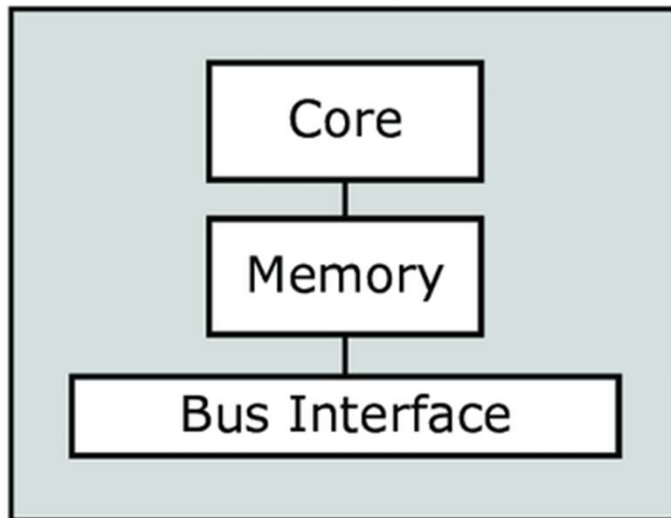


1. 가상 머신 환경

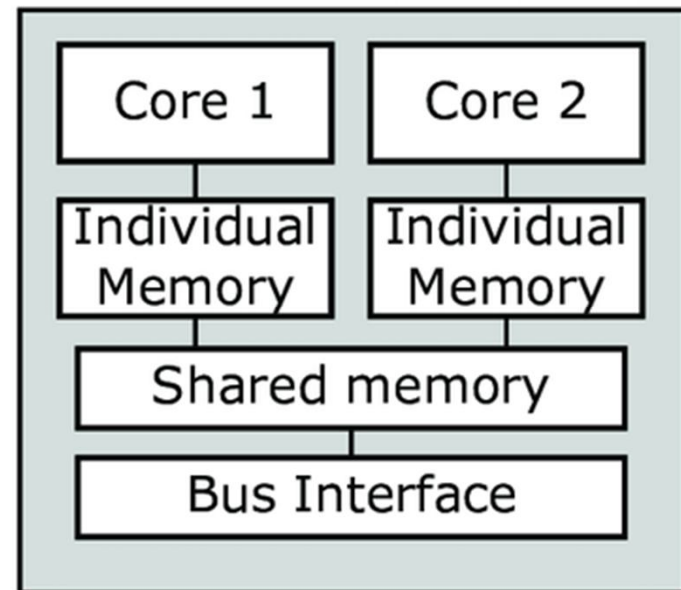
1 Processor

Processor



Single Core

Processor

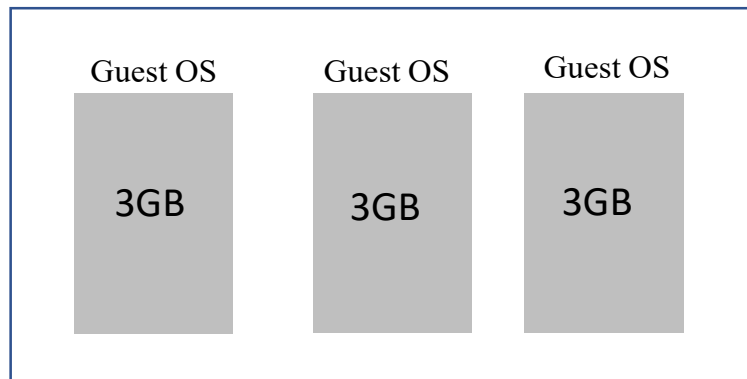


Multi-Core(Dual Core)

② RAM

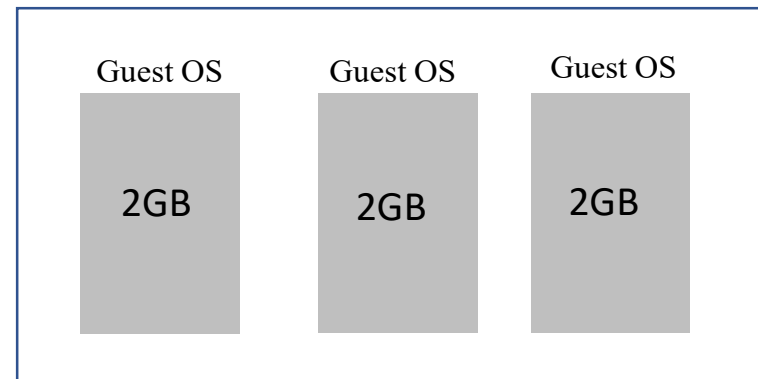
- 메모리의 할당은 가상머신을 만든 시점이 아니라 가상머신 ‘부팅’ 시 할당됨
- 게스트 컴퓨터에 모든 메모리를 할당하는 것은 바람직하지 않음

(예) 현재 컴퓨터의 RAM이 9GB인 경우



Host OS(0B)

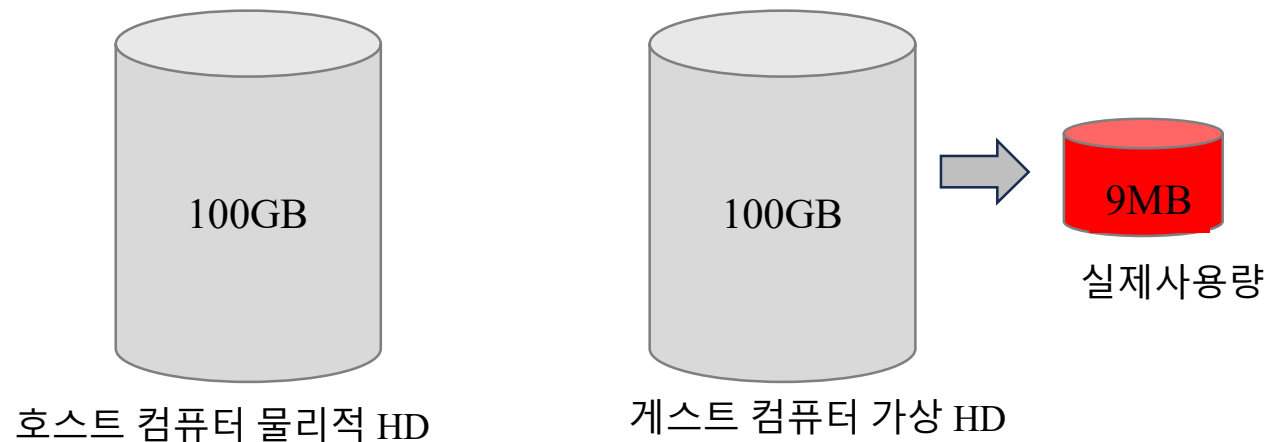
(예) 현재 컴퓨터의 RAM이 9GB인 경우



Host OS(3GB)

③ 하드디스크

- 게스트 컴퓨터의 하드디스크는 하나의 파일로 처리함
- 게스트 컴퓨터는 고정된 크기의 하드디스크를 할당 받는 것이 아님
 - 사용에 따라 하드 디스크를 가변적인 크기를 갖게 됨
 - 할당 시 설정된 최대 크기를 넘지는 못함



<<참고>>

- Debian계열
 - Debian
 - Ubuntu
- RedHat계열
 - Redhat
 - RHEL
 - Fedora – RHEL
 - CentOS

