

# AWS 기반 AI 애플리케이션 개발

2022. 9. (4일, 32H)

정 준 수 PhD

# AWS 기반 AI 애플리케이션 개발 교육 일정

| 일정  | 학습 내용           | 상세 내용                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1일차 | 과정 소개           | AI 관련 AWS 서비스의 개요 및 과정 소개                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|     |                 | AWS IAM(계정관리), EC2, S3 사용하기<br>Python 개발환경 구축: VSCode, AWS CLI, 가상환경 구축                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|     | 점심시간            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|     | AWS Chalice     | AWS CLI 연결 - WSL2 - Ubuntu 20.04 설치와 외부 터미널 연결<br>S3를 이용한 Static Web Hosting 만들기<br>(Bucket Policy 편집 - <a href="https://awspolicygen.s3.amazonaws.com/policygen.html">https://awspolicygen.s3.amazonaws.com/policygen.html</a> )<br>AWS Resources, Authentication, Authorization & access control, Policy, Role, Action 이해하기<br>( <a href="https://docs.aws.amazon.com/mediaconvert/latest/ug/auth_access_introduction.html">https://docs.aws.amazon.com/mediaconvert/latest/ug/auth_access_introduction.html</a> )<br>( <a href="https://docs.aws.amazon.com/mediaconvert/latest/ug/learn-more-iam.html">https://docs.aws.amazon.com/mediaconvert/latest/ug/learn-more-iam.html</a> ) |
| 2일차 | AWS Lambda함수    | 서버 리스 환경 이해하기: AWS Lambda 함수, API Gateway, DynamoDB                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|     | 점심시간            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|     | AWS Rekognition | Severless Microframework application 구현<br>AWS Chalice( <a href="https://aws.github.io/chalice/">https://aws.github.io/chalice/</a> ) Serverless 구현<br>- <a href="https://aws.github.io/chalice/quickstart.html">https://aws.github.io/chalice/quickstart.html</a><br>- REST API 실습 ( <a href="https://aws.github.io/chalice/tutorials/basicrestapi.html">https://aws.github.io/chalice/tutorials/basicrestapi.html</a> )<br>Cloud logwatch - Log 분석<br>[실습예제] Python boto3 활용 - Rekognition API                                                                                                                                                                                        |
|     |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 3일차 | AWS Rekognition | Vision Tasks 정의<br>AWS Rekognition<br>API 연동<br>[실습예제] 이미지 속 텍스트 번역기                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|     | 점심시간            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|     | AWS Reanslate   | AWS Translate<br>API 연동<br>[실습예제] 만능 통역기 만들기                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 4일차 | AWS Transcribe  | AWS Transcribe<br>API연동<br>AWS Polly<br>[실습예제 1] 다국어 지하철역 안내방송 만들기<br>[실습예제 2] 연락처 관리기 만들기                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|     | 점심시간            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|     | AWS Comprehend  | AWS Comprehend<br>API연동<br>[실습예제] 연락처 도우미 봇 만들기                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

Cloud Computing 환경이란?

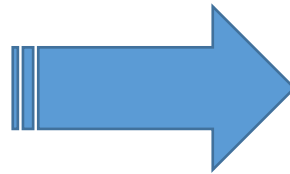
Hardware의 추상화

# 현금(Money)의 추상화(Abstraction) 사례

Data



현금(Money)



인터넷 뱅킹

Fetures: 교환가치 + 은행의 기능



신용카드

Fetures: 교환가치 + 신용

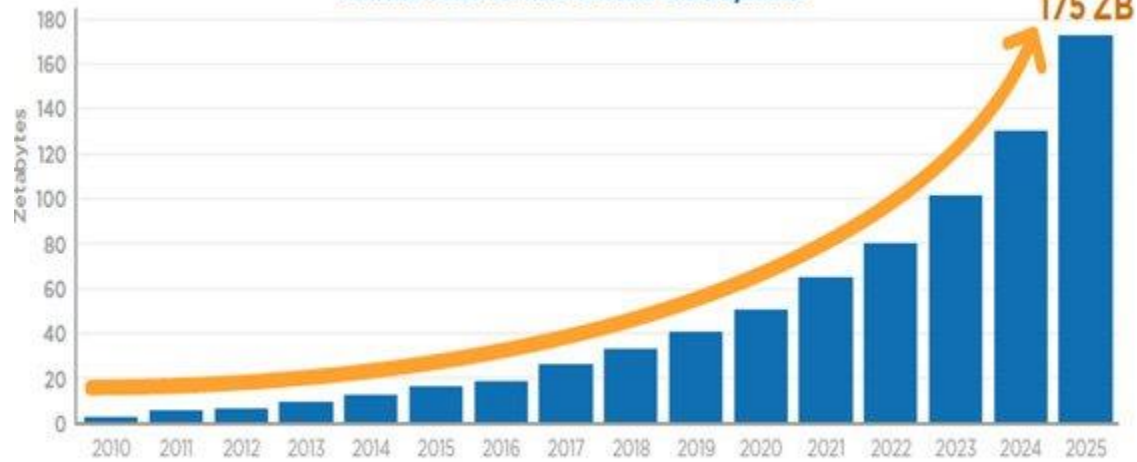


티머니

Fetures: 교환가치 +  
버스 지하철 갈아탈 수 있다

## Digital Data의 증가량

Annual Size of the Global Datasphere



Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018



데이터의 증가량: 2025년 전 세계 데이터 용량의 예상 규모

# 이번 과정에서 다룰 내용들

- **Severless 환경:** Lambda 함수, APIGateway, DynamoDB  
Chalice framework

- **AWS AI 서비스 기능**

Rekognition

Translate

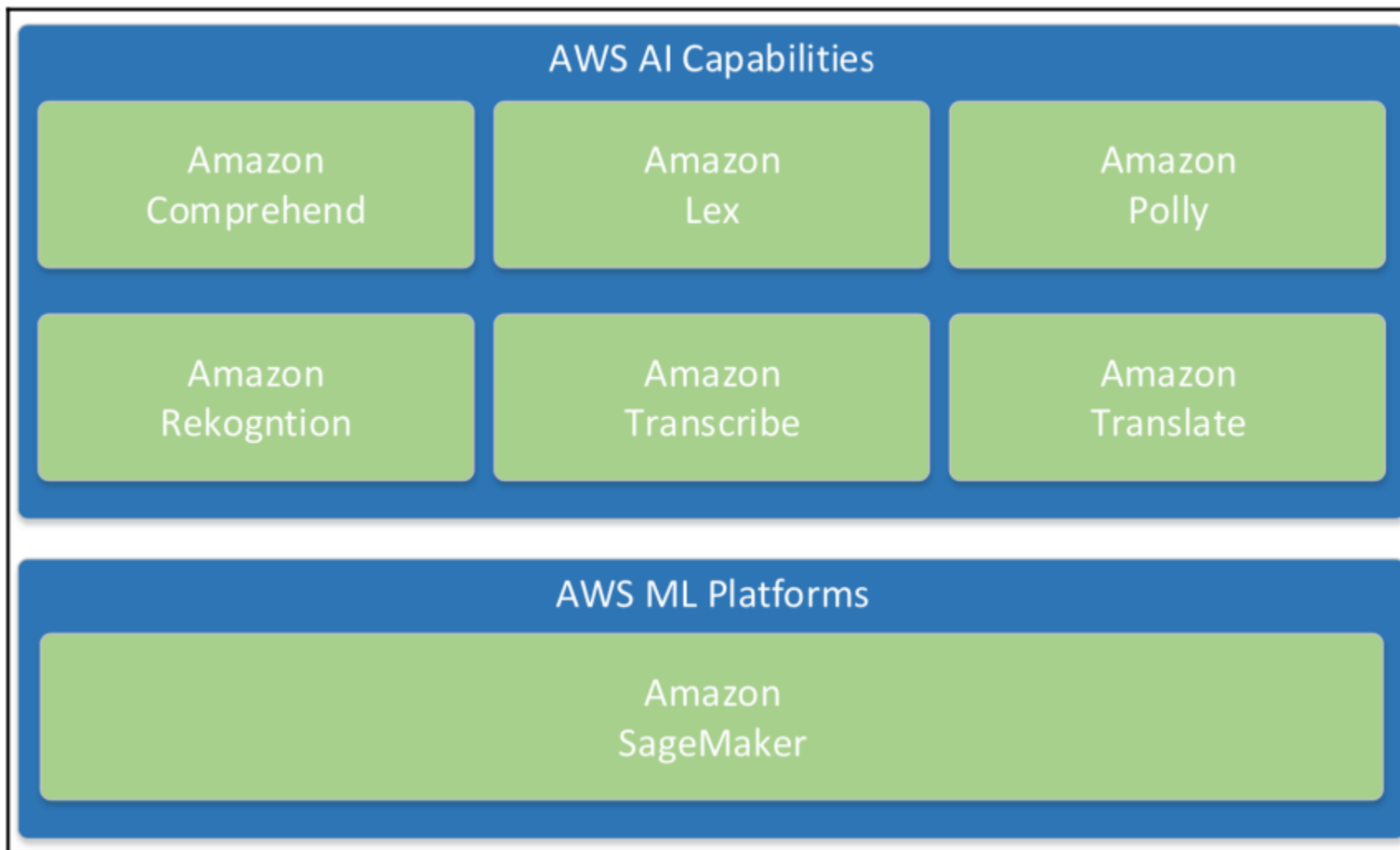
Transcribe

Polly

Comprehend

Lex

# AWS AI 개발 App과 ML Platform 구성도



# AWS AI 서비스 구성

- **AWS Comprehend**

NLP 서비스로, ML 기법을 통해 텍스트에서 단어 간의 관계와 같은 인사이트를 찾아준다. 즉, 비정형의 텍스트 더미에서 주제, 주요 문구, 감정 상태 등 가치 있는 정보를 발견하거나 엔티티(회사명, 사람, 주소 등) 식별, 자동 문서 분류 등의 기능을 제공한다.

- **AWS Lex**

애플리케이션에 음성이나 텍스트를 사용하는 대화형 인터페이스를 구축해 주는 서비스이다. 심층학습 기반의 자동 음성 인식ASR, Automatic Speech Recognition 및 자연어 이해 NLU, Natural Language Understanding 기법으로 음성을 텍스트로 변환(음성 인식)하고 텍스트에 있는 의도를 파악하는 등의 기능을 갖추고 있다. 음성 비서 Voice Assistant 인 아마존 알렉사Alexa 내부에서 사용하는 것과 동일한 기술이며 챗봇 등의 애플리케이션을 개발할 때 이용할 수 있다.

- **AWS Polly**

텍스트를 사람의 음성처럼 합성하는 기능이 있어 애플리케이션이 이 서비스를 사용해 사람의 목소리를 출력할 수 있다. 서비스의 내부 엔진에 있는 텍스트-음성 변환 Text-to-Speech 기술은 다양한 언어, 성별, 억양 별로 음성을 합성할 수 있는 고급 심층학습 기법을 사용했다.



# AWS AI 서비스 구성(계속)

- **AWS Rekognition**

이미지와 비디오를 분석해서 사물, 사람, 텍스트, 장면, 활동 등을 식별하는 서비스로, 다양한 종류의 애플리케이션 개발이 가능 하도록 정확한 안면 분석 및 인식 기능도 갖추고 있다. AWS는 정확성을 높이기 위해 서비스의 내부 엔진에 있는 심층 신경망을 방대한 양의 이미지와 비디오로 훈련했다.

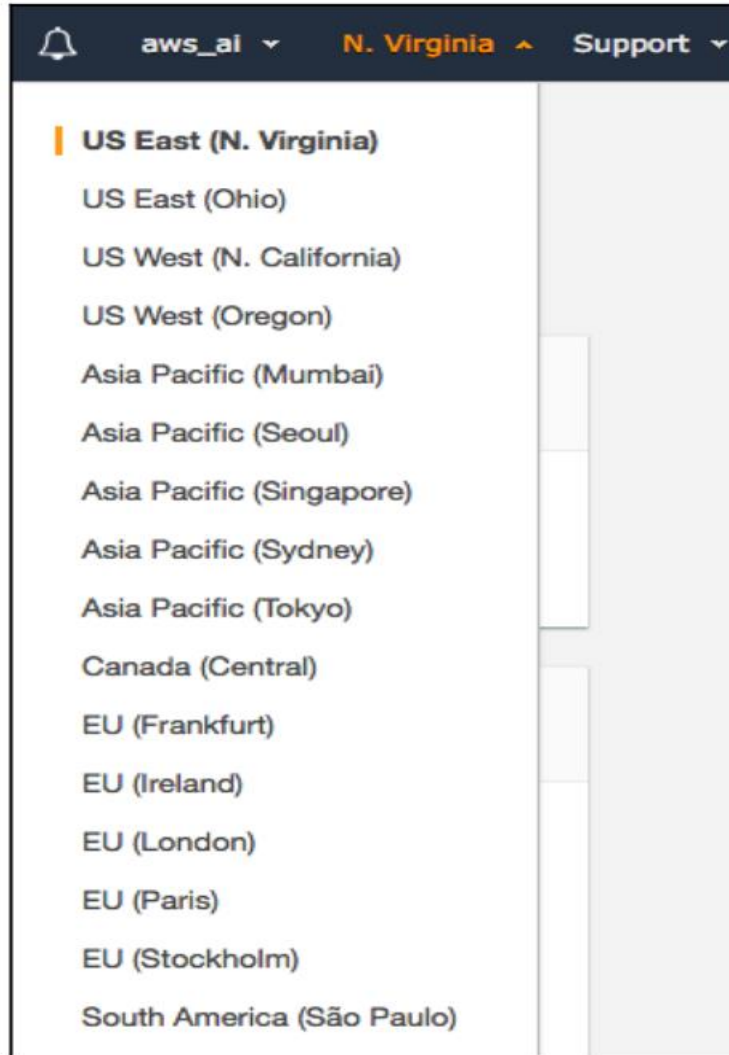
- **AWS Transcribe**

음성-텍스트 변환Speech-to-Text 기능을 제공하는 ASR 서비스로, 애플리케이션은 이 서비스를 사용해 실시간으로 저장된 오디오 파일 또는 라이브 오디오 스트림을 분석해서 텍스트로 변환할 수 있다.

