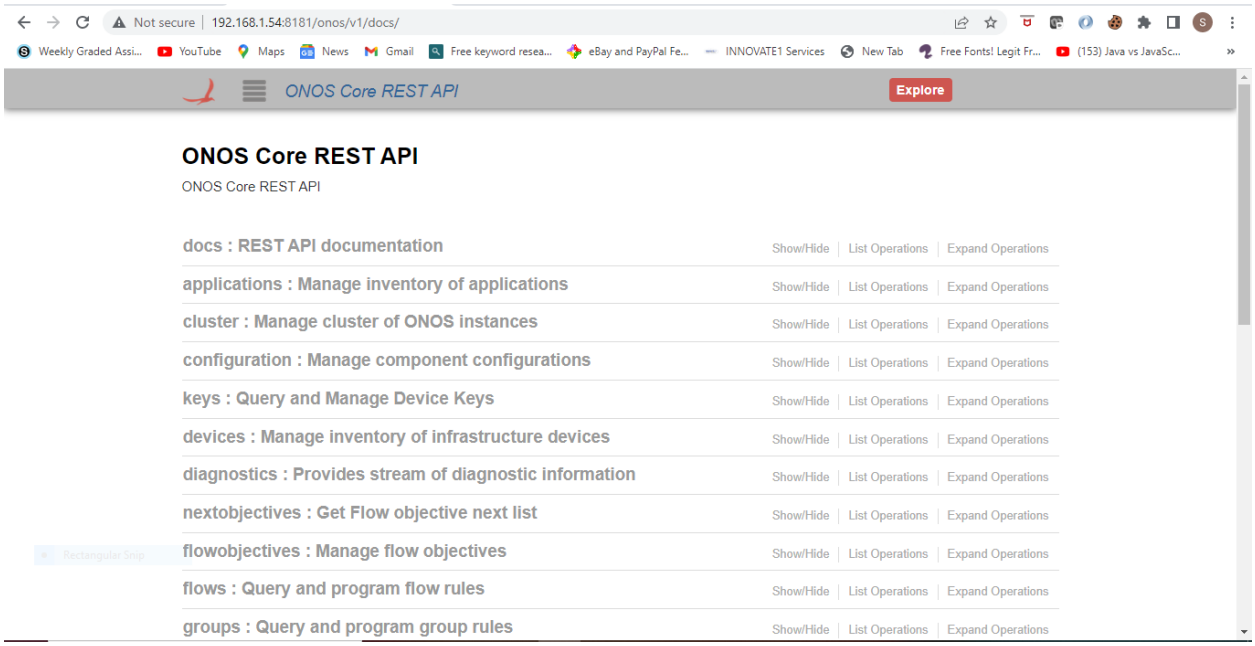

Not secure | 192.168.1.54:8181/onos/ui/#/topo2
Open Network Operating System
onos
172.17.0.1
172.17.0.1
Devices 9
ONOS Summary
Version 2.2.3
Devices 9
Links 36
Hosts 9
Topology SCCs 1
Intents 0
Flows 36
of:0000000000000203
URI of:0000000000000203
Vendor Nicira, Inc.
H/W Version Open vSwitch
S/W Version 2.9.8
Serial # None
Protocol OF_13


Not secure | 192.168.1.54:8181/onos/ui/#/flow?devId=of:0000000000000203
Open Network Operating System
onos
Flows for Device of:0000000000000203 (4 Total)
Search
All Fields
STATE
PACKETS
DURATION
FLOW PRIORITY
TABLE NAME
SELECTOR
TREATMENT
APP NAME
Added 19 2,715 5 0
Added 172 2,715 40000 0
Added 3,500 2,715 40000 0
Added 3,500 2,715 40000 0
0x10000edd2ada5
Flow ID 0x10000edd2ada5
State Added
Bytes 1,862
Packets 19
Duration 2,660
Flow Priority 5
Table Name 0
App Name *core
App ID 1
Group ID 0x0
Idle Timeout 0
Hard Timeout 0
Permanent true
Selector
ETH_TYPE ETH_TYPE:Ipv4

