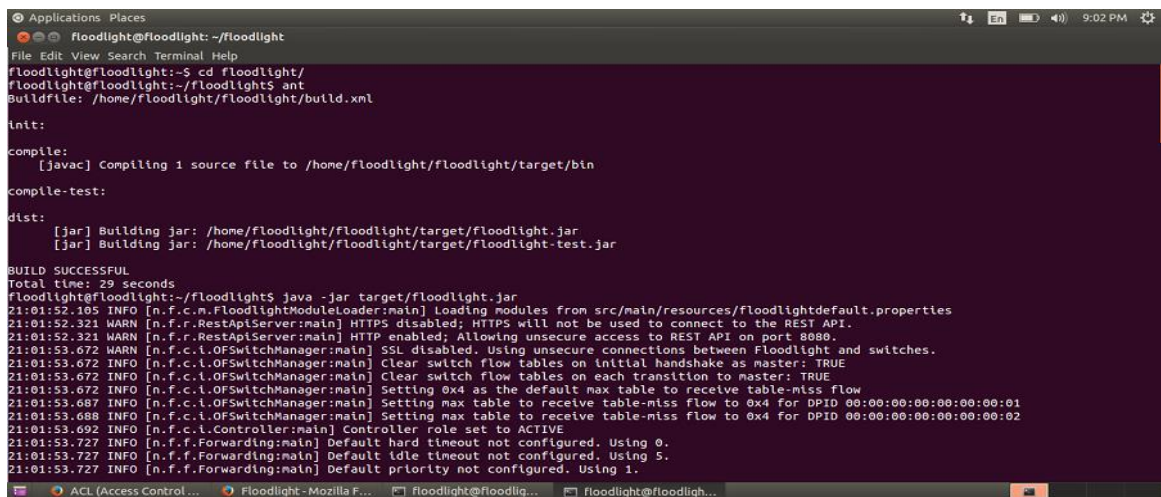


## Project 2 - Floodlight Firewall App

Firstly, to setup SDN environment we install Floodlight virtual box application from official website of Floodlight. Then after setting up floodlight virtual machine inside virtual box, we start to setup floodlight environment, enable firewall application (ACL REST API), and test Firewall REST API on our system through sequence of commands explained below.

1. Firstly, we use `ant` command to build all the java files inside the floodlight directory, Then, we use `java -jar target/floodlight.jar` to run the Floodlight within the VM.



```
Applications Places
Floodlight@Floodlight: ~/floodlight
File Edit View Search Terminal Help
Floodlight@Floodlight:~$ cd floodlight/
Floodlight@Floodlight:~/floodlight$ ant
Buildfile: /home/floodlight/floodlight/build.xml

Init:

Compile:
[javac] Compiling 1 source file to /home/floodlight/floodlight/target/bin

Compile-test:

Dist:
[jar] Building jar: /home/floodlight/floodlight/target/floodlight.jar
[jar] Building jar: /home/floodlight/floodlight/target/floodlight-test.jar

BUILD SUCCESSFUL
Total time: 29 seconds
Floodlight@Floodlight:~/floodlight$ java -jar target/floodlight.jar
21:01:52.105 INFO [n.f.c.m.FloodlightModuleLoader:main] Loading modules from src/main/resources/floodlightdefault.properties
21:01:52.321 WARN [n.f.r.RestApiServer:main] HTTPS disabled; HTTPS will not be used to connect to the REST API.
21:01:53.672 WARN [n.f.r.RestApiServer:main] HTTP enabled; Allowing unsecure access to REST API on port 8080.
21:01:53.672 INFO [n.f.c.l.OFSwitchManager:main] Clear switch flow tables on initial handshake as master: TRUE
21:01:53.672 INFO [n.f.c.l.OFSwitchManager:main] Setting max table to receive table-miss flow to 0x4 for DPID 00:00:00:00:00:00:01
21:01:53.687 INFO [n.f.c.l.OFSwitchManager:main] Setting max table to receive table-miss flow to 0x4 for DPID 00:00:00:00:00:00:02
21:01:53.692 INFO [n.f.c.l.Controller:main] Controller role set to ACTIVE
21:01:53.727 INFO [n.f.f.Forwarding:main] Default hard timeout not configured. Using 0.
21:01:53.727 INFO [n.f.f.Forwarding:main] Default idle timeout not configured. Using 5.
21:01:53.727 INFO [n.f.f.Forwarding:main] Default priority not configured. Using 1.
```

Fig. 1: Build Java files inside floodlight directory

- a) Then, using command `java -jar target/floodlight.jar` command shown above, we get our localhost setup complete as shown below in ss.

