



UNIVERSITE ABDELMALEK ESSAADI
FACULTE DES SCIENCES ET TECHNIQUES
DE TANGER



DEPARTEMENT GENIE INFORMATIQUE

Cycle Master: MBD & SIM (Semestre II)

Module: Cloud Computing & Virtualisation

Travaux Pratiques N°2: OpenDaylight MININET



Réalisé Par :

EL MSAOURI Tarik

MEDAGHRI ALAOUI Amine

Encadré Par :

Pr. BOUHORMA Mohammed

Année Universitaire 2020/2020

Mise en place d'une nouvelle topologie en utilisant une API

En vous servant des informations sur l'API python utilisée par Mininet <http://mininet.org/api/annotated.html>, il faut mettre en place l'architecture suivante :

- 4 commutateurs (S1,S2,S3,S4)
- 12 machines (h1 à h12)

Les connexions :

- S1 est directement connecté à s2 et s3 qui sont eux-mêmes connectés à s4.
- Les machines h1 à h3 sont connectés à S1, h4 à h6 à S2, h7 à h9 à s3 et les autres à s4.

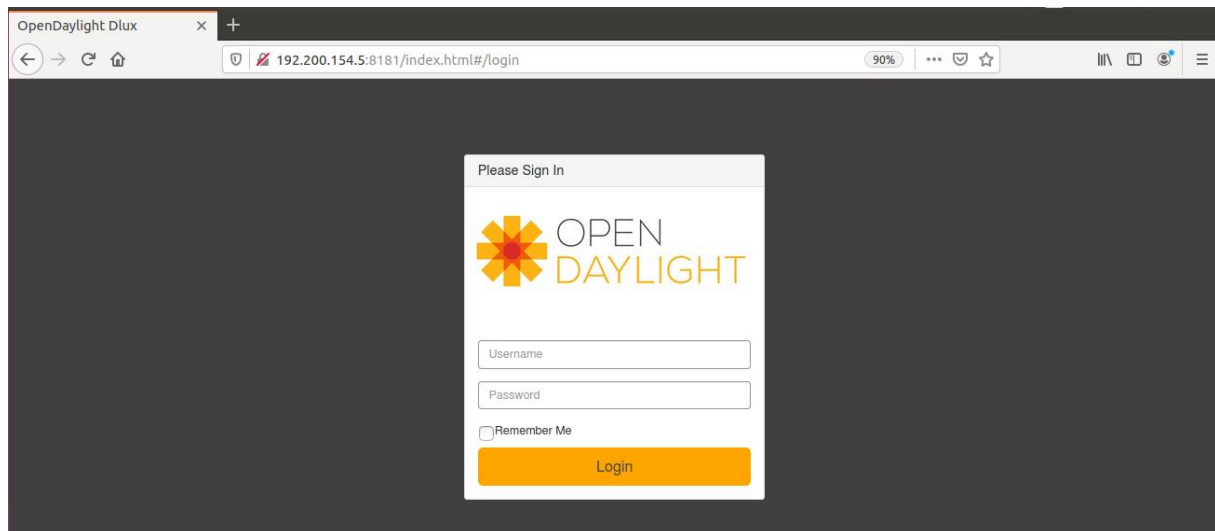
Etape 1

Nous avons démarré opendaylight avec la commande suivante :

```
sudo -E ./karaf
```

[illegible]

Maintenant, on peut accéder à l'URL et on se connecte à la console DLUX en utilisant les informations d'identification admin/admin. (<https://localhost:8181/index.html>)



On se connecte avec le user admin

Pour créer la topologie demandée on aura besoin d'un fichier python qui mettra en place l'architecture. Le fichier est nommé "tp2.py"

On commence par importer le package "topo"

```
from mininet.topo import Topo
```

Puis on définit les 12 machines de h1 à h12 avec la fonction "addHost()"

```
h1 = self.addHost( 'h1' )
h2 = self.addHost( 'h2' )
h3 = self.addHost( 'h3' )
h4 = self.addHost( 'h4' )
h5 = self.addHost( 'h5' )
h6 = self.addHost( 'h6' )
h7 = self.addHost( 'h7' )
h8 = self.addHost( 'h8' )
h9 = self.addHost( 'h9' )
h10 = self.addHost( 'h10' )
h11 = self.addHost( 'h11' )
h12 = self.addHost( 'h12' )
```

On ajoute les commutateurs S1, S2, S3 et S4

```
switch1 = self.addSwitch( 's1' )
switch2 = self.addSwitch( 's2' )
switch3 = self.addSwitch( 's3' )
switch4 = self.addSwitch( 's4' )
```

