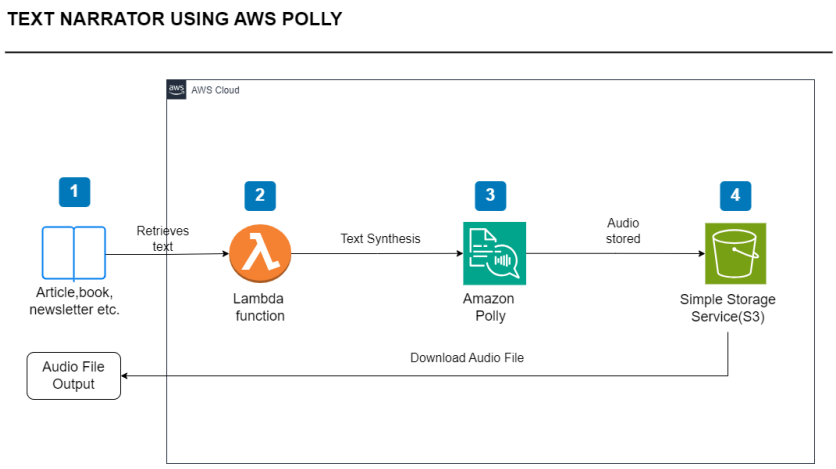
Develop a text narrator using Amazon Polly

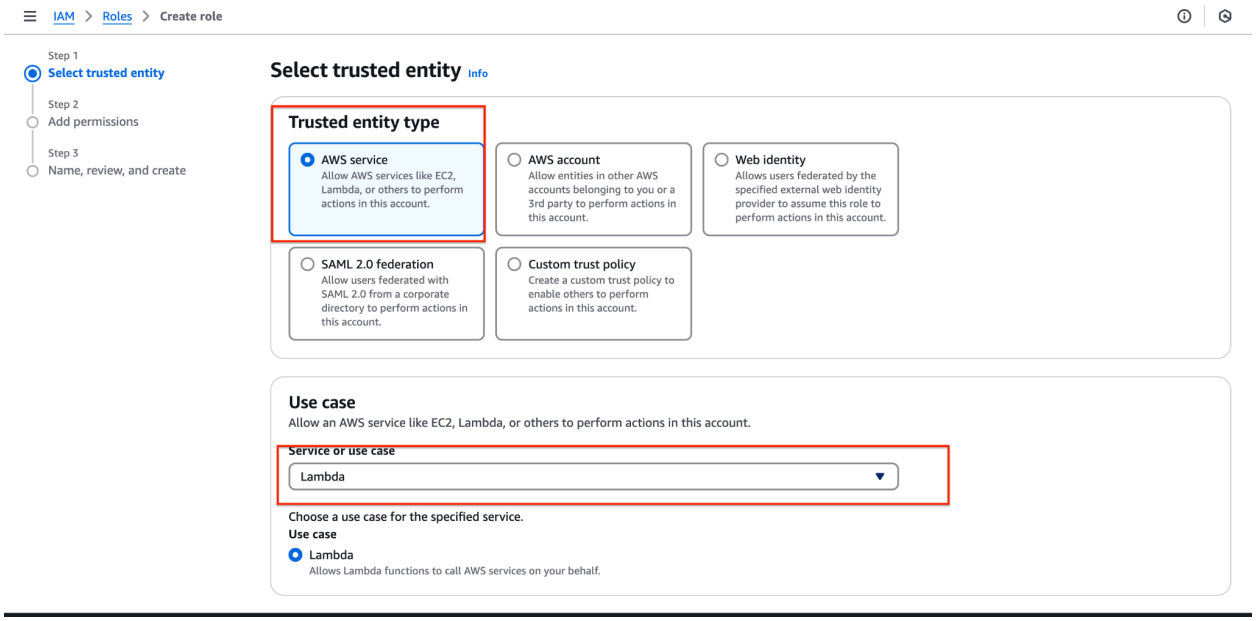
Here is the architectural plan of my application:

