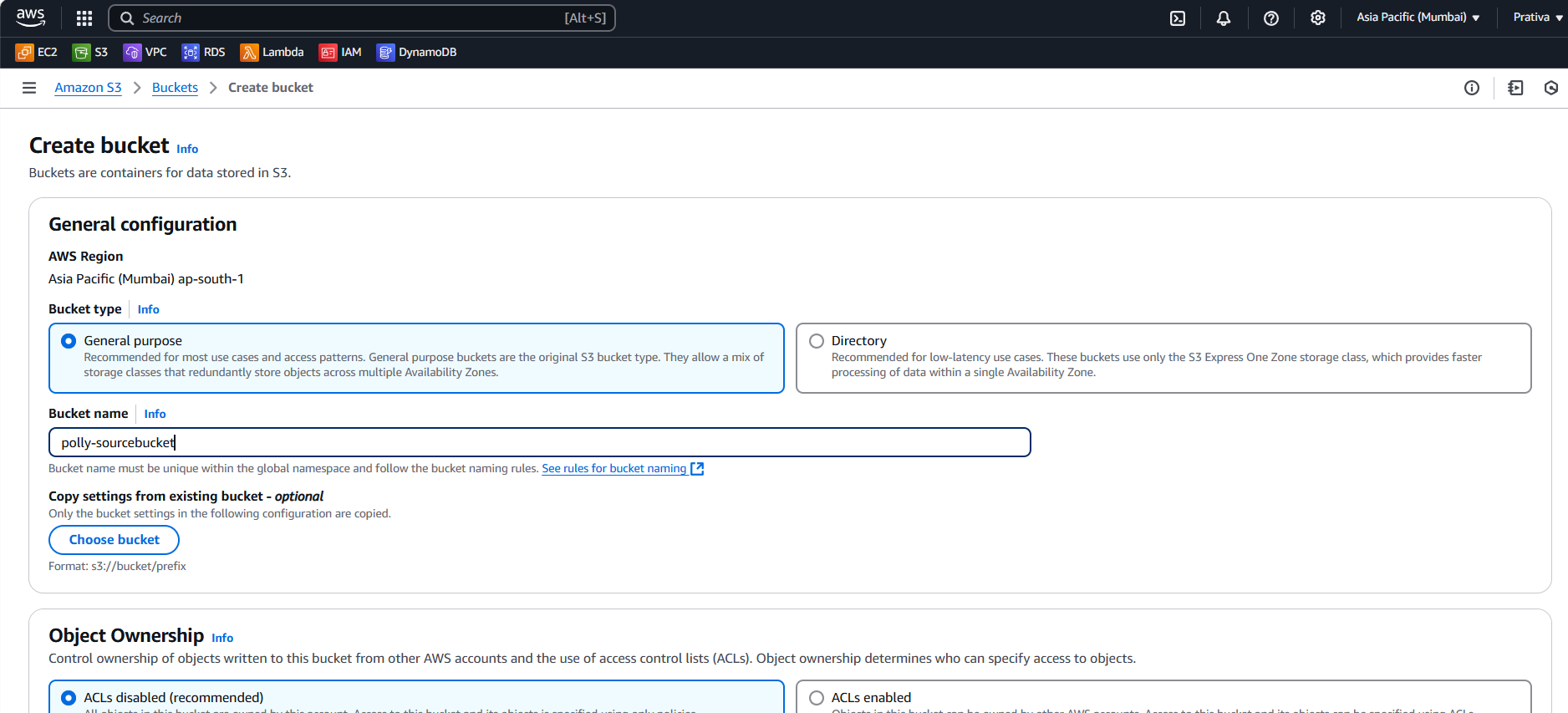
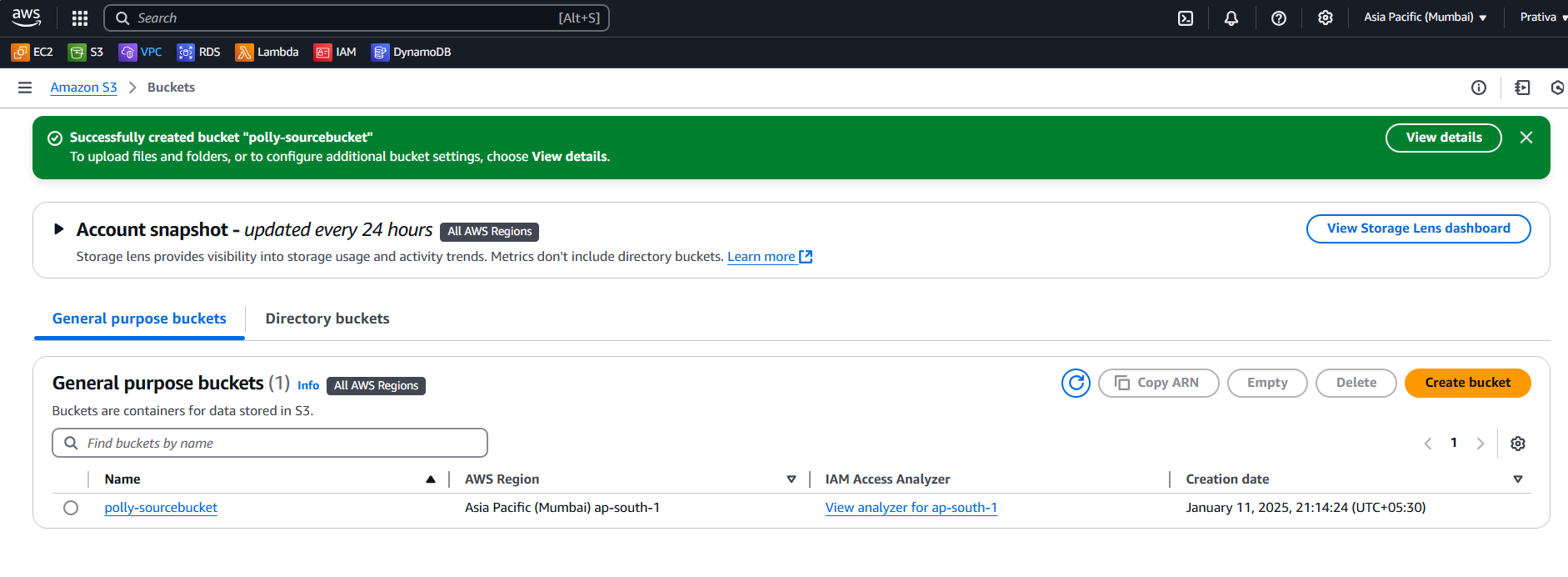
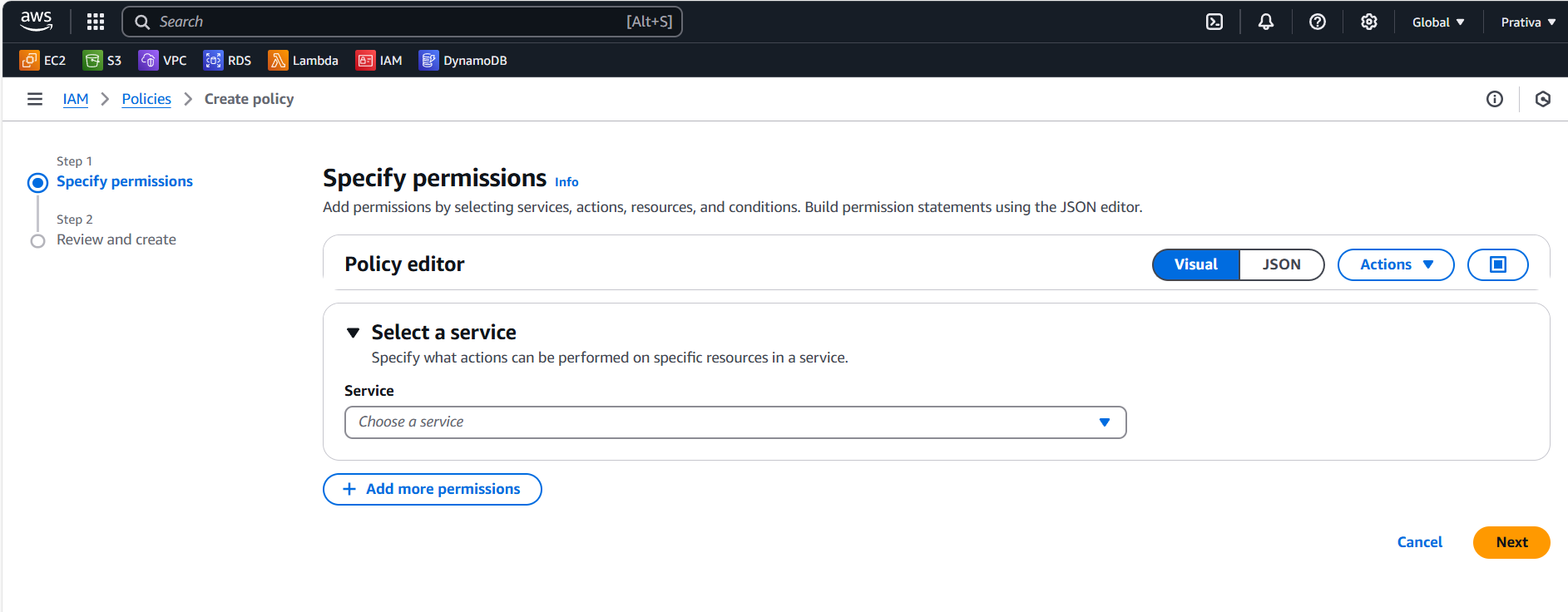
**Text narrator web app using amazon polly**

