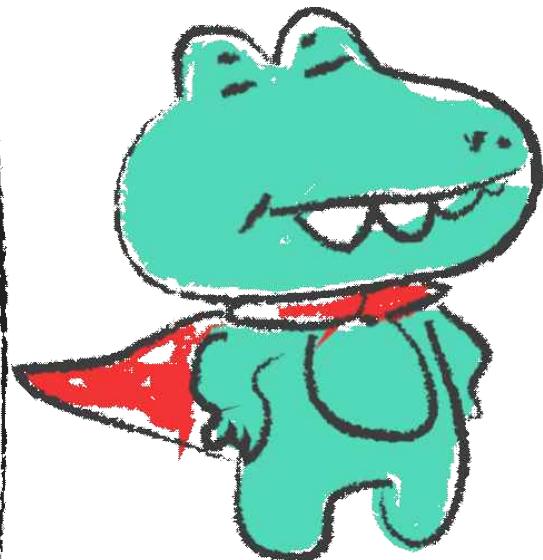


# 클라우딩 컴퓨터 이해

201695021 김으종

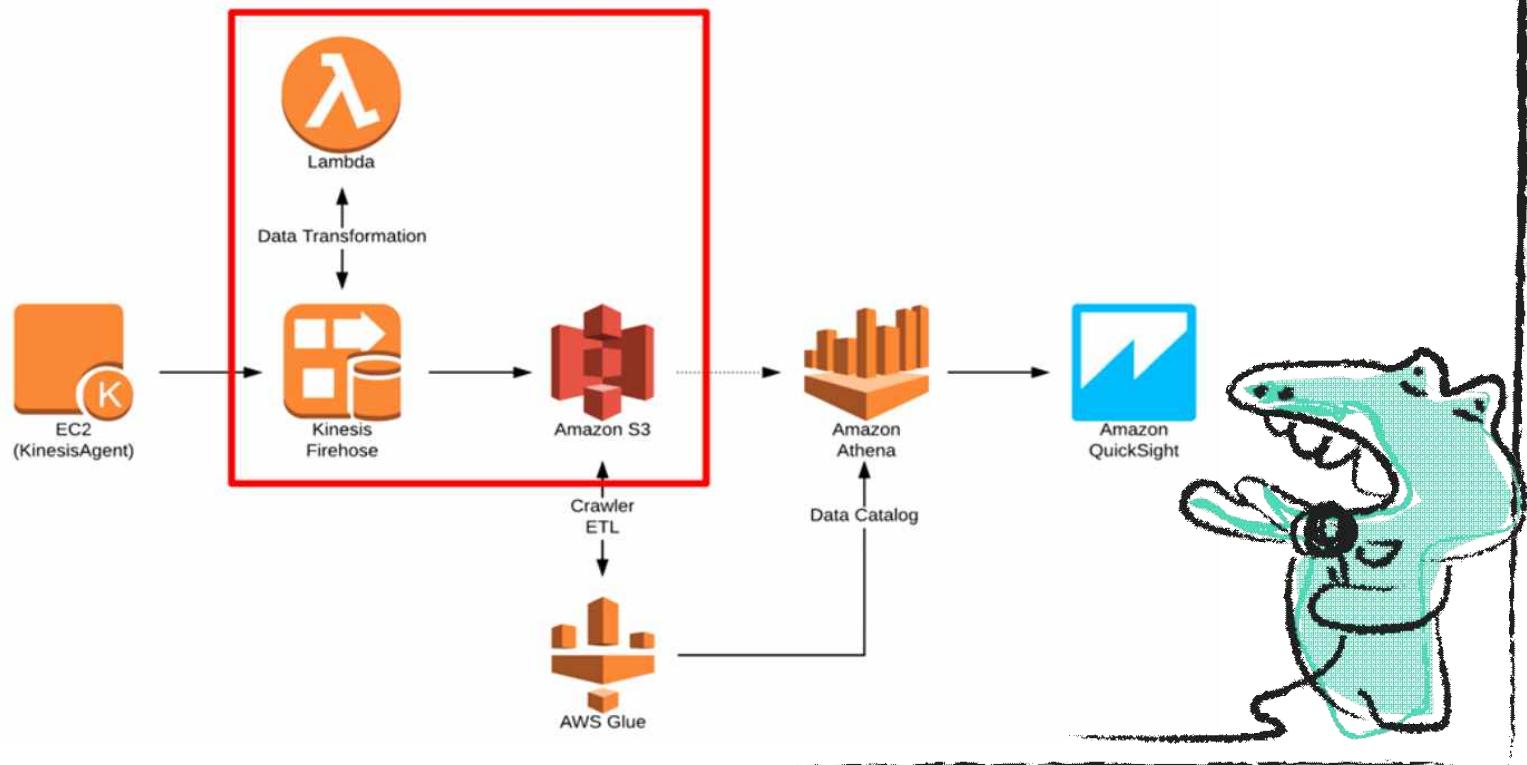
201595053 이도현





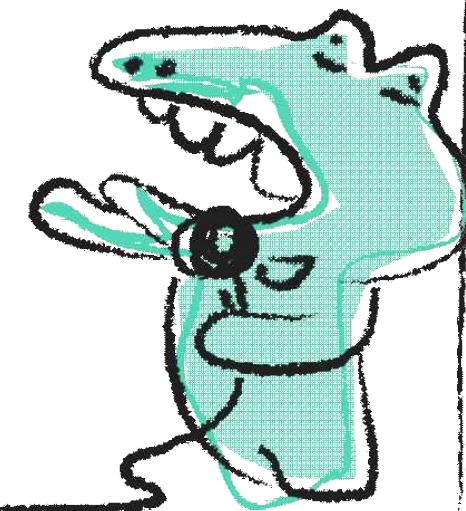
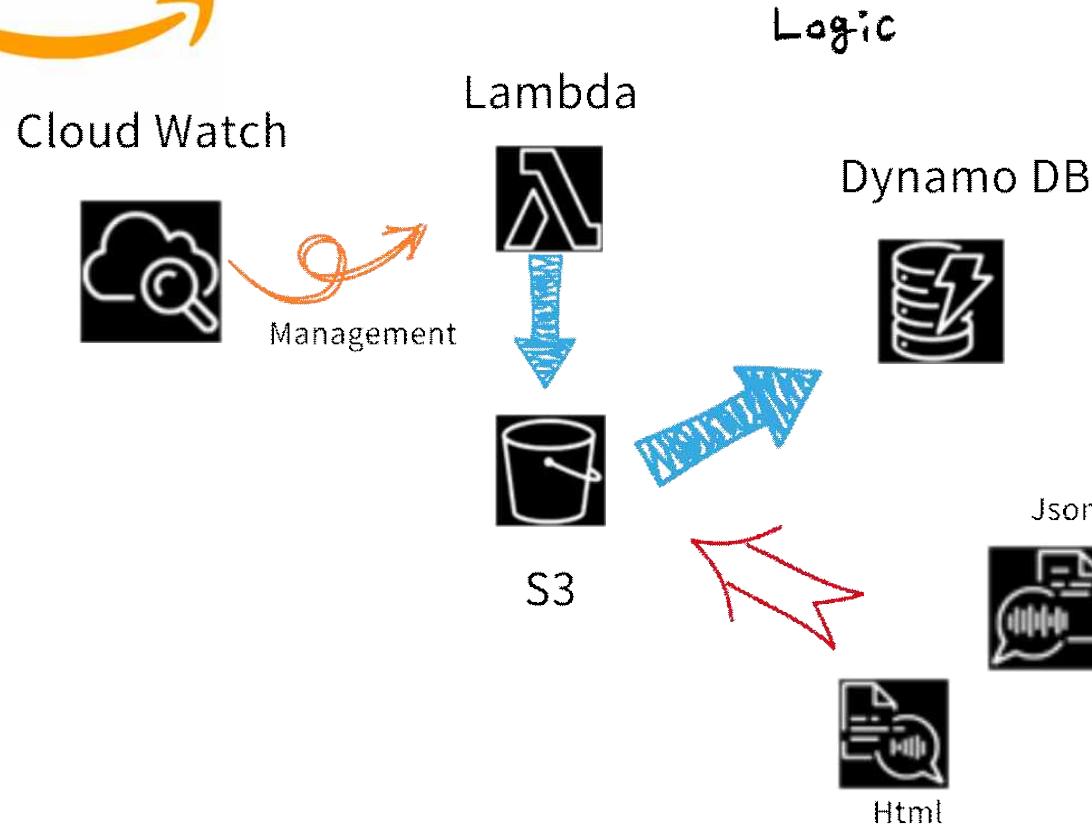
# 자동 복지 센터 MAP

