Manchester Metropolitan University

CLOUD COMPUTING

/AI as a SERVICE (AIaaS)
/AWS COMPREHEND

Dr Mohammed Kaleem

m.kaleem@mmu.ac.uk

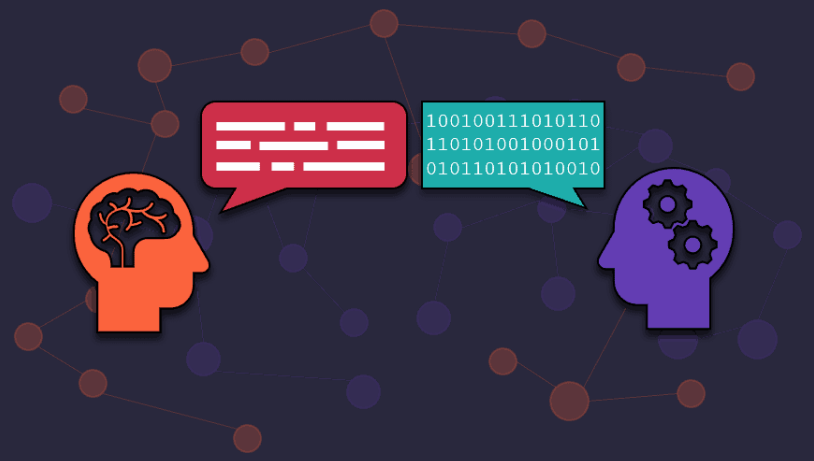
CONTENTS

AWS COMPREHEND – What is it?

Amazon Comprehend is **a natural language processing (NLP) service that uses machine learning (ML) to discover insights from text**.

Amazon Comprehend provides **custom entity recognition, custom classification, key phrase extraction, sentiment analysis, entity recognition**, and more APIs so you can easily integrate NLP into your applications.

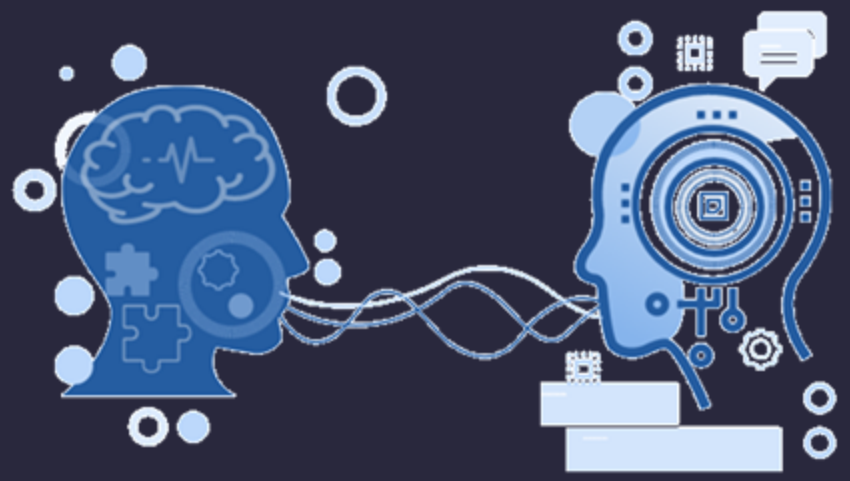
You simply call the Amazon Comprehend APIs in your application and provide the location of the source document or text. The APIs will **output entities, key phrases, sentiment, and language in a JSON format**, which you can use in your application.



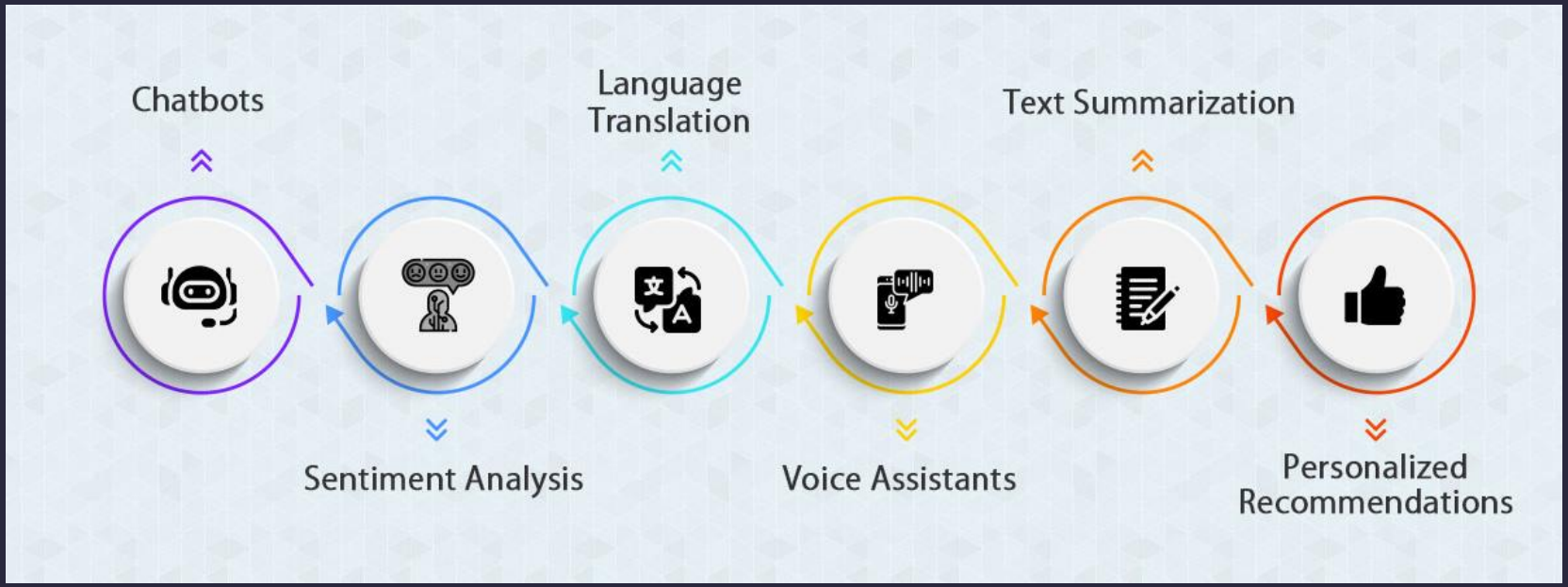
https://aws.amazon.com/rekognition/faqs/?nc=sn&loc=7