<<RedHat계열 변화 >>

- Fedora – CentOS Stream – RHEL

*CentOS Stream : RHEL의 신기능을 추가하기 위한 베타 버전으로 제작

* Rocky Linux (2021) : CentOS 대체 리눅스

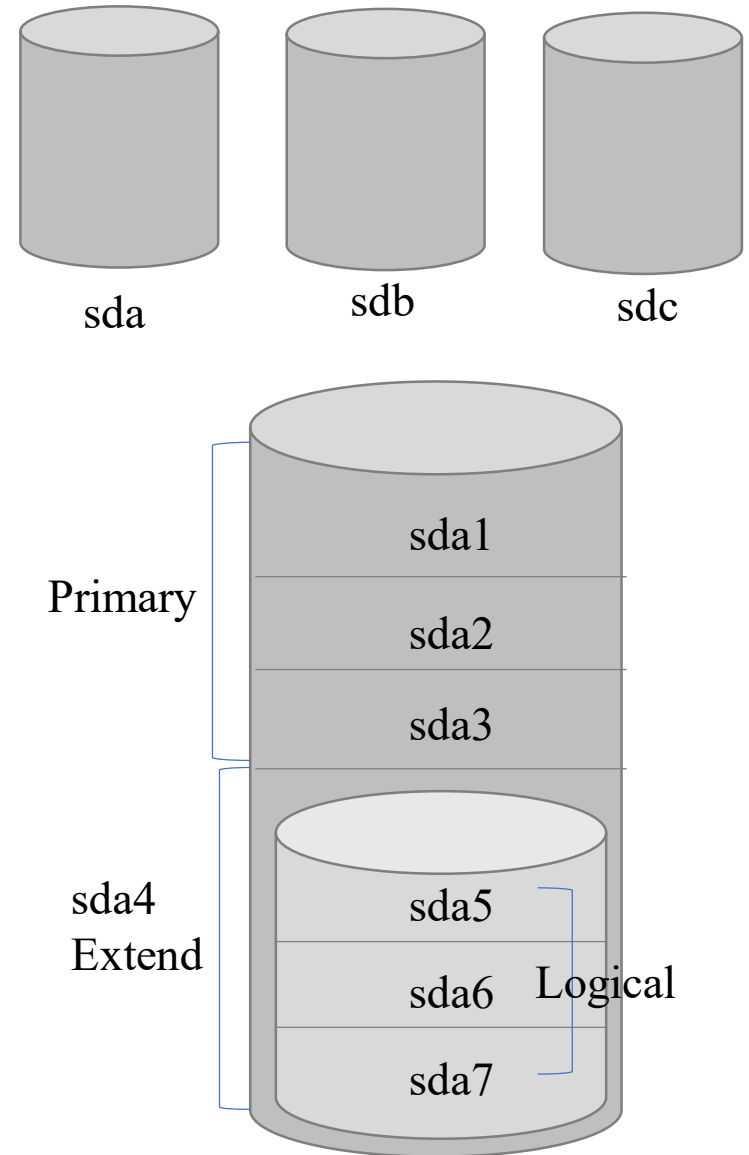
2. Linux 설치 시

1 디스크 분할

/dev/sd a 3

① ② ③

①	<ul style="list-style-type: none">· 하드 디스크 유형 지정- sd : SCSI또는 USB 방식 디스크
②	<ul style="list-style-type: none">· 한 케이블에 묶여진 하드 디스크 우선순위를 정함- 첫 번째 하드 디스크 : a- 두 번째 하드 디스크 : b
③	<ul style="list-style-type: none">· 파티션 번호- 1번에서 4번 : primary 또는 extended- 5번부터 : logical 파티션



Primary Partition 1	Primary Partition 2	Primary Partition 3	Primary Partition 4
---------------------	---------------------	---------------------	---------------------

주 파티션(Primary Partition)

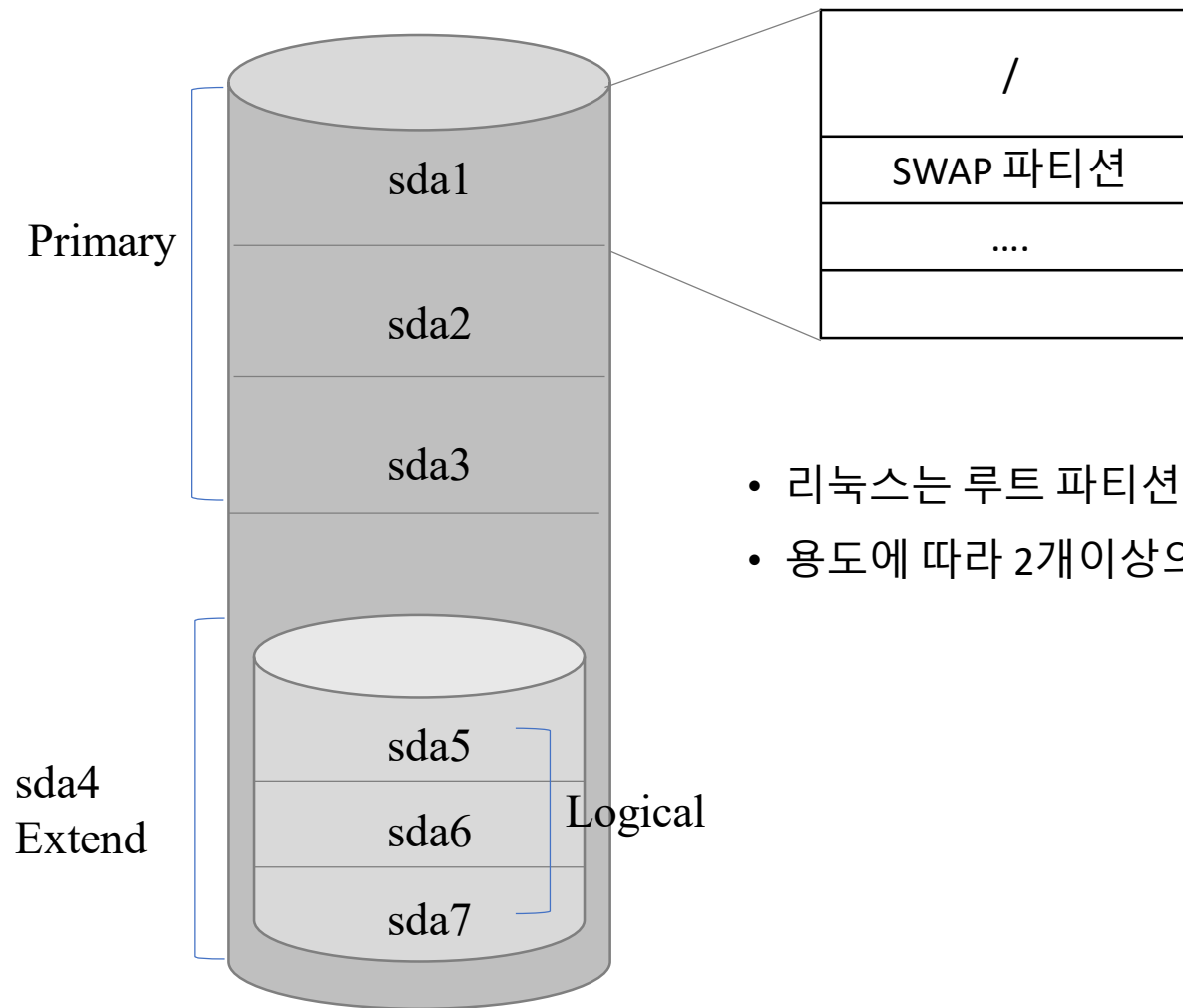
- 부팅이 가능한 기본 파티션
- 하나의 하드디스크에 최대 4개의 주 파티션 분할 가능

Primary Partition 1	Primary Partition 2	Primary Partition 3	Extended Partition 1
---------------------	---------------------	---------------------	----------------------

확장 파티션(Extended Partition)

- HDD를 여러 개의 파티션으로 나누고자 할 때 만드는 파티션
- 하나의 물리적 디스크에 1개만 생성
- 데이터 저장 영역을 위한 것이 아니라 논리 파티션을 생성

Logical Partition 1	Logical Partition 12
---------------------	------	------	----------------------



- 리눅스는 루트 파티션(/)와 swap 파티션은 설치 시 필수
- 용도에 따라 2개 이상의 파티션을 나눠 작업 가능

② 스왑(Swap) 파티션

- 하드디스크의 일부를 메모리처럼 사용하는 영역
- 주 파티션 또는 논리 파티션에 생성
- 메모리의 공간 부족 시 디스크의 일부분을 메모리로 사용되는 영역
- 리눅스 설치 시에 반드시 설치되어야 하는 영역
- 스왑 영역의 크기는 메모리의 2배를 설정하도록 권고
- $SWAP \text{ 영역} = RAM * 2$

