

Web Server

Apache Web Server

❖ Apache Web Server

- ✓ 웹의 역사와 함께해온 가장 오래되고 검증된 웹 서버
- ✓ 프로세스/스레드 기반 구조를 사용하며 .htaccess 파일을 통해 디렉토리별로 설정을 세밀하게 제어할 수 있음
- ✓ PHP 기반 웹사이트(WordPress 등) 나 복잡한 모듈 설정이 필요한 환경
- ✓ 수 많은 모듈이 존재하여 확장성이 뛰어나고 관련 자료가 매우 풍부

Apache Web Server

❖ Apache Web Server 설치 및 활성화

✓ 설치: apache2

```
user1@myubuntu:~$ sudo apt install apache2
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
(생략)
다음 새 패키지를 설치할 것입니다:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
0개 업그레이드, 9개 새로 설치, 0개 제거 및 0개 업그레이드 안 함.
1,610 k바이트 아카이브를 받아야 합니다.
이 작업 후 6,599 k바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n]
(생략)
Processing triggers for systemd (234-2ubuntu12.1) ...
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for ufw (0.35-5) ...
user1@myubuntu:~$
```

Apache Web Server

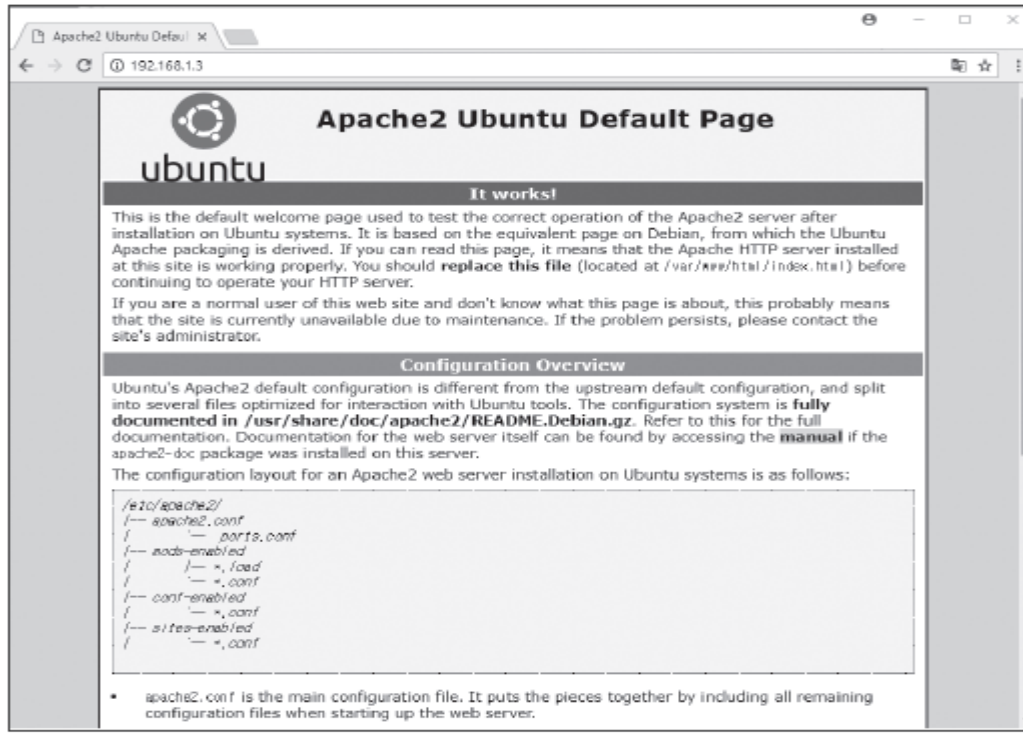
- ❖ Apache Web Server 설치 및 활성화
 - ✓ 서비스의 이름은 apache2

```
user1@myubuntu:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Sat 2017-11-25 15:38:49 KST; 39s ago
 Main PID: 22121 (apache2)
    Tasks: 55 (limit: 19660)
   CGroup: /system.slice/apache2.service
           └─22121 /usr/sbin/apache2 -k start
             └─22123 /usr/sbin/apache2 -k start
               └─22124 /usr/sbin/apache2 -k start

11월 25 15:38:49 myubuntu systemd[1]: Starting The Apache HTTP Server...
11월 25 15:38:49 myubuntu apachectl[22110]: AH00558: apache2: Could not reliably
11월 25 15:38:49 myubuntu systemd[1]: Started The Apache HTTP Server.
user1@myubuntu:~$ ps -ef | grep apache
root      22121      1  0 15:38 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  22123   22121  0 15:38 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  22124   22121  0 15:38 ?        00:00:00 /usr/sbin/apache2 -k start
user1     22329    5212  0 15:39 pts/2    00:00:00 grep --color=auto apache
user1@myubuntu:~$
```

