

ODL: Lab 1 Mise en oeuvre d'un contrôleur SDN

Objet	contrôleur SDN avec Opendaylight
Niveau requis	débutant, avisé
Features	odl-openflow-plugin
Débutant, à savoir	odl-overview
Suivi	:DONE:

Objectif

Ce didacticiel, permet d'apprendre à utiliser Floodlight conjointement avec OpenFlow.

À la fin de ce didacticiel, on doit être familiarisé avec les éléments suivants:

- Openflow
- opendaylight
- Création de flux OpenFlow
 - Matches OpenFlow
 - Actions OpenFlow

Description de la topologie

La topologie de test nécessite la mise en oeuvre:

- de deux hôtes, tous connectés l'un à l'autre via un seul commutateur activé par OpenFlow. Ceci est le plan de données.
- d'un commutateur connecté à un contrôleur. Ceci est le plan de contrôle.

A propos du contrôleur: Dans ce didacticiel, on utilise un contrôleur opendaylight.

Mise en oeuvre de la topologie

La topologie GNS3 est composée de 5 noeuds:

- **Noeud 1:** host 1
 - ifconfig eth0 10.0.0.1 netmask 255.255.255.0
 - route add default gateway 10.0.0.7
- **Noeud 2:** host 2
 - ifconfig eth0 20.0.0.2 netmask 255.255.255.0
 - route add default gateway 20.0.0.7
- **Noeud 3:** interface NAT pour accès au réseau via openvswitch et au contrôleur floodlight (réseau 192.168.122.0/24)
- **Noeud 4:** commutateur
 - vlan 10 : ip address 10.0.0.7 24

- vlan 20 : ip address 20.0.0.7 24
- vlan 212 : ip address 192.168.122.7 24
- ge1/0 : port access vlan 10 (host1)
- ge2/0 : port access vlan 20 (host2)
- ge3/0 : port access vlan 212 (NAT)

Configuration du contrôleur

Une fois la topologie installée et que toutes les ressources sont en place, lancer le contrôleur opendaylight à partir de la ressource contrôleur.

Se connecter sur l'UI opendaylight avec user/mdp 'karaf/karaf' :

```
$ ssh -p 8101 karaf@localhost
```

Les features [odl-openflow-plugin](#) doivent être installées :

Feature	Classe
odl-openflowjava-protocol	OpenDaylight::Openflow Java::Protocol
odl-openflowplugin-flow-services-ui	OpenDaylight::Openflow Plugin::Flow Services:
odl-openflowplugin-flow-services-rest	OpenDaylight::Openflow Plugin::Flow Services:
odl-openflowplugin-flow-services	OpenDaylight::Openflow Plugin::Flow Services
odl-openflowplugin-southbound	OpenDaylight::Openflow Plugin::Li southbound A
odl-openflowplugin-nsf-model	OpenDaylight::OpenflowPlugin::NSF::Model
odl-openflowplugin-app-config-pusher	OpenDaylight::Openflow Plugin::Application - d
odl-openflowplugin-app-topology	OpenDaylight::Openflow Plugin::Application - t
odl-openflowplugin-app-forwardingrules-manager	OpenDaylight::Openflow Plugin::Application - F

Configuration du commutateur

Maintenant que le contrôleur est configuré et fonctionne, il faut configurer le commutateur.

Configurer une instance

Pour configurer OpenFlow, dans la vue système, utiliser la commande `openflow` et spécifier une instance. Ceci amène aux options de configuration globales OpenFlow:

```
system-view
System View: return to User View with Ctrl+Z.
[HPE]openflow instance 1
```



Une instance OpenFlow doit être configurée. Sur les commutateurs ProVision, il existe un mappage un à un entre les instances et les VLAN. En d'autres termes, chaque VLAN nécessite une instance OpenFlow distincte. Ceci n'est pas vrai pour les commutateurs



Comware (plusieurs VLAN peuvent être mappés sur une seule instance OpenFlow).