NATURAL LANGUAGE PROCESSING (NLP)

In a nutshell, Natural Language Processing (NLP) is a **field of artificial intelligence** that enables **computers to understand, interpret, and generate human language**, allowing for tasks like machine **translation, sentiment analysis, and chatbot development**.



APPLICATIONS OF NLP



AWS COMPREHEND – How does it work?

1. **Input** → Provide text data (documents, reviews, emails, etc.)
2. **Processing** → AWS Comprehend applies NLP models
3. **Output** → It returns structured data (**JSON**) with insights

AWS COMPREHEND – Key features

Feature	What it Does?
Language Detection	Detects the language of the text
Sentiment Analysis	Determines if a text is positive, negative, neutral, or mixed
Entity Recognition	Identifies names, places, brands, dates, and more
Key Phrase Extraction	Extracts important words and phrases
PII Detection	Detects personal data (e.g., phone numbers, emails)
Topic Modeling	Groups documents by topics
Custom Classification	Classifies text into predefined categories

AWS COMPREHEND – USE CASES

Customer Support & Sentiment Analysis

Use Case:
Analyse customer feedback (emails, chat messages, support tickets) to detect customer sentiment (positive, negative, neutral).

Example:
A company analyses customer complaints and prioritizes negative ones for urgent responses.

Chatbots use sentiment analysis to escalate angry customers to human agents.

AWS Comprehend Feature Used:
Sentiment Analysis

AWS COMPREHEND – USE CASES

Social Media Monitoring & Brand Reputation

Use Case:
Track brand mentions on social media (Twitter, Facebook, Reddit) and analyse public sentiment.

Example:
A company monitors Twitter for brand mentions and detects trending complaints or praises.

Sentiment analysis helps gauge public reaction to product launches.

AWS Comprehend Feature Used:
Sentiment Analysis, Entity Recognition

AWS COMPREHEND – USE CASES

Healthcare & Medical Document Analysis

Use Case:
Extract important information from **medical records**, **prescriptions**, and **doctor notes**.

Example:
A hospital digitizes patient records and extracts details like medications, symptoms, and diagnoses.

AWS Comprehend Feature Used:
Entity Recognition, Key Phrase Extraction

AWS COMPREHEND – USE CASES

Legal Document Processing & Analysis

Use Case:
Automate contract review and extract key legal terms, parties, and dates.

Example:
A law firm uses AWS Comprehend to scan contracts and agreements and highlight important clauses and deadlines.

AWS Comprehend Feature Used:
Entity Recognition, Key Phrase Extraction

AWS COMPREHEND – USE CASES

HR & Recruitment – Resume Screening

Use Case:
Analyse resumes and categorize candidates based on skills, experience, and job roles.

Example:
A company scans thousands of resumes and automatically identifies top candidates based on key skills.

AWS Comprehend Feature Used:
Entity Recognition, Custom Classification

AWS COMPREHEND – USE CASES

Data Privacy & Compliance (GDPR, HIPAA…)

