

## **CASS : Cloud-based Automatic Sunvisor System**

**4조 : 정성윤, 최한슬**

## 프로젝트 개요

- 제목 : Cloud-based Automatic Sunvisor System (CASS)
- 주제 : 차량의 GPS, 방향, 시간 정보를 기반으로 태양의 위치를 계산 해 운전자 시야 내 눈부심을 예측하고 대응하는 시스템
- 동기 : 운전 중 눈부심은 사고 위험 요인 중 하나이나, 운전자가 운전 중 실시간으로 대처하기 어려움.



# 프로젝트 개요

---

- 목표
- 1. 한 점에서의 차량의 위치, 방향, 시간을 기준으로 태양의 위치 계산
- 2. 출발점, 도착점을 입력받고 경로 상에서의 태양의 위치 및 선바이저 조정 전개방향 설정
- 3. 클라우드를 통해 동시간대에 같은 경로를 지나는 다른 차량에 대한 계산 비용을 줄일 수 있도록 데이터 저장 기능 구현

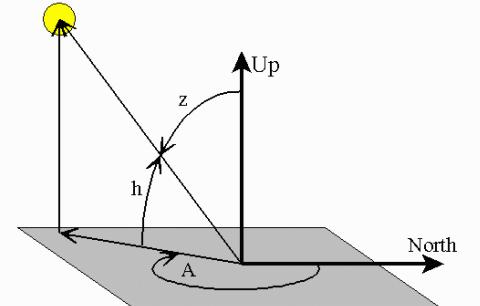
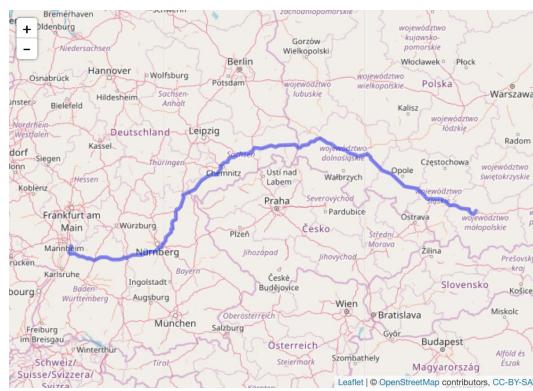
# 프로젝트 개요

- **유사제품: Bosch의 Virtual Visor**
  - 2020년 CES(국제전자제품박람회)에서 제안됨
- **구성**
  - 운전자의 얼굴에 비치는 소형 RGB 카메라와 전면의 투명한 LCD판으로 구성
  - AI를 이용해서 얼굴의 눈을 추적 후 정확히 햇빛이 비칠 때만 LCD 패널을 어둡게 한다.
- **한계점**
  - 항상 켜져있는 카메라 -> **프라이버시** 문제,
  - 실시간으로 눈 위치 추적 및 햇빛 감지 -> **높은 계산 비용**



# 시스템 구조 및 알고리즘

- **입력** : 출발지와 목적지의 **GPS 좌표**(latitude, longitude), 차량의 **방향**(degree), 현재 **시간**(UTC), 평균 속도 및 샘플 포인트간 거리
- **처리** : **OpenRouteService**의 API를 이용해서 최적의 경로를 계산 후, 각 지점마다 **NOAA**(미국 연방해양대기청)의 알고리즘에 기반해서 태양의 **방위각**(Azimuth)과 **고각**(elevation) 계산
- **출력** : 경로의 각 구간별 sunvisor의 배치 및 전체 경로에 대한 sunvisor의 배치



$h$  = elevation  
angle, measured  
up from horizon

$z$  = zenith angle,  
measured from  
vertical

$A$  = Azimuth angle,  
measured clockwise  
from North

# 시스템 구조 및 알고리즘

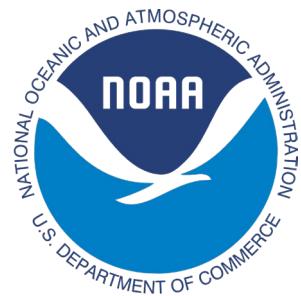
- NOAA(미국 연방해양대기청)의 Solar Calculator 사용
- Pysolar 라이브러리의 SPA보다 가벼워 시스템 부하 감소
- 약 2KB 정도의 가벼운 알고리즘

