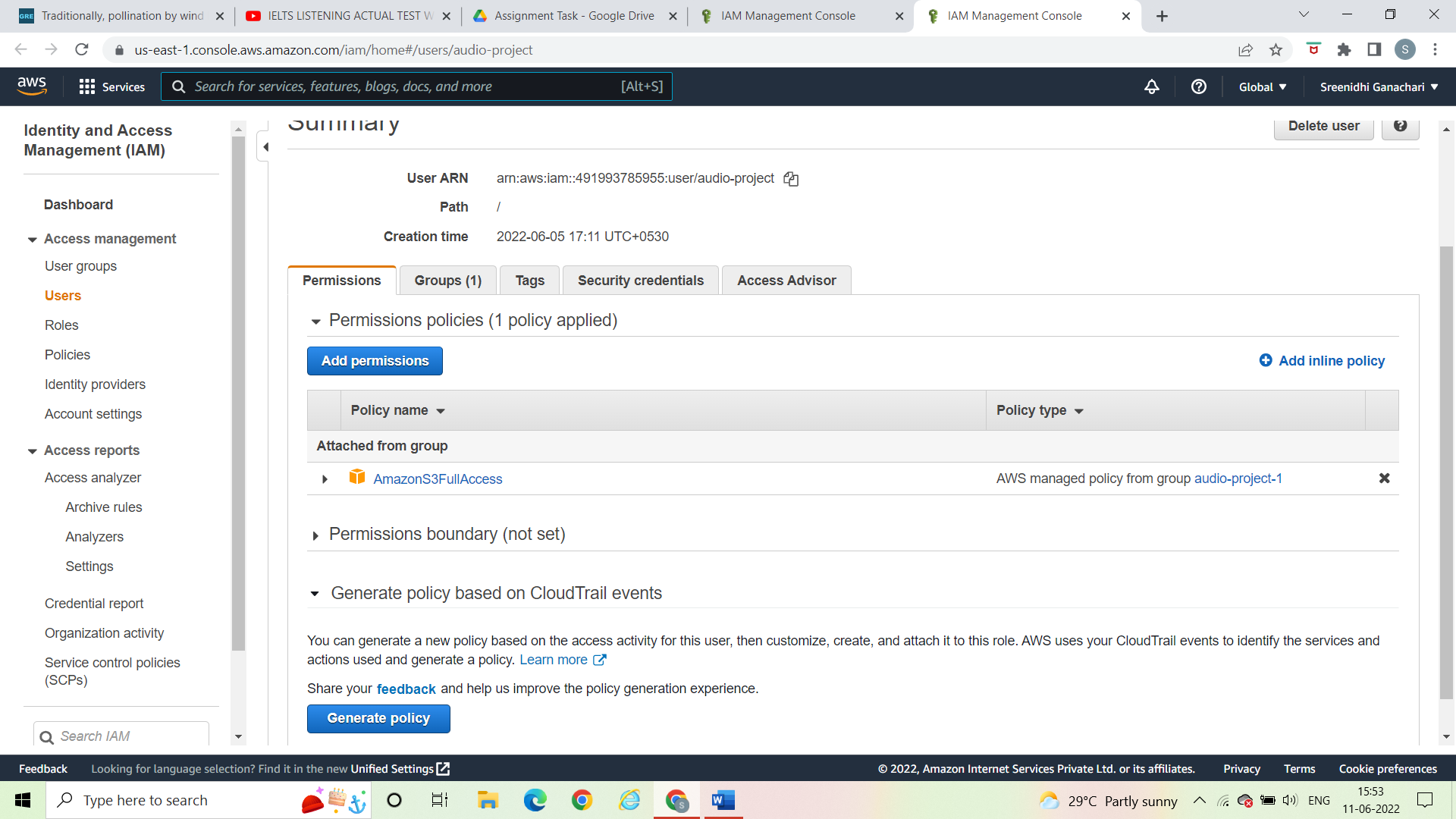
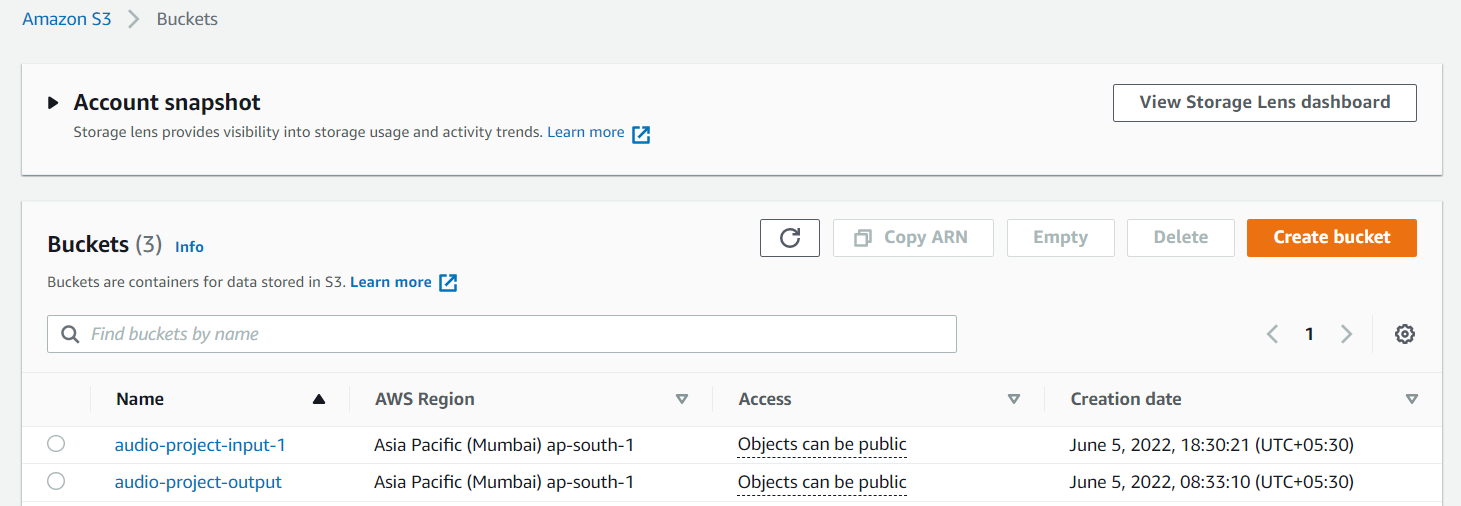
**STEPS TO SUMMARIZE TEXT IN AN AUDIO CLIP**