Apache Web Server

- ❖ Apache Web Server 설치 및 활성화
 - ✓ 브라우저에서 확인 : <http://ip주소>



Apache Web Server

- ❖ Apache Web Server 설치 및 활성화
 - ✓ 웹 서버의 기본 디렉토리: /var/www/html

```
user1@myubuntu:~$ cd /var/www/html
user1@myubuntu:/var/www/html$ ls
index.html
user1@myubuntu:/var/www/html$ sudo vi my.html
<html>
<head>
<title>html test</title>
</head>
<body>
My First Web Page!!!
</body>
</html>

:wq
user1@myubuntu:/var/www/html$
```

Apache Web Server

❖ 일반 사용자 계정에 웹 설정

- ✓ 일반 사용자 계정에서 웹 서비스를 제공하려면 아파치 설정 파일을 수정해야 하는데 아파치 웹 서버의 설정 파일 중 사용자 계정의 웹 설정과 관련된 파일은 /etc/apache2/mods-available/userdir.conf 파일
- ✓ userdir.conf 파일 수정(sudo vi /etc/apache2/mods-available/userdir.conf)

UserDir public_html

#UserDir disabled root

<Directory /home/*/public_html>

AllowOverride FileInfo AuthConfig Limit Indexes

Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec

Require method GET POST OPTIONS

</Directory>

Apache Web Server

❖ 일반 사용자 계정에 웹 설정

✓ 심볼릭 링크 설정

- `sudo ln -s /etc/apache2/mods-available/userdir.conf /etc/apache2/mods-enabled/userdir.conf`
- `sudo ln -s /etc/apache2/mods-available/userdir.load /etc/apache2/mods-enabled/userdir.load`

✓ 아파치 재시작

- `sudo systemctl stop apache2`
- `sudo systemctl start apache2`

Apache Web Server

- ❖ 일반 사용자 계정에 웹 설정
 - ✓ 사용자의 홈 디렉토리에 public_html 디렉토리 생성
 - mkdir public_html
 - ✓ 디렉토리 접근 권한 변경
 - chmod 701 .
 - chmod 701 public_html

Apache Web Server

❖ 일반 사용자 계정에 웹 설정

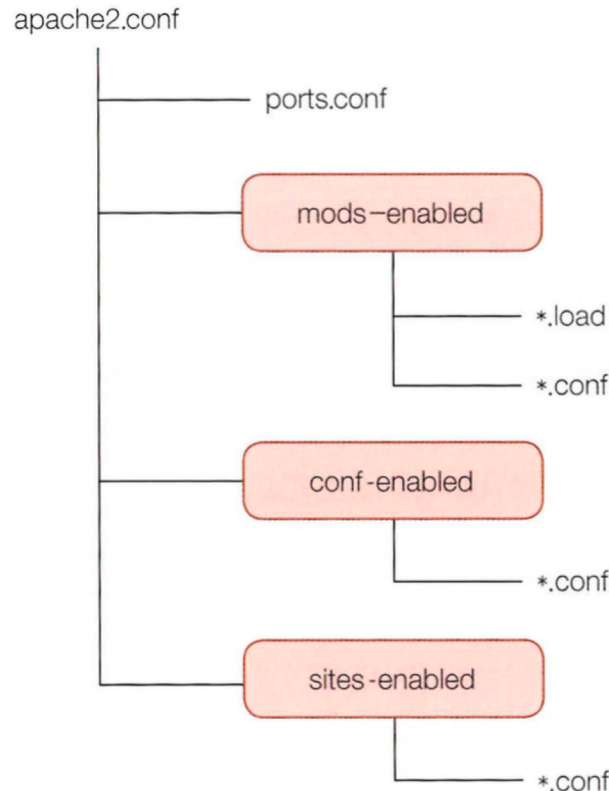
- ✓ public_html 디렉토리 안에 index.html 파일을 생성

```
<html>  
<head>  
<title>adam html test</title>  
</head>  
<body>  
Adam Web Page!!!  
</body>  
</html>
```

