

목차

- 00. 개요
- 01. 네트워크의 기초
- 02. 네트워크 설정
- 03. 호스트 이름 설정
- 04. 네트워크 상태 확인

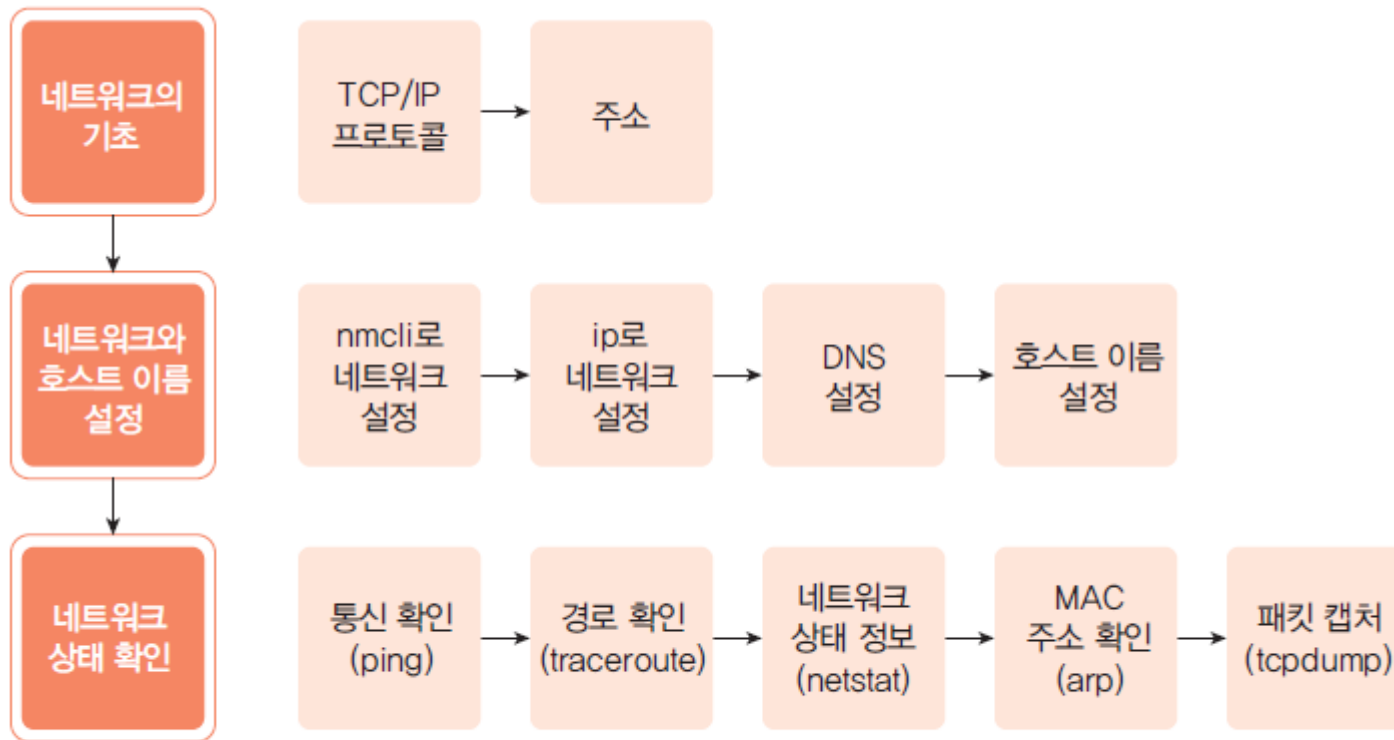


그림 11-1 11장의 내용 구성

01 네트워크의 기초

■ TCP/IP 프로토콜

- 프로토콜이란 컴퓨터와 컴퓨터 사이에서 데이터를 어떻게 주고받을 것인지를 정의한 통신 규약
- 인터넷이라고 부르는 네트워크는 TCP/IP라는 프로토콜에 따라 데이터를 주고받음
- TCP/IP 프로토콜은 [그림 11-2]와 같이 5계층으로 구성
- 전송 계층의 TCP와 네트워크 계층의 IP로 전체 프로토콜을 대표하여 TCP/IP 프로토콜이라고 함

표 11-1 TCP/IP 프로토콜 모델의 계층별 역할과 대표 프로토콜

계층	기능	프로토콜	전송 단위
응용 계층	서비스 제공 응용 프로그램	DNS, FTP, SSH, HTTP, 텔넷	메시지
전송 계층	응용 프로그램으로 데이터 전달, 데이터 흐름 제어 및 전송 신뢰성 담당	TCP, UDP	세그먼트
네트워크 계층	주소 관리 및 경로 탐색	IP, ICMP	패킷
링크 계층	네트워크 장치 드라이버	ARP	프레임
물리 계층	케이블 등 전송 매체	구리선, 광케이블, 무선	비트

응용 계층(application layer)

전송 계층(transport layer)

네트워크 계층(network layer)

링크 계층(link layer)

물리 계층(physical layer)

그림 11-2 TCP/IP 프로토콜 모델

■ 주소

- 일반적으로 네트워크 인터페이스는 거의 대부분 이더넷(ethernet) 방식을 사용
- 컴퓨터의 주소는 MAC 주소, IP 주소, 호스트명이 있음

■ MAC 주소

- MAC는 'media access control'의 약자
- MAC 주소는 하드웨어를 위한 주소이며 다른 말로 이더넷 주소, 하드웨어 주소, 물리 주소라고도 함
- MAC 주소는 네트워크 인터페이스 카드(다른 말로 랜 카드)에 저장된 주소라고 생각하면 됨
- MAC 주소는 쌍점(:)이나 붙임표(-)로 구분되는 여섯 개의 16진수로 구성되며, 총 48비트
 - 앞의 세 자리는 제조사 번호, 뒤의 세 자리는 일련번호

00:50:56:3e:3c:fe

00:50:56	3e:3c:fe
제조사 번호	일련번호
(IEEE에서 지정)	(제조사에서 지정)

그림 11-3 MAC 주소의 예

■ IP 주소

- 우리가 보통 인터넷 주소라고 부르는 것이 IP(internet protocol) 주소
- IP 주소는 인터넷으로 연결된 네트워크에서 각 컴퓨터를 구분하기 위해 사용
- IP 주소는 1바이트의 크기를 가진 네 자리 숫자로 구성되므로 총 4바이트
 - 예를 들어 192.168.100.5와 같이 숫자 네 가지와 마침표(.)로 구성
- IP 주소는 네트워크를 구분하는 네트워크 주소 부분과, 해당 네트워크 안에서 특정 컴퓨터를 식별하는 호스트 주소로 구분
- IP 주소는 총 32비트(4바이트) 중 몇 비트를 네트워크 부분으로 사용하고 나머지 몇 비트를 호스트 부분으로 사용하는지에 따라 A 클래스, B 클래스, C 클래스로 구분
- C 클래스의 구조
 - 앞의 3바이트가 네트워크 부분
 - 뒤의 1바이트만 호스트 부분으로 사용
 - 호스트 부분으로 사용할 수 있는 숫자 1~254
 - 예: 192.168.100.5는 C 클래스이므로 네트워크 부분은 앞의 세 자리인 192.168.100이고, 뒤의 5는 호스트 부분

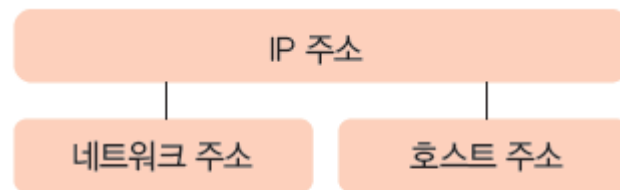


그림 11-5 IP 주소의 구성

- 넷마스크: IP 주소에서 네트워크 부분을 알려주는 역할
- 넷마스크는 하나의 네트워크를 다시 작은 네트워크(서브넷)로 분리할 때도 사용하는데, 그래서 서브넷 마스크라고 부르기도 함
- 넷마스크 예

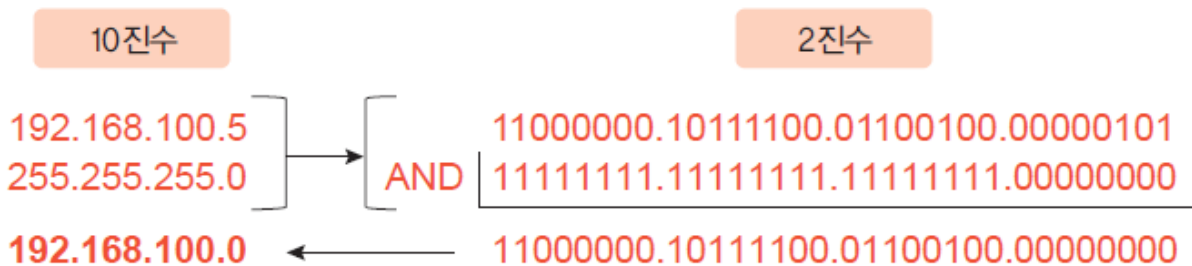


그림 11-6 넷 마스크 계산의 예

- IP 주소와 넷마스크를 10진수에서 2진수로 바꾼 다음, 두 값을 가지고 AND 연산을 수행
- AND연산을 하면 네트워크 부분만 남고 호스트 부분은 0이 됨
- 넷마스크는 IP 주소와 AND 연산을 수행하여 네트워크 부분만 남기는 역할
- 브로드캐스트 주소: 같은 네트워크에 있는 모든 컴퓨터에 메시지를 보낼 때 사용하는 것
 - 호스트 부분을 모두 1로 설정
 - 예: IP 주소에서 네트워크 부분이 192.168.100.0이면 브로드캐스트 주소는 192.168.100.255