Create a bucket (name: polly-sourcebucket) with as it is settings.





Create IAM Policy (name: Lambda\_Polly\_S3\_IAMPolicy)



Select the JSON and paste below code

{

    "Version": "2012-10-17",

    "Statement": [

        {

            "Effect": "Allow",

            "Action": [

                "s3:GetObject"

            ],

            "Resource": [

                "arn:aws:s3:::YOUR\_BUCKETNAME/\*"

            ]

        },

        {

            "Effect": "Allow",

            "Action": [

                "polly:SynthesizeSpeech"

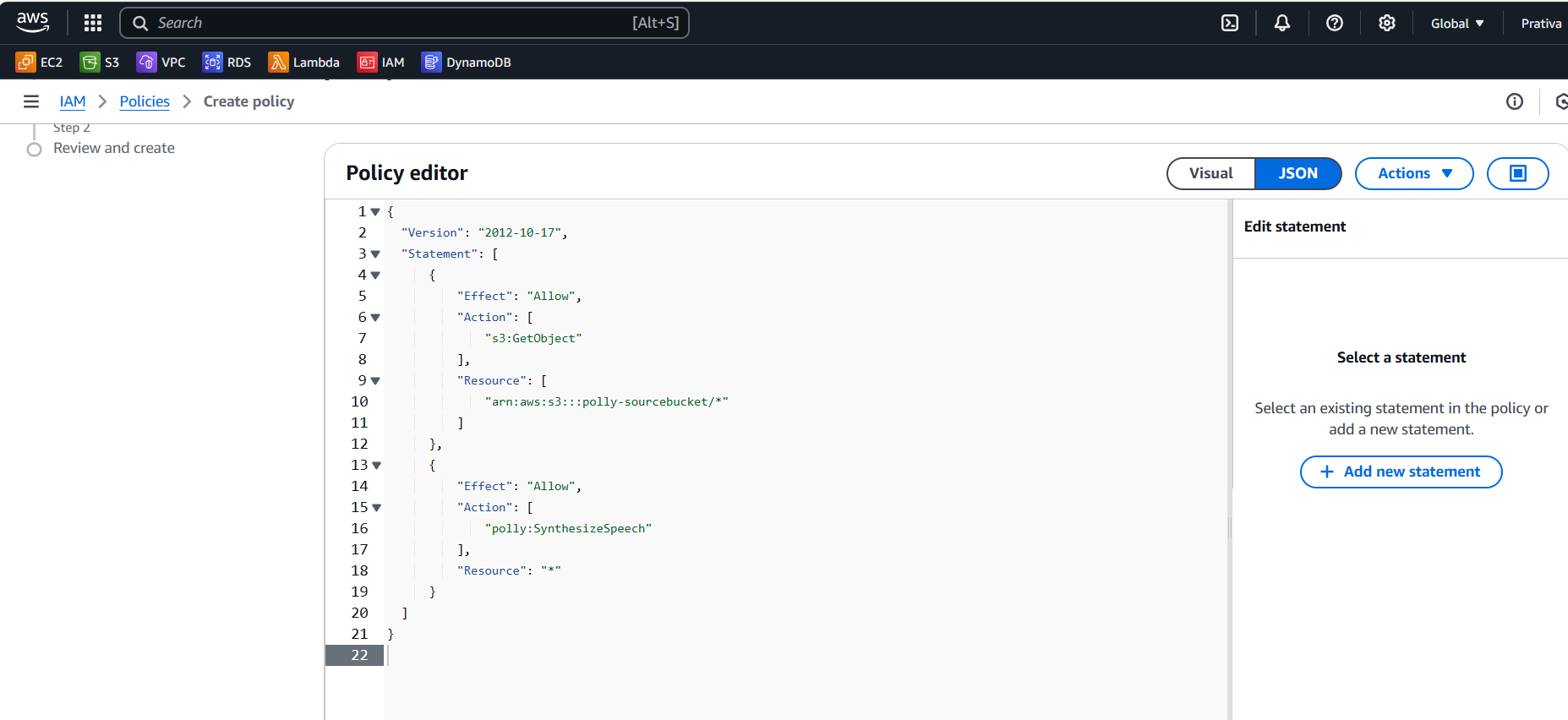
            ],

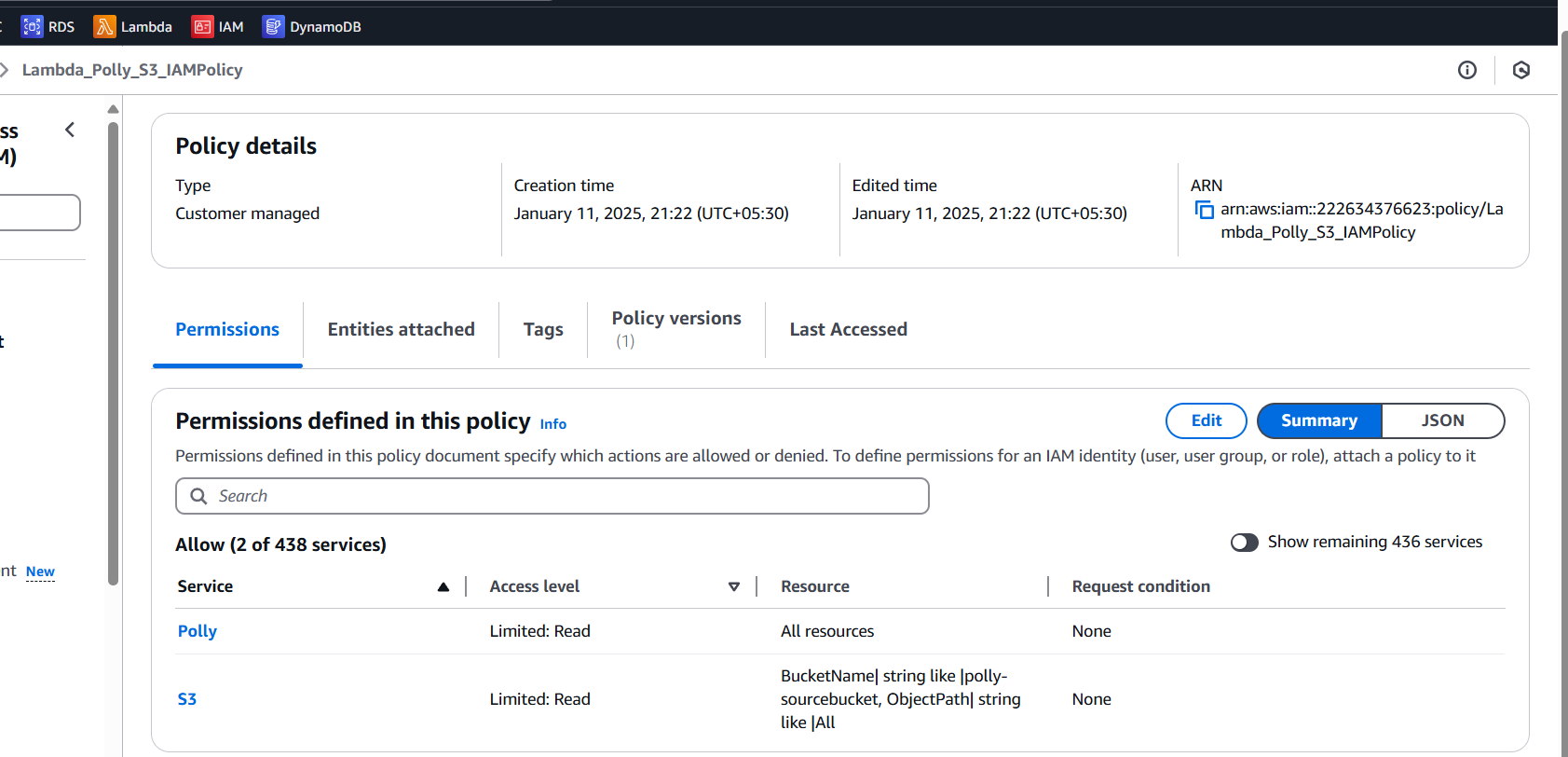
            "Resource": "\*"

