

Búsqueda de Videos, AWS

Franklin Canaza, Jose Torres, Kevin Salazar

Diciembre del 2020

1 Interfaz

Características principales de la interfaz:

- La consulta de una etiqueta retorna los videos que la contengan en algún momento de su duración. Estos videos retornados están ordenados por un *rating* que indica la mayor presencia de la etiqueta buscada.
- Debajo del video se muestra una lista de tiempos, la cual indica los momentos en los que aparece el objeto que pertenece a dicha etiqueta de búsqueda. Esta lista también está ordenada según el tiempo en el que se note la mayor presencia de frames consecutivos con esa etiqueta.
- Lenguaje de programación: Python 3
- Framework: Flask
- Integración de servicios con AWS: Boto3
- Servidor Web: nginx

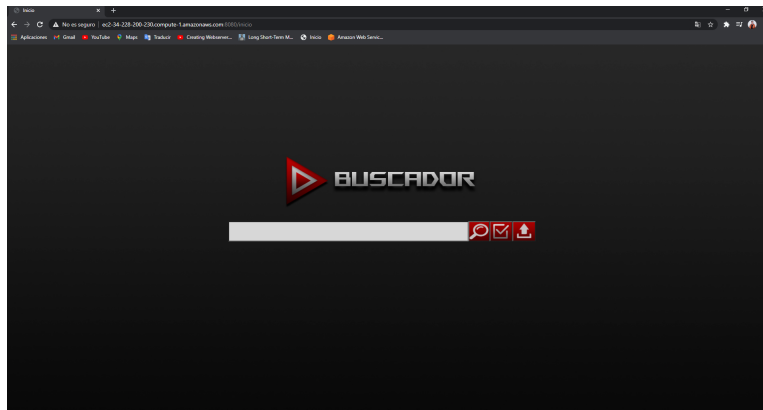


Figure 1: Interfaz de inicio.

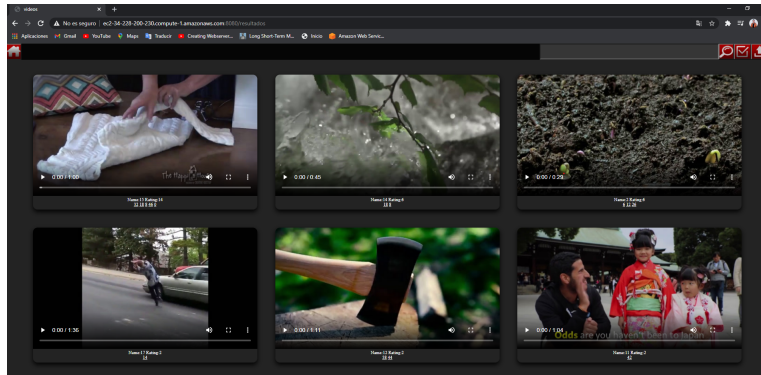


Figure 2: Interfaz de resultados.

Procedimiento para la implementación del servidor en terminal:

1. sudo apt-get update
2. sudo apt-get install python3
3. sudo apt-get install python3-pip
4. sudo pip3 install flask
5. sudo apt-get install nginx
6. sudo apt-get install gunicorn3
7. sudo apt install awscli
8. aws configure (Id de acceso y Clave secreta)
9. aws s3 cp s3://bucketservervideo/webserverflask /home/ubuntu --recursive