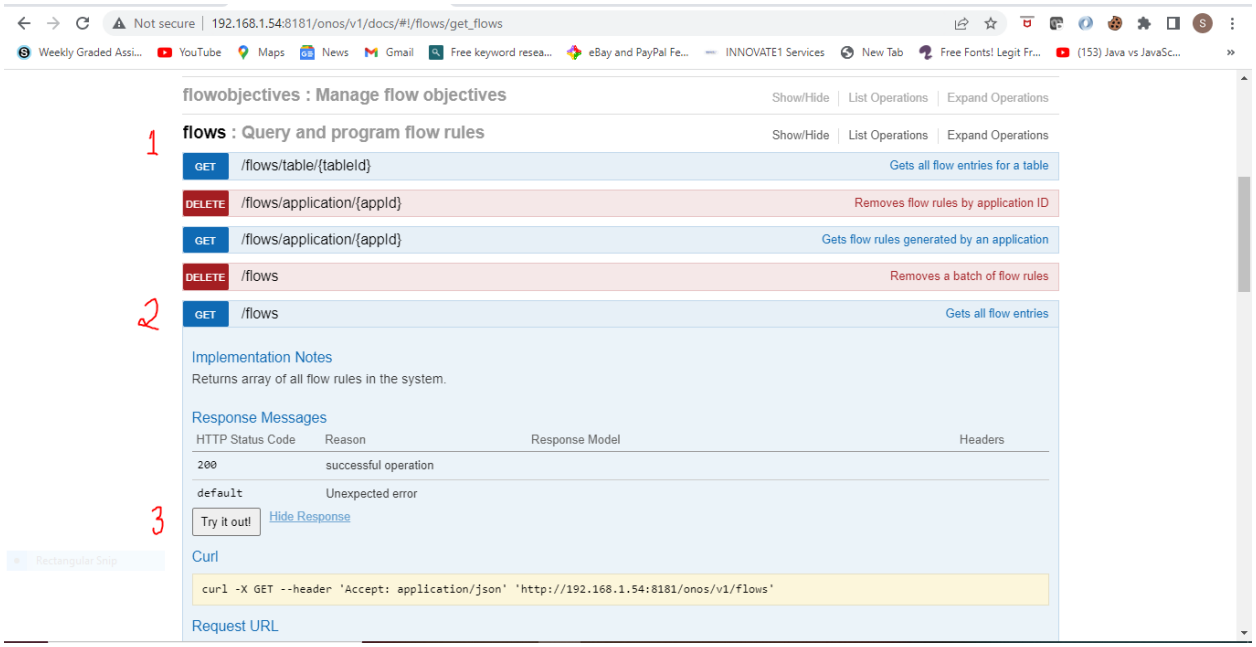


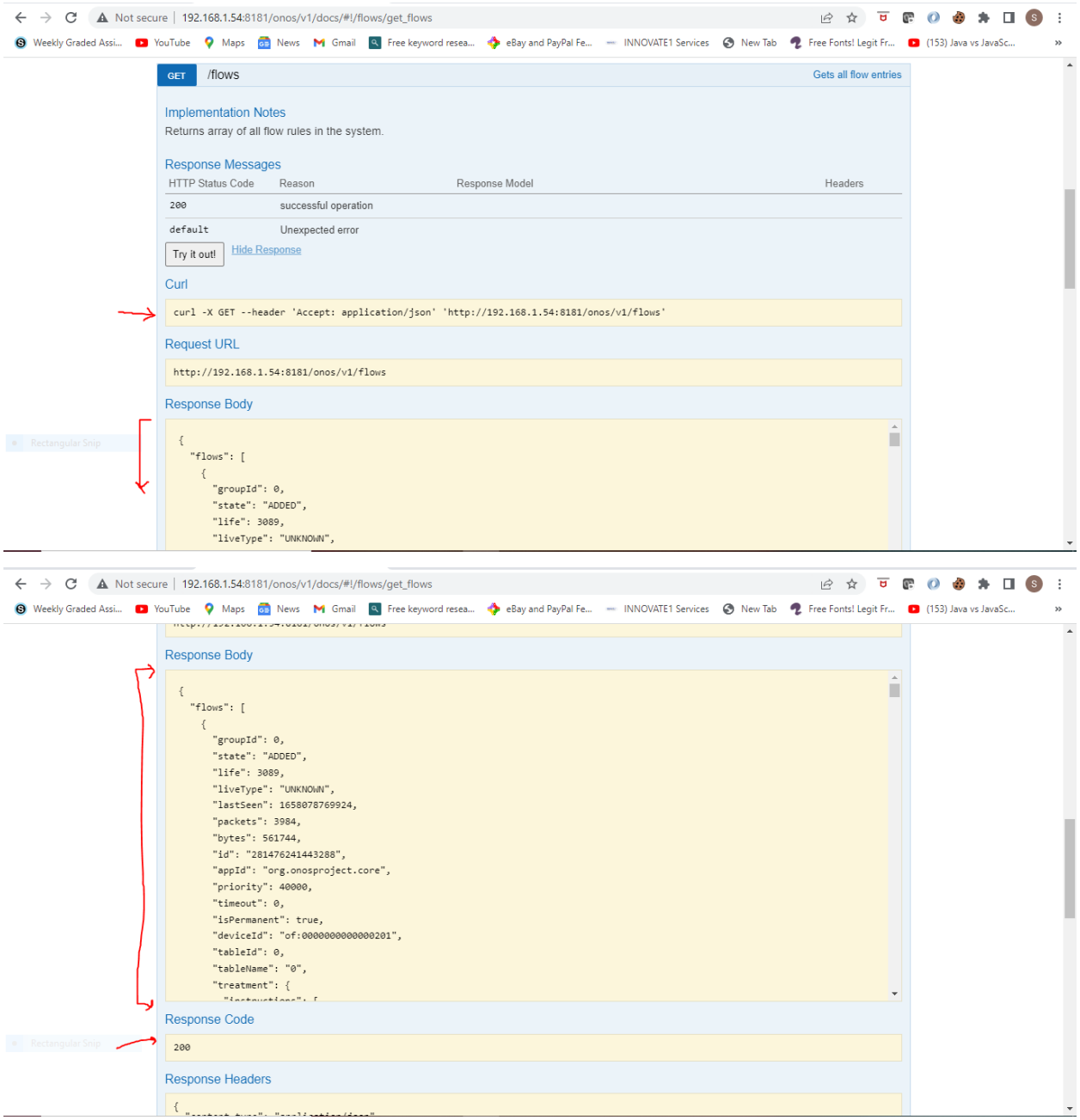
ONOS RESTCONF

- **ONOS API Documentation**
  - In a web browser, open the ONOS API Doc GUI by typing the below url in the address bar
  - **http://<IP-address-onos-server>:8181/onos/v1/docs/**
  - Note: my **<IP-address-onos-server>** is **192.168.1.54, user= onos, password= rocks**
  - See figure below:



- **Know the docs**
  - Click on **flows** to expand
  - Click on **GET/flows**
  - Click on button **Try it out!**
  - See figures below:





- **GETting info via REST API**
  - Test REST API
    - **$ curl -u onos:rocks http://localhost:8181/onos/v1/**
  - Get Devices
    - **$ curl -su onos:rocks http://localhost:8181/onos/v1/devices**
  - Get ONOS configuration

- **$ curl -X GET –header ‘Accept: application/json’ ‘http://localhost:8181/onos/v1/configuration’**