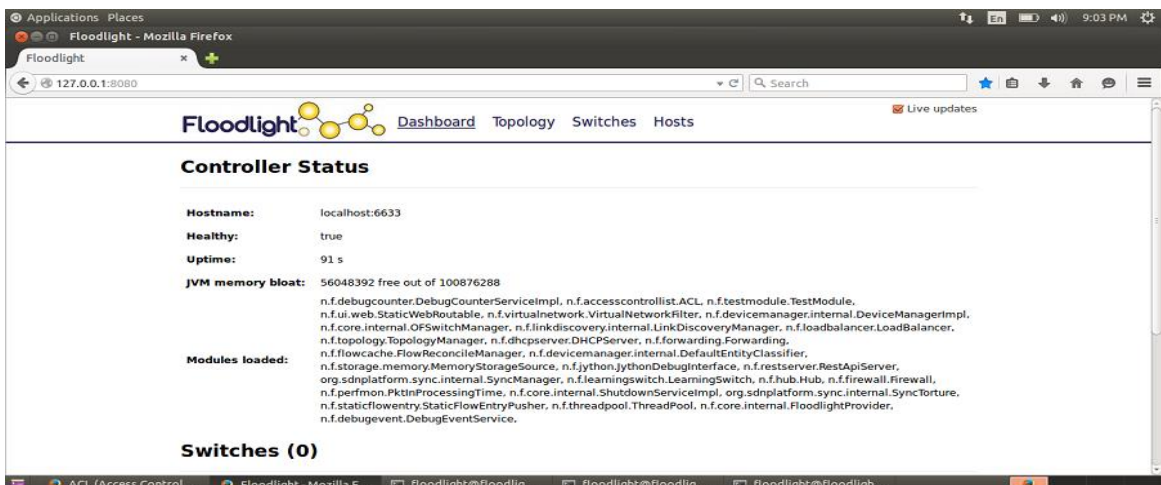
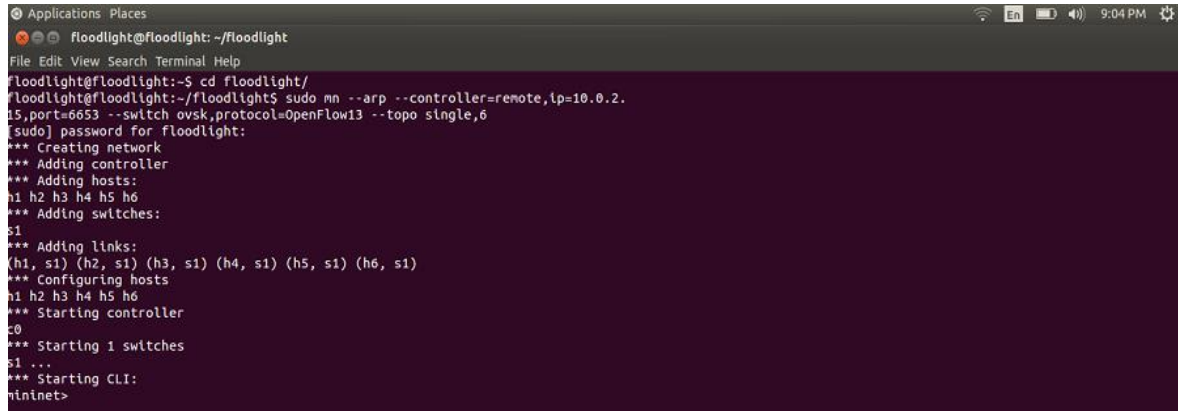


Fig. 1a: Checking localhost

2. Now, since floodlight is running we will attach it to OpenFlow network. We used Mininet which is one of the best tools for this purpose. Then, we used mininet against remote controller using command:

```
sudo mn - --arp - --controller=remote,ip=10.0.2.15,port=6653 -switch  
ovsk.protocol=OpenFlow13 -topo single,6
```

