# Testbed Presentation

ARTUR BARUCHI

MARCO GOMES

ROGERIO IOPE

# Agenda

1. Introduction
2. Network Devices and Computational Resources
3. Architecture
4. Testbed Usage Scenarios
5. Testbed Simulation
   i. What, How and Why
   ii. Preliminaries Results
   iii. SC’16 Kytos Demo
6. OpenStack and SDN
7. Huawei and Openflow
8. Current Status
9. Next Steps

SPRACE

# Introduction

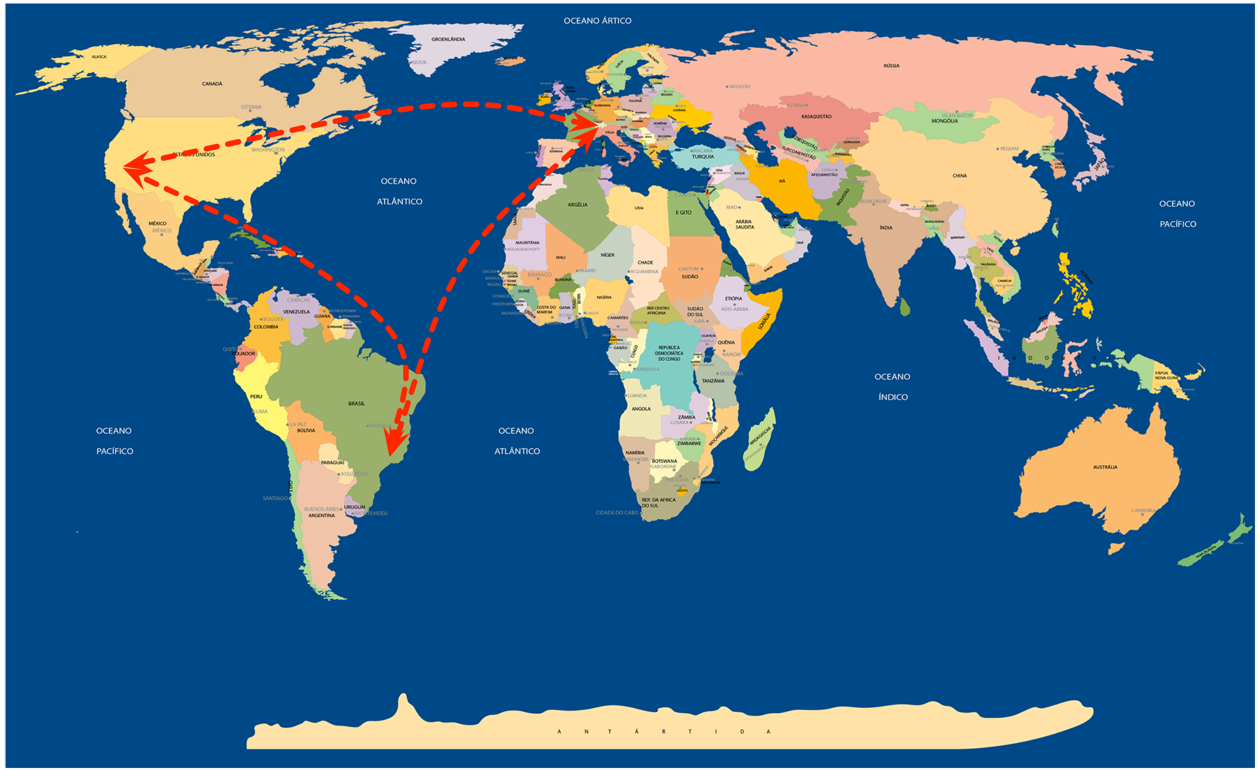
Geographically Distributed Environment to:

- Perform scientific research
- Proof of Concept (POC)
- Deploy and Evaluate new technologies (hardware and software)

# Introduction

Three sites

- North America: Caltech
- Europe: CERN
- South America: NCC (main site)



# Introduction

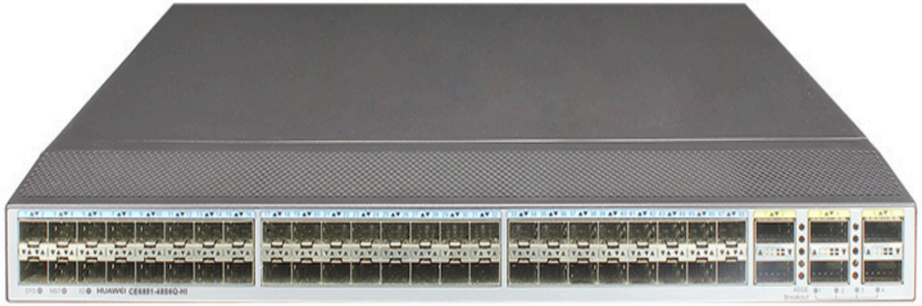
## Who is supporting us?

# Network Devices

## Huawei Switches

- Model: CE8860
  - 2x 100GbE
  - 24x 10G or 25GbE
  - 16x 40GbE
  - Total: 42 Ports
  - Supports SDN (OpenFlow > 1.3)

- Model: CE6851
  - 48x 10GbE
  - 5x 40GbE
  - Total: 53 Ports
  - Supports SDN (Openflow > 1.3)



# Computational Resources

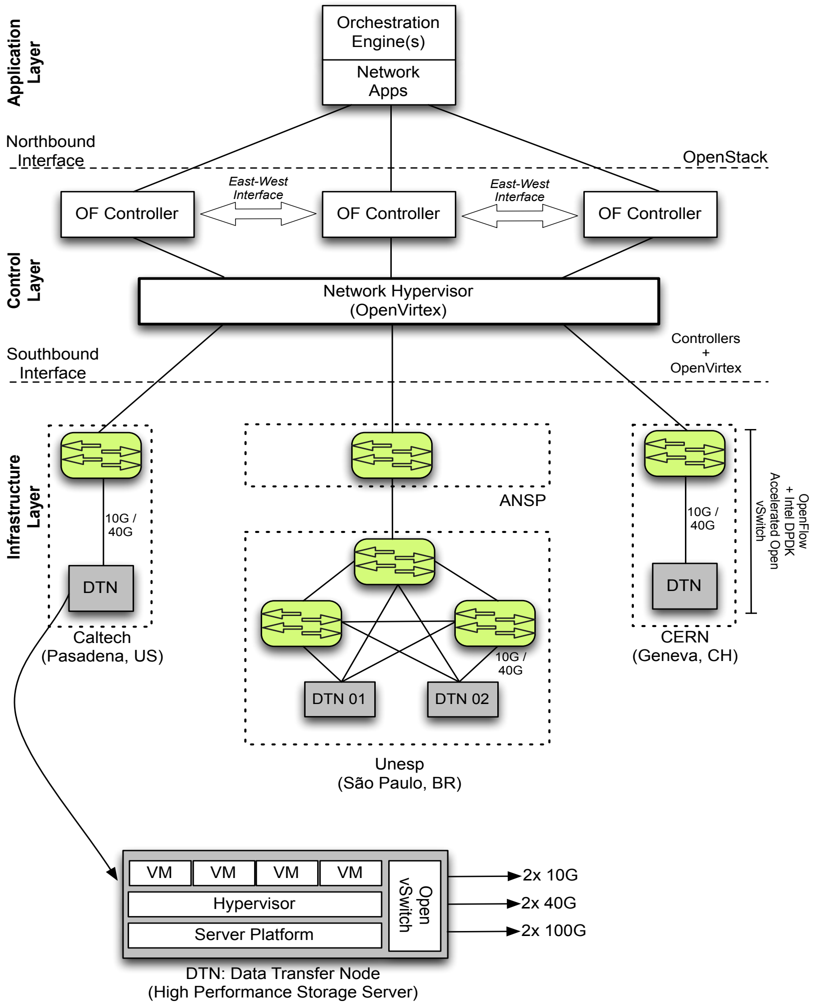
Huawei Servers (Data Transfer Nodes)