01 네트워크의 기초

■ 호스트 이름

- 사람은 숫자보다는 이름으로 된 것을 더 잘 기억한다. 그래서 나온 것이 호스트 이름.
- 호스트 이름도 IP 주소처럼 두 부분으로 구성
 - 예: 네이버를 예로 든다면 naver.com이 네트워크 부분, www가 호스트 부분에 해당

■ 포트 번호

- 각 서비스를 구분하는 번호
- 포트 번호는 TCP/IP 프로토콜의 4계층인 전송 계층에서 사용하는 번호
- /etc/services 파일에 포트 번호 저장

```
user1@myubuntu:~$ cat /etc/services
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services .
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.
(생략)
user1@myubuntu:~$
```

■ 네트워크를 설정하려면

- IP 주소, 넷마스크와 브로드캐스트 주소, 게이트웨이(라우터) 주소, DNS 주소

■ 네트워크 관리자

- 네트워크 관리자는 네트워크의 제어와 설정을 관리하는 데몬
- 네트워크 관리자를 사용하여 IP 주소 설정, 고정 라우트 설정, DNS 설정 등을 수행할 수 있음
- 사용자는 네트워크 관리자와 기존의 스크립트 파일 방식 모두 사용 가능
- 스크립트 방식으로 네트워크를 동작시킬 때

```
sudo systemctl start NetworkManager.service
```

표 11-2 네트워크 관리 도구

도구	기능
네트워크 관리자	기본 네트워킹 데몬이다.
nmcli 명령	네트워크 관리자를 사용하는 명령 기반 도구이다.
[설정]-[네트워크]	그놈에서 제공하는 GUI 기반 도구이다.
nm-connection-editor	네트워크 관리자를 사용하는 GUI 기반 도구로, [제어판]-[네트워크]에서 설정할 수 없는 부분도 설정할 수 있다.
ip 명령	네트워크를 설정하는 명령을 제공한다.

02 네트워크 설정

■ 네트워크 관리자 설치하기: apt 명령

```
sudo apt install network-manager
```

■ 네트워크 관리자 실행하기

- systemctl status 명령으로 동작 확인

```
user1@myubuntu:~$ systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor p
   Active: active (running) since Tue 2017-11-21 23:50:04 KST; 1 day 22h ago
     Docs: man:NetworkManager(8)
   Main PID: 530 (NetworkManager)
      Tasks: 4 (limit: 19660)
    CGroup: /system.slice/NetworkManager.service
            └─ 530 /usr/sbin/NetworkManager --no-daemon
               └─ 21217 /sbin/dhclient -d -q -sf /usr/lib/NetworkManager/nm-dhcp-help
```

(생략)

```
lines 1-20/20 (END)
```

```
user1@myubuntu:~$
```

- 네트워크 관리자의 상태가 **inactive**라면 다음 명령으로 동작

```
sudo systemctl start NetworkManager.service
```

- **enable** 명령: 부팅할 때마다 네트워크 관리자가 동작

```
sudo systemctl enable NetworkManager.service
```

- **네트워크 관리자와 작업하기**

- 네트워크 관리자는 네트워크 설정 정보를 연결 프로파일(connection profile)에 저장
- 사용자는 네트워크 관리자를 직접 제어하지 않고 명령 기반 도구나 GUI 기반 도구를 사용
- nmcli는 네트워크 관리자를 사용하는 명령 기반 도구
- 그놈의 [설정]-[네트워크]나 nm-connectioneditor는 GUI 기반 도구

02 네트워크 설정

■ GUI로 네트워크 설정하기

- 우분투에서는 네트워크 설정을 위한 두 가지 GUI 도구를 제공
 - 그놈에 포함된 [설정]-[네트워크]
 - 네트워크 관리자와 함께 설치되는 nm-connection-editor

■ 그놈의 [설정] -[네트워크]로 설정하기

- 설정 화면에서 네트워크를 선택하면 네트워크 설정 창 실행

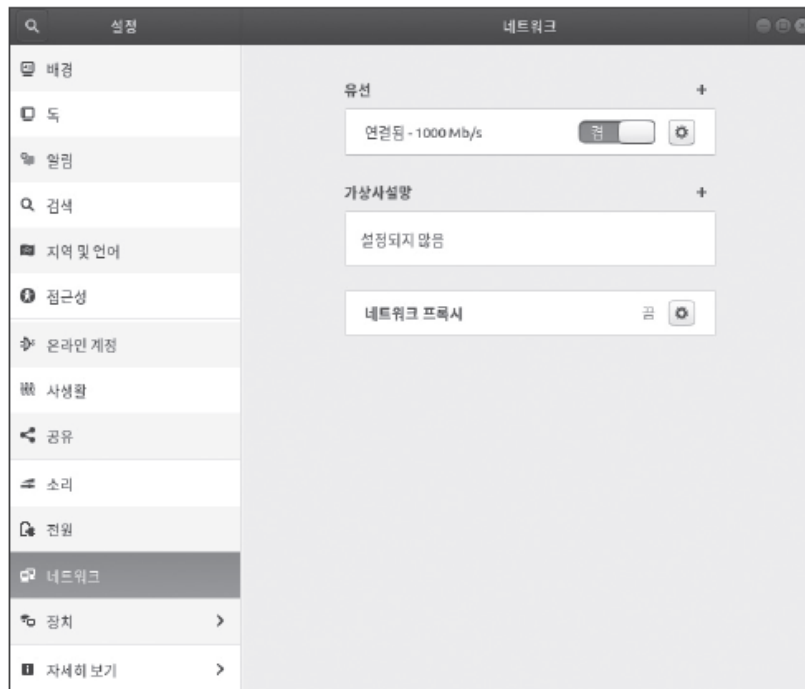


그림 11-7 그놈의 네트워크 설정 창

02 네트워크 설정

■ 그놈의 [설정] -[네트워크]로 설정하기

- IPv4 설정에서 IP 주소와 넷마스크, 게이트웨이, 네임서버(DNS), 라우팅을 설정

취소(C) 케이블 연결

자세히 보기 신원 IPv4 IPv6 보안

IPv4 방식

☐ 자동(DHCP) ☐ 링크 로컬만

☒ 수동 ☐ 사용 않기

주소

주소 넷마스크 게이트웨이

네임서버(DNS) 자동

IP 주소 여러 개는 쉼표로 구분합니다

라우팅 자동

주소 넷마스크 게이트웨이 계속

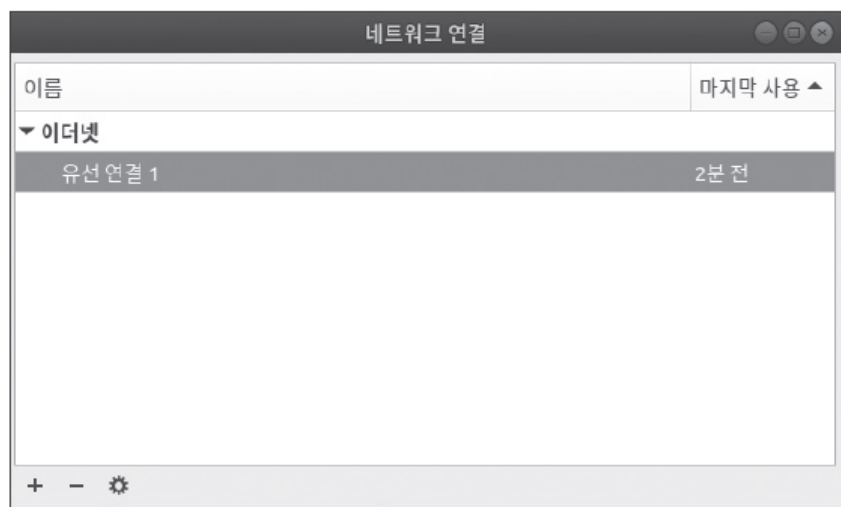
☐ 이 연결은 이 네트워크의 리소스에 대해서만 사용(O)

그림 11-8 유선 설정 창

02 네트워크 설정

■ nm-connection-editor로 설정하기

- (a) 터미널에서 nm-connection-editor를 실행
- (b) '유선 연결 1'을 선택하고 하단의 설정 버튼을 클릭: 세부 설정 가능



(a) 네트워크 연결 창



(b) 유선 연결1 편집 창

■ nmcli 명령으로 네트워크 설정하기

- 네트워크를 설정하는 명령은 네트워크 관리자와 함께 설치되는 nmcli 명령이며, 이와 별도로 리눅스가 제공하는 ip 명령을 사용 가능
- Nmcli 명령으로 유선 네트워크뿐만 아니라 와이파이 등 무선 네트워크, 보안 등 네트워크와 관련된 거의 모든 설정을 관리
- nmcli는 명령 행에서 사용하는 명령은 물론이고 대화식 인터페이스도 제공

nmcli

- **기능** 명령 기반으로 네트워크 관리자를 설정한다.
- **형식** nmcli [옵션] {명령} [서브 명령]
- **옵션**
 - t: 실행 결과를 간단하게 출력한다.
 - p: 사용자가 읽기 좋게 출력한다.
 - v: nmcli의 버전을 출력한다.
 - h: 도움말을 출력한다.
- **명령 {서브 명령}**
 - general {status | hostname}: 네트워크 관리자의 전체적인 상태를 출력하고, 호스트명을 읽거나 변경할 수 있다.
 - networking {on | off | connectivity}: 네트워크를 시작·종료하고 연결 상태를 출력한다.
 - connection {show | up | down | modify | add | delete | reload | load}: 네트워크를 설정한다.
 - device {status | show}: 네트워크 장치의 상태를 출력한다.
- **사용 예**

```
nmcli general
nmcli networking on
nmcli con add type ethernet con-name test-net ifname ens33 ip4
192.168.1.10/24 gw4 192.168.1.254
```