        }

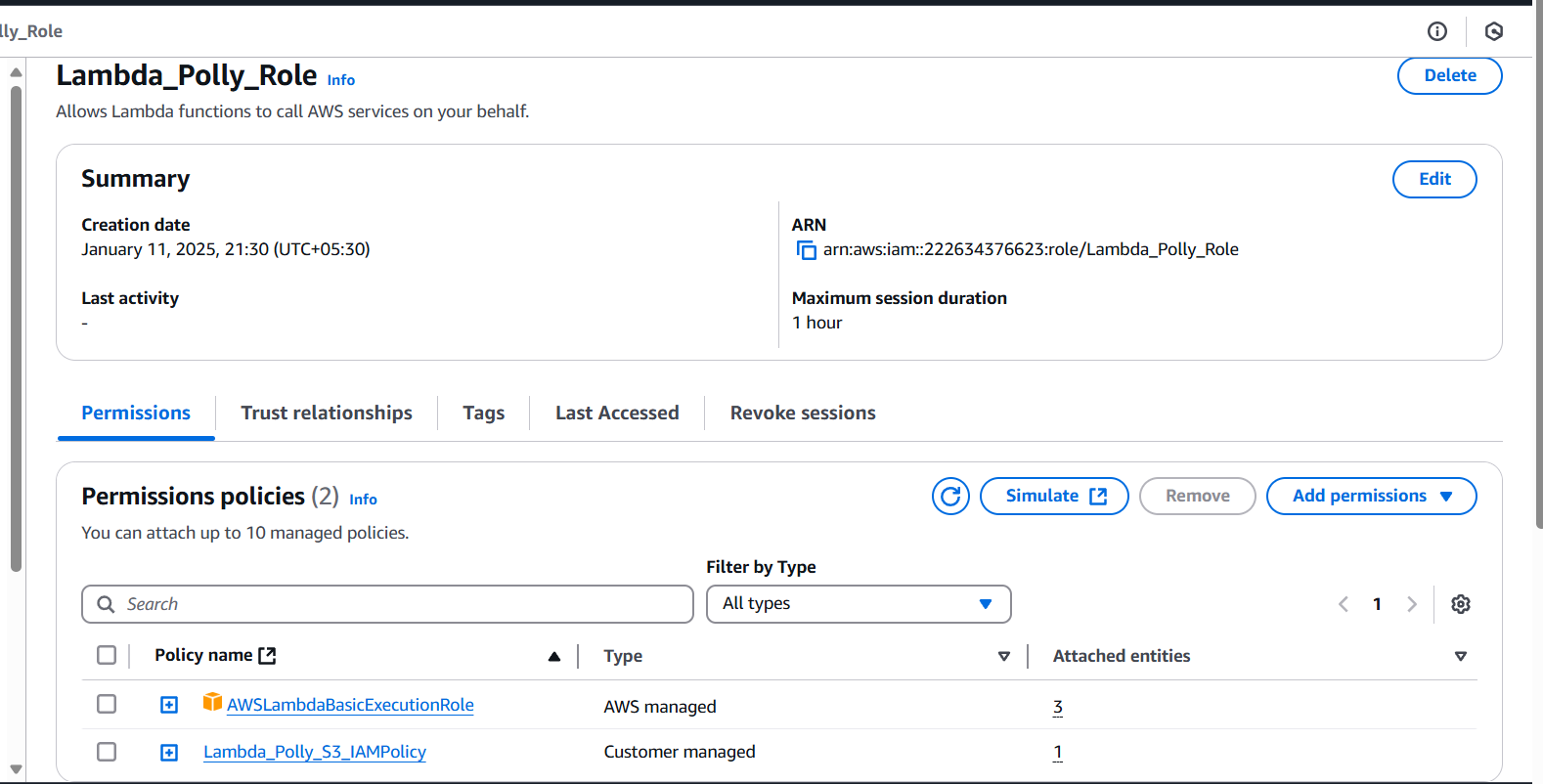
    ]

  }





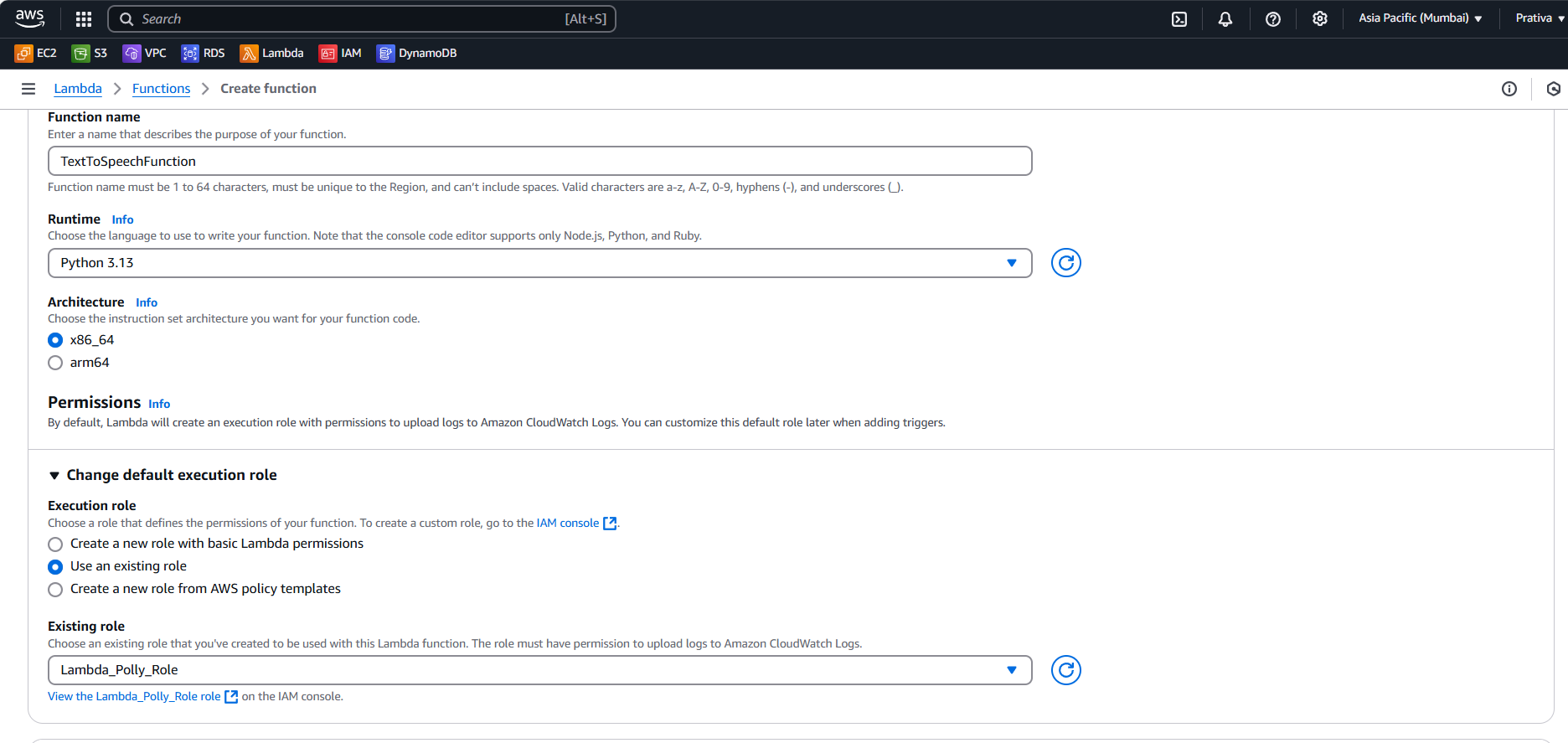
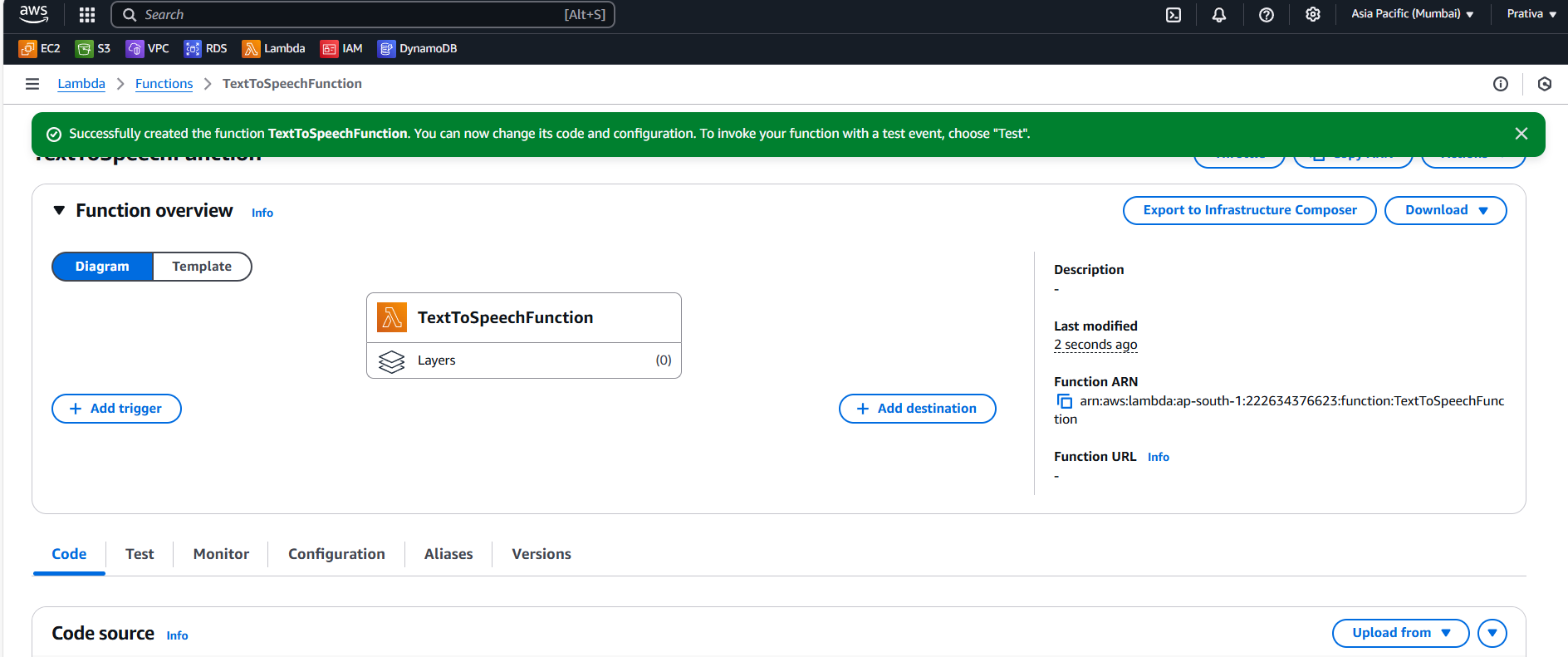
Create an IAM Role (name: Lambda\_Polly\_Role) and attach Lambda\_Polly\_S3\_IAMPolicy and AWSLambdaBasicExecutionRole Policies