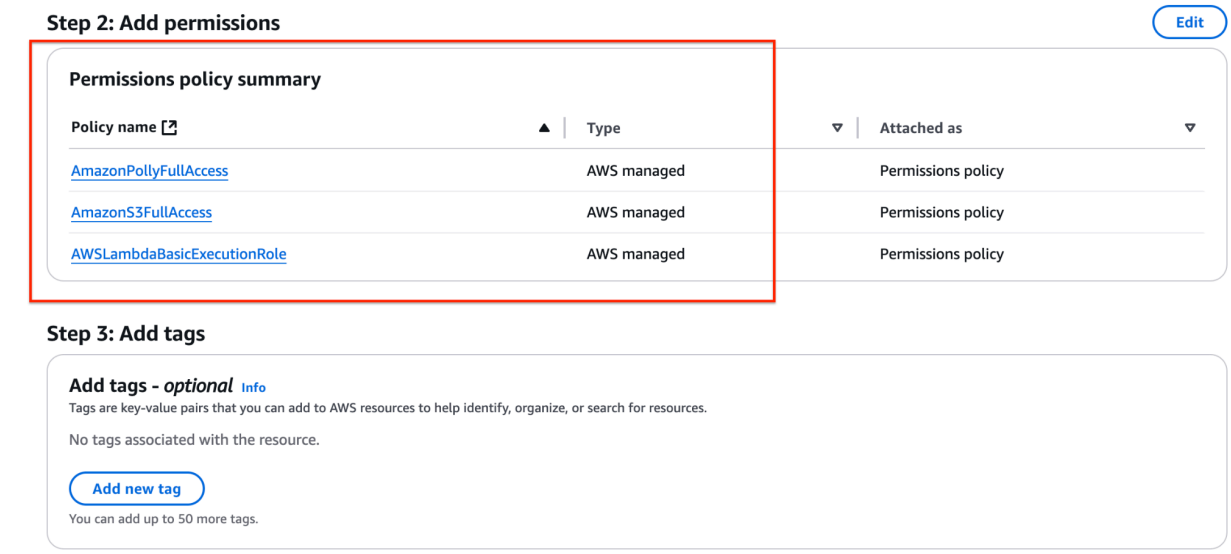


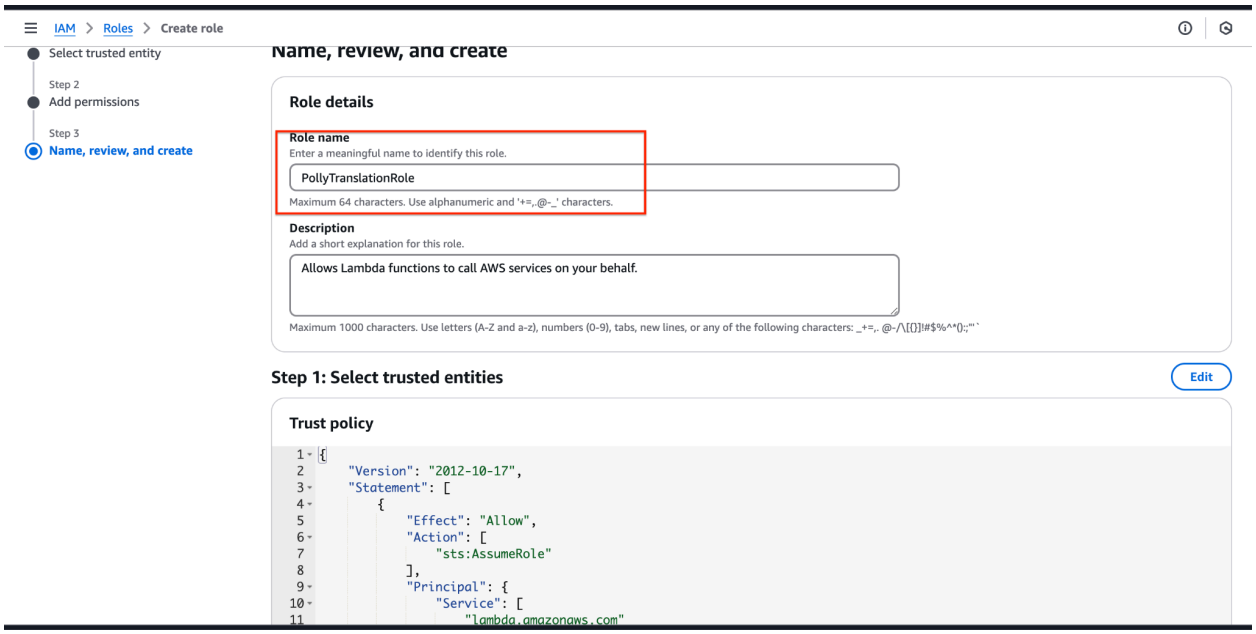
STEP 1

Creating an IAM role

I created an IAM role with suitable policies attached to it in order to access to POLLY service and store the audio output in a S3 bucket using a Lambda function.



