

**Control Mininet hosts using ONOS SDN controller**

**Subject:** Advance topic in information security

**Lecturer:** Dr.Karthikeyan Supramanian

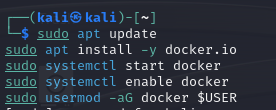
Ali Sulaiman Albalushi – 2021293061

Abdulaziz Salim Alkalbani – 2021293039

Ali Ibrahim Almaaini -2021293015

* **In this project we explor how to connect mininet in with ONOS controller, and controller the hosts of mininet, by blocking traffic from h1 and h2 using access list**

## **This command is used to update linux machine, download and start docker**



sudo apt update

sudo apt install -y docker.io

sudo systemctl start docker

sudo systemctl enable docker

sudo usermod -aG docker $USER

## **This command used to run and install ONOS SDN controller**

A screen shot of a computer

AI-generated content may be incorrect.

docker run -d --name onos \

-p 8181:8181 -p 6653:6653 -p 8101:8101 \

onosproject/onos

* **In order to run ACL and open ONOS web interface we need to download some features first, so after running ONOS we can access it using SSH with the password for onos user rocks**
* **We can also use ssh -p 8101 karaf@localhost and password is karaf**



## **The feature we download is**

ssh -p 8101 onos@localhost

password: rocks

app activate org.onosproject.openflow

app activate org.onosproject.fwd

app activate org.onosproject.acl

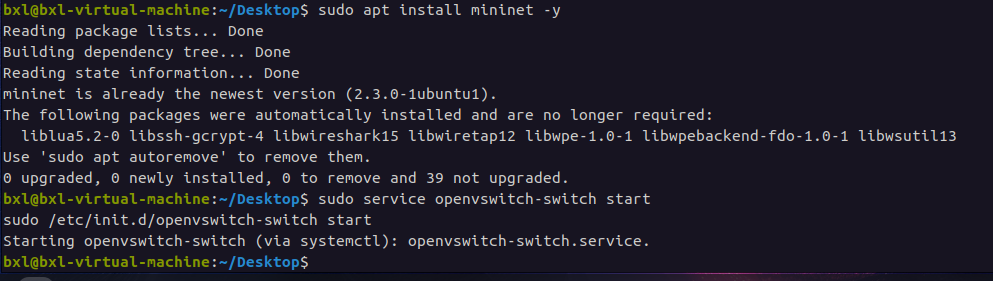
## **now the ONOS sdn web interface is can be access successfully**

* **Username:** onos
* **Password:** rocks
  + - * <http://localhost:8181/onos/ui>

A screenshot of a computer

AI-generated content may be incorrect.