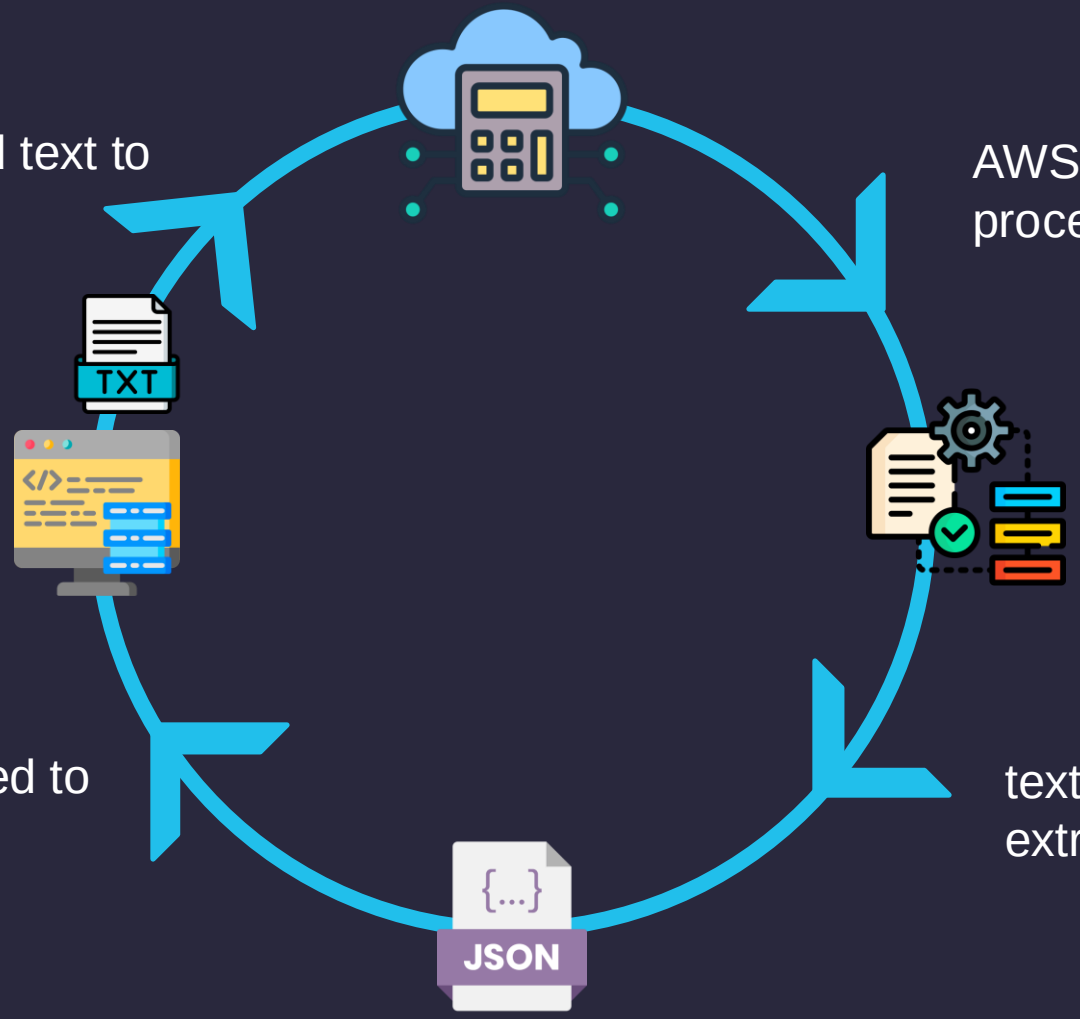
Use Case:
Automatically detect and mask sensitive user data in emails, chat logs, and customer records to comply with privacy laws.

Example:
A healthcare company ensures HIPAA compliance by redacting patient names, SSNs, and medical details from chat logs and reports.

A financial institution removes credit card details from customer support transcripts to meet PCI-DSS regulations.

AWS Comprehend Feature Used:
Personally Identifiable Information (PII) detection

AWS COMPREHEND – HIGH LEVEL OVERVIEW



Application: upload text to AWS Comprehend

AWS Comprehend processes text

text features are extracted

Text data is returned to client application

AWS REKOGNITION API CALL RESULT

When you make a call to the Comprehend API, AWS Comprehend analyses the text and **returns a JSON formatted object** containing key insights extracted from the input text.

The exact fields in the response depend on the API operation used.


```
{
  "Entities": [
    {
      "Text": "Khabib Nurmagomedov",
      "Type": "PERSON",
      "Score": 0.99,
      "BeginOffset": 0,
      "EndOffset": 19
    },
    {
      "Text": ”UFC",
      "Type": "ORGANIZATION",
      "Score": 0.98,
      "BeginOffset": 14,
      "EndOffset": 20
    }
  ]
}
```


Entity Recognition Result

AWS COMPREHEND – Sentiment Analysis Example

📄 **Input:** *"I love the Cloud Computing Module! It's amazing, Kaleem is the best!!"*

