**I. Software-defined Networking**

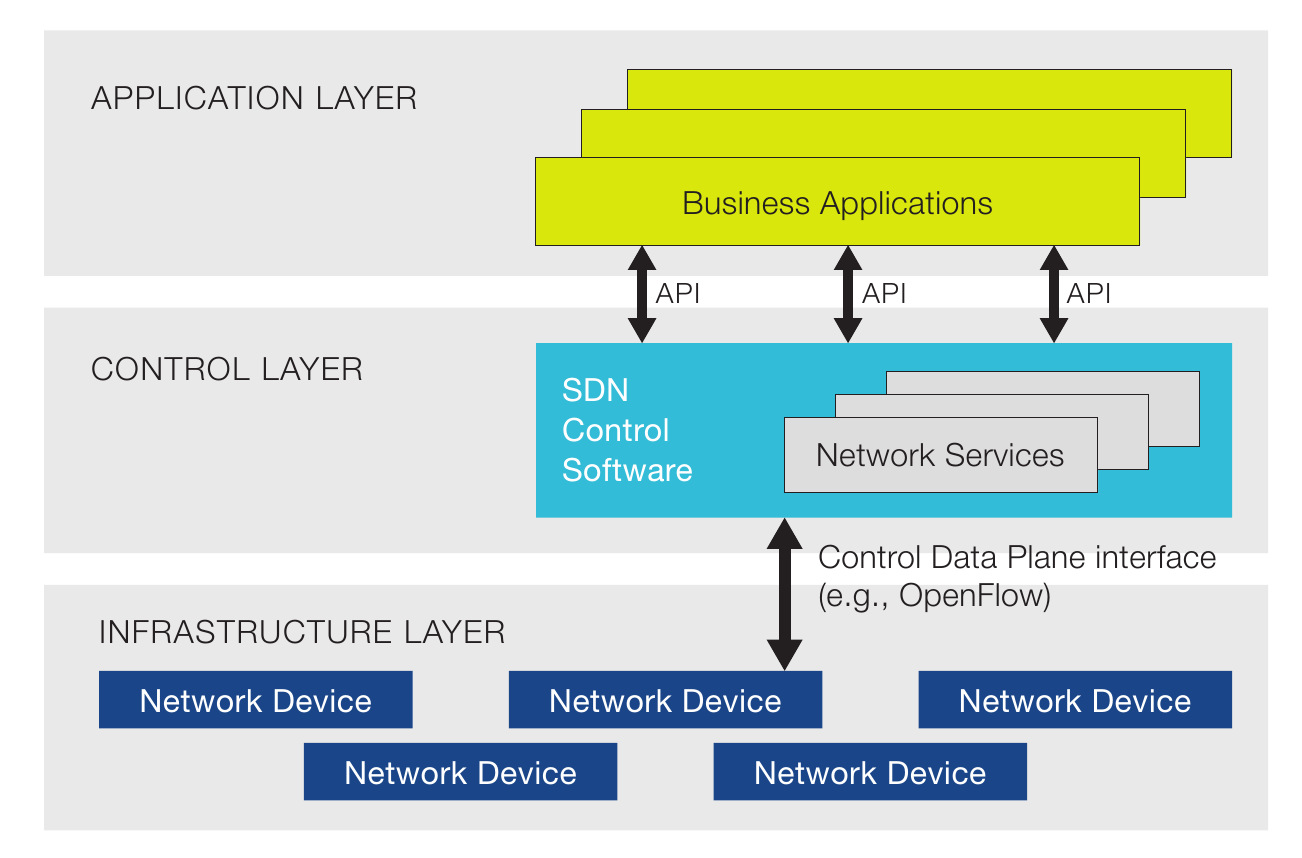
**1. Sofware-defined Networking (SDN)**

**1.1 Định nghĩa**

- Software-Defined Networking (SDN) là kiến trúc mạng linh động, dễ dàng quản lý, hiệu quả về chi phí, có khả năng đáp ứng cao, lý tưởng cho các ứng dụng đòi hỏi băng thông lớn và có tính năng động cao. Kiến trúc này tách biệt hai cơ chế đang tồn tại trong kiến trúc mạng hiện tại là cơ chế điều khiển (control plane) và cơ chế chuyển tiếp (dataplane), cho phép phần điều khiển có khả năng lập trình được và hạ tầng bên dưới trở nên trừu tượng với các ứng dụng và các dịch vụ mạng. Các đặc tính trong kiến trúc của SDN:

* **Khả năng lập trình trực tiếp:**Việc điều khiển network được lập trình trực tiếp bởi nó đã được tách biệt với các chức năng chuyển tiếp
* **Nhanh chóng:**Việc tách biệt các chức năng điều khiển và chức năng chuyển tiếp cho phép các nhà quản trị linh hoạt trong việc điều chỉnh luồng lưu lượng của network khi có yêu cầu thay đổi
* **Quản lý tập trung:**Việc điều khiển tập trung được thực hiện bởi SDN Controller(một phần mềm) duy trì khung nhìn toàn cục về mạng
* **Việc cấu hình lập trình được:**SDN cho phép người quản lý mạng cấu hình, quản lý, thiết lập bảo mật, tối ưu hóa tài nguyên mạng nhanh chóng nhờ có các chương trình hỗ trợ SDN đã tự động hóa, những chương trình đó hoàn toàn có thể tự lập trình được mà không phụ thuộc vào phần mềm.
* **Cung cấp các tiêu chuẩn mở:**Khi triển khai thông qua các tiêu chuân mở, SDN đã đơn gian hóa việc thiết kế mạng và vận hành bởi vì các chỉ dẫn được cung cấp bởi SDN controller thay vì các giao thức hay các thiết bị chuyên biệt của các nhà cung cấp.

**1.2 Kiến trúc hệ thống của SDN**



- Theo Open Networking Foundation, kiến trúc của SDN bao gồm ba lớp tách biệt truy cập thông qua các APIs mở:

* Lớp ứng dụng: bao gồm các ứng dụng của người dùng cuối sử dụng các dịch vụ truyền thông qua SDN. Ranh giới giao tiếp giữa lớp ứng dụng và lớp điều khiển được thực hiện bởi northbound API.
* Lớp điều khiển (SDN control plance): cung cấp chức năng quản lý tập trung làm nhiệm vụ quản lý việc chuyển tiếp trong mạng thông các các open interfaces. Các chức năng này bao gồm: định tuyến, khai báo tên, chính sách và thực hiện kiểm tra vấn đề bảo mật.
* Lớp cơ sở hạ tầng (SDN data plane): bao gồm các thiết bị mạng cung cấp chức năng chuyển mạch và forward các gói tin.

- SDN Controller định nghĩa nên các luồng dữ liệu trong SDN Data plane. Mỗi luồng đi qua mạng trước hết phải lấy thông tin về quyền hạn trên controller, để chứng thực các giao tiếp được phép thực hiện quy định bởi network policy. Nếu controller cho phép một luồng đi qua mạng, nó sẽ tính toán đường đi cho luồng dữ liệu này, và bổ sung một entry tương ứng với flow đó trong mỗi switch trên đường đi đã tính toán. Nhờ có controller xử lý rất nhiều các chức năng phức tạp, các switch đơn giản hơn trong việc quản lý các flow tables mà các entry của các table này chỉ được tính toán bởi controller. Việc truyền thông giữa controller và các switch sử dụng một giao thức chuẩn (Openflow) và southbound API.

