

1. 리눅스 (Linux)

1.1. 개요

리눅스는 Unix를 기반으로 하는 오픈소스 운영체제입니다. 리눅스는 안정적이고, 보안적으로 뛰어나며 성능이 극단적으로 제한된 임베디드 시스템부터, 여러 대의 물리적인 컴퓨터를 묶어 동작하는 슈퍼컴퓨터까지 거의 모든 상황에서 유연하게 사용할 수 있어 서버 또는 인프라스트럭처 구축시 보편적으로 사용됩니다.

1.2. 기본 명령어

- `ls`
현재 위치에 존재하는 파일, 폴더 리스트를 보여줍니다.
- `cd`
지정된 위치로 이동합니다.
- `mkdir`
새로운 폴더를 생성합니다.
- `rm`
파일 또는 폴더를 삭제합니다.
- `pwd`
현재 경로를 출력합니다.
- `which`
명령어 뒤에 입력된 명령어의 파일 위치를 출력합니다.
- `sudo`
명령어 뒤에 입력된 명령어를 관리자 권한으로 실행합니다.
- `$ ()`
() 내부에 있는 명령어를 먼저 수행하고, 그 결과값을 사용합니다.

2. 우분투 (Ubuntu)

2.1. 개요

우분투는 리눅스의 가장 대표적인 배포판입니다. 우분투는 일반 사용자를 타겟으로 개발되었기에, 사용자 편의를 위한 유려한 GUI와 다양한 도구가 내장되어 있어 간편하게 사용할 수 있습니다. 그 덕에 우분투는 현재(2023년) 일반 사용자가 가장 많은 리눅스 배포판이 되었습니다.

우분투는 개인 사용자용 리눅스 배포판 중 가장 인기가 많기에, 어떤 문제가 생겼을 때 그 문제를 해결하기 위한 솔루션을 찾기 쉽습니다. 그래서 만약 리눅스를 처음 사용한다면 우분투를 사용해 리눅스에 익숙해진 후, 자신에 맞는 다른 배포판을 사용해 보는걸 추천합니다.

2.2. 패키지 설치

리눅스는 기본적으로 패키지 관리자를 사용해 새로운 패키지를 설치하거나, 삭제합니다. 우분투는 데비안 계열의 리눅스 배포판이기 때문에 `deb` 를 사용한 수동 패키지 관리 또한 가능하지만, 우분투가 자체적으로 제공하는 `apt` 를 사용하면, 시스템에 맞는 패키지를 원격 패키지 저장소에서 자동으로 다운로드 받아 설치할 수 있습니다.

`apt` 는 기본적으로 다음과 같이 사용합니다.

```
sudo apt-get install package-name
```

위 코드에서 사용한 `apt-get` 은 대부분의 상황에서 `apt` 와 동일하게 동작합니다. 다만 `apt` 는 사용자 편의를 위한 `apt-get` 에서 몇 가지 기능이 추가되어 있습니다.

하지만, 위 명령어를 입력 했을 때 모든 명령어가 정상적임에도 `Package not found` 오류가 발생할 수 있습니다. 그 이유는 로컬 컴퓨터의 `apt` 패키지 목록이 업데이트 되지 않았기 때문입니다. 이 오류를 해결하려면 다음 줄을 입력하고 기다리면 됩니다.

```
sudo apt-get update
```

Tip.

우분투 기본 저장소가 해외로 지정되어 있는 경우, 패키지 다운로드시 속도가 매우 느릴 수 있습니다. 이런 경우 구글에 "Ubuntu mirror kakao"로 검색하여 카카오의 패키지 저장소를 사용하는 방법을 찾아 적용해보세요.

2.3. 방화벽 (UFW)

인프라스트럭처 또는 서버 구축시 모든 설정이 정상인데도 네트워크 연결이 안되는 경우가 있습니다. 만약 그런 상황을 마주한다면 가장 먼저 확인해 보아야 하는 것 중 하나가 바로 어플리케이션이 구동중인 서버의 방화벽 설정입니다.

방화벽이란 시스템의 일부분만을 외부에 공개하고, 나머지를 차단함으로써 시스템을 격리하고 보호하기 위한 도구입니다. 우분투는 과거 사용되던 `iptables` 를 더 간편하게 사용할 수 있도록 `ufw` 라는 도구를 제공합니다. `ufw` 의 모든 명령어 옵션을 보기 위해서는 아래 명령어를 실행하면 됩니다.

```
# 설치
sudo apt-get update
sudo apt-get install ufw

# UFW 모든 명령어 옵션
sudo ufw help
```

`ufw` 를 사용하는데 가장 기본적이고, 중요한 것은 방화벽을 활성화 / 비활성화 하는 것입니다. 이에 해당하는 명령어는 아래와 같습니다.

```
# UFW 방화벽 활성화
sudo ufw enable

# UFW 방화벽 비활성화
sudo ufw disable

# UFW 방화벽 활성 / 비활성 상태 확인
sudo ufw status
```

