



Cliente de almacenamiento de archivos en la nube utilizando S3 de AWS

Integrantes:

Angel Nolasco Serrano A01365726

Marco Antonio Almazán Martínez A01769046

Alberto Navarrete Ramirez A01422954

José Luis Hernández Hurtado A01365190

Myron Clarence Molina Bean A01369053

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS TOLUCA

*TC3003B.1: Desarrollo de aplicaciones avanzadas de ciencias computacionales
(Gpo 1)*

SEMESTRE FEBRERO-JUNIO 2023

Link del repo: <https://github.com/ANVRRT/Actividad3S3>

En esta entrega, se realizó un cliente y una función serverless alojada en lambda que permite el poder alojar archivos (en este caso imágenes) en un bucket de S3 en AWS.

El código proporcionado del repositorio en la carpeta "Client" se encuentra un script de Python que contiene dos funciones: `send_image` y `download_image`.

La función `send_image` toma una imagen en formato jpg y la convierte en una cadena codificada en base64. Luego, envía la imagen codificada en una solicitud POST a una URL específica para luego procesar la respuesta de la solicitud POST que se guarda en un archivo JSON con el nombre de la imagen.

```
def send_image(image_path, image_name):
    import requests
    import base64

    with open(image_path, "rb") as image_file:
        encoded_string = base64.b64encode(image_file.read()).decode('utf-8')

    payload = {
        "image": encoded_string,
        "image_name": image_name
    }

    url = "https://dv4d6d7smmnxnqqgxfncmdmk4m0gpsxz.lambda-url.us-east-1.on.aws/"
    response = requests.post(url, json=payload)

    with open(f"{image_name}.json", "w") as outfile:
        outfile.write(response.text)
```

La función `download_image` descarga una imagen de un bucket de Amazon S3. Esta función toma dos argumentos: la ruta remota de la imagen en el bucket de S3 y la ruta local donde se guardará la imagen descargada.

```
def download_image(remote_path, local_path):

    s3_manager = S3Manager()

    s3_manager.download_file_from_bucket("drimteam", remote_path, local_path)
```

El script importa los módulos `json`, `base64` y `S3Manager` de nuestro cliente administrador de S3.

En la sección principal del código, se llama a la función `send_image` con una imagen de prueba y se guarda la respuesta en un archivo JSON. Luego, se llama a la función `download_image` con la ruta remota de la imagen descargada del archivo JSON y la ruta local donde se guardará la imagen descargada.

```
if __name__ == "__main__":
    image_name = "imagetest"
    image_path = "./image.jpg"
    send_image(image_path, image_name)

    # Reads json with imagename
    with open(f"{image_name}.json", "r") as json_file:
        data = json.load(json_file)

    download_image(data["remote_path"], f"./downloaded_image.jpg")
```

El código proporcionado del repositorio en la carpeta “Lambda” es una función de AWS Lambda que carga una imagen en un bucket de Amazon S3 y devuelve la URL de la imagen cargada. La función se llama `lambda_handler` y se activa cuando se recibe una solicitud HTTP POST. La solicitud debe contener un cuerpo JSON que incluya la imagen codificada en base64 y el nombre de la imagen.

La función `upload_image` es la encargada de cargar la imagen en el bucket de S3. Primero, decodifica la imagen codificada en base64 y la guarda en un archivo temporal en el sistema de archivos local, luego, utiliza una instancia de la clase `S3Manager` para cargar el archivo en el bucket de S3 especificado para finalmente devolver un diccionario que contiene la ruta remota de la imagen cargada y la URL de la imagen.

```

def upload_image(image64, filename):
    try:
        s3_manager = S3Manager()

        bucket = "drimteam"
        filepath = f"dreamteam/{filename}.jpg"

        imageBytes = base64.b64decode(image64)

        with open(f"/tmp/{filename}.jpg", "wb") as img:
            img.write(imageBytes)

        s3_manager.upload_file_to_bucket(bucket, f"/tmp/{filename}.jpg", filepath)

        url = f"https://drimteam.s3.amazonaws.com/{filepath}"

        data = {
            "remote_path": filepath,
            "url": url
        }

        return data
    except:
        data = {
            "url": None
        }
        return data

```

La función `lambda_handler` extrae la imagen y el nombre de la imagen del cuerpo de la solicitud y llama a la función `upload_image` para cargar la imagen en S3. Luego, devuelve un diccionario que contiene el código de estado HTTP 200, el cuerpo de la respuesta que contiene la ruta remota de la imagen cargada y la URL de la imagen.

```
def lambda_handler(event, context):

    body = json.loads(event["body"])

    image = body["image"]
    image_name = body["image_name"]

    data = upload_image(image, image_name)

    return {
        "statusCode": 200,
        "body": data,
        "isBase64Encoded": False
    }
```

Con estos dos componentes, del cliente y de la función Lambda, es posible subir un archivo a un bucket de AWS. Y gracias a nuestro cliente manejador de S3, es posible subirlo desde el mismo cliente sin la necesidad de utilizar una función lambda, solo bastando conexión a internet y la imagen a la mano.