- **AWS Translate**

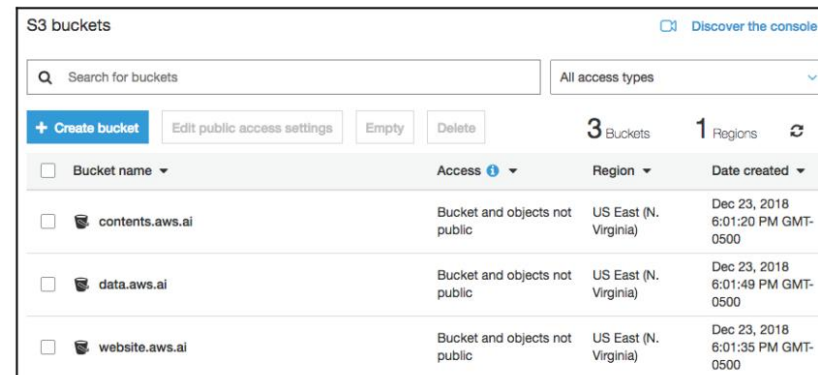
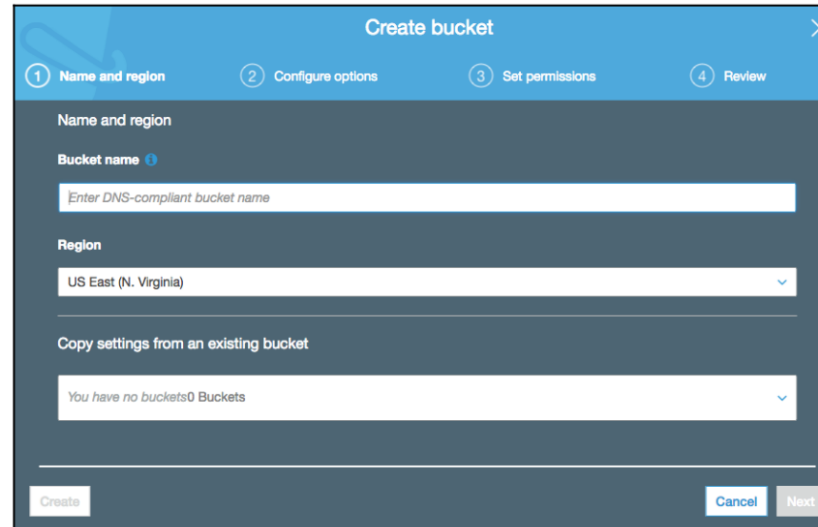
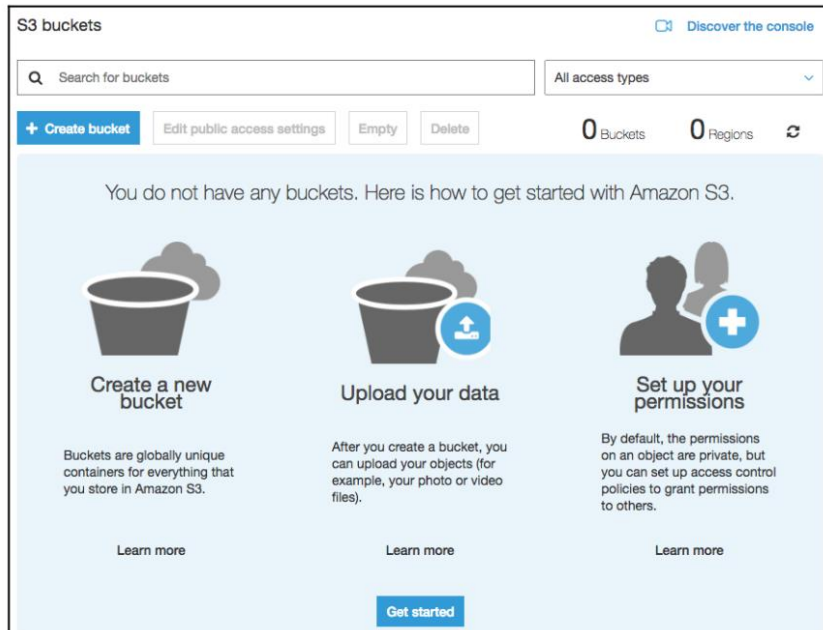
다양한 언어를 정확하고 자연스럽게 번역할 수. 있는 심층 신경망 모형 기반의 기계 번역 서비스이다. 브랜드 이름, 제품 이름 등을 사용자가 따로 정의해 두면 번역 시에 반영할 수도 있다.

# AWS 서비스 Region 선택하기



각 지역에 클라우드 데이터 센터를 구축하고 있는데 Region별로 AWS 서비스 배포 현황이 다르다. 따라서, 모든 Region에서 전체 AWS 서비스를 사용할 수 있는 것은 아니다.

# AWS – S3 사용하기(버킷 만들기)



- 1.버킷 만들기 클릭
- 2.버킷 이름, 리전 입력
- 3.퍼블릭 액세스 설정
- 4.버킷버전관리, 기본 암호화
- 5.버킷 생성 확인
- 6.업로드 시도
- 7.업로드된 사진 확인
- 8.업로드 이미지 확인하기

# AWS Access Key Pair 생성

The screenshot shows the AWS IAM Management Console for the user 'me'. The left sidebar contains navigation links for '대시보드', '액세스 관리', '사용자 그룹', '사용자', '역할', '정책', '자격 증명 공급자', '계정 설정', '보고서 액세스', '액세스 분석기', '아카이브 규칙', '분석기', '설정', '자격 증명 보고서', '조직 활동', and 'SCP(서비스 제어 정책)'. The main content area shows the '요약' (Summary) tab for the user 'me', displaying details like '사용자 ARN', '경로', and '생성 시간'. The '로그인 자격 증명' (Login Credentials) tab is selected, showing the '요약' (Summary) section with a console login link. Below this, the '액세스 키' (Access Keys) section is visible, containing a warning about key security and a red circle around the '액세스 키 만들기' (Create Access Key) button. At the bottom, there is a table of existing access keys and a section for 'AWS CodeCommit에 대한 SSH 키' (SSH Keys for AWS CodeCommit).

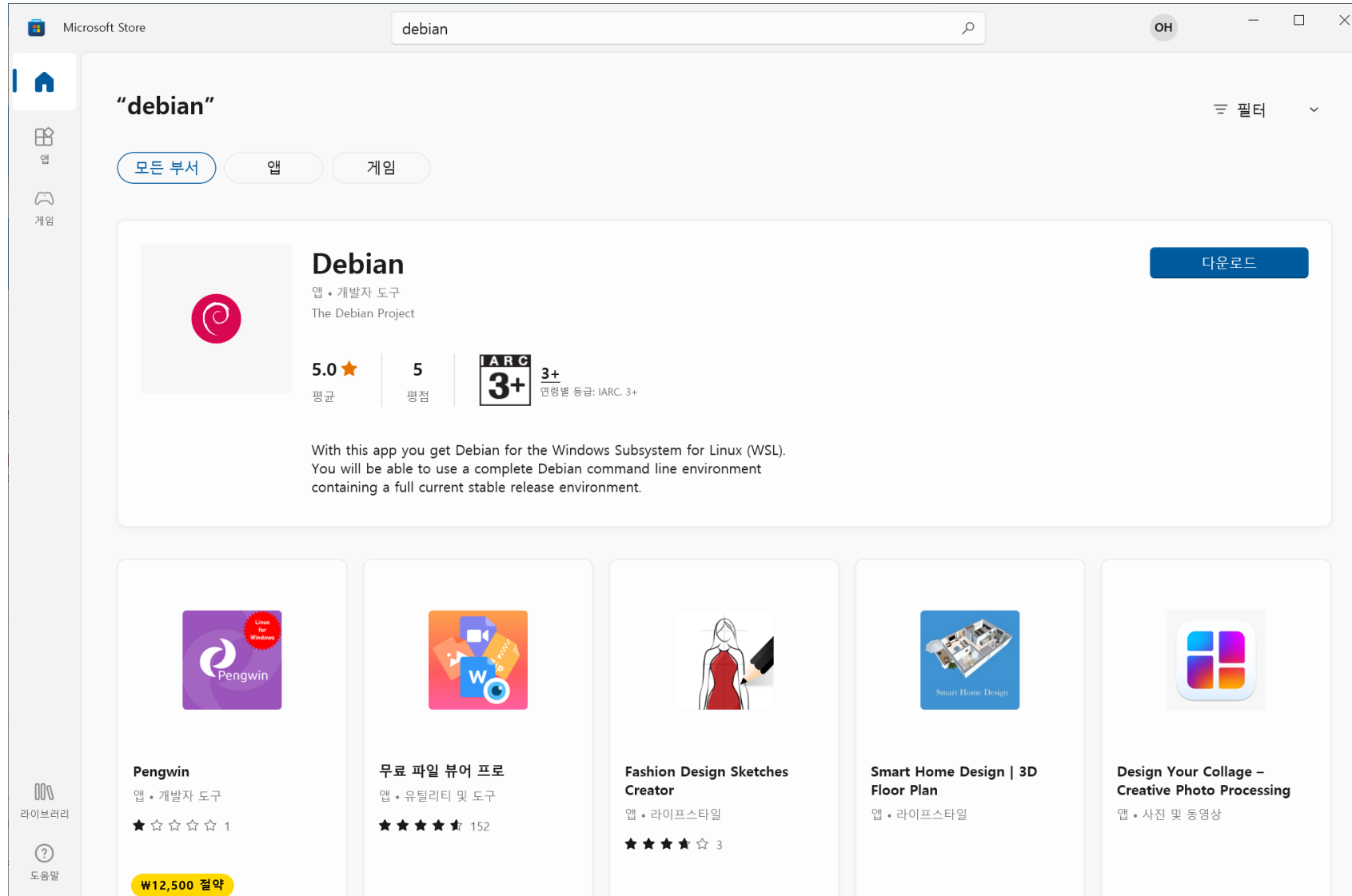
| 액세스 키 ID             | 생성 완료                     | 마지막 사용                     | 상태          |
|----------------------|---------------------------|----------------------------|-------------|
| AKIATDNDJNAD6HFRT3WC | 2022-02-05 19:58 UTC+0900 | ap-northeast-1에서 2022-0... | (활성)   비활성화 |

# PC(Local) 환경구축: WSL2

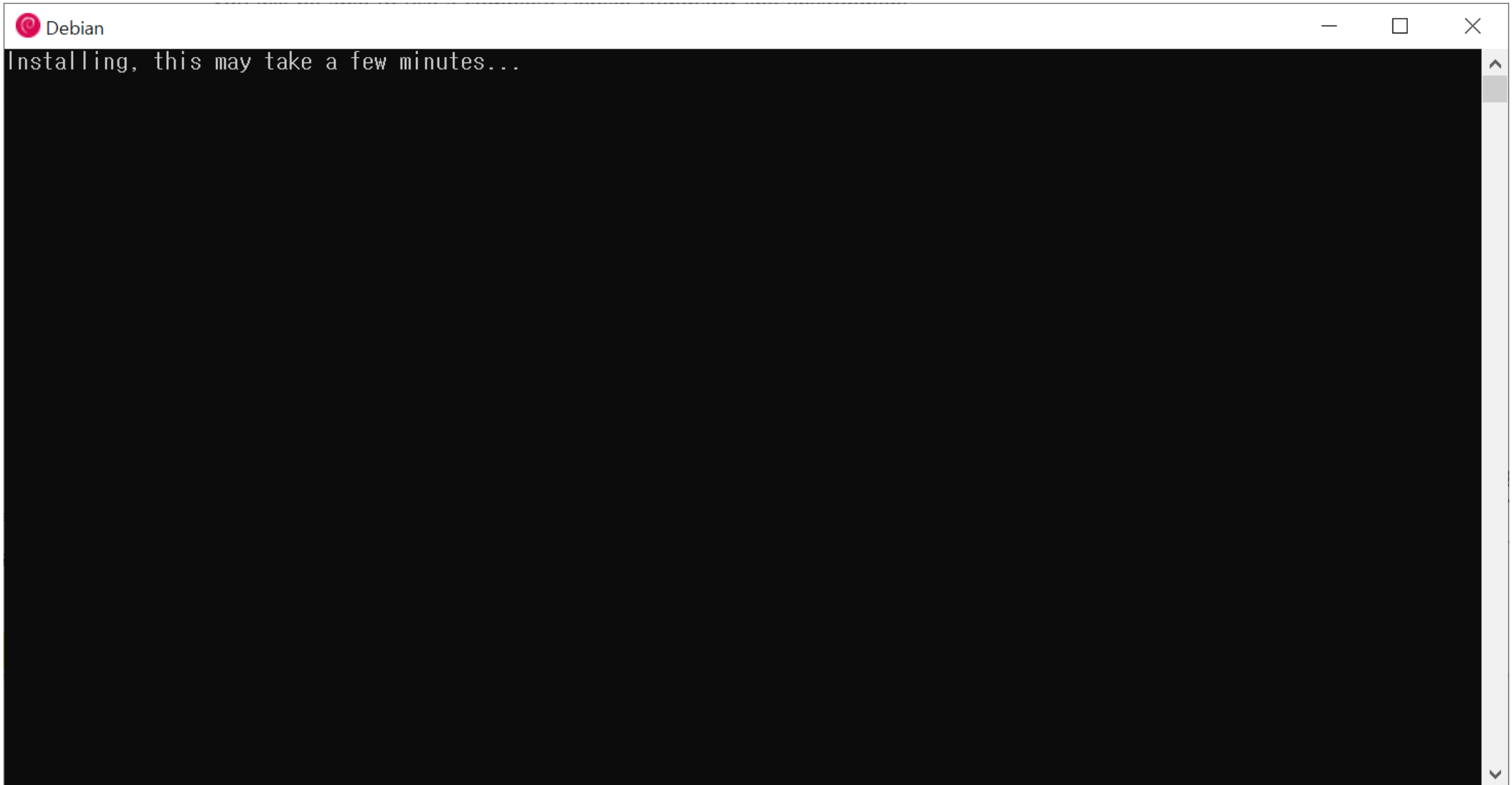
윈도우 환경에서 원하는 Linux를 실행

1. 관리자 권한으로 명령 프롬프트(CMD) 실행
2. <https://docs.microsoft.com/ko-kr/windows/wsl/tutorials/gui-apps>
3. C> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-windows-Subsystem-Linux
4. 또는 <https://ivyit.tistory.com/264>
5. Microsoft Store 열기
6. Ubuntu 또는 Debian 선택