Logic





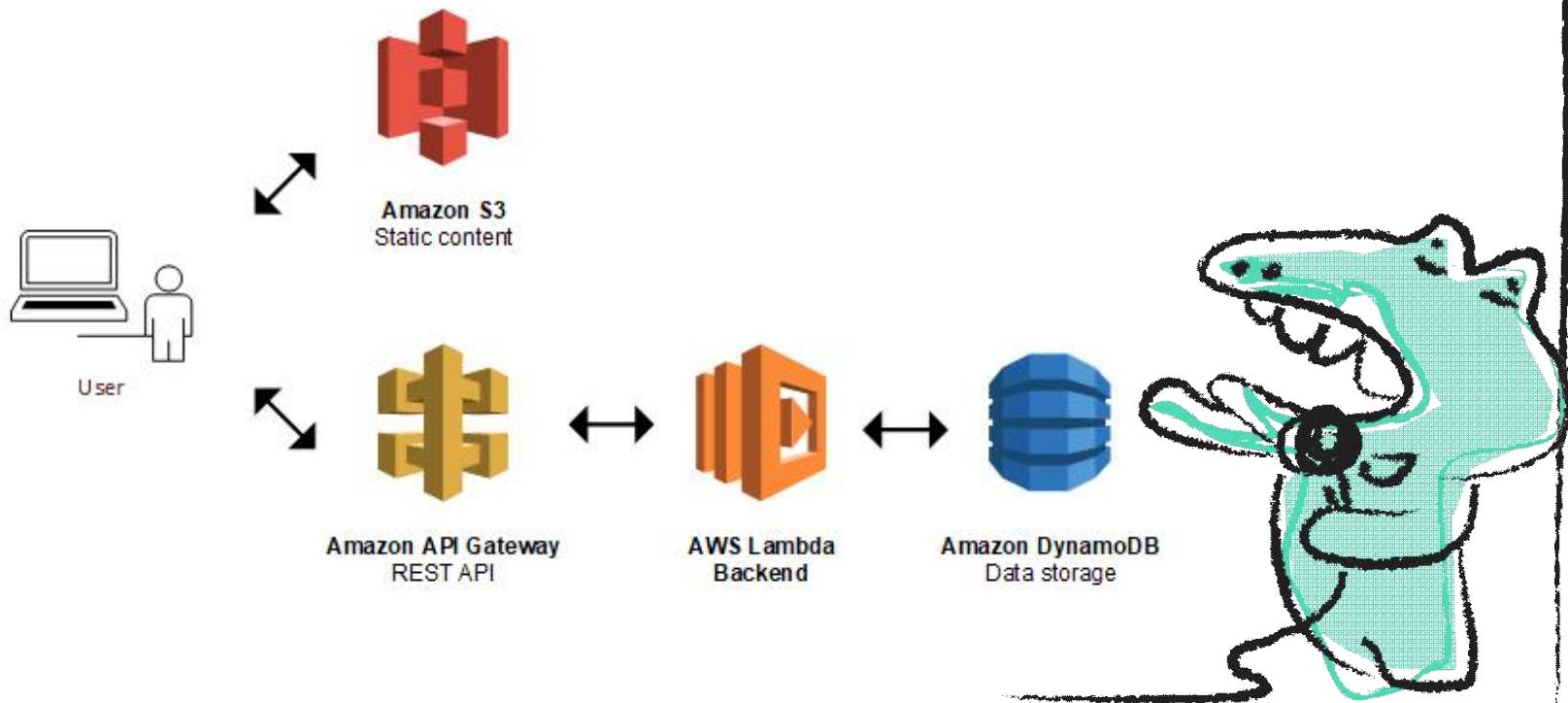
# 자동 복지 센터 MAP





# 자동 복지 센터 MAP

Logic



# 팀원들을 소개합니다



김은종

201695021

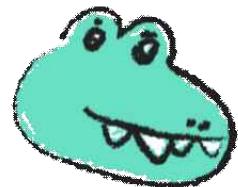


이도현

201595053



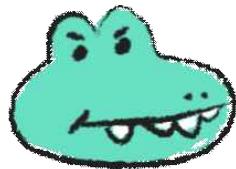
발표주제는요...



아동 복지센터 MAP 구현

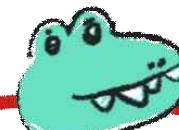


## 핵심 요소는..



기우드1

공공 api에서 데이터를  
받아 s3에 업로드



기우드2

업로드 된 내용에 따른  
MAP 구현자동화



기우드3

시각화 자료에 편리성

# 계획서 초안

## 프로젝트 Team 명

시리야 오늘 날씨 알려줘

## 프로젝트 목적 및 선정이유

요즘 환절기로 인해 일교차가 심하여 옷을 어떻게 입고 나가야할지 매우 난해한 날씨이다.

이러한 불편을 겪고 나니 내가 외출해서 활동하는 시간에 최고기온과 최저기온이 언제 언제 형성 되는지 알려주는 프로그램이 있으면 좋겠다. 라는 생각이 들어서 이 프로그램을 만들면 좋을것 같다 라고 판단했다.

## 프로젝트 범위

공공 API에서  
기상현상 받아서  
시각화 하자

# 공공 데이터

## 오픈API 상세

JSON+XML 부산광역시 아동 복지 및 시설 정보

아동양육시설, 자립시설, 일시보호소(구별 시설명, 소재지, 정원, 현원, 총사자 등)

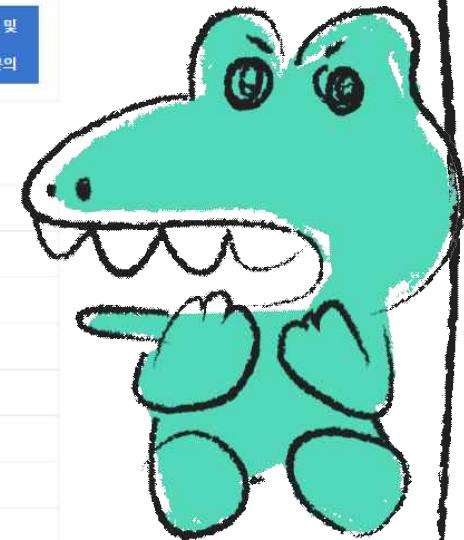
0 0 관상

f t URL 복사

활용신청  
오류신고 및  
담당자 문의

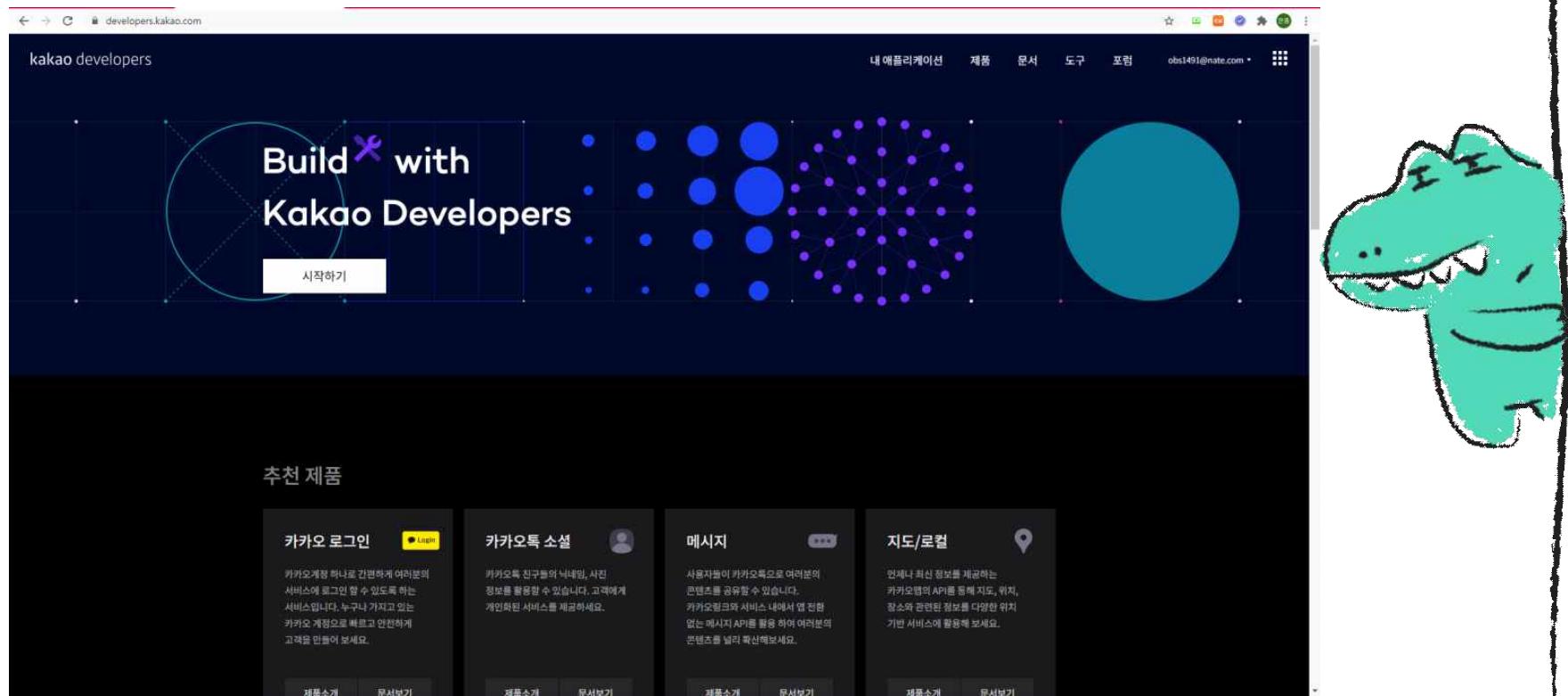
## OpenAPI 정보

분류체계	사회복지 - 사회복지일반	제공기관	부산광역시
관리부서명	통계빅데이터담당관	관리부서 전화번호	051-888-2364
API 유형	REST	데이터포맷	JSON+XML
활용신청	15	키워드	아동, 아동시설, 아동양육시설
등록	2019-10-19	수정	2019-10-29
심의유형	개발단계 : 허용 / 운영단계 : 허용		
비용부과유무	무료		
이용허락범위	<a href="#">이용허락범위 제한 없음</a>		
참고문서	<a href="#">부산 아동복지시설현황 정보 서비스.docx</a>		

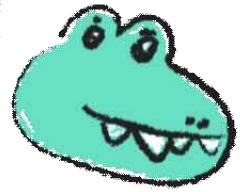


부산 광역시에 있는 아동 복지 및 시설 정보에 대한 공공데이터가 제공된다.  
공공데이터는 <https://www.data.go.kr/>에서 찾아서 쓸 수 있다.

# Kakao Developers



카카오 Map API를 사용하기에 우선 먼저 Kakao Developers에 개발자로 등록한다.