Steps –

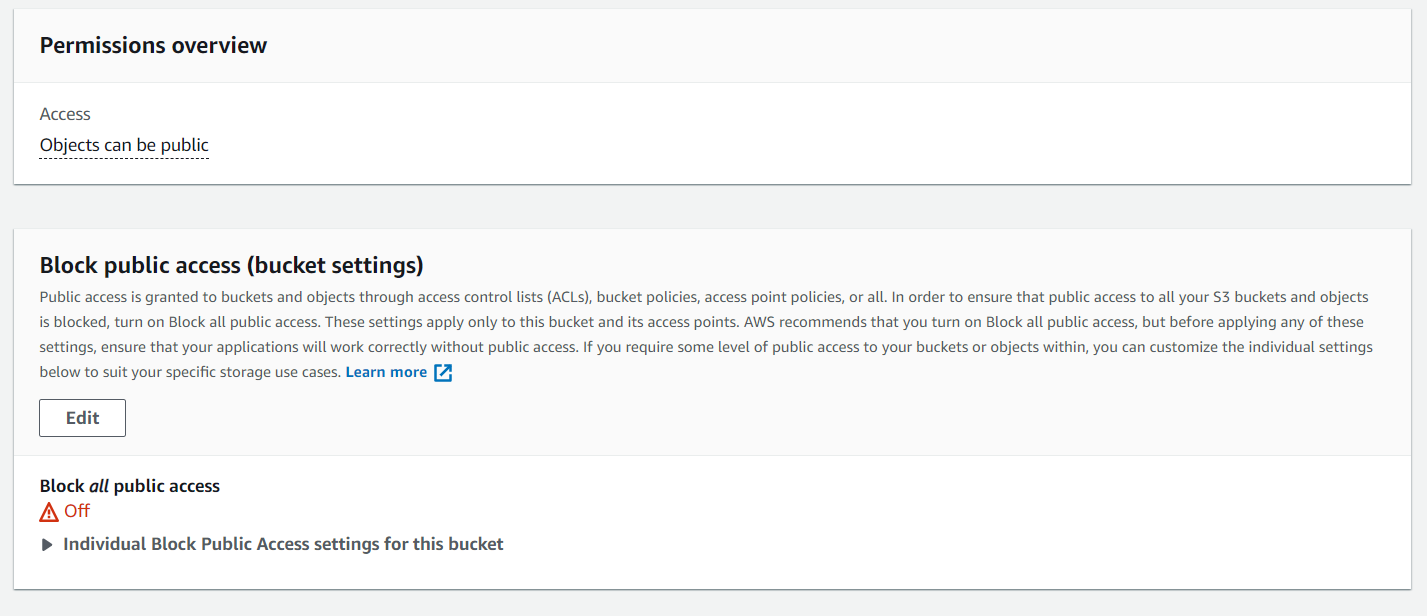
1. The first step is the creation of a basic html page using the code uploaded in the drive .
2. We must also write a javascript code to upload the audio file into the s3 bucket .
3. We need to provide an access key and a secret key in the index.js file , this can be obtained by creating an IAM role and provide access to the s3 bucket as shown below .



1. In the S3 cloud service create 2 buckets – one for input and one for output



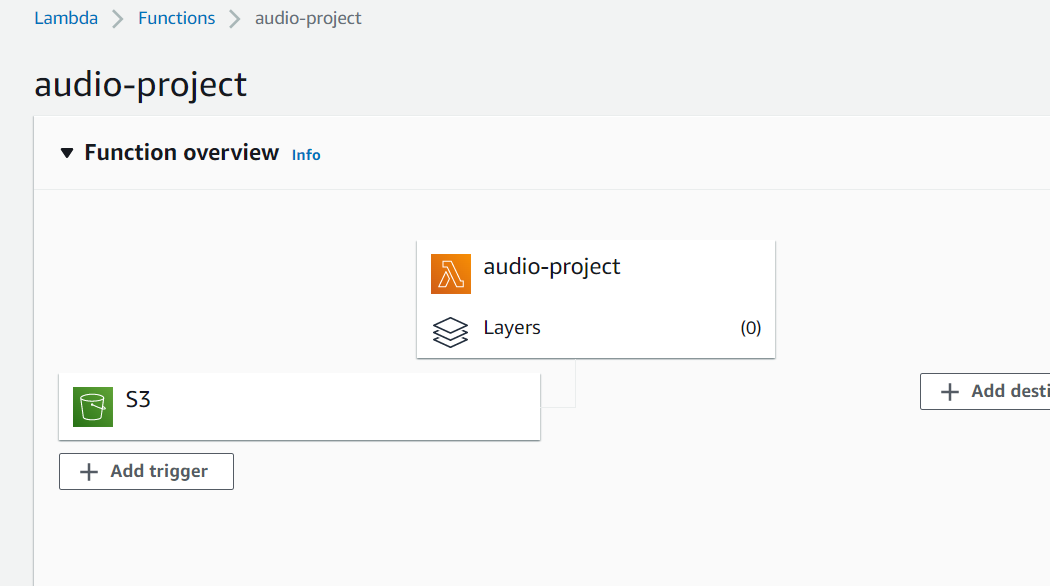
1. We have to provide complete public access to the input bucket and also ad this IAM role to the input bucket .