Et finalement on ajoute les liens entre les machines et les liens entre les commutateurs

```
# Add links
self.addLink( h1, switch1 )
self.addLink( h2, switch1 )
self.addLink( h3, switch1 )

self.addLink( h4, switch2 )
self.addLink( h5, switch2 )
self.addLink( h6, switch2 )

self.addLink( h7, switch3 )
self.addLink( h8, switch3 )
self.addLink( h9, switch3 )

self.addLink( h10, switch4 )
self.addLink( h11, switch4 )
self.addLink( h12, switch4 )

self.addLink( switch1, switch2 )
self.addLink( switch2, switch3 )
self.addLink( switch3, switch4 )
self.addLink( switch4, switch1 )
```

Dernièrement, on fait appelle à la fonction qui fera fonctionner notre fichier python.

```
topos = { 'tp2': ( lambda: Tp2() ) }
```

Pour lancer la topologie, on fait la commande suivante :

```
sudo mn -controller=remote,ip=192.200.154.5 -custom tp2.py -topo tp2
```

```
amine@tarik-amine:~$ nano tp2.py
amine@tarik-amine:~$ sudo mn --controller=remote,ip=192.200.154.5 --custom tp2.py --topo tp2
sudo: unable to resolve host tarik-amine: Connection refused
[sudo] password for aminegr:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s2) (h5, s2) (h6, s2) (h7, s3) (h8, s3) (h9, s3) (h10, s4) (h11, s4) (
h12, s4) (s1, s2) (s2, s3) (s3, s4) (s4, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
```

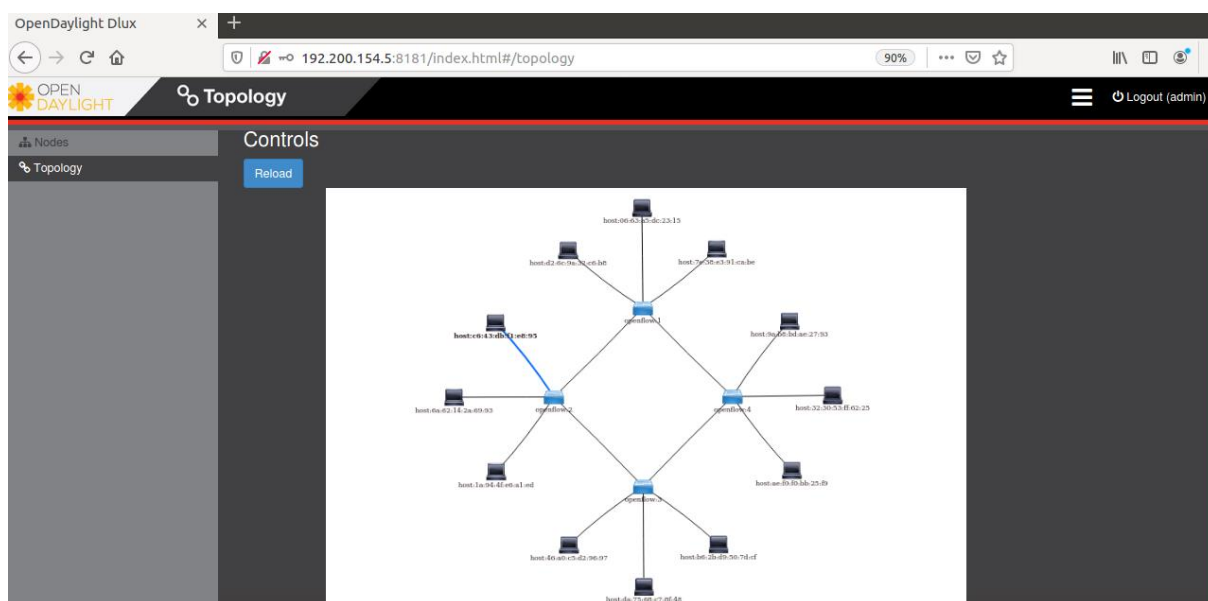
Voici un test de ping pour voir que tous les pings fonctionnent correctement

```

mininet> pingall
*** Ping: testing ping reachability
h1 -> X h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10 h11 h12
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10 h11 h12
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10 h11 h12
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10 h11 h12
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10 h11 h12
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10 h11 h12
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10 h11 h12
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h11 h12
h11 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h12
h12 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11
*** Results: 0% dropped (131/132 received)
mininet>

```

Puis, on retourne dans notre fenêtre et on la relance avec le bouton « Reload », et on obtiendra la topologie suivante



Voici la topologie obtenu, elle contient 4 commutateurs liés l'un a l'autre, avec 3 machines pour chacun d'eux