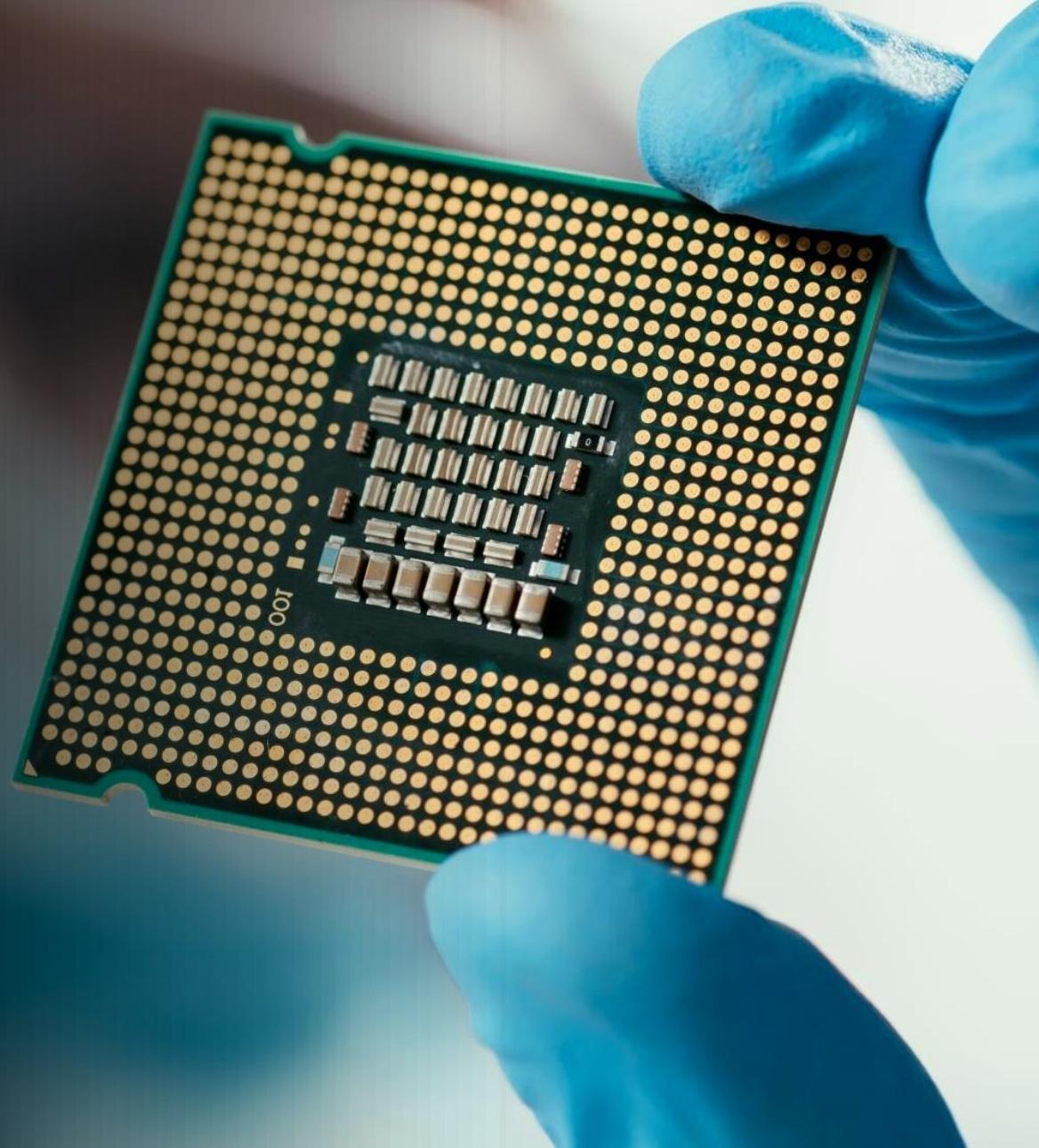


2024. 리눅스 서버구축 실무

충북교육연구정보원 정보영재교육원

| SW·AI교실 | 정보아카데미 | 정올 마스터 과정 | 강사

흥덕고등학교 교사 박정진



1. Linux 기초
2. Domain Name System
3. Docker (podman)의 활용
4. 멀티도메인 SSL 웹서버 구축
5. 실전 웹서버 구축

2024 리눅스 서버구축 실무

1. Linux 기초

- 1) Linux 및 VPS 소개
- 2) 서버 초기 설정
- 3) 기초 명령어 실습
- 4) 리눅스 계정
- 5) 방화벽(firewalld, firewall-cmd)

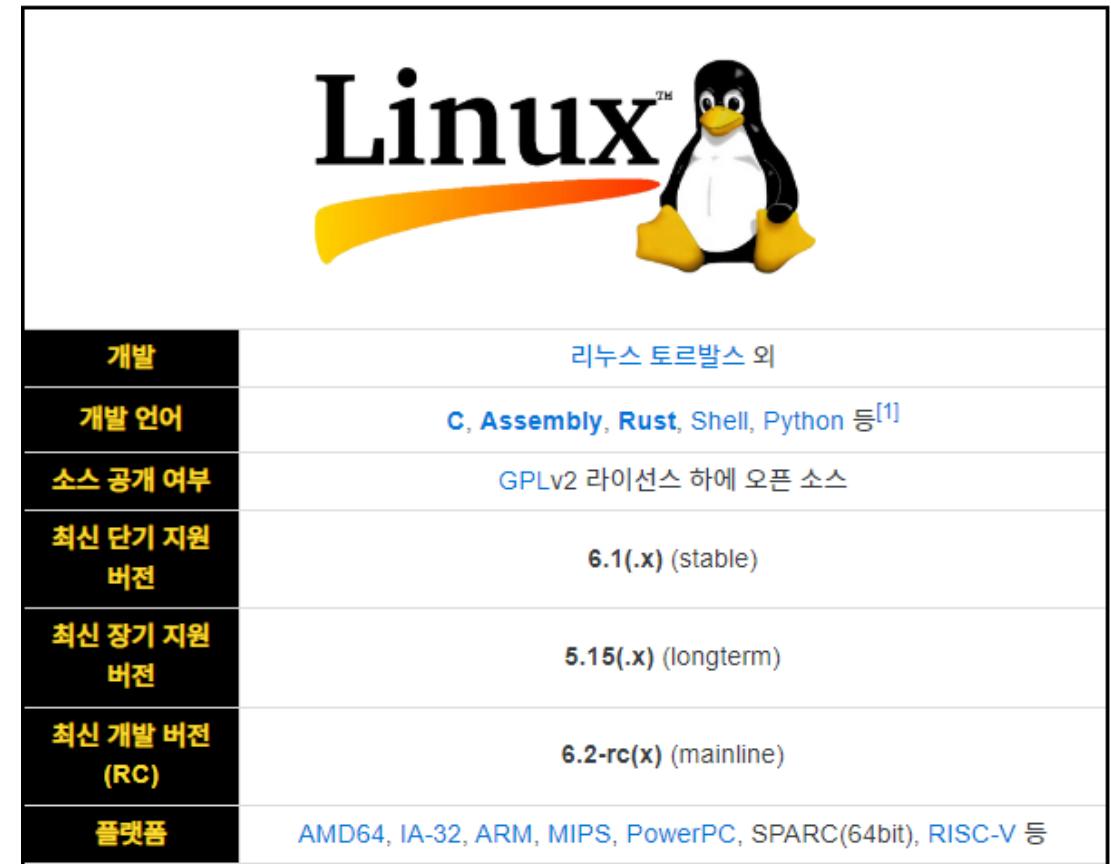
충북교육연구정보원 강사 박정진



리눅스(Linux)

■ 개요

- 1991년 9월 17일 리누스 토르발스가 처음 출시한 운영체제 커널인 리눅스 커널에 기반을 둔 오픈 소스 유닉스 계열 운영 체제
- Linux라는 이름은 Linus' *nix, 리누스의 유닉스라는 뜻
- 컴퓨터 역사상 가장 많은 참여자가 관여하고 있는 오픈 소스 프로젝트
- 모바일 운영체제로 유명한 안드로이드 역시 리눅스 커널을 가져다 사용



리눅스(Linux)

- 다양한 리눅스 배포판

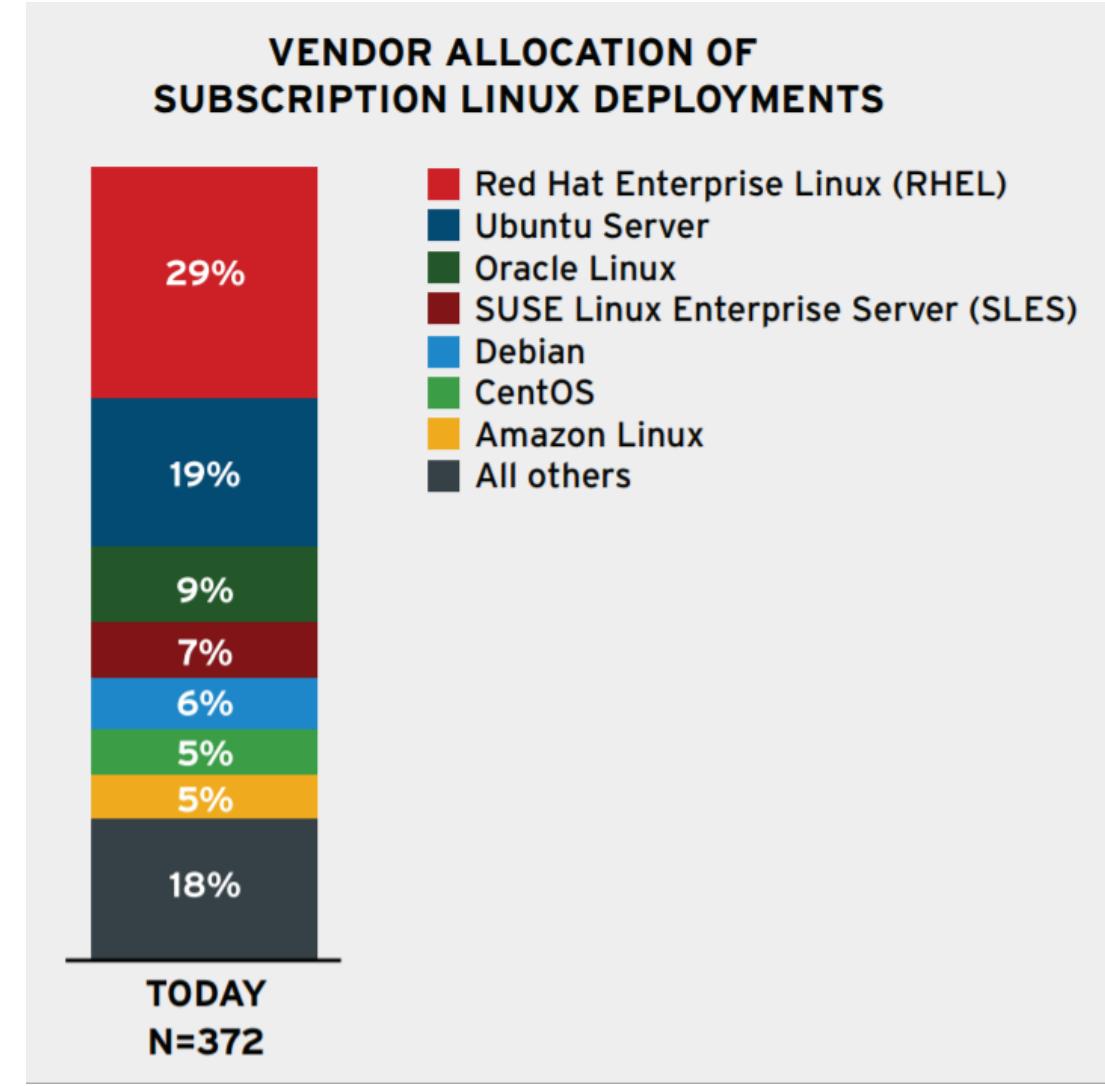
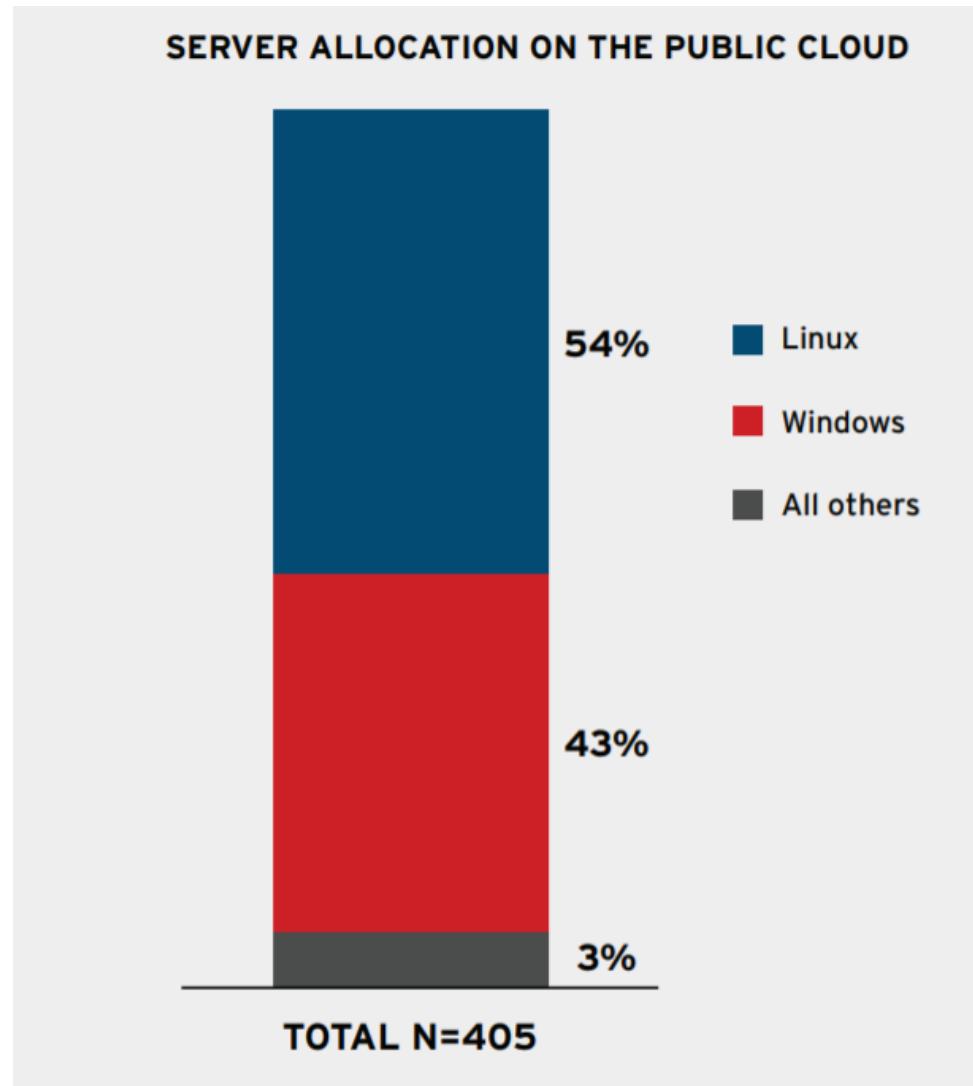


메이저 리눅스 배포판 비교

CentOS(Community Enterprise Operating System)

	데스크탑용	서버용
Redhat 계열	 fedora	 redhat  CentOS
Debian 계열	 ubuntu	 debian
기타	 KALI	침투 테스트에 최적화된 특수한 리눅스 배포판

The state of Linux in the public cloud for enterprises





- 현재 Fedora, rhel, centos의 관계



- 커뮤니티 프로젝트
- 빠른 릴리즈 (6개월)
- 실험적 기능
- 무료

- RHEL의 Upstream 프로젝트
- RHEL을 위한 테스트 및 검증
- RHEL과 바이너리 호환성 보장 안됨
- 무료

- 페도라 기반, 레드햇 개발
- 긴 지원기간
- 높은 수준의 안정성
- 상용

DigitalOcean vs AWS vs Google Cloud vs Linode vs Vultr: Which is Best for Hosting WordPress?

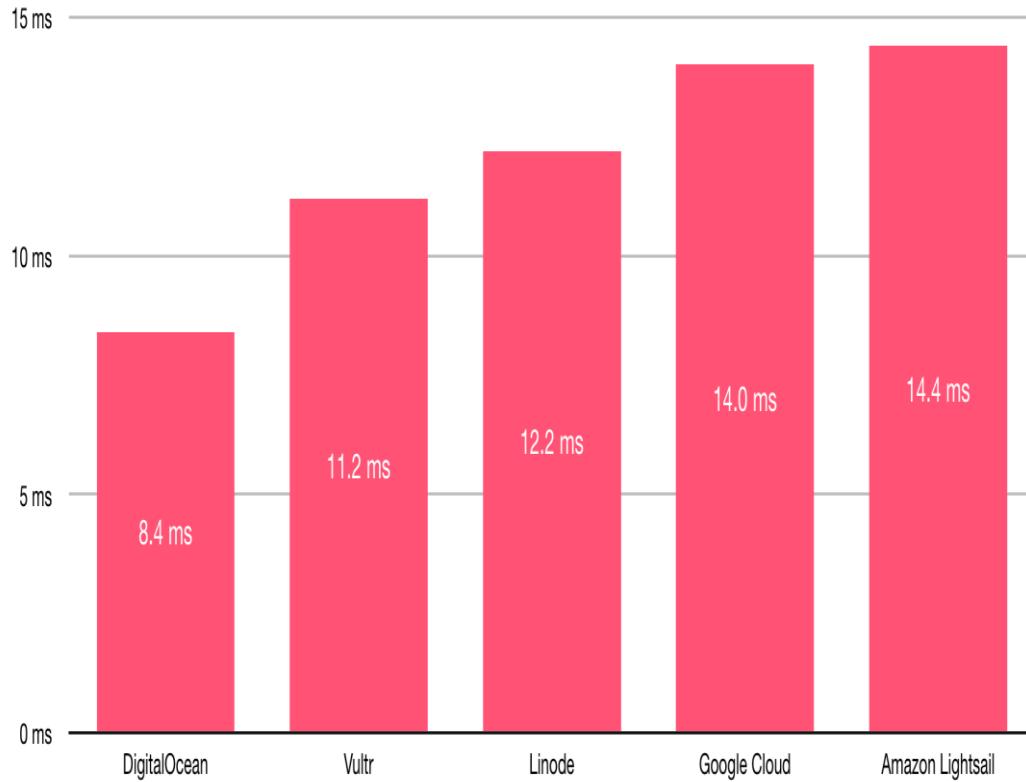
- 동급 VPS 비교 (최소한의 서버 스펙)

	Memory	CPU	Storage	Price
DigitalOcean	2 GB	1 vCPU	50 GB	\$10/mo
Amazon Lightsail	2 GB	1 vCPU	60 GB	\$10/mo
Google Cloud	2 GB	1 vCPU	50 GB	\$20.73/mo*
Linode	2 GB	1 vCPU	50 GB	\$10/mo
Vultr	2 GB	1 vCPU	55 GB	\$10/mo

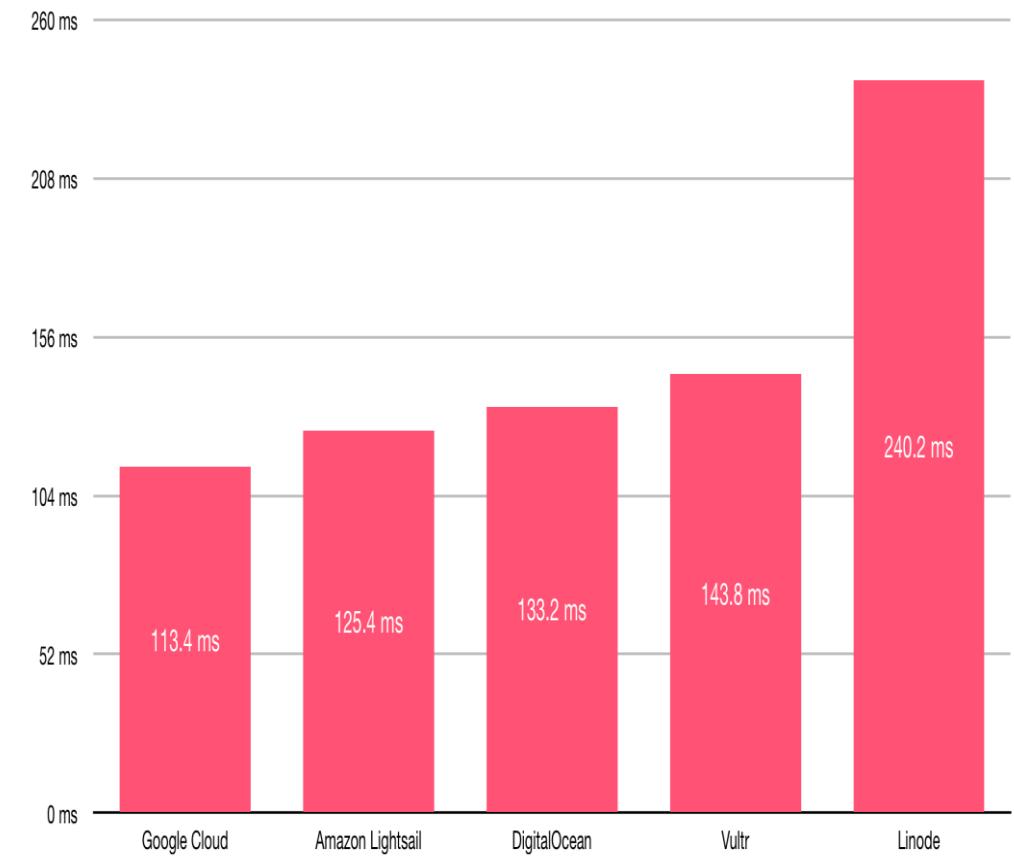
*Google Cloud price is estimated using their [pricing calculator](#).

VPS 성능 비교 (작은 숫자가 우수)

■ Cached Performance



■ Uncached Performance



Vultr VPS 만들기

The screenshot shows the Vultr Billing section of the dashboard. On the left sidebar, under the Account category, the "Billing" option is selected. A modal window titled "Make a Credit Card Payment" is displayed, featuring a "Use Code: CLOUD250FREE" button and a message about trying Vultr for free with \$250 credit. Below the modal, there's a "Credit Card Details" section with fields for Card Number, MM/YY, and CVV, along with a list of supported credit card brands.

Account

NEWS: Vultr Offers NVIDIA GH200 Grace Hopper Superchip: Tap Into Ultimate Power and Efficiency

Billing

Welcome to the everywhere cloud. [Continue setting up your account.](#)

You are almost ready to deploy servers - please link a payment method to get started!

Credit Card

Paypal

Crypto

Alipay

Wire Transfer

Make a Credit Card Payment

Use Code: **CLOUD250FREE**

Try Vultr for free with \$250 free credit. New customers only.

[View details →](#)

Credit Card Details

Card Number

MM/YY

CVV

Credit Card brands we accept:

Summary

Enter Code Apply **CLOUD250FREE** +\$250.00

\$250 for 30 days **Remove**

(credit expires 30 days after signup)

Total Billed to Credit Card +\$0.00

Total Available Credit **\$250.00**

Vultr VPS 만들기

NEWS: Vultr Managed Databases now supports Redis

PARK JEONG JIN

Products

Instances Snapshots ISOs Scripts DNS Block Storage Objects Firewall Network Load Balancers Kubernetes Databases

+ Deploy New Server

Add Managed Database

Add Block Storage

Add Kubernetes

Add Load Balancers

Add Object Storage

Add Reserved IP

View More Options

Introducing Vultr Cloud GPUs powered by the NVIDIA A40. [Provision your instance now.](#)

Search...

<input type="checkbox"/> Server	OS	Location	Charges
<input type="checkbox"/> datahub.pe.kr 1024.00 MB Regular Cloud Compute - 141.164.52.98		Seoul	\$1.91



High Performance

- Powered by new generations of AMD EPYC or Intel Xeon CPUs and NVMe SSD.

vCPUs	Memory	Bandwidth	Storage	Monthly Price	Hourly Price
1 vCPU	1 GB	2.00 TB	25 GB	\$6.00 /mo	\$0.009 /hr
1 vCPU	2 GB	3.00 TB	50 GB	\$12.00 /mo	\$0.018 /hr
2 vCPU	2 GB	4.00 TB	60 GB	\$18.00 /mo	\$0.027 /hr
2 vCPU	4 GB	5.00 TB	100 GB	\$24.00 /mo	\$0.036 /hr
4 vCPU	8 GB	6.00 TB	180 GB	\$48.00 /mo	\$0.071 /hr
4 vCPU	12 GB	7.00 TB	260 GB	\$72.00 /mo	\$0.107 /hr
8 vCPU	16 GB	8.00 TB	350 GB	\$96.00 /mo	\$0.143 /hr
12 vCPU	24 GB	12.00 TB	500 GB	\$144.00 /mo	\$0.214 /hr

Deploy New Instance

Choose Server



Optimized Cloud Compute

Virtual machines for more demanding business apps, e.g. production websites, CI/CD, video transcoding, or larger databases.

Starting from
\$28.00/mo

Dedicated vCPU



Cloud Compute

Virtual machines for apps with bursty performance, e.g. low traffic websites, blogs, CMS, dev/test environments, and small databases.

Starting from
\$2.50/mo

Shared vCPU



Cloud GPU

Virtual machines with fractional or full NVIDIA GPUs for AI, machine learning, HPC, visual computing and VDI. Also available as Bare Metal.

Starting from
\$21.50/mo

NVIDIA GPU +
Dedicated vCPU



Bare Metal

Single tenant bare metal for apps with the most demanding performance or security requirements.

Starting from
\$120.00/mo

Physical CPU +
Optional GPU

CPU & Storage Technology

NEW

AMD High Performance

Powered by latest generation AMD EPYC CPUs and NVMe SSD.

NEW

intel. High Performance

Powered by new generations of Intel Xeon CPUs and NVMe SSD.

✓

intel. High Frequency

Powered by 3GHz+ Intel Xeon CPUs and NVMe SSD.

intel. Regular Performance

Powered by previous generation Intel CPUs and regular SSD.

Server Location

All Locations America Europe Australia Asia

Tokyo Japan

Bangalore India

Delhi NCR

Seoul South Korea

Singapore Singapore

London

Server Image

Operating System Marketplace Apps Upload ISO ISO Library Backup Snapshot

[Server 위치별 Ping 차이]

C:\Users\akapo>ping 149.28.65.59

[미국 서부 LA]

Ping 149.28.65.59 32바이트 데이터 사용:

149.28.65.59의 응답: 바이트=32 시간=141ms TTL=52

149.28.65.59의 응답: 바이트=32 시간=142ms TTL=52

149.28.65.59의 응답: 바이트=32 시간=140ms TTL=52

149.28.65.59의 응답: 바이트=32 시간=141ms TTL=52

149.28.65.59에 대한 Ping 통계:

패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),

왕복 시간(밀리초):

최소 = 140ms, 최대 = 142ms, 평균 = 141ms



Rocky Linux
Select Version



Ubuntu
Select Version

Arch Linux
test x64

Fedora CoreOS
Select Version

CentOS
8 Stream x64
9 Stream x64
8 Stream x64

7 x64 Without SELinux

7 x64



Windows Core Stand...
Select Version



OpenBSD
Select Version



Windows Standard
Select Version

[Server 위치별 Ping 차이]

C:\Users\akapo\Downloads>ping 158.247.208.252

[한국]

Ping 158.247.208.252 32바이트 데이터 사용:

158.247.208.252의 응답: 바이트=32 시간=6ms TTL=51

158.247.208.252에 대한 Ping 통계:

패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),

왕복 시간(밀리초):

최소 = 6ms, 최대 = 6ms, 평균 = 6ms

C:\Users\akapo\Downloads>ping 202.182.107.70

[일본]

Ping 202.182.107.70 32바이트 데이터 사용:

202.182.107.70의 응답: 바이트=32 시간=32ms TTL=52

202.182.107.70의 응답: 바이트=32 시간=31ms TTL=52

202.182.107.70의 응답: 바이트=32 시간=31ms TTL=52

202.182.107.70의 응답: 바이트=32 시간=31ms TTL=52

202.182.107.70에 대한 Ping 통계:

패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),

왕복 시간(밀리초):

최소 = 31ms, 최대 = 32ms, 평균 = 31ms

Additional Features

- Enable IPv6 [?](#)
- No Public IPv4 Address [?](#)
- Enable DDOS Protection [\\$10/mo](#)
- Enable Cloud-Init User-Data [?](#)
- Enable Virtual Private Clouds [?](#)
- Enable VPC 2.0 [?](#)
- Enable Limited User Login [?](#)

SSH Keys [\(Manage \)](#)

[Add New](#)

Server Hostname & Label

Enter server hostname
Enter server label

Servers Qty:

[-](#) [1](#) [+](#)

Summary:

\$12.00 /month (\$0.018/hour)[Deploy Now](#) [Additional 10% VAT applicable to your account](#)[Update Settings](#)



Products

NEWS: Vultr Offers NVIDIA GH200 Grace Hopper Superchip: Tap Into Ultimate Power and Efficiency



JEONG JIN PARK

Deploy +



Compute

Products

Cloud Storage



Kubernetes



Container Registry



Databases



Load Balancers



Network



Orchestration



Cloud Compute

Location

Search

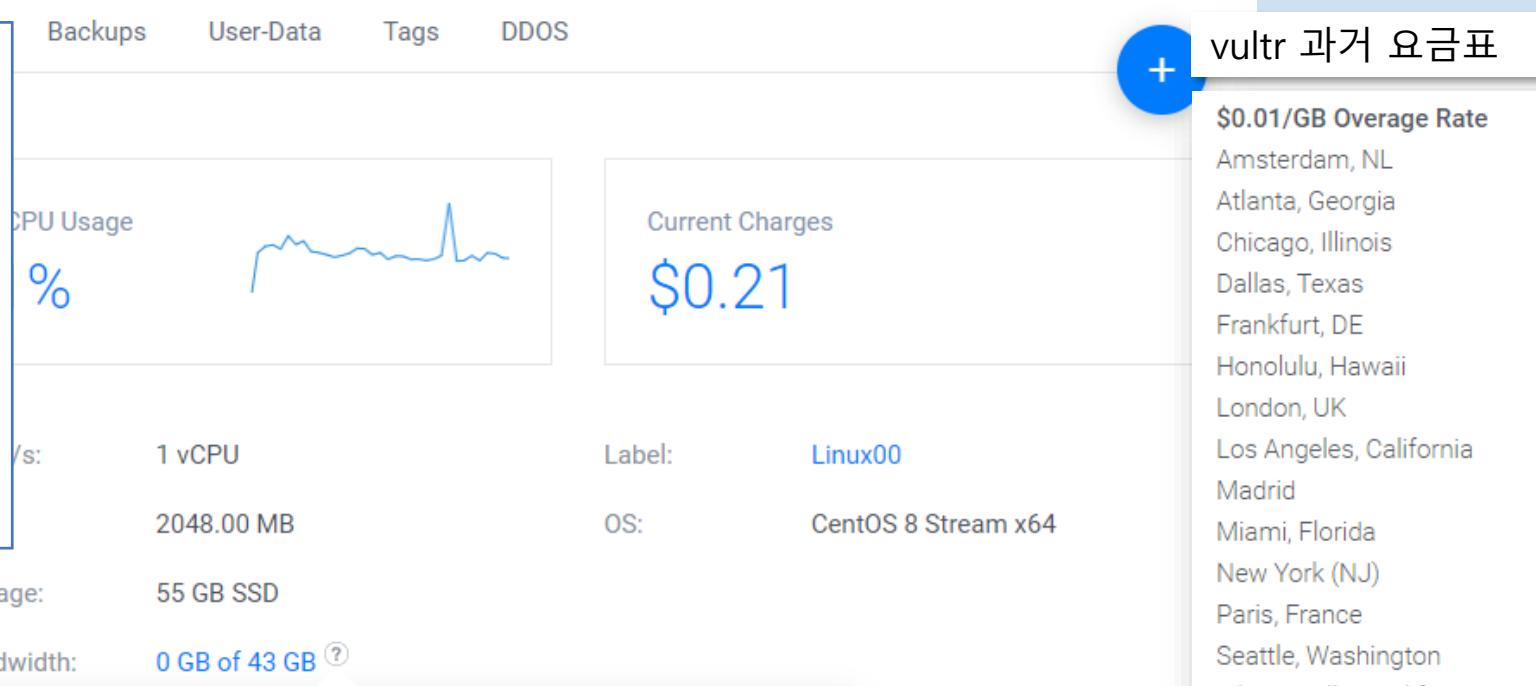
+ Deploy

<input type="checkbox"/>	Name	OS	Location	Charges	Status	...
<input type="checkbox"/>	cberi.shop 2048.00 MB AMD High Performance - 207.148.95.144		Tokyo	\$0.02	Running	



- Bandwidth(허용 트래픽) :

- 서버로 들어오는 데이터는 무료
- 2000GB/월(672시간) = 3GB/시간
- 3시간 동안 활성화된 서버 → 9GB
- 1개월(672시간) → 2000GB
- 매 결재 주기(월)마다 반복
- 초과 트래픽: \$0.01/GB
 - 한 계정당 2TB 차감 후 계산



업체	용량	가격
Vultr (23년부터)	GB (업로드만)	\$0.01 (약 13원)
AWS EC2 한국 리전 기준	GB (업로드만)	\$0.1386 (처음 10TB)(약 181원)
Azure 아시아 내 지역간	GB (업로드만)	107원
Linode	GB (업로드만)	\$0.01
CloudV (국제)	GB	500원
가비아	GB	100원

\$0.01/GB Overage Rate
Amsterdam, NL
Atlanta, Georgia
Chicago, Illinois
Dallas, Texas
Frankfurt, DE
Honolulu, Hawaii
London, UK
Los Angeles, California
Madrid
Miami, Florida
New York (NJ)
Paris, France
Seattle, Washington
Silicon Valley, California
Stockholm, Sweden
Toronto, Canada
Warsaw, Poland

\$0.03/GB Overage Rate
Singapore
Tokyo, Japan

\$0.05/GB Overage Rate
Bangalore, India
Delhi NCR, India
Johannesburg, South Africa
Melbourne, Australia
Mexico City, Mexico
Mumbai, India
Seoul, South Korea
Sydney, Australia
São Paulo, Brazil

VPS(Virtual Private Server) 정보

순번	소속교	성명	VPS IP	root p/w	순번	소속교	성명	VPS IP	root p/w
1	남성중	김민건	45.32.255.157	?3nQ+2](TUY)idMC	11	홍덕고	남하준	45.32.26.235	(8Bnh!-(jE8w+,a-
2	일신여고	방윤정	167.179.108.128	a[T3i(z9kjy4q]HL	12	일신여고	이정민	149.28.22.90	T8p)G+S7EoEMNvkh
3	대성고	오미현	167.179.108.93	E}o7Kz+vYWz?r7*B	13		VPS13	167.179.102.224	_j5L3i}_NAcKEFpg=
4	대성고	방민영	45.76.204.169	qT5*nuLPbVjF7gNj	14				
5	IT과학고	김민재	149.28.26.135	8kU.SD\$3@[aM[jam	15				
6	홍덕고	이주원	198.13.54.39	Bx%4e(CGbz9Ep7n[16				
7			198.13.54.39	Bx%4e(CGbz9Ep7n[17				
8	홍덕고	김지겸	108.61.183.138	v7M[VJyji9Q_ory_	18				
9	홍덕고	정호	66.42.40.63	R4!g(T4sn(ETh9MP	19				
10	정보원	박정진	108.61.180.91	6Jz?vbLuhc}bh+.K	20				

※ 자신의 서버만 사용해야 함. 비밀번호 변경에 신중을 기할 것. (분실 시 복구 불가능)

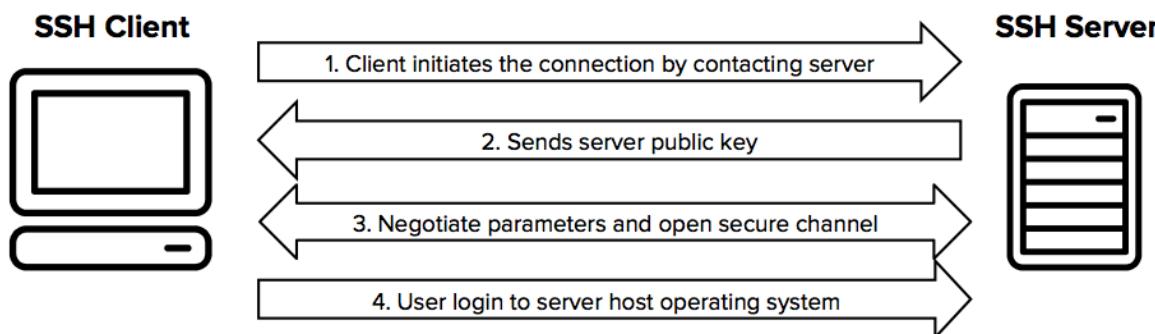
SSH (Secure Shell)

■ 기본개념

- 원격지 호스트 컴퓨터에 접속하기 위해 사용되는 인터넷 프로토콜
- 기존에 사용하던 telnet은 암호화가 이루어지지 않아 계정 정보가 탈취될 위험이 높음
- 여기에 **암호화 기능**을 추가하여 1995년에 나온 프로토콜이다. (**기본포트 22번**)

■ SSH의 주요 기능

- 보안 접속을 통한 rsh, rcp, rlogin, rexec, **telnet**, **ftp** 등을 제공.
- IP spoofing (IP스푸핑, 아이피 위변조 기법중 하나)을 방지하기 위한 기능을 제공.
- X11 패킷 포워딩 및 일반적인 TCP/IP 패킷 포워딩을 제공.



SSH 접속도구

- putty (<https://www.putty.org>) 다운로드



Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

[Download PuTTY](#)

Below suggestions are independent of PuTTY. They are not endorsements by the PuTTY project.

Bitvise SSH Client

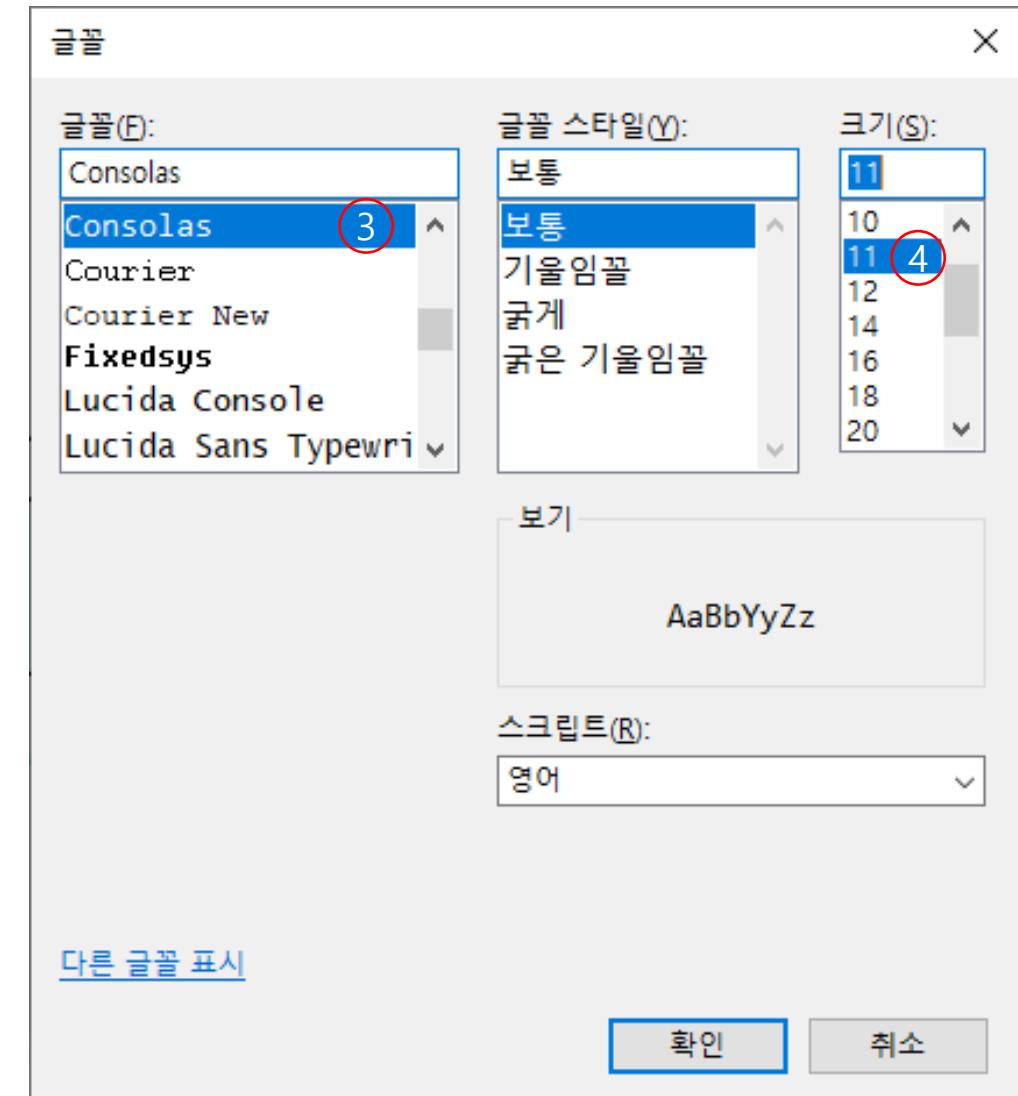
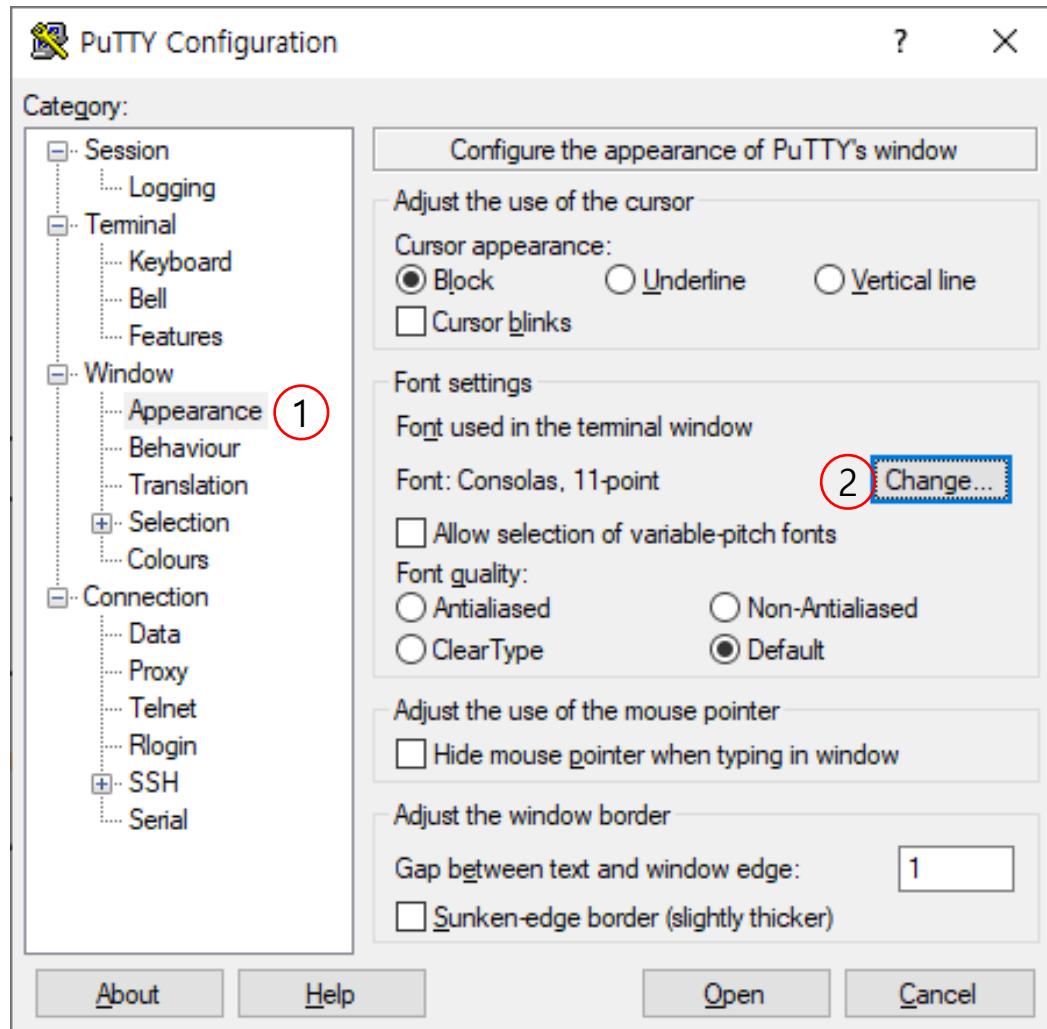
Bitvise SSH Client is an SSH and SFTP client for Windows. It is developed and supported professionally by Bitvise. The SSH Client is robust, easy to install, easy to use, and supports all features supported by PuTTY, as well as the following:

- graphical SFTP file transfer;
- single-click Remote Desktop tunneling;
- auto-reconnecting capability;
- dynamic port forwarding through an integrated proxy;
- an FTP-to-SFTP protocol bridge.

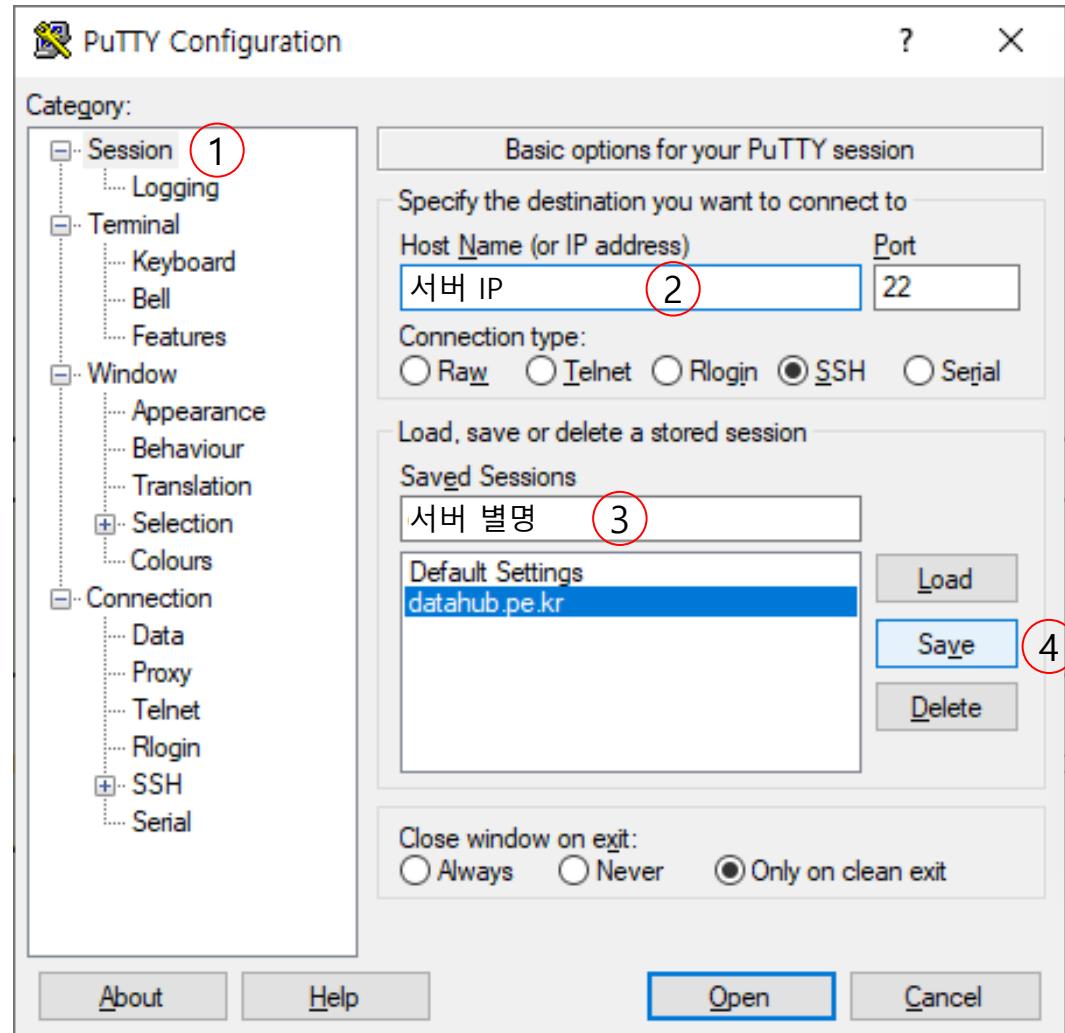
Bitvise SSH Client is **free to use**.

[Download Bitvise SSH Client](#)

putty



putty



※ 원고 작성시 필자가 사용한 VPS의 IP가 **207.148.95.144**임.
앞으로 위 IP가 나오면 수강생 VPS의 IP를 사용해야 함.

root 계정 패스워드 변경

■ 패스워드 변경

- 명령어 형식

```
passwd [OPTION] <accountName>
```

- 실습

```
[root@vultr ~]# passwd  
Changing password for user root.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@vultr ~]#
```

- 프롬프트

```
[root@vultr ~]#
```

사용자명

호스트명

현재 작업폴더

명령어 도움말

■ --help 옵션

```
[root@vultr ~]# passwd --help
Usage: passwd [OPTION...] <accountName>
  -k, --keep-tokens      keep non-expired authentication tokens
  -d, --delete           delete the password for the named account (root
                        only); also removes password lock if any
  -l, --lock              lock the password for the named account (root only)
  -u, --unlock            unlock the password for the named account (root only)
  -e, --expire             expire the password for the named account (root only)
  -f, --force              force operation
  -x, --maximum=DAY(S)    maximum password lifetime (root only)
  -n, --minimum=DAY(S)    minimum password lifetime (root only)
  -w, --warning=DAY(S)   number of days warning users receives before
                        password expiration (root only)
  -i, --inactive=DAY(S)  number of days after password expiration when an
                        account becomes disabled (root only)
  -S, --status             report password status on the named account (root
                        only)
  --stdin                 read new tokens from stdin (root only)

Help options:
  -?, --help                Show this help message
  --usage                  Display brief usage message
```

명령어 도움말

■ man(manual) 명령어

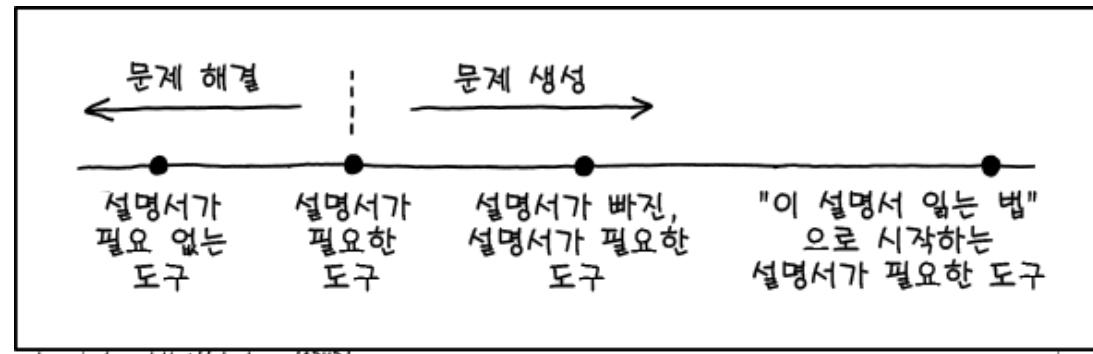
```
man [조회할 명령어]
```

```
[root@vultr ~]# man passwd
```

```
root@vultr:~  
PASSWD(1) User utilities PASSWD(1)  
  
NAME  
passwd - update user's authentication tokens  
  
SYNOPSIS  
passwd [-k] [-l] [-u [-f]] [-d] [-e] [-n mindays] [-x maxdays] [-w  
warndays] [-i inactivedays] [-S] [--stdin] [-?] [--usage] [username]  
  
DESCRIPTION  
The passwd utility is used to update user's authentication token(s).  
  
This task is achieved through calls to the Linux-PAM and Libuser API.  
Essentially, it initializes itself as a "passwd" service with Linux-PAM  
and utilizes configured password modules to authenticate and then  
update a user's password.  
  
A simple entry in the global Linux-PAM configuration file for this service would be:  
  
#  
# passwd service entry that does strength checking of  
# a proposed password before updating it.  
Manual page passwd(1) line 1 (press h for help or q to quit)
```

■ 한글 man page 프로젝트

- 현재 중단된 상태 - 그냥 영어로 읽으시오!
- 유닉스에 능숙한 개발자들이 쓴 것이기 때문에 초보자 입장에서는 불친절하거나 어려울 수 있음
- 매뉴얼에 대한 패러디



- q 키로 탈출 할 것.

리눅스 디렉토리 구조

■ tree 명령어로 디렉토리 구조 확인

```
tree [옵션] [디렉토리]
```

```
[root@cberi ~]# tree
.
├── thinclient_drives
├── 공개
├── 다운로드
├── 문서
├── 바탕화면
├── 비디오
├── 사진
├── 서식
└── 음악
```

- 옵션

- -L level : 레벨

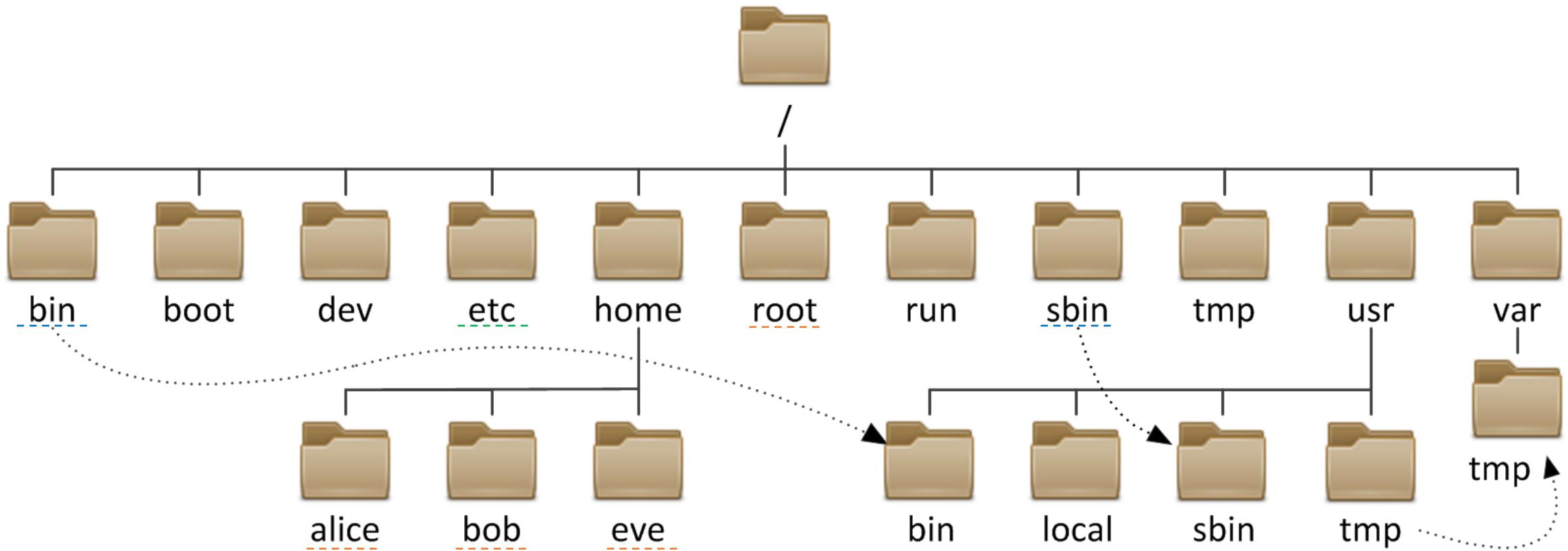
■ 루트 디렉토리 확인 (1레벨만)

```
[root@cberi ~]# tree -L 1 /
```

```
/
├── bin  -> usr/bin
├── boot
├── dev
├── etc
├── home
├── lib  -> usr/lib
├── lib64 -> usr/lib64
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin -> usr/sbin
├── srv
├── swapfile
├── sys
├── tmp
└── usr
└── var
```

리눅스 디렉토리 구조

- 설치 직후 기본 디렉토리 구조



리눅스 디렉토리 구조

위치	설명
/bin	binary(이진파일, 실행파일) 기본적인 명령어가 저장된 디렉토리 리눅스에서 자주 사용하는 mv, cp, rm등과 같은 명령어들이 이 디렉토리에 존재함
/sbin	system binary ifconfig, ethtool, halt, e2fsck와 같은 시스템 명령어들을 저장하고 있는 디렉토리
/boot	리눅스의 부팅에 필요한 정보를 가진 파일들이 있는 디렉토리
/dev	device 시스템 장치 파일을 저장하고 있는 디렉토리 즉, 물리적 장치가 파일화 되어 저장
/etc	et cetera 설정 파일을 두는 디렉토리
/lib	library 커널이 필요로 하는 각종 라이브러리 파일, 커널 모듈 파일 등이 존재하는 디렉토리
/opt	optional 추가 응용프로그램 패키지 설치 장소

위치	설명
/media	DVD, CD-ROM, USB등의 외부장치 들의 마운트 포인트로 사용하는 디렉토리
/proc	process '가상파일시스템'이라고 하는 곳으로 현재 메모리에 존재하는 작업들이 파일 형태로 존재하는 곳
/home	사용자들의 홈 디렉토리가 있는 곳 사용자를 추가하면 사용자의 id와 동일한 디렉토리가 이곳에 자동으로 생성
/root	관리자계정 root 사용자의 홈 디렉토리
/usr	일반 사용자들이 사용하는 디렉토리
/var	기타 모든 다용도로 사용될 수 있는 파일들이 저장되며 로그파일, 데이터베이스 캐싱파일, 웹서버 이미지 파일 등이 위치하는 디렉토리
/sys	디바이스를 관리하기 위한 가상파일시스템 디렉토리
/run	Run-time variable data를 관리한다. 부팅한 후의 시스템 정보를 관리하는 디렉토리이다.
/tmp	임시 파일을 저장하기 위한 디렉토리이다. 재부팅 시에 삭제되며, 정기적으로 10일 정도 간격으로 삭제된다.

절대 경로와 상대경로, 홈 폴더

■ 절대 경로

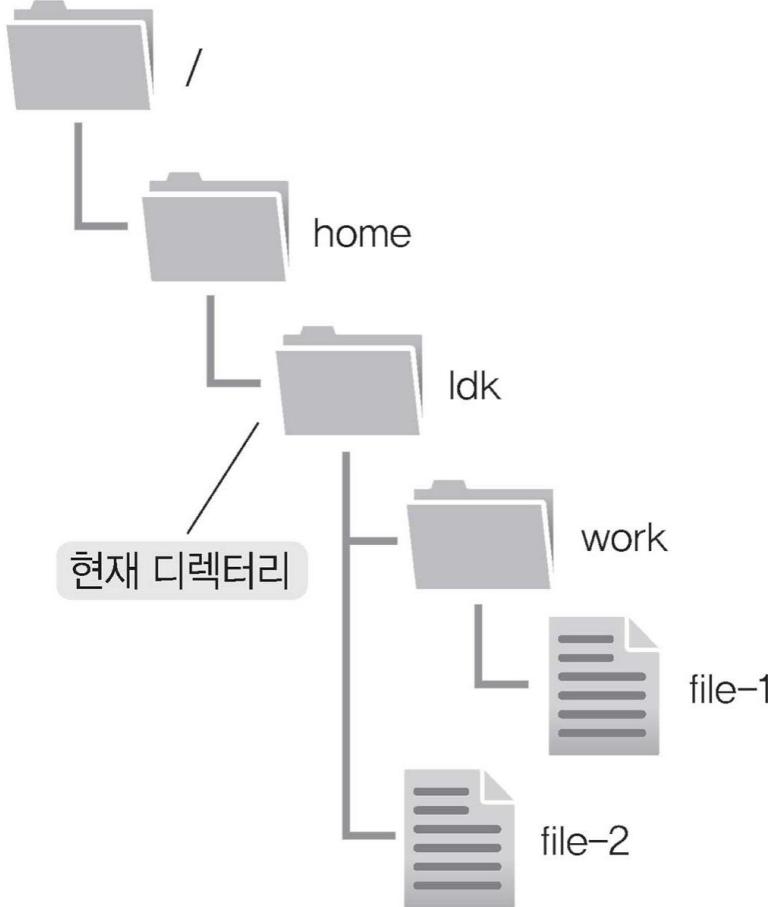
- 루트 디렉토리부터 해당 파일에 이르는 경로를 표시하는 것
- 해당 파일의 위치를 명확하게 나타내지 만, 디렉토리의 깊이가 깊으면 경로가 너무 길어지는 문제
- 시스템마다 절대 경로가 달라서 호환성 문제 가능성
- 사용자 **홈 폴더(~)**가 기준이 될 수도 있음.