📊 **AWS Comprehend Output:**

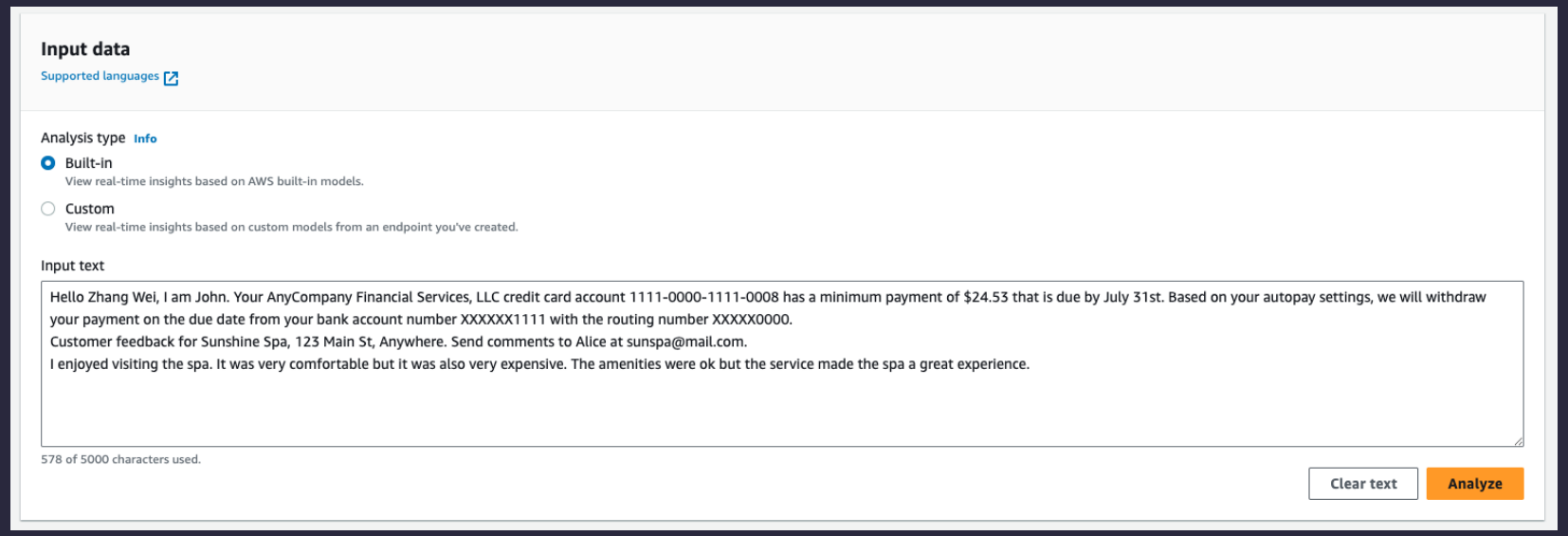

```
{
    "Sentiment": "POSITIVE",
    "SentimentScore": {
        "Positive": 0.98,
        "Negative": 0.01,
        "Neutral": 0.00,
        "Mixed": 0.01
    }
}
```

AWS COMPREHEND API REPONSE KEY ELEMENTS

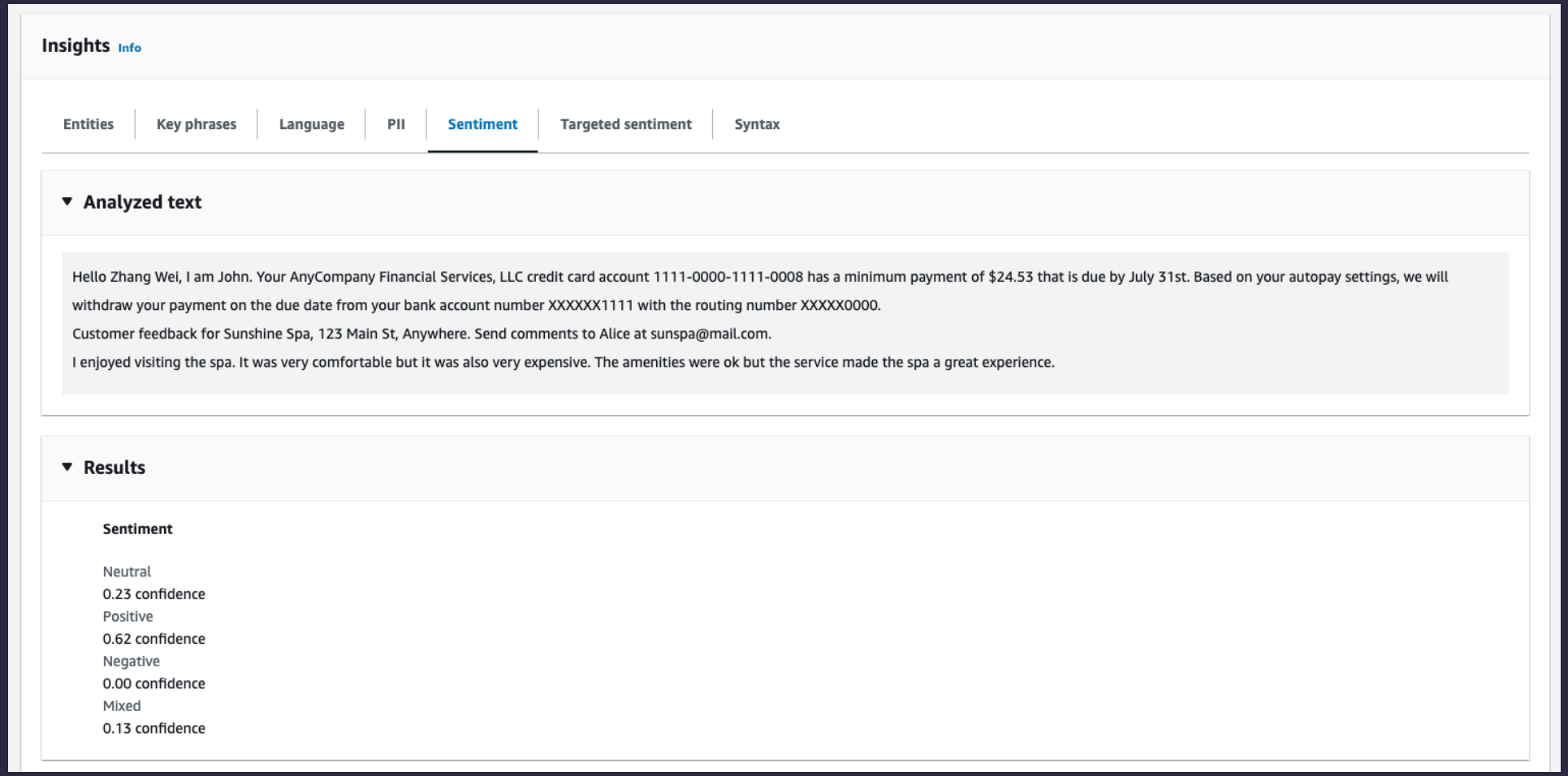
API Call	Key Features in Result	Description
detect-dominant-language	LanguageCode, Score	Detects the language of the input text.
detect-sentiment	Sentiment, SentimentScore (Positive, Negative, Neutral, Mixed)	Analyzes emotional tone of the text.
detect-entities	Entities → Text, Type, Score, BeginOffset, EndOffset	Identifies names, places, dates, and organizations.
detect-key-phrases	KeyPhrases → Text, Score, BeginOffset, EndOffset	Extracts important words or phrases.
detect-pii-entities	Entities → Text, Type (EMAIL, PHONE, SSN), Score	Detects personally identifiable information (PII).
start-topics-detection-job	Topics → TopicIndex, Terms	Groups documents into topics with related terms.
classify-document	Classes → Name, Score	Assigns predefined categories to text.