# Add My Application

전체 애플리케이션 (1)

애플리케이션 이름



애플리케이션 추가하기



Hey Siri

ID 506566 OWNER Web

애플리케이션 추가하기

## 애플리케이션 추가하기

앱 아이콘



파일 선택

JPG, GIF, PNG

권장 사이즈 128px, 최대 250KB

앱 이름

Hey 은종 아동복지센터 앱 보여줘

사업자명

김은종

- 입력된 정보는 사용자가 키카오 로그인을 할 때 표시됩니다.
- 정보가 정확하지 않은 경우 서비스 이용이 제한될 수 있습니다.

취소

저장

My 애플리케이션 상세정보 입력

# App Key And Flatorm

Hey 은종 아동복지센터 맵 보여줘 ⚙  
ID 510476 OWNER

## 앱 키

네이티브 앱 키	bd215cbb20d6771000d
REST API 키	e06e0000000000000000000000000000
JavaScript 키	00000000000000000000000000000000
Admin 키	bb487052000000000000000000000000

발급 받은 각종 App 키

중요!

Android      Web 플랫폼 등록

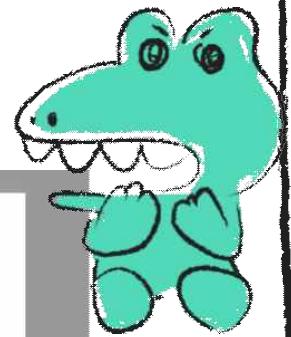
iOS      iOS 플랫폼 등록

Web      Web 플랫폼 등록

사이트 도메인  
JavaScript SDK, 카카오 링크, 카카오 맵, 메시지 API 사용 시 등록이 필요합니다.  
여러개의 도메인은 줄바꿈으로 추가해주세요. 최대 10까지 등록 가능합니다. 추가 등록은 후원(데브옵)  
으로 문의주세요.

예시: (O) https://example.com (X) https://www.example.com

취소      저장



S3에서 웹호스팅 이후 도메인 추가할 창

# Kakao Map API



Maps API

Kakao 지도 API는 웹사이트와 모바일 애플리케이션에서  
지도를 이용한 서비스를 제공할 수 있도록 다양한 기능을 제공하고 있습니다.

준비하기

Kakao 지도 JavaScript API는 카카오맵을 통하여 사용할 수 있습니다.  
그리고 기본 설정과 서비스는 서비스 개발자 설정입니다.

기본 설정은 아래 과정의 일정입니다.

1. 카카오 개발자센터 [http://developer.kakao.com] 접속
2. 개발자 팀 및 단 청장
3. 팀 설정을 추가 및 선택 - [설정]->[Web 플랫폼 팀]-> 팀 설정
4. 서비스 프로토콜 티어 (웹) 및 토큰을 선택하고, [나이트 도색인]을 선택합니다. (예: http://localhost:8080)
5. 퍼미션 설정을 JavaScript 키를 적용 서비스 앱으로 사용합니다.
6. 업을 실행합니다.

▶ 등록한 도메인에 [http://localhost:8080]에서 웹 서버를 실행하거나 위 링크를 클릭합니다.

▶ Python으로 실행할 경우에는 해당 경로로 사용 후,  
# cd /path/to/your/folder

▶ 그 다음으로 터미널을 통해 서버를 실행한 후 ./app.py  
▶ Python -> SimpleHTTPServer 8080

▶ 전화번호를 들어 쪽 도색인에 접속합니다.

총동원 사이트 도색인에서는 카카오맵을 사용할 수 있기 때문에 반드시 등록해주세요.

감독 책임의 [결과판]을 클릭하면 [카카오맵]을 이용합니다. 아직 빙글판은 키가 없지만 해당 페이지에 접속하여 키를 청급 받으세요.

시작하기

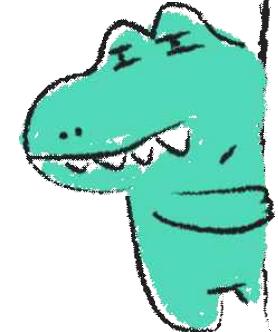
▶ 예제들은 간단한 코드를 통해 빙글판의 카카오맵을 적용할 수 있습니다.

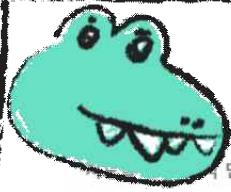
▶ 카카오맵은 웹 단계에 접속해 코드를 확인합니다.

지도를 단계별로 만들기

Kakao Map API 첫 화면  
Web 으로 접속

Guide 화면에 있는데로 잘 따라한다





만들기

먼저 지도를 달기 위한 영역이 필요합니다. 500x400 의 크기로 만들어 보겠습니다.

```
<div id="map" style="width:500px; height:400px;"></div>
```

지도를 담을 영역으로 스타일이 지정된 <div> 태그를 선언합니다.  
<div> 태그의 id값은 map 으로 하도록 하겠습니다.

### 실제 지도를 그리는 Javascript API를 불러오기

```
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=발급받은 APP KEY를 넣으시면 됩니다."></script>
```

// 이라는 상태 프로토콜을 사용하면, 사용자의 http, https 환경에 따라 자동으로 해당 프로토콜을 따라가게 됩니다.

API를 로딩하는 스크립트 태그는 HTML파일안의 head, body 둘 어떤 위치에 넣어도 상관없습니다.  
하지만, 반드시 실행 코드보다 먼저 선언되어야 합니다.

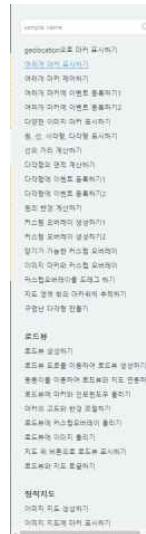
### 지도를 띄우는 코드 작성

```
var container = document.getElementById('map'); //지도를 담을 영역의 DOM 객체입니다.
var options = { //지도를 생성할 때 필요한 기본 옵션
  center: new kakao.maps.LatLng(33.450701, 126.570667), //지도의 중심좌표,
  level: 3 //지도의 레벨(真切, 확대, 확대 확대)
};

var map = new kakao.maps.Map(container, options); //지도 생성 및 객체 리턴
```

지도 생성 코드와 다양한 오픈소스를 제공  
아까 발급받은 자바스크립트 앱키를 코딩에 삽입

# Kakao Map API



### 여러개 마커 표시하기

지도 위에 마커 이미지를 사용하여 여러개의 마커를 표시합니다.



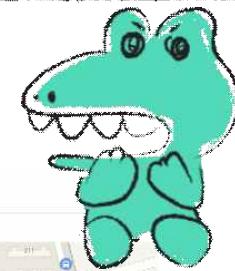
JavaScript

```
var mapContainer = document.getElementById('map'), //지도를 표시할 div
    mapOption = {
      center: new kakao.maps.LatLng(33.450701, 126.570667), //지도의 중심좌표
      level: 3 //지도의 레벨(真切, 확대, 확대 확대)
    };

var map = new kakao.maps.Map(mapContainer, mapOption); //지도 생성
```

여러개의 아동복지센터의 위치를 마커로 표시하기 위해  
'여러개 마커 표시하기' 오픈코드를 사용한다

# Kakao Map API



## 마커에 인포윈도우 표시하기

마커 위에 인포윈도우를 칠판에 표시합니다. 인포윈도우에 들어갈 내용은 HTML로 자유롭게 입력할 수 있습니다.



```
JavaScript | JavaScript + HTML | 자필 채점기

var mapContainer = document.getElementById('map'), // 칠판을 표시할 div
    section = [
        center: new kakao.maps.LatLng(35.46070, 128.57067), // 지도를 표시할
        level: 3 // 지도의 확대 레벨
    ],
    map = new kakao.maps.Map(mapContainer, section);

// 마커가 표시될 위치입니다.
var markerPosition = new kakao.maps.LatLng(35.46070, 128.57067);

// 마커를 생성합니다
var marker = new kakao.maps.Marker({
    position: markerPosition
});

// 마커가 지도 위에 표시되는 위치입니다
marker.setMap();
```

생성한 마커에 내용을 삽입하기 위해  
마커에 인포윈도우 표시하기 오픈코드 사용

## 원, 선, 사각형, 다각형 표시하기

지도 위에 원, 선, 사각형, 다각형을 표시합니다.



```
JavaScript | JavaScript + HTML | 자필 채점기

var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    section = [
        center: new kakao.maps.LatLng(35.46070, 128.57067), // 지도를 표시할
        level: 3 // 지도의 확대 레벨
    ],
    map = new kakao.maps.Map(mapContainer, section);

// 원형을 표시할 좌표, 반지름입니다
var circle = new kakao.maps.Circle({
    center: new kakao.maps.LatLng(35.46070, 128.57067), // 원의 중심좌표
    radius: 50 // 원의 반지름, 단위는 미터입니다
    strokeWeight: 5, // 원의 두께입니다
    strokeColor: '#FF0000', // 원의 색깔입니다
    strokeDash: [1], // 경계 끝부분 모양입니다. 0:实线, 1:虚线
    fillStyle: 'dashed', // 원의 속도 모양입니다
    fillColor: '#FFFF00' // 원의 속도 색깔입니다
```

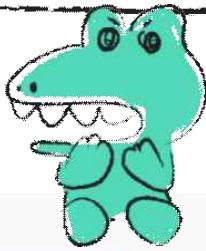
부산진구와 사상구의 구역을 표시해줄 다각형 표시하기  
오픈코드 사용



## AWS, S3 버킷 생성

## AWS 루트사용자로 접속

53 버킷명을 정하고 모든 퍼블릭 액세스 차단을 하지 않고  
위에 두가지만 풀어준다.



# AWS Lambda 함수 생성

함수 생성 Info

다음 옵션 중 하나를 선택하여 함수를 생성합니다.

새로 작성      블루프린트 사용      서비스 엔진 리포지토리 찾아보기

간단한 Hello World 예제는 시작하십시오.

