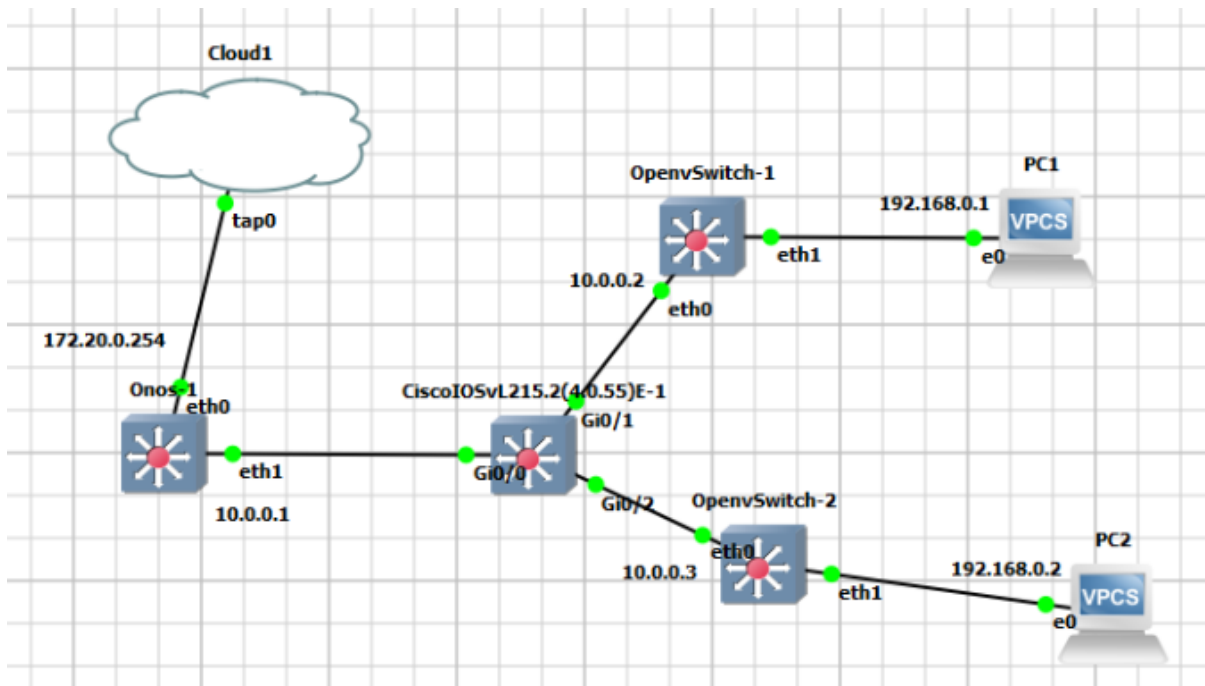


## SDN-LAB

In this exercise, we are going to use a sample app called *Reactive Forwarding*. It is shipped with ONOS and is a simple application that installs flows in response to every *miss* packet in that arrives at the controller.

