

การพัฒนาแอปพลิเคชันและตัวควบคุมตามสถาปัตยกรรมเอสดีเอ็น เพื่อควบคุม
การกระจายแพทเทรฟฟิกแบบที่ผู้ใช้งานกำหนดได้ บนเครือข่ายที่อุปกรณ์ไม่รองรับ
มาตรฐานเอสดีเอ็น

**IMPLEMENTATION OF SDN APPLICATION AND CONTROLLER
FOR USER-DEFINED TRAFFIC DISTRIBUTION IN TRADITIONAL
(NON-SDN) NETWORKS**

โดย
พงศ์พนิช อรัญรัตน์โสภณ
ภูริณัฐ จิตมนัส

อาจารย์ที่ปรึกษา
ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2564

สารบัญ

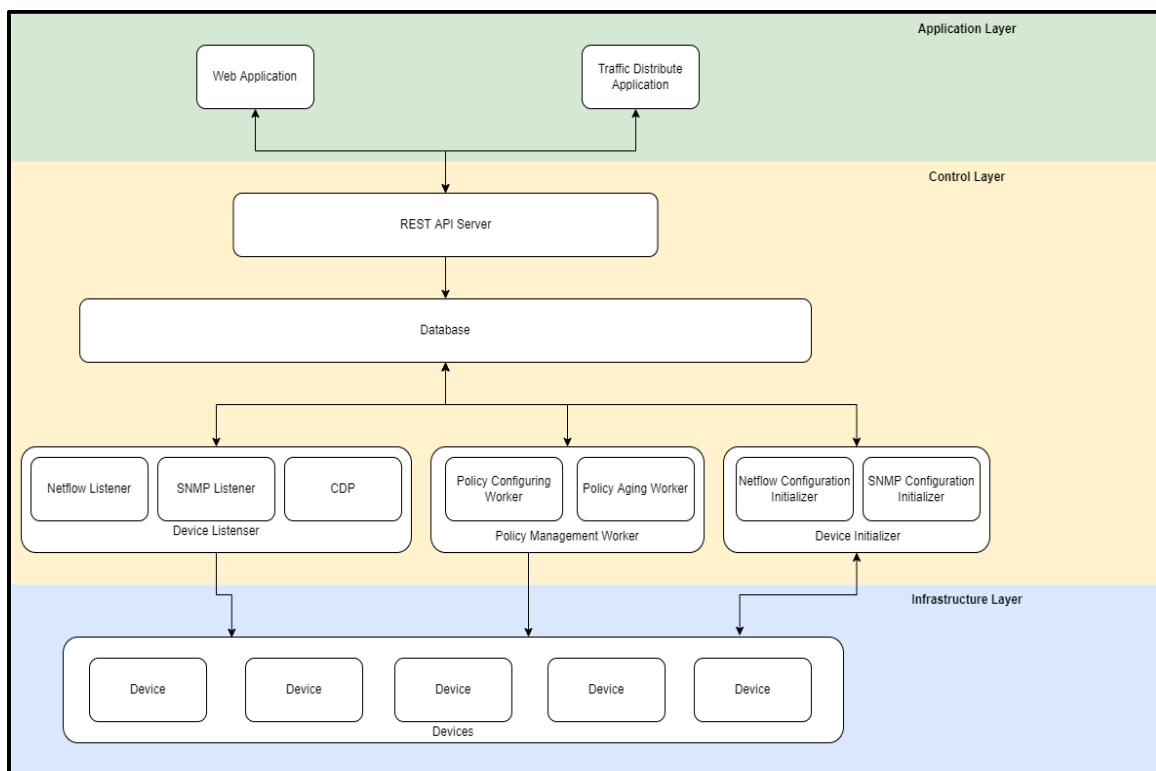
หน้าที่

สารบัญ	i
บทที่ 1 ภาพรวมระบบ	1
บทที่ 2 การติดตั้งระบบ และ โปรแกรมที่จำเป็นต่อการใช้งาน	3
2.1 การติดตั้งภาษาไพธอน.....	3
2.2 การติดตั้งคอนโทรลเลอร์	5
2.3 การติดตั้งฐานข้อมูล	5
2.3 การติดตั้งแอปพลิเคชันส่วน Frontend (Node.js Yarn และ Vue3).....	6
2.4 การติดตั้งโมดูลอื่นๆที่จำเป็น	10
2.5 การเปิดใช้งานคอนโทรลเลอร์	11
บทที่ 3 การเชื่อมต่อคอนโทรลเลอร์กับอุปกรณ์เครือข่าย	14
3.1 การตั้งค่าอุปกรณ์เครือข่าย	14
3.2 การใช้งานหน้าเว็บเพื่อควบคุมคอนโทรลเลอร์.....	15
3.3 การตั้งค่าเริ่มต้นให้อุปกรณ์เครือข่ายผ่านการเรียกใช้คำสั่งจากโค้ดไพธอนโดยตรง.....	20
บทที่ 4 ข้อมูลชุดคำสั่งเอพีไอที่สามารถใช้งานได้และตัวอย่างการใช้งาน	21
4.1 รายละเอียดคำสั่ง “device”	21
4.2 รายละเอียดคำสั่ง “flow”	22
4.3 รายละเอียดคำสั่ง “link”	23
4.4 รายละเอียดคำสั่ง “path”	24
4.5 รายละเอียดคำสั่ง “flow/routing”	25

บทที่ 1

ภาพรวมระบบ

ชุดทดลองระบบจัดการเครือข่ายตามสถาปัตยกรรมแบบเอสดีเอ็น เป็นระบบที่ถูกออกแบบมาเพื่อให้อุปกรณ์เครือข่ายที่ไม่รองรับการทำงานภายใต้สถาปัตยกรรมแบบเอสดีเอ็น (Software Defined Network: SDN) สามารถถูกควบคุมจากระบบดังกล่าวเพื่อที่จะช่วยอำนวยความสะดวกให้กับผู้ควบคุมระบบเครือข่าย ให้สามารถมองเห็นภาพรวม และควบคุมระบบเครือข่ายนั้นผ่านระบบคอนโทรลเลอร์ ซึ่งทำหน้าที่เสมือนระบบที่ถูกออกแบบภายใต้สถาปัตยกรรมแบบเอสดีเอ็นได้ โดยภาพรวมของระบบที่ถูกพัฒนาขึ้นจะถูกแสดงเอาไว้ตามรูปที่ 1.1



รูปที่ 1.1 ภาพรวมของระบบ

ระบบประกอบไปด้วยอุปกรณ์เครือข่ายที่เชื่อมต่อเข้ากับตัวควบคุม ซึ่งตัวควบคุมจำเป็นต้องสามารถเชื่อมต่อโดย Secure Shell (SSH) ไปยังอุปกรณ์เครือข่ายได้ เพื่อส่งคำสั่งตั้งค่า สำหรับเปิดใช้งาน SNMP และ NetFlow สำหรับเก็บข้อมูลเครือข่าย และส่งคำสั่งตั้งค่า Policy Based Routing

ข้อมูลที่เก็บมาจากอุปกรณ์เครือข่ายจะถูกบันทึกลงในฐานข้อมูล โดยจะมีหน้าเว็บแอปพลิเคชันที่จะดึงข้อมูลจากฐานข้อมูลผ่านทาง REST API ตัวแอปพลิเคชันจะทำหน้าที่แสดงข้อมูลจากระบบเครือข่ายให้ผู้ใช้สามารถดูและทำความเข้าใจได้ง่าย

จากที่กล่าวมาข้างต้นทำให้แอปพลิเคชันตัวควบคุมจะประกอบไปด้วยระบบหลักๆ 4 ส่วนดังต่อไปนี้

1. Controller Application เป็นแอปพลิเคชันที่ทำหน้าที่ประมวลผล ส่งคำสั่งตั้งค่าไปยังอุปกรณ์เครือข่าย รวมถึงรับข้อมูลจากอุปกรณ์เครือข่ายเพื่อบันทึกลงฐานข้อมูล
2. Database เป็นส่วนที่เก็บข้อมูลต่าง ๆ ของระบบเครือข่าย
3. Web Application เป็นส่วนที่ช่วยให้ผู้ใช้สามารถส่งคำสั่งเพิ่ม-ลบอุปกรณ์ เปิดการเก็บข้อมูลจาก SNMP และ NetFlow และใช้แสดงผลเครือข่าย
4. API Module เป็นส่วนที่ถูกใช้ในการรับส่งคำสั่งระหว่างผู้ใช้ ตัวควบคุม ละอุปกรณ์เครือข่าย ช่วยสามารถใช้เขียน โปรแกรมเพื่อควบคุมโดยตรง หรือสั่งการผ่านหน้าเว็บแอปพลิเคชันได้

บทที่ 2

การติดตั้งระบบ และโปรแกรมที่จำเป็นต่อการใช้งาน

ระบบคอนโทรลเลอร์ ถูกพัฒนาขึ้นภายใต้ระบบปฏิบัติการ Linux ซึ่งในที่นี้ผู้พัฒนาได้ทำการพัฒนาระบบภายใต้ระบบปฏิบัติการ CentOS 7 แต่ในขณะเดียวกันระบบคอนโทรลเลอร์นี้ก็ยังสามารถนำไปใช้กับระบบปฏิบัติการอื่นได้เช่นกัน แต่รูปแบบการติดตั้งอาจจะแตกต่างกันไปตามแต่คำสั่งของระบบปฏิบัติการ โดยในคู่มือฉบับนี้ผู้พัฒนาระบบจะแสดงวิธีการติดตั้งผ่านระบบคอนโทรลเลอร์ภายใต้ระบบปฏิบัติการ CentOS 7 เพื่อเป็นตัวอย่าง

เพื่อให้ระบบคอนโทรลเลอร์สามารถเข้าถึงอุปกรณ์เครือข่ายได้นั้น ผู้ใช้จะต้องติดตั้งระบบคอนโทรลเลอร์นั้นลงไปในเซิร์ฟเวอร์หรือคอมพิวเตอร์ที่สามารถติดต่อกับอุปกรณ์เครือข่ายที่ต้องการควบคุมได้โดยในขั้นแรกผู้ใช้ต้องทำการติดตั้งโปรแกรมที่จำเป็นต่อการใช้งานระบบเสียก่อน โดยมีขั้นตอนดังต่อไปนี้

2.1 การติดตั้งภาษาไพธอน

ระบบคอนโทรลเลอร์นั้นได้ถูกพัฒนาผ่านภาษาไพธอนเวอร์ชัน 3.6 (Python 3.6) ดังนั้นผู้ใช้งานจำเป็นต้องติดตั้งภาษาดังกล่าว รวมทั้งเครื่องมือที่เกี่ยวข้องก่อนเริ่มใช้งาน โดยมีขั้นตอนดังต่อไปนี้ (การติดตั้งจำเป็นต้องดาวน์โหลดข้อมูลผ่านอินเทอร์เน็ต)

