

**DCFabric SDN控制器**

**管理员手册**

**文档编号：**DCFabricSDN-Controller-Admin

**文档名称：**开源DCFabricSDN控制器管理员手册

**版本信息：**V2.0.2

**建立日期：**2014年04月13日

**最后修改日期：**2015年05月24日

# 文档修订记录

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **版本编号** | **修订原因** | **简要说明**  **（变更内容和变更范围）** | **日期** | **变更人** | **批准日期** | **批准人** |
| V0.1 | 创建 |  | 2014-11-18 | 邓超 |  |  |
| V1.0 | 补充 |  | 2014-11-23 | 邓超 |  |  |
| V1.1 | 修改 |  | 2014-12-31 | 邓超 |  |  |
| V1.2 | 修改 | 整理管理员手册 | 2015-5-24 | 赵良智 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**修订原因：**创建；增加；修改；删除

# 文档审批信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **审批人** | **角色** | **审批日期** | **签字** | **备注** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 简介

## 编写目的

本文旨在辅助管理员安装和管理SDN控制器以及使用本SDN控制器连接管理Openflow交换机。

## 读者对象

本文档预期读者包括：

* 数据中心SDN控制器开发项目管理和开发相关人员
* 系统需求组、系统架构组、系统设计组
* 管理员、运营维护人员

## 文档约定

本手册所有描述适用于x86架构64位CentOS 6.5版本或x86 GNOS系统，如非必须，用于部署数据中心SDN控制器的服务器不得有部署其他非OS自带的第三方服务，否则需要重新评估能否正常兼容安装本数据中心SDN控制器。本文下文中所描述的安装部署服务器均默认符合此约定。

如果使用本手册遇到问题请联系相关人员。

## 术语定义

|  |  |  |
| --- | --- | --- |
| **术语和缩写** | **全拼** | **解释** |
| Java sdk | Java Software Development Kit | Java软件开发工具包 |
| Hbase | Hadoop DataBase | Hadoop数据库 |
| OpenStack |  | 云计算管理平台 |
|  |  |  |

# 安装部署

## 安装前准备

### 部署服务器配置

用于安装部署的所有服务器（不论上面是否部署SDN控制器）必须先配置好主机名。保证所有服务直接能够ping通主机名。

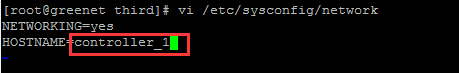
修改主机名

由于主机直接能通过主机名ping通，所有主机的主机名不能重复。

如假设主机名的规则为“controller\_N”，“N”从数字“1”开始递增。设置第一台服务器的主机名为“controller\_1”：

使用“vi”命令打开“/etc/sysconfig/network”文件，将里面的“HOSTNAME”的值修改为“controller\_1”，如果没有“HOSTNAME”则需 要新增，保存修改：

|  |
| --- |
| vi /etc/sysconfig/network |



修改完成后重启系统生效，重启命令：

|  |
| --- |
| reboot |

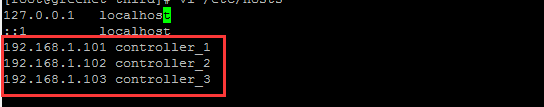
按此方法，修改所有服务器的主机名。

配置服务器hosts文件

为了所有主机之间通过主机名能相互ping通，需要在服务器hosts文件中配置主机IP和主机名的对应关系表，使用“vi”命令打开“/etc/hosts”文件，在文件最后面增加所有主机IP和主机名的对应关系并保存修改：

|  |
| --- |
| vi /etc/hosts |

假如有三台主机，主机“controller\_1”的IP地址为“192.169.1.101”、 主机“controller\_2”的IP地址为“192.169.1.102”、 主机“controller\_3”的IP地址为“192.169.1.103”，那么修改后所有服务器的“/etc/hosts”文件都应该应包含有如下内容：