(예) RAM 이 2GM인 경우

$$SWAP \text{ 영역} = 2048(2GB) * 2 = 4096$$

3 디렉터리

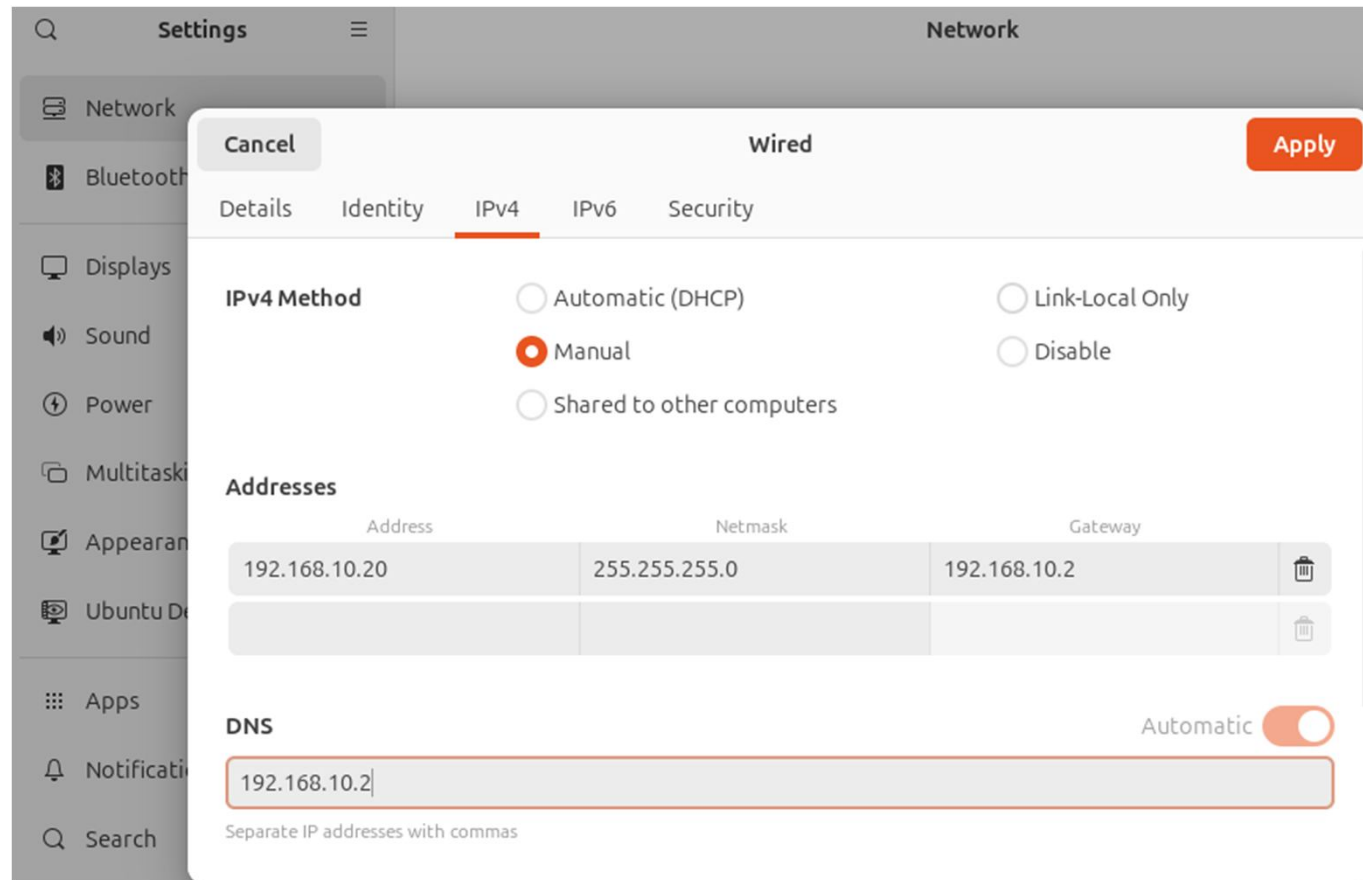
- 운영 용도에 따라 분할해서 사용 가능

마운트 포인터	설명	마운트 포인터	설명
/	루트 파티션	/boot	부팅 커널 저장
/bin	기본 명령어	/media	외부 장치 마운트 제공
/sbin	시스템 관리 명령어	/proc	프로세스에 대한 정보 저장 실제로는 빈 디렉토리이며 시스템이 부팅되면서 시스템의 프로세스 정보가 저장
/etc	환경설정 관련 파일	/tmp	임시 파일 저장
/dev	장치 파일 저장	/lost+found	파일 시스템 복구용 Fsck로 점검 후 깨진 file이 이 디렉토리 내에서 생성
/usr	응용 프로그램 저장	/home	사용자별 공간
/var	로그, 캐시 파일 등	swap	RAM 부족 시 사용