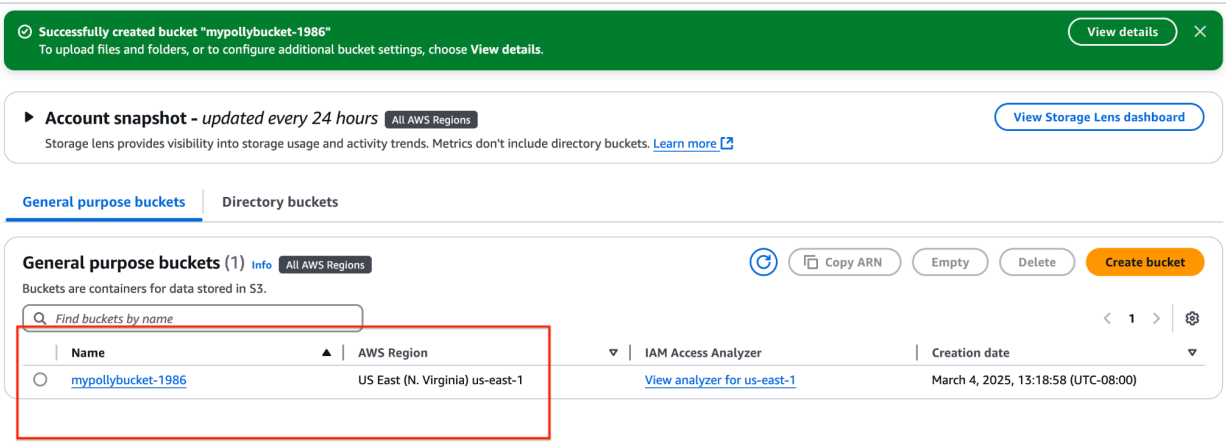




The role has been created

STEP 2

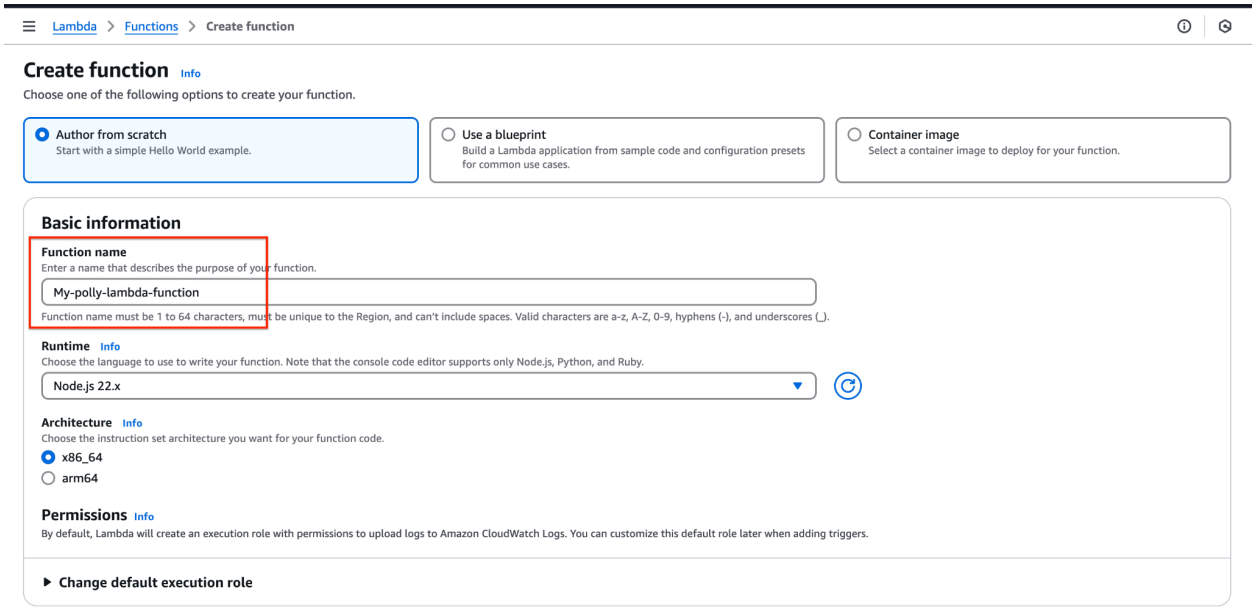
Creating a S3 bucket

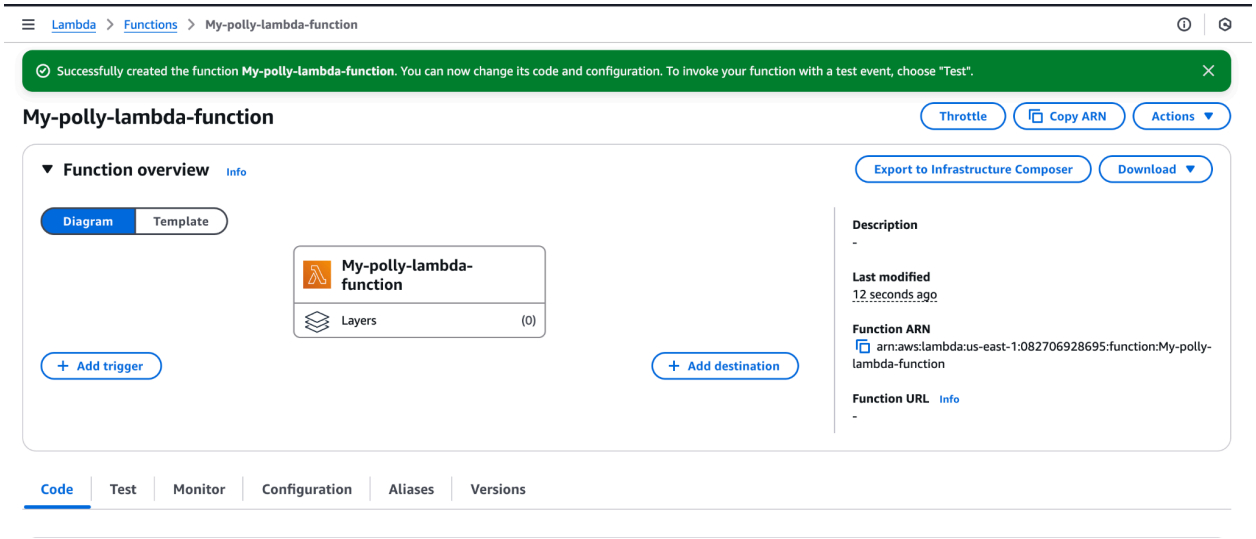


The Bucket has been created

STEP 3

Creating a Lambda function





My lambda function has been created, while i was creating the lambda function i added the role that we created so lambda can access to S3 and invoke Polly.

STEP 4

Writing the Code

Before i wrote the code i had to modify the index file name from **index.mjs'** file to **'index.js'**

In order for my lambda function to communicate with other service like (POLLY and S3) i had to use AWS SDK.