설명 코드 및 구축 Lambda 애플리케이션을 위한 구름 사전 설정을 일반적인 사용 사례를 살펴봅니다.

기본 정보

함수 이름  
함수의 이름을 결정하는 이름을 입력합니다.  
**adongbokji\_lambda**

언어  
함수를 실행할 때 사용할 언어를 선택합니다.  
Node.js 12.x

기본 Info

기본으로 Lambda는 Amazon CloudWatch Log에 로그를 기록하는 권한을 가진 동일한 권한을 제공합니다. 이 기본 권한은 나중에 보안하게 사용할 때 사용할 수 있습니다.

▼ 기본 실행 역할 변경

설정 이름  
함수의 다른 옵션을 정하는 역할을 선택합니다. 사용자 지정 역할을 설정하면 IAM 권한을 사용하십시오.

기본 Lambda 권한을 가진 새 역할 생성

기본 역할 사용

AWS 정책 템플릿에서 새 역할 생성

기본 역할  
당장은 기본 역할 중에Lambda 함수의 초기 사용할 역할을 선택합니다. 이 역할에는 Amazon CloudWatch Log에 로그를 기록할 수 있는 권한이 있어야 합니다.

role-europang  
새로운 역할 **role-europang** 역할을 확인하십시오.

Lambda 함수 명을 정하고  
런타임은 Node.js 12.x를 사용한다.  
Role은 기존 수업시간에 사용한 Role을 지정.

Lambda > 함수 > adongbokji\_lambda

adongbokji\_lambda

ARN - arn:aws:lambda:ap-northeast-2:74114656551:function:adongbokji\_lambda

조회      구분자      작업      서비스      대상      테스트      대상 추가

▼ 디자인

adongbokji\_lambda

Layers

+ 드래그 추가      + 대상 추가

함수 코드 Info

File Edit Find View Go Tools Window Test Deploy

생성된 Lambda 함수에 트리거(S3)를 추가한다.

# Lambda 함수 트리거 설정, DB 테이블 생성

## 추가 트리거

트리거 구성

S3 aws storage

버킷  
이벤트 소스의 역할을 하는 S3 버킷을 선택하십시오. 버킷은 함수와 같은 리전에 있어야 합니다.  
cloudingbucket

이벤트 유형  
Lambda 함수를 트리거하라는 이벤트를 선택합니다. 필요에 따라 이벤트의 첨두사 또는 접미사를 설정할 수 있습니다. 하지만 각 버킷에서 개별 이벤트는 첨두사나 접미사가 겹쳐서 객체 키가 중복될 수 있는 구성을 여러 개 가질 수 없습니다.

모든 객체 생성 이벤트

첨두사 - 선택 사항  
필요할 경우, 일치하는 문자로 시작하는 키를 사용하여 객체에 대해 알림을 제한하려는 단일 첨두사를 입력합니다.  
예: images/

접미사 - 선택 사항  
필요할 경우, 일치하는 문자로 끝나는 키를 사용하여 객체에 대해 알림을 제한하려는 단일 접미사를 입력합니다.  
.json

Lambda는 Amazon S3이(가) 이 트리거에서 Lambda 함수를 호출하는 데 필요한 권한을 추가합니다. Lambda 권한 모델에 대해 자세히 알아보기.

The Lambda console no longer supports disabling S3 and CloudWatch Logs triggers. Delete these triggers to stop further actions.

아까 생성한 S3 버킷을 지정하고  
접미사는 .json을 선택한다.

## DynamoDB 테이블 만들기

DynamoDB는 스키마가 없는 데이터베이스로서, 테이블 이름과 기본 키만 필요로 합니다. 테이블의 기본 키는 각 파티션에서 항목을 고유하게 식별하고, 데이터를 분할하며, 데이터를 정렬하는 한 개 또는 여러 개의 속성으로 구성됩니다.

테이블 이름: bokji\_center\_db  
기본 키: 파티션 키  
manage@ [문자열]

테이블 설정  
기본 설정을 사용하면 테이블을 가장 빠르게 시작할 수 있습니다. 이러한 기본 설정은 지금 또는 테이블을 만든 후에 수정할 수 있습니다.

기본 설정 사용

- 보조 인덱스 없음
- 프론비저닝된 용량은 임기 5개와 쓰기 5개로 설정
- 기본 일정은 SNS 주제 "dynamodb"을 사용하여 80% 상한 설정
- 유형 데이터를 기본 암호화 유형으로 암호화합니다.

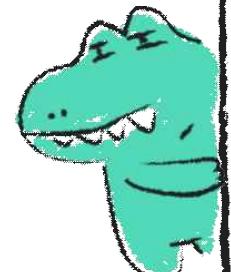
기본적으로 Auto Scaling을 활성화하기 위해 필요한 권한이 없습니다.  
설명서 참조하십시오.

+ 태그 추가 [선택 가능!](#)

CloudWatch 또는 SNS(Simple Notification Service)에 대한 AWS 프리 티어를 초과할 경우 추가 요금이 발생할 수 있습니다. 고급 놀랄 설정은 CloudWatch 관리 콘솔에서 사용할 수 있습니다.

DynamoDB 테이블명을 정하고 기본 키는 공공 API에서  
가져온 json 파일에 있는 하나의 변수를 지정

# 파일 업로드, Elastic Web-Hosting



Amazon S3 > cloudingbucket > 업로드

## 업로드

S3에 업로드할 파일 및 폴더를 추가합니다. 160GB보다 큰 파일을 업로드하려면 AWS CLI, AWS SDK 또는 Amazon S3 REST API를 사용합니다. [자세히 알아보기](#)

여기에 업로드할 파일과 폴더를 끌어서 놓거나, [파일 추가] 또는 [폴더 추가]를 선택합니다.

### 파일 및 폴더 (2 합계, 48.9KB)

이 테이블의 모든 파일과 폴더가 업로드됩니다.

선택	이름	유형	크기
<input type="checkbox"/>	centor.json	application/json	32.0KB
<input type="checkbox"/>	index.html	text/html	17.0KB

### 대상

대상  
<s3://cloudingbucket>

### 대상 세부 정보

다음 버킷 설정은 지정된 대상에 저장된 새 객체에 영향을 줍니다.

**버킷 버전 관리**  
활성화하면 버킷에 객체의 여러 변경을 저장하여 의도하지 않은 사용자 작업 및 애뮬리케이션 장애로부터 손쉽게 복구할 수 있습니다.  
[자세히 알아보기](#)

**기본 암호화**  
활성화하면 이 버킷에 저장된 새 객체가 자동으로 암호화됩니다. [자세히 알아보기](#)

**비활성화**

스캔: [B] bokji\_center\_db: manager

스캔: [B] bokji\_center\_db: manager

검색 시작

한국 1~31개를 보는 중

manager	addr	facility_nm	gugun	installation_date	lat	lon	num_of_j	num_of_wc	presel	reference_da	tel	type_of_facility
김성운	부산진구 일감서로 72 (금암동)	부산진지역아동센터	부산광역시 부	2016-12-07	35.167138	129.036938	19	2	14	2020-07-31	051-893-0160	지역아동센터
최은희	부산진구 동봉로 94번길 28 (	금정지역아동센터	부산광역시 부	1999-06-10	35.164564	129.041024	19	2	15	2020-07-31	051-896-5000	지역아동센터
김정희	부산진구 동봉로 94번길 213.	무른울지역아동센터	부산광역시 부	2013-05-22	35.166782	129.031192	19	2	16	2020-07-31	051-892-7833	지역아동센터
윤성태	부산진구 벽암순환로 127번	신애지역아동센터	부산광역시 부	2012-07-31	35.169749	129.041022	19	2	16	2020-07-31	051-817-8547	지역아동센터
김경우	부산진구 흥왕대로 978 (광장)	광장지역아동센터	부산광역시 부	2000-03-01	35.173496	129.074212	19	2	16	2020-07-31	051-861-3927	지역아동센터
이영재	부산진구 벽영대로 60번길 59	매일보육원	부산광역시 부	1992-05-12	35.164620	129.042363	17	10	17	2020-07-31	051-896-7569	어린양육시설
유재용	부산광역시 사상구 가야대로 2	고개지역아동센터	부산광역시 사	2011-06-24	35.148143	129.01328	19	2	17	2020-07-31	050-7763-1649	지역아동센터
김진아	부산광역시 사상구 사성로 27	밀양지역아동센터	부산광역시 사	2015-01-06	35.169366	129.98224	19	2	18	2020-07-31	051-304-5098	지역아동센터
한경미	부산진구 복지로 117-12 (가)	온하수지역아동센터	부산광역시 부	2002-02-01	35.142849	129.019559	19	2	18	2020-07-31	050-7546-4926	지역아동센터
이경희	부산진구 기아공원로 38번길	예촌지역아동센터	부산광역시 부	2015-09-04	35.150329	129.029243	19	2	18	2020-07-31	051-894-9991	지역아동센터
강명자	부산진구 모금로 98번길 20(	송북지역아동센터	부산광역시 부	2005-12-30	35.166995	129.056648	19	2	18	2020-07-31	051-816-4252	지역아동센터
윤승철	부산진구 양지로 94번길 20-1	우지역아동센터	부산광역시 부	2010-01-06	35.170502	129.056439	19	2	18	2020-07-31	051-800-4964	지역아동센터

S3 버킷에 Json 파일과 지도를 뜨워줄  
Html 파일을 업로드 한다.

업로드한 json 파일이 정상적으로  
DynamoDB에 들어갔다.