- ✓ http://ip/~계정 으로 접속해서 확인

Apache Web Server

- ❖ 웹 서버 설정 파일과 디렉토리
 - ✓ 설정 파일: /etc/apache2/apache2.conf
 - ✓ 설정 파일 포함 내역



Apache Web Server

❖ 웹 서버 설정 파일과 디렉토리

✓ 주요 의미

- ServerRoot `"/etc/apache2"` : 웹 서버의 설정 파일, 로그 파일 등이 저장되는 최상위 디렉터리
- Listen 80: 웹 서버의 포트 번호로 기본적으로 대부분의 웹 서버는 80을 사용(ports.conf)
- Include `conf.modules.d/*.conf`: 설정 파일에 포함될 파일의 경로와 파일 이름
- User `${APACHE_RUN_USER}`: 웹 서비스를 작동하는 사용자로 `envvars` 파일에 `www-data`로 설정
- Group `${APACHE_RUN_GROUP}`: 웹 서비스를 작동하는 그룹 설정으로 `envvars` 파일에 `www-data`로 설정
- MaxKeepAliveRequests 100: 처리할 수 있는 최대 요청 수
- `<Directory /var/www/> Option Indexes FollowSymLinks AllowOverride All Require all granted </Directory>`: 웹 서버는 디렉터리 단위로 설정하는데 각 디렉터리마다 허가/거부 등의 접근 권한을 다르게 할 수 있는데 Option Indexes FollowSymLinks의 의미는 현재 디렉터리에서 DirectoryIndex에 지정된 파일이 없다면 파일 목록을 대신 출력하라는 의미

nginx

❖ 개요

- ✓ 웹 서버 소프트웨어로 가볍고 빠르며 동시에 많은 접속자를 처리하는 데 특화된 도구
- ✓ 과거에는 Apache(아파치)가 시장을 지배했지만 현대의 웹 서비스는 수만 명의 사용자가 동시에 접속하는 경우가 많아지면서 이를 효율적으로 해결하기 위해 Nginx가 탄생함
- ✓ 주요 역할
 - 정적 웹 서버(Web Server): HTML, CSS, 이미지, JavaScript 파일 같은 정적 콘텐츠를 브라우저에 전달하는 역할로 Nginx는 메모리 사용량이 매우 적어 단순히 파일을 보내주는 작업에서 매우 높은 성능을 발휘
 - 리버스 프록시(Reverse Proxy): 클라이언트(사용자)와 백엔드 서버(Node.js, Python, Java 등) 사이에서 중개자 역할을 하는데 실제 서버의 IP 주소를 숨길 수 있어서 보안이 우수하며 자주 요청되는 데이터를 미리 저장해 두어 서버 부하를 줄임
 - 로드 밸런서(Load Balancer): 한 대의 서버에 트래픽이 몰리지 않도록 여러 대의 서버에 요청을 분산시켜주는 역할을 수행

nginx

❖ 비교

	Apache	Nginx
작동 방식	스레드/프로세스 기반	비동기 이벤트 기반
성능	대규모 동시 접속 시 메모리 부족	대규모 동시 접속에 매우 강함
유연성	다양한 모듈 제공 (확장성 좋음)	가볍고 성능 중심적

nginx

❖ 설치 및 확인

✓ 설치

- `sudo apt update`
- `sudo apt install nginx`

✓ 확인

- `systemctl status nginx`
- `curl http://localhost`



nginx

❖ 정적 웹 사이트 게시

✓ 디렉토리 생성

- `sudo mkdir -p /var/www/example.com/html`
- `sudo chown -R $USER:$USER /var/www/example.com/html`

✓ html 파일 작성

- `nano /var/www/example.com/html/index.html`
`<h1>Hello, Nginx on Ubuntu!</h1>`

nginx

❖ 정적 웹 사이트 게시

✓ 설정 파일 생성 및 작성

- /etc/nginx/sites-available/default 파일의 http 블록 안에 작성

```
server {  
    listen 80;  
    listen [::]:80;  
  
    root /var/www/example.com/html;  
    index index.html;  
  
    server_name example.com www.example.com;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

nginx

❖ 정적 웹 사이트 게시

✓ 설정 활성화 및 재시작

● 설정 오류 확인 후 재시작

```
sudo nginx -t
```

```
sudo systemctl reload nginx
```



nginx

❖ 리버스 프록시(Reverse Proxy) 설정

✓ 설정 변경

```
server {  
    listen 80;  
    server_name api.example.com;
```

```
    location / {  
        # 요청을 전달할 내부 서버 주소 (IP 또는 도메인)  
        # 모든 요청을 내부의 3000번 포트로 전달  
        proxy_pass http://localhost:3000;
```

```
        # 실제 접속자의 정보를 백엔드 서버에 전달하기 위한 헤더 설정  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;
```

```
    }  
}
```

nginx

❖ LoadBalancer 설정

✓ 설정 파일 수정

1. 백엔드 서버 그룹 정의

```
upstream backend_servers {  
    server 192.168.0.100:8000;  
    server 192.168.0.201:8000;  
    server 192.168.0.202:8000;  
}
```