Contrairement aux commutateurs 5900 prenant en charge 4094 instances d'OpenFlow, les commutateurs de la série 5500 ne prennent en charge que 8 instances.

Configurer le contrôleur

Un ID de contrôleur et une adresse IP doivent être spécifiés. On peut configurer plusieurs contrôleurs pour la redondance et l'équilibrage de la charge. Dans ce cas, un seul contrôleur est configuré.

```
[HPE-of-inst-1]controller 1 address ip 192.168.56.7
```

Associer les vlans

Les commutateurs hybrides OpenFlow prennent en charge le fonctionnement OpenFlow et le fonctionnement de commutation Ethernet normal. Certains VLAN utilisent la commutation Ethernet L2 traditionnelle, l'isolation de VLAN, le routage L3, le traitement des listes de contrôle d'accès et de qualité de service, etc. Ce type de commutateur doit fournir un mécanisme de classification qui achemine le trafic vers le pipeline OpenFlow ou le pipeline normal. Les commutateurs HP utilisent des VLAN (un ou plusieurs) à cette fin.

```
[5500-1-of-inst-1]classification vlan 10  
This command isn't effective until the active instance command is issued.  
[5500-1-of-inst-1]
```

Activer l'instance

La dernière étape consiste à activer l'instance:

```
[5500-1-of-inst-1] active instance
```

C'est tout. Il n'est pas trop difficile de configurer une seule instance de base d'OpenFlow sur les commutateurs HP ProVision

On doit voir le commutateur se connecter au contrôleur.

Pour s'assurer de pouvoir communiquer avec tout le monde:

- sur l'hôte 1 passer la commande:

```
ping 10.0.0.2
```

- sur l'hôte 2 passer la commande:

```
ping 10.0.0.1
```

A ce stade les hôtes ne peuvent pas communiquer entre eux, car la seule règle installée est la règle table MISS qui drop tous les flux qui ne correspondent à aucune règle.

Recherche des informations sur les périphériques connectés

Sur le commutateur récupérer le DPID de l'instance OPENFLOW

```
show openflow instance 1
```

Puis demander au contrôleur quels périphériques sont connectés. Cela donnera toutes les informations dont on aura besoin, par exemple. ports, adresses MAC, adresses IP. Pour cela avec postman faire un GET à l'url:

<http://x.x.x.x:8181/restconf/operational/opendaylight-inventory:nodes/node/openflow:#>

où # est le DPID valide du commutateur

Dans la sortie, on obtiendra une liste des périphériques que Floodlight a appris, récupérer les informations des vlans 10 et 20 et des ports GE1/0 (host1) et GE2/0 (host 2).

Par exemple pour élaborer ce lab, la sortie pour le VLAN 10 ressemble à ceci:

```
{
  "id": "openflow:506043239107447:404",
  "opendaylight-port-statistics:flow-capable-node-
connector-statistics": {
    "transmit-drops": 18446744073709551615,
    "receive-frame-error": 18446744073709551615,
    "receive-drops": 18446744073709551615,
    "receive-crc-error": 18446744073709551615,
    "bytes": {
      "transmitted": 18446744073709551615,
      "received": 18446744073709551615
    },
    "duration": {
      "second": 4719,
      "nanosecond": 4294967295
    },
    "receive-errors": 18446744073709551615,
```

```
        "transmit-errors": 18446744073709551615,  
        "receive-over-run-error": 18446744073709551615,  
        "collision-count": 18446744073709551615,  
        "packets": {  
            "transmitted": 18446744073709551615,  
            "received": 18446744073709551615  
        }  
    },  
    "flow-node-inventory:peer-features": "",  
    "flow-node-inventory:advertised-features": "",  
    "flow-node-inventory:port-number": "404",  
    "flow-node-inventory:hardware-address":  
"cc:3e:5f:82:0b:77",  
    "flow-node-inventory:supported": "",  
    "flow-node-inventory:current-speed": 0,  
    "flow-node-inventory:current-feature": "",  
    "flow-node-inventory:state": {  
        "live": true,  
        "link-down": false,  
        "blocked": false  
    },  
    "flow-node-inventory:maximum-speed": 0,  
    "flow-node-inventory:name": "Vlan10",  
    "flow-node-inventory:configuration": ""  
},
```

Les informations importantes à récupérer sont le flow-node-inventory:name (le nom canonique de l'interface), l'id et le flow-node-inventory:hardware-address (l'adresse MAC).

