EduTranscribe: AI-Powered Lecture Search

This project focuses on creating an **AI-powered Knowledge Base** using **AWS services**, which allows users to upload audio lecture, transcribe it into text, store it in Amazon OpenSearch Serverless, and query the information using Amazon Bedrock.

By leveraging **Amazon Transcribe, OpenSearch Serverless, and Bedrock**, this system enables efficient retrieval of lecture teachings.

Technologies Used

- **Amazon S3** – Storage for lecture audio and transcription files.
- **Amazon Transcribe** – Converts audio files into text format.
- **Amazon OpenSearch Serverless** – Stores and indexes text for retrieval.
- **Amazon IAM** – Provides controlled access to AWS resources.
- **Amazon Bedrock** – Enables AI-based querying over the knowledge base.

Step 1: Upload Lecture Audio to Amazon S3

- **Storage Class:** Standard Storage
- **Bucket Access Configuration:**
  - Access is **restricted to IAM users**.
  - Public access is **not allowed**.

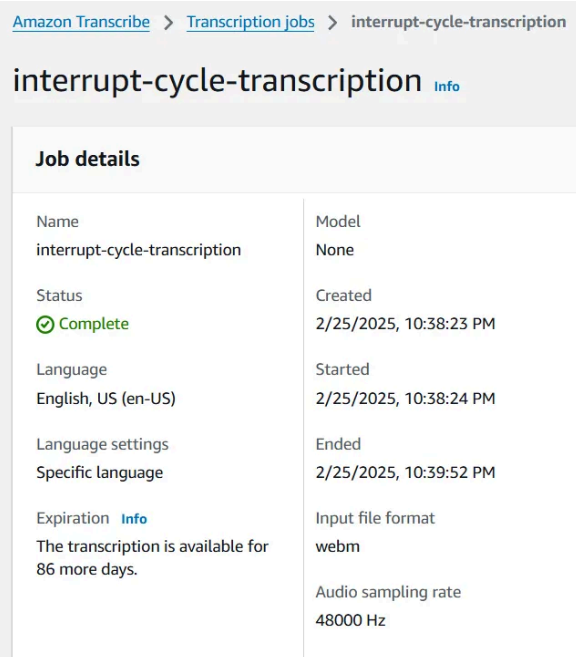
Command used for uploading the audio file:

```
aws s3 cp lecture_audio.mp3 s3://lecture-knowledge-base/
```

Step 2: Convert Audio to Text Using Amazon Transcribe

Amazon Transcribe was used to process the audio file and generate a textual transcript.

- **Features Enabled:**
    - Automatic punctuation.
    - Speaker identification.
    - Custom vocabulary (limited use, shortforms were preserved).



Amazon Transcribe > Transcription jobs > interrupt-cycle-transcription

interrupt-cycle-transcription

Job details

Name
interrupt-cycle-transcription

Status
Complete

Language
English, US (en-US)

Language settings
Specific language

Expiration
The transcription is available for 86 more days.

Model
None

Created
2/25/2025, 10:38:23 PM

Started
2/25/2025, 10:38:24 PM

Ended
2/25/2025, 10:39:52 PM

Input file format
webm

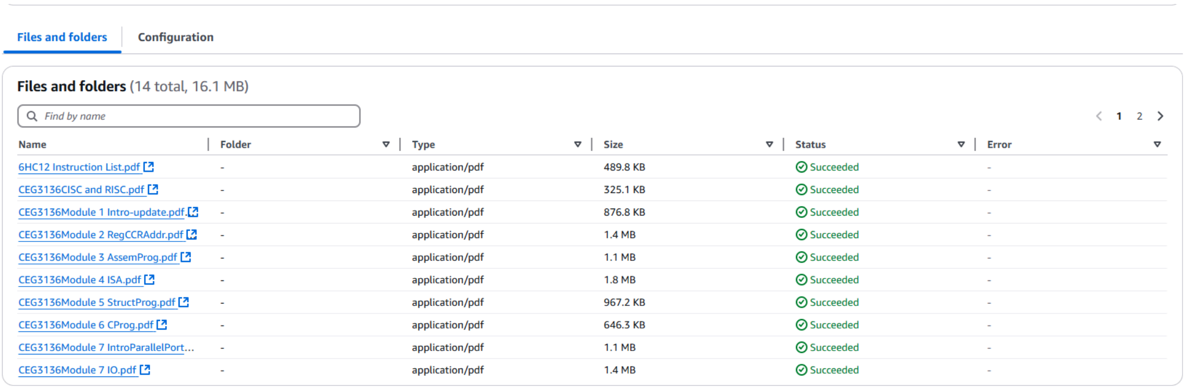
Audio sampling rate
48000 Hz

After the transcription process was completed, the resulting text file was **downloaded in JSON format**.

