

Chapter 11 네트워크 설정

목차

- 01 네트워크 기초
- 02 네트워크 설정
- 03 호스트 이름 설정
- 04 네트워크 상태 확인

학습목표



- TCP/IP 프로토콜의 계층 구조를 이해하고 설명할 수 있다.
- MAC 주소와 IP 주소의 차이를 설명할 수 있다.
- 라우팅 테이블을 확인하고 기본 게이트웨이를 설정할 수 있다.
- 네트워크를 설정하여 외부 네트워크와 연결할 수 있다.
- 네트워크가 정상적으로 동작하는지 확인할 수 있다.
- 패킷을 캡처하여 저장하고 내용을 분석할 수 있다.

00 Preview

00 Preview

■ 11장의 내용 구성

- 리눅스에서 네트워크를 설정하려면 IP 주소, 넷마스크, 게이트웨이, DNS가 모두 정확하게 설정되어야 함. 이를 설정하는 명령을 익히고 관련된 파일을 살펴볼 것이다
- 네트워크가 설정된 이후에는 정상적으로 동작하는지 주기적으로 확인해야 하고 네트워크 관련 통계를 확인하는 방법도 알아야 함
- 네트워크가 연결되지 않으면 리눅스는 서버 역할을 수행할 수 없음



01 네트워크 기초

01 네트워크 기초

■ TCP/IP 프로토콜

- 프로토콜: 컴퓨터와 컴퓨터 사이에 데이터를 어떻게 주고받을 것인지를 정의한 통신 규약
- 같은 프로토콜을 사용하는 기기 간에는 통신이 가능
- TCP/IP 프로토콜은 5계층으로 구성 되어 있음
- 계층별로 수행하는 역할이 구분되며, 계층별로 다양한 서비스를 제공하는 프로토콜이 지원됨
- TCP/IP 프로토콜은 다양한 프로토콜의 집합이라고 할 수 있는데, 이 중 전송 계층의 TCP와 네트워크 계층의 IP로 전체 프로토콜을 대표하여 TCP/IP 프로토콜이라고 일컬음

응용 계층(application layer)

전송 계층(transport layer)

네트워크 계층(network layer)

링크 계층(link layer)

물리 계층(physical layer)

그림 11-1 TCP/IP 프로토콜 모델

01 네트워크 기초

■ TCP/IP 프로토콜

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

01 네트워크 기초

■ MAC 주소

- MAC은 'media access control'의 약자로, MAC 주소는 하드웨어를 위한 주소이며 다른 말로 이더넷 주소, 하드웨어 주소, 물리 주소라고도 함
- MAC 주소는 각 하드웨어를 구별하는 역할을 수행.
MAC 주소는 네트워크 인터페이스 카드(랜 카드)에 저장된 주소 라고 생각하면 됨
- MAC 주소는 기본적으로 네트워크 인터페이스 카드가 만들어질 때 부여되며 원칙적으로는 수정할 수 없지만, 일부 네트워크 인터페이스 카드의 경우 사용자가 MAC 주소를 수정할 수 있도록 허용
- 특별한 경우가 아니면 MAC 주소는 수정하지 않는 것이 좋음

01 네트워크 기초

■ MAC 주소

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

제조사 번호 일련번호
(IEEE에서 지정) (제조사에서 지정)

그림 11-2 MAC 주소의 예

- MAC 주소는 :이나 -으로 구분되는 여섯 개의 16진수로 구성되며 총 48bit
- 이 중 앞의 세 자리는 제조사 번호이고 뒤의 세 자리는 일련번호
- 예를 들어 [그림 11-2]에서 앞의 세 자리인 00:50:56은 제조사 번호이고, 뒤의 세 자리인 3e:3c:fe는 일련번호. 제조사 번호는 국제 표준 기구 중 하나인 IEEE에서 지정함

01 네트워크 기초

■ 랜 카드 제조사 확인 방법

- MAC 주소로 제조사를 확인하는 방법이 있음. www.coffer.com/mac_find에서 MAC 주소 중 제조사 번호 부분을 검색하면 제조사를 알 수 있음.

예를 들어 00:50:56을 검색하면 vmware, inc.라는 검색 결과가 나옴

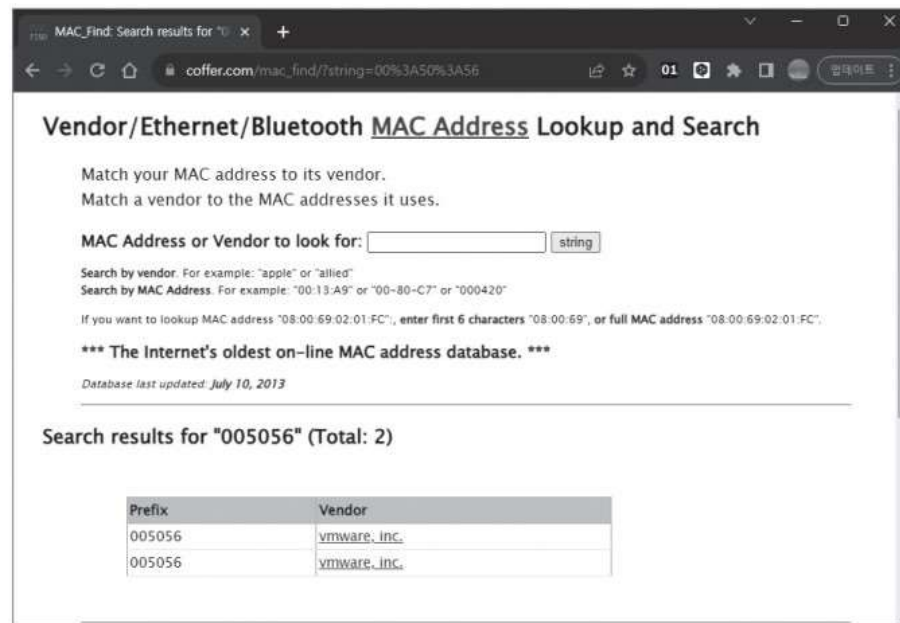


그림 11-3 MAC 주소로 제조사 찾기

01 네트워크 기초

■ IP 주소

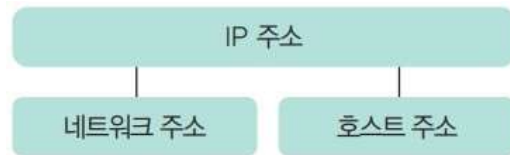


그림 11-4 IP 주소의 구성

- 컴퓨터가 인터넷에 연결하려면 IP 주소가 할당되어 있어야 함. 우리가 보통 인터넷 주소라고 부르는 것이 IPinternet protocol 주소
- IP 주소는 인터넷으로 연결된 네트워크에서 각 컴퓨터를 구분하기 위해 사용. IP 주소는 1바이트 크기의 네 개 숫자로 구성되므로 총 4바이트
- TCP/IP 프로토콜의 3~5계층은 IP 주소를 사용. IP 주소는 네트워크를 구분하는 네트워크 주소 부분과 해당 네트워크 안에서 특정 컴퓨터를 식별하는 호스트 주소 부분으로 나뉨

01 네트워크 기초

■ IP 주소

- IP 주소는 총 32bit(4B) 중 몇 비트를 네트워크 부분으로 사용하고 나머지 몇 비트를 호스트 부분으로 사용하는지에 따라 A 클래스, B 클래스, C 클래스로 구분
- 이 가운데 주로 접하게 되는 C 클래스는 앞의 3바이트가 네트워크 부분이고 뒤의 1바이트만 호스트 부분. 따라서 호스트 부분으로 사용할 수 있는 숫자는 0~255인데, 0은 네트워크 주소를 나타내는 데 사용하고 255는 브로드캐스트 주소로 사용하므로 1~254를 호스트 주소로 할당할 수 있음
- 192.168.100.5와 같은 형식의 IP 주소를 IPv4(IP 버전 4)라고 하는데 이미 이 주소는 고갈되어 더 이상 새로운 주소를 배정받을 수 없음. 이를 대체하기 위해 개발된 주소는 IPv6(IP 버전 6)

01 네트워크 기초

■ 넷마스크와 브로드캐스트 주소

- IP 주소에서 네트워크 부분을 알려주는 역할을 하는 것이 넷마스크. 넷마스크는 하나의 네트워크를 다시 작은 네트워크(서브넷)로 분리할 때도 사용하므로 서브넷 마스크라고 부르기도 함
- C 클래스 IP 주소의 경우 기본 넷마스크가 255.255.255.0

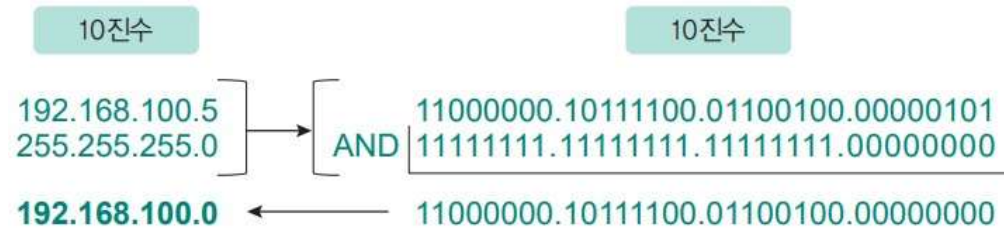


그림 11-5 넷마스크 계산의 예

01 네트워크 기초

■ 넷마스크와 브로드캐스트 주소

- IP 주소와 넷마스크를 10진수에서 2진수로 바꾼 다음, 두 값을 가지고 AND 연산을 수행하면 네트워크 부분만 남고 호스트 부분은 0이 됨
- AND 연산은 $a \text{ AND } b$ 에서 b 가 모두 1이면 a 값이 그대로 남고, b 가 모두 0이면 결과 값은 0이 됨
- 넷마스크는 IP 주소와 AND 연산을 수행하여 네트워크 부분만 남기는 역할을 하는 것
- 브로드캐스트 주소는 같은 네트워크에 있는 모든 컴퓨터에 메시지를 보낼 때 사용하는 것으로 호스트 부분을 모두 1로 설정함

01 네트워크 기초

■ 호스트 이름

- 컴퓨터가 인터넷에 연결하려면 IP 주소가 있어야 함. 이 주소를 사용하여 메일도 보내고 웹 사이트에 접속함
- 컴퓨터는 숫자를 좋아하지만 사람은 숫자보다 이름으로 된 것을 더 잘 기억함. 그래서 등장한 것이 호스트 이름
- 호스트 이름도 IP 주소처럼 두 부분으로 구성됨
- 네이버의 호스트 이름은 www.naver.com
naver.com이 네트워크 부분, www가 호스트 부분에 해당함
- 개인용 PC라면 호스트 이름을 붙일 필요가 없겠지만, 웹 서버와 같이 네트워크 서비스를 제공하는 서버 컴퓨터는 용도에 따라 호스트 이름을 붙여서 사용해야 함