# Microsoft Store에서: Linux 선택



# Ubuntu 또는 Debian 설치 과정



# Ubuntu 또는 Debian 설치 과정

```
oakyo@DESKTOP-HAAIOJO: ~  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows username.  
For more information visit: https://aka.ms/wslusers  
Enter new UNIX username: oakyo  
New password:  
Retype new password:  
passwd: password updated successfully  
Installation successful!  
oakyo@DESKTOP-HAAIOJO:~$
```



# Python 설치 과정

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

```
$ sudo apt-get install wget
```

**Anaconda 설치 :** <https://repo.anaconda.com/archive/>

```
$ wget https://repo.anaconda.com/archive/Anaconda3-2021.11-Linux-x86\_64.sh
```

```
$ bash Anaconda3-2021.11-Linux-x86_64.sh
```

**.profile 에 PATH 추가**

```
$ export PATH=/home/{login ID}/anaconda3/bin:$PATH
```

```
$ . .profile
```

## Python version 확인

\$ python -version

## PC(Local) 파일 확인

\$ explorer.exe .

# AWS CLI 설치하기

**\$ sudo pip install awscli**

\$ sudo apt-get install python3-pip

**\$ aws --version**

**\$ aws configure**

**설치 확인 방법**

**\$ cd**

**\$ ls -al**

**\$ aws s3 ls**

Ap-northeast-2  
json

# AWS CLI 설치 확인

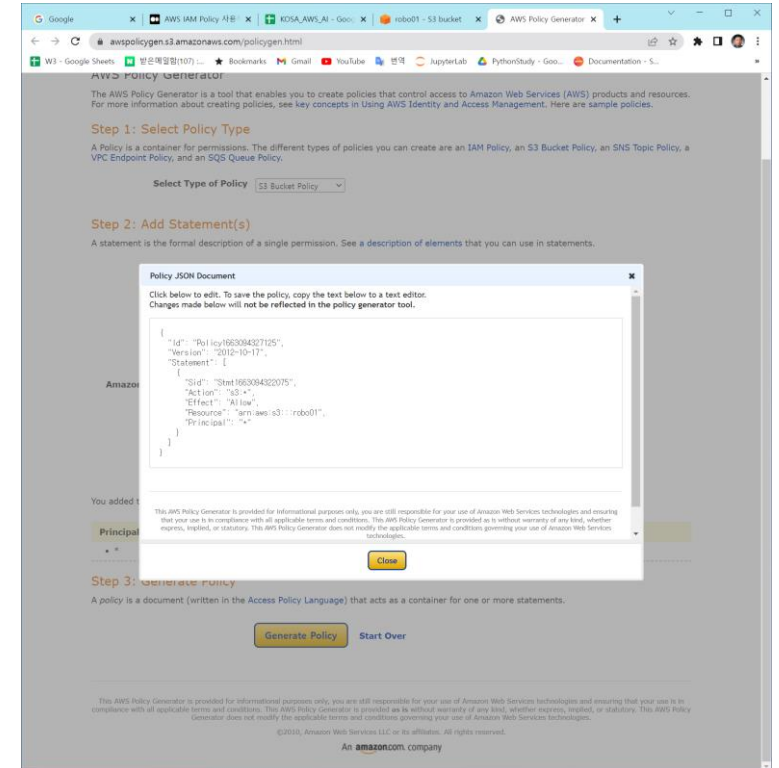
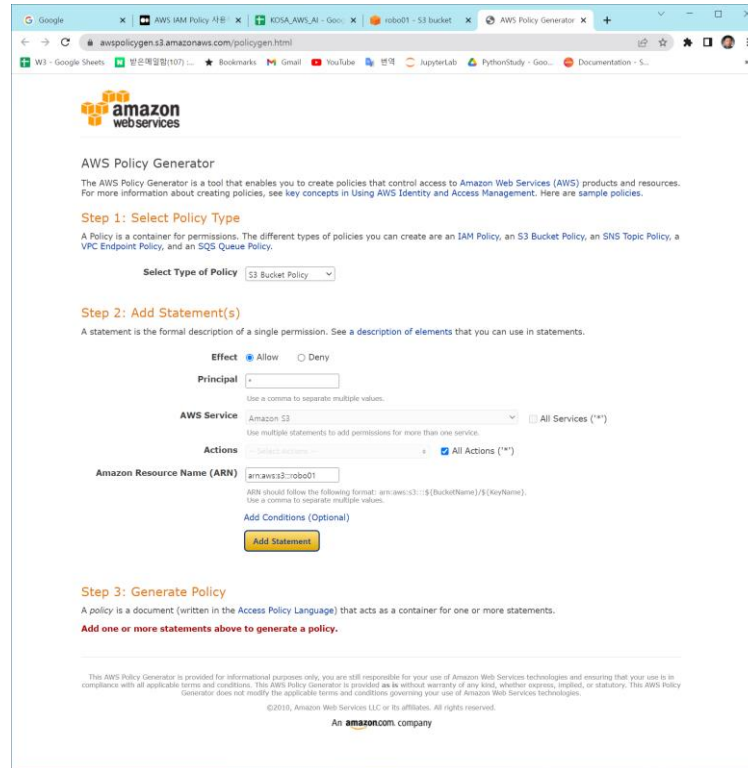
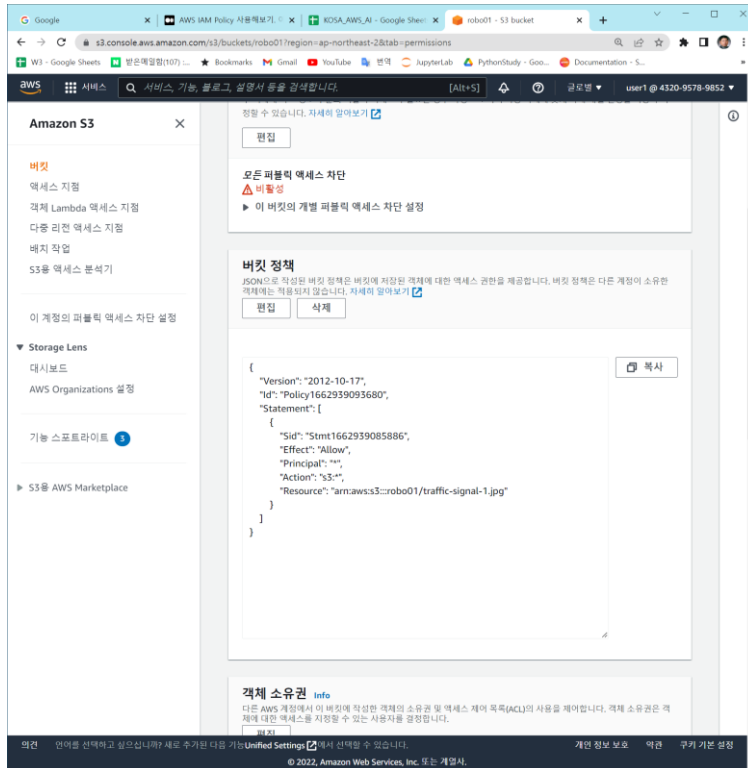
```
$ aws s3 ls s3://kosa-aws
```

```
$ aws rekognition detect-labels --image  
S3Object=W{Bucket=robo01,Name=traffic-signal-1.jpgW}
```

```
KOSA_AWS_AI/CLI/
```

```
$ python object_detection_demo.py
```

# AWS IAM Policy 사용해보기 – Bucket 정책 편집



<https://medium.com/@labcloud/aws-iam-policy-%EC%82%AC%EC%9A%A9%ED%95%B4%EB%B3%B4%EA%B8%B0-ca93cb6c868>

# Chalice Static Web Hosting

The screenshot shows the AWS S3 console interface for configuring static website hosting on a bucket named 'kosa-aws'. The left sidebar contains navigation options like '버킷' (Buckets), 'Storage Lens', and 'S3용 AWS Marketplace'. The main content area is titled '정적 웹 사이트 호스팅 편집' (Edit Static Website Hosting) and includes the following sections:

- 정적 웹 사이트 호스팅**: A note stating that the bucket is used for hosting websites and a link to '자세히 알아보기' (Learn more).
- 정적 웹 사이트 호스팅**: Radio buttons for '비활성화' (Deactivated) and '활성화' (Activated), with '활성화' selected.
- 호스팅 유형**: Radio buttons for '정적 웹 사이트 호스팅' (Static website hosting) and '객체에 대한 요청 리디렉션' (Redirect requests to objects). '정적 웹 사이트 호스팅' is selected, with a note that bucket endpoints are used as website URLs and a link to '자세히 알아보기' (Learn more).
- 고객이 웹 사이트 엔드포인트의 콘텐츠에 액세스할 수 있게 하려면 모든 콘텐츠를 공개적으로 읽기 가능하도록 설정해야 합니다.**: A warning box stating that to allow public access to website endpoint content, all content must be made public and bucket public access settings must be edited. It includes a link to 'Amazon S3 퍼블릭 액세스 차단 사용' (Use Amazon S3 public access blocking) for reference.
- 인덱스 문서**: A text input field for the index document, containing 'index.html'.
- 오류 문서 - 선택 사항**: A text input field for the error document, containing 'error.html'.
- 리디렉션 규칙 - 선택 사항**: A note stating that JSON-based redirect rules are automatically created for specific content and a link to '자세히 알아보기' (Learn more).

The bottom of the console shows a table with one entry, '1', and a footer with copyright information: '© 2022, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정'.

# Chalice Static Web Hosting

The screenshot shows the Amazon S3 console interface for configuring static website hosting on a bucket named 'static-web-me' in the 'ap-northeast-1' region. The breadcrumb navigation is 'Amazon S3 > static-web-me > 정적 웹 사이트 호스팅 편집'. The main title is '정적 웹 사이트 호스팅 편집' with an 'Info' link.

**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 엔드포인트를 웹 주소로 사용합니다. [자세히 알아보기](#)  
☐ 객체에 대한 요청 리디렉션  
요청을 다른 버킷 또는 도메인으로 리디렉션합니다. [자세히 알아보기](#)

**고객이 웹 사이트 엔드포인트의 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 참조하십시오.**

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

**오류 문서 - 선택 사항**  
오류가 발생하면 반환됩니다.

**리디렉션 규칙 - 선택 사항**  
JSON으로 작성된 리디렉션 규칙은 특정 콘텐츠에 대한 웹 페이지 요청을 자동으로 리디렉션합니다. [자세히 알아보기](#)

1

의견 한국어 ▼ © 2022, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

# Bucket 소유자의 정식 ID 확인

The screenshot shows the AWS S3 console interface. The left sidebar contains navigation links for '버킷' (Buckets), '액세스 지점' (Access Points), '객체 Lambda 액세스 지점' (Object Lambda Access Points), '다중 리전 액세스 지점' (Multi-Region Access Points), '배치 작업' (Batch Operations), 'S3용 액세스 분석기' (S3 Access Analyzer), '이 계정의 퍼블릭 액세스 차단 설정' (Public Access Block Settings), 'Storage Lens' (with sub-links '대시보드' and 'AWS Organizations 설정'), '기능 스포트라이트' (3 items), and 'S3용 AWS Marketplace'.

The main content area has a title 'Amazon S3' and a close button. It features two informational messages:

- 이 버킷에 대해 퍼블릭 액세스 차단 설정이 활성화되어 있기 때문에 퍼블릭 액세스가 차단됩니다.** 활성화된 설정을 확인하려면 이 버킷의 퍼블릭 액세스 차단 설정을 확인하세요. [Amazon S3 퍼블릭 액세스 차단 사용](#)에 대해 자세히 알아보기
- 중복 피부여자에 대한 통합 액세스 권한 부여가 콘솔에 표시됩니다.** 전체 ACL 목록을 보려면 Amazon S3 REST API, AWS CLI 또는 AWS SDK를 사용하세요.

