Preface

Originally, I do not think it necessary to write this tutorial, because the installation of floodlight and mininet build data center topology tutorial have been on the Internet, why write this, this is because when I was in the process of reproduction, found some problem brought by the floodlight update process, in a nutshell, that is, floodlight is in the update, but the tutorial is too old. This tutorial is actually my summary when I face problems and solve the problem.

background  
Mininet uses lightweight virtualization technology, based on Mininet researchers can build a custom topology on the laptop SDN network, and SDN-related innovative design test and verification. Once the validation is successful, it can be deployed in the actual environment. The simulation test based on Mininet can evaluate the effect of multi-data center network application design, and it can be used as a reference for data center network application development.

First, the preparation before installation

(1) Linux: Ubuntu14.04 version here to provide a mirror, a variety of versions are here http://pan.baidu.com/s/1skKN5CT

(2) install the JDK, Ant provide the command line

sudo apt-get install build-essential default-jdk ant python-dev

（3）Install git to provide command line  
Sudo apt-get install git

Second, install Floodlight  
2.1 download source  
(1) Download and compile Floodlight from Github to provide the command line

git clone git://github.com/floodlight/floodlight.git

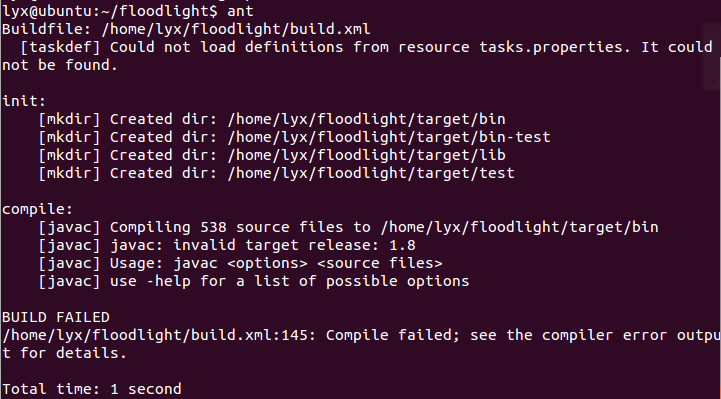
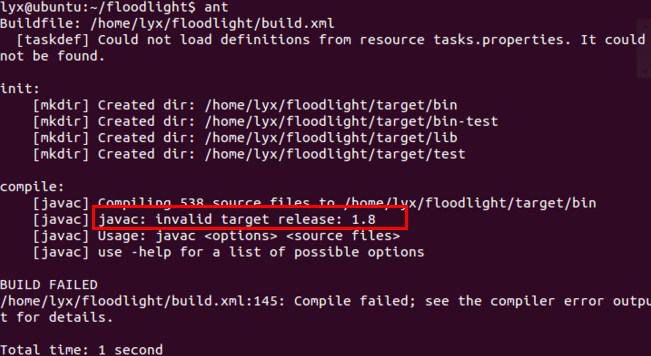
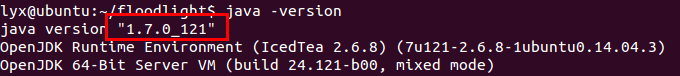
(2) If git download speed is slow, you can choose to download directly from github, and then extract, or can also be downloaded from the network disk, and then copy the virtual machine http://pan.baidu.com/s / 1qYe2s8k

2.2 Compile and install

cd floodlight

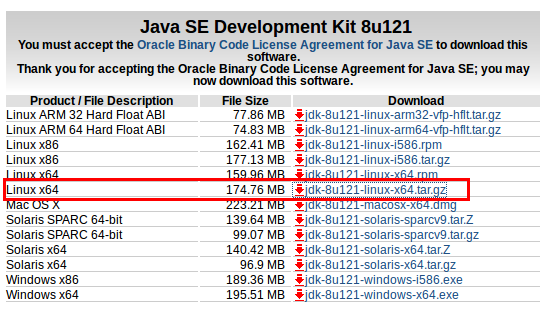
ant

Direct compilation will appear the following error, see the following error message

[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-1.png)  
[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-12.png)  
We found that the error message suggested that jdk version of the need for 1.8, let's look at our jdk version  
[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-3.png)

Found that is 1.7, which is because the previous installation jdk orders, the installation is default-jdk, and ubuntu14.04 acquiescence jdk is 1.7, why would this problem appear?Because this article refers to the tutorial, floodlight is not now 1.2 version, jdk needs below 1.7, so there is no problem, so we now need to install jdk 1.8.