**Step 1:** Create the Topology as per below



**Step 2:** Configure the ONOS Controller eth0 and eth1 IP address

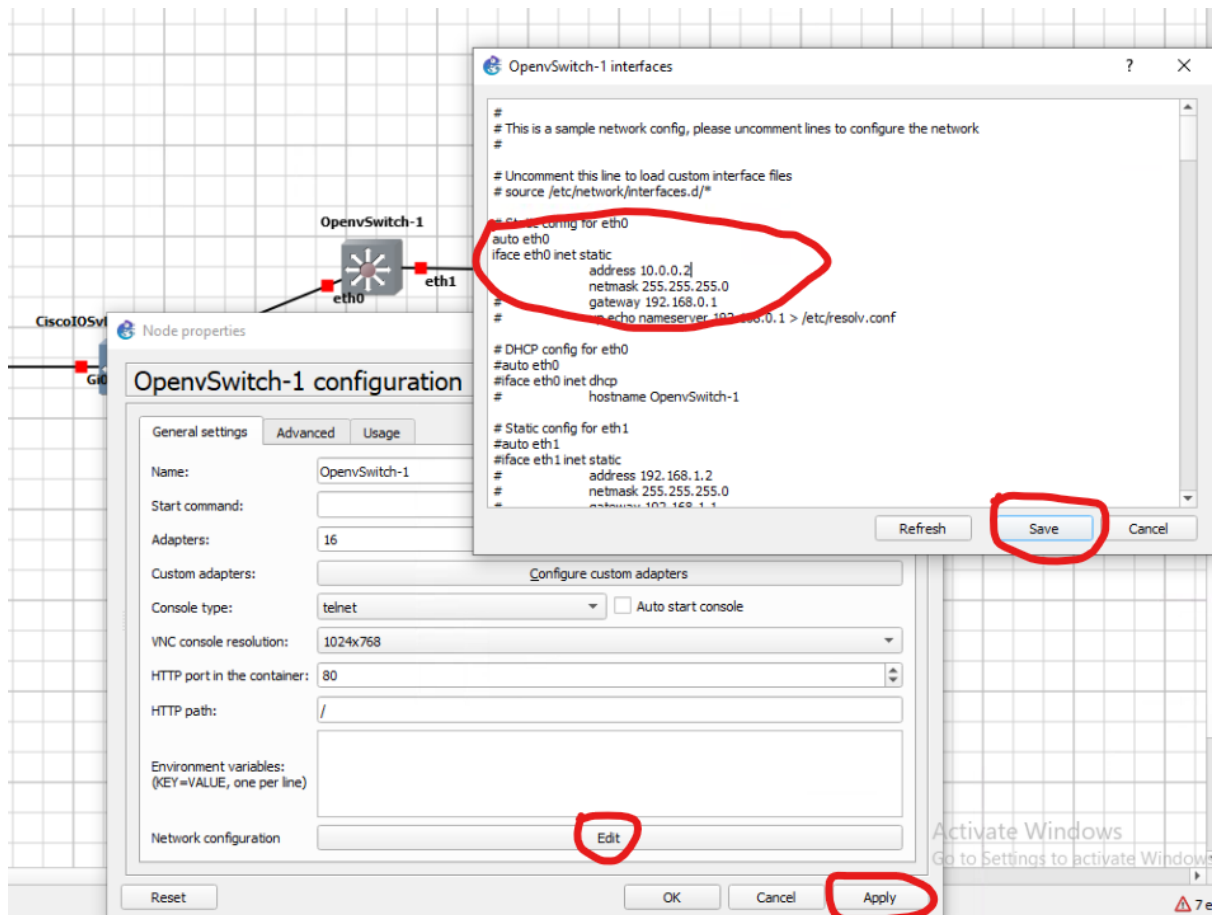
```
# This is a sample network config, please uncomment lines to configure the network
#
# Uncomment this line to load custom interface files
# source /etc/network/interfaces.d/*

# Static config for eth0
auto eth0
iface eth0 inet static
    address 172.20.0.254
    netmask 255.255.255.0
    gateway 172.20.0.1
    up echo nameserver 8.8.8.8 > /etc/resolv.conf

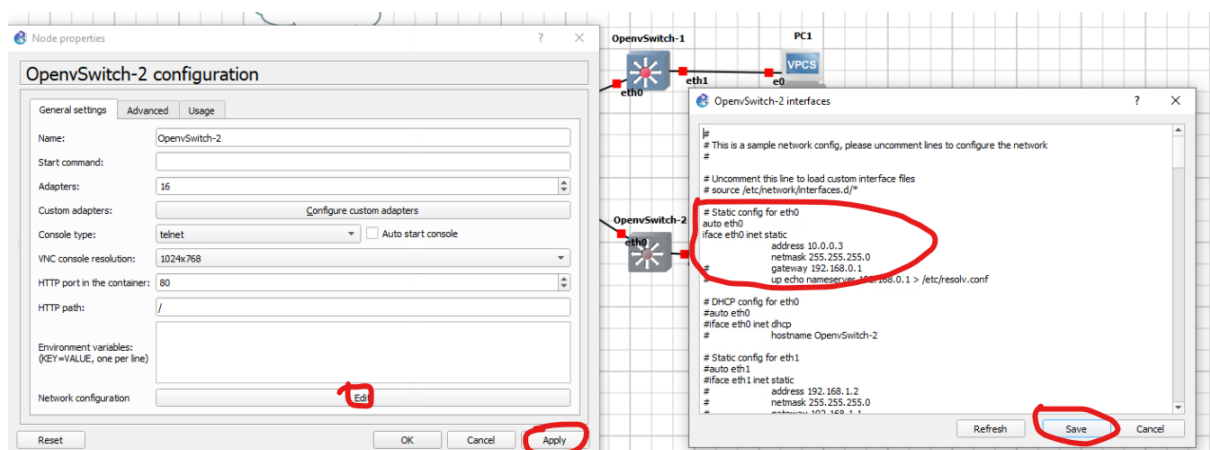
# Static config for eth1
auto eth1
iface eth1 inet static
    address 10.0.0.1
    netmask 255.255.255.0
    gateway 192.168.0.1
    up echo nameserver 192.168.0.1 > /etc/resolv.conf

# DHCP config for eth0
#auto eth0
#iface eth0 inet dhcp
#    hostname Onos-1
```