USING AWS COMPREHEND

You can use all Comprehend features through the browser.



USING AWS COMPREHEND



USING AWS COMPREHEND IN JAVA APPLICATIONS

Some examples of using AWS Comprehend in a simple Java Application

ACCESS KEYS

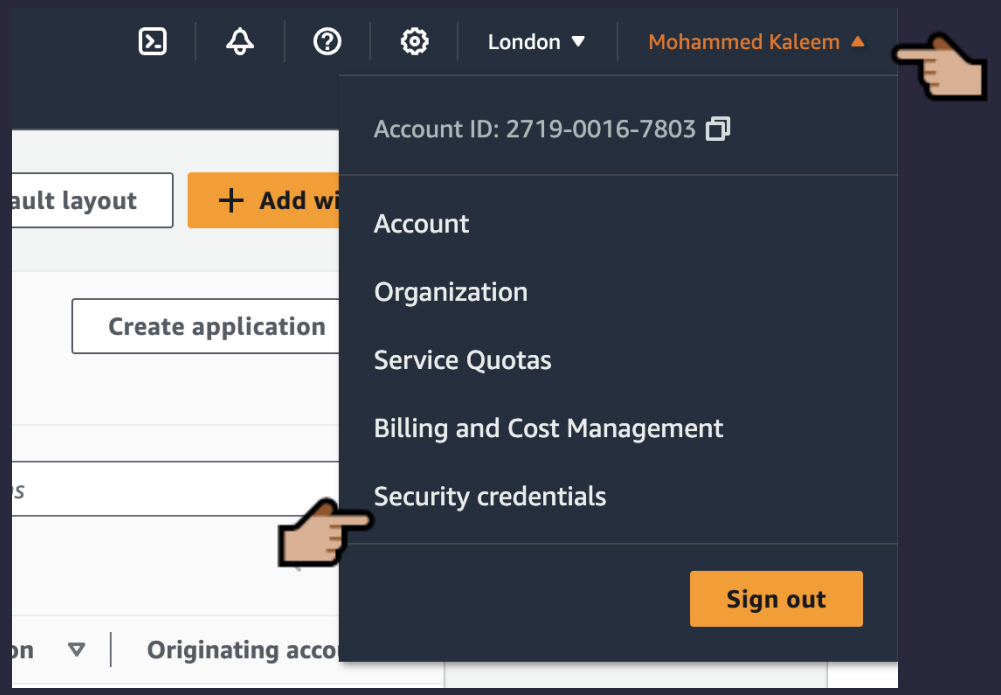
Before we start coding, we need to create access keys.

Access keys are alpha numeric strings that are unique to your account and allow you to control certain AWS services through code.

You are allowed a maximum of 2 access keys.

