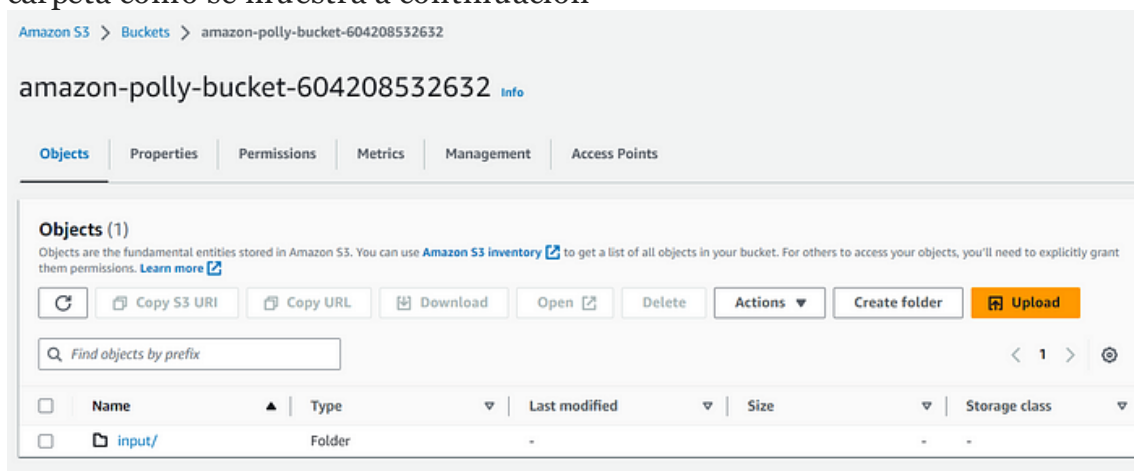


*En este caso de uso, veremos cómo convertir texto a voz usando Amazon Polly, AWS Lambda y S3 Bucket.*

## Creación de S3 Bucket para almacenar el archivo de texto de entrada

Vaya a la consola de AWS, busque S3. Cree un depósito S3 y cree una entrada de carpeta como se muestra a continuación



## Creación de función Lambda para crear el disparador

Vaya a la consola de AWS, busque Lambda. Cree una función de texto a voz y agregue el siguiente código Python

```
import json
import boto3
import logging
from botocore.exceptions import ClientError

logger = logging.getLogger()
logger.setLevel(logging.INFO)

s3 = boto3.client('s3')

def lambda_handler(event, context):
```

```

logger.info(event)

output_key = 'output/polly_response.json'
response = {}

for record in event['Records']:

    bucket = record['s3']['bucket']['name']
    key = record['s3']['object']['key']
    print(key)

    filename = key.split("/")[-1]
    print(filename)

    try:
        local_file_name = '/tmp/' + filename
        with open(local_file_name, 'wb') as data:
            s3.download_fileobj(bucket, key, data)
    except ClientError as e:
        if e.response['Error']['Code'] == "404":
            continue
        else:
            raise

    polly_client = boto3.client('polly')

    task_status = "null"
    line = ""

    voice_id='Aditi*'

    language_code = 'en-IN'

    try:
        with open(local_file_name, 'r') as file:
            line = file.read().replace("\n", " ")
            print(line)
        response = polly_client.start_speech_synthesis_task( #
            using start_speech_synthesis_task API
            Engine='neural',
            LanguageCode=language_code,
            OutputFormat='mp3',
            OutputS3BucketName=bucket,
            OutputS3KeyPrefix="output/"+filename,
            Text=line,
            TextType='text',
            VoiceId=voice_id
        )

        taskid = response['SynthesisTask']['TaskId']
        task_status = response['SynthesisTask']['TaskStatus']
        output_filename = filename + "." + taskid + ".mp3"

        return_result = {
            "FileName":output_filename, "TaskStatus":task_status}

```

```

except Exception as error:
    print(error)
    return_result = {"Status": "Failed", "Reason": error}

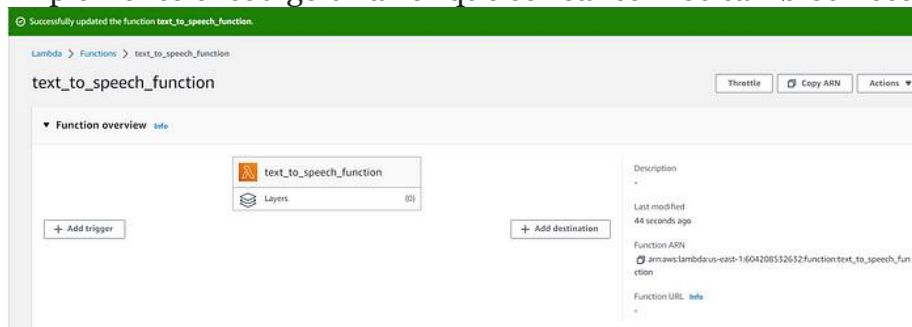
# Save response in S3 bucket
s3.put_object(
    Bucket=bucket,
    Key=output_key,
    Body=json.dumps(response, default=str, indent=4)
)

return return_result

```


En el código anterior, debemos ingresar el código de idioma y la identificación de voz según [la documentación](#).

Implemente el código una vez que se realicen los cambios necesarios.



Haga clic en Agregar activador, agregue el S3 Bucket y la carpeta de entrada que se ha creado.

**Trigger configuration** [Info](#)

 **S3**  
aws storage

**Bucket**  
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

× ↻

Bucket region: us-east-1

**Event types**  
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

PUT ×

**Prefix - optional**  
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

**Suffix - optional**  
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

**Recursive invocation**  
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

☒ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.



Recibiremos la siguiente notificación una vez que el activador se haya configurado correctamente


Lambda > Functions > text\_to\_speech\_function

**text\_to\_speech\_function** Throttle Copy ARN Actions

✓ The trigger amazon-polly-bucket-604208532632 was successfully added to function text\_to\_speech\_function. The function is now receiving events from the trigger.

**Function overview** [Info](#)


 **text\_to\_speech\_function**  
 Layers (0)

 **S3**  
+ Add trigger

+ Add destination

**Description**  
+

**Last modified**  
5 minutes ago

**Function ARN**  
 arn:aws:lambda:us-east-1:604208532632:function:text\_to\_speech\_function

**Function URL** [Info](#)  
-

## Demostración práctica

Con esto, veremos la demostración de cómo este archivo de texto que estamos cargando en la carpeta de entrada s3 se convierte a voz (formato mp3) en la carpeta de salida del depósito s3.