- Kiến trúc SDN nổi bật về sự linh hoạt, nó có thể vận hành với nhiều loại switch và nhiều lớp giao thức khác nhau. Các SDN Controller và các switch có thể được triển khai cho các Switch layer 2, các router, chuyển mạch giao vận hoặc chuyển mạch và định tuyến lớp ứng dụng.

**II. OpenFlow Protocol**

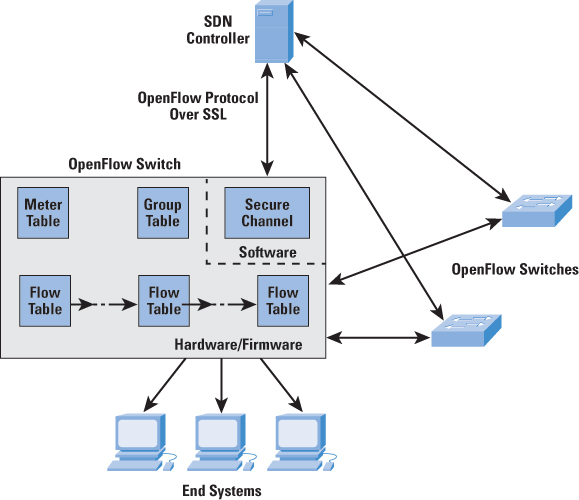
**2. Open Flow**

- Khái niệm SDN đặt ra 2 vấn đề khi triển khai thực tế:

* Cần phần có một kiến trúc logic chung cho tất cả các switch, router và các thiết bị mạng khác được quản lý bởi SDN Controller. Kiến trúc này có thể được triển khai bằng nhiều cách khác nhau trên các thiết bị của các nhà cung cấp khác nhau và phụ thuộc vào nhiều loại thiết bị mạng, miễn là SDN controller thấy được chức năng chuyển mạch thống nhất
* Một giao thức chuẩn, bảo mật để giao tiếp giữa SDN controller và các thiết bị mạng

- OpenFlow được đưa ra để giao quyết cả hai vấn để đó.

**2.1 Kiến trúc logical của OpenFlow Switch**



- Mỗi switch kết nối với các OpenFlow switch khác và kết nối với các thiết bị của người dùng cuối là nguồn và đích của luồng dữ liệu. Một OpenFlow switch bao gồm ít nhất ba thành phần:

* **Flow table:**có trách nhiệm "nói chuyện" với switch để chỉ ra rằng phải xử lý flow ra sao, mỗi hành động tương ứng với 1 flow-entry.
* **Secure Channel:**kết nối switch với controller sử dụng giao thức OpenFlow chạy qua Secure Sockets Layer (SSL), để gửi các commands và các packets.
* **OpenFlow Protocol**

**Flow table**

Có ba loại tập hợp các flow tables:

* Một flow table sẽ ghép các gói tin tới với một flow nhất định và chỉ định các chức năng được thực hiện trên các gói tin đó. Có thể có nhiều flow tables vận hành trong một pipeline.
* Một flow table có để chuyển một luồng vào một Group Table, tại đó có thể kích hoạt cùng một lúc nhiều hành động ảnh hưởng tới một hoặc nhiều flow.
* Một Meter Table có thể kích hoạt nhiều hành động liên quan tới hiệu năng trên một flow.

**Flow-entry**

Mỗi flow-entry trong flow table có một hành động tương ứng với nó và gồm 3 trường:

* **Packet header định nghĩa nên flow**
* **Hành động (Action) định nghĩa cách mà gói tin sẽ được xử lý**
* **Thống kê (Statistics) giữ thông tin theo dõi về số lượng gói tin và kích thước theo bytes của mỗi flow, thời gian kể từ lúc gói tin cuối đưa vào flow (nhằm mục đích loại bỏ các flow đã ngừng hoạt động).**

Mỗi flow-entry có một hành động tương ứng với nó, và có ba loại hành động cơ bản:

* Chuyển các gói tin của một flow tới port (hoặc các port) đã chỉ định. Điều này cho phép các gói tin được định tuyến qua mạng.
* Đóng gói và chuyển tiếp các gói tin của flow tới controller. Gói tin sẽ được đưa tới Secure Channel, tại đó nó được đóng gói và gửi tới controller. Điển hình như gói tin đầu tiên của mỗi flow mới sẽ được gửi tới controller để được quyết định xem liệu flow có được đưa vào trong flow table hay không.
* Hủy các gói tin của flow. Hành động này được sử dụng nhằm mục đích bảo mật, như tấn công từ chối dịch vụ (DoS).

[**2.2. OpenFlow Protocol**](https://github.com/hocchudong/thuctap032016/blob/master/ThaiPH/SDN/ThaiPH_SDN_OpenFlow.md#ofp)

- Giao thức OpenFlow mô tả bản tin trao đổi giữa OpenFlow Controller và một OpenFlow switch. Giao thức này được triển khai trên Secure Socket Layer (SSL) hoặc Transport Layer Security (TLS), cung cấp kênh OpenFlow bảo mật. Giao thức OpenFlow cho phép controller thực hiện các thao tác bổ sung, cập nhật và xóa các hành động vào các flow entry trong các flow tables. Nó hỗ trợ 3 loại bản tin:

* **Controller-to-Switch:** Các bản tin này được khởi tạo bởi controller, có thể yêu cầu hoặc không yêu cầu phản hồi từ switch. Lớp bản tin này cho phép controller quản lý các trạng thái logic của switch, bao gồm cấu hình và chi tiết về các flow entry trong flow table hoặc group table. Ngoài ra, nó cũng bao gồm các bản tin Packet-out (bản tin đi tới một port nhất định trên switch). Bản tin này được sử dụng khi switch gửi một packet tới controller và controller sẽ quyết định không hủy gói nhưng sẽ chuyển hướng tới một output port của switch.
* **Asynchronous:** Các bản tin loại này được gửi mà không có yêu cầu từ controller. Lớp các bản tin này bao gồm nhiều bản tin trạng thái khác nhau gửi tới controller. Ngoài ra lớp này còn bao gồm các bản tin packet-in - các packet không phù hợp với flow table nào, và như vậy các bản tin đó sẽ được gửi tới controller để xử lý.
* **Symmetric:** Các bản tin này được gửi đi mà không có yêu cầu cả từ controller và switch, bao gồm các bản tin:
  + **Hello:** là bản tin trao đổi giữa switch và controller dựa trên kết nối đã được đã được thiết lập.
  + **Echo:** các bản tin <b>echo request/reply</b> có thể được gửi từ switch hoặc controller, và chúng sẽ phải trả về một bản tin echo reply.
  + **Experimenter:** Các bản tin cho các chức năng bổ sung

- Giao thức OpenFlow cho phép controller quản lý cấu trúc logic của switch mà không liên quan tới chi tiết việc switch triển khai kiến trúc OpenFlow ra sao.

**III. OpenDaylight SDN Controller**

**1. OpenDaylight SDN Controller**

- Dự ánOpenDaylight là một dự án phần mềm mã nguồn mở được tổ chức The Linux Foundation thành lập. Mục tiêu của dự án là để thúc đẩy mạng lưới định nghĩa phần mềm (SDN) và ảo hóa các chức năng mạng (NFV). Phần mềm được viết bằng ngôn ngữ lập trình Java.