### 版本编译

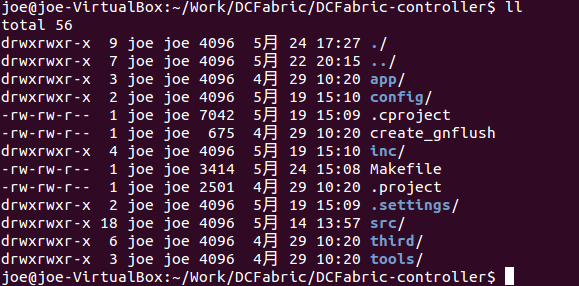
1. 获取DCFabric源码

1.从GitHub上获取到的控制器源码（注：需要有GitHub的权限才能下载）：

|  |
| --- |
| git clone https://github.com/China863SDN/DCFabric.git |



2.进入git下载目录，DCFabric，进入控制器源码目录DCFabric-controller，目录和文件结构和对应描述如下图示：



**app：**拓展应用目录，开发人员可以在此目录实现DCFabric的拓展，暂为空；

**inc：**控制器头文件目录，包含.h文件；

**src：**控制器源文件目录，包含所有.c文件；

**tools：**对接OpenStack工具

**third:** 第三方工具，包含书

**hbase：**hbase数据库相关头文件、java API调用实现文件、jvm运行依赖包；

**install-tools：**安装时需要用到的文件，如数据库建表脚本；

**json-lib：**第三方json开发工具；

**zookeeper：**zookeeper头文件及静态库文件；

编译DCFabric源码

1. 进入到“DCFabric-controller”目录

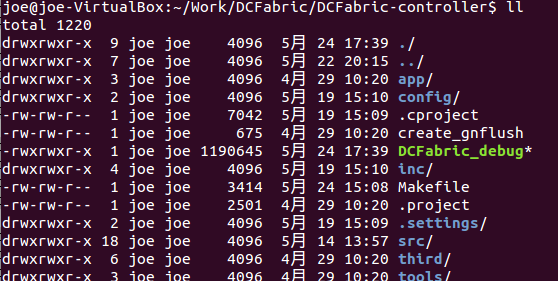


1. 执行编译

编译默认生成的是debug版本，该版本不具备hbase数据库访问模块。如需生成完整版本执行如需命令编译：

|  |
| --- |
| make TYPE=release |

编译生成“DCFabric”可执行文件，如果是debug版本则名称为“DCFabric\_debug”：



1. 至此版本编译完成。

### Java SDK安装

1. 获取JDK安装包

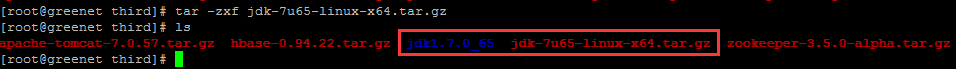
在安装部署SDN控制器、HBase数据库、zookeeper、WEB APP之前，需要获取并安装Java SDK安装包“jdk-7u65-linux-x64.tar.gz”。

由“2.1.2版本安装包获取 - 步骤三：解压安装包”，当前在“third”目录下已经有了该安装包。

解压JDK安装包

进入安装包目录，解压JDK安装包：

|  |
| --- |
| tar -zxf jdk-7u65-linux-x64.tar.gz |



1. 将解压后的文件移动安装到系统目录

执行如下命令解压后的目录解压安装到系统目录：

|  |
| --- |
| mkdir –p /usr/lib/java-1.7.0/  mv jdk1.7.0\_65 /usr/lib/java-1.7.0/  ln -s /usr/lib/java-1.7.0/jdk1.7.0\_65/jre/lib/amd64/server/libjvm.so /usr/lib64/libjvm.so |

Java环境变量配置

使用“vi”命令打开“/etc/profile”文件。

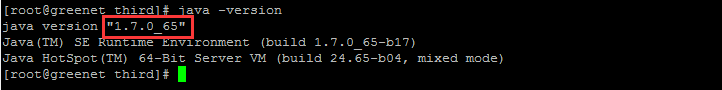
|  |
| --- |
| vi /etc/profile |

在打开文件最下面添加如下命令：

|  |
| --- |
| export JAVA\_HOME=/usr/lib/java-1.7.0/jdk1.7.0\_65  export JRE\_HOME=/usr/lib/java-1.7.0/jdk1.7.0\_65/jre  export PATH=/lib64:$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin:$PATH  export CLASSPATH=$CLASSPATH:.:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib |

执行完后输入如下命令查看当前Java版本号是否为“1.7.0\_65”，判断安装是否成功：

|  |
| --- |
| source /etc/profile  ldconfig  java -version |



### HBase数据库安装

HBase用于SDN控制器的数据持久化保存，供控制器集群完成数据同步，因此在SDN控制器单机部署的情况意义不大，但是需要部署，否则会因为缺失依赖导致错误。

HBase数据库部署在任意一台服务器上即可。

1. 获取HBase数据库安装包

在安装部署SDN控制器之前，需要获取并安装“hbase-0.94.22.tar.gz”。

由“2.1.2版本安装包获取 - 步骤三：解压安装包”，当前在“third”目录下已经有了该安装包。

解压HBase安装包

进入安装包目录，解压JDK安装包：

|  |
| --- |
| tar -zxf hbase-0.94.22.tar.gz |



将解压后的文件移动安装到系统目录

执行如下命令解压后的目录解压安装到系统目录：

|  |
| --- |
| mv hbase-0.94.22 /usr/hbase  cp /usr/hbase/hbase-0.94.22.jar /usr/hbase/lib/ |

HBase环境变量配置

使用“vi”命令打开“/etc/profile”文件。

|  |
| --- |
| vi /etc/profile |

在打开文件最下面添加如下命令：

|  |
| --- |
| export HBASE\_HOME=/usr/hbase  export PATH=$HBASE\_HOME/bin:$PATH |

执行如下命令使环境变量生效：

|  |
| --- |
| source /etc/profile  ldconfig |

修改HBase配置

使用“vi”命令打开“/usr/hbase/conf/hbase-site.xml”：

|  |
| --- |
| vi /usr/hbase/conf/hbase-site.xml |

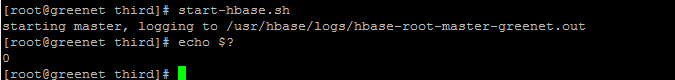
将里面的内容替换成如下内容，并保存修改：

|  |
| --- |
| <configuration>  <property>  <name>hbase.rootdir</name>  <value>/root/hbase\_dir/rootdir</value>  </property>  <property>  <name>hbase.zookeeper.property.dataDir</name>  <value>/root/hbase\_dir/datadir</value>  </property>  </configuration> |

启动HBase服务

执行如下命令启动HBase数据库服务：

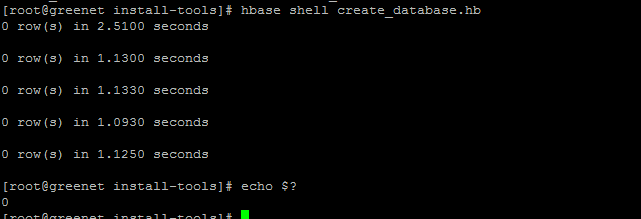
|  |
| --- |
| start-hbase.sh |



初始化数据库表

进入到“2.1.2版本安装包获取 - 步骤三：解压安装包”解压后的目录“install-tools”中，执行如下命令：

|  |
| --- |
| hbase shell create\_database.hb |



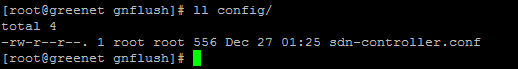
记录当前安装HBase数据库的服务的IP地址，后续安装部署SDN控制器时需要配置此IP地址。

## 控制器单机环境部署

### 获取SDN控制器版本

由“2.1.2版本编译”，当前在“DCFabric-controller”目录下已经有了控制器可执行文件“DCFabric”。

控制配置文件在“config”目录下：



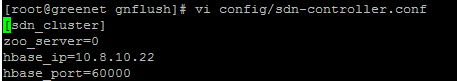
### 配置HBase数据库

此处仅描述HBase数据库的配置，如果需要详细了解和配置其他SDN控制器配置，请参考“3配置说明 - 3.1控制器配置”。

根据“2.1.5 HBase数据库安装”部署HBase数据库的服务IP地址修改SDN控制配置文件。假设IP地址为“192.168.1.101”，使用“vi”命令打开SDN控制器配置文件“config/sdn-controller.conf”：