3. Ubuntu 24.04 환경 설정

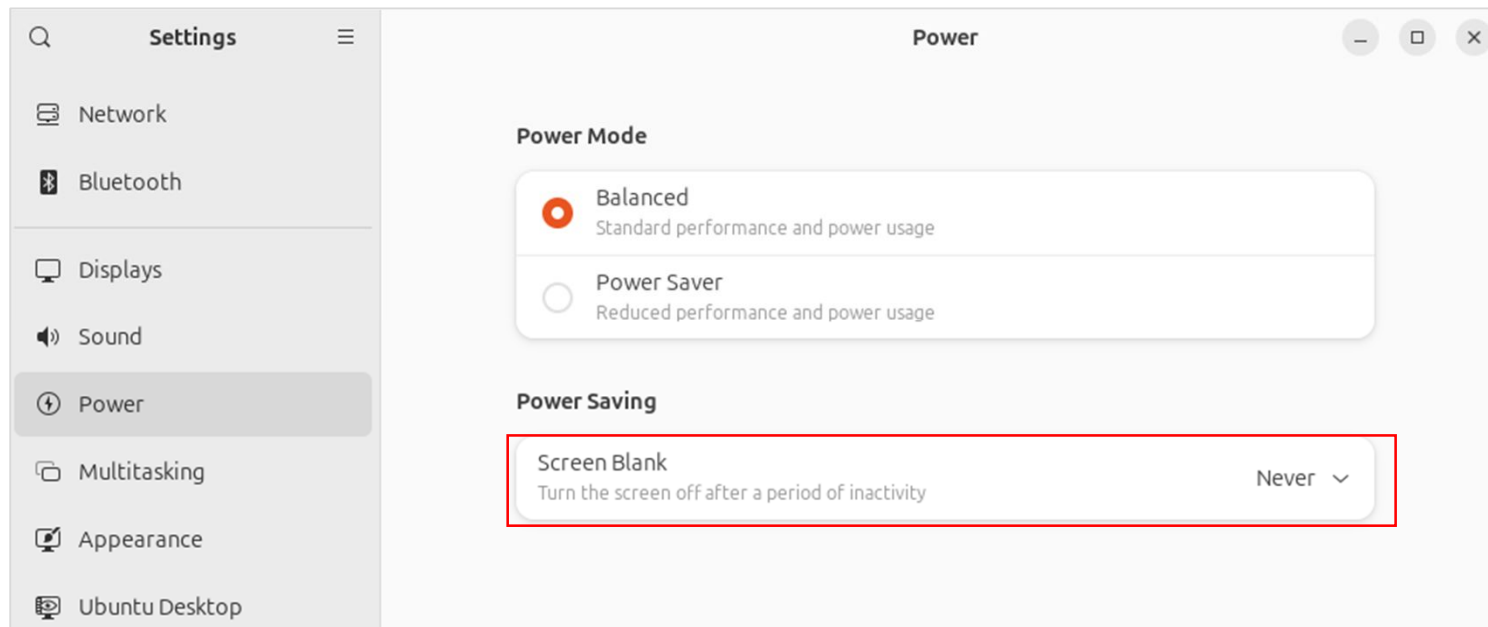
1 네트워크 설정

- Settings > network



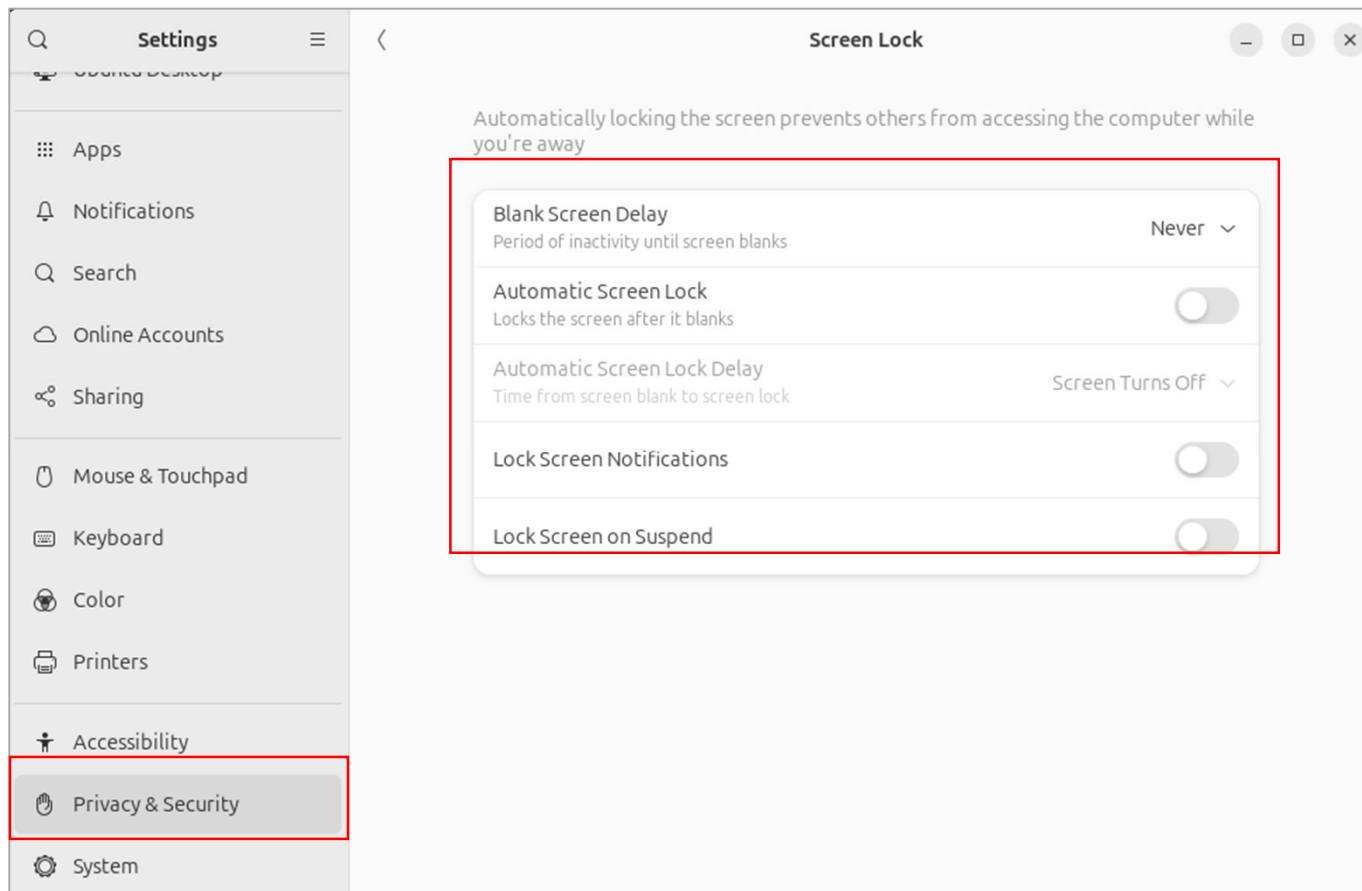
② 빈 화면(screen blank) 비활성화

- Settings > Power



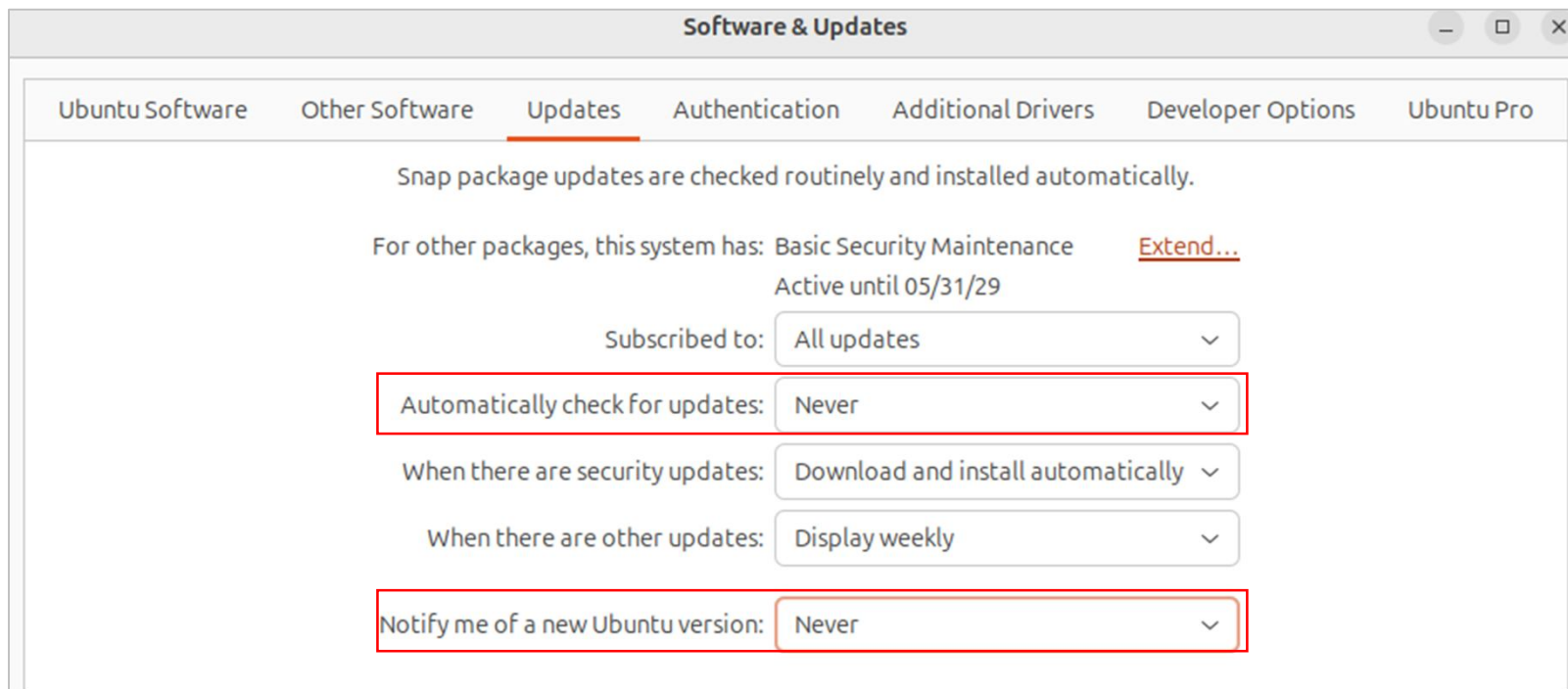
③ 화면 잠금(screen lock) 비활성화

- Settings > Privacy & Security



④ 소프트웨어 업데이트 기능 비활성화

- Software & Updates



5 패키지 업데이트

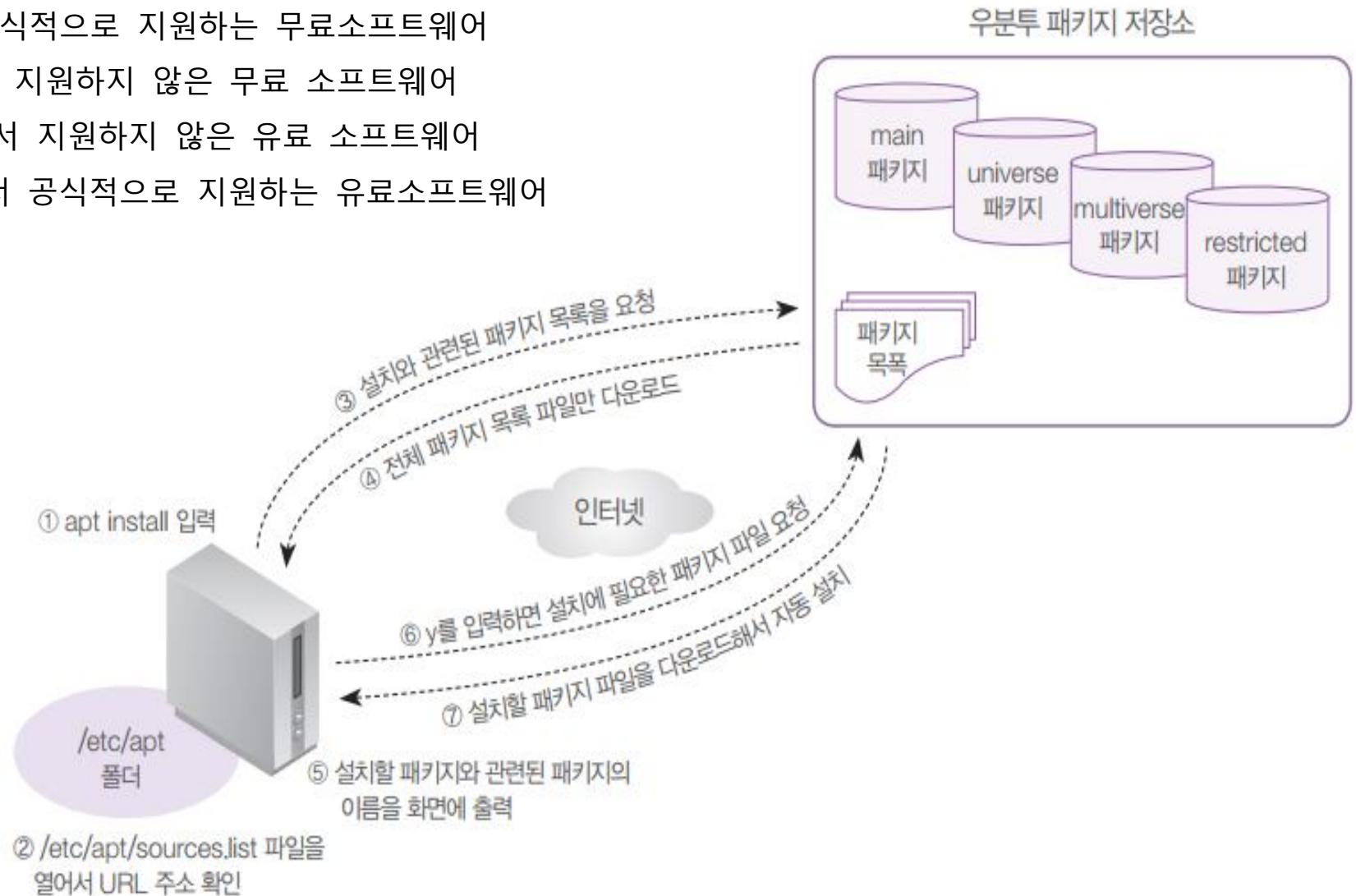
#cat /etc/apt/sources.list.d/ubuntu.sources

```
root@splunk:/etc/apt/sources.list.d# ls
ubuntu.sources  ubuntu.sources.curtin.orig
root@splunk:/etc/apt/sources.list.d# cat ubuntu.sources
Types: deb
URIs: http://archive.ubuntu.com/ubuntu/
Suites: noble noble-updates noble-backports
Components: main restricted universe multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

Types: deb
URIs: http://security.ubuntu.com/ubuntu/
Suites: noble-security
Components: main restricted universe multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg
root@splunk:/etc/apt/sources.list.d#
```