# Json 파일 설정



## 정적 웹 사이트 호스팅 편집

**정적 웹 사이트 호스팅**  
이 버킷을 사용하여 웹 사이트를 호스팅하거나 요청을 리디렉션합니다. [자세히 알아보기](#)

**정적 웹 사이트 호스팅**

비활성화  
 활성화

**호스팅 유형**

정적 웹 사이트 호스팅  
버킷 앤드포인트를 절 주소로 사용합니다. [자세히 알아보기](#)

객체에 대한 요청 리디렉션  
요청을 다른 버킷 또는 도메인으로 리디렉션합니다. [자세히 알아보기](#)

**고객이 웹 사이트 앤드포인트의 콘텐츠에 액세스할 수 있게 하려면 모든 콘텐츠를 공개적으로 읽기 가능하도록 설정해야 합니다. 이렇게 하려면, 버킷에 대한 S3 퍼블릭 액세스 차단 설정을 편집하면 됩니다. 자세한 내용은 [Amazon S3 퍼블릭 액세스 차단 사용](#)을 참조하십시오.**

**인덱스 문서**  
웹 사이트의 출 페이지 또는 기본 페이지를 지정합니다.

**오류 문서**  
오류가 발생하면 반환됩니다.

**리디렉션 규칙 - 선택 사항**  
JSON으로 작성된 리디렉션 규칙을 특정 쿠맨드에 대한 출 페이지 요청을 자동으로 리디렉션합니다. [자세히 알아보기](#)

Web-Hosting을 하기 위해 S3에서 정적 웹사이트 호스팅을 해주고 웹사이트에 메인 html을 지정해준다.

Amazon S3 > cloudingbucket > centor.json

**centor.json**

[세부 정보](#) [버전](#)

**객체 개요**

소유자

AWS 리본  
아시아 태평양(서울) ap-northeast-2

마지막 수정  
2020. 11. 27. pm 6:41:58 PM KST

크기  
32.0KB

유형  
json

키  
[centor.json](#)

S3 URI  
[s3://cloudingbucket/centor.json](#)

Amazon 리소스 이름(ARN)  
[arn:aws:s3:::cloudingbucket/centor.json](#)

엔터티 태그(Etag)  
[27eb5e57b927940a8200302c005047f4](#)

**액체 URL 복사됨**  
<https://cloudingbucket.s3.ap-northeast-2.amazonaws.com/centor.json>

Html 코딩에 사용하기 위해 업로드 한 centor.json의 객체 URL을 복사한다.

# ACL(액세스 제어목록 편집)

```
<head>
<meta charset="utf-8">
<title>다음 지도 API</title>
<script type="text/javascript" src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>//--json받아오는거
var html = [];
$.getJSON('https://totalclouding-buc.s3.ap-northeast-2.amazonaws.com/center.json', function (data) { //<
    $.each(data, function (i, item) {
        html.push(Number(item.lat));
        html.push(Number(item.lng));
        html.push(item.facility_nm);
        html.push(item.addr);
    });
});
</script>
</head>
```

Amazon S3 > [cloudingbucket] > centor.json > 액세스 제어 목록 편집

## 액세스 제어 목록 편집

이 버킷에 대해 퍼블릭 액세스 차단 설정이 활성화되어 있기 때문에 퍼블릭 액세스가 차단됩니다.  
활성화된 설정을 확인하려면 퍼블릭 액세스 차단을 위한 버킷 설정을 확인하세요. Amazon S3 퍼블릭 액세스 차단 사용에 대해 자세히 알아보세요.

### ACL(액세스 제어 목록)

AWS 개정에 따른 읽기/쓰기 권한을 부여합니다. 자세히 알아보기

파부여자	액체	액체 ACL
액체 소유자(AWS 계정)	<input checked="" type="checkbox"/> 읽기	<input checked="" type="checkbox"/> 읽기 <input checked="" type="checkbox"/> 쓰기
정식 ID: <a href="#">db8dd953fc5690f3dd51c2227erfd0a89411ef7b8d6afdf5e1dc1d0446/d8a9d0</a>		
모든 사용자(퍼블릭 액세스) 그룹: <a href="#">http://acs.amazonaws.com/groups/global&gt;AllUsers</a>	<input checked="" type="checkbox"/> 읽기 <input checked="" type="checkbox"/> 쓰기	<input checked="" type="checkbox"/> 읽기 <input checked="" type="checkbox"/> 쓰기
인증된 사용자 그룹(AWS 계정이 있는 모든 사용자) 그룹: <a href="#">http://acs.amazonaws.com/groups/global/AuthenticatedUsers</a>	<input type="checkbox"/> 읽기 <input type="checkbox"/> 쓰기	



모든 사용자 또는 인증된 사용자 그룹 파부여자에게 액세스 권한을 부여하면 누구나 이 객체에 액세스할 수 있습니다.

자세히 알아보기

이러한 변경 사항이 이 객체에 미치는 영향을 이해합니다.

복사한 json 객체 URL을 Html 코드에 삽입

Json 파일의 액세스 제어목록에서  
퍼블릭 액세스의 읽기를 적용한다

# web 사이트 도메인 등록

Amazon S3 > cloudingbucket > index.html > 액세스 제어 목록 편집

## 액세스 제어 목록 편집

이 버킷에 대해 퍼블릭 액세스 차단 설정이 활성화되어 있기 때문에 퍼블릭 액세스가 차단됩니다.  
활성화된 설정을 확인하려면 퍼블릭 액세스 차단을 위한 버킷 설정을 확인하세요. Amazon S3 퍼블릭 액세스 차단 사용에 대해 자세히 알아보세요.

**ACL(액세스 제어 목록)**  
AWS 계정에 기본 읽기/쓰기 권한을 부여합니다. 자세히 알아보기

피부여자	액체	액체 ACL
액체 소유자(AWS 계정)	<input checked="" type="checkbox"/> 읽기	<input checked="" type="checkbox"/> 읽기 <input checked="" type="checkbox"/> 쓰기
정식 ID: db8dd955fc5690f3dd51c2227efdd0a80411cf7b8d6af5e1dc1d0446fd8a8d0		
모든 사람(퍼블릭 액세스)	<input checked="" type="checkbox"/> 읽기	<input checked="" type="checkbox"/> 읽기 <input type="checkbox"/> 쓰기
그룹: http://acs.amazonaws.com/groups/global/AllUsers		
인증된 사용자 그룹(AWS 계정이 있는 모든 사용자)	<input type="checkbox"/> 읽기	<input type="checkbox"/> 읽기 <input type="checkbox"/> 쓰기
그룹: http://acs.amazonaws.com/groups/global/AuthenticatedUsers		

⚠ 모든 사용자 또는 인증된 사용자 그룹 피부여자에게 액세스 권한을 부여하면 누구나 이 액체에 액세스할 수 있습니다.

자세히 알아보기

이러한 변경 사항이 이 액체에 미치는 영향을 이해합니다.

[취소](#) [저장](#)

업로드한 index.html 파일의 액세스 제어목록도  
퍼블릭 액세스의 읽기를 허용한다.

## 정적 웹 사이트 호스팅

이 버킷을 사용하여 웹 사이트를 호스팅하거나 요청을 리디렉션합니다. 자세히 알아보기

정적 웹 사이트 호스팅 활성화됨

호스팅 유형  
버킷 호스팅

버킷 웹 사이트 엔드포인트  
버킷을 정적 웹 사이트로 구성하면, 해당 웹 사이트를 버킷의 AWS 리전별 웹 사이트 엔드포인트에서 사용할 수 있습니다. 자세히 알아보기

버킷 웹 사이트 엔드포인트 복사됨 ip-northeast-2.amazonaws.com

## Web 플랫폼 등록

### 사이트 도메인

JavaScript SDK, 카카오 링크, 카카오 맵, 메시지 API 사용시 등록이 필요합니다.

여러개의 도메인은 줄바꿈으로 추가해주세요. 최대 10까지 등록 가능합니다. 추가 등록은 포함(데브록)으로 문의주세요.

예시: (O) https://example.com (X) https://www.example.com

http://cloudingbucket.s3-website.ap-northeast-2.amazonaws.com

S3의 웹사이트 호스팅의 도메인을  
Kakao Developer의 도메인 플랫폼에 저장

# S3 CORS 편집

CORS(cross-origin 리소스 공유) 편집

CORS(Cross-origin 리소스 공유)

JSON으로 작성된 CORS 구성은 한 도메인에 로드되어 다른 도메인의 리소스와 상호 작용하는 클라이언트 헤더 애플리케이션에 대한 방법을 정의합니다. [자세히 알아보기](#)

```
1 > [
  2 >   {
    3 >     "AllowedHeaders": [
    4 >       "*"
    5 >     ],
    6 >     "AllowedMethods": [
    7 >       "GET",
    8 >       "PUT",
    9 >       "POST",
    10 >      "HEAD",
    11 >      "DELETE"
    12 >    ],
    13 >    "AllowedOrigins": [
    14 >      "*"
    15 >    ],
    16 >    "ExposeHeaders": [],
    17 >    "MaxAgeSeconds": 3000
    18 >  }
  19 >]
```

까먹지 말고!