Para el acceso a los objetos del bucket es necesario agregar una política en la configuración de permisos del bucket que contiene los recursos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPublicRead",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::bucketservervideo/*"
    }
  ]
}
```

2 Creación de los Lambdas y procesamiento con ffmpeg

- Creación del Lambda para obtener las miniaturas (frame de los vídeos de tamaño reducido) que se activa cuando un vídeo es subido al Bucket de vídeos. Las miniaturas son obtenidas usando ffmpeg que es un software de código libre que nos permite procesar un video, los binarios de ffmpeg son usados para esta tarea, que son colocados en un bucket y luego son cargados a la capa de la Lambda. Ver Fig 3.
- Creación del Lambda para obtener los gifs de los vídeos. Para una vista previa para cada vídeo subido usando ffmpeg para lo obtención del gif.
- Creación del Lambda para obtener un Json con la información del vídeo (titulo, fecha, duración, id).
- Creación del Lambda para obtener un Json de los tiempos de los frames de un vídeo.
- Creación de un tema de *Amazon Simple Notification Service* para que las notificaciones de eventos de creación de objetos de S3 se envíen a este tema.
- Un evento de notificación de S3 para publicar un evento en el tema de SNS cuando se carga un archivo en el bucket de entrada del vídeo. Ver Fig. 4

Pasos realizados para calcular las miniaturas del vídeo.

1. Un archivo de vídeo se carga en un bucket de S3 (bucket de entrada de vídeo)
2. Se envía un evento ObjectCreate (All) desde S3 a un tema de SNS (*Amazon Simple Notification Service*) llamado new-video-topic
3. La notificación de SNS activa las funciones Lambdas para calcular las miniaturas, el gif, y los json con la información del vídeo.
4. La función calcular miniaturas descarga el vídeo de S3, ejecuta el código de generación de miniaturas y carga varias miniaturas en un bucket de S3 (bucket de miniaturas).

Debe tenerse en cuenta que S3 no activa directamente la función Lambda. Debido a que S3 no admite la notificación de múltiples funciones de Lambda para el mismo tipo de evento, se utilizó *Amazon Simple Notification Service*(SNS) para distribuir el evento a múltiples funciones Lambdas. Ver la estructura en la Fig 4

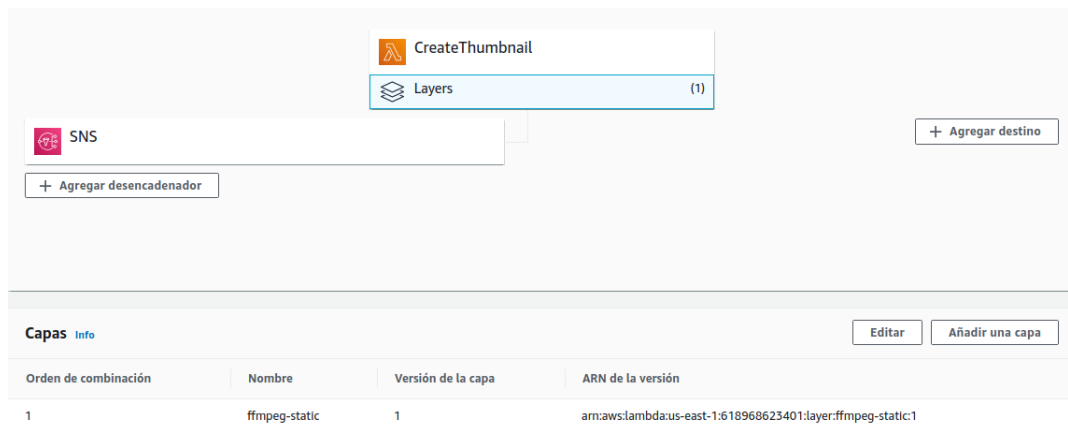


Figure 3: Activando el Lambda desde SNS y mostrando la capa del Lambda con ffmpeg

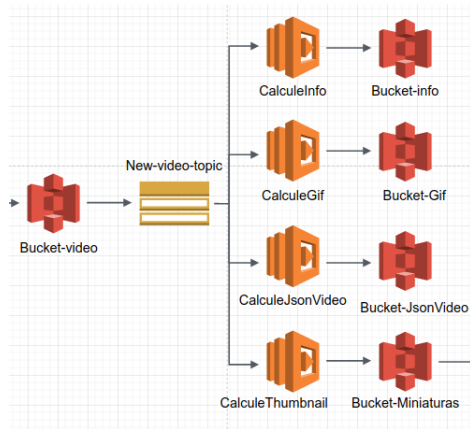


Figure 4: Diseño y comunicación de las Lambdas, buckets y el tema SNS

Políticas de accesos al tema new-topic-video desde S3, que publica una notificación cuando un objeto vídeo se sube en el bucket vídeos.

