菜鸟水平初步设置 OpenDaylight OVSDB + Openstack 测试环境

Hannibal （SDNAP 首发）

刚接触 SDN 和 OpenDaylight 两个多月时间，还处于人云亦云照葫芦画瓢的水平，在很多大牛的指导文章帮助下，初步搭建一个很简单的 OpenDaylight OVSDB + Openstack 调试环境。第一次写技术文章，请多包涵。

一、准备

硬件：

双核 Core i7，内存 4GB，一个以太网卡的 Thinkpad X201t，普通个人用笔记本

Host 环境：

64 位 Ubuntu 13.10，OVS 2.0.90

VM 环境：

2 个 Virtualbox VM，Fedora 19 + OVS 2.0.0 + Devstack 。导入 Virtualbox 都是缺省配置。两个 VM 的下载地址：

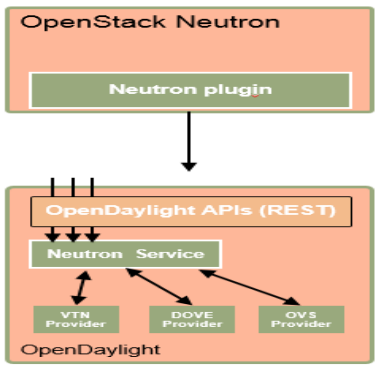
https://region-a.geo-1.objects.hpcloudsvc.com:443/v1/96991703573236/imgs/Fedora19--2node-Devstack.tar.bz2

Size: 4983728003 bytes
MD5sum: dfd791a989603a88a0fa37950696608c

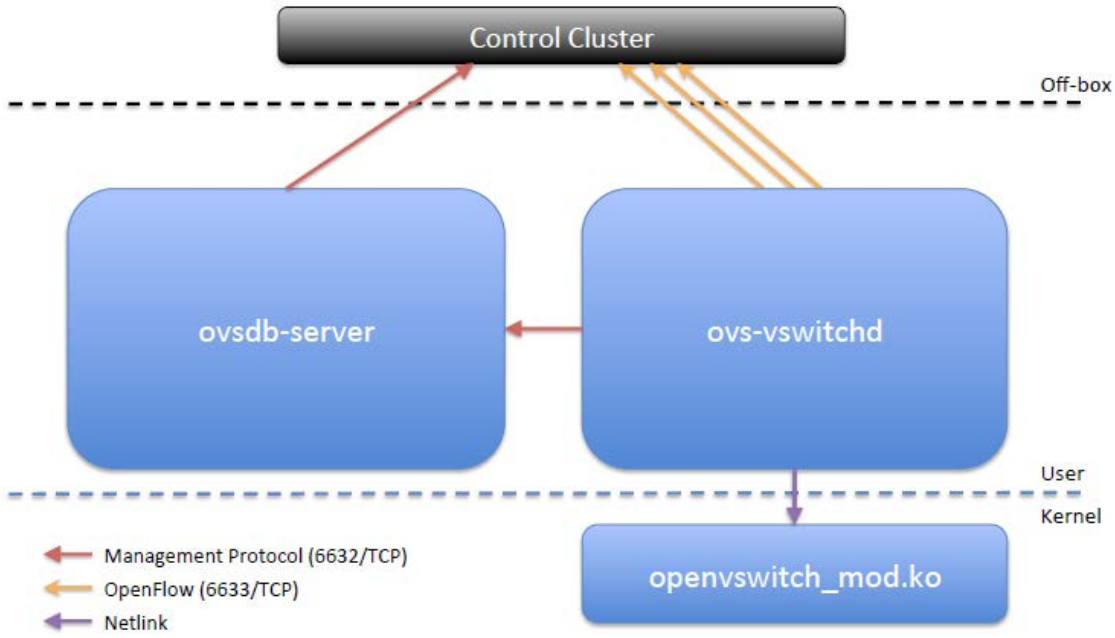
二、原理

OpenDaylight（ODL）是一个由 Linux 基金会支持，多个网络厂商参与的开源 SDN 控制器项目。Openstack 是开源的 IaaS 项目。如何让两个平台整合以便更好的发挥作用是本环境搭建的目的。

现有的解决方案之一，就是利用 Openstack Neutron 的 ML2 Plugin，将网络复杂性丢到 ODL。也就是说，Openstack 通过 ML2 Plugin，与 OpenDaylight 的 NB API 进行会话，具体网络部署的实现交由 OpenDaylight Controller 来实现。

