네트워크 관리

네트워크 기초

- TCP/IP 프로토콜
  - ✓ 개요
    - 프로토콜이란 컴퓨터와 컴퓨터 사이에 데이터를 어떻게 주고받을 것인지를 정의한 통신 규약으로 동일한 프로토콜을 사용하는 기기 간에 통신이 가능
    - 인터넷이라고 부르는 네트워크는 TCP/IP 라는 프로토콜에 따라 데이터를 주고 받음

네트워크 기초

- ❖ TCP/IP 프로토콜
  - ✓ 구조
    - ● 계층

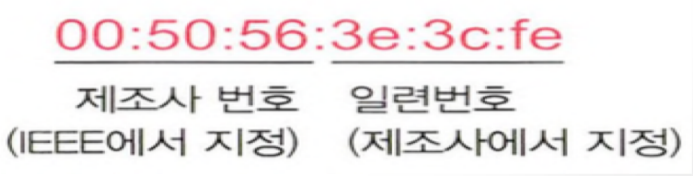
네트워크 기초

- TCP/IP 프로토콜
  - ✓ 구조
    - 계층 별 역할과 대표 프로토콜

계층	기능	프로토콜	전송 단위
응용 계층	서비스 제공 응용 프로그램	DNS, FTP, SSH, HTTP, Telnet	Message
전송 계층	응용프로그램으로 데이터 전달, 데이터 흐름 제어 및 전송 신뢰성 담당	TCP, UDP	Segment
네트워크 계층	주소 관리 및 경로 탐색	IP, ICMP	Packet
링크 계층	네트워크 장치 드라이버	ARP	Frame
물리 계층	케이블 등 전송 매체	구리선, 광케이블, 무선	Bit

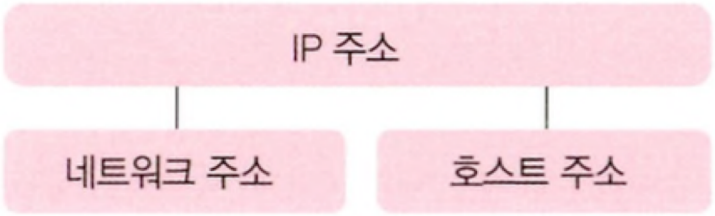
네트워크 기초

- ❖ 주소
  - ✓ MAC Address
    - MAC는 'media access control'의 약자로 MAC 주소는 하드웨어를 위한 주소이며 다른 말로 이더넷 주소, 하드웨어 주소, 물리 주소 라고도 함
    - MAC 주소는 네트워크 인터페이스 카드(랜 카드)에 저장된 주소
    - MAC 주소는 기본적으로 네트워크 인터페이스 카드가 만들어질 때 부여되며 원칙적으로는 수정할 수 없지만 일부 네트워크 인터페이스 카드의 경우 사용자가 MAC 주소를 수정할 수 있도록 허용함
    - 특별한 경우가 아니면 MAC 주소는 수정하지 않는 것이 좋음
    - MAC 주소는 각 하드웨어를 구별하는 역할을 수행
    - MAC 주소는 : 이나 -으로 구분되는 여섯 개의 16진수로 구성되며 총 48bit
    - 앞의 세 자리는 제조사 번호이고 뒤의 세 자리는 일련번호



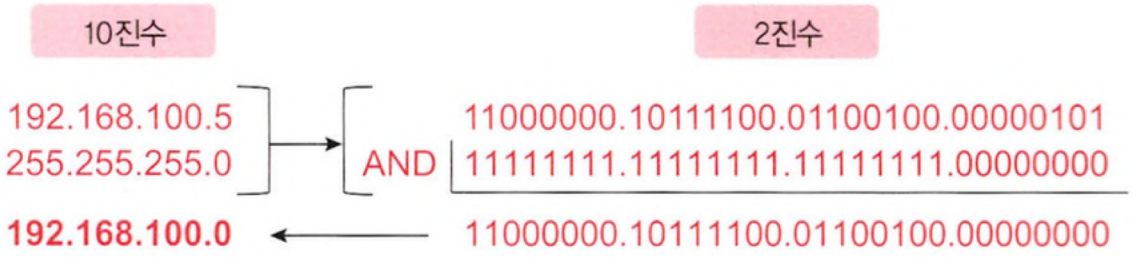
네트워크 기초

- ❖ 주소
  - ✓ IP Address
    - 컴퓨터가 인터넷에 연결되려면 IP 주소가 할당되어 있어야 함
    - 보통 인터넷 주소라고 부르는 것이 IPinternet protocol 주소
    - IP 주소는 인터넷으로 연결된 네트워크에서 각 컴퓨터를 구분하기 위해 사용
    - IPV4 주소는 1바이트 크기의 숫자 네 개로 구성되므로 총 4 바이트
    - 192.168.100.5와 같이 숫자 네 가지와 마침표 (.) 로 구성된 주소가 IPv4 주소
    - IPv4 주소는 고갈되어 더 이상 새로운 주소를 배정받을 수 없는데 이를 대체하기 위해 개발된 주소는 IPv6 예를 들어 fe80::2500::56ff::fe3e::3cfe와 같이 16 진수로 표기하며 128bit 크기
    - TCP/IP 프로토콜의 3~5계층은 IP 주소를 사용
    - 구성



네트워크 기초

- ❖ 주소
  - ✓ NetMask
    - IP 주소에서 네트워크 부분을 알려주는 역할을 하는 것이 netmask
    - 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



    - 위의 경우 192.168.100.5/24 이렇게 표현하기도 함

네트워크 기초

- ❖ 주소
  - ✓ Broadcast Address
    - 네트워크에 있는 모든 컴퓨터에 메시지를 보낼 때 사용하는 것으로 호스트 부분을 모두 1로 설정
    - 192.168.100.0/24 이면 브로드캐스트 주소는 192.168.100.255

네트워크 기초

- ❖ 주소
  - ✓ 호스트 이름
    - 컴퓨터가 인터넷에 연결되려면 IP 주소가 있어야 하는데 이 주소를 사용하여 메일도 보내고 웹 사이트에도 접속
    - 컴퓨터가 숫자를 좋아하는 데 비해 사람은 숫자로 된 주소를 기억하기가 쉽지 않음
    - 사람은 숫자보다 이름으로 된 것을 더 잘 기억하기 때문에 등장한 것이 호스트 이름으로 202.179.177.21 대신 www.naver.com을 사용하는 것
    - 호스트 이름도 IP 주소처럼 두 부분으로 구성
    - 네이버의 경우 naver.com이 네트워크 부분이고 www가 호스트 부분에 해당
    - 개인용 PC라면 호스트 이름을 붙일 필요가 없겠지만 웹 서버와 같이 네트워크 서비스를 제공하는 서버 컴퓨터는 용도에 따라 호스트 이름을 붙여서 사용해야 함

네트워크 기초

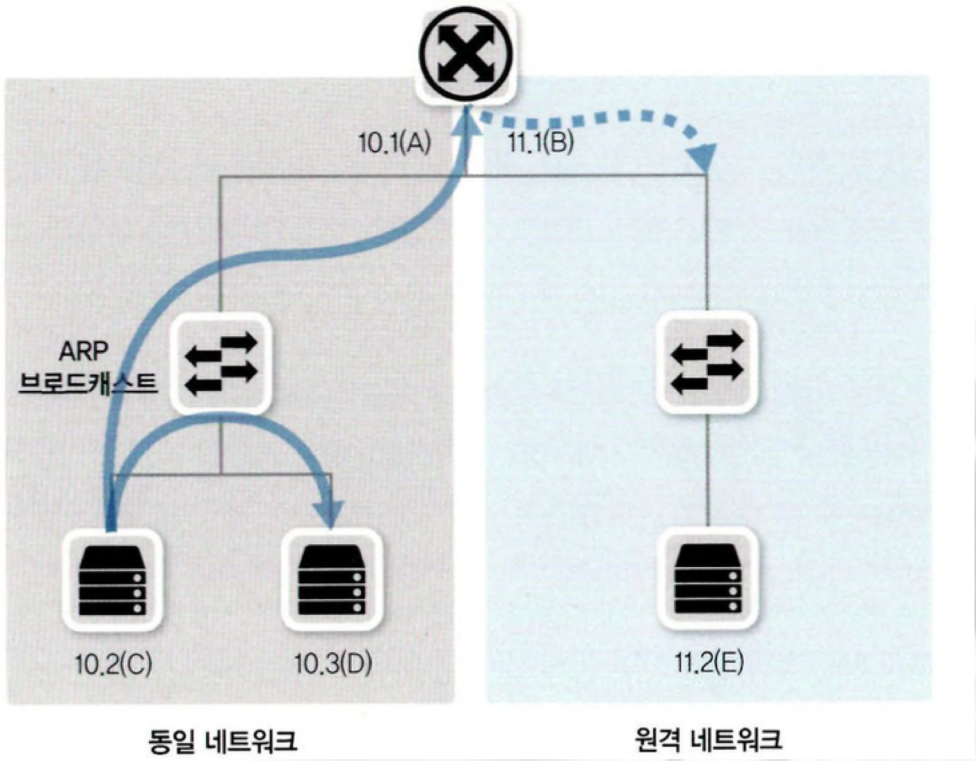
- DNS
  - ✓ DNS(Domain Name System)는 우리가 흔히 사용하는 웹사이트 주소(예: google.com)를 컴퓨터가 이해할 수 있는 IP 주소(예: 142.250.190.46)로 변환해 주는 시스템
  - ✓ 인터넷상의 모든 기기는 숫자로 된 IP 주소를 통해 서로를 식별하지만 사람이 이를 모두 외우기 어렵기 때문에 DNS를 통해 문자로 된 이름을 사용
  - ✓ 웹 브라우저에 google.com 등과 같은 URL을 사용하는데 실제 원하는 서버에 접근하려면 이 URL을 해당 컴퓨터의 IP 주소로 변환시켜야 하는데 이 일을 네임 서버 또는 DNS 서버라고 하는 컴퓨터가 담당
  - ✓ www.google.com을 IP 주소로 변환하는 과정을 이름 해석(resolution)이라고 함
  - ✓ 네트워크에서 컴퓨터를 구분하는 유일한 방법은 IP 주소인데 인터넷에 연결된 모든 컴퓨터에는 중복되지 않는 IP 주소가 있음
  - ✓ DNS Server를 이용하지 않는다면 IP 주소를 알아내는 과정이 생략되므로 인터넷 속도가 더 빨라짐