- OpenDaylight sử dụng giao thức mở cung cấp khả năng kiểm soát tập trung, có khả năng lập trình được và theo dõi các thiết bị mạng. Giống như nhiều SDN Controllers khác, OpenDaylight hỗ trợ OpenFlow, cũng như cung cấp các giải pháp mạng khác sẵn sàng để cài đặt khi có yêu cầu.

- OpenDaylight cung cấp giao diện cho phép kết nối các thiết bị mạng nhanh chóng và thông minh để tối ưu hiệu năng mạng

- OpenDaylight Controller cung cấp northbound APIs, được sử dụng bởi các ứng dụng. Các ứng dụng này sử dụng controller để thu thập thông tin về mạng, chạy các thuật toán để kiểm soát, phân tích, sau đó sử dụng OpenDaylight Controller tạo các rules mới cho mạng.

- OpenDaylight Controller viết bằng ngôn ngữ Java, có nghĩa là có thể sử dụng OpenDaylight Controller trên bất kì môi trường nào hỗ trợ Java. Tuy nhiên để đạt hiệu năng tốt nhất, OpenDaylight nên chạy trên môi trường Linux hỗ trợ JVM tối thiểu 1.7.

**2. Mininet Emulator**

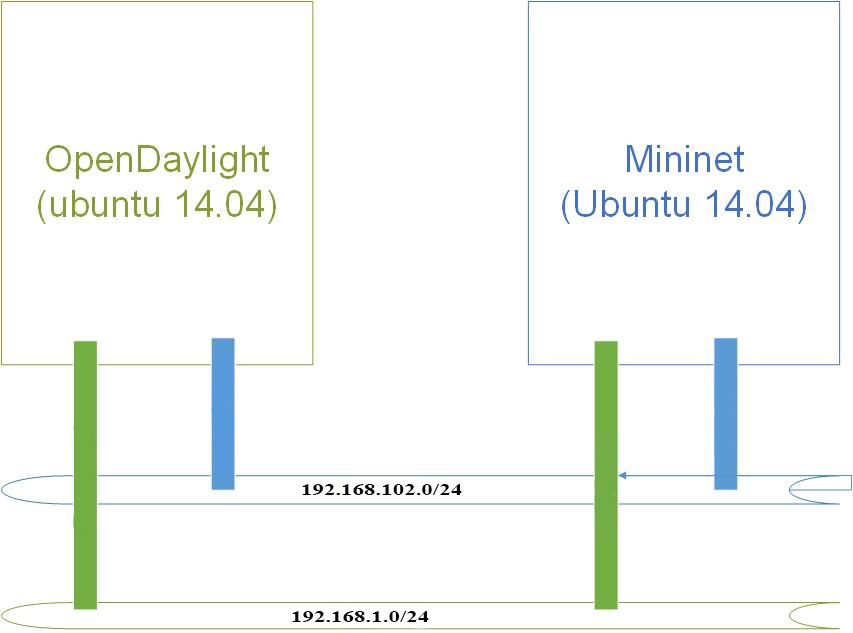
- Mininet là một công cụ giả lập mạng, bao gồm tập hợp các hosts đầu cuối, các switches, routers và các liên kết trên một Linux kernel. Mininet sử dụng công nghệ ảo hóa (ở mức đơn giản) để tạo nên hệ thống mạng hoàn chỉnh, chạy chung trên cùng một kernel, hệ thống và user code.

- Các host ảo, switch, liên kết và các controller trên mininet là các thực thể thực sự, được giả lập dưới dạng phần mềm thay vì phần cứng. Một host mininet có thể thực hiện ssh vào đó, chạy bất kì phần mềm nào đã cài trên hệ thống linux (môi trường mà mininet đang chạy). Các phần mềm này có thể gửi gói tin thông các ethernet interface của mininet với tốc độ liên kết và trễ đặt trước.

- Mininet cho phép tạo topo mạng nhanh chóng, tùy chỉnh được topo mạng, chạy được các phần mềm thực sự như web servers, TCP monitoring, Wireshark; tùy chỉnh được việc chuyển tiếp gói tin. Mininet cũng dễ dàng sử dụng và không yêu cầu cấu hình đặc biệt gì về phần cứng để chạy: mininet có thể cài trên laptop, server, VM, cloud (linux).

**IV. Dựng lab SDN sử dụng OpenDaylight và Mininet**

**1. Topology**

****



Chuẩn bị 2 máy ảo (chạy trên VMWare Workstation) cài ubuntu server 14.04, mỗi máy có hai card mạng như sau:

* **Card NAT:**Thuộc dải *192.168.1.0/24* (dải NAT hay VMNet8 của VMWare)
* **Card Host Only:**Thuộc dải *192.168.102.0/24* (ở đây là dải VMNet1 chế độ Host Only)

**2. Cài đặt OpenDaylight Controller và Mininet**

**2.1 Cài đặt mininet**

- Đảm bảo cấu hình mạng tương tự như sau:

mininet@mininet:~$ landscape-sysinfo

System load: 0.02 Processes: 362

Usage of /: 8.7% of 19.07GB Users logged in: 1

Memory usage: 28% IP address for eth0: 192.168.1.136

Swap usage: 0% IP address for eth1: 192.168.102.139

Graph this data and manage this system at:

<https://landscape.canonical.com/>

- Cài đặt mininet, sử dụng một trong hai cách sau:

* Tải Mininet VM image đã cài sẵn Mininet và các thành phần cần thiết rồi cấu hình lại card mạng như trên. Có thể tải Mininet VM <https://github.com/mininet/mininet/wiki/Mininet-VM-Images>
* Cài đặt mininet manual, thực hiện các lệnh sau để cài đặt mininet:

# Lệnh này cài đặt một số tập tin lõi cho mininet

sudo apt-get install mininet

# Bước tiếp theo để “kill any controller” mà mininet có thể đã kích hoạt

sudo mn -c

# Cài đặt git để mininet có thể tải được xuống từ Github và tải mininet

sudo apt-get install git

git clone git://github.com/mininet/mininet

# Chuyển mininet sang phiên bản mới nhất

cd mininet

git tag # list available versions

git check out -b “version”

# Cài đặt các phần tử mà mininet cung cấp để kết nối với controller và chạy được wireshark

mininet/util/install.h -a

**2.2 Cài đặt OpenDaylight Controller**

- Cài đặt java và cấu hình biến môi trường JAVA\_HOME:

sudo add-apt-repository ppa:webupd8team/java -y

sudo apt-get update

sudo apt-get install oracle-java8-installer

# Cài đặt Maven để build ODL

sudo mkdir -p /usr/local/apache-maven

wget <http://ftp.wayne.edu/apache/maven/maven-3/3.3.3/binaries/apache-maven-3.3.3-bin.tar.gz>

sudo mv apache-maven-3.3.3-bin.tar.gz /usr/local/apache-maven

sudo tar -xzvf /usr/local/apache-maven/apache-maven-3.3.3-bin.tar.gz -C /usr/local/apache-maven/

sudo update-alternatives --install /usr/bin/mvn mvn /usr/local/apache-maven/apache-maven-3.3.3/bin/mvn 1

sudo update-alternatives --config mvn

sudo apt-get install vim

vim ~/.bashrc

# thêm vào ~/.bashrc

export M2\_HOME=/usr/local/apache-maven/apache-maven-3.3.3

export MAVEN\_OPTS="-Xms256m -Xmx512m"