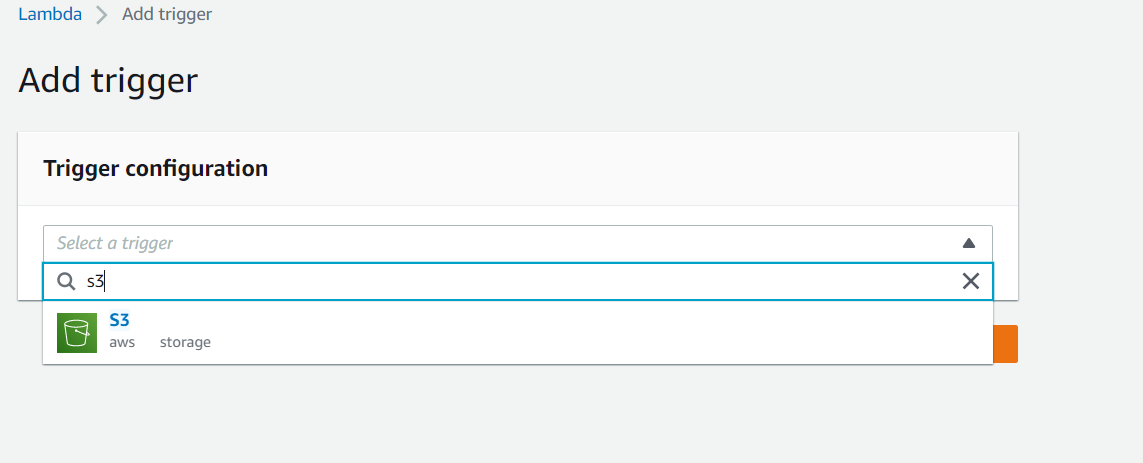
1. Next , we need to use lambda to trigger the s3 bucket to send the uploaded file directly to AWS Transcribe .

In Lambda follow the below steps –

1. Create a function and after it is created , we can create a trigger here as shown below .



1. When we create a trigger , choose s3



Also select the input bucket name in the next requirements asked .

This will create a trigger .

1. Using python we are sending the uploaded mp3 extension file to AWS Transcribe , after that the output we get is automatically stored in the output S3 bucket we had created previously .

Below is the python code used –

import boto3

import uuid

import json

s3 = boto3.client('s3')

transcribe = boto3.client('transcribe')

def lambda\_handler(event, context):

try:

file\_bucket = event['Records'][0]['s3']['bucket']['name']

file\_name = event['Records'][0]['s3']['object']['key']

object\_url = 'https://s3.amazonaws.com/{0}/{1}'.format(file\_bucket, file\_name)

transcriptionJobDetails=startTranscriptionJob(file\_name,object\_url)

status = getTranscriptionJob(file\_name)

url=status['TranscriptionJob']['Transcript']['TranscriptFileUri']

Text\_Data = (requests.get(url).json())['results']['transcripts'][0]['transcript']

file = open(f"/tmp/{file\_name}.txt", "w")

file.write(Text\_Data)

file.close()

s3.upload\_file(

Filename = f"/tmp/{file\_name}.txt" ,

# Bucket = "audio-project-output" ,

Key = f"{file\_name}.txt"

)

except Exception as e:

raise e

def startTranscriptionJob(file\_name,object\_url):

response = transcribe.start\_transcription\_job(

TranscriptionJobName=file\_name.replace('/','')[:10],

IdentifyLanguage= True,

MediaFormat='mp3',

Media={

'MediaFileUri': object\_url

},

OutputBucketName = "audio-project-output"

)

return response

def getTranscriptionJob(file\_name):

while True:

status = transcribe.get\_transcription\_job(

TranscriptionJobName=file\_name.replace('/','')[:10]

)

if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:

break

return status

1. The output would be uploaded in json format in output bucket .

Now , we need to download this output to our local system . Using java swing we are accessing the s3 bucket file using a presigned URL .

1. The content of the file from the URL is uploaded into an empty text file available on our local system .
2. The next part is summarizing the text we have got from the audio file .
3. For this purpose we have used jupyter notebook and used Spacy libraries which is mainly used for Natural Language Processing .
4. We upload the text file to the colab notebooks file in the drive and the code accesses it from there and takes its content .
5. The code used is uploaded in the drive and it helps us to summarize the text we have .