네트워크 기초

- DHCP
  - ✓ DHCP(Dynamic Host Configuration Protocol, 동적 호스트 설정 프로토콜)는 네트워크 환경에서 IP 주소 관리를 자동화하기 위해 필수적으로 사용되는 네트워크 프로토콜
  - ✓ 컴퓨터, 스마트폰, 서버 등 네트워크에 연결되는 모든 기기(호스트)가 서로 통신하려면 고유한 IP 주소가 필요한데 DHCP는 이를 자동으로 할당하고 관리해 주는 역할을 수행
  - ✓ 새로운 컴퓨터를 네트워크에 연결할 때마다 관리자가 직접 사용 가능한 IP 주소, 서브넷 마스크, 기본 게이트웨이, DNS 서버 주소를 수동으로 입력해야 하지만 DHCP 서버가 있는 환경에서는 기기가 네트워크에 연결되는 순간 네트워크상에 있는 DHCP 서버에게 "내가 사용할 수 있는 IP 주소를 달라"고 요청을 하고 서버는 자신이 가지고 있는 IP 주소 풀(Pool)에서 남는 주소를 하나 꺼내어 기기에게 일정 기간 대여(Lease)해준 후 대여 기간이 만료되거나 기기가 네트워크에서 이탈하면 해당 IP 주소는 다시 회수되어 다른 기기가 사용할 수 있게 함

네트워크 기초

- Gateway
  - ✓ 로컬 네트워크에서는 ARP 브로드캐스트를 이용해 도착지 MAC 주소를 학습할 수 있고 이 MAC 주소를 이용해 직접 통신할 수 있지만 원격 네트워크 통신은 네트워크를 넘어 전달되지 못하는 브로드캐스트의 성질 때문에 네트워크 장비의 도움이 필요한데 이 장비를 게이트웨이(Gatway)라고 하고 게이트웨이에 대한 정보를 PC나 네트워크 장비에 설정하는 항목이 기본 게이트웨이(Default Gateway)



네트워크 설정

- ❖ 주소
  - ✓ 포트 번호
    - ● 서비스 별로 포트 번호를 설정한 파일
      - ◆ adam@itstudy:~$ cat /etc/services | grep ftp

```
ftp-data        20/tcp
ftp             21/tcp
tftp            69/udp
ftps-data       989/tcp                 # FTP over SSL
(data)
ftps            990/tcp
venus-se        2431/udp                # udp sftp side effect
codasrv-se      2433/udp                # udp sftp side effect
gsiftp          2811/tcp
```

        - /etc/services 파일에 저장된 포트 번호는 국제 표준으로 합의하여 사용하고 있는 것
        - 사용자가 개발한 네트워크 프로그램은 이 파일에 정의되지 않은 번호를 사용하여 서비스를 제공하는 것을 권장

네트워크 설정

- 인터넷에 연결하려면 설정해야 할 주소
  - ✓ IP Address
  - ✓ subnet mask
  - ✓ broadcast address
  - ✓ gateway address
  - ✓ DNS Address

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ 개요
    - 우분투에서는 NetworkManager가 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
    - 전통적으로 유닉스와 리눅스에서 제공하던 스크립트 파일인 ifcfg 형식의 네트워크 설정 파일도 계속 지원
    - 사용자는 네트워크 관리자와 기존 스크립트 파일 방식을 모두 사용할 수 있음
    - 네트워크 관리와 관련된 도구
      - ◆ 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 [설정] – [네트워크] 에서 설정할 수 없는 부분도 설정 가능
      - ◆ ip 명령: 네트워크를 설정하는 명령을 제공

네트워크 설정

- 네트워크 관리자
  - ✓ 개요
    - 네트워크 관리자는 네트워크 설정 정보를 connection profile에 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 nm-connection-editor는 GUI 기반 도구
    - 도구를 사용하여 네트워크 설정을 변경하면 네트워크 관리자가 자동으로 인식
    - ip 명령으로도 네트워크를 설정할 수 있지만 이 명령으로 네트워크의 설정을 변경할 경우 네트워크 관리자가 자동으로 인식하지 못함

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ 설치
    - sudo apt install network-manager
  - ✓ 실행: sudo systemctl start NetworkManager
  - ✓ 부팅 시 자동 실행: sudo systemctl enable NetworkManager
  - ✓ 상태 확인
    - systemctl status NetworkManager

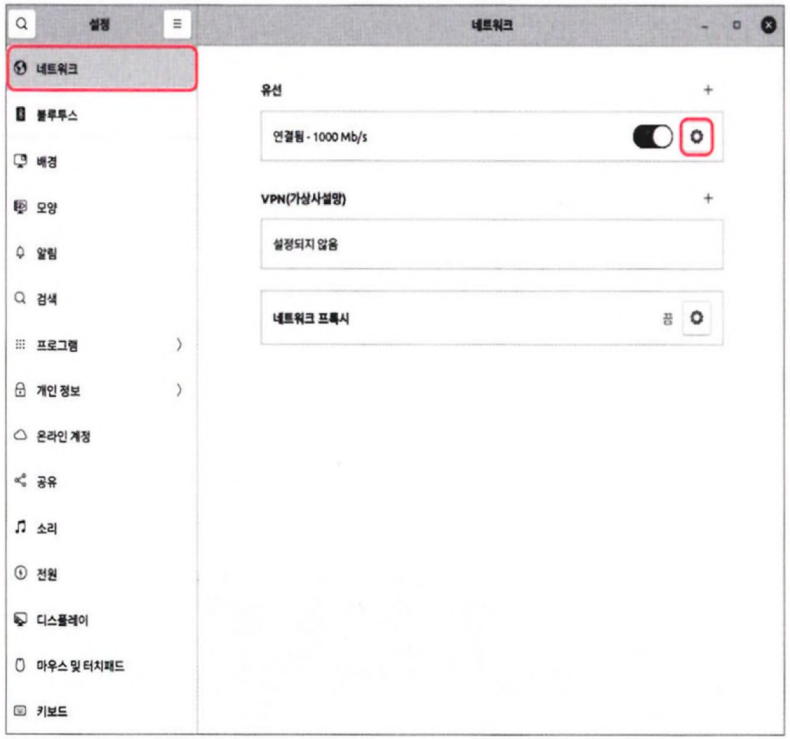
```
Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled;
preset: enabled)
     Active: active (running) since Mon 2024-09-09 09:14:56 UTC; 1min 6s ago
       Docs: man:NetworkManager(8)
   Main PID: 8640 (NetworkManager)
      Tasks: 4 (limit: 4548)
     Memory: 2.9M (peak: 20.0M)
        CPU: 229ms
     CGroup: /system.slice/NetworkManager.service
             └─8640 /usr/sbin/NetworkManager --no-daemon
```