■ 상대 경로

- 현재 디렉토리를 기준으로 표기하는 경로
- 특수 디렉토리인 .와 ..를 사용
- . : 현재 디렉토리를 의미
- .. : 상위 디렉토리를 의미
- 예)
 - ./test : 현재 디렉토리에 있는 test
 - ../../ : 상위 디렉토리의 상위 디렉토리

절대 경로와 상대경로, 홈 폴더

■ 예시



■ 비교표

상대 경로	절대 경로
.	/home/ldk
~/	~/ldk/
..	/home
~/..	~/..
../..	/
work 또는 ./work	/home/ldk/work
~/work	~/work
~ldk/work	~ldk/work
work/file-1 또는 ./work/file-1	/home/ldk/work/file-1
file-2 또는 ./file-2	/home/ldk/file-2

기본 명령어

■ ls : 리스트팅

ls [옵션] [디렉토리]

```
[root@vultr ~]# ls -lhF
total 0
[root@vultr ~]# ls -alhF
total 44K
dr-xr-x---. 4 root root 4.0K Jan 27 17:34 .
dr-xr-xr-x. 18 root root 4.0K Jan 27 03:48 ../
-rw-----. 1 root root 30 Jan 28 06:28 .bash_history
-rw-r--r--. 1 root root 18 May 11 2019 .bash_logout
-rw-r--r--. 1 root root 176 May 11 2019 .bash_profile
-rw-r--r--. 1 root root 239 Jan 25 15:04 .bashrc
drwx-----. 3 root root 4.0K Jan 25 15:05 .cache/
-rw-r--r--. 1 root root 100 May 11 2019 .cshrc
drwx-----. 2 root root 4.0K Jan 27 03:46 .ssh/
-rw-r--r--. 1 root root 129 May 11 2019 .tcshrc
```

■ 자주 사용하는 옵션

- l : 파일들을 나열할 때 자세히 출력한다.
- a : 숨김파일 포함 경로안의 모든 파일을 나열한다.
- R : 하부 디렉토리의 파일까지 모두 출력한다.
- h : K, M, G 단위를 사용하여 파일 크기를 사람이 보기 좋게 표시합니다.
- r : 출력 결과를 내림차순으로 정렬한다.
- t : 출력 결과를 파일이 수정된 시간을 기준으로 정렬한다
- F : 실행파일은 *, 경로 /, 소켓=, 링크 @ 등의 지시자로 출력한다.
- b : 알파벳 순으로 목록을 출력합니다.
- x : 확장자의 알파벳순으로 정렬하여 출력합니다.

기본 명령어

■ alias

```
cat [-p] [name[=value]]
```

```
[root@cberi ~]# alias lsl='ls -alhF'
[root@cberi ~]# lsl
합계 148K
dr-xr-x---. 19 root root 4.0K 1월 28 12:26 .
dr-xr-xr-x. 18 root root 4.0K 1월 12 10:34 ../
-rw-----. 1 root root 950 1월 27 21:25 .ICEauthority
-rw-----. 1 root root 113 1월 27 21:25 .Xauthority
-rw-----. 1 root root 23K 1월 29 03:28 .bash_history
-rw-r--r--. 1 root root 18 5월 12 2019 .bash_logout
-rw-r--r--. 1 root root 196 1월 16 22:17 .bash_profile
-rw-r--r--. 1 root root 239 11월 4 09:18 .bashrc
drwx-----. 13 root root 4.0K 1월 27 21:19 .cache/
drwx-----. 15 root root 4.0K 1월 12 16:42 .config/
-rw-r--r--. 1 root root 100 5월 12 2019 .cshrc
drwxr-xr-x. 5 root root 4.0K 1월 16 01:32 .local/
```

■ 설정된 alias 모두 출력

```
[root@cberi ~]# alias -p
alias cp='cp -i'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias lsl='ls -alhF'
alias mv='mv -i'
alias rm='rm -i'
alias xzegrep='xzegrep --color=auto'
alias xzfgrep='xzfgrep --color=auto'
alias xzgrep='xzgrep --color=auto'
alias zegrep='zegrep --color=auto'
alias zfgrep='zfgrep --color=auto'
alias zgrep='zgrep --color=auto'
```

웹 콘솔 활성화

■ 웹 콘솔

- 네트워크 환경에 있는 Linux 서버는 물론 로컬 시스템을 관리 및 모니터링하도록 설계된 Red Hat Enterprise Linux 8 웹 기반 인터페이스
- 활성화

```
login as: root
root@207.148.95.144's password:
Activate the web console with: systemctl enable --now cockpit.socket
```

```
[root@vultr ~]# systemctl enable --now cockpit.socket
Created symlink /etc/systemd/system/sockets.target.wants/cockpit.socket → /usr/lib/systemd/system/cockpit.socket.
```

```
[root@vultr ~]# netstat -lntp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 0.0.0.0:22              0.0.0.0:*            LISTEN    1388/sshd
tcp6     0      0 :::9090                ::::*               LISTEN    1/systemd
tcp6     0      0 :::22                 ::::*               LISTEN    1388/sshd
```

```
[root@vultr ~]#
```

웹 콘솔(Web Console) 접속

- `https://__.__.__.__:9090` 로 접속



연결이 비공개로 설정되어 있지 않습니다.

공격자가 **207.148.95.144**에서 정보(예: 비밀번호, 메시지, 신용카드 등)를 도용하려고 시도
증일 수 있습니다. [자세히 알아보기](#)

NET::ERR_CERT_AUTHORITY_INVALID

고급 버튼을 선택하고,
안전하지 않음으로 이동 선택



Chrome에서 가장 강력한 보안 기능을 사용하려면 [향상된 보호 모드를 사용 설정](#)
하세요.

고급

안전한 페이지로 돌아가기

도메인 네임 정하기

- 여러분이 운영하는 서버의 이름을 무엇으로 지을 것인가?
- 희망 도메인 네임 제출
 - <https://forms.gle/pF9QCpnSSpT48egu6>
- 이용가능 최상위 도메인
 - .shop
 - .site
 - .xyz

3. 당신의 이메일 주소는

내 답변

4. 희망하는 도메인 명은? *

짧은 이름이 좋음. 길면 타이핑 할 때 마다 힘듦.
(5~8글자 추천)

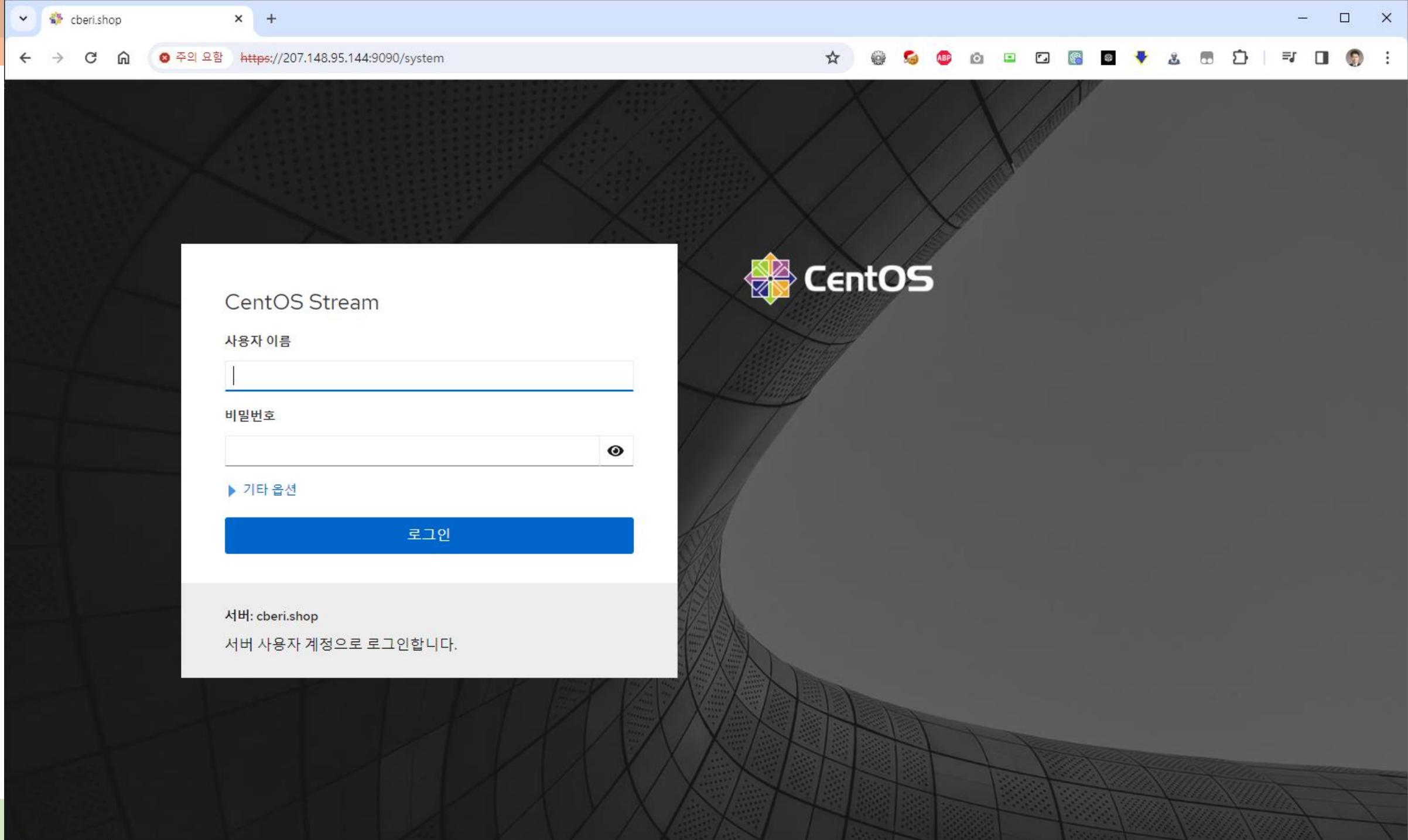
____.shop _____.site _____.xyz 사용 가능

ex) cberi.shop, cbero.site , iceman.xyz

내 답변

5. 위에서 지정한 도메인 생성에 실패했을 경우 두 번째로 희망하는

내 답변



개요 - root@cberi.shop

주의 요함 https://207.148.95.144:9090/system

root@
cberi.shop

검색

시스템

개요

기록

저장소

네트워킹

포드맨 컨테이너

계정

서비스

도구

소프트웨어 최신화

응용프로그램

진단 보고서

커널 덤프

터미널

cberi.shop CentOS Stream 8 실행 중

재시작

상태

! 1 서비스 실패

- 꾸러미 최신화 점검 중 ...

▲ 15 로그인 시도 실패

12월 26일 오후 11:34 pts/1의 175.205.86.54에서
마지막으로 성공한 로그인: 12월 26일 오후 11:34
[로그인 내역 보기](#)

사용량

중앙처리장치 97% 1 CPU

메모리 0.54 / 1.9 GiB

측정 항목과 내역 보기

시스템 정보

모델	Vultr VHP
자산 태그	70575508
장치 ID	7000c79093de42d79c79f6cbea10bf48
가동 시간	1일

하드웨어 세부 사항보기

설정

호스트 이름	cberi.shop 편집
시스템 시간	2023.12.26. 오후 2:57
도메인	도메인 가입
성능 프로파일	virtual-guest
암호화 정책	기본
보안 키	지문 표시

검색

시스템

개요

기록

저장소

네트워킹

포드먼 컨테이너

계정

서비스



도구

소프트웨어 최신화



응용프로그램

진단 보고서

커널 덤프

터미널

상태

 69 사용 가능한 최신화
마지막 접속 날짜: 7분 전

설정

자동 최신화 설정 없음

활성화

커널 실시간 패치 중 사용 안함

활성화

사용 가능한 최신화

 위험 경고: 웹 콘솔이 재시작됩니다 [자세한 정보...](#)[모든 최신화 설치](#)

이름	버전	심각도	상세정보
> alsa-sof-firmware	2023.09.2...		
> avahi-libs	0.7-26.el8		
> bpftool, kernel-headers, kernel-tools, kernel-tools-libs, ...	4.18.0-52...		
> cloud-init	23.1.1-12....		
> cmake, cmake-data, cmake-filesystem, cmake-rpm-macros	3.26.5-2.el8		
> cockpit, cockpit-bridge, cockpit-system, cockpit-ws, ...	306-1.el8		
> device-mapper-multipath, device-mapper-multipath-libs, kpartx	0.8.4-40.el8		
> dracut, dracut-config-rescue, dracut-network, dracut-squash	049-231.g...		
> git, git-core, git-core-doc, perl-Git	2.43.0-1.el8		
> kexec-tools	2.0.26-12....		

dnf (Dandified Yum)

■ dnf 개요

- YUM(Yellow dog Updater, Modified)
- yum의 단점인 느린 속도와 과다한 메모리 사용, 의존성 결정 느림 문제를 개선한 패키지 관리자
- RHEL/CentOS 8에 정식 도입
- 기존 사용자들을 위해 dnf를 백엔드로 사용하는 yum 유틸리티 제공
- 기존 yum의 명령어 그대로 사용가능

■ 기본사용법

기능	명령어
패키지 설치	yum install 패키지명
패키지 삭제	yum remove 패키지명
패키지 업데이트	yum update 패키지명
패키지 정보 확인	yum info 패키지명
패키지 검색	yum search 검색어
패키지 목록 보기 (설치된 항목 + 설치가능 항목)	yum list
설치된 패키지 목록 보기	yum list installed
저장소 확인하기	yum repolist

EPEL 저장소 설치

■ EPEL (Extra Packages for Enterprise Linux)

- RHEL이나 CentOS에 기본적으로 탑재되어 있지 않은 패키지를 제공 받기 위함
- repository(저장소) 확인

```
[root@cberi ~]# dnf repolist
```

레포지터리 ID

apprepo

appstream

baseos

extras

extras-common

레포지터리 이름

apprepo

CentOS Stream 8 - AppStream

CentOS Stream 8 - BaseOS

CentOS Stream 8 - Extras

CentOS Stream 8 - Extras common packages

```
[root@cberi ~]# dnf list installed epel-release
```

오류: 목록과 일치하는 패키지가 없습니다.

- 설치

```
[root@cberi ~]# dnf install epel-release
```

```
[root@cberi ~]# dnf list installed epel-release
```

설치된 꾸러미

epel-release.noarch

8-11.el8

@extras

locale(로케일) 설정

▪ locale

- 시스템 언어를 의미
- 실제 사용언어와 시스템 언어가 일치하지 않으면 글자 깨짐 현상 발생 할 수 있음

```
[root@localhost test5555]# ls -al
total 28
drwx----- 4 test5555 test5555 4096 8 14 02:22 .
drwxr-xr-x. 5 root      root    4096 8 14 01:41 ..
-rw-r--r--  1 test5555 test5555  18 11  8 2019 .bash_logout
-rw-r--r--  1 test5555 test5555 141 11  8 2019 .bash_profile
-rw-r--r--  1 test5555 test5555 312 11  8 2019 .bashrc
drwxr-xr-x  2 root      root    4096 8 14 02:22 .bashrc
drwxr-xr-x  2 root      root    4096 8 14 02:22 .bashrc
[root@localhost test5555]#
```

- 요즘엔 영문판도 UTF-8 인코딩을 사용하므로 글자 안 깨짐.
- 영문 메시지로 출력되는 것이 좋으면 바꾸지 말 것!

• 설정 가능한 모든 로케일 확인

```
[root@vultr ~]# locale -a
C
C.utf8
POSIX
en_AG
en_AU
en_AU.utf8
en_BW
en_BW.utf8
en_CA
en_CA.utf8
en_DK
en_DK.utf8
en_GB
en_GB.iso885915
en_GB.utf8
en_HK
en_HK.utf8
:
```

```
[root@vultr ~]# locale -a | grep -i kr
```

```
[root@cberi ~]# dnf install glibc-langpack-ko -y
Last metadata expiration check: 1:52:20 ago on Thu 12 Jan 2023 02:19:47 AM UTC.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
glibc-langpack-ko	x86_64	2.28-224.el8	baseos	361 k

```
Transaction Summary
```

```
Install 1 Package
```

```
Total download size: 361 k
```

```
Installed size: 2.5 M
```

```
Downloading Packages:
```

glibc-langpack-ko-2.28-224.el8.x86_64.rpm	757 kB/s
---	----------

Total	439 kB/s
-------	----------

```
Running transaction check
```

```
Transaction check succeeded.
```

```
Running transaction test
```

```
Transaction test succeeded.
```

```
Running transaction
```

Preparing :	
-------------	--

Installing :	glibc-langpack-ko-2.28-224.el8.x86_64
--------------	---------------------------------------

Running scriptlet:	glibc-langpack-ko-2.28-224.el8.x86_64
--------------------	---------------------------------------

Verifying :	glibc-langpack-ko-2.28-224.el8.x86_64
-------------	---------------------------------------

```
Installed:
```

glibc-langpack-ko-2.28-224.el8.x86_64

```
Complete!
```

```
[root@cberi ~]# locale -a | grep -i kr
ko_KR.euckr
ko_KR.utf8
```

```
[root@cberi ~]# localedctl
System Locale: LANG=en_US.UTF-8
VC Keymap: us
X11 Layout: us
```

```
[root@cberi ~]# localedctl set-locale LANG=ko_KR.utf8
```

```
[root@cberi ~]# localedctl
System Locale: LANG=ko_KR.utf8
VC Keymap: us
X11 Layout: us
[root@cberi ~]#
```

타임 존

■ 타임 존 확인

```
[root@cberi ~]# date  
Thu Jan 12 04:37:57 UTC 2023
```

지금 시간이
아닌데?

```
[root@cberi ~]# timedatectl  
    Local time: Thu 2023-01-12 05:04:23 UTC  
Universal time: Thu 2023-01-12 05:04:23 UTC  
      RTC time: Thu 2023-01-12 05:04:22  
    Time zone: UTC (UTC, +0000)  
System clock synchronized: yes  
      NTP service: active  
RTC in local TZ: no
```

■ 타임 존 설정

```
[root@cberi ~]# timedatectl set-timezone Asia/Seoul
```

```
[root@cberi ~]# date  
Thu Jan 12 14:05:17 KST 2023
```

```
[root@cberi ~]# timedatectl  
    Local time: Thu 2023-01-12 14:12:03 KST  
Universal time: Thu 2023-01-12 05:12:03 UTC  
      RTC time: Thu 2023-01-12 05:12:02  
    Time zone: Asia/Seoul (KST, +0900)  
System clock synchronized: yes  
      NTP service: active  
RTC in local TZ: n
```

시간 동기화

■ time 서버 설정

```
[root@cberi ~]# vi /etc/chrony.conf
# These servers were defined in the installation:
server 1.time.constant.com iburst
server 2.time.constant.com iburst
server 3.time.constant.com iburst
server time.bora.net iburst
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool
(http://www.pool.ntp.org/join.html).

# Record the rate at which the system clock
gains/losses time.
driftfile /var/lib/chrony/drift

# Allow the system clock to be stepped in the first
three updates
```

■ chronyd의 서비스 중지 / 재시작

```
[root@cberi ~]# systemctl stop chronyd
[root@cberi ~]# systemctl start chronyd
```

■ chronyd의 서비스 확인

```
[root@cberi ~]# chronyc sources
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^- hydrogen.constant.com      2   6    17      4   -1803us[ -146us
^- helium.constant.com       2   6    17      4   -6292us[ -6292us
^- lithium.constant.com      2   6    17      4   -377us[ -377us
^* time.bora.net             2   6    17      4   -117us[+1540us
```

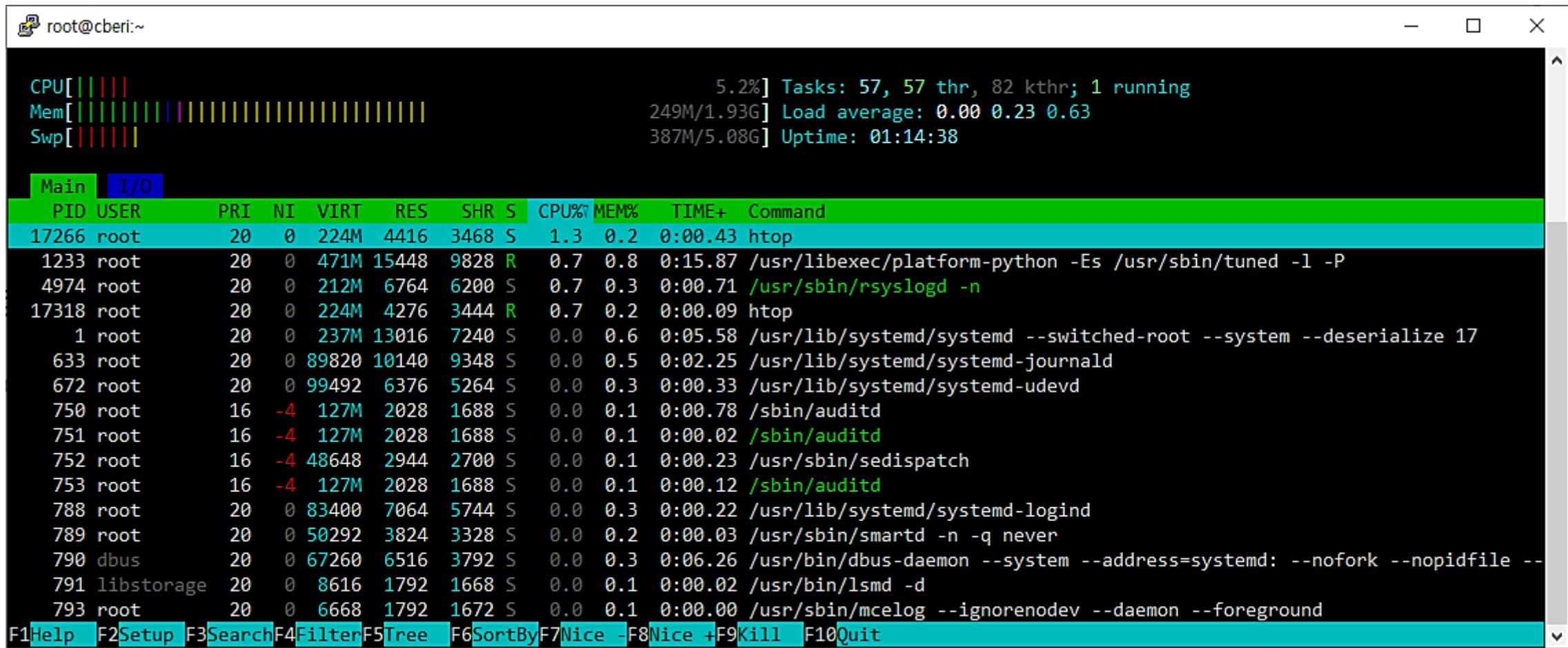
■ 메인보드 시간과 동기화

```
[root@cberi ~]# hwclock -w
[root@cberi ~]# hwclock -v
```

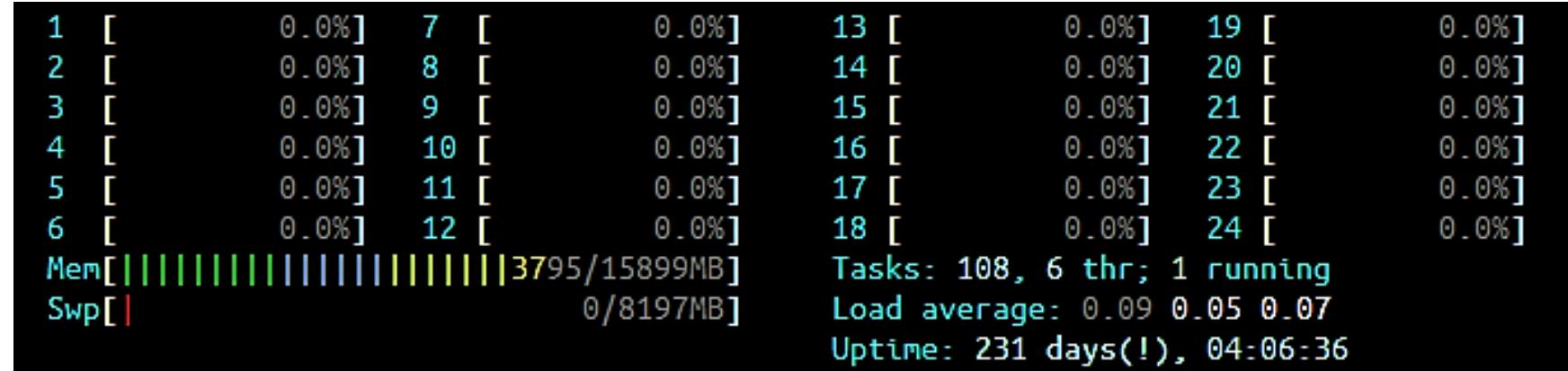
htop(실시간 시스템 모니터링) 설치

```
[root@cberi ~]# dnf install htop
```

```
[root@cberi ~]# htop
```



htop 사용법



■ CPU 상태

- CPU 코어를 프로세스가 점유하고 있는 비율을 나타냄
 - 각 Bar는 해당 코어의 사용된 % 를 표현

■ Memory

- 초록: 사용됨
 - 파랑: 버퍼
 - 노랑: 캐쉬

■ 평균 부하율 (Load Average)

- 첫번째 필드: 1분 평균 수치
 - 두번째 필드: 5분 평균 수치
 - 세번째 필드: 15분 평균 수치
 - 싱글 코어라면, 1.0이 100%를 의미
 - 쿼드 코어라면, 4.0이 100%를 의미

htop 사용법

- [F2 Setup]
 - F2 (또는 S[대문자]) 를 입력하면 설정 화면으로 진입합니다.
- [F3 Search]
 - F3 (또는 /) 를 입력하면 특정 프로세스를 검색할 수 있는 command bar를 화면에 보여줍니다. (VIM하고 유사합니다.)
- [F4 Filter]
 - F4 (또는 -, \) 를 입력하면 걸러낼 명령어를 입력할 수 있는 기능입니다.
- [F5 Tree/Sorted]
 - F5(또는 t) 를 입력하면 정렬(트리형식으로)해서 화면에 노출해줍니다. + 를 특정 프로세스에서 입력하면 해당 프로세스가 열고 있는 파일의 스냅샷을 보여줍니다.
 - 트리 노출 방식일 때는 정렬기능으로 변경됩니다. 즉 트리 노출 이전의 형태로 돌아갑니다.
- [F6 SortBy/Collap]
 - F6을 입력하면 정렬 기능으로 들어갑니다. 좌측에 정렬하고 싶은 필드가 노출되죠. 만일 트리화면일 경우, 각 부모 트리를 접었다 펼쳤다 할 수 있는 기능으로 변경됩니다.
- [F7 Nice -]
 - 프로세스의 우선순위를 높입니다. 또 다른 단축키로는 "]" 이 있습니다.
- [F8 Nice +]
 - 프로세스의 우선순위를 낮춥니다. 또 다른 단축키로는 "[" 이 있습니다.
- [F9 Kill]
 - 특정 프로세스 종료시키는 기능입니다. 다수의 프로세스를 종료하려면 스페이스를 눌러 체크(노란색으로 변함) 시킨 후 F9 또는 "k" 를 입력하시면 됩니다.
- [F10 Quit]
 - htop 종료합니다.

네트워크 설정

■ IP주소 및 netmask 확인

```
[root@vultr ~]# ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 207.148.95.144 brd 207.148.95.255 netmask 255.255.254.0
        broadcast 207.148.95.255
    inet6 2001:19f0:7001:59b1:5400:4ff:feb3:64d6 prefixlen 64 scopeid
          0x0<global>
        inet6 fe80::5400:4ff:feb3:64d6 prefixlen 64 scopeid 0x20<link>
    ether 56:00:04:b3:64:d6 txqueuelen 1000 (Ethernet)
    RX packets 168916 bytes 228462770 (217.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 170002 bytes 18475015 (17.6 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

■ IP:

207.148.95.144

■ netmask:

255.255.254.0

네트워크 설정

▪ Gateway 및 네임서버 확인

```
[root@cberi ~]# ip route
default via 207.148.94.1 dev enp1s0 proto dhcp src 207.148.95.144
metric 100
169.254.169.254 via 207.148.94.1 dev enp1s0 proto dhcp src
207.148.95.144 metric 100
207.148.94.0/23 dev enp1s0 proto kernel scope link src
207.148.95.144 metric 100
```

```
[root@cberi ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search guest
nameserver 108.61.10.10
nameserver 2001:19f0:300:1704::6
```

▪ Gateway:

207.148.94.1

▪ DNS1:

108.61.10.10

▪ DNS2:

2001:19f0:300:1704::6

네트워크 설정

▪ CentOS cockpit 이용

The screenshot shows the CentOS cockpit web interface. On the left, a sidebar menu includes '검색' (Search), '시스템' (System), '개요' (Overview), '기록' (Logs), '저장소' (Repositories), '네트워킹' (Networking) which is selected and highlighted in dark grey, '포드맨 컨테이너' (Podman Containers), '계정' (Accounts), '서비스' (Services) with a red exclamation mark icon, '도구' (Tools), and '소프트웨어 최신화' (Software Updates). The main content area displays network details for the interface 'enp1s0' (Red Hat, Inc.):

상태	207.148.95.14
캐리어	네
일반	<input checked="" type="checkbox"/> 자동으로 연결
IPv4	자동 추가 DNS 1.1.1
IPv6	자동 편집
MTU	자동 편집

Below this, a '연결장치' (Connection Device) section shows '이름' (Name) as 'enp0s2'. At the bottom are '저장' (Save) and '취소' (Cancel) buttons.

A modal window titled 'IPv4 설정' (IPv4 Configuration) is open on the right, detailing the IPv4 configuration for the interface:

주소	자동
DNS	<input checked="" type="checkbox"/> 자동
서버	1.1.1.1
서버	8.8.4.4
DNS 검색 도메인	<input checked="" type="checkbox"/> 자동
경로	<input checked="" type="checkbox"/> 자동

Buttons for '저장' (Save) and '취소' (Cancel) are at the bottom of the modal.



DNS: 인터넷의 디렉터리

인터넷의 거의 모든 작업은 DNS 요청에서 시작됩니다. DNS는 인터넷의 디렉터리입니다. 링크를 클릭하고, 앱을 열고, 이메일을 보낼 때 장치가 가장 먼저 수행하는 작업은 디렉터리에 다음을 묻는 것입니다. 이를 어디에서 찾을 수 있나요?

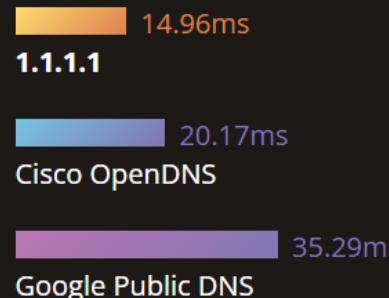
불행하게도 DNS는 기본적으로 느리며 안전하지 않습니다. ISP와 인터넷을 수신하는 모든 사람은 콘텐츠가 암호화되어 있더라도 사용자가 방문하는 모든 사이트와 사용자가 사용하는 모든 앱을 볼 수 있습니다. 끔찍하게도 일부 DNS 공급자는 사용자의 인터넷 활동에 대한 데이터를 팔거나 광고 목적으로 사용합니다.

이는 아주 기분 나쁜 일입니다. 이 기분 나쁜 상황을 방지할 수 있는 대안이 있습니다. **1.1.1.1**입니다.

비교 불가능한 속도

Cloudflare가 만든 1.1.1.1은 가장 빠른 인터넷 DNS 딕렉터리입니다. 이 주장은 직접 확인할 수 있습니다. 독립 DNS 모니터인 [DNSPerf](#)의 순위에 따르면 1.1.1.1은 전 세계에서 가장 빠른 DNS 서비스입니다.

인터넷에서 수행하는 거의 모든 작업은 DNS 요청에서 시작되므로 모든 기기에서 가장 빠른 DNS 디렉터리를 선택하면 온라인으로 수행하는 거의 모든 활동이 가속화됩니다.



네트워크 설정

- 네트워크 설정 오류로 SSH 접속이 불가능하게 되면...

The screenshot shows a VNC session titled "noVNC (Linux00) - ID 216f2944-b56e-4ee5-b908-5f84f073f5cf - Chrome". The main window displays a terminal log from "cloud-init". The log shows the setup of swap space, network configuration (IPv6, default qdisc bbr), and SSH host key fingerprints. It ends with the message "Cloud-init v. 22.4.2 finished at Fri, 27 Jan 2023 03:48:20 +0000. Datasource DataSourceVultr. Up 128.13 seconds". Below the log, there's a user session where "root@vultr:~\$" is typed.

The right side of the screen features a sidebar with the following sections:

- Kubernetes**: Shows a cluster icon and a button labeled "+".
- Databases**: Shows a database icon.
- Sort**: A dropdown menu set to "Label".
- Status**: Shows the server is "Running".
- Server Details**: Includes a "View Console" button, which is highlighted with a red dashed border.
- Actions**: Includes "Server Stop", "Server Restart", "Server Reinstall", and "Server Destroy".

At the bottom of the sidebar, there are icons for "Products", "Billing", "Support", "Referral Program", and "Account".

At the very bottom of the screen, the footer reads "2048.00 MB Regular Cloud Compute - 108.61.161.138".

Linux 사용자 (GUI 사용 가능)

The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical sidebar with various system management options: 검색 (Search), 시스템 (System), 개요 (Overview), 기록 (Logs), 저장소 (Storage), 네트워킹 (Networking), 포드맨 컨테이너 (Podman Container), 계정 (Accounts), 서비스 (Services), 도구 (Tools), 소프트웨어 최신화 (Software Updates), 응용프로그램 (Applications), 진단 보고서 (Diagnostic Report), and 커널 뎁프 (Kernel Dump). The '계정' (Accounts) option is currently selected.

The main window displays a '신규 계정 만들기' (Create New User) dialog. The '그룹' (Group) tab is selected, showing 'wheel: 1' and 'root: 5'. The '계정' (Account) tab is also visible. The form fields include:

- 성명 (Name): Empty input field.
- 사용자 이름 (User Name): Empty input field.
- 홈 디렉토리 (Home Directory): '디렉토리 경로' (Directory Path) dropdown set to '/bin/bash'.
- 쉘 (Shell): '/bin/bash' dropdown.
- 사용자 ID (User ID): '1002' input field.
- 인증 (Authentication):
 - 비밀번호 사용 (Password Use):
 - 처음 로그인에서 비밀번호를 변경해야 합니다 (Change password on first login).
 - 비밀번호 인증을 허용하지 않습니다 (Do not allow password authentication).
- 비밀번호 (Password): Empty input field with eye icon.
- 비밀번호 확인 (Password Confirmation): Empty input field with eye icon.

At the bottom of the dialog are two buttons: '생성' (Create) and '취소' (Cancel).

On the right side of the screen, there is a sidebar with 'ID 검색' (ID Search) and '신규 계정 만들기' (Create New User) buttons. Below these are lists of existing users: 'wheel' and 'admin (root)'.

At the bottom left, there is a note: '두 사용자 추가' (Add two users) with a bulleted list: '• www' and '• word'.

Linux 사용자

■ useradd : 사용자 추가

```
[root@cberi ~]# useradd www  
[root@cberi ~]# passwd www  
www 사용자의 비밀 번호 변경 중  
새 암호:  
새 암호 재입력:  
passwd: 모든 인증 토큰이 성공적으로 업데이트  
되었습니다.  
[root@cberi ~]#
```

■ su: 사용자 전환

```
[root@cberi ~]# su - www  
[www@cberi ~]$ pwd  
/home/www
```

■ 확인

```
[www@cberi ~]$ ls -alhF /home  
합계 16K  
drwxr-xr-x. 4 root root 4.0K 12월 27 15:49 ./  
dr-xr-xr-x. 18 root root 4.0K 12월 27 01:01 ../  
drwx-----. 3 1000 1000 4.0K 12월 25 23:15 linuxuser/  
drwx-----. 2 www www 4.0K 12월 27 15:58 www/  
  
[www@cberi ~]$ cd /  
[www@cberi /]$ cd ~  
[www@cberi ~]$ pwd  
/home/www  
[www@cberi ~]$ cd ~www  
[www@cberi ~]$ pwd  
/home/www  
[www@cberi ~]$ cd ~linuxuser  
-bash: cd: /home/linuxuser: 허가 거부  
[www@cberi ~]$ exit
```

Linux 사용자

■ 사용자 조회

```
[root@cberi ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
:
linuxuser:x:1000:1000:Linux User:/home/linuxuser:/bin/bash
www:x:1001:1001::/home/www:/bin/bash
```

■ 그룹 조회

```
[root@cberi ~]# cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mem:x:8:
wheel:x:10:
cdrom:x:11:
mail:x:12:
man:x:15:
floppy:x:19:
video:x:39:
ftp:x:50:
:
linuxuser:x:1000:linuxuser
www:x:1001:
```

Linux 사용자

■ userdel

- 사용자 삭제

```
[root@cberi home]# userdel --help
Usage: userdel [options] LOGIN

Options:
  -f, --force          force some actions that would fail otherwise
                       e.g. removal of user still logged in
  -h, --help            or files, even if not owned by the user
                       display this help message and exit
  -r, --remove          remove home directory and mail spool
  -R, --root CHROOT_DIR directory to chroot into
  -P, --prefix PREFIX_DIR prefix directory where are located the
  /etc/* files
  -Z, --selinux-user    remove any SELinux user mapping for the user
```

■ 실습

- r 옵션 사용

```
[root@cberi ~]# userdel -r linuxuser
[root@cberi ~]# ls -lhF /home
합계 16K
drwxr-xr-x.  4 root root 4.0K 12월 27 15:49 ./
dr-xr-xr-x. 18 root root 4.0K 12월 27 01:01 ../
drwx-----.  2 www www 4.0K 12월 27 15:58 www/
```

기본 명령어

■ mkdir (make directory)

: 디렉토리 만들기

```
[root@cberi ~]$ su - www  
[www@cberi ~]$ mkdir html  
[www@cberi ~]$ mkdir logs  
[www@cberi ~]$ ls -l  
합계 8  
drwxrwxr-x. 2 www www 4096 1월 15 12:27 html  
drwxrwxr-x. 2 www www 4096 1월 15 12:34 logs
```

■ cd (change directory)

: 작업 디렉토리 변경

```
[www@cberi ~]$ cd html  
[www@cberi html]$ pwd  
/home/www/html  
[www@cberi html]$ cd ~  
[www@cberi www]$ pwd  
/home/www
```

■ touch : 빈 파일 생성

```
[root@cberi ~]$ cd html  
[www@cberi html]$ touch index.html  
[www@cberi html]$ ls -l  
합계 0  
-rw-rw-r--. 1 www www 0 1월 15 12:27 index.html
```

■ tree : 트리구조 확인

```
[www@cberi ~]$ tree  
.└─ html  
1 directory, 0 files
```

```
[www@cberi html]$ tree /home/  
/home/  
└─ www  
   ├─ html  
   └─ index.html  
   └─ logs  
2 directories, 1 file
```

기본 명령어

■ cat

```
cat [옵션] <파일명>
```

```
[root@cberi ~]# cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
[root@cberi ~]# cat .bash_logout
# ~/.bash_logout
```

- 라인 번호 출력

```
[root@cberi ~]# cat -n .bash_profile
1 # .bash_profile
2
3 # Get the aliases and functions
4 if [ -f ~/.bashrc ]; then
5     . ~/.bashrc
6 fi
7
8 # User specific environment and startup programs
9
10 PATH=$PATH:$HOME/bin
11
12 export PATH
```

기본 명령어

■ cat (concatenate)

- 파일 병합

```
[root@cberi ~]# cat .bash_profile .bash_logout > merge.txt
[root@cberi ~]# cat merge.txt
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
# ~/.bash_logout
```

- 입력내용을 파일로 생성

```
[root@cberi ~]# cat > hi.txt
hi world.
I'm newbie.
[Ctrl]+D
[root@cberi ~]# cat hi.txt
hi world.
I'm newbie.
```

기본 명령어

■ less : 파일 스크롤 표시

```
less [옵션] <파일명>
```

```
[root@cberi ~]# less /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
:
```

■ 스크롤 조작 명령어

명령어	내용
[Space], [f], [Ctrl] + [v]	한 화면 아래로 스크롤
[b], [Meta] + [v]	한 화면 위로 스크롤
[j], [Ctrl] + [n], [Enter]	한 행 아래로 스크롤
[k], [Ctrl] + [p]	한 행 위로 스크롤
[q]	less 명령어 종료

- 요즘에는   키도 지원한다.

기본 명령어

- cp : (copy) 파일/디렉토리 복사

```
cp [옵션] <복사할 파일> ... <복사할 위치>
```

```
[root@cberi ~]# cp hi.txt hello.txt
[root@cberi ~]# ls -lhF
합계 48K
-rw-r--r--. 1 root root    10  1월 29 20:01 hello.txt
-rw-r--r--. 1 root root    10  1월 29 20:40 hi.txt
-rw-r--r--. 1 root root   214  1월 29 20:19 merge.txt
[root@cberi ~]# mkdir text
[root@cberi ~]# cp hello.txt hi.txt text/
[root@cberi ~]# cp *.txt text/
cp: overwrite 'text/hello.txt'? y
cp: overwrite 'text/hi.txt'? y
[root@cberi ~]# ls -lhF text/
합계 12K
-rw-r--r--. 1 root root    10  1월 29 20:43 hello.txt
-rw-r--r--. 1 root root    10  1월 29 20:43 hi.txt
-rw-r--r--. 1 root root   214  1월 29 20:43 merge.txt
```

- 폴더 통째로 복사

```
[root@cberi ~]# cp -R text/ test/
[root@cberi ~]# tree text/ test/
text
├── hello.txt
├── hi.txt
└── merge.txt
test
├── hello.txt
├── hi.txt
└── merge.txt
```

- 옵션

- R, -r : 하위 디렉토리/파일 포함하여 적용
- i, --interactive : 덮어쓰기 전에 확인
- f, --force : 묻지말고 그냥 해

기본 명령어

■ mv : (move)

- 파일/디렉토리 이동 (이름변경)

```
mv [옵션] <이동할 파일> ... <이동할 위치>
```

```
[root@vultr ~]# mkdir test  
[root@vultr ~]# ls  
hello.txt  hi.txt  merge.txt  test  text  
[root@vultr ~]# mv merge.txt sum.txt  
[root@vultr ~]# ls  
hello.txt  hi.txt  sum.txt  test  text  
[root@vultr ~]# mv sum.txt test/  
[root@vultr ~]# tree test/  
test/  
├── hello.txt  
├── hi.txt  
├── merge.txt  
└── sum.txt
```

■ rm : (remove) 삭제

```
[root@vultr ~]# cd ~/test  
[root@vultr test]# rm hello.txt hi.txt  
[root@vultr test]# rm -f *
```

■ rmdir : (remove dir) 폴더 삭제

```
[root@vultr ~]# cd ~/test  
[root@vultr ~]# rmdir test  
[root@vultr ~]# ls test  
ls: cannot access 'test': No such file or directory
```

■ 하위 폴더 포함 모두 삭제

```
[root@vultr ~]# cd ~  
[root@vultr ~]# rm -Rf text  
[root@vultr ~]# ls text  
ls: cannot access 'text': No such file or directory
```

링크(별명) 만들기

■ 하드 링크

- 한 개의 파일에 이름을 여러 개 붙임

```
ln TARGET LINK_NAME
```

```
ln file1 file2
```

- file1에 접근하는 것은 file2에 접근하는 것과 동일
- 하드 링크로 만든 파일은 원본/복사본 구분이 없음. 모두 원본임.
- 둘 중 하나만 삭제 시 지정한 파일만 삭제됨.
- 모든 하드 링크 파일이 삭제 되어야 사라짐.
- 디렉토리는 하드링크 불가.

■ 심볼릭 링크

- 바로 가기의 개념

```
ln -s TARGET LINK_NAME
```

- 디렉토리도 링크 가능, 다른 디스크의 파일도 링크 가능.
- 심볼릭 링크 삭제는 원본에 어떠한 영향도 주지 않음.
- 그 반대는 깨진 링크
- 실습

```
[root@cberi ~]# ln -s hello.txt hi.txt
[root@cberi ~]# ls -lhF *.txt
-rw-r--r--. 3 root root 10 1월 29 20:01 hello.txt
lrwxrwxrwx. 1 root root 9 1월 30 00:42 hi.txt -> hello.txt
```

grep, 파일(|)

■ grep

```
grep [옵션] <검색패턴> [<파일이름>]
```

- 문자열을 검색하는 명령어
- 예시

```
[root@cberi ~]# grep bash /etc/passwd
root:x:0:0:root:/root:/bin/bash
www:x:1000:1000::/home/www:/bin/bash
laravel:x:1001:1001::/home/laravel:/bin/bash
node:x:1002:1002::/home/node:/bin/bash
yii:x:1003:1003::/home/yii:/bin/bash
django:x:1004:1004::/home/django:/bin/bash
```

- 옵션

- -i : 대소문자 구별하지 않음
- -n : 줄번호 출력
- -v : 나타나지 않는 열 출력

■ 파일 라인 (|)

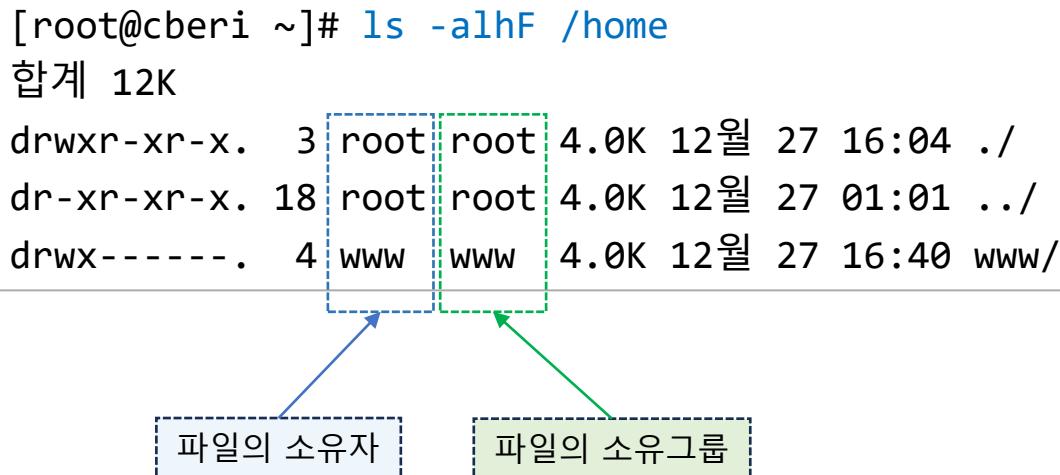
- 한 명령어의 실행결과를 다른 명령어의 입력으로 사용

```
[root@cberi ~]# ls /etc | grep cron
anacrontab
cron.d
cron.daily
cron.deny
cron.hourly
cron.monthly
cron.weekly
crontab
```

파일의 소유자와 그룹

■ 소유자

- 리눅스의 모든 파일(폴더)에는 소유자가 존재

```
[root@cberi ~]# ls -alhF /home
합계 12K
drwxr-xr-x.  3 root root 4.0K 12월 27 16:04 .
dr-xr-xr-x. 18 root root 4.0K 12월 27 01:01 ..
drwx-----.  4 www www 4.0K 12월 27 16:40 www/

```

파일의 소유자 파일의 소유그룹

- 파일의 소유자는 해당 파일에 대한 접근 권한을 자유롭게 설정 가능

- 새로운 파일(폴더)을 만들면 현재 사용자 소유로 자동 지정

```
[root@cberi home]# su - www
[www@cberi ~]$ cd html/
[www@cberi html]$ touch temp.html
[www@cberi html]$ ls -alhF
합계 8.0K
drwxrwxr-x.  2 www www 4.0K 12월 27 22:47 .
drwx-----.  4 www www 4.0K 12월 27 16:40 ../
-rw-rw-r--.  1 www www     0 12월 27 16:42 index.html
-rw-rw-r--.  1 www www     0 12월 27 22:47 temp.html
[www@cberi html]$
```

파일의 소유자와 그룹

■ 그룹

- 사용자들을 묶은 개념
- 계정 생성시 같은 이름의 그룹이 자동 생성됨
- 한 사용자는 여러 그룹에 소속될 수 있음

■ 현재 소속된 그룹 확인

```
[root@cberi home]# groups  
root
```

▶ 그룹 wheel:1 root:5 tss:2 74 더 알아보기...

■ 그룹 추가/삭제

```
groupadd [options] GROUP
```

```
groupdel [options] GROUP
```

■ 사용자를 그룹에 추가

```
usermod -a -G groupname username
```

• 예시

```
[root@cberi home]# usermod -a -G wheel root  
  
[root@cberi home]# cat /etc/group | grep root  
root:x:0:  
wheel:x:10:root
```

파일의 소유자와 그룹

■ 편집

root

세션 종료 삭제

성명	root
사용자 이름	root
그룹	wheel <input checked="" type="button"/> root
마지막 로그인	로그인 상태
옵션	<input type="checkbox"/> 대화형 비밀번호 허용 안 함 ③ 계정 잠금을 하지 않습니다 편집
비밀번호	비밀번호 설정 강제 변경 비밀번호가 만료되어서는 안됩니다 편집
홈 디렉토리	/root
쉘	/bin/bash 변경

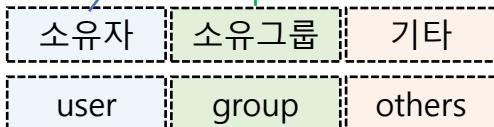
파일의 퍼미션

■ 퍼미션

- 리눅스의 각 파일에는 '누구에게 어떤 권한을 허가할 것인가?'에 대한 정보가 설정됨

파일타입

```
[root@cberi ~]# ls -l /bin/cat  
-rwxr-xr-x. 1 root root 38360 10월 15 14:11 /bin/cat
```

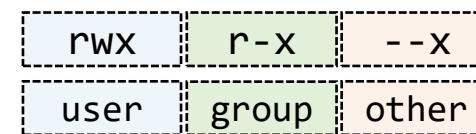


- 파일 타입
 - : 파일
 - d : 디렉토리
 - l : 심볼릭 링크

■ 의미

r	w	x
읽기	쓰기	실행

• 예시



- 소유자는 읽기/쓰기/실행 가능
- 소유그룹은 읽기/실행 가능
- 그 밖의 사용자는 실행만 가능

파일의 퍼미션

■ 예시

```
[root@cberi ~]# ls -l /etc/crontab  
-rw-r--r--. 1 root root 451 Jan 12 2021 /etc/crontab
```

소유자	소유그룹	기타
rw-	r--	r--

- 소유자는 읽기/쓰기 가능
- 소유 그룹은 읽기만 가능
- 그 밖의 사용자는 읽기만 가능

■ 테스트

```
[root@cberi etc]# ls -alhF /etc/shadow  
-----. 1 root root 1.5K 1월 14 18:31 /etc/shadow  
[root@cberi etc]# su - www
```

```
[www@cberi ~]$ cat /etc/shadow  
cat: /etc/shadow: 허가 거부
```

```
[www@cberi ~]$ touch test.txt  
[www@cberi ~]$ chmod 000 test.txt  
[www@cberi ~]$ ls -l  
합계 11  
-----. 1 www www 11 1월 14 22:57 test.txt
```

```
[www@cberi ~]$ cat test.txt  
cat: test.txt: 허가 거부
```

디렉토리의 퍼미션

■ 퍼미션

```
[root@cberi ~]# ls -l /home/
total 12
drwxr-xr-x 4 www www 4096 Jan  2 00:03 sqladmin
drwxr-xr-x 4 word word 4096 Jan 12 18:57 word
drwxr-xr-x 7 www www 4096 Jan 12 12:10 www
```

파일타입



- 파일 타입
 - : 파일
 - d : 디렉토리
 - l : 심볼릭 링크

■ 의미

기호	의미
r	디렉토리 내부의 파일 리스트 취득 가능
w	디렉토리 내부에 파일 또는 디렉토리 생성 가능
x	디렉토리로 이동 가능

- 왼쪽 sqladmin, word, www 디렉토리 퍼미션의 의미를 추측해보자.

chmod 명령어

- 기호 모드로 퍼미션 변경

```
chmod [ugoa] [+-=] [rwx] <파일이름>
```

- 사용자 지정기호

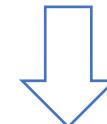
기호	의미
u	(user), 소유자
g	(group), 소유그룹
o	(others), 기타
a	(all), uga 모두

- + - = : 추가/금지/설정
- rwx : 읽기/쓰기/실행

- 기호 모드로 퍼미션 추가

```
chmod go+w file.txt
```

소유자	소유그룹	기타
r w -	r - -	r - -



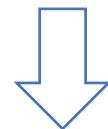
소유자	소유그룹	기타
r w -	r w -	r w -

chmod 명령어

- 기호 모드로 퍼미션 제거

```
chmod go-rw file.txt
```

소유자	소유그룹	기타
r w -	r w -	r w -



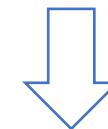
소유자	소유그룹	기타
r w -	- - -	- - -

- go에 rw를 삭제(금지)

- 기호 모드로 퍼미션 설정

```
chmod ugo=w file.txt
```

소유자	소유그룹	기타
r w -	- - -	- - -



소유자	소유그룹	기타
- w -	- w -	- w -

- ugo에 w를 설정

chmod 명령어

- 수치 모드로 퍼미션 변경

```
chmod <8진수8진수8진수> <파일이름>
```

- 퍼미션 수치

의미	숫자
읽기(r)	4
쓰기(w)	2
실행(x)	1

- 예시

- $rwx = 4+2+1 = 7$
- $rw = 4+2 = 6$

- 수치 모드로 퍼미션 변경

```
chmod 751 dir
```

소유자	소유그룹	기타
r w x	r - x	- - x
4 2 1	4 - 1	- - 1

- 서로 다른 계정으로 터미널 열어서 상대방 홈 폴더 진입 테스트

- www와 word 계정으로 실행

소유권 변경

▪ 명령어 형식

```
chown [옵션] owner[:group] file ...
```

- change the owner of a file
- 파일 또는 디렉토리의 소유자와 소유그룹을 변경한다
- 옵션
 - -f, --silent, --quiet
 - 파일 권한 변경 실패 시 에러를 출력하지 않음.
 - -R, --recursive
 - 하위 디렉터리에 있는 모든 디렉터리와 파일을 변경함.
 - -v, --verbose
 - 각 파일에 대해 변경한 정보나 변경되지 않은 정보를 상세히 출력함.