Below the messages is a table titled '버킷 ACL' (Bucket ACL) with columns '객체' (Object) and '버킷 ACL' (Bucket ACL). The table lists several groups and their permissions:

| 객체                            | 버킷 ACL |
|-------------------------------|--------|
| 정식 ID 복사됨 (AWS 계정)            | 읽기, 쓰기 |
| 모든 사람(퍼블릭 액세스)                | -      |
| 인증된 사용자 그룹(AWS 계정이 있는 모든 사용자) | -      |
| S3 로그 전달 그룹                   | -      |

Below the table is a section for 'CORS(Cross-origin 리소스 공유)' with a description and a 'JSON으로 작성된 CORS 구성은 한 도메인에 로드되어 다른 도메인의 리소스와 상호 작용하는 클라이언트 웹 애플리케이션에 대한 방법을 정의합니다.' link. There is a '편집' (Edit) button and a '표시할 구성 없음' (No configurations to display) message with a '복사' (Copy) button.

The footer contains the text: '© 2022, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정'.



# ACL(엑세스 제어 목록 확인)

Google

static-web-me - S3 bucket

객체 탐지기

403 Forbidden

s3.console.aws.amazon.com/s3/bucket/static-web-me/property/acl/edit?region=ap-northeast-1

aws 서비스 서비스, 기능, 블로그, 설명서 등을 검색합니다. [Alt+S]

글로벌 me @ 2134-8020-4295

ACL(엑세스 제어 목록)

다른 AWS 계정에 기본 읽기/쓰기 권한을 부여합니다. 자세히 알아보기

| 피부여자                                                                                           | 객체                                                                               | 버킷 ACL                                                                           |
|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| 버킷 소유자(AWS 계정)<br>정식 ID: 963bb1530939f6612d66a80617f7bf65ecfe113275038ca6a52eefbca0658053      | <input checked="" type="checkbox"/> 나열<br><input checked="" type="checkbox"/> 쓰기 | <input checked="" type="checkbox"/> 읽기<br><input checked="" type="checkbox"/> 쓰기 |
| 모든 사람(퍼블릭 액세스)<br>그룹: http://acs.amazonaws.com/groups/global/AllUsers                          | <input type="checkbox"/> 나열<br><input type="checkbox"/> 쓰기                       | <input checked="" type="checkbox"/> 읽기<br><input type="checkbox"/> 쓰기            |
| 인증된 사용자 그룹(AWS 계정이 있는 모든 사용자)<br>그룹: http://acs.amazonaws.com/groups/global/AuthenticatedUsers | <input type="checkbox"/> 나열<br><input type="checkbox"/> 쓰기                       | <input type="checkbox"/> 읽기<br><input type="checkbox"/> 쓰기                       |
| S3 로그 전달 그룹<br>그룹: http://acs.amazonaws.com/groups/s3/LogDelivery                              | <input type="checkbox"/> 나열<br><input type="checkbox"/> 쓰기                       | <input type="checkbox"/> 읽기<br><input type="checkbox"/> 쓰기                       |

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

자세히 알아보기

☒ 이러한 변경 사항이 객체와 버킷에 미치는 영향을 이해합니다.

다른 AWS 계정에 대한 액세스

리소스와 연결된 다른 AWS 계정이 없습니다.

피부여자 추가

취소 변경 사항 저장

의견 한국어

© 2022, Amazon Web Services, Inc. 또는 계열사. 개인 정보 보호 약관 쿠키 기본 설정

# AWS 서버리스(Serverless)

서버리스([Serverless: The Serverless Application Framework](#))는 AWS 람다를 비롯해 애저 Function, 구글 클라우드 Function 등을 지원하는 범용 서버리스 프레임워크입니다.

AWS 람다의 경우, 람다 함수 배포 뿐 아니라 클라우드 포메이션 스택을 통해 **IAM, DynamoDB, S3, 클라우드 와치 이벤트(CloudWatch Events)** 등 아마존 리소스를 함께 지원하고 있습니다.

# AWS Lambda

AWS 람다AWS Lambda는 [아마존 웹 서비스](#)Amazon Web Services에서 제공하는 서버리스 컴퓨팅 서비스입니다.

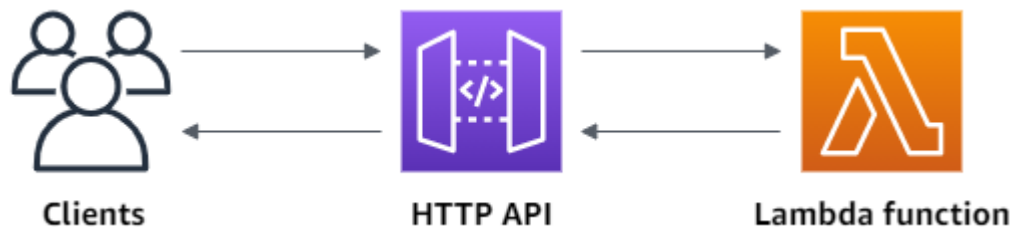
2014년 11월 AWS 리인벤트AWS re:Invent에서 처음 발표 되었습니다. 서버리스 컴퓨팅은 애플리케이션을 실행하기 위한 별도의 서버 셋업 없이 곧바로 코드를 실행해주는 서비스를 의미하며, 고정 비용 없이 사용 시간에 대해서만 비용이 발생합니다. 아마존 EC2Amazon EC2는 현재 초 단위로 비용을 계산하는 반면\* 람다는 1ms당 요금을 계산해 정확히 사용한만큼만 비용이 발생합니다.

# API Gateway

이번 시작하기 연습에서는 서버리스 API를 생성합니다. 서버리스 API를 사용하면 서버를 프로비저닝하고 관리하는 데 시간을 소비하는 대신 애플리케이션에 집중할 수 있습니다. 이 연습은 완료하는 데 20분을 넘지 않으며 [AWS 프리 티어](#) 내에서 가능합니다.

먼저, AWS Lambda 콘솔을 사용하여 Lambda 함수를 생성합니다. 그런 다음 API Gateway 콘솔을 사용하여 HTTP API를 생성합니다. 그런 다음 API를 호출합니다.

HTTP API를 호출하면 API Gateway는 요청을 Lambda 함수로 라우팅합니다. Lambda는 Lambda 함수를 실행하고 API Gateway에 응답을 반환합니다. 그리고 나면 API Gateway가 응답을 반환합니다.



# Chalice Python Serverless Microframework

<https://github.com/aws/chalice>

```
$ pip install chalice
```

```
$ sudo apt-get install curl
```

```
$ sudo apt-get install httpie
```

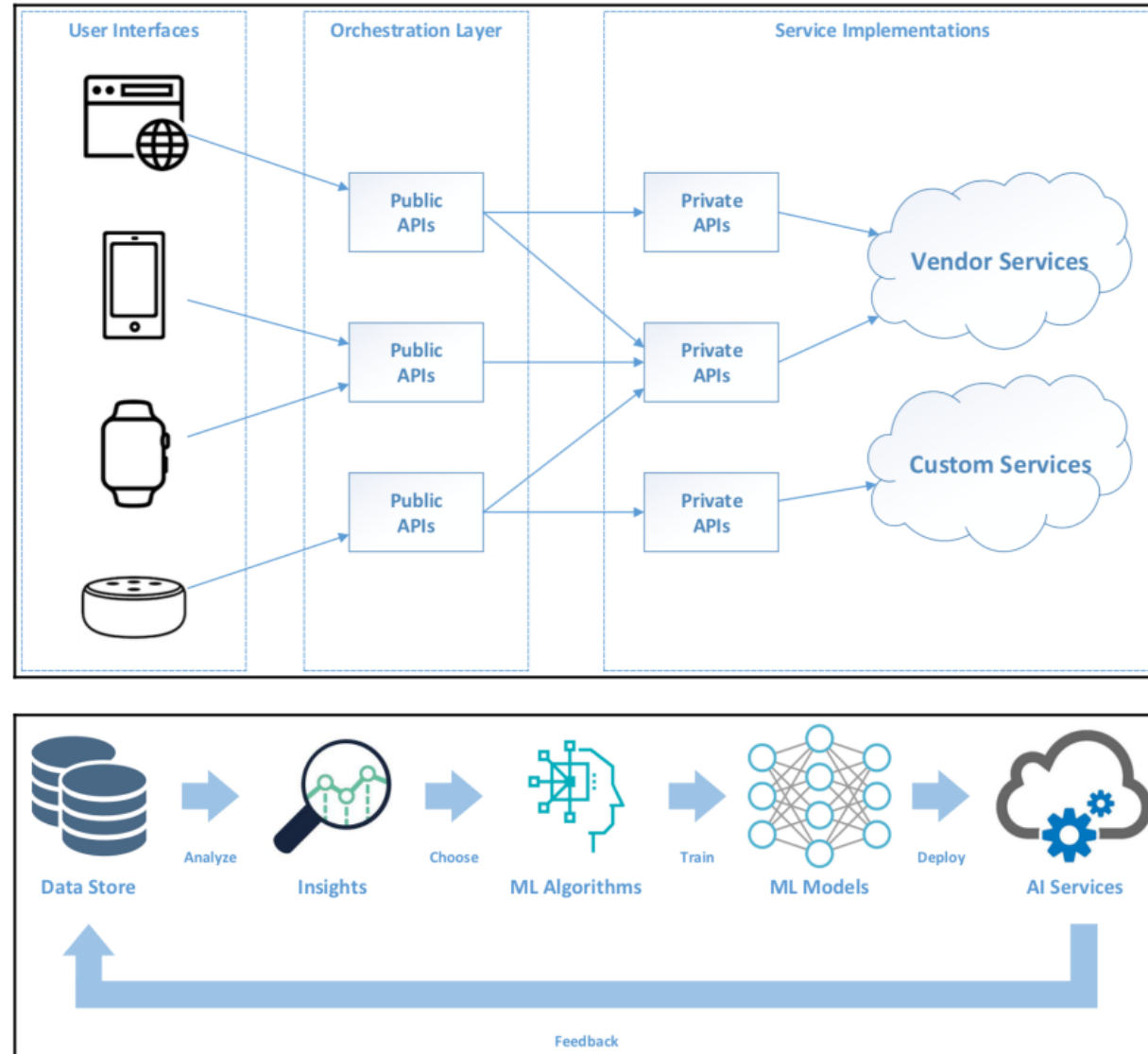
```
$ chalice new-project helloworld
```

```
$ chalice local
```

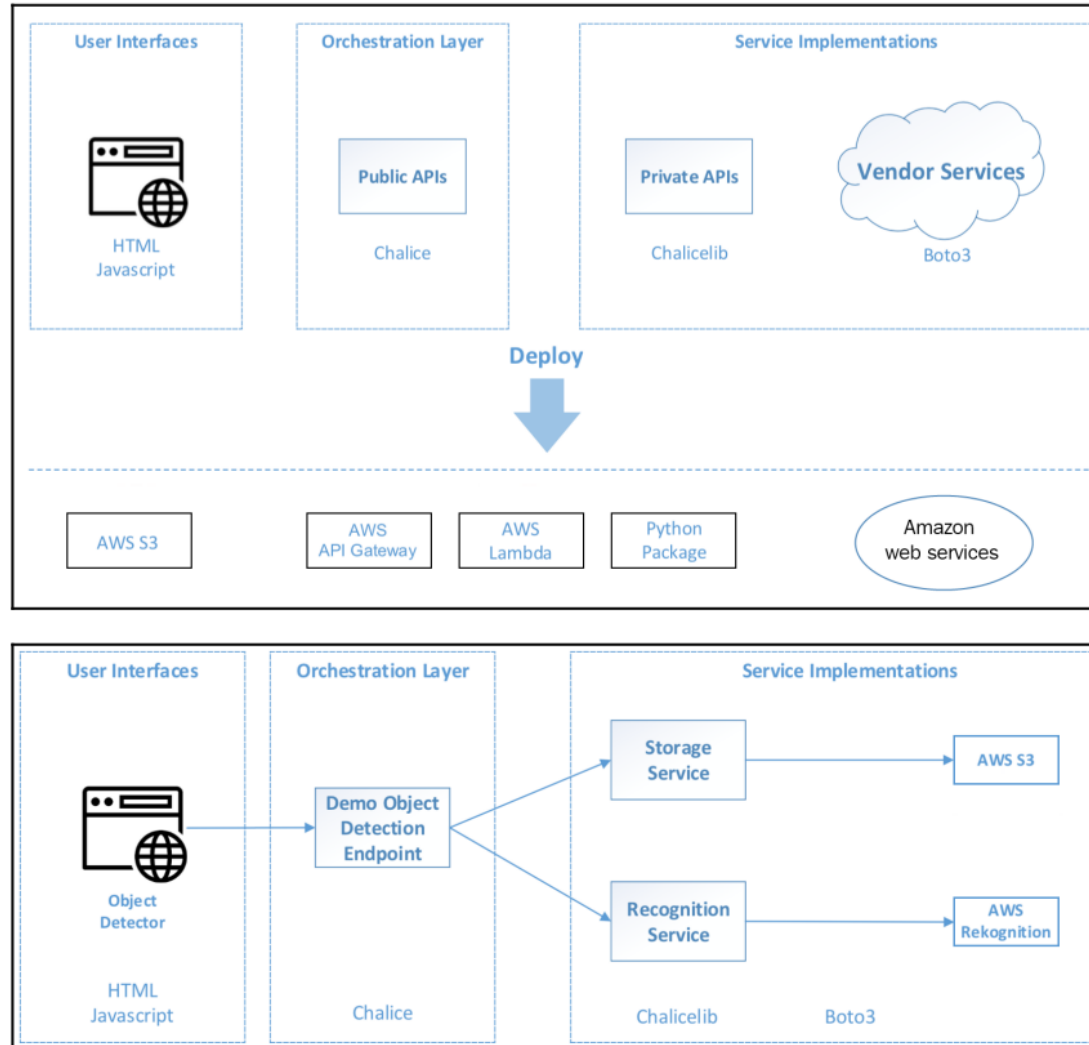
```
$ chalice deploy
```

**Local에서 AI app개발 환경을 제공**

# AWS AI 애플리케이션의 아키텍처 이해



# Chalice: AI 애플리케이션 아키텍처



# Project01: AWS Rekognition (Local)

[https://github.com/JSJeong-me/KOSA\\_AWS\\_AI/Project01](https://github.com/JSJeong-me/KOSA_AWS_AI/Project01)

```
$ cd Project01
```

```
$ chalice new-project Capabilities
```

```
$ cd Capabilities
```

```
$ mkdir chalicelib
```

```
$ chalice local
```

```
$ curl http://127.0.0.1:8000/demo-object-detection
```



# Project02: AWS Rekognition (Web UI)

**[https://github.com/JSJeong-me/KOSA\\_AWS\\_AI/Project02](https://github.com/JSJeong-me/KOSA_AWS_AI/Project02)**

\$ chalice local

./Website/index.html 실행

# Project03: AWS Rekognition (AWS Deploy)

**[https://github.com/JSJeong-me/KOSA\\_AWS\\_AI/Project03](https://github.com/JSJeong-me/KOSA_AWS_AI/Project03)**

```
$ cd Project03
```

```
$ cd Capabilities
```

```
$ chalice deploy
```

```
$ curl https://6pku28v9ok.execute-api.ap-northeast-1.amazonaws.com/api/demo-object-detection
```

# AWS Rekognition

Amazon Rekognition

Metrics

Demos

**Object and scene detection**

Image moderation

Facial analysis

Celebrity recognition

Face comparison

Text in image

Video Demos

Video analysis

Additional Resources

Getting started guide

Download SDKs

Developer resources


Pricing

FAQ

Forum

## Object and scene detection

Rekognition automatically labels objects, concepts and scenes in your images, and provides a confidence score.