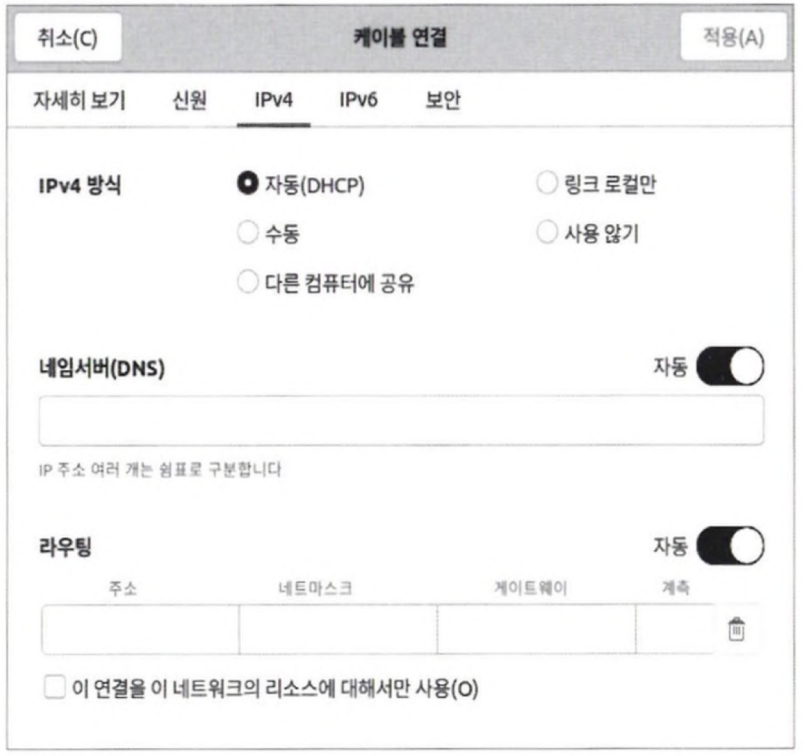
네트워크 설정

- ❖ 네트워크 관리자
  - ✓ 그놈의 [설정] – [네트워크]
    - 그놈은 윈도의 제어판처럼 시스템과 네트워크 설정을 위한 기능을 제공
    - [설정]은 우분투에서 [프로그램 표시] – [설정]을 선택하여 실행하거나 바탕화면에서 마우스 오른쪽 버튼을 클릭하여 선택
    - [설정]에서 [네트워크]를 선택하면 네트워크 설정 창이 뜸
    - 네트워크 설정 창에서 유선의 오른쪽에 있는 o 를 선택하면 유선 네트워크 설정 창이 뜨고 IPv4 메뉴에서 IP 주소와 네임서버(DNS), 라우팅 정보를 설정할 수 있음

네트워크 설정

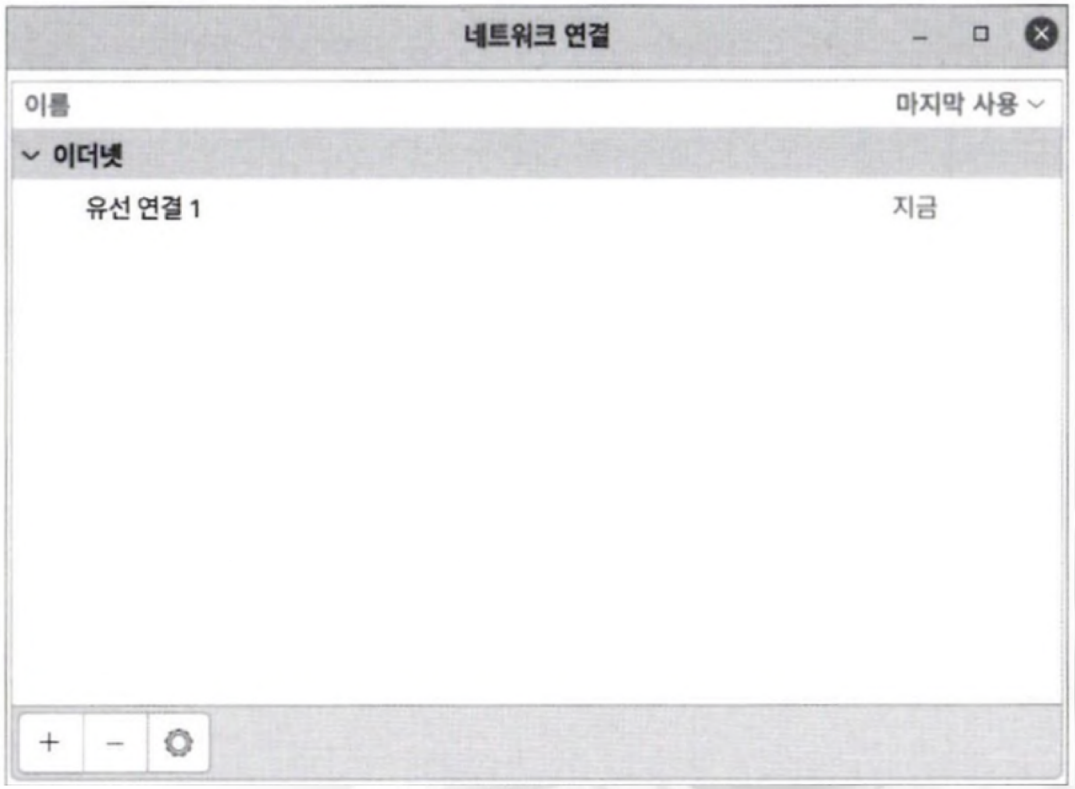
- 네트워크 관리자
  - ✓ 그놈의 [설정] – [네트워크]로 설정





네트워크 설정

- ❖ 네트워크 관리자
  - ✓ nm-connection-editor로 설정
    - ● nm-connection-editor는 네트워크 관리자와 함께 설치되며 터미널에서 실행



네트워크 설정

- ❖ 네트워크 관리자
  - ✓ nm-connection-editor로 설정
    - nm-connection-editor 창에서 [유선 연결 1]을 선택하고 하단의 아이콘을 클릭하면 설정 창이 뜨는데 여기서 세부적인 설정을 할 수 있음

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ nmcli 명령으로 관리
    - 네트워크를 설정하는 명령은 네트워크 관리자와 함께 설치되는 nmcli 명령이며 이와 별도로 리눅스가 제공하는 ip 명령을 사용할 수도 있음
    - nmcli 명령으로 유선 네트워크뿐만 아니라 와이파이 등 무선 네트워크, 보안 등 네트워크와 관련된 거의 모든 설정을 관리할 수 있음
    - nmcli는 명령 행에서 사용하는 명령은 물론이고 대화식 인터페이스도 제공
    - 형식

      nmcli [옵션] 명령 { 서브 명령 }
    - 옵션
      - ◆ -t: 실행 결과를 간단하게 출력
      - ◆ -p: 사용자가 읽기 좋게 출력
      - ◆ -v: nmcli의 버전을 출력
      - ◆ -h: 도움말을 출력

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ nmcli 명령으로 관리
    - ● 서브 명령
      - ◆ general {status I hostname}: 네트워크 관리자의 전체적인 상태를 출력하고 호스트 이름을 읽거나 변경할 수 있음
      - ◆ networking {on | off | connectivity}: 네트워크를 시작 및 종료하고 연결 상태를 출력
      - ◆ connection {show | up I down | modify | add | delete | reload | load }: 네트워크를 설정
      - ◆ device {status I show}: 네트워크 장치의 상태를 출력

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ nmcli 명령으로 관리
    - ● 네트워크의 전체 상태 살펴보기: nmcli general status
      - ◆ 네트워크의 전체적인 상태는 nmcli의 general 명령으로 확인할 수 있음
      - ◆ nmcli를 사용할 때 명령을 줄여서 사용할 수도 있는데 general 대신에 gen만 입력해도 됨
      - ◆ general 명령의 서브 명령인 status가 없어도 같은 결과를 출력

        STATE CONNECTIVITY WIFI-HW WIFI WWAN-HW WWAN

        disconnected unknown missing enabled missing enabled
      - ◆ connectivity가 출력하는 네트워크 상태는 다음 중 하나
        - ▪ none(없음): 호스트가 아직 네트워크에 연결되어 있지 않음
        - ▪ limited(제한적): 호스트가 네트워크에 연결되어 있지만 인터넷과 연결되지는 않음
        - ▪ full(전체): 호스트가 네트워크에 연결되어 있고 인터넷도 사용할 수 있음
        - ▪ unknown(알 수 없음): 네트워크 연결 상태를 알 수 없음

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ nmcli 명령으로 관리
    - ● 네트워크 활성화 또는 비활성화

```
user1@myubuntu : ~$ nmcli net con
        full
user1@myubuntu : ~$ nmcli net off
user1@myubuntu : ~$ nmcli net con
        none
user1@myubuntu : ~$ nmcli net on
user1@myubuntu : ~$ nmcli net con
        full
```

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ nmcli 명령으로 관리
    - ● 네트워크 설정: connection(con) 명령
      - ◆ 서브 명령
        - ▪ show: 메모리와 디스크에 저장된 네트워크 연결 프로파일을 출력하는데 서브 명령을 지정하지 않을 경우 기본적으로 show를 실행
        - ▪ up: 네트워크 연결을 시작
        - ▪ down: 네트워크 연결을 중지
        - ▪ modify: 연결 프로파일에서 속성을 추가 및 수정 그리고 삭제
        - ▪ add: 새로운 연결을 생성
        - ▪ delete: 연결의 설정을 삭제
        - ▪ reload: 연결과 관련된 파일을 디스크에서 다시 읽음
        - ▪ load: 디스크에서 하나 이상의 연결 파일을 읽음