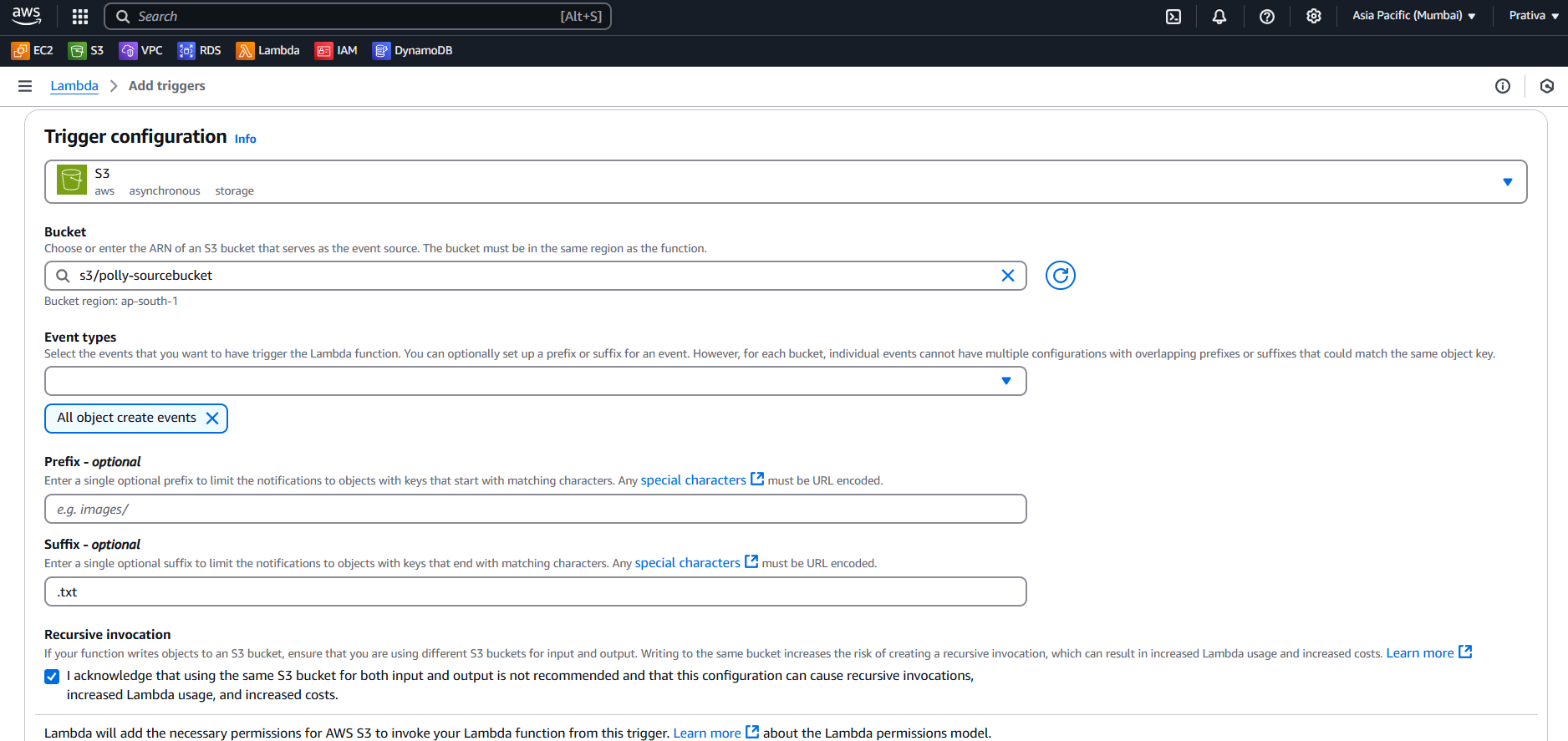
Once Policy, Role and bucket are created, its time for Lambda function creation