Done with the demo?  
[Learn more](#)

▼ Results



|                |        |
|----------------|--------|
| Transportation | 98.8 % |
| Vehicle        | 98.8 % |
| Automobile     | 98.8 % |
| Car            | 98.8 % |
| Person         | 98.3 % |
| Human          | 98.3 % |

Show more

► Request

► Response

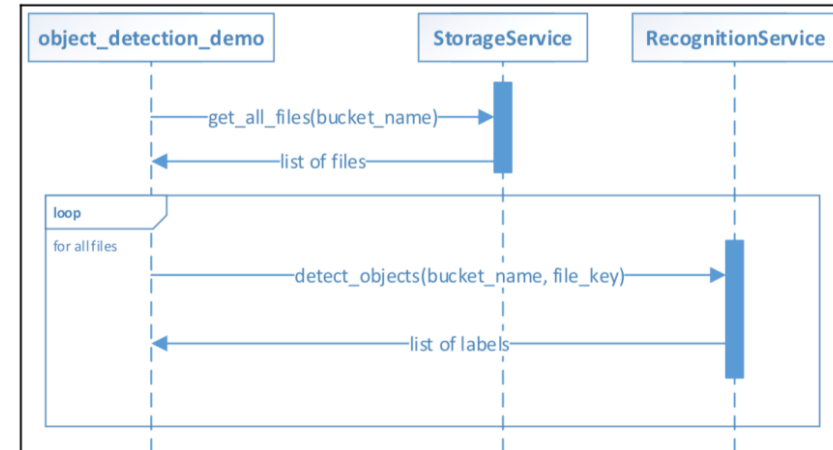
Choose a sample image



Use your own image  
Image must be .jpg or .png format and no larger than 5MB. Your image isn't stored.

[Upload](#) or drag and drop

Use image URL [Go](#)



<https://ap-northeast-1.console.aws.amazon.com/rekognition/home?region=ap-northeast-1#/label-detection>

# Amazon Rekognition Image, Video, and Custom labels examples

[참조]

[https://docs.aws.amazon.com/code-library/latest/ug/python\\_3\\_rekognition\\_code\\_examples.html](https://docs.aws.amazon.com/code-library/latest/ug/python_3_rekognition_code_examples.html)

[https://github.com/awsdocs/aws-doc-sdk-examples/tree/main/python/example\\_code/rekognition#code-examples](https://github.com/awsdocs/aws-doc-sdk-examples/tree/main/python/example_code/rekognition#code-examples)

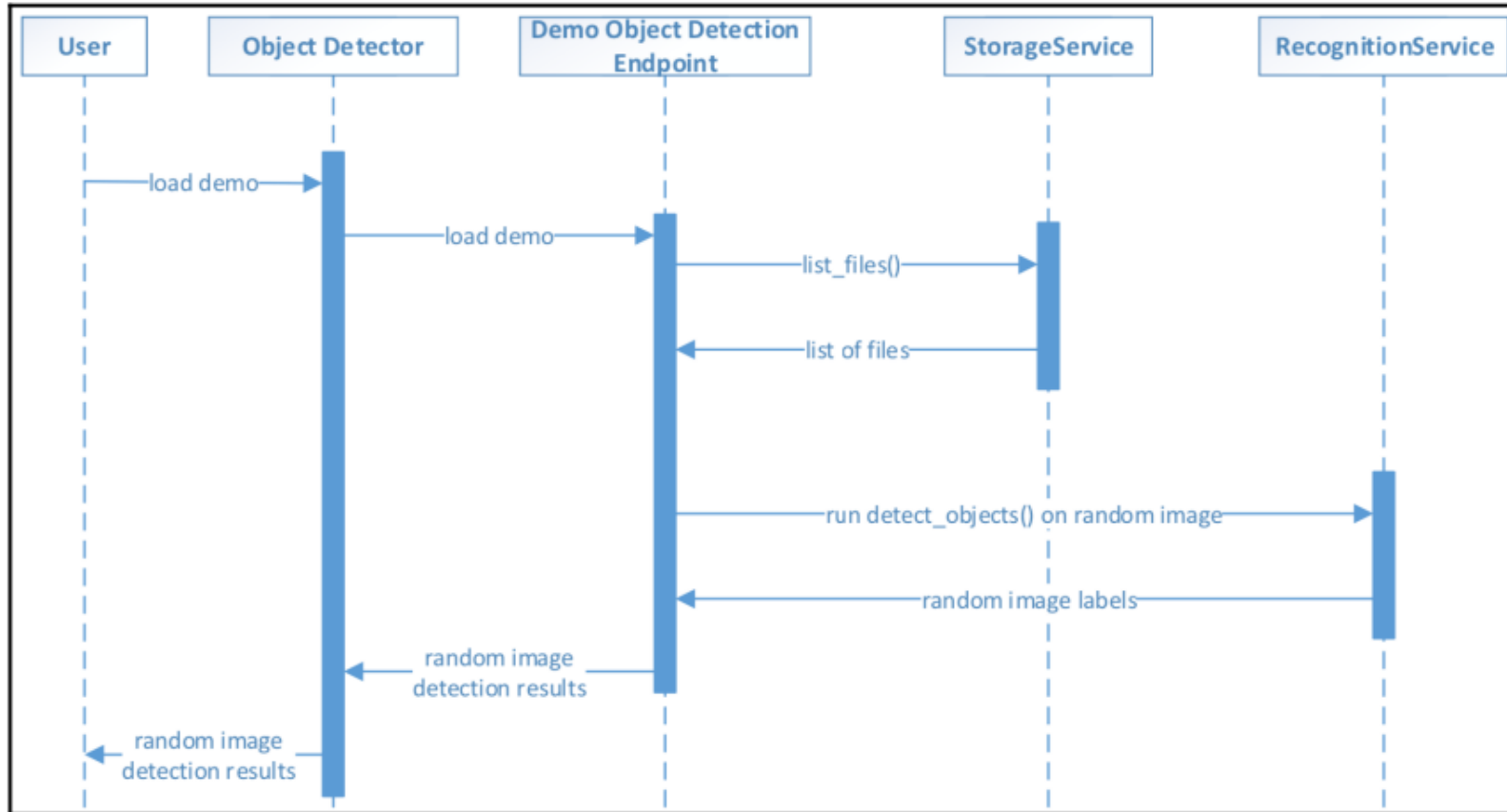
[예제]

```
$ pip install boto3
```

Boto3: Python 개발 환경 구축

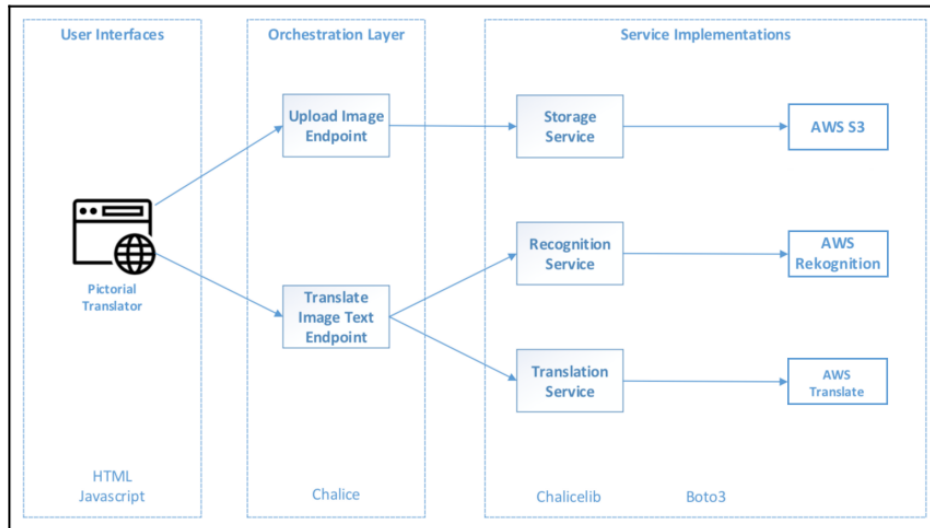
```
$ python rekognition_image_detection.py
```

# Rekognition Prj. Component간의 상호작용



# AWS Translate

Amazon Translate 사용을 시작하려면 AWS 계정을 설정하고 AWS Identity and Access Management(IAM) 사용자. AWS Command Line Interface(AWS CLI)를 사용하려면 다운로드하여 구성합니다.



## 주제

- 1단계: AWS 계정 설정 및 관리자 생성
- 2단계: 설정AWS Command Line Interface(AWS CLI)
- 3단계: 시작하기(콘솔)
- 4단계: 시작하기(AWS CLI)
- 5단계: 시작하기 (SDK)

[https://docs.aws.amazon.com/ko\\_kr/translate/latest/dg/getting-started.html](https://docs.aws.amazon.com/ko_kr/translate/latest/dg/getting-started.html)

# AWS Transcribe

Amazon Transcribe를 사용하려면AWS 계정을 만들고AWS Identity and Access Management(IAM) 사용자. AWS Command Line Interface(AWS CLI)를 사용하려면 다운로드하여 구성합니다.

## 주제

- 1단계: 를 설정합니다.AWS 계정관리자 사용자를 생성합니다.
- 2단계: 설정AWS Command Line Interface(AWS CLI)
- 3단계: 콘솔을 사용하여 시작하기
- 4단계: API 사용을 시작하기
- 5단계: 스트리밍 오디오 시작하기

[https://docs.aws.amazon.com/ko\\_kr/ko\\_kr/transcribe/latest/dg/getting-started.html](https://docs.aws.amazon.com/ko_kr/ko_kr/transcribe/latest/dg/getting-started.html)

# AWS Comprehend

Amazon Comprehend를 사용하려면 AWS 계정을 설정하고AWS Identity and Access Management(IAM) 사용자. Amazon Comprehend 컴프리엔드를 사용하려면 (AWS CLI) 를 다운로드하여 구성합니다.

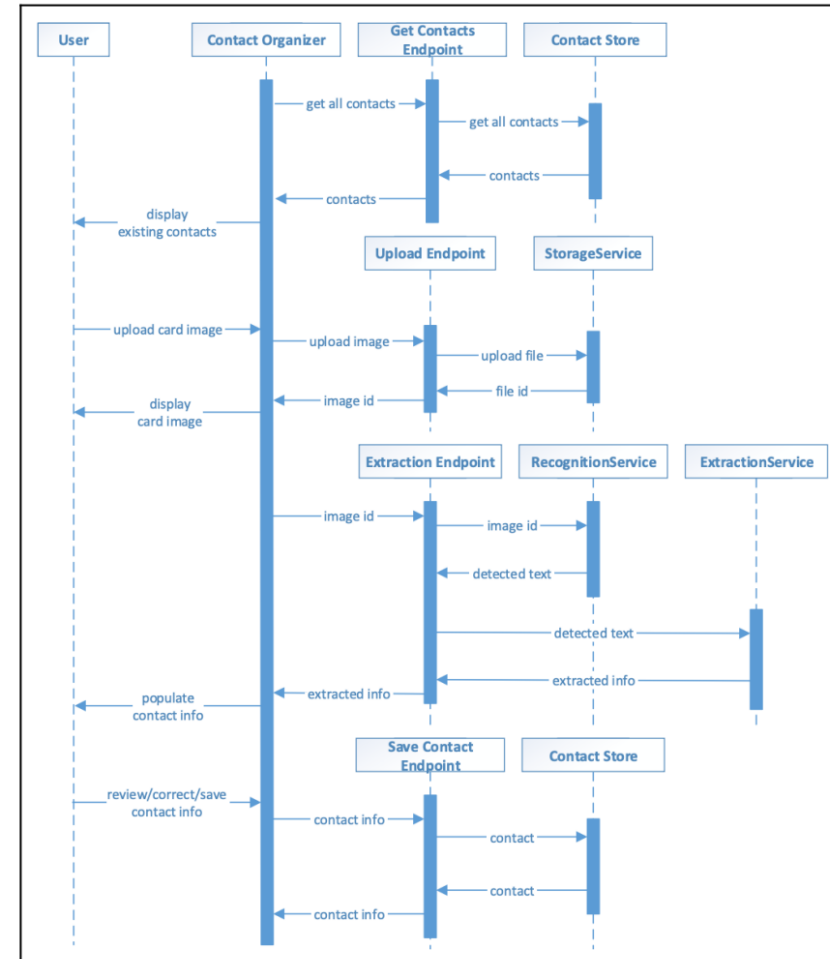
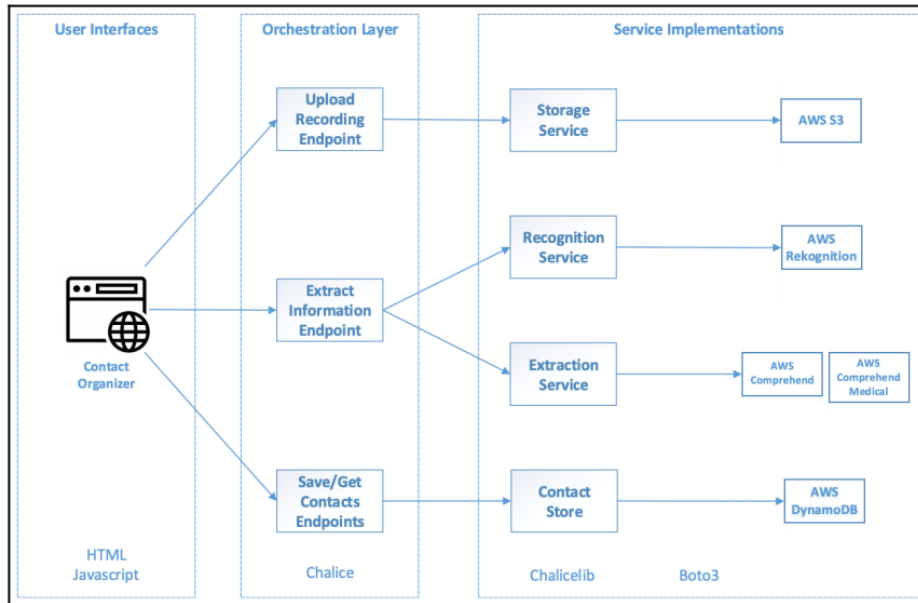
## 주제