1. source package preparation:  
   First to the official website to download jdk,

http：//www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html，download jdk-8u121-linux-x64.tar.gz[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-4.png)  
[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-5.png)  
(2) extract the source package  
Through the terminal in the / usr / local directory to create a new java folder, the command line

1. # sudo mkdir /usr/local/java

And then download the compressed package to the java folder, the command line:

# sudo cp jdk-8u121-linux-x64.tar.gz /usr/local/java   
Into the jdk source package directory

Then enter the java directory, the command line:

# cd /usr/local/java

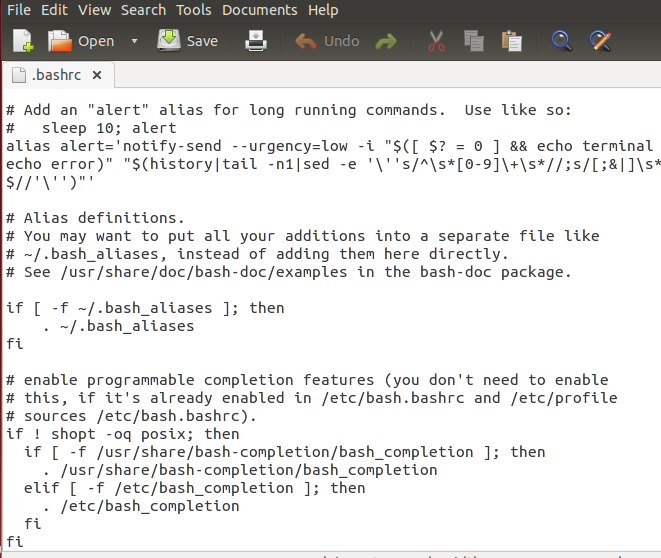
Decompress the compressed package, the command line:

# sudo tar xvf jdk-8u121-linux-x64.tar.gz

Then you can delete the compressed package, the command line:

# sudo rm jdk-8u121-linux-x64.tar.gz

(3) set the jdk environment variable  
Here is the global setting method, which is the common environment variable for all users

$ sudo gedit ~/.bashrc .As shown below:   
[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-6.png)  
After opening, add at the endJava

export JAVA\_HOME=/usr/local/java/jdk1.8.0\_121

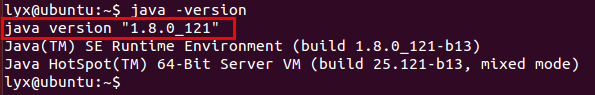
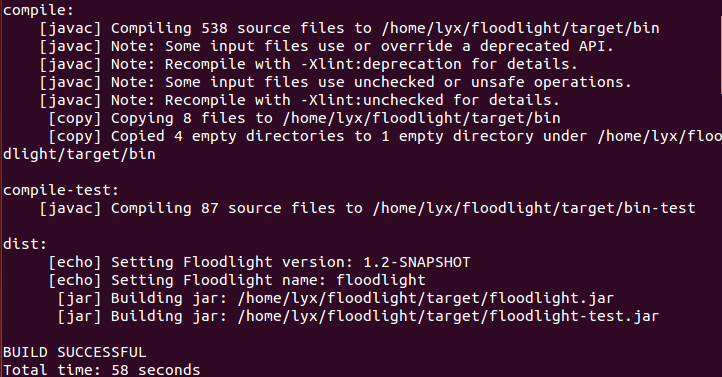
export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:JAVAHOME/lib:{JRE\_HOME}/lib

export PATH=JAVAHOME/bin:PATH

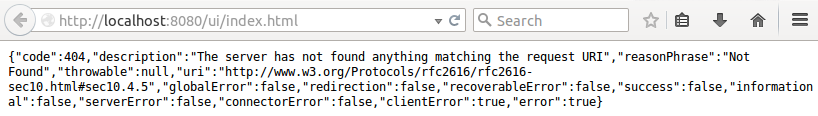
|  |  |
| --- | --- |
| 1  2  3  4 | Export JAVA\_HOME=/usr/local/java/jdk1.8.0\_121  export JRE\_HOME=${JAVA\_HOME}/jre  export CLASSPATH=.:JAVAHOME/lib:{JRE\_HOME}/lib  export PATH=JAVAHOME/bin:PATH |