Les valeurs collectées en sortie de liste sont présentées dans le tableau suivant :

flow-node-inventory:name	id	flow-node-inventory:hardware-address
Vlan10	openflow:506043239107447:405	cc:3e:5f:82:0b:77
Vlan20	openflow:506043239107447:406	cc:3e:5f:82:0b:77
GE1/0	openflow:506043239107447:17	0c:8b:f6:6a:0b:00
GE2/0	openflow:506043239107447:33	0c:8b:f6:6a:0b:01

Fabrication des flux statiques

Afin de faire communiquer les hôtes installer deux règles supplémentaires sur ODL sur la table-0 (à l'aide de son API REST et de Postman), comme suit:

flux statique de host1 vers host2

Etape 1: Indiquer l'id du switch

La première chose à faire est d'indiquer à opendaylight le commutateur où on veut insérer ce flux.

Pour ce faire, il faut indiquer le DatapathID (DPID) du commutateur dans le champ "commutateur" de l'objet JSON.

On peut obtenir ce DPID en entrant la commande suivante sur le commutateur:

```
show openflow instance 1
```

Dans la sortie du commutateur le DPID est retourné au format hexa, pour l'insérer dans l'objet JSON, il faut le saisir au format numérique'

Etape 2: Indiquer un nom

La prochaine chose à faire est de donner un nom au flux. Ce nom doit être unique, Il ne peut y avoir d'autre flux dans la topologie avec le même nom. On va simplement nommer ce premier flux «flux-1» dans le champ "name" de notre objet JSON.

Etape 3: Indiqué une priorité et activer

Il faut donner à nos flux une très haute priorité afin de s'assurer qu'ils sont toujours en première ligne! on indiquera '32768' dans le champ "priority".

Le prochain paramètre est assez facile. Il faut décider si on veut que le flux soit actif ou non. Pour cela simplement définir le champ "actif" sur "true". Cela garantira que le flux fait réellement son travail.

Etape 4: Fabriquer les correspondances (Match Maker)

Il faut maintenant définir les correspondances ou, en d'autres termes, le type de paquets sur lequel on veut effectuer les actions. C'est une bonne pratique d'être aussi précis que possible.

Les champs matchs à faire correspondre sont :

- **eth_type**: doit correspondre à un EtherType spécifique, qui dans notre cas est IPv4. Pour ce faire, utiliser la valeur '2048' (ou '0x0800' en hexadécimal) .
- **ipv4_src**: doit correspondre à l'adresse IP l'adresse IP de la source (l'hôte 1: 10.0.0.1).

Voilà pour les matchs! Maintenant qu'on a déterminé la partie matchs des paquets IPv4 dans la direction de l'hôte 1 vers l'hôte 2, il faut créer des actions pour manipuler ces paquets ...

Etape 5: Fabriquer les actions

L'objectif du tuto est de permettre les paquets, de l'hôte 1 destiné à l'hôte 2. Il faut donc spécifier le port sur lequel les paquets doivent sortir. Récupérer dans la liste des périphériques l'information du port relatif au vlan de l'hôte 2 (openflow:506043239107447:406).

Pour cela on a juste besoin de définir des actions pour:

- indiquer au paquet sur quel port sortir, (le vlan de l'hôte 2) dans champ **output**, qui devra toujours être la dernière entrée de la liste d'actions.

