



02-Network-Interface-Command

The `ifconfig` command

ก่อนที่จะเริ่มใช้งาน command นี้เราควรโหลด package net-tool มาต่อโดยใช้คำสั่ง

```
$ sudo apt install net-tools
```

สามารถใช้คำสั่ง `ifconfig` เพื่อกำหนดค่าของอินเทอร์เฟชเครือเบลก์อยู่ในระบบ kernel มักถูกใช้ในเวลาเริ่มต้นของระบบเพื่อตั้งค่าอินเทอร์เฟชตามที่จำเป็น หลังจากนั้น บันบัดจะใช้เฉพาะเบื้องมีการเก็บัญหาหรือเบื้องต้องการปรับแต่งระบบ

โครงสร้างคำสั่ง

```
$ ifconfig [-v] [-a] [-s] [interface]  
$ ifconfig [-v] interface [aftype] options | address ...
```

Command	Description
<code>ifconfig</code>	แสดงข้อมูลรายละเอียด interface ทั้งหมดที่ถูก Activate
<code>ifconfig -a</code>	แสดงข้อมูลรายละเอียด interface ทั้งหมดทั้งเมืองไม่ Activate
<code>ifconfig -v</code>	แสดงข้อมูล interface ที่เกิด error condition ทั้งหมด
<code>ifconfig -s</code>	แสดงข้อมูล interface ทั้งหมดเป็น list แบบย่อ
<code>ifconfig [interface]</code>	แสดงข้อมูล interface ที่เฉพาะเจาะจงตามที่ระบุไว้
<code>ifconfig [interface] up</code>	เปิดการใช้งาน interface ที่ระบุ
<code>ifconfig [interface] down</code>	ปิดการใช้งาน interface ที่ระบุ
<code>ifconfig [interface] [IP address] netmask [Subnet mask] broadcast [Broadcast IP]</code>	กำหนดค่า configure ของ interface โดยรวมทั้งหมด
<code>ifconfig [interface] promisc</code>	เปิดใช้งาน promiscous mode ให้กับ interface ที่ระบุไว้
<code>ifconfig [interface] promisc</code>	ปิดใช้งาน promiscous mode ให้กับ interface ที่ระบุไว้
<code>ifconfig [interface] [IP address]</code>	กำหนดค่า IP เอกพากให้กับอินเทอร์เฟชเครือข่ายที่ระบุไว้
<code>ifconfig [interface] mtu [rate]</code>	เพื่อตั้งค่าหน่วยการส่งข้อมูลสูงสุดให้เป็นอินเทอร์เฟชเครือข่าย
<code>ifconfig [interface]:[subinterface] [IP address]</code>	หากต้องการเพิ่มค่า IP เพิ่มเติมให้กับอินเทอร์เฟชเครือข่าย คุณสามารถกำหนด sub-interface ได้
<code>ifconfig [interface]:[subinterface] down</code>	ลบหรือปิดการใช้งานของ sub-interface นั้น

Promiscous mode คือ โหมดที่จะยอมรับทุกแพ็คเก็ตเก็บกั้งหมดที่ให้ผ่านการเดเครือข่ายมาเก็บไว้ โดยถ้าเป็นโหมดปกติ เมื่อ แพ็คเก็ตถูกรับโดยการเดเครือข่าย มันจะตรวจสอบว่ามันเป็นของตัวเองหรือไม่ หากไม่ใช่ มันจะปิดกั้นแพ็คเก็ตไป

ตัวอย่างการใช้คำสั่ง

- แสดงข้อมูลรายละเอียด interface กั้งหมด

```
User@ubuntu:~$ ifconfig -a
```

OUTPUT

```
putter@anthority:~$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.191 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::a00:27ff:feb1:d582 prefixlen 64 scopeid 0x20<link>
        inet6 2001:fb1:a9:adac:a00:27ff:feb1:d582 prefixlen 64 scopeid 0x0<global>
            ether 08:00:27:b1:d5:82 txqueuelen 1000 (Ethernet)
            RX packets 871 bytes 208649 (208.6 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 70 bytes 9282 (9.2 KB)
            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 94 bytes 7940 (7.9 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 94 bytes 7940 (7.9 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

[ภาพที่ 1] การแสดงผลคำสั่ง ifconfig -a

- สร้าง sub-interface แล้วให้ IP Address, Subnet mask, Broadcast Address

```
User@ubuntu:~$ sudo ifconfig enp0s3:0 192.168.1.120 netmask 255.255.255.128 broadcast 192.168.1.127
```

OUTPUT