1. เปิดใช้งานเทอร์มินัล
2. พิมพ์คำสั่ง `yum install update -y` เพื่อปรับปรุงเวอร์ชันแพ็คเกจให้เป็นปัจจุบันพร้อมใช้งาน
3. พิมพ์คำสั่งต่อไปนี้เพื่อติดตั้งไพธอนและไลบรารีที่จำเป็น
 - a. `yum install -y python3`
 - b. `pip3.6 install --upgrade pip`

```

python3-libs                               x86_64                               3.6.8-10.el7                               updates                               6.9 M
python3-pip                                noarch                                9.0.3-0.el7                               1.6 M
python3-setuptools                          noarch                                39.2.0-10.el7                             base                                629 k
Transaction Summary
-----
Install 1 Package (+4 Dependent packages)

Total download size: 9.3 M
Installed size: 40 M
Downloading packages:
(1/5): python3-3.6.8-10.el7.x86_64.rpm                                1.70 kB  00:00:00
(2/5): libtirpc-0.2.4-0.16.el7.x86_64.rpm                             1.09 kB  00:00:00
(3/5): python3-setuptools-39.2.0-10.el7.noarch.rpm                    6.29 kB  00:00:00
(4/5): python3-pip-9.0.3-0.el7.noarch.rpm                             1.16 MB  00:00:00
(5/5): python3-libs-3.6.8-10.el7.x86_64.rpm                           1.6.9 MB  00:00:00
-----
Total                                                                    33 MB/s | 9.3 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : libtirpc-0.2.4-0.16.el7.x86_64                               1/5
Installing : python3-setuptools-39.2.0-10.el7.noarch                     2/5
Installing : python3-pip-9.0.3-0.el7.noarch                             3/5
Installing : python3-3.6.8-10.el7.x86_64                                4/5
Installing : python3-libs-3.6.8-10.el7.x86_64                           5/5
Verifying : libtirpc-0.2.4-0.16.el7.x86_64                               1/5
Verifying : python3-setuptools-39.2.0-10.el7.noarch                     2/5
Verifying : python3-libs-3.6.8-10.el7.x86_64                           3/5
Verifying : python3-3.6.8-10.el7.x86_64                                4/5
Verifying : python3-pip-9.0.3-0.el7.noarch                             5/5

Installed:
python3.x86_64 0:3.6.8-10.el7

Dependency Installed:
libtirpc.x86_64 0:0.2.4-0.16.el7      python3-libs.x86_64 0:3.6.8-10.el7      python3-pip.noarch 0:9.0.3-0.el7      python3-setuptools.noarch 0:39.2.0-10.el7

Complete!
[root@localhost ~]# pip3.6 install --upgrade pip
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3.6 install --user' instead.
Collecting pip
  downloading https://files.pythonhosted.org/packages/a4/64/646349333547439a6b5b90b46a0f0742cc83ae83543e4d7600c248f0b/pip-21.3.1-py3-none-any.whl (1.7MB)
    100% |#####| 1.7MB 760kB/s
Installing collected packages: pip
Successfully installed pip-21.3.1
[root@localhost ~]# _

```

รูปที่ 2.2 ติดตั้ง Python3 และ Pip3 สำเร็จ

2.2 การติดตั้งคอนโทรลเลอร์

ระบบถูกแบ่งออกเป็นสามส่วน ได้แก่ Control module, API module และ Webpage module โดยวิธีการติดตั้งนั้นผู้ใช้สามารถดาวน์โหลดไฟล์ดังกล่าวจากเว็บไซต์ Github และติดตั้งผ่านเทอร์มินัลได้ โดยวิธีการติดตั้งผ่านเทอร์มินัลนั้นมีรายละเอียดดังต่อไปนี้

1. พิมพ์คำสั่ง `yum install git -y` เพื่อติดตั้ง `github`
2. พิมพ์คำสั่ง `git clone https://github.com/FameIllusionMaya/SDN-handmade_v3.git`

2.3 การติดตั้งฐานข้อมูล

ผู้พัฒนาระบบได้เลือกใช้ฐานข้อมูล `mongo` เพื่อเก็บข้อมูลของระบบเครือข่าย โดยมีขั้นตอนดังต่อไปนี้

1. เรียกใช้คำสั่ง `vi /etc/yum.repos.d/mongodb-org.repo` เพื่อสร้าง และเปิดไฟล์ใหม่เพื่อใช้ในการติดตั้งฐานข้อมูล
2. จากนั้นให้เพิ่มรายละเอียดของไฟล์ตามที่แสดงไว้ตามรูปที่ 2.3 หลังจากนั้นบันทึกและปิดไฟล์ด้วยคำสั่ง `:wq`
3. พิมพ์คำสั่ง `yum install mongodb-org -y` เพื่อติดตั้ง `mongoDB`
4. พิมพ์คำสั่ง `systemctl start mongod` เพื่อเปิดใช้งานฐานข้อมูล
5. เพื่อทดสอบว่าฐานข้อมูลถูกติดตั้งสำเร็จ และสามารถใช้งานได้ ให้ผู้ใช้พิมพ์คำสั่ง `mongo` ลงไปในคอนโซล ถ้าระบบแสดงหน้าต่างดังรูปที่ 2.4 แสดงว่าผู้ใช้สามารถติดตั้งฐานข้อมูลสำเร็จ

```
CentOS_Test2
[mongodb-org-3.4]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.4/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc_
```

รูปที่ 2.3 รายละเอียดข้อมูลที่ต้องเพิ่มลงไปในไฟล์ที่ถูกสร้างขึ้น

```

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : mongodb-org-server-3.4.24-1.el7.x86_64                                1/5
Created symlink from /etc/systemd/system/multi-user.target.wants/mongod.service to /usr/lib/systemd/system/mongod.service.
  Installing : mongodb-org-shell-3.4.24-1.el7.x86_64                                2/5
  Installing : mongodb-org-mongos-3.4.24-1.el7.x86_64                                3/5
  Installing : mongodb-org-tools-3.4.24-1.el7.x86_64                                4/5
  Installing : mongodb-org-3.4.24-1.el7.x86_64                                       5/5
  Verifying  : mongodb-org-tools-3.4.24-1.el7.x86_64                                1/5
  Verifying  : mongodb-org-mongos-3.4.24-1.el7.x86_64                                2/5
  Verifying  : mongodb-org-shell-3.4.24-1.el7.x86_64                                3/5
  Verifying  : mongodb-org-3.4.24-1.el7.x86_64                                       4/5
  Verifying  : mongodb-org-server-3.4.24-1.el7.x86_64                               5/5

Installed:
  mongodb-org.x86_64 0:3.4.24-1.el7

Dependency Installed:
  mongodb-org-mongos.x86_64 0:3.4.24-1.el7          mongodb-org-server.x86_64 0:3.4.24-1.el7          mongodb-org-shell.x86_64 0:3.4.24-1.el7
  mongodb-org-tools.x86_64 0:3.4.24-1.el7

Complete!
[root@localhost ~]# systemctl start mongod
[root@localhost ~]# mongo
MongoDB shell version 3.4.24
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.24
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
http://docs.mongodb.org/
Questions? Try the support group
http://groups.google.com/group/mongodb-user
Server has startup warnings:
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten]
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten] == WARNING: Access control is not enabled for the database.
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten] ==          Read and write access to data and configuration is unrestricted.
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten]
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten]
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten]
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten] == WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten] ==          We suggest setting it to 'never'
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten]
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten]
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten]
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten] == WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten] ==          We suggest setting it to 'never'
2022-06-16T13:28:35.011-0400 I CONTROL [initandlisten]
>

```

รูปที่ 2.4 ติดตั้งฐานข้อมูลสำเร็จ

2.3 การติดตั้งแอปพลิเคชันส่วน Frontend (Node.js Yarn และ Vue3)

เนื่องจากผู้พัฒนาได้ใช้ Node.js Yarn และ Vue3 ในการพัฒนาเว็บไซต์ที่ใช้ในการติดต่อสื่อสารกับระบบดังนั้นผู้ใช้งานจำเป็นต้องติดตั้ง Node.js Yarn และ Vue3 เพื่อใช้งานหน้าเว็บดังกล่าว โดยให้ผู้ใช้งานไปยัง Directory SDN-handmade_v3/sdn_frontendv3 และเรียกใช้คำสั่งดังต่อไปนี้

1. `curl -sL https://rpm.nodesource.com/setup_16.x | sudo bash -`
2. `yum install nodejs -y`
3. `npm install -g @vue/cli`
4. `curl --silent --location https://dl.yarnpkg.com/rpm/yarn.repo | sudo tee /etc/yum.repos.d/yarn.repo`
5. `rpm --import https://dl.yarnpkg.com/rpm/pubkey.gpg`
6. `yum install yarn -y`


```

Loading mirror speeds from cached hostfile
* base: mirrors.bangmod.cloud
* extras: mirrors.bangmod.cloud
* updates: mirrors.bangmod.cloud
nodesource
nodesource.x86_64/primary.db
Resolving dependencies
--> Running transaction check
--> Package nodejs.x86_64 2:8.17.8-1nodesource will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                arch                Version                Repository             Size
=====
Installing:
nodejs                  x86_64                2:8.17.8-1nodesource   nodesource              18 M

Transaction Summary
=====
Install 1 Package
Total download size: 18 M
Installed size: 55 M
Downloading packages:
warning: /var/cache/yum/x86_64/2/nodesource/packages/nodejs-8.17.8-1nodesource.x86_64.rpm: Header V4 RSA/SHA512 Signature, key ID 34fa74dd: NOKEY --:--:-- ETh
Public key for nodejs-8.17.8-1nodesource.x86_64.rpm is not installed
nodejs-8.17.8-1nodesource.x86_64.rpm
Retrieving key from file:///etc/pki/rpm-gpg/NODESOURCE-GPG-SIGNING-KEY-EL
Importing GPG key 8C3FA74DD:
  Userid : "Nodesource <gpg-rpm@nodesource.com>"
  Fingerprint: 2e55 287a 95d9 944b 8cc9 3261 5dd8 e8d4 34fa 74dd
  Package : nodesource-release-el7-1.noarch (installed)
  From    : /etc/pki/rpm-gpg/NODESOURCE-GPG-SIGNING-KEY-EL
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Warning: RPMDB altered outside of yum.
Installing : 2:nodejs-8.17.8-1nodesource.x86_64
Verifying : 2:nodejs-8.17.8-1nodesource.x86_64
1/1
1/1

Installed:
nodejs.x86_64 2:8.17.8-1nodesource

Complete!
[root@localhost ~]#