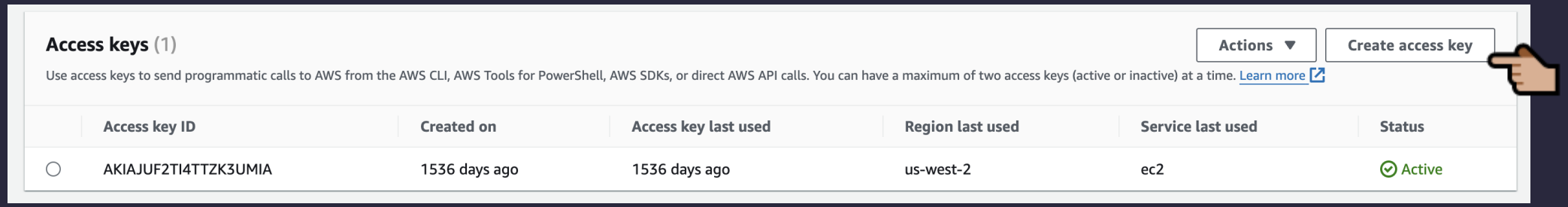
GENERATING ACCESS KEYS

Once you've logged in, click on your name (top right of screen) and select "Security credentials".

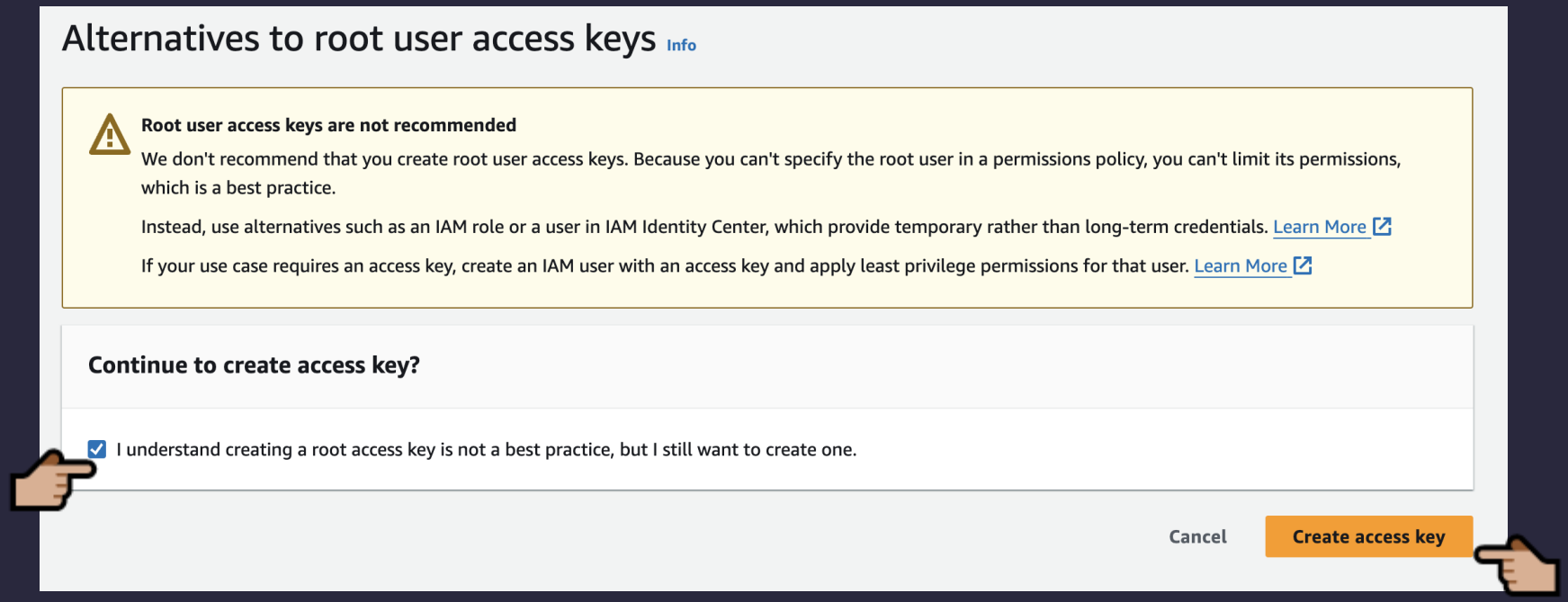


GENERATING ACCESS KEYS – continued

Scroll down till you see the access key section and select “Create access key”.