01 네트워크 기초

■ 포트 번호

- 포트 번호는 보통 주소에 포함되지 않지만 각 서비스를 구분하는 번호
- 사용자가 네트워크 서비스를 이용할 때 사용자의 패킷은 IP 주소를 보고 해당 서버 컴퓨터를 찾아 감
- 서버 컴퓨터에 도착한 사용자의 패킷은 어떤 서비스를 요청했는지 확인한 다음 해당 데몬에 패킷을 전달
- 예를 들어 웹 서비스를 요청했으면 웹 서버 데몬(httpd)에 전달하는 것임. 이때 사용자가 어떤 서비스를 요청했는지 구분해 주는 것이 포트 번호다
- 포트 번호는 TCP/IP 프로토콜의 4계층인 전송 계층에서 사용하는 번호
전송 계층이 하는 일이 해당 프로그램에 데이터가 정확하게 전달되도록 하는 것임을 기억하자

01 네트워크 기초

■ 포트 번호

- /etc/services 파일을 기억할 것. 이 파일의 본 역할은 서비스별로 포트 번호가 무엇인지 정의하는 것

```
[user1@localhost ~]$ cat /etc/services
# /etc/services:
# $Id: services,v 1.49 2017/08/18 12:43:23 ovasik Exp $
#
(생략)
# 21 is registered to ftp, but also used by fsp
ftp          21/tcp
ftp          21/udp          fsp fspd
ssh          22/tcp          # The Secure Shell (SSH) Protocol
ssh          22/udp          # The Secure Shell (SSH) Protocol
telnet       23/tcp
telnet       23/udp
(생략)
```

- /etc/services 파일에 저장된 포트 번호는 국제 표준으로 합의하여 사용하고 있는 번호. 사용자가 개발한 네트워크 프로그램은 이 파일에 정의되지 않은 번호를 사용하여 서비스를 제공할 수 있음

02 네트워크 설정

02 네트워크 설정

■ 네트워크 설정

- 네트워크를 사용하기 위해 설정해야 할 주소
 - IP 주소
 - 넷마스크와 브로드캐스트 주소
 - 게이트웨이(라우터) 주소
 - DNS 주소
- 사용자가 임의로 설정하여 사용할 수 있는 것이 아니며, 반드시 해당 기관의 네트워크 관리자에게 문의하여 정확한 주소를 받아야 함. 하나라도 틀리면 네트워크 연결에 문제가 발생할 수 있기 때문

02 네트워크 설정

■ 네트워크 관리자

- 네트워크 관리자가 네트워킹 서비스를 제공
- 네트워크 관리자는 네트워크의 제어와 설정을 관리하는 데몬
- 네트워크 관리자를 사용하여 IP 주소 설정, 고정 라우터 설정, DNS 설정 등을 수행할 수 있음

표 11-2 네트워크 관리 도구

도구	기능
네트워크 관리자	기본 네트워킹 데몬
nmcli 명령	네트워크 관리자를 사용하는 명령 기반 도구
[설정]-[네트워크]	그놈에서 제공하는 GUI 기반 도구
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 실행하기

- 네트워크 관리자는 로키 리눅스를 설치할 때 기본적으로 설치되지만, 설치되지 않은 경우에는 dnf 명령으로 설치. 네트워크 관리자는 시스템이 부팅될 때 자동으로 동작
- 네트워크 관리자가 동작하고 있는지는 systemctl status 명령으로 확인할 수 있음

```
[root@localhost ~]# systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; preset:
enabled)
   Active: active (running) since Sun 2023-09-24 19:30:44 KST; 6 days ago
     Docs: man:NetworkManager(8)
  Main PID: 1131 (NetworkManager)
    Tasks: 3 (limit: 22862)
   Memory: 10.0M
      CPU: 2.386s
   CGroup: /system.slice/NetworkManager.service
           └─1131 /usr/sbin/NetworkManager --no-daemon

(생략)
```

02 네트워크 설정

■ 네트워크 관리자 실행하기

- 만약 네트워크 관리자의 상태가 inactive라면 다음 명령으로 동작 시킴

```
[root@localhost ~]# systemctl start NetworkManager
```

- 시스템이 부팅될 때마다 네트워크 관리자가 동작하게 하려면 다음과 같이 enable 명령을 실행

```
[root@localhost ~]# systemctl enable NetworkManager
```


02 네트워크 설정

■ 네트워크 관리자와 작업하기

- 네트워크 관리자는 네트워크 설정 정보를 연결 프로파일에 저장
- 사용자는 네트워크 관리자를 직접 제어하지 않고 명령 기반 도구나 GUI 기반 도구를 사용
- nmcli는 네트워크 관리자를 사용하는 명령 기반 도구이고, 그놈의 [설정]-[네트워크]나 nmconnection-editor는 GUI 기반 도구
- 이러한 도구를 사용하여 네트워크 설정을 변경하면 네트워크 관리자가 자동으로 인식
- ip 명령으로도 네트워크를 설정할 수 있지만 이 명령으로 네트워크의 설정을 변경하면 네트워크 관리자가 자동으로 인식하지 못함

02 네트워크 설정

■ 그놈의 [설정]-[네트워크]로 설정하기

- 그놈은 윈도의 제어판처럼 시스템과 네트워크 설정을 위한 기능을 제공
- [설정]은 로키 리눅스에서 [현재 활동]-[프로그램 표시]-[설정]을 선택하여 실행하거나, 바탕화면에서 마우스 오른쪽 버튼을 클릭하여 선택
- [설정]에서 [네트워크]를 선택하면 네트워크 설정 창이 뜬. 네트워크 설정 창에서 유선의 우측에 있는  을 선택하면 오른쪽과 같은 유선

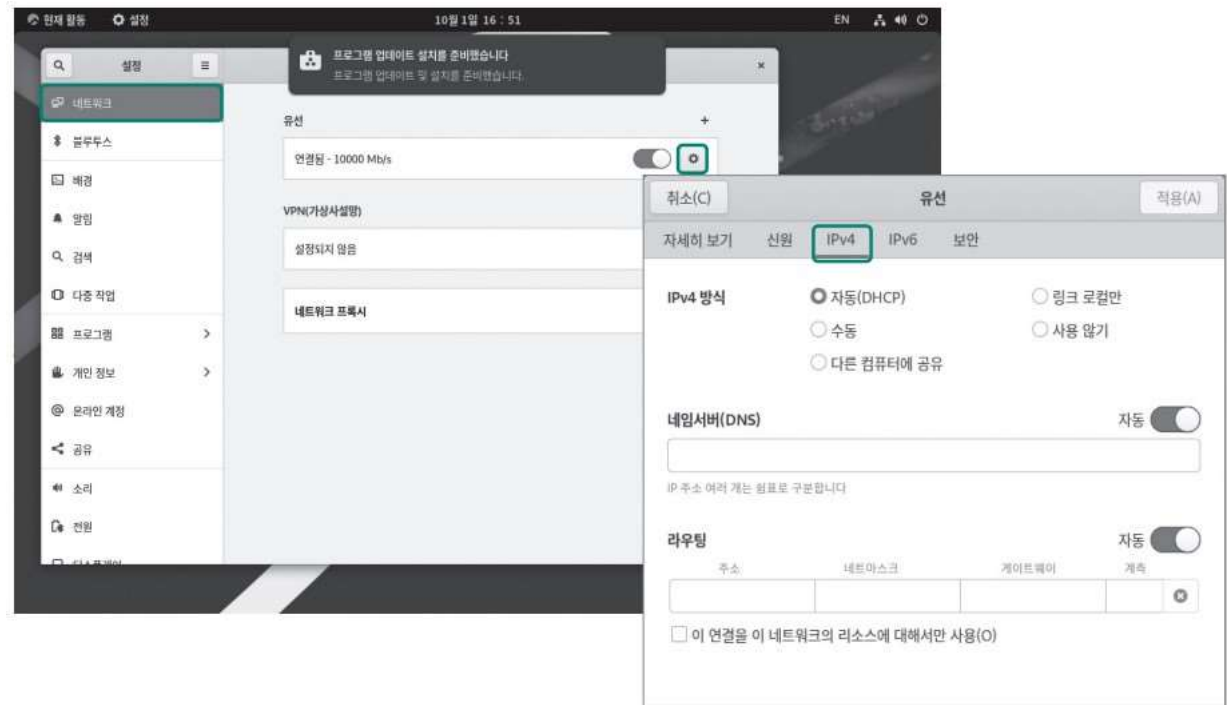



그림 11-6 그놈의 네트워크 설정 창과 유선 네트워크 설정 창

네트워크 설정 창이 뜨고, IPv4 메뉴에서 IP 주소와 네임서버(DNS), 라우팅 정보를 설정할 수 있음

02 네트워크 설정

■ nm-connection-editor로 설정하기

- nm-connection-editor는 네트워크 관리자와 함께 설치되며, 터미널에서 실행하면 [그림 11-7]의 왼쪽과 같은 창이 뜬다
- 이 창에서 설정할 장치명(ens160)을 선택하고 하단의  을 클릭하면 오른쪽과 같은 편집창이 뜬다. 이 편집 창에서 IPv4뿐 아니라 이더넷, 802.1x 보안, IPv6, DCB, Proxy 등 네트워크와 관련된 다양한 기능을 설정할 수 있다