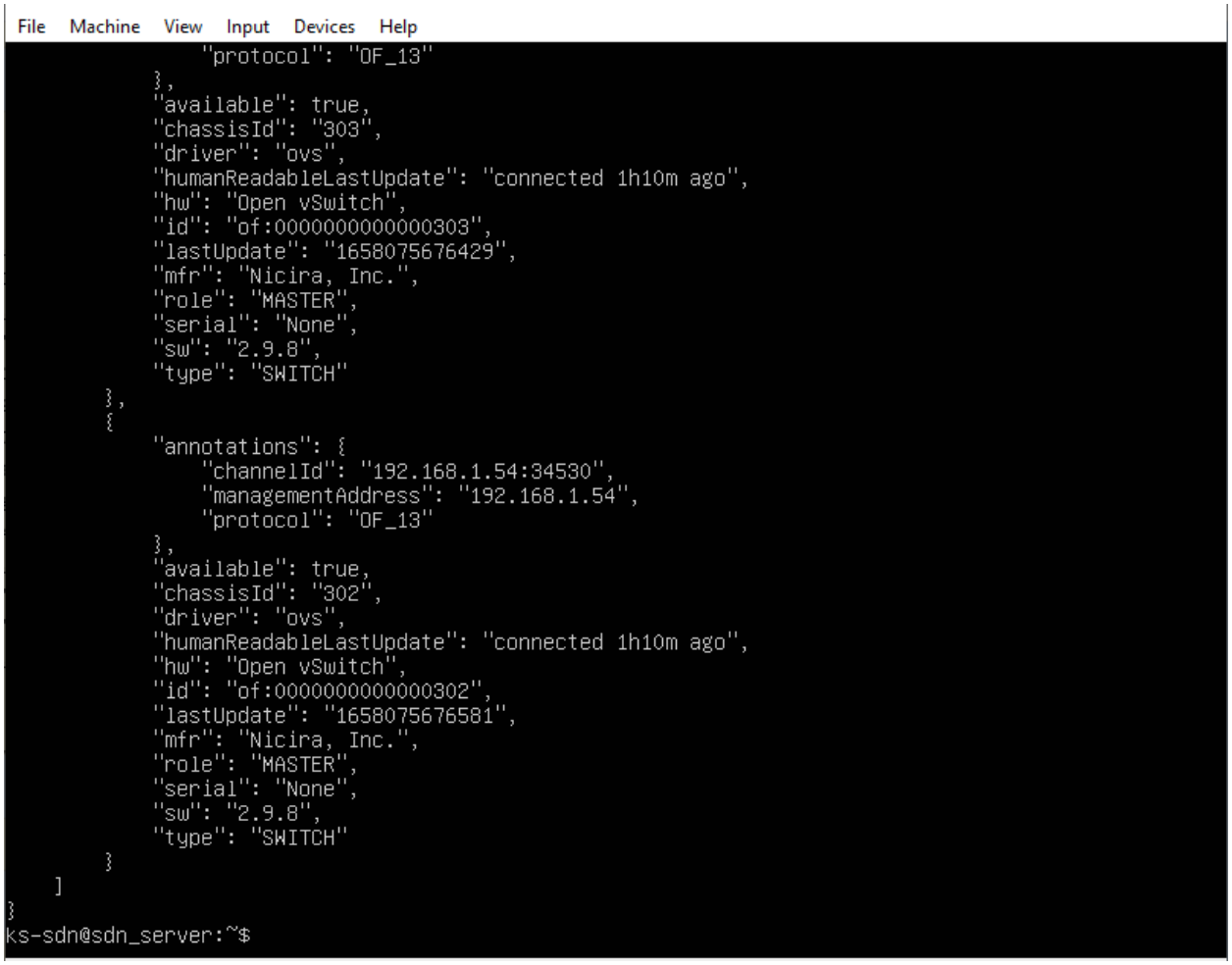
- Get Devices and print it in JSON format
  - **$ curl -su onos:rocks http://localhost:8181/onos/v1/devices | python -m json.tool**
- See figures below:

File Machine View Input Devices Help

```
ks-sdn@sdn_server:~$ curl -u onos:rocks http://localhost:8181/onos/v1/
{"code":404,"message":"HTTP 404 Not Found"}ks-sdn@sdn_server:~$ curl -u onos:rocks http://localhost:

ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/devices
{"devices":[{"id":"of:0000000000000201","type":"SWITCH","available":true,"role":"MASTER","mfr":"Nici
ra, Inc.","hw":"Open vSwitch","sw":"2.9.8","serial":"None","driver":"ovs","chassisId":"201","lastUpd
ate":"1658075673389","humanReadableLastUpdate":"connected 1h4m ago","annotations":{"channelId":"192.
168.1.54:34516","managementAddress":"192.168.1.54","protocol":"OF_13"}},{"id":"of:0000000000000301",
"type":"SWITCH","available":true,"role":"MASTER","mfr":"Nicira, Inc.","hw":"Open vSwitch","sw":"2.9.
8","serial":"None","driver":"ovs","chassisId":"301","lastUpdate":"1658075674292","humanReadableLastU
pdate":"connected 1h4m ago","annotations":{"channelId":"192.168.1.54:34518","managementAddress":"192
.168.1.54","protocol":"OF_13"}},{"id":"of:0000000000000102","type":"SWITCH","available":true,"role":
"MASTER","mfr":"Nicira, Inc.","hw":"Open vSwitch","sw":"2.9.8","serial":"None","driver":"ovs","chass
isId":"102","lastUpdate":"1658075676731","humanReadableLastUpdate":"connected 1h4m ago","annotations
":{"channelId":"192.168.1.54:34522","managementAddress":"192.168.1.54","protocol":"OF_13"}},{"id":"o
f:0000000000000202","type":"SWITCH","available":true,"role":"MASTER","mfr":"Nicira, Inc.","hw":"Open
 vSwitch","sw":"2.9.8","serial":"None","driver":"ovs","chassisId":"202","lastUpdate":"1658075673410"
,"humanReadableLastUpdate":"connected 1h4m ago","annotations":{"channelId":"192.168.1.54:34514","man
agementAddress":"192.168.1.54","protocol":"OF_13"}},{"id":"of:0000000000000103","type":"SWITCH","ava
ilable":true,"role":"MASTER","mfr":"Nicira, Inc.","hw":"Open vSwitch","sw":"2.9.8","serial":"None","
driver":"ovs","chassisId":"103","lastUpdate":"1658075679520","humanReadableLastUpdate":"connected 1h
4m ago","annotations":{"channelId":"192.168.1.54:34528","managementAddress":"192.168.1.54","protocol
":"OF_13"}},{"id":"of:0000000000000101","type":"SWITCH","available":true,"role":"MASTER","mfr":"Nici
ra, Inc.","hw":"Open vSwitch","sw":"2.9.8","serial":"None","driver":"ovs","chassisId":"101","lastUpd
ate":"1658075674292","humanReadableLastUpdate":"connected 1h4m ago","annotations":{"channelId":"192.
168.1.54:34520","managementAddress":"192.168.1.54","protocol":"OF_13"}},{"id":"of:0000000000000203",
"type":"SWITCH","available":true,"role":"MASTER","mfr":"Nicira, Inc.","hw":"Open vSwitch","sw":"2.9.
8","serial":"None","driver":"ovs","chassisId":"203","lastUpdate":"1658075674501","humanReadableLastU
pdate":"connected 1h4m ago","annotations":{"channelId":"192.168.1.54:34524","managementAddress":"192
.168.1.54","protocol":"OF_13"}},{"id":"of:0000000000000303","type":"SWITCH","available":true,"role":
"MASTER","mfr":"Nicira, Inc.","hw":"Open vSwitch","sw":"2.9.8","serial":"None","driver":"ovs","chass
isId":"303","lastUpdate":"1658075676429","humanReadableLastUpdate":"connected 1h4m ago","annotations
":{"channelId":"192.168.1.54:34526","managementAddress":"192.168.1.54","protocol":"OF_13"}},{"id":"o
f:0000000000000302","type":"SWITCH","available":true,"role":"MASTER","mfr":"Nicira, Inc.","hw":"Open
 vSwitch","sw":"2.9.8","serial":"None","driver":"ovs","chassisId":"302","lastUpdate":"1658075676581"
,"humanReadableLastUpdate":"connected 1h4m ago","annotations":{"channelId":"192.168.1.54:34530","man
agementAddress":"192.168.1.54","protocol":"OF_13"}}]}ks-sdn@sdn_server:~$
```

```
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/devices | python -m json.tool
_
```



- **Get a specific Device**
    - See figures below:

```
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/devices/of:0000000000000101 |
 python -m json.tool
{
    "annotations": {
        "channelId": "192.168.1.54:34520",
        "managementAddress": "192.168.1.54",
        "protocol": "OF_13"
    },
    "available": true,
    "chassisId": "101",
    "driver": "ovs",
    "humanReadableLastUpdate": "connected 1h14m ago",
    "hw": "Open vSwitch",
    "id": "of:0000000000000101",
    "lastUpdate": "1658075674292",
    "mfr": "Nicira, Inc.",
    "role": "MASTER",
    "serial": "None",
    "sw": "2.9.8",
    "type": "SWITCH"
}
ks-sdn@sdn_server:~$ _
```

```
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/devices/of:0000000000000102 |
 python -m json.tool
{
    "annotations": {
        "channelId": "192.168.1.54:34522",
        "managementAddress": "192.168.1.54",
        "protocol": "OF_13"
    },
    "available": true,
    "chassisId": "102",
    "driver": "ovs",
    "humanReadableLastUpdate": "connected 1h15m ago",
    "hw": "Open vSwitch",
    "id": "of:0000000000000102",
    "lastUpdate": "1658075676731",
    "mfr": "Nicira, Inc.",
    "role": "MASTER",
    "serial": "None",
    "sw": "2.9.8",
    "type": "SWITCH"
}
ks-sdn@sdn_server:~$ _
```

