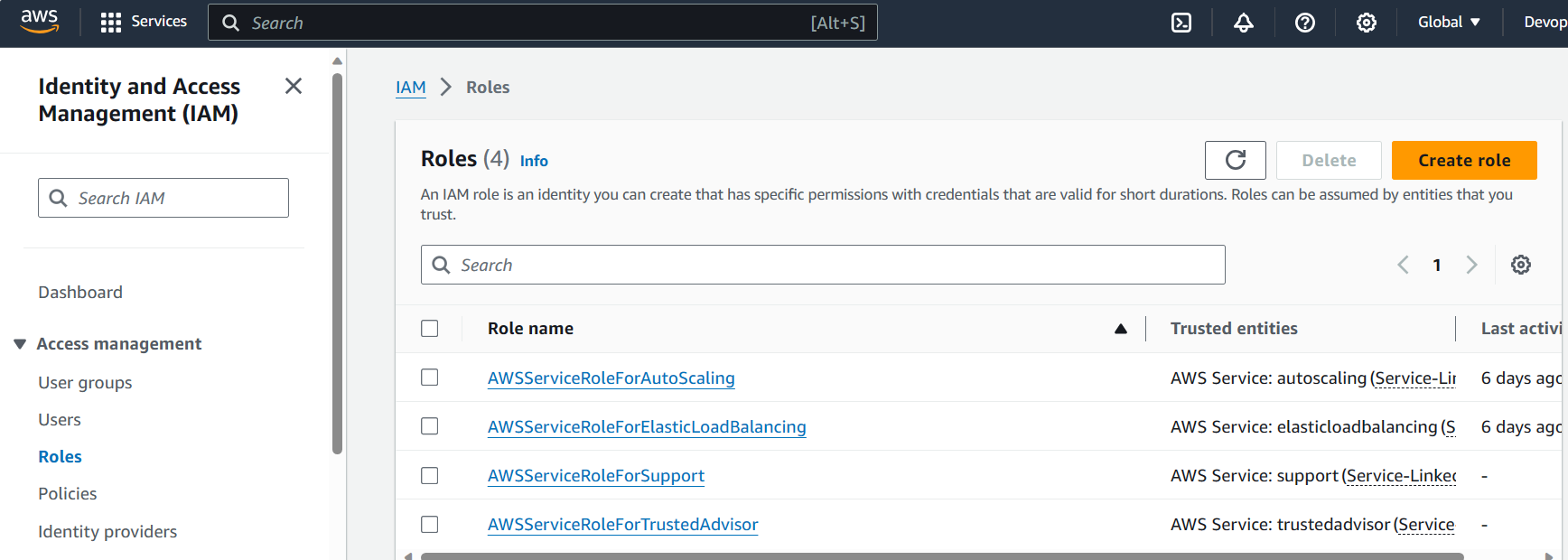
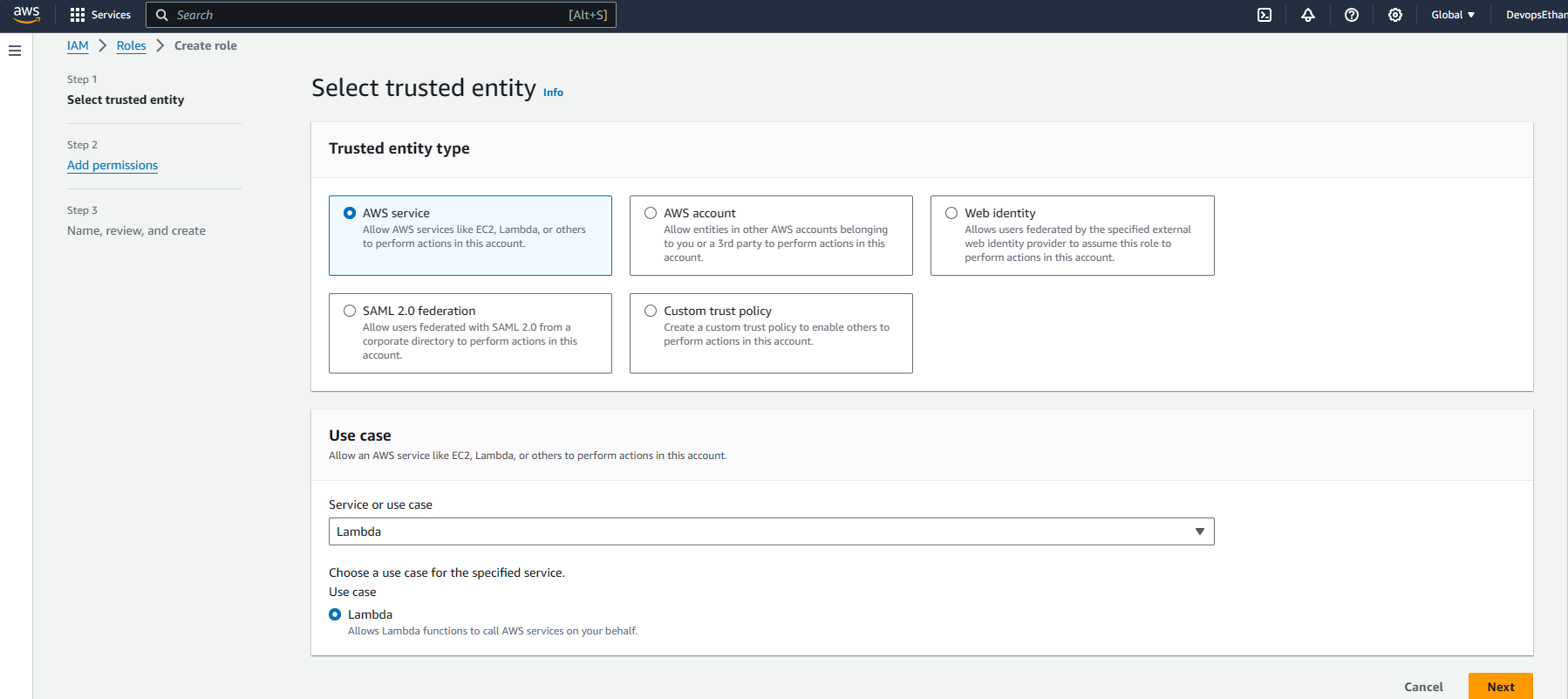
**AWS Lambda using an Amazon