활성 / 비활성 상태를 확인했다면, 이제는 특정 포트와 프로토콜에 대해 방화벽 설정을 하는 법을 알아야 합니다.

```
# allow는 허용을 의미합니다.
sudo ufw allow <port>/<protocol>

# protocol은 TCP와 UDP가 있으며, 생략한다면 둘 모두에게 적용됩니다.
sudo ufw allow <port>

# deny는 차단을 의미하며, 사용법은 allow와 동일합니다.
sudo ufw deny <port>/<protocol>
sudo ufw deny <port>
```

만약 포트 번호가 동적인 서비스가 있다면, 서비스 명을 사용해 위와 동일하게 허용 / 차단을 할 수 있습니다.

```
sudo ufw allow <service-name>
sudo ufw deny <service-name>
```

보편적으로, 서버에서 웹 서비스를 제공하는 경우 `http` 프로토콜에서 사용하는 80 포트와, `https` 프로토콜에서 사용하는 443 포트를 다음과 같은 명령어로 허용합니다.

```
# http
sudo ufw allow 80

# https
sudo ufw allow 443

# 적용 확인
sudo ufw status
```

Tip.

최근 추가된 HTTP/3 사양의 경우 기존에 사용하던 TCP 프로토콜을 사용하지 않고, UDP 프로토콜을 사용하므로, TCP 프로토콜에 대해서만 `allow` 한다면 HTTP/3 의 사용이 불가능할 수 있습니다.

3. 엔진엑스 (Nginx)

3.1. 개요

엔진엑스는 잘 알려진 경량의 고성능 웹 서버입니다. 엔진엑스는 스레드 - 동기 방식으로 동작하는 아파치 (Apache)와 달리 이벤트 기반 비동기 방식으로 요청을 처리하여 비교적 적은 리소스로도 고성능을 낼 수 있습니다.

3.2. 설치

우분투에서 엔진엑스를 설치하는 방법은 간단합니다. 터미널에서 아래 명령어 입력하고 기다리면 모든 설치가 완료됩니다.

Tip.

설치 중 타임존 설정을 위해 타임존 번호를 입력해야 하는 부분이 있는데, 글 작성 시점(2023.01) 기준으로 KST를 선택하려면 6 (Asia), 69 (Seoul) 순서로 입력하면 됩니다.

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

3.3. 기본 명령어

- `nginx -v`
현재 시스템에 설치된 엔진엑스 버전을 표시합니다.
- `nginx -t`
. 엔진엑스 설정 파일을 테스트합니다. 만약 문법 오류를 비롯한 어떤 문제가 있다면 이 명령어를 사용해 확인할 수 있습니다.
- `nginx -s reload`
엔진엑스 서비스의 정지 또는 재시작 없이 엔진엑스 설정 파일을 불러와 적용합니다.
- `nginx -s stop`
엔진엑스 서비스를 정지합니다.
- `nginx -s quit`
엔진엑스 서비스를 종료하고, 프로세스에서 제거합니다.
- `nginx -s reopen`
엔진엑스 재시작시 로그 파일을 다시 엽니다.
- `nginx -c /path/to/nginx.conf`
특정 경로에 위치한 설정 파일을 적용해 엔진엑스를 시동합니다.
- `nginx -T`
현재 가동중인 엔진엑스 서비스의 설정 파일을 보여줍니다.

3.4. 설정

엔진엑스의 설정은 엔진엑스의 동작을 지정하는 것으로, 파일에 설정 값을 저장하여 적용합니다. 설정 파일은 `conf` 폴더 하위에 위치하는데, 엔진엑스 설치 방법(컴파일 , `apt-get` , ...)에 따라 설정 파일의 경로가 다를 수 있습니다. 때문에, 설치 후 아래 명령어를 사용하여 설정 파일의 위치를 확인해야 합니다.

```
sudo find / -name nginx.conf
```

설정 파일은 위치에 따른 부분 과 중괄호로 쌓인 블록 으로 이뤄져 있습니다. 가장 단순한 형태로 만들어진 설정 파일은 아래와 같습니다.

설명을 위해 작성한 예시이므로, 서비스에서 사용하지 말 것

```
# Core
worker_processes 1;

# Events
events {
    worker_connections 1024;
}

# Http
http {
    include mime.types;

    # Server
    server {
        listen 80;

        # Location
        location / {
            root html;
            index index.html index.htm;
        }
    }
}
```

3.4.1. core

설정 파일 최상단에 위치하면서, 엔진엑스의 기본적인 동작 방식을 정의하는 부분입니다. 코어 부분에서 사용하는 대표적인 지시어들은 다음과 같습니다.

- **user**
엔진엑스가 실행되는 권한. `root` 권한은 보안상 위험함
example : `www-data`
- **worker_processes**
엔진엑스가 사용할 프로세스의 개수. CPU의 물리적 코어 하나당 최소 1개의 프로세스가 배정되도록 변경 권

장

default : 1

- error_log <save-path> <log-level>

엔진엑스의 로그가 저장될 위치와 로그 레벨 설정

log level : debug - info - notice - warn - error - crit

3.4.2. http

3.4.3. server

3.4.5. location

3.4.6. events

3.5 포트포워딩