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ nmcli 명령으로 관리
    - ● 네트워크 장치 상태 확인: device(dev) 명령
      - ◆ 네트워크 장치 상태 보기: nmcli dev status

```
DEVICE   TYPE       STATE       CONNECTION
enp0s8   ethernet   unmanaged   --
lo       loopback   unmanaged   --
```

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ nmcli 명령으로 관리
    - ● 네트워크 장치 상태 확인: device(dev) 명령
      - ◆ 네트워크 장치 상태 상세 보기: nmcli dev show

```
GENERAL.DEVICE:                         enp0s8
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         08:00:27:4C:65:BC
GENERAL.MTU:                            1500
GENERAL.STATE:                          10 (unmanaged)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         192.168.56.101/24
```

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ nmcli 명령으로 관리
    - ● 네트워크 장치 상태 확인: device(dev) 명령
      - ◆ 네트워크 특정 장치 상태 상세 보기: nmcli dev show enp0s8

```
GENERAL.DEVICE:                         enp0s8
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         08:00:27:4C:65:BC
GENERAL.MTU:                            1500
GENERAL.STATE:                          10 (unmanaged)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         192.168.56.101/24
```

네트워크 설정

- 네트워크 관리자
  - ✓ ip 명령으로 주소 관리
    - ip 명령으로 설정 가능하지만 재부팅하면 내용이 사라지므로 내용을 보존하기 위해서는 설정 파일에 저장해야 함
    - 명령
      - 형식: ip [옵션] 객체 [서브 명령]
      - 옵션
        - -V: 버전을 출력
        - -s: 자세한 정보를 출력
      - 객체 [서브명령]
        - address [add | del | show | help] 장치의 IP 주소를 관리
        - route [add l del | help]: 라우팅 테이블을 관리
        - link [set]: 네트워크 인터페이스를 활성화 또는 비활성화

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ ip 명령으로 주소 관리
    - ● 전체 장치에 대한 상세 정보 출력: 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 noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 08:00:27:4c:65:bc brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe4c:65bc/64 scope link
       valid_lft forever preferred_lft forever
```

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ ip 명령으로 주소 관리
    - ● 특정 장치에 대한 상세 정보 출력: ip addr show enp0s8

```
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 08:00:27:4c:65:bc brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe4c:65bc/64 scope link
       valid_lft forever preferred_lft forever
```

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ ip 명령으로 주소 관리
    - ip 주소 추가: sudo ip addr add 192.168.1.20/24 dev enp0s8
    - ip addr show enp0s8

```
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 08:00:27:76:14:eb brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet 192.168.1.20/24 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe76:14eb/64 scope link
       valid_lft forever preferred_lft forever
```

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ ip 명령으로 주소 관리
    - ip 주소 삭제: sudo ip addr del 192.168.1.20/24 dev enp0s8
    - ip addr show enp0s8

```
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 08:00:27:76:14:eb brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe76:14eb/64 scope link
       valid_lft forever preferred_lft forever
```

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ ip 명령으로 주소 관리
    - ● 라우팅 테이블과 게이트웨이 주소 관리: route 명령
      - ◆ ip route 명령은 라우팅 테이블을 출력하거나 게이트웨이를 설정하고 삭제
      - ◆ gateway
        - ▪ 네트워크를 다른 네트워크와 연결할 때 연결점이 되는 장치
        - ▪ 게이트웨이도 하나의 컴퓨터로 보통 라우터라고 부름
        - ▪ 게이트웨이는 패킷을 보고 같은 네트워크로 보내는 것이 아니면 외부로 전송
        - ▪ 게이트웨이 주소가 설정되어 있지 않으면 같은 네트워크가 아닌 컴퓨터와는 접속할 수 없음

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ ip 명령으로 주소 관리
    - ● 라우팅 테이블과 게이트웨이 주소 관리: route 명령
      - ◆ 라우팅 테이블 보기: ip route show

        default via 192.168.56.1 dev enp0s8 proto static

        192.168.56.0/24 dev enp0s8 proto kernel scope link src 192.168.56.101
        - ▪ 기본 게이트웨이 설정
          - default: 기본 게이트웨이를 설정
          - via 192.168.56.1: 패킷이 이 IP 주소를 통해 전송
          - dev enp0s8: 이 네트워크 인터페이스를 사용
          - proto static: 이 경로는 정적(static)으로 설정되어 있음
        - ▪ 서브넷 설정
          - 192.168.56.0/24: 이 네트워크의 주소 범위를 정의 (서브넷 마스크 255.255.255.0).
          - dev enp0s8: 이 네트워크 인터페이스를 사용
          - proto kernel: 커널에 의해 자동으로 생성된 경로
          - scope link: 로컬 링크에서 사용되는 경로
          - src 192.168.56.101: 패킷의 출발지 IP 주소

네트워크 설정

- 네트워크 관리자
  - ✓ ip 명령으로 주소 관리
    - ● 라우팅 테이블과 게이트웨이 주소 관리: route 명령
      - ◆ 기본 게이트웨이 설정: sudo ip route add default via 192.168.56.1 dev enp0s8
      - ◆ 192.168.2.0 네트워크를 192.168.147.2를 통해서 접속하도록 라우팅 경로 설정: sudo ip route add 192.168.2.0/24 via 192.168.147.2 dev enp0s8
      - ◆ 라우팅 경로 삭제: sudo ip route del 192.168.2.0/24

네트워크 설정

- 네트워크 관리자
  - ip 명령으로 주소 관리
    - 네트워크 인터페이스 관리 명령: link 명령
      - 네트워크 인터페이스 비활성화: sudo ip link set enp0s8 down
      - 네트워크 인터페이스 활성화: sudo ip link set enp0s8 up

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ ifconfig 명령으로 주소 관리
    - ● 설치: sudo apt install net-tools
    - ● 사용법
      - ◆ 형식: ifconfig [인터페이스명] [옵션] [값]
      - ◆ 옵션
        - ▪ -a: 시스템의 전체 인터페이스에 대한 정보를 출력
        - ▪ up/down: 인터페이스를 활성화 및 비활성화
        - ▪ netmask 주소: 넷마스크 주소를 설정
        - ▪ broadcast 주소: 브로드캐스트 주소를 설정

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ ifconfig 명령으로 주소 관리
    - ● 모든 네트워크 인터페이스 확인: ifconfig

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe76:14eb  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:76:14:eb  txqueuelen 1000  (Ethernet)
        RX packets 5706  bytes 2539233 (2.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4962  bytes 528698 (528.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 134  bytes 11693 (11.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 134  bytes 11693 (11.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ ifconfig 명령으로 주소 관리
    - ● 모든 네트워크 인터페이스 확인: ifconfig

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe76:14eb  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:76:14:eb  txqueuelen 1000  (Ethernet)
        RX packets 5706  bytes 2539233 (2.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4962  bytes 528698 (528.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 134  bytes 11693 (11.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 134  bytes 11693 (11.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

네트워크 설정

- 네트워크 관리자
  - ✓ ifconfig 명령으로 주소 관리
    - ● 모든 네트워크 인터페이스 확인: ifconfig
      - ◆ MAC 주소(ether): 08:00:27:76:14:eb
      - ◆ IP 주소(inet): 192.168.56.101
      - ◆ 넷마스크(netmask): 255.255.255.0
      - ◆ 브로드캐스트 주소(broadcast): 192.168.56.255
      - ◆ IPv6 주소(inet6): fe80::a00:27ff:fe76:14eb
      - ◆ RX: 부팅 후 현재까지 받은 패킷 수와 바이트 수
      - ◆ TX: 부팅 후 현재까지 보낸 패킷 수와 바이트 수

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ ifconfig 명령으로 주소 관리
    - ● 특정 네트워크 인터페이스 확인: ifconfig enp0s8

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe76:14eb  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:76:14:eb  txqueuelen 1000  (Ethernet)
        RX packets 5785  bytes 2546159 (2.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5084  bytes 551544 (551.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

네트워크 설정

- 네트워크 관리자
  - ✓ ifconfig 명령으로 주소 관리
    - 네트워크 인터페이스 비활성화: sudo ifconfig enp0s8 down
    - 네트워크 인터페이스 활성화: sudo ifconfig enp0s8 up
    - 네트워크 인터페이스 설정: sudo ifconfig enp0s8 192.168.147.129 netmask 255.255.255.0 broadcast 192.168.147.255

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ 게이트웨이 설정
    - ● 사용법
      - ◆ 형식: route [명령]
      - ◆ 명령
        - ▪ add: 라우팅 경로나 기본 게이트웨이 추가
        - ▪ del: 라우팅 경로나 기본 게이트웨이 제거

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ 게이트웨이 설정
    - ● 라우팅 테이블 보기: route

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    0      0        0 enp0s8
192.168.56.0    0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
```