```
putter@anthority:~$ sudo ifconfig enp0s3:0 192.168.1.120 netmask 255.255.255.128 broadcast 192.168.1.127
putter@anthority:~$ ifconfig enp0s3:0
enp0s3:0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.120 netmask 255.255.255.128 broadcast 192.168.1.127
        ether 08:00:27:b1:d5:82 txqueuelen 1000 (Ethernet)

putter@anthority:~$ sudo ifconfig enp0s3:0 down
putter@anthority:~$ _
```

[ภาพที่ 2] การแบบ sub-interfaces

The `ip` command

คำสั่ง `ip` เป็นคำสั่งที่มีอยู่ในไดเรกทอรี่ `/sbin` หรือ `/usr/sbin` ซึ่งเป็นไดเรกทอรี่ที่เก็บคำสั่งที่ต้องการสิทธิ์ root (superuser) เพื่อใช้งานได้ ดังนั้นเมื่อใช้งาน `ip` ควรเรียกใช้โดยใช้สิทธิ์ root โดยการใช้คำสั่ง `sudo ip` หรือเข้าสู่ระบบด้วยบัญชี root ก่อนเพื่อใช้งานคำสั่งได้อย่างถูกต้อง ไม่ต้องติดตั้ง Package เพิ่มเติมในการใช้งาน

โครงสร้างคำสั่ง

```
$ ip [OPTIONS] OBJECT {COMMAND | help}
```

มีองค์ประกอบดังนี้

- OPTIONS** : ตัวเลือกเพิ่มเติมที่ปรับเปลี่ยนพฤติกรรมของ nmcli
- OBJECT** : ระบุเป้าหมายของการดำเนินการ (เช่น การเชื่อมต่อ อุปกรณ์ ฯลฯ)
- COMMAND** : กำหนดการกระทำที่จะดำเนินการกับวัตถุที่ระบุ

Object	Description
<code>ip link l</code>	แสดงและแก้ไข network interface IPv6
<code>ip address addr</code>	แสดงและแก้ไข IP Address ในอินเตอร์เฟซ
<code>ip addrlabel addrl</code>	กำหนดค่าป้ายกำกับสำหรับการเลือกที่อยู่ของโปรดิคอล

<code>ip neighbour n</code>	แสดงและจัดการวัตถุข้างเคียง (ตาราง ARP)
<code>ip route r</code>	แสดงและแก้ไขตารางเส้นทาง (routing table)
<code>ip rule ru</code>	กฎเกณฑ์ในชุดข้อมูลที่กำหนดนโยบายการกำหนดเส้นทาง
<code>ip maddress maddr</code>	Multicast address
<code>ip mroute mr</code>	Multicast routing cache entry
<code>ip tunnel t</code>	แสดงและแก้ไข Tunnel ที่ผ่านมา IP
<code>ip xfrm x</code>	Framework ของ IPsec protocol

| ในคำสั่งต่อๆ สามารถพิมพ์รูปแบบย่อได้ เช่น `ip address` → `ip a / ip addr` เป็นต้น

ใช้คำสั่ง `ip` เพื่อแสดงและ **กำหนดค่าพาრามิเตอร์เครือข่าย** สำหรับอินเทอร์เฟซของเครื่องทั้งนี้:

1. หาว่าอินเทอร์เฟซใดบ้างที่มีกำหนดค่าบนระบบ
2. สอบถามสถานะของอินเทอร์เฟซ IP
3. กำหนดค่า loop-back ภายใน, Ethernet และอินเทอร์เฟซ IP อื่น ๆ
4. เปลี่ยนสถานะอินเทอร์เฟซเป็น up หรือ down

```
# ip link set {interface-name} {up|down}
```

5. กำหนดค่าและแก้ไขการส่ง static route และ default route

- Add a new **static route**

```
# ip route add {NETWORK/MASK} via {GATEWAYIP}
# ip route add {NETWORK/MASK} dev {DEVICE}
```

- Add a new **default route**

```
# ip route add default {NETWORK/MASK} via {GATEWAYIP}
# ip route add default {NETWORK/MASK} dev {DEVICE}
```

- Delete a route

```
ip route del default
```

6. ติดตั้ง tunnel ผ่าน IP

7. แสดงรายการcache ARP หรือ NDISC

8. กำหนด, ลบ, ตั้งค่า IP Address, routes, subnet และข้อมูล IP อื่น ๆ ไปยัง IP interfaces

- **กำหนดค่า IP address ให้กับ interface**

```
# ip a add {ip_addr/mask} dev {interface-name}
```

- **set broadcast address**

```
# ip addr add broadcast {brd_address} dev {interface}
```

- **ลบ IP Address ออกจาก Interface**

```
# ip a del {ip_addr/mask} dev {interface-name}
```

9. แสดงรายการที่อยู่ IP และข้อมูลคุณสมบัติ

```
# ip add show
```

10. จัดการและแสดงสถานะของเครือข่ายทั้งหมด

```
$ sudo ip -br -c addr show
```

11. เก็บข้อมูลที่อยู่ IP เป็นหน่วย
12. แสดงวัตถุเพื่อบ้าน กล่าวคือ แคช ARP, กำให้แคช ARP เป็นโโนน, เพิ่มรายการเข้าแคช ARP และอ่าน ๆ

- **Display neighbour/arp cache**