**Step 3:** Assign the static IP address on eth0 port of Open Switch -1



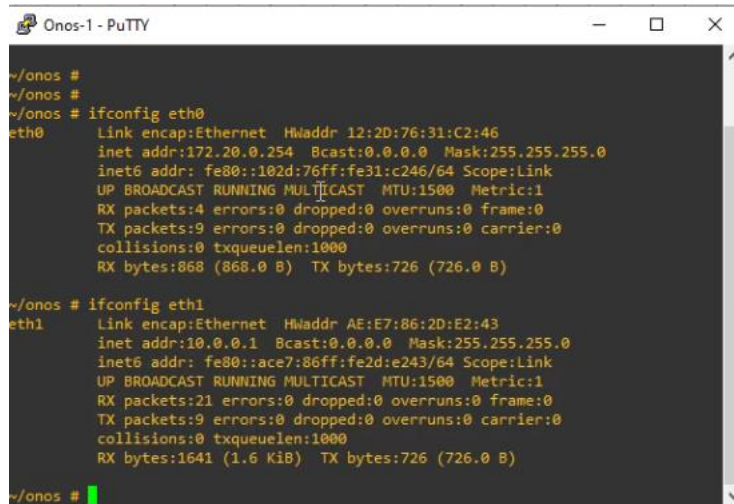
**Step 4:** Assign the static IP address on eth0 port of Open Switch -2



**Step 5:** Start the Cisco Switch first and wait till it come up.

**Step 6:** Start the ONOS Controller once SW come up

**Step 7:** Post ONOS come up, run the command ifconfig eth 0 and ifconfig eth1 and verify the IP



```
~/onos #  
~/onos #  
~/onos # ifconfig eth0  
eth0      Link encap:Ethernet  HWaddr 12:2D:76:31:C2:46  
          inet addr:172.20.0.254  Bcast:0.0.0.0  Mask:255.255.255.0  
          inet6 addr: fe80::102d:76ff:fe31:c246/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:868 (868.0 B)  TX bytes:726 (726.0 B)  
  
~/onos # ifconfig eth1  
eth1      Link encap:Ethernet  HWaddr AE:E7:86:2D:E2:43  
          inet addr:10.0.0.1  Bcast:0.0.0.0  Mask:255.255.255.0  
          inet6 addr: fe80::ace7:86ff:fe2d:e243/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:21 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:1641 (1.6 KiB)  TX bytes:726 (726.0 B)  
  
~/onos #
```