```
server {  
    listen 80;  
    server_name example.com;
```

```
    location / {
```

```
        # 2. 위에서 정의한 upstream 그룹으로 요청 전달  
        proxy_pass http://backend_servers;
```

```
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;
```

```
    }
```

```
}
```

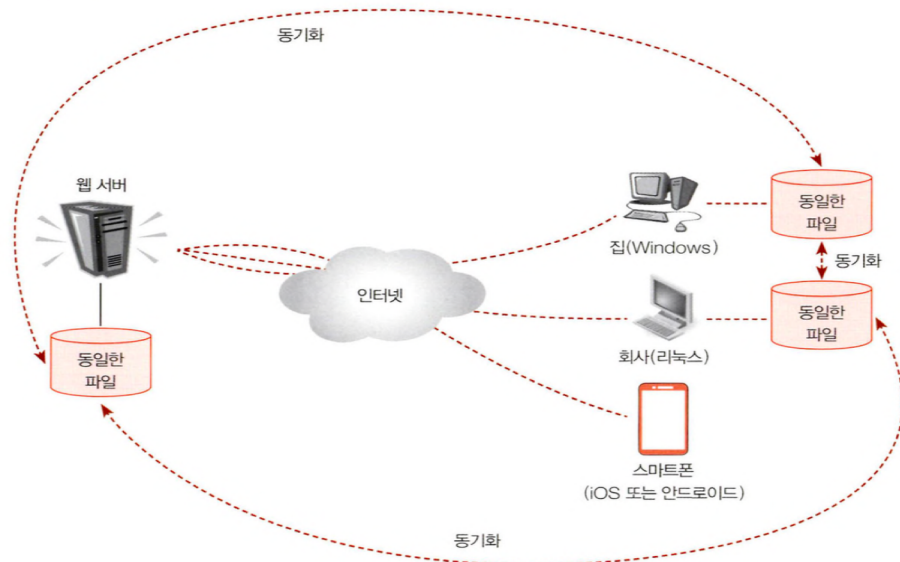
웹 서버 응용 서비스

❖ 클라우드 서비스

✓ 클라우드 서비스 구축

● 개요

- ◆ 예전에는 Web Hard라 불리는 웹 응용 서비스가 많이 사용되었지만 웹 브라우저로만 접속한다는 한계점이 있기 때문에 최근 들어서는 인기가 예전 같지 않음
- ◆ 대신 요즘에는 클라우드 저장소가 많이 활용되고 있는데 클라우드 저장소는 대표적으로 네이버의 N드라이브, 구글 드라이브, Microsoft OneDrive 등이 있는데 기존의 웹하드 기능까지도 포함하고 있어 별도의 웹하드가 필요 없음



웹 서버 응용 서비스

❖ 클라우드 서비스

✓ 클라우드 서비스 구축

● 구축

- ◆ 클라우드 서비스 기능을 제공하는 오픈 소스인 ownCloud 커뮤니티 에디션을 설치하기 위해서는 LAPM 환경이 구성되어야 함
- ◆ 기존 apache2 패키지를 제거하고 다시 설치
 - 아파치 서비스 중단: `sudo systemctl stop apache2`
 - 패키지 및 설정 파일 삭제: `sudo apt-get purge apache2 apache2-utils apache2-bin apache2.2-common`
 - 의존성 및 불필요한 패키지 자동 삭제: `sudo apt-get autoremove --purge`
 - 남은 디렉토리 수동 삭제
 - `sudo rm -rf /etc/apache2`
 - `sudo rm -rf /usr/lib/apache2`
 - `sudo rm -rf /var/log/apache2`
 - apache2 설치
 - `sudo apt -y install apache2`
 - `sudo systemctl restart apache2`
 - `sudo systemctl enable apache2`

웹 서버 응용 서비스

❖ 클라우드 서비스

✓ 클라우드 서비스 구축

● 구축

◆ 기존 Maria DB 삭제

- `sudo systemctl stop mariadb`
- `sudo apt-get purge mariadb-server mariadb-client mariadb-common mariadb-server-core-* mariadb-client-core-*`
- `sudo apt-get autoremove`
- `sudo apt-get autoclean`
- `sudo rm -rf /var/lib/mysql`
- `sudo rm -rf /etc/mysql`
- `sudo rm -rf /var/log/mysql`

◆ Maria DB 설치 및 초기화

- `sudo apt -y install mariadb-server mariadb-client`
- `sudo systemctl restart mariadb`
- `sudo systemctl enable mariadb`
- `sudo mysql_secure_installation` 명령으로 데이터베이스 초기화

웹 서버 응용 서비스

❖ 클라우드 서비스

✓ 클라우드 서비스 구축

● 구축

◆ 데이터베이스 및 유저 생성

- 접속: `sudo mysql -u root -p`
- `CREATE DATABASE cloud_db;`
- `CREATE USER 'cloud'@'localhost' IDENTIFIED BY '1234';`
- `GRANT ALL PRIVILEGES ON cloud_db.* TO 'cloud'@'localhost';`
- `FLUSH PRIVILEGES;`

◆ 방화벽 설정

- `sudo ufw allow 80`
- `sudo ufw allow 443`
- `sudo ufw allow 3306`

웹 서버 응용 서비스

❖ 클라우드 서비스

✓ 클라우드 서비스 구축

● 구축

◆ PHP 저장소 추가

- `sudo apt update && sudo apt upgrade -y`
- `sudo apt install -y software-properties-common`
- `sudo add-apt-repository ppa:ondrej/php -y`
- `sudo apt update`

◆ PHP 7 버전 설치 – 8 버전 아직 지원 안함

- `sudo apt install -y php7.4 php7.4-gd php7.4-mysql php7.4-curl ₩`
- `php7.4-xml php7.4-zip php7.4-intl php7.4-mbstring ₩`
- `php7.4-common libapache2-mod-php7.4`

웹 서버 응용 서비스

❖ 클라우드 서비스

✓ 클라우드 서비스 구축

● 구축

◆ bz2 압축 해제를 위한 패키지 설치: `sudo apt install bzip2 -y`

◆ owncloud 패키지 설치

- `cd /var/www/`
- `sudo wget https://download.owncloud.com/server/stable/owncloud-complete-latest.tar.bz2`
- `sudo tar -xjf owncloud-complete-latest.tar.bz2`
- `sudo chown -R www-data:www-data owncloud/`
- `sudo chmod -R 755 owncloud/`

웹 서버 응용 서비스

❖ 클라우드 서비스

✓ 클라우드 서비스 구축

● 구축

- ◆ owncloud 가상 호스트 설정: `sudo nano /etc/apache2/sites-available/owncloud.conf`