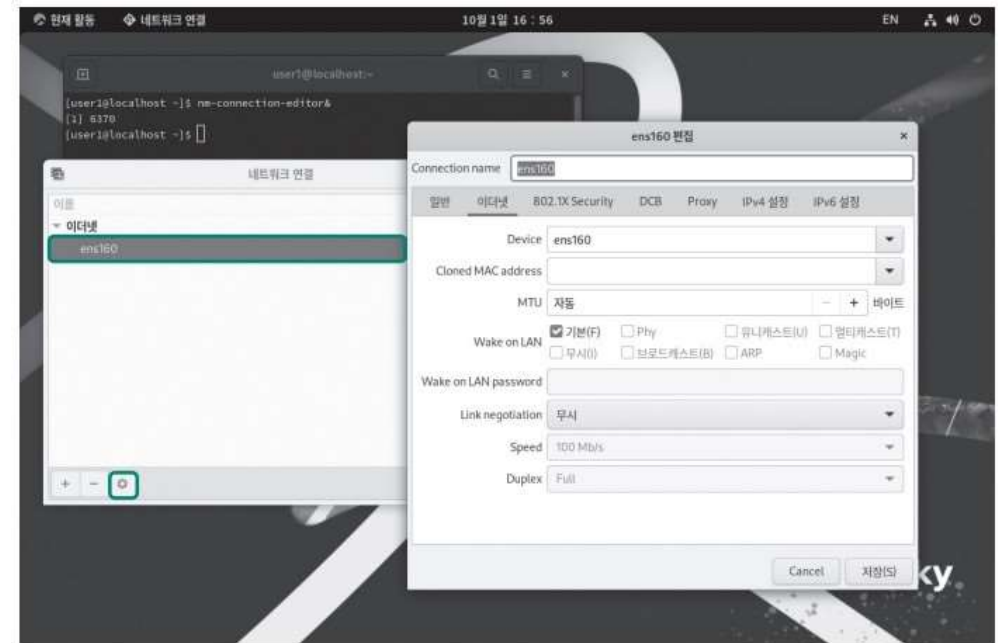


그림 11-7 nm-connection-editor 창과 편집 창

02 네트워크 설정

■ nmcli 명령으로 네트워크 설정

- 네트워크를 설정하는 명령은 네트워크 관리자와 함께 설치되는 nmcli 명령
- nmcli 명령으로 유선 네트워크뿐 아니라 와이파이 등의 무선 네트워크, 보안 등 네트워크와 관련된 거의 모든 설정을 관리할 수 있음
- nmcli는 명령 행에서 사용하는 명령은 물론이고 대화식 인터페이스도 제공

02 네트워크 설정

■ nmcli 명령으로 네트워크 설정

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

02 네트워크 설정

■ 네트워크의 전체 상태 살펴보기: general(gen) 명령

- 네트워크의 전체적인 상태는 nmcli의 general 명령으로 확인할 수 있음
- nmcli를 사용할 때 명령을 줄여서 사용할 수도 있음
- 예를 들어 general 대신에 gen만 입력해도 됨. general 명령의 서브 명령인 status가 없어도 같은 결과를 출력. 출력 결과를 보면 네트워크가 연결되어 있고 와이파이와 인터넷(WWAN)을 사용한다는 것을 알 수 있음

```
[root@localhost ~]# nmcli general status
```

STATE	CONNECTIVITY	WIFI-HW	WIFI	WWAN-HW	WWAN
연결됨	전체	missing	사용	missing	사용

```
[root@localhost ~]# nmcli gen
```

STATE	CONNECTIVITY	WIFI-HW	WIFI	WWAN-HW	WWAN
연결됨	전체	missing	사용	missing	사용

02 네트워크 설정

■ 네트워크를 활성화하거나 비활성화하기: `networking(net)` 명령

- `networking` 명령은 네트워크를 활성화on하거나 비활성화off
- `connectivity` 서브 명령으로 네트워크의 연결 상태를 알려줌
- `connectivity`가 출력하는 네트워크 상태는 다음 중 하나다
 - `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.

02 네트워크 설정

■ 네트워크를 활성화하거나 비활성화하기: networking(net) 명령

- 다음 예와 같이 nmcli net off를 실행하면 네트워크의 연결이 비활성화됨
- nmcli net on 으로 다시 네트워크를 활성화

```
[root@localhost ~]# nmcli net con
full
[root@localhost ~]# nmcli net off
user1@localhost ~]$ nmcli net con
none
[user1@localhost ~]$ nmcli net on
[user1@localhost ~]$ nmcli net con
full
```

02 네트워크 설정

■ 네트워크 설정하기: connetion(con) 명령

- connection은 네트워크 설정과 관련된 대부분의 기능을 수행

표 11-3 connection의 서브 명령

서브 명령	기능
show	메모리와 디스크에 저장된 네트워크 연결 프로파일을 출력한다. 서브 명령을 지정하지 않으면 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.

02 네트워크 설정

■ 네트워크 연결 프로파일 출력하기: show

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

```
[root@localhost ~]# nmcli con show
```

NAME	UUID	TYPE	DEVICE
ens160	94b2a0a1-f941-3654-b41e-0ece74decf48	ethernet	ens160
lo	8563b4bc-fd03-4426-9107-91b3a42f206c	loopback	lo

- 다음 예를 보면 연결 프로파일의 이름이 ens160이고, UUID가 94b2a0a1-f941-3654-b41e-0ece74decf48이며, 네트워크 유형은 이더넷, 연결된 장치의 이름은 ens160임을 알 수 있음
- 출력 내용 중 이름은 연결 프로파일에 사용자가 지정한 ID이며, UUID는 시스템이 지정하는 것으로 둘 다 연결을 구분하는 역할을 함. 여기서는 이름이 장치명과 같게 출력되었지만 장치명을 의미하는 것은 아님
- 출력 결과를 보면 ens160과 lo가 출력됨. lo는 로컬 루프백으로 시스템 내부 통신용으로 사용
- ens160이 실제로 외부와 통신할 때 사용되는 네트워크 인터페이스의 명칭

02 네트워크 설정

■ 네트워크 연결 중지하기: down

- nmcli connection down 명령은 네트워크 연결을 중지함
- 네트워크의 연결을 중지한 다음 show 명령으로 확인하면 ens160과 연결된 장치가 없다고 나옴

```
user1@localhost ~]$ nmcli con down ens160
'ens160' 연결이 성공적으로 비활성화되었습니다 (D-Bus 활성 경로: /org/freedesktop/NetworkManager/ActiveConnection/4)
[user1@localhost ~]$ nmcli con show
```

NAME	UUID	TYPE	DEVICE
lo	8563b4bc-fd03-4426-9107-91b3a42f206c	loopback	lo
ens160	94b2a0a1-f941-3654-b41e-0ece74decf48	ethernet	--

02 네트워크 설정

■ 네트워크 연결 시작하기: up

- nmcli connection up 명령은 네트워크 연결을 시작
- 네트워크 연결을 시작한 후 show 명령으로 확인하면 ens160 장치가 다시 연결되었음을 알 수 있음

```
[user1@localhost ~]$ nmcli con up ens160
연결이 성공적으로 활성화되었습니다 (D-버스 활성 경로: /org/freedesktop/NetworkManager/
ActiveConnection/5)
[user1@localhost ~]$ nmcli con show
```

NAME	UUID	TYPE	DEVICE
ens160	94b2a0a1-f941-3654-b41e-0ece74decf48	ethernet	ens160
lo	8563b4bc-fd03-4426-9107-91b3a42f206c	loopback	lo

02 네트워크 설정

■ 네트워크 연결 추가하기: add

- nmcli connection add 명령은 네트워크 연결을 추가함
- 예를 들어 IPv4 이더넷 연결을 추가하려면 다음과 같은 형식으로 실행함

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

- 명령의 형식에서 볼드체 부분을 사용자가 지정해야 함
- 예를 들어 고정 IP를 사용할 경우 다음과 같이 지정할 수 있음

```
[root@localhost ~]# nmcli con add type ethernet con-name test-net ifname ens160  
ip4 192.168.147.130/24 gw4 192.168.147.1  
'test-net' (8007ec89-92f3-45b8-a3ef-a1677e1c8ef4) 연결이 성공적으로 추가되었습니다.
```

02 네트워크 설정

■ 네트워크 연결 추가하기: add

- 연결 프로파일 이름(connection name): test-net
- 네트워크 장치명(ifname): ens160
- IPv4 주소: 192.168.147.130/24 (/24는 넷마스크를 표시: 255.255.255.0)
- 게이트웨이 주소(gw4): 192.168.147.1/24 (/24는 넷마스크를 표시: 255.255.255.0)
- show 명령으로 확인해 보면 test-net으로 연결이 추가되었음을 알 수 있음

```
root@localhost ~]# nmcli con show
```

NAME	UUID	TYPE	DEVICE
ens160	94b2a0a1-f941-3654-b41e-0ece74decf48	ethernet	ens160
lo	8563b4bc-fd03-4426-9107-91b3a42f206c	loopback	lo
test-net	8007ec89-92f3-45b8-a3ef-a1677e1c8ef4	ethernet	--

02 네트워크 설정

■ 네트워크 연결 추가하기: add

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

```
[root@localhost ~]# nmcli con up test-net ifname ens160
연결이 성공적으로 활성화되었습니다 (D-버스 활성 경로: /org/freedesktop/NetworkManager/ActiveConnection/6)
```

- show 명령으로 확인해 보면 test-net 연결에 장치명이 할당되었음
- ifconfig 명령으로 IPv4 주소를 확인해 보면 192.168.147.130으로 바뀐 것을 알 수 있음

```
root@localhost ~]# nmcli con show
NAME      UUID                                  TYPE      DEVICE
test-net  8007ec89-92f3-45b8-a3ef-a1677e1c8ef4  ethernet  ens160
lo        8563b4bc-fd03-4426-9107-91b3a42f206c  loopback  lo
ens160    94b2a0a1-f941-3654-b41e-0ece74decf48  ethernet  --
[root@localhost ~]# ifconfig ens160
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.147.130  netmask 255.255.255.0  broadcast 192.168.147.255
(생략)
```

02 네트워크 설정

■ 네트워크 연결 추가하기: add

- 만약 동적 IP 연결을 추가하려면 다음과 같은 형식을 사용
- IP 주소가 동적으로 할당 될 것이므로 IP 주소를 지정하는 부분이 없음
- 고정 IP 연결과 마찬가지로 연결을 추가한 후 up 명령으로 연결을 시작해야 함

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

■ 네트워크 연결 수정하기: modify(mod)

- nmcli connection modify 명령은 기존 연결 프로파일을 수정함

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

02 네트워크 설정

■ 네트워크 연결 수정하기: modify(mod)

- modify 명령에서 사용할 수 있는 setting과 property는 네트워크 관리자의 연결 프로파일에 사용하는 설정과 속성으로 매우 다양한 값을 가지고 있음

표 11-4 설정과 속성의 예

설정(setting)	속성(property)	값의 유형	기능
connection	autoconnection	boolean (TRUE/FALSE)	자원이 사용 가능해지면 네트워크 관리자가 자동으로 연결할지를 지정한다.
	id	문자열	사용자가 읽을 수 있는 연결의 이름
	interface-name	문자열	네트워크 장치의 이름
	type	문자열	연결의 유형
ipv4	addresses	주소	IP 주소
	dns	주소	DNS 서버의 IP 주소
	gateway	주소	게이트웨이 주소
	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").