```

รูปที่ 2.5 ติดตั้ง Nodejs สำเร็จ

```

Transaction test succeeded
Running transaction
Warning: RPMDB altered outside of yum.
Installing : 2:nodejs-16.15.1-1nodesource.x86_64
Verifying : 2:nodejs-16.15.1-1nodesource.x86_64
1/1
1/1

Installed:
nodejs.x86_64 2:16.15.1-1nodesource

Complete!
[root@localhost sdn_frontendv3]# npm install -g @vue/cli
npm WARN config global '--global', '-local' are deprecated. Use '--location=global' instead.
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated apollo-tracing@0.15.8: The 'apollo-tracing' package is no longer part of Apollo Server 3. See https://www.apollographql.com/docs/apollo-server/migration/#tracing for details
npm WARN deprecated graphql-extensions@0.15.8: The 'graphql-extensions' API has been removed from Apollo Server 3. Use the plugin API instead: https://www.apollographql.com/docs/apollo-server/integrations/plugins/
npm WARN deprecated apollo-cache-control@0.14.8: The functionality provided by the 'apollo-cache-control' package is built in to 'apollo-server-core' starting with Apollo Server 3. See https://www.apollographql.com/docs/apollo-server/migration/#cachecontrol for details.
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated subscriptions-transport-ws@0.9.19: The 'subscriptions-transport-ws' package is no longer maintained. We recommend you use 'graphql-ws' instead. For help migrating Apollo software to 'graphql-ws', see https://www.apollographql.com/docs/apollo-server/data/subscriptions/#switching-from-subscriptions-transport-ws
For general help using graphql-ws, see https://github.com/enisdenjo/graphql-ws/blob/master/README.md
npm WARN deprecated graphql-tools@0.8: This package has been deprecated and now it only exports makeExecutableSchema. We recommend you to migrate to scoped packages such as @graphql-tools/schema, @graphql-tools/utils and etc. Check out https://www.graphql-tools.com to learn what package you should use instead

added 897 packages, and audited 898 packages in 45s
89 packages are looking for funding
  run 'npm fund' for details

6 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run 'npm audit' for details.
npm notice
npm notice New minor version of npm available! 8.11.0 -> 8.12.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.12.2
npm notice Run npm install -g npm@8.12.2 to update!
npm notice
[root@localhost sdn_frontendv3]# curl --silent --location_

```

รูปที่ 2.6 ติดตั้ง Vue3 สำเร็จ

```
baseurl=https://dl.yarnpkg.com/rpm/
enabled=1
gpgcheck=1
gpgkey=https://dl.yarnpkg.com/rpm/pubkey.gpg
root@localhost ~# rpm --import https://dl.yarnpkg.com/rpm/pubkey.gpg
root@localhost ~# yum install yarn
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.bangmod.cloud
 * extras: mirrors.bangmod.cloud
 * updates: mirrors.bangmod.cloud
yarn
yarn/primary.db
Resolving Dependencies
--> Running transaction check
--> Package yarn.noarch 0:1.22.19-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
yarn noarch 1.22.19-1 yarn 1.2 M

Transaction Summary
=====
Install 1 Package
Total download size: 1.2 M
Installed size: 5.1 M
Is this ok [y/d/N]: y
Downloading packages:
yarn-1.22.19-1.noarch.rpm 1.2 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : yarn-1.22.19-1.noarch 1/1
Verifying : yarn-1.22.19-1.noarch 1/1

Installed:
yarn.noarch 0:1.22.19-1

Complete!
root@localhost ~#
```

รูปที่ 2.7 ติดตั้ง Yarn สำเร็จ

และผู้ใช้ปิด Firewall โดยคำสั่ง

1. systemctl stop firewalld
2. systemctl disable firewalld
3. systemctl mask --now firewalld

หลังจากนั้นให้ผู้ใช้แก้ไขข้อมูลในไฟล์ที่ชื่อ httpclient.ts (อยู่ใน directory SDN/sdn_frontendv3/src/) โดยเปลี่ยน baseURL ให้ตรงกับไอพีของเครื่องคอนโทรลเลอร์

```
=====
Install 1 Package
Total download size: 1.2 M
Installed size: 5.1 M
Is this ok [y/d/N]: y
Downloading packages:
yarn-1.22.19-1.noarch.rpm 1.2 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : yarn-1.22.19-1.noarch 1/1
Verifying : yarn-1.22.19-1.noarch 1/1

Installed:
yarn.noarch 0:1.22.19-1

Complete!
root@localhost sdn_frontendv3# systemctl stop firewalld
root@localhost sdn_frontendv3# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
root@localhost sdn_frontendv3# systemctl mask --now firewalld
Failed to parse lines 'on'
root@localhost sdn_frontendv3# systemctl mask --now firewalld
Created symlink from /etc/systemd/system/firewalld.service to /dev/null.
root@localhost sdn_frontendv3# ifconfig
-bash: ifconfig: command not found
root@localhost sdn_frontendv3# ipaddr show
-bash: ipaddr: command not found
root@localhost sdn_frontendv3# show ipaddr
-bash: show: command not found
root@localhost sdn_frontendv3# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:55:e2:03 brd ff:ff:ff:ff:ff:ff
    inet 10.50.34.25/24 brd 10.50.34.255 scope global noprefixroute dynamic ens192
        valid_lft 344866sec preferred_lft 344866sec
    inet6 fc00::1:40:1000::de:00000000/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
root@localhost sdn_frontendv3# _
```

รูปที่ 2.8 ปิด Firewall สำเร็จ และ ไอพีของคอนโทรลเลอร์คือ 10.50.34.25

```

import axios, { AxiosInstance } from "axios";

const apiClient: AxiosInstance = axios.create({
  baseURL: "http://19:58.34.25:5001/api/v1",
  headers: {
    "Content-type": "application/json",
  },
});

export default apiClient;
--
--
-- INSERT --

```

รูปที่ 2.9 แก้ไขไอพีให้ตรงกับคอนโทรลเลอร์

กลับไป Directory SDN/sdn_frontendv3 ลบตัว node modules เก่าและลงใหม่โดยใช้คำสั่ง

1. `rm -rf node_modules/`
2. `npm install`

```

[root@localhost sdn]#
[root@localhost sdn]#
[root@localhost sdn]#
[root@localhost sdn]#
[root@localhost sdn]#
[root@localhost sdn]#
[root@localhost sdn]# yarn serve
yarn run v1.22.19
$ vue-cli-service serve
/bin/sh: /root/.nvm/versions/node/v14.18.0/bin/vue-cli-service: Permission denied
error Command failed with exit code 126.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.
[root@localhost sdn]# cd ..
[root@localhost sdn]# yarn serve
yarn run v1.22.19
$ vue-cli-service serve
/bin/sh: /root/.nvm/versions/node/v14.18.0/bin/vue-cli-service: Permission denied
error Command failed with exit code 126.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.
[root@localhost sdn_frontendv3]# rm -rf node_modules/
[root@localhost sdn_frontendv3]# npm install
npm WARN EBADENGINE Unsupported engine {
  npm WARN EBADENGINE   package: 'amplify@5.2',
  npm WARN EBADENGINE   required: { node: '>=8.8 <9' },
  npm WARN EBADENGINE   current: { node: 'v16.15.1', npm: '8.11.0' }
}
npm WARN EBADENGINE
npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with 15x less dependencies.
npm WARN deprecated uuid@3.3.2: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

added 227 packages, removed 151 packages, changed 82 packages, and audited 1627 packages in 28s

116 packages are looking for funding
  run `npm fund` for details

46 vulnerabilities (7 moderate, 25 high, 14 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
[root@localhost sdn_frontendv3]#

```

รูปที่ 2.10 ติดตั้ง npm สำเร็จ

2.4 การติดตั้งโมดูลอื่น ๆ ที่จำเป็น

ผู้ใช้งานจำเป็นต้องติดตั้ง โมดูลที่เหลือนบางส่วนเพื่อให้คอนโทรลเลอร์สามารถใช้งานได้ โดยให้ผู้ใช้ไปยัง Directory SDN-handmade_v3/backend/src และเรียกใช้คำสั่งดังต่อไปนี้

1. pip3.6 install --upgrade pip
2. pip3 install -r requirements.txt
3. pip3 install -U sanic-cors
4. python3 -m pip install --upgrade textfsm==0.4.1
5. pip3 install requests
6. yum install python3-tkinter

```
Collecting setuptools>=40.8
  Downloading setuptools-59.6.0-py3-none-any.whl (952 kB)
  | 952 kB 90.7 MB/s
Collecting ply
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
  | 49 kB 2.2 MB/s
Collecting future
  Downloading future-0.18.2.tar.gz (829 kB)
  | 829 kB 96.3 MB/s
  Preparing metadata (setup.py) ... done
Collecting cffi-1.15.0-cp36-cp36m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (485 kB)
  | 485 kB 93.3 MB/s
Collecting idna
  Downloading idna-3.3-py3-none-any.whl (61 kB)
  | 61 kB 5.9 MB/s
Collecting pycparser
  Downloading pycparser-2.21-py2.py3-none-any.whl (118 kB)
  | 118 kB 90.6 MB/s
Collecting immutables>=0.9
  Downloading immutables-0.10-cp36-cp36m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl (115 kB)
  | 115 kB 90.4 MB/s
Collecting typing_extensions>=3.7.4.3
  Downloading typing_extensions-4.1.1-py3-none-any.whl (26 kB)
Using legacy 'setup.py install' for netatlas, since package 'wheel' is not installed.
Using legacy 'setup.py install' for networkx, since package 'wheel' is not installed.
Using legacy 'setup.py install' for future, since package 'wheel' is not installed.
Using legacy 'setup.py install' for contextvars, since package 'wheel' is not installed.
Installing collected packages: typing_extensions, immutables, contextvars, sniffio, rfc3986, pycparser, idna, h11, httpcore, cffi, certifi, websockets, uvloop, ujson, six, pyyaml, multidict, httpx, httptools, cryptography, bcrypt, aiofiles, setuptools, sanic, ply, paramiko, future, textfsm, scp, sanic-plugins-framework, pyserial, pytz, python-dateutil, pyyaml, pycryptodome, pyserial, numpy, kiwisolver, decorator, cyclical, Sanic-Cors, pyserial, pymongo, networkx, netatlas, netaddr, matplotlib
Running setup.py install for contextvars ... done
Attempting uninstall: setuptools
  Found existing installation: setuptools 39.2.0
  Uninstalling setuptools-39.2.0:
  Successfully uninstalled setuptools-39.2.0
Running setup.py install for future ... done
Running setup.py install for networkx ... done
Running setup.py install for netatlas ... done
Successfully installed Sanic-Cors-0.9.3 aiofiles-0.8.0 bcrypt-3.2.2 certifi-2022.6.15 cffi-1.15.0 contextvars-2.4 cryptography-37.0.2 cyclical-0.11.0 decorator-5.1.1 future-0.18.2 h11-0.9.0 httpcore-0.11.1 httpx-0.4.0 httptools-0.4.0 idna-3.3 immutables-0.10 kiwisolver-1.3.1 matplotlib-3.7.0 multidict-5.2.0 netaddr-0.7.19 netatlas-2.1.1 networkx-2.4 numpy-1.19.5 paramiko-2.11.0 ply-3.11 pyserial-0.4.0 pycparser-2.21 pycryptodome-3.14.1 pymongo-3.6.1 pyyaml-1.5.0 pyparsing-3.0.9 pyserial-0.4.0 pytz-2022.1 python-dateutil-2.8.2 pytz-2022.1 pyyaml-1.5.0 rfc3986-1.5.0 sanic-20.12.6 sanic-plugins-framework-0.9.5 scp-0.14.4 setuptools-59.6.0 six-1.16.0 sniffio-1.2.0 textfsm-1.1.2 typing_extensions-4.1.1 ujson-4.3.0 uvloop-0.14.0 websockets-9.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a v
irtual environment instead: https://pip.pypa.io/en/latest/using/virtual-environments/
root@localhost src#
```