**Step 8:** Go to browser and open the URL 172.20.0.254:8181/onos/ui

Username: onos

password: rocks

**Step 9:** Start both the open vswitch

**Step 10:** check the IP in both the switch via ifconfig eth0 command

**Step 11:** Ping the controller IP 10.0.0.1 from both the SW

```
OpenSwitch-1 - PuTTY
OpenSwitch-1 console is now available... Press RETURN to get started.
2023-12-14T14:48:15Z|00001|ovs_numa|INFO|Discovered 8 CPU cores on NUMA node 0
2023-12-14T14:48:15Z|00002|ovs_numa|INFO|Discovered 1 NUMA nodes and 8 CPU cores
2023-12-14T14:48:15Z|00003|reconnect|INFO|unix:/var/run/openvswitch/db.sock: connecting...
2023-12-14T14:48:15Z|00004|reconnect|INFO|unix:/var/run/openvswitch/db.sock: connected
sh: 1: unknown operand
/#
/#
/# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 52:83:51:28:28:01
          inet addr:10.0.0.2  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::5083:51ff:fe28:28d1/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:53 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3973 (3.8 KiB)  TX bytes:1452 (1.4 KiB)

OpenSwitch-2 - PuTTY
OpenSwitch-2 console is now available... Press RETURN to get started.
2023-12-14T14:48:19Z|00001|ovs_numa|INFO|Discovered 8 CPU cores on NUMA node 0
2023-12-14T14:48:19Z|00002|ovs_numa|INFO|Discovered 1 NUMA nodes and 8 CPU cores
2023-12-14T14:48:19Z|00003|reconnect|INFO|unix:/var/run/openvswitch/db.sock: connecting...
2023-12-14T14:48:19Z|00004|reconnect|INFO|unix:/var/run/openvswitch/db.sock: connected
sh: 1: unknown operand
/# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 66:EF:87:13:41:4B
          inet addr:10.0.0.3  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::64ef:b7ff:fe13:414b/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:47 errors:0 dropped:1 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3311 (3.2 KiB)  TX bytes:1522 (1.4 KiB)

/#
```

**Step 12:** Configure the SW with controller information

**Setup protocol:** ovs-vsctl set bridge br0 protocols=OpenFlow13

**Setup controller:** ovs-vsctl set-controller br0 tcp:10.0.0.1:6633

```
OpenSwitch-1 - PuTTY
          inet6 addr: fe80::bc79:9eff:fe94:78a5/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:200 errors:0 dropped:1 overruns:0 frame:0
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14497 (14.1 KiB)  TX bytes:1802 (1.7 KiB)

/# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1): 56 data bytes
64 bytes from 10.0.0.1: seq=0 ttl=64 time=11.266 ms
64 bytes from 10.0.0.1: seq=1 ttl=64 time=5.703 ms
64 bytes from 10.0.0.1: seq=2 ttl=64 time=6.077 ms
64 bytes from 10.0.0.1: seq=3 ttl=64 time=6.462 ms
64 bytes from 10.0.0.1: seq=4 ttl=64 time=6.769 ms
64 bytes from 10.0.0.1: seq=5 ttl=64 time=5.061 ms
64 bytes from 10.0.0.1: seq=6 ttl=64 time=7.114 ms
^Z[1]+  Stopped                  ping 10.0.0.1
/#
/#
/#
/#
/# ovs-vsctl set bridge br0 protocols=OpenFlow13
/# ovs-vsctl set-controller br0 tcp:10.0.0.1:6633
/#

OpenSwitch-2 - PuTTY
64 bytes from 10.0.0.1: seq=7 ttl=64 time=5.418 ms
64 bytes from 10.0.0.1: seq=8 ttl=64 time=5.936 ms
64 bytes from 10.0.0.1: seq=9 ttl=64 time=6.403 ms
64 bytes from 10.0.0.1: seq=10 ttl=64 time=5.292 ms
64 bytes from 10.0.0.1: seq=11 ttl=64 time=5.515 ms
64 bytes from 10.0.0.1: seq=12 ttl=64 time=6.857 ms
64 bytes from 10.0.0.1: seq=13 ttl=64 time=6.287 ms
64 bytes from 10.0.0.1: seq=14 ttl=64 time=6.555 ms
64 bytes from 10.0.0.1: seq=15 ttl=64 time=4.295 ms
64 bytes from 10.0.0.1: seq=16 ttl=64 time=5.403 ms
64 bytes from 10.0.0.1: seq=17 ttl=64 time=5.329 ms
64 bytes from 10.0.0.1: seq=18 ttl=64 time=5.721 ms
64 bytes from 10.0.0.1: seq=19 ttl=64 time=5.942 ms
64 bytes from 10.0.0.1: seq=20 ttl=64 time=6.868 ms
64 bytes from 10.0.0.1: seq=21 ttl=64 time=6.901 ms
^Z[1]+  Stopped                  ping 10.0.0.1
/#
/#
/#
/#
/# ovs-vsctl set bridge br0 protocols=OpenFlow13
/# ovs-vsctl set-controller br0 tcp:10.0.0.1:6633
/#
```