Step 3: Save the Transcribed Text to Amazon S3

Once transcription was complete, the **generated text was stored back in S3** for further processing.

- **Storage Class:** Standard Storage



Files and folders (14 total, 16.1 MB)

Name	Folder	Type	Size	Status	Error
6HC12 Instruction List.pdf	-	application/pdf	489.8 KB	✓ Succeeded	-
CEG3136CISC and RISC.pdf	-	application/pdf	325.1 KB	✓ Succeeded	-
CEG3136Module 1 Intro-update.pdf	-	application/pdf	876.8 KB	✓ Succeeded	-
CEG3136Module 2 RegCCRAddr.pdf	-	application/pdf	1.4 MB	✓ Succeeded	-
CEG3136Module 3 AssemProg.pdf	-	application/pdf	1.1 MB	✓ Succeeded	-
CEG3136Module 4 ISA.pdf	-	application/pdf	1.8 MB	✓ Succeeded	-
CEG3136Module 5 StructProg.pdf	-	application/pdf	967.2 KB	✓ Succeeded	-
CEG3136Module 6 CProg.pdf	-	application/pdf	646.3 KB	✓ Succeeded	-
CEG3136Module 7 IntroParallelPort...	-	application/pdf	1.1 MB	✓ Succeeded	-
CEG3136Module 7 IO.pdf	-	application/pdf	1.4 MB	✓ Succeeded	-

Step 4: Create an IAM User for Access Control

A dedicated **IAM user** was created to manage access to AWS resources.

- **Permissions Assigned:**
    - `AmazonBedrockFullAccess`
    - `AmazonOpenSearchServiceFullAccess`
    - `AmazonS3FullAccess`
    - **A custom policy** was also applied, with a **limited session duration for security reasons**.

This configuration ensures that the IAM user has the **necessary permissions** to work with Bedrock, OpenSearch, and S3 while maintaining security best practices.

Step 5: Set Up a Knowledge Base in Amazon Bedrock

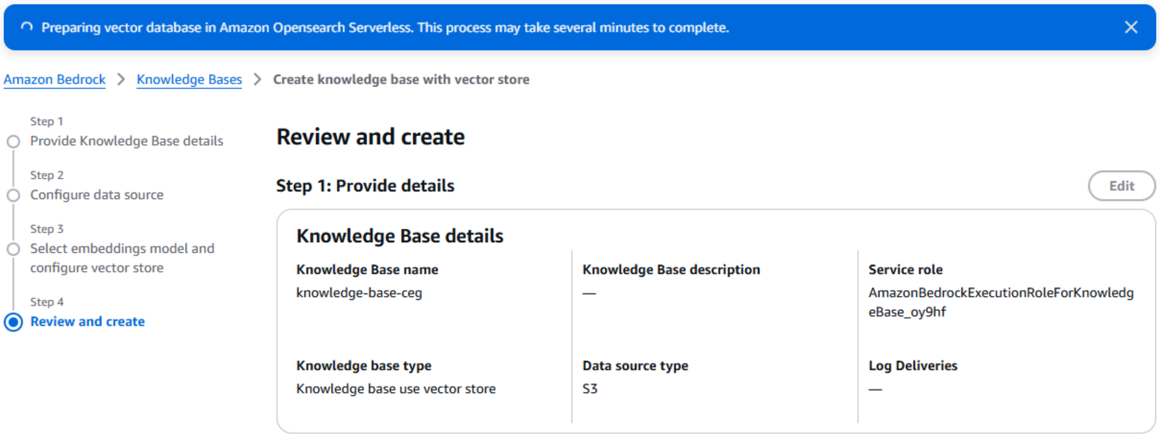
Amazon Bedrock was used to create a **Knowledge Base**, enabling AI-powered retrieval of lecture content.