■ 네트워크의 전체 상태 살펴보기: general(gen) 명령

- 네트워크의 전체적인 상태는 nmcli의 general 명령으로 확인
- nmcli를 사용할 때 명령을 줄여서 사용 가능

```
user1@myubuntu:~$ nmcli general status
상태      연결   WIFI-HW  WIFI    WWAN-HW  WWAN
연결됨    전체   사용      사용     사용      사용
user1@myubuntu:~$ nmcli gen
상태      연결   WIFI-HW  WIFI    WWAN-HW  WWAN
연결됨    전체   사용      사용     사용      사용
user1@myubuntu:~$
```

■ 네트워크 활성화·비활성화하기: **networking(net)** 명령

- **networking** 명령은 네트워크를 활성화(on)하거나 비활성화(off)함
- **connectivity** 서브 명령으로 네트워크의 연결 상태를 알려줌
- **connectivity**가 출력하는 네트워크 상태
 - none(없음): 호스트가 아직 네트워크에 연결되어 있지 않다.
 - limited(제한적): 호스트가 네트워크에 연결되어 있지만 인터넷과 연결되지는 않았다.
 - full(전체): 호스트가 네트워크에 연결되어 있고 인터넷도 사용할 수 있다.
 - unknown(알 수 없음): 네트워크 연결 상태를 알 수 없다.
- **nmcli net off**: 네트워크 비활성화 / **nmcli net on**: 네트워크 활성화

```
user1@myubuntu:~$ nmcli net off
user1@myubuntu:~$ nmcli net con
없음
user1@myubuntu:~$ nmcli net on
user1@myubuntu:~$ nmcli net con
전체
user1@myubuntu:~$
```


02 네트워크 설정

■ 네트워크 설정하기: connetion(con) 명령

표 11-3 connection의 서브 명령

서브 명령	기능
show	메모리와 디스크에 저장된 네트워크 연결 프로파일을 출력한다. 서브 명령을 지정하지 않을 경우 기본적으로 show를 실행한다.
up	네트워크 연결을 시작한다.
down	네트워크 연결을 중지한다.
modify	연결 프로파일에서 속성을 추가·수정·삭제한다.
add	새로운 연결을 생성한다.
delete	연결의 설정을 삭제한다.
reload	연결과 관련된 파일을 디스크에서 다시 읽어온다.
load	디스크에서 하나 이상의 연결 파일을 읽어온다.

■ 네트워크 연결 프로파일 출력하기: show

- nmcli connection show 명령은 연결 프로파일의 이름과 UUID, 네트워크 유형, 연결된 장치명을 출력

```
user1@myubuntu:~$ nmcli con show
```

이름	UUID	유형	장치
유선 연결 1	ed452947-83cd-3836-89f5-e706409c26e1	802-3-ethernet	ens33

```
user1@myubuntu:~$
```

■ 네트워크 연결 중지하기: down

- nmcli connection down 명령은 네트워크의 연결을 중지

```
user1@myubuntu:~$ nmcli con down '유선 연결 1'
연결 '유선 연결 1'이(가) 성공적으로 비활성화되었습니다(D-Bus 활성 경로: /org/freedesktop/NetworkManager/ActiveConnection/3).
user1@myubuntu:~$ nmcli con show
```

이름	UUID	유형	장치
유선 연결 1	ed452947-83cd-3836-89f5-e706409c26e1	802-3-ethernet	--

```
user1@myubuntu:~$
```

■ 네트워크 연결 시작하기: up

- nmcli connection up 명령은 네트워크의 연결을 시작

```
user1@myubuntu:~$ nmcli con up '유선 연결 1'
연결이 성공적으로 활성화되었습니다 (D-Bus 활성 경로: /org/freedesktop/NetworkManager/ActiveConnection/4)
user1@myubuntu:~$ nmcli con show
```

이름	UUID	유형	장치
유선 연결 1	ed452947-83cd-3836-89f5-e706409c26e1	802-3-ethernet	ens33

```
user1@myubuntu:~$
```

■ 네트워크 연결 추가하기: add

- 예) IPv4 이더넷 연결을 추가하려면 다음과 같은 형식으로 실행

```
nmcli connection add type ethernet con-name connection-name ifname interface-  
name ip4 address gw4 address
```

- 고정 IP를 사용할 경우

```
user1@myubuntu:~$ sudo nmcli con add type ethernet con-name test-net ifname  
ens33 ip4 192.168.1.10/24 gw4 192.168.1.1  
연결 'test-net' (b3172697-5f67-47bc-bfd5-82b9f7ed9487)이 성공적으로 추가되었습니다.  
user1@myubuntu:~$
```

- 연결 프로파일 이름(connection name): test-net
- 네트워크 장치명(ifname): ens33
- IPv4 주소: 192.168.1.10/24(/24는 넷마스크를 표시: 255.255.255.0)
- 게이트웨이 주소(gw4): 192.168.1.1
- show 명령으로 확인해보면 test-net으로 연결이 추가되었음

```
user1@myubuntu:~$ nmcli con show
```

이름	UUID	유형	장치
유선 연결 1	ed452947-83cd-3836-89f5-e706409c26e1	802-3-ethernet	ens33
test-net	b3172697-5f67-47bc-bfd5-82b9f7ed9487	802-3-ethernet	--

```
user1@myubuntu:~$
```

■ 네트워크 연결 추가하기: add

- add 명령으로 연결을 추가한 후 새로운 이더넷 연결을 시작하려면 up 명령을 사용

```
nmcli con up test-net ifname ens33
```

- show 명령으로 확인: test-net 연결에 장치명이 할당

```
user1@user1-pc:~$ nmcli con show
```

이름	UUID	유형	장치
test-net	b3172697-5f67-47bc-bfd5-82b9f7ed9487	802-3-ethernet	ens33
유선 연결 1	ed452947-83cd-3836-89f5-e706409c26e1	802-3-ethernet	--

```
user1@user1-pc:~$
```

■ 네트워크 연결 추가하기: add

- nmcli 명령으로 확인: IPv4 주소가 192.168.1.10으로 바뀜

```
user1@myubuntu:~$ nmcli
ens33: 연결됨 to test-net
    "Intel 82545EM Gigabit Ethernet Controller (Copper) (PRO/1000 MT Single
Port Adapter)"
    ethernet (e1000), 00:0C:29:74:D8:B8, hw, mtu 1500
    ip4 default
    inet4 192.168.1.10/24
    route4 169.254.0.0/16
    inet6 fe80::54ac:60a3:7199:3a25/64
(생략)
```

- 동적 IP 연결을 추가할 경우

```
nmcli connection add type ethernet con-name connection-name ifname interface-
name
```

02 네트워크 설정

■ 네트워크 연결 수정하기: modify(mod)

- nmcli connection modify 명령은 기존의 연결 프로파일을 수정

```
nmcli connection modify connection-name setting.property value
```

표 11-4 설정과 속성의 예

설정	속성	값의 유형	기능
connection	autoconnection	boolean (TRUE/FALSE)	자원이 사용 가능해지면 네트워크 관리자가 자동으로 연결 할지를 지정한다.
	id	문자열	사용자가 읽을 수 있는 연결의 이름이다.
	interface-name	문자열	네트워크 장치의 이름이다.
	type	문자열	연결의 유형이다.
ipv4	addresses	주소	IP 주소이다.
	dns	주소	DNS 서버의 IP 주소이다.
	gateway	주소	게이트웨이 주소이다.
	method	문자열	IP 구성 방법으로 manual은 고정 IP 사용, auto는 동적 IP 사용을 의미한다.
	routes	주소	네트워크의 경로를 설정한다(예: ipv4.routes "192.168. 1.0/24 192.168.1.1").