- [1단계: AWS 계정 설정 및 관리자 생성](#)
- [2단계: 설정AWS Command Line Interface\(AWS CLI\)](#)
- [3단계: Amazon Comprehend 콘솔 사용 시작하기](#)
- [4단계: Amazon Comprehend API 사용 시작하기](#)
- [솔루션: Amazon Comprehend 및 Amazon Elasticsearch Service 텍스트 분석](#)

[https://docs.aws.amazon.com/ko\\_kr/ko\\_kr/comprehend/latest/dg/getting-started.html](https://docs.aws.amazon.com/ko_kr/ko_kr/comprehend/latest/dg/getting-started.html)



# AWS Comprehend(계속)



# AWS Polly

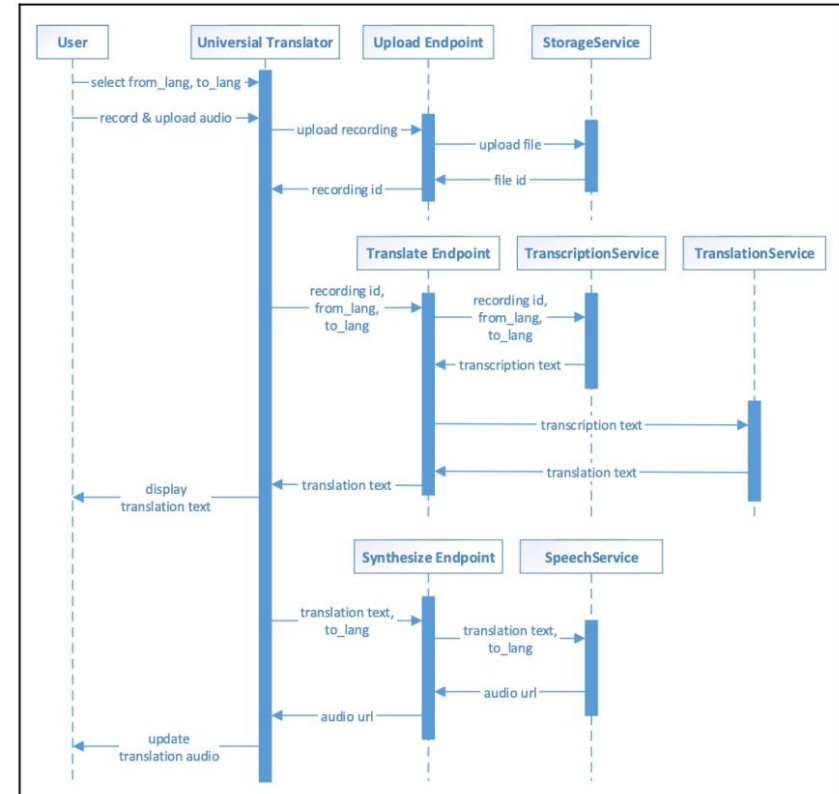
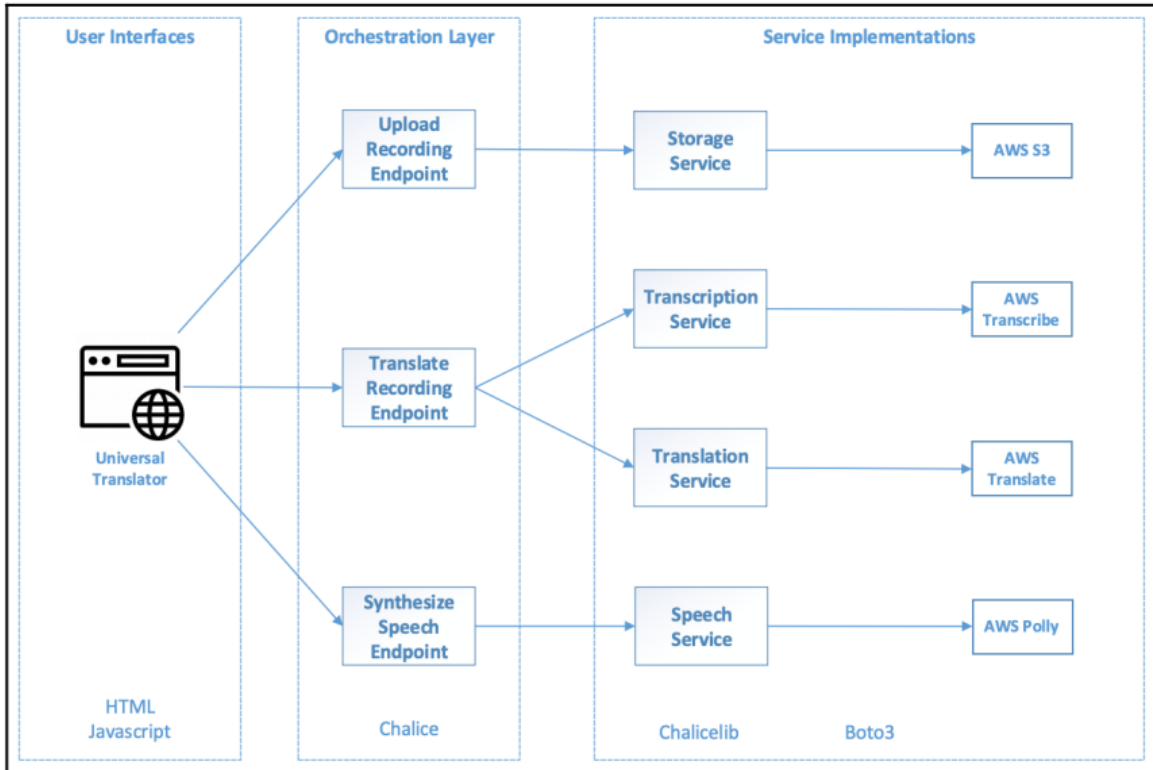
Amazon Polly는 기존 애플리케이션과 쉽게 통합할 수 있는 간단한 API 작업을 제공합니다. 지원되는 작업 목록은 [작업](#) 항목을 참조하십시오. 다음 두 가지 옵션 중 하나를 사용할 수 있습니다.

- AWSSDK — SDK를 사용할 때 Amazon Polly에 대한 요청은 사용자가 제공한 자격 증명을 사용하여 자동으로 서명되고 인증됩니다. 이것은 애플리케이션을 구축할 때 권장되는 선택 사항입니다.
- AWS CLI — 를 사용해도 됩니다. AWS CLI 코드를 작성하지 않고 Amazon Polly 기능에 액세스하려면 다음 섹션에서는 설정 및 입문용 실습을 제공하는 방법에 대해 설명합니다.

## 주제

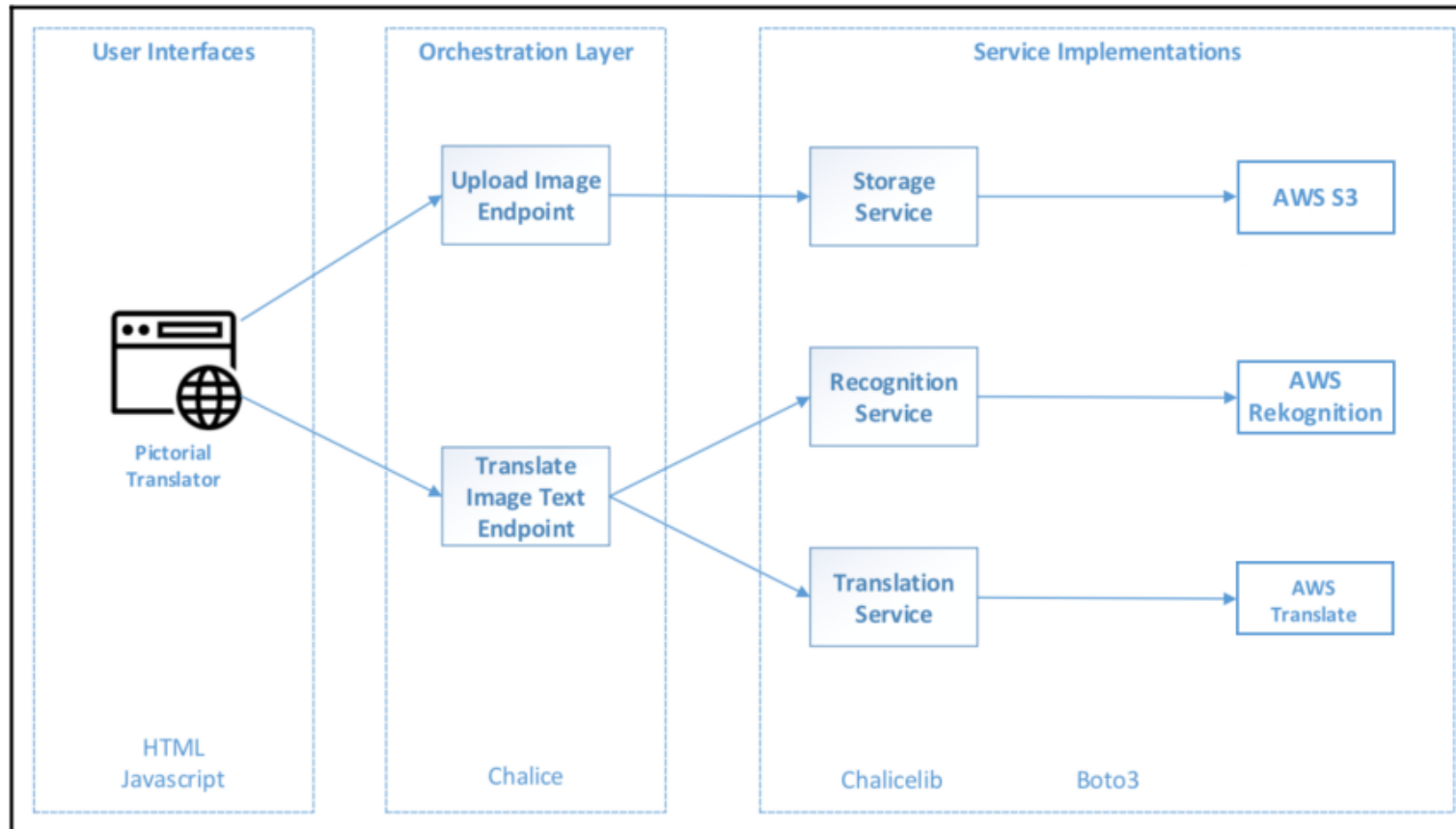
- [1단계: 설정AWS계정 및 사용자 생성](#)
- [2단계: 시작하기\(콘솔\)](#)
- [3단계: 시작하기\(AWS CLI\)](#)
- [Python 예제](#)

# AWS Polly (계속)



# Project IV: 사진 번역기(Rekognition & Translate)

App 사용자 인터페이스를 이용하여 외국어 텍스트가 포함된 사진을 upload하여 번역 결과를 볼 수 있다.