|  |
| --- |
| vi config/ sdn-controller.conf |

找到“hbase\_ip”，配置项，将下面的IP地址替换为“192.168.1.101”后保存：



### 启动控制器服务

启动控制器有两种方式，区别在于控制台是否打印调试信息。

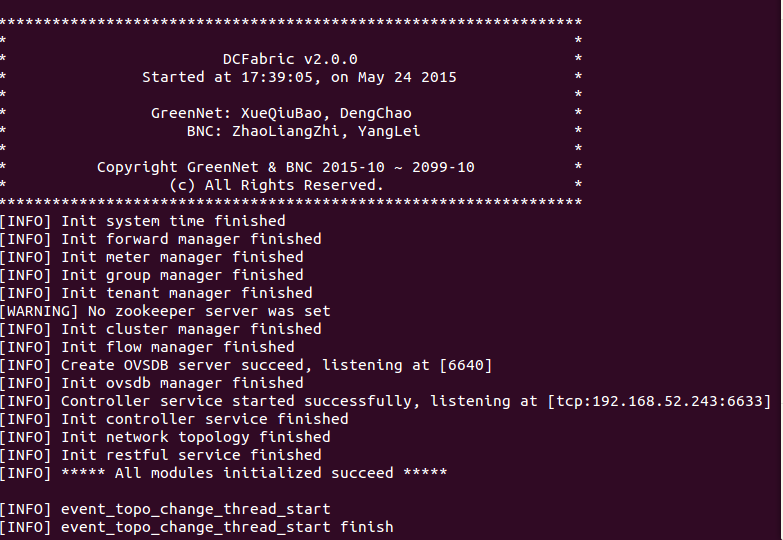
1. 正常启动

|  |
| --- |
| ./ DCFabric |

1. Debug启动

|  |
| --- |
| ./DCFabric 1 |

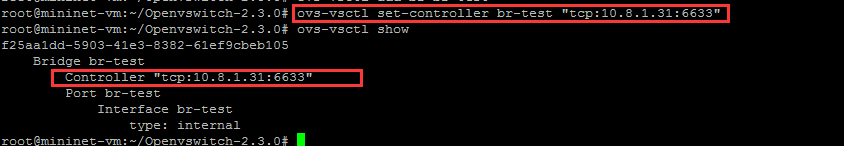
启动效果：



当控制台打印“\*\*\*\*\* All modules initialized succeed \*\*\*\*\*”时，表示控制器服务启动完成，如上图示。

### 配置交换机

控制器服务启动完成后，需要配置Openflow交换机连接到控制器，即设置交换机的“Controller”。如以Openvswitch为例：



## 控制器集群环境部署

### 集群部署前准备

请分别在两台及以上服务器上完成“2.2 控制器单机环境部署”中的“步骤一”至“步骤二”。

### 安装zookeeper服务

由于HBase数据库自带使用了zookeeper服务，如果不想再另外部署zookeeper服务，可以直接使用HBase数据库自带的zookeeper服务。将zookeeper服务IP设置为HBase数据库所在服务器的IP地址即可，跳过本步骤执行“步骤三：配置控制器”。

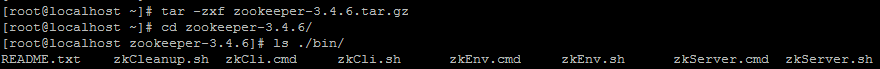
如果想另外单独部署，请在与HBase数据库不同的服务器上按照下面的指导部署zookeeper。

1. 获取zookeeper版本

由“2.1.2版本安装包获取 - 步骤三：解压安装包”，当前在“third”目录下已经有了zookeeper安装包文件“zookeeper-3.5.0-alpha.tar.gz”。

1. 安装zookeeper

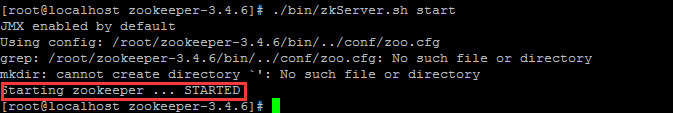
使用如下命令解压zookeeper安装包，后进入到解压目录：



1. 启动zookeeper服务

使用如下命令启动zookeeper服务：

|  |
| --- |
| ./bin/zkServer.sh start |



### 配置控制器

此处仅描述zookeeper服务的配置，如果需要详细了解和配置其他SDN控制器配置，请参考“3配置说明 - 3.1控制器配置”。

在所有部署SDN控制器的服务器上完成以下配置步骤：

1. 打开配置文件

打开版本解压生成目录中“config”子目录中的配置文件“sdn-controller.conf”

