AWS EC2로 Flask_API 구현하기

AWS를 사용한 이유

- 클라우드 컴퓨터를 제공하는 가장 대표적인 회사가 AWS , AZURE 가 있다고 나는 생각한다. 그런데 왜 AZURE를 안쓰고 AWS를 사용했냐 하면은 지표상 전세계 클라우드 시장에서 MS ,Google , IBM 보다 AWS의 시장 점유률이 10%나 높다는 기사가 있었다. 같은 기능 같은 서비스라고 생각했을때 시장 점유율이 높은 AWS을 사용하면 더 좋다고 생각했다. 시장 점유율 관련 기사

Flask를 사용한 이유

- 데이터 분석 프로젝트를 배포하는 것이 주 목적이다. Web Server는 Spring FrameWork를 사용해 구축 하였고 데이터 분석을 처리할 로직들은 외부 Flask_api에 작성했다. 외부 api를 사용한 이유는 Spring과 flask는 RESTfull 한 Framework라는 장점이 있기 때문에 무거운 분석 모델을 따로 호스팅 서버에 올리고 간단히 url요청을 통해 모델을 호출해서 결과값을 가져 올수 있고, 서버에 부하를 줄일 수 있기 때문에 외부 api로 구현 해 놓았다.

구현

- AWS에 EC2 인스턴스(가상머신)를 생성한다 스팩은 다음과 같다
  - Ubuntu 16.04
  - RAM 1GB -> free tier
  - Storage 8GB
- Command-Line 툴은 MobaXterm을 사용했다 왜냐하면 ubuntu 디렉토리를 GUI로 볼 수 있고, scp같이 파일을 호스트로 전송할때 터미널 작성이 아닌 **dreg & drop** 방식으로 쉽게 전송 할 수 있다.

1. **app.py**를 담고 있는 **Flask** 폴더를 가져온다. 데이터를 가져올때 **GIt** 저장소를 사용해서 가져왔다.

```
$ sudo apt install git-all
$ git clone "git ssh url"
```

- clone을 통해 가져왔다면 Flask를 실행 시킬 파이썬을 install 하자 이때 사용할 툴은 `virtualenv` 이다 이 프로그램은 독립된 파이썬 개발환경을 만들어주기 때문에 시스템에 영향을 주지 않고 개발 할 수 있다. 설치 법은 다음과 같다

```
$ sudo apt-get install python-virtualenv
```

2. 클론을 한 폴더에 가상 개발 환경을 생성한다.

```
$ virtualenv venv
```

- 다음과 같이 환경을 만들어 주면 user id 앞에 (venv)라고 가상환경이 준비되었다고 알려준다

3. 가상환경을 만들었으면 **Flask를** 설치 해준다

```
$ pip install Flask
```

- 가상환경에서 pip을 지원하기 때문에 따로 설치할 필요는 없다.

4. app.py를 실행 시킬 폴더에 가서 다음과 같은 명령어를 주면 Flask가 실행된다.

```
$ python app.py
```

이렇게 외부 api를 서버에 올려두고 web server에서 url로 호출을 한다면 그에 맞는 데이터 처리를 한뒤 응답을 해줄 것이다.

처음은 더미데이터를 Web Server로 보내는 연습을 했다.

- app.py

```
from flask import Flask

app = Flask(__name__)

@app.route("/json",methods = ['GET'])
def getString():
    return "안녕하세요"

if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

flask를 어디서든 접근 할 수 있겠금 `app.run` 매개변수로 host = 0.0.0.0으로 설정했다.

Web Server에서 app.py 가 실행 되어 있는 ip주소와 포트번호 5000 그리고 url에 /json을 붙혀주면 "안녕하세요" 라는 string 타입의 데이터를 return 해준다.