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle

source ~/.bashrc

- Tải file nén zip Opendaylight bản “Boron” từ trang chủ

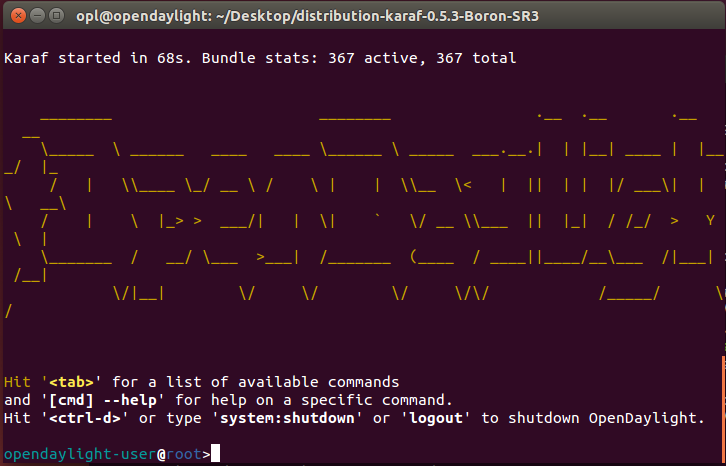
<https://www.opendaylight.org/downloads>

# Giải nén chạy Opendaylight và cài đặt feauture cần thiết

Unzip distribution-karaf-0.5.3-Boron-SR3.tar.gz

Cd Desktop/distribution-karaf-0.5.3-Boron-SR3/

./bin/karaf



# Sau khi khỏi xong như hình trên thì cài đặt các feature cần thiết như sau:

feature:install odl-restconf odl-l2switch-switch-all odl-mdsal-all odl-dlux-all

# Tạo topo và kiểm tra thông tin flow trên OpenDaylight

chạy lệnh sau tạo topo đơn giản:

sudo mn --topo tree,3 --mac --controller=remote,ip=192.168.1.144,port=6633 --switch ovs,protocols=OpenFlow13

# ping kiểm tra kết nối trên mininet

Pingall

Kết quả:

mininet> pingall

\*\*\* Ping: testing ping reachability

h1 -> h2 h3

h2 -> h1 h3

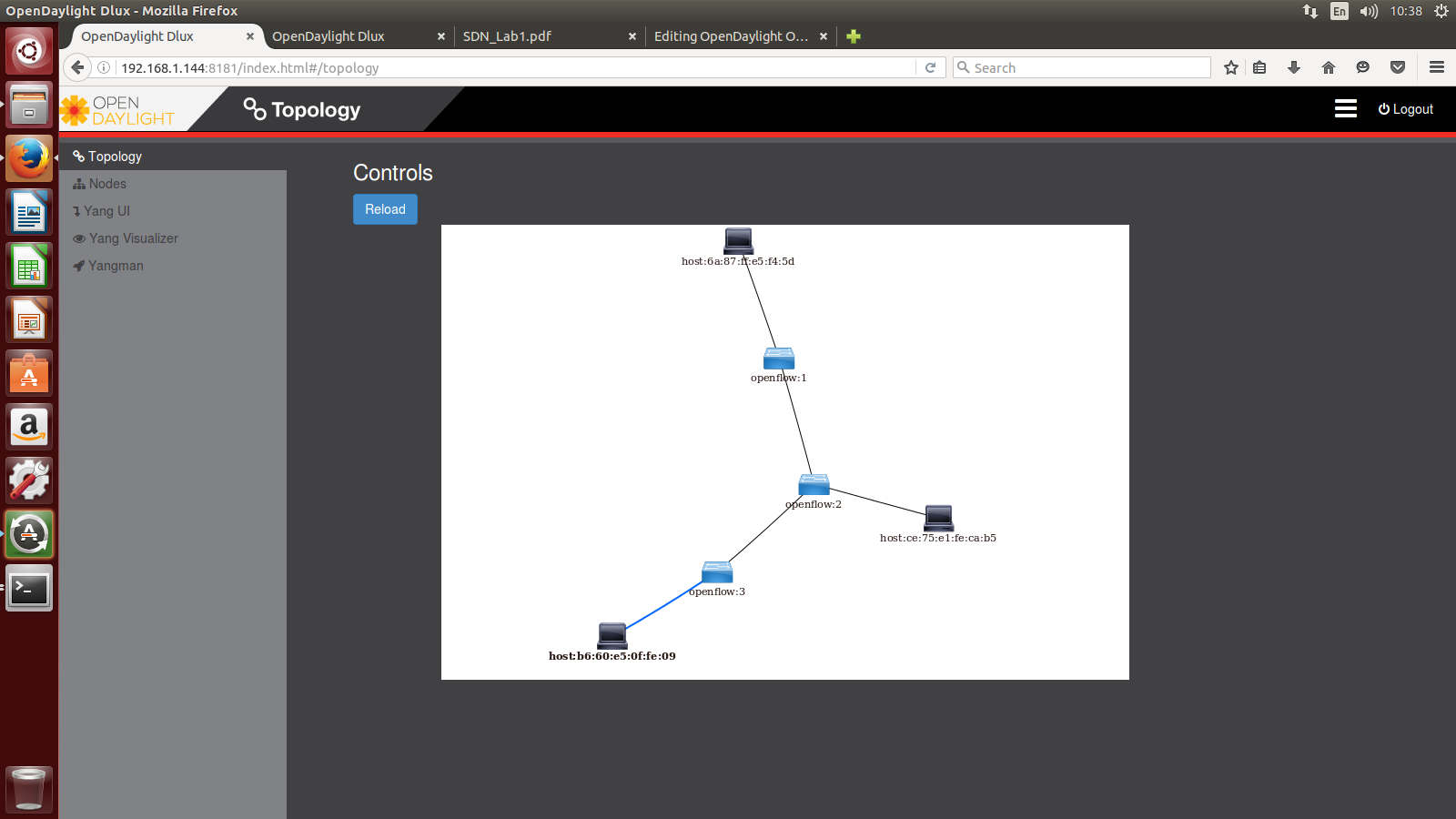
h3 -> h1 h2

\*\*\* Results: 0% dropped (6/6 received)

- Mở trình duyệt truy cập địa chỉ: <http://192.168.1.144:8181/index.html> chính là giao diện DLUX của OpenDaylight

192.168.1.144: IP của OpenDaylight controller

- Đăng nhập user/password là admin/admin sẽ thấy topology như hình dưới đây:



**3. Authentication, Authorization and Accounting (AAA) Services**

- Các nhà quản trị mạng ngày nay phải điều khiển việc truy cập cũng như giám sát thông tin mà người dùng đầu cuối đang thao tác. Những việc làm đó có thể đưa đến thành công hay thất bại của công ty. Với ý tưởng đó, AAA là cách thức tốt nhất để giám sát những gì mà người dùng đầu cuối có thể làm trên mạng. Ta có thể xác thực (authentication) người dùng, cấp quyền (authorization) cho người dùng, cũng như tập hợp được thông tin như thời gian bắt đầu hay kết thúc của người dùng (accounting). Như ta thấy, bảo mật là vấn đề rất quan trọng.

