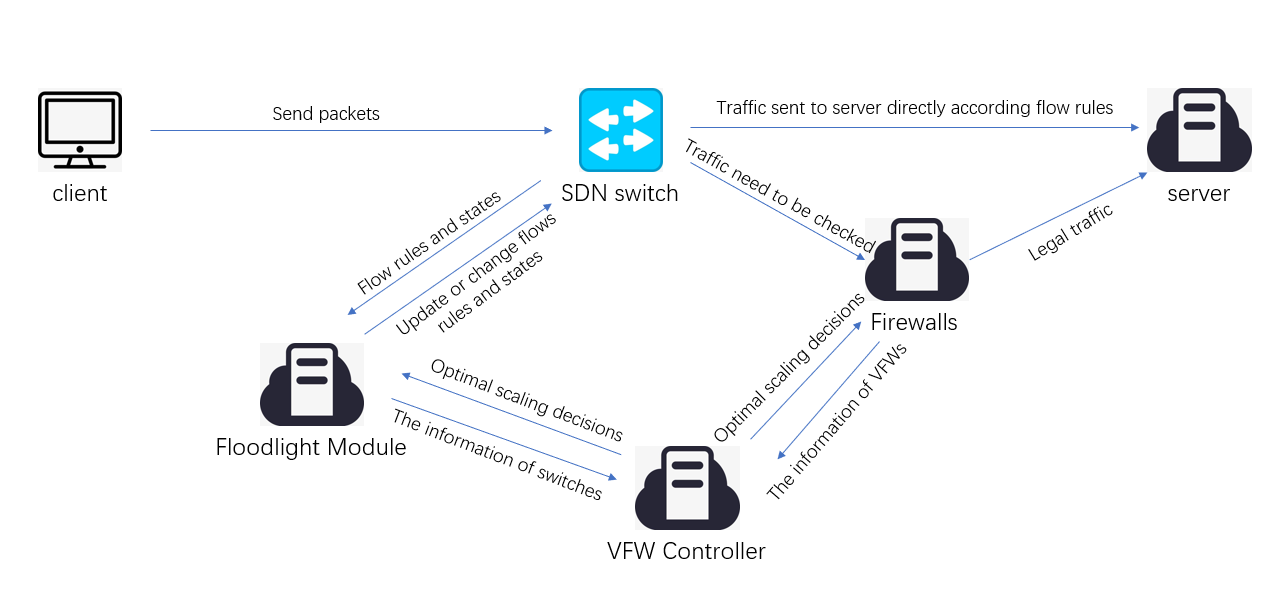
**IDS（NFV/SDN）实验环境配置说明**

初步设计一个实验结构如下图所示：



为了方便管理、优化实验效果，整个实验使用尽量少的物理机（建设传输时延和其他因素带来的影响），将各个模块独立打包成docker镜像，接下来按照模块进行环境配置说明。

目前决定使用两台虚拟机A和B。

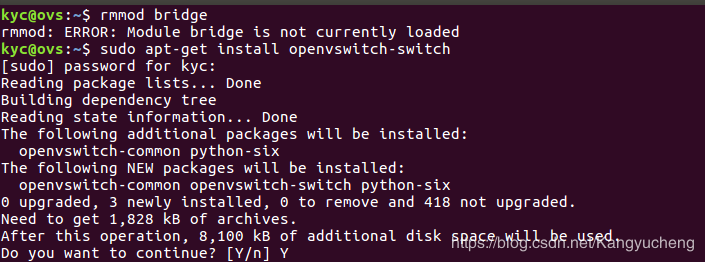
虚拟机A：宿主机安装floodlight，具体过程见后文。安装配置docker，docker开容器运行各个程序，包括Client、Server、IDS控制器、IDS实例，均以docker容器的方式运行，接下来会描述每个容器的镜像制作过程（或者直接从dockerhub拉取）。

虚拟机B：由于SDN交换机使用OVS（Openvswitch）项目，安装过程需要替换内核模块，因此无法运行在docker容器中，单独开一个虚拟机运行OVS。

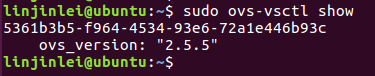
1. **Openvswitch**

**安装：**

1. sudo apt-get update
2. rmmod bridge
3. sudo apt-get install openvswitch-switch