취소 변경 사항 저장

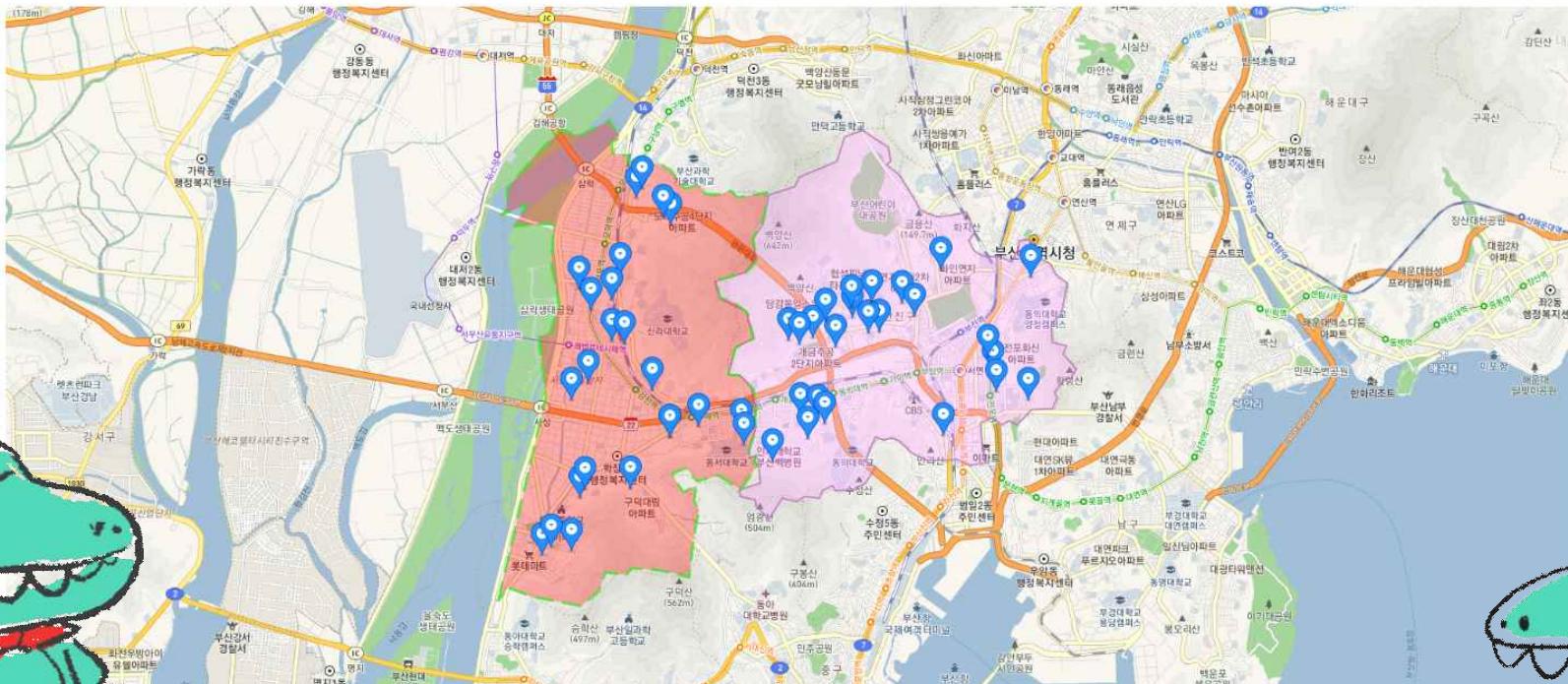
이후 호스팅한 도메인에 들어가도 HTTP Request 오류가 발생한다.

오류를 해결하기 위해 S3 버킷에서 CORS 편집에 코드를 입력한다.

# 실행화면 #1

◀ ▶ ⚡ 주의 요함 | cloudingbuckets3-website.ap-northeast-2.amazonaws.com

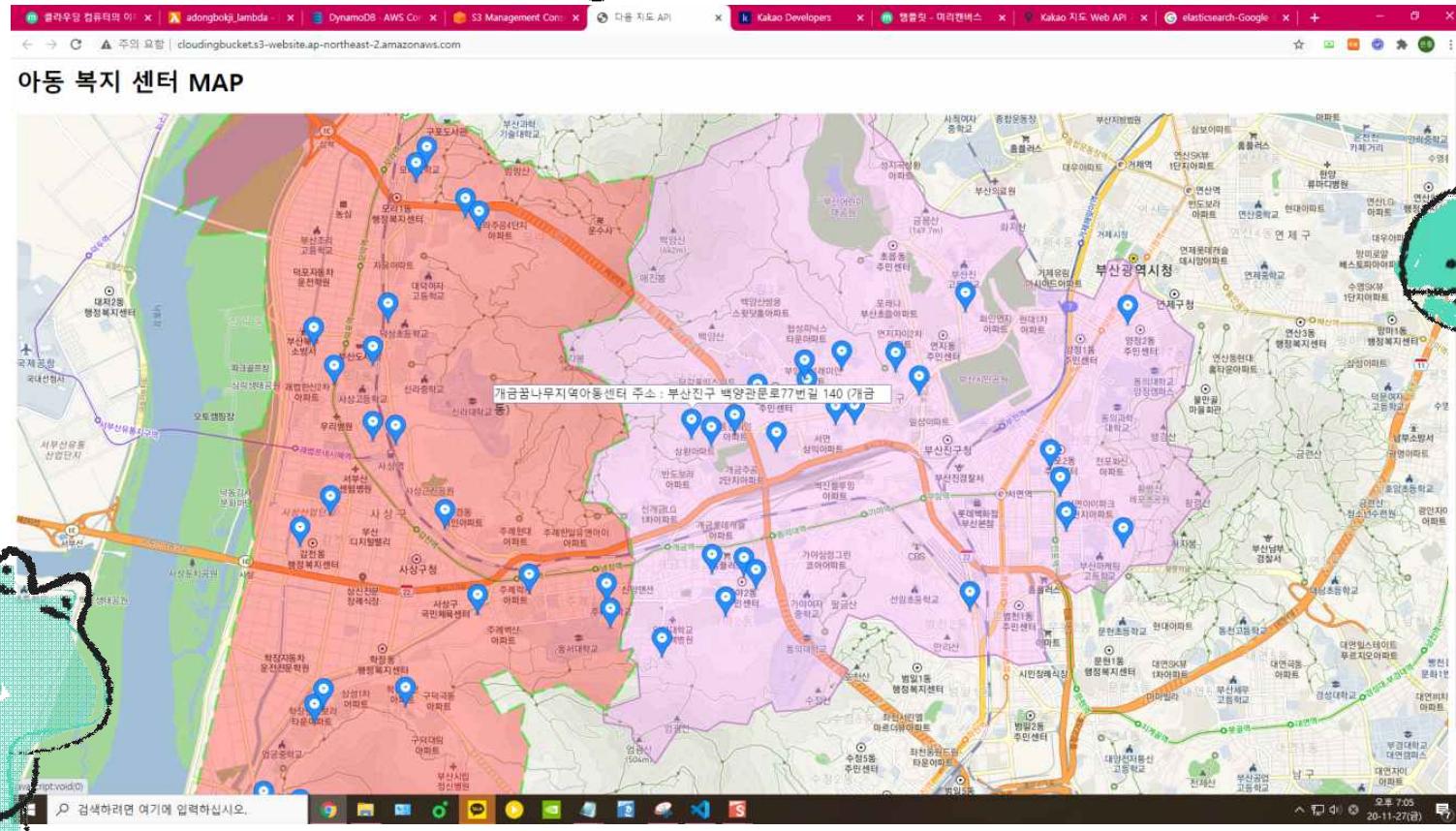
## 아동 복지 센터 MAP



진주와 사성구의 구역이 다른색깔로 잘 표시 되었다.



# 실행화면 #2



마우스를 올리면 복지센터의 이름과 주소가 나온다.

# Lambda 코드(Node.js 12.x)

```
1 const AWS = require('aws-sdk')
2 const s3 = new AWS.S3()
3 const documentClient = new AWS.DynamoDB.DocumentClient();
4
5 exports.handler = async (event) => {
6     let statusCode = 0;
7     let responseBody = '';
8     const {name} = event.Records[0].s3.bucket;
9     const {key} = event.Records[0].s3.object;
10
11    const getParms = {
12        Bucket: name,
13        Key: key
14    };
15
16    try {
17        const s3Data = await s3.getObject(getParms).promise();
18        const usersStr = s3Data.Body.toString();
19        const usersJSON = JSON.parse(usersStr);
20        console.log('Users :: ${usersStr}');
21
22        await Promise.all(usersJSON.map(async user => {
23            const {facility_nm, manager, addr, installation_date, num_of_people,
24            present_num_of_pp, num_of_worker, type_of_facility, tel, lat, lng,
25            gugun, reference_date} = user;
26            // json 파일안 항목을 나열하고 그 값을 받는다.
```

```
27    const putParams = {
28        TableName: "bokji_center_db", // json 파일데이터를 받을 DB table
29        Item: {
30            facility_nm: facility_nm,
31            manager: manager,
32            addr: addr,
33            installation_date: installation_date,
34            num_of_people: num_of_people,
35            present_num_of_pp: present_num_of_pp,
36            num_of_worker: num_of_worker,
37            type_of_facility: type_of_facility,
38            tel: tel,
39            lat: lat,
40            lng: lng,
41            gugun: gugun,
42            reference_date: reference_date
43        }
44    };
45    await documentClient.put(putParams).promise();
46 });
47
48    responseBody = 'Sucess adding data';
49    statusCode = 201;
50
51 } catch (err) {
52     responseBody = 'Error adding data';
53     statusCode = 403;
54 }
```



# Lambda 코드(Node.js 12.x)

```
1 const AWS = require('aws-sdk')
2 const s3 = new AWS.S3()
3 const documentClient = new AWS.DynamoDB.DocumentClient()
4
5 exports.handler = async (event) => {
6     let statusCode = 0;
7     let responseBody = '';
8     const {name} = event.Records[0].s3.bucket;
9     const {key} = event.Records[0].s3.object;
10
11    const getParms = {
12        Bucket: name,
13        Key: key
14    };
15
16    try {
17        const s3Data = await s3.getObject(getParms).promise();
18        const usersStr = s3Data.Body.toString();
19        const usersJSON = JSON.parse(usersStr);
20        console.log('Users :::: ${usersStr}');
21
22        await Promise.all(usersJSON.map(async user => {
23            const {facility_nm, manager, addr, installation_date, num_of_people,
24            present_num_of_pp, num_of_worker, type_of_facility, tel, lat, lng,
25            gugun, reference_date} = user;
26            // json 파일안 항목을 나열하고 그 값을 받는다.
```