■ 네트워크 연결 수정하기: modify(mod)

- test-net 연결의 IPv4 주소를 변경할 경우

```
nmcli con mod test-net ipv4.addresses 192.168.1.12
```

- 주소 추가

```
nmcli con mod test-net +ipv4.addresses 192.168.1.13
```

- 게이트웨이 수정

```
nmcli con mod test-net ipv4.gateway 192.168.1.254
```

- 특정 네트워크로 가는 경로 지정

```
nmcli con mod test-net +ipv4.routes "192.168.2.0/24 192.168.1.1"
```

- 연결 프로파일의 내용을 수정하면 up 명령으로 다시 적용

```
nmcli con up test-net ifname ens33
```

■ 네트워크 연결 삭제하기: delete(del)

- nmcli connection delete 명령은 연결 프로파일을 삭제

```
nmcli connection delete connection-name
```

- 연결 프로파일 test-net을 삭제할 경우

```
user1@myubuntu:~$ nmcli connection delete test-net
연결 'test-net' (b3172697-5f67-47bc-bfd5-82b9f7ed9487)이(가) 성공적으로 삭제되었습니다.
user1@myubuntu:~$ nmcli con show
```

이름	UUID	유형	장치
유선 연결 1	ed452947-83cd-3836-89f5-e706409c26e1	802-3-ethernet	ens33

```
user1@myubuntu:~$
```

■ 네트워크 연결 프로파일 읽어오기: reload, load

- 연결 프로파일이 수정되었는지 관리자에게 알려주기 위해 reload나 load 명령 사용

```
nmcli connection reload
nmcli connection load connection-name
```


02 네트워크 설정

■ 네트워크 장치의 상태 보기: `device(dev)` 명령

- `device` 명령은 네트워크 장치의 상태를 출력하고 관리

■ 네트워크 장치의 상태 보기: `status`

```
user1@myubuntu:~$ nmcli dev status
```

장치	유형	상태	연결
ens33	ethernet	연결됨	유선 연결 1
lo	loopback	관리되지 않음	--

```
user1@myubuntu:~$
```

■ 네트워크 장치의 상세 정보 보기: `show`

```
user1@myubuntu:~$ nmcli dev show
```

GENERAL.장치:	ens33
GENERAL.유형:	ethernet
GENERAL.하드웨어주소:	00:0C:29:74:D8:B8
GENERAL.MTU:	1500
GENERAL.상태:	100 (연결됨)

(생략)

■ 네트워크 장치의 상세 정보 보기: show

- show 명령 다음에 장치명을 지정하면 해당 장치의 상세 정보(ens33)가 출력

```
user1@myubuntu:~$ nmcli dev show ens33
GENERAL.장치:                ens33
GENERAL.유형:                ethernet
GENERAL.하드웨어주소:      00:0C:29:74:D8:B8
GENERAL.MTU:                1500
GENERAL.상태:                100 (연결됨)
GENERAL.연결:                유선 연결 1
GENERAL.CON-경로:            /org/freedesktop/NetworkManager/ActiveConnection/7
WIRED-PROPERTIES.캐리어:    켜짐
IP4.주소[1]:                192.168.1.3/24
IP4.게이트웨이:              192.168.1.1
IP4.라우팅[1]:              dst = 169.254.0.0/16, nh = 0.0.0.0, mt = 1000
IP4.DNS[1]:                 192.168.1.1
IP4.도메인[1]:              netis
IP6.주소[1]:                fe80::14c1:f467:d0ee:1539/64
IP6.게이트웨이:              --
user1@myubuntu:~$
```

■ ip 명령으로 네트워크 설정하기

- 네트워크 설정은 ip 명령으로도 가능하지만, 시스템을 재시작하면 사라짐
- 설정한 내용이 시스템을 다시 시작한 후에도 적용되게 하려면 설정 파일에 저장해야 함

ip

- **기능** IP 주소, 게이트웨이, 네트워크 장치의 상태 등을 출력하고 관리한다.
- **형식** ip [옵션] 객체 [서브 명령]
- **옵션** -V: 버전을 출력한다.
-s: 자세한 정보를 출력한다.
- **객체 [서브 명령]** address [add|del|show|help]: 장치의 IP 주소를 관리한다(ip-address).
route [add|del|help]: 라우팅 테이블을 관리한다(ip-route).
link [set]: 네트워크 인터페이스를 활성화·비활성화한다.
- **사용 예** ip addr show
ip addr add 192.168.1.20/24 dev ens33
ip route show
ip route add 192.168.2.0/24 via 192.168.1.1 dev ens33

■ 네트워크 장치의 주소 관리하기: `address(addr)` 명령

- `address` 명령은 IP 주소의 정보를 출력하거나 설정하고 삭제

■ 네트워크 장치의 정보 보기: `show`

- `show` 명령은 네트워크 장치의 정보를 출력
- `show` 명령 다음에 장치명을 지정하지 않으면 전체 장치에 대한 상세 정보가 출력
- 예) 전체 장치의 정보를 출력한 것

```
user1@myubuntu:~$ 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:0c:29:74:d8:b8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.3/24 brd 192.168.1.255 scope global dynamic ens33
        valid_lft 85982sec preferred_lft 85982sec
    inet6 fe80::14c1:f467:d0ee:1539/64 scope link
        valid_lft forever preferred_lft forever
user1@myubuntu:~$
```

02 네트워크 설정

■ 네트워크 장치의 정보 보기: show

- show 명령에 특정 장치를 지정하면 해당 장치(ens33)의 정보만 출력

```
user1@myubuntu:~$ ip addr show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:0c:29:74:d8:b8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.3/24 brd 192.168.1.255 scope global dynamic ens33
        valid_lft 85946sec preferred_lft 85946sec
    inet6 fe80::14c1:f467:d0ee:1539/64 scope link
        valid_lft forever preferred_lft forever
user1@myubuntu:~$
```

■ IP 주소 설정하기: add

- ens33 장치에 ip 명령으로 고정 IP 주소를 설정

```
user1@myubuntu:~$ sudo ip addr add 192.168.1.20/24 dev ens33
```

■ IP 주소 삭제하기: del

- ens33 장치에 설정된 192.168.1.20 주소를 삭제

```
user1@myubuntu:~$ sudo ip addr del 192.168.1.20/24 dev ens33
user1@myubuntu:~$ ip addr show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:0c:29:74:d8:b8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.3/24 brd 192.168.1.255 scope global dynamic ens33
        valid_lft 85387sec preferred_lft 85387sec
    inet6 fe80::14c1:f467:d0ee:1539/64 scope link
        valid_lft forever preferred_lft forever
user1@myubuntu:~$
```

02 네트워크 설정

■ 라우팅 테이블과 게이트웨이 주소 관리하기: route 명령

- ip route 명령은 라우팅 테이블을 출력하거나 게이트웨이를 설정하고 삭제
- 네트워크를 다른 네트워크와 연결할 때 연결점이 되는 장치를 게이트웨이
- 게이트웨이는 패킷을 보고 같은 네트워크로 보내는 것이 아니면 외부로 전송

■ 라우팅 테이블 보기: show

- 기본 게이트웨이 주소가 192.168.1.1이며, ens33 장치를 통해 접속

```
user1@myubuntu:~$ ip route show
default via 192.168.1.1 dev ens33 proto static metric 100
169.254.0.0/16 dev ens33 scope link metric 1000
192.168.1.0/24 dev ens33 proto kernel scope link src 192.168.1.3 metric 100
user1@myubuntu:~$
```

■ 기본 게이트웨이 주소 설정하기: add

```
user1@myubuntu:~$ sudo ip route add default via 192.168.1.1 dev ens3
```

■ 라우팅 경로 설정하기: add

- 기본 게이트웨이 이외에 경로 추가: 192.168.2.0 네트워크를 192.168.1.1을 통해 접속

```
user1@myubuntu:~$ sudo ip route add 192.168.2.0/24 via 192.168.1.1 dev ens33
user1@myubuntu:~$ ip route show
default via 192.168.1.1 dev ens33
default via 192.168.1.1 dev ens33 proto static metric 100
169.254.0.0/16 dev ens33 scope link metric 1000
192.168.1.0/24 dev ens33 proto kernel scope link src 192.168.1.3 metric 100
192.168.2.0/24 via 192.168.1.1 dev ens33
user1@myubuntu:~$
```

■ 라우팅 경로 삭제하기: del

- 라우팅 테이블에서 경로를 삭제하려면 ip route del 명령을 사용

```
user1@myubuntu:~$ sudo ip route del 192.168.2.0/24
user1@myubuntu:~$ ip route show
default via 192.168.1.1 dev ens33
default via 192.168.1.1 dev ens33 proto static metric 100
169.254.0.0/16 dev ens33 scope link metric 1000
192.168.1.0/24 dev ens33 proto kernel scope link src 192.168.1.3 metric 100
user1@myubuntu:~$
```


02 네트워크 설정

■ 네트워크 인터페이스 활성화·비활성화하기: link 명령

- 네트워크 인터페이스를 활성화 또는 비활성화하려면 ip link set 명령을 사용

■ 네트워크 인터페이스 비활성화하기: down

- 네트워크를 비활성화하면 state가 DOWN이 되고 IP 주소도 할당되지 않음

```
user1@myubuntu:~$ sudo ip link set ens33 down
user1@myubuntu:~$ ip addr show ens33
2: ens33: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN group
default qlen 1000
    link/ether 00:0c:29:7b:79:49 brd ff:ff:ff:ff:ff:ff
user1@myubuntu:~$
```

■ 네트워크 인터페이스 활성화하기 : up

- 네트워크를 활성화하면 state가 UP이 되고 IP 주소도 할당

```
user1@myubuntu:~$ sudo ip link set ens33 up
user1@myubuntu:~$ ip addr show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:0c:29:74:d8:b8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.3/24 brd 192.168.1.255 scope global dynamic ens33
        valid_lft 86339sec preferred_lft 86339sec
    inet6 fe80::14c1:f467:d0ee:1539/64 scope link
        valid_lft forever preferred_lft forever
user1@myubuntu:~$
```

■ 기존 명령으로 네트워크 설정하기: ifconfig

ifconfig

- **기능** 네트워크 인터페이스의 IP 주소를 설정한다.
- **형식** ifconfig [인터페이스명] [옵션] [값]
- **옵션**
 - a: 시스템의 전체 인터페이스에 대한 정보를 출력한다.
 - up/down: 인터페이스를 활성화·비활성화한다.
 - netmask 주소: 넷마스크 주소를 설정한다.
 - broadcast 주소: 브로드캐스트 주소를 설정한다.
- **사용 예**

```
ifconfig ens33
ifconfig ens33 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255
```