02 네트워크 설정

■ 네트워크 연결 수정하기: modify(mod)

- 예를 들어, 기존 test-net 연결의 IPv4 주소를 변경하려면 다음과 같이 지정

```
nmcli con mod test-net ipv4.addresses 192.168.147.131
```

- 다른 IP 주소를 추가하려면 + 기호를 사용하고, 주소를 제거하려면 - 기호를 사용함
다음은 주소를 추가한 예

```
nmcli con mod test-net +ipv4.addresses 192.168.147.132
```

- 게이트웨이를 수정하려면 다음과 같이 함

```
nmcli con mod test-net ipv4.gateway 192.168.147.254
```

- 특정 네트워크로 가는 경로를 지정하려면 다음과 같이 함

```
nmcli con mod test-net +ipv4.routes "192.168.2.0/24 192.168.147.1"
```

- 연결 프로파일의 내용을 수정하면 up 명령으로 다시 적용해야 함

```
nmcli con up test-net ifname ens160
```

02 네트워크 설정

■ 네트워크 연결 삭제하기: delete(del)

- nmcli connection delete 명령은 연결 프로파일을 삭제

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

- 예를 들어 연결 프로파일 test-net을 삭제하려면 다음과 같이 함

```
[root@localhost ~]# nmcli con delete test-net
'test-net' (8007ec89-92f3-45b8-a3ef-a1677e1c8ef4) 연결이 성공적으로 삭제되었습니다.
[root@localhost ~]# nmcli con show
```

NAME	UUID	TYPE	DEVICE
ens160	94b2a0a1-f941-3654-b41e-0ece74decf48	ethernet	ens160
lo	8563b4bc-fd03-4426-9107-91b3a42f206c	loopback	lo

02 네트워크 설정

■ 네트워크 연결 프로파일 읽어오기: reload, load

- 네트워크 관리자는 연결 프로파일이 수정되었는지 자동으로 인식하지 않음
- reload는 디스크에서 모든 연결 프로파일을 다시 읽어오고, load는 특정 연결 프로파일을 지정하여 읽어 옴
- 연결 프로파일을 수작업으로 수정했다면 네트워크 관리자에게 이를 알려주기 위해 reload나 load 명령을 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 관리함
- device 명령의 서브 명령에는 여러 가지가 있지만 status와 show만 살펴봄

■ 네트워크 장치의 상태 보기: status

- status 명령은 네트워크 장치의 상태를 요약해서 출력

```
[root@localhost ~]# nmcli dev status
```

DEVICE	TYPE	STATE	CONNECTION
ens160	ethernet	연결됨	ens160
lo	loopback	연결됨 (외부)	lo

02 네트워크 설정

■ 네트워크 장치의 상세한 정보 보기: show

- show 명령은 네트워크 장치의 상세한 정보를 출력
- show 명령 다음에 장치명을 지정하지 않으면 전체 장치에 대한 상세 정보가 출력되고, 장치명을 지정하면 해당 장치의 상세 정보만 출력

```
[root@localhost ~]# nmcli dev show
GENERAL.DEVICE:           ens160
GENERAL.TYPE:             ethernet
GENERAL.HWADDR:           00:0C:29:C5:25:AB
GENERAL.MTU:              1500
GENERAL.STATE:            100 (연결됨)
GENERAL.CONNECTION:       ens160
GENERAL.CON-PATH:         /org/freedesktop/NetworkManager/ActiveConnection/7
WIRED-PROPERTIES.CARRIER: 켜짐
IP4.ADDRESS[1]:           192.168.147.129/24
IP4.GATEWAY:              192.168.147.2
(생략)
```

02 네트워크 설정

■ ip 명령으로 네트워크 설정

- 네트워크 설정은 ip 명령으로도 가능. ip 명령을 사용하여 네트워크 상태 확인, IP 주소 설정, 게이트웨이 설정을 할 수 있는데, 이처럼 ip 명령으로 설정한 것은 시스템을 재시작하면 사라짐
- 시스템을 다시 시작한 후에도 설정한 내용이 적용되게 하려면 이를 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

02 네트워크 설정

■ 네트워크 장치의 주소 관리하기: address(addr) 명령

- address 명령은 IP 주소의 정보를 출력하거나 설정하고 삭제함

■ 네트워크 장치의 정보 보기: show

- show 명령은 네트워크 장치의 정보를 출력
- show 명령 다음에 장치명을 지정하지 않으면 전체 장치에 대한 상세 정보가 출력되고, 특정 장치를 지정하면 해당 장치의 정보만 출력됨

02 네트워크 설정

■ 네트워크 장치의 정보 보기: show

```
[root@localhost ~]# 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: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 00:0c:29:c5:25:ab brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.147.129/24 brd 192.168.147.255 scope global dynamic noprefixroute
ens160
        valid_lft 1384sec preferred_lft 1384sec
    inet6 fe80::20c:29ff:fec5:25ab/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

- 다음은 전체 장치의 정보를 출력한 예
- show 명령 없이 ip addr만 실행해도 같은 결과가 출력 됨
- link/ether는 이더넷 주소, inet은 IPv4 주소, inet6는 IPv6 주소

02 네트워크 설정

■ IP 주소 설정하기: add

- add 명령은 네트워크 장치에 IP 주소를 설정. ens160 장치에 ip 명령으로 고정 IP 주소를 설정하려면 다음과 같이 작성. 설정된 주소를 확인하면 ens160 장치에 IPv4 주소가 두 개 설정되어 있음

```
[root@localhost ~]# ip addr add 192.168.147.130/24 dev ens160
[root@localhost ~]# ip addr show ens160

2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 00:0c:29:c5:25:ab brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.147.129/24 brd 192.168.147.255 scope global dynamic noprefixroute
ens160
        valid_lft 1649sec preferred_lft 1649sec
    inet 192.168.147.130/24 scope global secondary ens160
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fec5:25ab/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

02 네트워크 설정

■ IP 주소 삭제하기: del

- del 명령은 네트워크 장치에 설정된 IP 주소를 삭제
- ens160 장치에 설정된 192.168.47.130 주소를 삭제하려면 다음과 같이 작성
- show 명령으로 주소가 삭제 된 것을 확인할 수 있음

```
[root@localhost ~]# ip addr del 192.168.147.130/24 dev ens160
[root@localhost ~]# ip addr show ens160
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 00:0c:29:c5:25:ab brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.147.129/24 brd 192.168.147.255 scope global dynamic noprefixroute
ens160
        valid_lft 1560sec preferred_lft 1560sec
    inet6 fe80::20c:29ff:fec5:25ab/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

02 네트워크 설정

■ 라우팅 테이블과 게이트웨이 주소 관리하기: route 명령

- ip route 명령은 라우팅 테이블을 출력하거나 게이트웨이를 설정하고 삭제
- 인터넷은 네트워크와 네트워크를 연결한 것이라고 할 수 있음
- 네트워크를 다른 네트워크와 연결할 때 연결점이 되는 장치를 게이트웨이라고 함
게이트웨이도 하나의 컴퓨터로 보통 라우터라고 부름
- 게이트웨이는 패킷을 보고 같은 네트워크로 보내는 것이 아니면 외부로 전송함. 우체부 역할
- 게이트 웨이 주소가 설정되어 있지 않으면 같은 네트워크가 아닌 컴퓨터와는 접속할 수가 없음
- 같은 네트워크 간에는 통신이 되는데 외부와 연결이 안 된다면 게이트웨이 주소를 올바르게 설정했는지 확인해야 함

02 네트워크 설정

■ 라우팅 테이블 보기: show

- ip route show 명령은 현재 설정된 라우팅 테이블을 출력
- 라우팅 테이블은 게이트웨이 정보를 가지고 있음
- 다음 예를 보면 기본 게이트웨이 주소가 192.168.147.2이며, ens160 장치를 통해 접속한다는 것을 알 수 있음

```
[root@localhost ~]# ip route show
default via 192.168.147.2 dev ens160 proto dhcp src 192.168.147.129 metric 100
192.168.147.0/24 dev ens160 proto kernel scope link src 192.168.147.129 metric 100
```

02 네트워크 설정

■ 기본 게이트웨이 주소 설정하기: add default

- ip route add 명령은 게이트웨이를 추가함. 기본 게이트웨이 설정은 다음과 같이 함

```
[root@localhost ~]# ip route add default via 192.168.147.1 dev ens160
```

■ 라우팅 경로 설정하기: add

- 기본 게이트웨이 외에 경로를 추가하려면 다음과 같이 함

다음 예는 192.168.2.0 네트워크를 192.168.147.1을 통해서 접속한다는 의미

```
[root@localhost ~]# ip route add 192.168.2.0/24 via 192.168.147.1 dev ens160
[root@localhost ~]# ip route show
default via 192.168.147.1 dev ens160
default via 192.168.147.2 dev ens160 proto dhcp src 192.168.147.129 metric 100
192.168.2.0/24 via 192.168.147.1 dev ens160
192.168.147.0/24 dev ens160 proto kernel scope link src 192.168.147.129 metric 100
```

02 네트워크 설정

■ 라우팅 경로 삭제하기: del

- 라우팅 테이블에서 경로를 삭제하려면 ip route del 명령을 사용

```
[root@localhost ~]# ip route del 192.168.2.0/24
[root@localhost ~]# ip route show
default via 192.168.147.1 dev ens160
default via 192.168.147.2 dev ens160 proto dhcp src 192.168.147.129 metric 100
192.168.147.0/24 dev ens160 proto kernel scope link src 192.168.147.129 metric 100
```

02 네트워크 설정

■ 네트워크 인터페이스를 활성화하거나 비활성화하기: link set 명령

- 네트워크 인터페이스를 활성화 또는 비활성화하려면 ip link set 명령을 사용

■ 네트워크 인터페이스 비활성화하기: down

- 네트워크 인터페이스 비활성화는 다음과 같이 함
- 네트워크를 비활성화하면 state가 DOWN이 된다

```
[root@localhost ~]# ip link set ens160 down
[root@localhost ~]# ip addr show ens160
2: ens160: <BROADCAST,MULTICAST> mtu 1500 qdisc mq state DOWN group default qlen
1000
    link/ether 00:0c:29:c5:25:ab brd ff:ff:ff:ff:ff:ff
    altname enp3s0
```

02 네트워크 설정

■ 네트워크 인터페이스 활성화하기: up

- 네트워크 인터페이스 활성화는 다음과 같이 함
- 네트워크를 활성화하면 state가 UP이 되고 IP 주소도 할당됨

```
[root@localhost ~]# ip link set ens160 up
[root@localhost ~]# ip addr show ens160
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 00:0c:29:c5:25:ab brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.147.129/24 brd 192.168.147.255 scope global dynamic noprefixroute
ens160
        valid_lft 1798sec preferred_lft 1798sec
    inet6 fe80::20c:29ff:fec5:25ab/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```


02 네트워크 설정

■ ifconfig 명령으로 네트워크 설정

- 네트워크 인터페이스를 설정하는 전통적인 명령은 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 ens160`
 - `ifconfig ens160 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255`