▪ 파일에 적용

```
[root@cberi word]# cd /home/word/  
[root@cberi word]# ls -l test.txt  
-rw-r--r--. 1 word word 0 1월 16 11:36 test.txt  
[root@cberi word]# chown www:www test.txt  
[root@cberi word]# ls -l test.txt  
-rw-r--r--. 1 www www 0 1월 16 11:36 test.txt  
[root@cberi word]#
```

▪ 폴더 전체에 적용

```
[root@cberi word]# ls -ld logs/  
drwxrwxr-x. 2 word word 4096 1월 14 13:28 logs/  
[root@cberi word]# chown -Rc :www logs/  
changed ownership of 'logs/access.log' from root:root  
to :www  
changed ownership of 'logs/error.log' from root:root  
to :www  
[root@cberi word]# ls -ld logs/  
drwxrwxr-x. 2 word word 4096 1월 14 13:28 logs/  
[root@cberi word]#
```

systemd(system daemon)

■ 개요

- systemd(=systemd init)은 리눅스의 가장 최신 init 구현으로 PID 1번을 사용하며 시스템이 부팅될 때 최초로 실행되는 프로세스
- systemd는 초기화 스크립트, 로그, 서비스, 네트워크 등 시스템의 모든 부분을 관리
- 제어 명령어로는 systemctl 사용

systemctl [OPTIONS] {COMMAND}

■ 명령어 예시

명령어	설명
systemctl start name.service	서비스 시작
systemctl stop name.service	서비스 중지
systemctl restart name.service	서비스 재시작
systemctl try-restart name.service	서비스가 구동중이었을 경우 에만 재시작
systemctl reload name.service	설정 재구동
systemctl status name.service	서비스 구동 여부 확인
systemctl list-units --type service --all	모든 서비스의 상태 표시

systemctl 실습

▪ systemctl status 유닛명

- 현재 상태 확인

```
[root@cberi ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2023-01-12 15:39:37 KST; 6h ago
    Docs: man:firewalld(1)
   Main PID: 823 (firewalld)
     Tasks: 3 (limit: 12392)
    Memory: 11.6M
      CGroup: /system.slice/firewalld.service
              └─823 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid
```

```
1월 12 15:39:35 cberi.kro.kr systemd[1]: Starting firewalld - dynamic firewall daemon...
1월 12 15:39:37 cberi.kro.kr systemd[1]: Started firewalld - dynamic firewall daemon.
1월 12 15:39:37 cberi.kro.kr firewalld[823]: WARNING: AllowZoneDrifting is enabled. >
```

systemctl 실습

■ systemctl stop 유닛명

- 데몬 중지

```
[root@cberi ~]# systemctl stop firewalld
```

■ systemctl start 유닛명

- 데몬 시작

```
[root@cberi ~]# systemctl start firewalld
```

■ systemctl restart 유닛명

- 데몬 재시작

```
[root@cberi ~]# systemctl restart firewalld
```

■ systemctl disable 유닛명

- 데몬 등록 해제(부팅시 시작 안됨)

```
[root@cberi ~]# systemctl disable firewalld
```

■ systemctl enable 유닛명

- 데몬 등록(부팅시 자동 시작)

```
[root@cberi ~]# systemctl enable firewalld
```

systemctl 실습

■ systemctl list-units

- 실행중인 유닛 조회

```
# systemctl list-units --all --state=active
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
proc-sys-fs-binfmt misc.automount	loaded	active	waiting	Arbitrary Executable File Format
dev-cdrom.device	loaded	active	plugged	QEMU DVD-ROM
dev-disk-by\x2did-ata\x2dQEMU_DVD\x2dROM_QM00005.device	loaded	active	plugged	QEMU DVD-ROM
dev-disk-by\x2dpartuuid-3a490f39\x2d01.device	loaded	active	plugged	/dev/disk/by-partuu
dev-disk-by\x2xpath-pci\x2d0000:00:1f.2\x2data\x2d3.device	loaded	active	plugged	QEMU DVD-ROM
dev-disk-by\x2xpath-pci\x2d0000:03:00.0.device	loaded	active	plugged	/dev/disk/by-path/p
dev-disk-by\x2xpath-pci\x2d0000:03:00.0\x2dpart1.device	loaded	active	plugged	/dev/disk/by-path/p
dev-disk-by\x2xpath-virtio\x2dpcl\x2d0000:03:00.0.device	loaded	active	plugged	/dev/disk/by-path/v
dev-disk-by\x2xpath-virtio\x2dpcl\x2d0000:03:00.0\x2dpart1.device	loaded	active	plugged	/dev/disk/by-p
dev-disk-by\x2duuid-fbbc43f1\x2d7f62\x2d4259\x2d8112\x2d28c26224aa9b.device	loaded	active	plugged	/dev/disk/by-p
dev-sr0.device	loaded	active	plugged	QEMU DVD-ROM
dev-ttyS0.device	loaded	active	plugged	/dev/ttyS0
dev-ttyS1.device	loaded	active	plugged	/dev/ttyS1
dev-ttyS2.device	loaded	active	plugged	/dev/ttyS2
dev-ttyS3.device	loaded	active	plugged	/dev/ttyS3
dev-vda.device	loaded	active	plugged	/dev/vda
dev-vda1.device	loaded	active	plugged	/dev/vda1
sys-devices-pci0000:00-0000:00:02.0-0000:01:00.0-virtio0-net-enp1s0.device	loaded	active	plugged	Virti
sys-devices-pci0000:00-0000:00:02.2-0000:03:00.0-virtio1-block-vda-vda1.device	loaded	active	plugged	/>
sys-devices-pci0000:00-0000:00:02.2-0000:03:00.0-virtio1-block-vda.device	loaded	active	plugged	/sys/d
sys-devices-pci0000:00-0000:00:1b.0-sound-card0.device	loaded	active	plugged	82801I (ICH9 Family)
sys-devices-pci0000:00-0000:00:1f.2-ata3-host2-target2:0:0-0:0:0-block-sr0.device	loaded	active	plugged	82801I (ICH9 Family)
sys-devices-platform-serial18250-tty-ttyS0.device	loaded	active	plugged	/sys/devices/platfo
sys-devices-platform-serial18250-tty-ttyS1.device	loaded	active	plugged	/sys/devices/platfo
sys-devices-platform-serial18250-tty-ttyS2.device	loaded	active	plugged	/sys/devices/platfo
sys-devices-platform-serial18250-tty-ttyS3.device	loaded	active	plugged	/sys/devices/platfo

■ systemctl list-unit-files

- 설치된 모든 unit 파일 조회

```
# systemctl list-unit-files
```

UNIT FILE	STATE
proc-sys-fs-binfmt misc.automount	static
-.mount	generated
dev-hugepages.mount	static
dev-mqueue.mount	static
proc-fs-nfsd.mount	static
proc-sys-fs-binfmt_misc.mount	static
run-vmblock\x2dfuse.mount	disabled
sys-fs-fuse-connections.mount	static
sys-kernel-config.mount	static
sys-kernel-debug.mount	static
tmp.mount	disabled
var-lib-machines.mount	static
var-lib-nfs-rpc_pipefs.mount	static
cups.path	enabled
ostree-finalize-staged.path	disabled
systemd-ask-password-console.path	static
systemd-ask-password-plymouth.path	static
systemd-ask-password-wall.path	static
session-1.scope	transient
session-3.scope	transient
accounts-daemon.service	enabled
alsa-restore.service	static
alsa-state.service	static
anaconda-direct.service	static
anaconda-fips.service	static
anaconda-nm-config.service	static

firewalld 방화벽

■ 개요

- CentOS 6까지는 iptables, CentOS7 부터 firewalld 채택
- firewalld(소프트웨어 방화벽) 서비스가 자동으로 시작되도록 설정되어 있음
- firewall-cmd 명령어로 제어

```
[root@cberi ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2023-01-12 22:31:22 KST; 29min ago
    Docs: man:firewalld(1)
   Main PID: 25328 (firewalld)
     Tasks: 2 (limit: 12392)
    Memory: 27.5M
      CGroup: /system.slice/firewalld.service
              └─25328 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid
```

```
1월 12 22:31:22 cberi.kro.kr systemd[1]: Starting firewalld - dynamic firewall daemon...
1월 12 22:31:22 cberi.kro.kr systemd[1]: Started firewalld - dynamic firewall daemon.
```

```
[root@cberi ~]# firewall-cmd --state
running
[root@cberi ~]#
```

firewalld 방화벽

■ Zone 조회

```
[root@cberi ~]# firewall-cmd --get-zones
block dmz drop external home internal libvirt
nm-shared public trusted work
[root@cberi ~]# firewall-cmd --list-all-zones
block
  target: %%REJECT%
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

public (active)
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
```

■ Zone 기본설정

영역(Zone)	설 명
block	<ul style="list-style-type: none">o 들어오는(inncomming) 모든 패킷 거부(reject)o 나가는(outgoing) 모든 패킷 허용(accept)
dmz	<ul style="list-style-type: none">o SSH를 제외하고 들어오는(inncomming) 모든 패킷 거부(reject)o 나가는(outgoing) 모든 패킷 허용(accept)
drop	<ul style="list-style-type: none">o 들어오는(inncomming) 모든 패킷 폐기(drop)o 나가는(outgoing) 모든 패킷 허용(accept)
external	<ul style="list-style-type: none">o SSH를 제외하고 들어오는(inncomming) 모든 패킷 거부(reject)o 나가는(outgoing) 모든 패킷 허용(accept)o masquerading 활성화
home	<ul style="list-style-type: none">o SSH, dhcpcv6-client, ipp-client, mdns, samba-client를 제외하고 들어오는(inncomming) 모든 패킷 거부(reject)o 나가는(outgoing) 모든 패킷 허용(accept)
internal	<ul style="list-style-type: none">o SSH, dhcpcv6-client, ipp-client, mdns, samba-client를 제외하고 들어오는(inncomming) 모든 패킷 거부(reject)o 나가는(outgoing) 모든 패킷 허용(accept)
public	<ul style="list-style-type: none">o SSH, dhcpcv6-client를 제외하고 들어오는(inncomming) 모든 패킷 거부(reject)o 나가는(outgoing) 모든 패킷 허용(accept)
trusted	<ul style="list-style-type: none">o 들어오는(inncomming) 모든 패킷 허용(accept)o 나가는(outgoing) 모든 패킷 허용(accept)
work	<ul style="list-style-type: none">o SSH, ipp-client, dhcpcv6-client를 제외하고 들어오는(inncomming) 모든 패킷 거부(reject)o 나가는(outgoing) 모든 패킷 허용(accept)

firewalld 방화벽

■ default zone 설정

```
[root@cberi ~]# firewall-cmd --get-default-zone
public
[root@cberi ~]# firewall-cmd --info-zone=public
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp1s0
  sources:
  services: cockpit dhcpcv6-client ssh
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@cberi ~]# firewall-cmd --set-default-zone=trusted
success
[root@cberi ~]# firewall-cmd --set-default-zone=public
success
```

■ 새로운 서비스 정의 (선택사항)

```
# firewall-cmd --get-services | grep xrdp
# firewall-cmd --permanent --new-service=xrdp
success

# cat /etc/firewalld/services/xrdp.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
</service>

# firewall-cmd --permanent --service=xrdp --add-port=3389/tcp
success

# cat /etc/firewalld/services/xrdp.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <port port="3389" protocol="tcp"/>
</service>

# firewall-cmd --reload
```

firewalld 방화벽

■ 서비스 포트 추가 개방

```
# firewall-cmd --permanent --add-service=xrdp  
success  
  
# firewall-cmd --reload  
success  
  
# firewall-cmd --info-zone=public  
public  
  target: default  
  icmp-block-inversion: no  
  interfaces:  
  sources:  
  services: cockpit dhcpcv6-client ssh xrdp  
  ports:  
  protocols:  
  forward: no  
  masquerade: no  
  forward-ports:  
  source-ports:  
  icmp-blocks:  
  rich rules:
```

■ 서비스 포트 삭제

```
# firewall-cmd --permanent --remove-service=xrdp  
# firewall-cmd --permanent --zone=public --remove-port=3128/tcp  
success  
  
# firewall-cmd --reload  
success  
  
# firewall-cmd --info-zone=public  
public  
  target: default  
  icmp-block-inversion: no  
  interfaces:  
  sources:  
  services: cockpit dhcpcv6-client ssh  
  ports:  
  protocols:  
  forward: no  
  masquerade: no  
  forward-ports:  
  source-ports:  
  icmp-blocks:  
  rich rules:
```

firewalld 방화벽

■ 포트 범위 개방

```
# firewall-cmd --permanent --add-port=50001-50010/tcp  
success  
  
# firewall-cmd --reload  
success  
  
# firewall-cmd --info-zone=public  
public  
  target: default  
  icmp-block-inversion: no  
  interfaces:  
  sources:  
  services: cockpit dhcpcv6-client ssh  
  ports: 50001-50010/tcp  
  protocols:  
  forward: no  
  masquerade: no  
  forward-ports:  
  source-ports:  
  icmp-blocks:  
  rich rules:
```

■ 특정 IP에 대한 정책

IP 10.20.30.0/24에서 ftp서비스 접근이 가능하도록

```
# firewall-cmd --permanent --add-rich-rule="rule  
family=ipv4 source address=10.20.30.0/24 service name=ftp  
accept"  
success  
  
# firewall-cmd --reload  
success  
# firewall-cmd --list-all  
public (active)  
  target: default  
  icmp-block-inversion: no  
  interfaces: eth0  
  sources:  
  services: dhcpcv6-client http mssql ssh  
  ports: 1024-2048/tcp  
  protocols:  
  masquerade: no  
  forward-ports:  
  source-ports:  
  icmp-blocks:  
  rich rules:  
    rule family="ipv4" source address="10.20.30.0/24"  
    service name="ftp" accept
```

방화벽



사용 기본적으로 들어오는 요청은 차단됩니다. 외부로 나가는 요청은 차단되지 않습니다.

신규 영역 추가

Public 영역

연결장치 enp1s0 허용된 주소 전체 서브넷

서비스 추가



서비스	TCP	UDP	
ssh	22		⋮
dhcpv6-client		546	⋮
cockpit	9090		⋮
xrdp	원격데스크톱	3389	⋮
http	웹서비스	80	⋮
https	웹서비스(보안)	443	⋮
ftp	proftpd	21	⋮
squid	web proxy	3128	⋮
ipsec	VPN	4500, 500, 4500	⋮

GNOME GUI 설치 및 XRDP 설정

```
[root@cberi ~]# dnf groupinstall "Server with GUI" --skip-broken
```

```
[root@cberi ~]# dnf install xrdp
```

```
[root@cberi ~]# systemctl enable --now xrdp
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/xrdp.service → /usr/lib/systemd/system/xrdp.service.
```

```
[root@cberi ~]# systemctl status xrdp
```

```
● xrdp.service - xrdp daemon
```

```
  Loaded: loaded (/usr/lib/systemd/system/xrdp.service; enabled; vendor preset: disabled)
```

```
  Active: active (running) since Thu 2023-01-12 16:19:25 KST; 24s ago
```

```
    Docs: man:xrdp(8)
```

```
          man:xrdp.ini(5)
```

```
Main PID: 13878 (xrdp)
```

```
  Tasks: 1 (limit: 12392)
```

```
  Memory: 1012.0K
```

```
  CGroup: /system.slice/xrdp.service
```

```
      └─13878 /usr/sbin/xrdp --nodaemon
```

```
[root@cberi ~]# netstat -lntp
```

```
Active Internet connections (only servers)
```

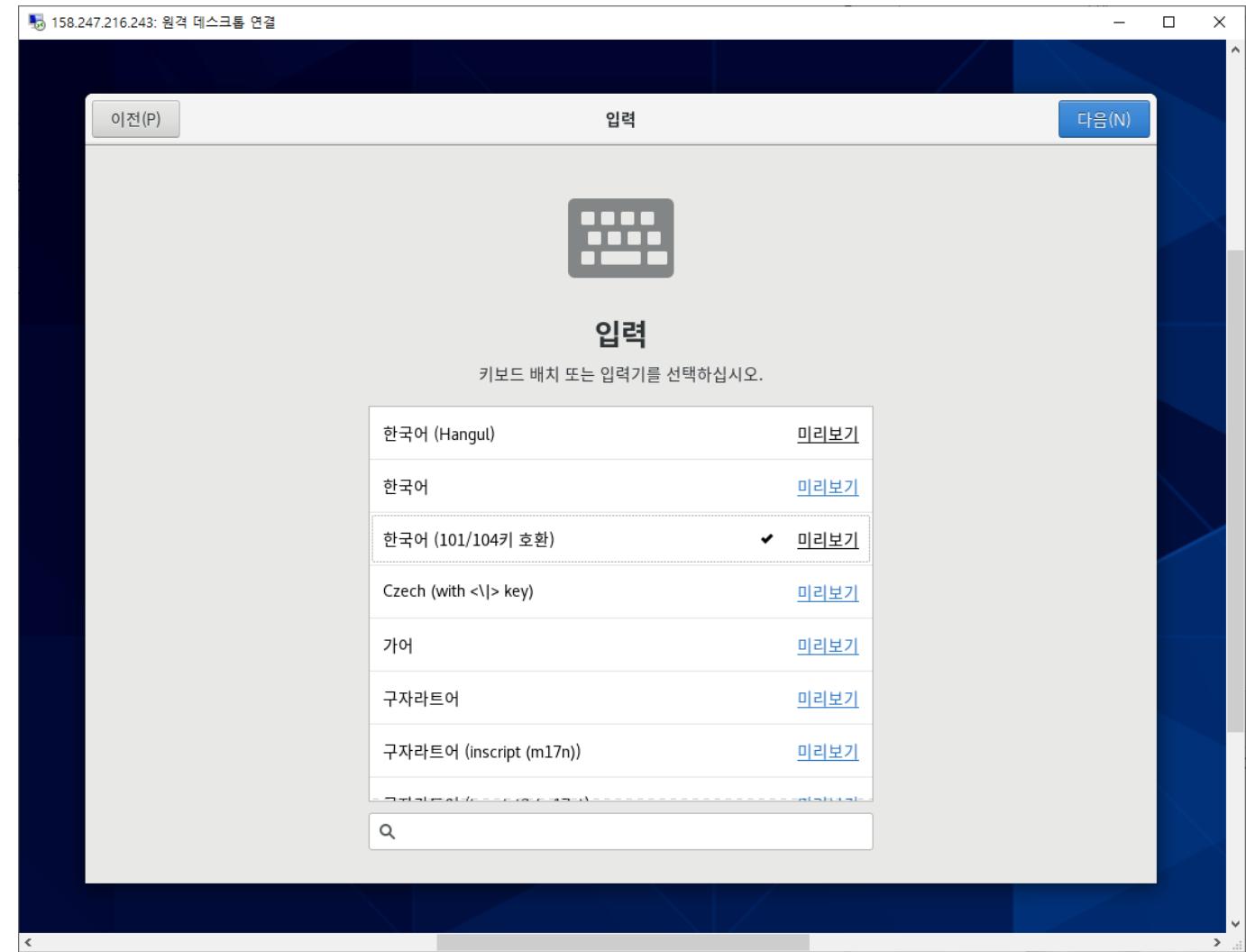
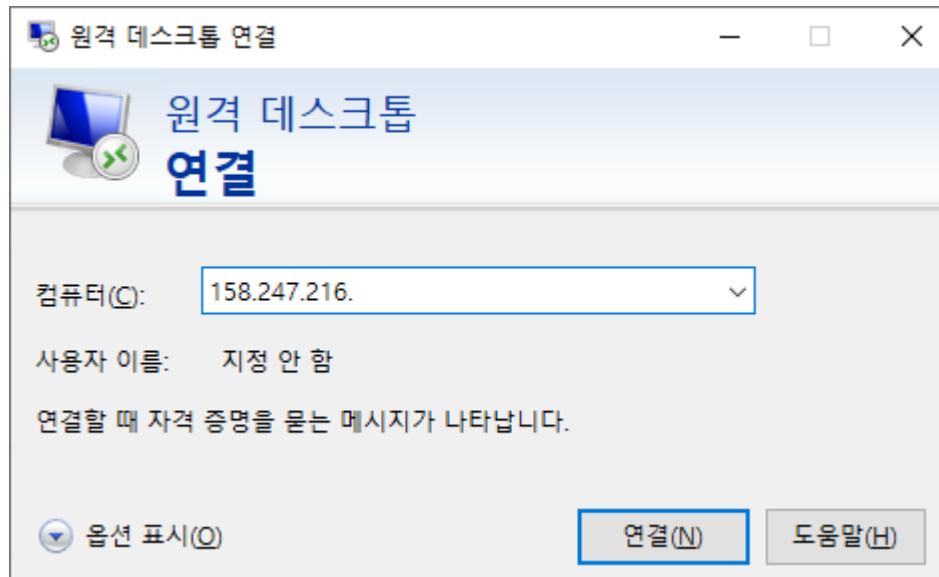
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN	1/systemd
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	1422/sshd
tcp6	0	0	:::9090	:::*	LISTEN	1/systemd
tcp6	0	0	:::111	:::*	LISTEN	1/systemd
tcp6	0	0	:::1:3350	:::*	LISTEN	13877/xrdp-sesman
tcp6	0	0	:::22	:::*	LISTEN	1422/sshd
tcp6	0	0	:::3389	:::*	LISTEN	13878/xrdp

```
[root@cberi ~]# firewall-cmd --permanent --add-service=xrdp  
success
```

```
[root@cberi ~]# firewall-cmd --reload  
success
```

GNOME GUI 접속

■ 윈도우 원격 데스크톱 이용



GNOME GUI 접속

■ wfreerdp 이용

- <https://ci.freerdp.com/job/freerdp-nightly-windows/>

The screenshot shows the GitHub PreBuilds page for FreeRDP. It includes sections for 'Stable' builds (with instructions for flatpak), 'Windows' (with links to cloudbase.it and Jenkins), and 'Mac' (with Homebrew and MacPorts instructions). A GitHub footer at the bottom provides links to Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, and Training.

The screenshot shows the Jenkins dashboard for the 'freerdp-nightly-windows' job. It features a sidebar with links for Dashboard, 상태 (Status), 변경사항 (Changes), 작업공간 (Workspaces), and Git Polling Log. The main area displays the 'Build History' with recent builds (#1234, #1233, #1232, #1231, #1230, #814, #806) and a 'Filter builds...' search bar. To the right, there are sections for 'Disk Usage' (Job, All builds, Locked builds, All workspaces, Slave workspaces, Non-slave workspaces) and performance metrics for MTTR, MTTF, and Standard Deviation over Last 7 Days, Last 30 Days, and All Time periods.

GNOME GUI 접속

■ wfreerdp 이용

- 명령창에서 직접 입력

```
C:\Users\akapo\Downloads>wfreerdp /v:<host ip>
```

예시)

```
C:\Users\akapo\Downloads>wfreerdp /v:45.111.11.1 /size:1200x1400
```

Usage: xfreerdp [file] [options] [/v:<server>[:port]]

Syntax:

```
/flag (enables flag)
	option:<value> (specifies option with value)
+toggle -toggle (enables or disables toggle, where '/' is a synonym of '+')

/v:<server>[:port]      Server hostname
/port:<number>          Server port
/w:<width>               Width
/h:<height>              Height
/size:<width>x<height>   Screen size
/f                        Fullscreen mode
/bpp:<depth>             Session bpp (color depth)
/kbd:<0x<layout id> or <layout name>>   Keyboard layout
/kbd-list                 List keyboard layouts
/kbd-type:<type id>       Keyboard type
/kbd-subtype:<subtype id>  Keyboard subtype
/kbd-fn-key:<function key count>   Keyboard function key count
/admin                    Admin (or console) session
/multimon                 Multi-monitor
/workarea                 Work area
/t:<title>                Window title
+decorations (default:off) Window decorations
/a                        Addin
/vc                       Static virtual channel
/dvc                      Dynamic virtual channel
/u:<domain>\<user> or <user>[@<domain>]   Username
/p:<password>              Password
/d:<domain>                Domain
```

GNOME GUI 접속

- 램 사용량 250MB에서 1.06GB로 치솟음
- GUI 대신 CLI를 사용하는 이유이다.

```
CPU[||| 3.2%] Tasks: 115, 227 thr, 84 kthr; 1 running
Mem[||||| 1.06G/1.93G] Load average: 2.99 1.25 0.60
Swp[||| 231M/5.08G] Uptime: 00:53:30

Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
14975 root 20 0 225M 3880 2792 R 2.6 0.2 0:01.97 htop
1233 root 20 0 471M 22052 11260 S 0.6 1.1 0:11.40 /usr/libexec/platform-python -Es /usr/sbin/tuned -l -P
1402 root 20 0 471M 22052 11260 S 0.6 1.1 0:03.88 /usr/libexec/platform-python -Es /usr/sbin/tuned -l -P
1624 root 20 0 463M 20688 8212 S 0.6 1.0 0:07.07 cockpit-bridge
 1 root 20 0 237M 12548 7156 S 0.0 0.6 0:05.27 /usr/lib/systemd/systemd --switched-root --system --deserialize 17
633 root 20 0 89820 10616 9288 S 0.0 0.5 0:01.60 /usr/lib/systemd/systemd-journald
672 root 20 0 99492 8092 5568 S 0.0 0.4 0:00.29 /usr/lib/systemd/systemd-udevd
750 root 16 -4 127M 1868 1728 S 0.0 0.1 0:00.58 /sbin/auditd
751 root 16 -4 127M 1868 1728 S 0.0 0.1 0:00.01 /sbin/auditd
752 root 16 -4 48648 3092 2732 S 0.0 0.2 0:00.17 /usr/sbin/sedispatch
753 root 16 -4 127M 1868 1728 S 0.0 0.1 0:00.09 /sbin/auditd
788 root 20 0 83400 7784 6036 S 0.0 0.4 0:00.19 /usr/lib/systemd/systemd-logind
789 root 20 0 50292 4664 3664 S 0.0 0.2 0:00.02 /usr/sbin/smard -n -q never
790 dbus 20 0 67260 6500 4184 S 0.0 0.3 0:05.56 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile -
791 libstorage 20 0 8616 1816 1668 S 0.0 0.1 0:00.02 /usr/bin/lsmd -d
793 root 20 0 6668 1884 1736 S 0.0 0.1 0:00.00 /usr/sbin/mcelog --ignorenodev --daemon --foreground
795 polkitd 20 0 1585M 16668 11980 S 0.0 0.8 0:00.65 /usr/lib/polkit-1/polkitd --no-debug
798 root 20 0 8144 1672 1424 S 0.0 0.1 0:00.00 /usr/bin/less /var/log/cloud-init.log
799 chrony 20 0 136M 4376 3844 S 0.0 0.2 0:00.06 /usr/sbin/chronyd
803 dbus 20 0 67260 6500 4184 S 0.0 0.3 0:00.00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile -
809 polkitd 20 0 1585M 16668 11980 S 0.0 0.8 0:00.00 /usr/lib/polkit-1/polkitd --no-debug
810 polkitd 20 0 1585M 16668 11980 S 0.0 0.8 0:00.21 /usr/lib/polkit-1/polkitd --no-debug
811 polkitd 20 0 1585M 16668 11980 S 0.0 0.8 0:00.00 /usr/lib/polkit-1/polkitd --no-debug
812 polkitd 20 0 1585M 16668 11980 S 0.0 0.8 0:00.00 /usr/lib/polkit-1/polkitd --no-debug
821 polkitd 20 0 1585M 16668 11980 S 0.0 0.8 0:00.00 /usr/lib/polkit-1/polkitd --no-debug
823 root 20 0 440M 33472 12968 S 0.0 1.7 0:02.45 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid

F1 Help F2 Setup F3 Search F4 Filter F5 Tree F6 SortBy F7 Nice -F8 Nice +F9 Kill F10 Quit
```

2. Domain Name System

- 1) vim 에디터
- 2) DNS 소개
- 3) CND 및 Cloudflare
- 4) proftpd 서버
- 5) Squid 프록시 서버



도메인 생성

The screenshot shows the homepage of the Namecheap website. At the top, there is a navigation bar with the Namecheap logo on the left and links for Domains, Hosting, WordPress, Email, Marketing Tools, Security, Transfer to Us, Help Center, and Account. A message about supporting Ukraine is displayed, followed by a link to a graphic design tool. The main banner features a purple winter theme with snowflakes and a cartoon character at the bottom. It promotes registering domains for less, specifically mentioning .COM for \$5.98 and .NET for \$11.98.

namecheap

Domains NEW Hosting WordPress Email NEW Marketing Tools NEW Security NEW Transfer to Us TRY ME Help Center Account

We stand with our friends and colleagues in Ukraine. To support Ukraine in their time of need visit this [page](#).

Create eye-catching graphics with the web's favorite online graphic design tool →

Make more online, for less

Register Transfer

Register a domain name to start

.COM only \$5.98* .NET only \$11.98

도메인 관리를 위한 로그인 정보

그룹	순	소속교	성명	VPS IP	도메인	id / pw
1	1	남성중	김민건	45.32.255.157	minocle.site	[namecheap] cberidom1 / pororo23@dom1 [google] domain1@cberi.go.kr / pororo23@dom1
	2	일신여고	방윤정	167.179.108.128	yoon.lol	
	3	청주대성고	오미현	167.179.108.93	schmh.site	
	4	청주대성고	방민영	45.76.204.169	alsdud.site	
	5	IT과학고	김민재	149.28.26.135	choi0217.site	
2	1	흥덕고	이주원	198.13.54.39	juwon.site	[namecheap] cberidom2 / pororo23@dom2 [google] domain2@cberi.go.kr / pororo23@dom2
	2	흥덕고	남하준	45.32.26.235	hajun.site	
	3	흥덕고	김지겸	108.61.183.138	wisck.site	
	4	흥덕고	정호	66.42.40.63	rinlv.site	
	5	일신여고	이정민	149.28.22.90	curri.xyz	
3	1	정보원	박정진	108.61.180.91	cberi.site	

- 도메인 관리: <https://www.namecheap.com>
- 이메일 접속: <https://mail.google.com>

hostname (기기명, 서버명) 변경

■ 동적변경

- 재부팅시 원래대로 돌아감

```
[root@vultr ~]# hostname  
vultr.guest  
[root@vultr ~]# hostname cberi.shop  
[root@vultr ~]# hostname  
cberi.shop
```

■ 정적변경

```
[root@vultr ~]# hostnamectl set-hostname cberi.shop  
[root@vultr ~]#  
[root@vultr ~]# hostnamectl status  
Static hostname: cberi.shop  
          Icon name: computer-vm  
          Chassis: vm  
        Machine ID: 7000c79093de42d79c79f6cbea10bf48  
          Boot ID: ca5cd14663ba4371b3a6527bf04d7956  
Virtualization: microsoft  
Operating System: CentOS Stream 8  
      CPE OS Name: cpe:/o:centos:centos:8  
        Kernel: Linux 4.18.0-526.el8.x86_64  
Architecture: x86-64  
[root@vultr ~]#
```

hostname (기기명, 서버명) 변경

■ 실습

```
[root@vultr ~]# hostnamectl -h
hostnamectl [OPTIONS...] COMMAND ...

Query or change system hostname.

-h --help           Show this help
--version          Show package version
--no-ask-password Do not prompt for password
-H --host=[USER@]HOST Operate on remote host
-M --machine=CONTAINER Operate on local container
--transient        Only set transient hostname
--static          Only set static hostname
--pretty          Only set pretty hostname

Commands:
status            Show current hostname settings
set-hostname NAME Set system hostname
set-icon-name NAME Set icon name for host
set-chassis NAME Set chassis type for host
set-deployment NAME Set deployment environment for host
set-location NAME Set location for host
[root@vultr ~]#
```

```
[root@vultr ~]# hostnamectl status
Static hostname: vultr.guest
Transient hostname: cberi.kro.kr
Icon name: computer-vm
Chassis: vm
Machine ID: 9b0b9092c72541aa8dd1a5db8e4a603e
Boot ID: 2753caed0dca4c8498a3f138b31b8dbf
Virtualization: microsoft
Operating System: CentOS Stream 8
CPE OS Name: cpe:/o:centos:centos:8
Kernel: Linux 4.18.0-408.el8.x86_64
Architecture: x86-64
[root@vultr ~]# hostnamectl set-hostname cberi.shop
[root@vultr ~]# hostnamectl status
Static hostname: cberi.shop
Icon name: computer-vm
Chassis: vm
Machine ID: 9b0b9092c72541aa8dd1a5db8e4a603e
Boot ID: 2753caed0dca4c8498a3f138b31b8dbf
Virtualization: microsoft
Operating System: CentOS Stream 8
CPE OS Name: cpe:/o:centos:centos:8
Kernel: Linux 4.18.0-408.el8.x86_64
Architecture: x86-64
[root@vultr ~]#
```

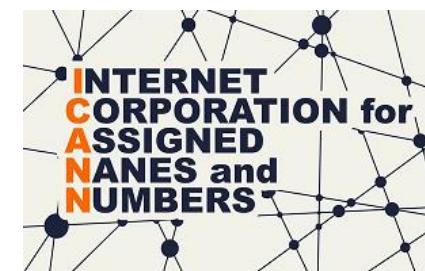
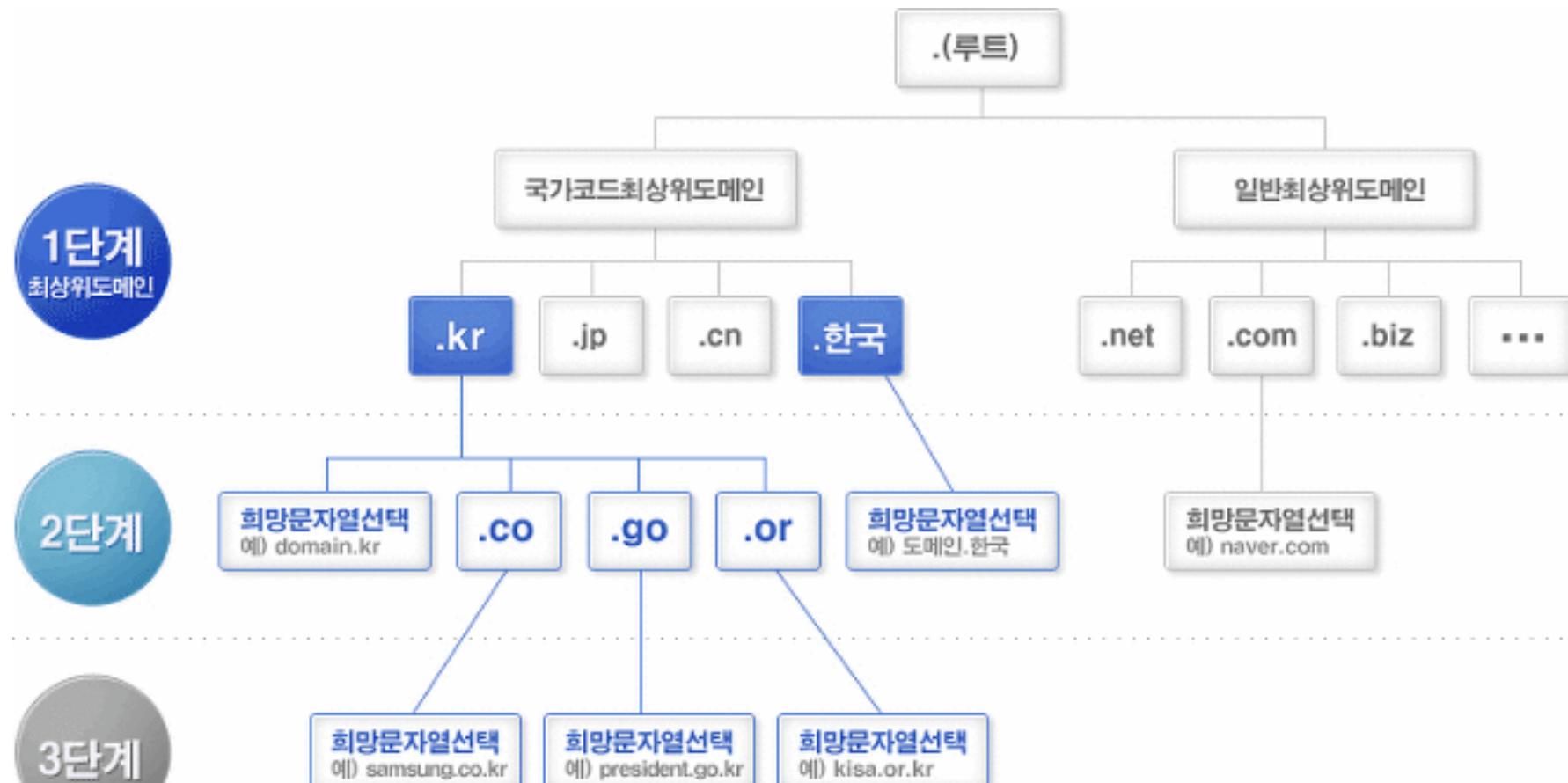
DNS (Domain Name System)

■ 도메인 네임 구조

1단계
최상위도메인

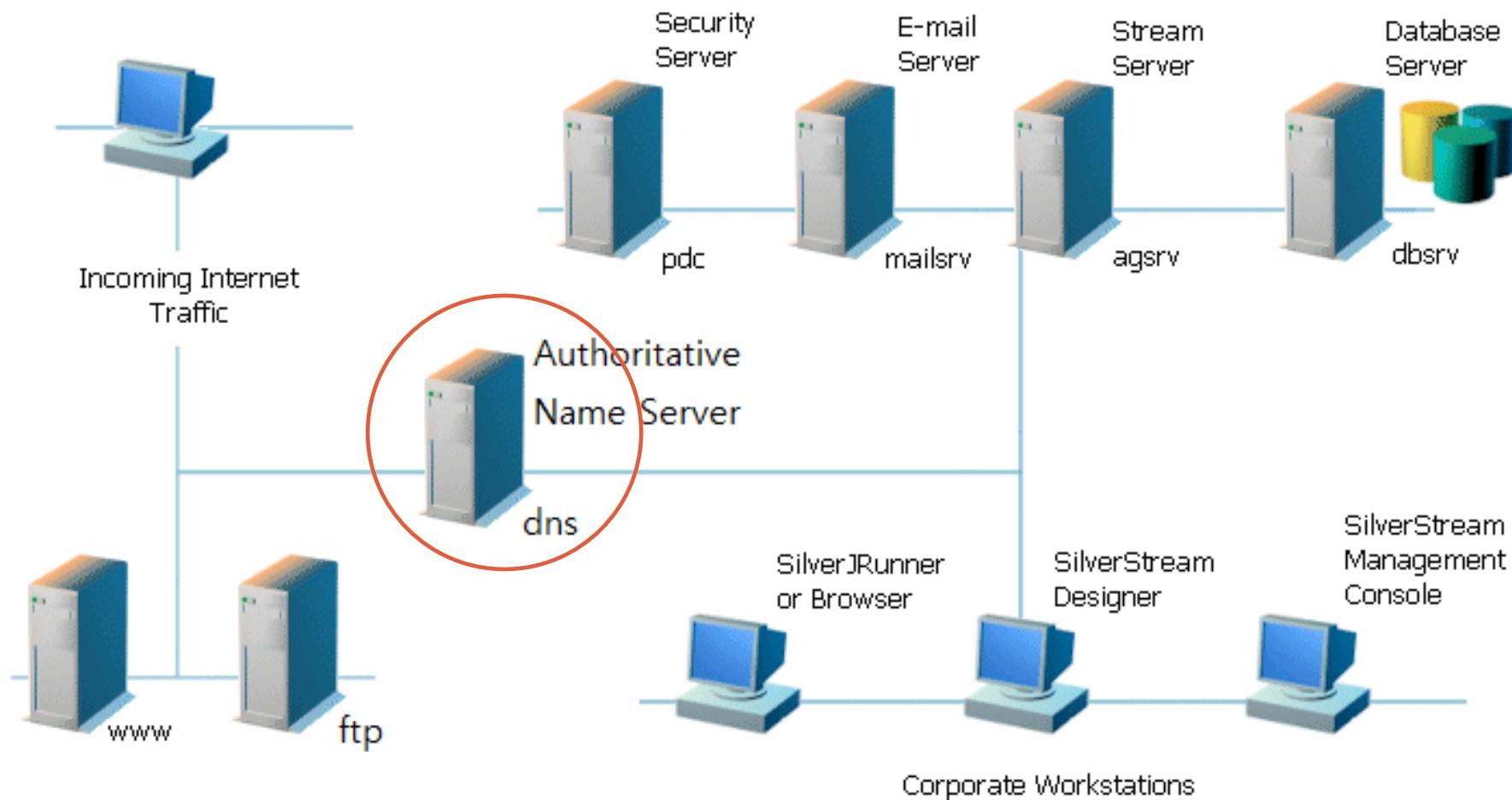
2단계

3단계



DNS (Domain Name System)

■ Authoritative Name Server 예시

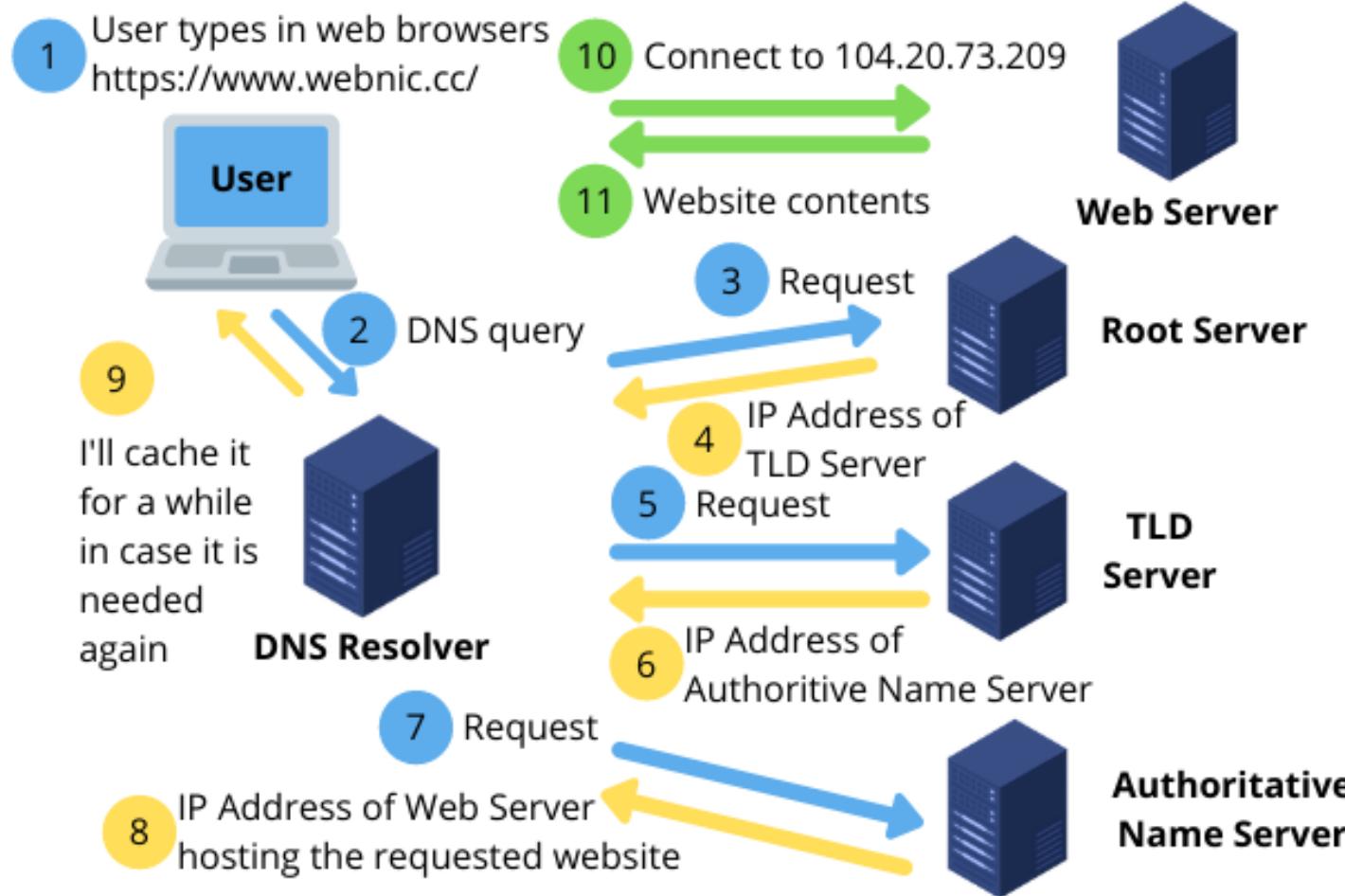


■ DNS record

name	ip
www	12.53.9.11
ftp	12.53.9.13
ns	12.53.9.14
pdc	12.53.9.15
mail	12.53.9.16
db	12.53.9.17
stream	12.53.9.18
shop	12.53.10.1

DNS (Domain Name System)

- host에 접속하기 위해 컴퓨터 내부에서 벌어지는 일



- Root Server, Top Level Domain(TLD) Server : com, org, net 등과 같은 상위 레벨 도메인과 모든 국가의 상위 레벨 도메인에 대한 관리를 담당하는 DNS 서버.
- Authoritative DNS Server : DNS 레코드 정보를 저장, 관리한다. 일반적으로 DNS 호스팅 공급자 또는 도메인 등록기관 / DDNS 서비스 등록 기관등의 DNS 서버와 회사, 기관, 개인이 일반적으로 구축하는 DNS 서버가 여기에 속한다.
- DNS Resolver(Cache DNS Server) : 일반적으로 우리가 사용하는 통신사 등의 서버는 IP 매핑을 위한 도메인 정보도 갖지 않고, 도메인 호스트의 질의에 대해 DNS Cache나 다른 네임 서버가 가진 데이터로 응답만을 제공하는 서버로 로컬 DNS 서버 또는 퍼블릭 도메인 네임 서버로 불리우며 엄격한 계층 구조에 포함되지 않는 서버이다. 실제로 authoritative DNS 서버보다 많이 통신하는 서버이다.

DNS (Domain Name System)

- DNS 레코드 종류
 - A : IPv4 형식의 호스트 주소
 - AAAA : IPv6 형식의 호스트 주소
 - CNAME (canonical name) : 다른 이름의 별칭. 하나의 도메인이나 하위 도메인을 다른 도메인으로 전달하며, IP 주소를 제공하지 않음.
 - MX (mail exchange) : 메일 교환(MX) 레코드는 도메인 이메일을 도메인의 사용자 계정을 호스팅하는 서버로 전송
 - NS (name server) : 도메인에 대해 어떤 서버가 DNS 정보를 전달할지 결정. 일반적으로 도메인에 기본 및 보조 네임서버 레코드가 존재.]
 - TXT : 다양한 목적으로 사용될 수 있는 텍스트 정보를 제공
 - SOA (Start of Authority) : 해당 DNS 서버 자체의 설정 정보를 정의. 도메인에 대한 관리자 정보를 저장.
- ※ TTL(time to live) : 다음 레코드 변경사항이 적용될 때까지 걸리는 시간(초)을 결정하는 값.

DNS RECORDS CHEAT SHEET - CONSTELLIX

A (address)

1

A (address) - Most commonly used to map a fully qualified domain name (FQDN) to an IPv4 address and acts as a translator by converting domain names to IP addresses.



AAAA (quad A)

2

AAAA (quad A) - Similar to A Records but maps to an IPv6 address (smartphones prefer IPv6, if available).



ANAME

3

ANAME - This record type allows you to point the root of your domain to a hostname or FQDN.



CNAME

4

CNAME (Canonical Name) - An alias that points to another domain or subdomain, but never an IP address. Alias record mapping FQDN to FQDN, multiple hosts to a single location. This record is also good for when you want to change an IP address over time as it allows you to make changes without affecting user bookmarks, etc.



SOA (start of authority)

5

SOA (Start of Authority) - Stores information about domains and is used to direct how a DNS zone propagates to secondary name servers.



NS (name server)

6

NS (name server) - Specifies which name servers are authoritative for a domain or subdomains (these records should not be pointed to a CNAME).



MX (mail exchange)

7

MX (Mail eXchange) - Uses mail servers to map where to deliver email for a domain (should point to a mail server name and not to an IP address).



TXT (text)

8

TXT (text) - Allows administrators to add limited human and machine-readable notes and can be used for things such as email validation, site, and ownership verification, framework policies, etc., doesn't require specific formatting.



SRV (service)

9

SRV (service) - Allows services such as instant messaging or VoIP to be directed to a separate host and port location.



SPF (sender policy framework)

10

SPF (sender policy framework) - Helps prevent email spoofing and limits spammers.



PTR (pointer)

11

PTR (pointer) - A reverse of A and AAAA records, which maps IP addresses to domain names. These records require domain authority and can't exist in the same zone as other DNS record types (put in reverse zones).



QUICK TIP

12

Tip: Always check for typos and mistakes when entering your DNS record information, especially your IPs. The Zone Config File is a good place to check your work and spot any mistyped information.



DNS (Domain Name System)

■ DNS 레코드 예시

Type ▲	Name	Content	TTL
CNAME	www	cberi.art	Auto
A	*	158.247.216.243	1 min
A	cberi.art	158.247.216.243	Auto
A	docker	158.247.216.243	Auto
A	sftp	158.247.216.243	Auto
A	sqladmin	158.247.216.243	Auto
A	ssh	158.247.216.243	Auto
A	vpn	158.247.216.243	Auto
MX	*	mx.cberi.art	Auto
MX	cberi.art	mx.cberi.art	Auto
TXT	cberi.art	v=spf1 ip4:66.96.128.0/18 ?	Auto

■ Name Server

Type	Value
NS	jacob.ns.cloudflare.com
NS	tia.ns.cloudflare.com

Domain

ap.www.namecheap.com/Domains/DomainControlPanel/cberi.shop/domain/

Dashboard Expiring / Expired Domain List Hosting List Private Email SSL Certificates Apps Profile

cberi.shop

Domain Products Sharing & Transfer Advanced DNS

STATUS & VALIDITY ALERT Dec 28, 2023 - Dec 28, 2024 AUTO-RENEW RENEW

WithheldforPrivacy PROTECTION Dec 28, 2023 - Dec 28, 2024 AUTO-RENEW ADD YEARS SHOW DETAILS

PremiumDNS ? Enable PremiumDNS protection in order to switch your domain to our PremiumDNS platform. With our PremiumDNS platform, you get 100% DNS uptime and DDoS protection at the DNS level. BUY NOW

NAMESERVENS ? Namecheap BasicDNS Authoritative Name Server ✓ ✕

REDIRECT DOMAIN ? Source URL Destination URL

cberi.shop → http://www.cberi.shop/

ADD REDIRECT ADD WILDCARD REDIRECT

This screenshot shows the Namecheap Domain Control Panel for the domain cberi.shop. The left sidebar includes links for Dashboard, Expiring / Expired, Domain List (selected), Hosting List, Private Email, SSL Certificates, Apps, and Profile. The main content area displays the domain's status as valid until Dec 28, 2024, with auto-renew enabled. It also shows privacy protection is withheld for privacy. A PremiumDNS section offers protection for \$12.99. The NameServers section lists Namecheap BasicDNS as the authoritative name server. Below, a redirect configuration shows a rule from cberi.shop to http://www.cberi.shop. Buttons for adding new redirects or wildcard redirects are at the bottom.

Advanced DNS

ap.www.namecheap.com/Domains/DomainControlPanel/cberi.shop/advanceddns

cberi.shop

Dashboard Expiring / Expired Domain List Hosting List Private Email SSL Certificates Apps Profile

DNS TEMPLATES Choose DNS Template

HOST RECORDS Actions Filters Search

Type	Host	Value	TTL
CNAME Record	www	parkingpage.namecheap.com.	30 min
URL Redirect Record	@	http://www.cberi.shop/	Unmasked
A Record	ssh	207.148.95.144	Automatic
A Record	sftp	207.148.95.144	Automatic

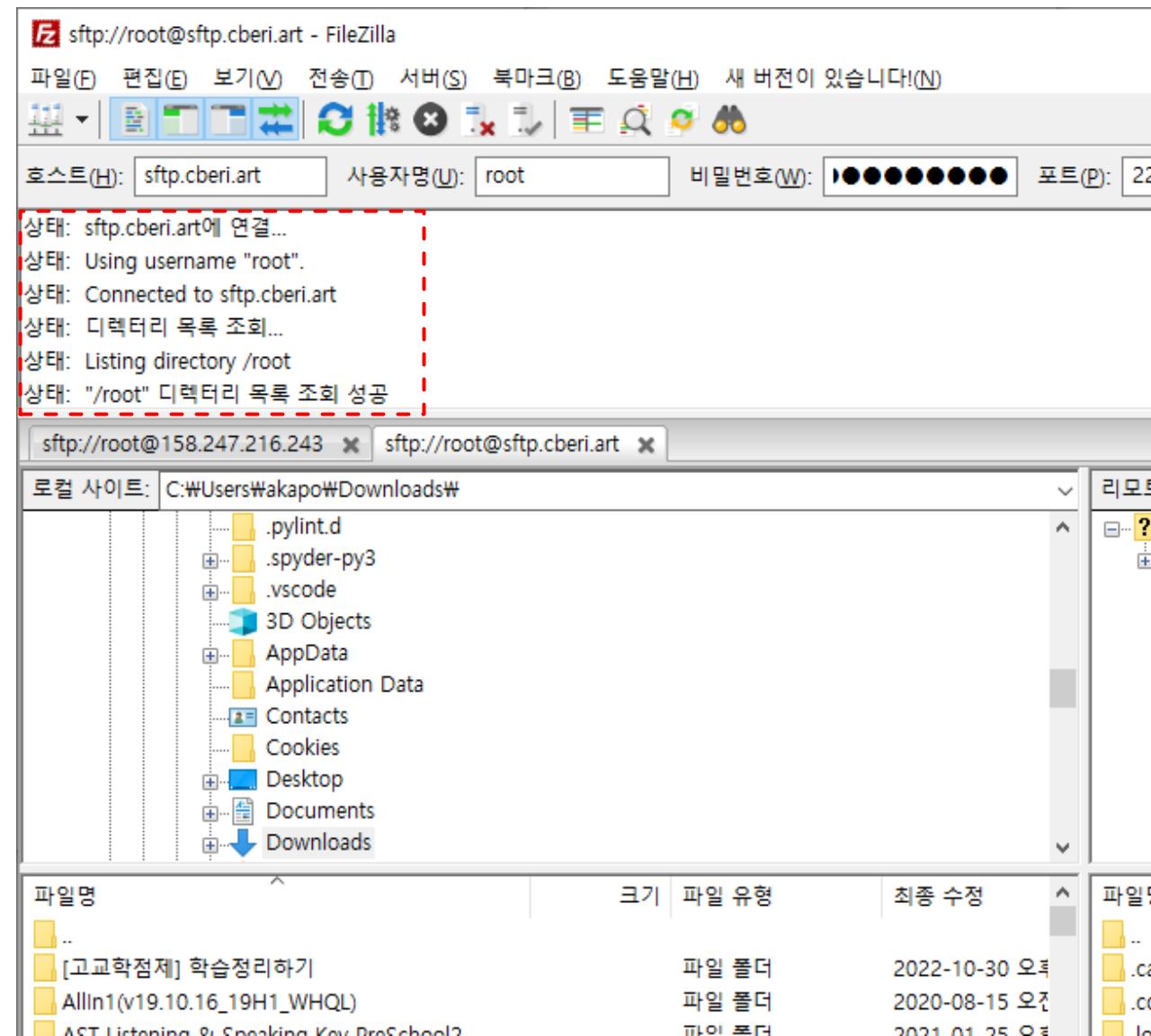
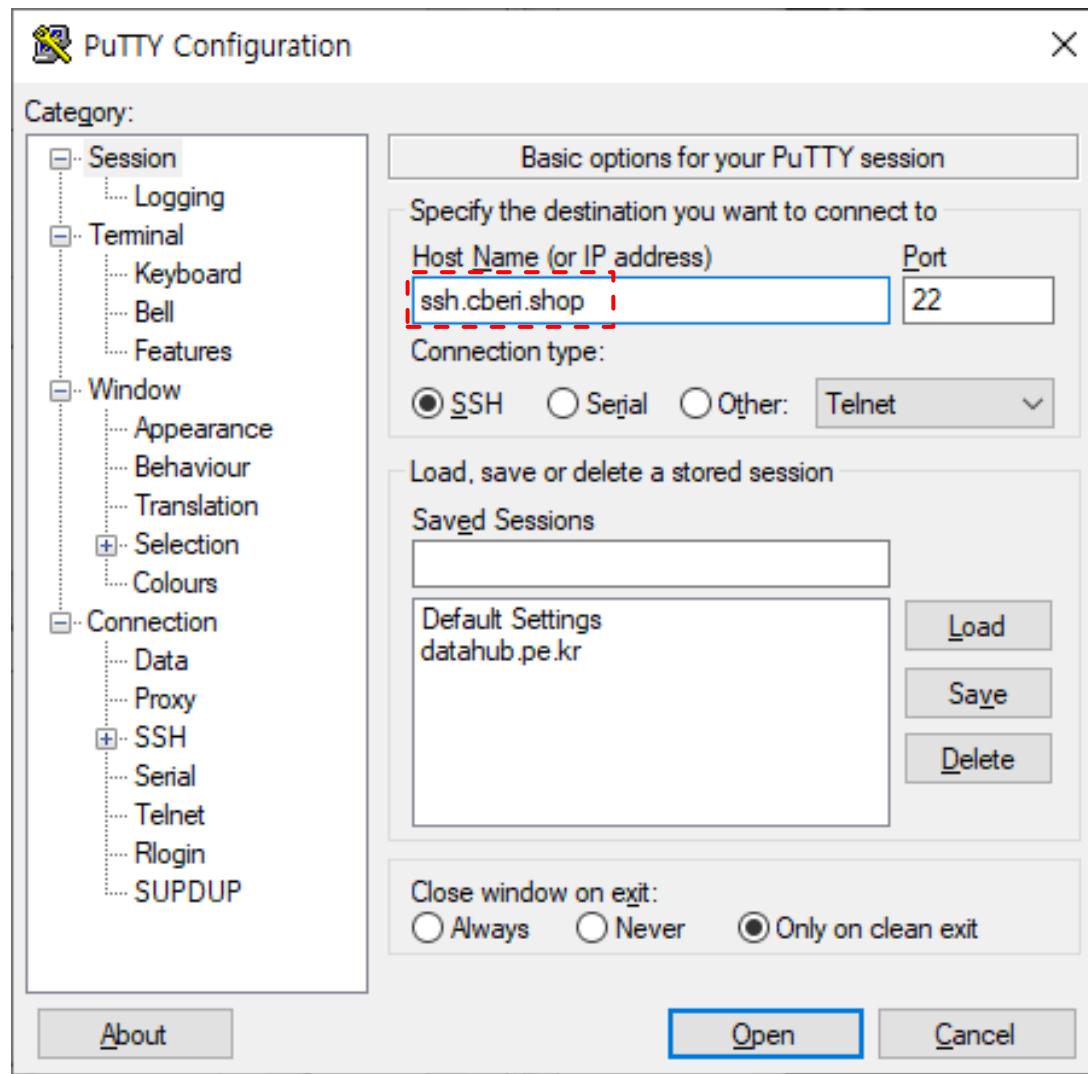
A레코드 ssh 추가
A Record ssh 207.148.95.144 Automatic ✓ ✗

A레코드 admin 추가
A Record sftp 207.148.95.144 Automatic ✓ ✗

ADD NEW RECORD SAVE ALL CHANGES

The screenshot shows the 'Advanced DNS' section of the Namecheap Domain Control Panel for the domain 'cberi.shop'. The left sidebar includes links for Dashboard, Expiring / Expired, Domain List (selected), Hosting List, Private Email, SSL Certificates, Apps, and Profile. The main area displays DNS records: a CNAME record for 'www' pointing to 'parkingpage.namecheap.com.' with a TTL of 30 min; and a URL Redirect Record for '@' pointing to 'http://www.cberi.shop/' with an Unmasked TTL. Two new A records are being added: one for 'ssh' with IP '207.148.95.144' and another for 'sftp' with the same IP, both set to automatic TTL. The 'ssh' record is highlighted with a red dashed box and labeled 'A레코드 ssh 추가'. The 'sftp' record is also highlighted with a red dashed box and labeled 'A레코드 admin 추가'. At the bottom, there are buttons for 'ADD NEW RECORD' and 'SAVE ALL CHANGES'.

새롭게 추가한 이름으로 접속 시도





Zero Trust 서비스

최고 성능의 제로 트러스트 응용 프로그램 액세스 및 인터넷 브라우징 솔루션으로 데이터 손실, 맬웨어, 피싱을 차단하세요.

- ✓ Zero Trust 네트워크 액세스(ZTNA)
- ✓ 보안 웹 게이트웨이(SWG)
- ✓ 브라우저 격리 및 이메일 보안
- ✓ 클라우드 액세스 보안 브로커(CASB)

[요금제 및 가격 보기 >](#)

[자세한 정보](#)



웹 사이트 및 앱 성능

Cloudflare 전역 네트워크를 통해 웹 사이트, 앱, API 속도를 개선해 콘텐츠를 최적화하고 사용자 위치와 더욱 가깝게 전달하세요.

- ✓ CDN
- ✓ DNS
- ✓ Load Balancing
- ✓ 웹 사이트 및 애플리케이션 가속화

[요금제 및 가격 보기 >](#)

[자세한 정보](#)



웹 사이트 및 앱 보안

봇, DDoS 공격 등으로부터 웹 사이트 및 애플리케이션을 보호합니다. 의심스러운 활동과 잠재적인 공격까지 함께 모니터링합니다.

- ✓ DDoS 방어
- ✓ 봇 관리(Bot Management)
- ✓ WAF

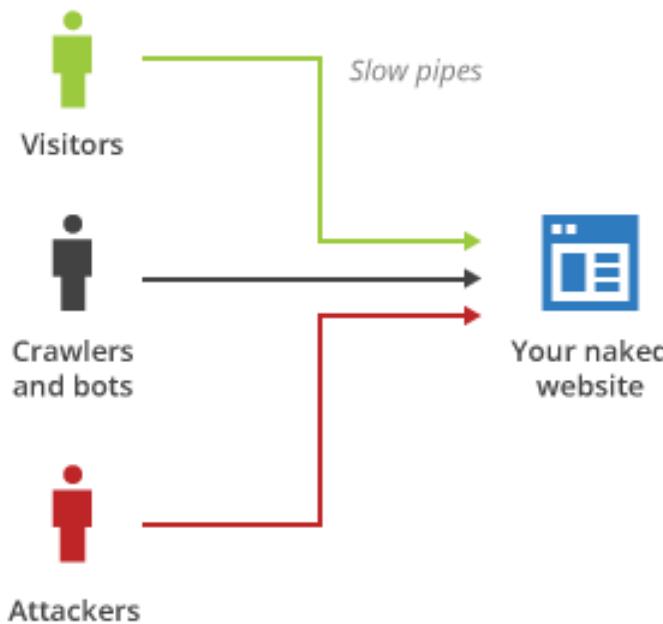
[요금제 및 가격 보기 >](#)

[자세한 정보](#)

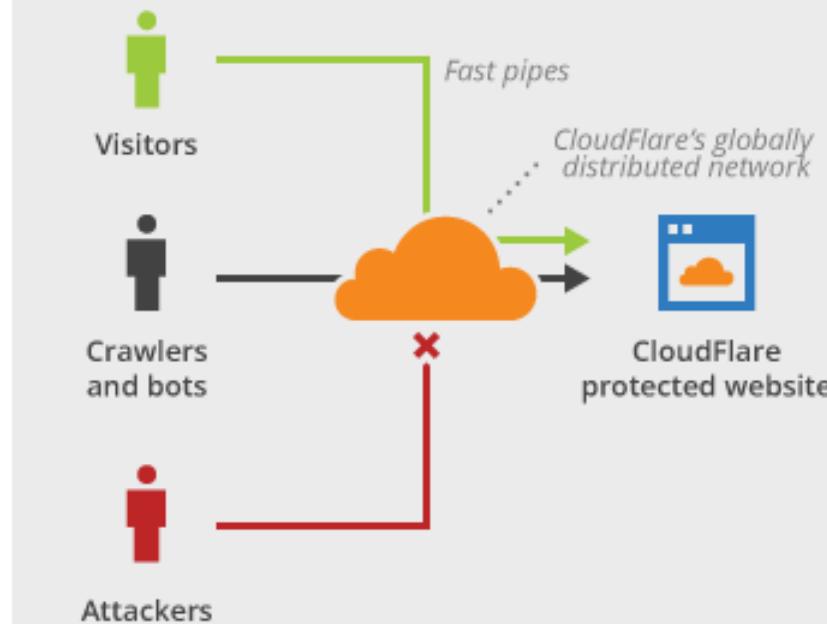
Cloudflare

■ Cloudflare 작동 원리 개념도

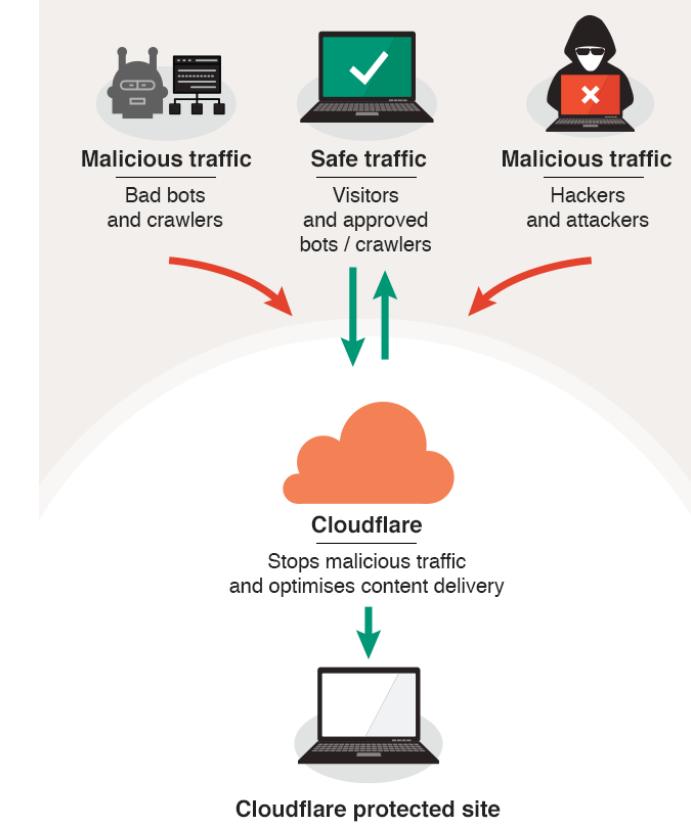
Without CloudFlare



With CloudFlare

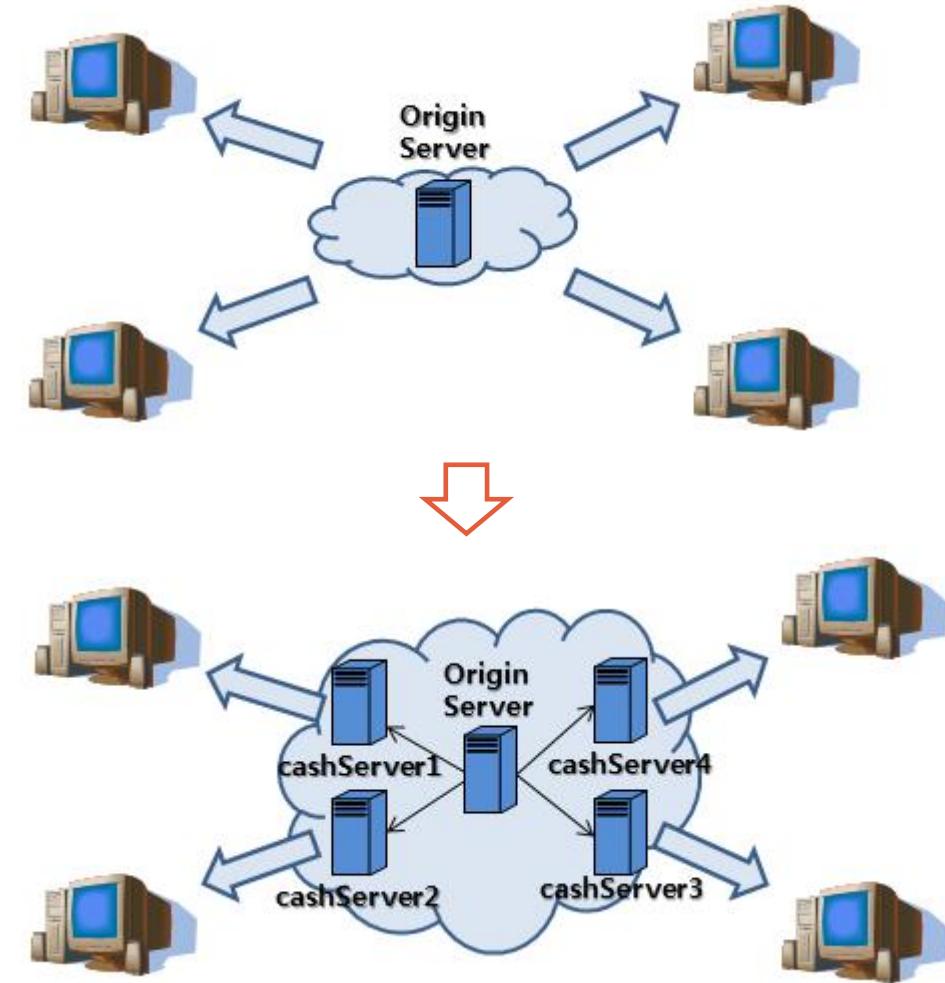


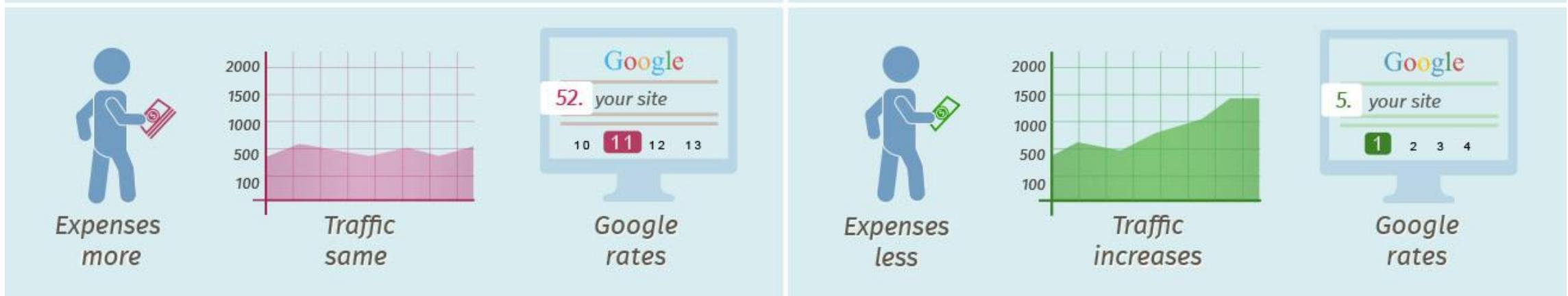
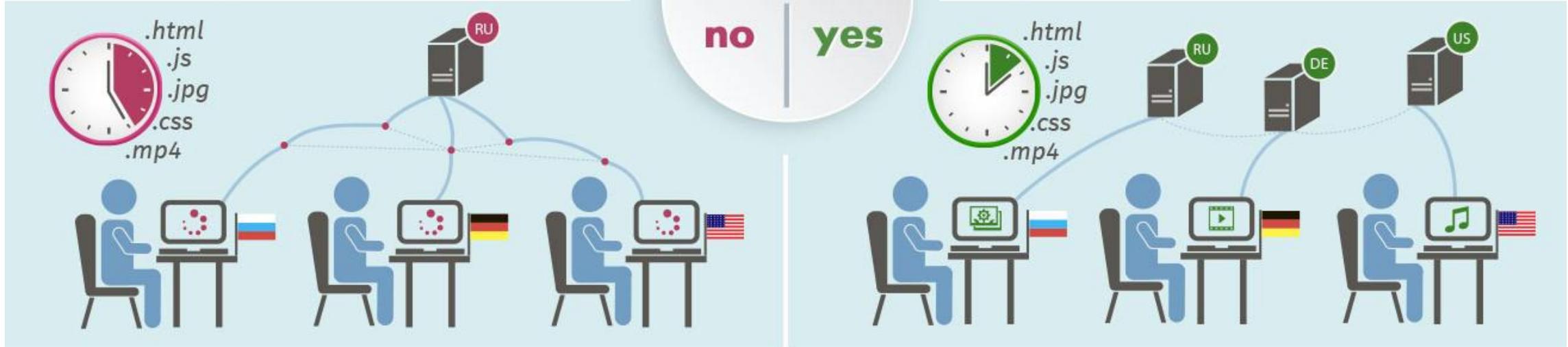
How Cloudflare works



Cloudflare 제공기능

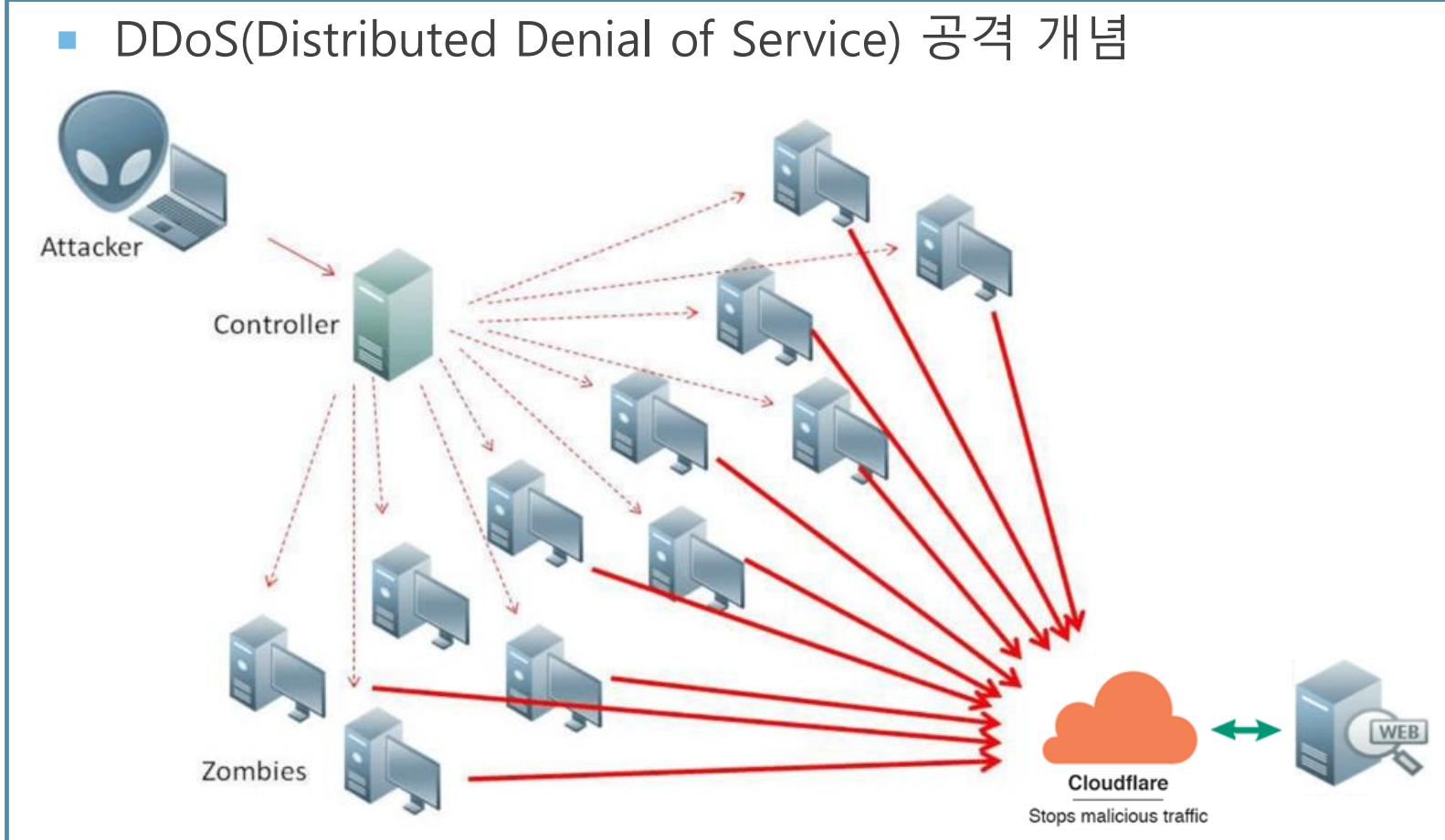
- CDN(Contents Delivery Network)
 - 지리, 물리적으로 떨어져 있는 사용자에게 컨텐츠를 더 빠르게 제공할 수 있는 기술
 - 느린 응답속도 / 다운로딩 타임을 극복하기 위한 기술
 - 효과
 - 트래픽 분산
 - 웹사이트 로딩 속도 개선
 - 인터넷 회선 비용 절감
 - 컨텐츠 제공의 안정성
 - 웹사이트 보안 개선





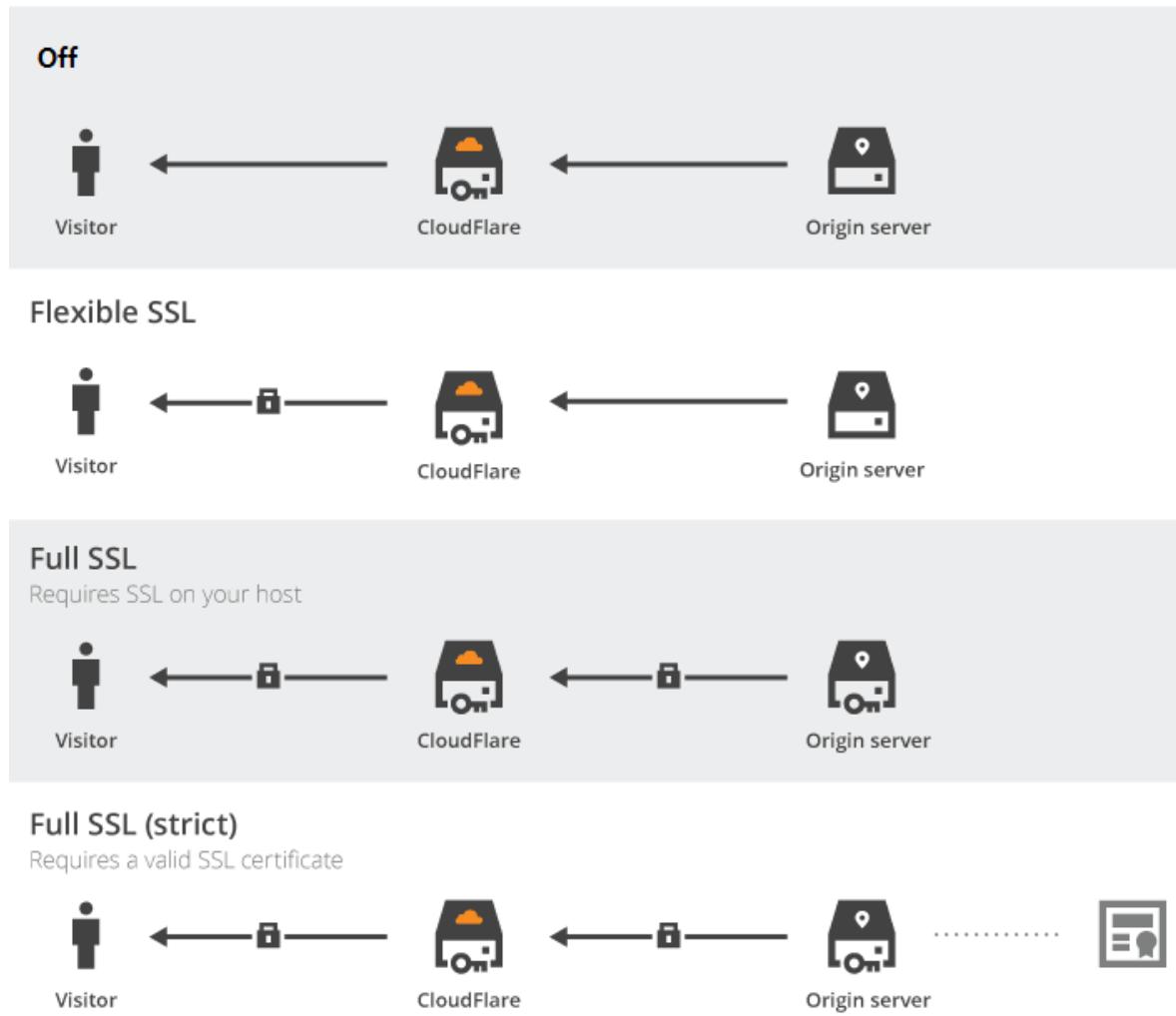
Cloudflare 제공 기능

- DDoS 방어
 - 서버를 숨기고 트래픽을 대신 받음
 - 전제조건: 보호하고자 하는 호스트의 IP가 유출되어서는 안됨.



Cloudflare 제공기능

- 암호화
 - SSL 인증서 제공
 - https 통신
 - SSL 인증서를 설치하지 않아 https 통신이 불가능한 웹서버에도 https 연결 기능 제공



Cloudflare 계정 생성 (맘아이 off 필요, 동일 IP에서 6개 이상 계정생성 안됨)

The screenshot shows the Cloudflare account creation interface. At the top right, there are links for 'Support' and 'English (US)'. The main heading is 'Get started with Cloudflare'. Below it, there are fields for 'Email' (containing 'dianox@cberi.go.kr') and 'Password' (containing a masked password). A 'Show' link is next to the password field. Below the password field, a message says 'Password requirements met!' followed by four green checkmarks with corresponding text: '8 characters', '1 number', '1 special character e.g., \$, !, @, %, &', and 'No leading or trailing whitespace'. At the bottom, a note states 'By clicking Create Account, I agree to Cloudflare's [terms](#), [privacy policy](#), and [cookie policy](#)'. A large blue button at the bottom is labeled 'Create Account'.

Support ▾ English (US) ▾

Get started with Cloudflare

Email

dianox@cberi.go.kr

Password

Show

.....

Password requirements met!

- ✓ 8 characters
- ✓ 1 number
- ✓ 1 special character e.g., \$, !, @, %, &
- ✓ No leading or trailing whitespace

By clicking Create Account, I agree to Cloudflare's [terms](#), [privacy policy](#), and [cookie policy](#).

Create Account

dash.cloudflare.com/ef0997f2690d7176b7afde0532df2fad/add-site

CLOUDFLARE

Dianox@cberi.go.kr...

- 웹 사이트
- Discover
- 도메인 등록
- 분석 및 로그
- Security Center
- 추적 베타
- Turnstile
- Zero Trust
- Area 1
- Workers 및 Pages
- AI 베타
- R2
- 사이드바 축소

Cloudflare에 웹 사이트 또는 응용 프로그램 추가

방문자를 위해 보안, 성능 및 안정성을 최적화하고 모니터링.

[가이드 학습 경로 따라가기](#)

도메인 입력
Cloudflare 구성을 적용하고 웹 사이트 또는 응용 프로그램에 대한 영향을 모니터링하는 공간의 이름입니다.

도메인 이름 입력(example.com)

계속

또는 Cloudflare Registrar로 [새 도메인을 등록](#)하십시오.

연락처	하는 일	리소스	지원	Cloudflare 소개
지원팀에 문의	요금제	설명서	기술 자료	Cloudflare 팀
영업팀에 문의	개요	블로그	커뮤니티	채용
영업 상담 전화번호: +1 (888) 993 5273	기능	사례 연구	시스템 상태	보도
	네트워크	파트너	신뢰 및 안전	웹 사이트 사용 약관
	Aops	고객		세프 서비스 계약

Pro

가장 인기 있는 항목

US\$20¹ 개월당

US\$240
연간 청구

Cloudflare

업무상 중요하지 않은 전문 웹 사이트용.

핵심 기능

Free의 모든 항목 및

- 향상된 WAF(웹 응용 프로그램 방화벽) 기능
- 무손실 이미지 최적화
- 자동 모바일 최적화
- Cache Analytics
- 향상된 봇 완화
- 슈퍼 봇 차단 모드
- Cloudflare 관리 규칙 집합
- WordPress용 DPO 플러그인

Free

US\$0

지원

커뮤니티 및 개발자 문서.

Business

US\$200¹ 개월당

US\$2,400
연간 청구

온라인으로 운영되는 성장 중인 소규모 비즈니스용.

핵심 기능

Pro의 모든 항목 및

- 100% 작동 시간 SLA
- 연중무휴 24시간 채팅 지원
- PCI DSS 3.2 규정 준수
- CNAME 설정 호환성
- 봇 분석
- 사용자 설정 + BYO SSL

Enterprise

사용자 설정

비즈니스의 핵심인 업무상 중요한 응용 프로그램용.

핵심 기능

Business의 모든 항목 및

- 우선 IP 범위
- 솔루션 엔지니어 지원
- 25배 배상 작동 시간 SLA
- 역할 기반 계정 액세스
- 모든 봇에 대한 완화
- 계약 외 서비스에 대한 액세스

개인 또는 취미 프로젝트용.

핵심 기능

- 빠르고 사용하기 쉬운 DNS
- 무제한 DDoS 보호

- 전역 CDN

- 범용 SSL 인증서

- 무료 관리 규칙 집합

- 간단한 봇 완화

3 Page Rules

5 WAF 규칙 ①

DNS를 cloudflare로 이전

[cberi.shop](#)[개요](#)[분석 및 로그](#)[DNS](#)[이메일](#)[SSL/TLS](#)[보안](#)[액세스](#)[Speed](#)[Caching](#)[Workers 경로](#)[규칙](#)[네트워크](#)[Traffic](#)[사용자 설정 페이지](#)[Apps](#)[사이트 바 출소](#)

cberi.shop에 대한 DNS 레코드 추가

클라우드 아이콘 옆의 토글을 클릭하여 A, AAAA 및 CNAME 레코드에 대한 트래픽을 프록싱합니다.



프록싱됨: 트래픽 가속 및 보호



DNS 확인만: Cloudflare 바이패스

참고: 클라우드 아이콘이 없는 레코드는 DNS 확인을 사용하지만 프록시 설정할 수 없습니다.

cberi.shop에 대한 DNS 관리

[DNS 레코드 가져오기 ▾](#)[DNS 레코드 검색](#)

내 서버
IP로
수정해야함

모두 프록싱
OFF로 변경

[레코드 추가](#)

유형	이름	콘텐츠	프록시 상태	TTL	작업
A	ssh	207.148.95.144	<input checked="" type="checkbox"/> DNS 전용	자동	삭제
MX	cberi.shop	eforward3.registrar-ser...	10 DNS 전용	자동	편집 ▶
MX	cberi.shop	eforward2.registrar-ser...	10 DNS 전용	자동	편집 ▶
MX	cberi.shop	eforward1.registrar-ser...	10 DNS 전용	자동	편집 ▶
MX	cberi.shop	eforward4.registrar-ser...	15 DNS 전용	자동	편집 ▶
MX	cberi.shop	eforward5.registrar-ser...	20 DNS 전용	자동	편집 ▶

[삭제](#)

DNS를 cloudflare로 이전

[cberi.shop](#)[개요](#)[분석 및 로그](#)[DNS](#)[이메일](#)[SSL/TLS](#)[보안](#)[액세스](#)[Speed](#)[Caching](#)[Workers 경로](#)[규칙](#)[네트워크](#)[Traffic](#)[사용자 설정 페이지](#)[Apps](#)[사이드바 축소](#)

3. 이름 서버 업데이트

등록 기관에서 이름 서버 목록을 찾습니다. 할당된 Cloudflare 이름 서버를 모두 추가하고 다른 이름 서버를 모두 제거한 다음 변경 내용을 저장합니다.

할당된 Cloudflare 이름 서버:

[jacob.ns.cloudflare.com](#)[클릭하여 복사](#)[tia.ns.cloudflare.com](#)[클릭하여 복사](#)

등록 기관이 변경 내용을 처리하는 데 최대 24시간이 걸립니다(대부분의 경우 더 빠름). Cloudflare에서 cberi.shop이(가) 활성화되면 이메일을 보내드립니다.

이 보류 상태인 동안 Cloudflare가 할당된 이름 서버에 대한 DNS 쿼리에 응답합니다.

활성화되면 SSL/TLS, DDoS 보호, 캐싱 및 기타 자동 최적화가 미리 구성한 사용자 지정 설정과 함께 프록시된 DNS 레코드에 대해 실행됩니다.

[보류 중인 도메인에 관해 더 알아보기](#)

Cloudflare는 주기적으로 이름 서버 업데이트를 확인합니다.

[지금 이름 서버 확인](#)

도움이 필요하십니까? [설정 설명서](#)를 따르거나 [지원 포털](#)을 방문하십시오.

[계속](#)

Dashboard

Domain

Products

Sharing & Transfer

Advanced DNS

Expiring / Expired

Domain List

Hosting List

Private Email

SSL Certificates

Apps

Profile

STATUS & VALIDITY

ALERT

Dec 28, 2023 - Dec 28, 2024

AUTO-RENEW

RENEW

Withheld for Privacy

PROTECTION

Dec 28, 2023 - Dec 28, 2024

AUTO-RENEW

ADD YEARS

SHOW DETAILS

Premium

Custom DNS
로 변경하고

PremiumDNS protection in order to switch your domain to our PremiumDNS platform.
Our PremiumDNS platform, you'll get 99.99% uptime and DDoS protection at the DNS level.

BUY NOW

NAMESERVERS

Custom DNS

cloudflare
에서 알려준
대로 변경

✓ ✗

jacob.ns.cloudflare.com
tia.ns.cloudflare.com

+ ADD NAMESERVER

REDIRECT DOMAIN

You haven't defined any Redirect Domain yet.

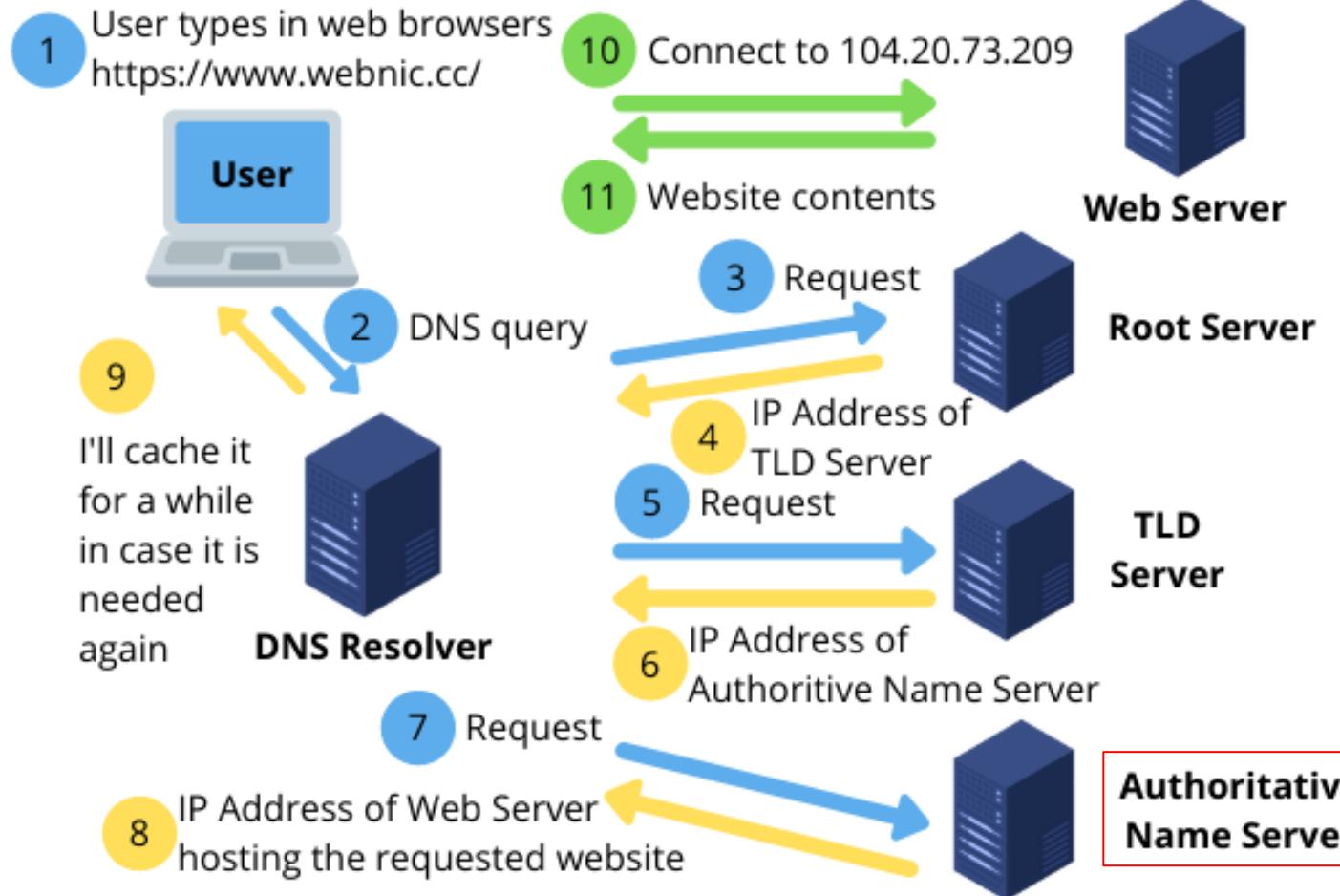
+ ADD REDIRECT

⚡ ADD WILDCARD REDIRECT

REDIRECT EMAIL

You haven't defined any Email Redirect yet.

도대체 무슨 일을 하고 있는 것인가?



- cberi.shop 도메인에 대한 Authoritative Name Server를 변경하고 있는 중
- namecheap.com에서 제공하는 네임서버에서 클라우드 플레이어 제공 네임서버로 이전
- TLD 서버의 정보가 변경되어야 함.
- 변경에 최대 48시간 소요
- 변경 정보가 업데이트 되어야 cloudeflare에서 DNS 기능을 수행할 수 있음.

[cberi.shop](#)

개요

분석 및 로그

DNS

이메일

SSL/TLS

보안

액세스

Speed

Caching

Workers 경로

규칙

네트워크

Traffic

사용자 설정 페이지

Apps

사이드바 축소

빠른 시작 가이드

[뒤로](#)

보안을 개선하고, 성능을 최적화하고, 계정을 최대한 활용하도록 도메인 설정을 구성하십시오.

1 보안 향상

2 성능 최적화

3 요약

일단 제공되는
기능을 모두
꺼놓자

자동 HTTPS 다시 쓰기: [OFF](#)

항상 HTTPS 사용: [OFF](#)

Brotli: [OFF](#)

[마침](#)

연락처

지원팀에 문의

영업팀에 문의

하는 일

요금제

개요

리소스

설명서

블로그

지원

기술 자료

커뮤니티

Cloudflare 소개

Cloudflare 팀

채용

[cberi.art](#)[개요](#)[Analytics](#)[DNS](#)[이메일](#)[SSL/TLS](#)[보안](#)[액세스](#)[Speed](#)[Caching](#)[Workers 경로](#)[규칙](#)[내보이기](#)[사이드바 축소](#)

이름 서버 설정 완료

cberi.art이(가) Cloudflare에서 아직 활성화되지 않았습니다.

1. 등록 기관 계정으로 로그인

[WHOIS](#)를 통해 등록 기관을 확인합니다.

이러한 이름 서버 제거:

ns1.domain.com
ns2.domain.com

2. Cloudflare의 이름 서버로 바꿉니다

 이름 서버 1

jacob.ns.cloudflare.com

[클릭하여 복사](#)

 이름 서버 2

tia.ns.cloudflare.com

[클릭하여 복사](#)

변경사항을 저장합니다.

등록 기관은 이름 서버 업데이트를 처리하는 데 24시간이 걸릴 수 있습니다.

Cloudflare에서 사이트가 활성화되면 이메일을 받습니다.

Cloudflare는 주기적으로 이름 서버 업데이트를 확인합니다. 지금 이름 서버 확인을 시작하려면 이름 서버 확인을 누르십시오.

[이름 서버 확인](#)

[Cloudflare로 전송](#)

활성 구독

[Page Rules](#)

3개의 규칙이 허용됨

[Free 요금제](#)

[변경](#)

지원 리소스

[설명서](#)

[커뮤니티](#)

API

[영역 ID](#)

0fa5bd24fabfde504a6d6d20030d6323

[클릭하여 복사](#)

[계정 ID](#)

ef0997f2690d7176b7afde0532df2fad

[클릭하여 복사](#)

[API 토큰 가져오기](#)

Dianox@cberi.go.kr...

계정 업데이트 및 알림을 받으려면 [이메일 주소 인증](#)을 수행하십시오.

웹 사이트

Discover

도메인 등록

분석 및 로그

Security Center

추적 베타

Turnstile

Zero Trust

Area 1

Workers 및 Pages

AI 베타

R2

Stream

Images

<< 사이드바 축소

Dianox@cberi.go.kr's account

홈

이 계정에서 Cloudflare에 추가된 웹 사이트 및 응용 프로그램이 여기에 나열됩니다. 구성은 업데이트하거나 보안 및 성능을 모니터링하려면 하나를 선택하십시오.

최적화된 이미지 제공으로 웹 성능 평가



간편한 최적화 원활한 크기 조정 글로벌 캐싱 더 빠른 로드 시간 경험 및 개선된 사용자 경험 체감.

[지금 시작하기 >](#)

Dianox@cberi.go.kr's account에서 검색...



검색

+ 사이트 추가

필터링 기준

☆ 별표 표시

cberi.art

 활성

활성화까지
15분 가량
소요되었다.

cberi.shop

 활성

cberi.click

 활성

[← cberi.art](#)[📄 개요](#)[🕒 Analytics](#)[DNS](#)[레코드](#)[설정](#)[✉️ 이메일](#)[🔒 SSL/TLS](#)[🛡️ 보안](#)[⌚ 액세스](#)[⚡ Speed](#)[📋 Caching](#)[◀ 사이드바 축소](#)

cberi.art에 대한 DNS 관리

편집 문서함의 모든 변경 내용은 저장하면 구현됩니다.

가져오기 및 내보내기 ▾ ⚙️ 대시보드 디스플레이 설정

DNS 레코드 검색

+ 필터 추가



검색

+ 레코드 추가

ssh.cberi.art은(는) 158.247.216.243을(를) 가리킵니다.

형식

A

이름 (필수 항목)

ssh

IPv4 주소 (필수 항목)

158.247.216.243

프록시 상태



DNS 전용

TTL

자동

루트에 @ 사용

레코드 특성

☞ 설명서

여기에 제공된 정보는 DNS 레코드 확인에 영향을 주지 않으며 참조용으로만 사용됩니다.

메모

여기에 메모를 입력하십시오(최대 50자).

취소

저장



+ 사이트 추가



지원

한국어



cberi.shop

개요

분석 및 로그

DNS

레코드

설정

이메일

SSL/TLS

보안

액세스

Speed

Caching

Workers 경로

<< 사이드바 축소

cberi.shop에 대한 DNS 관리

DNS 레코드를 검토, 추가 및 편집합니다. 편집 내용이 저장되면 적용됩니다.

가져오기 및 내보내기 ▾ ⚙ 대시보드 디스플레이 설정

DNS 레코드 검색

▽ 필터 추가



검색

+ 레코드 추가

유형	이름	콘텐츠	프록시 상태	TTL	작업
A	admin	207.148.95.144	DNS 전용	자동	편집 ▶
A	cberi.shop	207.148.95.144	DNS 전용	자동	편집 ▶
A	ftp	207.148.95.144	DNS 전용	자동	편집 ▶
A	sftp	207.148.95.144	DNS 전용	자동	편집 ▶
A	sqladmin	207.148.95.144	DNS 전용	자동	편집 ▶
A	squid	207.148.95.144	DNS 전용	자동	편집 ▶
A	ssh	207.148.95.144	DNS 전용	자동	편집 ▶
A	vpn	207.148.95.144	DNS 전용	자동	편집 ▶
A	www	207.148.95.144	DNS 전용	자동	편집 ▶

dns 응답 확인

https://toolbox.googleapps.com/apps/dig

Google 관리 콘솔 도구 상자 Dig 도움말

이름
ssh.cberi.shop

A AAAA ANY CAA CNAME DNSKEY DS MX NS PTR SOA SRV TLSA TSIG TXT

TTL:
5 minutes

A
DATA:
207.148.95.144

proftpd (FTP 서버 데몬) 설치

