# OVS常用命令与使用总结

说明

在平时使用ovs中，经常用到的ovs命令，参数，与举例总结，持续更新中…

## 进程启动

### 1.先准备ovs的工作目录，数据库存储路径等

mkdir -p /etc/openvswitch

mkdir -p /var/run/openvswitch

1

2

### 2.先启动ovsdb-server

ovsdb-server /etc/openvswitch/conf.db \

-vconsole:emer -vsyslog:err -vfile:info \

--remote=punix:/var/run/openvswitch/db.sock \

--private-key=db:Open\_vSwitch,SSL,private\_key \

--certificate=db:Open\_vSwitch,SSL,certificate \

--bootstrap-ca-cert=db:Open\_vSwitch,SSL,ca\_cert --no-chdir \

--log-file=/var/log/openvswitch/ovsdb-server.log \

--pidfile=/var/run/openvswitch/ovsdb-server.pid \

--detach --monitor

ps: 如果想清除配置，可以先删除/etc/openvswitch/\*，然后再手动创建一个数据库：

ovsdb-tool create /etc/openvswitch/conf.db /usr/share/openvswitch/vswitch.ovsschema

1

### 3.初始化ovsdb

针对于新创建的数据库才需要初始化

ovs-vsctl --no-wait init

1

### 4.启动vswitchd进程

ovs-vswitchd unix:/var/run/openvswitch/db.sock \

-vconsole:emer -vsyslog:err -vfile:info --mlockall --no-chdir \

--log-file=/var/log/openvswitch/ovs-vswitchd.log \

--pidfile=/var/run/openvswitch/ovs-vswitchd.pid \

--detach --monitor

## ovs常用命令

### 控制管理类

#### 1.查看网桥和端口

ovs-vsctl show

#### 2.创建一个网桥

ovs-vsctl add-br br0

ovs-vsctl set bridge br0 datapath\_type=netdev

1

2

#### 3.添加/删除一个端口

# for system interfaces

ovs-vsctl add-port br0 eth1

ovs-vsctl del-port br0 eth1

# for DPDK

ovs-vsctl add-port br0 dpdk1 -- set interface dpdk1 type=dpdk options:dpdk-devargs=0000:01:00.0

# for DPDK bonds

ovs-vsctl add-bond br0 dpdkbond0 dpdk1 dpdk2 \

-- set interface dpdk1 type=dpdk options:dpdk-devargs=0000:01:00.0 \

-- set interface dpdk2 type=dpdk options:dpdk-devargs=0000:02:00.0

# or new version

ovs-vsctl add-port br0 dpdkbond0 \

-- set interface dpdkbond0 type=dpdk options:dpdk-devargs=0000:01:00.0,0000:02:00.0

#### 4.设置/清除网桥的openflow协议版本

ovs-vsctl set bridge br0 protocols=OpenFlow13

ovs-vsctl clear bridge br0 protocols

#### 5.查看某网桥当前流表

ovs-ofctl dump-flows br0

ovs-ofctl -O OpenFlow13 dump-flows br0

ovs-appctl bridge/dump-flows br0

#### 6.设置/删除控制器

ovs-vsctl set-controller br0 tcp:1.2.3.4:6633

ovs-vsctl del-controller br0

#### 7.查看控制器列表

ovs-vsctl list controller

#### 8.设置/删除被动连接控制器

ovs-vsctl set-manager tcp:1.2.3.4:6640

ovs-vsctl get-manager

ovs-vsctl del-manager

#### 9.设置/移除可选选项

ovs-vsctl set Interface eth0 options:link\_speed=1G

ovs-vsctl remove Interface eth0 options link\_speed

#### 10.设置fail模式，支持standalone或者secure

standalone(default)：清除所有控制器下发的流表，ovs自己接管

secure：按照原来流表继续转发

ovs-vsctl del-fail-mode br0

ovs-vsctl set-fail-mode br0 secure

ovs-vsctl get-fail-mode br0

#### 11.查看接口id等

ovs-appctl dpif/show

1

#### 12.查看接口统计

ovs-ofctl dump-ports br0

### 流表类

#### 流表操作

##### 1.添加普通流表

ovs-ofctl add-flow br0 in\_port=1,actions=output:2

##### 2.删除所有流表

ovs-ofctl del-flows br0

##### 3.按匹配项来删除流表

ovs-ofctl del-flows br0 "in\_port=1"