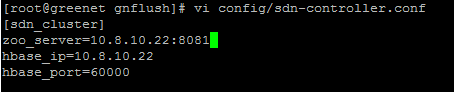
|  |
| --- |
| vi config/ sdn-controller.conf |

1. 编辑zookeeper配置项

找到配置项“zookeeper\_server”，将其值按如下格式修改填充（zookeeper服务默认监听端口为2181）：

|  |
| --- |
| zoo\_server=ServerIp:Port |

如：

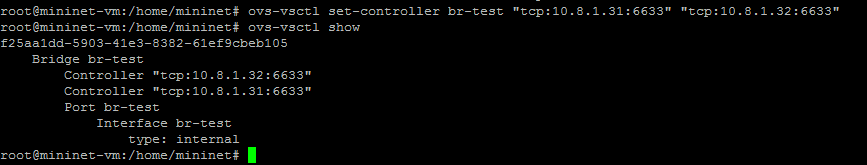


### 启动控制器

请参考“2.2 控制器单机环境部署”中的“[步骤三](#_启动控制器服务)”启动所有控制器。

### 配置交换机

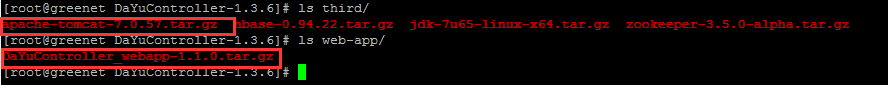
控制器服务启动完成后，需要配置Openflow交换机连接到所有控制器，即设置交换机的“Controller”。如以Openvswitch为例：



## 控制器WEB APP安装部署

### 获取SDN控制器WEB APP安装包

安装包共包括两部分：tomcat安装包和SDN控制器WEB APP包。由“2.1.2版本安装包获取 - 步骤三：解压安装包”，当前在“third”目录下已经有了tomcat安装包，在“web-app”目录以及有了WEB APP安装包。：



### 安装tomcat

进入“third”目录执行如下命令解压tomcat安装包即可：

|  |
| --- |
| tar -zxf apache-tomcat-7.0.57.tar.gz |



### 安装WEB APP

进入“web-app”目录解压WEB APP包,进入tomcat/webRoot/sdn\_center文件夹修改phy-route.properties文件加入物理交换机dpid。

### 启动WEB服务

进入到“apache-tomcat-7.0.57”目录，启动tomcat。



启动完毕后按如下URL格式通过浏览器访问WEB APP（tomcat默认使用8080端口）：

|  |
| --- |
| http://ServerIP: Port/sdn\_center/ |

## 控制器Openstack对接部署

Openstack使用不通的部署工具部署出来的环境存在较大差异，对接部署原理都一样。本文所描述的对接步骤适用于RDO方式部署的Openstack icehouse版本。

### Openstack控制节点配置

1. 工具脚本获取

在“2.1.2版本安装包获取 - 步骤三：解压安装包”，当前在“install-tools”目录下有shell脚本 “mechanism\_gnflush.py”是用于数据中心SDN与Openstack控制中心对接的。

将这个脚本拷贝到Openstack控制节点服务器上如下目录下：

|  |
| --- |
| /usr/lib/python2.6/site-packages/neutron/plugins/ml2/drivers/ |

关闭相关服务

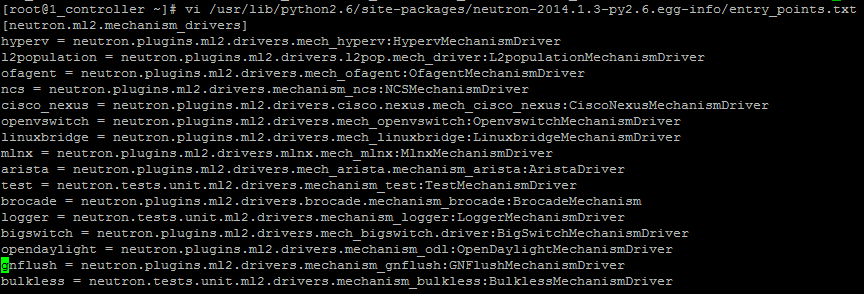
在Openstack控制中心服务器上执行如下命令停止相关服务：

|  |
| --- |
| service neutron-server stop |

配置ml2 plugin

在数据中心文件“/usr/lib/python2.6/site-packages/neutron-2014.1.3-py2.6.egg-info/entry\_points.txt”中“[neutron.ml2.mechanism\_drivers]”配置节点下增加GNFlush控制器ML2 plugin的entry point：

|  |
| --- |
| vi /usr/lib/python2.6/site-packages/neutron-2014.1.3-py2.6.egg-info/entry\_points.txt |