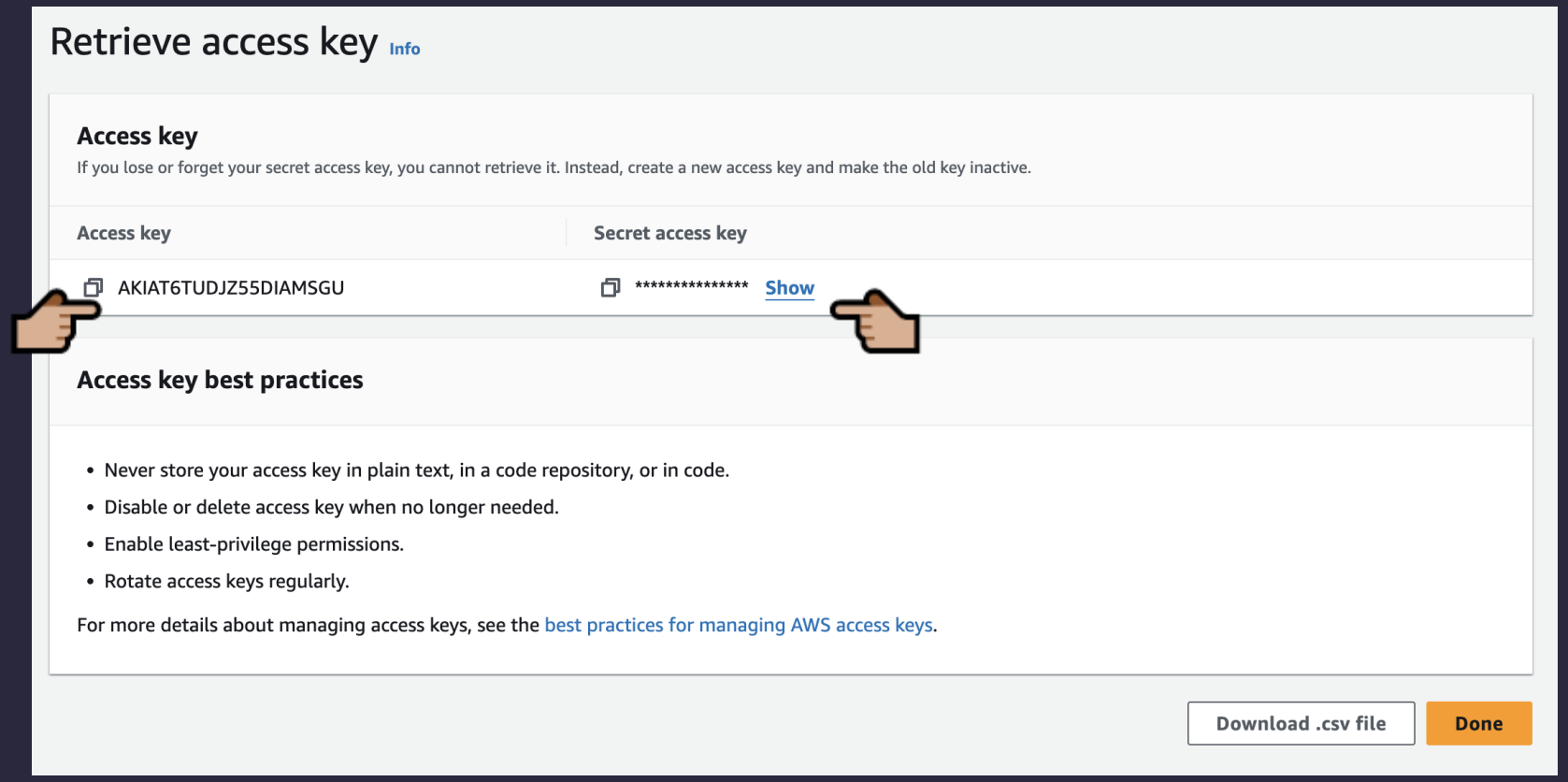


GENERATING ACCESS KEYS – continued



GENERATING ACCESS KEYS – continued

Make a note of both keys (or just download them in a csv file).



REQUIRED MAVEN DEPENDENCIES

Create a new maven project (*skip archetype selection*) and add the following dependency to the pom.xml file.

```
<dependency>
  <groupId>com.amazonaws</groupId>
  <artifactId>aws-java-sdk-rekognition</artifactId>
  <version>1.12.114</version>
</dependency>
```

Alternatively use the starter project on eclipse.

AWS Comprehend Client Object

Start by instantiating a global Amazon Comprehend client object (exactly the same as using AWS Rekognition).
We will use this object in all our Comprehend methods. It is therefore, recommended that you implemented all this in a “utils” class (as shown below).

```
public class AWSComprehendUtils {

    private static final String AWS_ACCESS_KEY_ID = "AK                    5FM7";
    private static final String AWS_SECRET_KEY = "qHY5C...                    phA3B1C0OXcnF";

    ComprehendClient client;

    public AWSComprehendUtils() {

        AwsBasicCredentials awsCreds = AwsBasicCredentials.create(AWS_ACCESS_KEY_ID, AWS_SECRET_KEY);

        client = ComprehendClient.builder()
                .region(Region.US_EAST_1)
                .credentialsProvider(StaticCredentialsProvider.create(awsCreds))
                .build();

    }
```

Your keys here

AWS Comprehend – Detect Language

```java
public void detectLanguage(String text) {

    DetectDominantLanguageRequest request = DetectDominantLanguageRequest.builder()
            .text(text)
            .build();

    DetectDominantLanguageResponse response = this.client.detectDominantLanguage(request);
    List<DominantLanguage> languages = response.languages();
    if(languages.isEmpty()) {
        System.out.println("No languages detected.");
        return;
    }

    for(DominantLanguage lang: languages) {
        System.out.println("Detected language: " + lang.languageCode() + " with confidence: " + lang.score());
    }
}
```

```java
public class AWSComprehendController {

    public static void main(String[] args) {

        AWSComprehendUtils utils = new AWSComprehendUtils();
        String langText = "السلام علیکم، آپ کیسے ہیں؟";
        utils.detectLanguage(langText);

    }
}
```

