Exercises - ONOS

These exercises will make you feel comfortable with ONOS. The source material of this exercise is in: https://wiki.onosproject.org/display/ONOS/Basic+ONOS+Tutorial

This exercise is divided in three parts:

1. Deploying ONOS - ONOS commands and GUI
2. ONOS Applications
3. ONOS Intents

Part 1: Deploying ONOS - ONOS commands and GUI

Start ONOS as a single container instance.

```
$ docker run -t -d -p 8181:8181 -p 8101:8101 -p 5005:5005 -p 830:830 --name onos onosproject
```

To login inside the ONOS command line use ssh (password karaf), but first get ONOS IP address.

```
docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' onos
```

```
ssh -p 8101 karaf@`ONOS IP Address`
```

From now on, commands willb e typed in the ONOS interactive CLI. For instance, check which apps are activated.

```
onos> apps -a -s
```

OBS: To end the ONOS ssh session just type logout + ENTER in the ONOS CLI.

Checking the ONOS GUI, open a browser and enter http://localhost:8181/onos/ui. Login as user onos with password rocks.

Probably you will not see any Topology devices in the initial screen, so let's start Mininet. In another terminal (tab or window) start mininet, don't forget to fill the right `ONOS IP Address`.

```
sudo mn --topo=att --custom=onos-exercises/rftesttopo.py --controller remote,ip=`ONOS IP Ad
```

See what happened in the ONOS GUI. No devices yet, right?

In the mininet console try to ping hosts.

```
mininet> h1 ping -c3 h2
```

Part 2: ONOS Applications

Let's start loading Apps in ONOS, so hosts can reach each other.

Get back to the ONOS ssh session terminal (interactive CLI), and load some applications.

```
onos> app activate org.onosproject.openflow
onos> app activate org.onosproject.fwd
```

See what happened in the ONOS GUI. Some devices showed up, right?

Try to ping hosts in the mininet terminal, and let it happening for a while.

```
mininet> h1 ping h2
```

Go back to the ONOS CLI, activate and deactivate the apps, see how it affects ping.

```
onos> app deactivate fwd
```

```
onos> app activate fwd
```

Ping all hosts in Mininet.

```
mininet> pingall
```

Some important ONOS commands in the CLI, go typing them one by one and see what happens.

ONOS has many CLI commands.

```
onos> help onos
```

ONOS has a convenient command to list the device currently known in the system. Running

```
onos> devices
```

Similarly, the links command is used to list the links detected by ONOS. At the ONOS prompt run

```
onos> links
```

And hosts too.

```
onos> hosts
```

The flows command allows you to observe which flow entries are currently registered in the system. Flow entries may be in several states:

- PENDING_ADD - The flow has been submitted and forwarded to the switch.
- ADDED - The flow has been added to the switch.
- PENDING_REMOVE - The request to remove the flow has been submitted and forwarded to the switch.
- REMOVED - The rule has been removed.

Keep mininet pinging:

```
mininet> h1 ping h2
```

Check the flows

```
onos> flows
```

In the shown output, ONOS provides many details about he the flows at the switches. For example each flow entry defines a selector and treatment which is the set of traffic matched by the the flow entry and how this traffic should be handled. Notice as well that each flow entry it tagged by an appId (application id), this appId identifies which application installed this flow entry. This is a useful feature because it can help an admin identify which application may be misbehaving or consuming many resources.

Given a network topology, ONOS computes all the shortest paths between any two nodes. This is especially useful for your applications to obtain path information for either flow installation or some other use. The paths command takes two arguments, both of them are devices. To make things easy for you ONOS provides CLI autocompletion by simply hitting the key.

Try it:

```
onos> paths <TAB>
onos> paths of:0000000000000001 of:0000000000000009
```

Part 3: ONOS Intents

The intent command allows one to see what intents are stored in the system. Intents can be in several states:

- SUBMITTED - The intent has been submitted and will be processed soon.
- COMPILING - The intent is being compiled. This is a transient state.
- INSTALLING - The intent is in the process of being installed.
- INSTALLED - The intent has been installed.
- RECOMPILING - The intent is being recompiled after a failure.
- WITHDRAWING - The intent is being withdrawn.
- WITHDRAWN - The intent has been removed.
- FAILED - The intent is in a failed state because it cannot be satisfied.

```
onos> intents
```

Note: You will not see any intents until some have been added. So, you will add explicit host connectivity intent.

The command can also tell you what type of sub-intents the intent has been compiled to:

```
onos> intents -i
```

First, let's deactivate the Reactive Forwarding application though:

```
onos> app deactivate fwd
```

Let's add a host connectivity intent for some two end-station hosts. You can use argument completion by pressing the Tab key.

```
onos> add-host-intent 00:00:00:00:00:01/None 00:00:00:00:00:03/None
```

```
onos> intents
```

Go to the ONOS GUI, in the side panel click in the Intents options, select the listed Intent and click in the graphic signal at the corner of the screen (show in topology).

If you still have the h1 ping h2 command running, you will see that the ICMP pings are still working beetween the two hosts.

So now that the intent is installed, let's have a look what path it is using by using the flows command with summarized output using the -s flag for better readability:

```
onos> flows -s
```

You can visualize the intent using ONOS GUI by selecting a node in the GUI and press the V key to show paths provisioned by intents that pass through the selected node.

To realize if your intent is actually working in ONOS, let's play with mininet, and check if the connectivity between h1 and h2 keeps existing.

```
mininet> link s3 s4 down
```

Check in the ONOS GUI how the path changed, and see its flows and the ping connectivity in mininet too.

To bring the link back

```
mininet> link s3 s4 up
```

To tear-down mininet and onos:

```
mininet> exit
$ sudo mn -c
```

```
$ docker stop -t0 onos
$ docker rm onos
```

Congratulations!

Now you know a little about ONOS!