รูปที่ 2.11 ติดตั้ง requirement สำเร็จ

```

Downloading Sanic_Cors-1.8.1-py2.py3-none-any.whl (17 kB)
Downloading Sanic_Cors-1.8.0-py2.py3-none-any.whl (17 kB)
Requirement already satisfied: sanic-plugin-framework<0.9.0 in /usr/local/lib/python3.6/site-packages (from sanic-cors) (0.9.5)
Requirement already satisfied: sanic<10.12.0 in /usr/local/lib/python3.6/site-packages (from sanic-cors) (20.12.6)
Requirement already satisfied: httpcore<0.15.4 in /usr/local/lib/python3.6/site-packages (from sanic-cors) (0.15.4)
Requirement already satisfied: aiofiles<0.6.0 in /usr/local/lib/python3.6/site-packages (from sanic-cors) (0.9.0)
Requirement already satisfied: ujson<0.5.3 in /usr/local/lib/python3.6/site-packages (from sanic-cors) (0.14.0)
Requirement already satisfied: ujson<1.35 in /usr/local/lib/python3.6/site-packages (from sanic-cors) (4.3.0)
Requirement already satisfied: websockets<9.1.0 in /usr/local/lib/python3.6/site-packages (from sanic-cors) (9.1)
Requirement already satisfied: httptools<0.9.10 in /usr/local/lib/python3.6/site-packages (from sanic-cors) (0.4.0)
Requirement already satisfied: multidict<6.0.0 in /usr/local/lib/python3.6/site-packages (from sanic-cors) (5.2.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.6/site-packages (from httpcore<0.15.4>sanic-cors) (2022.6.15)
Requirement already satisfied: httpcore<0.11.0 in /usr/local/lib/python3.6/site-packages (from httpcore<0.15.4>sanic-cors) (0.11.1)
Requirement already satisfied: rfc3986(idna2008)<2.0.1.3 in /usr/local/lib/python3.6/site-packages (from httpcore<0.15.4>sanic-cors) (1.5.0)
Requirement already satisfied: sniffio in /usr/local/lib/python3.6/site-packages (from httpcore<0.15.4>sanic-cors) (1.2.0)
Requirement already satisfied: h11<0.10.0 in /usr/local/lib/python3.6/site-packages (from httpcore<0.11.0>httpcore<0.15.4>sanic-cors) (0.9.0)
Requirement already satisfied: contextvars<2.1 in /usr/local/lib/python3.6/site-packages (from sniffio->httpcore<0.15.4>sanic-cors) (2.4)
Requirement already satisfied: setuptools<40.0 in /usr/local/lib/python3.6/site-packages (from sanic-plugin-framework<0.9.0>sanic-cors) (59.6.0)
Requirement already satisfied: idna in /usr/local/lib/python3.6/site-packages (from rfc3986(idna2008)<2.0.1.3>httpcore<0.15.4>sanic-cors) (3.3)
Requirement already satisfied: immutables<0.9 in /usr/local/lib/python3.6/site-packages (from contextvars<2.1>sniffio->httpcore<0.15.4>sanic-cors) (0.10)
Requirement already satisfied: typing-extensions<3.7.4.3 in /usr/local/lib/python3.6/site-packages (from immutables<0.9>contextvars<2.1>sniffio->httpcore<0.15.4>sanic-cors) (4.1.1)
Installing collected packages: sanic-cors
  attempting uninstall: sanic-cors
    Found existing installation: Sanic-Cors 0.9.3
    Uninstalling Sanic-Cors-0.9.3:
      Successfully uninstalled Sanic-Cors-0.9.3
Successfully installed sanic-cors-0.10.0.post3
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a v
irtual environment instead: https://pip.pypa.io/environments/venv/
root@localhost src# python3 -m pip install --upgrade textfsm==0.4.1
Collecting textfsm==0.4.1
  Downloading textfsm-0.4.1.tar.gz (30 kB)
  Preparing metadata (setup.py) ... done
Using legacy 'setup.py install' for textfsm, since package 'wheel' is not installed.
Installing collected packages: textfsm
  attempting uninstall: textfsm
    Found existing installation: textfsm 1.1.2
    Uninstalling textfsm-1.1.2:
      Successfully uninstalled textfsm-1.1.2
  Running setup.py install for textfsm ... done
Successfully installed textfsm-0.4.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a v
irtual environment instead: https://pip.pypa.io/environments/venv/
root@localhost src#

```

รูปที่ 2.12 ติดตั้ง sanic-cors และ textfsm สำเร็จ

2.5 การเปิดใช้งานคอนโทรลเลอร์

1. เปิดใช้งานฐานข้อมูลโดยคำสั่ง **systemctl start mongod**
2. เปิดใช้งานคอนโทรลเลอร์โดย ไปที่ Directory /SDN-handmade_v3/backend/src พิมพ์คำสั่ง **python3 main.py**
3. เปิดใช้งาน โมดูลเอฟไอโอโดย ไปที่ Directory /SDN-handmade_v3/backend/src พิมพ์คำสั่ง **python3 main_web.py**
4. เปิดใช้งานหน้าเว็บโดย ไปที่ Directory /SDN-handmade_v3/sdn_frontend พิมพ์คำสั่ง **yarn serve** (กรณีเปิดใช้งานหน้าเว็บมีปัญหาให้ลบ node_modules และลง npm ใหม่)

```

SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:20:55 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:20:59 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:20:59 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:02 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:02 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:06 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:06 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:09 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:10 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:13 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:13 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:17 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:17 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:20 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:21 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:24 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:24 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:28 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:28 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:31 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:32 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:35 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:35 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:39 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:39 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:42 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:43 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:46 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:46 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
(2022-06-21 00:21:50 INFO) [MainProcess-SSH-WORKER]: 0
SDN Handmade (0.0.1 Beta)# -----
(2022-06-21 00:21:50 INFO) [MainProcess-SSH-WORKER]: Policy monitor task is running...
SDN Handmade (0.0.1 Beta)#

```

รูปที่ 2.13 เปิดใช้งานคอนโทรลเลอร์สำเร็จ

```

CentOS Linux 7 (Core)
Kernel 3.10.0-1160.66.1.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Tue Jun 21 07:29:29 on tty1
root@localhost ~#
root@localhost ~# cd sdn-handmade_v3/backend/src/
root@localhost src# python3 main_web.py
default
/usr/local/lib/python3.6/site-packages/paramiko/transport.py:33: CryptographyDeprecationWarning: Python 3.6 is no longer supported by the Python core team. Ther
efore, support for it is deprecated in cryptography and will be removed in a future release.
  from cryptography.hazmat.backends import default_backend
(2022-06-21 00:14:24 -0400) [28544] [INFO] Goin' Fast @ http://0.0.0.0:5001
(2022-06-21 00:14:24 INFO) [Process-1-MainThread]: Goin' Fast @ http://0.0.0.0:5001
(2022-06-21 00:14:24 INFO) [Process-1-MainThread]: 28544:28525 -> (get) app
(2022-06-21 00:14:24 -0400) [28544] [INFO] Starting worker [28544]
(2022-06-21 00:14:24 INFO) [Process-1-MainThread]: Starting worker [28544]

```

รูปที่ 2.14 เปิดใช้งานโมดูลเอพียูสำเร็จ

■ 4 problems (0 errors, 4 warnings)

Module Warning (from ./node_modules/eslint-loader/index.js):

```

/root/.SDN-handmade_v3/sdn_frontend3/src/views/home.vue
240:35 warning Unexpected any. Specify a different type @typescript-eslint/no-explicit-any
241:11 warning 'selectedNodes' is assigned a value but never used @typescript-eslint/no-unused-vars
242:11 warning 'selectedNode' is assigned a value but never used @typescript-eslint/no-unused-vars
245:11 warning 'tooltip' is assigned a value but never used @typescript-eslint/no-unused-vars
261:32 warning Unexpected any. Specify a different type @typescript-eslint/no-explicit-any
277:79 warning Unexpected any. Specify a different type @typescript-eslint/no-explicit-any
294:73 warning Unexpected any. Specify a different type @typescript-eslint/no-explicit-any
294:79 warning 'layouts' is defined but never used @typescript-eslint/no-unused-vars
294:87 warning Unexpected any. Specify a different type @typescript-eslint/no-explicit-any
310:11 warning 'eventHandlers' is assigned a value but never used @typescript-eslint/no-unused-vars
323:26 warning '...' is defined but never used @typescript-eslint/no-unused-vars

```

■ 11 problems (0 errors, 11 warnings)

Module Warning (from ./node_modules/eslint-loader/index.js):

```

/root/.SDN-handmade_v3/sdn_frontend3/src/views/initialization.vue
18:10 warning 'defineComponent' is defined but never used @typescript-eslint/no-unused-vars
18:27 warning 'onMounted' is defined but never used @typescript-eslint/no-unused-vars
18:30 warning 'reactive' is defined but never used @typescript-eslint/no-unused-vars
18:40 warning 'watchEffect' is defined but never used @typescript-eslint/no-unused-vars
18:66 warning 'watch' is defined but never used @typescript-eslint/no-unused-vars
23:3 warning Missing return type on function @typescript-eslint/explicit-module-boundary-types

```

■ 6 problems (0 errors, 6 warnings)

You may use special comments to disable some warnings.
 Use `// eslint-disable-next-line` to ignore the next line.
 Use `/* eslint-disable */` to ignore all warnings in a file.

App running at:
 - Local: <http://localhost:8080/>
 - Network: <http://10.50.34.25:8080/>

Note that the development build is not optimized.
 To create a production build, run `yarn build`.

No issues found.

-

รูปที่ 2.15 เปิดใช้งานหน้าเว็บสำเร็จ

บทที่ 3

การเชื่อมต่อคอนโทรลเลอร์กับอุปกรณ์เครือข่าย

การที่ระบบคอนโทรลเลอร์จะสามารถติดต่อกับอุปกรณ์เครือข่ายได้นั้น จะเกิดขึ้นจากการที่ผู้ใช้ตั้งค่าอุปกรณ์เครือข่ายเพื่อให้ระบบคอนโทรลเลอร์สามารถส่งคำสั่งเพื่อตั้งค่า SNMP ให้แก่อุปกรณ์เครือข่ายก่อนที่ Collector module (SNMP และ NetFlow) จะสามารถดึงข้อมูลของเราเตอร์เข้ามาเก็บไว้เพื่อการประมวลผลในฐานข้อมูลได้ และในขณะเดียวกันคอนโทรลเลอร์ก็ต้องทราบว่าจะต้องติดต่อไปยัง อุปกรณ์ใน subnet ไค และผ่านเส้นทางใด รวมถึงมีอุปกรณ์เครือข่ายใดอยู่ในระบบบ้าง