- Model: RH2288H V3
- 2x Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz
- Memory: 128Gb
- 2xSSD Disks in RAID 0 (240GB) and 4xDisks in RAID 1 (7.3TB)

Huawei Servers (Controllers)

- Model: RH2288 V3
- 2x Intel(R) Xeon(R) CPU E5-2695 v3 @ 2.30GHz
- Memory: 64Gb
- 2xSSD Disks in RAID 0 (240GB)

# Architecture



# Testbed Simulation

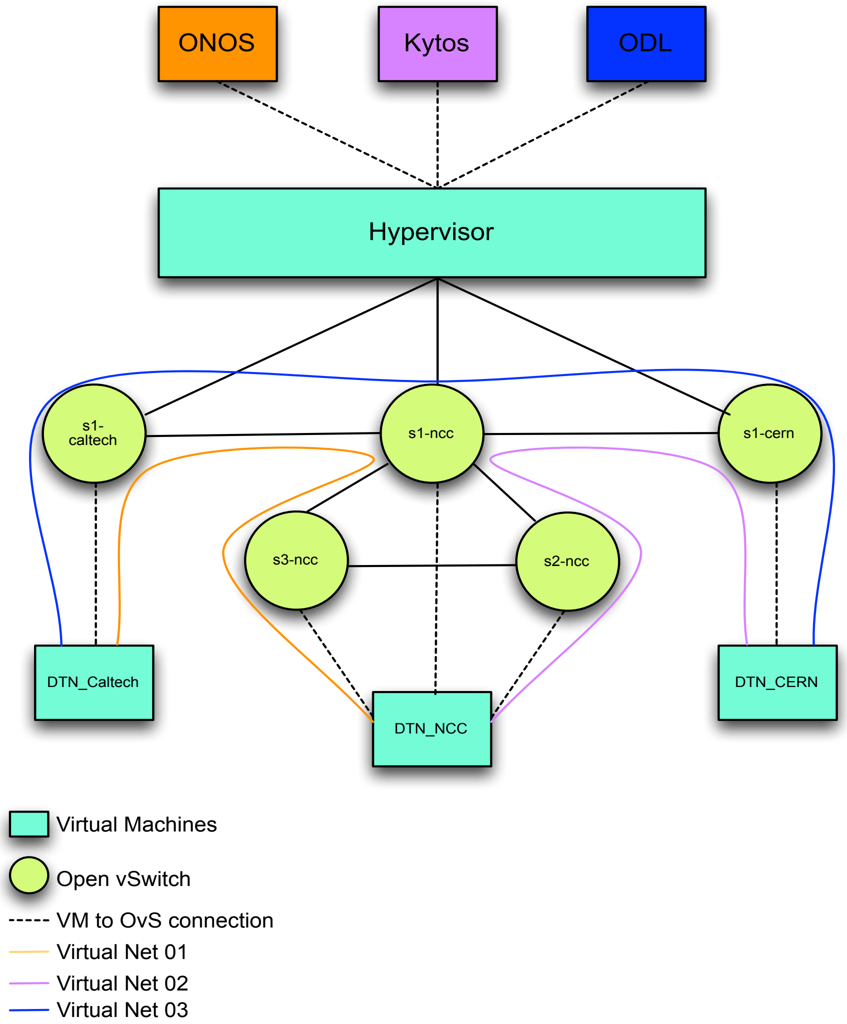
- What is it?
  - Reduction of our testbed that mimics our testbed architecture and topology

- How?
  - Using virtual machines (KVM) and Open vSwitch (Accelerated by Intel DDPK)

- Why?
  - Several usage scenarios, such as:

    - ❑ QA Environment
    - ❑ SW Evaluation
    - ❑ Topology Evaluation
    - ❑ POC
    - ❑ ETC…

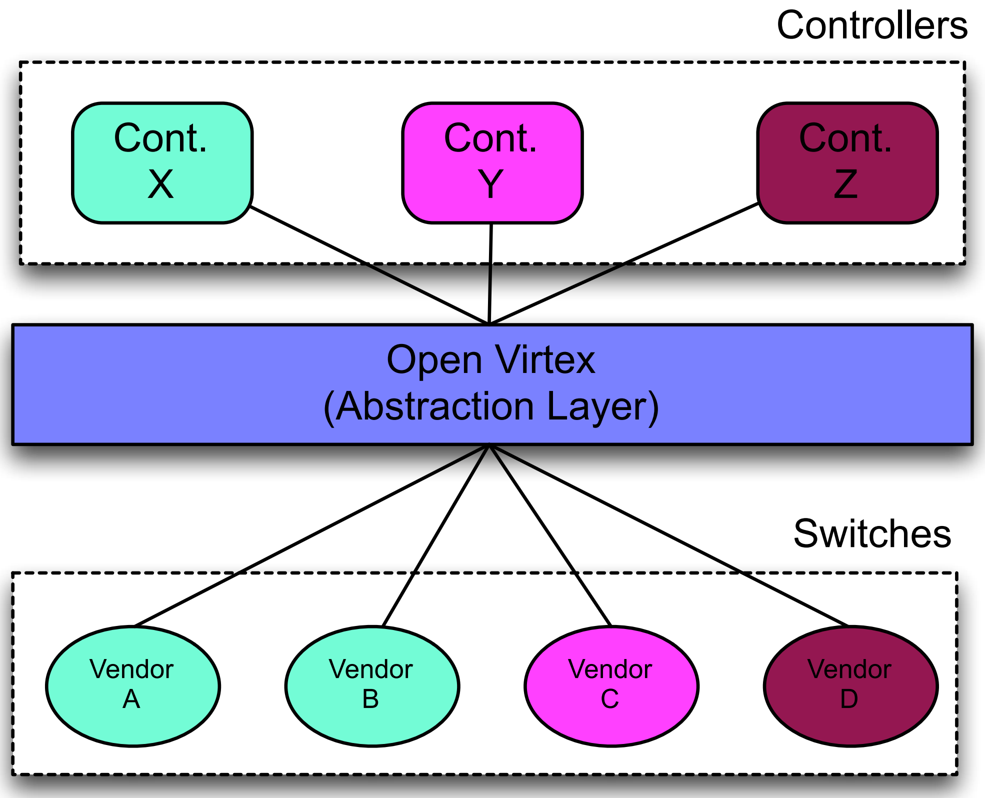
# Testbed Simulation



VM's Configuration:

- DTNs:
  - 4 vCPU
  - Memory: 10GB
- Controllers:
  - 1 vCPU
  - Memory: 2GB
- Hypervisor:
  - 2 vCPU
  - Memory: 5GB