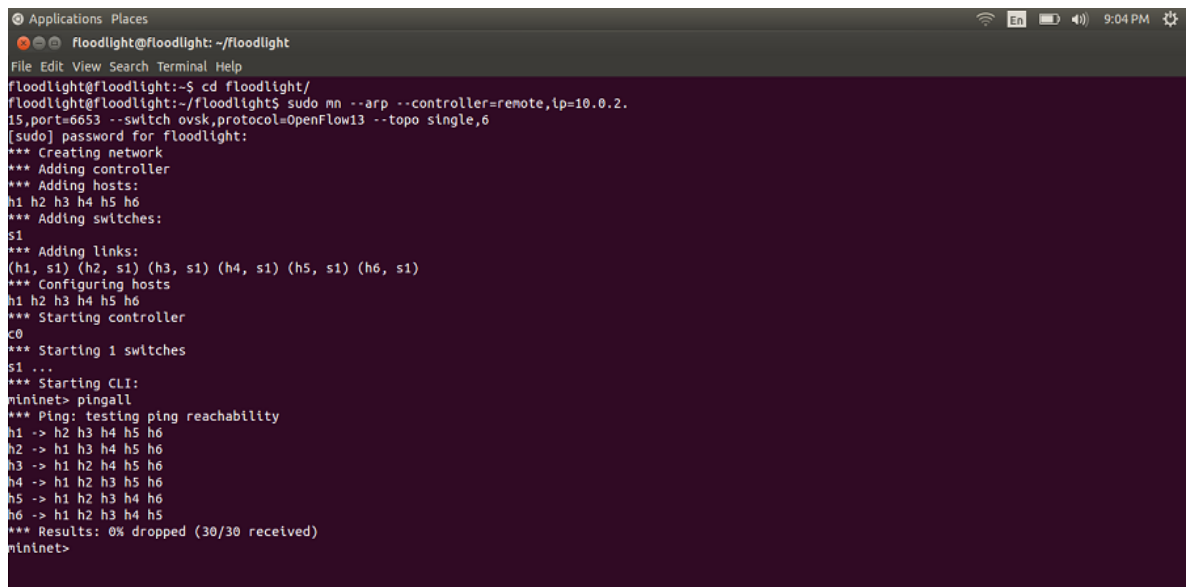
A single topology consists of a single switch connected to a number of hosts as specified in the topology build command. Here, we are using a single topology with 6 hosts connected to the switch.



```
Applications Places
Floodlight@Floodlight: ~/Floodlight
File Edit View Search Terminal Help
Floodlight@Floodlight:~$ cd Floodlight/
Floodlight@Floodlight:~/Floodlight$ sudo mn --arp --controller=remote,ip=10.0.2.
15,port=6653 --switch ovsk,protocol=OpenFlow13 --topo single,6
[sudo] password for floodlight:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Fig.2: Using Mininet tool to implement OpenFlow

- a) Then using pingall command we check if hosts are communicating with each other through switch as shown below.



```
Applications Places
Floodlight@Floodlight: ~/Floodlight
File Edit View Search Terminal Help
Floodlight@Floodlight:~$ cd Floodlight/
Floodlight@Floodlight:~/Floodlight$ sudo mn --arp --controller=remote,ip=10.0.2.
15,port=6653 --switch ovsk,protocol=OpenFlow13 --topo single,6
[sudo] password for floodlight:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
mininet>
```

Fig 2.a: Using pingall command

b) Below SS shows the single topology with 6 hosts and one switch.



Applications Places  
Floodlight - Mozilla Firefox

For Developers - Flo... x Floodlight REST API ... x ACL (Access Control... x Floodlight x +

127.0.0.1:8080

Search

Live updates

Floodlight [Dashboard](#) Topology Switches Hosts

### Switches (1)

DPID	IP Address	Vendor	Packets	Bytes	Flows	Connected Since
00:00:00:00:00:00:01	/10.0.2.15:38132	Nicira, Inc.	55	4878	5	10/22/2018, 9:04:26 PM

### Hosts (6)

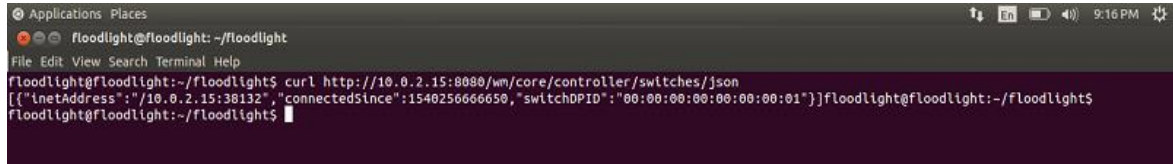
MAC Address	IP Address	Switch Port	Last Seen
d2:cc:af:a2:0f:1e		00:00:00:00:00:00:01-2	10/22/2018, 9:04:50 PM
6a:c6:ec:e6:95:b8		00:00:00:00:00:00:01-5	10/22/2018, 9:04:50 PM
b2:ac:7c:9a:16:4c		00:00:00:00:00:00:01-4	10/22/2018, 9:04:50 PM
4e:7f:eb:c6:f3:b3		00:00:00:00:00:00:01-1	10/22/2018, 9:04:50 PM
be:3a:93:cf:98:c0	0.0.0.0	00:00:00:00:00:00:01-6	10/22/2018, 9:04:50 PM

Fig.2.c: Checking Hosts and Switches in localhost

## Now we, will be executing FIREWALL REST API and Floodlight REST API calls

### Using ACL REST API's

1. **Using CURL to access REST API:** An example REST call that retrieves data from Floodlight at IP address <controller-ip> i.e. 10.0.2.15 is:

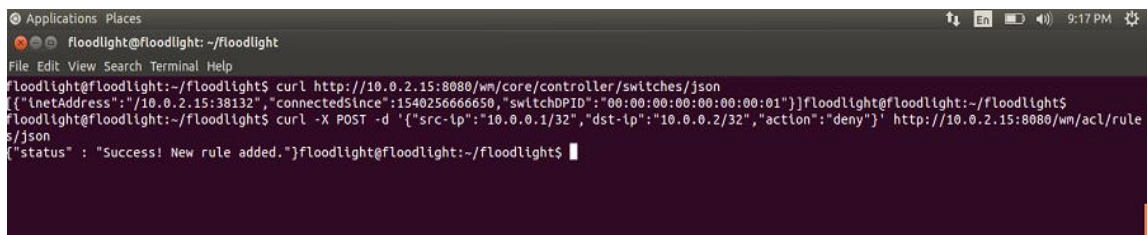