## **Install and configure mininet:**



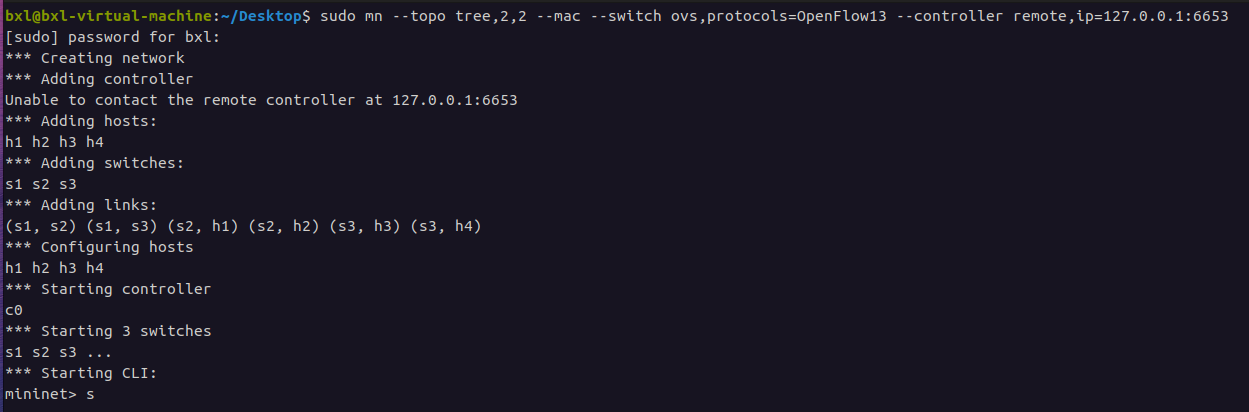
sudo apt install mininet

sudo service openvswitch-switch start

sudo /etc/init.d/openvswitch-switch start

## **After running mininet using the next command that will create 3 switches and 4 hosts**

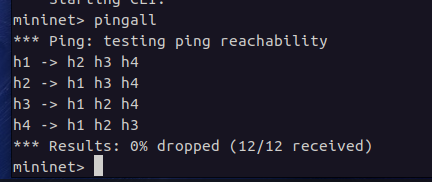
sudo mn --topo tree,2,2 --mac --switch ovs,protocols=OpenFlow13 --controller remote,ip=127.0.0.1:6653



A screenshot of a computer

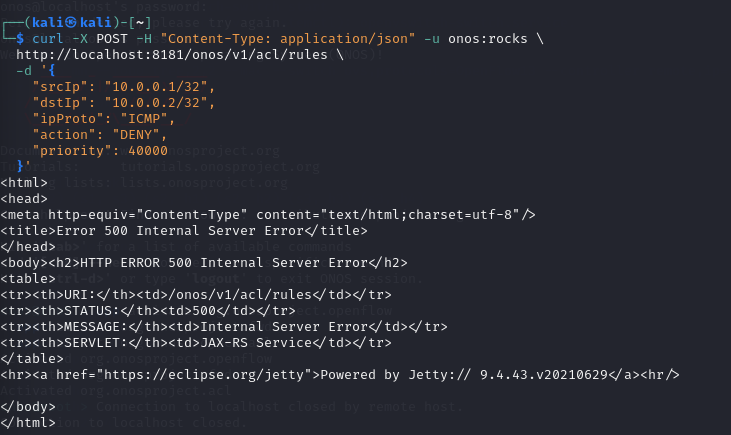
AI-generated content may be incorrect.

## **If we PING all hosts, we can see that all hosts receive and respond to ping**



Mininet> pingall

## **This Command is used to block ICMP access from H1 to H2.**



curl -X POST -H "Content-Type: application/json" -u onos:rocks \

http://localhost:8181/onos/v1/acl/rules \

-d '{

"srcIp": "10.0.0.1/32",

"dstIp": "10.0.0.2/32",

"ipProto": "ICMP",

"action": "DENY",

"priority": 40000

}'

**This command is used to allow ICMP access from H2 to H1 – NOT WORKING**

**And the second command is used to show all rules set for this network**

A computer screen shot of white text

AI-generated content may be incorrect.

curl -X POST -H "Content-Type: application/json" -u onos:rocks \

http://localhost:8181/onos/v1/acl/rules \

-d '{

"srcIp": "10.0.0.2/32",

"dstIp": "10.0.0.1/32",

"ipProto": "ICMP",

"action": "ALLOW",

"priority": 50000

}'

## **Check available rules**

curl -u onos:rocks http://localhost:8181/onos/v1/acl/rules

## **After we applied all rules we can ping all hosts again, we noticed the neither h1 can ping h2 nor h2 can ping h1**

A black background with white text

AI-generated content may be incorrect.

Mininet> pingall

## **This page from ONOS web interface that show all switches connected to our netwok**

A screenshot of a computer

AI-generated content may be incorrect.

## **This page showing all hostes connected to the network**

A screenshot of a computer

AI-generated content may be incorrect.

## **Other commands**

* **Stop ONOS sdn**

docker stop onos

* **Remove ONOS sdn**

docker rm docker

* **Enter ONOS docker container**

docker exec -it onos /bin/bash

* **ssh as root**

ssh -p 8101 karaf@localhost

password: karaf

* **List rules**

curl -u onos:rocks http://localhost:8181/onos/v1/acl/rules

* **Delete rules**

curl -X DELETE -u onos:rocks [http://localhost:8181/onos/v1/acl/rules/{ruleId}](http://localhost:8181/onos/v1/acl/rules/%7bruleId%7d)

**The END**