- **Storage Backend:** Amazon OpenSearch Serverless
- **Data Source:** Amazon S3
- **Embedding Model Used:** Titan (AWS)
- **Search Method:** Vector Search

- **Auto-Sync Configuration:** Enabled for automatic updates when new data is added.

The system was configured to **automatically sync** new lecture transcripts into the Knowledge Base, ensuring **real-time updates**.

Since the data was sourced **only from trusted lecture transcripts**, **ranking or filtering mechanisms were not applied**.



Preparing vector database in Amazon Opensearch Serverless. This process may take several minutes to complete.

Amazon Bedrock > Knowledge Bases > Create knowledge base with vector store

Step 1
Provide Knowledge Base details
Step 2
Configure data source
Step 3
Select embeddings model and configure vector store
Step 4
Review and create

Review and create

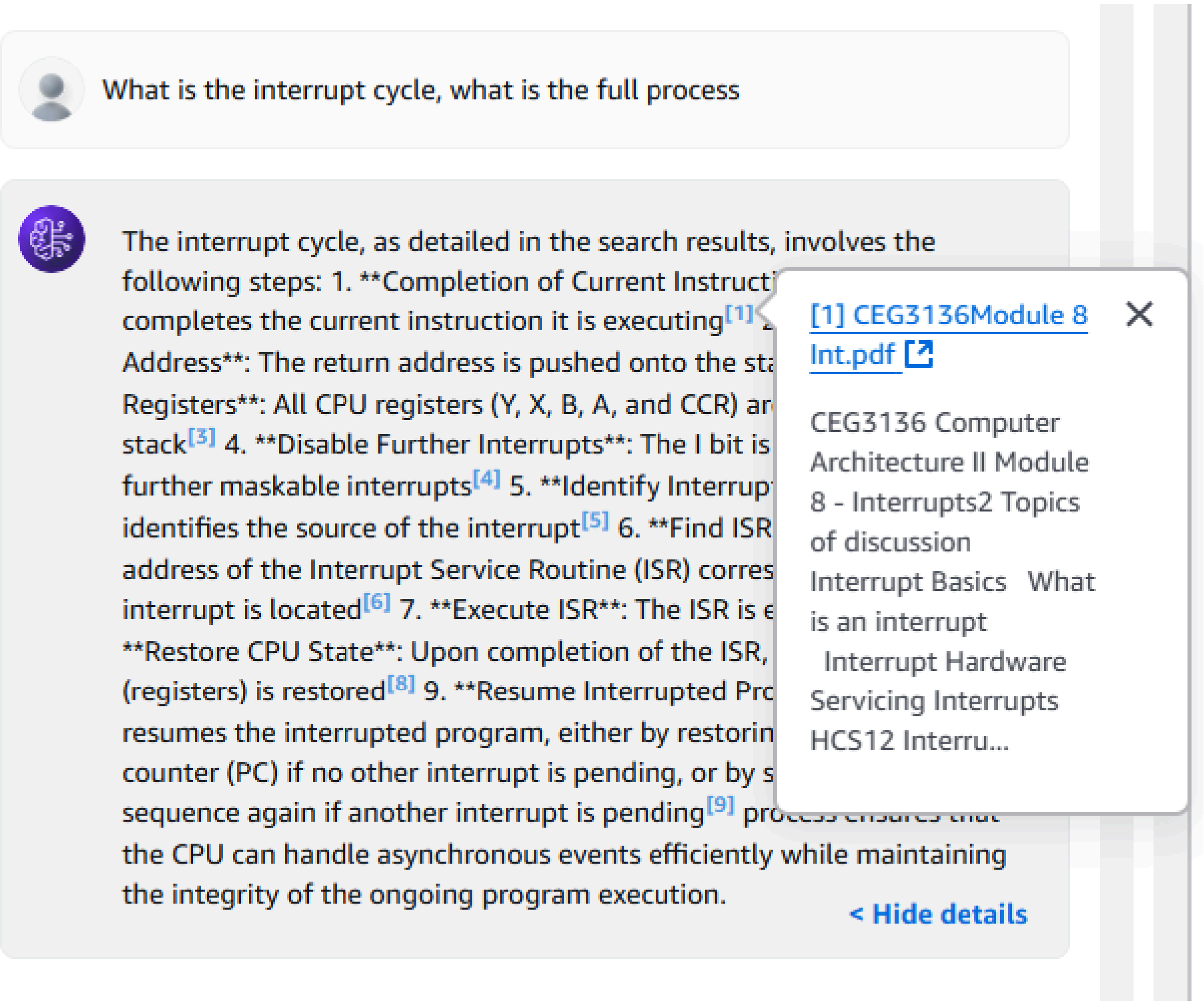
Step 1: Provide details

Knowledge Base details

Knowledge Base name	Knowledge Base description	Service role
knowledge-base-ceg	—	AmazonBedrockExecutionRoleForKnowledgeBase_oy9hf
Knowledge base type	**Data source type**	**Log Deliveries**
Knowledge base use vector store	S3	—