1

#### 匹配项

##### 1.匹配vlan tag，范围为0-4095

ovs-ofctl add-flow br0 priority=401,in\_port=1,dl\_vlan=777,actions=output:2

1

##### 2.匹配vlan pcp，范围为0-7

ovs-ofctl add-flow br0 priority=401,in\_port=1,dl\_vlan\_pcp=7,actions=output:2

1

##### 3.匹配源/目的MAC

ovs-ofctl add-flow br0 in\_port=1,dl\_src=00:00:00:00:00:01/00:00:00:00:00:01,actions=output:2

ovs-ofctl add-flow br0 in\_port=1,dl\_dst=00:00:00:00:00:01/00:00:00:00:00:01,actions=output:2

1

2

##### 4.匹配以太网类型，范围为0-65535

ovs-ofctl add-flow br0 in\_port=1,dl\_type=0x0806,actions=output:2

1

##### 5.匹配源/目的IP

条件：指定dl\_type=0x0800，或者ip/tcp

ovs-ofctl add-flow br0 ip,in\_port=1,nw\_src=10.10.0.0/16,actions=output:2

ovs-ofctl add-flow br0 ip,in\_port=1,nw\_dst=10.20.0.0/16,actions=output:2

1

2

##### 6.匹配协议号，范围为0-255

条件：指定dl\_type=0x0800或者ip

# ICMP

ovs-ofctl add-flow br0 ip,in\_port=1,nw\_proto=1,actions=output:2

1

2

##### 7.匹配IP ToS/DSCP，tos范围为0-255，DSCP范围为0-63

条件：指定dl\_type=0x0800/0x86dd，并且ToS低2位会被忽略(DSCP值为ToS的高6位，并且低2位为预留位)

ovs-ofctl add-flow br0 ip,in\_port=1,nw\_tos=68,actions=output:2

ovs-ofctl add-flow br0 ip,in\_port=1,ip\_dscp=62,actions=output:2

1

2

##### 8.匹配IP ecn位，范围为0-3

条件：指定dl\_type=0x0800/0x86dd

ovs-ofctl add-flow br0 ip,in\_port=1,ip\_ecn=2,actions=output:2

1

##### 9.匹配IP TTL，范围为0-255

ovs-ofctl add-flow br0 ip,in\_port=1,nw\_ttl=128,actions=output:2

1

##### 10.匹配tcp/udp，源/目的端口，范围为0-65535

# 匹配源tcp端口179

ovs-ofctl add-flow br0 tcp,tcp\_src=179/0xfff0,actions=output:2

# 匹配目的tcp端口179

ovs-ofctl add-flow br0 tcp,tcp\_dst=179/0xfff0,actions=output:2

# 匹配源udp端口1234

ovs-ofctl add-flow br0 udp,udp\_src=1234/0xfff0,actions=output:2

# 匹配目的udp端口1234

ovs-ofctl add-flow br0 udp,udp\_dst=1234/0xfff0,actions=output:2

##### 11.匹配tcp flags

tcp flags=fin，syn，rst，psh，ack，urg，ece，cwr，ns

ovs-ofctl add-flow br0 tcp,tcp\_flags=ack,actions=output:2

1

##### 12.匹配icmp code，范围为0-255

条件：指定icmp

ovs-ofctl add-flow br0 icmp,icmp\_code=2,actions=output:2

1

##### 13.匹配vlan TCI

TCI低12位为vlan id，高3位为priority，例如tci=0xf123则vlan\_id为0x123和vlan\_pcp=7

ovs-ofctl add-flow br0 in\_port=1,vlan\_tci=0xf123,actions=output:2

1

##### 14.匹配mpls label

条件：指定dl\_type=0x8847/0x8848

ovs-ofctl add-flow br0 mpls,in\_port=1,mpls\_label=7,actions=output:2

1

##### 15.匹配mpls tc，范围为0-7

条件：指定dl\_type=0x8847/0x8848

ovs-ofctl add-flow br0 mpls,in\_port=1,mpls\_tc=7,actions=output:2

1

##### 16.匹配tunnel id，源/目的IP

# 匹配tunnel id

ovs-ofctl add-flow br0 in\_port=1,tun\_id=0x7/0xf,actions=output:2

# 匹配tunnel源IP

ovs-ofctl add-flow br0 in\_port=1,tun\_src=192.168.1.0/255.255.255.0,actions=output:2