02 네트워크 설정

■ 현재 설치된 네트워크 인터페이스 설정 보기

```
[root@localhost ~]# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.147.129 netmask 255.255.255.0 broadcast 192.168.147.255
    inet6 fe80::20c:29ff:fec5:25ab prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:c5:25:ab txqueuelen 1000 (Ethernet)
    RX packets 406 bytes 38066 (37.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 274 bytes 34428 (33.6 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 86 bytes 9888 (9.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 86 bytes 9888 (9.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- 옵션 없이 ifconfig 명령을 사용하면 현재 설치된 네트워크 인터페이스의 설정 내용을 출력
- IP 주소와 넷마스크, 브로드캐스트 주소는 사용자의 네트워크 환경에 따라 다르게 출력

02 네트워크 설정

■ 현재 설치된 네트워크 인터페이스 설정 보기

- 보통 시스템에서 네트워크 인터페이스는 하나이지만 경우에 따라 두 개 이상 장착할 수도 있음
- 위의 예를 보면 네트워크가 다음과 같이 설정되었음을 알 수 있음
 - MAC 주소(ether): 00:0c:29:c5:25:ab
 - IP 주소(inet): 192.168.147.129
 - 넷마스크(netmask): 255.255.255.0
 - 브로드캐스트 주소(broadcast): 192.168.147.255
 - IPv6 주소(inet6): fe80::20c:29ff:fec5:25ab
- RX는 부팅 후 현재까지 받은 패킷 수와 바이트 수를 알려주며, TX는 부팅 후 현재까지 보낸 패킷 수와 바이트 수를 알려줌

02 네트워크 설정

■ 특정 네트워크 인터페이스 설정 보기

- ifconfig 명령 다음에 네트워크 인터페이스의 이름을 지정하면 해당 인터페이스의 설정 내용만 출력

```
[root@localhost ~]# ifconfig ens160
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.147.129 netmask 255.255.255.0 broadcast 192.168.147.255
    inet6 fe80::20c:29ff:fec5:25ab prefixlen 64 scopeid 0x20<link>
(생략)
```

02 네트워크 설정

■ 네트워크 인터페이스 사용 해제하기: down 옵션

- 네트워크 인터페이스를 사용하지 않으려면 다음과 같이 down 옵션을 사용하여 비활성화

```
[root@localhost ~]# ifconfig ens160 down
[root@localhost ~]# ifconfig ens160
ens160: flags=4098<BROADCAST,MULTICAST> mtu 1500
        ether 00:0c:29:c5:25:ab txqueuelen 1000 (Ethernet)
(생략)
```

- 인터페이스가 다운되었을 때 flags를 보면 UP과 RUNNING이 출력되지 않음.
네트워크 연결이 끊어진 것

02 네트워크 설정

■ 네트워크 인터페이스 활성화하기: up 옵션

- 네트워크 인터페이스를 비활성화 상태에서 다시 활성화하려면 up 옵션을 사용

```
[root@localhost ~]# ifconfig ens160 up
[root@localhost ~]# ifconfig ens160
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.147.129 netmask 255.255.255.0 broadcast 192.168.147.255
(생략)
```

- 인터페이스를 up 옵션으로 활성화하면 flags에 UP과 RUNNING이 표시
- 현재 시스템이 동적으로 IP 주소를 받아서 사용하고 있는 경우라면 자동으로 IP 주소가 설정될 것
- 동적 IP를 사용하고 있지 않다면 관리자가 수작업으로 인터페이스를 설정해야 함

02 네트워크 설정

■ 네트워크 인터페이스 설정하기

- 네트워크 인터페이스에 IP 주소를 수작업으로 설정하려면 IP 주소와 넷마스크를 함께 설정해야 함
- 만약 서브넷으로 나누지 않고 기본 C 클래스를 사용한다면 넷마스크나 브로드캐스트 주소를 생략할 수도 있음

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

- ifconfig 명령으로 ip 주소를 다른 것으로 설정하면 게이트웨이 경로 등 네트워크 정보를 잃어버릴 수 있으므로 ens160의 네트워크 설정을 바꾸는 실습은 생략

02 네트워크 설정

■ 게이트웨이 설정하기

- 게이트웨이의 설정과 확인은 route 명령으로 할 수 있음
- route는 게이트웨이의 연결 정보를 관리하는 라우팅 테이블을 편집하는 명령
- 라우팅 테이블이 제대로 설정되어 있지 않으면 외부 네트워크와 연결할 수 없음

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 192.168.147.1 dev ens160

02 네트워크 설정

■ 게이트웨이 설정하기

- route 명령으로 라우팅 테이블을 편집할 때 주로 사용하는 형식

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 ens160
라우팅 경로 추가(호스트)	route add -host 호스트 주소 dev 인터페이스명 route add -host 192.168.1.5 dev ens160
라우팅 경로 제거(네트워크)	route del -net 네트워크 주소 netmask 넷마스크 [dev 인터페이스명] route del -net 192.168.1.0 netmask 255.255.255.0
라우팅 경로 제거(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 192.168.147.1 dev ens160
기본 게이트웨이 제거	route del default gw 게이트웨이 주소 route del default gw 192.168.147.1
루프백(lo) 추가	route add -net 127.0.0.0 netmask 255.0.0.0 dev lo

02 네트워크 설정

■ 라우팅 테이블 보기: route

- route 명령만 사용하면 현재 라우팅 테이블을 출력

```
[root@localhost ~]# route
Kernel IP routing table
Destination    Gateway      Genmask      Flags    Metric    Ref    Use    Iface
default        _gateway    0.0.0.0      UG       100       0       0     ens160
192.168.147.0  0.0.0.0     255.255.255.0 U        100       0       0     ens160
```

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

- 기본 게이트웨이를 설정하려면 게이트웨이 주소를 알아야 함
- 여기서는 192.168.147.2를 예로 사용

```
[root@localhost ~]# route add default gw 192.168.147.2 dev ens160
[root@localhost ~]# route
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	_gateway	0.0.0.0	UG	0	0	0	ens160
default	_gateway	0.0.0.0	UG	100	0	0	ens160
192.168.147.0	0.0.0.0	255.255.255.0	U	100	0	0	ens160

02 네트워크 설정

■ 기본 게이트웨이 삭제하기: del

- 기본 게이트웨이 삭제는 다음과 같이 함
- 만약 DHCP를 사용하는 실습 환경이 아니라면 기본 게이트웨이를 삭제하지 않도록 함
게이트웨이가 설정되어 있지 않으면 네트워크를 사용할 수 없기 때문

```
[root@localhost ~]# route del default gw 192.168.147.2
[root@localhost ~]# route
Kernel IP routing table
Destination    Gateway         Genmask         Flags   Metric      Ref    Use    Iface
default        _gateway       0.0.0.0         UG      100         0      0      ens160
192.168.147.0  0.0.0.0        255.255.255.0   U        100         0      0      ens160
```

02 네트워크 설정

■ DNS 설정

- 네트워크를 설정하기 위해 끝으로 알아야 할 것은 DNS 주소
- DNS가 이름을 주소로 변환하는 역할을 담당
- DNS는 'domain name service'의 약자로 호스트명을 IP 주소로 바꾸는 역할을 수행
- 만약 DNS가 설정되어 있지 않으면 이름으로 서버에 접속할 수 없으며 직접 IP 주소를 사용하여 접속해야 함

02 네트워크 설정

■ DNS 서버 지정하기

- 리눅스는 DNS 서버의 주소를 /etc/resolv.conf 파일에 저장
- /etc/resolv.conf 파일의 내용을 보면 다음과 같음. 서버의 IP 주소는 다를 수 있음
- resolv.conf 파일은 키워드 nameserver 다음에 DNS 서버의 IP 주소를 지정
- 네임 서버는 DNS 서버를 뜻하며 첫 번째 서버가 동작하지 않으면 두 번째 서버로 연결

```
[root@localhost ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search localdomain
nameserver 192.168.147.2
```

02 네트워크 설정

■ nmcli 명령으로 DNS 설정하기

- 예전에는 DNS 서버의 주소가 담긴 /etc/resolv.conf 파일을 수작업으로 편집했으나 요즘에는 명령을 사용하여 편집. nmcli 명령으로 DNS를 지정하는 방법은 다음과 같음

```
nmcli con mod connection-name ipv4.dns DNS주소
```

- 연결 프로파일 ens160에 구글의 DNS 서버 주소인 8.8.8.8과 8.8.4.4를 지정하려면 다음과 같이 함

```
[root@localhost ~]# nmcli con mod ens160 ipv4.dns "8.8.8.8 8.8.4.4"
[root@localhost ~]# nmcli con up ens160
연결이 성공적으로 활성화되었습니다 (D-버스 활성 경로: /org/freedesktop/NetworkManager/
ActiveConnection/3)
[root@localhost ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search localdomain
nameserver 8.8.8.8
nameserver 8.8.4.4
nameserver 192.168.147.2
```


02 네트워크 설정

■ DNS 서버에 질의하기

- DNS 서버에 특정 도메인의 정보를 질의하는 명령어는 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
- nslookup 명령을 실행하면 프롬프트가 >로 바뀜
- 이 상태에서 알고 싶은 도메인명을 입력하면 해당 도메인의 IP 주소가 출력됨

02 네트워크 설정

■ DNS 서버에 질의하기

```
[root@localhost ~]# nslookup
> www.hanbit.co.kr
Server:      8.8.8.8
Address:     8.8.8.8#53

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

[root@localhost ~]#
```

- 다음 예를 보면 www.hanbit.co.kr의 IP 주소는 218.38.58.195임을 알 수 있음
- nslookup 명령은 exit로 종료

02 네트워크 설정

■ 네트워크 설정하기

① 현재 네트워크의 연결 상태를 확인

```
[root@localhost ~]# nmcli gen status
```

STATE	CONNECTIVITY	WIFI-HW	WIFI	WWAN-HW	WWAN
연결됨	전체	missing	사용	missing	사용

② 새로운 연결 프로파일 follow-me를 생성

- 여기서는 IP 주소를 192.168.147.131로 설정
- 실습할 때는 실습 환경의 네트워크에서 사용하지 않는 IP 주소를 지정해야 함
- 게이트웨이 주소는 192.168.147.2

```
[root@localhost ~]# nmcli con add type ethernet con-name follow-me ifname ens160 ip4  
192.168.147.131 gw4 192.168.147.2  
'follow-me' (d7ec74ed-7a1b-4b76-9b63-7f813596fb52) 연결이 성공적으로 추가되었습니다.
```

02 네트워크 설정

■ 네트워크 설정하기

③ 연결 프로파일을 확인

```
[root@localhost ~]# nmcli con show
```

NAME	UUID	TYPE	DEVICE
ens160	94b2a0a1-f941-3654-b41e-0ece74decf48	ethernet	ens160
lo	2b8ed1a0-3363-4beb-8ea5-df4a0ef8329d	loopback	lo
follow-me	d7ec74ed-7a1b-4b76-9b63-7f813596fb52	ethernet	--