```
[root@cberi ~]# dnf list installed | grep proftpd
```

```
[root@cberi ~]# dnf list | grep proftpd
```

proftpd.x86_64	1.3.6e-6.el8	epel
proftpd-devel.x86_	1.3.6e-6.el8	epel
proftpd-ldap.x86_	1.3.6e-6.el8	epel
proftpd-mysql.x86	1.3.6e-6.el8	epel
proftpd-postgresql.x86	1.3.6e-6.el8	epel
proftpd-sqlite.x86_	1.3.6e-6.el8	epel
proftpd-utils.x86	1.3.6e-6.el8	epel

```
[root@cberi ~]# dnf install -y proftpd
```

Installed:

libmemcached-libs-1.0.18-17.el8.x86_64	proftpd-1.3.6e-6.el8.x86_64
--	-----------------------------

Complete!

```
[root@cberi ~]# systemctl start proftpd
```

```
[root@cberi ~]# systemctl enable proftpd
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/proftpd.service →
/usr/lib/systemd/system/proftpd.service.
```

proftpd (FTP 서버 데몬) 설치

■ SELinux 관련 설정

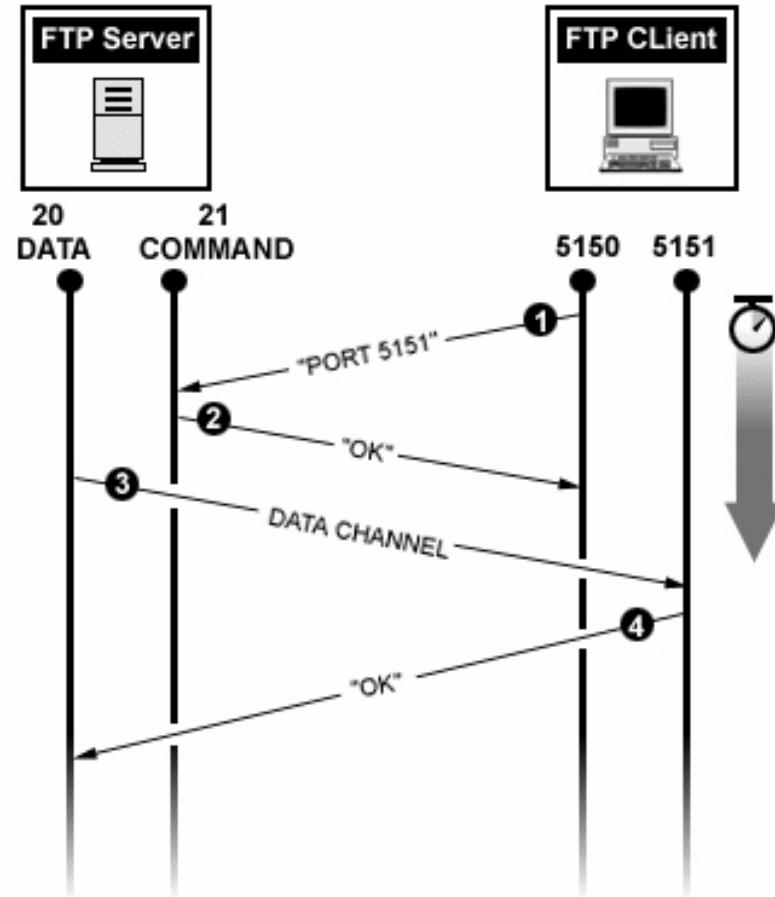
```
[root@cberi etc]# setsebool -P ftpd_full_access=1  
[root@cberi etc]# setsebool -P ftpd_use_passive_mode=1  
  
[root@cberi etc]# ls -lhF proftpd.conf proftpd/  
-rw-r-----. 1 root root 6.5K Jan 12 16:08 proftpd.conf  
  
proftpd/:  
total 28K  
-rw-r-----. 1 root root 1.9K Dec 29 00:09 anonftp.conf  
drwxr-xr-x. 2 root root 4.0K Dec 29 00:10 conf.d/  
-rw-r-----. 1 root root 597 Dec 29 00:09 mod_ban.conf  
-rw-r-----. 1 root root 359 Dec 29 00:09 mod_qos.conf  
-rw-r-----. 1 root root 604 Dec 29 00:09 mod_tls.conf  
-rw-r-----. 1 root root 7.0K Jan 12 14:00 modules.conf
```

■ 설정 파일 수정(옵션사항)

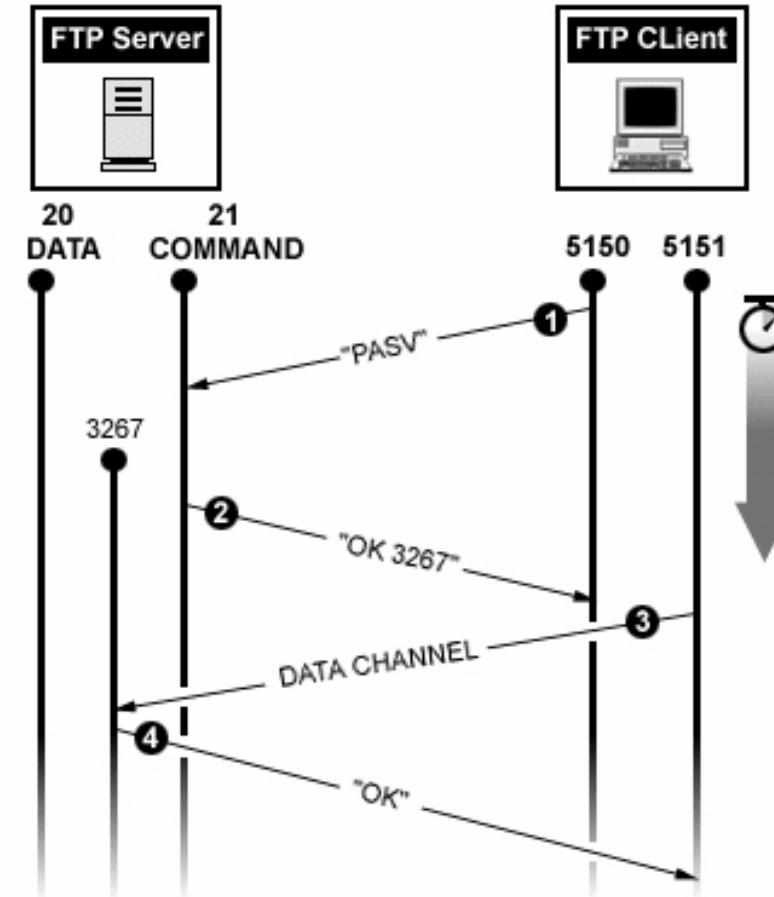
```
[root@cberi ~]# vim /etc/proftpd.conf  
:  
:  
# 영문 메시지로 출력되도록 변경  
<IfModule mod_lang.c>  
    LangDefault en_US  
    LangEngine on  
    UseEncoding utf-8 utf-8  
    LangPath /usr/share/locale  
</IfModule>
```

ftp 전송 모드 비교

■ Active Mode



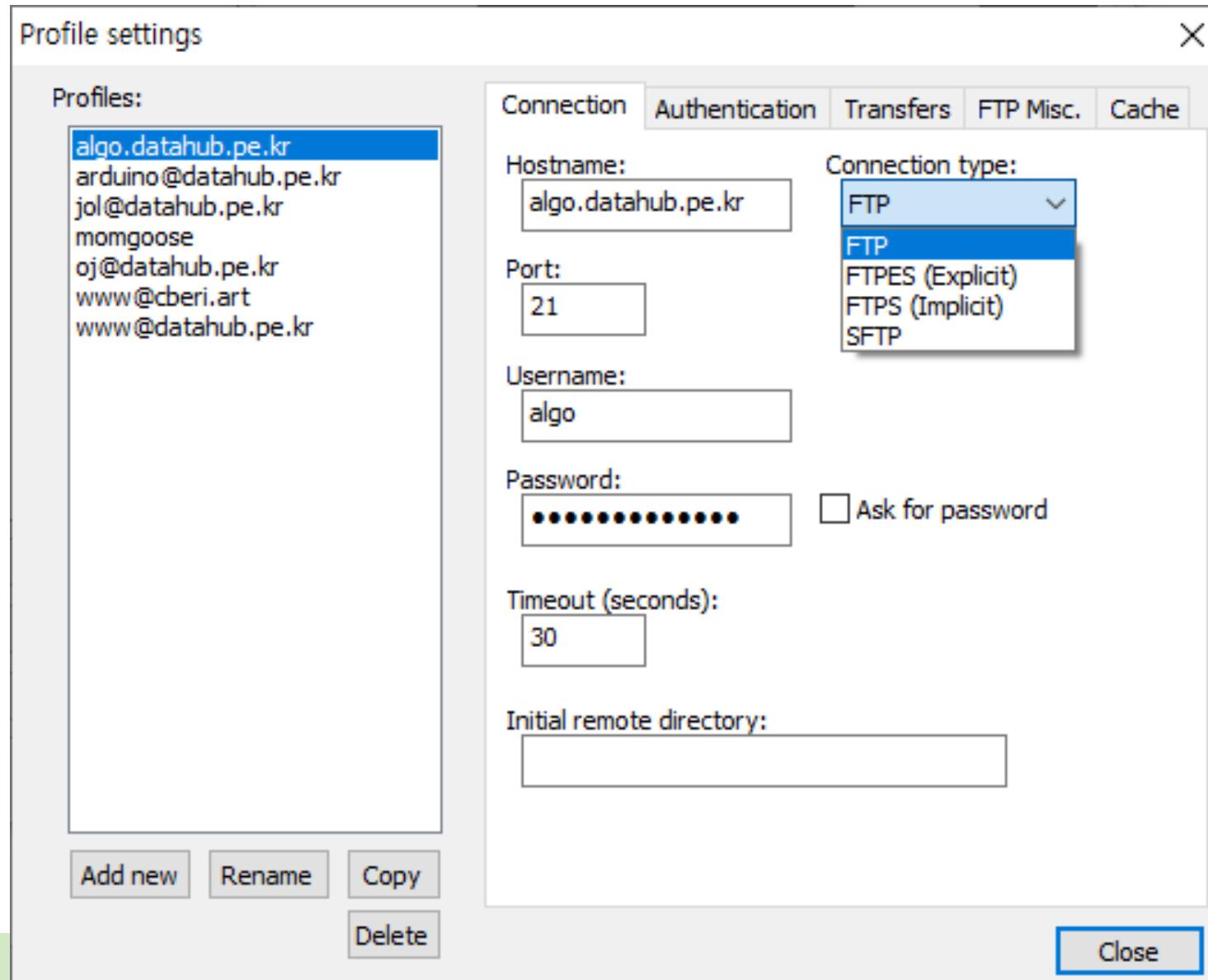
■ Passive Mode



Passive Mode
를 활성화하기
위한 별도의
proftpd 설정
필요 없음.

클라언트가
Passive
Mode를
요구하면
서버가 그렇게
응대해줌.

FTP 프로토콜 4가지 비교



■ Connection type???

- FTP
- FTPES(Explicit)
- FTPS(Implicit)
- SFTP

FTP 프로토콜 4가지 비교

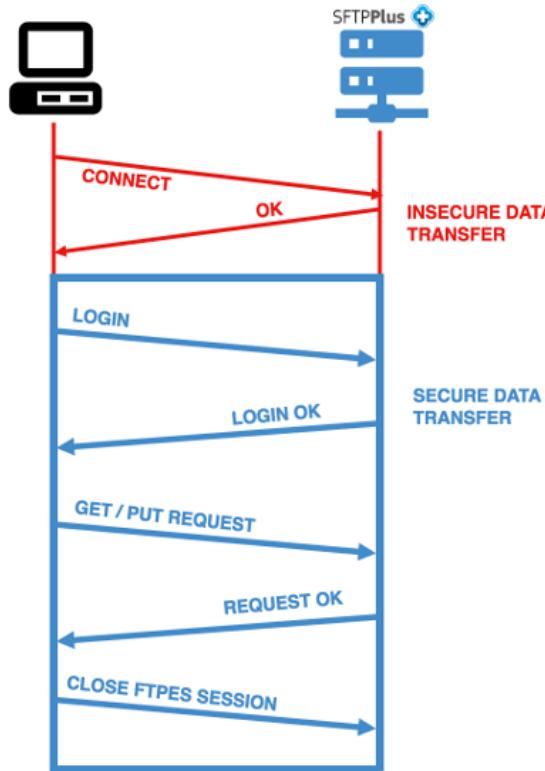
- SFTP(Secure shell FTP)
- FTP(암호화 없는 전통적인 방식)



FTP 프로토콜 4가지 비교

■ FTPS(Implicit, 암시적)

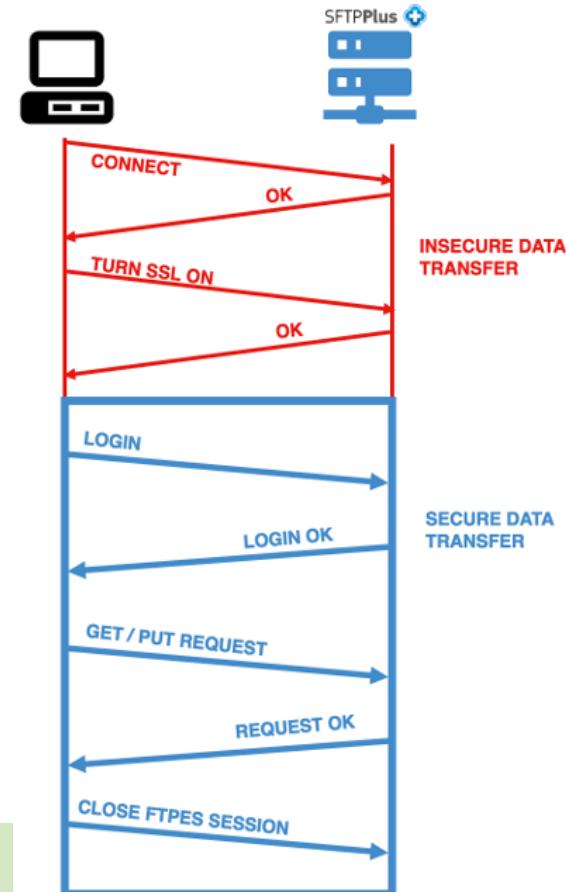
- 최초의 FTP 보안 프로토콜(고전방식)
- 990 포트 사용



요구하지
않아도
암호화
통신이 항상
켜져 있다

■ FTPES(Explicit, 명시적)

- 요즘 많이 사용되는 방식
- 기존 21번 포트 사용



클라이언트에서
암호화 통신을
요구해야 켜진다.

준비물

■ Filezilla FTP client

- <https://filezilla-project.org/>

The screenshot shows the official website for FileZilla. The header features the "FileZilla" logo and the tagline "The free FTP solution". The left sidebar contains links for Home, Features, Screenshots, Download, Documentation, and FileZilla Pro. Under "FileZilla Server", there are links for Download, Community, Forum, and Wiki. The "General" section includes links for FAQ, Support, Contact, License, Privacy Policy, and Trademark Policy. The "Development" section lists Source code, Nightly builds, Translations, Version history, Changelog, and Issue tracker. The "Other projects" section includes libfilezilla and Octochess. Sponsors are listed at the bottom, including NDC (Netwrix Data Center Audit). The main content area has a "Promotion" banner for FileZilla. The "Overview" section welcomes visitors to the homepage of FileZilla, the free FTP solution. It mentions the GNU General Public License and offers FileZilla Pro for additional protocol support. The "Quick download links" section provides two prominent buttons: "Download FileZilla Client" (All platforms) and "Download FileZilla Server" (All platforms). A note below the buttons says, "Pick the client if you want to transfer files. Get the server if you want to r...". The "News" section highlights the release of FileZilla Client 3.63.1 on January 26, 2023, with a note about MSW icon sizing issues.

■ Nodepad++

- <https://notepad-plus-plus.org/downloads/v8.4.8/>

Download 64-bit x64



- [Installer | GPG Signature](#)
- [Portable \(zip\) | GPG Signature](#)
- [Portable \(7z\) | GPG Signature](#)
- [Mini-portable \(7z\) | GPG Signature](#)

텍스트 에디터 vim (vi improve)

■ vim 모드

vi 를 처음 실행시키면 입력모드가 아닌 명령모드 상태에 놓이게 된다. 이 상태에서는 문자의 입력이 이루어지지 않으며, 찾기 등과 같은 간단한 문서관련 명령을 실행할 수 있다.

이 명령모드 상태에서 ":" 키를 누르면 ex 상태로 a, i, o 키 등을 누르면 입력 상태로 넘어가게 된다. 일단 입력상태로 들어가게 되면 문서 편집을 할 수 있게 되는데, 이때 ESC 키를 누르면 명령모드 상태로 넘어가게 된다.

■ vim의 상태

명령 상태	처음 vim을 실행 했을때, 입력상태 / 명령상태에서 ESC 입력시	간단한 찾기, 커서 이동, ex 상태로 넘어가기
ex 상태	명령 상태에서 ("")	각종 치환, 저장, 파일읽기, vim 설정등과 같은 대부분의 작업들
입력 상태	명령 상태에서 (a,i,o 키 입력)	내용 입력

vim (vi) 명령어 모드

■ 커서 이동

k	커서를 위로 움직임
j	커서를 아래로 움직임
h	커서를 왼쪽으로 움직임
l	커서를 오른쪽으로 움직임
-	커서를 줄의 처음으로 옮김
e, E	다음단어의 끝으로, 문자단위 이동
w, W	다음단어의 처음으로, 문자단위 이동
\$	줄의 마지막으로
0	줄의 처음으로
^	줄의 처음으로(공백이 아닌 처음시작되는 문자)
Shift+g	문서의 마지막으로 이동한다.
gg, 1g	문서의 처음으로 이동한다. 1대신 다른 숫자를 입력하면 해당 숫자의 라인으로 이동한다.
), (다음, 이전 문장의 처음으로
}, {	다음, 이전문단의 처음으로
]], [[다음, 이전 구절의 처음으로

■ 화면 스크롤

^F	한 화면을 앞으로 스크롤
^B	한 화면을 뒤로 스크롤
^D	반 화면을 앞으로 스크롤
^U	반 화면을 뒤로 스크롤
^E	한줄 앞으로 스크롤
^Y	한줄 뒤로 스크롤
Shift + h	화면의 맨 윗줄로
Shift + m	화면의 중간줄로
Shift + l	화면의 맨 아랫줄로

■ 입력 명령

i	현재위치에서 삽입
I	현재줄의 처음 위치에서 삽입
a	현재위치에서 한칸앞으로 이동해서 삽입
A	현재줄의 마지막위치에서 삽입
o	새로운 줄을 커서 아래에 연다
O	새로운 줄을 커서 위연다
s	현재 위치의 문자를 지우고 입력모드로 들어간다.
S	현재위치의 라인을 지우고 입력모드로 들어간다.

vim (vi) 명령어 모드

■ 편집명령(non-visual block모드)

y	한줄 복사
yn	현재 라인에서부터 n라인만큼을 복사
p	복사된 내용 붙이기
dd	한줄삭제
dw	한단어 삭제
Shift+d, d\$	현재커서 위치에서 마지막까지 삭제
Shift+j	현재 행의 개행문자를 제거한다. 즉 아래라인을 현재라인에 덧붙인다.

■ 되돌리기(undo)

- ‘u’ 키만 입력하면 된다.

■ 블록 지정

v	단어단위로 블럭지정이 가능하다. 블럭범위는 이동명령인 'hjkl'로 원하는 범위 만큼 지정할수 있다.
Shift+v	라인단위 블럭지정이다. 라인전체가 선택되며, 위아래 이동명령 'hj'으로 범위 지정이 가능하다.
Ctrl+v	블럭단위 블록 지정이다. 4각형의 블록 지정이 가능하며 이동명령인 'hjkl'로 원하는 범위를 지정할 수 있다.
^U	반 화면 을 뒤로 스크롤
^E	한줄 앞으로 스크롤
^Y	한줄 뒤로 스크롤
Shift + h	화면의 맨 윗줄로
Shift + m	화면의 중간줄로
Shift + l	화면의 맨 아랫줄로

■ 편집(visual block 모드)

y	지정된 블럭을 복사한다.
p	복사된 블럭을 현재라인(커서) 아래에 붙인다.
d	지정된 블럭을 삭제한다.

vim (vi) ex 모드

■ 파일 저장, 열기, 종료

:e [filename]	filename 으로 파일열기
:q, :q!, :wq	종료, 강제종료, 저장후 종료
:w, :w [filename]	현재파일명으로 저장, filename 로 저장
:<범위>w [filename]	지정한 범위만 다른 파일로 저장
:e [filename]	filename 을 편집하기 위해서 연다
zz	지금파일을 저장하고 vim을 종료한다.

■ 문자열 찾기

찾기 기능은 ':/패턴/' 를 이용 하면된다. 찾기 원하는 문자혹은 패턴을 입력하고 엔터키를 누르면 현재 커서위치에서 가장 가까운 곳에 위치한 문자열로 커서를 이동시킨다(문서 아래방향으로). 다음 문자열을 찾기를 원한다면 'n'키를 누르면 된다.

■ 문자열 치환

:[범위]s/[oldpattern]/[newpattern]/ 의 형식으로 사용하면 된다. 범위 지정은 visual block 을 이용할수도 있으며, 직접 범위를 입력할수도 있다. visual block 를 이용한 치환은 visual block 를 지정한 다음 ':' 를 입력해서 ex 모드로 넘어가면 된다.

vim (vi) ex 모드

■ 문자열 치환

- 명령어 형식

:시작행, 끝행s/원래문자열/변경문자열/옵션

- 문자열에 정규 표현식 사용 가능
- 이스케이프 문자 \
 - http:// → http:\/\/\//
- 1~10행 사이에서 → :1,10
- 현재 커서가 위치한 행 → .
- 마지막행 → \$
- 현재행부터 마지막 행까지 → :.,\$

■ 옵션

- g : 여러 번 나오면 계속 변경
- i : 대소문자 구분하지 않고 검색
- c : 치환 시 사용자 확인 요구

■ 예시

- 1~10행의 모든 Hello를 Bye로 변경
:1,10s/Hello/Bye/g
- 문서 전체의 모든 Hello를 Bye로 변경
:%s/Hello/Bye/g
- 현재 행에서 마지막 행 까지
:.,\$s/Hello/Bye/g

sshd 보안설정

■ 고찰

- 일반 사용자는 웹 호스팅을 위한 파일 업로드가 주목적
- 비관리자 계정에 ssh 접속을 허용할 이유 없음
- ssh를 비관리자 계정에 열어주면 보안 취약점으로 작용 가능성
- 일반 사용자에게는 ssh 접속은 막고, sftp만 열어주기
- sftp로 접속해도 자신의 홈 폴더에서 탈출하지 못하도록 설정 필요

■ chroot 적용

```
# useradd sql
# echo 'pororo23@sql' > passwd --stdin sql
# vi /etc/ssh/sshd_config

# 맨 마지막 줄에 추가
Match User sql
    ChrootDirectory %h
    ForceCommand internal-sftp

# cd ~sql
# chown root.root .
# chmod 775 .
# mkdir html
# chown sql.sql html
```

Squid로 Proxy Server 구축

- 프록시 서버(Proxy server)
 - 사전적인 의미로는 '대리인'
 - 클라이언트가 자신을 통해서 다른 네트워크 서비스에 간접적으로 접속할 수 있게 해 주는 컴퓨터나 응용 프로그램
 - 프록시 서버에 요청된 내용들을 캐시에 저장 후, 캐시 안에 있는 정보를 이용함으로써 불필요하게 외부와의 연결을 하지 않아 전송 시간을 절약됨
 - 사용목적
 - 원하지 않는 사이트를 차단을 위해
 - IP 추적을 당하지 않을 목적으로
 - 악의적인 의도로 바이러스, 악성 루머 전파 또는, 다른 정보들을 빼낼 목적으로
- 스퀴드(Squid)
 - <http://www.squid-cache.org/>
 - 사전적인 의미로는 '오징어'
 - 오픈 소스(GPL) 소프트웨어 프록시 서버이자 웹 캐시
 - 반복된 요청을 캐싱함으로 웹서버의 속도를 향상시킴
 - 네트워크 자원을 공유하려는 사람들에게 웹, DNS와 다른 네트워크 검색의 캐싱을 제공함
 - 트래픽을 걸러줌으로써 안정성에 도움을 주는 등에 이르기까지 광범위하게 이용됨

Squid로 HTTP Proxy Server 구축

- Squid 설치

```
# dnf install -y squid
```

완료되었습니다!

- 설정파일 수정

- 모두에게 허용

```
# vi /etc/squid/squid.conf
```

```
48 #
49 # INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM \
50 #
51 acl anybody src 0.0.0.0/0.0.0.0
52 http_access allow anybody
```

- 일부 IP에만 허용 (추천하는 방식)

```
51 acl members src 220.86.241.0/24
52 http_access allow members
```

- 방화벽 개방 (이미 앞에서 했음)

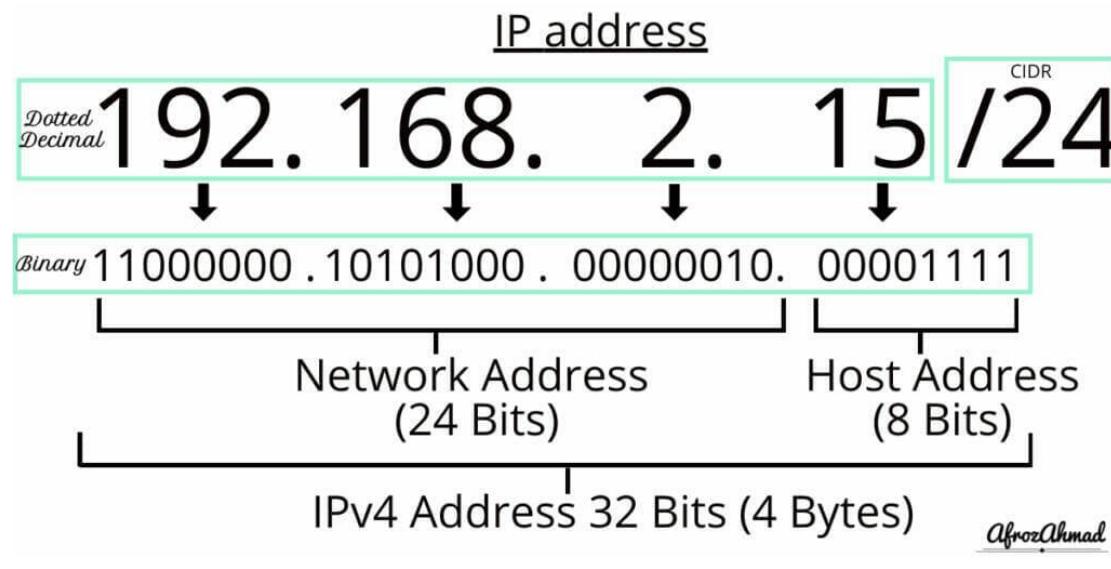
```
# firewall-cmd --permanent --add-port=3128/tcp
success
# firewall-cmd --reload
success
```

- 서비스 등록 및 시작