## Project Files 위치 : PictorialTranslator

[https://github.com/JSJeong-me/KOSA\\_AWS\\_AI/tree/main/translate/PictorialTranslator](https://github.com/JSJeong-me/KOSA_AWS_AI/tree/main/translate/PictorialTranslator)

### 프로젝트 진행 순서

1. Static html 작성 – image upload -> upload된 image display -> 번역된 텍스트 display
2. Rekognition Module 구현
3. Translate Module 구현
4. 단위 테스트: test.py
5. \$ chalice local
6. \$ chalice deploy

```
$ aws rekognition detect-text --image  
"S3Object={Bucket=bucketname,Name=input.jpg}"
```

```
$ aws translate translate-text --text "Einbahnstraße" --source-language-code auto --  
target-language-code en
```

### **CUSTOM Terminology**

**[https://docs.aws.amazon.com/ko\\_kr/ko\\_kr/translate/latest/dg/creating-custom-terminology.html](https://docs.aws.amazon.com/ko_kr/ko_kr/translate/latest/dg/creating-custom-terminology.html)**

## Upload test

```
$ (echo -n '{"filename": "stop.jpg", "filebytes": ""; base64 ./stop.jpg; echo ""}' ) | curl --header "Content-Type: application/json" -d @- http://127.0.0.1:8000/images
```

## Detect & Translate

```
$ curl --header 'Content-Type: application/json' --request POST --data  
'{"fromLang": "auto", "toLang": "fr"}' http://127.0.0.1:8000/images/german-one-way-street-sign.jpg/translate-text
```

## 이미지의 텍스트 위치

이미지에서 인식된 텍스트의 위치를 확인하려면 DetectText가 반환하는 경계 상자([Geometry](#)) 정보를 사용합니다. Geometry 객체에는 감지된 선과 단어에 대한 두 가지 유형의 경계 상자 정보가 있습니다.

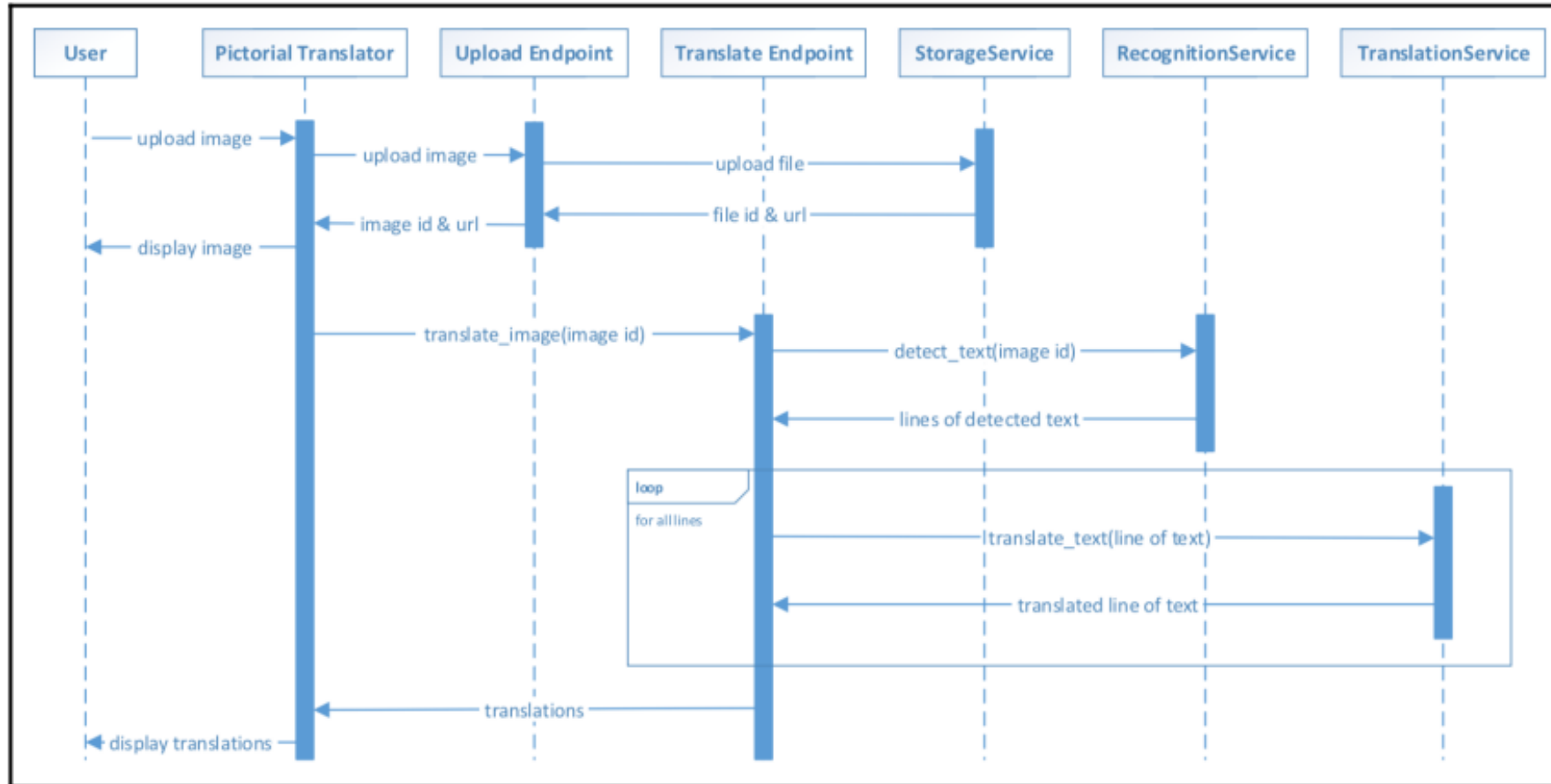
- [BoundingBox](#) 객체의 축으로 정렬된 거친 직사각형 윤곽 [Point](#) 배열에서 여러 개의 X와 Y 좌표로 구성된, 세분화된 다각형 테두리 상자와 다각형 좌표는 원본 이미지의 텍스트 위치를 나타냅니다. 좌표 값은 전체 이미지 크기의 비율입니다. 자세한 정보는 [BoundingBox](#)을 참조하십시오.

DetectText 작업의 다음 JSON 응답은 다음 이미지에서 감지된 단어와 줄을 표시합니다.

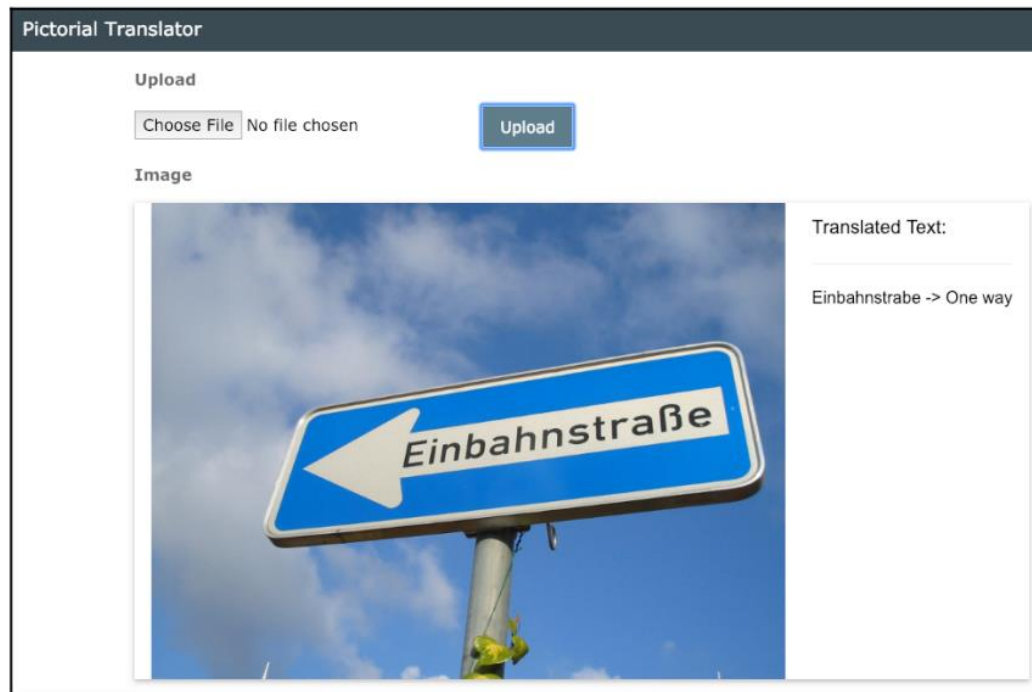
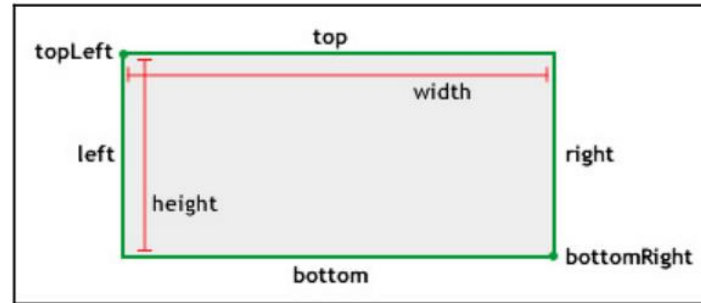