#apt update

- Main : 우분투에서 공식적으로 지원하는 무료소프트웨어
- Universe : 우분투에서 지원하지 않은 무료 소프트웨어
- Multiverse : 우분투에서 지원하지 않은 유료 소프트웨어
- Restricted : 우분투에서 공식적으로 지원하는 유료소프트웨어



⑥ 네트워크 툴 설치

#apt install -y net-tools

```
root@splunk:/# apt install -y net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 194 not upgra
Need to get 204 kB of archives.
After this operation, 811 kB of additional disk space will b
Get:1 http://archive.ubuntu.com/ubuntu noble/main amd64 net-
Fetched 204 kB in 2s (87.3 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 161234 files and directories currently
Preparing to unpack .../net-tools_2.10-0.1ubuntu4_amd64.deb
Unpacking net-tools (2.10-0.1ubuntu4) ...
Setting up net-tools (2.10-0.1ubuntu4) ...
Processing triggers for man-db (2.12.0-4build2) ...
```

#ifconfig

```
root@splunk:/# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.10.10  netmask 255.255.255.0  broadcast 192.168.10.255
    ether 00:0c:29:0a:aa:90  txqueuelen 1000  (Ethernet)
    RX packets 24909  bytes 36537031 (36.5 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 2366  bytes 195972 (195.9 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

7 방화벽 설정

```
#ufw enable
```

```
#ufw allow 80/tcp
```

```
#ufw status
```

```
<<Rule 삭제 또는 방화벽 비활성화>>
```

```
#ufw disable
```

```
#ufw delete allow 80/tcp
```

⑧ SSH 설치

```
#apt install -y openssh-server
```

```
#nano /etc/ssh/sshd_config
```

Port 22

```
#systemctl restart ssh
```

```
#systemctl status ssh
```

```
#ufw allow 22/tcp
```

```
#ufw status
```

```
root@zeekids:~# cat /etc/ssh/sshd_config

# This is the sshd server system-wide configuration file.
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

# The strategy used for options in the default sshd_config file is to allow
# OpenSSH to specify options with their default values where possible, but
# leave them commented. Uncommented options specify values overriding the
# default value.

Include /etc/ssh/sshd_config.d/*.conf
```

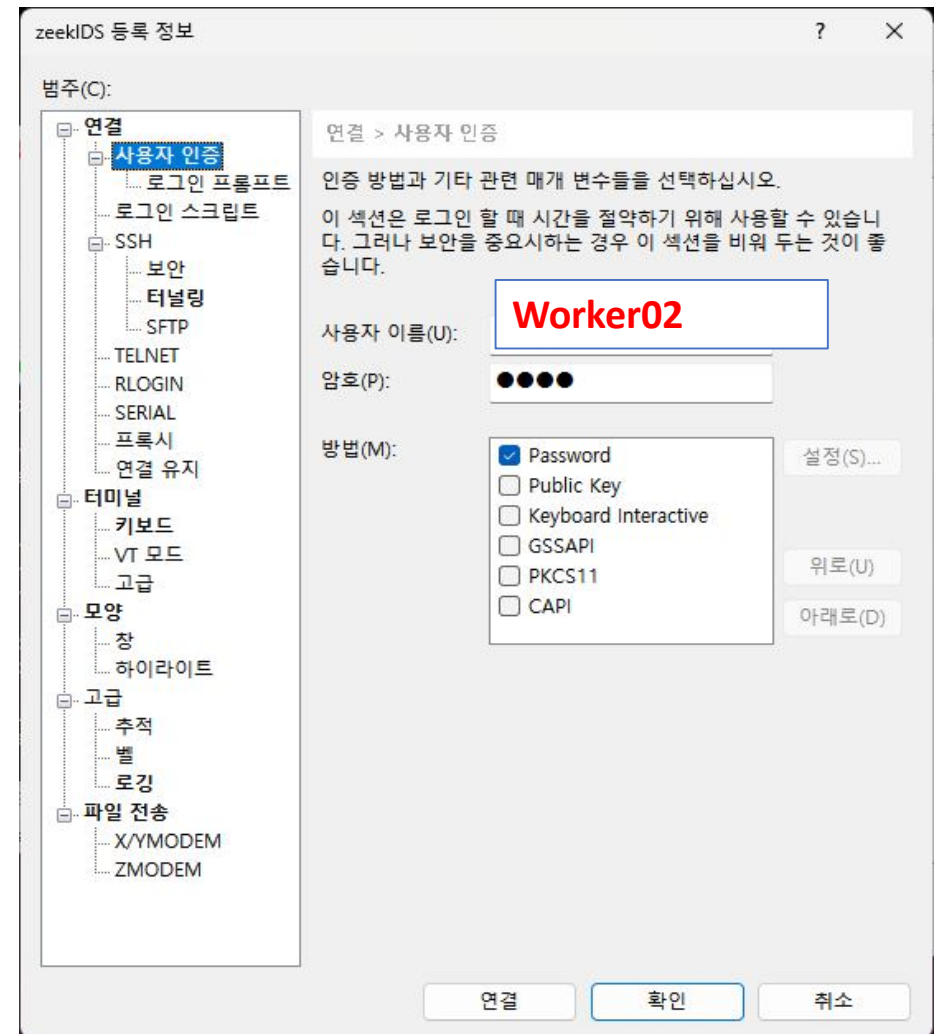
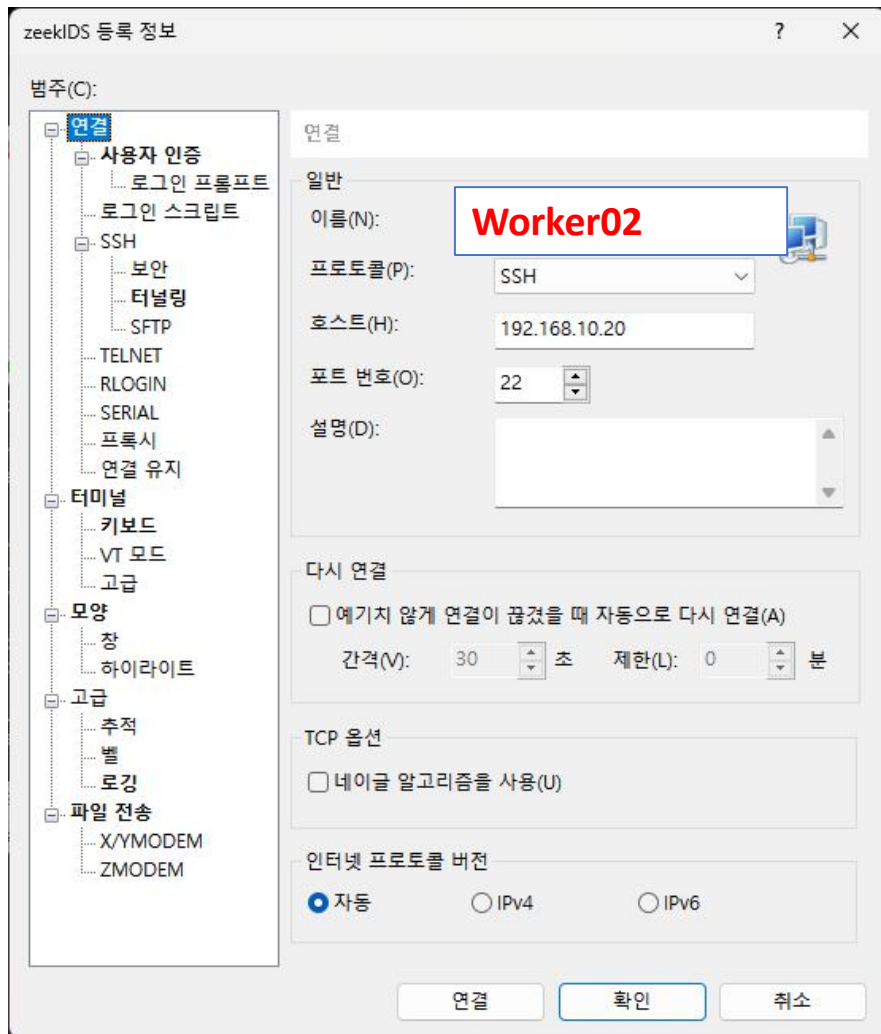
Port 22