      - ◆ Destination: 라우팅 대상 네트워크나 호스트의 주소
      - ◆ Gateway: 게이트웨이 주소 또는 설정되어 있지 않으면 * 출력
      - ◆ Genmask: 대상 네트워크의 넷마스크로 255.255.255.255 호스트이고 0.0.0.0 이면 default 경로
      - ◆ Metrics: 대상까지의 거리로 최근 커널에서는 사용되지 않지만 라우팅 데몬에서 사용할 수 도 있음
      - ◆ Ref: 해당 경로에 대한 참조 수이지만 리눅스 커널에서는 사용하지 않음
      - ◆ Iface: 패킷이 전달되는 인터페이스 이름

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ 게이트웨이 설정
    - ● 라우팅 테이블 보기: route
      - ◆ Flags
        - ▪ U: 경로활성화(UP)
        - ▪ H: 대상이 호스트
        - ▪ G: 게이트웨이로사용
        - ▪ 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

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ 게이트웨이 설정
    - 기본 게이트웨이 설정: sudo route add default gw 192.168.147.2 dev ens33
    - 기본 게이트웨이 삭제: sudo route del default gw 192.168.147.2

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ DNS 설정
    - DNS는 domain name service의 약자로 호스트 이름을 IP 주소로 바꾸는 역할을 수행
    - DNS가 설정되어 있지 않으면 이름으로 서버에 접속할 수 없으며 직접 IP 주소를 사용하여 접속해야 함
    - 17.04 버전부터 DNS를 관리하는 systemd-resolved 서비스를 도입
    - 리눅스는 DNS 서버의 주소를 /etc/resolv.conf 파일에 저장
    - 확인: cat /etc/resolv.conf

      nameserver 127.0.0.53

      options edns0 trust-ad

      search .

      - ◆ 127.0.0.53이 기본으로 설정되어 있지만 이 주소가 실제 DNS 주소는 아님
      - ◆ $₩text{Linux}$ 시스템에서는 /etc/resolv.conf 파일에 $₩text{DNS}$ 서버 목록이 직접 명시되지만 $₩text{systemd-resolved}$를 사용하는 $₩text{Ubuntu}$에서는 이 파일이 **로컬 스터브 주소($₩text{127.0.0.53}$)를 가리키는 심볼릭 링크** 인 경우가 많음

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ DNS 설정
    - ● DNS 정보 확인: resolvectl status명령으로 확인

```
Global
        Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
  resolv.conf mode: stub

Link 2 (enp0s8)
    Current Scopes: DNS
        Protocols: +DefaultRoute -LLMNR -mDNS -DNSOverTLS
DNSSEC=no/unsupported
Current DNS Server: 8.8.8.8
       DNS Servers: 8.8.8.8 8.8.4.4
```

- ◆ Global 섹션: 전역 $₩text{DNS}$ 설정을 보여줌
- ◆ Link 섹션에서는 각 네트워크 인터페이스($₩text{eth0}$, $₩text{wlan0}$ 등)별 $₩text{DNS}$ 설정을 보여주는데 $₩text{Current DNS Server}$: 현재 활발하게 사용되는 $₩text{DNS}$ 서버 주소이며 $₩text{DNS Servers}$: 해당 링크에 설정된 모든 $₩text{DNS}$ 서버 목록입니다.

네트워크 설정

- ❖ 네트워크 관리자
  - ✓ DNS 설정
    - nmcli 명령으로 DNS 정보 설정: nmcli con mod connection-name ipv4.dns DNS주소
    - DNS 서버에 질의(exit로 종료): nslookup

```
adam@server:~$ nslookup
> www.choongang.co.kr
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
www.choongang.co.kr canonical name = choongang.co.kr.
Name:   choongang.co.kr
Address: 35.213.149.53
```

네트워크 설정

- ❖ netplan
  - ✓ 개요
    - netplan은 Ubuntu에서 네트워크 구성을 관리하는 도구
    - 네트워크 구성을 변경하려면 netplan 설정 파일을 수정해서 할 수 있는데 netplan 설정 파일은 YAML 형식으로 작성되며 /etc/netplan/ 디렉토리에 존재하는데 파일 이름은 의미없음
    - 파일을 수정하고 적용할 때는 sudo netplan apply 또는 재부팅
    - 예시

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.56.101/24
      gateway4: 192.168.56.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

    - 일반 유저나 그룹에 대한 읽기/쓰기 권한을 제한하기 위한 권한 수정

```
sudo chmod 600 /etc/netplan/01-netcfg.yaml
```

Host Name 설정

- ❖ 호스트 이름 출력
  - ✓ uname
    - ● 명령어
      - ◆ 형식: uname [옵션]
      - ◆ 옵션
        - ▪ -m: 하드웨어 종류를 출력
        - ▪ -n: 호스트 이름을 출력
        - ▪ -r: 운영체제의 릴리즈 정보를 출력
        - ▪ -s: 운영체제의 이름을 출력
        - ▪ -v: 운영체제의 버전을 출력
        - ▪ -a: 위의 모든 정보
    - ● 명령어 사용
      - ◆ uname -n

        server
      - ◆ uname -a

        Linux server 6.8.0-79-generic #79-Ubuntu SMP PREEMPT_DYNAMIC Tue Aug 12 14:33:58 UTC 2025 aarch64 aarch64 aarch64 GNU/Linux

Host Name 설정

- ❖ 호스트 이름 출력 및 설정
  - ✓ hostname
    - ● 명령어
      - ◆ 형식: hostname [호스트 이름]
    - ● 명령어 사용
      - ◆ hostname

      server
      - ◆ sudo hostname ubuntu
      - ◆ hostname

      Ubuntu

Host Name 설정

- ❖ 호스트 이름 출력 및 설정
  - ✓ hostnamectl
    - ● 명령어
      - ◆ 형식: hostnamectl [옵션] [명령]
      - ◆ 옵션
        - ▪ -h: 도움말 출력
        - ▪ --version: 버전을 출력
      - ◆ 명령
        - ▪ status: 현재 호스트 이름과 관련 정보 출력
        - ▪ set-hostname: 호스트 이름을 설정

Host Name 설정

- ❖ 호스트 이름 출력 및 설정
  - ✓ hostnamectl
    - ● 명령어 사용
      - ◆ adam@server:~$ hostnamectl

```
   Static hostname: server
Transient hostname: ubuntu
         Icon name: computer
        Machine ID: 894d805ea846482e94d8c705a4e40b98
           Boot ID: b8fc6627ff784cf4b983abb8dad856b6
  Operating System: Ubuntu 24.04.3 LTS
            Kernel: Linux 6.8.0-79-generic
      Architecture: arm64
```

      - ◆ sudo hostnamectl set-hostname myubuntu
      - ◆ hostnamectl

```
 Static hostname: myubuntu
         Icon name: computer
        Machine ID: 894d805ea846482e94d8c705a4e40b98
           Boot ID: b8fc6627ff784cf4b983abb8dad856b6
Operating System: Ubuntu 24.04.3 LTS
            Kernel: Linux 6.8.0-79-generic
      Architecture: arm64
adam@server:~$
```

Host Name 설정

- 호스트 이름 파일에 저장
  - ✓ /etc/hostname
    - hostname 명령으로 호스트 이름이 바뀌기는 했지만 시스템을 재시작하면 원래 이름으로 돌아가는데 재시작해도 바뀐 호스트 이름이 유지되게 하려면 호스트 이름을 설정하는 파일 자체를 수정해야 함
    - 리눅스에서 호스트 이름을 저장하는 파일은 /etc/hostname
    - 명령어 사용
      - cat /etc/hostname

      myubuntu

Host Name 설정

- /etc/hosts
  - ✓ 개요
    - 도메인 이름(hostname)을 IP 주소에 수동으로 매핑하는 데 사용되는 텍스트 파일
    - DNS(Domain Name System) 조회를 거치지 않고 특정 호스트 이름을 IP 주소로 변환하는 데 사용되며 주로 로컬 시스템에서 가장 먼저 참조하는 파일
  - ✓ 용도
    - 로컬 호스트 및 개발: 웹 사이트 개발 시 아직 도메인에 연결되지 않은 개발 서버의 IP 주소로 특정 도메인 이름을 매핑하여 로컬에서 테스트 할 때 유용
    - 시스템 최적화: 자주 방문하는 호스트 이름을 DNS 조회 고정을 생략하여 약간의 속도 향상을 기대할 수 있음
    - 특정 사이트 차단: 악성 사이트나 광고 서버의 도메인 이름을 루프백 주소에 매핑하여 접속을 차단
    - 자체 호스트 이름 정의: 로컬 네트워크 환경에서 서버나 장치들의 이름을 편리하게 지정하고 접근하는데 사용
    - 형식: IP_주소 호스트이름 [별칭1, 별칭2...]