# index.html 코드 #1

```
1  <html>
2  <head>
3      <meta charset="utf-8">
4      <title>아동 복지 센터 API</title>
5      <script type="text/javascript" src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
6      <script>
7          var html = [];
8          $.getJSON('https://cloudingbucket.s3.ap-northeast-2.amazonaws.com/centor.json', function (data) {// 업로드한 json 파일의 객체 url 지정
9              $.each(data, function (i, item) {
10                  html.push(Number(item.lat));
11                  html.push(Number(item.lng));
12                  html.push(item.facility_nm );
13                  html.push(item.addr); // 업로드한 json 객체에서 총 4개(위도, 경도, 센터이름, 센터주소)를 가져온다.
14              });
15          });
16      </script>
17  </head>
18  <body>
19      <h1>아동 복지 센터 MAP</h1>
20      <div id="map" style="width:100%;height:100vh;"></div>
21      <script src="https://dapi.kakao.com/v2/maps/sdk.js?appkey=03dc9e8b4ec4db78e0af85cde8d0bfe6&libraries=clusterer"></script>
22      <script type="text/javascript">
23          var mapContainer = document.getElementById('map'), // 지도를 표시할 div
24          mapOption = {
25              center: new kakao.maps.LatLng(html[0], html[1]), // 지도의 중심좌표
26              level: 7, // 지도의 확대 레벨
27              mapTypeId: kakao.maps.MapTypeId.ROADMAP // 지도종류
28          };
29          // 지도를 생성한다
30          var map = new kakao.maps.Map(mapContainer, mapOption);
31          // 마커 클러스터러를 생성합니다
32          var clusterer = new kakao.maps.MarkerClusterer({
33              map: map, // 마커들을 클러스터로 관리하고 표시할 지도 객체
34              averageCenter: true, // 클러스터에 포함된 마커들의 평균 위치를 클러스터 마커 위치로 설정
35              minLevel: 10 // 클러스터 할 최소 지도 레벨
36          });
37      </script>
```



## index.html 코드 #2

```
38     var markers = [];
39     for (var i = 0; i < html.length; i++) {
40         if(i%4==0){// json파일에서 꺼내올 4개의 항목이 잘 추출이 되었다면 지도에 마커를 찍는다.
41             var markerPosition = new kakao.maps.LatLng(html[i], html[i + 1]);
42             // 마커를 생성합니다
43             var marker = new kakao.maps.Marker({
44                 position: markerPosition
45             });
46             // 마커가 지도 위에 표시되도록 설정합니다
47             marker.setMap(map);
48             var iwContent = (html[i+2] + " 주소 :" + html[i+3]), // 인포윈도우에 표출될 내용
49             iwPosition = new kakao.maps.LatLng(html[i], html[i + 1]); //인포윈도우 표시 위치입니다
50             // 인포윈도우를 생성합니다
51             var infowindow = new kakao.maps.InfoWindow({
52                 position: iwPosition,
53                 content: iwContent
54             });
55             markers.push(marker);
56             // 이벤트 리스너로는 클로저를 만들어 등록합니다
57             kakao.maps.event.addListener(marker, 'mouseover', makeOverListener(map, marker, infowindow));
58             kakao.maps.event.addListener(marker, 'mouseout', makeOutListener(infowindow));
59         }
60     };
61     // 클러스터리에 마커들을 추가합니다
62     cluster.addMarkers(markers);
63     // 인포윈도우를 표시하는 클로저를 만드는 함수입니다
64     function makeOverListener(map, marker, infowindow) {
65         return function () {
66             infowindow.open(map, marker);
67         };
68     }
69     // 인포윈도우를 닫는 클로저를 만드는 함수입니다
70     function makeOutListener(infowindow) {
71         return function () {
72             infowindow.close();
73         };
74     }
```



# index.html 코드 #3

```
76 // 다각형을 구성하는 좌표 배열입니다. 이 좌표들을 이어서 다각형을 표시합니다
77 var polygonPath = [
78     new kakao.maps.LatLng(35.12020094572115, 128.96202850286903),
79     new kakao.maps.LatLng(35.15459410542651, 128.97267150793138),
80     new kakao.maps.LatLng(35.16483905325401, 128.97146987832758),
81     new kakao.maps.LatLng(35.16540038296907, 128.97370147616323),
82     new kakao.maps.LatLng(35.17045217605399, 128.97576141262692),
83     new kakao.maps.LatLng(35.17676647606333, 128.97438812165115),
84     new kakao.maps.LatLng(35.184062389729554, 128.97490310576705),
85     new kakao.maps.LatLng(35.18378178977535, 128.9668350212843),
86     new kakao.maps.LatLng(35.1875698874116, 128.9694099418639),
87     kakao.maps.LatLng(35.185745969109355, 128.96408843933273),
88     kakao.maps.LatLng(35.186868335985785, 128.96529006893655),
89     kakao.maps.LatLng(35.18841156512946, 128.96546173030885),
90     new kakao.maps.LatLng(35.19023534360404, 128.96546173030885),
91     kakao.maps.LatLng(35.192760507757356, 128.9680366508881),
92     kakao.maps.LatLng(35.19753004813447, 128.97352981479125),
93     kakao.maps.LatLng(35.20019525755188, 128.97713470360273),
94     kakao.maps.LatLng(35.19907307477199, 128.9805679310422),
95     kakao.maps.LatLng(35.20145769464684, 128.98709106317716),
96     kakao.maps.LatLng(35.19865225223237, 128.98915099964086),
97     new kakao.maps.LatLng(35.19767032449459, 128.98743438592112),
98     new kakao.maps.LatLng(35.19584671291816, 128.9877770866508),
99     new kakao.maps.LatLng(35.195566153657374, 128.9908676133606),
100    new kakao.maps.LatLng(35.19416334281916, 128.99069595198864),
101    new kakao.maps.LatLng(35.19528559342764, 128.9939575180561),
102    new kakao.maps.LatLng(35.19402306040303, 128.99464416354402),
103    new kakao.maps.LatLng(35.1938827774468, 128.99584579314782),
104    new kakao.maps.LatLng(35.19416334281916, 128.99739074549558),
105    new kakao.maps.LatLng(35.19304187670759, 128.99910735921532),
106    new kakao.maps.LatLng(35.19290079235359, 128.9996223433125),
107    new kakao.maps.LatLng(35.18967418535645, 129.0025405866548),
108    new kakao.maps.LatLng(35.19023534360404, 129.00563049135033),
109    new kakao.maps.LatLng(35.188692149097655, 129.00666045958218),
```

간드아



```
121     new kakao.maps.LatLng(35.17003120528376, 129.01215362348535),
122     new kakao.maps.LatLng(35.16820697342803, 129.01507186680888),
123     new kakao.maps.LatLng(35.163856716930034, 129.01627349641268),
124     new kakao.maps.LatLng(35.16217268419877, 129.0123252848573),
125     new kakao.maps.LatLng(35.157681759784715, 129.01078033250954),
126     new kakao.maps.LatLng(35.157401068776416, 129.0135269144611),
127     new kakao.maps.LatLng(35.155015156103055, 129.01618183504073),
128     new kakao.maps.LatLng(35.152488819358354, 129.01644515778466),
129     new kakao.maps.LatLng(35.149962404170005, 129.01627349641268),
130     new kakao.maps.LatLng(35.14785699825927, 129.01627349641268),
131     new kakao.maps.LatLng(35.14490933847226, 129.0149002054369),
132     new kakao.maps.LatLng(35.143505653430886, 129.01455688269294),
133     new kakao.maps.LatLng(35.143505653430886, 129.01627349641268),
134     new kakao.maps.LatLng(35.142523059497975, 129.0155868509248),
135     new kakao.maps.LatLng(35.142101944181256, 129.01661681915664),
136     new kakao.maps.LatLng(35.13789067125778, 129.013011929917),
137     new kakao.maps.LatLng(35.14041746119824, 129.00820541150176),
138     new kakao.maps.LatLng(35.13943482998552, 129.00391387720242),
139     new kakao.maps.LatLng(35.141821199497265, 129.00168227936675),
140     new kakao.maps.LatLng(35.14083858522975, 128.99859237467123),
141     new kakao.maps.LatLng(35.13760991197858, 129.00082397250688),
142     new kakao.maps.LatLng(35.13873294328575, 129.00374221583044),
143     new kakao.maps.LatLng(35.13620610105878, 129.00305557034255),
144     new kakao.maps.LatLng(35.134661881089, 129.00408553857437),
145     new kakao.maps.LatLng(35.130029045449255, 129.00288390897057),
146     new kakao.maps.LatLng(35.12314950021117, 129.00168227936675),
147     new kakao.maps.LatLng(35.12381512124639, 128.9987640360432),
148     new kakao.maps.LatLng(35.124553536231005, 128.9948158244878),
149     new kakao.maps.LatLng(35.124834340530704, 128.9898376447006),
150     new kakao.maps.LatLng(35.12371111752337, 128.98880767646875),
151     new kakao.maps.LatLng(35.12427273096322, 128.98434448807974),
152     new kakao.maps.LatLng(35.121843400802535, 128.98297118982165),
153     new kakao.maps.LatLng(35.11851600963394, 128.9791946396382),
154     new kakao.maps.LatLng(35.12034135556551, 128.96906661869176),
155     new kakao.maps.LatLng(35.12034135556551, 128.96906661869176),
156     new kakao.maps.LatLng(35.12202625396973, 128.96391677753257),
157     new kakao.maps.LatLng(35.11992012551916, 128.96202850244086)
158 ];
159 // 지도에 표시할 다각형을 생성합니다
```