After the restart we can find jdk has been updated to the 1.8 version

[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-7.png)  
Then compile:  
[http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-8.png](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-8.png)  
Compiled successfully  
[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-9.png)  
Then start floodlight through the command line

# java -jar target/floodlight.jar

After the start, through the browser to access floodlight management interface, [http://localhost:8080/ui/index.html](http://localhost:8080/ui/index.html )

The results of the following problems, floodlight1.2 management interface can not be accessed[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-10.png)  
By finding the information, it seems that it is because the latest version of the problem, there are two ways to solve

1) Configure with git Submodule

A set of settings through the git Submodule command, through the command line (here slightly explained under the git Submodule command, git Submodule is a very good multi-project tool to use the common class library, he allows the class library project as a repository , The subproject is as an own independent git project in the parent project. commit, push, pull, while the parent project contains subprojects as Submodule, and the parent project can specify the subproject header, parent item Submitted messages will contain Submodule information, and then clone parent items can be initialized when Submodule.

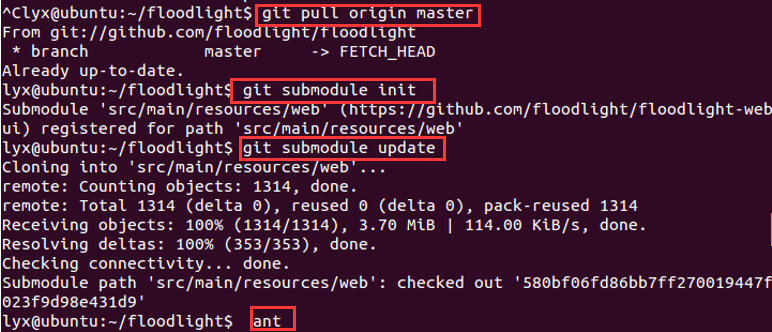
But this method can access the management interface, but can not interact with mininet, the specific reasons are unknown, so so far still obediently with the old version

# git pull origin master

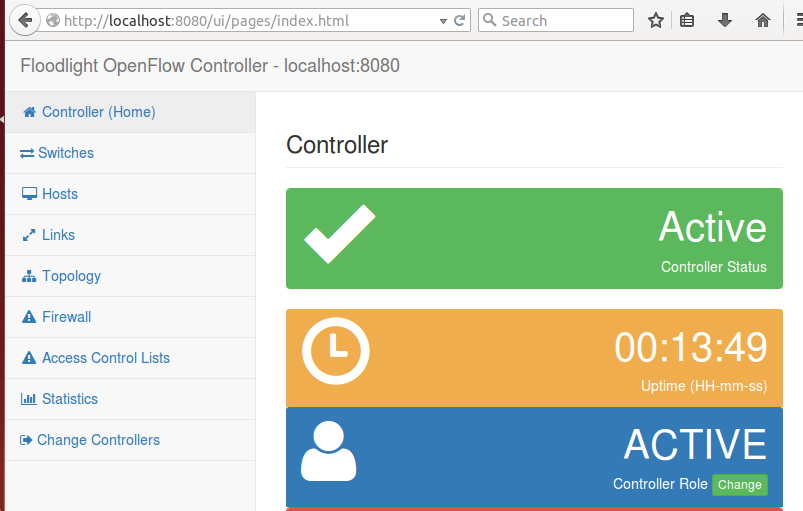
# git submodule init

# git submodule update

# ant

[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-11.png)  
Start floodlight after compilation:

# java -jar target/floodlight.jar

And then through http: // localhost: 8080 / ui / index.html visit floodlight management interface, found to be able to visit[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-12.png)  
**2) with the old version 0.91**

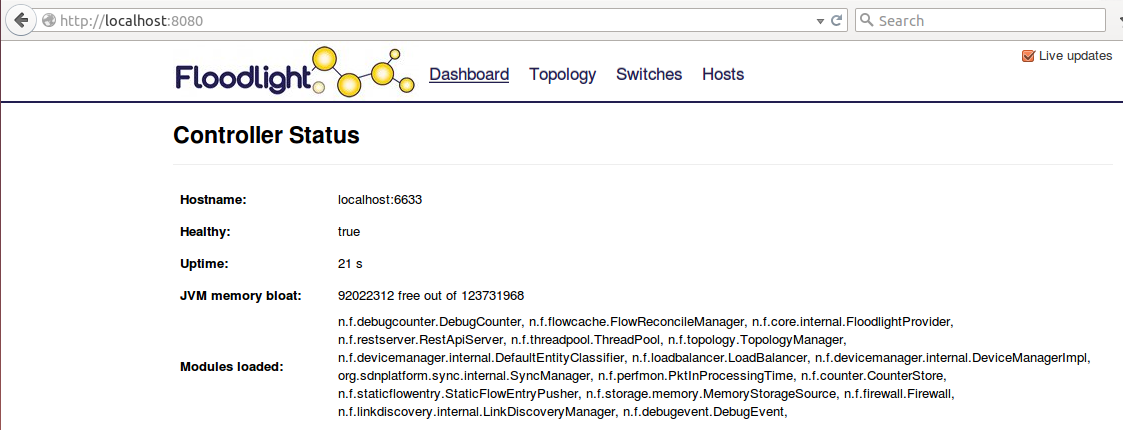
**Another is that so far is effective to download the 0.91 version, that is, no problem with the old version, download address**

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

[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-13.png)  
Then decompress and compile

ant

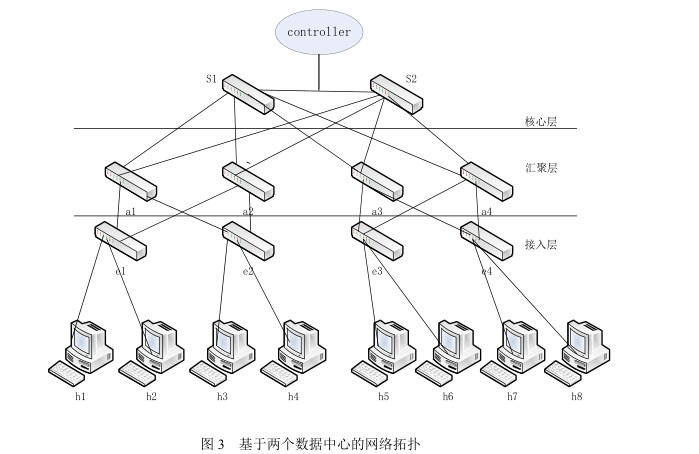
java -jar target/floodlight.jar

And then through http: // localhost: 8080 / ui / index.html visit floodlight management interface, found to be able to visit[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-14.png)

**3.Install mininet and connect to Floodlight**

**After installing the controller, we need to install mininet topology structure, the command line is as follows** sudo apt-get install mininet

Then need to build a network topology, on the data center network, for example, I built a script

fattree.py，  
[[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-15.png)](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-15.png)  
Upon completion, we need to use mininet to execute the script:

# sudo mn --custom /home/lyx/fattree.py --topo mytopo --controller=remote,ip=218.193.113.249,port=6633 --switch ovsk,protocols=OpenFlow10

According to the actual situation, in the floodlight whose ip is server's ip, add the protocols parameter specified OpenFlow protocol version. And then for the port, when the 6633 web interface does not respond, try to use with 6653, it is possible 6633 is Occupied, especially floodlight version1.2 (special reminder)  
Mn is the mininet boot command.