3.1 การตั้งค่าอุปกรณ์เครือข่าย

ระบบคอนโทรลเลอร์ถูกออกแบบมาให้สามารถตั้งค่า SNMP ให้กับอุปกรณ์เครือข่ายผ่านทาง การเชื่อมต่อระยะไกลผ่านโพรโทคอล Secure shell (SSH) ดังนั้นหลังจากที่ผู้ใช้ได้ทำการ กำหนดที่อยู่ไอพี และกำหนดวิธีการค้นหาเส้นทางเพื่อเชื่อมต่ออุปกรณ์เครือข่ายเข้าด้วยกันแล้ว ผู้ใช้จำเป็นจะต้องตั้งค่าอุปกรณ์เครือข่ายทั้งหมด ผ่านชุดคำสั่งเหล่านี้เพื่อให้ระบบคอนโทรลเลอร์ สามารถส่งคำสั่งผ่านโพรโทคอล SSH ได้อย่างถูกต้อง

```
Router(config)#ip domain-name pcn //ผู้ใช้งานตั้งชื่อโดเมนเนมได้อย่างอิสระ (ในตัวอย่างนี้ถูกตั้งเป็น pcn)
```

```
Router (config)#crypto key generate rsa
```

```
How many bits : 1024
```

```
Router (config)#username cisco password cisco
```

```
Router (config)#ip ssh version 2 //ผู้ใช้งานสามารถตั้ง username และ password ได้อย่างอิสระ แต่ในตัวอย่างจะใช้ทั้ง username และ password เป็น cisco
```

```
Router (config)#enable password cisco
```

```
Router (config)#line vty 0 xxx //กำหนดจำนวน channel ในการเชื่อมต่อโดยค่า xxx ผู้พัฒนาแนะนำให้ตั้งค่าสูงสุดที่เป็นไปได้เพื่อให้ระบบคอนโทรลเลอร์สามารถติดต่อกับเราเตอร์ได้อย่างมีประสิทธิภาพ
```

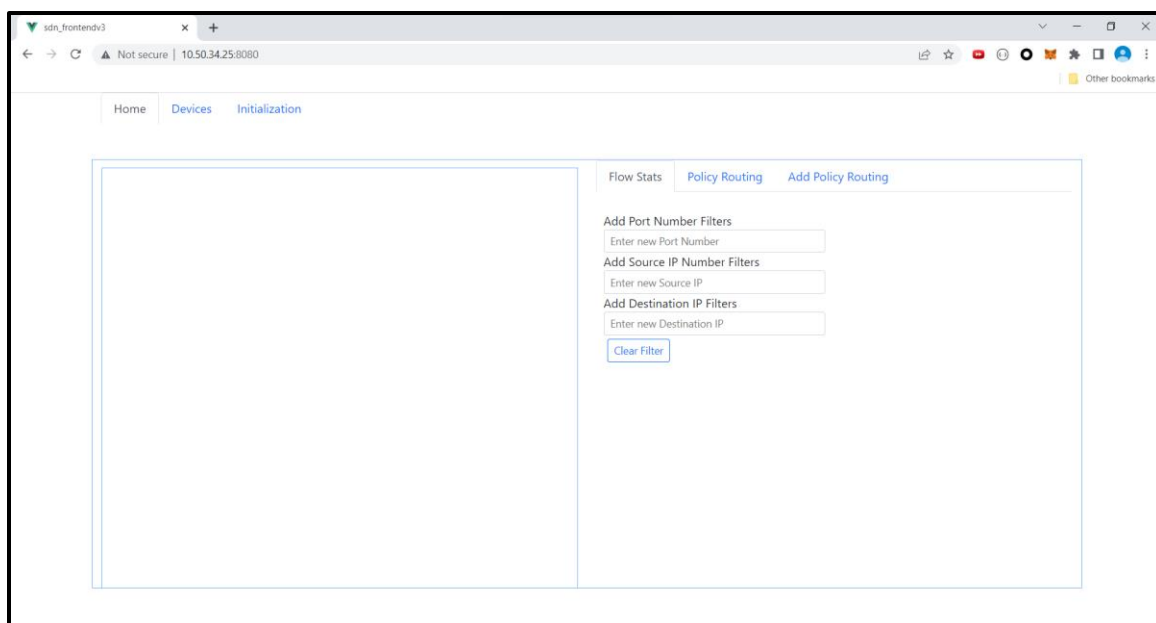
```
Router (config-line)#login local
```

```
Router (config-line)#transport input ssh
```

```
Router (config-line)#exit
```


3.2 การใช้งานหน้าเว็บเพื่อควบคุมคอนโทรลเลอร์

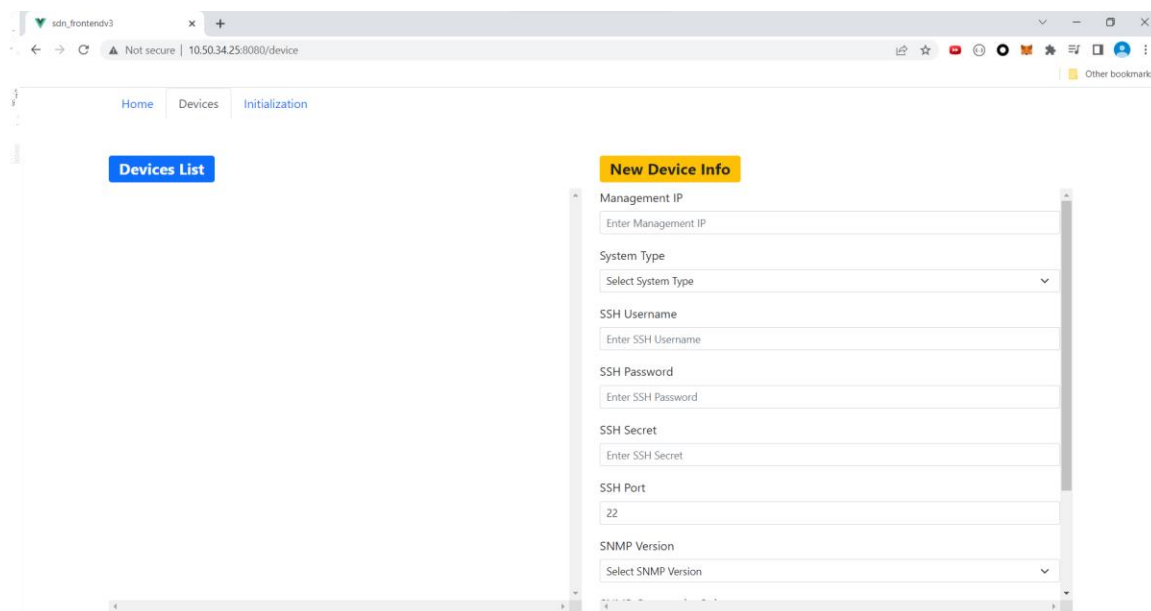
ในการเข้าถึงหน้าเว็บเพื่อใช้ควบคุมระบบคอนโทรลเลอร์นั้น ผู้ใช้สามารถเปิดใช้งานเบราว์เซอร์ใด ๆ และพิมพ์ไอพีของเครื่องเซิร์ฟเวอร์ที่ทำการเปิดโมดูลทั้งหมดเอาไว้ตามด้วยหมายเลขพอร์ตที่ถูกตั้งเอาไว้ตามหัวข้อ 2.4 ในตัวอย่างตามรูปที่ 3.1 หมายเลขพอร์ตที่กำหนดไว้ค่าเท่ากับ 8080 และไอพีของเครื่องเซิร์ฟเวอร์คือ 10.50.34.25



รูปที่ 3.1 ตัวอย่างหน้าเว็บไซต์สำหรับแสดงผล

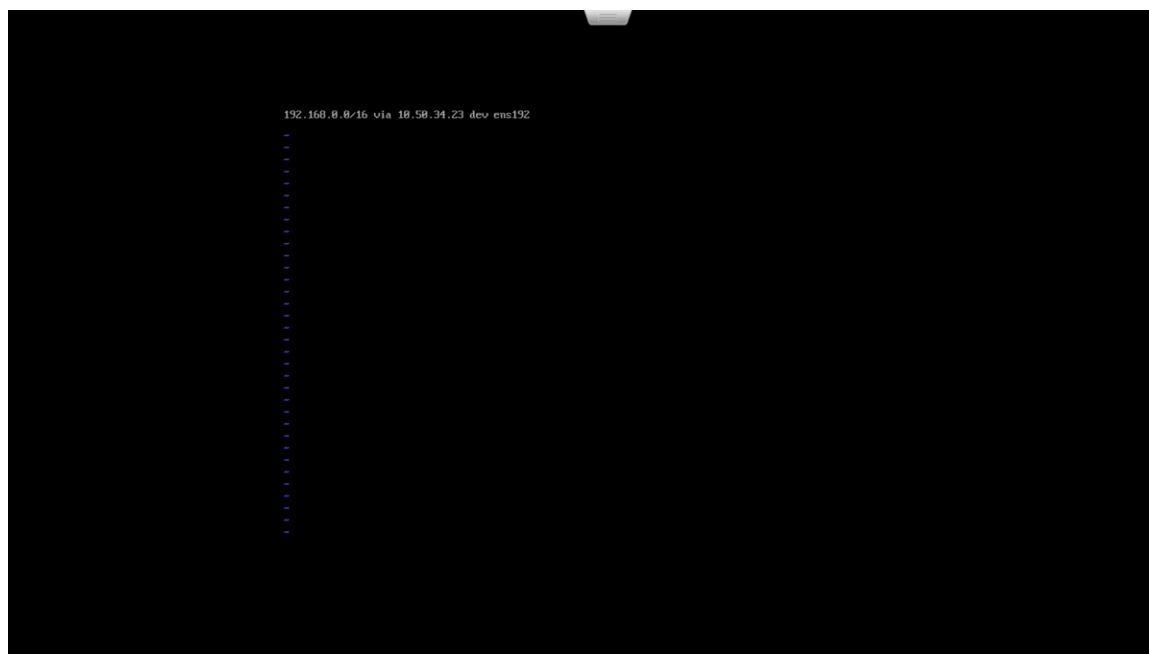
เพื่อให้หน้าเว็บสามารถเข้าถึงอุปกรณ์เครือข่ายได้ ให้ผู้ใช้เลือกไปที่แท็บ “Devices” ซึ่งก็จะปรากฏหน้าเว็บตามรูปที่ 3.2 โดยผู้ใช้งานจะต้องทำการเพิ่มข้อมูลอุปกรณ์เครือข่ายทั้งหมดลงไปในฟอร์มทางขวามือ โดยกำหนด SSH username และ SSH password ให้ตรงกับที่ตั้งไว้ในอุปกรณ์ตามหัวข้อที่ 3.1 โดยรายละเอียดอื่น ๆ มีดังต่อไปนี้

1. ใส่หมายเลข Management IP ของตัว Router
2. ใส่ประเภทอุปกรณ์เครือข่าย ซึ่งปัจจุบันรองรับเพียง Cisco IOS
3. ใส่ชื่อผู้ใช้งานสำหรับการเข้าถึงอุปกรณ์เครือข่ายผ่าน SSH (SSH Username)
4. ใส่รหัสผ่านสำหรับการเข้าถึงอุปกรณ์เครือข่ายผ่าน SSH (SSH Password)
5. ใส่รหัสการเปิดใช้งานอุปกรณ์เครือข่าย (Enable Password)
6. ใส่หมายเลขพอร์ต SSH ซึ่งค่าเริ่มต้นคือ 22
7. ใส่เวอร์ชันของ SNMP ซึ่งปัจจุบันรองรับเพียงเวอร์ชัน 2c
8. ใส่ค่า SNMP Community string ซึ่งค่าเริ่มต้นคือ public
9. ใส่หมายเลขพอร์ตของ SNMP ซึ่งค่าเริ่มต้นคือ 161