配置ml2

配置ml2前，需要准备如下信息：

* 租户网络类型：管理员可以根据需要选择“gre”或者“vlan”。
* 控制器IP地址：数据中心SDN控制器所在服务器的IP地址。
* 控制器Rest服务端口：Rest服务监听端口需要查看SDN控制器配置文件中“[rest\_port]”配置，默认为“8081”

在Openstack控制中心服务器上执行如下命令修改ml2配置，以租户网络类型为“gre”为例：

|  |
| --- |
| crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 mechanism\_drivers gnflush  crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 tenant\_network\_types <租户网络类型> |

在“/etc/neutron/plugins/ml2/ml2\_conf.ini”文件最后面添加“ml2\_gnflsuh”的选项:

|  |
| --- |
| [ml2\_gnflush]  password = admin  username = admin  url = http://<控制器IP>:<控制Rest服务端口>/controller/nb/v2/neutron |

创建ml2数据库

在Openstack控制中心服务器上执行如下命令：

|  |
| --- |
| mysql -e "drop database if exists neutron\_ml2;"  mysql -e "create database neutron\_ml2 character set utf8;"  mysql -e "grant all on neutron\_ml2.\* to 'neutron'@'%';"  neutron-db-manage --config-file /usr/share/neutron/neutron-dist.conf --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugin.ini upgrade head |

启动相关服务

在Openstack控制中心服务器上执行如下命令：

|  |
| --- |
| service neutron-server start |

### Openstack计算节点和网络节点配置

1. 工具脚本获取

在“2.1.2版本安装包获取 - 步骤三：解压安装包”，当前在“install-tools”目录下有shell脚本“gnflush-controller.sh”用于设置计算节点和网络节点与数据中心SDN控制器对接的配置。

将这个脚本拷贝到Openstack所有计算节点和网络节点服务器上任意目录下。

关闭相关服务

在所有Openstack计算节点和网络节点上服务器上执行如下命令：

|  |
| --- |
| service neutron-openvswitch-agent stop |

配置Openvswitch

配置前，需要准备如下信息：

* 本地IP：当前服务器的IP地址。
* 控制器IP地址：数据中心SDN控制器所在服务器的IP地址。

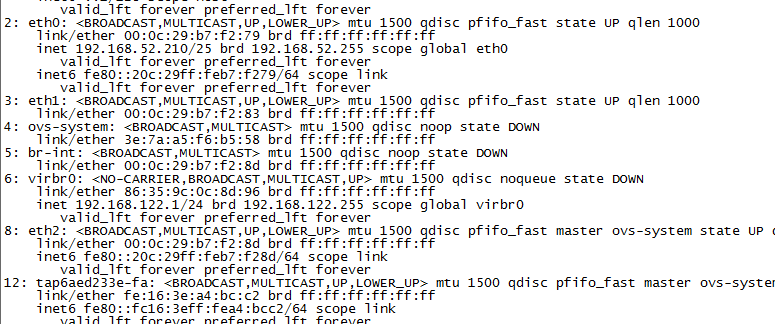
在所有Openstack计算节点和网络节点上服务器上执行如下命令：

|  |
| --- |
| ./gnflush-controller.sh --local\_ip <本地IP> --gnflush\_ip <控制器IP地址> |

网络和计算节点对接物理交换机配置

配置前，需要在网络节点和计算节点配置至少2块网卡，一块作为openstack内部控制网络使用，另一块网卡作为对接Openflow物理交换机使用。

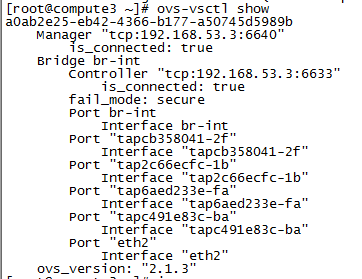
如下图，eth0为openstack内部控制使用的网络，eth2用来对接物理交换机，eth2可以不需要配置网络地址。