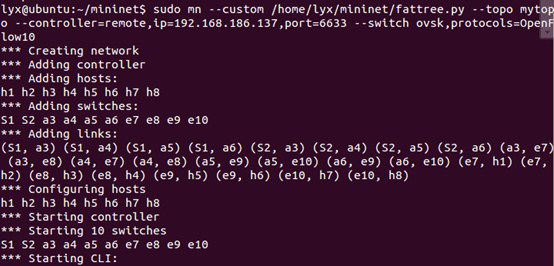
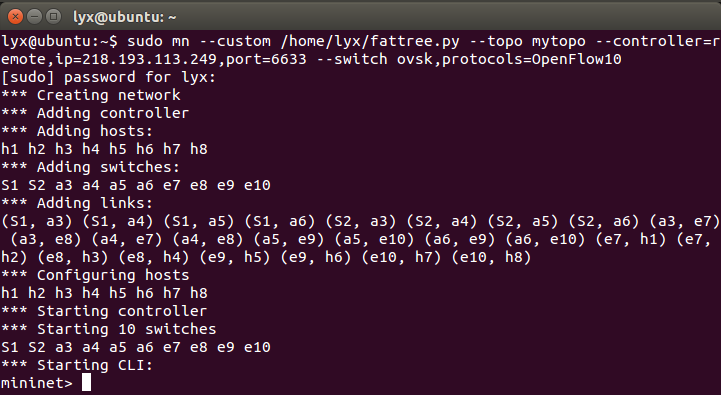
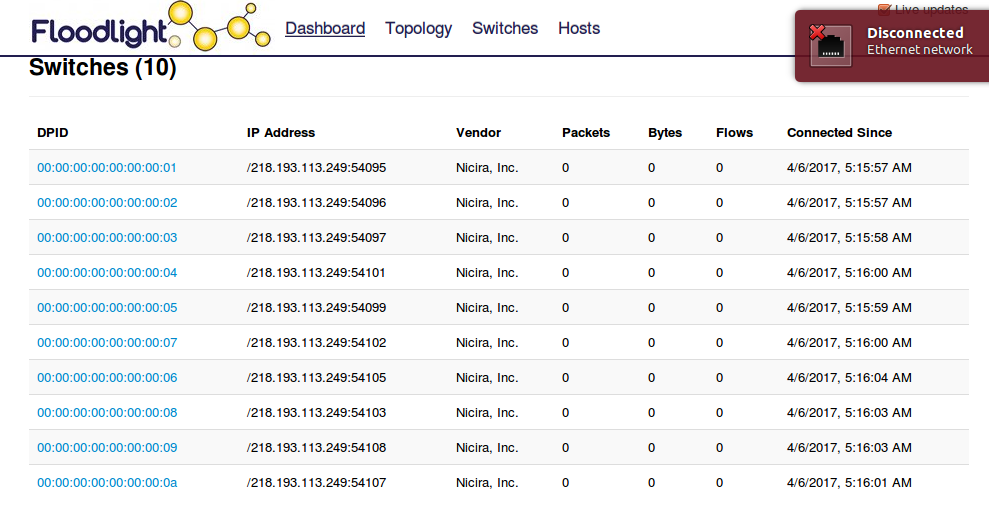
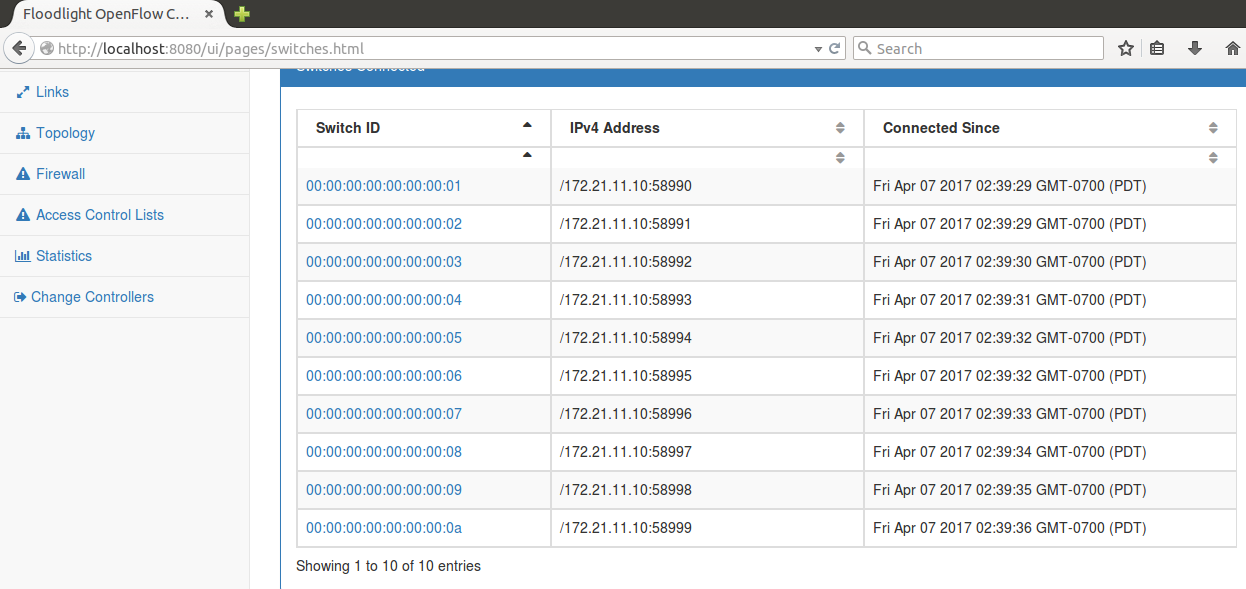
--mac Specifies the mac address sequence number of the virtual host, and if it don't have this parameter, it is randomly numbered.