```
floodlight@floodlight: ~/floodlight
File Edit View Search Terminal Help
floodlight@floodlight:~/floodlight$ curl http://10.0.2.15:8080/wm/core/controller/switches/json
[{"inetAddress":"/10.0.2.15:38132","connectedSince":1540256666650,"switchDPID":"00:00:00:00:00:00:01"}]floodlight@floodlight:~/floodlight$
```

Fig.1: Using CURL to access REST APIs

2. **Adding an ACL Rule:** Now, we will add an ACL rule through ACL REST Interface using command `curl -X POST -d '{"src-ip":"10.0.0.1/32","dst-ip":"10.0.0.2/32","action":"deny"}' http://10.0.2.15:8080/wm/acl/rules/json`

This command states that any traffic from source ip:10.0.0.1 to destination ip:10.0.0.2 will be denied by the switch.



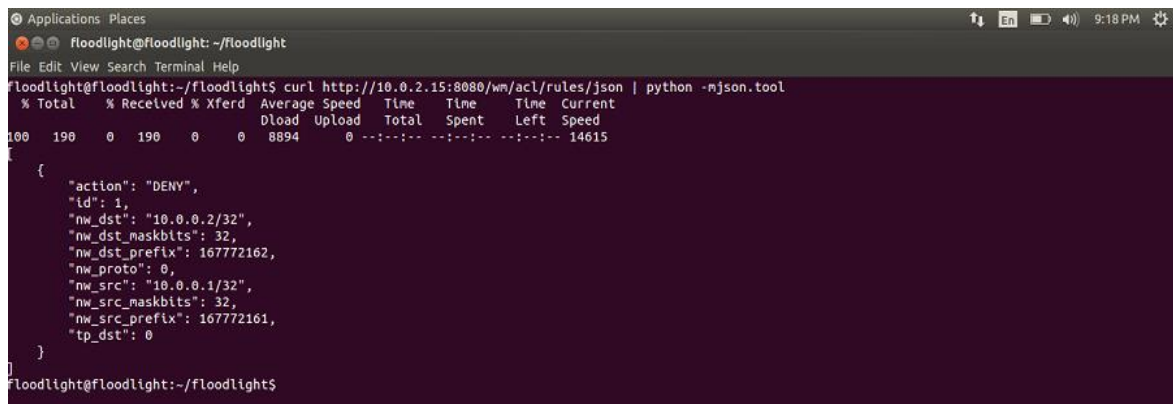
```
floodlight@floodlight: ~/floodlight
File Edit View Search Terminal Help
floodlight@floodlight:~/floodlight$ curl http://10.0.2.15:8080/wm/core/controller/switches/json
[{"inetAddress":"/10.0.2.15:38132","connectedSince":1540256666650,"switchDPID":"00:00:00:00:00:00:01"}]floodlight@floodlight:~/floodlight$
floodlight@floodlight:~/floodlight$ curl -X POST -d '{"src-ip":"10.0.0.1/32","dst-ip":"10.0.0.2/32","action":"deny"}' http://10.0.2.15:8080/wm/acl/rules/json
{"status": "Success! New rule added."}floodlight@floodlight:~/floodlight$
```

Fig.2: Adding an ACL Rule

3. **Listing All ACL Rules:** To list all the ACL rules currently active on the switch, we type the command

`curl http://10.0.2.15:8080/wm/acl/rules/json | python -mjson.tool`

This lists all the ACL rules currently active as shown below.

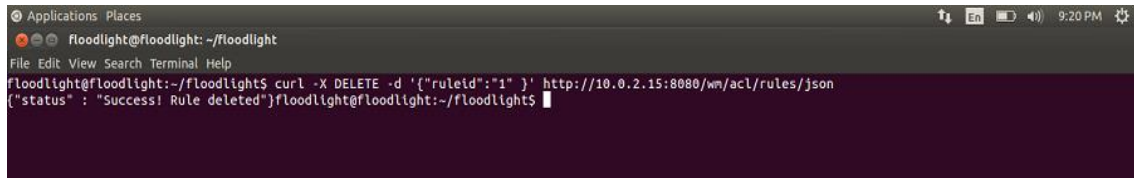