```

{
  "Version": "2008-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SNS:Publish",
    }
  ]
}

```

```

    "Resource": "arn:aws:sns:us-east-1:618968623401:new-topic-video",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:s3:::fjk2-bucket-video"
      }
    }
  }
]
}

```

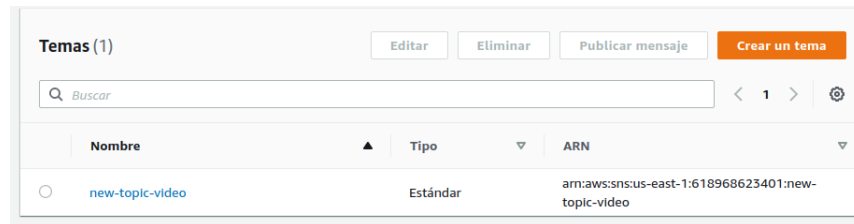


Figure 5: Tema creado para recibir notificaciones desde S3

	Nombre de la función ▾	Descripción	Tipo de paquete ▾	Tiempo de ejecución ▾	Tamaño del código ▾	Última modificación ▾
<input type="radio"/>	CreateJsonvideo		Zip	Node.js 12.x	1.2 kB	ayer
<input type="radio"/>	agregarATablaInfo		Zip	Python 3.8	828 bytes	hace 4 horas
<input type="radio"/>	agregarATablaDynamoDB		Zip	Python 3.8	821 bytes	hace 3 días
<input type="radio"/>	CreateGif		Zip	Node.js 12.x	1.4 kB	ayer
<input type="radio"/>	IndexarJsonVideo		Zip	Python 3.8	1.7 kB	hace 7 horas
<input type="radio"/>	CreateInfo		Zip	Node.js 12.x	1.3 kB	ayer
<input type="radio"/>	CreateThumbnail		Zip	Node.js 12.x	1.5 kB	ayer

Figure 6: Funciones Lambda creadas y configuradas con el tema SNS




	Nombre ▲	Región ▼	Acceso ▼	Fecha de creación ▼
		east-1	publicos	-05
<input type="radio"/>	fjk2-bucket-gif	EE. UU. Este (Norte de Virginia) us-east-1	 Público	28 Dec 2020 9:54:11 PM -05
<input type="radio"/>	fjk2-bucket-img	EE. UU. Este (Norte de Virginia) us-east-1	 Público	28 Dec 2020 9:52:12 PM -05
<input type="radio"/>	fjk2-bucket-index	EE. UU. Este (Norte de Virginia) us-east-1	Bucket y objetos que no son públicos	24 Dec 2020 10:05:29 PM -05
<input type="radio"/>	fjk2-bucket-info	EE. UU. Este (Norte de Virginia) us-east-1	Bucket y objetos que no son públicos	24 Dec 2020 10:05:50 PM -05
<input type="radio"/>	fjk2-bucket-jsonvideo	EE. UU. Este (Norte de Virginia) us-east-1	Bucket y objetos que no son públicos	24 Dec 2020 10:06:05 PM -05
<input type="radio"/>	fjk2-bucket-video	EE. UU. Este (Norte de Virginia) us-east-1	 Público	28 Dec 2020 9:52:12 PM -05

Figure 7: Buckets creados

3 Indexación

Se consideran los siguientes buckets en S3:

- *bucket-info*

Archivos *id.json*, donde *id* es el ID del video; por ejemplo, *523.json*:

```
{
  "video": 523,
  "titulo": "Madagascar",
  "fecha": "23/12/2020 14:34",
  "duracion": "5:23"
}
```

- *bucket-img*

Archivos *id_minuto_segundo.jpg* que representan el frame tomado en ese minuto y segundo del video *id*. *523_0_0.jpg*

- *bucket-jsonvideo*

523.json:

```
{
  "video": 523,
  "frames":
  [
    { "minuto": 0, "segundo": 0 },
    { "minuto": 0, "segundo": 2 },
    { "minuto": 0, "segundo": 4 },
    { "minuto": 0, "segundo": 6 },
    { "minuto": 0, "segundo": 8 }
  ]
}
```

Archivos *id.json* que desencadenan la ejecución de la función lambda *indexarJsonVideo*, la cual detecta las etiquetas presentes y genera un archivo, en *bucket-index*, que servirá para indexar esta data en DynamoDB.

- *bucket-index*

```
[
  {
    "etiqueta": "Person",
    "valorVideo": "0000002,0000523",
    "tiempos": [
      {
        "minuto": 0,
        "segundo": 0
      }
    ]
  },
  {
    "etiqueta": "Wheel",
    "valorVideo": "0000004,0000523",
    "tiempos": [
      {
        "minuto": 0,
        "segundo": 0
      },
      {
        "minuto": 0,
        "segundo": 8
      }
    ]
  },
  .
  .
  .
]
```

El bucket contiene archivos *id.json* que indican las etiquetas presentes en el video *id*, el *valor* de estas para el video, y los *tiempos* representativos, de mayor a menor representatividad, de esta etiqueta en el video.

Estos archivos desencadenan la función lambda *agregarATablaDynamoDB* que indexa esta data, tal cual está, en la tabla *etiquetaVideo* de DynamoDB.

<input type="checkbox"/>	etiqueta ⓘ	valorVideo	tiempos
<input type="checkbox"/>	Bicycle	0000005,0002323	[{ "M" : { "minuto" : { "N" : "0" }, "segundo" : { "N" : "4" } } }, { "M" : { "minuto"...
<input type="checkbox"/>	Bicycle	0000004,0000523	[{ "M" : { "minuto" : { "N" : "0" }, "segundo" : { "N" : "0" } } }, { "M" : { "minuto"...
<input type="checkbox"/>	Bike	0000001,0002323	[{ "M" : { "minuto" : { "N" : "0" }, "segundo" : { "N" : "10" } } }]
<input type="checkbox"/>	Bike	0000001,0000523	[{ "M" : { "minuto" : { "N" : "0" }, "segundo" : { "N" : "8" } } }]

Figure 8: Tabla *etiquetaVideo* en DynamoDB. Notar que *etiqueta* en la Primary Key y el string *valorVideo* es el Sort Key.






De esta manera se podrá consultar sobre la tabla, y a partir del *valor* y

los *tiempos* se podrán obtener los videos más relevantes con los minutos y segundos más representativos del video según sea la etiqueta consultada.

3.1 Configuración


Funciones Lambda

Rol:

Nombre de la política ▾	
▶	 AmazonS3FullAccess
▶	 CloudWatchFullAccess
▶	 AmazonDynamoDBFullAccess
▶	 AmazonRekognitionFullAccess
▶	 AWSLambda_FullAccess

- indexar.JsonVideo

☐ Desencadenador

 **S3: fjk-bucket-jsonvideo** (Habilitado)
arn:aws:s3:::fjk-bucket-jsonvideo

☐ **Detalles**

Nombre de la notificación: **b91c08e2-f5d3-4502-bf97-a7cfff89eef**
Tipo de evento: **ObjectCreated**

Configuración básica
Info

Editar

Descripción

-

Memoria (MB)

128

Tiempo de espera


10 min 0 s

Figure 9: El timeout por defecto es de 3 segundos, lo cual es insuficiente. Se colocan 10 min porque pueden haber videos largos.

- agregarATablaDynamoDB

☐

Desencadenador



S3: fjk-bucket-index (Habilitado)

arn:aws:s3:::fjk-bucket-index

☐

▼ Detalles

Nombre de la notificación: **ddda85f0-589c-40a1-8324-67e48285642b**

Tipo de evento: **ObjectCreated**

Configuración básica
Info

Editar

Descripción

-

Memoria (MB)

128

Tiempo de espera

1 min 0 s

Rekognition y DynamoDB

Las funciones lambda las manejan y configuran. Solamente la eliminación de la tabla es manual.

4 Código

Se muestra en <https://github.com/cieloblanco/videoSearch>