--controller Specifies the of controller of the switch

--switch Specifies the type of virtual switch, ovsk represents the virtual switch for ovs Kernel mode

--custom Specifies a custom topology file

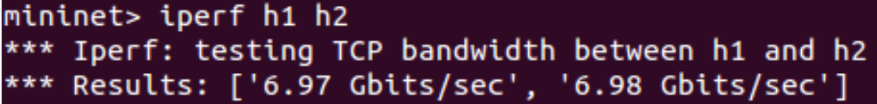
--topo Specifies the name of the loaded topology.

The implementation process is as follows:   
[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-16.png)  
[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-17.png)  
This is the 0.91 version  
[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-18.png)  
This is 1.2 version  
[](http://img1.sdnlab.com/wp-content/uploads/2017/05/Floodlight-mininet-fig-19.png)  
Claim:

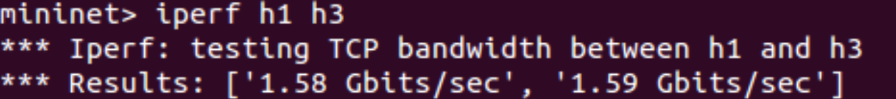
In the Mininet command line interface, the iperf command analyzes the bandwidth performance between h1 and h2, h1 and h3, h1, and h5 hosts (also by write testing script implement completely automated test), including the transmission rate and the acceptance rate.

Experimental steps:

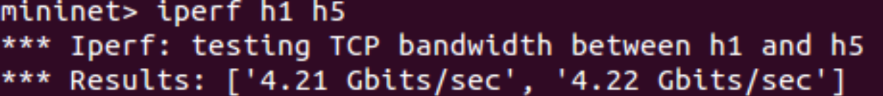
（1）Connectivity between host and host communication within the same switch and communication bandwidth test, and ping between h1 and h2.



（2）the same convergent switch between the different rack host test, between h1 and h3 ping operation.



1. the same core switches under different convergent switches under the host test, between h1 and h5 between the ping.Reference material



[http://www.sdnlab.com/2909.html](http://www.sdnlab.com/2909.html )  
[http://www.linuxidc.com/Linux/2015-01/112030.htm](http://www.linuxidc.com/Linux/2015-01/112030.htm )  
[https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/8650780/Floodlight+VM](https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/8650780/Floodlight+VM )  
<https://segmentfault.com/a/1190000003076028>