```
floodlight@floodlight: ~/floodlight
File Edit View Search Terminal Help
floodlight@floodlight:~/floodlight$ curl http://10.0.2.15:8080/wm/acl/rules/json | python -mjson.tool
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %    190    0    190    0    0    8894    0 --:--:-- --:--:-- --:--:-- 14615
{
  {
    "action": "DENY",
    "id": 1,
    "nw_dst": "10.0.0.2/32",
    "nw_dst_maskbits": 32,
    "nw_dst_prefix": 167772162,
    "nw_proto": 0,
    "nw_src": "10.0.0.1/32",
    "nw_src_maskbits": 32,
    "nw_src_prefix": 167772161,
    "tp_dst": 0
  }
}
floodlight@floodlight:~/floodlight$
```

Fig.3: Listing all ACL Rules

4. **Removing ACL Rule:** To remove ACL rule, we use command `curl -X DELETE -d '{"ruleid":"1"}' http://10.0.2.15:8080/wm/acl/rules/json`

After executing the command, it deletes the ACL rule with a prompt Success! Rule deleted

A terminal window titled 'Applications Places' with the prompt 'floodlight@floodlight: ~/floodlight'. The command 'curl -X DELETE -d '{"ruleid":"1"}' http://10.0.2.15:8080/wm/acl/rules/json' is entered. The output is '{"status": "Success! Rule deleted"}'.

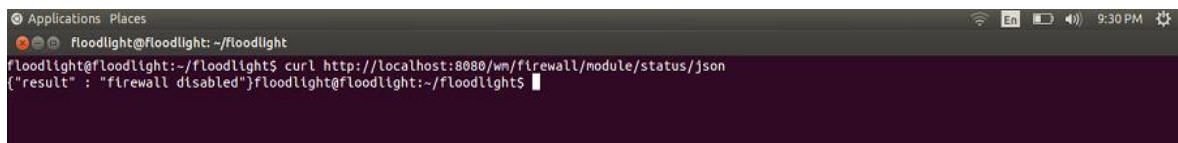
```
floodlight@floodlight:~/floodlight$ curl -X DELETE -d '{"ruleid":"1"}' http://10.0.2.15:8080/wm/acl/rules/json
{"status": "Success! Rule deleted"}floodlight@floodlight:~/floodlight$
```

Fig.4: Removing ACL Rule

## Examples Using CURL – Firewall REST API

1. **Check Status of Firewall:** To check the status of Firewall if it is enabled or disabled, we used command `curl http://localhost:8080/wm/firewall/module/status/json`

After executing the command, it shows that firewall is disabled in the terminal.

A terminal window titled 'Applications Places' with the prompt 'floodlight@floodlight: ~/floodlight'. The command 'curl http://localhost:8080/wm/firewall/module/status/json' is entered. The output is '{"result": "firewall disabled"}'.

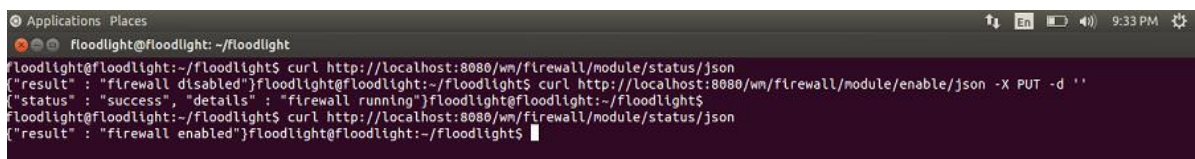
```
floodlight@floodlight:~/floodlight$ curl http://localhost:8080/wm/firewall/module/status/json
{"result": "firewall disabled"}floodlight@floodlight:~/floodlight$
```

Fig.1: Check Status of Firewall

2. **Enable the Firewall:** Now, to enable the firewall using command

`curl http://localhost:8080/wm/firewall/module/enable/json -X PUT -d ''`

After executing the command, we rechecked the status of firewall and it showed enabled as shown below.

A terminal window titled 'Applications Places' with the prompt 'floodlight@floodlight: ~/floodlight'. The first command 'curl http://localhost:8080/wm/firewall/module/status/json' is entered, showing '{"result": "firewall disabled"}'. The second command 'curl http://localhost:8080/wm/firewall/module/enable/json -X PUT -d ''' is entered, showing '{"status": "success", "details": "firewall running"}'. The third command 'curl http://localhost:8080/wm/firewall/module/status/json' is entered, showing '{"result": "firewall enabled"}'.

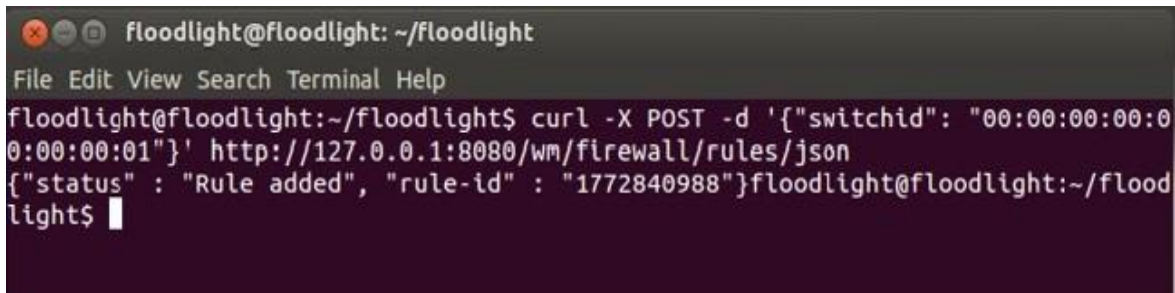
```
floodlight@floodlight:~/floodlight$ curl http://localhost:8080/wm/firewall/module/status/json
{"result": "firewall disabled"}floodlight@floodlight:~/floodlight$ curl http://localhost:8080/wm/firewall/module/enable/json -X PUT -d ''
{"status": "success", "details": "firewall running"}floodlight@floodlight:~/floodlight$ curl http://localhost:8080/wm/firewall/module/status/json
{"result": "firewall enabled"}floodlight@floodlight:~/floodlight$
```

Fig.2: Enable the Firewall



3. **Adding ALLOW Rule for flows through Switch:** To add ALLOW rule for all flows through switch we used the command