- Với mức độ điều khiển, thật dễ dàng để cài đặt bảo mật và quản trị mạng. Ta có thể định nghĩa các vai trò (role) đưa ra cho user những lệnh mà họ cần để hoàn thành nhiệm vụ của họ và theo dõi những thay đổi trong mạng. Với khả năng log lại các sự kiện, ta có thể có những sự điều chỉnh thích hợp với từng yêu cầu đặt ra.

- Tất cả những thành phần này là cần thiết để duy trì tính an toàn, bảo mật cho mạng. Với thông tin thu thập được, ta có thể tiên đoán việc cập nhật cần thiết theo thời gian. Yêu cầu bảo mật dữ liệu, gia tăng băng thông, giám sát các vấn đề trên mạng,… tất cả đều có thể tìm thấy trên dịch vụ AAA.

**Security Framework for AAA services:**

Kể từ khi phiên bản Boron phát hành, các dịch vụ AAA của OpenDaylight được dựa trên Java Apache Shiro Framework. Tập tin cấu hình chính cho AAA nằm ở "etc / shiro.ini" tương ứng với thư mục chính của OpenDaylight Karaf.

**Cấu hình dịch vụ AAA trong OpenDaylight sử dụng Idmtool:**

# Kích hoạt dịch vụ AAA bằng cách cài đặt tính năng odl-aaa-shiro

opendaylight-user@root> feature:install odl-aaa-shiro

Mặc định đã được cài đặt.

# Thêm 1 user mới

python etc/idmtool admin add-user “user mới”

Ví dụ:

python etc/idmtool admin add-user hungpm

root@opendaylight:/home/opl/Desktop/distribution-karaf-0.5.3-Boron-SR3# python etc/idmtool admin add-user hungpm

Password:

Enter new password:

Re-enter password:

add\_user(admin)

command succeeded!

json:

{

"description": "",

"domainid": "sdn",

"email": "",

"enabled": true,

"name": "hungpm",

"password": "\*\*\*\*\*\*\*\*\*\*",

"salt": "\*\*\*\*\*\*\*\*\*\*",

"userid": "hungpm@sdn"

}

- Password: password của admin

# Xóa 1 user

root@opendaylight:/home/opl/Desktop/distribution-karaf-0.5.3-Boron-SR3# python etc/idmtool admin delete-user hungpm@sdn

Password:

delete\_user(hungpm@sdn

command succeeded!

# Kiểm tra danh sách user

root@opendaylight:/home/opl/Desktop/distribution-karaf-0.5.3-Boron-SR3# python etc/idmtool admin list-users

Password:

list\_users

command succeeded!

json:

{

"users": [

{

"description": "admin user",

"domainid": "sdn",

"email": "",

"enabled": true,

"name": "admin",

"password": "\*\*\*\*\*\*\*\*\*\*",

"salt": "\*\*\*\*\*\*\*\*\*\*",

"userid": "admin@sdn"

},

{

"description": "user user",

"domainid": "sdn",

"email": "",

"enabled": true,

"name": "user",

"password": "\*\*\*\*\*\*\*\*\*\*",

"salt": "\*\*\*\*\*\*\*\*\*\*",

"userid": "user@sdn"

}

]

}

# Thay đổi mật khẩu của 1 user

root@opendaylight:/home/opl/Desktop/distribution-karaf-0.5.3-Boron-SR3# python etc/idmtool admin change-password admin@sdn

Password:

Enter new password:

Re-enter password:

change\_password(admin)

command succeeded!

json:

{

"description": "admin user",

"domainid": "sdn",

"email": "",

"enabled": true,

"name": "admin",

"password": "\*\*\*\*\*\*\*\*\*\*",

"salt": "\*\*\*\*\*\*\*\*\*\*",

"userid": "admin@sdn"

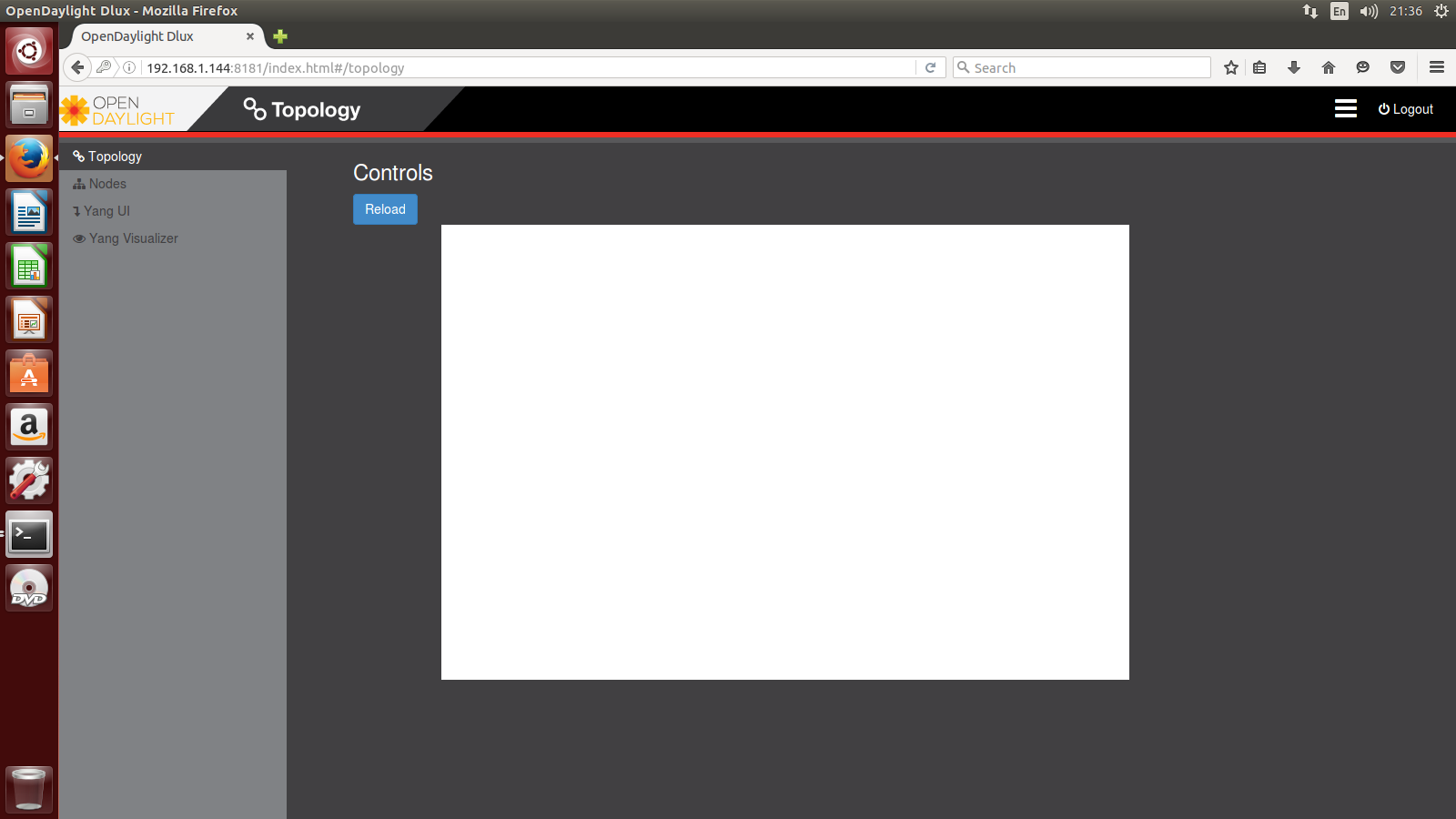
}

- Để user có thể đăng nhập vào sử dụng DLUX thì phải tạo grant và thêm grant cho user đó bằng các câu lệnh sau:

python etc/idmtool admin get-grants hungpm@sdn

python etc/idmtool admin add-grant hungpm@sdn admin@sdn

- Đăng nhập bằng user/password: hungpm/1



**4. Defense4all**

**4.1 Defense4all là gì?**

- Defense4All là một ứng dụng SDN bảo mật để phát hiện và giảm thiểu các cuộc tấn công DoS và DDoS trong các topology SDN khác nhau. Nó nhận ra chống DoS trong chế độ OOP cho môi trường SDM ProgrammableFlow.