# 匹配tunnel目的IP

ovs-ofctl add-flow br0 in\_port=1,tun\_dst=192.168.1.0/255.255.255.0,actions=output:2

# **一些匹配项的速记符**

### 速记符 匹配项

ip dl\_type=0x800

ipv6 dl\_type=0x86dd

icmp dl\_type=0x0800,nw\_proto=1

icmp6 dl\_type=0x86dd,nw\_proto=58

tcp dl\_type=0x0800,nw\_proto=6

tcp6 dl\_type=0x86dd,nw\_proto=6

udp dl\_type=0x0800,nw\_proto=17

udp6 dl\_type=0x86dd,nw\_proto=17

sctp dl\_type=0x0800,nw\_proto=132

sctp6 dl\_type=0x86dd,nw\_proto=132

arp dl\_type=0x0806

rarp dl\_type=0x8035

mpls dl\_type=0x8847

mplsm dl\_type=0x8848

### 指令动作

#### 1.动作为出接口

从指定接口转发出去

ovs-ofctl add-flow br0 in\_port=1,actions=output:2

1

#### 2.动作为指定group

group id为已创建的group table

ovs-ofctl add-flow br0 in\_port=1,actions=group:666

1

#### 3.动作为normal

转为L2/L3处理流程

ovs-ofctl add-flow br0 in\_port=1,actions=normal

1

#### 4.动作为flood

从所有物理接口转发出去，除了入接口和已关闭flooding的接口

ovs-ofctl add-flow br0 in\_port=1,actions=flood

1

#### 5.动作为all

从所有物理接口转发出去，除了入接口

ovs-ofctl add-flow br0 in\_port=1,actions=all

1

#### 6.动作为local

一般是转发给本地网桥

ovs-ofctl add-flow br0 in\_port=1,actions=local

1

#### 7.动作为in\_port

从入接口转发回去

ovs-ofctl add-flow br0 in\_port=1,actions=in\_port

1

#### 8.动作为controller

以packet-in消息上送给控制器

ovs-ofctl add-flow br0 in\_port=1,actions=controller

1

#### 9.动作为drop

丢弃数据包操作

ovs-ofctl add-flow br0 in\_port=1,actions=drop

1

#### 10.动作为mod\_vlan\_vid

修改报文的vlan id，该选项会使vlan\_pcp置为0

ovs-ofctl add-flow br0 in\_port=1,actions=mod\_vlan\_vid:8,output:2

1

#### 11.动作为mod\_vlan\_pcp

修改报文的vlan优先级，该选项会使vlan\_id置为0

ovs-ofctl add-flow br0 in\_port=1,actions=mod\_vlan\_pcp:7,output:2

1

#### 12.动作为strip\_vlan

剥掉报文内外层vlan tag

ovs-ofctl add-flow br0 in\_port=1,actions=strip\_vlan,output:2

1

#### 13.动作为push\_vlan

在报文外层压入一层vlan tag，需要使用openflow1.1以上版本兼容

ovs-ofctl add-flow -O OpenFlow13 br0 in\_port=1,actions=push\_vlan:0x8100,set\_field:4097-\>vlan\_vid,output:2

1

ps: set field值为4096+vlan\_id，并且vlan优先级为0，即4096-8191，对应的vlan\_id为0-4095

14.动作为push\_mpls

修改报文的ethertype，并且压入一个MPLS LSE

ovs-ofctl add-flow br0 in\_port=1,actions=push\_mpls:0x8847,set\_field:10-\>mpls\_label,output:2

1

#### 15.动作为pop\_mpls

剥掉最外层mpls标签，并且修改ethertype为非mpls类型

ovs-ofctl add-flow br0 mpls,in\_port=1,mpls\_label=20,actions=pop\_mpls:0x0800,output:2

1

#### 16.动作为修改源/目的MAC，修改源/目的IP

# 修改源MAC

ovs-ofctl add-flow br0 in\_port=1,actions=mod\_dl\_src:00:00:00:00:00:01,output:2

# 修改目的MAC

ovs-ofctl add-flow br0 in\_port=1,actions=mod\_dl\_dst:00:00:00:00:00:01,output:2