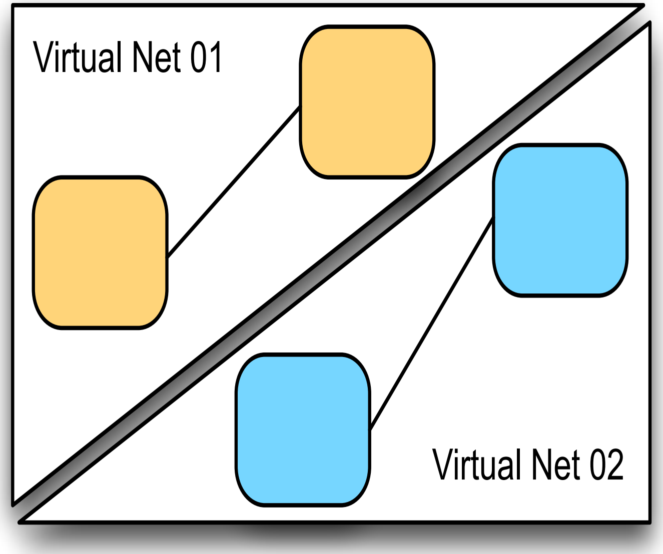
# Testbed Simulation



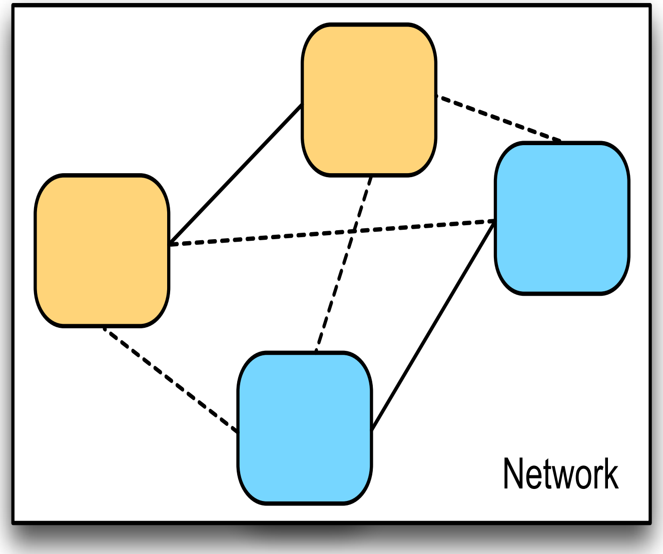
- Controller’s Point of View:
  - There are only switches of a kind

- Switch’s Point of View:
  - There is only one switch available

# OpenVirtex Vs FlowVisor



OpenVirtex



FlowVisor

- Network completely segmented (by sw port)
- Hosts inside VN01 can not reach VN02

- Network is segmented by packet type or protocol
- Hosts can reach each other

# Testbed Simulation

**Cria a rede, dizendo qual Controller será usado:**

```
  ovxctl.py -n createNetwork tcp:10.10.0.20:6633 10.0.0.0 16
```

**Cria os switches virtuais:**

```
  ovxctl.py -n createSwitch 1 00008664b7c93d40
  ovxctl.py -n createSwitch 1 0000ca63a25d5c41
  ovxctl.py -n createSwitch 1 00007a1088f88a46
```

**Cria as portas virtuais dos Switches:**

```
  ovxctl.py -n createPort 1 00008664b7c93d40 6
  ovxctl.py -n createPort 1 00008664b7c93d40 7
  ovxctl.py -n createPort 1 00:00:ca:63:a2:5d:5c:41 12
  ovxctl.py -n createPort 1 00:00:ca:63:a2:5d:5c:41 8
  ovxctl.py -n createPort 1 00:00:7a:10:88:f8:8a:46 13
  ovxctl.py -n createPort 1 00:00:7a:10:88:f8:8a:46 9
```

**Cria os links:**

```
 ovxctl.py -n connectLink 1 00:a4:23:05:00:00:00:01 2 00:a4:23:05:00:00:00:03 1 spf 1
  ovxctl.py -n connectLink 1 00:a4:23:05:00:00:00:03 2 00:a4:23:05:00:00:00:02 2 spf 1
```

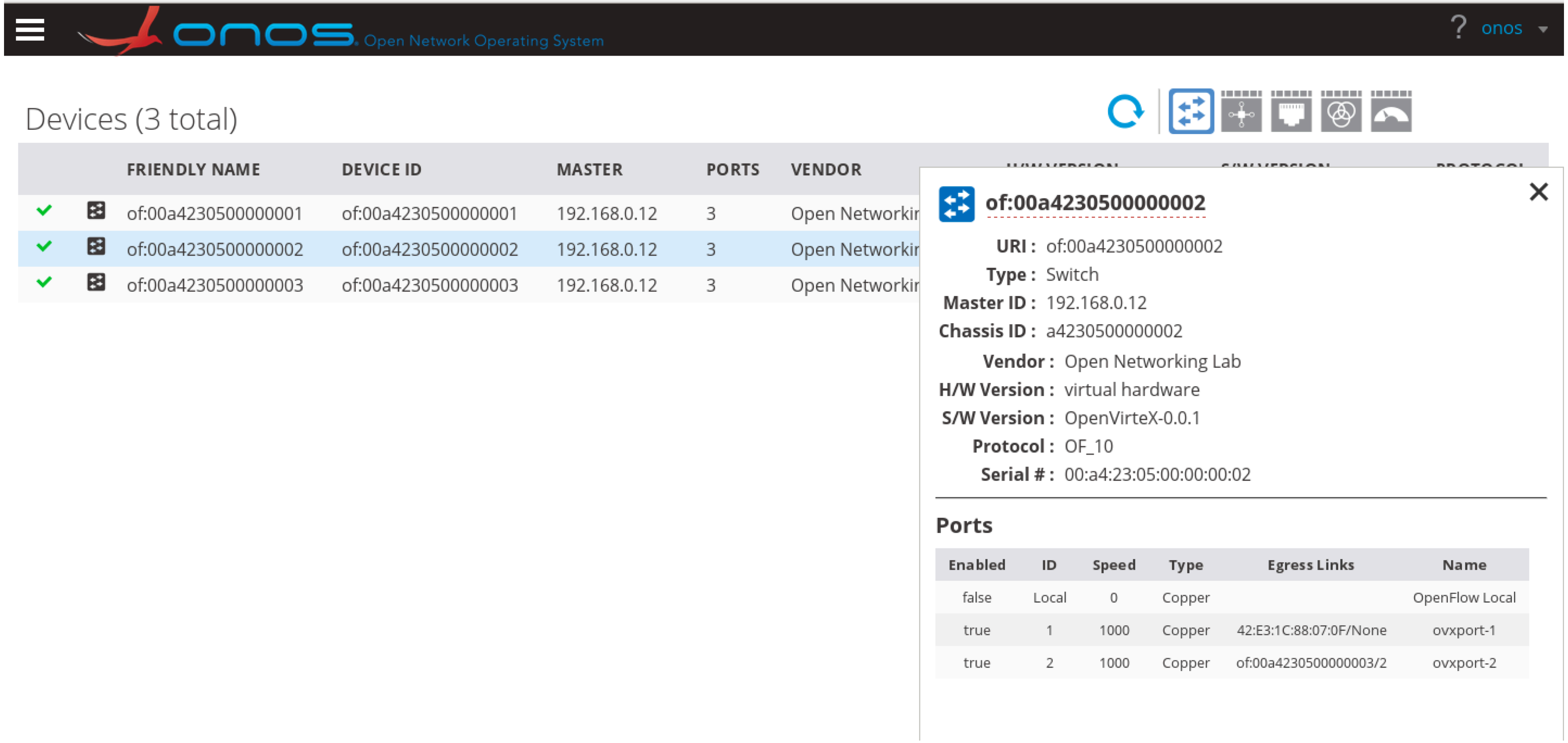
**Cria os hosts:**

```
  ovxctl.py -n connectHost 1 00:a4:23:05:00:00:00:01 1 36:CD:3F:36:71:92
  ovxctl.py -n connectHost 1 00:a4:23:05:00:00:00:02 1 42:e3:1c:88:07:0f
```

**Sobre a rede virtual:**

```
  ovxctl.py -n startNetwork 1
```

# Testbed Simulation



Devices (3 total)

|  |  | FRIENDLY NAME | DEVICE ID | MASTER | PORTS | VENDOR |
|---|---|---|---|---|---|---|
| ✓ |  | of:00a4230500000001 | of:00a4230500000001 | 192.168.0.12 | 3 | Open Networkin |
| ✓ |  | of:00a4230500000002 | of:00a4230500000002 | 192.168.0.12 | 3 | Open Networkin |
| ✓ |  | of:00a4230500000003 | of:00a4230500000003 | 192.168.0.12 | 3 | Open Networkin |