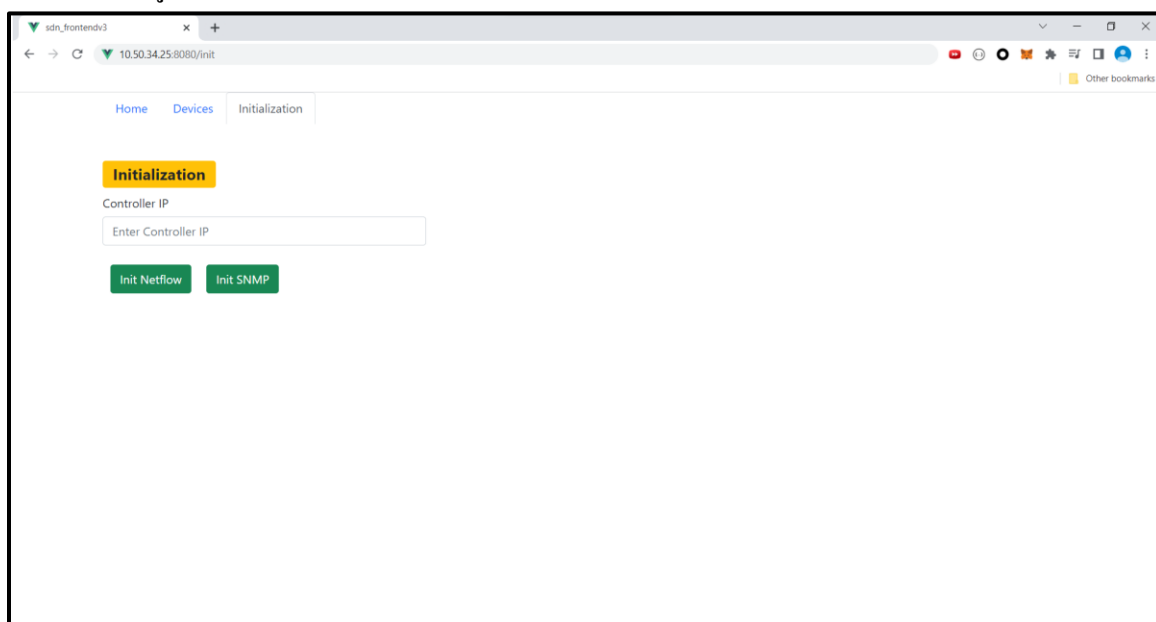
รูปที่ 3.2 ตัวอย่างหน้าเว็บไซต์สำหรับแสดงผลหน้า Devices

จากนั้น เพื่อให้ระบบสามารถส่งคำสั่งเข้าไปตั้งค่าอุปกรณ์เครือข่ายได้ ผู้ใช้ต้องทำการตั้งค่าเราท์ติ้งเพื่อไปยัง Subnet ของอุปกรณ์เครือข่าย ผ่านคำสั่ง `vi /etc/sysconfig/network-scripts/route-ens192` โดย `ens192` คือชื่ออินเตอร์เฟซ จากตัวอย่างตามรูปที่ 3.3 ผู้ใช้ได้ทำการตั้งค่าไอพี ของอุปกรณ์เครือข่ายเอาไว้ในกลุ่ม Subnet 192.168.0.0/16 โดยคอนโทรลเลอร์จะสามารถติดต่อกับ อุปกรณ์ผ่านอินเตอร์เฟซที่มีหมายเลขไอพีคือ 10.50.34.23 (Gateway)

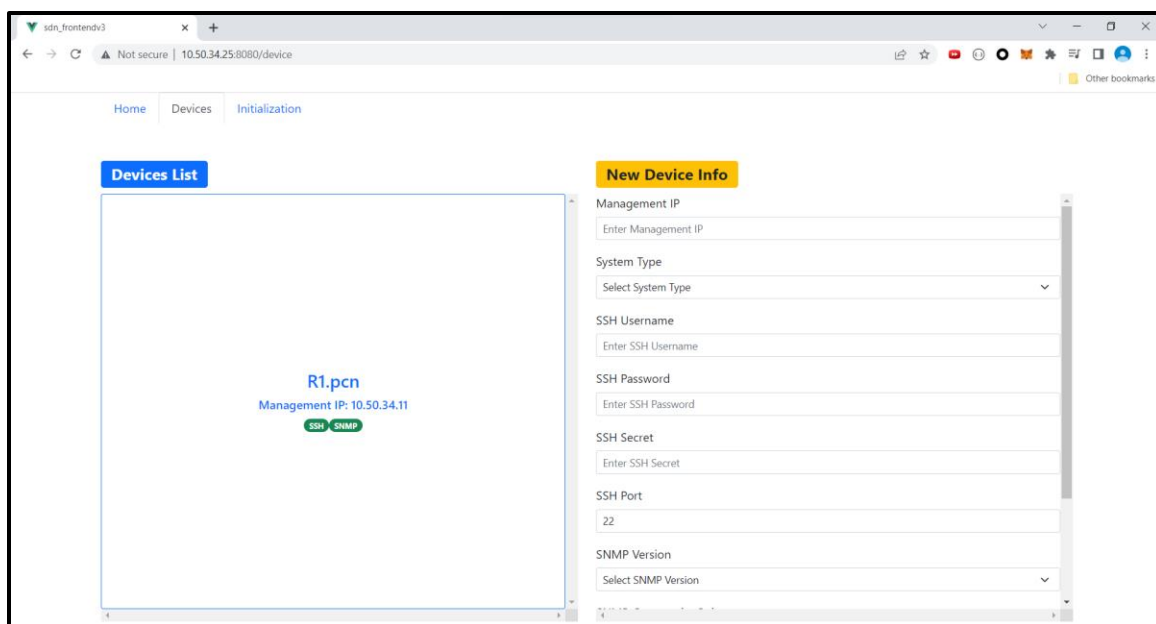


รูปที่ 3.3 ตัวอย่างการตั้งค่าเราท์ติ้ง

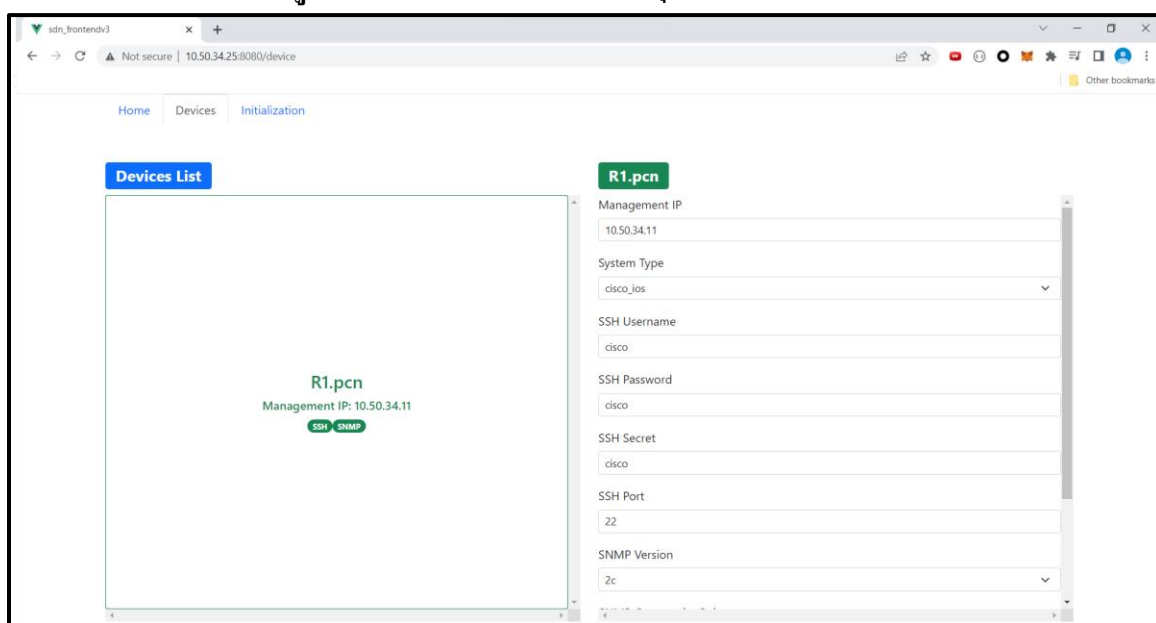
ในขั้นต่อไปให้ผู้ใช้เลือกไปที่แท็บ “Initialization” จะปรากฏหน้าตามรูปที่ 3.4 จากนั้นใส่ข้อมูลหมายเลขไอพีของเครื่องที่ทำหน้าที่เป็นเซิร์ฟเวอร์ให้ระบบคอนโทรลเลอร์แล้วกด คลิกที่ปุ่ม “Init SNMP” เพื่อส่งคำสั่งในการตั้งค่า SNMP ไปยังอุปกรณ์ทั้งหมดที่ได้ทำการเพิ่ม เอาไว้ในหน้าเว็บ เมื่อระบบเชื่อมต่อกับอุปกรณ์สำเร็จ และตั้งค่าอุปกรณ์เครือข่ายสำเร็จแล้วหน้าเว็บจะสามารถแสดงผลในแท็บ “Devices” ดังรูปที่ 3.5 โดยที่สถานะ SSH และ SNMP ต้องขึ้นเป็นไอคอนสีเขียว โดยผู้ใช้สามารถกดดูข้อมูลของอุปกรณ์เครือข่ายใด ๆ ได้โดยกดไปที่แถบข้อมูลของ อุปกรณ์เครือข่ายนั้น ซึ่งจะแสดงผลตามรูปที่ 3.6



