SUMMARY

Our VOD pipeline on GCP consists of several functions, and in this document, we will present the analysis of one of these functions named *vod-vid-to-aud-conversion*. The *vod-vid-to-aud-conversion* cloud function extracts audio from MP4 video files stored in a Google Cloud Storage bucket. When a file is added or modified in the source bucket, the function is triggered. It downloads the video file to a temporary local directory, uses the FFmpeg tool to convert the video content into a WAV audio file, and then transfers this audio file to a destination bucket structured for further processing. It also includes logging mechanisms to track critical steps and cleaning up temporary files after execution.

Table of Contents

[I- INTRODUCTION 1](#_Toc182953968)

[I-1 Overview of the Cloud Function's purpose and primary operations 1](#_Toc182953969)

[I-2 Architecture Overview 1](#_Toc182953970)

[Figure 1: Algorithmic diagram of the *vod-vid-to-aud-conversion* cloud function 1](#_Toc182953971)

[I-2-1 Function Details 2](#_Toc182953972)

[II- IMPLEMENTATION OF THE MAIN.PY FUNCTION 4](#_Toc182953973)

[Figure 2: Import Statements 4](#_Toc182953974)

[Figure 3: Client and Logger Initialization 4](#_Toc182953975)

[Figure 4: Defining the Cloud Function 4](#_Toc182953976)

[Figure 5: Extracting Event and File Information 5](#_Toc182953977)

[Figure 6 : Defining Destination Bucket and Local Paths 5](#_Toc182953978)

[Figure 7: Downloading the Video File 5](#_Toc182953979)

[Figure 8: Converting Video to Audio 6](#_Toc182953980)

[Figure 9: Uploading the Audio File to the Destination Bucket 6](#_Toc182953981)

[Figure 10: Cleaning Up Temporary Files 7](#_Toc182953982)

[III- TESTING AND VALIDATION 8](#_Toc182953983)

[IV- REFERENCES AND EXTERNAL DOCUMENTATION 9](#_Toc182953984)

[V- GLOSSARY 10](#_Toc182953985)

# 

# INTRODUCTION

## I-1 Overview of the Cloud Function's purpose and primary operations

The *vod-vid-to-aud-conversion* cloud function is triggered by Google Cloud Storage events. It extracts audio from an MP4 video file and stores the resulting WAV audio file in a specified GCP bucket for further processing.

## I-2 Architecture Overview

Une image contenant capture d’écran, Caractère coloré, carré, conception

Description générée automatiquement

Figure 1: Algorithmic diagram of the *vod-vid-to-aud-conversion* cloud function

**Workflow Explanation:**

* **Trigger:** A file is uploaded to a Google Cloud Storage bucket.
* **Download:** The video file is downloaded to a local temporary directory.
* **Conversion:** The video file is processed with FFmpeg to extract its audio.
* **Upload:** The audio file is uploaded to a destination bucket in Google Cloud Storage.
* **Cleanup:** Temporary files are deleted from the local directory.

The function relies on several libraries to achieve its goals:

* **google-cloud-storage** handles file management before and after processing:
  + Downloading the video from the source bucket: “vodunprocessedgcp”
  + Uploading the converted audio file to the destination bucket: “vodprocessedgcp”
* **ffmpeg-python** performs the core processing:
  + Extracting the audio from the video file.

### I-2-1 Function Details

The *vod-vid-to-aud-conversion* cloud function consists of two files which are in the same directory:

* main.py: Contains the main logic for handling events, extracting audio from video, converting audio format and storing the converted audio file in an output bucket.
* requirements.txt: Lists Python dependencies required for the function to operate.
  + **google-cloud-storage**

**Role**: used for managing files in your cloud buckets (object storage).

**In the function**: The library is used to retrieve the video file from the source bucket and after converting the video file to audio (WAV format), the library is used to upload the generated file to another bucket (the destination bucket). Additionally, the library simplifies the creation and management of structured paths, by efficiently organizing files within buckets.

* + **ffmpeg-python**

**Role :** FFmpeg is a collection of open-source libraries and tools used for processing multimedia files, including encoding, decoding, format conversion, editing, and streaming.

**In the function:** FFmpeg is used to extract audio from an MP4 video file and save it in WAV format. This library is also known for its speed and accuracy in handling multimedia formats, which is essential for tasks such as audio extraction and file conversion.

**Environment and Configuration**

* **Permissions:** The function requires the following IAM permissions:
  + storage.objects.get: To download video files from the source bucket.
  + storage.objects.create: To upload audio files to the destination bucket.
* **Dependencies :** Python Packages:
  + google-cloud-storage: To interact with GCP storage buckets.
  + subprocess: To execute ffmpeg commands.
* **External Tool**:
  + FFmpeg: Used for MP4 to WAV conversion.

# IMPLEMENTATION OF THE MAIN.PY FUNCTION



Figure 2: Import Statements

* **from google.cloud import storage**: Allows interaction with Google Cloud Storage, enabling file upload and download operations.
* **import os**: Provides utilities for file and path management, such as creating and deleting temporary files.
* **import subprocess**: Enables running shell commands, used here to execute FFmpeg for video-to-audio conversion.
* **import logging**: Facilitates logging information, warnings, and errors for debugging and monitoring.