打开控制端，输入ovs-vsctl add-port br-int eth2



然后输入ovs-vsctl show即可看到br-int下面已经连接上了eth2，



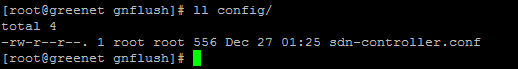
同时eth2口需要物理网线连接到openflow物理交换机，同时在物理交换机控制口输入set-controller 192.168.53.3:6633。该命令各个品牌交换机均不相同，需要咨询交换机厂商后获取命令。这样网络和计算节点对接物理服务器已经成功。

### DCFabric控制器配置文件修改

#### 找到控制器配置文件

当前在“DCFabric-controller”目录下已经有了控制器可执行文件“DCFabric”。

控制配置文件在“config”目录下：

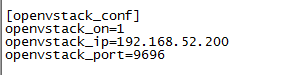


#### 配置openstack IP 和端口

假设openstack架设在192.168.52.200服务器上，使用“vi”命令打开SDN控制器配置文件“config/sdn-controller.conf”：

|  |
| --- |
| vi config/ sdn-controller.conf |

找到“openvstack\_conf”，配置项，将下面的openstack\_ip=地址替换为“192.168.52.200”，openvstack\_on=1，openvstack\_port=9696后保存：



# 配置说明

## 控制器配置

本节针对“[2.1 安装前准备部署](#_控制器安装部署) - 2.1.2 版本安装包获取”中的配置文件以添加注释的方式详细说明。

|  |
| --- |
| [controller]  #控制器使用的物理网卡名称，即当前服务器服务IP使用的网卡名称。管理员需要根据实际情况配置，一般情况下默认为服务器第一张网卡即可  manager\_eth=eth0  #控制器监听端口，即OpenFlow交换机连接控制器时使用的端口，默认使用6633，建议不要改动  of\_port=6633  #命令行工具监听端口，管理员可以根据当前空闲端口情况配置  cmd\_port=7744  #北向rest接口监听端口，管理员可以根据当前空闲端口情况配置  rest\_port=8081  #LLDP发送时间间隔（单位：秒），管理员可以根据当前网络实际需求更新。该时间配置越短，网络拓扑刷新越快  lldp\_interval=8  #能支持网络拓扑中主机的数量  MacUser\_hsize=198971  #主机Mac地址有效时间  MacUser\_lifetime=100  #能支持动态流表项的最大条数  FlowEntry\_hsize=98971  #能支持静态流表项的最大条数  StaticFlow\_hsize=8971  #l2 flow entry config  # L2层流表项有效时间：流表项经历“l2\_flow\_idle\_time”时间没匹配到包后删除（单位：秒），管理员可以根据当前网络实际需求更新，如果网络拓扑固定，建议设置较大值  l2\_flow\_idle\_time=100  # L2层流表项有效时间：流表项从产生起经历“l2\_flow\_hard\_time”时间后删除（单位：秒），管理员可以根据当前网络实际需求更新，如果网络拓扑固定，建议设置较大值  l2\_flow\_hard\_time=200  #l3 flow entry config  # L3层流表项有效时间：流表项经历“l2\_flow\_idle\_time”时间没匹配到包后删除（单位：秒），管理员可以根据当前网络实际需求更新，如果网络拓扑固定，建议设置较大值  l3\_flow\_idle\_time=100  # L3层流表项有效时间：流表项从产生起经历“l2\_flow\_hard\_time”时间后删除（单位：秒），管理员可以根据当前网络实际需求更新，如果网络拓扑固定，建议设置较大值  l3\_flow\_hard\_time=200  #reverse\_flow\_mod configure is just for test: 0,off; 1,on;  #测试开关，有些仪表测试时只会发送package\_in包，在没有目的mac的情况下不能产生正向流表，只会下发package\_out。开启此模式后下发package\_out的同时会下发一条反向流表，用于测试流表下发吞吐量，默认关闭使用正常处理逻辑。 0：关闭； 1：开启；  reverse\_flow\_mod=0  [sdn\_cluster]  #zookeeper服务访问地址，配置为“0”时不使用zookeeper（配置示例： 10.8.1.145:2181  ），使用场景请参考本手册“2.3节”，“步骤二”和“步骤三”  zoo\_server=0  #Hbase数据库服务访问IP，使用场景请参考本手册“2.1.5节”  hbase\_ip=10.8.10.22  #Hbase数据库服务访问端口，初始化预设，如无特殊情况，请不要随意修改  hbase\_port=60000  [openstack]  #Openvswitch控制器管理端口，仅控制器被设置为Openvswitch的manager时使用，Openvswitch默认使用6640，请不要随意修改  ovsdb\_port=6640  #下发给Openvswitch交换机OpenFlow版本号：of10或of13。仅当控制器被设置为Openvswitch的manager时使用，建议使用新的Openflow1.3版本，即配置of13  ovsdb\_of\_version=of13  #创建tunnel的类型。仅当Openstack环境下使用控制器时使用。可以配置为“gre”或“vlan”  ovsdb\_tunnel\_type=gre  #openstack 配置相关  [openvstack\_conf]  #是否对接openstack，1：对接，0：无对接  openvstack\_on=1  #对接openstack 控制节点ip地址  openvstack\_ip=192.168.52.200  #对接openstack端口  openvstack\_port=9696 |