รูปที่ 3.4 ตัวอย่างหน้าเว็บไซต์สำหรับแสดงผลหน้า Initialization

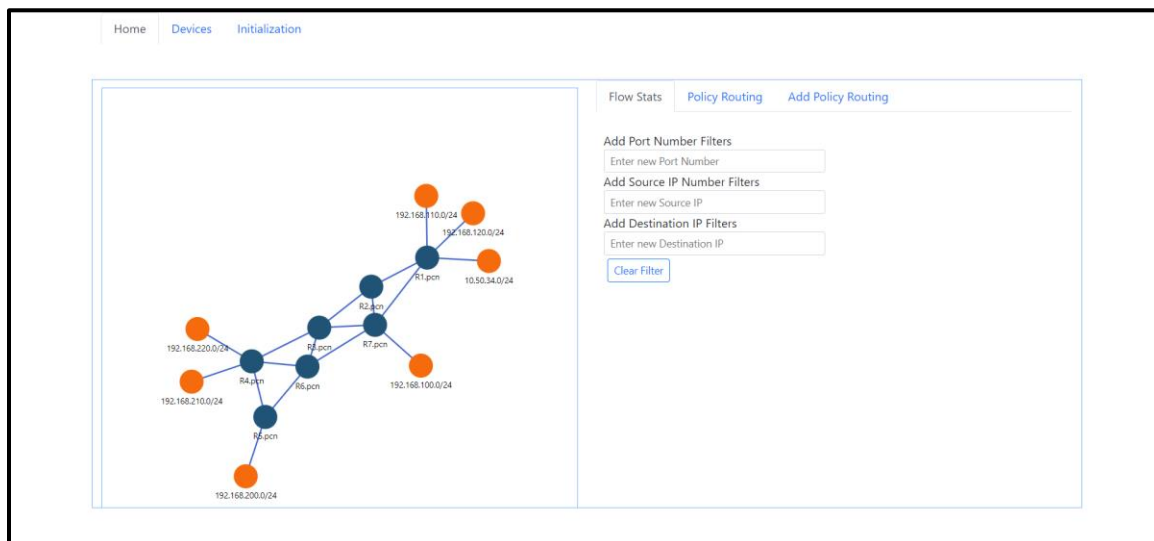


รูปที่ 3.5 หน้า Devices แสดงผลอุปกรณ์ที่เพิ่มเข้ามา



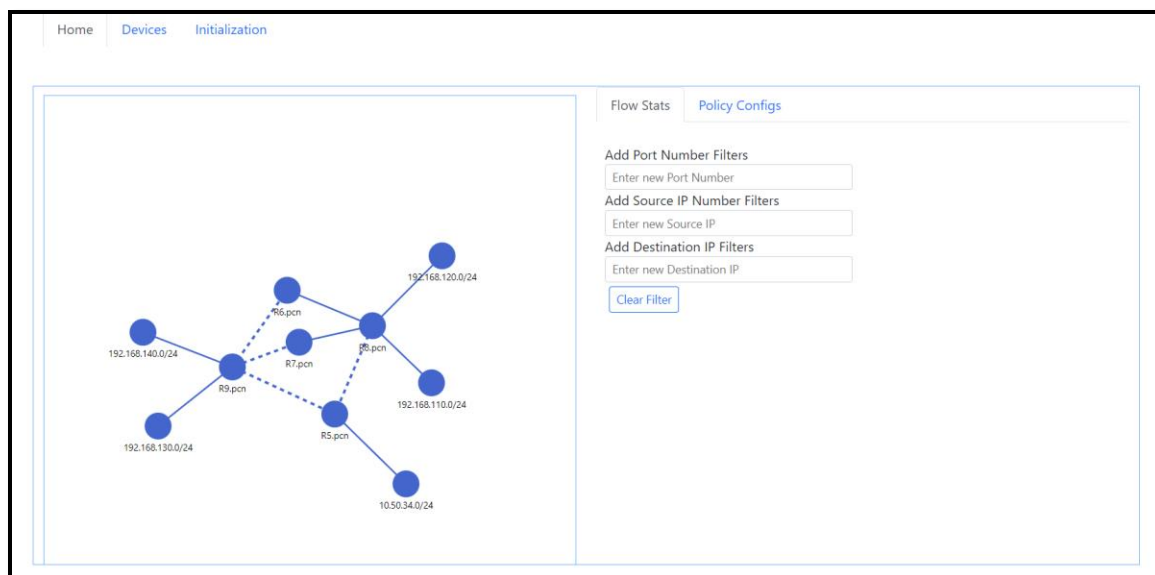
รูปที่ 3.6 หน้า Devices สามารถกดที่อุปกรณ์เพื่อดูรายละเอียด

เมื่อระบบสามารถเชื่อมต่อกับอุปกรณ์ทั้งหมดได้สำเร็จ ระบบจะสามารถแสดงโทโพโลยีของเครือข่ายออกมาได้โดยให้ผู้ใช้เลือกไปที่แท็บ “Home” ซึ่งจะแสดงผลออกมาตามรูปที่ 3.7

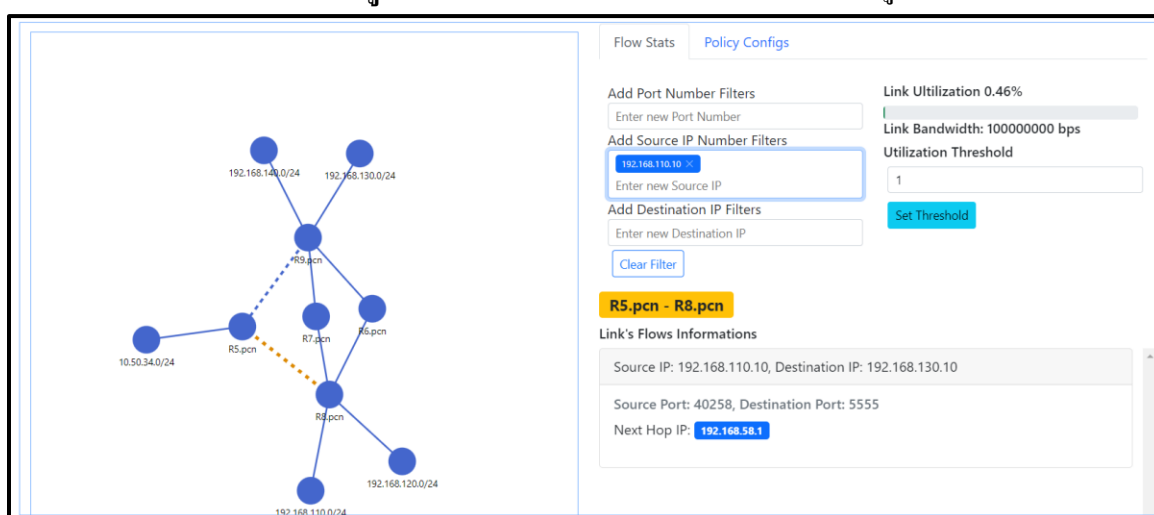


รูปที่ 3.7 ตัวอย่างการแสดงผลโทโพโลยีเครือข่าย

ข้อมูลที่วิ่งอยู่ในอุปกรณ์เครือข่ายได้ผู้ใช้ต้องเลือก ไปที่แท็บ “Initialization” และกดเลือกไปที่หัวข้อ “Initialization” ตามรูปที่ 3.4 จากนั้นใส่ข้อมูล หมายเลขไอพีของเครื่องที่ทำหน้าที่เป็นเซิร์ฟเวอร์ให้ระบบคอนโทรลเลอร์แล้วกดที่ปุ่ม “Init Netflow” เพื่อให้ระบบคอนโทรลเลอร์สามารถดึงข้อมูลของโพล์มาแสดงได้ (ขั้นตอนนี้อาจจำเป็นต้องทำหลังจากการตั้งค่า SNMP เรียบร้อยแล้วเท่านั้น) โดยผู้ใช้สามารถกดไปที่แท็บ “Home” จะแสดงเอฟเฟกการเคลื่อนไหวซึ่งแสดงให้เห็นว่าลิงก์ใดมีการไหลของโพล์เกิดขึ้น โดยผู้ใช้สามารถกดเข้าไปในลิงก์เพื่อดูข้อมูลโพล์ตามรูปที่ 3.8 และสามารถใส่ filter เพื่อกรองเฉพาะข้อมูลโพล์ที่ต้องการตามรูปที่ 3.9



รูปที่ 3.8 หน้าเว็บแสดงผลลิงก์ที่มีโฟลว์ไหลอยู่



รูปที่ 3.9 ผู้ใช้สามารถคลิกที่ลิงก์เพื่อแสดงข้อมูลโฟลว์ และกรองโฟลว์เพื่อแสดงผลตามที่ผู้ใช้งานต้องการได้

3.3 การตั้งค่าเริ่มต้นให้อุปกรณ์เครือข่ายผ่านการเรียกใช้คำสั่งจากโค้ดไพธอนโดยตรง

นอกจากการใช้งานผ่านหน้าเว็บแล้ว ผู้ใช้สามารถเรียกใช้ฟังก์ชันที่มีในหน้าเว็บผ่านทางเอพีไอโดยตรงได้เช่นกัน ซึ่งสามารถศึกษาวิธีเรียกใช้งานผ่านไฟล์ `add_device.py` ที่อยู่ใน directory `SDN-handmade_v3/Fame_test` ไฟล์ประกอบไปด้วยฟังก์ชันต่างๆ และตัวอย่างการใช้งาน ผู้ใช้สามารถแก้ไขตัวแปร `device_list` ตามชุดไอพีอุปกรณ์ที่มี และทดลองรันโปรแกรมเพื่อทดสอบใช้งานฟังก์ชันต่างๆ ที่มีในหน้าเว็บปกติได้

บทที่ 4

ข้อมูลชุดคำสั่งเอพียูที่สามารถใช้งานได้และตัวอย่างการใช้งาน

การใช้งาน API module ผู้ใช้จะต้องเข้าผ่านเว็บเบราว์เซอร์โดยมีรูปแบบ URL ดังนี้

<http://ControllerIp:5001/api/v1/Command/>

ตัวอย่าง: <http://10.50.34.15:5001/api/v1/device>

1. Controller Ip เป็นหมายเลขไอพีของเครื่องเซิร์ฟเวอร์ที่รันระบบคอนโทรลเลอร์อยู่ (ในแลปนี้ใช้ไอพี 10.50.34.15)
2. Command เป็นกลุ่มชุดคำสั่งที่สามารถเรียกใช้ได้โดยตัวเลือกชุดคำสั่งมีดังต่อไปนี้
 - a. device
 - b. flow
 - c. link
 - d. path
 - e. flow/routing

4.1 รายละเอียดคำสั่ง “device”

ผู้ใช้งานสามารถเรียกดูข้อมูลผ่านทาง <http://10.50.34.15:5001/api/v1/device> โดยหน้านี้จะแสดงข้อมูลพื้นฐานอุปกรณ์ทั้งหมดในเครือข่าย เช่น ชื่ออุปกรณ์ ชื่อ รายละเอียดต่างๆที่เกี่ยวข้องกับอินเตอร์เฟซ ข้อมูล SSH และสถานะการทำงานของ SNMP CDP และ NetFlow โดยผู้ใช้งานสามารถเลือกดูข้อมูลอุปกรณ์ที่ต้องการแบบเฉพาะเจาะจงโดยการเพิ่ม /hostname ของอุปกรณ์ที่ต้องการลงไปต่อท้ายชุดคำสั่งข้างต้นได้ ตัวอย่าง <http://10.50.34.15:5001/api/v1/device/R1>



รูปที่ 4.1 ตัวอย่างการเรียกใช้ device เอพีไอ



รูปที่ 4.2 ผลลัพธ์จากการเรียกใช้โปรแกรมในรูปที่ 4.1

4.2 รายละเอียดคำสั่ง “flow”

ผู้ใช้งานสามารถเรียกดูข้อมูลผ่านทาง <http://10.50.34.15:5001/api/v1/flow> โดยหน้านี้จะ แสดงข้อมูลโฟลว์ที่ได้รับจากอุปกรณ์เครือข่าย เป็นรายละเอียดเกี่ยวกับข้อมูลที่มีการรับส่งผ่าน อุปกรณ์เครือข่าย โดยข้อมูลเหล่านี้สามารถนำมาประยุกต์ใช้ในการบังคับเปลี่ยนเส้นทางเพื่อทำ Network Traffic Engineering ได้


```
import requests

flow_info = requests.get("http://10.50.34.15:5001/api/v1/flow").json()
for flow in flow_info['flows']:
    print(flow)
```