Une image contenant texte, capture d’écran, Police, blanc

Description générée automatiquementFigure 3: Client and Logger Initialization

* **storage\_client = storage.Client()**: Initializes a client to communicate with Google Cloud Storage.
* **logger = logging.getLogger()**: Creates a logging object to capture messages.
* **logger.setLevel(logging.INFO)**: Configures the logger to display messages with a severity level of INFO or higher.

Figure 4: Defining the Cloud Function

* **event**: Contains metadata about the event that triggered the function (e.g., details of the uploaded file in the source bucket).
* **context**: (Optional) Provides additional event information like the event ID and timestamp.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquementFigure 5: Extracting Event and File Information

* **event['bucket']**: Retrieves the name of the source bucket where the video file was uploaded.
* **event['name']**: Retrieves the name of the video file in the bucket.
* **os.path.basename(video\_file\_name)**: Extracts the file name from the file path.
* **os.path.splitext(file\_name)[0]**: Removes the file extension to get the base name of the file, which is used for naming the output audio file.

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquementFigure 6 : Defining Destination Bucket and Local Paths

* **destination\_bucket\_name**: Specifies the bucket where the processed audio file will be uploaded.
* **local\_mp4\_path**: Temporary local path for downloading the video file.
* **local\_audio\_path**: Temporary local path for storing the converted audio file.

Une image contenant texte, capture d’écran, Police

Description générée automatiquementFigure 7: Downloading the Video File

* **storage\_client.bucket(source\_bucket\_name).blob(video\_file\_name)**: Accesses the video file in the source bucket.
* **download\_to\_filename(local\_mp4\_path)**: Downloads the video file to the local temporary directory.
* **Error Handling**: Logs and returns an error message if the download fails.

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquementFigure 8: Converting Video to Audio

* **subprocess.run()**: Executes FFmpeg to extract audio from the video file.
  + **-i**: Specifies the input video file.
  + **-vn**: Ignores the video stream.
  + **-acodec pcm\_s16le**: Sets the audio codec to PCM (16-bit little-endian).
  + **-ar 44100**: Sets the sampling rate to 44.1 kHz.
  + **-ac 1**: Converts the audio to mono.
* **Error Handling**: Logs and returns an error message if the conversion fails.

Une image contenant texte, capture d’écran, Police

Description générée automatiquementFigure 9: Uploading the Audio File to the Destination Bucket

* **processed\_bucket\_name**: Defines the destination bucket for the audio file.
* **wav\_blob\_path**: Specifies the folder structure and name for the audio file in the bucket.
* **upload\_from\_filename(local\_audio\_path)**: Uploads the audio file to the destination bucket.
* **logger.info**: Logs the success of the upload.

Une image contenant texte, capture d’écran, Police, algèbre

Description générée automatiquementFigure 10: Cleaning Up Temporary Files

* **os.path.exists()**: Checks if the temporary files exist in the local directory.
* **os.remove()**: Deletes the temporary files to free up storage and maintain efficiency.

# TESTING AND VALIDATION

**Processing Steps** :

1. Extract file metadata from the event payload (source bucket and file name).
2. Download the video file from the source bucket to a temporary local path (/tmp).
3. Convert the MP4 video file to WAV audio format using FFmpeg.
4. Upload the converted WAV file to the destination bucket with a structured file path.
5. Clean up temporary files to free up local storage space.

**Unit Tests** :

* Validate simulated events (event) for correctness.
* Test scenarios for download errors and insufficient permissions.

**Integration Tests** :

* Ensure that properly formatted video files in the source bucket are successfully converted and uploaded to the destination bucket.

# REFERENCES AND EXTERNAL DOCUMENTATION

* **Official GCP Documentation**:
  + Cloud Functions Overview : [https://cloud.google.com/functions/docs/concepts/overview?\_gl=1\*1odtn9b\*\_ga\*ODAwNzc2OTM2LjE3MTM3NDEzMjA.\*\_ga\_WH2QY8WWF5\*MTczMjA1NjIwMy4xMjkuMS4xNzMyMDU2MjM3LjI2LjAuMA..](https://cloud.google.com/functions/docs/concepts/overview?_gl=1*1odtn9b*_ga*ODAwNzc2OTM2LjE3MTM3NDEzMjA.*_ga_WH2QY8WWF5*MTczMjA1NjIwMy4xMjkuMS4xNzMyMDU2MjM3LjI2LjAuMA..)
  + Cloud Storage Client Library : [https://cloud.google.com/python/docs/reference/storage/1.22.0/client?hl=en\_GB&\_gl=1\*t1ya0g\*\_ga\*ODAwNzc2OTM2LjE3MTM3NDEzMjA.\*\_ga\_WH2QY8WWF5\*MTczMjA1NjIwMy4xMjkuMS4xNzMyMDU2NDQ0LjYwLjAuMA..](https://cloud.google.com/python/docs/reference/storage/1.22.0/client?hl=en_GB&_gl=1*t1ya0g*_ga*ODAwNzc2OTM2LjE3MTM3NDEzMjA.*_ga_WH2QY8WWF5*MTczMjA1NjIwMy4xMjkuMS4xNzMyMDU2NDQ0LjYwLjAuMA..)
* **FFmpeg Documentation**:
  + [FFmpeg Official Site](https://ffmpeg.org/documentation.html)

# GLOSSARY

VOD : Video-On-Demand

GCP : Google Cloud Platform

FFmpeg: Fast Forward Moving Pictures Expert Group

MP4 : MPEG-4 Part 14

WAV : Waveform Audio File Format