The solar zenith angle ( $\phi$ ) can then be found from the hour angle ( $ha$ ), latitude ( $lat$ ) and solar declination ( $decl$ ) using the following equation:

$$\cos(\phi) = \sin(lat)\sin(decl) + \cos(lat)\cos(decl)\cos(ha)$$

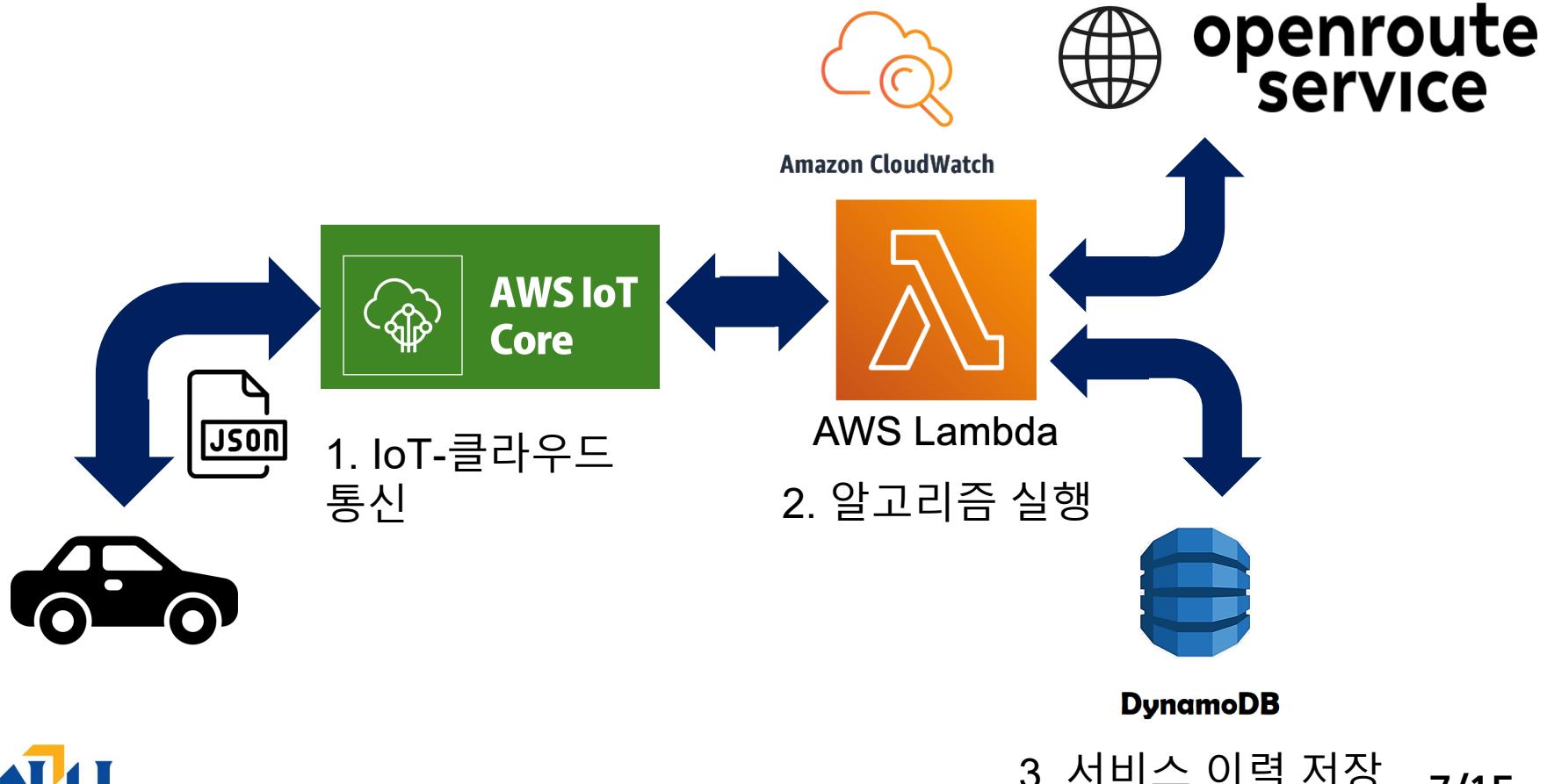
And the solar azimuth ( $\theta$ , degrees clockwise from north) is found from:

$$\cos(180 - \theta) = - \frac{\sin(lat)\cos(\phi) - \sin(decl)}{\cos(lat)\sin(\phi)}$$



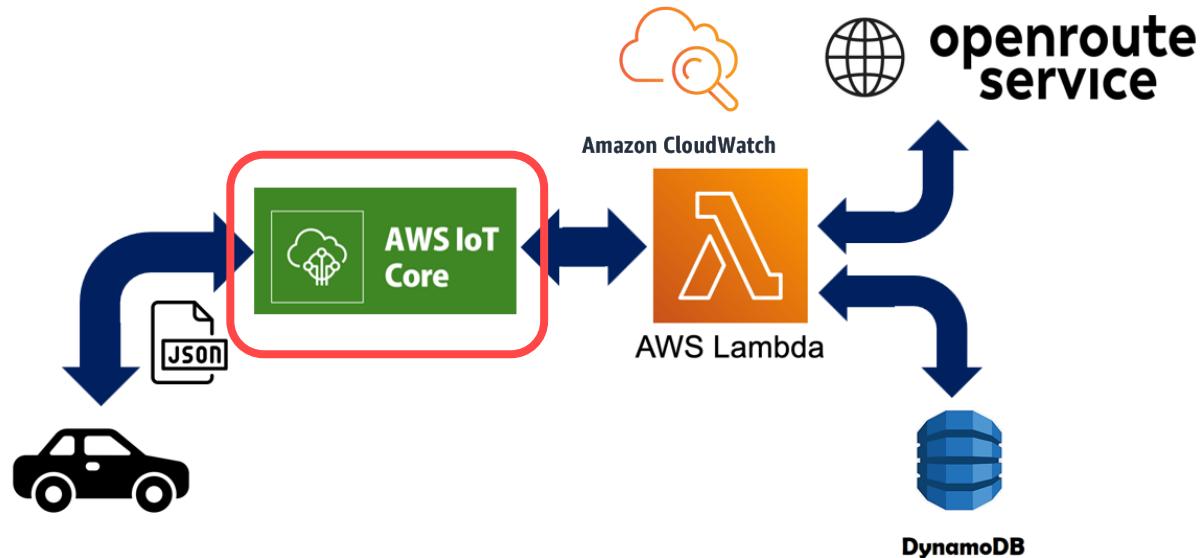
# 클라우드 서비스 활용

## 2-1. 최적 경로 계산



# 클라우드 서비스 활용

- **AWS IoT Core** : IoT 디바이스와 클라우드 애플리케이션을 안전하게 연결하는 완전 관리형 서비스
- 차량 클라우드 양방향 통신을 위한 **IoT Rule, Policy**를 규정
- MQTT 메시지 수신 시 자동으로 lambda 실행.