输入上面三个指令即可完成安装，验证安装结果：sudo ovs-vsctl show，如果出现一串16进制序列以及ovs的版本号即安装成功。



OVS中的网桥即表示交换机。

**使用：**

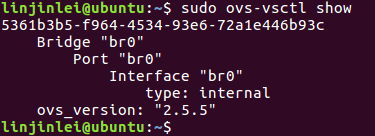
1. 添加网桥br0：sudo ovs-vsctl add-br br0
2. 列出所有的网桥：sudo ovs-vsctl list-br



1. 删除网桥：sudo ovs-vsctl del-br br0
2. 启动网桥：sudo ifconfig br0 up

这个一定要启动，不启动无法连接到控制器

1. 查看网桥详细信息：sudo ovs-vsctl show



如果与is\_connected:true字样，说明连接控制器成功。

1. 设置控制器：sudo ovs-vsctl set-controller br0 tcp:ip:6633

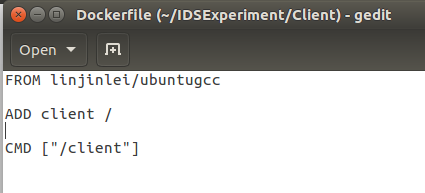
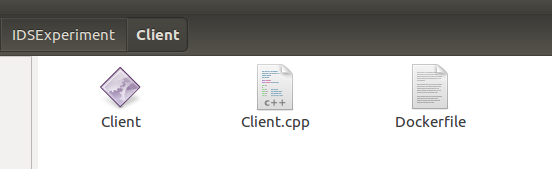
这里控制器的端口要和floodlight的端口匹配，默认端口是6633

1. 删除控制器：sudo ovs-vsctl del-controller br0

OVS这部分到这里配置就已经结束了，目前就剩下连接控制器，在后文会有描述。

1. **Client**

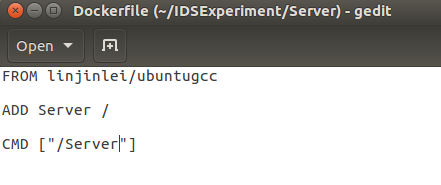
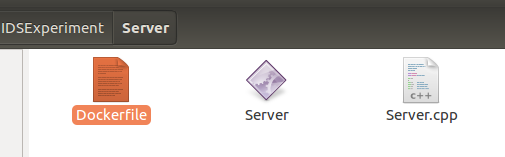
客户端用来发送流量到SDN交换机，经过SDN交换机转发到对应的IDS实例，最终由Server接收。目前由于不知道Client要跑的程序是什么，这里只给出一个Dockerfile，建议本机编译生成可执行文件（镜像中只配置gcc和g++），到时候只需要把可执行文件替换即可。



Dockerfile写了个比较简单的版本，如果后面实验有需要根据实际情况调整。

1. **Server**

具体情况和Client类似。



Server和Client目前看来都是持续运行的，所以在Dockerfile中最后都是直接执行，这样就不用进入镜像去手动执行。

1. **vIDS**

这部分暂时未配置。

1. **Floodlight**

这部分已经做成了docker的镜像，要用的话直接输入指令：

sudo docker pull linjinlei/floodlightimg

即可。实际实验的时候这个模块不需要重复构建，里面还有一些参数可能要手动去配置，所以还是进入容器手动编译开启服务，但是镜像的创建还是由docker-compose完成。

1. **整体环境搭建**

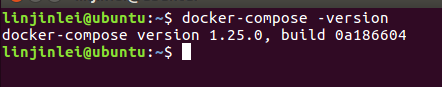
这部分是整个实验环境的搭建，分为两部分：openvswitch配置；其他模块配置。