# 修改源IP

ovs-ofctl add-flow br0 in\_port=1,actions=mod\_nw\_src:192.168.1.1,output:2

# 修改目的IP

ovs-ofctl add-flow br0 in\_port=1,actions=mod\_nw\_dst:192.168.1.1,output:2

#### 17.动作为修改TCP/UDP/SCTP源目的端口

# 修改TCP源端口

ovs-ofctl add-flow br0 tcp,in\_port=1,actions=mod\_tp\_src:67,output:2

# 修改TCP目的端口

ovs-ofctl add-flow br0 tcp,in\_port=1,actions=mod\_tp\_dst:68,output:2

# 修改UDP源端口

ovs-ofctl add-flow br0 udp,in\_port=1,actions=mod\_tp\_src:67,output:2

# 修改UDP目的端口

ovs-ofctl add-flow br0 udp,in\_port=1,actions=mod\_tp\_dst:68,output:2

#### 18.动作为mod\_nw\_tos

条件：指定dl\_type=0x0800

修改ToS字段的高6位，范围为0-255，值必须为4的倍数，并且不会去修改ToS低2位ecn值

ovs-ofctl add-flow br0 ip,in\_port=1,actions=mod\_nw\_tos:68,output:2

1

#### 19.动作为mod\_nw\_ecn

条件：指定dl\_type=0x0800，需要使用openflow1.1以上版本兼容

修改ToS字段的低2位，范围为0-3，并且不会去修改ToS高6位的DSCP值

ovs-ofctl add-flow br0 ip,in\_port=1,actions=mod\_nw\_ecn:2,output:2

1

#### 20.动作为mod\_nw\_ttl

修改IP报文ttl值，需要使用openflow1.1以上版本兼容

ovs-ofctl add-flow -O OpenFlow13 br0 in\_port=1,actions=mod\_nw\_ttl:6,output:2

1

#### 21.动作为dec\_ttl

对IP报文进行ttl自减操作

ovs-ofctl add-flow br0 in\_port=1,actions=dec\_ttl,output:2

1

#### 22.动作为set\_mpls\_label

对报文最外层mpls标签进行修改，范围为20bit值

ovs-ofctl add-flow br0 in\_port=1,actions=set\_mpls\_label:666,output:2

1

#### 23.动作为set\_mpls\_tc

对报文最外层mpls tc进行修改，范围为0-7

ovs-ofctl add-flow br0 in\_port=1,actions=set\_mpls\_tc:7,output:2

1

#### 24.动作为set\_mpls\_ttl

对报文最外层mpls ttl进行修改，范围为0-255

ovs-ofctl add-flow br0 in\_port=1,actions=set\_mpls\_ttl:255,output:2

1

#### 25.动作为dec\_mpls\_ttl

对报文最外层mpls ttl进行自减操作

ovs-ofctl add-flow br0 in\_port=1,actions=dec\_mpls\_ttl,output:2

1

#### 26.动作为move NXM字段

使用move参数对NXM字段进行操作

# 将报文源MAC复制到目的MAC字段，并且将源MAC改为00:00:00:00:00:01

ovs-ofctl add-flow br0 in\_port=1,actions=move:NXM\_OF\_ETH\_SRC[]-\>NXM\_OF\_ETH\_DST[],mod\_dl\_src:00:00:00:00:00:01,output:2

1

2

ps: 常用NXM字段参照表

NXM字段 报文字段

NXM\_OF\_ETH\_SRC 源MAC

NXM\_OF\_ETH\_DST 目的MAC

NXM\_OF\_ETH\_TYPE 以太网类型

NXM\_OF\_VLAN\_TCI vid

NXM\_OF\_IP\_PROTO IP协议号

NXM\_OF\_IP\_TOS IP ToS值

NXM\_NX\_IP\_ECN IP ToS ECN

NXM\_OF\_IP\_SRC 源IP

NXM\_OF\_IP\_DST 目的IP

NXM\_OF\_TCP\_SRC TCP源端口

NXM\_OF\_TCP\_DST TCP目的端口

NXM\_OF\_UDP\_SRC UDP源端口

NXM\_OF\_UDP\_DST UDP目的端口

NXM\_OF\_SCTP\_SRC SCTP源端口

NXM\_OF\_SCTP\_DST SCTP目的端口

#### 27.动作为load NXM字段

使用load参数对NXM字段进行赋值操作

# push mpls label，并且把10(0xa)赋值给mpls label

ovs-ofctl add-flow br0 in\_port=1,actions=push\_mpls:0x8847,load:0xa-\>OXM\_OF\_MPLS\_LABEL[],output:2

# 对目的MAC进行赋值

ovs-ofctl add-flow br0 in\_port=1,actions=load:0x001122334455-\>OXM\_OF\_ETH\_DST[],output:2