```
<VirtualHost *:80>
    DocumentRoot /var/www/owncloud
    Alias /owncloud "/var/www/owncloud/"

    <Directory /var/www/owncloud/>
        Options +FollowSymlinks
        AllowOverride All

        <IfModule mod_dav.c>
            Dav off
        </IfModule>

        SetEnv HOME /var/www/owncloud
        SetEnv HTTP_HOME /var/www/owncloud
    </Directory>
</VirtualHost>
```

웹 서버 응용 서비스

❖ 클라우드 서비스

✓ 클라우드 서비스 구축

● 구축

◆ 설정 적용 및 서버 재시작

- `sudo a2ensite owncloud.conf`
- `sudo a2enmod rewrite headers env dir mime`
- `sudo systemctl restart apache2`

◆ 위 설정 만으로 안되는 경우 apache2 재설정

- `sudo a2dissite 000-default.conf`
- `sudo a2ensite owncloud.conf`
- `sudo systemctl restart apache2`

웹 서버 응용 서비스

❖ 클라우드 서비스

✓ 클라우드 서비스 구축

● 구축

◆ Web UI에서 관리자 계정 설정: <http://IP/owncloud>

관리자 계정 만들기

사용자 이름

사용자 이름

암호

암호

저장소 및 데이터베이스 ▾

데이터 폴더

/var/www/owncloud/data

데이터베이스 설정

MySQL/MariaDB만 사용할 수 있습니다. 다른 데이터베이스 종류를 사용하려면 추가 PHP 모듈을 설치 및 활성화하십시오.

자세한 정보를 보려면 문서를 참고하십시오. ↗

데이터베이스 사용자

데이터베이스 사용자

데이터베이스 암호

데이터베이스 암호

데이터베이스 이름

데이터베이스 이름

데이터베이스 호스트

localhost

웹 서버 응용 서비스

❖ 클라우드 서비스

✓ 클라우드 서비스 구축

● 사용자 생성

+ 그룹 추가

모두 3

관리자 0

users 2

사용자 이름	이메일	그룹	만들기			
사용자 이름	전체 이름	암호	그룹	다음 그룹의 관리자:	할당량	
 admin	admin	●●●●●●●●	admin ▼	그룹 없음 ▼	기본값 ▼	
 ubuntu	ubuntu	●●●●●●●●	users ▼	그룹 없음 ▼	1 GB ▼	
 itstudy	itstudy	●●●●●●●●	users ▼	그룹 없음 ▼	1 GB ▼	