```
curl -X POST -d '{"switchid": "00:00:00:00:00:00:00:01"}'
http://localhost:8080/wm/firewall/rules/json
```



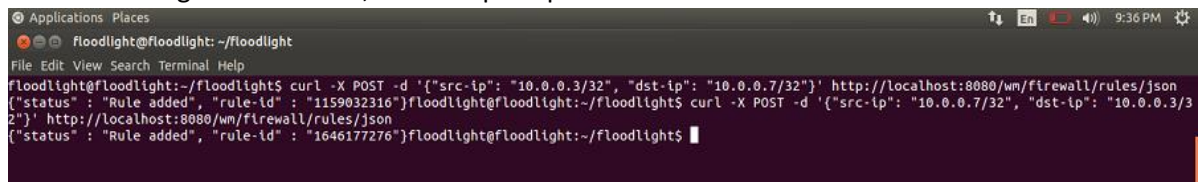
```
floodlight@floodlight: ~/floodlight
File Edit View Search Terminal Help
floodlight@floodlight:~/floodlight$ curl -X POST -d '{"switchid": "00:00:00:00:00:00:00:01"}' http://127.0.0.1:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1772840988"}floodlight@floodlight:~/floodlight$
```

Fig.3: Adding ALLOW rule for flow through Switch

4. **Adding ALLOW Rule for flows between two ip addresses:** To add ALLOW rule for all flows between IP hosts. Not specifying action implies ALLOW rule. We used command

```
curl -X POST -d '{"src-ip": "10.0.0.3/32", "dst-ip": "10.0.0.7/32"}'
http://localhost:8080/wm/firewall/rules/json
curl -X POST -d '{"src-ip": "10.0.0.7/32", "dst-ip": "10.0.0.3/32"}'
http://localhost:8080/wm/firewall/rules/json
```

After executing the command, it shows prompt: Rule added.



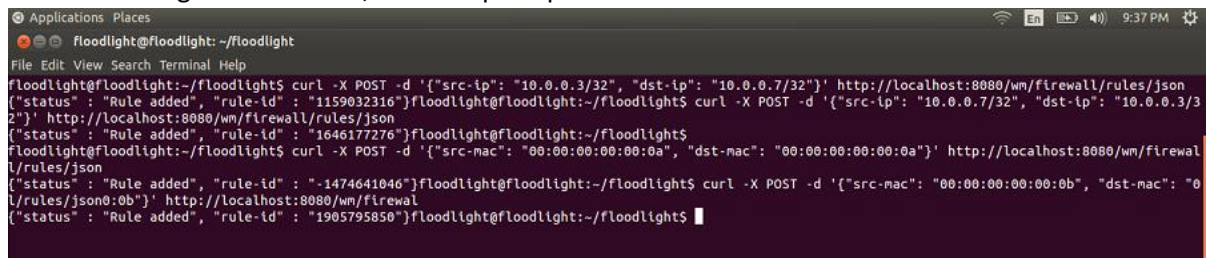
```
floodlight@floodlight: ~/floodlight
File Edit View Search Terminal Help
floodlight@floodlight:~/floodlight$ curl -X POST -d '{"src-ip": "10.0.0.3/32", "dst-ip": "10.0.0.7/32"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1159032316"}floodlight@floodlight:~/floodlight$ curl -X POST -d '{"src-ip": "10.0.0.7/32", "dst-ip": "10.0.0.3/32"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1646177276"}floodlight@floodlight:~/floodlight$
```

Fig.4: Adding ALLOW Rule for flow between two IP addresses

5. **Adding ALLOW rule between two MAC addresses:** To add ALLOW rule for all flows between host mac 00:00:00:00:00:0a and host 00:00:00:00:00:0b we use the command