Etape 6: Assembler l'objet json

L'objet json devrait ressembler à quelque chose comme ceci

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
<strict>false</strict>
<flow-name>host1-host2</flow-name>
<id>260</id>
<cookie_mask>255</cookie_mask>
<cookie>105</cookie>
<table_id>0</table_id>
<priority>1</priority>
<hard-timeout>0</hard-timeout>
<idle-timeout>0</idle-timeout>
<match>
<ethernet-match>
<ethernet-type>
<type>2048</type>
</ethernet-type>
</ethernet-match>
<ipv4-source>10.0.0.1/32</ipv4-source>
</match>
<instructions>
<instruction>
<order>0</order>
<apply-actions>
<action>
<order>0</order>
<output-action>
<output-node-connector>openflow:506043239107447:406</output-node-connector>
<max-length>65535</max-length>
</output-action>
</action>
</apply-actions>
</instruction>
</instructions>
</flow>
```

Etape 7: Soumettre l'objet json

Méthode	PUT
URI	/restconf/config/opendaylight-inventory:nodes/node/openflow:<swirchId>/table/<tableId>/flow/<flowId>

flux statique de host2 vers host1

Etape 1: Indiquer l'id du switch

Le commutateur DPID est identique au premier flux (il n'y a qu'un commutateur)

Etape 2: Indiquer un nom

On lui donnera le nom "flux-2".

Etape 3: Indiqué une priorité et activer

On lui donnera la priorité sera "32768" et on mettra active à "true"

Etape 4: Fabriquer les correspondances (Match Maker)

Les champs matchs à faire correspondre sont :

- **eth_type**: doit correspondre à un EtherType spécifique, qui dans notre cas est IPv4. Pour ce faire, utiliser la valeur '2048' (ou '0x0800' en hexadécimal) .
- **ipv4_src**: doit correspondre à l'adresse IP l'adresse IP de la source (l'hôte 2: 20.0.0.2).

Voilà pour les matchs! Maintenant qu'on a déterminé la partie matchs des paquets IPv4 dans la direction de l'hôte 2 vers l'hôte 1, il faut créer des actions pour manipuler ces paquets ...

Etape 5: Fabriquer les actions

Récupérer dans la liste des périphériques l'information du port relatif au vlan de l'hôte 1 (openflow:506043239107447:404).

Pour cela on a juste besoin de définir des actions pour:

- indiquer au paquet sur quel port sortir, (le vlan de l'hôte 1) dans champ **output**, qui devra toujours être la dernière entrée de la liste d'actions.

Etape 6: Assembler l'objet json

L'objet json devrait ressembler à quelque chose comme ceci

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<flow xmlns="urn:opendaylight:flow:inventory">
<strict>false</strict>
<flow-name>host2-host1</flow-name>
```

```

<id>261</id>
<cookie_mask>255</cookie_mask>
<cookie>105</cookie>
<table_id>0</table_id>
<priority>1</priority>
<hard-timeout>0</hard-timeout>
<idle-timeout>0</idle-timeout>
<match>
  <ethernet-match>
    <ethernet-type>
      <type>2048</type>
    </ethernet-type>
  </ethernet-match>
  <ipv4-source>20.0.0.2/32</ipv4-source>
</match>
<instructions>
  <instruction>
    <order>0</order>
    <apply-actions>
      <action>
        <order>0</order>
        <output-action>
          <output-node-connector>openflow:506043239107447:405</output-node-connector>
          <max-length>65535</max-length>
        </output-action>
      </action>
    </apply-actions>
  </instruction>
</instructions>
</flow>

```

Etape 7: Soumettre l'objet json

Méthode	PUT
URI	/restconf/config/opendaylight-inventory:nodes/node/openflow:<swirchId>/table/<tableId>/flow/<flowId>

Validation du lab

Suite à l'ajoute des flux bi directionnels, les hôtes doivent pouvoir communiquer les uns avec les autres, pour s'en assurer:

- sur l'hôte 1 passer la commande:

```
ping 20.0.0.2
```

- sur l'hôte 2 passer la commande:

```
ping 10.0.0.1
```

Les deux devraient fonctionner.

From:

<http://vps101364.serveur-vps.net/> - **dwndoc**

Permanent link:

<http://vps101364.serveur-vps.net/doku.php?id=labs:odl-lab-1-openflow>

Last update: **2025/02/19 10:59**