#### 28.动作为pop\_vlan弹出报文最外层vlan tag

ovs-ofctl add-flow br0 in\_port=1,dl\_type=0x8100,dl\_vlan=777,actions=pop\_vlan,output:2

1

### meter表常用操作

由于meter表是openflow1.3版本以后才支持，所以所有命令需要指定OpenFlow1.3版本以上

ps: 在openvswitch-v2.8之前的版本中，还不支持meter

在v2.8版本之后已经实现，要正常使用的话，需要注意的是datapath类型要指定为netdev，band type暂时只支持drop，还不支持DSCP REMARK

#### 1.查看当前设备对meter的支持

ovs-ofctl -O OpenFlow13 meter-features br0

1

#### 2.查看meter表

ovs-ofctl -O OpenFlow13 dump-meters br0

1

#### 3.查看meter统计

ovs-ofctl -O OpenFlow13 meter-stats br0

1

#### 4.创建meter表

# 限速类型以kbps(kilobits per second)计算，超过20kb/s则丢弃

ovs-ofctl -O OpenFlow13 add-meter br0 meter=1,kbps,band=type=drop,rate=20

# 同上，增加burst size参数

ovs-ofctl -O OpenFlow13 add-meter br0 meter=2,kbps,burst,band=type=drop,rate=20,burst\_size=256

# 同上，增加stats参数,对meter进行计数统计

ovs-ofctl -O OpenFlow13 add-meter br0 meter=3,kbps,burst,stats,band=type=drop,rate=20,burst\_size=256

# 限速类型以pktps(packets per second)计算，超过1000pkt/s则丢弃

ovs-ofctl -O OpenFlow13 add-meter br0 meter=4,pktps,band=type=drop,rate=1000

#### 5.删除meter表

# 删除全部meter表

ovs-ofctl -O OpenFlow13 del-meters br0

# 删除meter id=1

ovs-ofctl -O OpenFlow13 del-meter br0 meter=1

#### 6.创建流表

ovs-ofctl -O OpenFlow13 add-flow br0 in\_port=1,actions=meter:1,output:2

group表

由于group表是openflow1.1版本以后才支持，所以所有命令需要指定OpenFlow1.1版本以上

### 常用操作

group table支持4种类型

all：所有buckets都执行一遍

select： 每次选择其中一个bucket执行，常用于负载均衡应用

ff(FAST FAILOVER)：快速故障修复，用于检测解决接口等故障

indirect：间接执行，类似于一个函数方法，被另一个group来调用

#### 1.查看当前设备对group的支持

ovs-ofctl -O OpenFlow13 dump-group-features br0

1

#### 2.查看group表

ovs-ofctl -O OpenFlow13 dump-groups br0

1

#### 3.创建group表

# 类型为all

ovs-ofctl -O OpenFlow13 add-group br0 group\_id=1,type=all,bucket=output:1,bucket=output:2,bucket=output:3

# 类型为select

ovs-ofctl -O OpenFlow13 add-group br0 group\_id=2,type=select,bucket=output:1,bucket=output:2,bucket=output:3

# 类型为select，指定hash方法(5元组，OpenFlow1.5+)

ovs-ofctl -O OpenFlow15 add-group br0 group\_id=3,type=select,selection\_method=hash,fields=ip\_src,bucket=output:2,bucket=output:3

1

2

3

4

5

6

#### 4.删除group表

ovs-ofctl -O OpenFlow13 del-groups br0 group\_id=2

1

#### 5.创建流表

ovs-ofctl -O OpenFlow13 add-flow br0 in\_port=1,actions=group:2

1

#### 6. table配置

数据流先从table0开始匹配，如actions有goto\_table，再进行后续table的匹配，实现多级流水线，如需使用goto table，则创建流表时，指定table id，范围为0-255，不指定则默认为table0

##### 1.在table0中添加一条流表条目

ovs-ofctl add-flow br0 table=0,in\_port=1,actions=goto\_table=1

1

##### 2.在table1中添加一条流表条目

ovs-ofctl add-flow br0 table=1,ip,nw\_dst=10.10.0.0/16,actions=output:2

1

### tunnel配置

如需配置tunnel，必需确保当前系统对各tunnel的remote ip网络可达

gre

##### 1.创建一个gre接口，并且指定端口id=1001

ovs-vsctl add-port br0 gre1 -- set Interface gre1 type=gre options:remote\_ip=1.1.1.1 ofport\_request=1001