- 우분투는 ifconfig 명령이 기본으로 설치되지 않으므로 net-tools 패키지를 설치

```
user1@myubuntu:~$ sudo apt install net-tools
```

```
패키지 목록을 읽는 중입니다... 완료
```

```
의존성 트리를 만드는 중입니다
```

```
상태 정보를 읽는 중입니다... 완료
```

```
다음 새 패키지를 설치할 것입니다:
```

```
net-tools
```

```
0개 업그레이드, 1개 새로 설치, 0개 제거 및 57개 업그레이드 안 함.
```

```
194 k바이트 아카이브를 받아야 합니다.
```

```
이 작업 후 803 k바이트의 디스크 공간을 더 사용하게 됩니다.
```

```
(생략)
```

```
net-tools (1.60+git20161116.90da8a0-1ubuntu1) 설정하는 중입니다 ...
```

```
user1@myubuntu:~$
```

■ 현재 설치된 네트워크 인터페이스 설정 보기

- 옵션 없이 ifconfig 명령을 사용하면 현재 설치된 네트워크 인터페이스의 설정 내용을 출력

```
user1@myubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.3 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::14c1:f467:d0ee:1539 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:74:d8:b8 txqueuelen 1000 (Ethernet)
    RX packets 93380 bytes 80449450 (80.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 74715 bytes 9450141 (9.4 MB)
    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 1190 bytes 86700 (86.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1190 bytes 86700 (86.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

user1@myubuntu:~$
```

■ 현재 설치된 네트워크 인터페이스 설정 보기

- 앞의 예를 보면 네트워크가 다음과 같이 설정되었음
 - MAC 주소(ether): 00:0c:29:74:d8:b8
 - IP 주소(inet): 192.168.1.3
 - 넷마스크(netmask): 255.255.255.0
 - 브로드캐스트 주소(broadcast): 192.168.1.255
 - IPv6 주소(inet6): fe80::14c1:f467:d0ee:1539
- RX는 부팅 후 현재까지 받은 패킷 수와 바이트 수를 알려줌
- TX는 부팅 후 현재까지 보낸 패킷 수와 바이트 수를 알려줌

■ 특정 네트워크 인터페이스 설정 보기

```
user1@myubuntu:~$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.3 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::14c1:f467:d0ee:1539 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:74:d8:b8 txqueuelen 1000 (Ethernet)
    RX packets 93410 bytes 80452170 (80.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 74730 bytes 9452765 (9.4 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

user1@myubuntu:~$
```

■ 네트워크 인터페이스 사용 해제하기: down 옵션

```
user1@myubuntu:~$ sudo ifconfig ens33 down
user1@myubuntu:~$ ifconfig ens33
ens33: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 08:00:0c:29:74:d8:b8 txqueuelen 1000 (Ethernet)
    RX packets 93431 bytes 80454246 (80.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 74741 bytes 9454581 (9.4 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

user1@myubuntu:~$
```

- 인터페이스가 다운되었을 때 flags를 보면 UP과 RUNNING이 없어지고, 둘째 줄과 셋째 줄에 있던 IP 주소 관련 부분인 inet과 inet6이 출력되지 않음

■ 네트워크 인터페이스 활성화하기: up 옵션

```
user1@myubuntu:~$ sudo ifconfig ens33 up
user1@myubuntu:~$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.3 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::14c1:f467:d0ee:1539 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:74:d8:b8 txqueuelen 1000 (Ethernet)
    RX packets 93525 bytes 80465545 (80.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 74825 bytes 9464695 (9.4 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

user1@myubuntu:~$
```

- 인터페이스를 up 옵션으로 활성화하면 flags에 UP과 RUNNING이 표시됨

■ 네트워크 인터페이스 수동으로 설정하기

```
ifconfig 인터페이스명 IP 주소 netmask 넷마스크 주소 broadcast 브로드캐스트 주소
```

- 예: IP 주소를 192.168.1.3에서 192.168.1.10으로 바꾸어 설정

```
user1@myubuntu:~$ sudo ifconfig ens33 192.168.1.10 netmask 255.255.255.0
broadcast 192.168.1.255
user1@myubuntu:~$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::14c1:f467:d0ee:1539 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:74:d8:b8 txqueuelen 1000 (Ethernet)
    RX packets 93586 bytes 80472809 (80.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 74877 bytes 9471839 (9.4 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

user1@myubuntu:~$
```

■ 게이트웨이 설정하기

- 인터넷은 네트워크와 네트워크를 연결한 것
- 네트워크를 다른 네트워크와 연결할 때 연결점이 되는 장치가 게이트웨이
- 게이트웨이의 설정과 확인은 route 명령으로 수행

route

- **기능** 라우팅 테이블을 편집하고 출력한다.
- **형식** route [명령]
- **명령** add: 라우팅 경로나 기본 게이트웨이를 추가한다.
del: 라우팅 경로나 기본 게이트웨이를 삭제한다.
- **사용 예** route
route add default gw 192.168.1.1 dev ens33

02 네트워크 설정

■ 게이트웨이 설정하기

표 11-5 route 명령을 사용한 라우팅 테이블 편집

기능	명령 형식과 사용 예
라우팅 경로 추가(네트워크)	route add -net 네트워크 주소 netmask 넷마스크 dev 인터페이스명 route add -net 192.168.1.0 netmask 255.255.255.0 dev ens33
라우팅 경로 추가(호스트)	route add -host 호스트 주소 dev 인터페이스명 route add -host 192.168.1.5 dev ens33
라우팅 경로 삭제(네트워크)	route del -net 네트워크 주소 netmask 넷마스크 [dev 인터페이스명] route del -net 192.168.1.0 netmask 255.255.255.0
라우팅 경로 삭제(호스트)	route del -host 호스트 주소 route del -host 192.168.1.5
기본 게이트웨이 추가	route add default gw 게이트웨이 주소 dev 인터페이스명 route add default gw 192.168.1.1 dev ens33
기본 게이트웨이 삭제	route del default gw 게이트웨이 주소 route del default gw 192.168.1.1
루프백(lo) 추가	route add -net 127.0.0.0 netmask 255.0.0.0 dev lo

■ 라우팅 테이블 보기 : route

```
user1@myubuntu:~$ route
Kernel IP routing table
Destination      Gateway         Genmask        Flags   Metric      Ref    Use    Iface
192.168.1.0      0.0.0.0        255.255.255.0  U       0           0      0      ens33
user1@myubuntu:~$
```

02 네트워크 설정

■ 라우팅 테이블 보기 : route

표 11-6 라우팅 테이블의 출력 항목

항목	기능
Destination	라우팅 대상 네트워크나 호스트의 주소
Gateway	게이트웨이 주소 또는 설정되어 있지 않으면 * 출력
Genmask	대상 네트워크의 넷마스크 255.255.255.255: 대상이 호스트인 경우 0.0.0.0: 기본(default) 경로
Flags	U: 경로 활성화(UP) H: 대상이 호스트 G: 게이트웨이로 사용 R: 동적 라우팅을 위한 경로 재생성 D: 데몬 또는 리다이렉트에 의해 동적으로 재설치 M: 라우팅 데몬 또는 리다이렉트에 의해 경로 수정 A: addrconf에 의해 설치 C: 캐시 항목 !: 경로 거부
Metrics	대상까지의 거리로 최근 커널에서는 사용되지 않지만 라우팅 데몬에서 사용할 수도 있다.
Ref	해당 경로에 대한 참조 수이지만 리눅스 커널에서는 사용하지 않는다.
Use	경로를 탐색한 수
Iface	패킷이 전달되는 인터페이스 이름

02 네트워크 설정

■ 기본 게이트웨이 설정하기: add

```
user1@myubuntu:~$ sudo route add default gw 192.168.1.1 dev ens33
user1@myubuntu:~$ route
Kernel IP routing table
Destination    Gateway      Genmask      Flags    Metric     Ref     Use     Iface
default        gateway      0.0.0.0      UG        0          0        0       ens33
192.168.1.0    0.0.0.0     255.255.255.0  U        0          0        0       ens33
user1@myubuntu:~$
```

■ 기본 게이트웨이 삭제하기: del

```
user1@myubuntu:~$ sudo route del default gw 192.168.1.1
user1@myubuntu:~$ route
Kernel IP routing table
Destination    Gateway      Genmask      Flags    Metric     Ref     Use     Iface
192.168.1.0    0.0.0.0     255.255.255.0  U        0          0        0       ens33
192.168.1.0    0.0.0.0     255.255.255.0  U       100        0        0       ens33
user1@myubuntu:~$
```

02 네트워크 설정

■ 네트워크 인터페이스 설정을 파일에 저장하기

- 부팅할 때 네트워크가 설정되게 하려면 /etc/network/interfaces 파일에 설정

```
user1@myubuntu:~$ cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
user1@myubuntu:~$
```

- 우분투 서버의 경우 고정 IP를 사용하려면 /etc/network/interfaces 파일을 수정

```
user1@myubuntu:~$ sudo vi /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

iface eth0 inet static
address 192.168.1.10
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1
dns-nameservers 168.126.63.1 168.126.63.2
~
:wq
user1@myubuntu:~$
```