# 프로젝트 아키텍처



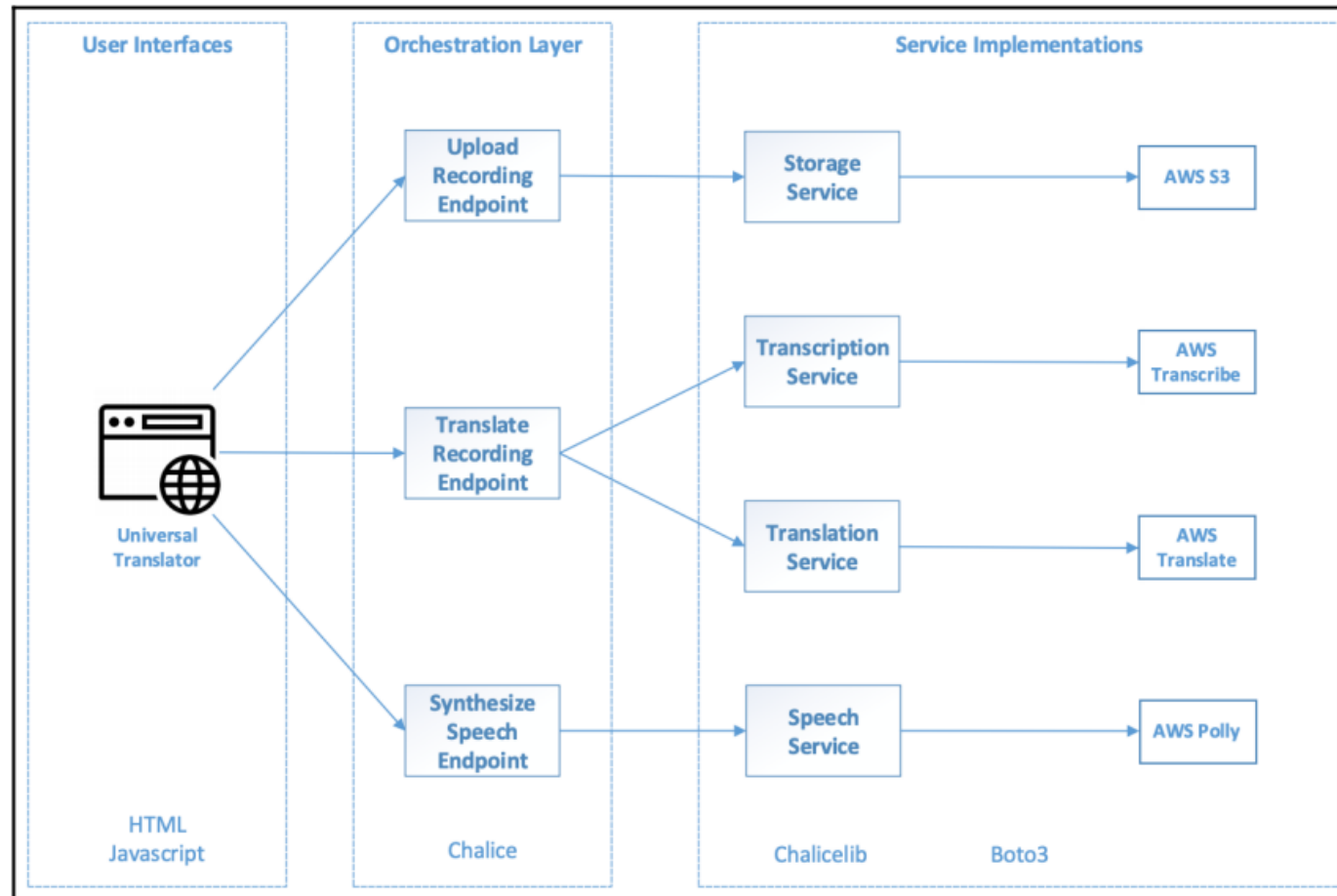
# 텍스트 Detect & Display



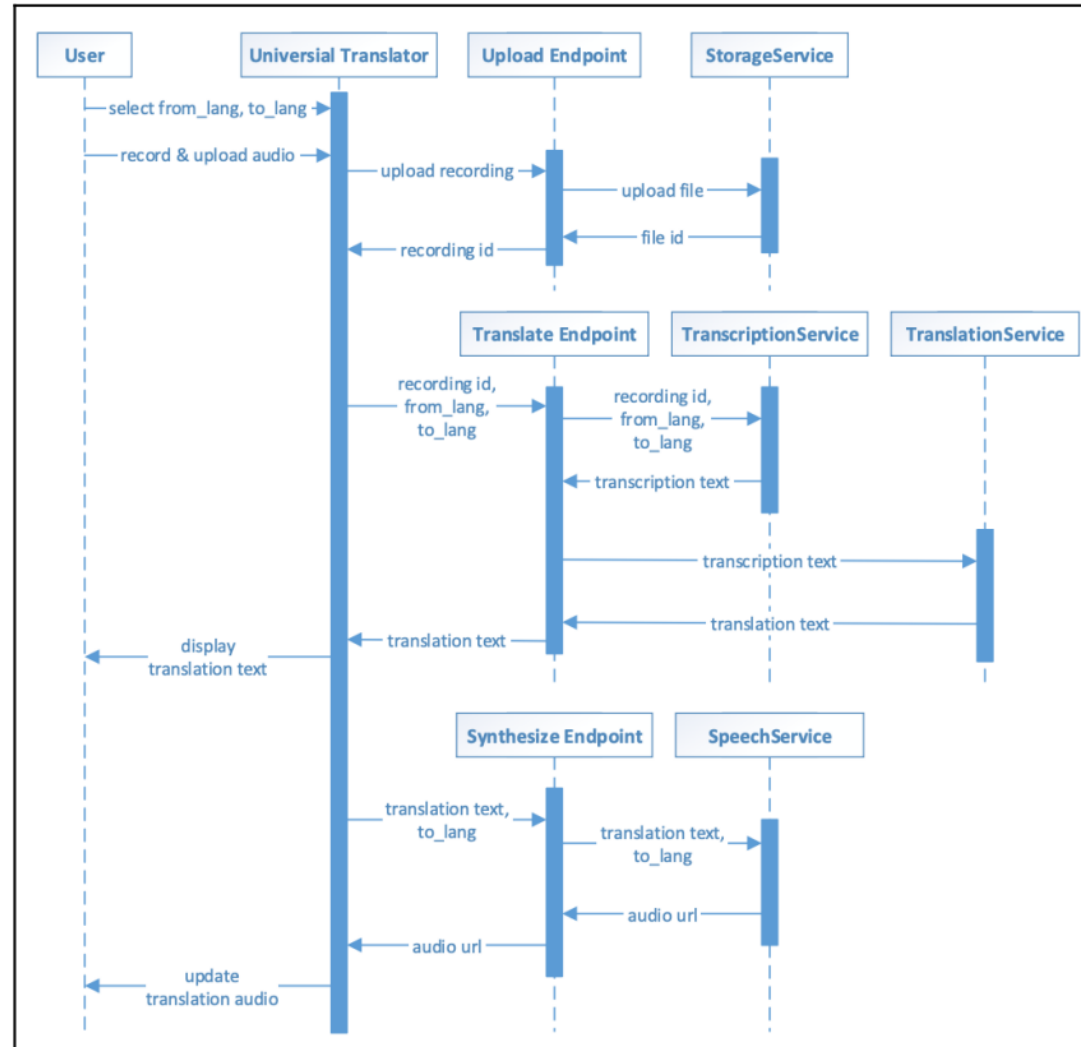
# Bucket 정책 수정

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyForAllowUploadWithACL",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::images-aws-ai/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control"
        }
      }
    }
  ]
}
```

# Project V: Speech-to-Text and Vice Versa with Amazon Transcribe and Polly



# 프로젝트 아키텍처



# Automatic Speech Recognition

```
$ aws transcribe start-transcription-job  
  --transcription-job-name <jobname>  
  --language-code en-US  
  --media-format wav  
  --media MediaFileUri=https://s3.amazonaws.com/contents.aws.a/<audio file>.wav  
  --output-bucket-name contents.aws.ai
```

<예시>

```
$ aws transcribe start-transcription-job --transcription-job-name me-asr --Language-code  
ko-KR --media-format wav --media MediaFileUri=s3://kosa-asr/20220312210201.wav --  
output-bucket-name kosa-asr
```

## 작업 Status 확인

```
$ aws transcribe get-transcription-job --transcription-job-name <jobname>
```

## Polly(텍스트 -> 음성)

```
$ aws polly start-speech-synthesis-task --output-format mp3 --output-s3-bucket-name  
<bucket> --text "testing testing 1 2 3" --voice-id Ivy
```

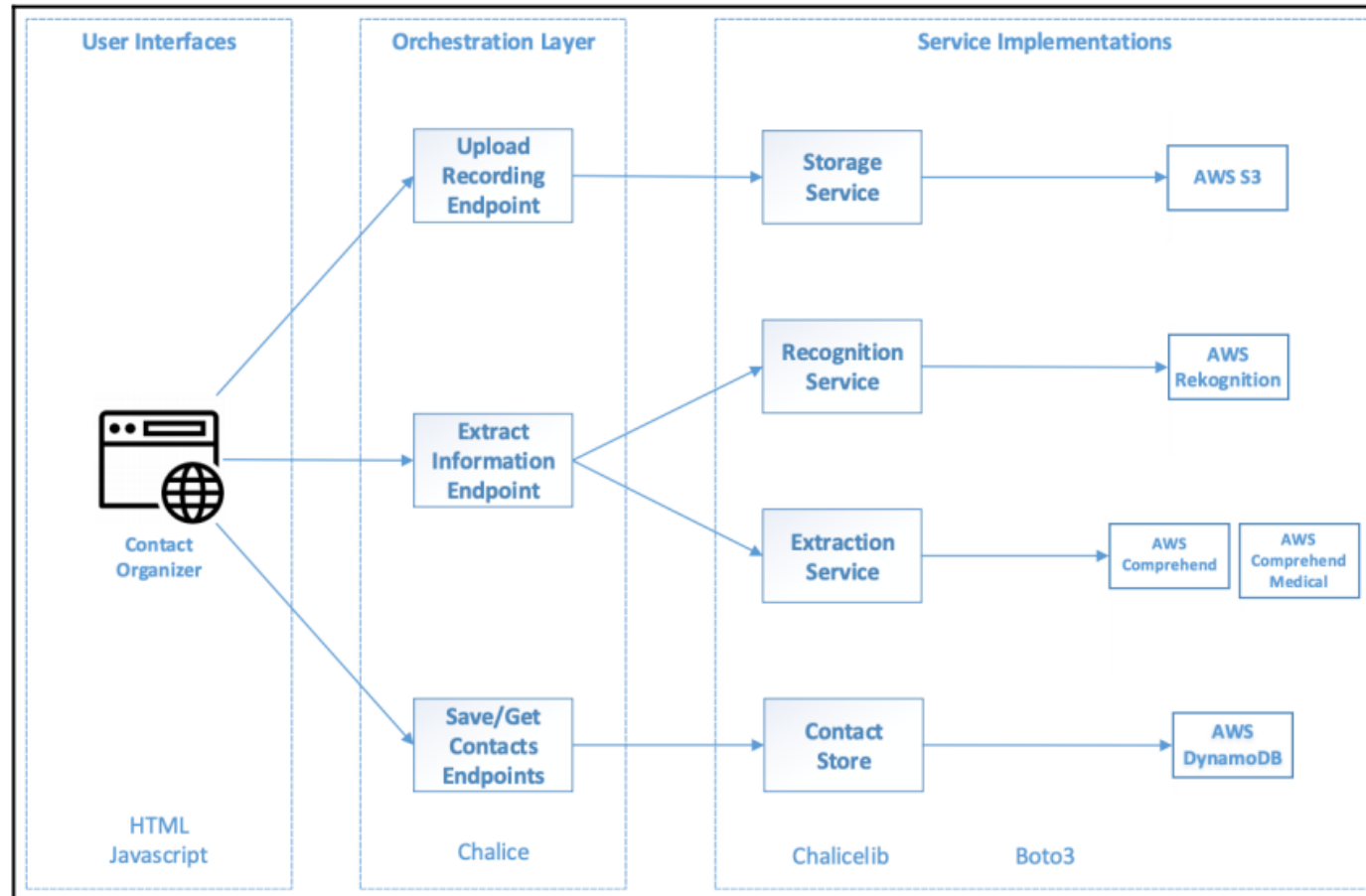
```
$ curl --header "Content-Type: application/json" --request POST --data '{"text":"Good  
afternoon Today is Friday","language": "en"}' http://127.0.0.1:8000/synthesize\_speech
```

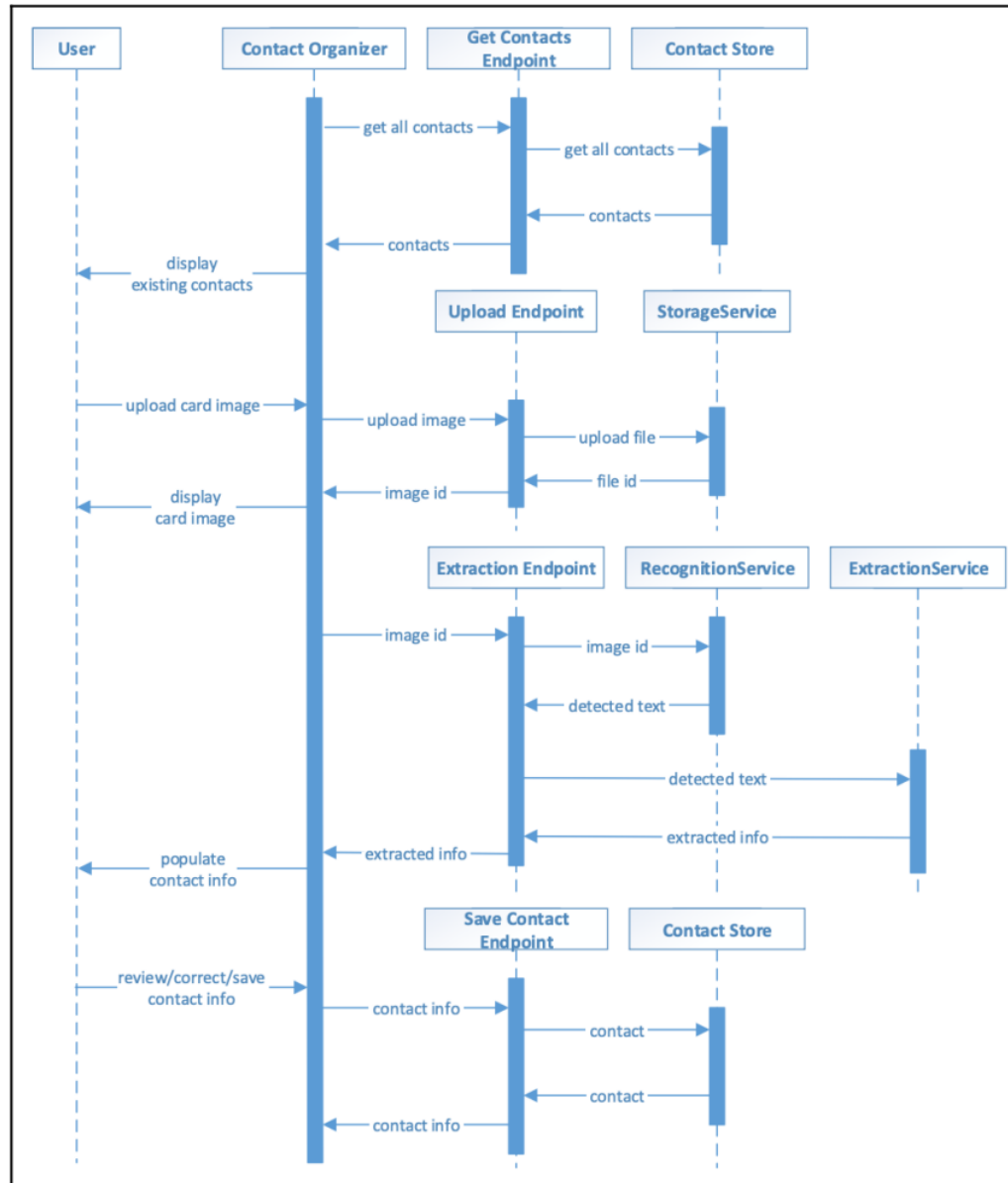
```
$ curl --header "Content-Type: application/json" --request POST --data '{"fromLang":"en",  
"toLang":"en"}' http://127.0.0.1:8000/recordings/sound-en.wav/translate-text
```



```
curl --header "Content-Type: application/json" --request POST --data '{"fromLang":"en", "toLang":"en"}'  
http://127.0.0.1:8000/recordings/sound-en.wav/translate-text
```

# Project VI: Extracting Information from Text with Amazon Comprehend





# DynamoDB Table 생성

DynamoDB

Dashboard

Tables

Backups

Reserved capacity

Preferences

DAX

Dashboard

## Create table

Amazon DynamoDB is a fully managed non-relational database service that provides fast and predictable performance with seamless scalability.

Create table

### Recent alerts

No CloudWatch alarms have been triggered. [View all in CloudWatch](#)

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

**Table name\***  ⓘ

**Primary key\*** Partition key

ⓘ

☐ Add sort key

### Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

- No secondary indexes.
- Auto Scaling capacity set to 70% target utilization, at minimum capacity of 5 reads and 5 writes.
- Encryption at Rest with DEFAULT encryption type.

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

[Cancel](#) [Create](#)

Contact Organizer

Add Contact

Choose File

No file chosen

Extract Info

Information

Name

Title

Email

Phone

Address

Organization

Address

City

State

Zip

Save Contact

My Contacts

Refresh

John Smith

Senior Software Engineer

AI Enterprise Inc.

123 Main Street

Washington D.C., undefined 20001

phone: (202) 123-4567

email: john.smith@aienterprise.com

61

# Comprehend Test

```
$ aws comprehend detect-entities --language-code en --text "AI Enterprise Inc. John Smith Senior Software Engineer 123 Main Street Washington D.C. 20001 john.smith@aienterprise.com (202) 123-4567"
```

```
$ aws comprehendmedical detect-phi --text "AI Enterprise Inc. John Smith Software Engineer 123 Main Street Washington D.C. 20001 john.smith@aienterprise.com (202) 123 4567"
```

정 준 수 / Ph.D ( jsjeong@hansung.ac.kr )

- 前) 삼성전자 연구원
- 前) 삼성의료원 (삼성생명과학연구소)
- 前) 삼성SDS (정보기술연구소)
- 現) 한성대학교 교수(겸)
- 전문분야: Computer Vision, 머신러닝(ML), RPA
- <https://github.com/JSJeong-me/>

