



Project

Nokia SR Linux

จัดทำโดย

นายกิตินันท์ กุณโฮง	633040148-3
นายตรีเพชร ตรีจันทร์	633040156-4

Section 1

รายงานเล่มนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชา EN814805 MODERN COMPUTER
NETWORKS

ภาคการศึกษาที่ 2 ปีการศึกษา 2565

Nokia SR Linux เป็นระบบปฏิบัติการสำหรับเครือข่ายคอมพิวเตอร์ (network operating system) ที่ถูกออกแบบมาเพื่อใช้งานกับอุปกรณ์เครือข่ายต่าง ๆ รวมถึงเครือข่ายคอมพิวเตอร์ขนาดใหญ่ (data center) โดยเป็นผลิตภัณฑ์ของบริษัท Nokia ที่เน้นความเร็ว ประสิทธิภาพ และความยืดหยุ่นในการใช้งาน ระบบปฏิบัติการนี้ได้รับความนิยมจากผู้ใช้งานในวงกว้างเนื่องจากความสามารถในการจัดการเครือข่ายที่มีประสิทธิภาพสูง และการปรับแต่งตามความต้องการของลูกค้าได้ง่าย ๆ โดยไม่ต้องมีการเปลี่ยนแปลงฮาร์ดแวร์หรือโครงสร้างของเครือข่ายใหม่แบบสมบูรณ์ นอกจากนี้ Nokia SR Linux ยังมีความปลอดภัยสูง และมีการอัปเดตเพื่อรองรับการเปลี่ยนแปลงในเครือข่ายได้อย่างมีประสิทธิภาพสูง

Nokia SR Linux: มีข้อดีดังนี้

1. Nokia SR Linux ระบบปฏิบัติการนี้มีประสิทธิภาพสูงเป็นอย่างมากในการจัดการเครือข่าย มีความเร็วในการประมวลผลและการสื่อสารที่ดี
2. Nokia SR Linux สามารถใช้งานร่วมกับฮาร์ดแวร์ต่างๆได้ ซึ่งช่วยเพิ่มความสามารถในการทำงานและปรับปรุงระบบได้อย่างสม่ำเสมอ
3. Nokia SR Linux มีการออกแบบระบบบริหารจัดการที่ง่ายต่อการใช้งาน และช่วยลดการใช้งานทรัพยากรของระบบเครือข่ายได้
4. Nokia SR Linux มีความสามารถในการออกแบบโมดูลาร์ที่ช่วยในการปรับแต่งและตั้งค่าการทำงานของระบบได้อย่างอิสระ ส่งผลให้ Nokia SR Linux มีประสิทธิภาพการทำงานที่สูง
5. Nokia SR Linux มีความปลอดภัยสูงเนื่องจากด้วยการรองรับการใช้งานร่วมกับฮาร์ดแวร์ต่าง ๆ และความสามารถในการปรับแต่งการทำงานของระบบ

Nokia SR Linux: มีข้อเสียดังนี้

1. Nokia SR Linux มีความซับซ้อนและมีการปรับแต่งต่าง ๆ ได้อย่างอิสระ จึงต้องการความเชี่ยวชาญในการตั้งค่าและการดูแลรักษาระบบ ทำให้ระบบทำงานได้อย่างมีประสิทธิภาพและปลอดภัยดังนั้นผู้ใช้งานควรมีความรู้เรื่อง Nokia SR Linux พอสมควร
2. Nokia SR Linux เป็นระบบปฏิบัติการเครือข่ายระดับสูง ดังนั้นราคาจะต้องสูงกว่าระบบปฏิบัติการเครือข่ายระดับทั่วไป จึงทำให้ Nokia SR Linux จะต้องใช้งบประมาณสูงมากขึ้นเนื่องจากประสิทธิภาพการทำงานที่สูง
3. Nokia SR Linux อาจจะมีข้อจำกัดที่ไม่รองรับการทำงานของฮาร์ดแวร์บางรุ่น