■ DNS 설정하기

- 호스트명을 IP 주소로 바꾸는 역할을 수행
- 만약 DNS가 설정되어 있지 않으면 이름으로 서버에 접속할 수 없으며 직접 IP 주소를 사용하여 접속 가능

■ DNS 관리 서비스와 DNS 서버 지정 파일

- DNS 서버의 주소를 /etc/resolv.conf 파일에 저장

```
user1@myubuntu:~$ cat /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# 127.0.0.53 is the systemd-resolved stub resolver.
# run "systemd-resolve --status" to see details about the actual nameservers.
nameserver 127.0.0.53

search netis
user1@myubuntu:~$
```

■ DNS 관리 서비스와 DNS 서버 지정 파일

- DNS 서버의 정보는 systemd-resolve 명령으로 확인

systemd-resolve

- **기능** DNS 서버에 질의하고 응답을 받는다.
- **형식** systemd-resolve [옵션] [호스트명|IP 주소]
- **옵션** --status: DNS 서버 정보를 출력한다.
- **사용 예** systemd-resolve www.daum.net
systemd-resolve --status

- systemd-resolve --status 명령으로 DNS 서버 정보를 확인

```
user1@myubuntu:~$ systemd-resolve --status
Global
    DNSSEC NTA: 10.in-addr.arpa
                  16.172.in-addr.arpa
(생략)
Link 2 (ens33)
    Current Scopes: DNS LLMNR/IPv4 LLMNR/IPv6
    LLMNR setting: yes
MulticastDNS setting: no
    DNSSEC setting: no
    DNSSEC supported: no
    DNS Servers: 192.168.1.1
    DNS Domain: netis
user1@myubuntu:~$
```

■ nmcli 명령으로 DNS 설정하기

- nmcli 명령으로 DNS를 지정하는 방법

```
nmcli con mod connection-name ipv4.dns DNS 주소
```

- 연결 프로파일 ens33에 구글의 DNS 서버 주소인 8.8.8.8과 8.8.4.4를 지정

```
user1@myubuntu:~$ sudo nmcli con mod '유선 연결 1' ipv4.dns "8.8.8.8 8.8.4.4"
user1@myubuntu:~$ sudo nmcli con up '유선 연결 1'
연결이 성공적으로 활성화되었습니다 (D-Bus 활성 경로: /org/freedesktop/NetworkManager/ActiveConnection/11)
user1@myubuntu:~$
```

- systemd-resolve --status 명령으로 확인해보면 DNS 주소가 추가되었음

```
user1@myubuntu:~$ systemd-resolve --status
(생략)
DNSSEC supported: no
DNS Servers: 192.168.1.1
              8.8.8.8
              8.8.4.4
DNS Domain: netis
user1@myubuntu:~$
```


■ DNS 서버에 질의하기: nslookup

nslookup

- **기능** DNS 서버와 대화식으로 질의하고 응답을 받는다.
- **형식** nslookup [도메인명]
- **사용 예** nslookup
 nslookup www.daum.net

- nslookup 명령으로 특정 도메인의 주소를 검색

```
user1@myubuntu:~$ nslookup
> www.hanbit.co.kr
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   www.hanbit.co.kr
Address: 218.38.58.195
> exit

user1@myubuntu:~$
```

■ DNS 서버에 질의하기: nslookup

- systemd-resolve 명령으로도 DNS 서버에 질의 가능

```
user1@myubuntu:~$ systemd-resolve www.hanbit.co.kr
www.hanbit.co.kr: 218.38.58.195

-- Information acquired via protocol DNS in 4.7ms.
-- Data is authenticated: no
user1@myubuntu:~$
```

■ 호스트 이름 확인하기

- 해당 기관의 도메인 이름에 서버에서 제공하는 대표적인 서비스의 이름을 붙이는 것이 편리
- 붙인 이름은 호스트 이름 설정 파일에 저장하고 DNS에 등록해야 서비스 제공 가능

■ `uname -n` 명령

`uname`

- **기능** 시스템 정보를 출력한다.
- **형식** `uname [옵션]`
- **옵션**
 - `-m`: 하드웨어 종류를 출력한다.
 - `-n`: 호스트 이름을 출력한다.
 - `-r`: 운영체제의 릴리스 정보를 출력한다.
 - `-s`: 운영체제 이름을 출력한다.
 - `-v`: 운영체제 버전을 출력한다.
 - `-a`: 위의 모든 정보를 출력한다.
- **사용 예** `uname -n` `uname -a`

03 호스트 이름 설정

■ `uname -n` 명령

- 리눅스를 설치할 때 설정한 호스트 이름인 myubuntu를 출력

```
user1@myubuntu:~$ uname -n  
myubuntu  
user1@myubuntu:~$
```

- `uname -a` 명령을 실행하면 호스트 이름을 포함하여 시스템 관련 정보가 출력

```
user1@myubuntu:~$ uname -a  
Linux myubuntu 4.13.0-16-generic #19-Ubuntu SMP Wed Oct 11 18:35:14 UTC 2017  
x86_64 x86_64 x86_64 GNU/Linux  
user1@myubuntu:~$
```

03 호스트 이름 설정

■ hostname 명령

hostname

- **기능** 호스트 이름을 출력하거나 설정한다.
- **형식** hostname [호스트 이름]
- **사용 예** hostname
hostname mail.han.server

- hostname 명령으로 호스트 이름을 검색하면 다음과 같이 출력

```
user1@myubuntu:~$ hostname
myubuntu
user1@myubuntu:~$
```

- nmcli gen host 명령으로도 같은 결과

```
user1@myubuntu:~$ nmcli gen host
myubuntu
user1@myubuntu:~$
```

■ hostnamectl 명령

hostnamectl

- **기능** 호스트 이름을 검색하거나 설정한다.
- **형식** hostnamectl [옵션] [명령]
- **옵션** -h: 도움말을 출력한다.
--version: 버전을 출력한다.
- **명령** status: 현재 호스트 이름과 관련 정보를 출력한다.
set-hostname 호스트 이름: 호스트 이름을 설정한다.
- **사용 예** hostnamectl
hostnamectl status
hostnamectl set-hostname mail.han.server

- hostnamectl 명령으로 호스트 이름을 검색하면 호스트 이름과 관련된 다른 정보도 출력

```
user1@myubuntu:~$ hostnamectl
  Static hostname: myubuntu
        Icon name: computer-vm
        Chassis: vm
    Machine ID: 520e8c5e038b410c90f79a1ed0400bb6
        Boot ID: 03598a6cb9a0410a977c08ec8e208caa
(생략)
user1@myubuntu:~$
```

03 호스트 이름 설정

■ hostname 명령으로 호스트 이름 설정하기

```
user1@myubuntu:~$ sudo hostname mail.han.server
user1@myubuntu:~$ hostname
mail.han.server
user1@myubuntu:~$
```

- 프롬프트의 호스트 이름 부분이 바뀌었음

```
user1@mail:~$
```

■ hostnamectl 명령으로 호스트 이름 설정하기

- hostnamectl 명령으로 호스트 이름을 다시 myubuntu로 설정

```
user1@myubuntu:~$ sudo hostnamectl set-hostname myubuntu
user1@myubuntu:~$ hostnamectl
  Static hostname: myubuntu
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 520e8c5e038b410c90f79a1ed0400bb6
        Boot ID: 03598a6cb9a0410a977c08ec8e208caa
  Virtualization: vmware
  Operating System: Ubuntu 17.10
        Kernel: Linux 4.13.0-16-generic
        Architecture: x86-64
user1@myubuntu:~$
```

■ 호스트 이름을 파일에 저장하기

- hostname 명령으로 바뀐 호스트 이름은 시스템을 재시작하면 원래의 이름으로 돌아감
- 재시작해도 바뀐 호스트 이름이 유지되게 하려면 호스트 이름을 설정하는 파일 자체를 수정
- 우분투에서 호스트 이름을 저장하는 파일은 /etc/hostname
- /etc/hostname 파일의 내용

```
user1@myubuntu:~$ cat /etc/hostname
myubuntu
user1@myubuntu:~$
```

- 호스트 이름을 새로 정의할 때는 한 네트워크에 같은 이름을 사용하는 다른 호스트가 있으면 안 됨
- 사용자가 임의로 호스트 이름을 수정하면 안 되고 반드시 시스템 관리자와 상의해야 함

04 네트워크 상태 확인

■ 통신 확인하기: ping

ping

- **기능** 네트워크 장비에 신호(ECHO_REQUEST)를 보낸다.
- **형식** ping [옵션] 목적지 주소
- **옵션**
 - a: 통신이 되면 소리를 낸다.
 - q: 테스트 결과를 지속적으로 보여주지 않고 종합 결과만 출력한다.
 - c 개수: 보낼 패킷 수를 지정한다.
- **사용 예**
 - ping 192.168.0.1
 - ping -a www.naver.com

- 옵션 없이 사용하는 경우: 패킷은 기본적으로 56바이트의 크기 + 8바이트 헤더 정보

```
user1@myubuntu:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data:
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=146 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=20.0 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=1.68 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=166 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=86.1 ms
(생략)
```

■ 통신 확인하기: ping

- ping은 보낼 패킷 수를 -c 옵션으로 지정하지 않으면 계속 패킷을 보내므로 Ctrl+c로 ping 종료
- ping이 종료되면 다음과 같이 통계 정보가 출력

```
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=86.1 ms
^C
--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.686/84.183/166.207/65.683 ms|
user1@myubuntu:~$
```