- **Get Device Ports**
  - **$ curl -su onos:rocks http://localhost:8181/onos/v1/devices/of:0000000000000102/ports | python -m json.tool**
  - See figures below:

```
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/devices/of:0000000000000102/p
orts | python -m json.tool_
```

```
            "element": "of:0000000000000102",
            "isEnabled": true,
            "port": "3",
            "portSpeed": 10000,
            "type": "copper"
        },
        {
            "annotations": {
                "adminState": "enabled",
                "portMac": "3a:35:74:f3:c8:e7",
                "portName": "s1x2-eth4"
            },
            "element": "of:0000000000000102",
            "isEnabled": true,
            "port": "4",
            "portSpeed": 10000,
            "type": "copper"
        },
        {
            "annotations": {
                "adminState": "enabled",
                "portMac": "da:0e:de:3c:60:02",
                "portName": "s1x2-eth5"
            },
            "element": "of:0000000000000102",
            "isEnabled": true,
            "port": "5",
            "portSpeed": 10000,
            "type": "copper"
        }
    ],
    "role": "MASTER",
    "serial": "None",
    "sw": "2.9.8",
    "type": "SWITCH"
}
ks-sdn@sdn_server:~$ _
```


- **Get Links**
  - **$ curl -su onos:rocks http://localhost:8181/onos/v1/links | python -m json.tool**
  - See figures below:

```
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/links | python -m json.tool
```

```
                "device": "of:0000000000000202",
                "port": "3"
            },
            "src": {
                "device": "of:0000000000000201",
                "port": "3"
            },
            "state": "ACTIVE",
            "type": "DIRECT"
        },
        {
            "dst": {
                "device": "of:0000000000000201",
                "port": "5"
            },
            "src": {
                "device": "of:0000000000000203",
                "port": "4"
            },
            "state": "ACTIVE",
            "type": "DIRECT"
        },
        {
            "dst": {
                "device": "of:0000000000000102",
                "port": "4"
            },
            "src": {
                "device": "of:0000000000000202",
                "port": "2"
            },
            "state": "ACTIVE",
            "type": "DIRECT"
        }
    ]
}
ks-sdn@sdn_server:~$
```


- **Get Hosts**
  - **$ curl -su onos:rocks http://localhost:8181/onos/v1/hosts | python -m json.tool**
  - See figures below:

```
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/hosts | python -m json.tool_
```

```
            "id": "00:00:00:00:00:06/None",
            "innerVlan": "None",
            "ipAddresses": [
                "10.0.0.6"
            ],
            "locations": [
                {
                    "elementId": "of:0000000000000203",
                    "port": "1"
                }
            ],
            "mac": "00:00:00:00:00:06",
            "outerTpid": "0x0000",
            "suspended": false,
            "vlan": "None"
        },
        {
            "configured": false,
            "id": "00:00:00:00:00:05/None",
            "innerVlan": "None",
            "ipAddresses": [
                "10.0.0.5"
            ],
            "locations": [
                {
                    "elementId": "of:0000000000000202",
                    "port": "1"
                }
            ],
            "mac": "00:00:00:00:00:05",
            "outerTpid": "0x0000",
            "suspended": false,
            "vlan": "None"
        }
    ]
}
ks-sdn@sdn_server:~$ _
```

```
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/hosts | python -m json.tool |
egrep -w "id"
            "id": "00:00:00:00:00:03/None",
            "id": "00:00:00:00:00:04/None",
            "id": "00:00:00:00:00:01/None",
            "id": "00:00:00:00:00:02/None",
            "id": "00:00:00:00:00:08/None",
            "id": "00:00:00:00:00:09/None",
            "id": "00:00:00:00:00:07/None",
            "id": "00:00:00:00:00:06/None",
            "id": "00:00:00:00:00:05/None",
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/hosts | python -m json.tool |
egrep -w "id|ipAddresses"
            "id": "00:00:00:00:00:03/None",
            "ipAddresses": [
            "id": "00:00:00:00:00:04/None",
            "ipAddresses": [
            "id": "00:00:00:00:00:01/None",
            "ipAddresses": [
            "id": "00:00:00:00:00:02/None",
            "ipAddresses": [
            "id": "00:00:00:00:00:08/None",
            "ipAddresses": [
            "id": "00:00:00:00:00:09/None",
            "ipAddresses": [
            "id": "00:00:00:00:00:07/None",
            "ipAddresses": [
            "id": "00:00:00:00:00:06/None",
            "ipAddresses": [
            "id": "00:00:00:00:00:05/None",
            "ipAddresses": [
ks-sdn@sdn_server:~$ _
```

```
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/hosts | python -m json.tool |
egrep -v -w "innerVlan|ipAddresses" | egrep -A2 -w "id"_
```

```
egrep -v -w "innerVlan|ipAddresses" | egrep -A2 -w "id"
            "id": "00:00:00:00:00:03/None",
                "10.0.0.3"
            ],
--
            "id": "00:00:00:00:00:04/None",
                "10.0.0.4"
            ],
--
            "id": "00:00:00:00:00:01/None",
                "10.0.0.1"
            ],
--
            "id": "00:00:00:00:00:02/None",
                "10.0.0.2"
            ],
--
            "id": "00:00:00:00:00:08/None",
                "10.0.0.8"
            ],
--
            "id": "00:00:00:00:00:09/None",
                "10.0.0.9"
            ],
--
            "id": "00:00:00:00:00:07/None",
                "10.0.0.7"
            ],
--
            "id": "00:00:00:00:00:06/None",
                "10.0.0.6"
            ],
--
            "id": "00:00:00:00:00:05/None",
                "10.0.0.5"
            ],
ks-sdn@sdn_server:~$
```