```
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

```
root@zeekids:~# ufw status
```

Status: active

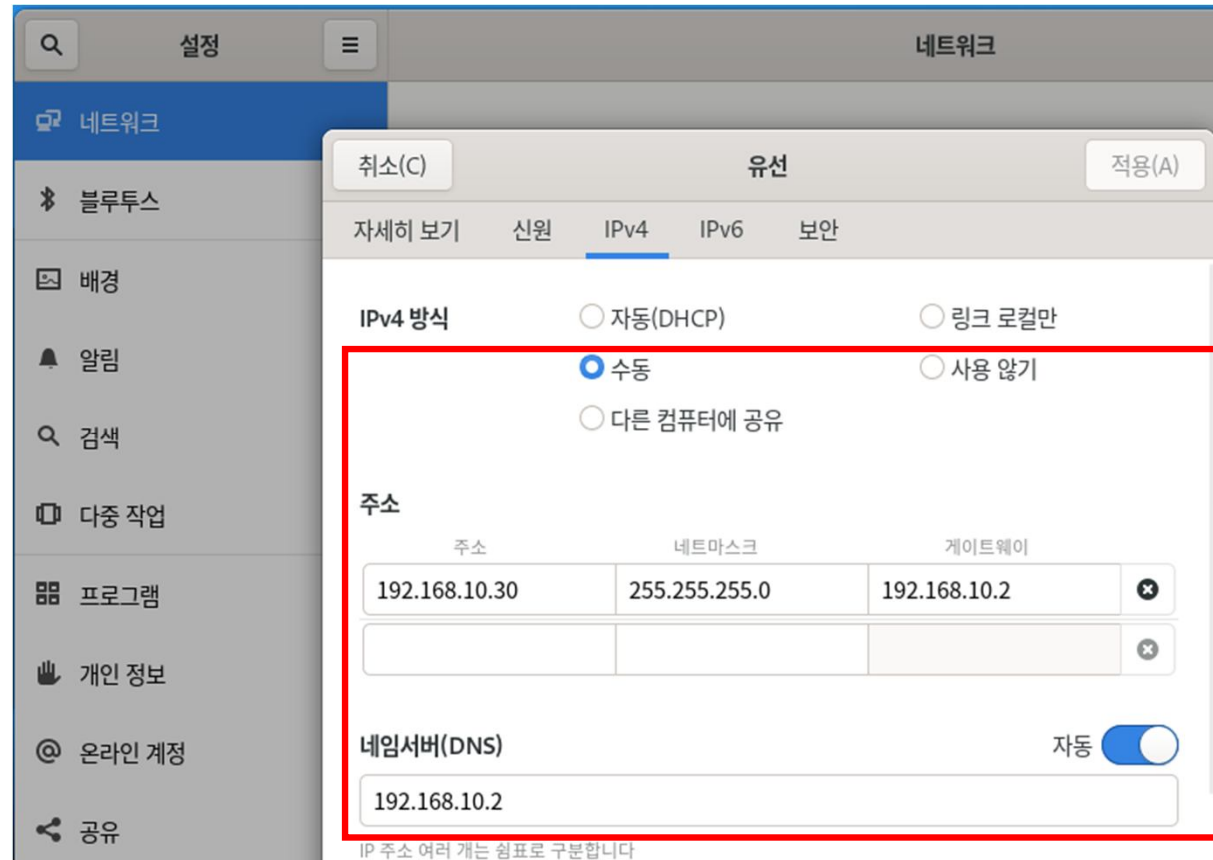
To	Action	From
--	-----	----
22/tcp	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)