- 통계 정보로는 보낸 전체 패킷 수, 보낸 패킷 중에서 받은 패킷 수, 보내고 받은 패킷 중 유실된 패킷의 비율, 통신 속도의 최솟값, 평균값, 최댓값이 출력
- 결과를 보면 다섯 개 패킷을 보내 모두 수신했으며, 유실된 것은 없고 평균 84.183밀리초가 걸렸음
- 통신 시간이 낮을수록 네트워크의 상태가 양호하다는 것을 의미

■ -q 옵션 사용하기

- -q 옵션을 사용하면 아무 메시지도 출력되지 않다가 +C로 종료하면 통계 정보만 출력

```
user1@myubuntu:~$ ping -q 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 1.651/3.138/4.919/1.498 ms
user1@myubuntu:~$
```

■ -c 옵션 사용하기

- -c 옵션을 사용하면 보낼 패킷 수를 지정

```
user1@myubuntu:~$ ping -c 3 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=4.38 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1.66 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=1.67 ms

--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 1.663/2.572/4.380/1.278 ms
user1@myubuntu:~$
```

■ 도메인 이름을 사용하는 경우

```
user1@myubuntu:~$ ping www.hanbit.co.kr
PING www.hanbit.co.kr (218.38.58.195) 56(84) bytes of data.
64 bytes from 218.38.58.195 (218.38.58.195): icmp_seq=1 ttl=54 time=3.05 ms
64 bytes from 218.38.58.195 (218.38.58.195): icmp_seq=2 ttl=54 time=3.34 ms
64 bytes from 218.38.58.195 (218.38.58.195): icmp_seq=3 ttl=54 time=3.09 ms
64 bytes from 218.38.58.195 (218.38.58.195): icmp_seq=4 ttl=54 time=3.39 ms
^C
--- www.hanbit.co.kr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 3.056/3.220/3.390/0.164 ms
user1@myubuntu:~$
```

- 시스템에 따라서 보안을 강화하기 위해 ping 패킷이 왔을 때 응답하지 않도록 설정하는 경우도 있으므로, ping으로 연결되지 않는다고 해서 무조건 해당 시스템이 동작하지 않는다는 것은 아니다

■ 통신 경로 확인하기

traceroute

- **기능** 목적지까지 패킷이 거치는 경로를 출력한다.
- **형식** traceroute 목적지 주소
- **사용 예** traceroute 192.168.1.1
traceroute www.naver.com

- 만약 traceroute가 설치되어 있지 않다면 apt 명령으로 설치

04 네트워크 상태 확인

■ 정상으로 경로가 확인되는 경우

```
user1@myubuntu:~$ traceroute www.hanbit.co.kr
traceroute to www.hanbit.co.kr (218.38.58.195), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.11.1)  0.305 ms  0.300 ms  0.335 ms
 2  192.168.200.3 (192.168.200.3)  0.126 ms  0.106 ms  0.106 ms
 3  121.157.127.17 (121.157.127.17)  0.931 ms  0.711 ms  0.770 ms
 4  * * *
 5  112.189.128.53 (112.189.128.53)  1.193 ms  1.201 ms  1.136 ms
 6  112.174.114.25 (112.174.114.25)  3.380 ms  2.801 ms  112.174.114.37 (112.174.114.37)
   3.060 ms
 7  112.174.51.102 (112.174.51.102)  5.389 ms  112.174.76.126 (112.174.76.126)  4.534
   ms  4.477 ms
 8  203.229.222.106 (203.229.222.106)  6.223 ms  6.867 ms  6.851 ms
 9  118.221.7.73 (118.221.7.73)  11.062 ms  175.124.170.85 (175.124.170.85)  7.199 ms
   7.191 ms
10  58.229.8.106 (58.229.8.106)  8.953 ms  211.108.63.138 (211.108.63.138)  3.414 ms
   3.379 ms
11  58.229.17.214 (58.229.17.214)  2.251 ms  2.799 ms  2.875 ms
12  114.202.0.218 (114.202.0.218)  3.034 ms  114.202.0.222 (114.202.0.222)  2.739 ms
   3.079 ms
13  * * *
14  218.38.58.195 (218.38.58.195)  2.470 ms  2.891 ms  2.955 ms
user1@myubuntu:~$
```

04 네트워크 상태 확인

■ 중간 노드의 기관 확인: whois 명령 사용

- 예: 5번에 출력된 112.189.128.53가 어느 기관의 것인지?

```
user1@myubuntu:~$ whois 112.189.128.53
query : 112.189.128.53
```

```
# KOREAN(UTF8)
```

조회하신 IPv4주소는 한국인터넷진흥원으로부터 아래의 관리대행자에게 할당되었으며, 할당 정보는 다음과 같습니다.

[네트워크 할당 정보]

IPv4주소	: 112.160.0.0 - 112.191.255.255 (/11)
기관명	: 주식회사 케이티
서비스명	: KORNET
주소	: 경기도 성남시 분당구 불정로 90
우편번호	: 13606
할당일자	: 20090210

이름	: IP주소 담당자
전화번호	: +82-2-500-6630
전자우편	: kornet_ip@kt.com

(생략)

```
user1@myubuntu:~$
```

04 네트워크 상태 확인

■ 경로가 확인되지 않는 경우

- 네트워크가 연결되지 않는 구간은 *가 출력: 통신에 장애가 있거나 traceroute 명령을 거부한다는 의미

```
user1@myubuntu:~$ traceroute www.naver.com
traceroute to www.naver.com (125.209.222.141), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  0.406 ms  0.420 ms  0.470 ms
 2  192.168.200.3 (192.168.200.3)  0.132 ms  0.125 ms  0.111 ms
 3  121.157.127.17 (121.157.127.17)  0.677 ms  0.743 ms  0.824 ms
 4  * * *
 5  112.189.127.53 (112.189.127.53)  1.100 ms  1.075 ms  1.242 ms
 6  112.174.98.45 (112.174.98.45)  2.363 ms  112.174.98.17 (112.174.98.17)  2.325 ms
    112.174.98.25 (112.174.98.25)  4.329 ms
 7  112.174.25.146 (112.174.25.146)  1.528 ms  112.174.71.162 (112.174.71.162)  2.236
ms  1.919 ms
 8  121.189.2.250 (121.189.2.250)  2.487 ms  121.189.2.6 (121.189.2.6)  2.137 ms
    121.189.2.206 (121.189.2.206)  41.055 ms
 9  * * *
10  * * *
11  * * *
12  * * *
(생략)
```


■ 네트워크 상태 정보 출력하기

netstat

- **기능** 네트워크의 상태 정보를 출력한다.
- **형식** netstat [옵션]
- **옵션**
 - a: 모든 소켓 정보를 출력한다.
 - r: 라우팅 정보를 출력한다.
 - n: 호스트명 대신에 IP 주소를 출력한다.
 - i: 모든 네트워크 인터페이스 정보를 출력한다.
 - s: 프로토콜별로 네트워크 통계 정보를 출력한다.
 - p: 해당 소켓과 관련된 프로세스의 이름과 PID를 출력한다.
- **사용 예** netstat -rn netstat -s

04 네트워크 상태 확인

■ 라우팅 테이블 확인하기 : -r 옵션

```
user1@myubuntu:~$ netstat -r
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
default	gateway	0.0.0.0	UG	0	0	0	ens33
link-local	0.0.0.0	255.255.0.0	U	0	0	0	ens33
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	ens33

```
user1@myubuntu:~$ netstat -rn
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
0.0.0.0	192.168.1.1	0.0.0.0	UG	0	0	0	ens33
169.254.0.0	0.0.0.0	255.255.0.0	U	0	0	0	ens33
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	ens33

```
user1@myubuntu:~$
```

04 네트워크 상태 확인

■ 현재 열려 있는 포트 확인하기

- 현재 통신이 진행 중인 서비스는 해당 서비스 포트가 LISTEN 상태

```
user1@myubuntu:~$ netstat -an | grep LISTEN
```

```
tcp        0      0 0.0.0.0:22                0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:631            0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:25               0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:5355             0.0.0.0:*                LISTEN
tcp6       0      0 :::22                   :::*                      LISTEN
tcp6       0      0 :::1:631                 :::*                      LISTEN
tcp6       0      0 :::25                    :::*                      LISTEN
tcp6       0      0 :::5355                  :::*                      LISTEN
unix  2      [ ACC ]     STREAM    LISTENING   29361      @/tmp/.ICE-unix/1532
(생략)
```

04 네트워크 상태 확인

■ 현재 열려 있는 포트를 사용 중인 프로세스 확인하기 : -p 옵션

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address      Foreign Address    State
PID/Program name
tcp        0      0 myubuntu:ssh      192.168.1.90:50410 ESTABLISHED
21261/sshd: user1 [
tcp        0      0 192.168.1.10:ssh  192.168.1.90:53957 ESTABLISHED
23563/sshd: user1 [
tcp        0      0 myubuntu:ssh      192.168.1.90:54019 ESTABLISHED
25332/sshd: user1 [
(생략)
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags      Type           State      I-Node  PID/Program name
Path
unix    2      [ ]        DGRAM          29082     1516/systemd
/run/user/1000/systemd/notify
unix    2      [ ]        DGRAM          21811     677/systemd
/run/user/121/systemd/notify
(생략)
```

04 네트워크 상태 확인

■ 인터페이스별 네트워크 통계 정보 확인하기 : -i 옵션

- RX-OK, TX-OK는 정상적으로 주고받은 패킷의 개수
- RX-ERR, RX-DROP, RX-OVR, TX-ERR, TX-DROP, TX-OVR는 송수신 중에 오류가 발생한 패킷의 개수