# index.html 코드 #4

```

159 // 지도에 표시할 사각구 구역 표시를 생성합니다
160 var polygon = new kakao.maps.Polygon({
161   path: polygonPath, // 그려질 다각형의 좌표 배열입니다
162   strokeWeight: 3, // 선의 두께입니다
163   strokeColor: '#39DE2A', // 선의 색깔입니다
164   strokeOpacity: 0.8, // 선의 불투명도입니다 1에서 0 사이의 값이며 0에 가까울수록 투명합니다
165   strokeStyle: 'longdash', // 선의 스타일입니다
166   fillColor: 'red', // 채우기 색깔입니다
167   fillOpacity: 0.3 // 채우기 불투명도입니다
168 });
169 // 지도에 표시할 부산진구 구역 표시를 생성합니다
170 var path = [
171   new kakao.maps.LatLng(35.19892896103547, 129.03947713926783),
172   new kakao.maps.LatLng(35.1979920968214, 129.03603767658151),
173   new kakao.maps.LatLng(35.193776674139464, 129.03431794523834),
174   new kakao.maps.LatLng(35.19119950811947, 129.02643584324883),
175   new kakao.maps.LatLng(35.18955983265371, 129.025432666632),
176   new kakao.maps.LatLng(35.188271493000334, 129.019986850712),
177   new kakao.maps.LatLng(35.183235060193034, 129.01869705220463),
178   new kakao.maps.LatLng(35.18077529351994, 129.01640407708804),
179   new kakao.maps.LatLng(35.17737835049462, 129.01712063180673),
180   new kakao.maps.LatLng(35.17503554845657, 129.0148276566825),
181   new kakao.maps.LatLng(35.17362983482581, 129.00837866414568),
182   new kakao.maps.LatLng(35.17070118669787, 129.00938184076253),
183   new kakao.maps.LatLng(35.16847534356975, 129.0148276566825),
184   new kakao.maps.LatLng(35.16367200080513, 129.01626076613516),
185   new kakao.maps.LatLng(35.1623832508198, 129.01253468155832),
186   new kakao.maps.LatLng(35.15769671509981, 129.010814958021518),
187   new kakao.maps.LatLng(35.15746238122601, 129.01325123628462),
188   new kakao.maps.LatLng(35.15547051604513, 129.01554421140884),
189   new kakao.maps.LatLng(35.1525412139507, 129.0161174551899),
190   new kakao.maps.LatLng(35.152072515828365, 129.01669069897093),
191   new kakao.maps.LatLng(35.1498461628774, 129.0159741424464),
192   new kakao.maps.LatLng(35.148557193853456, 129.01640407708844),
193   new kakao.maps.LatLng(35.14656511065761, 129.0154009004636),
194   new kakao.maps.LatLng(35.14504172001708, 129.0149709676278),
195   new kakao.maps.LatLng(35.143401113591615, 129.0148276566825),
196   new kakao.maps.LatLng(35.14375267489534, 129.01626076613516),
197   new kakao.maps.LatLng(35.14269798642866, 129.01583083329936),
198   new kakao.maps.LatLng(35.142229231612255, 129.01669069897093),
199   new kakao.maps.LatLng(35.13906506599215, 129.01697732086146),

```



```

250   new kakao.maps.LatLng(35.17362983482581, 129.07960420394153),
251   new kakao.maps.LatLng(35.17550411426556, 129.0786010273247),
252   new kakao.maps.LatLng(35.17667551697253, 129.07587811936472),
253   new kakao.maps.LatLng(35.1789011356128, 129.07530487558364),
254   new kakao.maps.LatLng(35.179721085014414, 129.07387176613102),
255   new kakao.maps.LatLng(35.179603949892, 129.07215203478785),
256   new kakao.maps.LatLng(35.17808117794007, 129.07172210195208),
257   new kakao.maps.LatLng(35.17597267737385, 129.0704323034447),
258   new kakao.maps.LatLng(35.1769097954884, 129.06785270642996),
259   new kakao.maps.LatLng(35.1769097954884, 129.0649864875247),
260   new kakao.maps.LatLng(35.17226121199627, 129.06398313189877),
261   new kakao.maps.LatLng(35.17796404045448, 129.06255020145522),
262   new kakao.maps.LatLng(35.179838219968, 129.06154702483838),
263   new kakao.maps.LatLng(35.18124382624465, 129.06283682334575),
264   new kakao.maps.LatLng(35.18464060772525, 129.06126040294785),
265   new kakao.maps.LatLng(35.18417289458215, 129.0579642512866),
266   new kakao.maps.LatLng(35.183937836997714, 129.0552413432468),
267   new kakao.maps.LatLng(35.18464060772525, 129.0528050571773),
268   new kakao.maps.LatLng(35.18522624535593, 129.05151525866992),
269   new kakao.maps.LatLng(35.188739982512395, 129.0492228354573),
270   new kakao.maps.LatLng(35.19119950811947, 129.0473592412573),
271   new kakao.maps.LatLng(35.19387338241955, 129.04477964424257),
272   new kakao.maps.LatLng(35.19354184424148, 129.04563950991417),
273   new kakao.maps.LatLng(35.19611833597631, 129.0458662661331),
274   new kakao.maps.LatLng(35.197874988034876, 129.04177011439202),
275   new kakao.maps.LatLng(35.198811853599665, 129.04005038304885)
];
var hole = [
276   new kakao.maps.LatLng(35.14752339588451, 129.01955416807806),
277   new kakao.maps.LatLng(35.14746342789317, 129.01976766851158),
278   new kakao.maps.LatLng(35.147348822275674, 129.01966662250487)
];
// 다각형을 생생하고 지도에 표시합니다
var polygon2 = new kakao.maps.Polygon({
280   map: map,
281   path: [path, hole], // 좌표 배열의 배열로 하나의 다각형을 표시할 수 있습니다
282   strokeWeight: 2,
283   strokeColor: '#b2bb22',
284   strokeOpacity: 0.8,
285   fillColor: '#f9f',
286   fillOpacity: 0.3
});
// 지도에 다각형을 표시합니다
polygon2.setMap(map);

```

</script>

</body>

</html>



## 총평 및 개선방안

이번 Team Project를 진행하며 웹 앱에 대한 기초와 AWS에 대한 기초를 알았다. 앞으로는 이번 프로젝트 보다 더욱 좋은 웹 앱을 만들 수 있을 것 같다.

### 아동 복지 센터는

지역사회 아동의 보호 교육, 건전한 놀이와 오락의 제공, 보호자와 지역사회의 연계 등 아동의 건전육성을 위하여 종합적인 아동복지서비스를 제공하는 곳이다.

아이들의 성정과 양육에 큰 도움이 되는 것 같아 부모님들이나 우리 같은 예비 아빠, 엄마들이 이용하기에 편리한 웹앱을 만들고 싶었다.

그러나 아직은 익숙치 못 한 클라우딩을 활용한 웹앱을 만들기에는 배움과 전문 지식이 많이 부족했고 공공API의 활용도가 많이 떨어졌다.

더욱더 지식과 경험을 쌓고 클라우딩 관련 어플리케이션을 만든다면 Gps 기능과 네비 게이션 기능 현 위치에서 근처에 있는 센터에 거리와 걸리는 시간 등 더욱 많은 실용성있는 기능을 추가 할 것이다.



# 팀원 별 담당파트



김은종

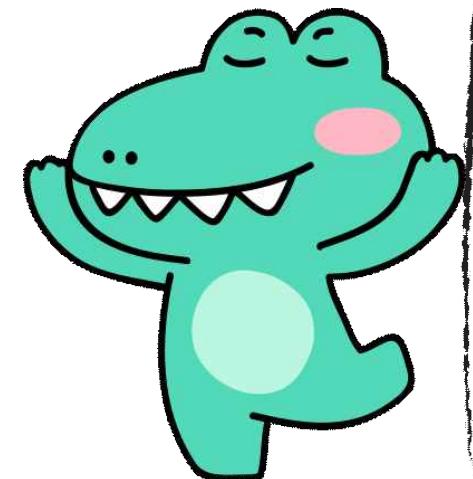
201695021



이도현

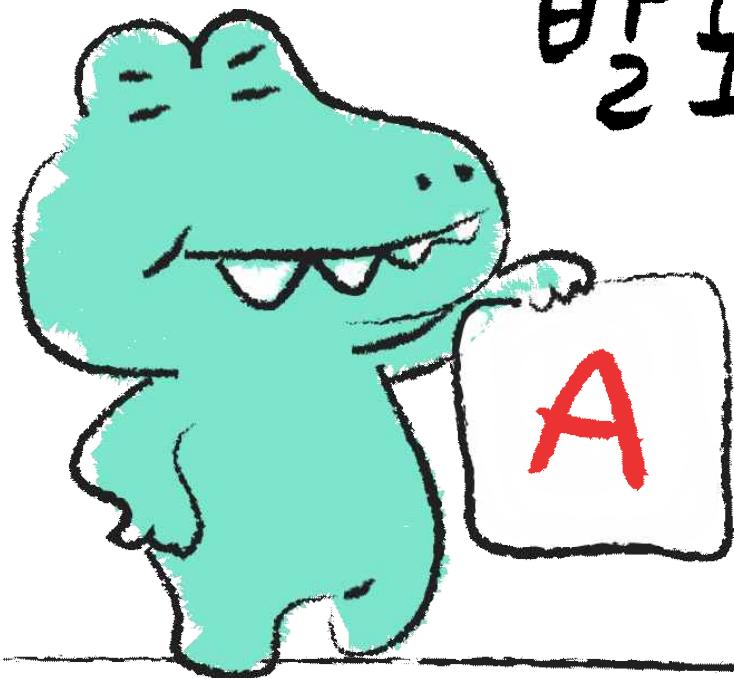
201695053

팀 프로젝트 방향성 결정, 공공 API 활용, AWS Lambda, S3,  
Dynamo DB 구축, html 코딩, PPT 작성



팀 프로젝트 방향성 결정, 공공 API 신청, 부산진구, 사상구 구역 표시  
코딩 작성, 보고서 작성

At Last



발표를 들팡 주셔서  
감사합니다