ขั้นตอนการติดตั้ง Nokia SR Linux

ขั้นตอนที่ 1: ทำการเปิด VMware จากนั้นทำการเปิด Terminal เพื่อทำการ Download image โดยคำสั่งที่ใช้คือ
docker pull ghcr.io/nokia/srlinux จะได้ผลลัพธ์ดังนี้

```
tii@tii-virtual-machine:~$ sudo docker pull ghcr.io/nokia/srlinux
[sudo] password for tii:
Using default tag: latest
latest: Pulling from nokia/srlinux
5021ece2e12c: Pull complete
Digest: sha256:39671cbffaa2e42d584ecacac2070c9fbef0cf5f0295abe13135506375d0e51e
Status: Downloaded newer image for ghcr.io/nokia/srlinux:latest
ghcr.io/nokia/srlinux:latest
```

ขั้นตอนที่ 2: ทำการสร้าง Topo ในไฟล์ที่ได้ว่าจากการ pull โดยมีชื่อไฟล์ว่า project.clab.yml

```
1 name: project
2
3 topology:
4   nodes:
5     srl1:
6       kind: srl
7       image: ghcr.io/nokia/srlinux
8     srl2:
9       kind: srl
10      image: ghcr.io/nokia/srlinux
11
12   links:
13     - endpoints: ["srl1:e1-1", "srl2:e1-1"]
```

ขั้นตอนที่ 3: จากนั้นทำการเข้าไปใน directory โดยใช้คำสั่ง cd clab-srl เมื่อเข้าไปใน directory ของ clab-srl จากนั้นใช้คำสั่ง sudo containerlab deploy + เพื่อที่จะใช้ docker สร้างอุปกรณ์ในการเชื่อมต่อที่เราเขียนไว้ใน topology เมื่อคำสั่งเสร็จจะได้ผลดังนี้

```
gunntreepaeach@gunntreepaeach-virtual-machine:~/project$ sudo containerlab deploy +
[sudo] password for gunntreepaeach:
INFO[0000] Containerlab v0.38.0 started
INFO[0000] Parsing & checking topology file: project.clab.yml
INFO[0000] Creating lab directory: /home/gunntreepaeach/project/clab-project
INFO[0000] Creating docker network: Name="clab", IPv4Subnet="172.20.20.0/24", IPv6Subnet="2001:172:20:20::/64", MTU="1500"
INFO[0000] Creating container: "srl2"
INFO[0000] Creating container: "srl1"
INFO[0001] Creating virtual wire: srl1:e1-1 <--> srl2:e1-1
INFO[0001] Running postdeploy actions for Nokia SR Linux 'srl2' node
INFO[0001] Running postdeploy actions for Nokia SR Linux 'srl1' node
INFO[0010] Adding containerlab host entries to /etc/hosts file
```

#	Name	Container ID	Image	Kind	State	IPv4 Address	IPv6 Address
1	clab-project-srl1	22e478858e66	ghcr.io/nokia/srlinux	srl	running	172.20.20.2/24	2001:172:20:20::2/64
2	clab-project-srl2	43b7cc0f9c0f	ghcr.io/nokia/srlinux	srl	running	172.20.20.3/24	2001:172:20:20::3/64

ขั้นตอนที่ 4: จากนั้นใช้คำสั่ง `sudo docker ps` เพื่อดูทำงานใน docker ที่เราได้ทำการสร้างมา

```
gunntreepaeach@gunntreepaeach-virtual-machine:~/project$ sudo docker ps
[sudo] password for gunntreepaeach:
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
22e478858e66	ghcr.io/nokia/srlinux	"/tini -- fixuid -q ..."	5 minutes ago	Up 5 minutes		clab-project-srl1
43b7cc0f9c0f	ghcr.io/nokia/srlinux	"/tini -- fixuid -q ..."	5 minutes ago	Up 5 minutes		clab-project-srl2

ขั้นตอนที่ 5 : ทดสอบการ ping โดยใช้คำสั่ง `ping 172.20.20.2` และ `ping 172.20.20.3`