4. Rocky Linux 환경 설정

1 네트워크 설정

- 설정 > 네트워크



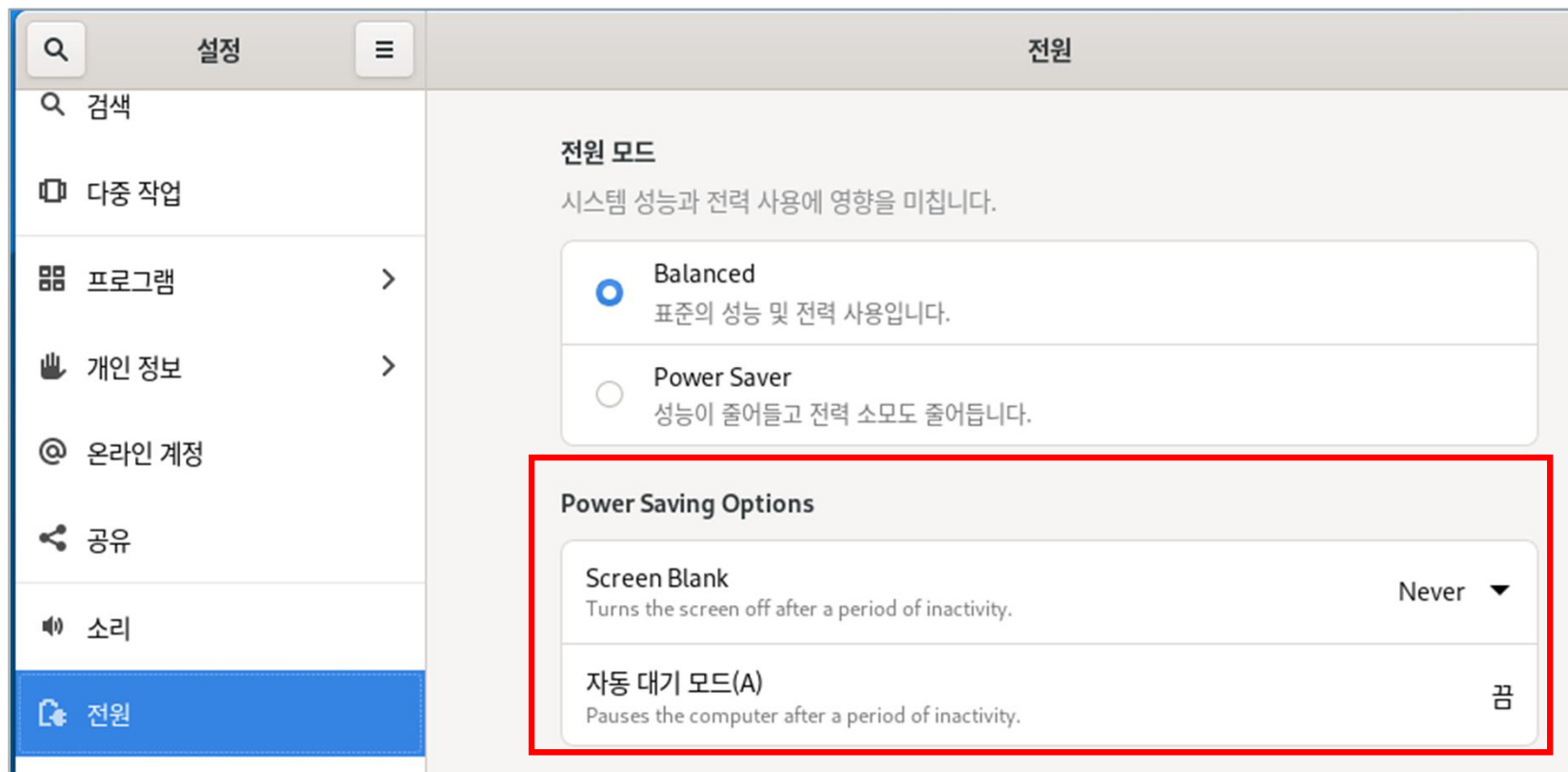
② 화면 잠금

- 설정 > 개인 정보 > 화면잠금



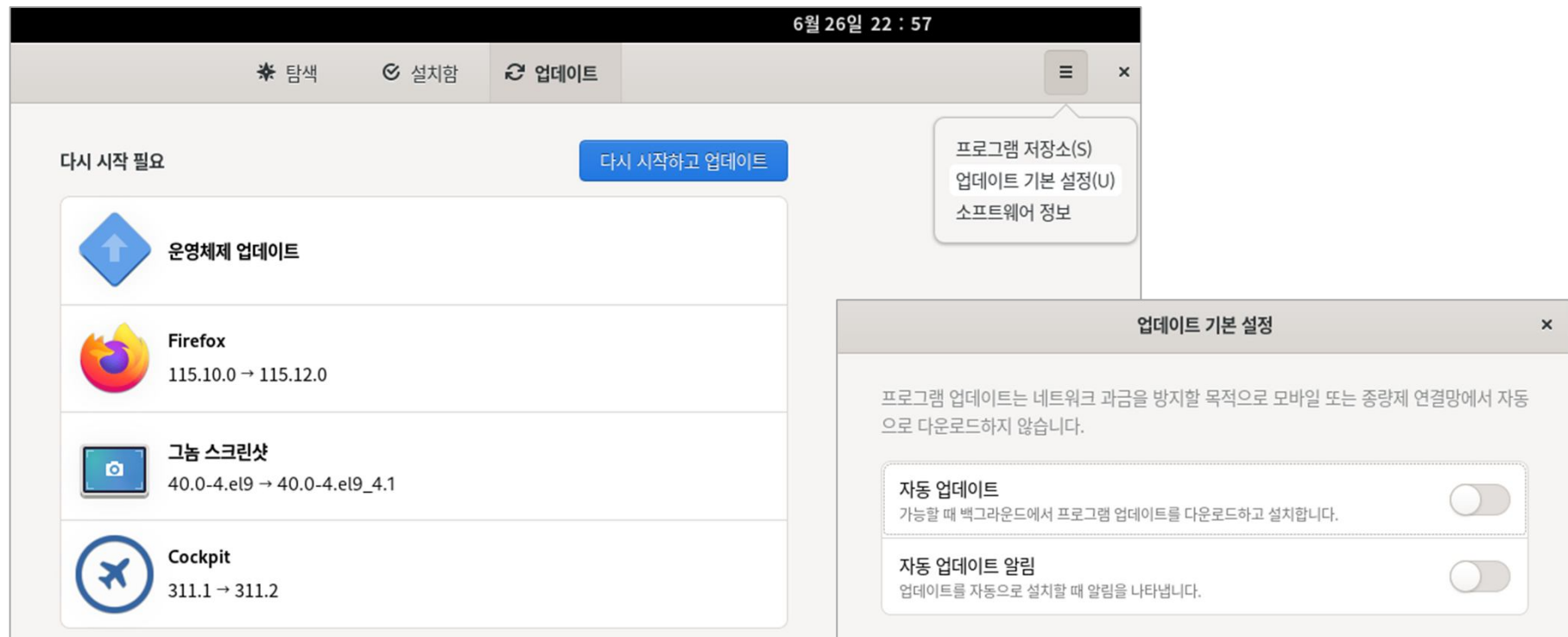
3 빈 화면 비활성

- 설정 > 전원



4 자동 업데이트 기능 끄

- 설정 > 정보 > 소프트웨어 업데이트 > 업데이트 기본설정



⑤ 저장소 update

```
#cd /etc/yum.repos.d
```

```
#mkdir backup
```

```
#mv *.repo backup
```

```
#cd /etc/yum.repos.d
```

```
#nano rocky.repo
```

```
#dnf clean all
```

```
[root@localhost yum.repos.d]# cat rocky.repo
[baseos]
name=Rocky Linux $releasever - BaseOS
baseurl=https://dl.rockylinux.org/vault/rocky/9.0/BaseOS/x86_64/os/
gpgcheck=0

[appstream]
name=Rocky Linux $releasever - AppStream
baseurl=https://dl.rockylinux.org/vault/rocky/9.0/AppStream/x86_64/os/
gpgcheck=0

[extras]
name= Rocky Linux $releasever - Extras
baseurl=https://dl.rockylinux.org/vault/rocky/9.0/extras/x86_64/os/
gpgcheck=0
```

```
[root@localhost yum.repos.d]# dnf clean all
25 파일이 삭제되었습니다
[root@localhost yum.repos.d]#
```

6 방화벽 설치

```
#dnf install -y firewall-config
```

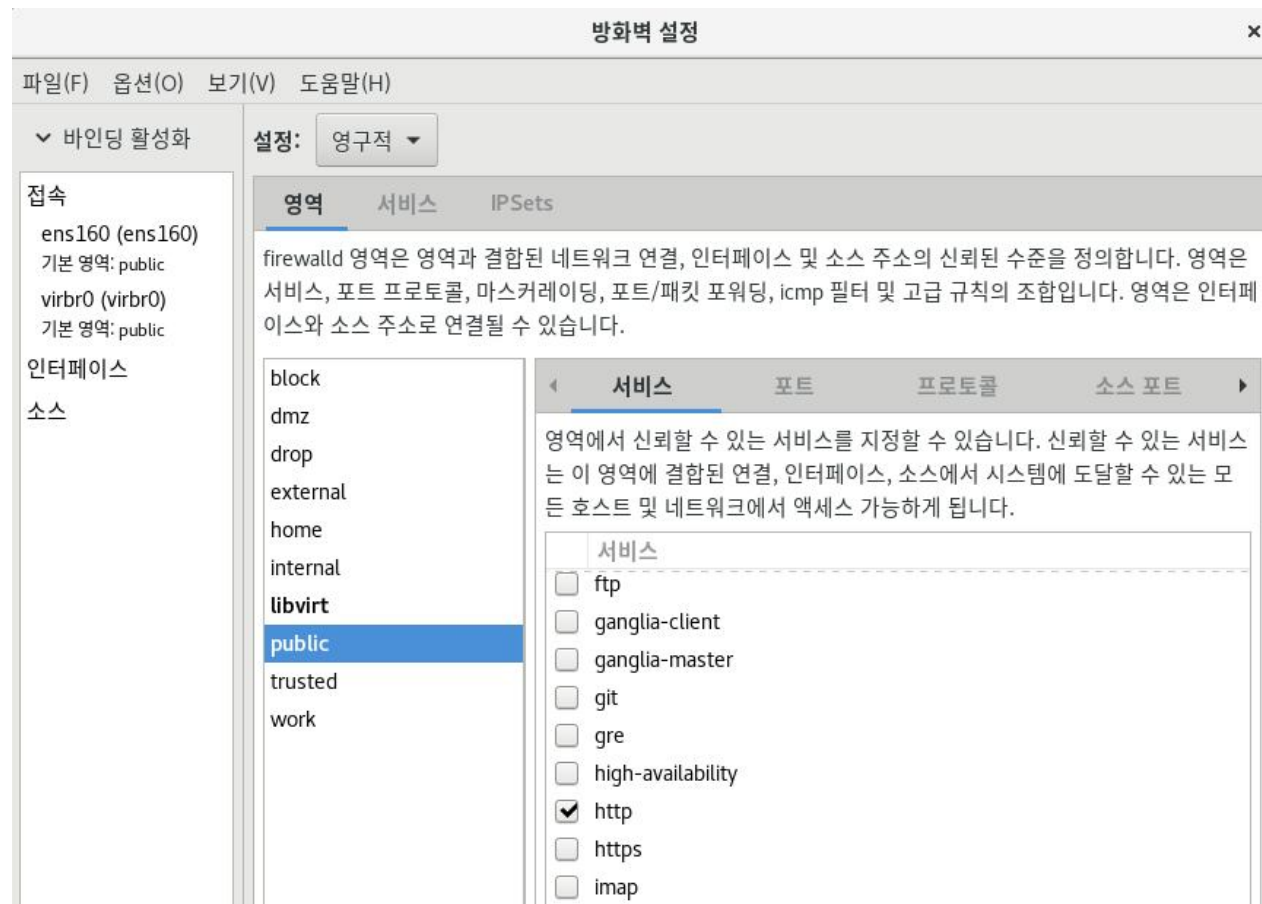
```
[root@rsyslog-client local]# dnf install -y firewall-config
마지막 메타 데이터 만료 확인 : 14:58:47 전에 2022년 08월 06일 (토) 오후 09시 40분 05초.
종속성이 해결되었습니다.
=====
꾸러미                아키텍처                버전                리포지토리                크기
=====
Installing:
firewall-config                noarch                0.6.3-7.el8                AppStream                157 k
=====
거래 요약
=====
설치 1 꾸러미

총 다운로드 크기 : 157 k
설치 크기 : 1.1 M
패키지 다운로드 중 :
firewall-config-0.6.3-7.el8.noarch.rpm                62 kB/s | 157 kB                00:02
-----
합계                62 kB/s | 157 kB                00:02
=====
트랜잭션 점검 실행 중
트랜잭션 검사가 성공했습니다.
트랜잭션 테스트 실행 중
트랜잭션 테스트가 완료되었습니다.
거래 실행 중
준비 중입니다 :
Installing : firewall-config-0.6.3-7.el8.noarch                1/1
스크립틀릿 실행 : firewall-config-0.6.3-7.el8.noarch                1/1
확인 중 : firewall-config-0.6.3-7.el8.noarch                1/1

설치 됨 :
firewall-config-0.6.3-7.el8.noarch

완료되었습니다!
```