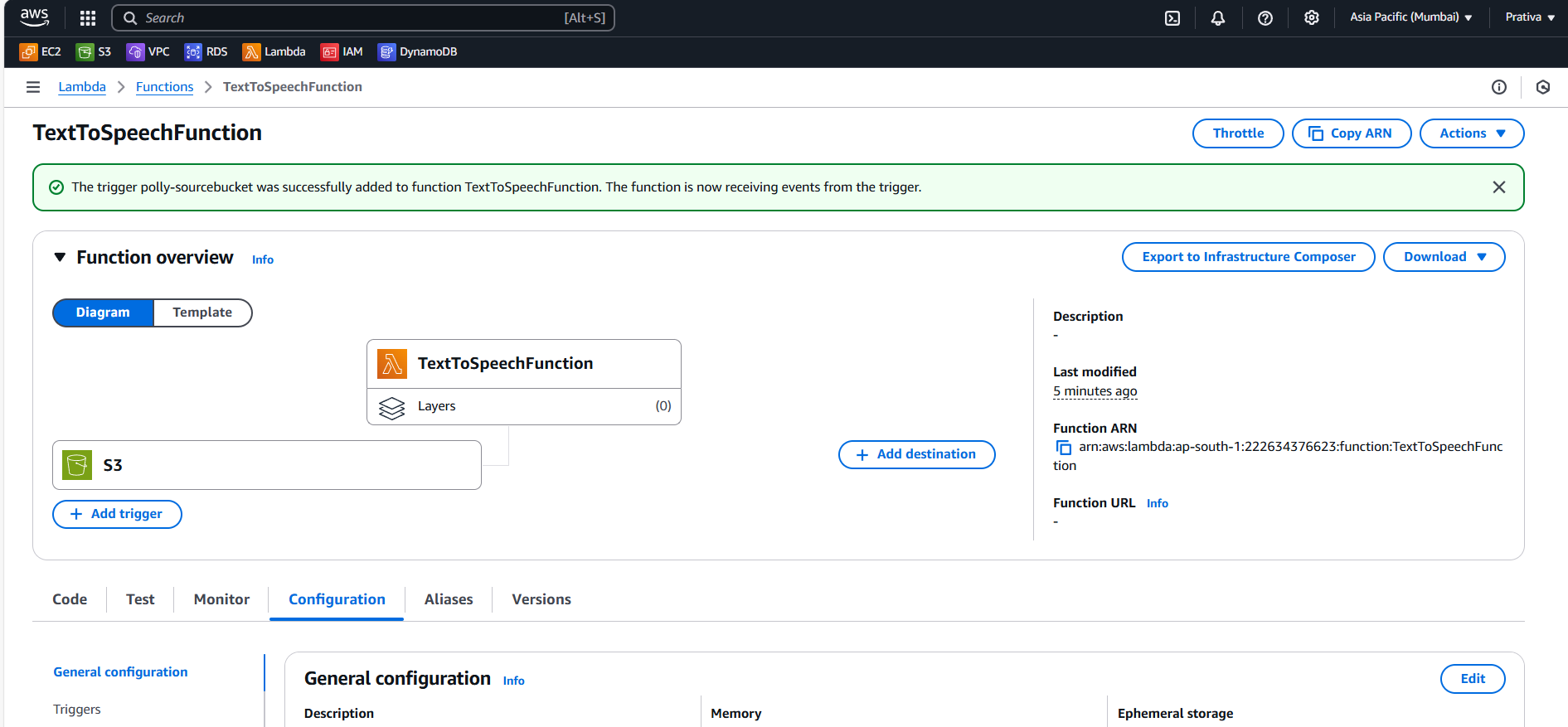
Create and Configure the Lambda Function (name: TextToSpeechFunction)

Set the runtime to Python(latestversion),the execution role(Lambda\_Polly\_Role)

Click on Add Trigger to set up an event notification in the source S3 bucket to trigger the Lambda function on new object creation events with the .txt suffix and acknowledge it.





Go to S3 bucket🡪 Permissions🡪update bucket policy

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": {

"AWS": "arn:aws:iam::accountid:role/Lambda\_Polly\_Role"

},

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::polly-sourcebucket/\*"

}

}

Go to IDE(VSCode), and create your logic to test the flow

Create Frontent.py file which is a Flask web application and paste below code

import os

import boto3

from flask import Flask, render\_template, request, send\_file, flash, redirect

from werkzeug.utils import secure\_filename

import io

import json

import base64

# Initialize Flask app

app = Flask(\_\_name\_\_)

# Set a secret key for session management (should be kept secret)

app.secret\_key = os.urandom(24)  # Or set a custom key if needed

# AWS Config

s3\_client = boto3.client('s3', region\_name='ap-south-1')

bucket\_name = 'polly-sourcebucket'

lambda\_client = boto3.client('lambda', region\_name='ap-south-1')

lambda\_name = 'TextToSpeechFunction'

ALLOWED\_EXTENSIONS = {'txt'}

def allowed\_file(filename):

    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

@app.route('/')

def index():

    return render\_template('index.html')

@app.route('/upload', methods=['POST'])

def upload\_file():

    if 'file' not in request.files:

        flash('No file part', 'error')

        print("No file part")

        return redirect('/')

    file = request.files['file']

    print(f"File received: {file.filename}")

    if file.filename == '':

        flash('No selected file', 'error')

        print("No selected file")

        return redirect('/')

    if file and allowed\_file(file.filename):

        try:

            filename = secure\_filename(file.filename)

            file\_content = file.read().decode('utf-8')  # Read content as a string

            # Reset the file pointer to the beginning

            file.seek(0)

            # Upload the file to S3

            s3\_client.upload\_fileobj(file, bucket\_name, filename)

            print(f"File uploaded to S3: {filename} with content: {file\_content} ")

            # Trigger Lambda function

            response = lambda\_client.invoke(

                FunctionName=lambda\_name,  # Replace with your Lambda function name

                InvocationType='RequestResponse',

                Payload=json.dumps({'bucket': bucket\_name, 'filename': filename})

            )

            print('response: ',response)

            # Get the audio content from Lambda response

            response\_payload = json.loads(response['Payload'].read())

            print(f"Lambda response: {response\_payload}")

            if response\_payload.get('statusCode') == 200:

                # Decode the base64-encoded audio

                audio\_content\_base64 = response\_payload['body']

                audio\_content = base64.b64decode(audio\_content\_base64)

                # Return the audio file to the user for download

                return send\_file(io.BytesIO(audio\_content),

                                 mimetype='audio/mpeg',

                                 as\_attachment=True,

                                 download\_name=filename.replace('.txt', '.mp3'))

            else:

                flash(f"Error: {response\_payload.get('body')}", 'error')

                print(f"Error: {response\_payload.get('body')}")

                return redirect('/')

        except Exception as e:

            flash(f'Error processing the file: {str(e)}', 'error')

            print(f'Error processing the file: {str(e)}')

            return redirect('/')

    flash('Invalid file type. Please upload a .txt file.', 'error')

    print('Invalid file type. Please upload a .txt file.')

    return redirect('/')

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

Create Lambdafunction.py file which will have lambda function logic which invokes polly to generate audio from text . Paste below code

import os

import boto3

from flask import Flask, render\_template, request, send\_file, flash, redirect

from werkzeug.utils import secure\_filename

import io

import json

import base64

# Initialize Flask app

app = Flask(\_\_name\_\_)

# Set a secret key for session management

app.secret\_key = os.urandom(24)

# AWS Config

s3\_client = boto3.client('s3', region\_name=ap-south-1)

bucket\_name = 'polly-sourcebucket'

lambda\_client = boto3.client('lambda', region\_name='ap-south-1'

lambda\_name = 'TextToSpeechFunction'

ALLOWED\_EXTENSIONS = {'txt'}

def allowed\_file(filename):

    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

@app.route('/')

def index():

    return render\_template('index.html')

@app.route('/upload', methods=['POST'])

def upload\_file():

    if 'file' not in request.files:

        flash('No file part', 'error')

        return redirect('/')

    file = request.files['file']

    if file.filename == '':

        flash('No selected file', 'error')

        return redirect('/')

    if file and allowed\_file(file.filename):

        try:

            filename = secure\_filename(file.filename)

            # Upload the file to S3

            s3\_client.upload\_fileobj(file, bucket\_name, filename)

            # Trigger Lambda function

            response = lambda\_client.invoke(

                FunctionName=lambda\_name,

                InvocationType='RequestResponse',

                Payload=json.dumps({'bucket': bucket\_name, 'filename': filename})

            )

            # Get the response from Lambda

            response\_payload = json.loads(response['Payload'].read().decode())

            if response\_payload.get('statusCode') == 200:

                # Decode the base64-encoded audio

                audio\_content\_base64 = response\_payload['body']

                audio\_content = base64.b64decode(audio\_content\_base64)

                # Return the audio file to the user for download

                return send\_file(io.BytesIO(audio\_content),

                                 mimetype='audio/mpeg',

                                 as\_attachment=True,

                                 download\_name=filename.replace('.txt', '.mp3'))

            else:

                flash(f"Error: {response\_payload.get('body')}", 'error')

                return redirect('/')

        except Exception as e:

            flash(f'Error processing the file: {str(e)}', 'error')

            return redirect('/')

    flash('Invalid file type. Please upload a .txt file.', 'error')

    return redirect('/')

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

Create index.html file under templates folder of your project for UI design and paste below code

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Upload Text and Get Audio</title>

    <style>

        /\* General Reset \*/

        body, h1, form {

            margin: 0;

            padding: 0;

        }

        /\* Body Styling \*/

        body {

            font-family: Arial, sans-serif;

            background: #f4f4f9;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

        }

        /\* Container Styling \*/

        .container {

            text-align: center;

            background: #ffffff;

            border-radius: 8px;

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

            padding: 20px;

            width: 400px;

        }

        /\* Header Styling \*/

        h1 {

            color: #333333;

            font-size: 24px;

            margin-bottom: 20px;

        }

        /\* Form Styling \*/

        form {

            display: flex;

            flex-direction: column;

            align-items: center;

            gap: 15px;

        }

        input[type="file"] {

            padding: 10px;

            border: 1px solid #ccc;

            border-radius: 4px;

            width: 100%;

            cursor: pointer;

        }

        /\* Button Styling \*/

        button {

            background: #007bff;

            color: white;

            border: none;

            border-radius: 4px;

            padding: 10px 15px;

            font-size: 16px;

            cursor: pointer;

            width: 100%;

            transition: background 0.3s ease;

        }

        button:hover {

            background: #0056b3;

        }

        /\* Footer Link \*/

        .footer {

            margin-top: 15px;

            font-size: 12px;

            color: #777;

        }

        .footer a {

            color: #007bff;

            text-decoration: none;

        }

        .footer a:hover {

            text-decoration: underline;

        }

    </style>

</head>

<body>

    <div class="container">

        <h1>Upload a Text File</h1>

        <form action="/upload" method="POST" enctype="multipart/form-data">

            <input type="file" name="file" accept=".txt" required>

            <button type="submit">Upload</button>

        </form>

        <div class="footer">

            <p>Need help? <a href="mailto:prativatest@example.com">Contact Support</a></p>

        </div>

    </div>

</body>

</html>

<!-- <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Upload Text and Get Audio</title>