```
gunntreepaeach@gunntreepaeach-virtual-machine:~/project$ ping 172.20.20.2
PING 172.20.20.2 (172.20.20.2) 56(84) bytes of data.
64 bytes from 172.20.20.2: icmp_seq=1 ttl=64 time=7.89 ms
64 bytes from 172.20.20.2: icmp_seq=2 ttl=64 time=6.19 ms
64 bytes from 172.20.20.2: icmp_seq=3 ttl=64 time=8.18 ms
^C
--- 172.20.20.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 6.190/7.419/8.177/0.877 ms
gunntreepaeach@gunntreepaeach-virtual-machine:~/project$ ping 172.20.20.3
PING 172.20.20.3 (172.20.20.3) 56(84) bytes of data.
64 bytes from 172.20.20.3: icmp_seq=1 ttl=64 time=8.37 ms
64 bytes from 172.20.20.3: icmp_seq=2 ttl=64 time=6.45 ms
64 bytes from 172.20.20.3: icmp_seq=3 ttl=64 time=8.34 ms
^C
--- 172.20.20.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 6.448/7.719/8.374/0.898 ms
```

ขั้นตอนที่ 6: ในกรณีที่ไม่ทราบ ansible_ssh_pass ให้เราทำการเข้าไปที่ในเครื่องที่ได้สร้างขึ้นโดยขั้นตอนตอนนี้
เครื่องที่ 1:

```
gunntreepaech@gunntreepaech-virtual-machine:~$ sudo docker exec -it clab-project-srl1 bash
[root@srl1 /]# passwd
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

เครื่องที่ 2:

```
gunntreepaech@gunntreepaech-virtual-machine:~$ sudo docker exec -it clab-project-srl2 bash
[root@srl2 /]# passwd
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

ขั้นตอนที่ 7: จากนั้นลองเข้าไปที่ root โดยใช้คำสั่ง ssh root@<ip address>

```
gunntreepaech@gunntreepaech-virtual-machine:~$ ssh root@172.20.20.2
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:5oHjYF0QAicg2KDwAztBnIasw3KRdWmkBjHJMAr414s.
Please contact your system administrator.
Add correct host key in /home/gunntreepaech/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /home/gunntreepaech/.ssh/known_hosts:14
  remove with:
    ssh-keygen -f "/home/gunntreepaech/.ssh/known_hosts" -R "172.20.20.2"
Host key for 172.20.20.2 has changed and you have requested strict checking.
Host key verification failed.
```

ถ้าขึ้นข้อความตามภาพด้านบน ให้ทำการล้าง ssh ตัวเก่าทิ้งโดยใช้คำสั่ง

ssh-keygen -f "/home/\$USER/.ssh/known_hosts" -R "<ip address>" ตามภาพด้านล่าง

```

gunntreepaech@gunntreepaech-virtual-machine:~$ ssh-keygen -f "/home/gunntreepaech/.ssh/known_hosts" -R "172.20.20.2"
# Host 172.20.20.2 found: line 12
# Host 172.20.20.2 found: line 13
# Host 172.20.20.2 found: line 14
/home/gunntreepaech/.ssh/known_hosts updated.
Original contents retained as /home/gunntreepaech/.ssh/known_hosts.old
gunntreepaech@gunntreepaech-virtual-machine:~$ ssh-keygen -f "/home/gunntreepaech/.ssh/known_hosts" -R "172.20.20.3"
# Host 172.20.20.3 found: line 12
# Host 172.20.20.3 found: line 13
# Host 172.20.20.3 found: line 14
/home/gunntreepaech/.ssh/known_hosts updated.
Original contents retained as /home/gunntreepaech/.ssh/known_hosts.old
gunntreepaech@gunntreepaech-virtual-machine:~$

```

จากนั้นใช้คำสั่ง ssh root@<ip address> อีกครั้งทั้ง 2 ip address ตามภาพด้านล่าง

เครื่องที่ 1:

```

gunntreepaech@gunntreepaech-virtual-machine:~$ ssh root@172.20.20.2
The authenticity of host '172.20.20.2 (172.20.20.2)' can't be established.
ED25519 key fingerprint is SHA256:5oHjYF0QAicg2KDwAztBnIasw3KRdWmkBjHJMAr414s.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.20.20.2' (ED25519) to the list of known hosts.
.....
:           Welcome to Nokia SR Linux!           :
:           Open Network OS for the NetOps era.     :
:           :                                       :
:   This is a freely distributed official container image. :
:           Use it - Share it                     :
:           :                                       :
: Get started: https://learn.srlinux.dev           :
: Container:   https://go.srlinux.dev/container-image :
: Docs:        https://doc.srlinux.dev/22-11       :
: Rel. notes:  https://doc.srlinux.dev/rn22-11-2   :
: YANG:        https://yang.srlinux.dev/v22.11.2   :
: Discord:     https://go.srlinux.dev/discord       :
: Contact:     https://go.srlinux.dev/contact-sales :
:           :                                       :
:           :                                       :
.....

root@172.20.20.2's password:
[root@srl1 ~]#

```

เครื่องที่ 2:

```

gunntreepaeach@gunntreepaeach-virtual-machine:~$ ssh root@172.20.20.3
The authenticity of host '172.20.20.3 (172.20.20.3)' can't be established.
ED25519 key fingerprint is SHA256:p7PI2qzqGF02T/r+HofmHIYk45Ha7YYCcFREF1WQp/8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.20.20.3' (ED25519) to the list of known hosts.

.....
:           Welcome to Nokia SR Linux!           :
:           Open Network OS for the NetOps era.     :
:           :                                       :
:           This is a freely distributed official   :
:           container image.                       :
:           Use it - Share it                      :
:           :                                       :
: Get started: https://learn.srlinux.dev           :
: Container:  https://go.srlinux.dev/container-image :
: Docs:       https://doc.srlinux.dev/22-11        :
: Rel. notes: https://doc.srlinux.dev/rn22-11-2    :
: YANG:       https://yang.srlinux.dev/v22.11.2    :
: Discord:    https://go.srlinux.dev/discord       :
: Contact:    https://go.srlinux.dev/contact-sales :
:           :                                       :
:           :                                       :
.....

root@172.20.20.3's password:
[root@srl2 ~]#

```

ขั้นตอนที่ 8: แก้ไข ไฟล์ ansible.cfg โดยใช้คำสั่ง `sudo nano /etc/ansible/ansible.cfg` แล้วพิมพ์ตามภาพด้านล่าง

```

❯ ansible.cfg
1  [defaults]
2  host_key_checking = false
3  interpreter_python = auto_silent
4  deprecation_warnings=False
5  command_warnings=False
6  retry_files_enabled = False

```

ขั้นตอนที่ 9: แก้ไข ไฟล์ hosts โดยใช้คำสั่ง `sudo nano /etc/ansible/hosts` แล้วพิมพ์ตามภาพด้านล่าง

```

≡ hosts
1  [lab]
2  clab-srl02-srl1 ansible_host=172.20.20.2
3  clab-srl02-srl2 ansible_host=172.20.20.3
4
5  [lab:vars]
6  ansible_connection= ssh
7  ansible_user='root'
8  ansible_ssh_pass='root123'|

```

ขั้นตอนที่ 10: ทดลอง ping จาก ansible

```

gunntreepaech@gunntreepaech-virtual-machine:~$ ansible all -m ping -v
Using /etc/ansible/ansible.cfg as config file
clab-srl02-srl1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
clab-srl02-srl2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}

```

ขั้นตอนที่ 11: ทำการเขียน playbook

ตัวอย่างที่ 1:

```

1|- name: show ifconfig with shell comman
2  hosts: all
3  become: yes
4
5  tasks:
6    - name: show ifconfig
7      shell: ifconfig -a
8      register: output
9    - debug:
10      var: output.stdout_lines

```


playbook มีการทำงานคือ จะใช้คำสั่ง ifconfig -a ที่ shell แล้วจะนำผลลัพธ์ไปเก็บไว้ที่ output จากนั้นจะมาแสดงในคำสั่ง debug ผลลัพธ์จะเป็นดังรูปด้านล่าง