④ 연결 프로파일 follow-me를 적용

```
[root@localhost ~]# nmcli con up follow-me
```

연결이 성공적으로 활성화되었습니다 (D-버스 활성 경로: /org/freedesktop/NetworkManager/ActiveConnection/5)

```
[root@localhost ~]# nmcli con show
```

NAME	UUID	TYPE	DEVICE
follow-me	d7ec74ed-7a1b-4b76-9b63-7f813596fb52	ethernet	ens160
lo	2b8ed1a0-3363-4beb-8ea5-df4a0ef8329d	loopback	lo
ens160	94b2a0a1-f941-3654-b41e-0ece74decf48	ethernet	--

02 네트워크 설정

■ 네트워크 설정하기

⑤ ip 명령으로 IP 주소를 확인

```
[root@localhost ~]# ip addr show ens160
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    link/ether 00:0c:29:c5:25:ab brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.147.131/32 scope global noprefixroute ens160
        valid_lft forever preferred_lft forever
(생략)
```

⑥ ip 명령으로 라우팅 테이블을 확인

```
[root@localhost ~]# ip route show
default via 192.168.147.2 dev ens160 proto static metric 100
192.168.147.2 dev ens160 proto static scope link metric 100
```

02 네트워크 설정

■ 네트워크 설정하기

⑦ 연결 프로파일 follow-me에 DNS를 설정하고 활성화

```
[root@localhost ~]# cat /etc/resolv.conf
# Generated by NetworkManager
[root@localhost ~]# nmcli con mod follow-me ipv4.dns "8.8.8.8 8.8.4.4"
[root@localhost ~]# nmcli con up follow-me
연결이 성공적으로 활성화되었습니다 (D-버스 활성 경로: /org/freedesktop/NetworkManager/
ActiveConnection/6)
[root@localhost ~]# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 8.8.8.8
nameserver 8.8.4.4
```

02 네트워크 설정

■ 네트워크 설정하기

⑧ nslookup 명령으로 www.daum.net의 IP 주소를 확인

```
[root@localhost ~]# nslookup
> www.daum.net
Server:      8.8.8.8
Address:     8.8.8.8#53

Non-authoritative answer:
www.daum.net  canonical name = daum-4vdtymgd.kgslb.com.
Name:   daum-4vdtymgd.kgslb.com
Address: 211.249.220.24
> exit
```

02 네트워크 설정

■ 네트워크 설정하기

혼자해보기 네트워크 설정하기

[따라해보기]에서 변경한 내용을 다시 원상태로 바꿔보도록 하자.

- ❶ nmcli device 명령으로 장치의 상태를 확인한다.
- ❷ 연결 프로파일의 상태를 확인한다.
- ❸ 연결 프로파일 ens160을 적용한다.
- ❹ 현재 시스템에 설정된 IP 주소를 확인한다.
- ❺ 현재 라우팅 테이블의 내용을 확인한다.
- ❻ 현재 DNS 설정을 확인한다.
- ❼ nslookup 명령으로 www.rockylinux.org의 IP 주소를 확인한다.

03 호스트 이름 설정

03 호스트 이름 설정

■ 호스트 이름 설정

- 네트워크 서비스를 제공하는 서버 시스템이라면 IP 주소 외에 호스트 이름을 설정해야 함
- 붙인 이름을 호스트 이름 설정 파일에 저장하고 DNS에 등록해야 서비스를 제공할 수 있음

■ 호스트 이름 출력하기: `uname -n` 명령

- `uname`은 원래 시스템 정보를 출력하는 명령인데, 여기에 `-n` 옵션을 사용하면 호스트 이름을 출력

uname

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

03 호스트 이름 설정

■ 호스트 이름 출력하기: `uname -n` 명령

- `uname -n` 명령을 실행하면 호스트 이름을 출력함
- 리눅스를 설치할 때 별도로 호스트 이름을 지정하지 않았기 때문에 기본값인 `localhost.localdomain`으로 설정되어 있음

```
[root@localhost ~]# uname -n  
localhost.localdomain
```

- `uname -a`를 실행하면 호스트 이름을 포함하여 시스템 관련 정보가 출력됨

```
[root@localhost ~]# uname -a  
Linux localhost.localdomain 5.14.0-284.11.1.el9_2.x86_64 #1 SMP PREEMPT_DYNAMIC Tue  
May 9 17:09:15 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
```

03 호스트 이름 설정

■ 호스트 이름 출력 및 설정하기: hostname 명령

- hostname 명령도 호스트 이름을 출력. 이 명령으로 호스트 이름을 설정할 수도 있음

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

```
[root@localhost ~]# hostname
localhost.localdomain
```

- hostname 명령으로 호스트 이름을 설정하면 다음과 같음

```
[root@localhost ~]# hostname mail.han.server
[root@localhost ~]# hostname
mail.han.server
```

03 호스트 이름 설정

■ 호스트 이름 검색 및 설정하기: hostnamectl 명령

- hostnamectl 명령은 시스템의 호스트 이름을 관리
- 이 명령으로 호스트 이름을 검색하거나 설정할 수 있음

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 호스트명: 호스트명을 호스트 이름으로 설정한다.
- 사용 예 hostnamectl
hostnamectl status
hostnamectl set-hostname mail.han.server

03 호스트 이름 설정

■ 호스트 이름 검색 및 설정하기: hostnamectl 명령

- hostnamectl 명령으로 호스트 이름을 검색해 보자

```
[root@localhost ~]# hostnamectl
  Static hostname: (unset)
Transient hostname: mail.han.server
      Icon name: computer-vm
      Chassis: vm
    Machine ID: b79f0cabfac84e2ab3527f85365b740d
      Boot ID: 125e9dbc420a481094feea33d894e70b
  Virtualization: vmware
 Operating System: Rocky Linux 9.2 (Blue Onyx)
(생략)
```

- hostnamectl 명령의 실행 결과에서 Static hostname은 아직 설정되어 있지 않음을 알 수 있음
- Transient hostname은 임시로 설정된 호스트 이름으로, 바로 앞에서 hostname으로 설정한 이름

03 호스트 이름 설정

■ 호스트 이름 검색 및 설정하기: hostnamectl 명령

- nmcli gen hostname 명령을 입력하면 Static hostname을 출력함
- 이 예에서는 아직 Static hostname이 설정되지 않았으므로 아무것도 출력하지 않음

```
[root@localhost ~]# nmcli gen hostname
```

- hostnamectl 명령으로 호스트 이름을 localhost.localdomain으로 설정하면 다음과 같이 출력됨

```
[root@localhost ~]# hostnamectl set-hostname localhost.localdomain
[root@localhost ~]# hostnamectl
    Static hostname: localhost.localdomain
    Transient hostname: localhost
(생략)
```

- Static hostname이 설정되었고, Transient hostname도 localhost로 바뀌었음
- hostnamectl 명령은 단순히 호스트 이름만 출력하는 것이 아니라 호스트 이름과 관련된 다른 정보도 출력함

03 호스트 이름 설정

■ 호스트 이름 검색 및 설정하기: hostnamectl 명령

- nmcli gen hostname 명령으로 다시 확인해 보면 Static hostname에 설정된 이름이 출력 됨

```
[root@localhost ~]# nmcli gen hostname  
localhost.localdomain
```


03 호스트 이름 설정

■ 호스트 이름을 파일에 저장하기

- hostname 명령으로 호스트 이름이 바뀌기는 했지만 시스템을 재시작하면 원래의 이름으로 돌아 감
- 재시작해도 바뀐 호스트 이름이 유지되게 하려면 호스트 이름을 설정하는 파일 자체를 수정해야 함
- 로키 리눅스에서 호스트 이름을 저장하는 파일은 /etc/hostname

```
[root@localhost ~]# cat /etc/hostname  
localhost.localdomain
```

- 이 파일은 단순히 도메인 이름을 포함한 호스트 이름만 저장하고 있음. 이 파일의 내용을 수정하면 재시작해도 호스트 이름을 유지할 수 있음. 이때 편집기로 수정해도 되지만 hostnamectl 명령이나 nmcli gen hostname [호스트명]으로 수정하는 편이 좋음
- 호스트 이름을 새로 정의할 때는 한 네트워크에서 같은 이름을 사용하는 다른 호스트가 있으면 안 된다는 점을 꼭 기억해야 함. 사용자가 임의로 호스트 이름을 수정하면 안되며 반드시 시스템 관리자와 상의해야 함

03 호스트 이름 설정

■ 호스트 이름 설정하기

- ① **hostname** 명령으로 호스트 이름을 myrocky.server로 설정

```
[root@localhost ~]# hostname mylocky.server  
[root@localhost ~]# hostname  
mylocky.server
```

- ② **hostnamectl** 명령으로 호스트 이름과 관련 정보를 확인

```
[root@localhost ~]# hostnamectl  
Static hostname: localhost.localdomain  
Transient hostname: mylocky.server  
(생략)
```

03 호스트 이름 설정

■ 호스트 이름 설정하기

③ nmcli 명령으로 호스트 이름을 localhost.localdomain으로 설정

```
[root@localhost ~]# nmcli gen hostname localhost.localdomain
[root@localhost ~]# nmcli gen hostname local.localdomain
[root@localhost ~]# hostnamectl
  Static hostname: local.localdomain
            Icon name: computer-vm
(생략)
```

④ hostnamectl 명령으로 호스트 이름을 localhost.localdomain으로 설정

```
[root@localhost ~]# hostnamectl set-hostname localhost.localdomain
[root@localhost ~]# hostnamectl
  Static hostname: localhost.localdomain
  Transient hostname: localhost
(생략)
```

04 네트워크 상태 확인

04 네트워크 상태 확인

■ 통신 확인

- 네트워크에서 통신이 가능한지를 확인하는 대표적인 명령은 ping
- ping은 시스템이 외부와 통신 되는지 확인하거나 외부 서버가 동작하는지 확인할 때 사용

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 192.168.1.1
ping -a www.naver.com

04 네트워크 상태 확인

■ 옵션 없이 사용하기

- 옵션을 지정하지 않고 ping 명령을 사용하면 계속 패킷을 보냄
- 패킷은 기본적으로 56B의 크기로 보냄
- 64B는 56B의 데이터에 8B의 헤더 정보를 더한 것

```
[root@localhost ~]# ping 192.168.147.2
PING 192.168.147.2 (192.168.147.2) 56(84) bytes of data.
64 bytes from 192.168.147.2: icmp_seq=1 ttl=128 time=0.374 ms
64 bytes from 192.168.147.2: icmp_seq=2 ttl=128 time=0.239 ms
64 bytes from 192.168.147.2: icmp_seq=3 ttl=128 time=0.215 ms
```