```
# systemctl start squid
# systemctl enable squid
Created symlink /etc/systemd/system/multi-
user.target.wants/squid.service →
/usr/lib/systemd/system/squid.service.
```

CIDR 표기법

- net id 와 host id

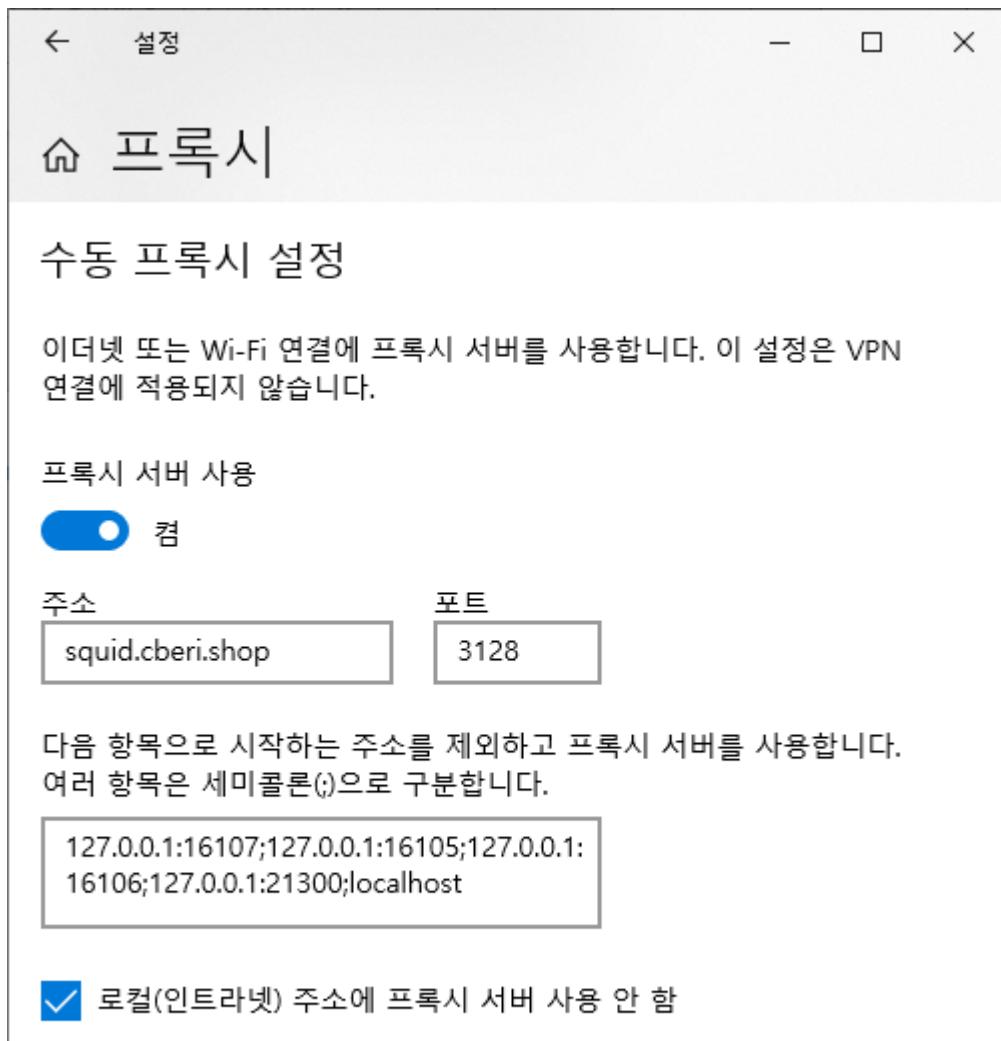


- subnet mask

CIDR	Subnet Mask	CIDR	Subnet Mask
/8	255.0.0.0	/21	255.255.248.0
/9	255.128.0.0	/22	255.255.252.0
/10	255.192.0.0	/23	255.255.254.0
/11	255.224.0.0	/24	255.255.255.0
/12	255.240.0.0	/25	255.255.255.128
/13	255.248.0.0	/26	255.255.255.192
/14	255.252.0.0	/27	255.255.255.224
/15	255.254.0.0	/28	255.255.255.240
/16	255.255.0.0	/29	255.255.255.248
/17	255.255.128.0	/30	255.255.255.252
/18	255.255.192.0	/31	255.255.255.254
/19	255.255.224.0	/32	255.255.255.255
/20	255.255.240.0		

Proxy Server 이용 클라이언트 설정

■ Windows 10 설정



■ 유의사항

- Squid 는 HTTP 프록시 임
- 따라서, 오직 웹 연결시에만 프록시 서버를 우회한다.
- 즉 SSH, FTP 등 웹을 제외한 다른 연결에는 나의 익명성을 보장해 주지 못한다.
- 모든 연결에 IP 익명성을 보장받으려면 VPN 을 사용할 것!!!

HTTP Proxy Server 테스트

- 접속지 변경 확인 (<https://dnschecker.org/ip-location.php>)

IP Address Lookup - Instantly

dnschecker.org/ip-location.php

IP2Location

IP: 2001:19f0:7001:59b1:5400:4ff:feb3:64d6

Country: Japan

State: Tokyo

City: Tokyo

Latitude: 35.6894

Longitude: 139.6923

ISP: The Constant Company LLC

IP Location Services by: IP2Location
[Get Protected Now!](#)

Load More

35°41'21.8"N 139°41'32.3"E
〒160-0023 東京都新宿区西新宿2丁目

View larger map

Directions

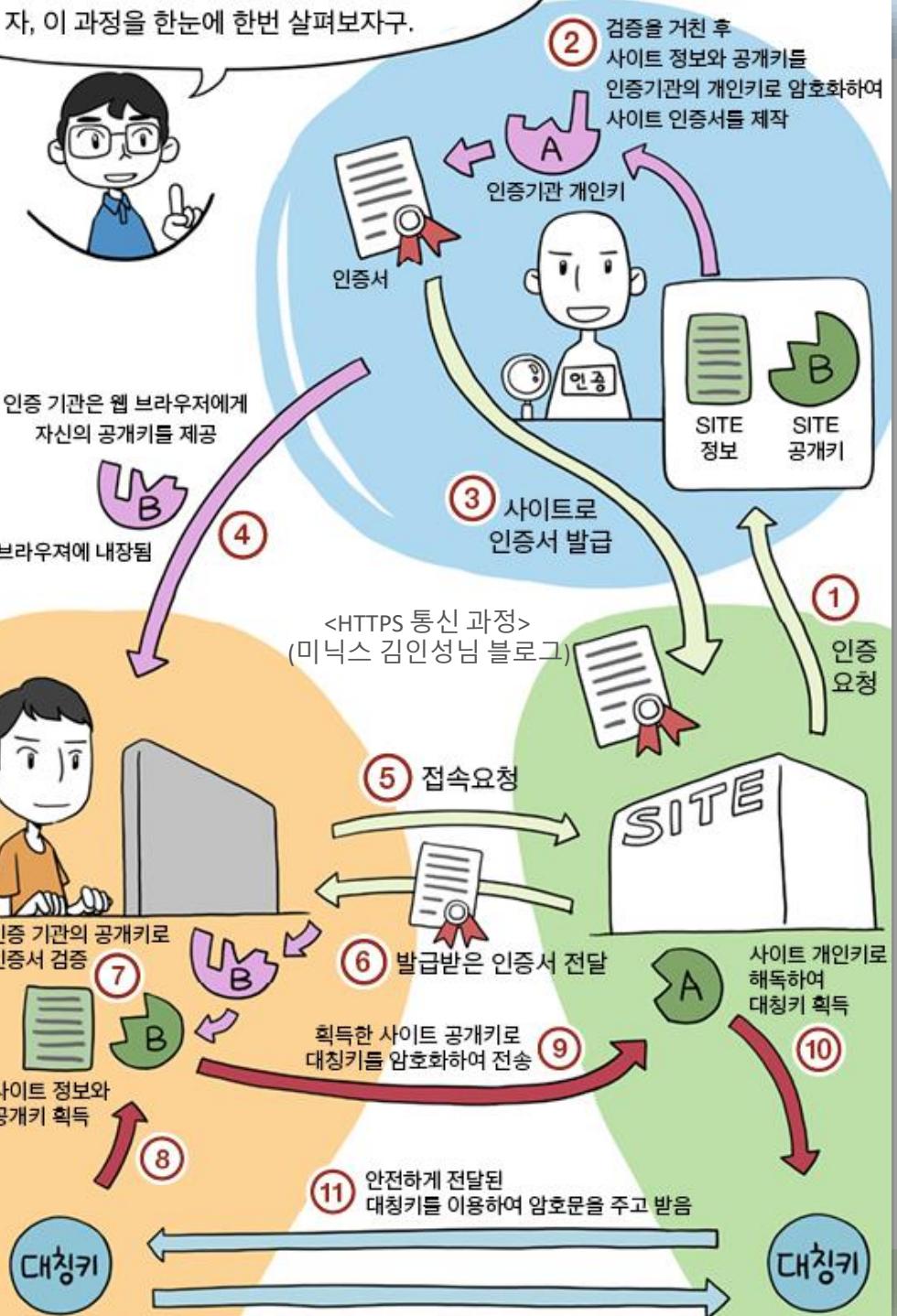
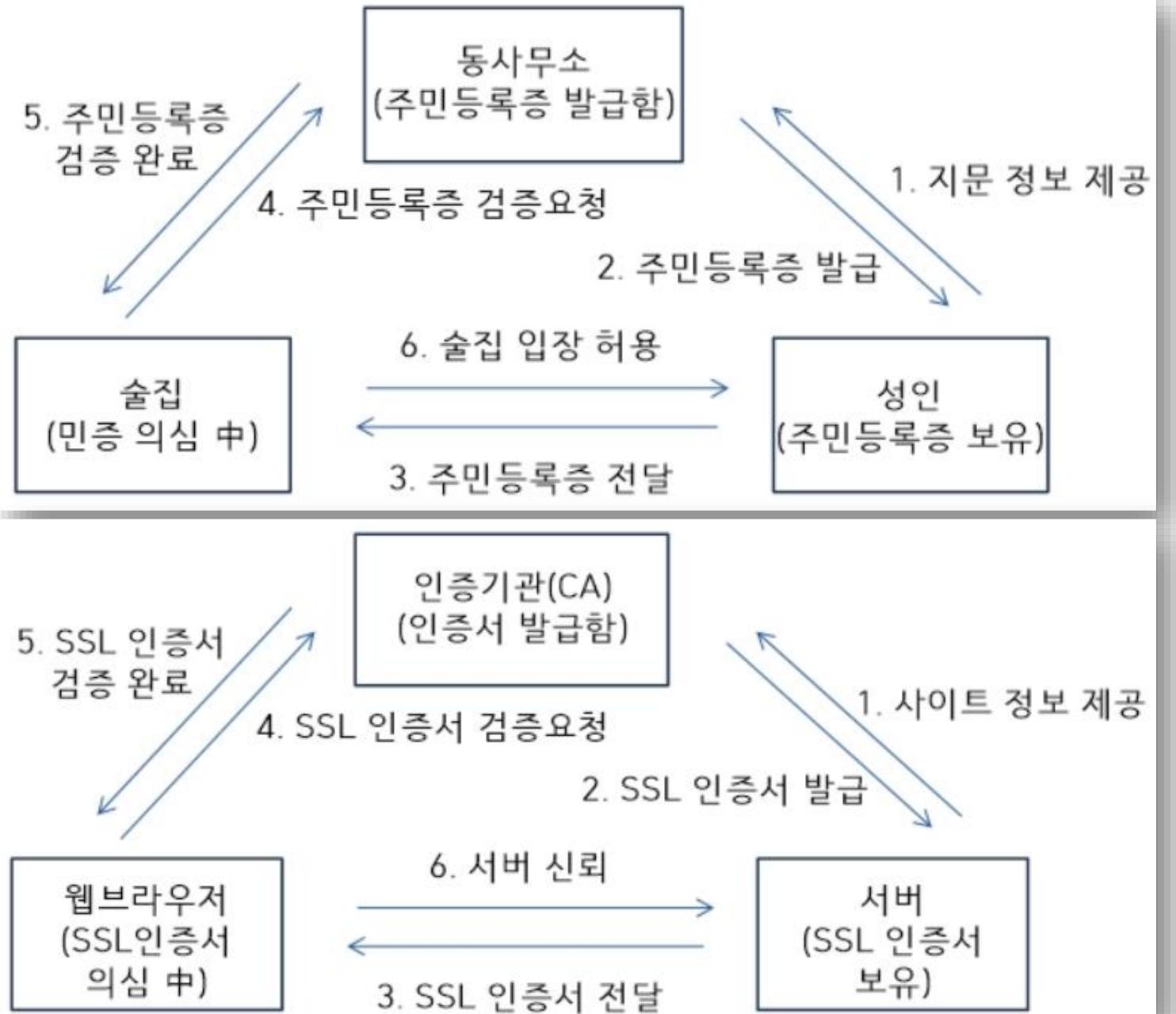


HTTP Proxy Server 테스트

- 한국에서는 접속 못하는 스웨덴 과학센터 navet.com 테스트

The screenshot shows the homepage of the Navet Science Center website. The header features the Navet logo (a stylized molecule icon) and the word "NAVET". The navigation menu includes "Start", "Allmänhet, helger & lov", "Skola & utbildning", "Möte & Konferens", "Kalas", and "Om Navet". Below the menu, there are four main columns: "Besökare" (Visitor) featuring a squirrel image and text about winter focus; "Sportlov" (Sports Day) featuring a cartoon character and text about the event; "Navet Science Club" featuring a plasma ball image and text about joining; and "Koda med Navet på Sportlovet!" (Coding with Navet at Sports Day) featuring a smartphone with code and text about coding. At the bottom, there are two more columns: "Kalas på Navet" (Party at Navet) and "Vi bygger ny fasad" (We are building a new facade), both with small images and text.

HTTPS 통신과정



2억 2500 만개의 사이트에 TLS 인증서를 제공하는 비영리 인증 기관입니다.

Read all about our nonprofit mission

시작하기



Create Free SSL Certificate

 HTTPS Enter Primary Domain

Get Free SSL

Features

Developer

Pricing

Partner Program

Log In

출처: 블로그

2023. 1. 19.

Thank you to our 2023 renewing sponsors!

Let's Encrypt is a nonprofit service and our long-time renewing sponsors play a major role in making that possible.

신청하기

상세보기

PremiumSSL WildCard

기관(소유자) 정보 표시! 및 2차 도메인 SSL 적용
무제한!

배상금:\$250,000

650,000원/년

신청하기

상세보기

 Trusted Certificate Authority

**SSL Protection
For Anyone
Fast. Reliable. Free.**

Easily secure any site by putting SSL management on autopilot,
supporting one-step validation and renewal via REST API.

1 Select Cert & Domain

- Select Domains
- 90-Day Certificate
- 1-Year Certificate



Amazon.com



90

2 CSR & Validation

- Generate CSR
- One-Step SSL Validation



3 Certificate Issued

- Install Certificate
- Site Secured



HTTPS

Securing Leaders in the Industry



cockpit에 Let's encrypt SSL 인증서 적용

■ 인증서 발급

```
[root@cberi ~]# sudo dnf install certbot -y

[root@cberi ~]# certbot certonly --manual --email akapo@naver.com \
--server https://acme-v02.api.letsencrypt.org/directory \
--agree-tos -d *.cberi.shop
Saving debug log to /var/log/letsencrypt/letsencrypt.log
```

```
- - - - -  
Account registered.
```

```
Requesting a certificate for *.cberi.shop  
- - - - -
```

```
Please deploy a DNS TXT record under the name:
```

```
_acme-challenge.cberi.shop.
```

```
with the following value:
```

```
LNTa0hd3k0AgmDs0JWCXeKNTJR4ajwUfA5UeTHF81b8
```

- 도메인 소유자임을 증명해야 함.
- cloudflare DNS에서 TXT 레코드 생성

cockpit 에 Let's encrypt SSL 인증서 적용

■ 발급된 인증서 적용

Before continuing, verify the TXT record has been deployed. Depending on the DNS provider, this may take some time, from a few seconds to multiple minutes. You can check if it has finished deploying with aid of online tools, such as the Google Admin Toolbox:

https://toolbox.googleapps.com/apps/dig/#TXT/_acme-challenge.cberi.site.

Look for one or more bolded line(s) below the line ';ANSWER'. It should show the value(s) you've just added.

Press Enter to Continue

```
[root@cberi ~]# cd /etc/letsencrypt/live/cberi.shop/  
# cp fullchain.pem /etc/cockpit/ws-certs.d/wildcart.cberi.shop.cert  
# cp privkey.pem /etc/cockpit/ws-certs.d/wildcart.cberi.shop.key  
# /usr/libexec/cockpit-certificate-ensure --check  
  
# systemctl restart cockpit.service
```

■ 3개월 기간의 인증서

인증서 뷰어: *.cberi.shop

일반(G) 세부 정보(D)

발급 대상

CN(일반 이름)	*.cberi.shop
조직 (O)	<인증서의 일부가 아님>
OU(조직 구성 단위)	<인증서의 일부가 아님>

발급자

CN(일반 이름)	R3
조직 (O)	Let's Encrypt
OU(조직 구성 단위)	<인증서의 일부가 아님>

유효 기간

발급 날짜	2024년 1월 15일 월요일 오전 12:29:28
만료 날짜	2024년 4월 14일 일요일 오전 12:29:27

SHA-256 지문

인증서	86e6137ec7dc94dae6fd3bd3adb14b67f07dedb8dc60f6b6a2ef2c
공개 키	e943701bdc4f43d2757704da4a0ff28990aebf873a00b6563bfaf880

3. Docker(podman)의 활용

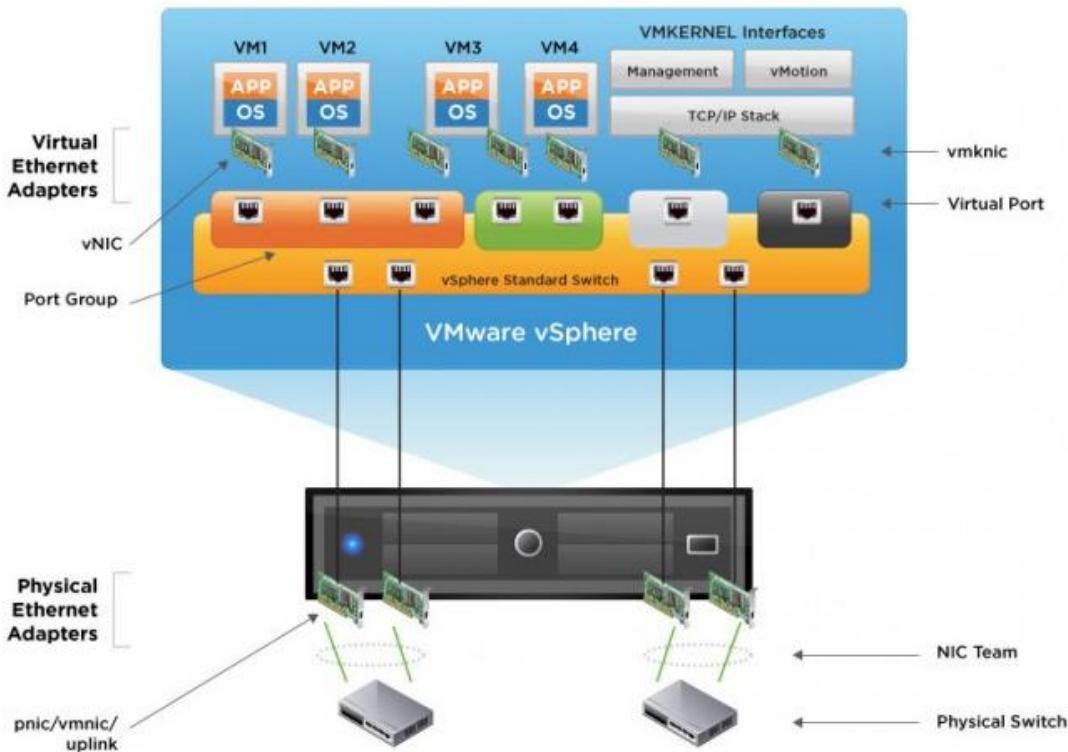
- 1) Docker 소개
- 2) Docker 명령어
- 3) Docker 활용 서비스 구축
 - ① nginx 웹 서버 구동
 - ② mysql (mariadb)
 - ③ l2tp / IKEv2 VPN Server

충북교육연구정보원 강사 박정진

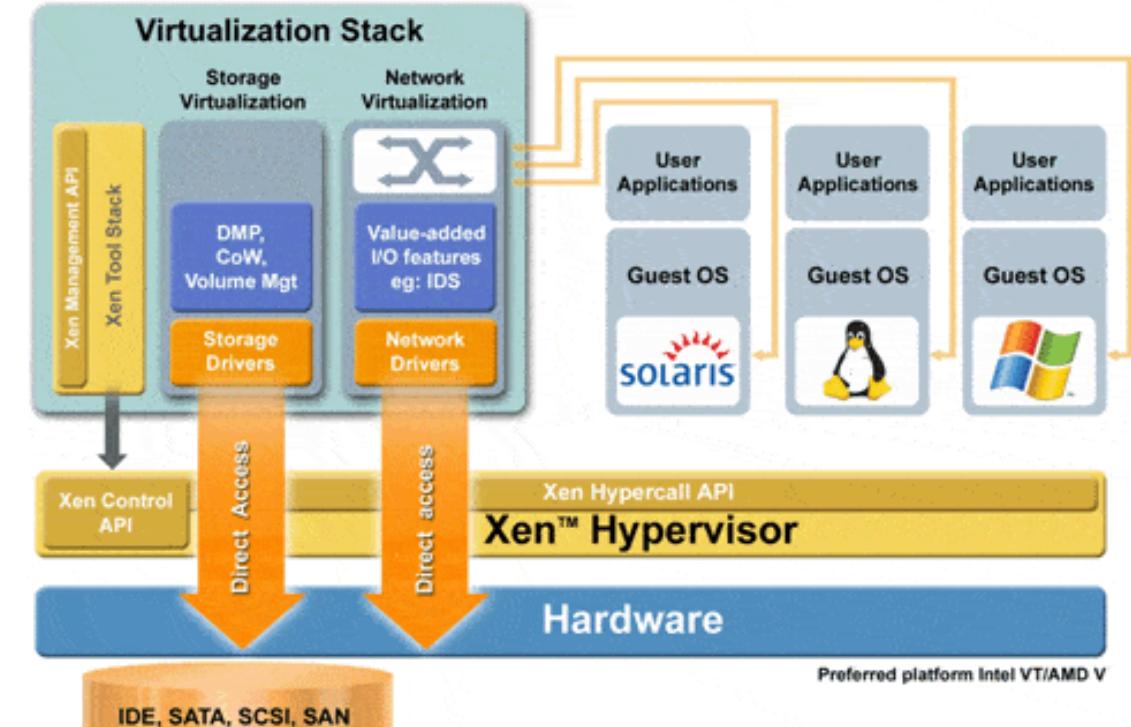


가상화 솔루션

■ VMware vSphere



■ Citrix Xen Server

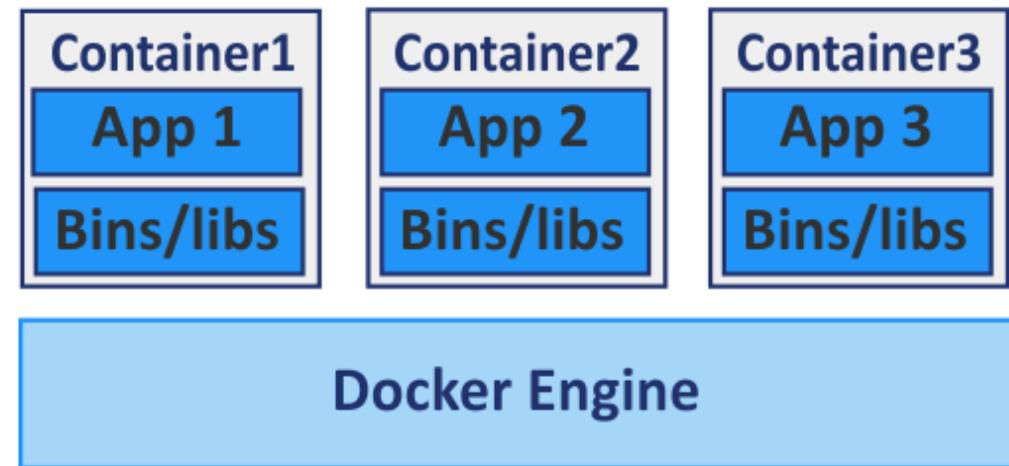


VM vs Docker

Virtual Machines



Containers





- 도커 (Docker) 란?

- 도커는 리눅스의 응용 프로그램들을 프로세스 격리 기술들을 사용해 컨테이너로 실행하고 관리하는 오픈 소스 프로젝트이다.
- 도커 컨테이너는 일종의 소프트웨어를 소프트웨어의 실행에 필요한 모든 것을 포함하는 완전한 파일 시스템 안에 감싼다. 여기에는 코드, 런타임, 시스템 도구, 시스템 라이브러리 등 서버에 설치되는 무엇이든 아우른다. 따라서 실행 중인 환경에 관계 없이 언제나 동일하게 실행될 것을 보증한다

- 컨테이너 (Container)

- 애플리케이션과 애플리케이션을 구동하는 환경을 격리하는 공간을 뜻하며, 호스트 OS의 커널을 공유하면서 격리된 컴퓨팅 자원을 제공하는 가상화 기술이다.
- 가상 머신(VM)은 하드웨어 스택을 가상화하지만, 컨테이너는 운영체제 수준에서 가상화를 실시하여 다수의 컨테이너를 호스트의 OS 커널에서 직접 구동한다.



- 컨테이너의 장점

- 신속성이 높다

VM은 가상 하드웨어를 준비하고 그 위에서 Guest OS를 부팅해야하기 때문에 부팅시간이 느린다. 하지만 컨테이너는 프로세스를 띄우는 정도만 수행하면 되기 때문에 부팅시간이 빠르다. 이와 더불어 추가 배포 시간도 짧다는 장점이 있다.

- 메모리 사용량이 적다

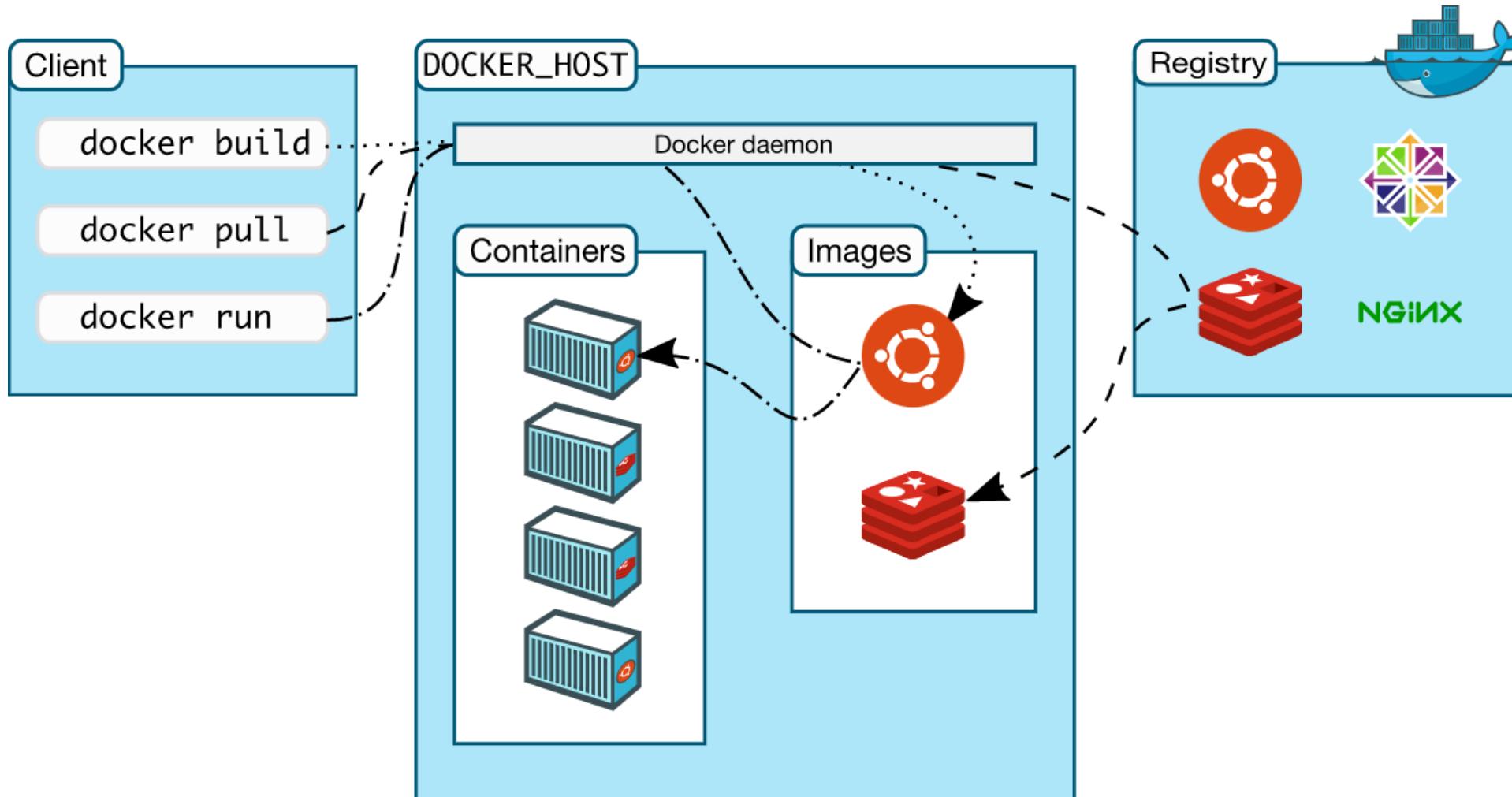
VM은 게스트 OS가 별도로 실행되어야 하기 때문에 시스템 메모리가 차지하는 메모리가 크다. 하지만 컨테이너는 게스트 OS를 띄우지 않기 때문에 애플리케이션과 애플리케이션이 구동될 때 필요한 라이브러리와 환경 정도의 적은 메모리를 사용한다.

- 효율성이 높다

많은 기업이 애플리케이션 서비스를 안정적으로 운영하기위해, 1개의 가상머신에 1개의 서비스를 구동하는 것을 권장한다. 하지만, 해당 애플리케이션이 가상머신의 모든 자원을 사용하는 것이 아니기 때문에 유휴자원이 생겨 성능적으로 오버헤드가 생길 수 있다. 반면 컨테이너는 운영체제의 자원을 공유하기 때문에 애플리케이션을 실행하는데 필요한 만큼의 자원만 할당받을 수 있다. 즉, 서버 전체 자원을 효율적으로 사용할 수 있다

도커의 구성 요소

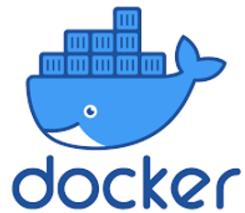
- Docker Container + Docker Image + Docker Registry



도커의 구성 요소

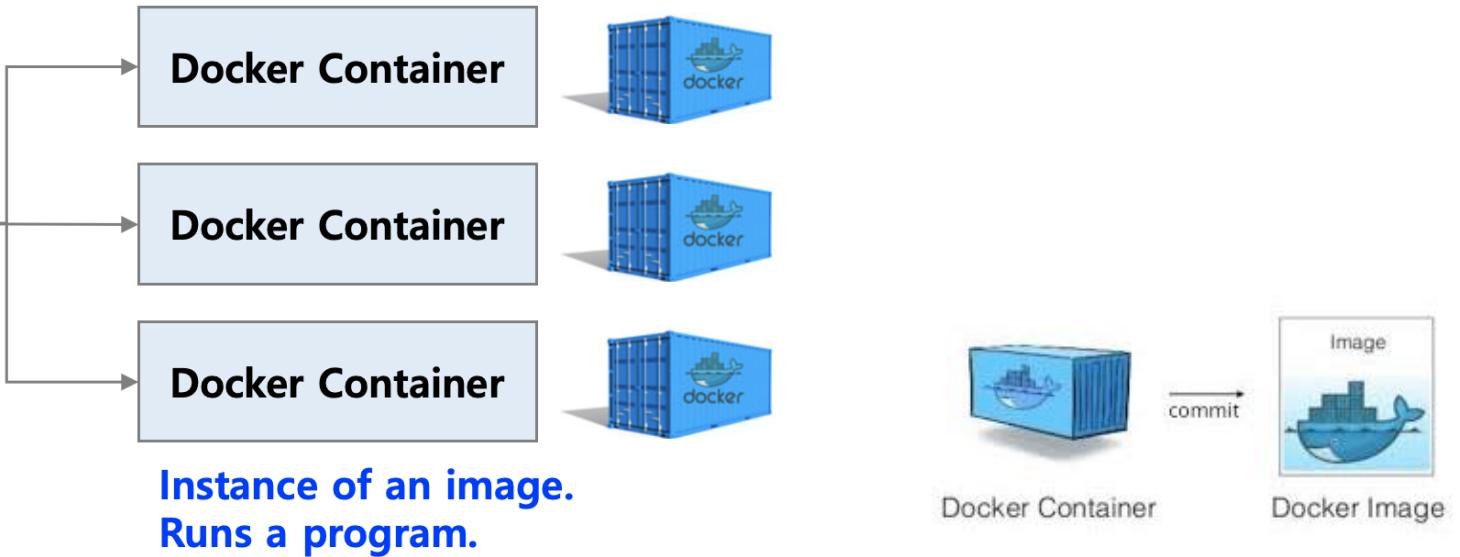
- 도커 이미지 (Docker Image)
 - 서비스 운영에 필요한 서버 프로그램, 소스 코드, 컴파일된 실행 파일을 묶은 형태
 - 상태 값을 가지지 않고 변하지 않는다.
 - 저장소(Registry)에 올리고(push) 받는(pull) 것은 이미지이다
 - 한 이미지로 여러 개의 컨테이너를 만들 수 있다
 - 도커는 베이스 이미지에서 바뀐 부분만 이미지로 생성하기 때문에, 컨테이너로 실행할 때 베이스 이미지와 바뀐 부분을 합쳐서 실행한다.
- 도커 컨테이너 (Docker Container)
 - 도커 이미지를 실행한 상태
 - OS로 치면 이미지는 실행 파일이고 컨테이너는 프로세스
- 도커 저장소 (Docker Registry)
 - 도커 이미지를 저장해놓는 원격 저장소로 저장소이다.
 - 로컬에 image를 만들었다고 자동으로 hub에 올라가는 것이 아니다. 따라서 hub에 올리기 위해서는 push를 해주어야 한다.

도커 이미지 vs. 도커 컨테이너 (Docker Image vs. Docker Container)



Docker Image

Single file with all the dependencies and configuration required to run a program.



도커 이미지 (Docker Image)	도커 컨테이너 (Docker Container)
컨테이너의 설계도 (container blueprint)	도커 이미지 인스턴스 (image instance)
변형 불가 (immutable)	쓰기 가능 (writable)
컨테이너 없이 존재 가능	컨테이너는 이미지를 실행시켜야 존재
컴퓨팅 자원을 필요로 하지 않음	실행을 위해 컴퓨팅 자원 필요
공개/비공개 레지스트리 플랫폼으로 공유	이미 실행 중인 개체는 공유 필요 없음
단 한번만 생성	같은 이미지로 부터 다수의 컨테이너 생성

도커의 구성 요소

- 가장 유명한 저장소인 docker hub (<https://hub.docker.com/>)

The screenshot shows the Docker Hub interface with a search query of "nginx". On the left, there are various filtering options: "Filters" (Images, Extensions, Plugins), "Products" (Docker Official Image, Verified Publisher, Sponsored OSS), "Trusted Content" (Docker Official Image, Verified Publisher, Sponsored OSS), "Operating Systems" (Linux, Windows), and "Architectures" (ARM, ARM 64). The main search results page displays three entries:

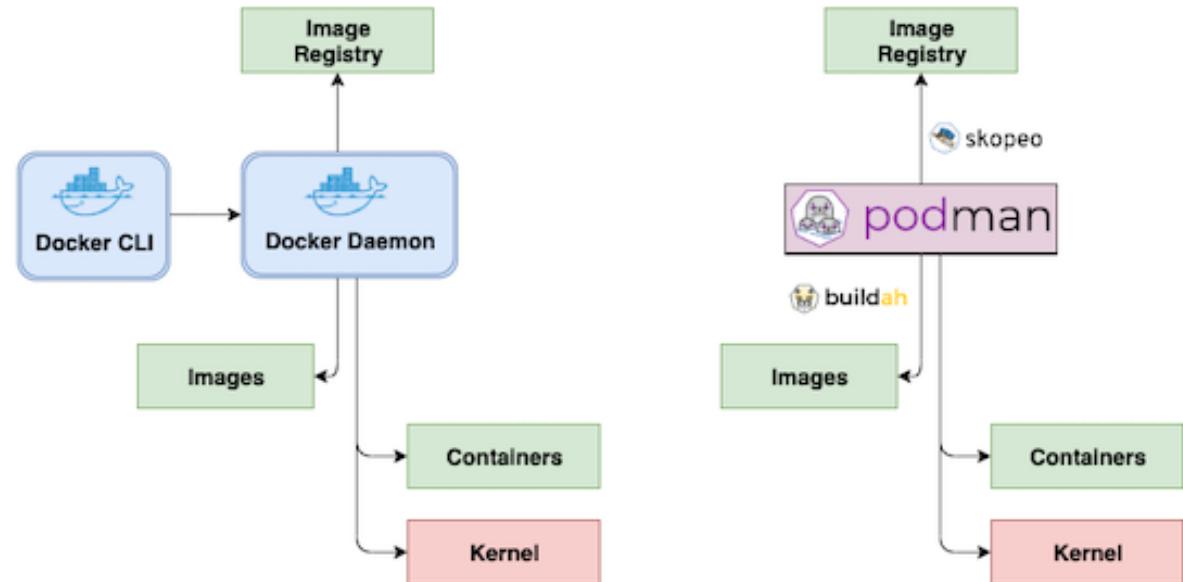
- nginx** (DOCKER OFFICIAL IMAGE) by NGINX: Updated 20 days ago. Description: Official build of Nginx. Tags: Linux, IBM Z, x86-64, ARM, ARM 64, 386, mips64le, PowerPC 64 LE. Downloads: 1B+, Stars: 10K+.
- bitnami/nginx** (VERIFIED PUBLISHER) by VMware: Updated 4 days ago. Description: Bitnami nginx Docker Image. Tags: Linux, x86-64. Downloads: 100M+, Stars: 150.
- bitnami/nginx-ingress-controller** (VERIFIED PUBLISHER) by VMware: Updated 4 days ago. Description: Bitnami Docker Image for NGINX Ingress Controller. Tags: None. Downloads: 10M+, Stars: 23.

docker vs podman

■ 도커의 문제점

- 데몬이 죽으면 모든 컨테이너가 죽음
(현재는 도커의 아키텍처의 변화로 데
몬 재시작시에도 컨테이너가 죽지는
않음)
- 데몬 프로세스는 실행중인 컨테이너에
대한 모든 자식 프로세스를 소유
- 모든 도커는 루트 권한을 가진 사용자
가 수행
- 컨테이너에서 보안 취약점 발생 가능
성 존재

■ Docker vs Podman



podman

- podman 이란?
 - podman 은 Pod Manager Tool 의 약자로 OCI 표준 container와 런타임을 개발, 관리, 실행할 수 있게 해주는 container engine
 - Red Hat Enterprise Linux 8 / CentOS 8 부터 Docker 대신 podman 사용
 - docker와 cli 인터페이스가 매우 유사함
 - podman 명령줄을 docker 명령어로 변환 해주는 podman-docker 설치
- 특징
 - daemon-less
 - 데몬이 필요 없음
 - root 권한 불필요
 - docker 는 docker daemon 에 모든 권한이 집중되다 보니 아무나 docker client 로 docker daemon 을 제어하지 못하도록 root 사용자만 docker client 사용 가능 하였음
 - Kubernetes 지원
 - podman 은 대부분이 명령어가 docker 와 호환되지만 docker가 제공하지 않는 기능도 제공합니다.

```
# dnf install podman-docker
```

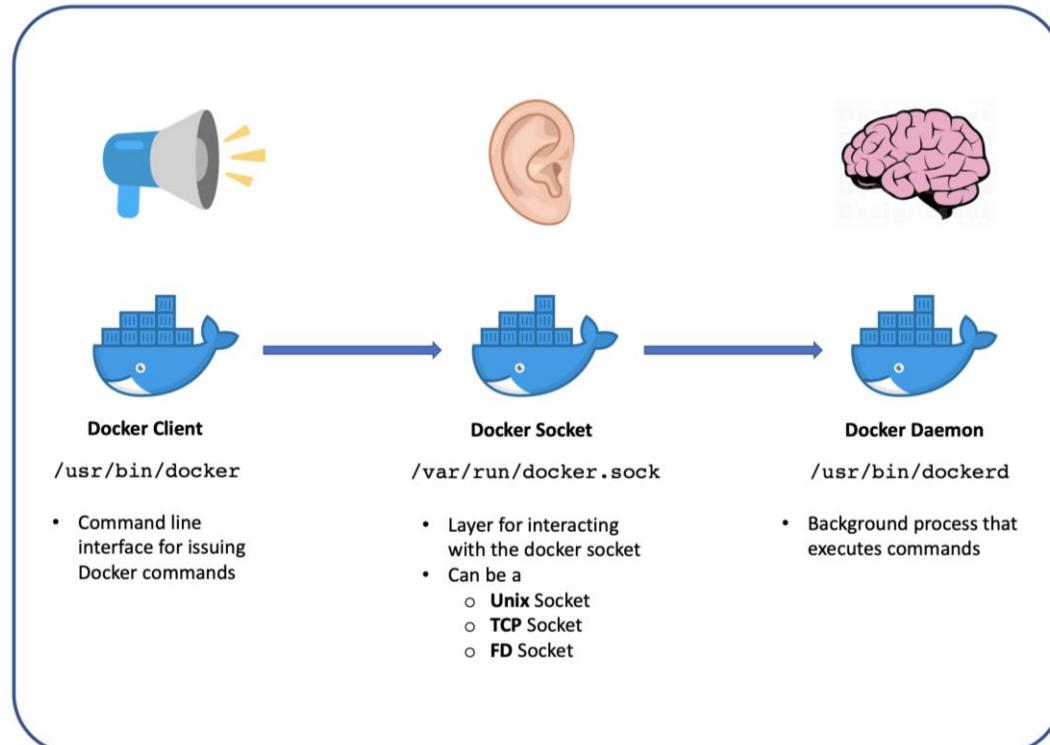
podman-compose 설치

- podman 과 docker는 대부분의 명령어가 호환되지만 docker-compose.yml 을 이용한 명령어는 지원되지 않으므로 별도 프로그램 설치 필요

```
[root@cberi ~]# cd /usr/local/bin  
[root@cberi bin]# pip3 install podman-compose  
[root@cberi bin]# ls -alhF  
total 16K  
drwxr-xr-x.  2 root root 4.0K Dec 31 22:42 ./  
drwxr-xr-x. 12 root root 4.0K Dec 12 05:59 ../  
-rwxr-xr-x.  1 root root  213 Dec 31 22:42 dotenv*  
-rwxr-xr-x.  1 root root  219 Dec 31 22:42 podman-compose*  
[root@cberi bin]# podman-compose --version  
podman-compose version: 1.0.6  
['podman', '--version', '']  
using podman version: 4.6.1  
podman-compose version 1.0.6  
podman --version  
podman version 4.6.1  
exit code: 0
```

podman.sock 설치

- docker.sock에 해당하는 역할
 - 다른 컨테이너들과 상호작용하기 위한 레이어



- 명령줄

```
# systemctl start podman.socket  
# systemctl enable podman.socket
```

- 위 두 명령어를 한번에...

```
# systemctl enable --now podman.socket
```

- 확인

```
# podman --remote info | grep sock  
path: /run/podman/podman.sock
```

- podman.sock 으로 접근 가능

curl

- curl 이란?

- 커맨드라인 데이터 전송 툴
- 업로드 / 다운로드 모두 가능
- HTTP/HTTPS/FTP/LDAP/SCP/TELNET/SFTP/POP3 등 주요한 프로토콜을 지원
- Usage: curl [options...] <url>
- 상세 사용법
 - <https://www.lesstif.com/software-architect/curl-http-get-post-rest-api-14745703.html>

- 예시

- 다운로드 받은 파일을 콘솔로 출력

```
$ curl https://linux.datahub.pe.kr/uc.html
<html>
<body>
    
</body>
```

- 지정한 이름으로 저장

```
$ curl -o index.html \
https://linux.datahub.pe.kr/uc.html
```

- HTTP 인증

- Basic Auth

```
$ curl -v -u userid:password \
http://www.example.com/user.html
```

docker 명령어

■ search

- docker는 docker hub를 통해 GitHub처럼 사용자들 간의 이미지 공유를 할 수 있는 환경이 구축되어 있음
- docker hub로부터 사용가능한 image를 찾는 명령어
- 공식이미지는 galid/centos처럼 /앞에 사용자의 이름이 붙지 아니함

■ pull

- docker hub로 부터 image를 다운받는 명령어

```
[root@cberi ~]# docker pull nginx
? Please select an image:
▶ registry.access.redhat.com/nginx:latest
registry.redhat.io/nginx:latest
docker.io/library/nginx:latest
```

```
[root@cberi ~]# docker pull mariadb
```

```
[root@cberi ~]# docker search mariadb
```

NAME

```
registry.access.redhat.com/rhscl/mariadb-101-rhel7
registry.access.redhat.com/rhscl/mariadb-100-rhel7
registry.access.redhat.com/openshift3/mariadb-app
registry.access.redhat.com/rhscl/mariadb-102-rhel7
registry.access.redhat.com/rhosp12/openstack-mariadb
:
```

DESCRIPTION

```
MariaDB server 10.1 for OpenShift and genera...
MariaDB 10.0 SQL database server
Ansible Playbook Bundle application definiti...
MariaDB is a multi-user, multi-threaded SQL...
Red Hat OpenStack Container image for openst...
```

docker 명령어

■ images

- 현재 host에 다운 받아져 있는 image들을 출력

```
[root@cberi data]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/library/nginx	latest	a99a39d070bf	4 days ago	146 MB
docker.io/hwdsl2/ipsec-vpn-server	latest	dfad1ea83c88	10 days ago	46.1 MB
docker.io/library/mariadb	latest	a748acbaccac	5 weeks ago	416 MB
docker.io/mobtitude/vpn-pptp	latest	4ce95136b93f	4 years ago	158 MB

```
[root@cberi data]#
```

■ rmi

- remove image (도커 이미지 삭제)

```
[root@cberi data]# docker rmi [삭제할 이미지명]
```

- 실행 중이지 않은 이미지만 삭제 가능

docker 명령어

■ run

- docker image로 부터 새로운 컨테이너를 만들어 실행

- 형식

```
docker run [옵션] <이미지이름 or 이미지ID>
```

- 예시

```
docker run -d --name web --privileged \
-p 80:80 \
nginx
```

```
docker run --detach --privileged \
--name mariadb \
--network nginx-proxy \
-p 3306:3306 \
-v /srv/mariadb/data:/var/lib/mysql \
-e MARIADB_ROOT_PASSWORD=pororo23@root \
mariadb:latest
```

옵션	설명
--add-host	사용자 지정 호스트-IP 매핑 추가(host:ip)
--attach , -a	STDIN, STDOUT 또는 STDERR에 연결
--detach , -d	백그라운드에서 컨테이너 실행 및 컨테이너 ID 출력
--detach-keys	컨테이너 분리를 위한 키 시퀀스 재정의
--env , -e	환경 변수 설정
--env-file	환경 변수 파일에서 읽기
--expose	포트 또는 포트 범위 노출
--group-add	가입할 그룹 추가
--hostname , -h	컨테이너 호스트 이름
--mount	파일 시스템 마운트를 컨테이너에 연결
--name	컨테이너에 이름 할당
--net	컨테이너를 네트워크에 연결
--net-alias	컨테이너에 대한 네트워크 범위 별칭 추가
--network	컨테이너를 네트워크에 연결
--network-alias	컨테이너에 대한 네트워크 범위 별칭 추가
--pull	실행하기 전에 이미지 가져오기
--publish , -p	<호스트 포트>:<컨테이너 포트> 호스트에 연결된 컨테이너의 특정 포트를 외부에 노출
--read-only	컨테이너의 루트 파일 시스템을 읽기 전용으로 마운트
--restart	컨테이너 종료 시 적용할 재시작 정책
--rm	컨테이너가 종료되면 자동으로 제거
--volume , -v	<호스트>:<컨테이너> 볼륨 마운트 바인딩
--volumes-from	지정된 컨테이너에서 볼륨 마운트
--workdir , -w	컨테이너 내부의 작업 디렉토리

docker 명령어

■ (표준) run

- 실험

```
# docker run \
--name web --privileged \
-p 80:80 \
nginx
```

- foreground로 컨테이너로 실행
- 컨테이너의 표준 입출력이 콘솔에 연결되므로 더 이상 프롬프트 사용이 불가능하게 되지만 컨테이너의 출력을 모니터링 할 수 있음

■ run --detach

- 실험

```
# docker run --detach \
--name web --privileged \
-p 80:80 \
nginx
```

- background로 컨테이너를 실행하므로 즉시 프롬프트 이용 가능
- 컨테이너의 출력을 확인하려면,

```
# docker logs nginx
```

-p 옵션으로
외부 포트와
매핑하려면,
--privileged
옵션 필요

웹 서버 SW

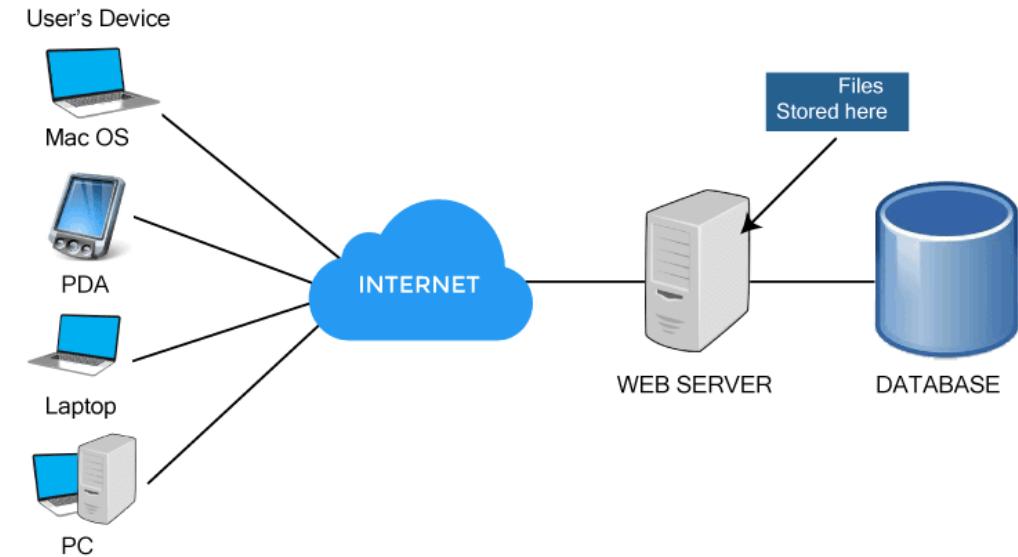
■ 웹 서버

- 웹 서버의 주된 기능은 웹 페이지를 클라이언트로 전달하는 것이다. 주로 그림, CSS, 자바스크립트를 포함한 HTML 문서가 클라이언트로 전달된다.
- 흔히 웹 브라우저 또는 웹 크롤러로 부르는 클라이언트는 HTTP를 통해 리소스를 요청하며 서버는 해당 리소스를 반환하거나 처리할 수 없을 경우 에러 메시지를 전달한다.
- 주된 기능은 콘텐츠를 제공하는 것이지만 클라이언트로부터 콘텐츠를 전달 받기도 한다. 이러한 기능은 파일 업로드를 포함하여 클라이언트에서 제출한 웹 폼을 수신하기 위해 사용된다.

■ 제품



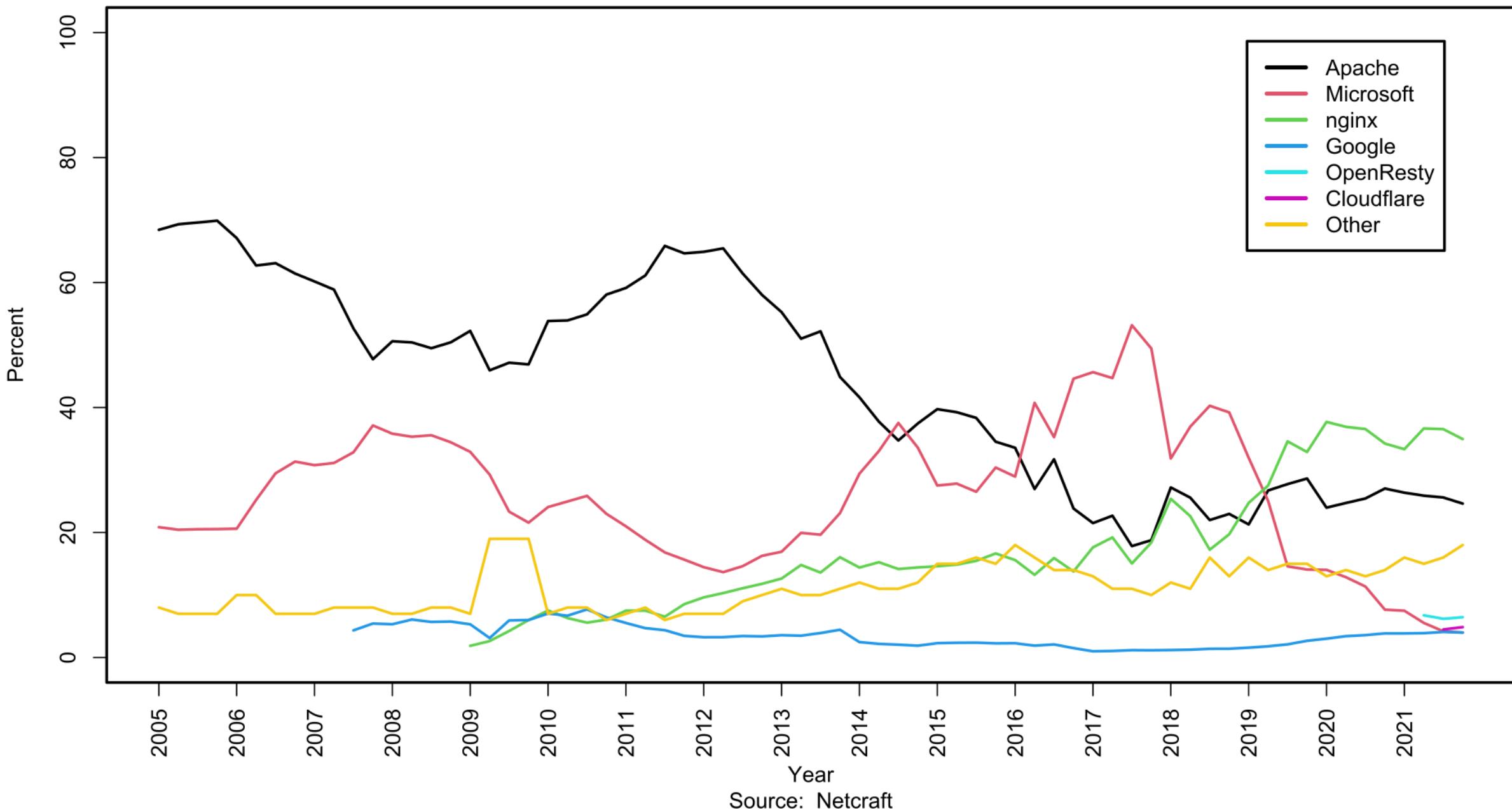
■ 개념도



NGINX vs Apache: Head to Head Comparison

Feature	Apache	NGINX
 Simplicity	Easy to develop and innovate because of its one-connection-per-process model	Complex to develop as it has a sophisticated architecture to handle multiple connections concurrently.
 Performance – Static Content	Slow	2.5 times faster than Apache and consumes less memory
 Performance – Dynamic Content	Excellent	Excellent
 Operating system support	Supports all Unix OS and Windows	Supports all Unix OS and Windows; however, performance on Windows isn't as stable.
 Security	Comparatively the same level of security.	Comparatively the same level of security.
 Flexibility	Can be customized by adding modules. Apache had dynamic module loading for a long time.	NGINX version 1.11.5 and NGINX Plus Release R11 introduced compatibility for dynamic modules.
 Support and Documentation	Excellent support and documentation are available, as it has been in the market for a very long time.	Though there was a weak start for support and documentation for NGINX, it has grown rapidly since.

Usage share of web servers



nginx 웹서버 실행 (docker 활용)

■ 준비

- index.html 생성 (건너뜀)

```
[root@cberi html]# su - www  
  
[www@cberi ~]$ chmod +x .  
  
[www@cberi ~]$ cd html/  
  
[www@cberi html]$ echo "web server worked!"  
web server worked!  
  
[www@cberi html]$ echo "web server worked!" > index.html  
  
[www@cberi html]$ tree ~www  
/home/www  
└── html  
    └── index.html  
└── logs  
  
2 directories, 2 files
```

■ 서버 실행

- Docker 컨테이너 생성

```
# docker run -d --privileged \  
--name web \  
-p 80:80 \  
-v /home/www/html:/usr/share/nginx/html:ro \  
nginx
```

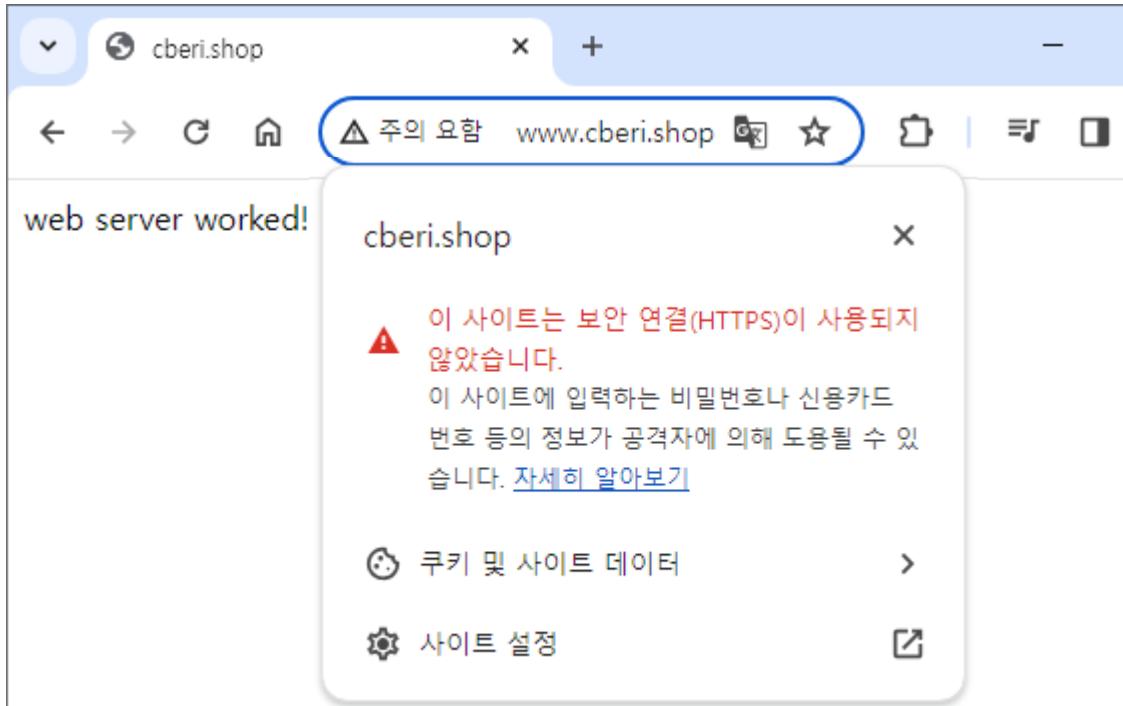
볼륨 마운트
지점은 어떻게
알아낸 것인가?

■ 방화벽 포트 개방(이미함)

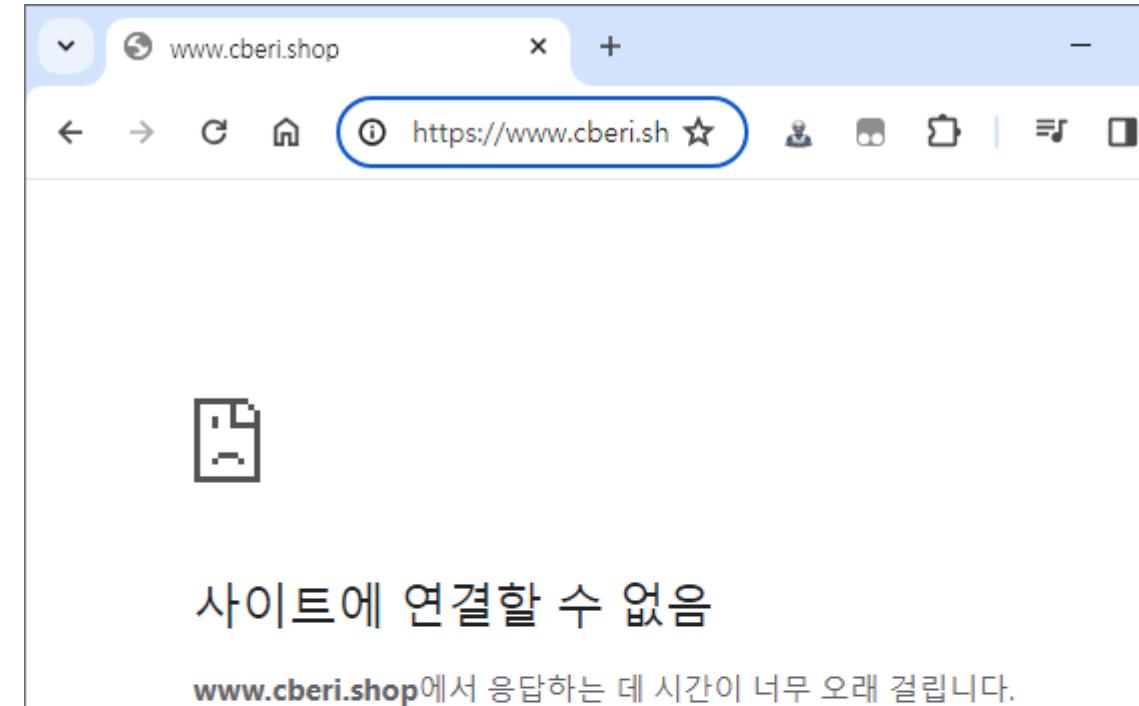
```
# firewall-cmd --permanent --add-service=http  
success  
  
# firewall-cmd --reload  
success
```

nginx 웹 서버 구축

■ http 접속 테스트



■ https 접속 테스트



cloudflare의 프록시 기능 활성화

■ Cloudflare DNS 항목의 proxy 설정 변경

Type	Name	Content	Proxy status	TTL	Actions
A	*	158.247.216.243	DNS only	Auto	Edit▶
A	cberi.art	158.247.216.243	DNS only	Auto	Edit▶
A	sftp	158.247.216.243	DNS only	Auto	Edit▶
A	ssh	158.247.216.243	DNS only	Auto	Edit▶
A	vpn	158.247.216.243	DNS only	Auto	Edit▶
A	www	158.247.216.243	DNS only	Auto	Edit▼

Type Name (required) IPv4 address (required) Proxy status TTL
A www 158.247.216.243 DNS only Auto
Use @ for root

Record Attributes [Documentation](#)
The information provided here will not impact DNS record resolution and is only meant for your reference.

Comment
Enter your comment here (up to 50 characters).

Delete **Cancel** **Save**



Type	Name	Content	Proxy status	TTL	Actions
A	*	158.247.216.243	DNS only	Auto	Edit▶
A	cberi.art	158.247.216.243	DNS only	Auto	Edit▶
A	sftp	158.247.216.243	DNS only	Auto	Edit▶
A	ssh	158.247.216.243	DNS only	Auto	Edit▶
A	vpn	158.247.216.243	DNS only	Auto	Edit▶
A	www	158.247.216.243	DNS only	Auto	Edit▼

Type Name (required) IPv4 address (required) Proxy status TTL
A www 158.247.216.243 Proxied Auto
Use @ for root

Record Attributes [Documentation](#)
The information provided here will not impact DNS record resolution and is only meant for your reference.

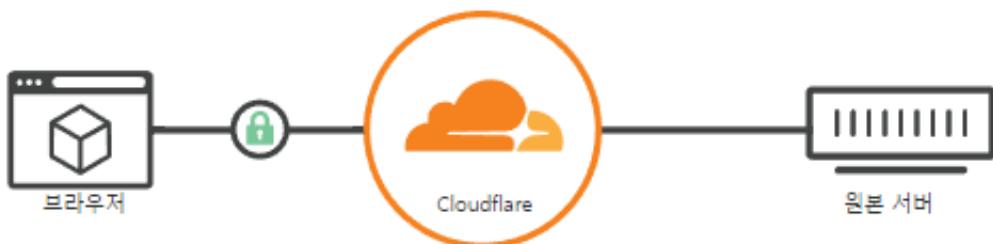
Comment
Enter your comment here (up to 50 characters).