</head>

<body>

    <h1>Upload a Text File</h1>

    <form action="/upload" method="POST" enctype="multipart/form-data">

        <input type="file" name="file" required>

        <button type="submit">Upload</button>

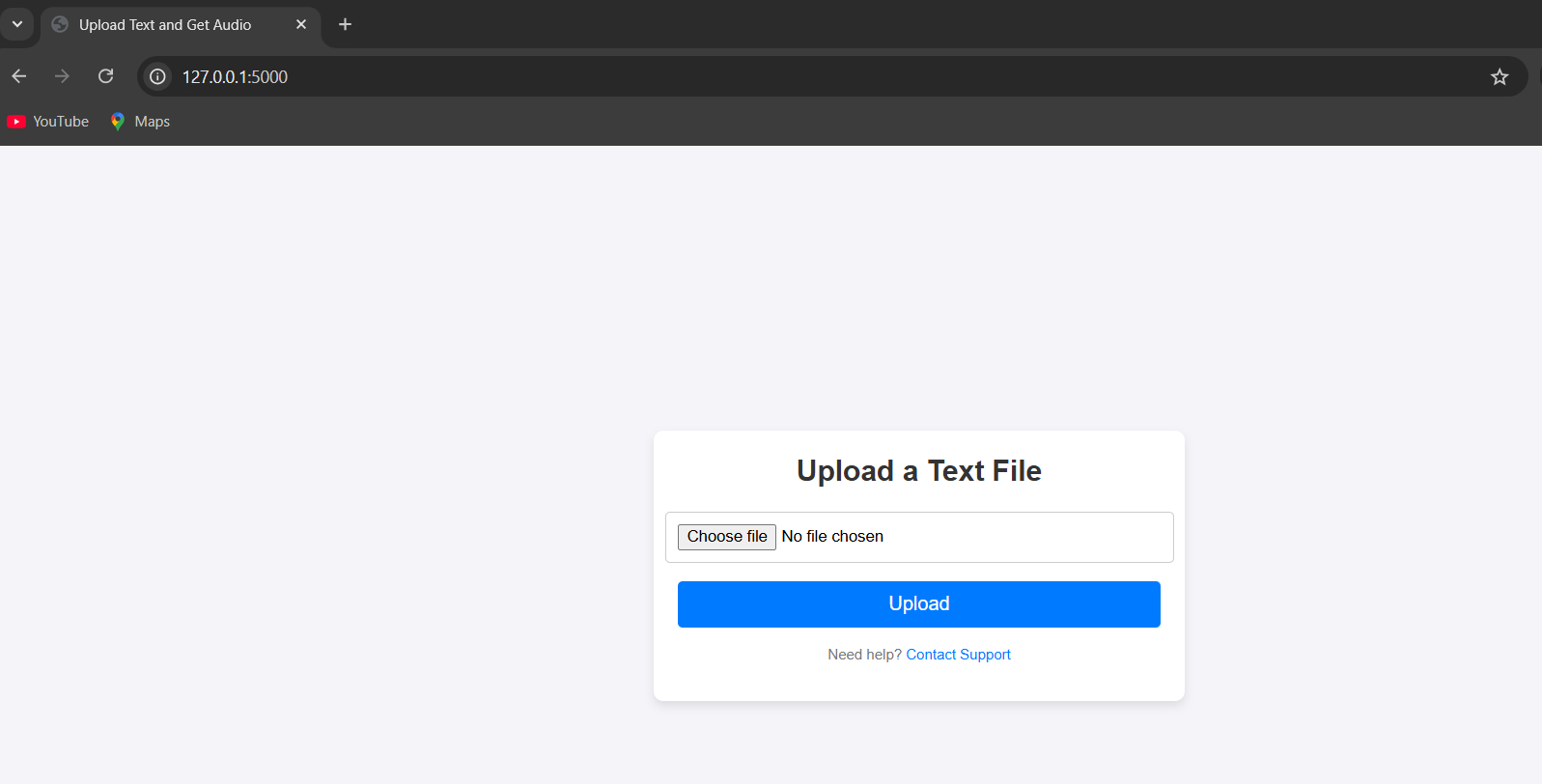
    </form>

</body>

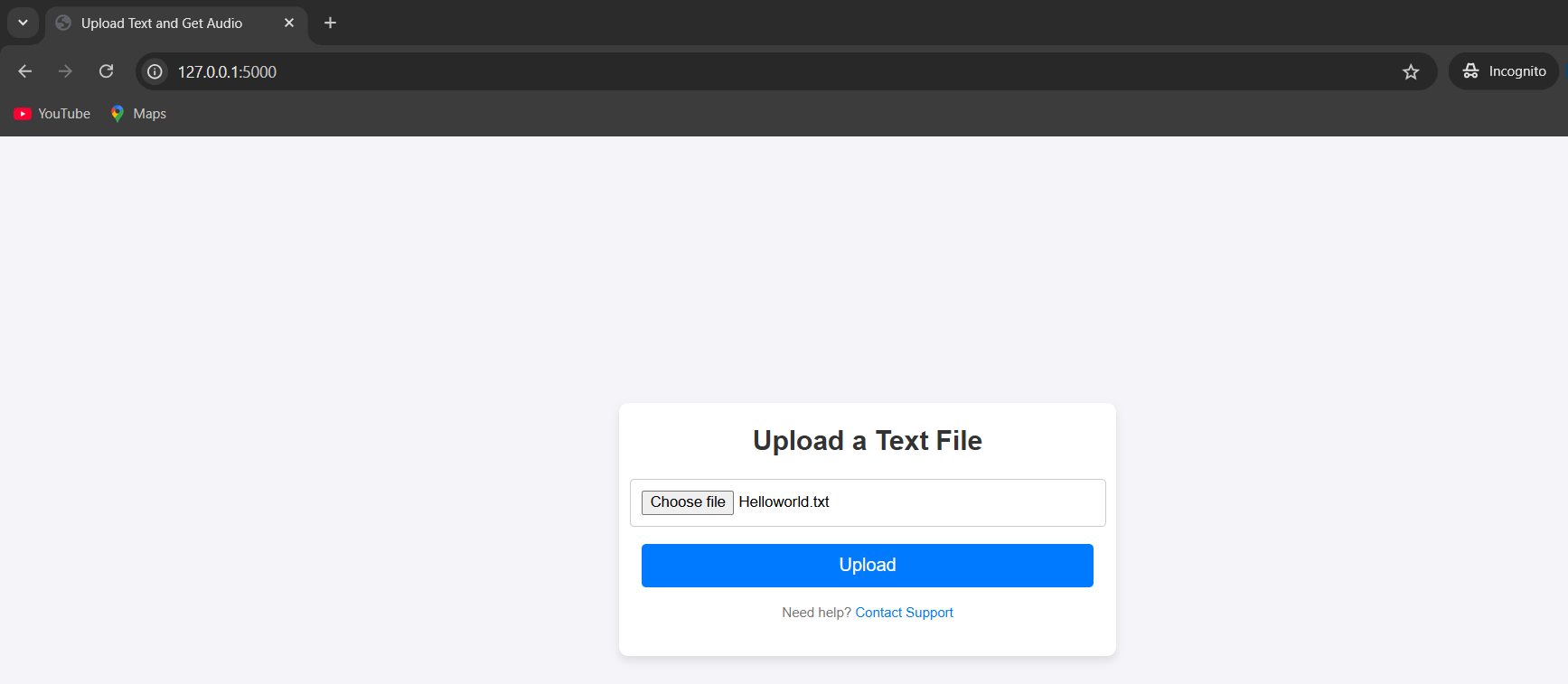
</html> -->

Test the Application

Run python Frontend.py and hit the url in browser



Choose file from your local



Once you click on upload, audio(mp3) file will be downloaded to your local system