Host Name 설정

- /etc/hosts
  - ✓ cat /etc/hosts

```
127.0.0.1 localhost
127.0.1.1 original

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

네트워크 상태 확인

- ❖ 통신 확인
  - ✓ ping
    - 외부와 통신되는지 확인하거나 외부 서버가 동작하는지 확인할 때 사용
    - 명령어
      - ◆ 형식: ping [옵션] [목적지 주소]
      - ◆ 옵션
        - -a: 통신이 되면 소리가 남
        - -q: 테스트 결과를 지속적으로 보여주지 않고 종합 결과만 출력
        - -c 개수: 보낼 패킷 수를 지정
    - 시스템에 따라서 보안을 강화하기 위해 ping 패킷이 왔을 때 응답하지 않도록 설정하는 경우도 있기 때문에 ping 으로 연결되지 않는다고 해서 무조건 해당 시스템이 동작하지 않는다고 단정할 수 없음
    - 옵션없이 사용: ping www.google.com
      - ◆ 계속해서 패킷을 보냄

네트워크 상태 확인

- ❖ 통신 확인
  - ✓ ping
    - ● 패킷 수를 지정해서 사용: ping -c 5 www.google.com

```
PING www.google.com (142.250.71.132) 56(84) bytes of data.
64 bytes from nchkga-aa-in-f4.1e100.net (142.250.71.132): icmp_seq=1 ttl=112 time=41.6 ms
64 bytes from nchkga-aa-in-f4.1e100.net (142.250.71.132): icmp_seq=2 ttl=112 time=41.6 ms
64 bytes from nchkga-aa-in-f4.1e100.net (142.250.71.132): icmp_seq=3 ttl=112 time=42.2 ms
64 bytes from nchkga-aa-in-f4.1e100.net (142.250.71.132): icmp_seq=4 ttl=112 time=43.3 ms
64 bytes from nchkga-aa-in-f4.1e100.net (142.250.71.132): icmp_seq=5 ttl=112 time=41.4 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 41.427/42.030/43.312/0.699 ms
```

네트워크 상태 확인

- ❖ 통신 확인
  - ✓ ping
    - ● 패킷 수를 지정해서 사용: ping -c 5 www.google.com
      - ◆ 64 bytes from nchkga-aa-in-f4.1e100.net (142.250.71.132) 은 패킷의 크기(56B의 데이터에 8B의 헤더) 와 목적지의 IP
      - ◆ icmp_seq 는 응답이 보낸 ICMP(Internet Control Message Protocol) 패킷의 순서
      - ◆ ttl=112는 Time To Live의 약자로 패킷이 목적지에 도달하기까지 거친 라우터(Router) 수를 간접적으로 알려주는데 패킷이 처음 출발할 때 설정된 값(예: 128 또는 64)에서 라우터를 하나 거칠 때마다 1씩 감소하므로 이 값이 높을수록 목적지까지의 경로가 상대적으로 짧거나 안정적일 수 있음
      - ◆ time=41.4 ms는 왕복 시간으로 패킷이 출발지에서 목적지까지 갔다가 돌아오는데 걸린 시간
      - ◆ 5 packets transmitted, 5 received, 0% packet loss, time 4008ms: 5개의 패킷을 전송해서 5개의 패킷을 응답받았고 0%의 손실률이며 명령을 시작해서 모든 패킷의 전송 및 응답을 완료하는데 걸린 시간이 4008ms
      - ◆ rtt min/avg/max/mdev = 41.427/42.030/43.312/0.699 ms: 가장 빨랐던 왕복 시간 과 평균 그리고 가장 느렸던 왕복 시간 및 표준 편차

네트워크 상태 확인

- 통신 확인
  - ✓ netstat
    - 네트워크 연결 상태, 라우팅 테이블, 인터페이스 관련 통계 정보 등을 출력하며 현재 시스템에 열려 있는 포트가 무엇인지도 확인할 수 있는 명령
    - 명령어
      - 형식: netstat [옵션]
      - 옵션
        - -a: 모든 소켓 정보를 출력
        - -r: 라우팅 정보를 출력
        - -n: 호스트 이름 대신 IP 주소로 출력
        - -i: 모든 네트워크 인터페이스 정보를 출력
        - -s: 프로토콜별로 네트워크 통계 정보를 출력
        - -p: 해당 소켓과 관련된 프로세스의 이름과 PID를 출력

네트워크 상태 확인

- ❖ 통신 확인
  - ✓ netstat
    - ● 열려있는 포트 확인: netstat -an | grep LISTEN

      tcp 0 0 127.0.0.54:53 0.0.0.0:* LISTEN

      unix 2 [ACC] STREAM LISTENING 56436 /run/systemd/resolve/io.systemd.Resolve

      - ◆ tcp 0 0 127.0.0.54:53 0.0.0.0:* LISTEN 결과는 TCP 프로토콜을 사용하고 두번째 항목의 0은 수신 대기 큐의 크기로 애플리케이션에 의해 아직 처리되지 않은 패킷의 수 이며 세번째 0은 현재 원격 호스트로 전송되었지만 승인받지 못한 데이터의 양을 나타내며 다음 2개의 주소는 로컬 시스템의 주소와 포트 와 원격 시스템의 주소와 포트이며 마지막은 현재 상태로 연결 요청을 기다리고 있는 상태를 의미
      - ◆ unix 2 [ACC] STREAM LISTENING 56436 /run/systemd/resolve/io.systemd.Resolve 은 유닉스 소켓 상태를 해석하는 것

네트워크 상태 확인

- ❖ 통신 확인
  - ✓ netstat
    - ● 열려있는 포트를 사용 중인 프로세스 확인: netstat -p

```
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 myubuntu:ssh            192.168.200.72:54770    ESTABLISHED -
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   PID/Program name
Path
unix  3      [ ]         STREAM     CONNECTED     56396    -
unix  3      [ ]         STREAM     CONNECTED     56397    -
/run/systemd/journal/stdout
unix  2      [ ]         DGRAM      CONNECTED     66289    -
```

네트워크 상태 확인

- ❖ 통신 확인
  - ✓ netstat
    - ● 인터페이스 별 네트워크 통계 정보 확인: netstat -i

```
Kernel Interface table
Iface             MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP
TX-OVR Flg
enp0s8           1500   987734      0      0 0        251221      0      0      0 BMRU
lo              65536      627      0      0 0           627      0      0      0 LRU
```

네트워크 상태 확인

- 통신 확인
  - ✓ netstat
    - ● 프로토콜 별 네트워크 통계 정보 확인: netstat -s

```
Ip:
    Forwarding: 2
    542794 total packets received
    0 forwarded
    0 incoming packets discarded
    542794 incoming packets delivered
    250203 requests sent out
    100 outgoing packets dropped
    OutTransmits: 250203
```

네트워크 상태 확인

- MAC 주소 와 IP 주소 확인
  - ✓ arp(address resolution protocol)
    - 같은 네트워크에 연결된 시스템들의 MAC 주소 와 IP 주소를 확인하는 명령
    - 명령어 형식: arp [IP 주소]
    - arp

```
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   52:54:00:12:35:00   C                     enp0s8
```

네트워크 상태 확인

- 패킷 캡쳐 명령
  - ✓ tcpdump – 네트워크 상태를 확인하기 위해 패킷을 캡쳐하여 분석할 때 사용
  - ✓ 명령어
    - 형식: tcp [옵션]
    - 옵션
      - -c 패킷 수: 지정한 패킷 수만큼 덤프 받고 종료
      - -i 인터페이스명: 특정 인터페이스를 지정
      - -n: IP 주소를 호스트 이름으로 바꾸지 않음
      - -q: 정보를 간단한 형태로 출력
      - -X: 패킷의 내용을 16진수와 ASCII로 출력
      - -w 파일명: 덤프한 내용을 지정한 파일에 저장
      - -r 파일명: 덤프를 저장한 파일에서 읽어옴
      - host 호스트 이름 또는 주소: 지정한 호스트가 받거나 보낸 패킷만 덤프
      - port 번호: 지정한 포트 번호 패킷만 덤프
      - ip: IP 패킷만 덤프

네트워크 상태 확인

- 패킷 캡쳐 명령
  - ✓ 현재 시스템에서 주고받는 모든 패킷을 캡쳐: sudo tcpdump

```
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), snapshot length 262144 bytes
07:25:01.033754 IP 192.168.200.72.54770 > myubuntu.ssh: Flags [P.], seq
796473:796509, ack 3170331571, win 31724, length 36
07:25:01.035892 IP myubuntu.ssh > 192.168.200.72.54770: Flags [P.], seq 1:37, ack 36,
win 65535, length 36
```

네트워크 상태 확인

- 패킷 캡쳐 명령
  - ✓ 특정 포트로 송수신되는 패킷을 캡쳐: sudo tcpdump -c 3 port 22

```
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), snapshot length 262144 bytes
07:26:31.966291 IP myubuntu.ssh > 192.168.200.72.54770: Flags [P.], seq 3170378587:3170378695, ack 819429, win 65535, length 108
07:26:31.966534 IP myubuntu.ssh > 192.168.200.72.54770: Flags [P.], seq 108:144, ack 1, win 65535, length 36
07:26:31.966665 IP 192.168.200.72.54770 > myubuntu.ssh: Flags [.], ack 108, win 32032, length 0
3 packets captured
33 packets received by filter
0 packets dropped by kernel
```

네트워크 상태 확인

- 패킷 캡쳐 명령
  - ✓ 캡쳐한 내용을 ASCII로 보기: sudo tcpdump -Xc 3 port 22