**of:00a4230500000002**

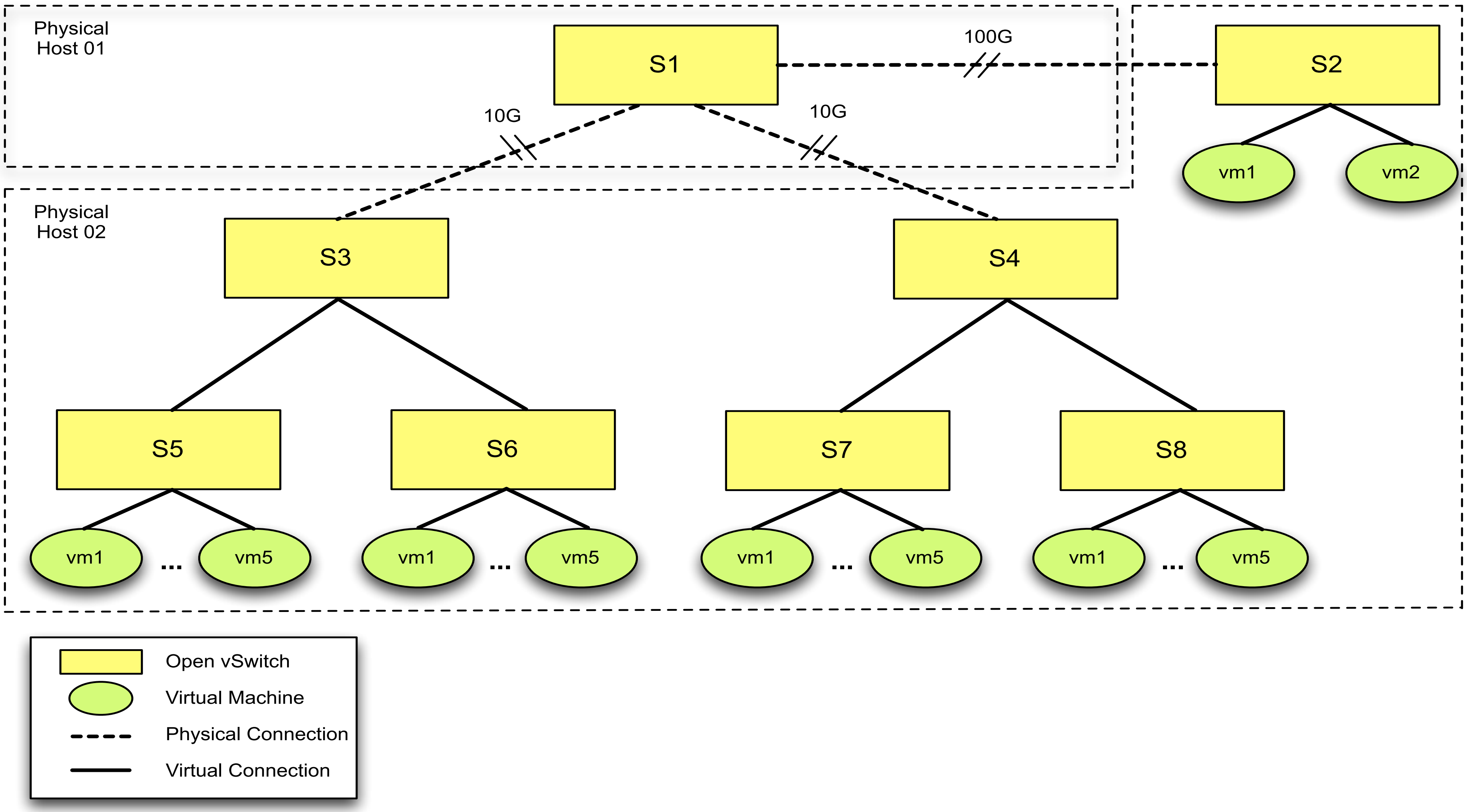
URI : of:00a4230500000002
Type : Switch
Master ID : 192.168.0.12
Chassis ID : a4230500000002
Vendor : Open Networking Lab
H/W Version : virtual hardware
S/W Version : OpenVirteX-0.0.1
Protocol : OF_10
Serial # : 00:a4:23:05:00:00:00:02

**Ports**

| Enabled | ID | Speed | Type | Egress Links | Name |
|---|---|---|---|---|---|
| false | Local | 0 | Copper |  | OpenFlow Local |
| true | 1 | 1000 | Copper | 42:E3:1C:88:07:0F/None | ovxport-1 |
| true | 2 | 1000 | Copper | of:00a4230500000003/2 | ovxport-2 |

Open Virtex GUI

# Kytos Controller Demo



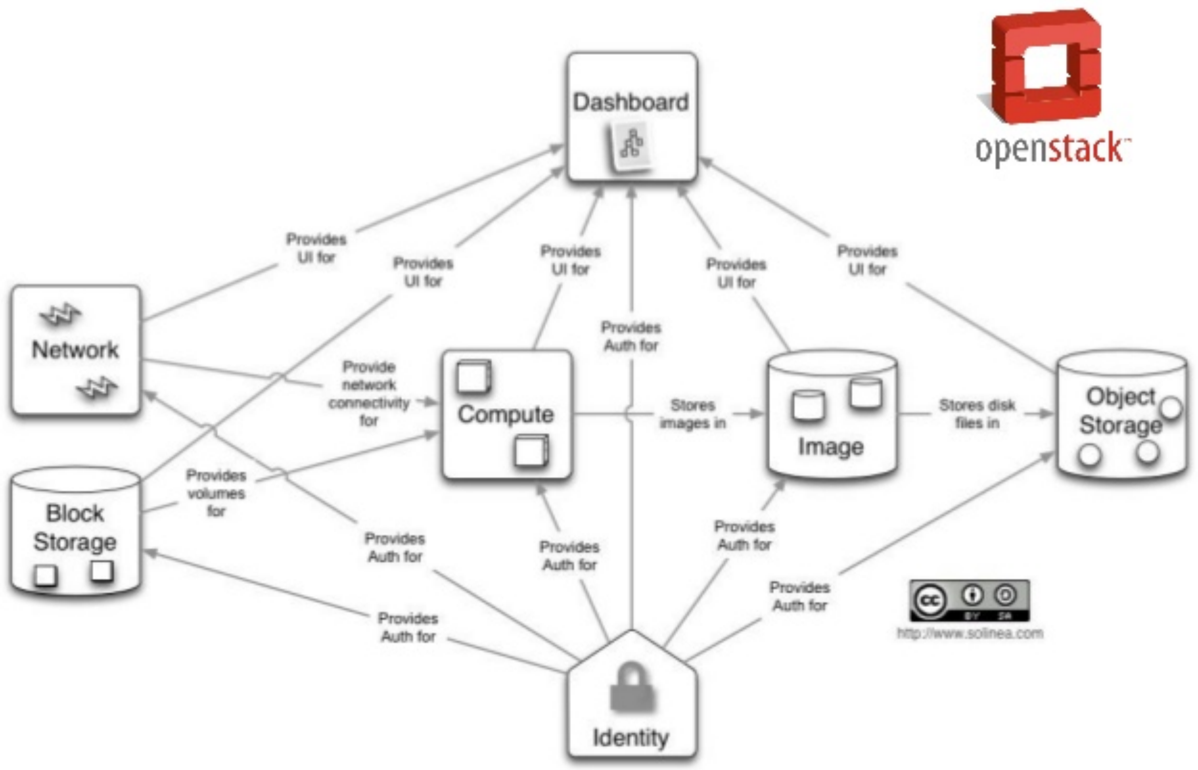
# OpenStack

Responsible for manage datacenter resources (compute nodes, storage and networking) to create a cloud environment.