Delete **Cancel** **Save**

cloudflare의 프록시 기능 활성화

SSL/TLS 암호화 모드 가변

이 설정은 16일 전전에 마지막으로 변경되었습니다



- 고기(안전하지 않음) ⓘ
암호화가 적용되지 않음
- 가변
브라우저와 Cloudflare 간 트래픽을 암호화합니다
- 전체
서버의 자체 서명된 인증서를 사용하여 엔드투엔드를 암호화합니다.
- 전체(엄격)
엔드투엔드를 암호화하지만 서버에 신뢰할 수 있는 CA 또는 Cloudflare 원본 CA 인증서가 필요합니다

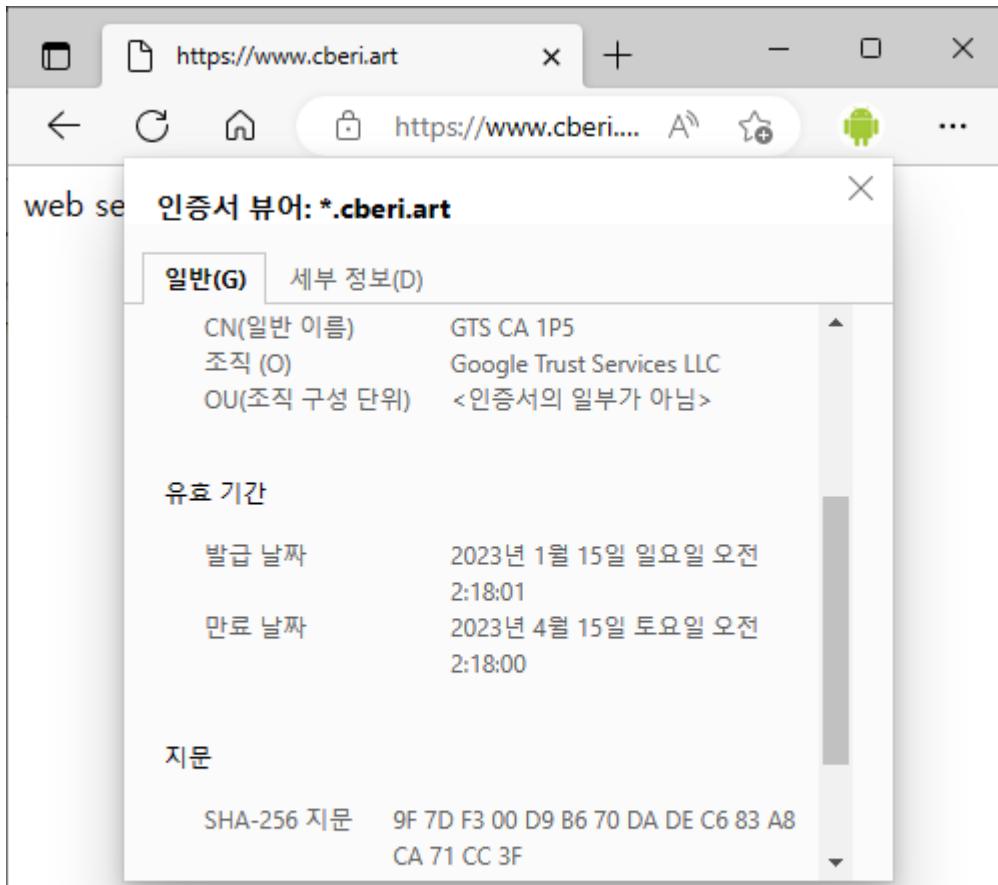
[Cloudflare로 엔드투엔드 암호화에 대해 자세히 알아보십시오.](#)

호스트 이름으로 이러한 설정을 사용자 설정하려면 [구성 규칙을 생성하십시오.](#)

nginx 웹 서버 구축

■ https 접속 다시 테스트

(다른 브라우저로 테스트 해야 함)



■ 고찰

- [cloudflare](#)의 proxy 서비스를 이용하여 SSL 인증서를 설치하지 않고도 보안 서버 운용 가능함을 확인
- 만약 하나의 호스트에서 여러 사이트를 운용해야 한다면 ... ?
 - <http://www.cberi.shop>
 - <http://sqladmin.cberi.shop>
 - <http://word.cberi.shop>

■ 해결책

- 1) 각 사이트별 포트를 다르게 운용
- 2) virtual host 세팅
- 3) reverse proxy 구성

cloudflare 설정

■ 고려할 만한 추가 설정

The screenshot shows the Cloudflare dashboard for the site 'cberi.art'. The left sidebar has a 'SSL/TLS' section selected. The main content area has two sections: '항상 HTTPS 사용' (Always HTTPS) and 'HSTS(HTTP 엄격한 전송 보안)' (HSTS(HTTP Strict Transport Security)).

항상 HTTPS 사용
"http" 구성표가 있는 모든 요청을 "https"로 리디렉션합니다. 이는 영역에 대한 모든 http 요청에 적용됩니다.

HSTS(HTTP 엄격한 전송 보안)
웹사이트에 대한 웹 보안 정책을 강제 적용합니다.

A callout bubble points to the '항상 HTTPS 사용' section, stating: "ON으로 설정하면 cloudflare가 http 접속시도를 자동으로 https로 리다이렉트 한다."

docker 명령어

■ ps

- 실행중인 컨테이너의 목록을 확인
- -a 옵션: 종료된 컨테이너 포함 모든 컨테이너 목록을 확인

```
[root@cberi data]# docker ps -a
CONTAINER ID   IMAGE               COMMAND                  CREATED             NAMES
STATUS          PORTS              COMMAND
3384b258ab4c   docker.io/hwdsl2/ipsec-vpn-server:latest   /opt/src/run.sh   30 hours ago   Up
17 hours ago   0.0.0.0:500->500/udp, 0.0.0.0:4500->4500/udp ipsec-vpn-server
f9a1fe77230a   docker.io/library/mariadb:latest           mariadb      4 seconds ago
Exited (1) 4 seconds ago
mariadb
```

■ logs

- 실행중인/실행되었던 도커 컨테이너의 로그 확인

```
[root@cberi mariadb]# docker logs mariadb
2023-01-15 05:15:19+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.10.2+maria~ubu2204
started.
find: '/var/lib/mysql/': Permission denied
chown: changing ownership of '/var/lib/mysql/': Permission denied
```

docker 명령어

- stop

- 형식

```
docker container stop <컨테이너 이름 or 컨테이너 ID>
```

- 실행중인 도커 컨테이너 중지
 - 실험

```
[root@cberi data]# docker container stop nginx
```

```
[root@cberi data]# docker ps -a
CONTAINER ID  IMAGE
COMMAND      CREATED      STATUS
PORTS          NAMES
e8d8093c8c09  docker.io/library/nginx:latest
nginx -g daemon o...  2 hours ago  Exited (0) 13 seconds ago
0.0.0.0:3306->3306/tcp                                nginx
```

- start

- 형식

```
docker container start <컨테이너 이름 or 컨테이너 ID>
```

- 중지된 도커 컨테이너 시작

- restart

- 형식

```
docker container restart <컨테이너 이름 or 컨테이너 ID>
```

- 컨테이너를 중지한 후 다시 실행
 - 컨테이너가 이미 중지되어 있다면 start와 동일

docker 명령어

- exec

- 형식

```
docker exec <CONTAINER_ID> <COMMAND>
```

- 실행중인 도커 컨테이너 내부에서 명령 실행

```
# docker exec -i www ls -alhF /usr/share/nginx/html
total 12K
drwxrwxr-x. 2 1001 1001 4.0K Dec 27 13:47 .
drwxr-xr-x. 3 root root 4.0K Dec 20 20:13 ..
-rw-rw-r--. 1 1001 1001    85 Dec 31 15:41 index.html
-rw-rw-r--. 1 1001 1001     0 Dec 27 13:47 temp.html
```

- exec 응용

- 컨테이너 내부의 배시셸 사용

```
[root@cberi ~]# docker exec -it web /bin/bash
root@e8d8093c8c09:/# pwd
/
root@e8d8093c8c09:/# hostname
fa9a72133578
root@e8d8093c8c09:/# exit
```

- i, --interactive : 컨테이너 표준 입력을 연다(컨테이너 내부에서 발생하는 일을 사용자에게 보여준다)
 - t, --tty : 단말 디바이스(tty)를 사용한다. (터미널과 비슷한 환경으로 조성해준다.)
 - exit로 탈출

docker 명령어

- cp
 - 호스트와 도커 컨테이너 간 파일 복사
 - 컨테이너 → 호스트
 - 호스트 → 컨테이너
 - 예시
- ```
docker cp [container]:[내부 경로] [host 파일경로]
```
- ```
docker cp [host 파일경로] [container]:[내부 경로]
```
- ```
docker cp -a mariadb:/var/lib/mysql ./
ls -pl
합계 4
drwxr-xr-x. 5 root root 4096 1월 15 14:21 mysql/
```
- 폴더가 통째로 복사됨
- inspect
    - 이미지 또는 컨테이너의 세부 정보 출력

```
[root@cberi www]# docker inspect web
[
 {
 "Id": "00d5c952730b8f42f56f4e7d6794aa233c835978b30e42c6fcbcfdf59af3aeba",
 "Created": "2023-02-01T00:19:22.186061279+09:00",
 "Path": "/docker-entrypoint.sh",
 "Args": [
 "nginx",
 "-g",
 "daemon off;"
],
 "State": {
 "OciVersion": "1.0.2-dev",
 "Status": "running",
 "Running": true,
 "Paused": false,
 "Restarting": false,
```

# docker network

- 개념
  - Docker 컨테이너(container)는 격리된 환경에서 돌아가기 때문에 기본적으로 다른 컨테이너와의 통신이 불가능
  - 여러 개의 컨테이너를 하나의 Docker 네트워크(network)에 연결시키면 서로 통신이 가능해짐
  - ls 명령어로 조회 가능
- 네트워크의 종류
  - bridge 네트워크는 하나의 호스트 컴퓨터 내에서 여러 컨테이너들이 서로 소통할 수 있도록 해줍니다.
  - host 네트워크는 컨테이너를 호스트 컴퓨터와 동일한 네트워크에서 컨테이너를 돌리기 위해서 사용됩니다.
  - overlay: 네트워크는 여러 호스트에 분산되어 돌아가는 컨테이너들 간에 네트워킹을 위해서 사용됩니다.

```
[root@cberi ~]# docker network ls
NETWORK ID NAME DRIVER
2f259bab93aa podman bridge
[root@cberi ~]#
```

# docker network

- create

- 새로운 도커 네트워크 생성

```
[root@cberi ~]# docker network create nginx-proxy
```

```
[root@cberi ~]# docker network ls
NETWORK ID NAME DRIVER
ee1e18cf9293 nginx-proxy bridge
2f259bab93aa podman bridge
```

- -d, --driver 옵션을 사용하지 않으면 기본값인 bridge 네트워크가 생성됨.
- 상세 옵션 설정 예시

```
docker network create --driver bridge \
--gateway 172.19.0.1 --subnet 172.19.0.0/24 \
mybridge
```

- inspect

- 네트워크 상세 정보 조회

```
[root@cberi data]# docker network inspect nginx-proxy
[
 {
 "name": "nginx-proxy",
 "id": "ee1e18cf9293ac52d6a040a4c3ac9e087759228562606a6c8d5e68c51fd963c2",
 "driver": "bridge",
 "network_interface": "cni-podman1",
 "created": "2023-01-15T13:47:19.346909853+09:00",
 "subnets": [
 {
 "subnet": "10.89.0.0/24",
 "gateway": "10.89.0.1"
 }
],
 "ipv6_enabled": false,
 "internal": false,
 "dns_enabled": false,
 "ipam_options": {
 "driver": "host-local"
 }
 }
]
```

# docker network

- connect

- 특정 컨테이너를 도커 네트워크에 연결
- 형식

```
docker network connect <network name> <container name>
```

- 실습

```
[root@cberi ~]# docker network connect nginx-proxy mariadb
Error: container
e8d8093c8c093f8df604665024628652ec82f86a92c5aab5cd405a66b1
30eb09 is already connected to network "nginx-proxy":
network already exists
[root@cberi ~]#
```

- disconnect

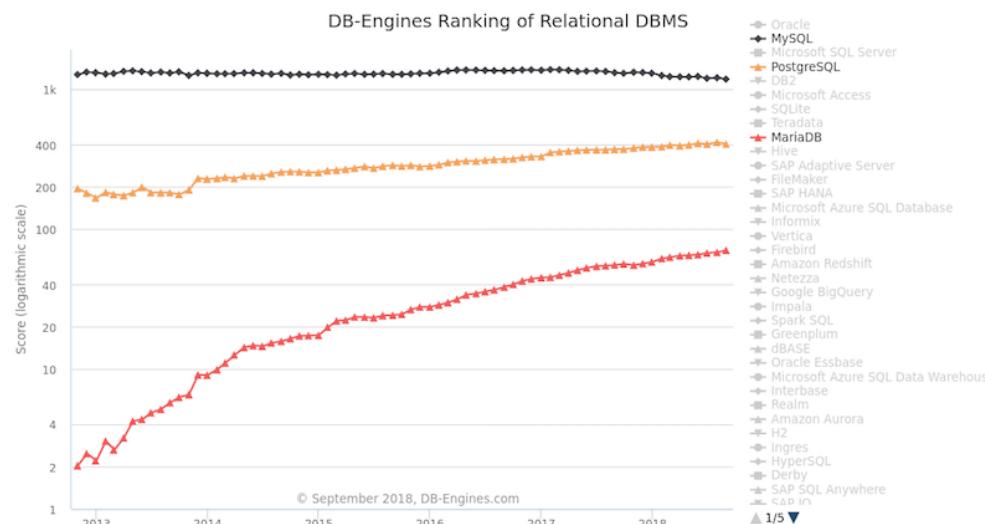
- 특정 컨테이너를 도커 네트워크에서 연결 해제
- 형식

```
docker network disconnect <net name> <container name>
```

# mariadb

## ■ mariadb란?

- 오픈 소스의 관계형 데이터베이스 관리 시스템(RDBMS)이다.
- MySQL과 동일한 소스 코드를 기반으로 하며, GPL v2 라이선스를 따른다.
- 오라클 소유의 현재 불확실한 MySQL의 라이선스 상태에 반발하여 만들어짐



## ■ MySql과 공통점

- MariaDB의 실행 프로그램들과 유ти리티는 모두 MySQL과 이름이 동일하며 호환됩니다.
- MySQL 5.x의 데이터 파일과 테이블 정의 파일(.frm)은 MariaDB 5.x와 호환합니다.
- 모든 클라이언트 API와 통신 프로토콜은 서로 호환합니다.
- MySQL Connector(Java 및 C 클라이언트 라이브러리 등)는 모두 MariaDB에서 변경없이 사용 가능
- MySQL 클라이언트 프로그램은 그대로 MariaDB 서버의 연결에 사용 가능



# MariaDB vs. MySQL

## ■ 스토리지 엔진

| 용도                       | MariaDB                                                                                                        | MySQL                                     |
|--------------------------|----------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| 메모리 스토리지 엔진              | MySQL 코드 베이스에 포함된 MEMORY 스토리지 엔진을 기본으로 사용합니다.                                                                  | MariaDB의 메모리 스토리지 엔진과 거의 동일합니다.           |
| 디스크 기반 임시 테이블 엔진 스토리지 엔진 | 기본적으로 Aria 스토리지 엔진을 사용합니다. 이것은 InnoDB와 비슷하게 인덱스하며 레코드 데이터까지 모두 메모리 캐시를 이용할 수 있기 때문에 MyISAM에 비해서는 빠른 처리를 보장합니다. | MyISAM 스토리지 엔진을 사용하는 테이블을 생성하여 작업을 처리합니다. |
| 트랜잭션 지원 스토리지 엔진          | PerconaServer에서 나온 XtraDB가 사용되었습니다만 10.0.7 버전부터는 InnoDB가 기본이 되었다고 합니다.                                         | InnoDB 스토리지 엔진을 사용합니다.                    |
| NoSQL 지원 엔진              | Cassandra의 데이터를 MariaDB 서버를 통해 접근할 수 있습니다.                                                                     | Memcached 플러그인을 제공합니다.                    |

## ■ 기능

| 용도                | MariaDB                                                                         | MySQL                                       |
|-------------------|---------------------------------------------------------------------------------|---------------------------------------------|
| 스레드 풀             | <a href="#">MariaDB 5.1 버전부터 지원합니다.</a>                                         | 엔터프라이즈 버전에서만 지원합니다.                         |
| 버퍼풀 프리 로드         | XtraDB에서 버퍼 풀의 내용을 덤프하고, 덤프된 버퍼 풀 정보를 MariaDB 재시작 후 다시 버퍼 풀로 로딩할 수 있는 기능을 제공합니다 | MySQL 5.6 버전에서부터 InnoDB 버퍼 풀의 덤프와 로딩 기능을 지원 |
| SSD 지원            | XtraDB에서는 SSD 기반의 디스크 IO를 위한 블록 플러시 알고리즘 지원                                     | -                                           |
| 룰(ROLE) 기반의 권한 관리 | 오라클과 같이 특정 유저 그룹을 위한 룰 생성 가능                                                    | -                                           |
| 반 동기화 레플리케이션      | -                                                                               | 플러그인 형태로 제공                                 |
| 가상 컬럼             | 1개 이상의 컬럼 값을 미리 별도의 컬럼에 저장하거나 쿼리처리 시점에 가공하여 보여 주는 기능                            | -                                           |
| 동적 컬럼             | NoSQL 처럼 사용할 수 있는 동적 컬럼 지원 가능                                                   | -                                           |

# mariadb 설치

## ■ 사용 이미지

[https://hub.docker.com/\\_/mariadb](https://hub.docker.com/_/mariadb)

The screenshot shows the Docker Hub page for the 'mariadb' image. At the top, there's a search bar with 'mariadb' and navigation links for 'Explore', 'Official Images', and 'mariadb'. Below this, the 'mariadb' logo (a red seal) and 'MariaDB Foundation' are displayed. The image card includes the text 'DOCKER OFFICIAL IMAGE', '1B+', '5.2K', and a brief description: 'MariaDB Server is a high performing open source relational database, forked from MySQL.' Below the card, there are 'Overview' and 'Tags' tabs, with 'Overview' being active. A note at the bottom states: 'Note: the description for this image is longer than the Hub length limit of 25000, so has been trimmed. The full description can be found at <https://github.com/docker-library/docs/tree/master/mariadb/README.md>. See [docker/hub-beta-feedback#238](#) for more information.' A 'Quick reference' section lists environment variables:

- Maintained by: MariaDB developer community

## ■ 명령어

```
mkdir /srv/mariadb
cd /srv/mariadb
mkdir data work backup
```

```
docker network create nginx-proxy
docker network ls
NETWORK ID NAME DRIVER
ee1e18cf9293 nginx-proxy bridge
2f259bab93aa podman bridge
```

```
curl -o /srv/mariadb/work/employee_db.sql \
https://linux.datahub.pe.kr/employee_db.sql
```

```
docker run --detach --privileged \
--name mariadb \
--network nginx-proxy \
-p 3306:3306 \
-v /srv/mariadb/data:/var/lib/mysql:z \
-v /srv/mariadb/work:/srv/mariadb/work:z \
-e MARIADB_ROOT_PASSWORD=pororo23@root \
mariadb:latest
```

# mariadb 설치

## ■ mariadb 명령줄 생성

```
[root@cberi ~]# cd /usr/local/bin
[root@cberi bin]# touch mariadb
[root@cberi bin]# chmod +x mariadb
[root@cberi bin]# vim mariadb
```

```
#!/bin/bash
podman exec -it mariadb mariadb $@
```

```
[root@cberi bin]# mariadb
ERROR 1045 (28000): Access denied for user
'root'@'localhost' (using password: NO)
```

## ■ mariadb-dump 명령줄 생성

```
[root@cberi ~]# cd /usr/local/bin
[root@cberi bin]# touch mariadb-dump
[root@cberi bin]# chmod +x mariadb-dump
[root@cberi bin]# vim mariadb
```

```
#!/bin/bash
podman exec -it mariadb mariadb-dump $@
```

```
[root@cberi bin]# mariadb-dump
Usage: mariadb-dump [OPTIONS] database [tables]
OR mariadb-dump [OPTIONS] --databases DB1
OR mariadb-dump [OPTIONS] --all-databases
OR mariadb-dump [OPTIONS] --
system=[SYSTEMOPTIONS]]
For more options, use mariadb-dump --help
```

# mariadb 설치

## ■ mariadb 데이터베이스 접속 테스트

|                                                              |                               |
|--------------------------------------------------------------|-------------------------------|
| [root@cberi ~]# mariadb -uroot -p<br>Enter password:         | 접속 명령<br>암호입력                 |
| MariaDB [(none)]> show databases;                            | 존재하는 모든 데이터베이스 조회             |
| MariaDB [(none)]> create database employee;                  | employee라는 데이터베이스 생성          |
| MariaDB [(none)]> show database;                             | 서버에 존재하는 모든 데이터베이스 조회         |
| MariaDB [(none)]> use employee;                              | 작업 DB를 employee로 변경           |
| Database changed                                             |                               |
| MariaDB [employee]> show tables;                             | 선택된 DB에 존재하는 모든 테이블 조회        |
| MariaDB [employee]> source /srv/mariadb/work/employee_db.sql | 백업파일(employee_db.sql)에서 DB 복원 |
| MariaDB [employee]> show tables;                             | 작업 DB안의 모든 테이블 조회             |
| MariaDB [employee]> exit;                                    | 나가기                           |
| Bye                                                          |                               |

# mariadb 백업/복원

## ■ DB 백업하기

### ■ 명령줄 형식

```
mariadb-dump -u [사용자 계정] -p [복원할 DB명] < [백업된 DB파일]
mariadb-dump --all-databases -u [사용자 계정] -p < [백업된 DB파일].sql
```

### ■ 예시

```
mariadb-dump -uroot -ppororo23@root employee > employee.sql
mariadb-dump --all-databases -uroot -ppororo23@root > all_db.sql
```

## ■ DB 복원

### ■ 명령줄 형식

```
mariadb -u [사용자 계정] -p [복원할 DB명] < [백업된 DB파일]
mariadb --all-databases -u [사용자 계정] -p < [백업된 DB파일].sql
```

### ■ 예시

```
mariadb -uroot -ppororo23@root employees < employee.sql
mariadb --all-databases -uroot -ppororo23@root < all_db.sql
```

# anacron

## ■ 특징

- crontab 과 달리 system demon이 아님
- 정해진 시간을 주기로 작업을 수행하는 것이 아니라, 해당 작업이 정해진 시간 내에 실행된 적이 있는지를 점검한 후 만약 정해진 시간내에 수행한 적이 없다면 해당 작업을 여유를 두고 실행
- 작업의 실행 간격을 1일 단위로 지정
- (1일 1회 이상 실행빈도는 설정 불가, 명시적인 실행 시각 지정 불가)
- 작업의 설정 장소는 /etc/anacrontab만 가능하며, 루트 유저 이외는 작업의 설정이 불가
- 작업의 실행 타이밍에 랜덤으로 연장 시간이 추가 됨
- 데몬 프로세스가 존재하지 않으므로 정기적으로 anacron 명령어를 실행하는 구성을 따로 준비.
- (cron의 매시 01분에 anacron을 실행하는 명령어가 기본으로 추가되어 있음)
- 장기간 서버가 정지된 경우는 서버 가동 후에 필요한 작업을 바로 할 수 있다.

# anacron

## ■ 설정 예시

```
vim /etc/anacrontab

/etc/anacrontab: configuration file for anacron
See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
the jobs will be started during the following hours only
START_HOURS_RANGE=3-22

#period in days delay in minutes job-identifier command
1 5 cron.daily nice run-parts /etc/cron.daily
7 25 cron.weekly nice run-parts /etc/cron.weekly
@monthly 45 cron.monthly nice run-parts /etc/cron.monthly
```

- 1라인 : 최근 1일동안 실행된 적이 있는지 점검하고, 만약 그렇지 않다면 시스템 부팅 5분 후에 수행
- 2라인 : 최근 7일동안 실행된 적이 있는지 점검하고, 만약 그렇지 않다면 시스템 부팅 25분 후에 수행
- 지연 시간을 두는 이유는 한꺼번에 실행할 경우 시스템에 과부하가 걸릴 수 있으므로 시간 간격을 두어 부하를 분산하기 위함

# date 명령어

## ■ 도움말

```
[root@cberi ~]# date --help
Usage: date [OPTION]... [+FORMAT]
 or: date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
Display the current time in the given FORMAT, or set the system date.
```

Mandatory arguments to long options are mandatory for short options too.  
-d, **--date**=STRING display time described by STRING, not 'now'

FORMAT controls the output. Interpreted sequences are:

|           |                                                     |
|-----------|-----------------------------------------------------|
| <b>%d</b> | day of month (e.g., 01)                             |
| <b>%D</b> | date; same as %m/%d/%y                              |
| <b>%F</b> | full date; same as %Y-%m-%d                         |
| <b>%g</b> | last two digits of year of ISO week number (see %G) |
| <b>%H</b> | hour (00..23)                                       |
| <b>%I</b> | hour (01..12)                                       |
| <b>%m</b> | month (01..12)                                      |
| <b>%M</b> | minute (00..59)                                     |
| <b>%s</b> | seconds since 1970-01-01 00:00:00 UTC               |
| <b>%S</b> | second (00..60)                                     |
| <b>%T</b> | time; same as %H:%M:%S                              |
| <b>%w</b> | day of week (0..6); 0 is Sunday                     |
| <b>%y</b> | last two digits of year (00..99)                    |
| <b>%Y</b> | year                                                |

## ■ 테스트

```
date
Mon Jan 1 20:28:44 KST 2024

date +'%Y%m%d'
20240101

date --date '10 days'
Thu Jan 11 20:28:55 KST 2024

date --date '10 days ago'
Fri Dec 22 20:29:03 KST 2023

date --date '10 days ago' +'%Y%m%d'
20231222
```

# 셸 스크립트

## ■ 셸 스크립트의 기본

```
#!/bin/bash

echo "root directory"
cd /
ls -lhF # 조회
```

공백은 무시됨.

# 문자로 주석처리

```
#!/bin/bash

echo "root dir";cd /;ls -lhF
```

한줄에 여러명령 가능

```
#!/bin/bash
appdir=/home/www/html
echo $appdir

filename=employee
echo ${filename}_db.sql
echo ${filename}_db.sql
```

변수 선언(띄어쓰기금지)  
변수 사용

{ }의 사용법에 주목

## ■ 쿼팅

```
#!/bin/bash
country=Korea
echo 'I came from $country'
echo "I came from $country"
```

이 둘의 차이점은?

```
[root@cberi ~]# ./test.sh
I came from $country
I came from Korea
```

- 작은따옴표 안의 \$는 특별한 의미를 가지지 않는 일반문자
- 큰따옴표 안에서는 \$로 시작하는 변수가 값으로 치환됨

```
[root@cberi ~]# ./test.sh
/home/www/html
.sql
employee_db.sql
```

# 셸 스크립트

## ▪ 명령어 치환

- 명령어의 출력 결과를 이용하고 싶을 때 이 기능을 사용
- \$( ) 와 같은 형식으로 괄호 안에 실행하려는 명령어를 작성

```
#!/bin/bash
filename=$(date '+%Y-%m-%d')
touch $filename

echo "Today is $(date '+%Y-%m-%d')."
```

## ▪ 위치 파라미터

```
$./params.sh aaa bbb ccc
```

```
$0 $1 $2 $3
```

```
[root@cberi ~]# vi params.sh
```

```
#!/bin/bash
echo "\$0 = $0"
echo "\$1 = $1"
echo "\$2 = $2"
echo "\$3 = $3"
echo "\$@ = $@"
```

```
[root@cberi ~]# chmod +x params.sh
```

```
[root@cberi ~]# ./params.sh a b c d
```

```
$0 = ./params.sh
$1 = a
$2 = b
$3 = c
$@ = a b c d
```

# mariadb 전체 DB 백업 스크립트

## ■ 쉘 스크립트 작성

```
[root@cberi ~]# cd /srv/mariadb/backup/
[root@cberi backup]# touch mariadb_backup.sh
[root@cberi backup]# chmod +x mariadb_backup.sh
[root@cberi backup]# vim mariadb_backup.sh

#!/bin/sh

DATE=$(date '+%Y%m%d')
PREV_DATE=$(date --date '10 days ago' '+%Y%m%d')

backup_file=all_db_${DATE}.sql
delete_file=all_db_${PREV_DATE}.sql

echo "backup file: $backup_file"
mariadb-dump \
 --all-databases --skip-lock-tables \
 -uroot -ppororo23@root > /srv/mariadb/backup/${backup_file}

echo "delete file: $delete_file"
rm -f /srv/mariadb/backup/${delete_file}
```

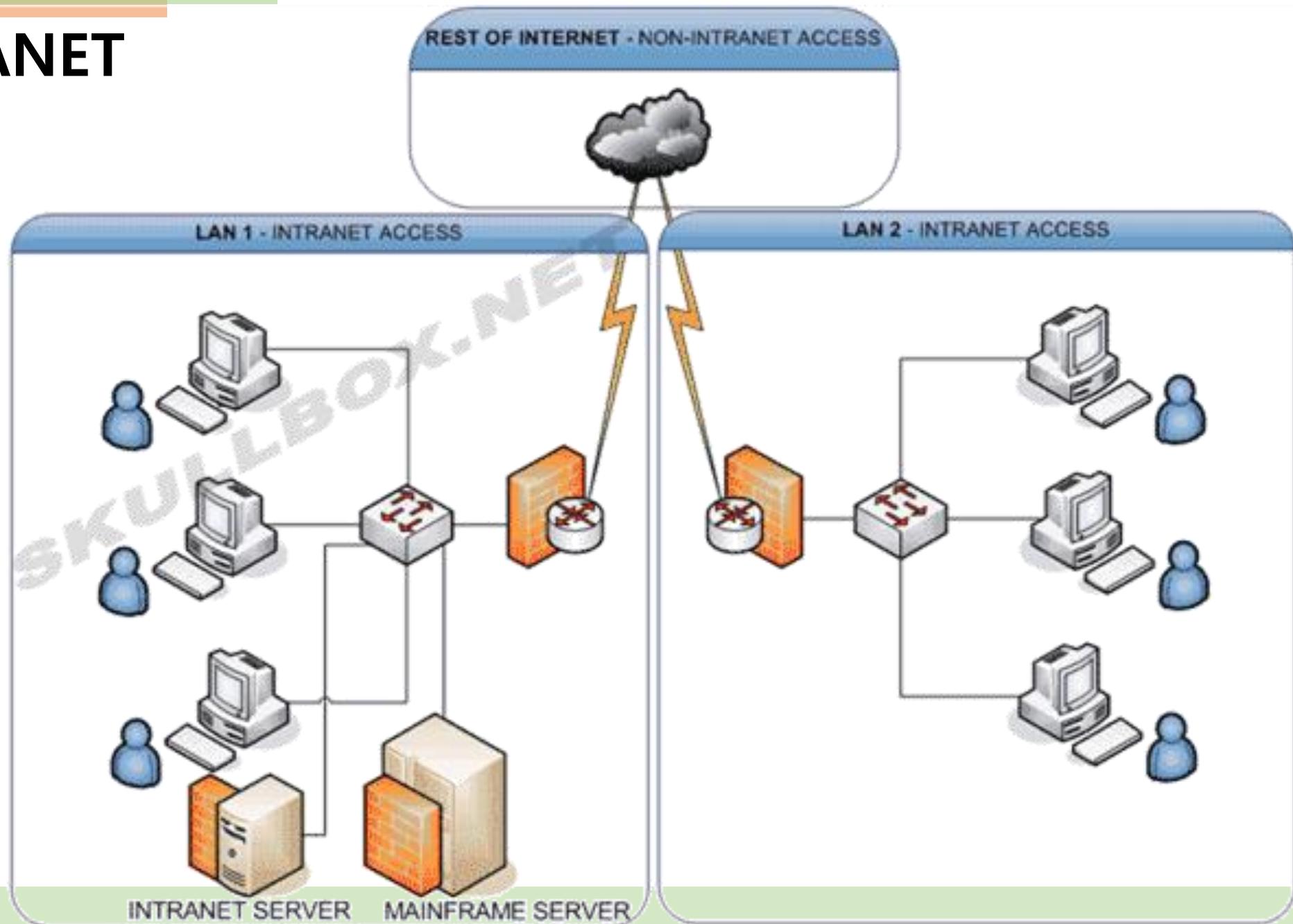
## ■ 실행 테스트

```
[root@cberi backup]# ./mariadb_backup.sh
backup file: all_db_20240101.sql
delete file: all_db_20231222.sql
```

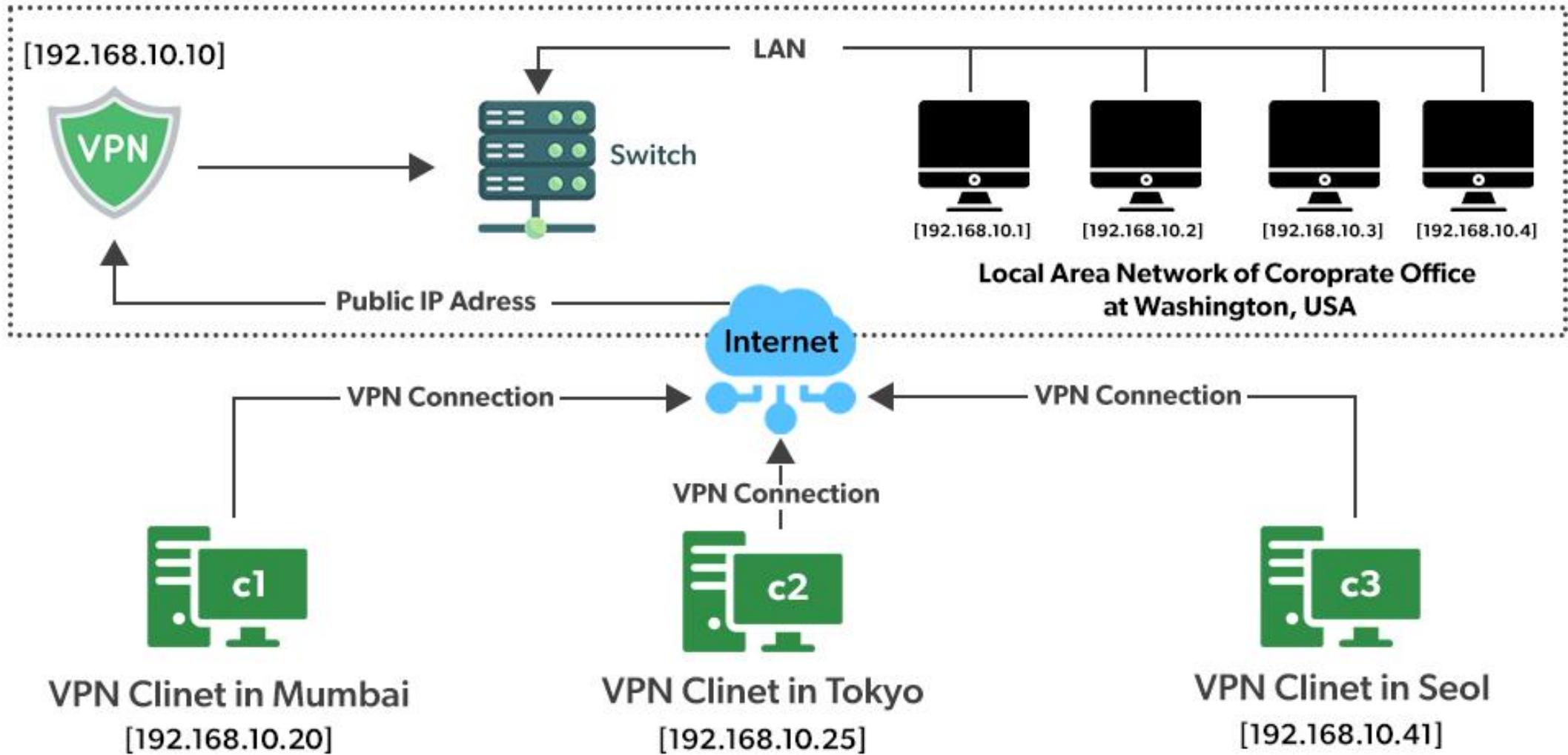
## ■ anacron에 등록

```
mv mariadb_backup.sh /etc/cron.daily/
```

# INTRANET



# VPN (Virtual Private Network)



# L2TP/IPSec VPN Server 접속 전 확인

## ■ IP주소 확인

The screenshot shows a mobile browser interface with the URL [m.search.naver.com/search....](https://m.search.naver.com/search....). The search query "ip주소확인" is entered. The results page has a green header bar with the Naver logo and a microphone icon. Below the header, there are tabs: 통합 (Unified), VIEW, 이미지 (Image), 지식iN (Knowledge), 인플루언서 (Influencer), 동영 (Video), and three dots. The main content area displays search results for "연관" (Related) terms: "내 ip 주소 확인, 컴퓨터 ip 주소 확인, ip 주소 확인 방법, ip주소확인 네이버, 내 아이피 주소 확인, 아이피 주소 확인, ip 위치". Below this, a section titled "이 기기의 IP주소 조회결과" shows the IP address "61.108.19.222". At the bottom, there is a link "관련 IP주소란?" and a button "공용 IP 주소 찾기".

## ■ 차단된 사이트 예시

The screenshot shows a mobile browser interface with the URL [manamo21.com](https://manamo21.com). The browser bar includes icons for back, forward, search, and a red "정진" (Block) button. The main content area displays a large blue circular icon with a white question mark. Below the icon, the text "사이트에 연결할 수 없음" is displayed. Further down, there is a message: "manamo21.com에 오타가 있는지 확인하세요." and "철자가 올바르다면 Windows 네트워크 진단을 실행해 보세요." At the bottom, the error code "DNS\_PROBE\_FINISHED\_NXDOMAIN" is shown, and a blue "새로고침" (Refresh) button is visible.

# kernel-modules-extra 설치

- 설치 이유 (<https://access.redhat.com/articles/3760101>)

To enhance Red Hat Enterprise Linux against possible future security vulnerabilities in lesser-known components which system administrators typically do not protect against, a set of kernel modules have been moved to the kernel-modules-extra package so they are not installed by default, and blacklisted by default so those components cannot be loaded by non-root users.

```
[root@cberi ~]# dnf install -y kernel-modules-extra
```

설치되었습니다:

```
kernel-modules-extra-4.18.0-408.el8.x86_64
```

완료되었습니다!

```
[root@cberi ~]# vi /etc/modprobe.d/l2tp_netlink-blacklist.conf
```

```
Remove the blacklist by adding a comment # at the start of the line.
#blacklist l2tp_netlink
```

```
[root@cberi ~]# vi /etc/modprobe.d/l2tp_ppp-blacklist.conf
```

```
Remove the blacklist by adding a comment # at the start of the line.
#blacklist l2tp_ppp
```

# L2TP/IPSec VPN Server 셋팅

## ■ 준비

```
[root@cberi srv]# cd /srv
[root@cberi srv]# mkdir ipsec-vpn
[root@cberi srv]# cd ipsec-vpn/
[root@cberi ipsec-vpn]# vim vpn.env

VPN_IPSEC_PSK=pororo@vpn
VPN_USER=vpnuser
VPN_PASSWORD=vpnpass

VPN_ADDL_USERS=vpnuser1 vpnuser2 vpnuser3
VPN_ADDL_PASSWORDS=vpnpass1 vpnpass2 vpnpass3
```

## ■ L2TP(Layer 2 Tunneling Protocol)

## ■ 도커 이미지 실행

```
docker run \
 --detach --privileged \
 --name ipsec-vpn-server \
 --env-file /srv/ipsec-vpn/vpn.env \
 -v /lib/modules:/lib/modules:ro \
 -p 500:500/udp \
 -p 4500:4500/udp \
 hwds12/ipsec-vpn-server
✓ docker.io/hwds12/ipsec-vpn-server:latest
Trying to pull docker.io/hwds12/ipsec-vpn-
server:latest...
```

## ■ IKEv1(Internet Key Exchange v2) 를 사 용하지 않는다면 여기서 끝

# L2TP/IPSec VPN Server 셋팅

## ■ 컨테이너 실행 상태 확인

```
[root@cberi ~]# docker ps -a
```

| CONTAINER ID                                 | IMAGE                                    | COMMAND          | CREATED   | STATUS       |
|----------------------------------------------|------------------------------------------|------------------|-----------|--------------|
| POR                                          | TS                                       | NAMES            |           |              |
| 3384b258ab4c                                 | docker.io/hwdsl2/ipsec-vpn-server:latest | /opt/src/run.sh  | 1 min ago | Up 1 min ago |
| 0.0.0.0:500->500/udp, 0.0.0.0:4500->4500/udp |                                          | ipsec-vpn-server |           |              |

```
[root@cberi ~]#
```

## ■ 방화벽 개방 (이미 열었음)

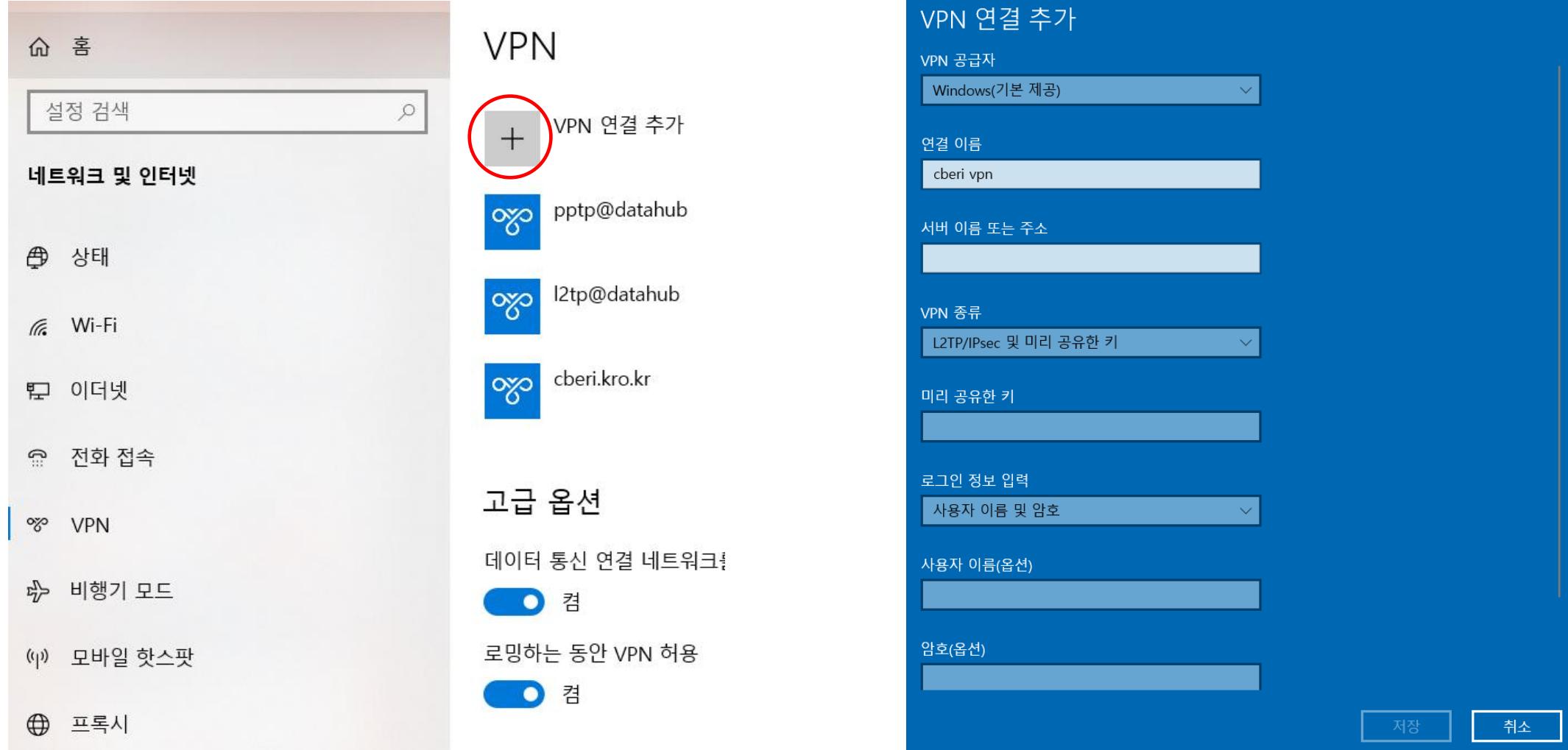
```
[root@cberi ipsec.d]# firewall-cmd --permanent --zone=public --add-port=500/udp
success
```

```
[root@cberi ipsec.d]# firewall-cmd --permanent --zone=public --add-port=4500/udp
success
```

```
[root@cberi ipsec.d]# firewall-cmd --reload
success
```

# L2TP/IPSec VPN Server 접속

- Windows 10 L2TP/IPSec PreSharedKey(미리 공유한 키)를 이용한 접속



# L2TP/IPSec VPN Server 접속

## ■ Windows error 809 에러 대응

- <https://github.com/hwdsl2/setup-ipsec-vpn/blob/master/docs/clients.md#windows-error-809>

Error 809: The network connection between your computer and the VPN server could not be established because the remote server is not responding. This could be because one of the network devices (e.g. firewalls, NAT, routers, etc) between your computer and the remote server is not configured to allow VPN connections. Please contact your Administrator or your service provider to determine which device may be causing the problem.

**Note:** The registry change below is only required if you use IPsec/L2TP mode to connect to the VPN. It is NOT required for the IKEv2 and IPsec/XAuth modes.

To fix this error, a [one-time registry change](#) is required because the VPN server and/or client is behind NAT (e.g. home router). Download and import the [.reg](#) file below, or run the following from an [elevated command prompt](#). **You must reboot your PC when finished.**

- For Windows Vista, 7, 8, 10 and 11 ([download .reg file](#))

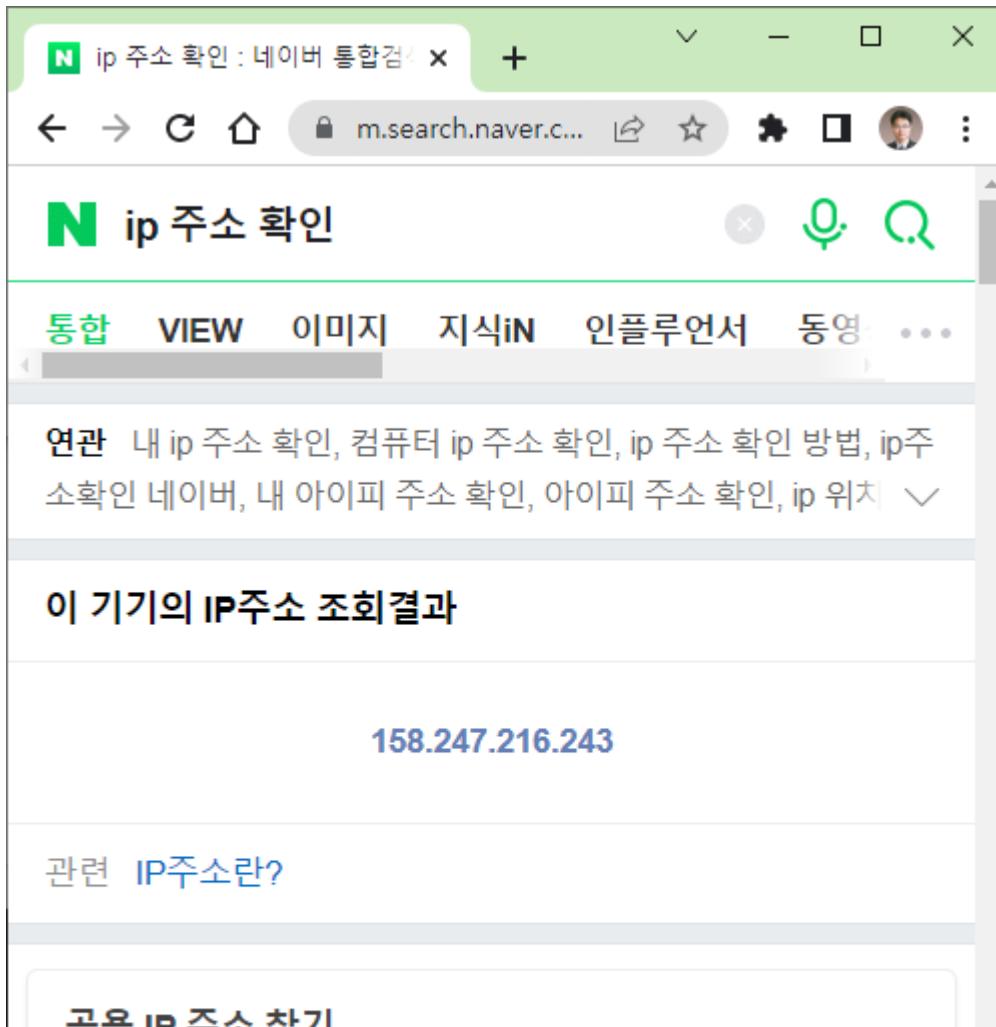
```
REG ADD HKLM\SYSTEM\CurrentControlSet\Services\PolicyAgent /v AssumeUDPEncapsulationContextOnSendRule /t REG_DWORD /d 0x2 /f
```

## ■ 또는 수작업

- 레지스트리에서 HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\PolicyAgent로 이동.
- 32비트 DWORD를 생성하고 이름을 AssumeUDPEncapsulationContextOnSendRule로 변경.
- 값을 2로 설정하고 재부팅

# L2TP/IPSec VPN Server 접속 후 확인

- IP주소 확인 (서버 주소로 변경됨)



- 고찰

- VPN은 Squid 프록시와는 달리 모든 연결에 IP익명성을 제공한다.
- 추적자는 나의 IP를 바로 알아낼 수 없다.
- 또한 모든 통신은 암호화 되어 전송된다.
- VPN의 용도가 익명성 제공인가?
- VPN의 원래 용도는 무엇인가?

# IKEv2/IPSec VPN 을 위한 추가 셋팅

## ■ IKEv2 인증서 생성 스크립트 실행

```
[root@cberi ~]# docker exec -it ipsec-vpn-server ikev2.sh
Emulate Docker CLI using podman. Create /etc/containers/nodocker
to quiet msg.
```

IKEv2 Script Copyright (c) 2020-2023 Lin Song 13 Dec 2023

Welcome! Use this script to set up IKEv2 on your VPN server.

I need to ask you a few questions before starting setup.  
You can use the default options and just press enter if you are  
OK with them.

Do you want IKEv2 clients to connect to this server using a DNS  
name,

e.g. vpn.example.com, instead of its IP address? [y/N] **y**

Enter the DNS name of this VPN server: **vpn.cberi.shop**

Provide a name for the IKEv2 client.  
Use one word only, no special characters except '-' and '\_'.  
Client name: [vpnclient] **vpnclient**

Specify the validity period (in months) for this client

```
Do you want to continue? [Y/n] y
=====
```

IKEv2 setup successful. Details for IKEv2 mode:

VPN server address: **vpn.cberi.shop**

VPN client name: **vpnclient**

Client configuration is available inside the  
Docker container at:

**/etc/ipsec.d/vpnclient.p12** (for Windows & Linux)

**/etc/ipsec.d/vpnclient.swan** (for Android)

**/etc/ipsec.d/vpnclient.mobileconfig** (for iOS & macOS)

Next steps: Configure IKEv2 clients. See:  
<https://vpnsetup.net/clients2>

```
=====
```

# IKEv2/IPSec VPN 을 위한 추가 셋팅

## ■ IKEv2 사용을 위한 인증서 복사

```
[root@cberi ~]# docker exec -it ipsec-vpn-server ls -l /etc/ipsec.d
total 152
-rw----- 1 root root 36864 Jan 13 16:03 cert9.db
-rw-r--r-- 1 root root 621 Jan 13 16:03 ikev2.conf
-rw----- 1 root root 877 Jan 13 16:03 ikev2setup.log
-rw----- 1 root root 61440 Jan 13 16:03 key4.db
-rw----- 1 root root 53 Jan 13 23:51 passwd
-rw----- 1 root root 423 Jan 13 16:02 pkcs11.txt
drwx----- 2 root root 4096 Nov 12 14:48 policies
-rw----- 1 root root 12546 Jan 13 16:03 vpnclient.mobileconfig
-rw----- 1 root root 4344 Jan 13 16:03 vpnclient.p12
-rw----- 1 root root 6313 Jan 13 16:03 vpnclient.sswan
```

```
[root@cberi ~]# docker cp ipsec-vpn-server:/etc/ipsec.d /etc/
```

# IKEv2 VPN 을 위한 추가 셋팅

## ■ 도커 재실행

### ■ IKEv2 설정 추가

```
docker run \
--detach --privileged \
--name ipsec-vpn-server \
--env-file /srv/ipsec-vpn/vpn.env \
-v ikev2-vpn-data:/etc/ipsec.d \
-v /lib/modules:/lib/modules:ro \
-p 500:500/udp \
-p 4500:4500/udp \
hwsl2/ipsec-vpn-server
✓ docker.io/hwsl2/ipsec-vpn-server:latest
Trying to pull docker.io/hwsl2/ipsec-vpn-
server:latest...
```

## ■ 인증서 복사

```
[root@cberi ~]# cp -Ra /etc/ipsec.d/ ~/
```

## ■ 인증서 다운로드

### ■ sftp로 접속하여 인증서 다운로드

| 파일명                    | 크기     | 파일 유형         | 최종 수정          | 권한         |
|------------------------|--------|---------------|----------------|------------|
| ..                     |        |               |                |            |
| policies               | 36,864 | 파일 풀더         | 2023-12-01 ... | drwx-----  |
| cert9.db               | 656    | Data Base ... | 2024-01-13 ... | -rw-----   |
| ikev2.conf             | 61,440 | CONF 파일       | 2024-01-13 ... | -rw-r--r-- |
| key4.db                | 53     | Data Base ... | 2024-01-13 ... | -rw-----   |
| passwd                 | 423    | 파일            | 2024-01-13 ... | -rw-----   |
| pkcs11.txt             | 12,546 | 텍스트 문서        | 2024-01-13 ... | -rw-----   |
| vpnclient.mobileconfig | 4,344  | MOBILECO...   | 2024-01-13 ... | -rw-----   |
| vpnclient.p12          | 6,311  | 개인 정보 ...     | 2024-01-13 ... | -rw-----   |
| vpnclient.sswan        |        | SSWAN 파일      | 2024-01-13 ... | -rw-----   |

# IKEv2 VPN 을 위한 추가 셋팅

## ■ 도커 로그 확인

```
[root@cberi ~]# docker logs ipsec-vpn-server
Trying to auto discover IP of this server...
Starting IPsec service...
=====
IPsec VPN server is now ready for use!
```

Connect to your new VPN with these details:

Server IP: 207.148.95.144

IPsec PSK: pororo@vpn

Username: vpnuser

Password: vpnpass

Write these down. You'll need them to connect!

VPN client setup: <https://vpnsetup.net/clients2>

```
=====
IKEv2 is already set up. Details for IKEv2 mode:
```

VPN server address: 207.148.95.144

VPN client name: vpnclient

Client configuration is available inside the Docker container at:

</etc/ipsec.d/vpnclient.p12> (for Windows & Linux)

</etc/ipsec.d/vpnclient.swan> (for Android)

</etc/ipsec.d/vpnclient.mobileconfig> (for iOS & macOS)

Next steps: Configure IKEv2 clients. See:  
<https://vpnsetup.net/clients2>

# IKEv2 프로토콜로 VPN Server 접속

- Configure your devices to use the IKEv2 VPN.
  - <https://github.com/hwdsl2/setup-ipsec-vpn/blob/master/docs/ikev2-howto.md>

## Windows 7, 8, 10 and 11

### Auto-import configuration

[Screencast: IKEv2 Auto Import Configuration on Windows](#)

Windows 8, 10 and 11 users can automatically import IKEv2 configuration:

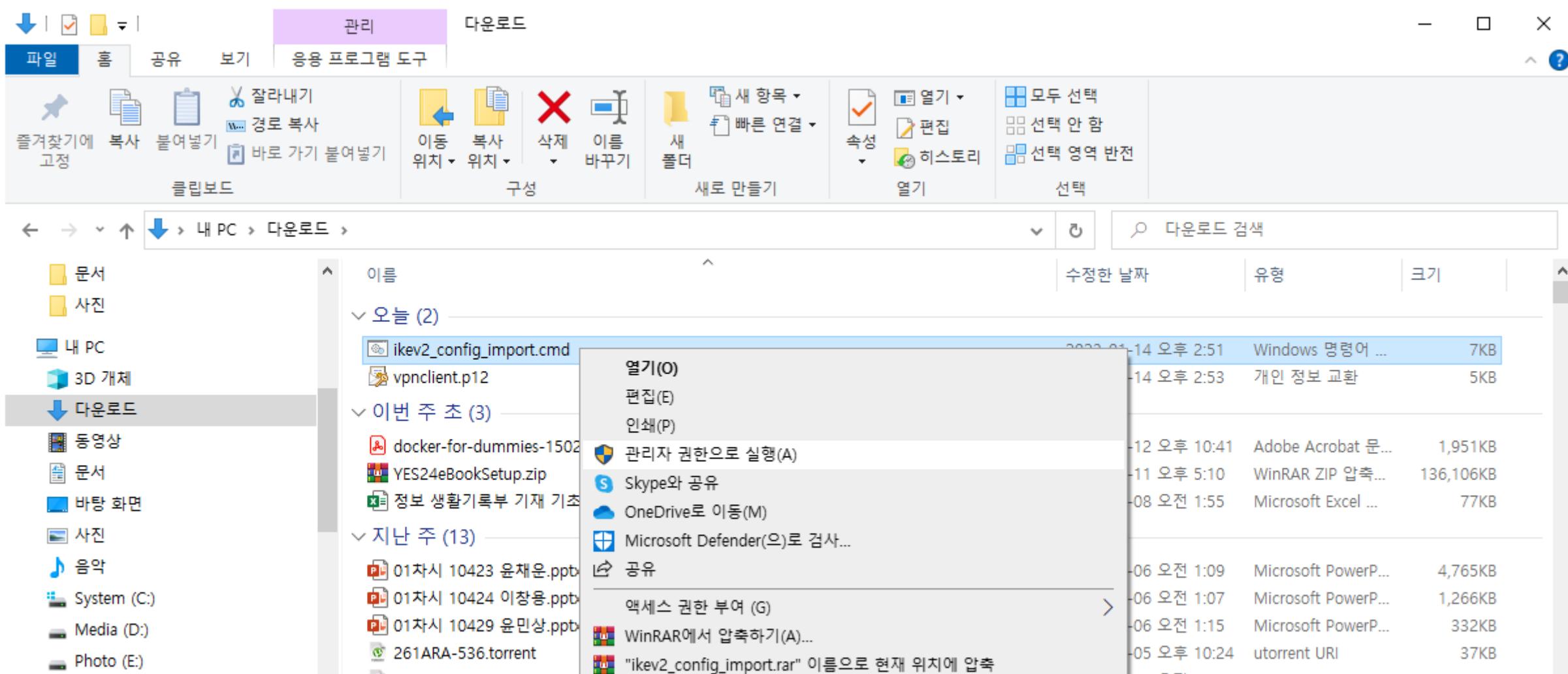
- Securely transfer the generated `.p12` file to your computer.
- Right-click on `ikev2_config_import.cmd` and save this helper script to the **same folder** as the `.p12` file.
- Right-click on the saved script, select **Properties**. Click on **Unblock** at the bottom, then click on **OK**.
- Right-click on the saved script, select **Run as administrator** and follow the prompts.

To connect to the VPN: Click on the wireless/network icon in your system tray, select the new VPN entry, and click **Connect**. Once connected, you can verify that your traffic is being routed properly by [looking up your IP address on Google](#). It should say "Your public IP address is `Your VPN Server IP`".