```
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), snapshot length 262144 bytes
07:27:47.684734 IP myubuntu.ssh > 192.168.200.72.54770: Flags [P.], seq
3170419187:3170419295, ack 858813, win 65535, length 108
0x0000:  4510 0094 26b3 4000 4006 91a2 c0a8 3865  E...&.@.@.....8e
0x0010:  c0a8 c848 0016 d5f2 bcf8 c1f3 000d 1abd  ...H............
0x0020:  5018 ffff 8285 0000 8eff e7f4 5b1c 0089  P...........[...
0x0030:  f05a 27b5 5b49 96cb fee6 71d1 91c7 6472  .Z'.[I....q...dr
0x0040:  194e 3492 b15d d918 e5c1 5fe5 0a70 79b8  .N4..]...._..py.
0x0050:  5437 10fe 05ae 0a23 041e ced6 5b38 fb78  T7.....#....[8.x
0x0060:  6d14 080f dfff 0c3a 007d 2c58 753d 9da7  m.......:.},Xu=..
0x0070:  b2dd 8488 0201 a5d4 1c16 280e 8ff2 d453  ..........(....S
0x0080:  ad76 2500 47ec 77d0 cbc8 a632 dd68 ed9b  .v%.G.w....2.h..
0x0090:  ba3f 795e
```

방화벽 설정

- ❖ iptable
  - ✓ 개요
    - iptables는 리눅스 커널 내부의 패킷 필터링 프레임워크인 Netfilter(넷필터)를 제어하여 방화벽, NAT(네트워크 주소 변환), 패킷 변형 등을 수행하는 명령줄(CLI) 기반의 강력한 보안 도구
    - 서버로 들어오고 나가는 모든 네트워크 패킷을 검사하고, 관리자가 설정한 규칙(Rule)에 따라 패킷을 통과시킬지, 차단할지, 혹은 다른 곳으로 전달할지 결정하는 네트워크 게이트키퍼 역할을 수행

방화벽 설정

- iptable
  - ✓ 개요
    - 장점
      - 커널 레벨의 고성능: 유저 공간이 아닌 리눅스 커널 레이어(Netfilter)에서 직접 패킷을 제어하므로 처리 속도가 매우 빠르고 자원을 적게 사용
      - 극도의 세밀함: 포트, IP뿐만 아니라 패킷의 상태(Connection Tracking), 네트워크 인터페이스 카드(NIC), MAC 주소, 패킷 문자열 감지 등 매우 정교한 조건 생성이 가능
      - 풍부한 레거시 인프라: 수십 년간 리눅스 표준 방화벽으로 군용, 기업용 인프라에서 검증되어 관련 레퍼런스와 스크립트 툴이 방대
    - 단점
      - 높은 진입 장벽: 명령어가 복잡하고 가독성이 떨어져서 규칙 하나를 잘못 추가하면 전체 네트워크가 마비되거나 접속이 끊길 수 있음
      - 순서 의존성: 규칙을 위에서부터 아래로 순서대로 검사하기 때문에 규칙 간의 우선순위(상하 관계)를 완벽히 이해하고 작성해야 함
      - 휘발성 메모리 작동: 명시적으로 별도 파일에 저장(iptables-save)하지 않으면 서버를 재부팅할 때 설정한 규칙이 모두 날아감

방화벽 설정

- iptable
  - ✓ iptables의 구조: 테이블 과 체인
    - 3대 주요 테이블(Tables): 패킷을 목적에 따라 분류
      - Filter(기본 테이블): 패킷을 통과시킬지(허용), 버릴지(차단) 결정하는 방화벽의 핵심 역할
      - NAT (Network Address Translation): 패킷의 출발지 IP/포트나 목적지 IP/포트를 다른 것으로 변환(공유기 기능이나 포트 포워딩에 사용)
      - Mangle: 패킷 헤더의 특정 정보(TTL, TOS 등)를 마킹하거나 미세하게 수정할 때 사용
    - 5대 체인(Chains): 패킷이 네트워크 카드를 통해 들어오고 나가는 과정에서 거치는 체크포인트
      - INPUT: 외부에서 서버 자체를 목적지로 들어오는 패킷이 거치는 통로(예: 내 웹 서버로 접속하는 트래픽)
      - OUTPUT: 서버 자체에서 외부로 나가는 패킷이 거치는 통로(예: 서버에서 apt update를 요청할 때)
      - FORWARD: 서버가 목적지가 아니라, 서버를 단순히 거쳐서 다른 곳으로 전달되는 패킷이 거치는 통로(라우터/공유기 역할 수행 시 사용)
      - PREROUTING: 패킷이 네트워크 카드에 도착하자마자 라우팅 경로를 결정하기 전에 거치는 통로(주로 NAT 목적지 변경 시 사용)
      - POSTROUTING: 패킷이 라우팅을 마치고 네트워크 카드를 통해 밖으로 나가기 직전에 거치는 통로(주로 NAT 출발지 변경 시 사용)

방화벽 설정

- iptable
  - ✓ 패킷 처리 규칙(Targets): 패킷이 특정 체인의 규칙에 매칭되었을 때 수행할 작업
    - ACCEPT: 패킷을 허용하고 통과
    - DROP: 패킷을 완전히 차단(보낸 이에게 차단되었다는 응답조차 주지 않고 증발시킴 - 보안상 추천)
    - REJECT: 패킷을 차단하되 보낸 이에게 거부되었다는 안내 메시지(ICMP 에러)를 되돌려 줌
    - LOG: 패킷의 정보를 시스템 로그(/var/log/syslog)에 기록(패킷 자체는 통과됨)

방화벽 설정

- iptable
  - ✓ 명령어 사용
    - 규칙 조회 및 상태 확인
      - 기본 filter 테이블의 규칙을 번호와 함께 상세히 조회: sudo iptables -L -n -v --line-numbers
      - NAT 테이블의 규칙 조회: sudo iptables -t nat -L -n --line-numbers
        - -L: 규칙 리스트 출력
        - -n: IP 주소와 포트를 도메인/서비스 이름 대신 숫자로 표시(조회 속도가 빨라짐)
        - -v: 패킷 수, 바이트 수 등 상세 정보 표시
        - --line-numbers: 규칙 앞에 행 번호를 붙여줌(삭제할 때 유용)

방화벽 설정

- iptable
  - ✓ 명령어 사용
    - 기본 정책(Policy) 변경: 규칙에 매칭되지 않은 나머지 모든 패킷을 어떻게 처리할지 기본 방침
      - 들어오는 패킷은 기본적으로 모두 차단(화이트리스트 방식 방화벽의 기본): sudo iptables -P INPUT DROP
      - 나가는 패킷은 기본적으로 모두 허용: sudo iptables -P OUTPUT ACCEPT
      - 거쳐가는 패킷은 기본적으로 모두 차단: sudo iptables -P FORWARD DROP

방화벽 설정

- iptable
  - ✓ 명령어 사용
    - 방화벽 규칙 추가
      - ◆ 옵션
        - -A (Append): 체인의 맨 아래에 규칙을 추가
        - -I (Insert): 체인의 맨 위에 규칙을 삽입(iptables는 위에서부터 규칙을 읽으므로 중요)
      - ◆ 사용
        - 로컬 호스트(loopback) 트래픽 허용(시스템 내부 통신용 필수 설정): sudo iptables -A INPUT -i lo -j ACCEPT
        - 이미 연결되었거나 연관된 패킷은 자동으로 허용(원격 접속 끊김 방지 필수 설정): sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
        - 특정 포트 허용 (SSH 22번 포트 허용): sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
        - 웹 서버 포트(80, 443) 허용
          - sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
          - sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
        - 특정 IP 주소 차단 (악성 IP 1.2.3.4 차단): sudo iptables -A INPUT -s 1.2.3.4 -j DROP

방화벽 설정

- iptable
  - ✓ 명령어 사용
    - 방화벽 규칙 삭제
      - 특정 체인의 번호 기반 삭제(INPUT 체인의 3번째 규칙 삭제): sudo iptables -D INPUT 3
      - 모든 테이블의 모든 규칙 일시적으로 전부 삭제(초기화): sudo iptables –F
    - 포트포워딩: iptables를 사용하면 특정 포트로 들어온 트래픽을 다른 IP나 포트로 토스 해줄 수 있음(외부에서 80포트로 오면 내부 8080포트로 전달)
      - sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
  - ✓ iptables 규칙 영구 저장: iptables 명령어로 설정한 규칙은 메모리에만 기록되기 때문에, 서버를 재부팅하면 모두 사라지므로 재부팅 후에도 유지되게 하려면 저장 도구를 사용해야 함
    - 저장용 패키지 설치(설정 저장 여부를 묻는 창이 뜨면 Yes 선택): sudo apt install iptables-persistent