Detected language: ur with confidence: 1.0

AWS Comprehend – Detect Sentiment

```java
public void sentimentAnalysis(String text) {

    DetectSentimentRequest request = DetectSentimentRequest.builder()
            .text(text)
            .languageCode("en")
            .build();

    DetectSentimentResponse response = this.client.detectSentiment(request);
    System.out.println("Sentiment: " + response.sentiment());
}
```

```java
public class AWSComprehendController {

    public static void main(String[] args) {

        AWSComprehendUtils utils = new AWSComprehendUtils();

        String textSentiment = "Cloud Computing is the best module at MMU and Kaleem is the best ever!!";
        utils.sentimentAnalysis(textSentiment);

    }
}
```

Sentiment: POSITIVE

AWS Comprehend – Detect Entities

```java
public void entityRecognition(String text) {

    DetectEntitiesRequest request = DetectEntitiesRequest.builder()
            .text(text)
            .languageCode("en") // Language of the text
            .build();

    DetectEntitiesResponse response = this.client.detectEntities(request);
    response.entities().forEach(entity ->
        System.out.println("Entity: " + entity.text() + " (Type: " + entity.type() + ")")
    );
}
```

```java
public class AWSComprehendController {

    public static void main(String[] args) {

        AWSComprehendUtils utils = new AWSComprehendUtils();
        String entityText = "Khabib Abdulmanapovich Nurmagomedov is a professional mixed martial artist. "
                + "He was the longest-reigning UFC Lightweight Champion ever, "
                + "having held the title from April 2018 to March 2021.  ";
        utils.entityRecognition(entityText);
    }
}
```

Entity: Khabib Abdulmanapovich Nurmagomedov (Type: PERSON)
Entity: UFC (Type: ORGANIZATION)
Entity: Lightweight Champion (Type: TITLE)
Entity: April 2018 (Type: DATE)
Entity: March 2021 (Type: DATE)

AWS Comprehend – Detect Key Phrases

```java
public void keyPhraseExtraction(String text) {

    DetectKeyPhrasesRequest request = DetectKeyPhrasesRequest.builder()
            .text(text)
            .languageCode("en") // Language of the text
            .build();

    // Get key phrases from AWS Comprehend
    DetectKeyPhrasesResponse response = this.client.detectKeyPhrases(request);
    List<KeyPhrase> keyPhrases = response.keyPhrases();

    // Print extracted key phrases
    System.out.println("Key Phrases Detected:");
    for (KeyPhrase phrase : keyPhrases) {
        System.out.println("- " + phrase.text() + " (Confidence: " + phrase.score() + ")");
    }

}
```

```java
public class AWSComprehendController {

    public static void main(String[] args) {

        AWSComprehendUtils utils = new AWSComprehendUtils();
        String keyPhrase = "Khabib Abdulmanapovich Nurmagomedov is a professional mixed martial artist. "
                + "He was the longest-reigning UFC Lightweight Champion ever, "
                + "having held the title from April 2018 to March 2021.  ";
        utils.keyPhraseExtraction(keyPhrase);
    }
}
```

Key Phrases Detected:
- Khabib Abdulmanapovich Nurmagomedov (Confidence: 0.99991786)
- a professional mixed martial artist (Confidence: 0.9595572)
- the longest-reigning UFC Lightweight Champion (Confidence: 0.9998174)
- the title (Confidence: 0.99997145)
- April 2018 (Confidence: 0.9999066)
- March 2021 (Confidence: 0.99979985)

AWS Comprehend – Personally Identifiable Information

```java
public void piiDetection(String text) {

    // Create a request for PII detection
    DetectPiiEntitiesRequest request = DetectPiiEntitiesRequest.builder()
            .text(text)
            .languageCode("en") // Language of the text
            .build();

    // Get PII entities from AWS Comprehend
    DetectPiiEntitiesResponse response = this.client.detectPiiEntities(request);
    List<PiiEntity> piiEntities = response.entities();

    // Print detected PII entities
    System.out.println("Detected PII Entities:");
    for (PiiEntity entity : piiEntities) {
        System.out.println("- " + entity.type() + " found at position [" + entity.beginOffset() + " - " + entity.endOffset() + "]");
    }
}
```

```java
public class AWSComprehendController {

    public static void main(String[] args) {

        AWSComprehendUtils utils = new AWSComprehendUtils();
        String piiText = "My name is Mohammed Kaleem, "
                + "my email is m.kaleem@mmu.ac.uk, "
                + "and my phone number is 123-456-7890.";
        utils.piiDetection(piiText);
    }
}
```

Detected PII Entities:
- NAME found at position [11 - 26]
- EMAIL found at position [40 - 58]
- PHONE found at position [83 - 95]

Summary

- AWS Comprehend is an excellent API for NLP and text analysis.
- It has pretrained AI models for text analysis.
- Very easy to set up and use within existing applications that require NLP or Text processing/analysis.
- Has SDKs for Java, C++, JavaScript, PHP and more.