```
$ ip neigh show
```

- **Add a new ARP entry**

```
# ip neigh add {IP-HERE} lladdr {MAC/LLADDRESS} dev {DEVICE} nud {STATE}
```

- **Delete a ARP entry**

```
# ip neigh del {IPAddress} dev {DEVICE}
```

- **Flush ARP entry**

```
# ip -s -s n f {IPAddress}
```

13. ตั้งค่าหรือลบรายการส่งเส้นทาง

14. หา route address (เช่น 8.8.8.8)

```
$ ip route list
```

15. กำหนดความยาวของคิวการส่ง

```
# ip link set txqueuelen {NUMBER} dev {DEVICE}
```

- สำหรับเครือข่ายกิกะบิต คุณสามารถตั้งค่า *maximum transmission units (MTU)* เพื่อประสิทธิภาพเครือข่ายที่ดีขึ้น

```
# ip link set mtu {NUMBER} dev {DEVICE}
```

| อาจมีการเพิ่ม option ก่อนตัว object ระบุไว้ก่อนด้านหน้า เพื่อ

ตัวอย่างการใช้คำสั่ง

- แสดงข้อมูลรายละเอียดของ interface โดยให้ highlight state และ IP address และสถานะ IPv4 เท่านั้น

```
User@ubuntu:~$ ip -4 -c addr show
```

OUTPUT

```
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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b1:d5:02 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.191/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
            valid_lft 84134sec preferred_lft 84134sec
        inet6 2001:fb1:ab:fb0b:a00:27ff:feb1:d582/64 scope global dynamic mngtmpaddr noprefixroute
            valid_lft 259192sec preferred_lft 172792sec
        inet6 fe80::a00:27ff:feb1:d582/64 scope link
            valid_lft forever preferred_lft forever
putter@anthority:~$
```

[ภาพที่ 3] ใช้ -c กับ ip addr เพื่อใช้สีในส่วนของ IP และ state

- ในการกำหนดที่อยู่ IP ให้กับอินเทอร์เฟซบ้านๆ หรือการแบ่ง subnet

```
User@ubuntu:~$ sudo ip addr add 192.168.1.120/25 dev enp0s3
```

OUTPUT

```

putter@anthority:~$ sudo ip addr add 192.168.1.120/25 dev enp0s3:0
putter@anthority:~$ ip -c addr show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
    link/ether 08:00:27:b1:d5:82 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.191/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
            valid_lft 83144sec preferred_lft 83144sec
        inet 192.168.1.120/25 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
            valid_lft forever preferred_lft forever
        inet6 2001:fbi:ab:fb3b:a00:27ff:feb1:d582/64 metric 100 scope global dynamic mngtmpaddr noprefixroute
            valid_lft 259181sec preferred_lft 172781sec
        inet6 fe80::a00:27ff:feb1:d582/64 metric 100 scope link
            valid_lft forever preferred_lft forever
putter@anthority:~$
```

[ภาพที่ 4] กำหนดค่า subnet ในบันทึก interface

The `nmcli` command

`nmcli` เป็นเครื่องมือ CLI ที่ใช้ในการควบคุม NetworkManager คำสั่ง `nmcli` ยังสามารถใช้เพื่อแสดงสถานะอุปกรณ์เครือข่าย สร้าง แก้ไข เปิด/ปิดใช้งาน และลบการเชื่อมต่อเครือข่าย

โครงสร้างคำสั่ง