- Quản trị viên có thể cấu hình Defense4All để bảo vệ các mạng và máy chủ nhất định, được gọi là các mạng được bảo vệ hoặc các đối tượng được bảo vệ (PO). Defense4All khai thác khả năng SDN để tính lưu lượng xác định, và cài đặt lưu lượng truy cập các luồng cho mỗi giao thức của mỗi PO được cấu hình trong mỗi vị trí mạng (VTN Vexternals) thông qua đó lưu lượng truy cập của các chủ đề PO chảy.

- Defense4All sẽ theo dõi lưu lượng truy cập của tất cả các cấu hình PO, tóm tắt các giá trị đọc, tỷ lệ và trung bình từ tất cả các vị trí mạng có liên quan. Nếu phát hiện ra một sai lệch từ hành vi giao thông học tập bình thường trong một giao thức (như TCP, UDP, ICMP, hoặc phần còn lại của giao thông) của một PO cụ thể.

**4.2 Cài đặt và cấu hình dịch vụ Defense4all**

* **Cài đặt Cassandra ( Hệ quản trị cơ sở dữ liệu theo kiểu NoSQL)**

echo "deb http://debian.datastax.com/community stable main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list

sudo apt-get install curl

curl -L http://debian.datastax.com/debian/repo\_key | sudo apt-key add -

sudo apt-get update

sudo apt-get install dsc20=2.0.11-1 cassandra=2.0.11

sudo service cassandra restart

sudo service cassandra status

* **Cài đặt Defense4all**

sudo apt-get install git

git clone https://github.com/opendaylight/defense4all.git

sudo apt-get install rpm

cd defense4all/dfapp/

mvn clean install

cd defense4all/dfapp.aggregate/target/

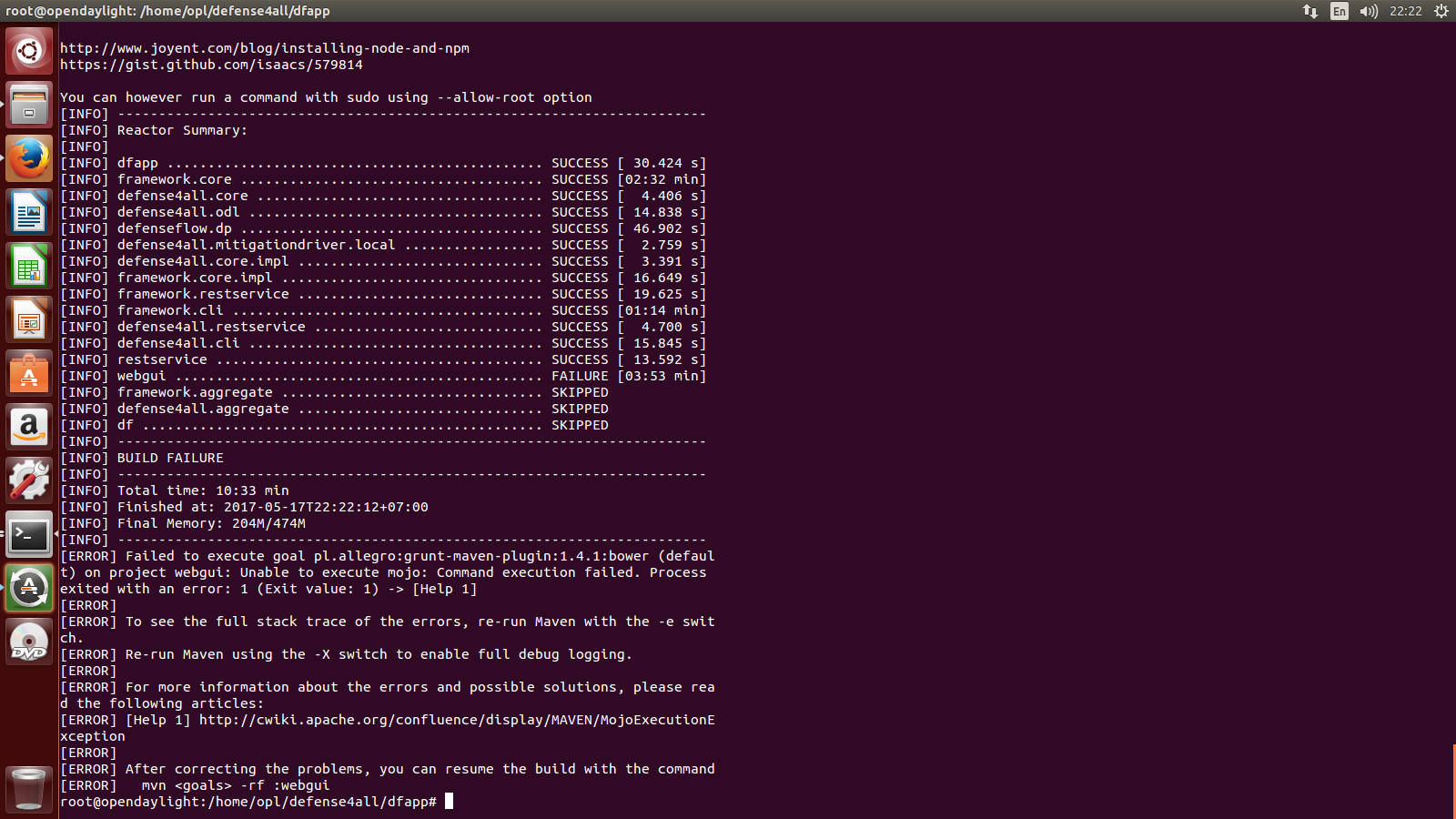
sudo dpkg -i defense4all\_1.1.0.deb

sudo dpkg -i defense4all\_cli\_1.1.0.deb

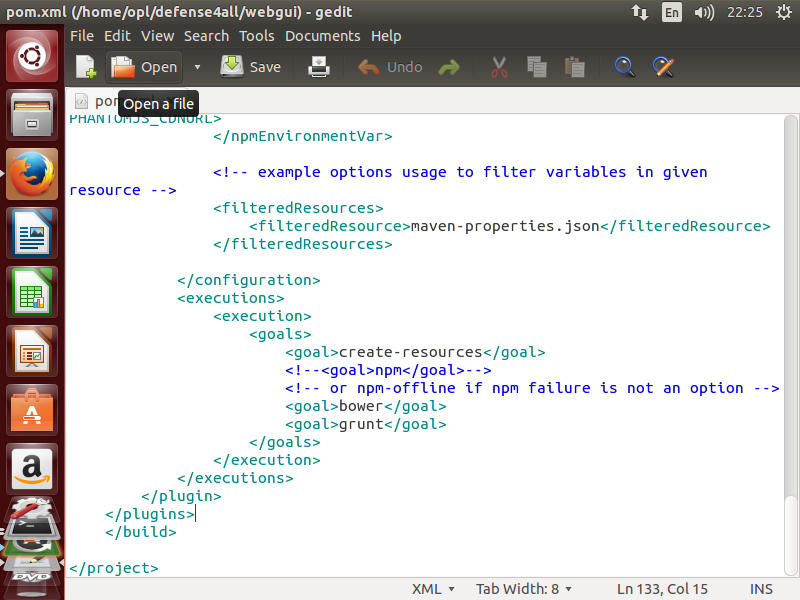
sudo service defense4all restart

sudo service defense4all status

- Nếu tại câu lệnh “mvn clean install” bị lỗi



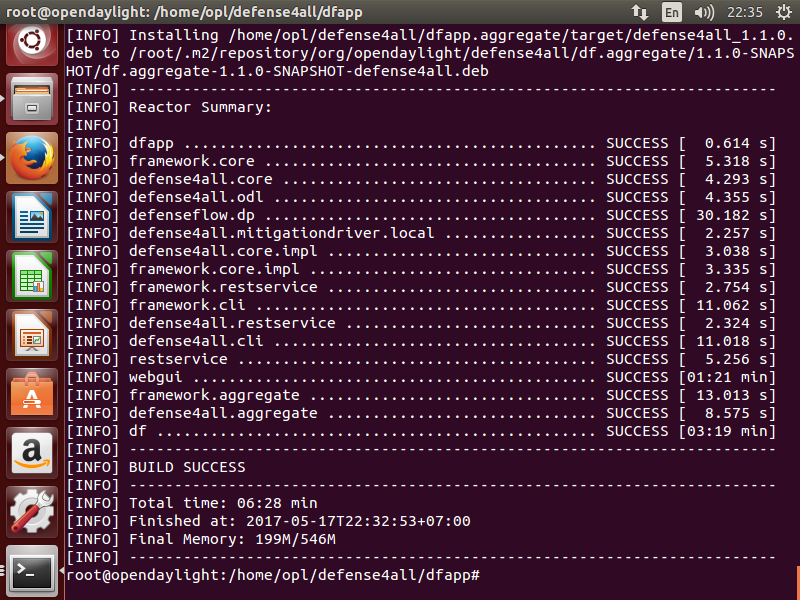