```
gunntreepaech@gunntreepaech-virtual-machine:~/project/playbooks$ ansible-playbook showifconfig.yml
PLAY [show ifconfig with shell command] *****

TASK [Gathering Facts] *****
ok: [clab-srl02-srl1]
ok: [clab-srl02-srl2]

TASK [show ifconfig] *****
changed: [clab-srl02-srl1]
changed: [clab-srl02-srl2]

TASK [debug] *****
ok: [clab-srl02-srl1] => {
  "output.stdout_lines": [
    "gateway: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500",
    "    inet 172.250.250.1 netmask 255.255.255.252 broadcast 172.250.250.3",
    "    inet6 fe80::50f5:68ff:fec2:4368 prefixlen 64 scopeid 0x20<link>",
    "    inet6 fe80::1 prefixlen 128 scopeid 0x20<link>",
    "    ether 1a:c8:00:ff:00:00 txqueuelen 1000 (Ethernet)",
    "    RX packets 14 bytes 1060 (1.0 KiB)",
    "    RX errors 0 dropped 0 overruns 0 frame 0",
    "    TX packets 12 bytes 960 (960.0 B)",
    "    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0",
    "",
    "lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536",
    "    inet 127.0.0.1 netmask 255.0.0.0",
    "    inet6 ::1 prefixlen 128 scopeid 0x10<host>",
    "    loop txqueuelen 1000 (Local Loopback)",
    "    RX packets 0 bytes 0 (0.0 B)",
    "    RX errors 0 dropped 0 overruns 0 frame 0",
    "    TX packets 0 bytes 0 (0.0 B)",
    "    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0",
    "",
    "mgmt0.0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500",
    "    inet 172.20.20.2 netmask 255.255.255.0 broadcast 172.20.20.255",
    "    inet6 fe80::42:acff:fe14:1402 prefixlen 64 scopeid 0x20<link>",
    "    inet6 2001:172:20:20::2 prefixlen 64 scopeid 0x0<global>",
    "    ether 02:42:ac:14:14:02 txqueuelen 1000 (Ethernet)",
    "    RX packets 511 bytes 461970 (451.1 KiB)",
    "    RX errors 0 dropped 0 overruns 0 frame 0",
    "    TX packets 357 bytes 50786 (49.5 KiB)",
    "    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0"
  ]
}
```

ตัวอย่างที่ 2:

```
1 - name: chang loopback ip address
2   hosts: all
3   become: yes
4
5   tasks:
6     - name: show ifconfig before chang ip address
7       shell: ifconfig -a
8       register: output
9     - debug:
10       var: output.stdout_lines
11
12     - name: chang ip address
13       shell: ifconfig lo 172.148.156.31 netmask 255.255.255.0
14
15     - name: show ifconfig after chang ip address
16       shell: ifconfig -a
17       register: output
18     - debug:
19       var: output.stdout_lines
20
```

การทำงานเป็นการเปลี่ยน ip address ของ loopback จาก ip 127.0.0.1 netmask 255.0.0.0 เป็น ip 172.148.156.31 netmask 255.255.255.0 ผลลัพธ์เป็นดังนี้

ก่อนเปลี่ยน:

```
gunntreepaech@gunntreepaech-virtual-machine:~/project/playbooks$ ansible-playbook changloopbackIP.yaml

PLAY [chang loopback ip address] *****

TASK [Gathering Facts] *****
ok: [clab-srl02-srl1]
ok: [clab-srl02-srl2]

TASK [show ifconfig before chang ip address] *****
changed: [clab-srl02-srl1]
changed: [clab-srl02-srl2]