รูปที่ 4.3 ตัวอย่างการเรียกใช้ flow เอพีไอ

```
{'_id': {'$oid': '6234bb956b3e4f9c9c7de928'}, 'cisco_51': 0, 'direction': 0, 'dst_as': 0, 'dst_mask': 24, 'flow_sampler_id': 0,
'from_ip': '192.168.2.2', 'input_snmp': 2, 'ipv4_dst_addr': '192.168.10.1', 'ipv4_next_hop': '192.168.3.2', 'ipv4_src_addr':
'10.50.34.15', 'l4_dst_port': 161, 'l4_src_port': 37786, 'output_snmp': 1, 'protocol': 17, 'src_as': 0, 'src_mask': 0, 'src_t
os': 0, 'tcp_flags': 16, 'last_switched': {'$date': 1647647504296}, 'first_switched': {'$date': 1647647504296}, 'in_bytes': 96
, 'in_pkts': 1, 'created_at': {'$date': 1647623061105}}
{'_id': {'$oid': '6234bb956b3e4f9c9c7de92e'}, 'cisco_51': 0, 'direction': 0, 'dst_as': 0, 'dst_mask': 24, 'flow_sampler_id': 0
, 'from_ip': '192.168.2.2', 'input_snmp': 2, 'ipv4_dst_addr': '192.168.9.1', 'ipv4_next_hop': '192.168.3.2', 'ipv4_src_addr':
'10.50.34.15', 'l4_dst_port': 161, 'l4_src_port': 52309, 'output_snmp': 1, 'protocol': 17, 'src_as': 0, 'src_mask': 0, 'src_t
os': 0, 'tcp_flags': 16, 'last_switched': {'$date': 1647647504296}, 'first_switched': {'$date': 1647647504296}, 'in_bytes': 96,
'in_pkts': 1, 'created_at': {'$date': 1647623061105}}
{'_id': {'$oid': '6234bb956b3e4f9c9c7de934'}, 'cisco_51': 0, 'direction': 0, 'dst_as': 0, 'dst_mask': 24, 'flow_sampler_id': 0
, 'from_ip': '192.168.2.2', 'input_snmp': 2, 'ipv4_dst_addr': '192.168.11.1', 'ipv4_next_hop': '192.168.3.2', 'ipv4_src_addr':
'10.50.34.15', 'l4_dst_port': 161, 'l4_src_port': 48077, 'output_snmp': 1, 'protocol': 17, 'src_as': 0, 'src_mask': 0, 'src_t
os': 0, 'tcp_flags': 16, 'last_switched': {'$date': 1647647504384}, 'first_switched': {'$date': 1647647504384}, 'in_bytes': 96
, 'in_pkts': 1, 'created_at': {'$date': 1647623061105}}
{'_id': {'$oid': '6234bb976b3e4f9c9c7deab7'}, 'cisco_51': 0, 'direction': 0, 'dst_as': 0, 'dst_mask': 24, 'flow_sampler_id': 0
, 'from_ip': '192.168.7.2', 'input_snmp': 2, 'ipv4_dst_addr': '192.168.12.1', 'ipv4_next_hop': '192.168.8.2', 'ipv4_src_addr':
'10.50.34.15', 'l4_dst_port': 161, 'l4_src_port': 41206, 'output_snmp': 1, 'protocol': 17, 'src_as': 0, 'src_mask': 0, 'src_t
```

รูปที่ 4.4 ผลลัพธ์จากการเรียกใช้โปรแกรมในรูปที่ 4.3

4.3 รายละเอียดคำสั่ง “link”

ผู้ใช้งานสามารถเรียกดูข้อมูลผ่านทาง `http://10.50.34.15:5001/api/v1/link` โดยหน้านี้จะ แสดง ข้อมูลการเชื่อมต่อของอุปกรณ์ สามารถรู้ได้ว่าอุปกรณ์กำลังต่ออยู่กับอุปกรณ์อื่นใดบ้าง และสามารถ แสดงข้อมูลปริมาณการใช้งานของแต่ละลิงก์ได้เช่นกัน

```
import requests

link_info = requests.get("http://10.50.34.15:5001/api/v1/link").json()
for link in link_info['links']:
    print('This link is connect between', link['src_node_hostname'], ' and', link['dst_node_hostname'])
    print('Connect with Port', link['src_port'], ' and', link['dst_port'])
```

รูปที่ 4.5 ตัวอย่างการเรียกใช้ link เอพีไอ

```

This link is connect between R1.pcn and R2.pcn
Connect with Port GigabitEthernet0/0 and GigabitEthernet0/1
This link is connect between R34.pcn and R1.pcn
Connect with Port GigabitEthernet0/0 and GigabitEthernet0/1
This link is connect between R2.pcn and R3.pcn
Connect with Port GigabitEthernet0/0 and GigabitEthernet0/1
This link is connect between R3.pcn and R4.pcn
Connect with Port GigabitEthernet0/0 and GigabitEthernet0/1
This link is connect between R4.pcn and R5.pcn
Connect with Port GigabitEthernet0/0 and GigabitEthernet0/1
This link is connect between R5.pcn and R6.pcn
Connect with Port GigabitEthernet0/0 and GigabitEthernet0/1
This link is connect between R6.pcn and R7.pcn
Connect with Port GigabitEthernet0/0 and GigabitEthernet0/1

```

รูปที่ 4.6 ผลลัพธ์จากการเรียกใช้โปรแกรมในรูปที่ 4.5

4.4 รายละเอียดคำสั่ง “path”

ผู้ใช้งานสามารถเรียกดูข้อมูลผ่านทาง `http://10.50.34.15:5001/api/v1/path/srcip,dstip` ตัวอย่าง `http://10.50.34.15:5001/api/v1/path/192.168.1.1,192.168.10.1` โดยหน้านี้จะ สามารถแสดงข้อมูล เส้นทางที่เป็นไปได้ทั้งหมด จากไอพีต้นทาง และไอพีปลายทาง

```

import requests

path_info = requests.get("http://10.50.34.15:5001/api/v1/path/192.168.1.1,192.168.10.1").json()
all_path = []
print('There are', len(path_info['paths']), 'possible paths')
for path in path_info['paths']:
    all_path.append(path['path'])
least_hop_path = min(all_path, key=len)
print('Lowest hop use =', len(least_hop_path), 'hops')
print('Path:', least_hop_path)

```

รูปที่ 4.7 ตัวอย่างการเรียกใช้ path เอพีไอ

```

There are 2 possible paths
Lowest hop use = 10 hops
Path: ['192.168.1.1', '192.168.2.1', '192.168.3.1', '192.168.4.1', '192.168.5.1', '192.168.6.1', '192.168.7.1', '192.168.8.1', '192.168.9.1', '192.168.10.1']

```

รูปที่ 4.8 ผลลัพธ์จากการเรียกใช้โปรแกรมในรูปที่ 4.7

4.5 รายละเอียดคำสั่ง “flow/routing”

ผู้ใช้สามารถจัดการ policy based routing สำหรับการตั้งค่าเส้นทางโดยเฉพาะตามที่ ต้องการได้ โดยผู้ใช้สามารถเรียกดู สร้าง และลบ policy ดังกล่าวผ่าน API นี้ได้โดย method GET POST DELETE ตามลำดับ ผ่าน URL `http://10.50.34.15:5001/api/v1/flow/routing` โดย พารามิเตอร์ที่ใช้สำหรับ DELETE และ POST method มีดังต่อไปนี้

Delete method parameter สำหรับลบ policy routing.

ผู้ใช้สามารถเรียกใช้ API โดยมี payload ดังต่อไปนี้

1. `flow_id`: ไอดี policy ที่ต้องการลบ โดยผู้ใช้สามารถดูไอดีผ่าน GET method ของ flow/routing API



รูปที่ 4.9 ตัวอย่าง Payload ของ DELETE Method

POST method parameter สำหรับสร้าง policy routing.

ผู้ใช้สามารถเรียกใช้ API โดยมี payload ดังต่อไปนี้

1. `name`: ชื่อ policy สามารถตั้งอะไรก็ได้
2. `src_ip/dst_ip`: ไอพีเครือข่ายต้นทาง และปลายทาง
3. `src_port/dst_port`: พอร์ตต้นทาง และปลายทาง หากเป็นพอร์ตอะไรก็ได้ให้ใส่ any
4. `src_subnet/dst_subnet`: ไรต์การ์ดต้นทาง และปลายทาง
5. `actions`: กำหนดอุปกรณ์ที่จะติดตั้ง policy และวิธีการรับมือของอุปกรณ์เหล่านั้นเมื่อมี Flow ตาม Policy ผ่านเข้ามา สามารถติดตั้งบนอุปกรณ์หลายตัวพร้อมกันได้โดยคำสั่งเดียว
 - a. `device_id`: ไอดีของอุปกรณ์ สามารถดูได้จาก API device

- b. action: วิธีการส่งข้อมูลตัวอย่างใช้ 2 หมายถึงส่งให้ Next Hop Ip (1 หมายถึง Next Hop Interface)
- c. data: ข้อมูล Next hop ต้องสอดคล้องกับ action ถ้า action เป็น 1 ให้ใส่ชื่อ Next Hop Interface เป็นต้น

```

payload = {
    'name': 'Test Policy',
    'src_ip': '192.168.1.1',
    'src_port': 'any',
    'src_subnet': '0.0.0.255',
    'dst_ip': '192.168.5.1',
    'dst_port': '8080',
    'dst_subnet': '0.0.0.255',
    'actions': [
        {'device_id': '62349df46b3',
         'action': 2,
         'data': '192.168.2.1'},
        {'device_id': 'atd54257s3',
         'action': 2,
         'data': '192.168.3.1'}
    ]
}

```

รูปที่ 4.10 ตัวอย่าง Payload ของ POST Method

```
import requests

src_net = '192.168.200.0'
src_port = 'any'
src_wildcard = '0.0.0.255'
dst_net = '192.168.201.0'
dst_port = 'any'
dst_wildcard = '0.0.0.255'

for router in requests.get("http://10.50.34.15:5001/api/v1/device").json()['devices']:
    if router['name'] == 'R1.pcn':
        router_id = router['_id']['$oid']

action = [{'device_id': router_id, 'action': 2, 'data': '192.168.2.1'}]
payload = {'name': 'Test_policy', 'src_ip': src_net, 'src_port': src_port, 'src_subnet': src_wildcard,
           'dst_ip': dst_net, 'dst_port': dst_port, \
           'dst_subnet': dst_wildcard, 'actions': action}
requests.post("http://10.50.34.15:5001/api/v1/flow/routing", json=payload)
```

รูปที่ 4.11 ตัวอย่างการใช้ POST method สำหรับสร้าง policy routing ตัวอย่างหาก Router R1 ได้รับโฟลว์ มีไอพีเครือข่ายต้นทางเป็น 192.168.200.0 ไอพีเครือข่ายปลายทางเป็น 192.168.201.0 พอร์ตต้นทางและปลายทางเป็นอะไรก็ได้ จะส่ง Flow ดังกล่าวไปยัง Next Hop IP ที่เป็น 192.168.2.1

```
import requests

all_policy = requests.get("http://10.50.34.15:5001/api/v1/flow/routing").json()['flows']
for policy in all_policy:
    if policy['name'] == 'Test_policy':
        policy_id = policy['flow_id']

payload = {'flow_id': policy_id}
requests.delete("http://localhost:5001/api/v1/flow/routing", params=payload)
```

รูปที่ 4.12 ตัวอย่างการใช้ DELETE method โดยการลบ policy ที่มีชื่อว่า Test_policy โดยดึง flow_id จาก GET method