04 네트워크 상태 확인

■ 옵션 없이 사용하기

- ng은 -c 옵션으로 보낼 패킷 수를 지정하지 않으면 계속 패킷을 보냄
- Ctrl +C로 ping을 종료해야 함. ping이 종료되면 다음과 같이 통계 정보가 출력됨

```
64 bytes from 192.168.147.2: icmp_seq=9 ttl=128 time=0.263 ms
64 bytes from 192.168.147.2: icmp_seq=10 ttl=128 time=0.194 ms
^C
--- 192.168.147.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9226ms
rtt min/avg/max/mdev = 0.194/0.267/0.374/0.058 ms
```

- 통계 정보로는 보낸 패킷 수, 보낸 패킷 중에서 받은 패킷 수, 보내고 받은 패킷 중 유실된 패킷의 비율, 통신 속도의 최솟값·평균값·최댓값이 출력
- 위의 결과를 보면 열 개 패킷을 보내 모두 수신했으며, 유실된 것은 없고 평균 0.267밀리초 걸렸음을 알 수 있음. 통신 시간이 낮을수록 네트워크의 상태가 양호하다는 것을 의미

04 네트워크 상태 확인

■ -q 옵션 사용하기

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

```
[root@localhost ~]# ping -q 192.168.147.2
PING 192.168.147.2 (192.168.147.2) 56(84) bytes of data.
^C
--- 192.168.147.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1055ms
rtt min/avg/max/mdev = 0.160/0.221/0.282/0.061 ms
```


04 네트워크 상태 확인

■ -c 옵션 사용하기

- -c 옵션을 사용하면 보낼 패킷 수를 지정할 수 있음
- 다음은 패킷 수를 세 개로 지정한 예

```
[root@localhost ~]# ping -c 3 192.168.147.2
PING 192.168.147.2 (192.168.147.2) 56(84) bytes of data.
64 bytes from 192.168.147.2: icmp_seq=1 ttl=128 time=0.134 ms
64 bytes from 192.168.147.2: icmp_seq=2 ttl=128 time=0.306 ms
64 bytes from 192.168.147.2: icmp_seq=3 ttl=128 time=0.220 ms

--- 192.168.147.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2037ms
rtt min/avg/max/mdev = 0.134/0.220/0.306/0.070 ms
```

04 네트워크 상태 확인

■ 도메인 이름 사용하기

- ping 명령을 사용할 때 IP 주소가 아닌 도메인 이름을 지정할 수도 있음. 이는 보통 해당 도메인이 동작하는지 확인하기 위해 사용
- 예를 들어 www.hanbit.co.kr 사이트가 통신 가능한 상태인지 확인하려면 다음과 같이 작성

```
[root@localhost ~]# 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=128 time=6.58 ms
64 bytes from 218.38.58.195 (218.38.58.195): icmp_seq=2 ttl=128 time=6.17 ms
64 bytes from 218.38.58.195 (218.38.58.195): icmp_seq=3 ttl=128 time=4.74 ms
64 bytes from 218.38.58.195 (218.38.58.195): icmp_seq=4 ttl=128 time=7.65 ms
^C
--- www.hanbit.co.kr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 4.739/6.287/7.654/1.044 ms
```

- ping으로 연결되지 않는다고 해서 무조건 해당 시스템이 동작하지 않는다고 단정할 수 없음

04 네트워크 상태 확인

■ 네트워크 상태 정보 출력

- netstat 명령은 네트워크 연결 상태, 라우팅 테이블, 인터페이스 관련 통계 정보 등을 출력
- 이 명령으로 현재 시스템에 열려 있는 포트가 무엇인지도 확인할 수 있음

netstat

- 기능 네트워크의 상태 정보를 출력한다.
- 형식 netstat [옵션]
- 옵션
 - a: 모든 소켓 정보를 출력한다.
 - r: 라우팅 정보를 출력한다.
 - n: 호스트명 대신 IP 주소로 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

- netstat 명령에 -r 옵션을 사용하면 라우팅 테이블을 확인할 수 있음
- -n 옵션을 함께 사용하면 이름 대신 IP 주소를 출력
- netstat -r로 출력되는 라우팅 테이블 정보는 route 명령의 출력 결과와 같음

```
[root@localhost ~]# netstat -r
Kernel IP routing table
Destination      Gateway         Genmask         Flags   MSS      Window    irtt     Iface
default          _gateway       0.0.0.0         UG      0         0         0        ens160
192.168.147.0    0.0.0.0        255.255.255.0   U        0         0         0        ens160
```

04 네트워크 상태 확인

■ 현재 열려 있는 포트 확인하기

- 네트워크로 통신을 할 때 현재 통신이 진행 중인 서비스는 해당 서비스 포트가 LISTEN 상태이므로 이를 통해 어떤 포트가 열려 있고 서비스 중인지 확인할 수 있음

```
[root@localhost ~]# netstat -an | grep LISTEN
tcp        0      0 127.0.0.1:631          0.0.0.0:*              LISTEN
tcp        0      0 0.0.0.0:111            0.0.0.0:*              LISTEN
tcp        0      0 0.0.0.0:22             0.0.0.0:*              LISTEN
tcp6       0      0 :::1:631               :::*                    LISTEN
tcp6       0      0 :::111                 :::*                    LISTEN
tcp6       0      0 :::22                  :::*                    LISTEN
unix  2      [ ACC ]     STREAM    LISTENING   35704    /tmp/.X11-unix/X0
(생략)
```

- 다음 예에서는 631 번, 111번, 22번 등이 서비스 중임을 알 수 있음

04 네트워크 상태 확인

■ 현재 열려 있는 포트를 사용 중인 프로세스 확인하기: -p 옵션

- -p 옵션으로 현재 열려 있는 포트를 사용하는 프로세스를 확인
- 다음 예를 보면 현재 인터넷으로 연결된 ssh 서비스가 있고, 그 외에 유닉스 도메인 소켓 (DGRAM)으로 내부 포트를 사용 중인 프로세스들이 있음을 알 수 있음

```
[root@localhost ~]# netstat -p | more
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/
Program name
tcp        0      64 localhost.localdoma:ssh 192.168.147.1:uadtc     ESTABLISHED 2195/sshd:
user1 [p
udp        0       0 localhost.locald:bootpc 192.168.147.254:bootps  ESTABLISHED 1157/
NetworkManager
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags Type  State      I-Node PID/Program name Path
unix  2      [ ]  DGRAM          33680 2202/systemd    /run/user/1000/systemd/notify
unix  4      [ ]  DGRAM CONNECTED 13707 1/systemd       /run/systemd/notify
(생략)
```

04 네트워크 상태 확인

■ 인터페이스별 네트워크 통계 정보 확인하기: -i 옵션

- 현재 사용 중인 인터페이스별로 네트워크 통계 정보를 확인하려면 -i 옵션을 사용
- RXOK, TX-OK는 정상적으로 주고받은 패킷의 개수
- RX-ERR, RX-DROP, RX-OVR, TX-ERR, TX-DROP, TX-OVR은 송수신 중에 오류가 발생한 패킷의 개수

```
[root@localhost ~]# netstat -i
```

```
Kernel Interface table
```

Iface	MTU	RX-OK	RX-ERR	RX-DROP	RX-OVR	TX-OK	TX-ERR	TX-DROP	TX-OVR	Flg
ens160	1500	3282	0	0	0	2417	0	0	0	BMRU
lo	65536	44	0	0	0	44	0	0	0	LRU

04 네트워크 상태 확인

■ 프로토콜별 네트워크 통계 정보 확인하기: -s 옵션

- 프로토콜별로 네트워크 통계 정보를 확인하려면 -s 옵션을 사용
- 다음은 IP 프로토콜, ICMP 프로토콜, ICMPMSG 프로토콜, TCP/UDP 프로토콜별로 통계 정보를 출력한 예

```
[root@localhost ~]# netstat -s
Ip:
  Forwarding: 2
  3339 total packets received
  5 with invalid addresses
  0 forwarded
  0 incoming packets discarded
  3258 incoming packets delivered
  2268 requests sent out
  57 dropped because of missing route
```


04 네트워크 상태 확인

■ 프로토콜별 네트워크 통계 정보 확인하기: -s 옵션

```
Icmp:
  27 ICMP messages received
  0 input ICMP message failed
  ICMP input histogram:
    destination unreachable: 8
    echo replies: 19
  46 ICMP messages sent
  0 ICMP messages failed
  ICMP output histogram:
    destination unreachable: 8
    echo requests: 38
IcmpMsg:
  InType0: 19
  InType3: 8
  OutType3: 8
  OutType8: 38
```

04 네트워크 상태 확인

■ 프로토콜별 네트워크 통계 정보 확인하기: -s 옵션

Tcp:

- 6 active connection openings
- 1 passive connection openings
- 0 failed connection attempts
- 0 connection resets received
- 1 connections established
- 2980 segments received
- 1926 segments sent out
- 1 segments retransmitted
- 0 bad segments received
- 4 resets sent

Udp:

- 243 packets received
- 8 packets to unknown port received

(생략)

04 네트워크 상태 확인

■ MAC 주소와 IP 주소 확인

- 같은 네트워크에 연결된 시스템들의 MAC 주소와 IP 주소를 확인하려면 arp 명령을 사용
- arp는 'address resolution protocol'의 약자

arp

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

04 네트워크 상태 확인

■ MAC 주소와 IP 주소 확인

- arp 명령을 수행하면 현재 같은 네트워크에 연결된 시스템의 MAC 주소와 IP 주소를 출력
- arp 명령으로 확인한 결과 두 개의 시스템이 연결되어 있음. HW address가 MAC 주소

```
[root@localhost ~]# arp
```

Address	HWtype	HWaddress	Flags	Mask	Iface
_gateway	ether	00:50:56:ed:23:8f	C		ens160
192.168.147.254	ether	00:50:56:f5:e5:42	C		ens160
192.168.147.1	ether	00:50:56:c0:00:08	C		ens160

- 특정 시스템의 MAC 주소를 확인하려면 해당 시스템의 IP 주소를 지정함

```
[root@localhost ~]# arp 192.168.147.2
```

Address	HWtype	HWaddress	Flags	Mask	Iface
_gateway	ether	00:50:56:ed:23:8f	C		ens160

04 네트워크 상태 확인

■ 패킷 캡처 명령

- 네트워크에서 주고받는 패킷을 캡처하여 확인하는 명령은 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

- 이 명령은 네트워크의 상태를 확인하기 위해 패킷을 캡처하여 분석할 때 사용
- 하지만 이 명령을 악용하면 해킹의 도구가 될 수도 있으므로 주의해야 함