POSTing via REST API

- **Verifying YANG-GUI feature via ONOS CLI**
  - **karaf@root> feature:list | grep yang**
  - See figure below:

```
karaf@root > feature:list | grep yang
onos-apps-yang                  | 2.2.3           |          | Uninstalled | org.onosproject-
onos-apps-yang                  | YANG Compiler and Runtime
onos-apps-yang-gui              | 2.2.3           |          | Uninstalled | org.onosproject-
onos-apps-yang-gui              | YANG Runtime GUI
karaf@root >
```

- **Varying YANG-GUI feature via REST API - GET**
  - **$ curl -su onos:rocks http://localhost:8181/onos/v1/applications | python -m**

json.tool | grep -A7 -w “‘name”: “org.onosproject.yang-gui”’ | egrep -w “name|state”

- See figure below:

- **Activate YANG-GUI feature via REST API - POST**
  - **$ curl -su onos:rocks -X POST http://localhost:8181/onos/v1/applications/org.onosproject.yang-gui/activate | python -m json.tool**
  - Check the current state feature again, its **active:**
    - **$ curl -su onos:rocks http://localhost:8181/onos/v1/applications | python -m json.tool | grep -A7 -w “‘name”: “org.onosproject.yang-gui”’ | egrep -w “name|state”**
  - See figures below:

```
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/applications | python -m json
.tool | grep -A7 -w '"name": "org.onosproject.yang-gui"' | egrep -w "name|state"
            "name": "org.onosproject.yang-gui",
            "state": "INSTALLED",
ks-sdn@sdn_server:~$
ks-sdn@sdn_server:~$ curl -su onos:rocks -X POST http://localhost:8181/onos/v1/applications/org.onos
project.yang-gui/active | python -m json.tool
{
    "category": "GUI",
    "description": "Adds GUI extension that allows the operator to view the list of currently regist
ered YANG models, and their YANG sources.",
    "features": [
        "onos-apps-yang-gui"
    ],
    "featuresRepo": "mvn:org.onosproject/onos-apps-yang-gui/2.2.3/xml/features",
    "id": 82,
    "name": "org.onosproject.yang-gui",
    "origin": "ONOS Community",
    "permissions": [],
    "readme": "Adds GUI extension that allows the operator to view the list of currently registered
YANG models, and their YANG sources. It also provides means for compiling source YANG files (*.yang,
 *.zip, *.jar) on the fly and directly from the GUI using drag-n-drop.User can simply compress a set
 of YANG source files and drop it on the YANG Models GUI view to automatically compile and then regi
ster the compiled YANG models.",
    "requiredApps": [
        "org.onosproject.yang"
    ],
    "state": "ACTIVE",
    "url": "http://onosproject.org",
    "version": "2.2.3"
}
ks-sdn@sdn_server:~$ _
```

```
ks-sdn@sdn_server:~$ curl -su onos:rocks http://localhost:8181/onos/v1/applications | python -m json
.tool | grep -A7 -w '"name": "org.onosproject.yang-gui"' | egrep -w "name|state"
            "name": "org.onosproject.yang-gui",
            "state": "ACTIVE",
ks-sdn@sdn_server:~$
ks-sdn@sdn_server:~$ curl -su onos:rocks -X DELETE http://localhost:8181/onos/v1/applications/org.on
osproject.yang-gui/active | python -m json.tool
No JSON object could be decoded
ks-sdn@sdn_server:~$ _
```

QED