1

##### 2.可选选项

将tos或者ttl在隧道上继承，并将tunnel id设置成123

ovs-vsctl set Interface gre1 options:tos=inherit options:ttl=inherit options:key=123

1

##### 3.创建关于gre流表

# 封装gre转发

ovs-ofctl add-flow br0 ip,in\_port=1,nw\_dst=10.10.0.0/16,actions=output:1001

# 解封gre转发

ovs-ofctl add-flow br0 in\_port=1001,actions=output:1

### vxlan

##### 1.创建一个vxlan接口，并且指定端口id=2001

ovs-vsctl add-port br0 vxlan1 -- set Interface vxlan1 type=vxlan options:remote\_ip=1.1.1.1 ofport\_request=2001

1

##### 2.可选选项

将tos或者ttl在隧道上继承，将vni设置成123，UDP目的端为设置成8472(默认为4789)

ovs-vsctl set Interface vxlan1 options:tos=inherit options:ttl=inherit options:key=123 options:dst\_port=8472

1

##### 3.创建关于vxlan流表

# 封装vxlan转发

ovs-ofctl add-flow br0 ip,in\_port=1,nw\_dst=10.10.0.0/16,actions=output:2001

# 解封vxlan转发

ovs-ofctl add-flow br0 in\_port=2001,actions=output:1

1

2

3

4

### sflow配置

##### 1.对网桥br0进行sflow监控

agent: 与collector通信所在的网口名，通常为管理口

target: collector监听的IP地址和端口，端口默认为6343

header: sFlow在采样时截取报文头的长度

polling: 采样时间间隔，单位为秒

ovs-vsctl -- --id=@sflow create sflow agent=eth0 target=\"10.0.0.1:6343\" header=128 sampling=64 polling=10 -- set bridge br0 sflow=@sflow

1

##### 2.查看创建的sflow

ovs-vsctl list sflow

1

##### 3.删除对应的网桥sflow配置，参数为sFlow UUID

ovs-vsctl remove bridge br0 sflow 7b9b962e-fe09-407c-b224-5d37d9c1f2b3

1

##### 4.删除网桥下所有sflow配置

ovs-vsctl -- clear bridge br0 sflow

1

### QoS配置

ingress policing

##### 1.配置ingress policing，对接口eth0入流限速10Mbps

ovs-vsctl set interface eth0 ingress\_policing\_rate=10000

ovs-vsctl set interface eth0 ingress\_policing\_burst=8000

1

2

##### 2.清除相应接口的ingress policer配置

ovs-vsctl set interface eth0 ingress\_policing\_rate=0

ovs-vsctl set interface eth0 ingress\_policing\_burst=0

1

2

##### 3.查看接口ingress policer配置

ovs-vsctl list interface eth0

1

##### 4.查看网桥支持的Qos类型

ovs-appctl qos/show-types br0

1

### 端口镜像配置

##### 1.配置eth0收到/发送的数据包镜像到eth1

ovs-vsctl -- set bridge br0 mirrors=@m \

-- --id=@eth0 get port eth0 \

-- --id=@eth1 get port eth1 \

-- --id=@m create mirror name=mymirror select-dst-port=@eth0 select-src-port=@eth0 output-port=@eth1

##### 2.删除端口镜像配置

ovs-vsctl -- --id=@m get mirror mymirror -- remove bridge br0 mirrors @m

1

##### 3.清除网桥下所有端口镜像配置

ovs-vsctl clear bridge br0 mirrors

1

##### 4.查看端口镜像配置

ovs-vsctl get bridge br0 mirrors

1

### 聚合口配置

##### 1.创建一个聚合口

ovs-vsctl add-port br0 dpdkbond0 \

-- set interface dpdkbond0 type=dpdk options:dpdk-devargs=0000:01:00.0,0000:02:00.0

1

2

##### 2.设置聚合口模式

# mode=1

ovs-vsctl set port dpdkbond0 bond\_mode=active-backup

# mode=2

ovs-vsctl set port dpdkbond0 bond\_mode=balance-slb

# mode=4

ovs-vsctl set port dpdkbond0 bond\_mode=balance-tcp

ovs-vsctl set port dpdkbond0 lacp=active

ovs-vsctl set port dpdkbond0 lacp=off

ovs-vsctl set port dpdkbond0 lacp=passive

##### 3.查看mode=4 lacp协商状态

ovs-appctl lacp/show

cle/details/73163041