So i had to include it in my lambda code:

```
const AWS = require('aws-sdk');
const polly = new AWS.Polly();
const s3 = new AWS.S3();
```

I wrote a function that AWS runs for me whenever something happens. It's like a little program that waits for a signal to start working:

```
exports.handler = async (event) => {
```

The Objective was, when a message is received by the lambda function it has to turn it into speech. I was able to choose how the speech will sound and in which format:

```
const text = event.text;
const params = {
    Text: text,
    OutputFormat: 'mp3',
    VoiceId: 'Joanna' // You can change this to the desired voice
};
```

I sent the text to Polly and asked it to turn it into speech. Polly did its magic and gave me back the speech as data.
I then saved this speech in my S3 storage:

```
const data = await polly.synthesizeSpeech(params).promise();
const key = `audio-${Date.now()}.mp3`;
const s3Params = {
    Bucket: 'your-bucket-name', // Replace with your S3 bucket name
    Key: key,
    Body: data.AudioStream,
    ContentType: 'audio/mpeg'
};
await s3.upload(s3Params).promise();
```

I made a message saying the speech has been saved successfully with its special name in my storage. If something goes wrong, there is an error message.

```
const outputMessage = `The audio file has been successfully stored in the S3 bucket by the name ${key}`;
return {
    statusCode: 200,
    body: JSON.stringify({ message: outputMessage })
};
} catch (error) {
    console.error('Error:', error);
    return {
        statusCode: 500,
        body: JSON.stringify({ message: 'Internal server error' })
    };
}
```

STEP 5

Deployin the Code

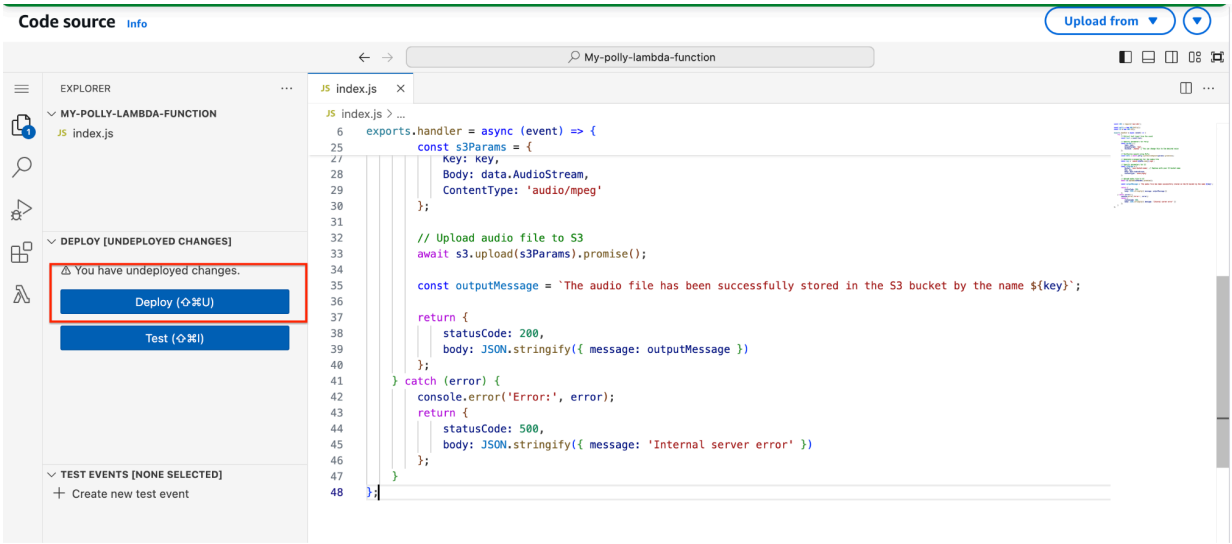
Here is the complete Lambda function code:

```
const AWS = require('aws-sdk');
const polly = new AWS.Polly();
const s3 = new AWS.S3();
exports.handler = async (event) => {
    try {
        // Extract text input from the event
        const text = event.text;

        // Specify parameters for Polly
        const params = {
            Text: text,
            OutputFormat: 'mp3',
            VoiceId: 'Joanna' // You can change this to the desired voice
        };
        // Synthesize speech using Polly
        const data = await polly.synthesizeSpeech(params).promise();
        // Generate a unique key for the audio file
        const key = `audio-${Date.now()}.mp3`;
        // Specify parameters for S3
        const s3Params = {
            Bucket: 'your-bucket-name', // Replace with your S3 bucket name
            Key: key,
            Body: data.AudioStream,
            ContentType: 'audio/mpeg'
        };
        // Upload audio file to S3
        await s3.upload(s3Params).promise();
        const outputMessage = `The audio file has been successfully stored in the S3 bucket by the
name ${key}`;
        return {
            statusCode: 200,
            body: JSON.stringify({ message: outputMessage })
        };
    } catch (error) {
        console.error('Error:', error);
        return {
            statusCode: 500,
            body: JSON.stringify({ message: 'Internal server error' })
        };
    }
```

```
};
```