TASK [debug] *****
ok: [clab-srl02-srl1] => {
  "output.stdout_lines": [
    "gateway: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500",
    "    inet 127.250.250.1 netmask 255.255.255.252 broadcast 127.250.250.3",
    "    inet6 fe80::50f5:68ff:fec2:4368 prefixlen 64 scopeid 0x20<link>",
    "    inet6 fe80::1 prefixlen 120 scopeid 0x20<link>",
    "    ether 1a:c8:00:ff:00:00 txqueuelen 1000 (Ethernet)",
    "    RX packets 14 bytes 1060 (1.0 KiB)",
    "    RX errors 0 dropped 0 overruns 0 frame 0",
    "    TX packets 13 bytes 1030 (1.0 KiB)",
    "    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0",
    "",
    "lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536",
    "    inet 127.0.0.1 netmask 255.0.0.0",
    "    inet6 ::1 prefixlen 128 scopeid 0x10<host>",
    "    loop txqueuelen 1000 (Local Loopback)",
    "    RX packets 0 bytes 0 (0.0 B)",
    "    RX errors 0 dropped 0 overruns 0 frame 0",
    "    TX packets 0 bytes 0 (0.0 B)",
    "    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0",
    "",
    "ngnt0.0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500",
    "    inet 172.20.20.2 netmask 255.255.255.0 broadcast 172.20.20.255",
    "    inet6 fe80::42:acff:fe14:1402 prefixlen 64 scopeid 0x20<link>",
    "    inet6 2001::172:20:20:2 prefixlen 64 scopeid 0x0<global>",
    "    ether 02:42:ac:14:14:02 txqueuelen 1000 (Ethernet)",
    "    RX packets 4145 bytes 4350182 (4.1 MiB)",
    "    RX errors 0 dropped 0 overruns 0 frame 0",
    "    TX packets 2669 bytes 317402 (309.9 KiB)",
    "    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0"
  ]
}
```

หลังเปลี่ยน:

```

TASK [chang ip address] *****
changed: [clab-srl02-srl1]
changed: [clab-srl02-srl2]

TASK [show ifconfig after chang ip address] *****
changed: [clab-srl02-srl1]
changed: [clab-srl02-srl2]

TASK [debug] *****
ok: [clab-srl02-srl1] => {
  "output.stdout_lines": [
    "gateway: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500",
    "    inet 127.250.250.1 netmask 255.255.255.252 broadcast 127.250.250.3",
    "    inet6 fe80::50f5:68ff:fec2:4368 prefixlen 64 scopeid 0x20<link>",
    "    ether 1a:c8:00:ff:00:00 txqueuelen 1000 (Ethernet)",
    "    RX packets 14 bytes 1060 (1.0 KiB)",
    "    RX errors 0 dropped 0 overruns 0 frame 0",
    "    TX packets 13 bytes 1030 (1.0 KiB)",
    "    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0",
    "",
    "lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536",
    "    inet 172.148.156.31 netmask 255.255.255.0",
    "    inet6 ::1 prefixlen 128 scopeid 0x10<host>",
    "    loop txqueuelen 1000 (Local Loopback)",
    "    RX packets 0 bytes 0 (0.0 B)",
    "    RX errors 0 dropped 0 overruns 0 frame 0",
    "    TX packets 0 bytes 0 (0.0 B)",
    "    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0",
    "",
    "mgmt0.0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500",
    "    inet 172.20.20.2 netmask 255.255.255.0 broadcast 172.20.20.255",
    "    inet6 fe80::42:acff:fe14:1402 prefixlen 64 scopeid 0x20<link>",
    "    inet6 2001:172:20:20::2 prefixlen 64 scopeid 0x0<global>",
    "    ether 02:42:ac:14:14:02 txqueuelen 1000 (Ethernet)",
    "    RX packets 4351 bytes 4562994 (4.3 MiB)",
    "    RX errors 0 dropped 0 overruns 0 frame 0",
    "    TX packets 2765 bytes 327906 (320.2 KiB)",
    "    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0"
  ]
}

```

คำสั่งพื้นฐานของ Nokia SR Linux

คำสั่งที่ 1:

Show version

คำสั่ง Show version ของ Nokia SR Linux คือ show version โดยสามารถใช้คำสั่งนี้ในโหมด configure หรือโหมด operational ได้ โดยจะแสดงเวอร์ชันของระบบปฏิบัติการ SR Linux ที่กำลังทำงานอยู่บนอุปกรณ์นั้น ๆ ในรูปแบบของตัวเลขเวอร์ชันและชื่อเวอร์ชันที่อาจมีการกำหนดโดยผู้ใช้งาน

