YouTube to MP3 Downloader with AWS Lambda

Introduction

This document provides a step-by-step guide for implementing a **YouTube to MP3 Downloader** using **AWS Lambda** and **S3**. It covers the prerequisites, setup steps, code, configuration, and deployment process for the Lambda function.

1 Prerequisites

- AWS Account with Lambda and S3 services enabled.
- Python 3.6 or later.
- Install **boto3** and **pytube** libraries: `pip install boto3 pytube`.

2 Step-by-Step Implementation

2.1 Step 1: Set Up S3 Bucket

1. Open the **Amazon S3** console.
2. Create a new S3 bucket, e.g., `youtube-2-mp3-files`, for storing the converted MP3 files.
3. Configure permissions to allow access from your Lambda function.

2.2 Step 2: Create the Lambda Function

1. Open the **AWS Lambda Console**.
2. Choose **Create Function**.
3. Select **Author from scratch**.
4. Name the function (e.g., `youtube-to-mp3-converter`).
5. Set **Runtime** to Python 3.6 or later.
6. Configure the Lambda Execution Role with basic Lambda permissions and add S3 access permissions.

2.3 Step 3: Set Up Environment Variables in Lambda

1. In the Lambda function's **Configuration tab**, go to **Environment variables**.
2. Add a new variable:
   - **Key**: `destination_bucket`
   - **Value**: Name of your S3 bucket (e.g., `youtube-2-mp3-files`).

2.4 Step 4: Write and Upload the Code

1. Copy and paste the following code into the Lambda editor:

```
import json
from pytube import YouTube
import os
import boto3
import logging
from botocore.exceptions import ClientError

def lambda_handler(event, context):
    # Get the destination bucket from the environment variable
    destination_bucket = os.getenv("destination_bucket")

    # URL input from event (assuming the URL is passed as an event parameter)
    video_url = event.get("video_url", "https://youtu.be/Mt-JJBabxiA")

    # Initialize YouTube and extract audio
    yt = YouTube(video_url)
    video = yt.streams.filter(only_audio=True).first()

    # Set destination directory to /tmp (Lambdas writable directory)
    destination = "/tmp"
    out_file = video.download(output_path=destination)

    # Convert to MP3 format
    base, ext = os.path.splitext(out_file)
    new_file = base + '.mp3'
    os.rename(out_file, new_file)

    # Upload the file to S3
    upload_success = upload_file(new_file, destination_bucket)
    if upload_success:
        return {
            'statusCode': 200,
            'body': json.dumps(f"{yt.title} has been successfully downloaded and uploaded to S3.")
        }
    else:
        return {
            'statusCode': 500,
```

```
            'body': json.dumps("Failed to upload the file to S3.")
        }

def upload_file(file_name, bucket, object_name=None):
    """Upload a file to an S3 bucket"""
    if object_name is None:
        object_name = os.path.basename(file_name)

    s3_client = boto3.client('s3')
    try:
        s3_client.upload_file(file_name, bucket, object_name)
    except ClientError as e:
        logging.error(e)
        return False
    return True
```

Listing 1: AWS Lambda Python Code

2.5 Step 5: Configure Lambda Timeout and Memory

- Increase **Timeout** in Settings ¿ Basic Settings to 1-2 minutes.
- Allocate more **Memory** (e.g., 512 MB or 1024 MB) for performance if needed.

2.6 Step 6: Testing the Function

1. In the Lambda function's **Test** section, create a new test event with the following JSON:

```
{
    "video_url": "https://youtu.be/Mt-JJBabxiA"
}
```

2. Run the test and check the logs in CloudWatch to monitor the download, conversion, and upload process.
3. Verify the MP3 file is uploaded to your S3 bucket.

3 Limitations and Considerations

- **Execution Time Limits**: Lambda functions have a maximum execution time limit of 15 minutes.
- **pytube Library Limitations**: Some videos may have restrictions.
- **Security and Permissions**: Ensure the Lambda function has only the necessary permissions.

4 Important Notes

Use this code responsibly and only with videos you have permission to download and convert. This code is provided for educational purposes only, and misuse may lead to legal consequences.