其他模块的配置用docker-compose完成：

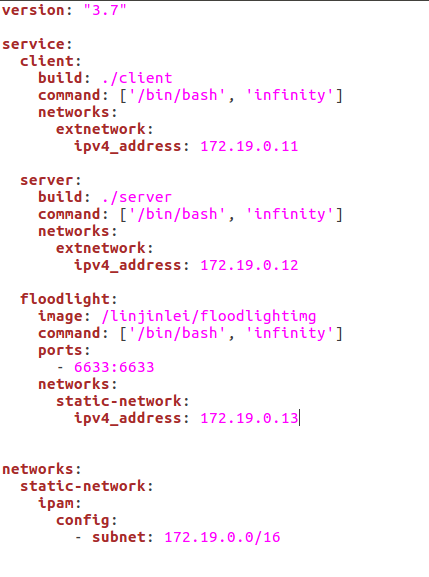
首先要安装docker-compose，步骤如下：



去GitHub下载一个可执行文件，版本自选即可，推荐1.25.0，因为docker用的版本是最新的。保存到指定位置，增加可执行权限即可。

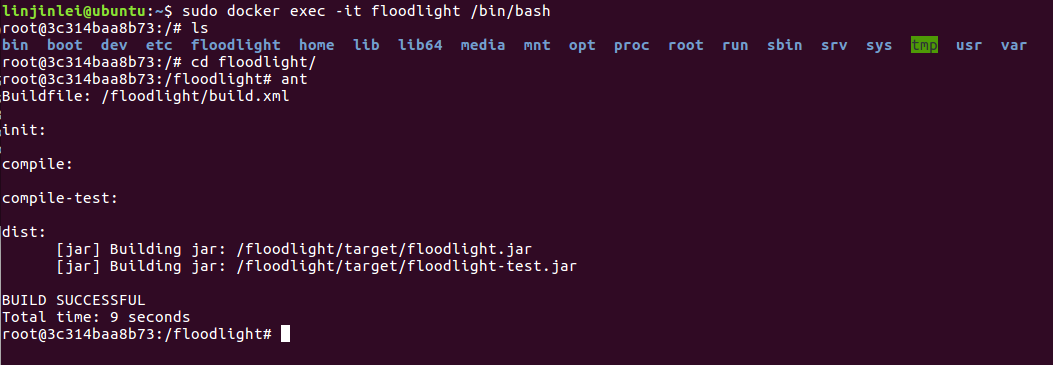


接着编写docker-compose文件：

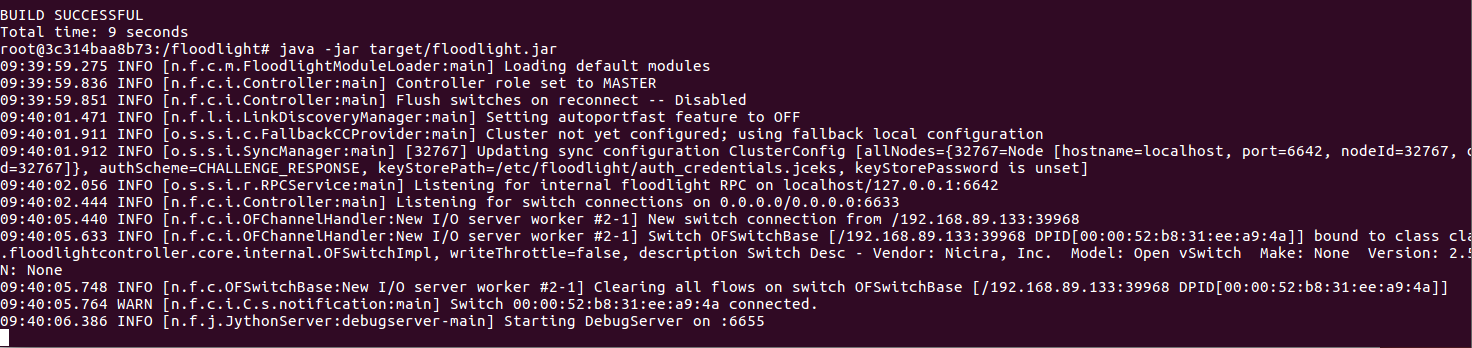


docker-compose.yml文件暂时只写了这三个部件，为了后面实验方便，都是配置了静态的IPv4地址，其中floodlight由于需要和另一台机器上的OVS模块进行交互，OVS需要访问floodlight的6633端口，因此需要做一个端口的映射。

之后docker-compose即可构建这些容器，floodlight是需要进入容器手动配置的。进入floodlight容器，然后输入ant进行编译，看到成功的提示即可。