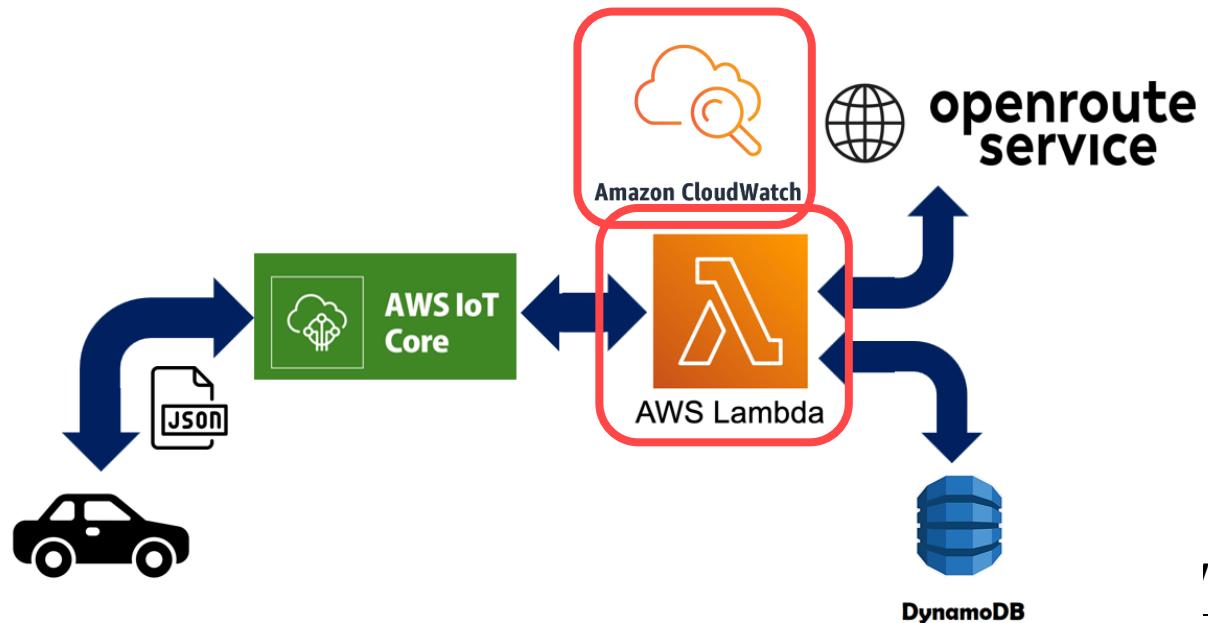


# 클라우드 서비스 활용

**AWS Lambda** : 서버리스 컴퓨팅 서비스, 실제 코드 실행

**AWS CloudWatch** : AWS 리소스 및 어플리케이션 모니터링 서비스.

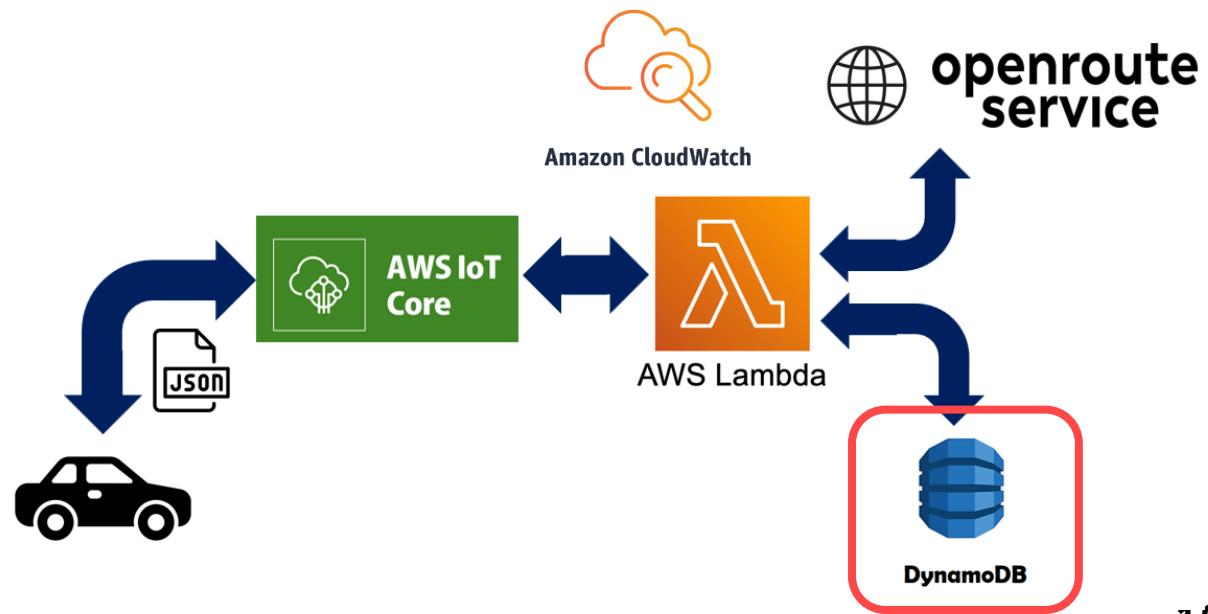
실행 시간, 메모리 사용량, 호출 횟수, 등 실시간으로 모니터링 가능

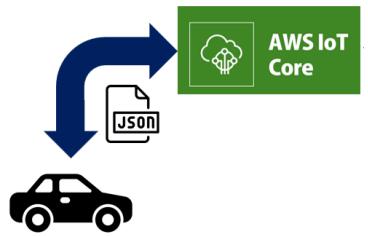


# 클라우드 서비스 활용

**Amazon DynamoDB** : NoSQL 데이터베이스 서비스.

차량별 서비스 내역 및 위치 기록 저장





# 실행 예시

## 입력

```
{"vehicle_id": "vehicle-GAILAB-003",
 "start": { "lat": 37.3, "lon": 126.6, "heading": 90,
   "timestamp_utc": "2025-11-15T07:00:00Z" },
 "destination": { "lat": 34.7, "lon": 127.7 },
 "speed_kmh": 50,
 "interval_km": 20}
```

## 출력

모드: Route

샘플링 포인트: 20개

전체 경로 추천: 우측(조수석) 선바이저 전개 (경로의 55.0%에서 필요)

통계: 우측 전개: 11회 (55.0%), 전면 전개: 8회 (40.0%), 권고 없음: 1회 (5.0%)

변경 지점:

- [0] 2025-11-15T01:00:00+00:00, 위치: (37.3778, 126.6352), 우측 전개