    - 현재 설정된 iptables 규칙을 파일로 저장: sudo iptables-save | sudo tee /etc/iptables/rules.v4
    - 나중에 수동으로 불러와야 할 때: sudo iptables-restore < /etc/iptables/rules.v4

방화벽 설정

- ❖ iptable
  - ✓ 현대 리눅스에서의 위치(UFW / nftables 와의 관계)
    - 최신 리눅스 환경에서는 iptables의 복잡함을 해결하기 위해 다양한 대안이 함께 사용되고 있음
    - UFW (Uncomplicated Firewall): Ubuntu 등에서 주로 쓰이며, iptables가 너무 복잡해서 초보자도 쉽게 다룰 수 있도록 명령어 체계를 직관적으로 래핑(Wrapping)해 놓은 단순화 도구(백엔드에서는 결국 iptables 규칙으로 치환되어 작동)
    - nftables: 현대 리눅스 커널(구현체에 따라 다름)에서는 구조적 한계가 있는 iptables를 대체하기 위해 차세대 패킷 필터링 프레임워크인 nftables를 개발하여 기본 탑재하고 있는데 문법이 훨씬 깔끔하고 효율적이지만 여전히 하위 호환성을 위해 iptables 명령어를 입력하면 nftables 규칙으로 자동 변환되는 브릿지 툴(iptables-nft)이 작동하는 경우가 많음

방화벽 설정

- ufw
  - ✓ 개요
    - UFW(Uncomplicated Firewall)는 이름 그대로 복잡하지 않은 방화벽이라는 뜻을 가진 Ubuntu 리눅스의 기본 방화벽 관리 도구
    - 리눅스 커널의 강력하지만 악명 높을 정도로 복잡한 방화벽 도구인 iptables를 사용자가 직관적이고 쉽게 다룰 수 있도록 래핑(Wrapping)하여 명령어를 단순화한 것이 특징
    - 리눅스 자체 방화벽 엔진인 iptables는 패킷의 흐름에 따라 테이블과 체인을 완벽히 이해해야 하고 명령어가 매우 길고 복잡해서 규칙 하나를 잘못 넣으면 관리자 본인의 SSH 접속마저 끊어버리는 대참사가 자주 일어나서 이러한 진입 장벽을 낮추기 위해 개발됨
    - 백엔드에서는 여전히 iptables 규칙을 생성하여 커널에 전달하지만 사용자는 sudo ufw allow 80 같이 단 한 줄의 직관적인 명령어로 방화벽을 제어할 수 있게 됨
  - ✓ UFW의 핵심 작동 방식
    - UFW는 기본적으로 화이트리스트(Whitelist) 방식의 보안 정책을 쉽게 수립할 수 있도록 설계됨
    - 기본 정책 (Default Policy): 외부에서 서버로 들어오는 모든 접근(Incoming)은 일단 전부 차단(Deny)하고 서버 내부에서 외부로 나가는 통신(Outgoing)은 전부 허용(Allow)하는 것이 기본 상태
    - 예외 허용: 기본 정책이 모두 차단이므로 관리자는 웹 서버(80, 443)나 원격 제어(SSH, 22)처럼 안전하고 필요한 포트만 콕 집어서 구멍을 뚫어주는(Allow) 방식으로 운영

방화벽 설정

- ufw
  - ✓ UFW의 주요 특징 및 장점
    - 인간 친화적인 명령 체계
      - 포트 번호를 몰라도 대중적인 서비스 이름(ssh, http, https)을 입력하면 UFW가 알아서 해당 포트를 찾아 규칙을 적용
      - iptables 예시: iptables -A INPUT -p tcp --dport 22 -j ACCEPT
      - UFW 예시: ufw allow ssh (또는 ufw allow 22)
    - 규칙 관리의 용이성(번호 기반 관리): 현재 적용된 방화벽 규칙들을 번호 순으로 정렬해서 보고 지우고 싶은 규칙의 번호만 지정해 간편하게 삭제할 수 있음(ufw status numbered -> ufw delete [번호])
    - 응용 프로그램 프로필(Application Profile) 지원: Nginx, Apache, OpenSSH 등 유명 패키지를 설치하면 해당 프로그램이 사용하는 포트 정보를 담은 프로필이 UFW에 자동으로 등록되며 관리자는 포트 번호를 일일이 외울 필요 없이 ufw allow Nginx Full 같은 명령으로 관련 포트를 통째로 제어할 수 있음
    - 다중 조건 조합 기능: 단순 포트 개방을 넘어 특정 IP 주소(192.168.x.x)가 TCP 프로토콜로 22번 포트에 접근할 때만 허용한다 와 같은 정교한 방화벽 규칙도 직관적인 문법으로 조합해 작성할 수 있음

방화벽 설정

- ufw
  - ✓ 명령어
    - UFW 상태 확인 및 활성화/비활성화
      - 상태 및 현재 규칙 확인: sudo ufw status verbose
      - UFW 방화벽 활성화: sudo ufw enable
      - 원격(SSH)으로 서버에 접속 중인 경우 SSH 포트(기본 22번)를 허용하지 않은 채 UFW를 켜면 접속이 즉시 끊기고 서버가 차단될 수 있으므로 반드시 SSH 허용 명령어를 먼저 실행하거나 활성화 시 경고 문구가 뜨면 SSH 포트가 열려있는지 확인
      - UFW 방화벽 비활성화: sudo ufw disable
    - 기본 정책(Default Policy) 설정
      - UFW는 일일이 규칙을 지정하지 않은 패킷에 대해 어떻게 처리할지 기본 정책을 정할 수 있습니다. 가장 안전하고 일반적인 표준 설정은 다음과 같음
      - 들어오는 모든 접속 차단 (기본값): sudo ufw default deny incoming
      - 나가는 모든 접속 허용 (기본값): sudo ufw default allow outgoing
      - 이렇게 설정해 두면 외부에서 서버로 해킹 시도는 막으면서 서버 내부에서 apt update나 웹 서핑을 위해 외부로 나가는 트래픽은 자유롭게 허용됨

방화벽 설정

- ufw
  - ✓ 명령어
    - 방화벽 규칙 추가 (허용 및 차단)
      - ◆ 포트 번호 기반 설정
        - 특정 포트 허용 (웹 서버 80번 포트): sudo ufw allow 80
        - 프로토콜 지정 허용(TCP 443번 포트만 허용): sudo ufw allow 443/tcp
        - 특정 포트 차단(구형 FTP 21번 포트 차단): sudo ufw deny 21
        - 포트 범위 허용(3000번부터 3010번 포트까지 한 번에 허용): sudo ufw allow 3000:3010/tcp
      - ◆ 서비스 이름 기반 설정: /etc/services 파일에 등록된 대중적인 서비스 이름을 포트 번호 대신 사용할 수 있음
        - SSH(22번 포트) 허용: sudo ufw allow ssh
        - HTTP 및 HTTPS 허용
          - sudo ufw allow http
          - sudo ufw allow https

방화벽 설정

- ufw
  - ✓ 명령어
    - 방화벽 규칙 추가 (허용 및 차단)
      - ◆ 특정 IP 주소 기반 설정
        - 특정 IP 주소의 접근만 허용(예: 내 PC IP가 192.168.0.50일 때): sudo ufw allow from 192.168.0.50
        - 특정 IP가 특정 포트에만 접근할 수 있도록 허용(가장 추천하는 SSH 보안 설정): sudo ufw allow from 192.168.0.50 to any port 22 proto tcp
        - 특정 서브넷(네트워크 대역) 전체 허용: sudo ufw allow from 192.168.0.0/24

방화벽 설정

- ufw
  - ✓ 명령어
    - 방화벽 규칙 삭제 및 초기화
      - 번호 기반 삭제 (가장 편리함)
        - 현재 등록된 규칙들을 앞에 번호([1], [2])를 붙여서 출력: sudo ufw status numbered
        - 삭제하고 싶은 규칙의 번호를 지정하여 삭제(2번 규칙 삭제): sudo ufw delete 2
      - 등록했던 명령어 그대로 삭제: 추가할 때 썼던 명령어 앞에 delete만 붙이면 됨
        - sudo ufw delete allow 80
        - sudo ufw delete allow from 192.168.0.50 to any port 22 proto tcp
      - 방화벽 전체 초기화(Reset): 설정이 너무 꼬여서 처음에 덤벼들었을 때의 깨끗한 상태로 되돌리고자 하는 경우
        - sudo ufw reset

방화벽 설정

- ufw
  - ✓ 명령어
    - ● 예시
      - ◆ SSH 포트 먼저 열기(팅김 방지): sudo ufw allow 22/tcp
      - ◆ 웹 서버 포트 열기
        - ▪ sudo ufw allow 80/tcp
        - ▪ sudo ufw allow 443/tcp
      - ◆ 프록시(Squid 등) 서버 포트 열기(필요시): sudo ufw allow 3128/tcp
      - ◆ 기본 정책 확인 후 방화벽 켜기
        - ▪ sudo ufw default deny incoming
        - ▪ sudo ufw default allow outgoing
        - ▪ sudo ufw enable
      - ◆ 최종 결과 확인: sudo ufw status verbose