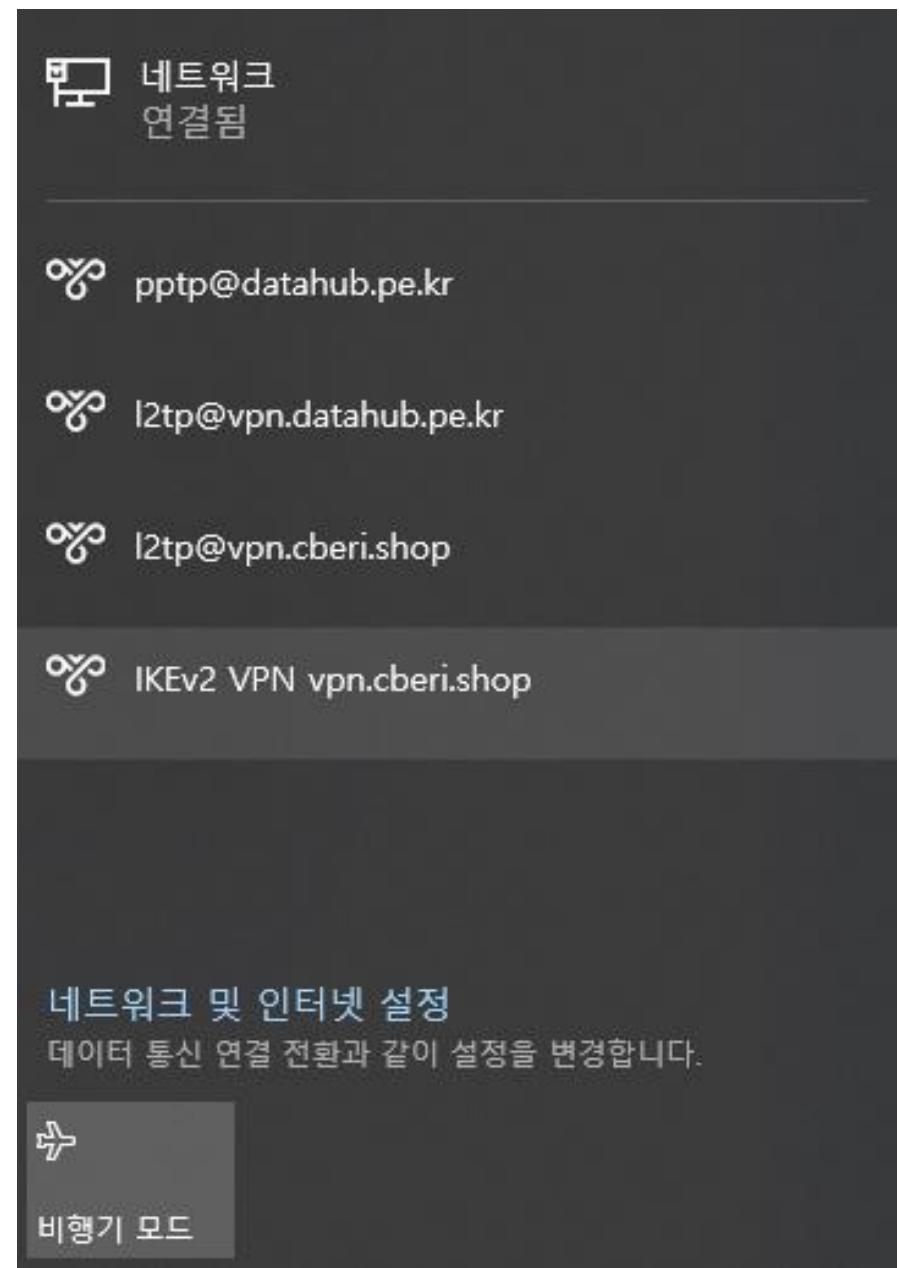
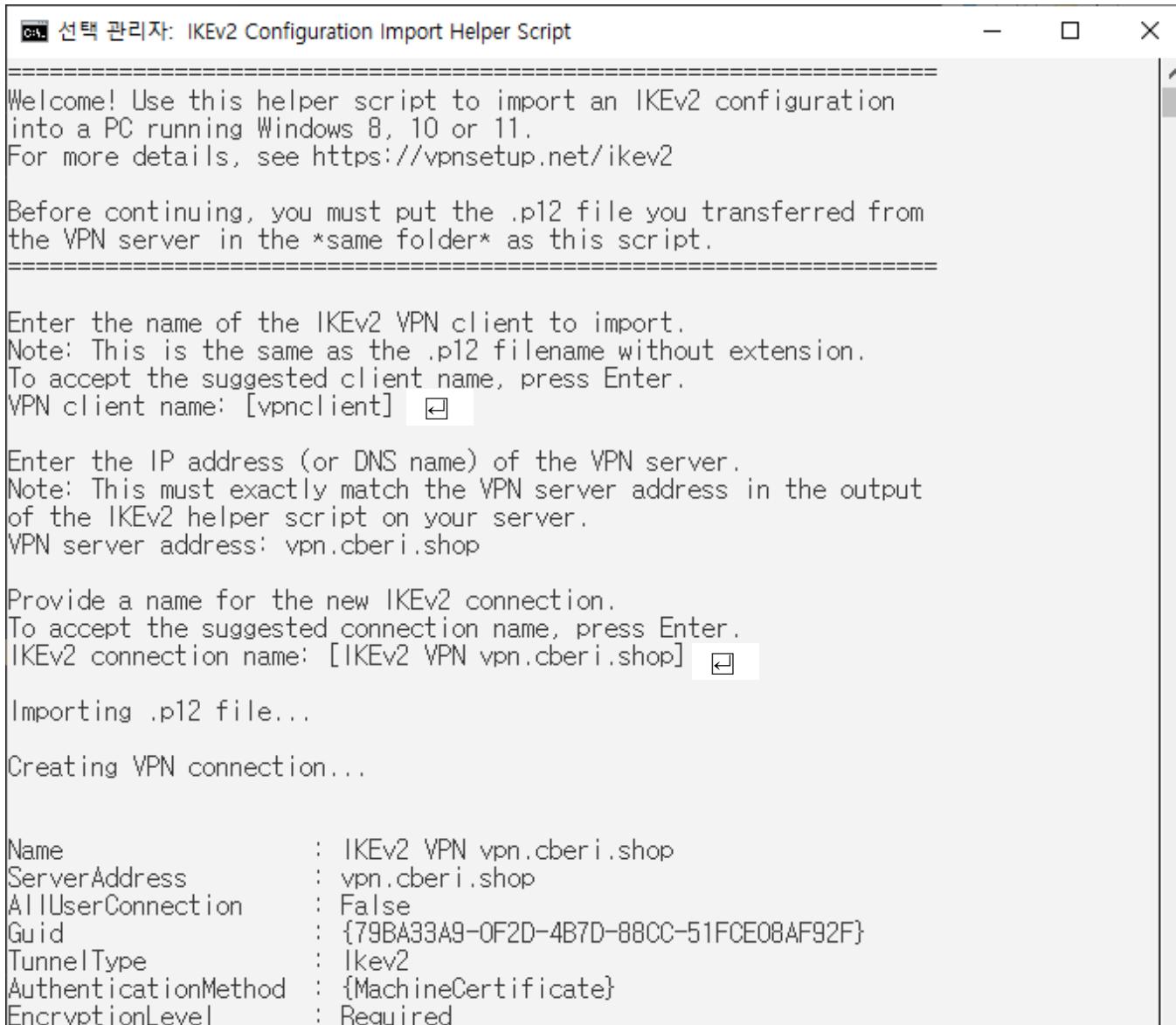
If you get an error when trying to connect, see [Troubleshooting](#).

# IKEv2 VPN 접속 프로필 생성

- Configure your devices to use the IKEv2 VPN.



# IKEv2 VPN 접속 프로필 생성



# IKEv2 프로토콜로 VPN Server 접속

## ■ IKEv2 용 추가 인증서 발급

- 앞서 사용한 인증서는 동일 NAT에 속해 있는 클라이언트에서 중복 접속 불가
- 이러한 경우 추가로 인증서를 발급받아 사용자마다 다른 인증서를 사용해야 함

```
[root@cberi ~]# docker exec -it ipsec-vpn-server ikev2.sh --addclient vpnclient2
```

```
IKEv2 Script Copyright (c) 2020-2023 Lin Song 4 Jan 2023
Adding a new IKEv2 client 'vpnclient2', using default options.
Creating client configuration...
=====
New IKEv2 client "vpnclient2" added!
VPN server address: 207.148.95.144
```

```
Client configuration is available inside the
Docker container at:
/etc/ipsec.d/vpnclient2.p12 (for Windows & Linux)
/etc/ipsec.d/vpnclient2.sswan (for Android)
/etc/ipsec.d/vpnclient2.mobileconfig (for iOS & m
Next steps: Configure IKEv2 clients. See:
https://vpnsetup.net/clients2
=====
```

```
[root@cberi ipsec-vpn]# docker exec -it ipsec-vpn-server ikev2.sh --listclients
```

```
IKEv2 Script Copyright (c) 2020-2023 Lin Song 4 Jan 2023
Checking for existing IKEv2 client(s)...

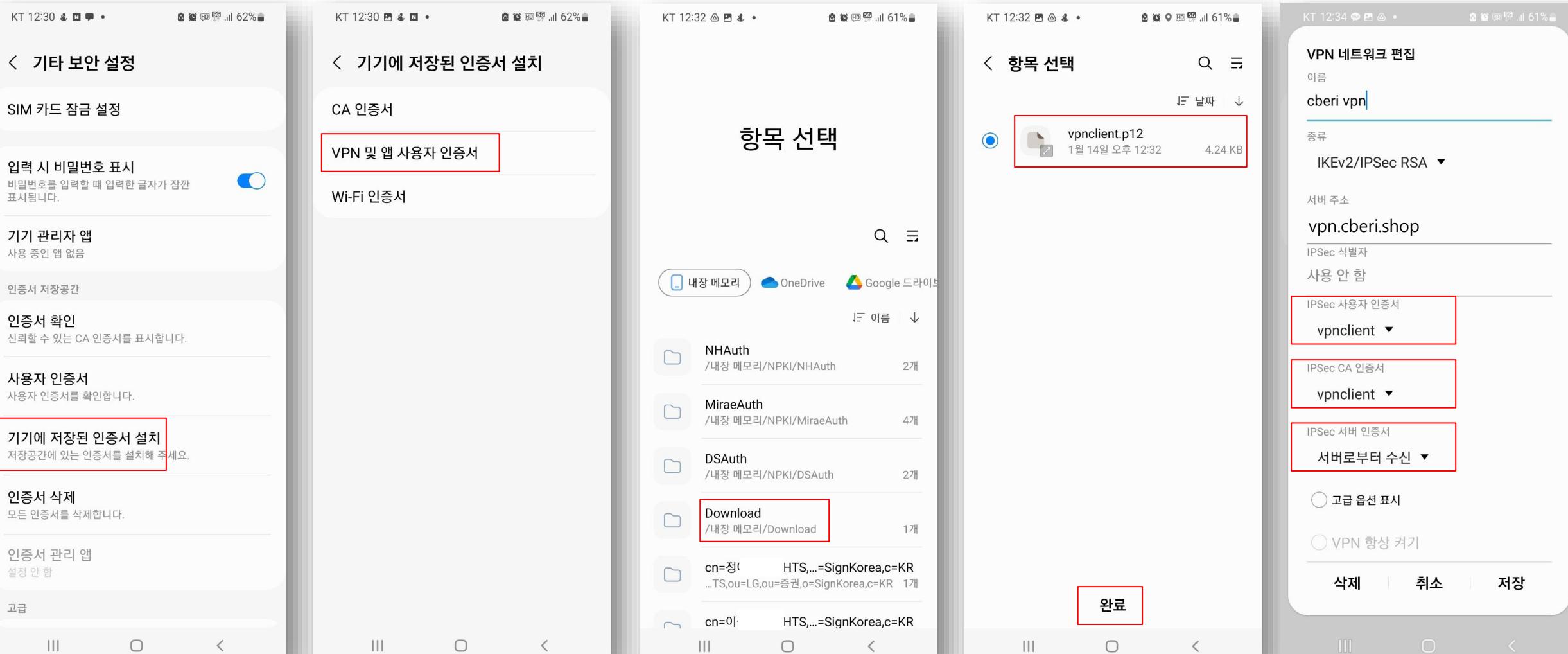
```

| Client Name | Certificate Status |
|-------------|--------------------|
| vpnclient   | valid              |
| vpnclient2  | valid              |

```
Total: 2 clients
```

# 안드로이드폰에서 VPN Server 접속

## ■ IKEv2 보안 접속 : 안드로이드 폰 (삼성 S21 예시)



# L2TP/IPSec VPN Server 셋팅

## ■ 실행 스크립트 파일 생성

```
[root@cberi ~]# cd /srv/ipsec-vpn/
[root@cberi ipsec-vpn]# vim start.sh

#!/bin/bash

docker run \
 --detach --privileged \
 --name ipsec-vpn-server \
 --env-file /srv/ipsec-vpn/vpn.env \
 --restart=always \
 -v /lib/modules:/lib/modules:ro \
 -p 500:500/udp \
 -p 4500:4500/udp \
 hwsl2/ipsec-vpn-server
```

```
[root@cberi ipsec-vpn]# chmod u+x start.sh
```

## ■ IKEv2 지원 스크립트

```
[root@cberi ~]# cd /srv/ipsec-vpn/
[root@cberi ipsec-vpn]# vim start_ikev2.sh

#!/bin/bash

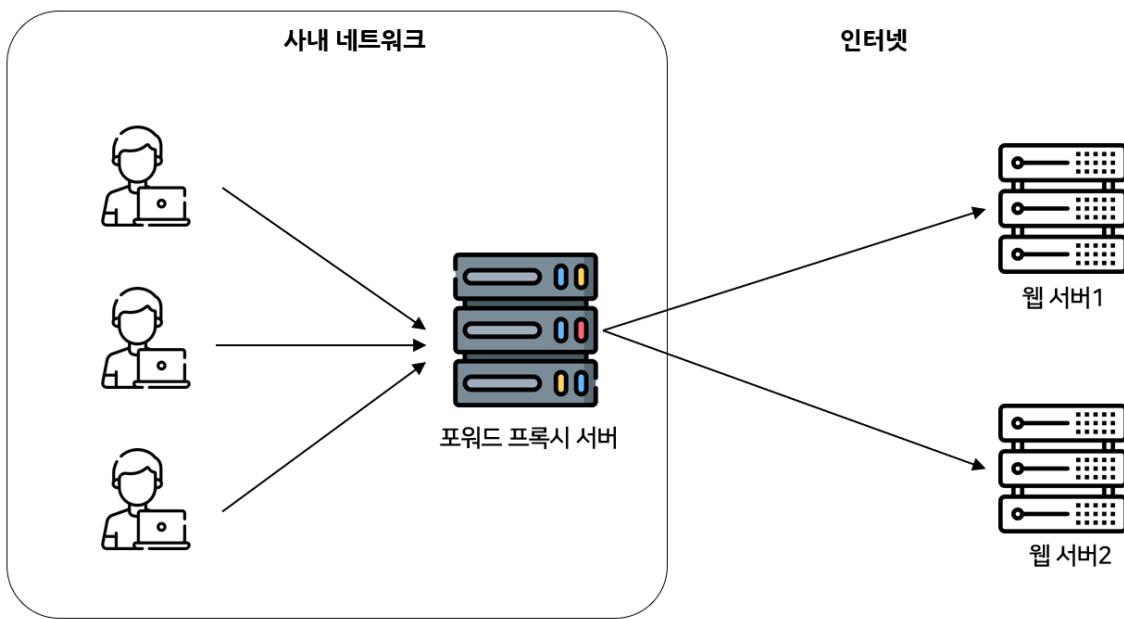
docker run \
 --detach --privileged \
 --name ipsec-vpn-server \
 --env-file /srv/ipsec-vpn/vpn.env \
 --restart=always \
 -v ikev2-vpn-data:/etc/ipsec.d \
 -v /lib/modules:/lib/modules:ro \
 -p 500:500/udp \
 -p 4500:4500/udp \
 hwsl2/ipsec-vpn-server
```

```
[root@cberi ipsec-vpn]# chmod u+x start_ikev2.sh
```

## 4. SSL 웹서버 구축

- 1) forward vs reverse proxy
- 2) nginx-proxy-acme-companion
- 3) php 웹서버 환경 구축
- 4) phpMyAdmin
- 5) wordpress 사이트 구축

# 포워드 프록시 (forward proxy)



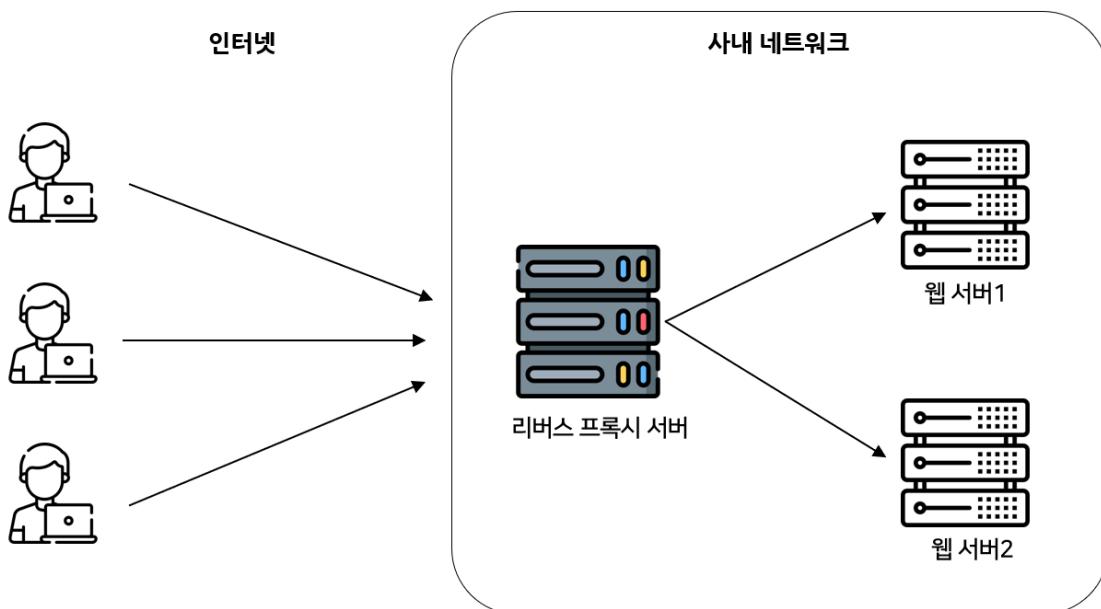
우선 정부, 학교, 기업 등과 같은 기관은 해당 기관에 속한 사람들의 제한적인 인터넷 사용을 위해 방화벽을 사용합니다. 포워드 프록시 서버는 이런 제한을 위해 사용합니다. 즉, 해당 기관에 속한 사람들이 그들이 방문하고자 하는 웹사이트에 직접적으로(directly) 방문하는 것을 방지합니다.

- 프록시: 서버와 클라이언트 사이에 중계기로서 대리로 통신을 수행하는 것

이 말은 곧 포워드 프록시 서버는 기관에 속한 유저가 특정 컨텐츠에 접근하는 것을 방지하는데 사용된다는 말과 동일합니다. 예를 들어, 포워드 프록시 서버에 룰을 추가해서 특정 사이트에 접속하는 것을 막을 수 있습니다.

마지막으로 포워드 프록시 서버를 사용하면 유저의 정체를 숨겨줍니다. 인터넷을 사용하는 대부분 유저들은 익명성을 바라는데 프록시 서버를 사용하지 않으면 자신의 정체가 탄로날 수 있습니다. 하지만 만약 포워드 프록시 서버를 사용하면 IP 주소를 역추적해도 정체를 파악하기 어렵습니다. 왜냐면 IP 추적해도 프록시 서버만 보이기 때문입니다.

# 리버스 프록시 (reverse proxy)



우선 리버스 프록시 서버는 로드 밸런싱 (load balancing)에 사용됩니다. 유명한 웹 사이트는 하루에도 수백만명이 방문합니다. 그리고 그러한 대량의 트래픽을 하나의 싱글 서버로 감당해 내기란 어렵습니다. 하지만 리버스 프록시 서버를 여러개의 서버 앞에 두면 특정 서버가 과부화 되지 않게 로드밸런싱이 가능합니다.

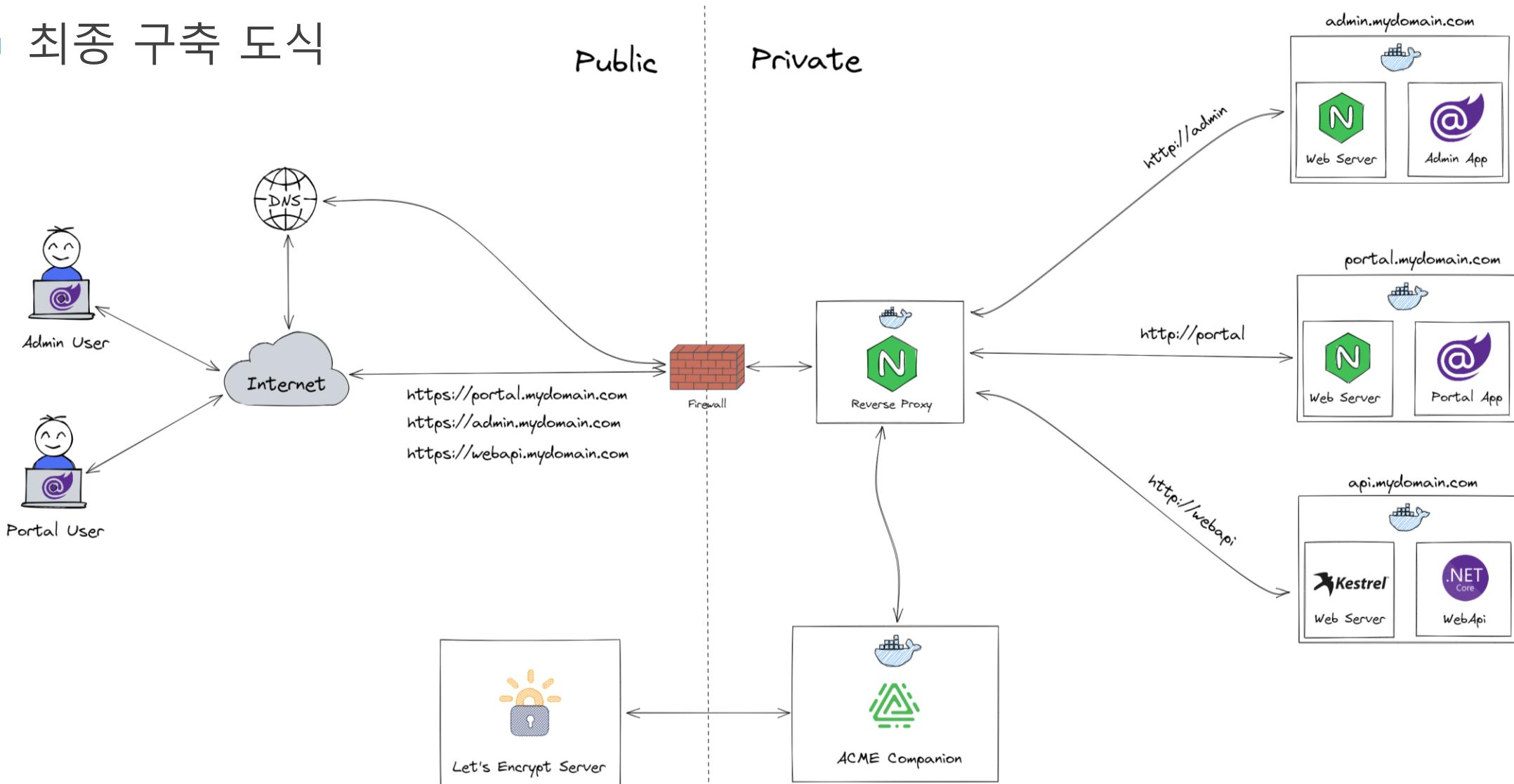
또한 리버스 프록시를 사용하면 보안에 좋습니다. 리버스 프록시를 사용하면 본래 서버의 IP 주소를 노출시킬 필요가 없습니다. 따라서 해커들의 DDoS 공격과 같은 공격을 막는데 유용합니다.

그리고 리버스 프록시 서버에는 성능 향상을 위해 캐시 데이터를 저장할 수 있습니다. 만약 어떤 한국에 있는 유저가 미국에 웹서버를 두고 있는 사이트에 접속할 때, 리버스 프록시 서버가 한국에 있다고 해봅시다. 그러면 한국에 있는 유저는 한국에 있는 리버스 프록시 서버와 통신합니다. 따라서 리버스 프록시 서버에 캐싱되어 있는 데이터를 사용할 경우에는 더 빠른 성능을 보여줄 수 있는 것입니다.

마지막으로 SSL 암호화에 좋습니다. 본래 서버가 클라이언트들과 통신을 할 때 SSL(or TSL)로 암호화, 복호화를 할 경우 비용이 많이 듭니다. 그러나 리버스 프록시를 사용하면 들어오는 요청을 모두 복호화하고 나가는 응답을 암호화해주므로 클라이언트와 안전한 통신을 할 수 있으며 본래 서버의 부담을 줄여줍니다.

# 단일 호스트 멀티 도메인 SSL 웹서버 구축 (feat. Let'sEncrypt)

## ■ 최종 구축 도식



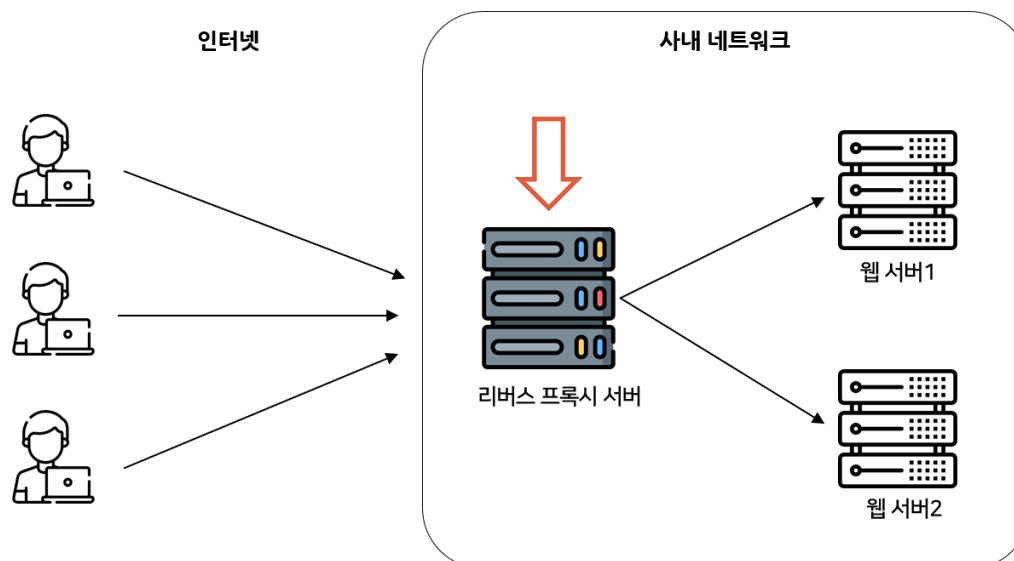
# 단일 호스트 멀티 도메인 SSL 웹서버 구축 (feat. LetsEncrypt)

- 특징
  - acme.sh 를 사용하여 Let's Encrypt(또는 다른 ACME CA) 인증서의 자동 생성/갱신.
  - http-01 챌린지를 통해서만 암호화 /ACME 도메인 유효성 검사
  - 인증서 생성/갱신 시 자동 업데이트 및 nginx 구성 다시 로드
  - 다중 도메인(SAN) 인증서 생성을 지원
  - 시작 시 강력한 RFC7919 Diffie-Hellman 그룹 생성
- 활용 도구
  - <https://github.com/nginx-proxy/acme-companion>
- 작업 순서
  - 1단계 : nginx-proxy 설치
  - 2단계 : acme-companion 설치
  - 3단계 : 내부 웹서버 설치(필요한 만큼)

cloudflare proxy 기능 끄고 작업할 것!!!  
모든 설정 완료 후 cloudflare 사용

# step1) nginx-proxy 설치

- nginx-proxy
  - 리버스 프록시 역할
  - 외부와의 통신을 담당
  - 80번 포트에서 http 통신
  - 443번 포트에서 https 통신



- 준비
  - 명령줄

```
[root@cberi ~]# cd /srv
[root@cberi srv]# mkdir nginx-proxy
[root@cberi srv]# cd nginx-proxy
[root@cberi srv]# mkdir certs html vhost.d
[root@cberi nginx-proxy]# tree /srv/nginx-proxy/
/srv/nginx-proxy/
└── certs
└── html
└── vhost.d
```

# step1) nginx-proxy 설치

- 도커 컨테이너 생성

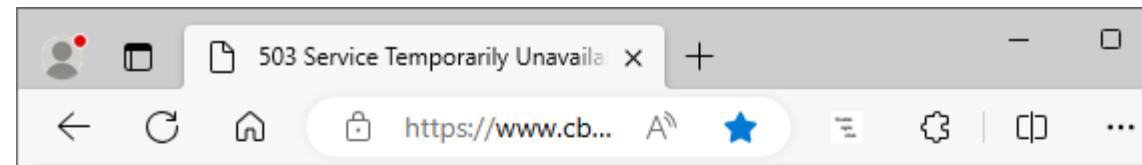
```
docker run --detach --privileged \
--name nginx-proxy \
--publish 80:80 \
--publish 443:443 \
--network="nginx-proxy" \
-v /srv/nginx-proxy/certs:/etc/nginx/certs \
-v /srv/nginx-proxy/vhost.d:/etc/nginx/vhost.d \
-v /srv/nginx-proxy/html:/usr/share/nginx/html \
-v /run/podman/podman.sock:/tmp/docker.sock:ro \
-e TZ=Asia/Seoul \
nginxproxy/nginx-proxy
```

- 방화벽 개방 (이미 열림)

```
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
firewall-cmd --reload
success
```

- 작동확인

- 웹브라우저에서 다음 주소 접속
  - <http://www.cberi.shop>
- 아직 프록시 된 서비스를 생성하지 않으므로 아래 오류가 정상임



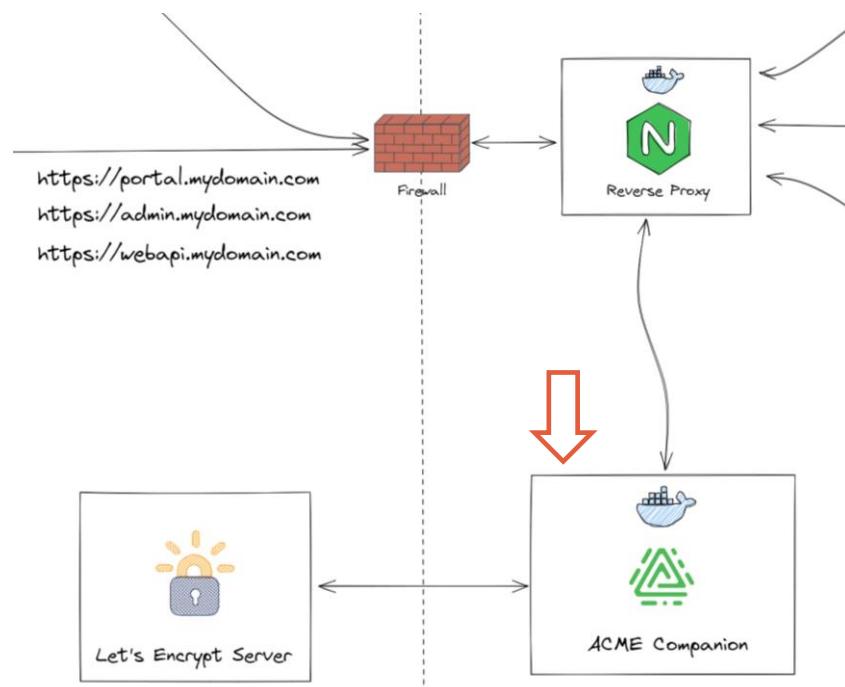
503 Service Temporarily Unavailable

nginx

## step2) acme-companion 설치

- acme-companion

- SSL 인증서 발급 자동화 프로그램
- Let's Encrypt에서 3개월 인증서 발급
- 3개월 지나면 갱신과정 자동 진행



- 도커 컨테이너 실행

- 명령줄

```
cd /srv
mkdir acme-companion
cd acme-companion
mkdir acme

docker run --detach --privileged \
--name nginx-proxy-acme \
--network="nginx-proxy" \
--volumes-from nginx-proxy \
-v /run/podman/podman.sock:/var/run/docker.sock:ro \
-v /srv/acme-companion/acme:/etc/acme.sh \
-e "DEFAULT_EMAIL=dianox@cberi.go.kr" \
-e TZ=Asia/Seoul \
nginxproxy/acme-companion
```

## step3) 프록시되는 내부 서버 설치

- 설치 서버 목록

|   | host name | 서비스 내용              | 설치 홈 폴더        | 컨테이너명     | 비고         |
|---|-----------|---------------------|----------------|-----------|------------|
| 1 | www       | Default web service | /home/www      | www       | php-fpm 필요 |
| 2 | sqladmin  | phpMyAdmin          | /home/sqladmin | sqladmin  | php-fpm 필요 |
| 3 | word      | 워드프레스 웹사이트          | /home/word     | word      | php-fpm 필요 |
| 4 | portainer | portainer           | (컨테이터 단독사용)    | portainer |            |
| 5 | dapi      | django REST API     | /home/dapi     | dapi      |            |
| 6 | sqlite    | sqlite-web 관리도구     | (컨테이터 단독사용)    | sqlite    |            |
| : | :         | :                   | :              | :         | :          |
|   |           |                     |                |           |            |

# php-fpm 설치

- php-fpm이란?
  - PHP FastCGI Process Manager
  - nginx에 php를 사용하려면 php-fpm 필요
  - 일반 CGI
    - 요청이 들어올 때마다 신규 프로세스를 생성/구동/종료하여 이 과정에서 부하 증가 등의 이슈가 발생
  - FastCGI
    - 미리 프로세스를 생성한 뒤 해당 프로세스를 요청이 들어올 때마다 재활용
    - 3~30배의 성능 개선 효과
- 도커 컨테이너 실행

```
docker pull php:8.2-fpm

docker run -d --privileged \
--name php-fpm \
--network="nginx-proxy" \
--expose 9000 \
-v /home:/home \
-e TZ=Asia/Seoul \
php:8.2-fpm
```
- php를 사용하는 여러 계정에 접근할 수 있도록 /home 폴더를 php-fpm 컨테이너와 공유해야 함에 유의

# proxied 서버로 www 구성

## ■ 준비

### ■ 관리 폴더 생성

```
[root@cberi ~]# mkdir /srv/sites
[root@cberi ~]# mkdir /srv/sites/www
```

### ■ 로그 폴더 생성

```
[root@cberi ~]# mkdir /home/www/logs
[root@cberi ~]# chwon www.www /home/www/logs
```

### ■ nginx 설정 파일 작성

```
[root@cberi ~]# mkdir /srv/sites/www/conf.d
[root@cberi ~]# cd /srv/sites/www/conf.d
[root@cberi ~]# vi nginx.conf
```

nginx.conf

```
server {
 listen 80;
 listen [::]:80;
 root /home/www/html;
 server_tokens off;

 location / {
 index index.php index.html;
 }
 #error_page 404 /404.html;

 # redirect server error pages to the static page /50x.html
 error_page 500 502 503 504 /50x.html;
 location = /50x.html {
 root /usr/share/nginx/html;
 }

 # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
 location ~ \.php$ {
 include fastcgi_params;
 fastcgi_pass php-fpm:9000;
 fastcgi_index index.php;
 fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
 }
}
```

# proxied 서버로 www 구성 (과거 버전과 비교)

- 단독 www 웹서버
  - docker run 명령줄

```
docker run -d --name web --privileged \
-p 80:80 \
-v /home/www/html:/usr/share/nginx/html:ro \
nginx
```

- proxied www 웹서버
  - docker run 명령줄

```
docker run -d \
--name www \
--network="nginx-proxy" \
-v /srv/sites/www/conf.d:/etc/nginx/conf.d:z \
-v /home/www/logs:/var/log/nginx:z \
-v /home/www:/home/www:z \
--env "VIRTUAL_HOST=www.cberi.shop" \
--env "LETSENCRYPT_HOST=www.cberi.shop" \
--env "HTTPS_METHOD=noredirect" \
nginx
```

# proxied 서버로 www 구성

- 시험 페이지 작성
  - 명령줄

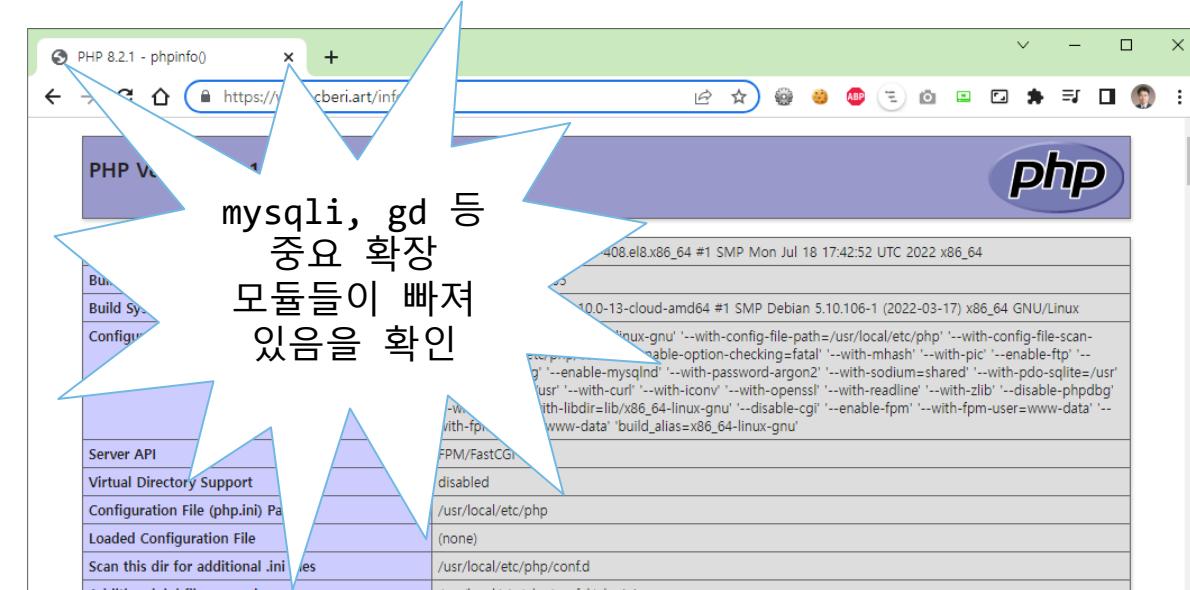
```
su - www
$ cd /home/www/html
$ vim info.php

<?php
 phpinfo();
?>

[www@cberi logs]$ tail -n 2 access.log
10.89.0.2 - - [01/Feb/2023:09:38:30 +0000] "GET
/info.php HTTP/1.1" 200 72796 "-" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/109.0.0.0 Safari/537.36" "220.86.241.91"
10.89.0.2 - - [01/Feb/2023:09:38:37 +0000] "GET
/info.php HTTP/1.1" 200 72713 "-" "Mozilla/5.0 (Windows
NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/109.0.0.0 Safari/537.36" "220.86.241.91"
```

## ■ 테스트

- <https://www.cberi.shop/info.php>



PHP 설치에 워드프레스가 요구하는 MySQL 확장이 없는 것 같습니다.

mysqli PHP 확장이 설치되고 활성화되었는지 확인하십시오.

이 용어가 무엇을 의미하는지 확실하지 않으면 호스트에게 문의해야 합니다. 아직 도움이 필요하면 [워드프레스 지원 포럼](#)에 방문할 수 있습니다.

# 해결책① - Dockerfile로 커스텀 이미지 빌드

- php:7.4 base 이미지 생성 예시
  - Dockerfile 생성

```
[root@cberi ~]# cd /srv/php-fpm
[root@cberi php-fpm]# mkdir build
[root@cberi php-fpm]# cd build
[root@cberi build]# vim Dockerfile
```

FROM php:7.4-fpm

```
RUN apt-get update && apt-get install -y \
 libfreetype6-dev \
 libjpeg62-turbo-dev \
 libpng-dev \
 && pecl install stats-2.0.3 \
 && docker-php-ext-configure gd --with-freetype --with-jpeg \
 && docker-php-ext-install -j$(nproc) gd \
 && docker-php-ext-install mysqli pdo_mysql \
 && docker-php-ext-enable stats
```

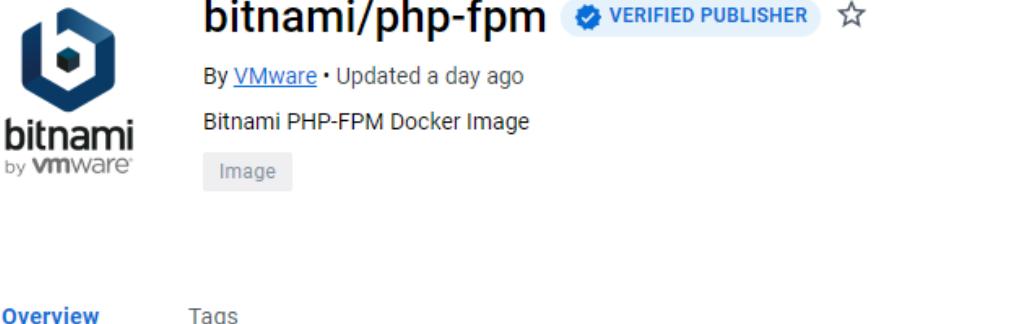
- 이미지 빌드

```
[root@cberi build]# docker build --tag php7.4-cberi .
STEP 1/2: FROM php:7.4-fpm
Resolved "php" as an alias
(/etc/containers/registries.conf.d/000-shortnames.conf)
Trying to pull docker.io/library/php:7.4-fpm...
Getting image source signatures
Copying blob a8e3b94fe6c1 done
```

- php-fpm 추가 실행

```
docker run -d --privileged \
--name php7-fpm -h php7-fpm \
--network="nginx-proxy" \
--expose 9000 \
-v /home:/home \
-e TZ=Asia/Seoul \
php7.4-cberi
```

# 해결책② - 원하는 확장 모듈이 들어있는 도커 이미지 사용



bitnami/php-fpm VERIFIED PUBLISHER ☆  
By VMware • Updated a day ago  
Bitnami PHP-FPM Docker Image  
Image

Overview Tags

## PHP-FPM packaged by Bitnami

### What is PHP-FPM?

PHP-FPM (FastCGI Process Manager) is an alternative PHP FastCGI implementation with some additional features useful for sites of any size, especially busier sites.

### Overview of PHP-FPM

Trademarks: This software listing is packaged by Bitnami. The respective trademarks mentioned in the offering are owned by the respective companies, and use of them does not imply any affiliation or endorsement.

### TL;DR

- 기존 php:8.2-fpm 컨테이너 삭제

```
docker rm -f php-fpm
```

- 재실행

```
docker pull bitnami/php-fpm
```

```
docker run -d --privileged \
--name php-fpm \
--network="nginx-proxy" \
--expose 9000 \
-v /home:/home \
-e TZ=Asia/Seoul \
bitnami/php-fpm
```

- www 컨테이너도 다시 띄워야 적용됨

```
$ docker run -it --name phpfp -v /path/to/app:/app bitnami/php-fpm
```

# php 추가 설치

- php cli 설정
  - php command line interface

```
vim /usr/local/bin/php

#!/bin/sh
docker run -it --name php-run --rm --privileged -v $(pwd):/app bitnami/php-fpm php $@

chmod +x /usr/local/bin/php
```

- composer 설치

- 명령줄

```
vim /usr/local/bin/composer

#!/bin/sh
docker run -it --name php-run --rm --privileged -v $(pwd):/app bitnami/php-fpm composer $@

chmod +x /usr/local/bin/composer
```

# 전송량 한도 문제

## ■ upload 테스트

```
[root@cberi ~]# su - www
[www@cberi www]$ cd html
[www@cberi html]$ mkdir submit submit/files
[www@cberi html]$ cd submit
[www@cberi submit]$ chmod o+rwx files
$ curl -o index.php \
 http://datahub.pe.kr/submit.php
```

## ■ <https://www.cberi.shop/submit/>

홍덕고등학교 정보교과 파일 제출

파일 제출

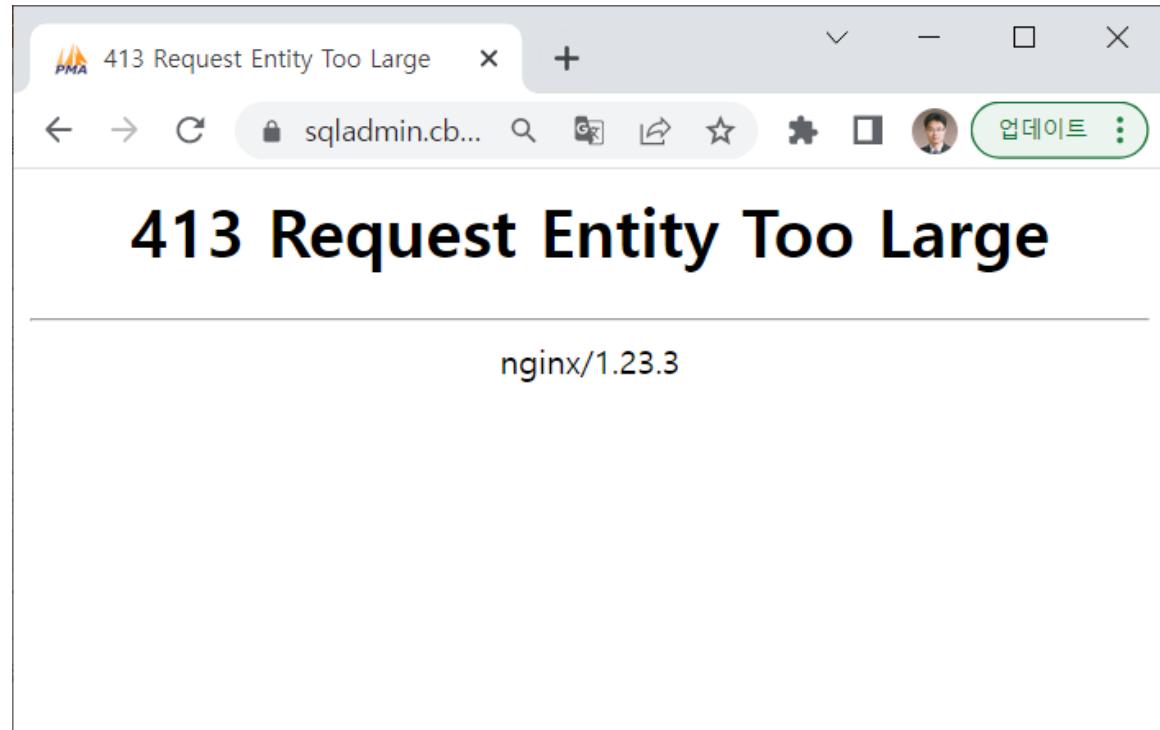
|               |    |    |    |                      |
|---------------|----|----|----|----------------------|
| 차시            | 학년 | 반  | 번호 | 이름                   |
| 01차 자료의 수집과 능 | 1  | 선택 | 선택 | <input type="text"/> |

제출파일

선택된 파일 없음

지금까지 제출된 전체 파일 목록

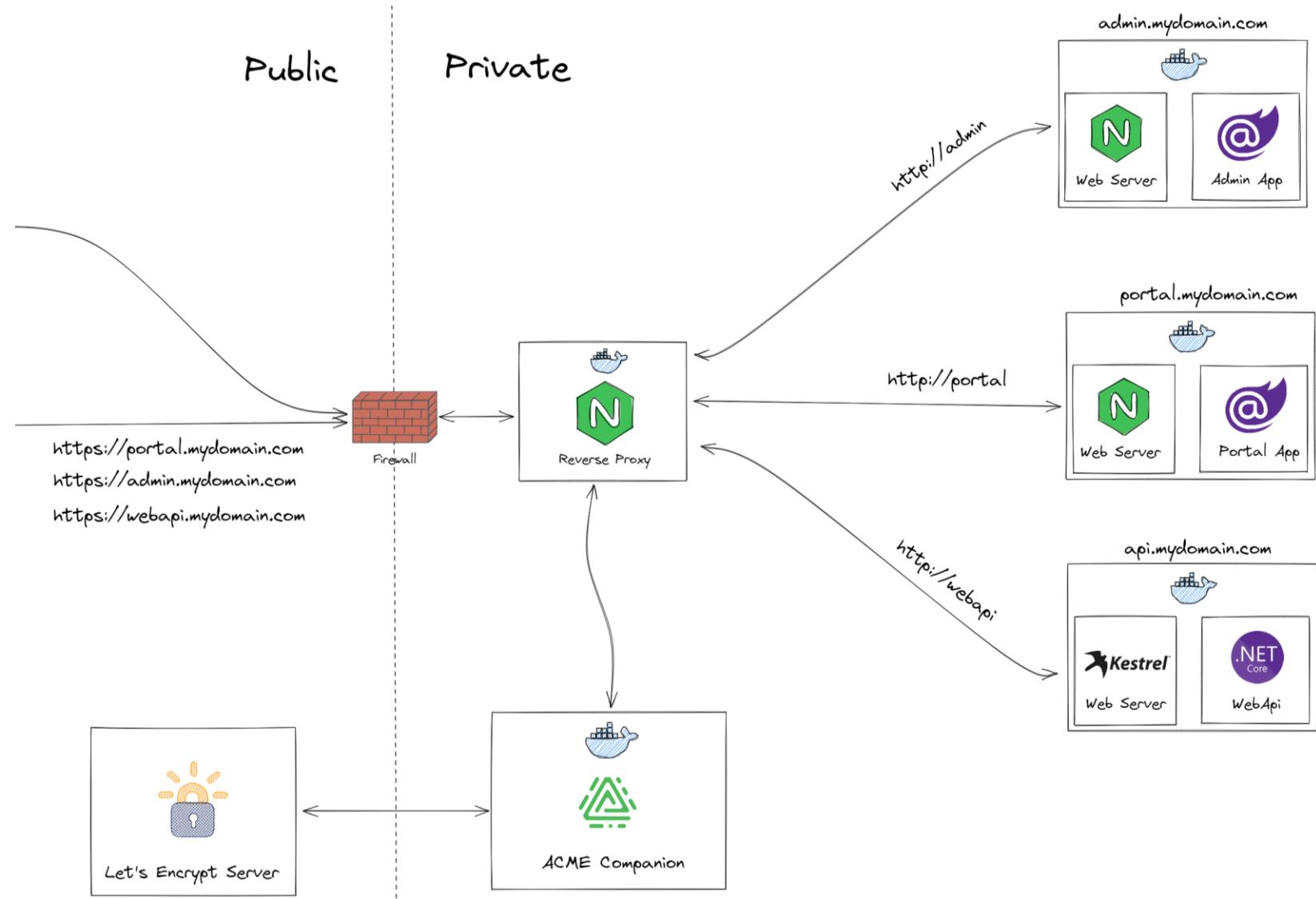
## ■ 2M 이상 파일 전송 시



# 업로드 최대 용량 설정 변경

- 3군데 수정 필요

- 1) 외부 nginx-proxy
- 2) php-fpm
- 3) 내부 nginx 서버



# 업로드 최대 용량 설정 변경

## 1) 외부 nginx-proxy

- 커스텀 프록시 설정 파일 생성

```
[root@cberi ~]# cd /srv/nginx-proxy/
[root@cberi nginx-proxy]# vi custom.conf

this file is custom proxy settings
client_max_body_size 128m;
server_tokens off;
```

- 기존에 만든 nginx.conf 파일에 내용을 추가하는 방식도 가능함

- nginx-proxy 재시작

```
docker run --detach --privileged \
--name nginx-proxy \
--publish 80:80 \
--publish 443:443 \
--network="nginx-proxy" \
-v /srv/nginx-proxy/certs:/etc/nginx/certs \
-v /srv/nginx-proxy/vhost.d:/etc/nginx/vhost.d \
-v /srv/nginx-proxy/html:/usr/share/nginx/html \
-v /srv/nginx-proxy/custom.conf:/etc/nginx/conf.d/custom.conf \
-v /run/podman/podman.sock:/tmp/docker.sock:ro \
nginxproxy/nginx-proxy
```

# 업로드 최대 용량 설정 변경

## 2) php-fpm 설정 변경 (php8버전)

- php 설정 파일 복사

```
docker exec -i php-fpm ls /opt/bitnami/php/etc
common.conf
conf.d
environment.conf
pear.conf
php-fpm.conf
php-fpm.conf.default
php-fpm.d
php.ini
php.ini-development
php.ini-production

cd /srv/php-fpm/
docker cp -a php-fpm:/opt/bitnami/php/etc etc8
```

- 개발용 설정 파일로 셋팅 할 때

```
[root@cberi ~]# cd /srv/php-fpm/etc8
[root@cberi etc8]# mv php.ini php.ini.bak
[root@cberi etc8]# ln -s php.ini-development php.ini
```

- 배포용 설정 파일로 셋팅 할 때

```
[root@cberi ~]# cd /srv/php-fpm/etc8
[root@cberi etc8]# mv php.ini php.ini.bak
[root@cberi etc8]# ln -s php.ini-production php.ini
```

- 커스텀 설정 파일 설정 (최대 업로드 등)

```
cd /srv/php-fpm/etc8/conf.d
vim custom.ini

max_file_uploads = 128M
post_max_size = 128M
upload_max_filesize = 128M
max_execution_time = 60
date.timezone = Asia/Seoul
```

# 업로드 최대 용량 설정 변경

## 2) php-fpm 설정 변경 (php7버전)

- php 설정 파일 복사

```
docker exec -i php7-fpm ls /usr/local/etc/php
conf.d
php.ini-development
php.ini-production # docker exec -i php-fpm ls
/opt/bitnami/php/etc/conf.d

cd /srv/php-fpm/
docker cp -a php7-fpm:/usr/local/etc/php etc7
```

- 개발용 설정 파일로 셋팅 할 때

```
[root@cberi ~]# cd /srv/php-fpm/etc7
[root@cberi etc7]# mv php.ini php.ini.bak
[root@cberi etc7]# ln -s php.ini-development php.ini
```

- 배포용 설정 파일로 셋팅 할 때

```
[root@cberi ~]# cd /srv/php-fpm/etc7
[root@cberi etc7]# mv php.ini php.ini.bak
[root@cberi etc7]# ln -s php.ini-production php.ini
```

- 커스텀 설정 파일 설정 (최대 업로드 등)

```
cd /srv/php-fpm/etc8/conf.d
vim custom.ini

max_file_uploads = 128M
post_max_size = 128M
upload_max_filesize = 128M
max_execution_time = 60
date.timezone = Asia/Seoul
```

# 업로드 최대 용량 설정 변경

- php:8.2-fpm 재시작

```
docker run -d --privileged \
--name php-fpm \
--network="nginx-proxy" \
-v /home:/home \
-v /srv/php-fpm/etc8:/opt/bitnami/php/etc \
bitnami/php-fpm
```

- php:7.4-cberi 재시작

```
docker run -d --privileged \
--name php7-fpm \
--network="nginx-proxy" \
-v /home:/home \
-v /srv/php-fpm/etc7:/usr/local/etc/php \
php7.4-cberi
```

- start.sh 스크립트 생성

# 업로드 최대 용량 설정 변경

## 3) 내부 nginx 설정 변경

- www로 테스트 중이므로 그곳에 적용

```
cd /srv/sites/www/conf.d
vi custom.conf

this file is custom proxy settings
client_max_body_size 128m;
server_tokens off;

ls
nginx.conf
custom.conf
```

- www 재시작

```
docker run -d \
--name www \
--network="nginx-proxy" \
-v /srv/sites/www/conf.d:/etc/nginx/conf.d:z \
-v /home/www/logs:/var/log/nginx:z \
-v /home/www:/home/www:z \
--env "VIRTUAL_HOST=www.cberi.shop" \
--env "LETSENCRYPT_HOST=www.cberi.shop" \
--env "HTTPS_METHOD=noredirect" \
nginx
```

- start.sh 스크립트 작성

podman 특유의 hostname resolve 방식 때문에 컨테이너 실행 순서가 중요함.  
참조되는 컨테이너가 먼저 실행되어야 정상작동 함.

# phpMyAdmin 사이트 개설

## ■ phpmyadmin은?

- MySQL 데이터베이스를 관리할 수 있는 웹 기반 인터페이스.
- 기능
  - 데이터베이스 및 테이블 생성, 삭제 및 수정
  - 다양한 형식의 데이터 가져오기 및 내보내기
  - SQL 쿼리 실행
  - 테이블 지정 편집
  - 사용자 및 권한 관리
  - 인덱스 생성 및 편집

## ■ 준비

```
[root@cberi ~]# mkdir /srv/sites
[root@cberi ~]# mkdir /srv/sites/sqladmin
[root@cberi ~]# mkdir /srv/sites/sqladmin/conf.d
[root@cberi ~]# mkdir /home/sqladmin
[root@cberi ~]# chown www.www /home/sqladmin
[root@cberi ~]# su - www

[www@cberi ~]$ cd /home/sqladmin
[www@cberi sqladmin]$ mkdir logs
[www@cberi sqladmin]$ wget
https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-all-languages.zip
[www@cberi sqladmin]$ unzip phpMyAdmin-5.2.1-all-languages.zip
[www@cberi html]$ mv phpMyAdmin-5.2.1-all-languages html
[www@cberi sqladmin]$ tree -L 1 /home/sqladmin/
/home/sqladmin/
├── html
└── logs
 └── phpMyAdmin-5.2.1-all-languages.zip
[www@cberi html]$ exit
```

# phpMyAdmin 사이트 개설

## ■ nginx 설정 파일 작성

```
[root@cberi ~]# cd /srv/sites/sqladmin/conf.d
[root@cberi ~]# vim nginx.conf
```

## ■ 도커 컨테이너 실행

```
docker run -d \
--name sqladmin \
--network="nginx-proxy" \
-v /srv/sites/sqladmin/conf.d:/etc/nginx/conf.d:z \
-v /home/sqladmin/logs:/var/log/nginx:z \
-v /home/sqladmin:/home/sqladmin:z \
--env "VIRTUAL_HOST=sqladmin.cberi.shop" \
--env "LETSENCRYPT_HOST=sqladmin.cberi.shop" \
nginx
```

nginx.conf

```
server {
 listen 80;
 listen [::]:80;
 server_name sqladmin.cberi.shop;
 root /home/sqladmin/html;

 server_tokens off;
 client_max_body_size 128m;

 location / {
 index index.php index.html;
 }
 #error_page 404 /404.html;

 # redirect server error pages to the static page /50x.html
 error_page 500 502 503 504 /50x.html;
 location = /50x.html {
 root /usr/share/nginx/html;
 }

 # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
 location ~ \.php$ {
 include fastcgi_params;
 fastcgi_pass php7-fpm:9000;
 fastcgi_index index.php;
 fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
 }
}
```

# phpMyAdmin 사이트 개설

## ■ config.inc.php 설정

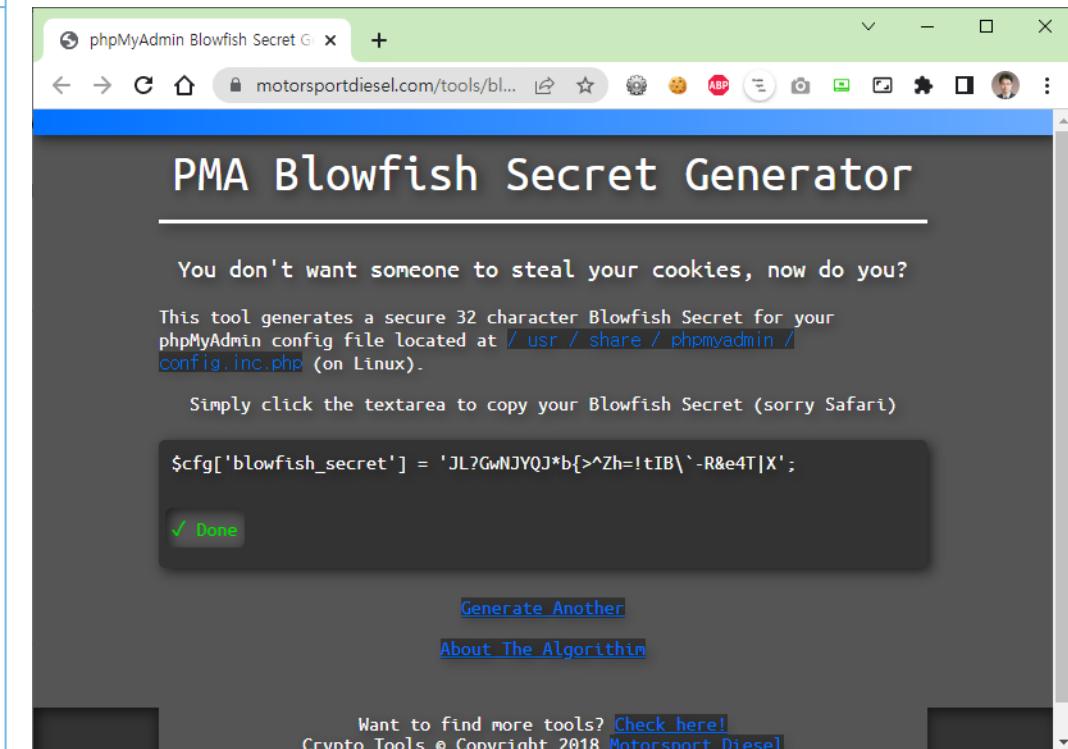
```
cd /home/sqladmin/html
mv config.sample.inc.php config.inc.php
vim config.inc.php
```

config.inc.php

```
13 * This is needed for cookie based authentication to encrypt password in
14 * cookie. Needs to be 32 chars long.
15 */
16 $cfg['blowfish_secret'] = 'JL?GwNjYQJ*b{>^Zh=!tIB\`-R&e4T|X';
/* YOU MUST FILL IN THIS FOR COOKIE AUTH!
17
18 /**
19 * Servers configuration
20 */
21 $i = 0;
22
23 /**
24 * First server
25 */
26 $i++;
27 /* Authentication type */
28 $cfg['Servers'][$i]['auth_type'] = 'cookie';
29 /* Server parameters */
30 $cfg['Servers'][$i]['host'] = 'mariadb';
31 $cfg['Servers'][$i]['compress'] = false;
```

## ■ PMA Blowfish secret generator

- <https://www.motorsportdiesel.com/tools/blowfish-salt/pma>



# phpMyAdmin 사이트 개설

- 접속 테스트

- <https://sqladmin.cberi.shop/>

The screenshot shows the phpMyAdmin landing page. At the top is the phpMyAdmin logo with a sailboat icon. Below it, a message says "phpMyAdmin에 오셨습니다". There is a language selection dropdown set to "한국어 - Korean". Below that is a login form with fields for "사용자명" (username) containing "root" and "암호" (password) which is empty. A red oval highlights the "Create" link in the configuration storage section of the page.

언어 (Language)

한국어 - Korean

로그인

사용자명: root

암호:

로그인

- 추가 설정

- tmp 폴더 생성 및 777권한 부여

```
cd /home/sqladmin/html
mkdir tmp
chmod 777 tmp
```

- configuration storage 생성

⚠️ phpMyAdmin 설정 스토리지가 완전히 설정되지 않아, 일부 확장 기능들이 비활성화 된 상태입니다.  
원인을 확인하려면 [여기](#)를 클릭하세요.  
대신 데이터베이스 작업 탭을 사용하여 설정할 수도 있습니다.

phpMyAdmin configuration storage

Configuration of pmadb... not OK

Create를 눌러준다.

릴레이션 기능 사용불가

⚠️ Create a database named 'phpmyadmin' and setup the phpMyAdmin configuration storage there.