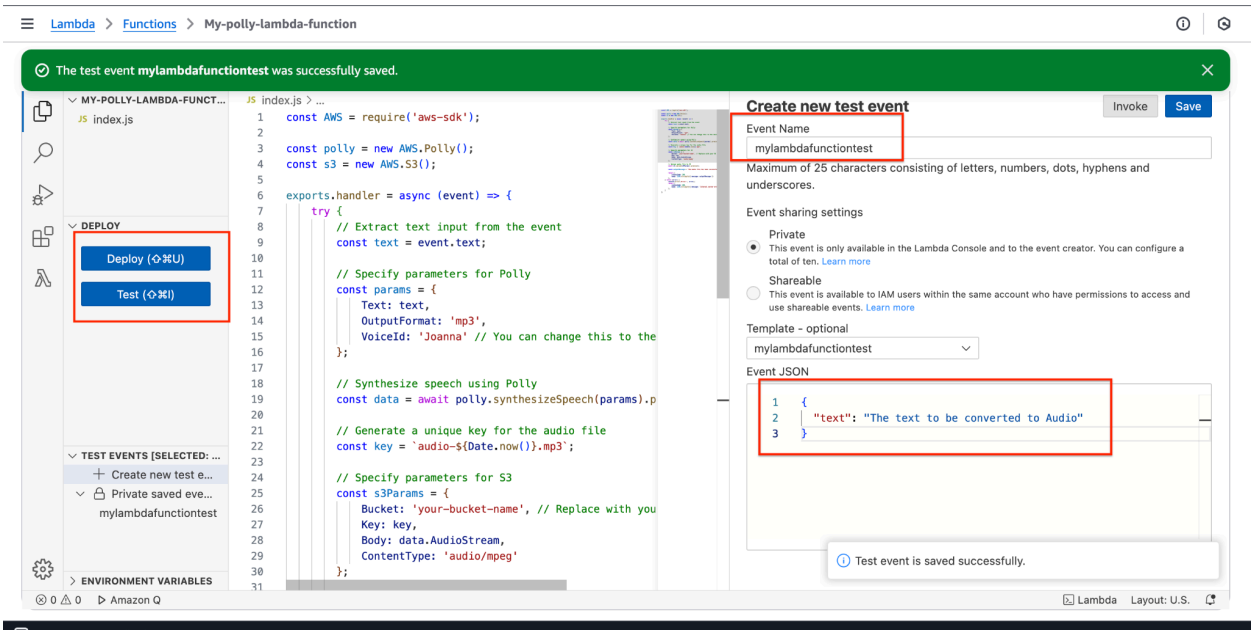
I deployed the code changes by clicking on Deploy



STEP 6

Checking the output

i tested and configured an event for my lambda function



I provided a name for my test configuration and in the Event JSON, I provided the text to be converted to audio.

"text":"The text to be converted to Audio"