**Step 13:** Set the IP on both the PC and PING – It will be successful

PC1: IP 192.168.1.1/24

PC2: ip 192.168.1.2/24

**Step 14:** Go to ONOS GUI and enable the OpenFlow base application, provider suit and OpenFlow overlay in application TAB

## Applications (170 Total)

openflow All Fields

	Title	App ID
✓	OpenFlow Base Provider	org.onosproject.openflow-base
✓	OpenFlow Provider Suite	org.onosproject.openflow
✓	Openflow overlay	org.onosproject.workflow.ofoverlay
■	Control Message Stats Provider	org.onosproject.openflow-message
■	OpenFlow Agent	org.onosproject.ofagent
■	Polatis OpenFlow Device Drivers	org.onosproject.drivers.polatis.openflow

**Step 15:** Check in Open switch , controller in connected mode and Ping from PC1 to PC2 fail

ovs-vsctl show

```

#
# ovs-vsctl show
89b1061a-ca0d-43c7-98cf-94637e6676f4
Bridge br3
  datapath type: netdev
  Port br3
    Interface br3
      type: internal
Bridge br1
  datapath type: netdev
  Port br1
    Interface br1
      type: internal
Bridge br0
  Controller "tcp:10.0.0.1:6633"
    is_connected: true
  datapath type: netdev
  Port eth14
    Interface eth14
      type: internal

```

```

#
# ovs-vsctl show
52bde531-8000-4000-8000-800080008000
Bridge br0
  Controller "tcp:10.0.0.1:6633"
    is_connected: true
  datapath type: netdev
  Port eth5
    Interface eth5
      type: internal
  Port br0
    Interface br0
      type: internal
  Port eth10
    Interface eth10
      type: internal
  Port eth11
    Interface eth11
      type: internal
  Port eth15
    Interface eth15
      type: internal

```

**Step 16:** Ping PC1 to PC2 – **Ping Fail** as Open V switch is connected with ONOS Controller and all the forwarding decision is not moved to ONOS Controller.

**Step 17:** Enable reactive forwarding in application and check, **PC1 to PC2 ping should be successful.**

## Applications (170 Total)

reactive All Fields

	Title	App ID
✓	Reactive Forwarding	org.onosproject.fwd
■	SDN-IP Reactive Routing	org.onosproject.reactive-routing

## Step 18: Check the Devices , Topology on ONOS GUI

Devices (2 total)						
Search		All Fields ▾				
	FRIENDLY NAME ▲	DEVICE ID	MASTER	PORTS	VENDOR	H/W VERSION
✓	of:000022abc5ae324e	of:000022abc5ae324e	172.20.0.254	17	Nicira, Inc.	Open vSwitch
✓	of:0000de63861dec45	of:0000de63861dec45	172.20.0.254	17	Nicira, Inc.	Open vSwitch

## Step 19: Connect to Postman

<http://172.20.0.254:8181/onos/v1/devices>

<http://172.20.0.254:8181/onos/v1/flows>

<http://172.20.0.254:8181/onos/v1/links>

<http://172.20.0.254:8181/onos/v1/topology>

The screenshot displays two Postman requests to the ONOS GUI. The first request is a GET to `http://172.20.0.254:8181/onos/v1/topology` with Basic Auth (Username: onos, Password: rocks). The response is a JSON object with the following structure:

```
{
  "time": 3879364248283487,
  "devices": 2,
  "links": 2,
  "clusters": 2
}
```

The second request is a GET to `http://172.20.0.254:8181/onos/v1/devices` with the same Basic Auth. The response is a JSON array of device objects:

```
[
  {
    "id": "of:0000de63861dec45",
    "type": "SWITCH",
    "available": true,
    "role": "MASTER",
    "mfr": "Nicira, Inc.",
    "hw": "Open vSwitch",
    "sw": "2.17.6",
    "serial": "None",
    "driver": "ova",
    "chassisId": "de63861dec45"
  }
]
```