- [1] 2025-11-15T01:24:00+00:00, 위치: (37.3924, 126.8343), 전면 전개
- [9] 2025-11-15T04:36:00+00:00, 위치: (36.1735, 127.0522), 우측 전개
- [19] 2025-11-15T08:36:00+00:00, 위치: (34.7535, 127.7483), 권고 없음



DynamoDB

# 실행 예시

- DB에서의 차량의 위치 조회 및 추천 통계

## 차량의 최신 상태

차량: vehicle-GAILAB-003  
최종 업데이트: 2025-11-15T14:15:00+00:00  
위치: (35.1791, 129.0719)  
방향: 96.13753520804858°

차량 : vehicle-test-001  
최종 업데이트 : 2025-11-26T09:24:00:00+00  
위치:(37.3776, 126.6351)  
방향: 24°

차량 : vehicle-GAILAB-002  
최종 업데이트 : 2025-11-13T14:30:00:00+00  
위치:(51.3000, 0.0700)  
방향:124°

...

## 차량의 선바이저 비율 통계

차량 : vehicle-test-001  
총 23개 기록  
선바이저 전개 비율:  
권고 없음 : 18회(78.3%)  
앞쪽 : 0회 (0.00%)  
오른쪽 : 2회(8.7%)  
왼쪽 : 3회(13.0%)  
뒤쪽: 0회(0.00%)

차량: vehicle-GAILAB-003  
총 95개 기록  
선바이저 전개 비율:  
권고 없음: 33회 (34.7%)  
앞쪽: 19회 (20.0%)  
오른쪽: 42회 (44.2%)  
왼쪽: 1회 (1.1%)  
뒷쪽: 0회 (0.0%)

...