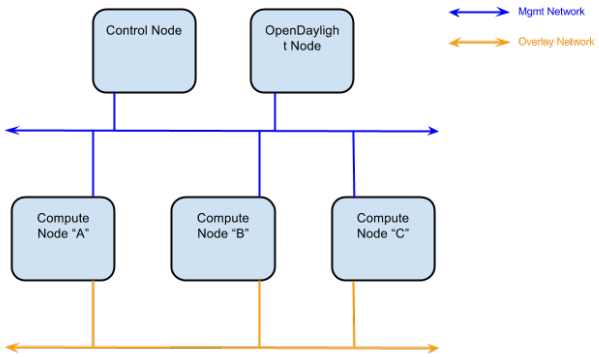


该方案的实现场景之一，便是 OVSDB 协议的简单应用。OVSDB 是 Openvswitch 的数据库管理协议



有关 Openvswitch 的详细内容请参见 Openvswitch 官网的[内容]

本环境的逻辑图如下



简单环境可以由一个 Openstack Control Node，一个 Openstack Compute Node 和一个 OpenDaylight Node 组成。

三、环境搭建

1. 安装 OpenDaylight Controller 和 OpenDaylight OVSDB

控制器的安装的具体事项请参考 wiki：
https://wiki.opendaylight.org/view/OpenDaylight_Controller:Pulling,_Hacking,_and_Pushing_the_Code_from_the_CLI

OpenDaylight 项目开发的 IDE 可以选择 Eclipse 或者 Intellij IDEA，个人推荐 Intellij IDEA Community Version，在完成上述步骤后直接导入即可。Eclipse 的步骤在 wiki 中可以找到，非常复杂且 bug 繁多。

OVSDB 的安装（在和 controller 的同级目录下）：

```
git clone https://git.opendaylight.org/gerrit/p/ovsdb.git

cd ovsdb
```

然后写一个 build_ovsdb.sh 的脚本，在当前目录下运行即可：

```
#!/bin/sh

git pull

cd neutron

echo "Refreshing ovsdb/neutron.."

pwd

mvn clean install

cd ../northbound/ovsdb/

echo "Refreshing northbound/ovsdb.. "

pwd

mvn clean install

cd ../../ovsdb

echo "Refreshing ovsdb/ovsdb.."

pwd
```

```
mvn clean install

cd ..

cp ~/odl/ovsdb/neutron/target/ovsdb.neutron-0.5.0-SNAPSHOT.jar
~/odl/controller/opendaylight/distribution/opendaylight/target/distribution.ope
ndaylight-osgipackage/opendaylight/plugins/

cp ~/odl/ovsdb/northbound/ovsdb/target/ovsdb.northbound-0.5.0-SNAPSHOT.jar
~/odl/controller/opendaylight/distribution/opendaylight/target/distribution.ope
ndaylight-osgipackage/opendaylight/plugins/

cp ~/odl/ovsdb/ovsdb/target/ovsdb-0.5.0-SNAPSHOT.jar
~/odl/controller/opendaylight/distribution/opendaylight/target/distribution.ope
ndaylight-osgipackage/opendaylight/plugins/

echo "done!"
```

2. 运行 OpenDaylight Controller

```
cd
~/odl/controller/opendaylight/distribution/opendaylight/target/distribution.ope
ndaylight-osgipackage/opendaylight

./run.sh
```

需要注意的是我们需要在 OSGI 命令行中停掉 simpleforward bundle

```
osgi> lb | grep simple

 132|Active | 4|samples.simpleforwarding (0.4.1.SNAPSHOT)

 true

 osgi> stop 132

 osgi> lb | grep simple

 132|Resolved | 4|samples.simpleforwarding (0.4.1.SNAPSHOT)

 true
```

3. 准备 Openstack 的两个 Node

在上面提供的下载链接下载并解压之后，里面提供了 control node 和 compute node 的 local.conf.control 和 local.conf.compute，但需要做出相应的修改，HOST IP 要改成当前 Node 的 ip 地址，Service ip 要改成 Control Node 的地址，在最下面[ml2_odl]中的 url 要填写 OpenDaylight Controller 所在的地址。以我的设置举例，对于 Control Node，HOST IP=128.195.169.202，SERVICE_HOST=128.195.169.202，最下面的 url=http://128.195.169.117:8080….对于 Compute Node，HOST IP=128.195.169.192，SERVICE_HOST=128.195.169.202,url 的配置与 Control Node 相同。