```

A:srl1# show version
-----
Hostname           : srl1
Chassis Type       : 7220 IXR-D2
Part Number        : Sim Part No.
Serial Number      : Sim Serial No.
System HW MAC Address: 1A:C8:00:FF:00:00
Software Version   : v22.11.2
Build Number       : 116-gf3be2e95f2
Architecture       : x86_64
Last Booted        : 2023-03-31T17:36:55.944Z
Total Memory       : 7914800 kB
Free Memory        : 2523980 kB
-----

```

คำสั่งที่ 2:

show network-instance route-table all

คำสั่ง `show network-instance route-table all` ใน Nokia SR Linux เป็นคำสั่งที่ใช้สำหรับแสดงข้อมูลเกี่ยวกับตารางเส้นทางของเครือข่ายในทุกๆ Network Instance ที่มีอยู่ในระบบ

```
A:srl1# show network-instance route-table all
```

IPv4 unicast route table of network instance mgmt									
Prefix	ID	Route Type	Route Owner	Active	Metric	Pref	Next-hop (Type)	Next-hop Interface	
0.0.0.0/0	1	dhcp	dhcp_client_mgr	True	0	5	172.20.20.1 (direct)	mgmt0.0	
172.20.20.0/24	0	linux	linux_mgr	False	0	5	172.20.20.0 (direct)	mgmt0.0	
172.20.20.0/24	1	local	net_inst_mgr	True	0	0	172.20.20.2 (direct)	mgmt0.0	
172.20.20.2/32	1	host	net_inst_mgr	True	0	0	None (extract)	None	
172.20.20.255/32	1	host	net_inst_mgr	True	0	0	None (broadcast)	None	

IPv4 routes total : 5
IPv4 prefixes with active routes : 4
IPv4 prefixes with active ECMP routes: 0

IPv6 unicast route table of network instance mgmt									
Prefix	ID	Route Type	Route Owner	Active	Metric	Pref	Next-hop (Type)	Next-hop Interface	
2001:172:20:20::/64	0	linux	linux_mgr	False	0	5	2001:172:20:20:: (direct)	mgmt0.0	
2001:172:20:20::/64	1	local	net_inst_mgr	True	0	0	2001:172:20:20::2 (direct)	mgmt0.0	
2001:172:20:20::2/128	1	host	net_inst_mgr	True	0	0	None (extract)	None	

IPv6 routes total : 3
IPv6 prefixes with active routes : 2
IPv6 prefixes with active ECMP routes: 0

คำสั่งที่ 3:

`show platform`

คำสั่ง `show platform` ใน Nokia SR Linux ใช้สำหรับแสดงข้อมูลเกี่ยวกับพลังงานและอุปกรณ์ฮาร์ดแวร์บนเครื่องเซิร์ฟเวอร์

```
A:srl1# show platform
```

Module Type	ID	Admin State	Operational State	Model	Last Change
control	A	N/A	up	imn48-25g-sfp28+8-100g-qsfp28	2023-03-31T17:36:55.944Z
linecard	1	N/A	empty		

คำสั่งที่ 4:

`Show interface`

คำสั่ง `Show interface` ใน Nokia SR Linux คือ "show interface". คำสั่งนี้จะแสดงรายละเอียดของอินเทอร์เฟซที่กำลังทำงานอยู่บนอุปกรณ์ รวมถึงสถานะของอินเทอร์เฟซ และข้อมูลต่างๆ

```
A:srl1# show interface
```

ethernet-1/1 is up, speed 25G, type None	
mgmt0 is up, speed 1G, type None	
mgmt0.0 is up	
Network-instance: mgmt	
Encapsulation : null	
Type : None	
IPv4 addr	: 172.20.20.2/24 (dhcp, preferred)
IPv6 addr	: 2001:172:20:20::2/64 (dhcp, preferred)
IPv6 addr	: fe80::42:acff:fe14:1402/64 (link-layer, preferred)

Summary

- 0 loopback interfaces configured
- 1 ethernet interfaces are up
- 1 management interfaces are up
- 1 subinterfaces are up