

概念学习

Port

normal

internal

patch

interface

flow管理

OVS工具使用及区别

ovs-vsctl

ovsdb-tool

ovsdb-client

ovs-ofctl

概念学习

Port

端口Port与物理交换机的端口概念类似，Port是OVS Bridge上创建的一个虚拟端口，每个Port都隶属于一个Bridge。Port有以下几种类型

normal

可以把操作系统中已有的网卡(物理网卡em1/eth0,或虚拟机的虚拟网卡tapxxx)挂载到ovs上，ovs会生成一个同名Port处理这块网卡进出的数据包。此时端口类型为Normal。

如下，主机中有一块物理网卡eth1，把其挂载到OVS网桥br-ext上，OVS会自动创建同名Port eth1。

```
ovs-vsctl add-port br-ext eth1
```

```
#Bridge br-ext中出现Port "eth1"
```

有一点要注意的是，挂载到OVS上的网卡设备不支持分配IP地址，因此若之前eth1配置有IP地址，挂载到OVS之后IP地址将不可访问。这里的网卡设备不只包括物理网卡，也包括主机上创建的虚拟网卡

internal

Internal类型是OVS内部创建的虚拟网卡接口，每创建一个Port，OVS会自动创建一个同名接口(Interface)挂载到新创建的Port上。接口的概念下面会提到。

下面创建一个网桥br0，并创建一个Internal类型的Port p0

```

1 ovs-vsctl add-br br0
2 ovs-vsctl add-port br0 p0 -- set Interface p0 type=internal
3
4 查看网桥br0
5 ovs-vsctl show br0
6     Bridge "br0"
7         fail_mode: secure
8         Port "p0"
9             Interface "p0"
10                type: internal
11         Port "br0"
12             Interface "br0"
13                type: internal

```

可以看到有两个Port。当ovs创建一个新网桥时，默认会创建一个与网桥同名的Internal Port。在OVS中，只有“internal”类型的设备才支持配置IP地址信息，因此我们可以为br0接口配置一个IP地址，当然p0也可以配置IP地址

```

1 ip addr add 192.168.10.11/24 dev br0
2 ip link set br0 up
3 #添加默认路由
4 ip route add default via 192.168.10.1 dev br0

```

上面两种Port类型区别在于，Internal类型会自动创建接口(Interface)，而Normal类型是把主机中已有的网卡接口添加到OVS中

patch

当主机中有多个ovs网桥时，可以使用Patch Port把两个网桥连起来。Patch Port总是成对出现，分别连接在两个网桥上，从一个Patch Port收到的数据包会被转发到另一个Patch Port，类似于Linux系统中的veth。使用Patch连接的两个网桥跟一个网桥没什么区别，OpenStack Neutron中使用到了Patch Port。上面网桥br-ext中的Port phy-br-ext与br-int中的Port int-br-ext是一对Patch Port

```

1 ovs-vsctl add-br br0
2 ovs-vsctl add-br br1
3 ovs-vsctl \
4 -- add-port br0 patch0 -- set interface patch0 type=patch
   options:peer=patch1 \
5 -- add-port br1 patch1 -- set interface patch1 type=patch
   options:peer=patch0
6
7 #结果如下
8 #ovs-vsctl show
9     Bridge "br0"
10         Port "br0"
11             Interface "br0"
12                type: internal
13         Port "patch0"
14             Interface "patch0"
15                type: patch
16                options: {peer="patch1"}
17     Bridge "br1"
18         Port "br1"
19             Interface "br1"
20                type: internal

```

```

21     Port "patch1"
22         Interface "patch1"
23             type: patch
24             options: {peer="patch0"}

```

interface

在通常情况下，Port和Interface是一对一的关系，只有在配置Port为 bond模式后，Port和Interface是一对多的关系。这个网络接口设备可能是创建Internal类型Port时OVS自动生成的虚拟网卡，也可能是系统的物理网卡或虚拟网卡(TUN/TAP)挂载在ovs上。OVS中只有“Internal”类型的网卡接口才支持配置IP地址

Interface是一块网络接口设备，负责接收或发送数据包，Port是OVS网桥上建立的一个虚拟端口，Interface挂载在Port上。

flow管理

介绍下上面三种flows管理工具，不具体说明，具体使用可以查看相关man手册。

- ovs-ofctl dump-flows
打印指定网桥内的所有OpenFlow flows，可以存在多个流表(flow tables)，按表顺序显示。不包括“hidden” flows。这是最常用的查看flows命令，当然这条命令对所有OpenFlow交换机都有效，不单单是OVS。
- ovs-appctl bridge/dump-flows
打印指定网桥内所有OpenFlow flows，包括“hidden” flows，in-band control模式下排错可以用到。
- ovs-dpctl dump-flows [dp] 打印内核模块中datapath flows，[dp]可以省略，默认主机中只有一个datapath system@ovs-system,man手册可以找到非常详细的用法说明，注意ovs-ofctl管理的是OpenFlow flows。