04 네트워크 상태 확인

■ 옵션 없이 사용하기

- tcpdump 명령을 옵션 없이 수행하면 현재 시스템에서 주고받는 모든 패킷을 캡처하여 패킷의 헤더 부분 정보를 출력

```
[root@localhost ~]# tcpdump
dropped privs to tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens160, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:09:49.543646 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
2374310413:2374310493, ack 1026663830, win 592, length 80
21:09:49.543750 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
80:144, ack 1, win 592, length 64
21:09:49.543844 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
144:272, ack 1, win 592, length 128
21:09:49.543920 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
272:416, ack 1, win 592, length 144
21:09:49.543970 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
416:496, ack 1, win 592, length 80
(생략)
```

- 다음 예를 보면 ens160을 통해 캡처한 패킷을 출력하며 로컬 호스트와 192.168.147.1.uadtc 사이에 주고받은 패킷이 캡처된 것을 알 수 있음

04 네트워크 상태 확인

■ 옵션 없이 사용하기

- tcpdump는 Ctrl +C로 종료하지 않으면 계속 캡처하여 출력함
- Ctrl +C로 종료하면 캡처한 패킷의 개수를 출력하고 종료함

```
21:09:50.883225 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
29216:29504, ack 321, win 638, length 288
21:09:50.932440 IP 192.168.147.1.uadtc > localhost.localdomain.ssh: Flags [.], ack
29504, win 4100, length 0
^C
159 packets captured
162 packets received by filter
0 packets dropped by kernel
```

04 네트워크 상태 확인

■ 캡처할 패킷 개수 지정하기: -c 옵션

- tcpdump 명령에 -c 옵션을 사용하여 캡처할 패킷 개수를 지정할 수 있음
- 예를 들어 패킷을 세 개만 캡처하려면 -c 3을 옵션으로 지정

```
[root@localhost ~]# tcpdump -c 3
dropped privs to tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens160, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:17:17.715477 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
2374340909:2374340989, ack 1026664662, win 638, length 80
21:17:17.715567 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
80:144, ack 1, win 638, length 64
21:17:17.715814 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
144:272, ack 1, win 638, length 128
3 packets captured
19 packets received by filter
0 packets dropped by kernel
```


04 네트워크 상태 확인

■ 캡처한 패킷 정보를 파일로 저장하기: -w 옵션

- tcpdump 명령으로 캡처한 패킷 정보를 파일에 저장하려면 -w 옵션을 사용
- 예를 들어 패킷 세 개를 캡처하여 dump.out 파일에 저장하려면 다음과 같이 함

```
[root@localhost ~]# tcpdump -c 3 -w dump.out
dropped privs to tcpdump
tcpdump: listening on ens160, link-type EN10MB (Ethernet), snapshot length 262144
bytes
3 packets captured
11 packets received by filter
0 packets dropped by kernel
```

- 그런데 패킷을 저장한 파일이 바이너리 파일이기 때문에 cat이나 vi 명령으로 파일 내용을 확인할 수 없음

04 네트워크 상태 확인

■ 캡처한 패킷 정보를 파일로 저장하기: -w 옵션

- file 명령으로 파일 종류를 확인하면 tcpdump 캡처 파일이라고 출력되며, cat 명령으로 dump.out 파일을 확인하면 이상한 문자가 출력됨
- 즉, 캡처 파일의 내용을 확인하려면 -r 옵션을 사용해야 함

```
[root@localhost ~]# file dump.out
dump.out: pcap capture file, microsecond ts (little-endian) - version 2.4 (Ethernet,
capture length 262144)
[root@localhost ~]# cat dump.out
tcePV
)%EHx7~@Z
=1VP~>zV.$SiWLRd;c;!d5hqQI{U}H]mV,
"5djrc1 tceuvvPV
)%EHhZ
=1VP~. ]U nA\ aS :G8 Y
~#1 Htce<<
(생략)
```

04 네트워크 상태 확인

■ 캡처한 패킷 파일 읽기: -r 옵션

- tcpdump 명령으로 캡처한 패킷 정보를 저장한 파일의 내용을 읽으려면 -r 옵션을 사용
- 앞에서 저장한 dump.out 파일의 내용을 확인해 보면 다음과 같음

```
[root@localhost ~]# tcpdump -r dump.out
reading from file dump.out, link-type EN10MB (Ethernet), snapshot length 262144
dropped privs to tcpdump
21:17:56.397985 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
2374343837:2374343917, ack 1026665558, win 638, length 80
21:17:56.398197 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
80:144, ack 1, win 638, length 64
21:17:56.398234 IP 192.168.147.1.uadtc > localhost.localdomain.ssh: Flags [.], ack 80,
win 4097, length 0
```

04 네트워크 상태 확인

■ 특정 포트로 송수신되는 패킷 캡처하기: tcp port 옵션

- 특정 포트로 송수신되는 패킷을 캡처하려면 tcp port 옵션을 사용
- 다음은 192.168.147.1과 주고받는 패킷 중에서 22번 포트에 해당하는 패킷 세 개를 캡처한 예

```
[root@localhost ~]# tcpdump -c 3 tcp port 22
dropped privs to tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens160, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:21:23.956670 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
2374355709:2374355885, ack 1026673494, win 661, length 176
21:21:23.956717 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: Flags [P.], seq
176:272, ack 1, win 661, length 96
21:21:23.956927 IP 192.168.147.1.uadtc > localhost.localdomain.ssh: Flags [.], ack 272,
win 4098, length 0
3 packets captured
3 packets received by filter
0 packets dropped by kernel
```

- 22번 포트를 캡처하기 위해 ssh를 동작시킴
- 22번 포트를 사용하고 있지 않다면 캡처되는 것이 없을 수도 있음

04 네트워크 상태 확인

■ 캡처한 내용을 ASCII로 보기: -X 옵션

- 캡처한 내용을 ASCII로 보려면 -X 옵션을 사용
- 앞에서 저장한 dump.out 파일을 -X 옵션으로 출력해 보면 다음과 같음

```
[root@localhost ~]# tcpdump -Xqr dump.out
reading from file dump.out, link-type EN10MB (Ethernet), snapshot length 262144
dropped privs to tcpdump
21:17:56.397985 IP localhost.localdomain.ssh > 192.168.147.1.uadtc: tcp 80
    0x0000:  4548 0078 377e 4000 4006 5ae6 c0a8 9381  EH.x7~@.Z.....
    0x0010:  c0a8 9301 0016 0acf 8d85 9c9d 3d31 ac56  .....=1.V
    0x0020:  5018 027e a83e 0000 0280 7aab 56c5 102e  P..~.>...z.V...
    0x0030:  ab24 53fb 69f6 808b 57ff 4c52 648b fe63  .$.i...W.LRd..c
    0x0040:  3bcc 8c21 648e 35c6 e913 8068 7190 bc51  ;..!d.5...hq..Q
    0x0050:  9f59 7f49 7b5f 1455 9311 9e48 5dfc 6de7  .Y.I{.U...H].m.
    0x0060:  562c c10c 1122 e8ee 3564 ebd7 6ad5 7263  V,..."..5d..j.rc
    0x0070:  f0f8 016c b7da a1dd                      ...l....
```

(생략)

04 네트워크 상태 확인

■ 캡처한 내용을 ASCII로 보기: -X 옵션

- -q 옵션은 일부 정보를 생략하고 출력함
- ASCII로 출력한 결과를 보면 무슨 의미인지 잘 모르겠지만, 관심을 가지고 분석하면 내용을 파악할 수도 있음. 네트워크를 통해 주고받는 데이터의 내용을 노출하기 때문
- 따라서 암호화되어 있지 않다면 중요 데이터가 그대로 공개될 수도 있음

04 네트워크 상태 확인

■ 네트워크 상태 확인하기

① ping 명령으로 www.google.co.kr이 동작하는지 확인해 봄

```
[root@localhost ~]# ping www.google.co.kr
PING www.google.co.kr (142.251.222.3) 56(84) bytes of data.
64 bytes from nrt13s71-in-f3.1e100.net (142.251.222.3): icmp_seq=1 ttl=128 time=63.9
ms
64 bytes from nrt13s71-in-f3.1e100.net (142.251.222.3): icmp_seq=2 ttl=128 time=65.4
ms
64 bytes from nrt13s71-in-f3.1e100.net (142.251.222.3): icmp_seq=3 ttl=128 time=64.6
ms
64 bytes from nrt13s71-in-f3.1e100.net (142.251.222.3): icmp_seq=4 ttl=128 time=65.3
ms
^C
--- www.google.co.kr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 63.910/64.818/65.445/0.609 ms
```

04 네트워크 상태 확인

■ 네트워크 상태 확인하기

- ② netstat 명령으로 현재 열려 있는 tcp 포트를 사용하는 프로세스를 확인

```
[root@localhost ~]# netstat -pt
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/
Program name
tcp      0      64 localhost.localdoma:ssh 192.168.147.1:uadtc ESTABLISHED 2195/
sshd: user1 [p
```

- ③ 웹 브라우저(파이어폭스)를 동작시키고 www.hanbit.co.kr 사이트에 접속

04 네트워크 상태 확인

■ 네트워크 상태 확인하기

- ④ tcp 포트 80으로 주고받는 패킷 다섯 개를 캡처하여 httpdump 파일에 저장

```
[root@localhost ~]# tcpdump -c 5 -w httpdump tcp port 80
dropped privs to tcpdump
tcpdump: listening on ens160, link-type EN10MB (Ethernet), snapshot length 262144
bytes
5 packets captured
6 packets received by filter
0 packets dropped by kernel
```

04 네트워크 상태 확인

■ 네트워크 상태 확인하기

⑤ httpdump 파일에 저장한 패킷의 내용을 ASCII로 확인

```
[root@localhost ~]# tcpdump -Xqr httpdump
reading from file httpdump, link-type EN10MB (Ethernet), snapshot length 262144
dropped privs to tcpdump
21:26:10.260503 IP localhost.localdomain.45624 > nrt12s51-in-f3.1e100.net.http: tcp 0
    0x0000: 4500 0028 3dd2 4000 4006 e117 c0a8 9381  E..(=.@.@.....
    0x0010: acd9 1ae3 b238 0050 7d02 9ec5 5d8b 622a  .....8.P}...].b*
    0x0020: 5010 f98a 1c01 0000                                P.....
21:26:10.260550 IP localhost.localdomain.35464 > nrt12s51-in-f3.1e100.net.http: tcp 0
    0x0000: 4500 0028 ce08 4000 4006 50e1 c0a8 9381  E..(..@.@.P.....
    0x0010: acd9 1ae3 8a88 0050 d7fb e0f7 40d7 54d1  .....P....@.T.
    0x0020: 5010 f98a 1c01 0000                                P.....
```

(생략)

Thank you!