```
$ nmcli [OPTIONS] OBJECT {COMMAND | help}
```

เมื่องค่าประกอบดังนี้

- **OPTIONS** : ตัวเลือกเพิ่มเติมที่ปรับเปลี่ยนพฤติกรรมของ `nmcli`
- **OBJECT** : ระบุเป้าหมายของการดำเนินการ (เช่น การเชื่อมต่อ อุปกรณ์ ฯลฯ)
- **COMMAND** : กำหนดการกระทำที่จะดำเนินการกับวัตถุที่ระบุ

Command	Description
<code>nmcli general { status hostname permissions logging } [ARGUMENTS ...]</code>	ใช้คำสั่งนี้เพื่อแสดงสถานะ NetworkManager และการอนุญาต
<code>nmcli networking { on off connectivity } [ARGUMENTS ...]</code>	สobelสถานะเครือข่าย NetworkManager เปิดใช้งานและปิดใช้งานเครือข่าย
<code>nmcli connection { show up down modify add edit clone delete monitor reload load import export } [ARGUMENTS ...]</code>	สามารถใช้การเชื่อมต่อเพิ่มเติมเพื่อให้สามารถสับเปลี่ยนเครือข่ายและการกำหนดค่าต่างๆ ได้อย่างรวดเร็ว
<code>nmcli connection show</code>	แสดงรายการการเชื่อมต่อเครือข่าย กั้งหนดที่มีอยู่ในระบบของคุณ โดยให้รายละเอียดต่างๆ เช่น ชื่อการเชื่อมต่อ UUID อุปกรณ์ ประเภท และสถานะ
<code>nmcli connection add type <connection_type> ifname <interface_name> con-name <connection_name></code>	เพิ่มการเชื่อมต่ออีกครั้งหนึ่ง
<code>nmcli connection modify <connection_name> <setting_name> <setting_value></code>	ปรับการตั้งค่าการเชื่อมต่อ
<code>nmcli device { status show set connect reapply modify disconnect delete monitor wifi lldp } [ARGUMENTS ...]</code>	แสดงและจัดการอินเทอร์เฟซเครือข่าย
<code>nmcli dev status</code>	แสดงสถานะปัจจุบันของอุปกรณ์เครือข่ายกั้งหนดระบบของคุณ
<code>nmcli device show <device_name></code>	แสดงรายละเอียดอุปกรณ์ตัวนั้น

interface ที่ระบุ	
<code>nmcli radio { all wifi wwan } [ARGUMENTS ...]</code>	แสดงสถานะสวิตซ์ วิทยุ หรือเปิดและปิด สวิตซ์
<code>nmcli monitor</code>	สังเกตวิจกรรม NetworkManager ติดตามการ เปลี่ยนแปลงสถานะ การเชื่อมต่อ อุปกรณ์ หรือไฟล์การเชื่อม ต่อ
<code>nmcli agent { secret polkit all }</code>	รัน nmcli ในฐานะ ตัวแทนลับของ NetworkManager หรือตัวแทน polkit

ตัวอย่างการใช้คำสั่ง

- เพิ่มการเชื่อมต่ออีเมลแบบไม่ได้ต้องบีบเข้าไปในตัวอย่างกับอินเทอร์เฟซ enp0s3 พร้อมการกำหนดค่า IP อัตโนมัติ (DHCP) และปิดใช้งานการตั้งค่าสถานะการเชื่อมต่ออัตโนมัติของการเชื่อมต่อ

```
User@ubuntu:~$ nmcli connection add type ethernet autoconnect no ifname enp0s3
```

```
putter@anthority:~$ nmcli connection add type ethernet autoconnect no ifname enp0s3
Connection 'ethernet-enp0s3' (4a124799-cb94-4433-aeb2-61ab6e9e5bed) successfully added.
putter@anthority:~$ nmcli connection show
NAME                UUID                                  TYPE      DEVICE
Ethernet connection 1  9ee6e7ef-6dca-4da8-b10a-27a5a523196d  ethernet  --
ethernet-enp0s3       4a124799-cb94-4433-aeb2-61ab6e9e5bed  ethernet  --
putter@anthority:~$ nmcli device status
DEVICE  TYPE      STATE      CONNECTION
enp0s3  ethernet  unmanaged  --
lo     loopback  unmanaged  --
putter@anthority:~$
```

[ภาพที่ 5] กำหนดค่าให้ใน network interface

- ส่งออกโปรไฟล์ NetworkManager VPN corp-vpnc เป็นการกำหนดค่าบันทึกฐานข้อมูล Cisco (vpnc)

```
User@ubuntu:~$ nmcli con export corp-vpnc /home/joe/corpvpn.conf
```

Reference

- Bobby Iliev. (2021). 101 Linux commands Open-source eBook. <https://github.com/bobbyiliev/101-linux-commands-ebook/tree/main>
- saixiii. (2017). ifconfig – Linux Command คำสั่งแสดงข้อมูลและเปลี่ยนค่า interface server. <https://saixiii.com/ifconfig-linux-command/>
- Vivek Gite. (2023). Linux ip Command Examples. <https://www.cyberciti.biz/faq/linux-ip—command-examples-usage-syntax/>
- ChatGPT. (2024). Linux Network Interface Configuration. <https://chat.openai.com/share/31898e1f-ee5c-4f0e-844a-a8d61edc9fc8>
- Gabriel Ramugila. (2023). nmcli Linux Command | NetworkManager Usage Guide. <https://ioflood.com/blog/nmcli-linux-command/>
- GNOME Devleoper. <https://developer-old.gnome.org/NetworkManager/stable/nmcli.html>

เขียนและเรียบเรียงเนื้อหาโดย นายกคณิต พันธุ์กฤศติวงศ์ 65070165 สื่อกิจกรรมทั่วไป เนื้อหาเรื่องการตั้งค่าเครือข่ายใน Linux สำหรับผู้เรียนของมหาวิทยาลัยเชียงใหม่ อนุญาตให้นำไปใช้ได้