Step 6: Query the Knowledge Base Using Amazon Bedrock

Once the Knowledge Base was set up, queries were performed using Amazon Bedrock.



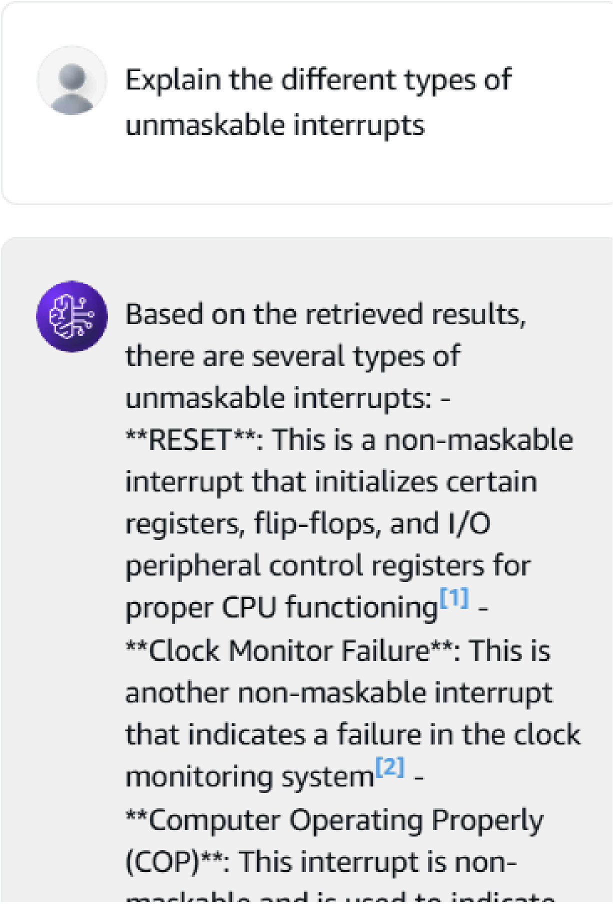
What is the interrupt cycle, what is the full process

The interrupt cycle, as detailed in the search results, involves the
following steps: 1. **Completion of Current Instructi
completes the current instruction it is executing[1] 2
Address**: The return address is pushed onto the sta
Registers**: All CPU registers (Y, X, B, A, and CCR) ar
stack[3] 4. **Disable Further Interrupts**: The I bit is
further maskable interrupts[4] 5. **Identify Interrup
identifies the source of the interrupt[5] 6. **Find ISR
address of the Interrupt Service Routine (ISR) corres
interrupt is located[6] 7. **Execute ISR**: The ISR is e
Restore CPU State: Upon completion of the ISR,
(registers) is restored[8] 9. **Resume Interrupted Pro
resumes the interrupted program, either by restorin
counter (PC) if no other interrupt is pending, or by s
sequence again if another interrupt is pending[9] process ensures that
the CPU can handle asynchronous events efficiently while maintaining
the integrity of the ongoing program execution.

< Hide details

[1] CEG3136Module 8 Int.pdf

CEG3136 Computer Architecture II Module 8 - Interrupts2 Topics of discussion Interrupt Basics   What is an interrupt Interrupt Hardware Servicing Interrupts HCS12 Interru...



Explain the different types of unmaskable interrupts

Based on the retrieved results, there are several types of unmaskable interrupts: - **RESET**: This is a non-maskable interrupt that initializes certain registers, flip-flops, and I/O peripheral control registers for proper CPU functioning[1] - **Clock Monitor Failure**: This is another non-maskable interrupt that indicates a failure in the clock monitoring system[2] - **Computer Operating Properly (COP)**: This interrupt is non-maskable and is used to indicate