Modular software concept – each module manages an specific kind of resource

- Keystone – identity
- Glance – image repository
- Nova – compute management
- Neutron – networking management
- Horizon – system dashboard
- Cinder – Block Storage provider
- and others…

# OpenStack Modules



Font: openstack.org

Management of network resources

Creation of network objects

- Network
- subnets
- switches
- routers
- Firewalls
- VPNs
- Load Balancers

Networking provided by agent plugins

- Native Linux networking mechanisms
- External devices
- SDN Controllers

# OpenStack – Neutron (Service and Components)

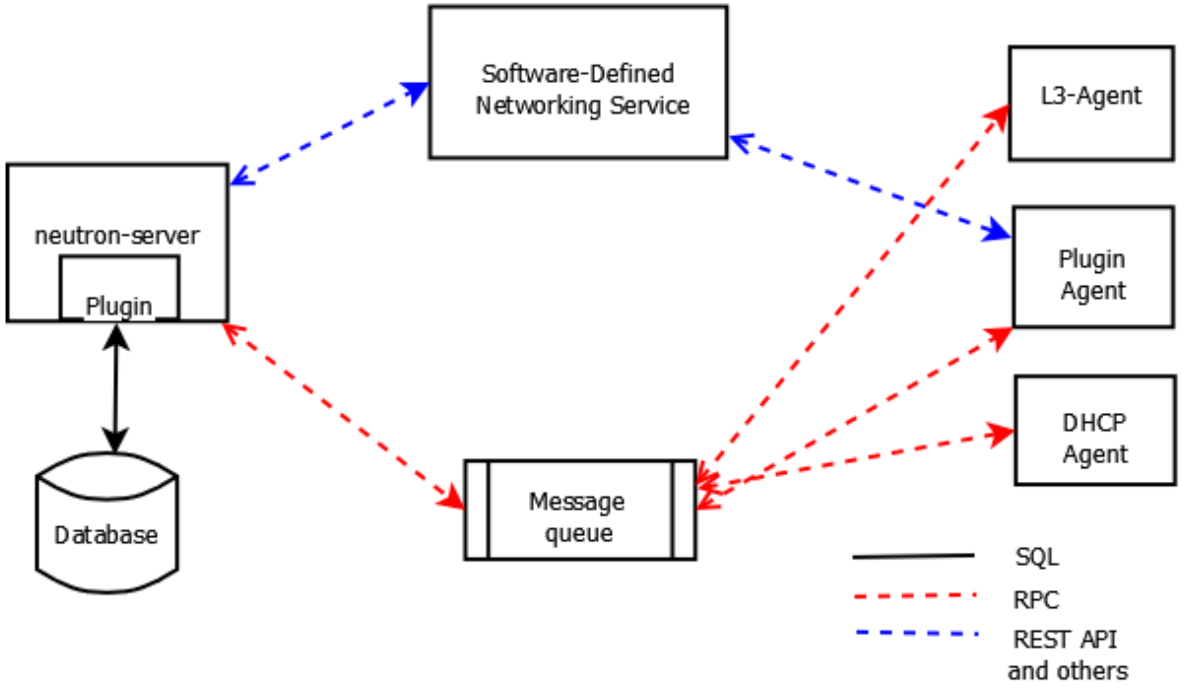
Neutron Server

- Plugins
- Agents
  - Layer 2 (Ethernet and Switching)
  - Layer 3 (IP and routing)
- Services
  - Routing services
  - VPNaaS
  - LBaaS
  - FWaaS

# OpenStack – Neutron

Integrate networking services with SDN controllers to provide virtual networking to the cloud infrastructure

- Integration between Neutron and SDN controller software through REST API

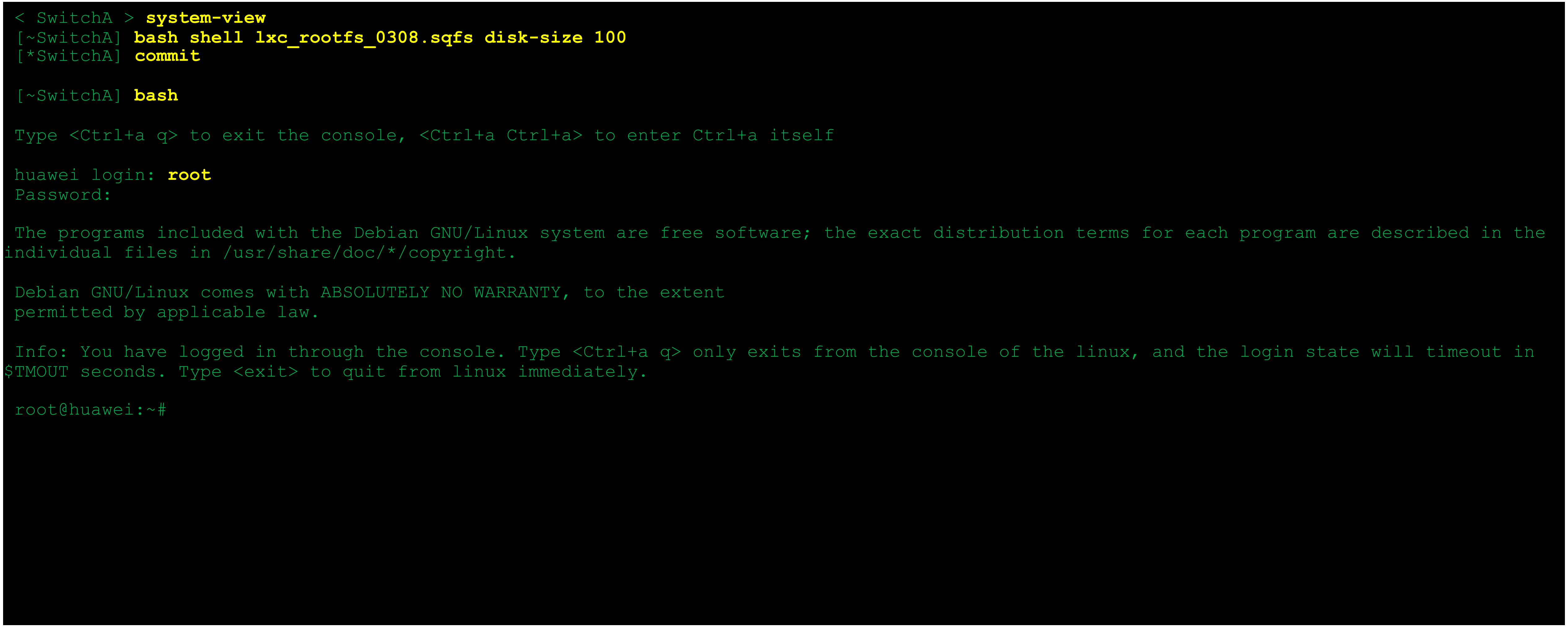


# Huawei and Openflow

Considerations:

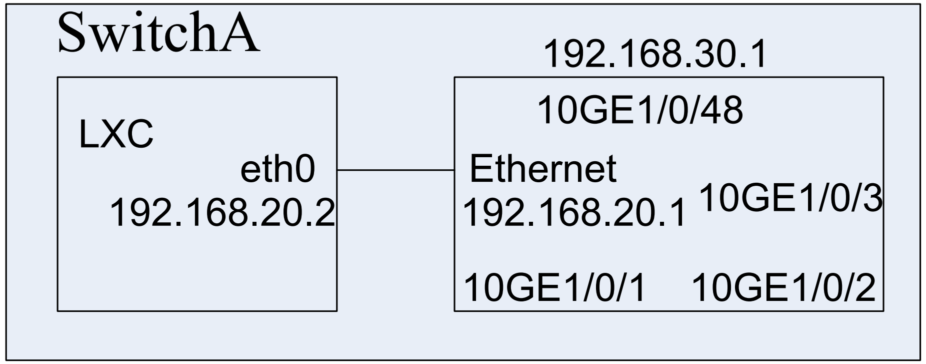
- Huawei Switches support version 1.3
- Necessary to create a *docker* inside the Switch to support controllers
- When using Huawei’s controller (Agile) *docker* is not necessary

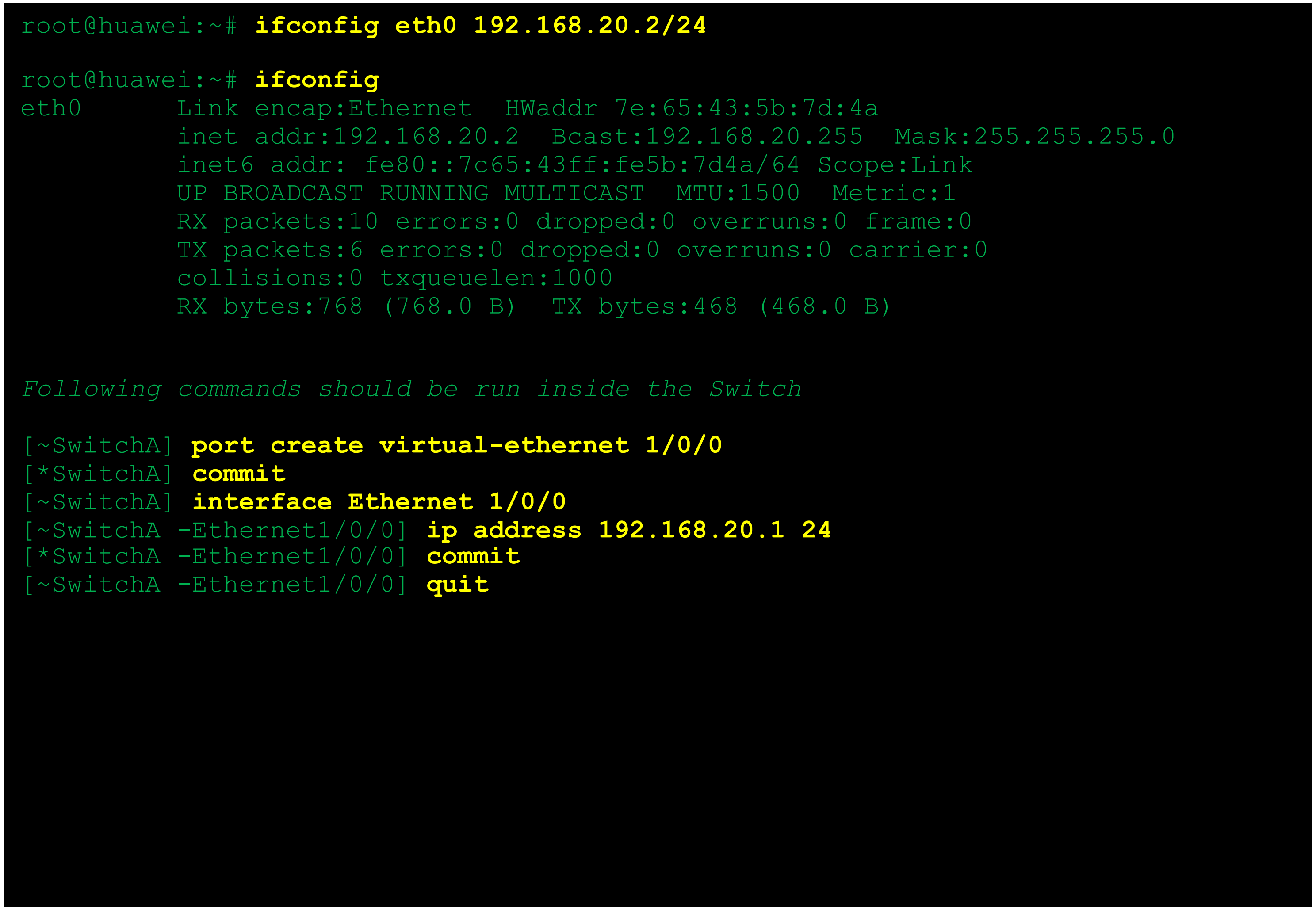
# Huawei and Openflow



```
< SwitchA > system-view
[~SwitchA] bash shell lxc_rootfs_0308.sqfs disk-size 100
[*SwitchA] commit

[~SwitchA] bash

Type <Ctrl+a q> to exit the console, <Ctrl+a Ctrl+a> to enter Ctrl+a itself

huawei login: root
Password:

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Info: You have logged in through the console. Type <Ctrl+a q> only exits from the console of the linux, and the login state will timeout in
$TMOUT seconds. Type <exit> to quit from linux immediately.

root@huawei:~#
```