```
user1@myubuntu:~$ netstat -i
```

```
Kernel Interface table
```

Iface	MTU	RX-OK	RX-ERR	RX-DROP	RX-OVR	TX-OK	TX-ERR	TX-DROP	TX-OVR	Flg
ens33	1500	99040	0	0	0	78359	0	0	0	BMRU
lo	65536	2087	0	0	0	2087	0	0	0	LRU

```
user1@myubuntu:~$
```

04 네트워크 상태 확인

■ 프로토콜별 네트워크 통계 정보 확인하기 : -s 옵션

- 예: IP 프로토콜, ICMP 프로토콜, TCP 프로토콜별로 통계 정보를 출력

```
user1@myubuntu:~$ netstat -s
Ip:
    Forwarding: 2
    63990 total packets received
    174 with invalid addresses
    0 forwarded
    0 incoming packets discarded
    63193 incoming packets delivered
    79144 requests sent out
    20 outgoing packets dropped
    68 dropped because of missing route
Icmp:
    79 ICMP messages received
    1 input ICMP message failed
    ICMP input histogram:
        destination unreachable: 47
        timeout in transit: 15
        redirects: 1
        echo replies: 16
```

(생략)

04 네트워크 상태 확인

■ MAC 주소와 IP 주소 확인하기: arp 명령

arp

- **기능** ARP 캐시 정보를 관리한다.
- **형식** arp [IP 주소]
- **사용 예** arp
arp 192.168.1.1

- arp 명령을 수행하면 현재 같은 네트워크에 연결되어 있는 시스템의 MAC 주소와 IP 주소를 출력

```
user1@myubuntu:~$ arp
```

Address	HWtype	HWaddress	Flags	Mask	Iface
gateway	ether	04:8d:39:33:52:f7	C		ens33
192.168.1.90	ether	b4:b6:76:ab:f7:48	C		ens33
192.168.1.10		(incomplete)			ens33

```
user1@myubuntu:~$
```

- 특정 시스템의 MAC 주소를 확인

```
user1@myubuntu:~$ arp 192.168.1.1
```

Address	HWtype	HWaddress	Flags	Mask	Iface
gateway	ether	04:8d:39:33:52:f7	C		ens33

```
user1@myubuntu:~$
```

■ 패킷 캡처하기 : tcpdump

tcpdump

- **기능** 네트워크상의 트래픽을 덤프한다.
 - **형식** tcpdump [옵션]
 - **옵션**
 - c 패킷 수: 지정한 패킷 수만큼 덤프 받고 종료한다.
 - i 인터페이스명: 특정 인터페이스를 지정한다.
 - n: IP 주소를 호스트명으로 바꾸지 않는다.
 - q: 정보를 간단한 형태로 보여준다.
 - X: 패킷의 내용을 16진수와 ASCII로 출력한다.
 - w 파일명: 덤프한 내용을 지정한 파일에 저장한다.
 - r 파일명: 덤프를 저장한 파일에서 읽어온다.
- host 호스트명 또는 주소: 지정한 호스트가 받거나 보낸 패킷만 덤프한다.
- tcp port 번호: 지정한 포트 번호 패킷만 덤프한다.
- ip: IP 패킷만 덤프한다.
- **사용 예**
 - tcpdump
 - tcpdump -i eth0
 - tcpdump -i eth0 -w DUMP.out
 - tcpdump tcp port 22 and host 192.168.0.7

■ 옵션 없이 사용하는 경우

- 현재 시스템에서 주고받는 모든 패킷을 캡처하여 패킷의 헤더 부분 정보를 출력

```
uuser1@myubuntu:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
15:53:26.893361 IP myubuntu.ssh > 192.168.1.90.54205: Flags [P.], seq
4089455547:4089455675, ack 302215993, win 260, length 128
15:53:26.893738 IP 192.168.1.90.54205 > myubuntu.ssh: Flags [.], ack 128, win 2047,
length 0
15:53:26.893827 IP myubuntu.ssh > 192.168.1.90.54205: Flags [P.], seq 128:192, ack
1, win 260, length 64
15:53:26.893981 IP myubuntu.ssh > 192.168.1.90.54205: Flags [P.], seq 192:320, ack
1, win 260, length 128
15:53:26.894190 IP 192.168.1.90.54205 > myubuntu.ssh: Flags [.], ack 320, win 2053,
length 0
(생략)
```

■ 옵션 없이 사용하는 경우

- Ctrl+c로 종료하면 다음과 같이 캡처한 패킷의 개수를 출력하고 종료

(생략)

```
15:53:27.891413 IP 192.168.1.90.54205 > myubuntu.ssh: Flags [P.], seq 2081:2145,  
ack 148368, win 2047, length 64
```

```
^C
```

```
1001 packets captured
```

```
1012 packets received by filter
```

```
11 packets dropped by kernel
```

```
user1@myubuntu:~$
```

■ 캡처할 패킷 개수 지정하기 : -c 옵션

- 예: 패킷을 3개만 캡처

```
user1@myubuntu:~$ sudo tcpdump -c 3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
15:54:37.038883 IP myubuntu.ssh > 192.168.1.90.54205: Flags [P.], seq
4089604987:4089605115, ack 302218585, win 260, length 128
15:54:37.039293 IP 192.168.1.90.54205 > myubuntu.ssh: Flags [.], ack 128, win 2050,
length 0
15:54:37.039428 IP myubuntu.ssh > 192.168.1.90.54205: Flags [P.], seq 128:192, ack
1, win 260, length 64
3 packets captured
18 packets received by filter
9 packets dropped by kernel
user1@myubuntu:~$
```

04 네트워크 상태 확인

■ 캡처한 패킷 정보를 파일로 저장하기 : -w 옵션

- 예: 패킷 세 개를 캡처하여 dump.out 파일에 저장

```
user1@myubuntu:~$ sudo tcpdump -c 3 -w dump.out
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144
bytes
3 packets captured
5 packets received by filter
0 packets dropped by kernel
user1@myubuntu:~$
```

- 예: cat 명령으로 dump.out 파일을 확인하면 이상한 문자가 출력되므로 -r 옵션으로 내용 확인

```
user1@myubuntu:~$ file dump.out
dump.out: tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture
length 262144)
user1@myubuntu:~$ cat dump.out
A ZK
vvvH
)tEh00000Z 00t000P_<0000451F0}*0mD 00(00a#)S0q0Q0~`x1M00l2*SX
004000A Z00
(생략)
user1@myubuntu:~$
```

■ 캡처한 패킷 파일 읽기 : -r 옵션

- 예: dump.out 파일의 내용 확인

```
user1@myubuntu:~$ sudo tcpdump -r dump.out
reading from file dump.out, link-type EN10MB (Ethernet)
15:54:57.713035 IP myubuntu.ssh > 192.168.1.90.54205: Flags [P.], seq
4089607323:4089607387, ack 302219481, win 260, length 64
15:54:57.713415 IP 192.168.1.90.54205 > myubuntu.ssh: Flags [.], ack 64, win 2047,
length 0
15:54:57.713546 IP myubuntu.ssh > 192.168.1.90.54205: Flags [P.], seq 64:192, ack 1,
win 260, length 128
user1@myubuntu:~$
```

■ 특정 포트로 송수신되는 패킷 캡처하기 : tcp port 옵션

- 예: 192.168.0.17과 주고받는 패킷 중에서 22번 포트에 해당하는 패킷 세 개를 캡처

```
user1@myubuntu:~$ sudo tcpdump -c 3 tcp port 22 and host 192.168.1.90
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
15:58:10.877891 IP myubuntu.ssh > 192.168.1.90.54205: Flags [P.], seq
4089617211:4089617339, ack 302226713, win 260, length 128
15:58:10.878241 IP 192.168.1.90.54205 > myubuntu.ssh: Flags [.], ack 128, win 2050,
length 0
15:58:10.878315 IP myubuntu.ssh > 192.168.1.90.54205: Flags [P.], seq 128:192, ack
1, win 260, length 64
3 packets captured
16 packets received by filter
7 packets dropped by kernel
user1@myubuntu:~$
```

04 네트워크 상태 확인

■ 캡처한 내용을 ASCII로 보기 : -X 옵션

```
user1@myubuntu:~$ sudo tcpdump -Xqr dump.out
reading from file dump.out, link-type EN10MB (Ethernet)
15:54:57.713035 IP myubuntu.ssh > 192.168.1.90.54205: tcp 64
    0x0000:  4510 0068 18f5 4000 4006 9ddd c0a8 0103  E..h..@.@.....
    0x0010:  c0a8 015a 0016 d3bd f3c2 749b 1203 80d9  ...Z.....t....
    0x0020:  5018 0104 8408 0000 5f3c 1795 8ffe e034  P.....<.....4
    0x0030:  3531 aa46 807d d02a 4fa7 6d44 2090 11ae  51.F.}*0.mD....
    0x0040:  288c d461 231b fa5d 53d8 71d3 51d1 7e60  (..a#..]S.q.Q.~`
    0x0050:  0378 31d9 4dc6 ef6c 3202 142a 5358 f3f9  .x1.M..l2..*SX..
    0x0060:  abb0 ac34 b340 80e2                          ...4.@..
```

(생략)