OVS工具使用及区别

ovs-vsctl

ovs-vsctl是一个管理或配置ovs-vswitchd的高级命令行工具，高级是说其操作对用户友好，封装了对数据库的操作细节。它是管理OVS最常用的命令，除了配置flows之外，其它大部分操作比如Bridge/Port/Interface/Controller/Database/Vlan等都可以完成。

```

1  #添加网桥br0
2  ovs-vsctl add-br br0
3  #列出所有网桥
4  ovs-vsctl list-br
5  #添加一个Port p1到网桥br0
6  ovs-vsctl add-port br0 p1
7  #查看网桥br0上所有Port
8  ovs-vsctl list-ports br0
9  #获取br0网桥的OpenFlow控制器地址，没有控制器则返回空
10 ovs-vsctl get-controller br0
11 #设置OpenFlow控制器,控制器地址为192.168.1.10,端口为6633
12 ovs-vsctl set-controller br0 tcp:192.168.1.10:6633
13 #移除controller
14 ovs-vsctl del-controller br0
15 #删除网桥br0
16 ovs-vsctl del-br br0
17 #设置端口p1的vlan tag为100

```

```

18 ovs-vsctl set Port p1 tag=100
19 #设置Port p0类型为internal
20 ovs-vsctl set Interface p0 type=internal
21 #添加vlan10端口, 并设置vlan tag为10, Port类型为Internal
22 ovs-vsctl add-port br0 vlan10 tag=10 -- set Interface vlan10 type=internal
23 #添加隧道端口gre0, 类型为gre, 远端IP为1.2.3.4
24 ovs-vsctl add-port br0 gre0 -- set Interface gre0 type=gre
    options:remote_ip=1.2.3.4
25

```

ovsdb-tool

ovsdb-tool是一个专门管理OVS数据库文件的工具, 不常用, 它不直接与ovsdb-server进程通信.

```

1  #可以使用此工具创建并初始化database文件
2  ovsdb-tool create [db] [schema]
3  #可以使用ovsdb-client get-schema [database]获取某个数据库的schema(json格式)
4  #可以查看数据库更改记录, 具体到操作命令, 这个比较有用
5  ovsdb-tool show-log -m
6  record 48: 2017-01-07 03:34:15.147 "ovs-vsctl: ovs-vsctl --timeout=5 -- --
    if-exists del-port tapcea211ae-10"
7      table Interface row "tapcea211ae-10" (151f66b6):
8          delete row
9      table Port row "tapcea211ae-10" (cc9898cd):
10         delete row
11      table Bridge row "br-int" (fddd5e27):
12      table Open_vSwitch row a9fc1666 (a9fc1666):
13
14  record 49: 2017-01-07 04:18:23.671 "ovs-vsctl: ovs-vsctl --timeout=5 -- --
    if-exists del-port tap5b4345ea-d5 -- add-port br-int tap5b4345ea-d5 -- set
    Interface tap5b4345ea-d5 "external-ids:attached-mac=\"fa:16:3e:50:1b:5b\""
    -- set Interface tap5b4345ea-d5 "external-ids:iface-id=\"5b4345ea-d5ea-
    4285-be99-0e4cadf1600a\"" -- set Interface tap5b4345ea-d5 "external-ids:vm-
    id=\"0aa2d71e-9b41-4c88-9038-e4d042b6502a\"" -- set Interface tap5b4345ea-
    d5 external-ids:iface-status=active"
15      table Port insert row "tap5b4345ea-d5" (4befd532):
16      table Interface insert row "tap5b4345ea-d5" (b8a5e830):
17      table Bridge row "br-int" (fddd5e27):
18      table Open_vSwitch row a9fc1666 (a9fc1666):
19  ...

```

ovsdb-client

ovsdb-client是ovsdb-server进程的命令行工具, 主要是从正在运行的ovsdb-server中查询信息, 操作的是数据库相关.

```

1 #列出主机上的所有databases，默认只有一个库Open_vSwitch
2 ovsdb-client list-dbs
3 #获取指定数据库的schema信息
4 ovsdb-client get-schema [DATABASE]
5 #列出指定数据库的所有表
6 ovsdb-client list-tables [DATABASE]
7 #dump指定数据库所有数据,默认dump所有table数据，如果指定table，只dump指定table数据
8 ovsdb-client dump [DATABASE] [TABLE]
9 #监控指定数据库中的指定表记录改变
10 ovsdb-client monitor DATABASE TABLE

```

ovs-ofctl

ovs-ofctl是专门管理配置OpenFlow交换机的命令行工具，我们可以用它手动配置OVS中的OpenFlow flows，注意其不能操作datapath flows和“hidden” flows.

```

1 #查看br-tun中OpenFlow flows
2 ovs-ofctl dump-flows br-tun
3 #查看br-tun端口信息
4 ovs-ofctl show br-tun
5 #添加新的flow: 对于从端口p0进入交换机的数据包，如果它不包含任何VLAN tag，则自动为它添加
  VLAN tag 101
6 ovs-ofctl add-flow br0
  "priority=3,in_port=100,dl_vlan=0xffff,actions=mod_vlan_vid:101,normal"
7 #对于从端口3进入的数据包，若其vlan tag为100，去掉其vlan tag，并从端口1发出
8 ovs-ofctl add-flow br0 in_port=3,dl_vlan=101,actions=strip_vlan,output:1
9 #添加新的flow: 修改从端口p1收到的数据包的源地址为9.181.137.1,show 查看p1端口ID为100
10 ovs-ofctl add-flow br0 "priority=1
  idle_timeout=0,in_port=100,actions=mod_nw_src:9.181.137.1,normal"
11 #添加新的flow: 重定向所有的ICMP数据包到端口 p2
12 ovs-ofctl add-flow br0
  idle_timeout=0,dl_type=0x0800,nw_proto=1,actions=output:102
13 #删除编号为 100 的端口上的所有流表项
14 ovs-ofctl del-flows br0 "in_port=100"

```