- Vào thư mục defense4all/webgui/ sau đó vào file pom.xml xóa dòng chứa “bower”.



- Sau đó vào thư mục defense4all/dfapp chạy “mvn clean install” lại rồi làm các câu lệnh như trên tiếp.

opl@opendaylight:~$ cd defense4all/dfapp

opl@opendaylight:~/defense4all/dfapp$ mvn clean install



- Khi khởi động dịch vụ defense4all mà bị lỗi:

bash: /var/log/defense4all/output.log: Permission denied

defense4all is not running

- Cấp quyền cho thư mục defense4all

chmod 777 /var/log/defense4all

- Khởi động lại dịch vụ defens4all

root@opendaylight:/home/opl# service defense4all restart

defense4all is not running

defense4all is running

**Reset:**

Ta cần reset lại hệ thống khi muốn thử test một Testbed mới.

* Reset framework trong cửa sổ duyệt web của Defense4all

Trong cửa sổ terminal

sudo service defense4all stop

sudo service cassandra stop

sudo rm /var/log/defense4all/server.log #Xóa file lưu chữ các thông báo khi chạy Defense4all

sudo rm -rf /var/lib/cassandra/data/system/\*

sudo shutdown -r now

* Khởi động lại cassandra và Defense4all.

sudo service cassandra restart

sudo service defense4all restart

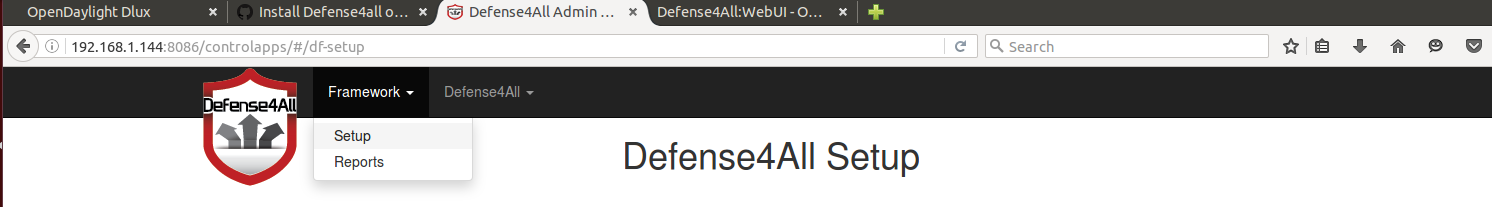
**Cấu hình Defense4all sử dụng Web Interface:**

* **Cấu hình Framework Environment**

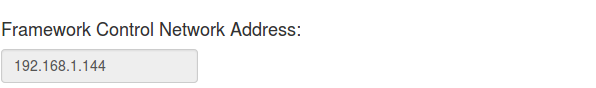
1. Nếu bạn chưa có trong giao diện người dùng Defense4All, từ một trình duyệt Internet, hãy truy cập http: // <ip address>: 8086 / controlapps, trong đó <ip address> là địa chỉ của máy chủ đang chạy Defense4All.

VD: <http://192.168.1.144:8086/controlapps>

1. Nếu chưa có Framework Setup thì chọn Framework -> Setup



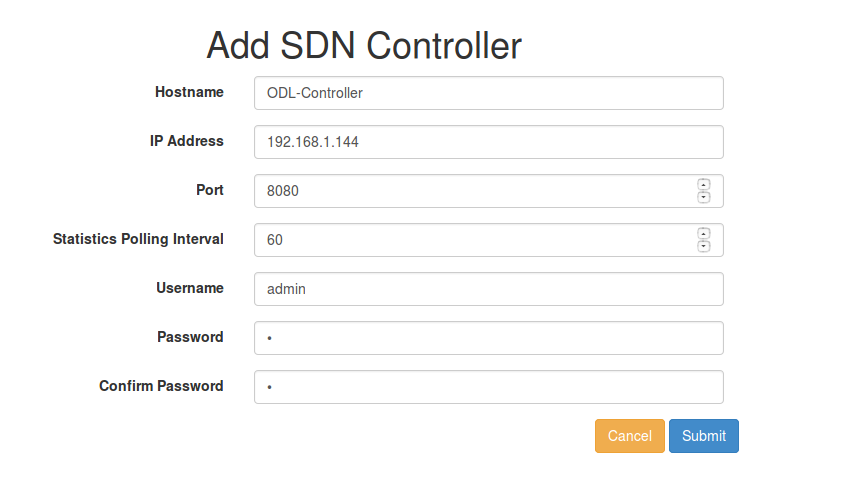
1. Thiết lập Framework Control Network Address cho địa chỉ IP Defense4All sử dụng để truy cập vào mạng điều khiển.