# 점검 및 고찰

## ■ www.cberi.shop

- cloudflare 프록시 사용함
- php 8.3.1로 작동

## ■ sqladmin.cberi.shop

- cloudflare 프록시 사용 안함
- php 7.4.33로 작동

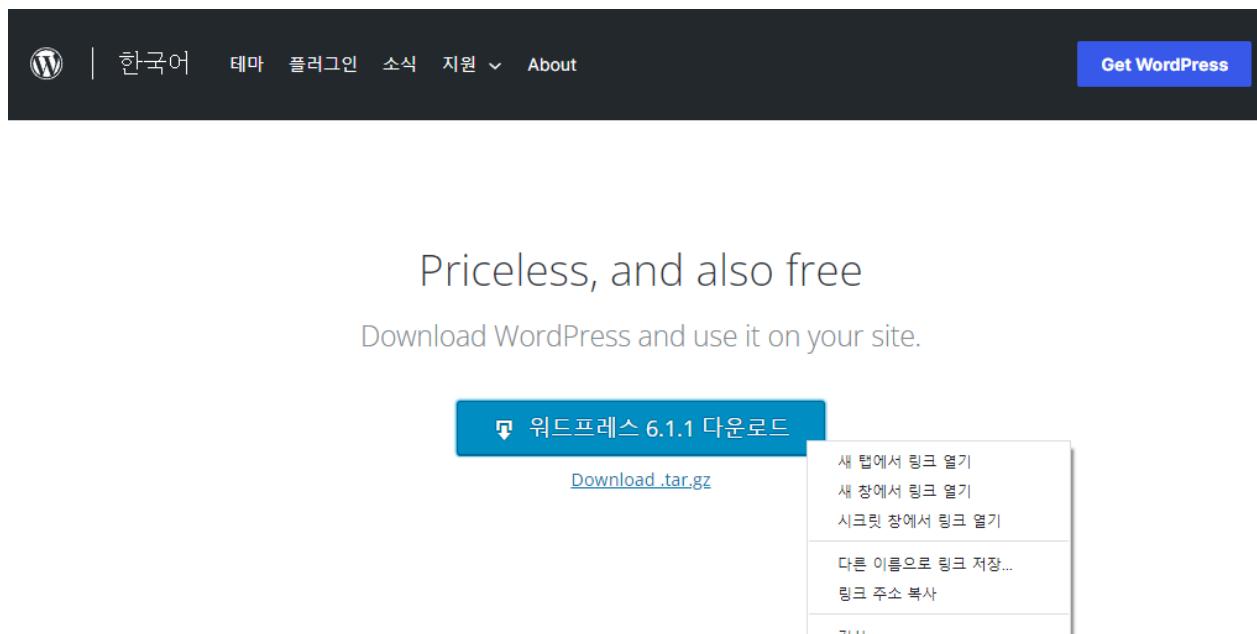
## ■ cberi.shop

- cloudflare 프록시 사용함
- www.cberi.shop의 CNAME
- 하지만 접속 안됨
- 문제를 해결하려면?

# wordpress 사이트 개설

## ■ 다운로드 링크 복사

- <https://ko.wordpress.org/download/>



## Installation

With our famous 5-minute installation, setting up WordPress for the first time is simple. We've created a [handy guide](#) to see you through the [installation process](#).

## Release no.

Want to get notified about WordPress releases? Join the [WordPress Announcements mailing list](#) and we will send a friendly message whenever there is a [new stable release](#).

## ■ 준비

```
su - word
$ wget https://ko.wordpress.org/latest-ko_KR.zip
[word@cberi ~]$ unzip latest-ko_KR.zip
[word@cberi ~]$ mv wordpress html
[word@cberi ~]$ mkdir logs
[word@cberi ~]$ chmod +x .
[word@cberi ~]$ exit
```

```
[root@cberi ~]# mkdir /srv/sites/word
[root@cberi ~]# cd /srv/sites/word
[root@cberi word]# mkdir conf.d
[root@cberi word]# vim conf.d/nginx.conf
```

# wordpress 사이트 개설

- nginx 컨테이너 생성
  - wordpress용 nginx.conf 준비
  - <https://www.nginx.com/resources/wiki/start/topics/recipes/wordpress/>

```
upstream php {
 server unix:/tmp/php-cgi.socket;
 server php-fpm:9000;
}

server {
 ## Your website name goes here.
 server_name domain.tld;
 ## Your only path reference.
 root /home/word/html;
 server_tokens off;
 client_max_body_size 128m;
```

- 실행

```
docker run -d --name word \
--network="nginx-proxy" \
-v /srv/sites/word/conf.d:/etc/nginx/conf.d:z \
-v /home/word/logs:/var/log/nginx:z \
-v /home/word:/home/word:z \
-e "VIRTUAL_HOST=word.cberi.shop" \
-e "LETSENCRYPT_HOST=word.cberi.shop" \
-e "HTTPS_METHOD=noredirect" \
nginx
```

- 주의

- server\_name 필드 삭제하지 않으면,  
**비공개 연결이 아닙니다.**  
공격자가 **word.cberi.shop**에서 사용자의 정보 (예: 암호, 메시지 또는 신용 카드)를 도용  
할 수 있습니다.

NET::ERR\_CERT\_AUTHORITY\_INVALID

# wordpress 사이트 개설

## ■ db 준비

## ■ 새 db 및 새 사용자 생성

The screenshot shows the 'User creation' page in phpMyAdmin for a MySQL database named 'mariadb'. The 'User creation' tab is selected in the top navigation bar. The main form fields are filled as follows:

- Username: word
- Host: localhost (%)
- Password: (hidden)
- Strength: 강력함 (Strong)
- Re-enter Password: (hidden)
- Authentication plugin: Native MySQL authentication
- Generate Password: (button)

A green success message at the bottom left states: "새 사용자를 추가했습니다." (A new user was added successfully). Below this message is the SQL command used to create the user:

```
CREATE USER 'word'@'%' IDENTIFIED VIA mysql_native_password USING '***';GRANT USAGE ON *.* TO 'word'@'%'
REQUIRE NONE WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0
MAX_USER_CONNECTIONS 0;CREATE DATABASE IF NOT EXISTS `word`;GRANT ALL PRIVILEGES ON `word`.* TO
'word'@'%';GRANT ALL PRIVILEGES ON `word`.* TO 'word'@'%';
```

In the bottom right corner, there is a link: "이리인 평점 1.5 수첩 1.5 PDF 코드 보기".

# wordpress 사이트 개설

아래에서 당신의 데이터베이스 연결 상세를 입력해야합니다. 이것을 잘 모른다면 당신의 호스트에 연락하세요.

데이터베이스 이름

word

워드프레스에 사용할 데이터베이스 이름.

사용자명

word

데이터베이스 사용자명.

비밀번호

pororo23@word



데이터베이스 비밀번호.

데이터베이스 호스트

mariadb

localhost(가) 작동하지 않는다면 이 정보는 자신의 웹호스팅에서 받을 수 있습니다.

테이블 접두어

wp\_

하나의 데이터베이스에서 여러 개의 워드프레스를 설치하여 운영하려면 이것을 변경하세요.

제출

wp-config.php 파일에 쓰기를 할 수 없음.

wp-config.php 파일을 직접 만들고 다음 문장을 붙일 수 있습니다.

wp-config.php에 대한 환경설정 규칙:

```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file
 * during the installation.
 * You don't have to use the web site, you can copy this file
 * to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 */
```

완료한 다음 "설치 실행"을 클릭하세요.

설치 실행

# wordpress 사이트 개설

## ■ wp-config.php 쓰기지원

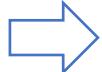
디렉토리 쓰기 권한 추가

```
[root@cberi ~]# cd /srv/sites/word/html
[root@cberi html]# chmod o+w .
[root@cberi html]# ls -alhdF
drwxr-xrwx. 5 word word 4.0K Jan 14 13:49 ./
```

웹브라우저에서 [F5] 새로고침

설치통과 후 쓰기 권한 삭제

```
[root@cberi html]# chmod o-w .
```



아주 좋습니다! 설치의 일부를 통과 했습니다. 워드프레스는 이제 데이터베이스와 통신할 수 있습니다. 준비됐다면, 다음을...

설치 실행

# wordpress 사이트 개설

## 환영합니다

유명한 워드프레스 5분 설치 절차에 오신 것을 환영합니다! 아래의 정보를 입력하기만 하면 세계에서 가장 확장성 있고 강력한 개인 발행 플랫폼을 사용할 수 있습니다.

## 정보가 필요합니다

다음 정보를 제공해주세요. 걱정하지 마세요. 이 설정을 나중에 언제든지 바꿀 수 있습니다.

사이트 제목

cberi's wordpress

사용자명

admin

사용자명은 알파벳, 숫자, 스페이스, 밑줄, 하이픈, 마침표, @ 기호만 가능합니다.

비밀번호

pororo23@admin



숨기기

강함

중요: 로그인할 비밀번호가 필요할 것입니다. 안전한 위치에서 저장해주세요.

이메일 주소

akapo@naver.com

계속하기 전에 이메일 주소를 다시 확인하세요.

검색 엔진 가시성

검색 엔진이 이 사이트를 검색하는 것을 차단

이 요청이 받아들여지는 것은 전적으로 검색 엔진에 좌우됩니다.

워드프레스 설치



## 성공!

워드프레스가 설치됐습니다. 감사합니다. 즐거운 시간을!

사용자명

admin

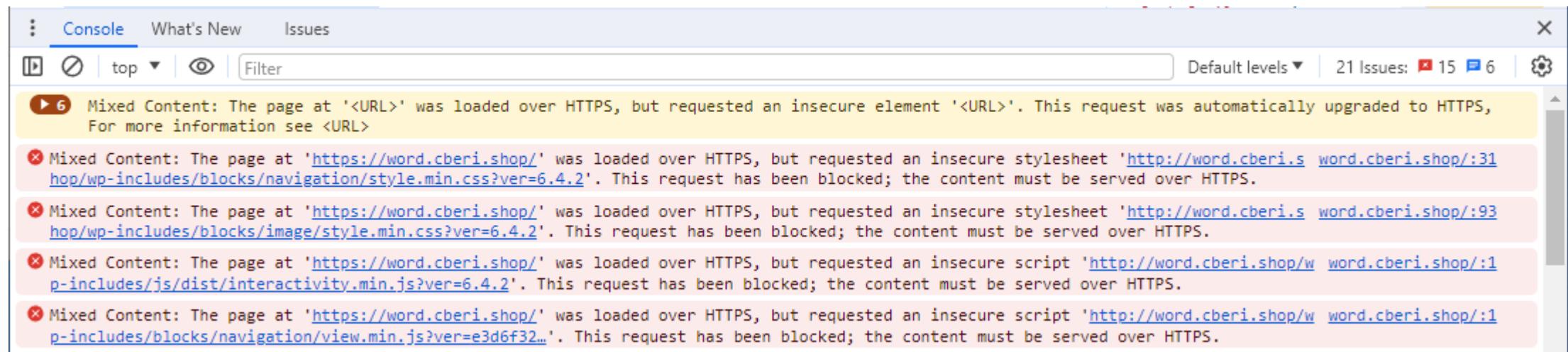
비밀번호

선택한 비밀번호

[로그인](#)

# wordpress 사이트 개설

## ■ mixed contents 오류 수정



The screenshot shows a browser's developer tools console tab. It displays several error messages related to mixed content. The first message is a warning about a page being loaded over HTTPS but requesting an insecure element. The subsequent five messages are errors indicating that pages loaded over HTTPS are requesting insecure stylesheets and scripts from HTTP URLs.

- Mixed Content: The page at '<URL>' was loaded over HTTPS, but requested an insecure element '<URL>'. This request was automatically upgraded to HTTPS, For more information see <URL>
- Mixed Content: The page at '<https://word.cberi.shop/>' was loaded over HTTPS, but requested an insecure stylesheet '<http://word.cberi.shop/wp-includes/blocks/navigation/style.min.css?ver=6.4.2>'. This request has been blocked; the content must be served over HTTPS.
- Mixed Content: The page at '<https://word.cberi.shop/>' was loaded over HTTPS, but requested an insecure stylesheet '<http://word.cberi.shop/wp-includes/blocks/image/style.min.css?ver=6.4.2>'. This request has been blocked; the content must be served over HTTPS.
- Mixed Content: The page at '<https://word.cberi.shop/>' was loaded over HTTPS, but requested an insecure script '<http://word.cberi.shop/wp-includes/js/dist/interactivity.min.js?ver=6.4.2>'. This request has been blocked; the content must be served over HTTPS.
- Mixed Content: The page at '<https://word.cberi.shop/>' was loaded over HTTPS, but requested an insecure script '<http://word.cberi.shop/wp-includes/blocks/navigation/view.min.js?ver=e3d6f32...>'. This request has been blocked; the content must be served over HTTPS.

```
$ vi wp-config.php
```

```
<?php
if (isset($_SERVER['HTTP_X_FORWARDED_PROTO']) && $_SERVER['HTTP_X_FORWARDED_PROTO'] == 'https')
 $_SERVER['HTTPS'] = 'on';

if (! defined('FS_METHOD')) define('FS_METHOD', 'direct');
/**
 * The base configuration for WordPress
 *
```

# wordpress 사이트 개설

## ■ mixed contents 오류 수정

The screenshot shows the phpMyAdmin interface for managing the WordPress database. The top navigation bar includes tabs for '보기' (View), '구조' (Structure), 'SQL', '검색' (Search), '삽입' (Insert), '내보내기' (Export), '가져오기' (Import), '권한' (Privileges), '테이블 작업' (Table Operations), and '트리거' (Triggers). A green status bar at the top indicates '0 - 24행 표시 중 (합계 141, 질의 실행시간 0.0006 초.)'. Below the status bar, the SQL query 'SELECT \* FROM `wp\_options`' is displayed. The results table shows four rows in the wp\_options table:

|                          | option_id | option_name     | option_value           | autoload |
|--------------------------|-----------|-----------------|------------------------|----------|
| <input type="checkbox"/> | 1         | siteurl         | https://word.cberi.art | yes      |
| <input type="checkbox"/> | 2         | home            | https://word.cberi.art | yes      |
| <input type="checkbox"/> | 3         | blogname        | cberi's wordpress      | yes      |
| <input type="checkbox"/> | 4         | blogdescription |                        | yes      |

A red dashed box highlights the 'option\_value' column for the first two rows (siteurl and home). The 'option\_value' for both rows is 'https://word.cberi.art', which is causing the mixed content error.

# wordpress 사이트 개설

## ■ mixed contents 오류 수정

The screenshot shows a MySQL query interface with the following details:

- Toolbar:** Includes buttons for 보기 (View), 구조 (Structure), SQL, 검색 (Search), 삽입 (Insert), 내보내기 (Export), 가져오기 (Import), 테이블 작업 (Table Operations), and 트리거 (Triggers).
- Message Bar:** Shows a green message: "✓ 0 - 0행 표시 중 (합계 1, 질의 실행시간 0.0004 초.)".
- SQL Query:** The query is: `SELECT * FROM `wp_users`;`
- Buttons:** Below the query are buttons for [프로파일링] (Profiling), [인라인 편집] (Inline Edit), [수정] (Edit), [SQL 해석] (SQL Explain), [PHP 코드 보기] (PHP Code View), and [다시 보기] (Revert).
- Update Statement:** A redacted update statement is shown: `UPDATE `wp_users` SET `user_url` = 'https://word.cberi.shop' WHERE `wp_users`.`ID` = 1;`
- Buttons:** Below the update statement are buttons for [인라인 편집] (Inline Edit), [수정] (Edit), and [PHP 코드 보기] (PHP Code View).
- Table Filter:** Includes checkboxes for "모두 보기" (All View) and "행 갯수:" (Row Count: 25), and a "행 필터링" (Filter Row) input field.
- Table Headers:** The table has columns: ID, user\_login, user\_pass, user\_nicename, user\_email, and user\_url.
- Data Row:** One row is displayed: ID 1, user\_login "admin", user\_pass "\$P\$BXWecPu0ukExAwVYHUewuSjZ4tm1Va1", user\_nicename "admin", user\_email "akapo@naver.com", and user\_url "https://word.cberi.shop". The "user\_url" cell is highlighted with a red dashed border.
- Action Buttons:** At the bottom of the table area are buttons for "수정" (Edit), "복사" (Copy), "삭제" (Delete), "내보내기" (Export), and checkboxes for "모두 체크" (Check All) and "선택한 것을:" (Selected). There are also buttons for "수정" (Edit), "복사" (Copy), "삭제" (Delete), and "내보내기" (Export).
- Bottom Filter:** Similar to the top filter, it includes checkboxes for "모두 보기" (All View) and "행 갯수:" (Row Count: 25), and a "행 필터링" (Filter Row) input field.

# wordpress 사이트 개설

- 관리자 페이지 <https://word.cberi.shop/wp-admin/> 접속

The screenshot shows the WordPress dashboard with a dark theme. The top navigation bar includes the WordPress logo, site title 'Web Agency Demo 1', message count '0', a '새로 추가' (Add New) button, and a greeting '안녕하세요 admin님' (Hello admin) with a user icon.

The left sidebar contains a navigation menu with icons and text: '홈' (Home), '업데이트' (Updates), '글' (Posts), '미디어' (Media), '페이지' (Pages), '댓글' (Comments), 'Neve' (Theme), '외모' (Appearance), '플러그인' (Plugins), '사용자' (Users), '도구' (Tools), and '설정' (Settings). The '알림판' (Dashboard) item is currently selected and highlighted in blue.

The main content area features a large banner with the text '워드프레스에 오신 것을 환영합니다!' (Welcome to WordPress!) and a link '6.4.2 버전에 대해 더 알아보기' (Learn more about version 6.4.2). Below the banner are three cards:

- 블록과 패턴으로 풍부한 콘텐츠를 만드세요** (Create rich content using blocks and patterns)  
Description: '블록 패턴은 사전 설정된 블록 레이아웃입니다. 영감을 얻기 위해 사용하거나 새 페이지를 바로 만드세요.'  
[블록 패턴에 대한 자세한 내용](#)
- 사용자 정의 시작** (Start customizing)  
Description: '사용자 정의하기에서 사이트 로고, 헤더, 메뉴, 더 많은 것을 환경설정하세요.'  
[사용자 정의하기 열기](#)
- 사이트를 제작하는 새 방법 알아보기** (Learn new ways to build your site)  
Description: '새로운 종류의 워드프레스 테마가 있습니다. 블록 테마라고 하며 블록과 스타일로 항상 만들고 싶었던 사이트를 제작할 수 있습니다.'

## 5. 실전 웹서버 구축

- 1) portainer.io 설치
- 2) podman container 자동실행
- 3) RESTful API 소개
- 4) django 이용 REST API 구축
- 5) sqlite 설치

---

충북교육연구정보원 강사 박정진

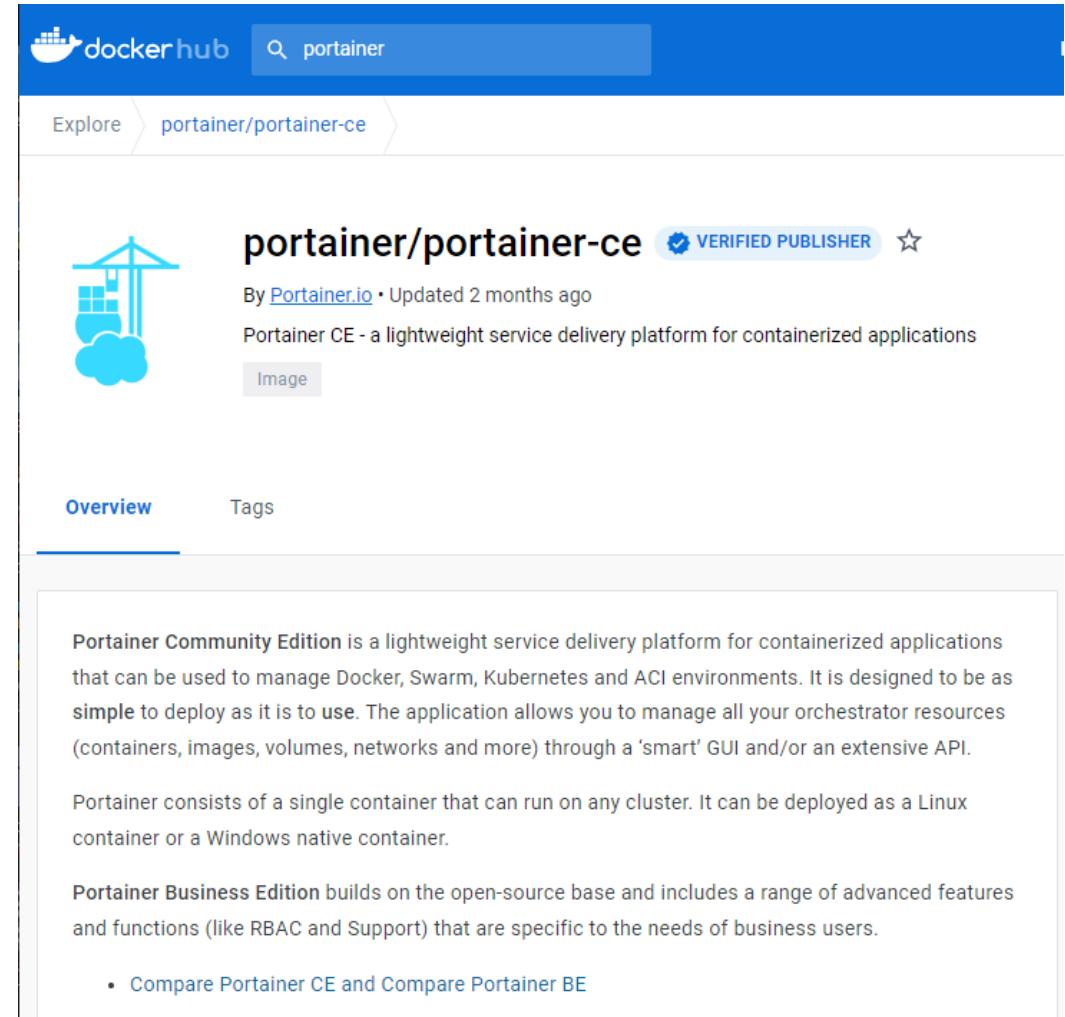


# portainer.io 설치

## ■ portainer란?

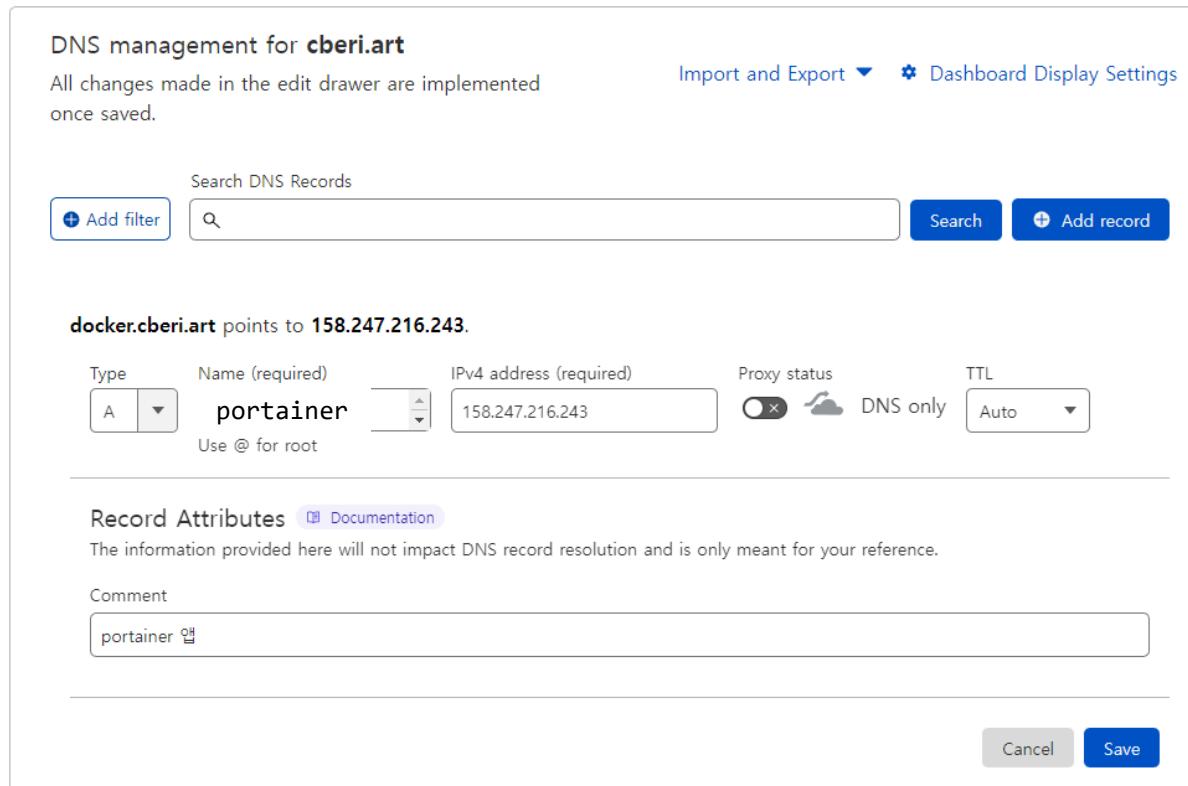
- 도커를 관리하는 웹 인터페이스
- 클라우드 환경에서 컨테이너를 쉽게 이동하고 배포할 수 있도록 하는 도구
- 실행 중인 컨테이너, 설치한 이미지, 볼륨 등을 직관적으로 확인할 수 있을 뿐 아니라 클릭 몇 번으로 상태를 변경할 수도 있고, 템플릿을 이용해 원하는 애플리케이션을 간단하게 구현 가능
- 7억 다운로드 수를 기록한 유용한 관리 도구

## ■ dockerhub에서 배포 중



# portainer.io 설치

- DNS 레코드 생성
  - cloudflare DNS 섹션



DNS management for **cberi.art**  
All changes made in the edit drawer are implemented once saved.

Import and Export ▾ **Dashboard Display Settings**

Search DNS Records  **Search** **Add record**

**docker.cberi.art** points to **158.247.216.243**.

| Type | Name (required) | IPv4 address (required) | Proxy status | TTL  |
|------|-----------------|-------------------------|--------------|------|
| A    | portainer       | 158.247.216.243         | DNS only     | Auto |

**Record Attributes** Documentation  
The information provided here will not impact DNS record resolution and is only meant for your reference.

Comment

**Cancel** **Save**

- 도커 컨테이너로 실행
  - 명령줄

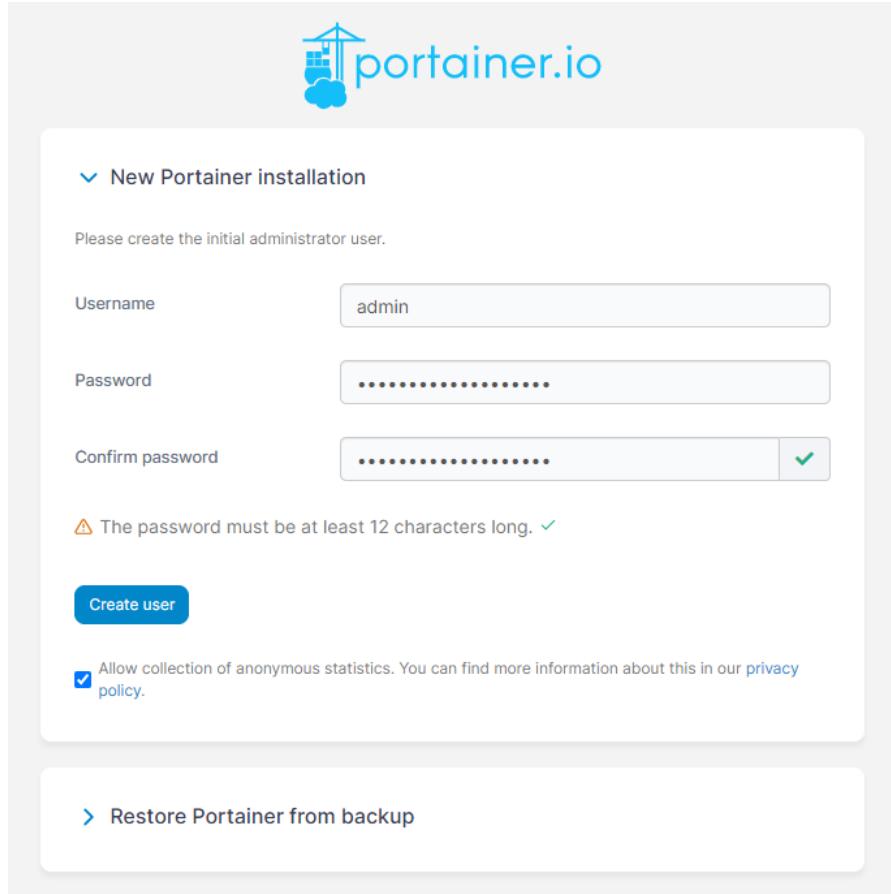
```
docker run -d --privileged \
--name portainer \
--network="nginx-proxy" \
-v /run/podman/podman.sock:/var/run/docker.sock:z \
-e "VIRTUAL_HOST=portainer.cberi.shop" \
-e "LETSENCRYPT_HOST=portainer.cberi.shop" \
-e "VIRTUAL_PORT=9000" \
-e "HTTPS_METHOD=noredirect" \
portainer/portainer-ce
```

환경변수를 자신의 환경에 맞게 설정 필요

- portainer 컨테이너는 자체 웹서버를 포함한다.
  - http: 9000, https: 9433

# portainer.io 설치

- 접속 주소
  - <https://docker.cberi.shop>



New Portainer installation

Please create the initial administrator user.

Username: admin

Password: .....  
Confirm password: ..... ✓

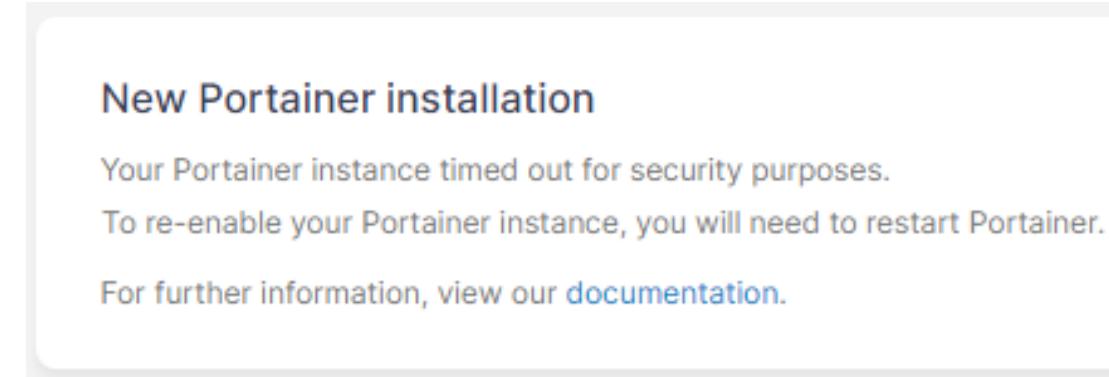
The password must be at least 12 characters long. ✓

Create user

Allow collection of anonymous statistics. You can find more information about this in our [privacy policy](#).

Restore Portainer from backup

- 초기 설정
  - 관리자 암호 지정
  - 만약 아래와 같은 메시지를 만나면...



- 컨테이너 재시작 후...

```
docker restart portainer
```
- 같은 과정을 반복하면 해결됨



## ■ 초기 화면

The screenshot shows the Portainer.io interface with a dark blue sidebar on the left and a light gray main content area.

**Left Sidebar (Dark Blue):**

- Home** (selected)
- Environment:** None selected
- Settings**
  - Users
  - Environments
  - Registries
  - Authentication logs
  - Notifications
  - Settings

**Bottom Left:** portainer.io Community Edition 2.16.2 Upgrade

**Right Main Area (Light Gray):**

Environment Wizard

### Quick Setup

admin

**Environment Wizard**

#### Welcome to Portainer

We have connected your local environment of Docker to Portainer.

Get started below with your local portainer or connect more container environments.

**Get Started** (Icon: Docker ship with a red checkmark) **Add Environments** (Icon: Power plug)

Proceed using the local environment which Portainer is running in

Connect to other environments



## ■ local (portainer를 실행중인 호스트) 선택

The screenshot shows the Portainer.io Home page. On the left, a dark sidebar menu includes "Home", "Environment: None selected", and "Settings" with sub-options: "Users", "Environments", "Registries", "Authentication logs", "Notifications", and "Settings". The main content area has a header "Environments" and "Home". It features a "Latest News From Portainer" section with a "dismiss" button and a decorative 3D cube graphic. Below this is the "Environments" section with a sub-header "Click on an environment to manage". It includes a "Refresh" button, a search bar, and filters for "Platform", "Connection Type", "Status", "Tags", "Groups", "Agent Version", "Clear all", and "Sort By". A single environment entry is listed: "local" (status "up"), last updated "2023-01-16 17:24:44". The entry details show "0 stacks", "6 containers" (with green status icons), "9 volumes", "6 images", "1 CPU", "2.1 GB RAM", "0 GPU", and "No tags". To the right, environment metadata is shown: "Group: Unassigned", "Standalone 4.2.0", and the path "/var/run/docker.sock". At the bottom right, there's a "Items per page" dropdown set to "10".

# portainer.io - container list

portainer.io COMMUNITY EDITION

Home

local

- Dashboard
- App Templates
- Stacks
- Containers
- Images
- Networks
- Volumes
- Events
- Host

Settings

- Users
- Environments
- Registries
- Authentication logs

portainer.io Community Edition 2.16.2 Upgrade

Containers

## Container list

Containers

| Name             | State   | Image                                      | Created             | IP Address | GPUs | Published Ports   | Ownership      |
|------------------|---------|--------------------------------------------|---------------------|------------|------|-------------------|----------------|
| ipsec-vpn-server | running | docker.io/hwdsl2/ipsec-vpn-server:latest   | 2023-01-14 08:51:55 | 10.88.0.2  | none | 500:500 4500:4500 | administrators |
| mariadb          | running | docker.io/library/mariadb:latest           | 2023-01-15 14:41:00 | 10.89.0.9  | none | 3306:3306         | administrators |
| nginx-proxy      | running | docker.io/nginxproxy/nginx-proxy:latest    | 2023-01-16 15:52:05 | 10.88.0.15 | none | 80:80 443:443     | administrators |
| nginx-proxy-acme | running | docker.io/nginxproxy/acme-companion:latest | 2023-01-16 15:54:57 | 10.88.0.17 | none | -                 | administrators |
| portainer        | running | docker.io/portainer/portainer-ce:latest    | 2023-01-16 16:56:22 | 10.88.0.22 | none | -                 | administrators |
| www              | running | docker.io/library/nginx:latest             | 2023-01-16 16:01:36 | 10.88.0.18 | none | -                 | administrators |

5 item(s) selected

Items per page 10

# DNS의 \* A레코드

## ■ \* A레코드

- 새로운 사이트를 추가할 때마다 A레코드 항목을 추가하는 일이 귀찮다면..

DNS management for cberi.art  
All changes made in the edit drawer are implemented once saved.

Import and Export ▾ Dashboard Display Settings

Search DNS Records

\*.cberi.art points to 158.247.216.243.

| Type | Name (required) | IPv4 address (required) | Proxy status                        | DNS only                 | TTL  |
|------|-----------------|-------------------------|-------------------------------------|--------------------------|------|
| A    | *               | 158.247.216.243         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Auto |

Use @ for root

Record Attributes [Documentation](#)  
The information provided here will not impact DNS record resolution and is only meant for your reference.

Comment

## ■ 변화 확인

- 따로 설정되지 않은 모든 서브 도메인에 대하여 지정한 아이피 반환

```
[root@cberi ~]# nslookup sqladmin.cberi.shop
Server: 1.1.1.1
Address: 1.1.1.1#53
```

```
Non-authoritative answer:
Name: sqladmin.cberi.shop
Address: 207.148.95.144
```

```
[root@cberi ~]# nslookup daum.cberi.shop
Server: 1.1.1.1
Address: 1.1.1.1#53
```

```
Non-authoritative answer:
Name: daum.cberi.shop
Address: 207.148.95.144
```

# podman container 자동실행

- 재부팅 시 모든 실행 중이던 모든 container가 멈추는 문제 해결 필요

| CONTAINER ID | IMAGE                                      | COMMAND                                                         | CREATED     | STATUS                    |
|--------------|--------------------------------------------|-----------------------------------------------------------------|-------------|---------------------------|
| PORTS        |                                            | NAMES                                                           |             |                           |
| 3384b258ab4c | docker.io/hwdsl2/ipsec-vpn-server:latest   | /opt/src/run.sh<br>0.0.0.0:500->500/udp, 0.0.0.0:4500->4500/udp | 2 weeks ago | Exited (1) 20 hours ago   |
| 301d457fd6cf | docker.io/library/mariadb:latest           | mariadb<br>0.0.0.0:3306->3306/tcp                               | 9 days ago  | Exited (0) 20 hours ago   |
| 68e87d601737 | docker.io/nginxproxy/acme-companion:latest | /bin/bash /app/st...<br>nginx-proxy-acme                        | 9 days ago  | Exited (0) 20 hours ago   |
| a60a9a8389aa | docker.io/nginxproxy/nginx-proxy:latest    | forego start -r<br>0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp     | 9 days ago  | Exited (0) 20 hours ago   |
| 66a1952d6705 | docker.io/library/nginx:latest<br>sqladmin | nginx -g daemon o...<br>php-fpm                                 | 9 days ago  | Exited (0) 23 minutes ago |
| 8e6e6106ec84 | docker.io/bitnami/php-fpm:latest           | php-fpm -F --pid ...<br>lapi                                    | 9 days ago  | Exited (0) 20 hours ago   |
| 1b1b6c4493f3 | docker.io/library/nginx:latest             | nginx -g daemon o...<br>papi                                    | 9 days ago  | Exited (0) 20 hours ago   |
| 7d46e58e78d2 | docker.io/library/nginx:latest             | nginx -g daemon o...<br>papi                                    | 8 days ago  | Exited (0) 20 hours ago   |

# podman container 자동실행

- docker의 --restart 옵션
  - no: container를 재시작 시키지 않는다. (default)
  - on-failure[:max-retries]: container가 정상적으로 종료되지 않은 경우(exit code가 0이 아님)에만 재시작. max-retries로 재시작 최대 시도횟수를 지정 가능
  - always: container를 항상 재시작 시킨다. exit (code 상관 없이 항상 재시작)
  - unless-stopped: container를 stop시키기 전 까지 항상 재시작 시킨다.
- podman
  - podman은 docker와는 다르게 container 옵션을 통해 restart 정책 지정 불가 (입력해도 무시됨)
  - podmand은 컨테이너 시작 순서, 종속성 확인 또는 컨테이너 복구작업을 직접적으로 관여하지 않음.
  - 해결책: 도커 컨테이너를 재시작하는 서비스를 만들어 등록하는 방법으로 접근해야 함.
  - 서비스 파일 내용 만들어 주는 명령어  
`podman generate systemd --name <컨테이너이름>`

```
[root@cberi ~]# podman generate systemd --name mariadb
container-mariadb.service
autogenerated by Podman 4.6.1
Thu Jan 18 21:48:18 KST 2024

[Unit]
Description=Podman container-mariadb.service
Documentation=man:podman-generate-systemd(1)
Wants=network-online.target
After=network-online.target
RequiresMountsFor=/run/containers/storage

[Service]
Environment=PODMAN_SYSTEMD_UNIT=%n
Restart=on-failure
TimeoutStopSec=70
ExecStart=/usr/bin/podman start mariadb
ExecStop=/usr/bin/podman stop \
 -t 10 mariadb
ExecStopPost=/usr/bin/podman stop \
 -t 10 mariadb
PIDFile=/run/containers/storage/overlay-
containers/9b9840776cdf0abec6e121246c3c1ca827d8aa0166eb66440c88f6ecc6d7aa37/userdata/common.pid
Type=forking

[Install]
WantedBy=default.target
```

# podman container 자동실행

- 서비스 파일 생성
  - 파일 위치: /etc/systemd/system/{서비스명}.service
  - {서비스명}: **podman.{컨테이너명}** 으로 생성
  - ex) nginx-proxy 컨테이너의 경우: **podman.nginx-proxy.service**

```
[root@cberi ~]# podman generate systemd --name mariadb > /etc/systemd/system/podman.mariadb.service
```

- 서비스 등록

```
[root@cberi ~]# systemctl enable --now podman.mariadb
Created symlink /etc/systemd/system/default.target.wants/podman.mariadb.service →
/etc/systemd/system/podman.mariadb.service.
```

- 자동 재시작이 필요한 다른 컨테이너들도 모두 똑같이 시행

# REST

## ■ 기본 개념

- 웹의 창시자(HTTP) 중의 한 사람인 Roy Fielding의 2000년 박사학위 논문에 의해서 소개
- 현재의 아키텍쳐가 웹의 본래 설계의 우수성을 많이 사용하지 못하고 있다고 판단
- 웹의 장점을 최대한 활용할 수 있는 네트워크 기반의 아키텍처를 소개했는데 이것이 바로
- REST = Representational State Transfer

# RESTful API

- HTTP URI(Uniform Resource Identifier)를 통해 자원(Resource)을 명시하고, HTTP Method(POST, GET, PUT, DELETE)를 통해 해당 자원에 대한 CRUD Operation을 적용하는 것을 의미한다.
- CRUD Operation
  - Create : 생성(POST)
  - Read : 조회(GET)
  - Update : 수정(PUT)
  - Delete : 삭제(DELETE)
  - HEAD: header 정보 조회(HEAD)
- 웹 사이트의 이미지, 텍스트, DB 내용 등의 모든 자원에 고유한 ID인 HTTP URI를 부여한다.

# REST 구성요소

## 1) 자원(Resource): URI

- 모든 자원에 고유한 ID가 존재하고, 이 자원은 Server에 존재한다.
- 자원을 구별하는 ID는 '/groups/:group\_id'와 같은 HTTP URI다.
- Client는 URI를 이용해서 자원을 지정하고 해당 자원의 정보에 대한 조작을 요청한다.

## 2) 행위(Verb): HTTP Method

- HTTP 프로토콜의 Method를 사용한다.
- HTTP 프로토콜은 GET, POST, PUT, DELETE 와 같은 메서드를 제공한다.

## 3) 표현(Representation of Resource)

- Client가 자원의 상태(정보)에 대한 조작을 요청하면 Server는 이에 적절한 응답 (Representation)을 보낸다.
- REST에서 하나의 자원은 JSON, XML, TEXT, RSS 등 여러 형태의 Representation으로 나타내어 질 수 있다.

# REST API 설계 예시

## ■ Resource

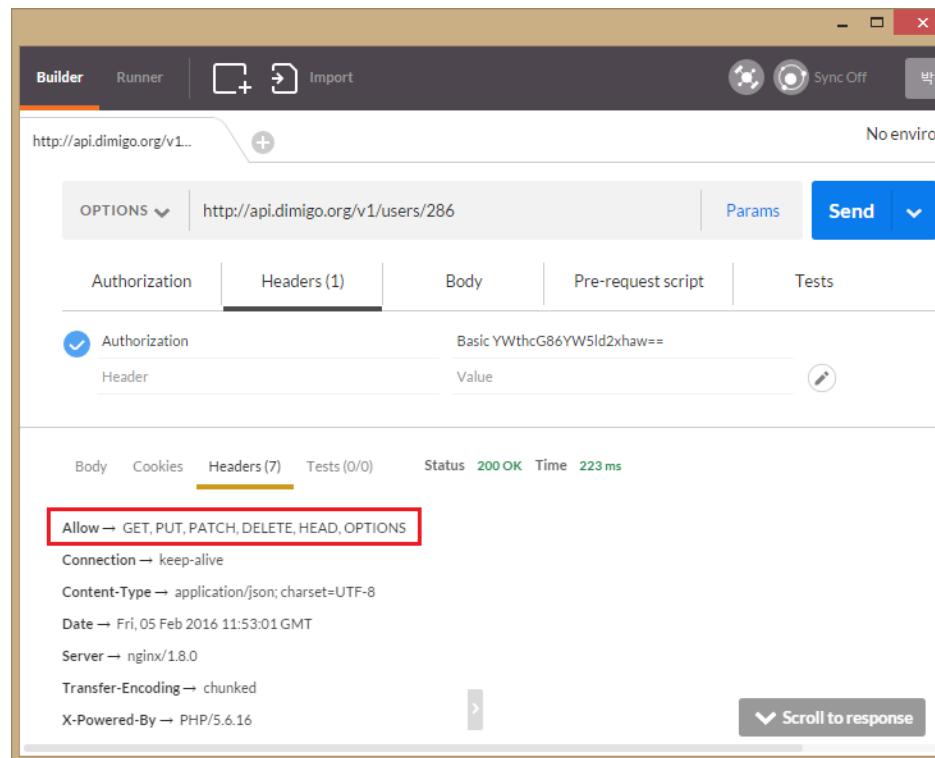
- URI 형태로 표현 (<http://api.cberi.shop/users>)
- 자원을 명사로 표현
- 세부 자원은 id를 추가 (<http://api.cberi.shop/users/2>)
- Collection URI vs Element URI

| Resource                                                                                                      | GET                                 | PUT                    | POST                                             | DELETE            |
|---------------------------------------------------------------------------------------------------------------|-------------------------------------|------------------------|--------------------------------------------------|-------------------|
| Collection URI, such as<br><a href="http://example.com/resources/">http://example.com/resources/</a>          | 컬렉션에 속한 자원들의 URI나 그 상세사항의 목록을 보여준다. | 전체 컬렉션은 다른 컬렉션으로 교체한다. | 해당 컬렉션에 속하는 새로운 자원을 생성한다. 자원의 URI는 시스템에 의해 할당된다. | 전체 컬렉션을 삭제한다.     |
| Element URI, such as<br><a href="http://example.com/resources/item17">http://example.com/resources/item17</a> | 요청한 컬렉션 내 자원을 반환한다.                 | 해당 자원을 수정한다.           | 해당 자원에 귀속되는 새로운 자원을 생성한다.                        | 해당 컬렉션내 자원을 삭제한다. |

# REST

## ■ HTTP Method

- OPTIONS: 각 Resource URI 별 허용되는 HTTP 메소드들은 해당 URI에 OPTIONS 메소드로 쿼리하여 알아낼 수 있음.
- 응답 헤더의 Allow를 확인.



# Postman API platform

A powerful GUI platform to make your API development faster & easier, from building API requests through testing, documentation and sharing.

The screenshot shows the Postman API Platform interface. At the top, there is a navigation bar with links for Product, Pricing, Enterprise, Resources and Support, and a search bar labeled "Search Postman". On the right side of the header are "Sign In" and "Sign Up for Free" buttons. Below the header, a yellow banner reads "Debug APIs together". The main content area is titled "Explore" and shows "Twitter API v2 / Tweet Lookup / Single Tweet". The request method is "GET" and the URL is "https://api.twitter.com/2/tweets/:id". The "Query params" section contains "tweet.fields" with a description of "Comma-separated list of fields for the tweet object. Expanded with attachments, author\_id, context\_annotations, conversation\_id, expansions, media.fields, poll.fields, place.fields, and user.fields". The "Path Variables" section has a "id" variable set to "1403216129661628420" with the note "Required. Enter a single Tweet ID". The "Body" section shows a JSON response with the following content:

```
1 {
2 "data": {
3 "id": "1403216129661628420",
4 "text": "Donovan Mitchell went down after a collision with Paul George towards the end of game 2. https://t.co/Y9ihXh0LDN"
5 }
}
```

On the right side of the interface, there is a sidebar with sections for "Documentation", "Authorization", "Request params", and "Allowed values". The "Documentation" section links to <https://api.twitter.com/2/tweets/:id>. The "Authorization" section notes "Bearer token" and "This request is using an authorization help collection Twitter API v2". The "Request params" section shows "tweet.fields" with a description of "Comma-separated list of fields for the tweet object." and "Allowed values" including "attachments,author\_id,context\_annotations,conversation\_id,expansions,media.fields,poll.fields,place.fields,user.fields". The "Path Variables" section shows "id" with "Default values: id;text". The bottom of the interface includes a "Find and Replace" and "Console" bar, along with download links for the desktop app (Windows, Mac, Linux).

# REST API TOOL 다운로드

- [https://www.postman.com/downloads/?utm\\_source=postman-home](https://www.postman.com/downloads/?utm_source=postman-home)

The screenshot shows the Postman download page and a preview of the Postman application interface.

**Postman Download Page:**

- The title bar says "Download Postman | Get Started".
- The address bar shows "postman.com/downloads/?utm\_source=postman-home".
- The menu bar includes "Product", "Pricing", "Enterprise", "Resources and support", "Explore", "Sign In", and "Sign Up for Free".
- The main content area features the heading "Download Postman" and the text: "Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman."
- A section titled "The Postman app" contains the text: "Download the app to get started with the Postman API Platform." and a large orange button labeled "Windows 64-bit".
- At the bottom, there's a note: "By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#)".

**Postman Application Preview:**

- The application has a top navigation bar with "Home", "Workspaces", "API Network", "Explore", and a search bar.
- The workspace "Twitter's Public Workspace" is selected, showing a "Collections" panel with "Twitter API v2" expanded, containing "Twitter Lookup" and two "GET Single Tweet" requests.
- The main request editor shows a "GET" request to "https://api.twitter.com/2/tweets/:id".
- Below the request editor, tabs for "Params", "Authorization", "Headers", "Body", "Pre-request Scripts", and "Tests" are visible.

# python & django 설치 (from docker image)

## ■ 준비

```
docker pull python:latest
mkdir /srv/django
mkdir /srv/django/build
cd /srv/django/build
```

## ■ 필요 파일 패키지 정의

```
vim requirements.txt
```

```
Django
djangorestframework
Markdown
django-filter
mysqlclient
psycopg2
```

- mysqlclient : MySQL 접속용
- psycopg2: PostgreSQL 접속용

## ■ docker image 빌드

```
vim Dockerfile

FROM python:latest
ENV PYTHONUNBUFFERED 1

WORKDIR /usr/src/app
COPY requirements.txt ./
RUN pip install -r requirements.txt

EXPOSE 8000
CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]

docker build --tag django-cberi .
```

# python & django 설치 (from docker image)

## ■ pip 란?

- 파이썬(python)으로 작성된 패키지 소프트웨어를 설치 · 관리하는 패키지 관리 시스템
- Python Package Index (PyPI)에서 많은 파이썬 패키지를 찾을 수 있음
- 2020년 1월 1일자로 파이썬 2의 지원이 종료
- 패키지 설치 및 제거 명령어
  - pip install [some-package-name]
  - pip uninstall [some-package-name]

```
[root@cberi django]# pip -V
pip 22.3.1 from /usr/local/lib/python3.11/site-packages/pip (python 3.11)
```

```
[root@cberi django]# pip list
Package Version

asgiref 3.6.0
Django 4.1.5
django-filter 22.1
djangorestframework 3.14.0
Markdown 3.4.1
mysqlclient 2.1.1
pip 22.3.1
psycopg2 2.9.5
pytz 2022.7.1
setuptools 65.5.1
sqlparse 0.4.3
wheel 0.38.4
```

# python & django 설치 (from docker image)

## 1. 기본 설치된 python3 이용

- python 명령어 (django는 별도 설치)

```
[root@cberi ~]# cd /etc/profile.d/
[root@cberi profile.d]# vim custom_alias.sh

alias python=python3
alias pip=pip3

[root@cberi profile.d]# source /etc/profile
```

## 2. docker 이미지를 이용하는 방법

### 1) python 명령어 생성

```
touch /usr/local/bin/python
chmod +x /usr/local/bin/python
vim /usr/local/bin/python

#!/bin/bash
docker run -it --rm --privileged --name python-run \
-v "$PWD":/usr/src/app -w /usr/src/app django-cberi \
python $@
```

### 2) pip 명령어 생성

```
touch /usr/local/bin/pip
chmod +x /usr/local/bin/pip
vim /usr/local/bin/pip

#!/bin/bash
docker run -it --rm --privileged --name pip-run \
-v "$PWD":/usr/src/app -w /usr/src/app django-cberi \
pip $@
```

### 3) django-admin 명령어 생성

```
touch /usr/local/bin/django-admin
chmod +x /usr/local/bin/django-admin
vim /usr/local/bin/django-admin

#!/bin/bash
docker run -it --rm --privileged \
--name django-admin-run \
-v "$PWD":/usr/src/app \
-w /usr/src/app \
django-cberi django-admin $@
```

# django REST API

## ■ Project setup

```
[root@cberi build]# adduser django
[root@cberi build]# su - django
[django@cberi ~]$ chmod +x /home/django/
[django@cberi ~]$ mkdir /home/django/dapi
[django@cberi ~]$ cd dapi/
$ django-admin startproject tutorial .
$ cd dapi
$ django-admin startapp quickstart
$ cd ..
$ pwd
/home/django/dapi

$ python manage.py migrate
$ python manage.py createsuperuser \
 --email admin@example.com --username admin
Password:
Password (again):
Superuser created successfully.
```

## ■ 셋업 후 폴더 구조

```
[root@cberi dapi]$ find .
.
./manage.py
./tutorial
./tutorial/quickstart
./tutorial/quickstart/tests.py
./tutorial/quickstart/models.py
./tutorial/quickstart/migrations
./tutorial/quickstart/migrations/__init__.py
./tutorial/quickstart/__init__.py
./tutorial/quickstart/apps.py
./tutorial/quickstart/admin.py
./tutorial/quickstart/views.py
./tutorial/__init__.py
./tutorial/settings.py
./tutorial/asgi.py
./tutorial/urls.py
./tutorial/wsgi.py
```

# django REST API

## ■ Serializer

```
$ vi tutorial/quickstart/serializers.py

from django.contrib.auth.models import User, Group
from rest_framework import serializers

class
UserSerializer(serializers.HyperlinkedModelSerializer):
 class Meta:
 model = User
 fields = ['url', 'username', 'email', 'groups']

class
GroupSerializer(serializers.HyperlinkedModelSerializer):
 class Meta:
 model = Group
 fields = ['url', 'name']
```

## ■ View

```
$ vi tutorial/quickstart/views.py

from django.contrib.auth.models import User, Group
from rest_framework import serializers

class
UserSerializer(serializers.HyperlinkedModelSerializer):
 class Meta:
 model = User
 fields = ['url', 'username', 'email', 'groups']

class
GroupSerializer(serializers.HyperlinkedModelSerializer):
 class Meta:
 model = Group
 fields = ['url', 'name']
```

# django REST API

## ■ URL

```
$ vi tutorial/urls.py

from django.urls import include, path
from rest_framework import routers
from tutorial.quickstart import views

router = routers.DefaultRouter()
router.register(r'users', views.UserViewSet)
router.register(r'groups', views.GroupViewSet)

Wire up our API using automatic URL routing.
Additionally, we include login URLs for the browsable
API.

urlpatterns = [
 path('', include(router.urls)),
 path('api-auth/', include('rest_framework.urls',
 namespace='rest_framework'))
]
```

## ■ Pagenation

```
$ vi tutorial/settings.py

:
REST_FRAMEWORK = {
 'DEFAULT_PAGINATION_CLASS':
 'rest_framework.pagination.PageNumberPagination',
 'PAGE_SIZE': 10
}
:
```

## ■ Settings

```
:
INSTALLED_APPS = [
 ...
 'rest_framework',
]
:
```

# django REST API

## ■ django app 구동

```
[root@cberi dapi]$ exit
[root@cberi dapi]# docker run -d --privileged \
--name dapi \
--network="nginx-proxy" \
-v /home/django/dapi/:/usr/src/app \
-w /usr/src/app \
-e "VIRTUAL_HOST=dapi.cberi.shop" \
-e "LETSENCRYPT_HOST=dapi.cberi.shop" \
-e "VIRTUAL_PORT=8000" \
-e TZ=Asia/Seoul \
django-cberi
```

## ■ 추가 설정

```
vi tutorial/settings.py
:
ALLOWED_HOSTS = ['*']
:
```

## ■ 사용자 인증 기능 미사용

```
vi tutorial/quickstart/views.py
class UserViewSet(viewsets.ModelViewSet):
 """
 API endpoint that allows users to be viewed or edited.
 """
 queryset = User.objects.all().order_by('-date_joined')
 serializer_class = UserSerializer
 # permission_classes = [permissions.isAuthenticated]

class GroupViewSet(viewsets.ModelViewSet):
 """
 API endpoint that allows groups to be viewed or edited.
 """
 queryset = Group.objects.all()
 serializer_class = GroupSerializer
 # permission_classes = [permissions.isAuthenticated]
```

# django REST API (과정 생략 버전)

## ■ Project setup

```
mkdir /home/django
chown www.www /home/django
su - www
$ cd /home/django
$ wget http://linux.datahub.pe.kr/dapi.zip
$ unzip dapi.zip
$ exit
$ tree -L 2 dapi/
dapi/
├── db.sqlite3
├── manage.py
└── tutorial
 ├── __init__.py
 ├── __pycache__
 ├── asgi.py
 ├── quickstart
 ├── settings.py
 ├── urls.py
 └── wsgi.py
```

## ■ django app 구동

```
docker run -d --privileged \
--name dapi \
--network="nginx-proxy" \
-v /home/django/dapi:/usr/src/app \
-w /usr/src/app \
-e "VIRTUAL_HOST=dapi.cberi.shop" \
-e "LETSENCRYPT_HOST=dapi.cberi.shop" \
-e "VIRTUAL_PORT=8000" \
-e TZ=Asia/Seoul \
django-cberi
```

# sqlite 설치

## ■ dockerhub 검색

- 1) command line tool
- 2) database browser

## ■ sqlite 강좌 사이트

■ <https://wikidocs.net/book/1530>

## ■ docker 이미지 다운로드

```
docker pull nochka/sqlite3
```

Docker Hub search results for "sqlite".

1 - 25 of 955 results for sqlite.

Best Match

Filters

Products

- Images
- Extensions
- Plugins

Trusted Content

- Docker Official Image
- Verified Publisher
- Sponsored OSS

Operating Systems

- Linux
- Windows

Architectures

- ARM
- ARM 64
- IBM POWER
- IBM Z
- PowerPC 64 LE
- x86
- x86-64

1 - 25 of 955 results for sqlite.

linuxserver/sqlitebrowser SPONSORED OSS 100K+ 24 Downloads Stars

By linuxserver.io • Updated 4 days ago

Linux x86-64 arm arm64

Azureiotedge SQLite/undefined 0 Stars

By Microsoft • Updated 37 minutes ago

An Azure IoT Edge module capable of accessing sqlite databases.

x86-64

nouchka/sqlite3 100K+ 42 Downloads Stars

By nouchka • Updated 3 days ago

Sqlite3 command line in a docker container

Linux x86-64

sqlitebrowsered/sqlitebrowser 1.1K 1 Downloads Star

By sqlitebrowsered • Updated 2 years ago

Linux x86-64

coleifer/sqlite-web 100K+ 9 Downloads Stars

By coleifer • Updated 3 years ago

Docker image for running web-based SQLite database browser.

Linux x86-64

# sqlite 설치

- command line tools 설치
- image에서 정보 확인

## Use

Use from command line:

```
docker run -it nouchka/sqlite3
```

or use with docker compose:

```
docker-compose run sqlite3
```

- 공식 사이트에 건질 만한 정보가 없음

```
[root@cberi srv]# docker image inspect nouchka/sqlite3
[
 {
 "Id": "83e8c76e66310a42354e6dfe6c099adb5edf876759ab46c1760595f42d5f503",
 "Digest": "sha256:20ad54a17b3bae5b76eb8023d6766ca08363d7b8153dc289e846",
 "RepoTags": [
 "docker.io/nouchka/sqlite3:latest"
],
 "Parent": "",
 "Comment": "",
 "Created": "2023-01-26T14:07:02.706487955Z",
 "Config": {
 "Env": [
 "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin"
],
 "Entrypoint": [
 "sqlite3"
],
 "WorkingDir": "/root/db",
 "Labels": {}
 }
 }
]
```

# sqlite 설치

## ■ sqlite 명령어 생성

```
touch /usr/local/bin/sqlite
chmod +x /usr/local/bin/sqlite
vim /usr/local/bin/sqlite

docker run -it --rm \
 --name sqlite-run \
 -v "$PWD":/root/db:z \
 nouchka/sqlite3 $@
```

## ■ 테스트

```
cd ~
mkdir db && cd db
sqlite3 test.db
SQLite version 3.34.1 2021-01-20 14:10:07
Enter ".help" for usage hints.
sqlite> .database
main: /root/db/test.db r/w
sqlite> .tables
sqlite> CREATE TABLE "auth_group" (
 "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
 "name" varchar(150) NOT NULL UNIQUE);
sqlite> .tables
auth_group
sqlite> INSERT INTO "auth_group" VALUES(NULL, 'guest');
sqlite> SELECT * FROM auth_group;
1|guest
sqlite> .exit
[root@cberi sqlite]# ls -lph
합계 16K
-rw-r--r--. 1 root root 16K 1월 29 17:10 test.db
```

# sqlite-web 설치

The screenshot shows the Docker Hub page for the repository `coleifer/sqlite-web`. The page includes the repository name, author, update time, description, and a link to the Dockerfile. Below the Dockerfile link, there is a command example for running the container.

```
docker run -it --rm \
-p 8080:8080 \
-v /path/containing/database:/data \
-e SQLITE_DATABASE="db_filename.db" \
coleifer/sqlite-web
```

## ■ 파라미터 커스터 마이징

```
docker run -d \
--name sqlite-web \
--network="nginx-proxy" \
-v /home/django/dapi:/data:z \
-e SQLITE_DATABASE="db.sqlite3" \
-e "VIRTUAL_HOST=sqlite.cberi.shop" \
-e "LETSENCRYPT_HOST=sqlite.cberi.shop" \
-e "VIRTUAL_PORT=8080" \
-e TZ=Asia/Seoul \
coleifer/sqlite-web
```

## ■ docker container 실행

SQLite Web: db.sqlite3

https://sqlite.cberi.shop

new\_table\_name Create

| table name...              |  |  |
|----------------------------|--|--|
| auth_group                 |  |  |
| auth_group_permissions     |  |  |
| auth_permission            |  |  |
| auth_user                  |  |  |
| auth_user_groups           |  |  |
| auth_user_user_permissions |  |  |
| django_admin_log           |  |  |
| django_content_type        |  |  |
| django_migrations          |  |  |
| django_session             |  |  |
| sqlite_sequence            |  |  |

[Toggle helper tables](#)

|                 |                                     |
|-----------------|-------------------------------------|
| <b>Sqlite</b>   | 3.32.3                              |
| <b>Filename</b> | /data/db.sqlite3                    |
| <b>Size</b>     | 135.2 kB                            |
| <b>Created</b>  | Sunday January 14, 2024 at 11:07:AM |
| <b>Modified</b> | Sunday January 14, 2024 at 10:31:AM |
| <b>Tables</b>   | 11                                  |
| <b>Indexes</b>  | 18                                  |
| <b>Triggers</b> | 0                                   |
| <b>Views</b>    | 0                                   |

Web-based SQLite database browser, powered by [Flask](#) and [Peewee](#). © 2024 Charles Leifer

---

---

# 질의 응답