另外注意在 Control Node 的 local.conf.control 里面还要加上 disable_service n-cpu

修改好两个 conf 文件后，在 virtualbox 里启动 vm，如果不能启动图形界面的话，直接右侧 Ctrl + F2 进入命令行界面，用户名：fedora，密码：reverse。先配置 Control Node，将 local.conf.control 改名外 local.conf，然后运行同样是压缩包里的 fedora19-devstack.sh 文件：./fedora19-devstack “local.conf”，该脚本的功能在压缩包里的 Readme 文件中有叙述。之后 cd /mnt/git/devstack 然后./stack.sh。顺利的话会一次成功，如果出现错误的话，先运行./unstack.sh，改正后再次运行./stack.sh。Compute Node 的配置基本重复上面的步骤，除了是将 local.conf.compute 改成 local.conf。

在两个 Node 都 devstack 安装完毕之后，可以在 Compute Node 运行 Generic—post-Devstack.sh 来创建一个 instance。注意 Readme 文件中所提到的，脚本最后注释掉了 nova boot 的命令，是因为 net-id 的选项需要倒数第二个 neutron net-list 所给出的 private network id

./fedora19-devstack.sh “local.conf”适用于第一次配置，再次启动 VM 重跑 devstack 的时候，只要在启动 VM 后记得启动 OVS：sudo /usr/share/openvswitch/script/ovs-ctl start 就好。之后直接 cd /mnt/git/devstack 后./stack.sh 即可。

4. 后续

到了这里，打开浏览器地址栏输入控制器地址:8080，然后用户名 admin 密码 admin 即可进入 OpenDaylight 的 UI 界面，界面里应该显示有两个 VM 中 OVS 所创造的虚拟接口以及相应的流。在 VM 的 CLI 里运行 sudo ovs-vsctl show 可以看到接口的具体信息 。下面是我的 Compute Node 以及 Control Node 的 OVS 信息，其中可以看到 6640 是 OVSDB 端口，6633 是 controller 监听端口。具体 OVSDB 调试可以参见：http://networkstatic.net/getting-started-ovsdb/


[fedora@fedora19-compute devstack]$ sudo ovs-vsctl show
3cc9dac3-9fa6-4c69-acc1-a2d463396fc8
Manager "tcp:128.195.169.177:6640"
is_connected: true
Bridge br-int
Controller "tcp:128.195.169.177:6633"
is_connected: true
Port "tapce0263f5-ca"
Interface "tapce0263f5-ca"
Port "tapee14668f-ba"
tag: 1
Interface "tapee14668f-ba"
Port br-int
Interface br-int
Port patch-tun
Interface patch-tun
type: patch
options: {peer=patch-int}
Port "tap7128389f-d3"
Interface "tap7128389f-d3"
Port "tapb76b9c94-28"
tag: 1
Interface "tapb76b9c94-28"
Bridge br-tun
Controller "tcp:128.195.169.177:6633"
is_connected: true
Port patch-int
Interface patch-int
type: patch
options: {peer=patch-tun}
Port br-tun
Interface br-tun
Port "vxlan-1001-128.195.169.202"
Interface "vxlan-1001-128.195.169.202"
type: vxlan
options: {key="1001", local_ip="128.195.169.177", remote_ip="128
.195.169.202"}
ovs_version: "2.0.0"
@JJ_nopainkiller
weibo.com/nopainkillerjr

四、总结

目前的问题在于如果 linked clone compute node 来创建第二个 compute node vm 的话，虽然有不同的 mac 地址，但是不知何原因 OpenDaylight 的 Forwarding Rule Manager 会进入一个死循环，不断的用第二个节点来代替第一个节点，然后再用第一个节点来代替第二个节点。另外在两个 openstack node vm 刚启动 ovs 之后和 devstack 之前，osgi 的 cli 会给出一些 ovsdb bundle 提出的错误。这些问题都需要进一步的修正。

总而言之一个比较简单的 OpenDaylight OVSDB + OpenStack 的环境是可以用以上方法在普通电脑上搭建成功的，欢迎大家进行更复杂的调试。

五、参考链接：http://www.siliconloons.com/?p=523