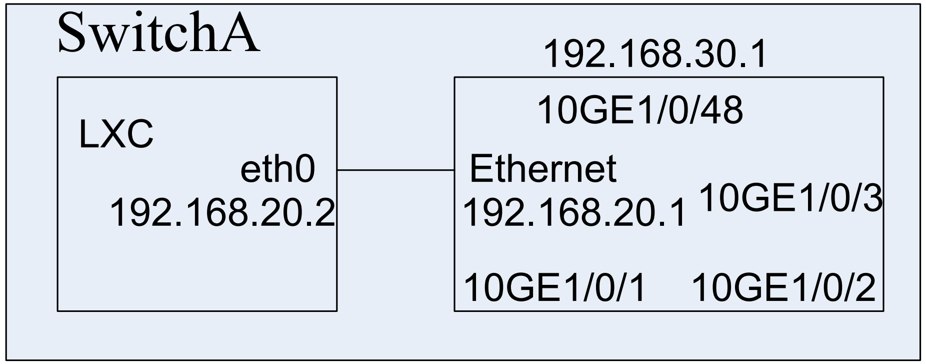
# Huawei and Openflow

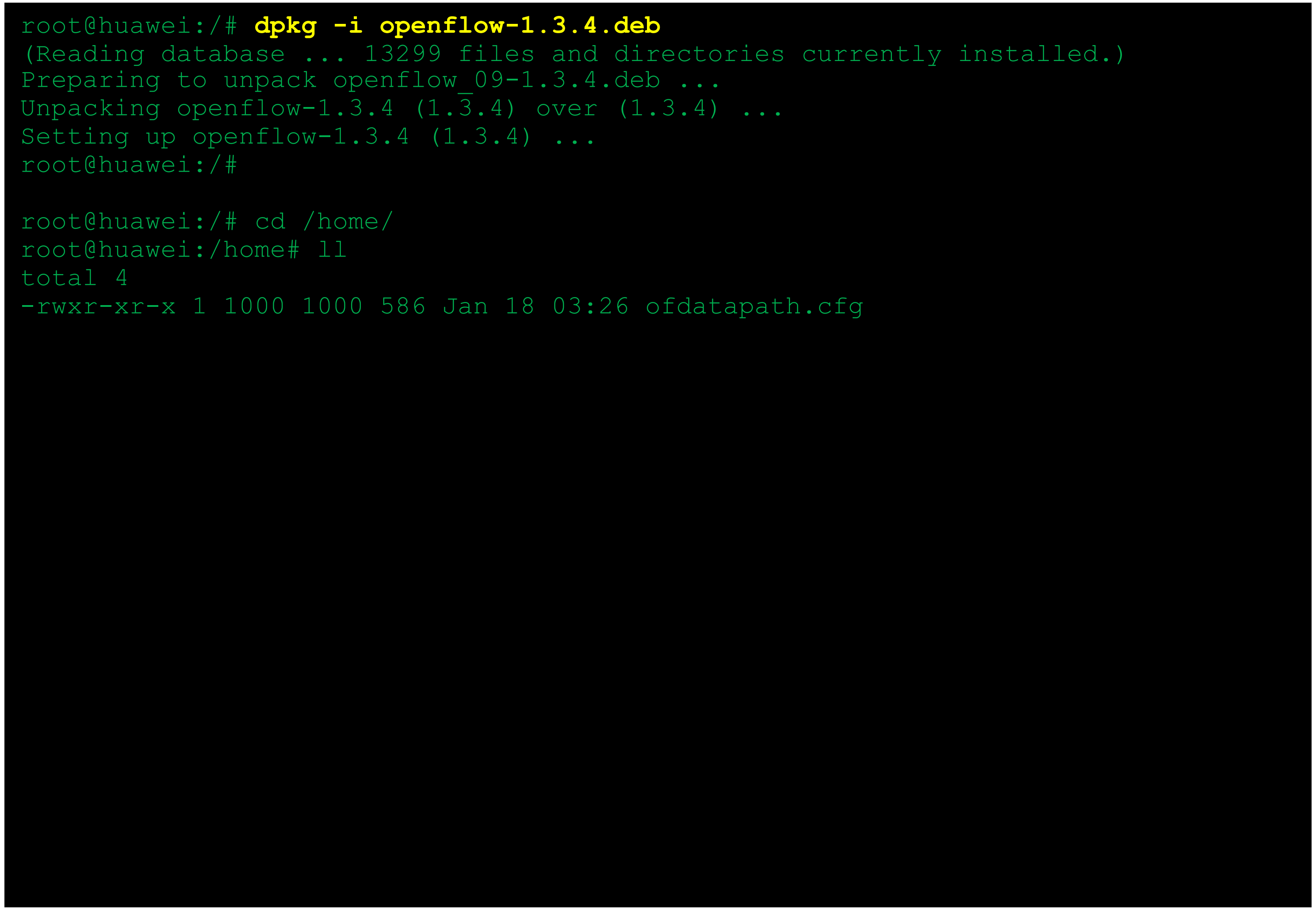




```
root@huawei:~# ifconfig eth0 192.168.20.2/24

root@huawei:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 7e:65:43:5b:7d:4a
          inet addr:192.168.20.2  Bcast:192.168.20.255  Mask:255.255.255.0
          inet6 addr: fe80::7c65:43ff:fe5b:7d4a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:768 (768.0 B)  TX bytes:468 (468.0 B)


Following commands should be run inside the Switch

[~SwitchA] port create virtual-ethernet 1/0/0
[*SwitchA] commit
[~SwitchA] interface Ethernet 1/0/0
[~SwitchA -Ethernet1/0/0] ip address 192.168.20.1 24
[*SwitchA -Ethernet1/0/0] commit
[~SwitchA -Ethernet1/0/0] quit
```

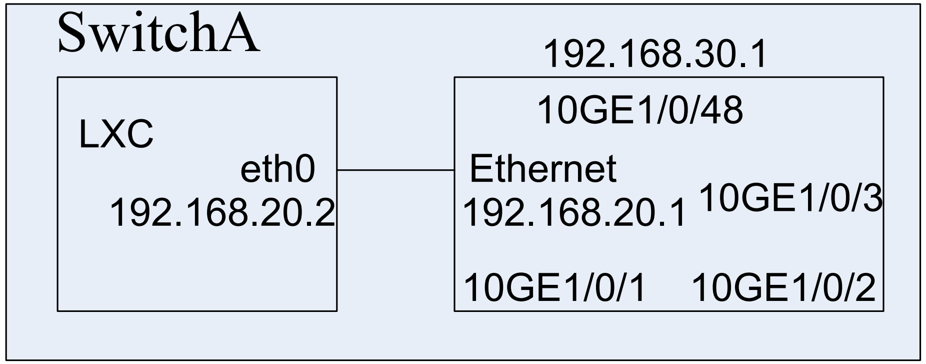
# Huawei and Openflow

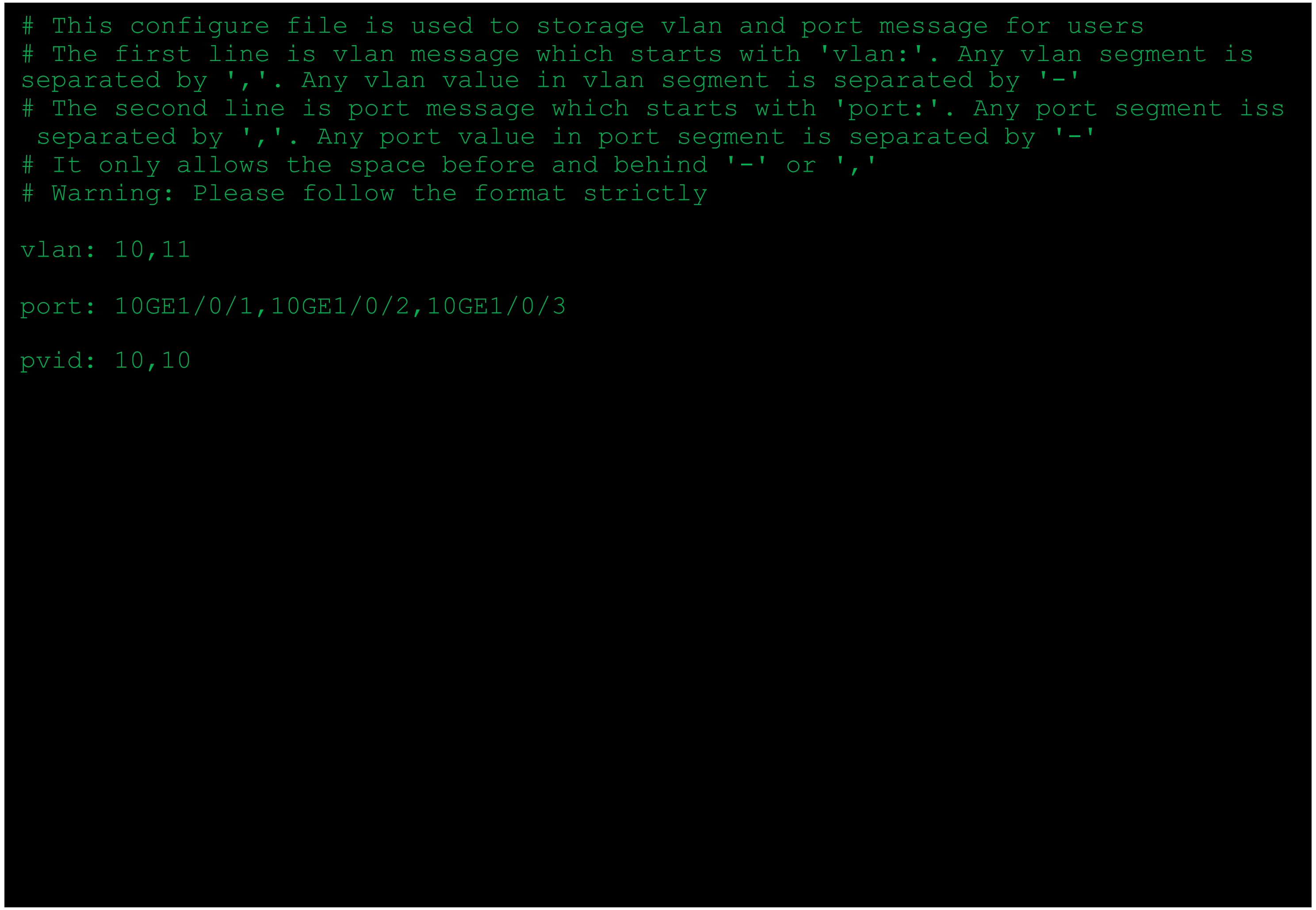




```
root@huawei:/# dpkg -i openflow-1.3.4.deb
(Reading database ... 13299 files and directories currently installed.)
Preparing to unpack openflow_09-1.3.4.deb ...
Unpacking openflow-1.3.4 (1.3.4) over (1.3.4) ...
Setting up openflow-1.3.4 (1.3.4) ...
root@huawei:/#

root@huawei:/# cd /home/
root@huawei:/home# ll
total 4
-rwxr-xr-x 1 1000 1000 586 Jan 18 03:26 ofdatapath.cfg
```