Amazon CloudWatch

# CloudWatch Log 결과

## Single point 모니터링

**Duration :** 272.32ms(0.3초 이내의 응답)

**Memory Size :** 86MB / 256 MB

### 로그 이벤트

아래의 필터 막대를 사용하여 로그 이벤트의 용어, 구문 또는 값을 검색하고 매칭할 수 있습니다. [필터 패턴에 대해 자세히 알아보기](#)

(C) 작업 ▾ 테일링 시작 지표 필터 생성

이벤트 필터링 - Enter 키를 눌러 검색		지우기	1분	30분	1시간	12시간	사용자 지정	UTC 시간대	디스플레이	⚙️
▶	타임스탬프		메시지							
			현재 이전 이벤트가 없습니다. <a href="#">재시도</a>							
▶	2025-11-26T07:38:38.495Z		INIT_START Runtime Version: python:3.10.v101 Runtime Version ARN: arn:aws:lambda:eu-north-1::runtime:6c8008e98a910e34349fc88f903d2ecdffcbf435318a92388cad2922...							
▶	2025-11-26T07:38:39.268Z		START RequestId: 8d83385a-bd3c-4cc5-99a7-08e6d115b059 Version: \$LATEST							
▶	2025-11-26T07:38:39.396Z		DynamoDB 저장 완료: vehicle-GAILAB-002 at 2025-11-26T04:30:40.462158+00:00							
▶	2025-11-26T07:38:39.539Z		차량으로 응답 전송 완료: vehicle/vehicle-GAILAB-002/command							
▶	2025-11-26T07:38:39.541Z		END RequestId: 8d83385a-bd3c-4cc5-99a7-08e6d115b059							
▶	2025-11-26T07:38:39.541Z		REPORT RequestId: 8d83385a-bd3c-4cc5-99a7-08e6d115b059 Duration: 272.32 ms Billed Duration: 1043 ms Memory Size: 256 MB Max Memory Used: 86 MB Init Duration:...							
			현재 최신 이벤트가 없습니다. 자동 재시도를 일시 중지했습니다. <a href="#">재개</a>							



Amazon CloudWatch

# CloudWatch Log 결과

## Route 계산 모니터링

**Duration :** 1176.62ms(1.2초 내외의 응답)

경로 계산(275ms) + DynamoDB 저장(879ms) + IoT응답 전송(20ms)

**Memory Size :** 87MB / 256 MB

```
▶ 2025-11-26T07:38:39.541Z      REPORT RequestId: 8d83385a-bd3c-4cc5-99a7-08e6d115b059 Duration: 272.32 ms Billed Duration: 1043 ms Memory Size: 256 MB Max Memory Used: 86 MB Init Duration:...
▶ 2025-11-26T07:40:35.601Z      START RequestId: d3a34c96-82c1-4502-80bd-caec170b80e4 Version: $LATEST
▶ 2025-11-26T07:40:35.602Z      Route 모드 감지: vehicle_id=vehicle-GAILAB-003
▶ 2025-11-26T07:40:35.602Z      Route 처리 시작: vehicle_id=vehicle-GAILAB-003
▶ 2025-11-26T07:40:35.877Z      Route 포인트 20개를 DynamoDB에 저장 중...
▶ 2025-11-26T07:40:36.756Z      Route DynamoDB 저장 완료: 20/20개 포인트
▶ 2025-11-26T07:40:36.756Z      Route 응답 크기: 7017 bytes (6.85 KB)
▶ 2025-11-26T07:40:36.776Z      Route 응답 전송 완료: vehicle/vehicle-GAILAB-003/command
▶ 2025-11-26T07:40:36.778Z      END RequestId: d3a34c96-82c1-4502-80bd-caec170b80e4
▶ 2025-11-26T07:40:36.778Z      REPORT RequestId: d3a34c96-82c1-4502-80bd-caec170b80e4 Duration: 1176.61 ms Billed Duration: 1177 ms Memory Size: 256 MB Max Memory Used: 87 MB
```

현재 최신 이벤트가 없습니다. 자동 재시도를 일시 중지했습니다. [재개](#)

---

## **QnA**