#firewall-config



⑦ SSH 활성화

#rpm -qa openssh-server

#systemctl status sshd

```
[root@localhost /]# rpm -qa openssh-server
openssh-server-8.7p1-38.el9.x86_64
[root@localhost /]#
[root@localhost /]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-06-26 22:16:08 KST; 59min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 1059 (sshd)
    Tasks: 1 (limit: 24456)
   Memory: 2.6M
      CPU: 64ms
   CGroup: /system.slice/sshd.service
           └─1059 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Locky 등록 정보

범주(C):

- 연결
 - 사용자 인증
 - 로그인 프롬프트
 - 로그인 스크립트
 - SSH
 - 보안
 - 터널링
 - SFTP
 - TELNET
 - RLOGIN
 - SERIAL
 - 프록시
 - 연결 유지
- 터미널
 - 키보드
 - VT 모드
 - 고급
- 모양
 - 창
 - 하이라이트
- 고급
 - 추적
 - 별
 - 로깅
- 파일 전송
 - X/YMODEM
 - ZMODEM

연결

일반

이름(N): Rocky |

프로토콜(P): SSH

호스트(H): 192.168.10.30

포트 번호(O): 22

설명(D):

다시 연결

☐ 예기치 않게 연결이 끊겼을 때 자동으로 다시 연결(A)

간격(V): 30 초 제한(L): 0 분

TCP 옵션

☐ 네이글 알고리즘을 사용(U)

인터넷 프로토콜 버전

☒ 자동 ☐ IPv4 ☐ IPv6

연결 확인 취소

Locky 등록 정보

범주(C):

- 연결
 - 사용자 인증
 - 로그인 프롬프트
 - 로그인 스크립트
 - SSH
 - 보안
 - 터널링
 - SFTP
 - TELNET
 - RLOGIN
 - SERIAL
 - 프록시
 - 연결 유지
- 터미널
 - 키보드
 - VT 모드
 - 고급
- 모양
 - 창
 - 하이라이트
- 고급
 - 추적
 - 별
 - 로깅
- 파일 전송
 - X/YMODEM
 - ZMODEM

연결 > 사용자 인증

인증 방법과 기타 관련 매개 변수들을 선택하십시오.

이 섹션은 로그인 할 때 시간을 절약하기 위해 사용할 수 있습니다. 그러나 보안을 중요시하는 경우 이 섹션을 비워 두는 것이 좋습니다.

사용자 이름(U): webadmin

암호(P): ●●●●

방법(M):

- ☒ Password
- ☐ Public Key
- ☐ Keyboard Interactive
- ☐ GSSAPI
- ☐ PKCS11
- ☐ CAPI

설정(S)...

위로(U)

아래로(D)

연결 확인 취소

7 Selinux 비활성화

```
#sestatus
```

```
#grubby --update-kernel ALL --args selinux=0
```

```
#sestatus
```

SELinux(Security Enhanced Linux)

- 리눅스의 근본적이고 구조적인 보안 약점(소스공개)을 보완하기 위한 리눅스 확장 도구이자 보안 모듈
- 강제 접근 통제(MAC; Mandatory Access Control)에 기반한 접근 제어
 - 미리 정해진 정책과 보안 등급에 의거하여 주체에게 허용된 접근 권한과 객체에게 부여된 허용 등급을 비교하여 접근을 통제

⑧ SSH 활성화

```
#rpm -qa openssh-server
```

```
#systemctl status sshd
```

```
[root@localhost /]# rpm -qa openssh-server
openssh-server-8.7p1-38.el9.x86_64
[root@localhost /]#
[root@localhost /]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-06-26 22:16:08 KST; 59min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 1059 (sshd)
    Tasks: 1 (limit: 24456)
   Memory: 2.6M
      CPU: 64ms
   CGroup: /system.slice/sshd.service
           └─1059 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

[참조] 수동 네트워크 활성화/비활성화

```
#cd /etc/NetworkManager/system-connections
```

```
#ls
```

```
#cat ens160.nmconnection
```

```
#nmcli connection down ens160
```

```
#nmcli connection up ens160
```

```
#reboot
```

```
[root@localhost /]# cd /etc/NetworkManager/system-connections/
[root@localhost system-connections]# ls
ens160.nmconnection
[root@localhost system-connections]# cat ens160.nmconnection
[connection]
id=ens160
uuid=be72033d-4de8-3ec4-afa9-a976273e622b
type=ethernet
autoconnect-priority=-999
interface-name=ens160

[ethernet]

[ipv4]
address1=192.168.10.30/24,192.168.10.2
dns=192.168.10.2;
method=manual

[ipv6]
addr-gen-mode=eui64
method=auto

[proxy]
[root@localhost system-connections]#
```

[참조] 패키지 설치 유틸리티

Debian 계열	<ul style="list-style-type: none">• dpkg• apt-get	<code>dpkg -l 패키지명</code> <code>apt-get install -y 패키지명</code>
Redhat 계열	<ul style="list-style-type: none">• rpm• yum• dnf (RHEL 8부터 사용)	<code>rpm -qa 패키지명</code> <code>yum install -y 패키지명</code> <code>dnf install -y 패키지명</code>

5. Web Server 구축

WebServer 구축(1)



Apache Web Server
(Rocky)

```
#rpm -qa | grep httpd
```

```
#dnf install httpd -y
```

```
#systemctl start httpd
```

```
#systemctl status httpd
```

```
#ps -ef | grep httpd
```

```
#systemctl enable httpd
```

```
#firewall-cmd --add-service=http
```

```
#gedit /var/www/html/index.html
```

```
<html>
<head><title>Hello~</title></head>
<body>
  Hello~ Rocky
</body>
</html>
```

WebServer 구축(2)



Apache Web Server
(Ubuntu)

```
#apt update
#dpkg -l | grep apache2
#apt install apache2 -y
#systemctl start apache2
#systemctl enable apache2
#systemctl status apache2
#ufw allow 'Apache'
#ufw status
#gedit /var/www/html/index.html
```

```
<html>
  <head><title>Hello~</title></head>
  <body>
    Hello~ Apache^^
  </body>
</html>
```

WebServer 구축(3)



Nginx Web Server
(Ubuntu)

```
#apt update
#dpkg -l | grep apache2
#apt install nginx -y
#systemctl start nginx
#systemctl enable nginx
#systemctl status nginx
#ufw allow 'Nginx HTTP'
#ufw status
#gedit /var/www/html/index.html
```

```
<html>
  <head><title>Hello~</title></head>
  <body>
    Hello~ Nginx^^
  </body>
</html>
```