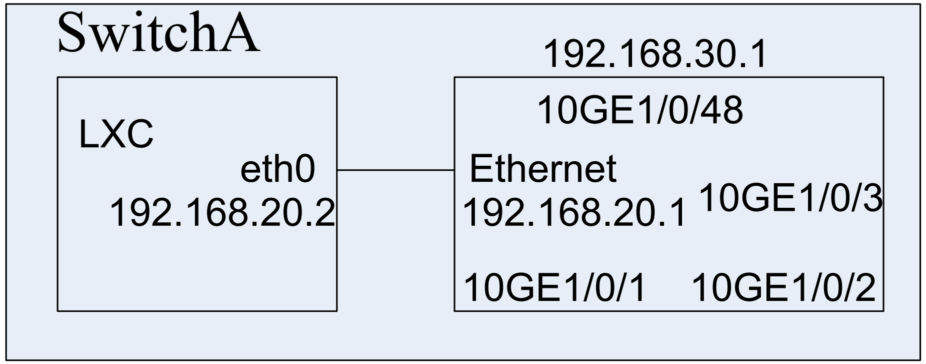
# Huawei and Openflow

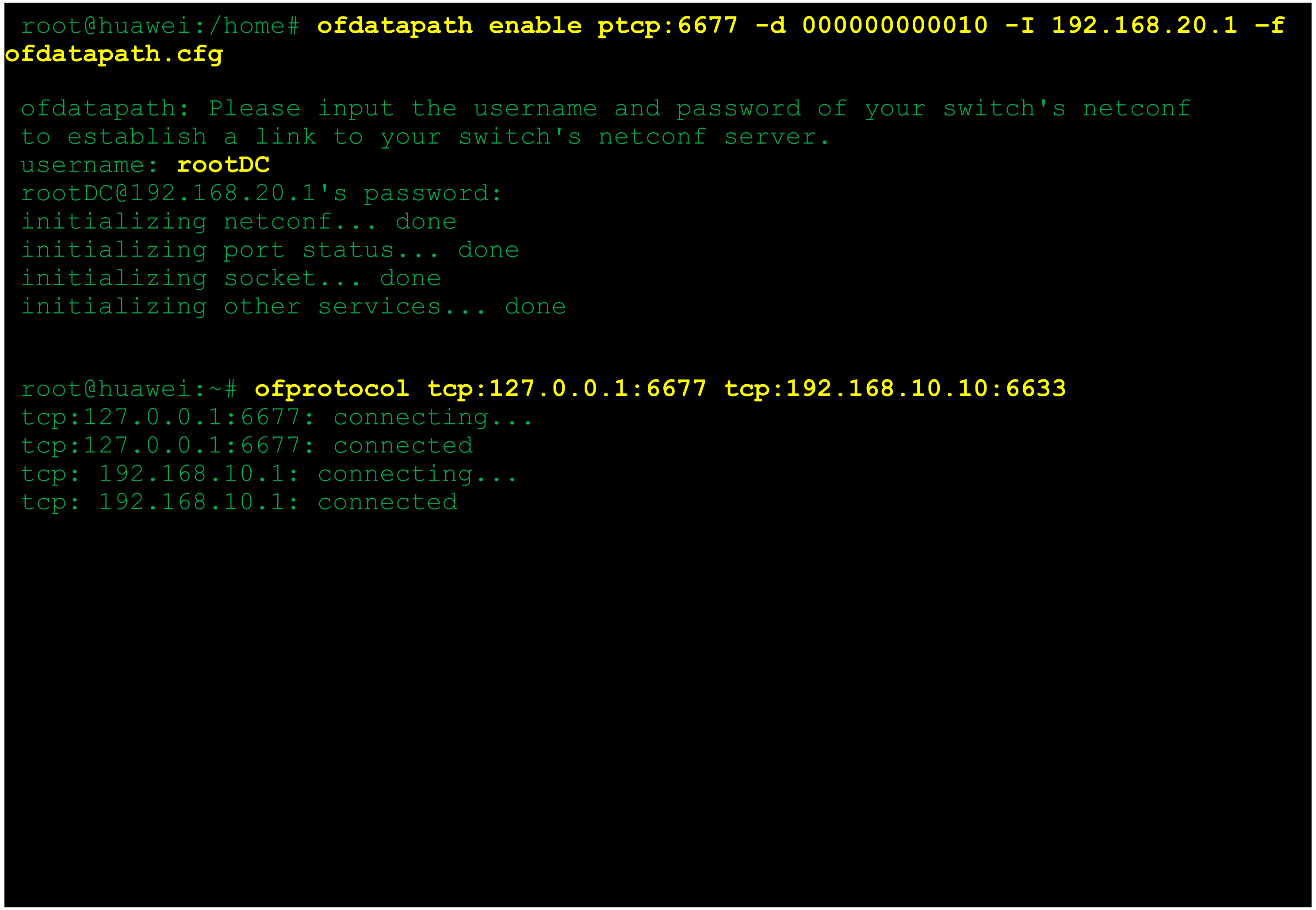




```
# This configure file is used to storage vlan and port message for users
# The first line is vlan message which starts with 'vlan:'. Any vlan segment is
separated by ','. Any vlan value in vlan segment is separated by '-'
# The second line is port message which starts with 'port:'. Any port segment iss
 separated by ','. Any port value in port segment is separated by '-'
# It only allows the space before and behind '-' or ','
# Warning: Please follow the format strictly

vlan: 10,11

port: 10GE1/0/1,10GE1/0/2,10GE1/0/3

pvid: 10,10
```

# Huawei and Openflow





```
root@huawei:/home# ofdatapath enable ptcp:6677 -d 000000000010 -I 192.168.20.1 -f
ofdatapath.cfg

ofdatapath: Please input the username and password of your switch's netconf
to establish a link to your switch's netconf server.
username: rootDC
rootDC@192.168.20.1's password:
initializing netconf... done
initializing port status... done
initializing socket... done
initializing other services... done


root@huawei:~# ofprotocol tcp:127.0.0.1:6677 tcp:192.168.10.10:6633
tcp:127.0.0.1:6677: connecting...
tcp:127.0.0.1:6677: connected
tcp: 192.168.10.1: connecting...
tcp: 192.168.10.1: connected
```

# Current Status

❑ NCC Side machines are already deployed (using CentOS 7)

❑Switches are installed and updated (used during SC’16)

❑ Image installation and scripts are ready

❑ Software and tools are under test in our testbed

# Next Steps

- ❑ Deploy CERN and CALTECH equipment
- ❑ Send equipment to US and Europe
- ❑ Define final architecture (OpenStack + OpenVirtex + XOS)
- ❑ Write and evaluate some scenarios to run in our Testbed

# Questions and Answers