# 后台管理

## 后台信息查询

|  |  |
| --- | --- |
| **管理内容** | **后台命令** |
| 查询所有交换机信息 | sendcmd -p 7744 "show sw" |
| 查询所有交换机路由 | sendcmd -p 7744 "show route" |
| 查询所有流表 | sendcmd -p 7744 "show flows" |
| 查询邻接矩阵 | sendcmd -p 7744 "show matrix" |
| 查询拓扑信息 | sendcmd -p 7744 "show topo" |
| 查询所有主机 | sendcmd -p 7744 "show hosts" |

# 常见问题

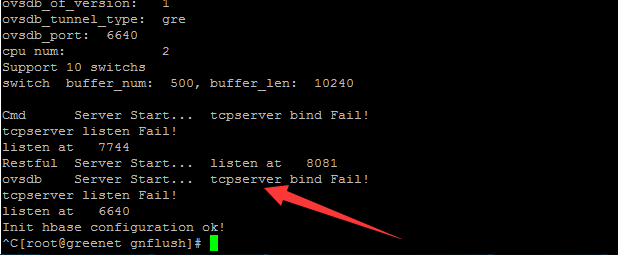
## 控制器无法启动成功

### 问题描述：

控制器启动时启动失败。

### 原因分析：

查看控制台打印信息，如果出现如下内容：



结束当前启动命令，执行如下命令查看是否存在其他控制器进程导致服务端口被占用：

|  |
| --- |
| ps -ef |grep -v "grep" |grep GNFlush --color |

例如：



### 解决方案：

如果需要重新启动控制器，请先结束当前控制器进程，然后重新启动控制器。

如果不用重新启动，因为控制器进程正在运行，无需再次启动。

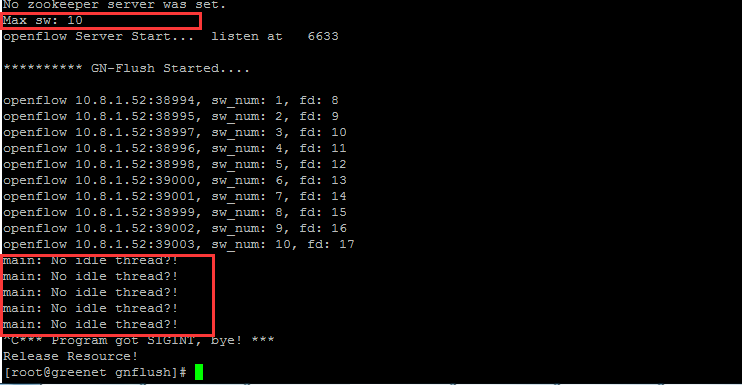
## 交换机无法正常连接

### 问题描述：

交换机连接控制器后发现交换机拓扑上没有新连接的交换机。

### 原因分析：

查看控制器后台打印信息：



如发现打印“main: No idle thread?!”，表示当前交换机连接已经到达上限，也可以查看控制台打印的“Max sw”的值和当前已经连接成功的交换机个数是否相等。

### 解决方案：

重新编译生成能够支持更多交换机连接的控制器版本。