```
curl -X POST -d '{"src-mac": "00:00:00:00:00:0a", "dst-mac": "00:00:00:00:00:0a"}' http://localhost:8080/wm/firewall/rules/json
curl -X POST -d '{"src-mac": "00:00:00:00:00:0b", "dst-mac": "00:00:00:00:00:0b"}' http://localhost:8080/wm/firewall/rules/json
```

After executing the command, it shows prompt: Rule added.



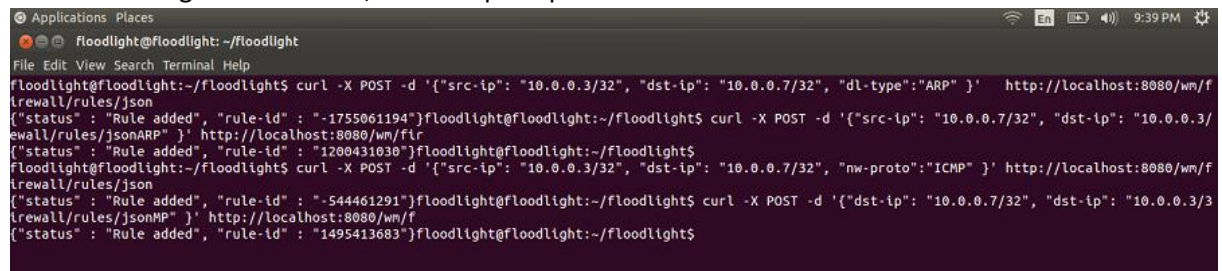
```
floodlight@floodlight: ~/floodlight
File Edit View Search Terminal Help
floodlight@floodlight:~/floodlight$ curl -X POST -d '{"src-ip": "10.0.0.3/32", "dst-ip": "10.0.0.7/32"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1159032316"}floodlight@floodlight:~/floodlight$ curl -X POST -d '{"src-ip": "10.0.0.7/32", "dst-ip": "10.0.0.3/32"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1646177276"}floodlight@floodlight:~/floodlight$
floodlight@floodlight:~/floodlight$ curl -X POST -d '{"src-mac": "00:00:00:00:00:0a", "dst-mac": "00:00:00:00:00:0a"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1474641046"}floodlight@floodlight:~/floodlight$ curl -X POST -d '{"src-mac": "00:00:00:00:00:0b", "dst-mac": "00:00:00:00:00:0b"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1905795850"}floodlight@floodlight:~/floodlight$
```

Fig.5: Adding ALLOW Rule between two MAC addresses

6. **Adding an ALLOW rule for ping to work between IP hosts:** To add ALLOW rule for ping to work between ip hosts we used the command