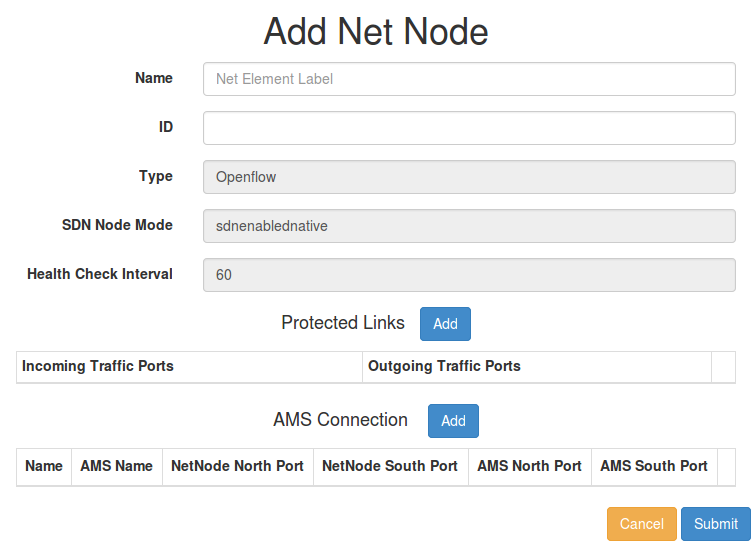
1. Ở bên phải nhãn của SDN Controller chọn Add

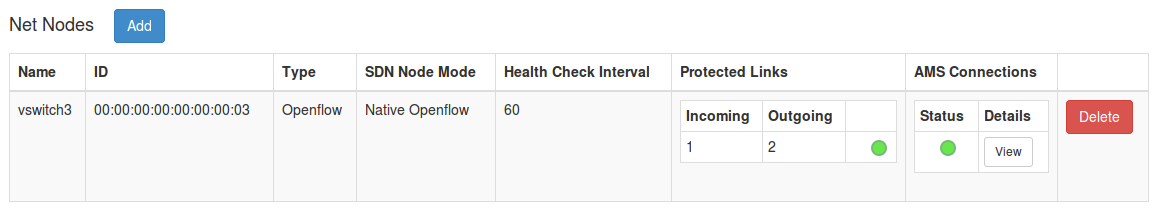


1. Trong phần Add SDN Controller đặt các thông số sau:



1. Chọn Submit
2. Thêm Net Node



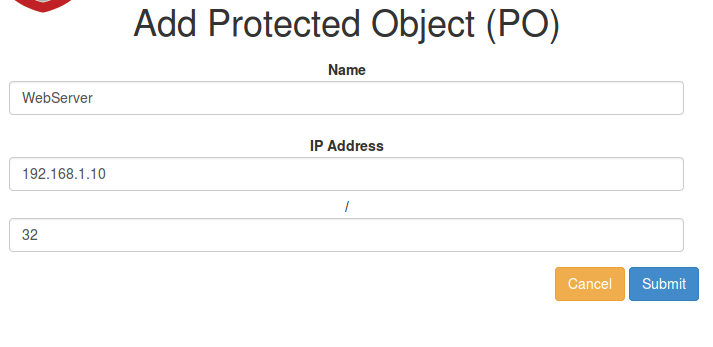


* **Cấu hình Defense4all Protected Objects (POs)**

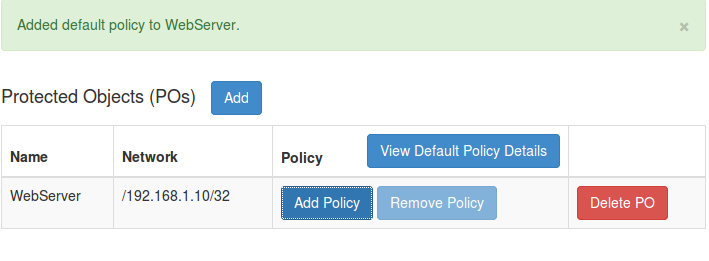
1. Chọn Defense4all vào phần Setup
2. Bên phải nhãn Protected Objects nhấp vào Add



1. Thiết lập các tham số

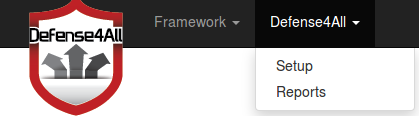


1. Chọn Submit
2. Add Policy



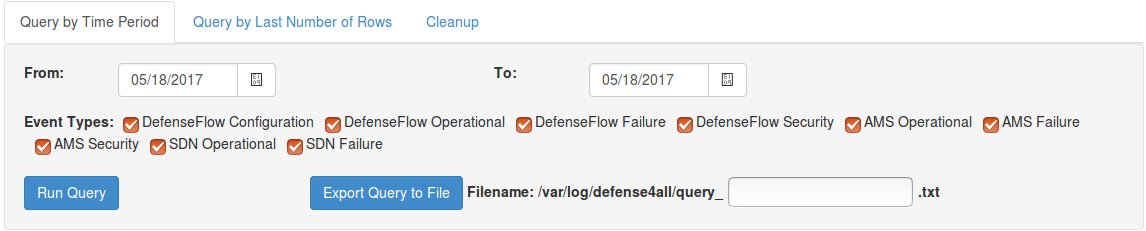
* **Defense4all Report**

1. Vào Defense4all chọn Report



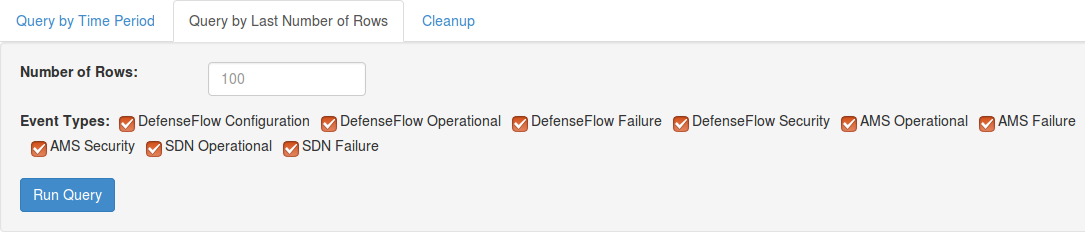
- Trong Report của Defense4all có 2 loại truy vấn và phần xóa các sự kiện

* + Query by Time Period:
  1. Trong các trường From và To, chọn các ngày thích hợp để xác định phạm vi của truy vấn.
  2. Chọn các loại sự kiện mà bạn muốn đưa vào báo cáo.
  3. Nhấp vào Chạy truy vấn. Kết quả hiển thị ở cuối bảng.
  4. Nếu bạn muốn lưu truy vấn vào một tệp tin, hãy nhập đường dẫn tệp vào tên tệp đã được tải và nhấp vào Xuất truy vấn vào tệp.



* + Query by Last Number of Rows:

1. Trong trường Số hàng, hãy nhập số cuối cùng của hàng trong cơ sở dữ liệu mà bạn muốn hiển thị trong báo cáo của mình.
2. Chọn các loại sự kiện mà bạn muốn đưa vào báo cáo.
3. Nhấp vào Chạy truy vấn. Kết quả hiển thị ở cuối bảng. Bạn không thể lưu truy vấn này vào tệp tin



* + Clean up:
    1. Trong trường Delete events older than, nhập một số ngày. Các sự kiện lớn hơn số ngày này sẽ bị xóa.
    2. Nhấp vào Submit. Kết quả hiển thị ở cuối bảng. Bạn không thể lưu truy vấn này vào một tệp.