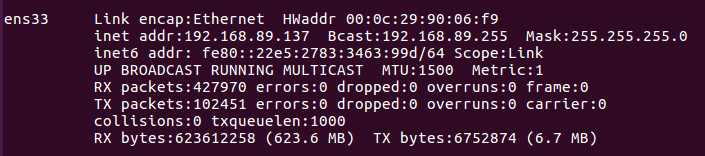
输入指令：java -jar target/floodlight.jar即可运行floodlight控制器，如果要进行其他的配置另行解决。



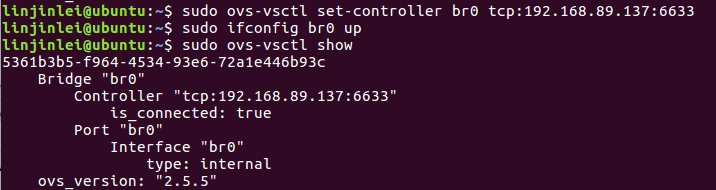
看到这样的输出就配置成功了。

接下来就是连接OVS到floodlight控制器：

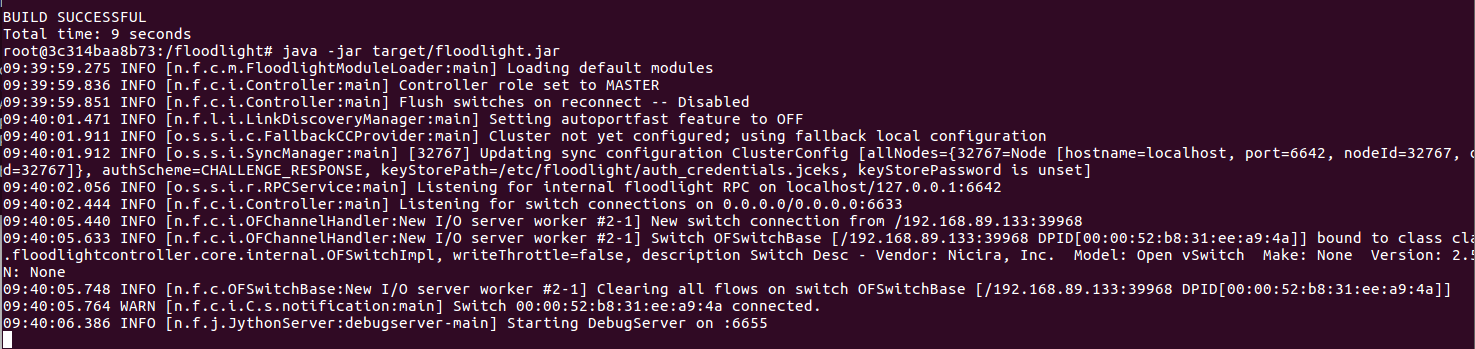
首先需要查看floodlight容器所在的虚拟机（虚拟机A）的IP地址：



之前docker-compose里面做了端口映射，所以接下来直接去虚拟机B配置OVS即可：



这里显示已经连接上了，返回虚拟机A看floodlight那边的状态：



这里也可以看到连接成功了。

到这里已经完成了配置，下面是另一种方式配置floodlight模块，可选。

1. **Floodlight直接配置在虚拟机A上**

上面的做法是将floodlight放到docker里面作为容器运行，需要做端口映射，也可以将floodlight模块直接装到虚拟机A的宿主机里面，这样就不需要做端口映射，同时，可以使用虚拟机A的可视化界面来管理OVS。

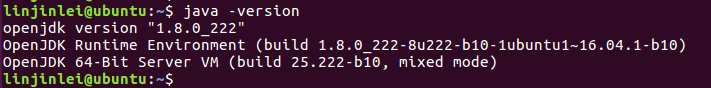
这种情况下配置floodlight相对简单，方法和上面docker中类似。

首先虚拟机里面要有java8环境，要安装ant：

sudo apt-get update

sudo apt-get install openjdk-8-jdk

java –version



之后就是进入floodlight的文件夹，输入ant进行编译，完成后输入：java -jar target/floodlight.jar即可。

Floodlight的版本选择：可以选择最新版，有比较美观、功能多的管理界面，但是配置起来会麻烦一些。我们这里选择的是0.91版本，链接如下：

<http://www.projectfloodlight.org/download/>

下载后解压即可。

之后的连接过程和上面一致。