```
curl -X POST -d '{"src-ip": "10.0.0.3/32", "dst-ip": "10.0.0.7/32", "dl-type":"ARP"}' http://localhost:8080/wm/firewall/rules/json
curl -X POST -d '{"src-ip": "10.0.0.7/32", "dst-ip": "10.0.0.3/32", "dl-type":"ARP"}' http://localhost:8080/wm/firewall/rules/json
curl -X POST -d '{"src-ip": "10.0.0.3/32", "dst-ip": "10.0.0.7/32", "nw-proto":"ICMP"}' http://localhost:8080/wm/firewall/rules/json
curl -X POST -d '{"dst-ip": "10.0.0.7/32", "src-ip": "10.0.0.3/32", "nw-proto":"ICMP"}' http://localhost:8080/wm/firewall/rules/json
```

After executing the command, it shows prompt: Rule added.



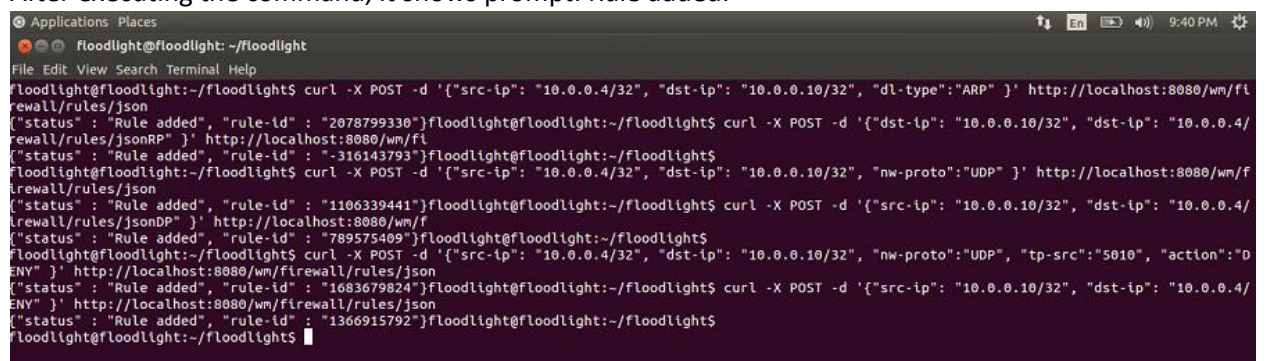
```
Applications Places
Floodlight@Floodlight: ~/Floodlight
File Edit View Search Terminal Help
Floodlight@Floodlight:~/Floodlight$ curl -X POST -d '{"src-ip": "10.0.0.3/32", "dst-ip": "10.0.0.7/32", "dl-type":"ARP"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "-1755061194"}Floodlight@Floodlight:~/Floodlight$ curl -X POST -d '{"src-ip": "10.0.0.7/32", "dst-ip": "10.0.0.3/32", "dl-type":"ARP"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1200431030"}Floodlight@Floodlight:~/Floodlight$ curl -X POST -d '{"src-ip": "10.0.0.3/32", "dst-ip": "10.0.0.7/32", "nw-proto":"ICMP"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "-544461291"}Floodlight@Floodlight:~/Floodlight$ curl -X POST -d '{"dst-ip": "10.0.0.7/32", "src-ip": "10.0.0.3/32", "nw-proto":"ICMP"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1495413683"}Floodlight@Floodlight:~/Floodlight$
```

Fig.6: Adding ALLOW Rule between two IP Hosts

7. **Adding an ALLOW rule for UDP (such as iperf) to work between IP hosts:** To add ALLOW rule for UDP to work between two ip hosts, we used command `curl -X POST -d '{"src-ip": "10.0.0.4/32", "dst-ip": "10.0.0.10/32", "dl-type":"ARP"}' http://localhost:8080/wm/firewall/rules/json`

```
curl -X POST -d '{"dst-ip": "10.0.0.10/32", "src-ip": "10.0.0.4/32", "dl-type":"ARP"}' http://localhost:8080/wm/firewall/rules/json
curl -X POST -d '{"src-ip": "10.0.0.4/32", "dst-ip": "10.0.0.10/32", "nw-proto":"UDP"}' http://localhost:8080/wm/firewall/rules/json
curl -X POST -d '{"src-ip": "10.0.0.10/32", "dst-ip": "10.0.0.4/32", "nw-proto":"UDP"}' http://localhost:8080/wm/firewall/rules/json
```

After executing the command, it shows prompt: Rule added.



```
Applications Places
Floodlight@Floodlight: ~/Floodlight
File Edit View Search Terminal Help
Floodlight@Floodlight:~/Floodlight$ curl -X POST -d '{"src-ip": "10.0.0.4/32", "dst-ip": "10.0.0.10/32", "dl-type":"ARP"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "2078799330"}Floodlight@Floodlight:~/Floodlight$ curl -X POST -d '{"dst-ip": "10.0.0.10/32", "src-ip": "10.0.0.4/32", "dl-type":"ARP"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "-316143793"}Floodlight@Floodlight:~/Floodlight$ curl -X POST -d '{"src-ip": "10.0.0.4/32", "dst-ip": "10.0.0.10/32", "nw-proto":"UDP"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1106339441"}Floodlight@Floodlight:~/Floodlight$ curl -X POST -d '{"src-ip": "10.0.0.10/32", "dst-ip": "10.0.0.4/32", "nw-proto":"UDP"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "789575409"}Floodlight@Floodlight:~/Floodlight$ curl -X POST -d '{"src-ip": "10.0.0.4/32", "dst-ip": "10.0.0.10/32", "nw-proto":"UDP", "tp-src":"5010", "action":"DENY"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1683679824"}Floodlight@Floodlight:~/Floodlight$ curl -X POST -d '{"src-ip": "10.0.0.10/32", "dst-ip": "10.0.0.4/32", "nw-proto":"UDP", "tp-src":"5010", "action":"DENY"}' http://localhost:8080/wm/firewall/rules/json
{"status": "Rule added", "rule-id": "1366915792"}Floodlight@Floodlight:~/Floodlight$
```

Fig.7: Adding ALLOW rule for UDP