S3 trigger to invoke a Lambda Function**

Steps followed to achieve this:  
1. Create an IAM Role in AWS

Go to IAM Service

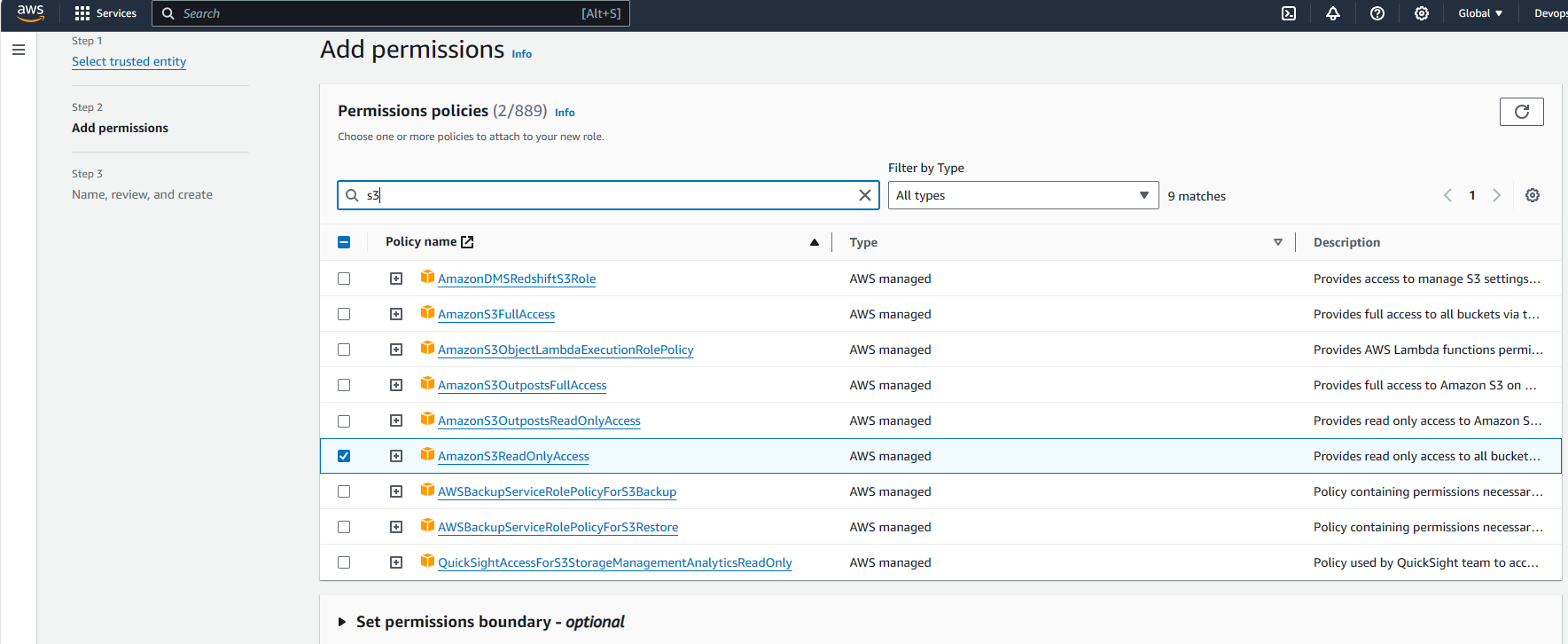


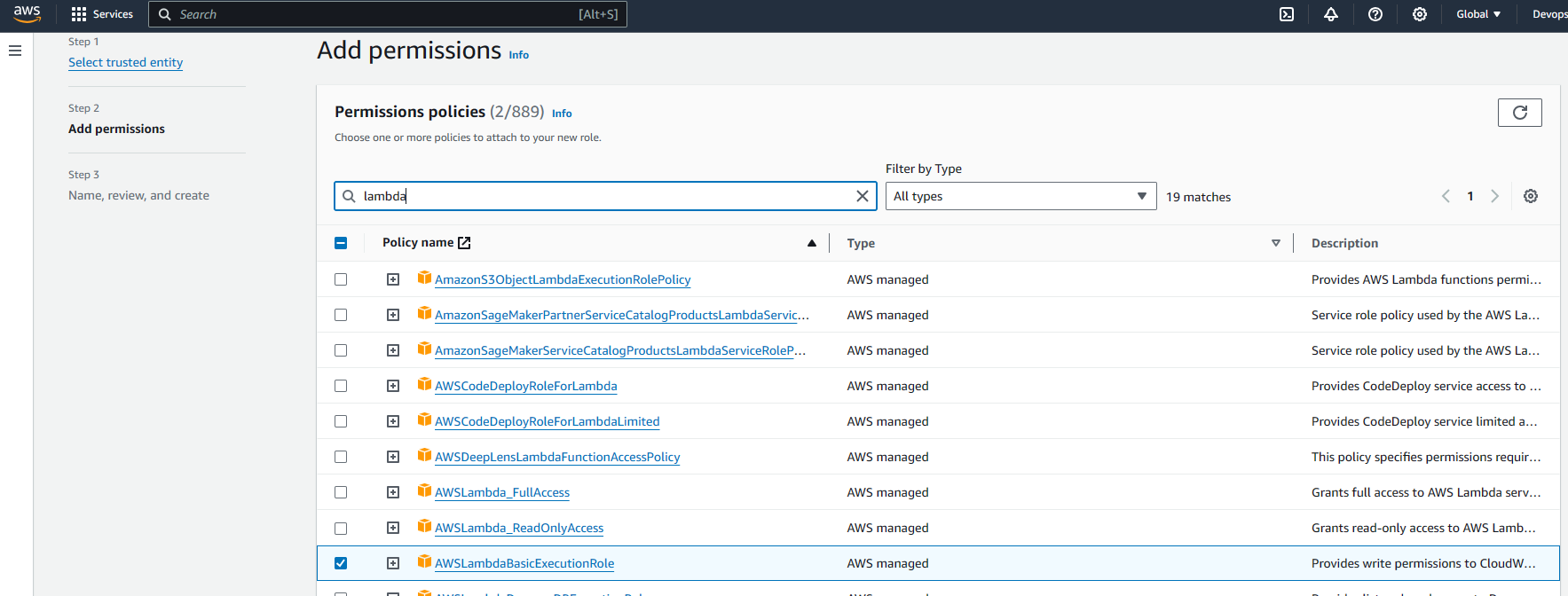
Create Role



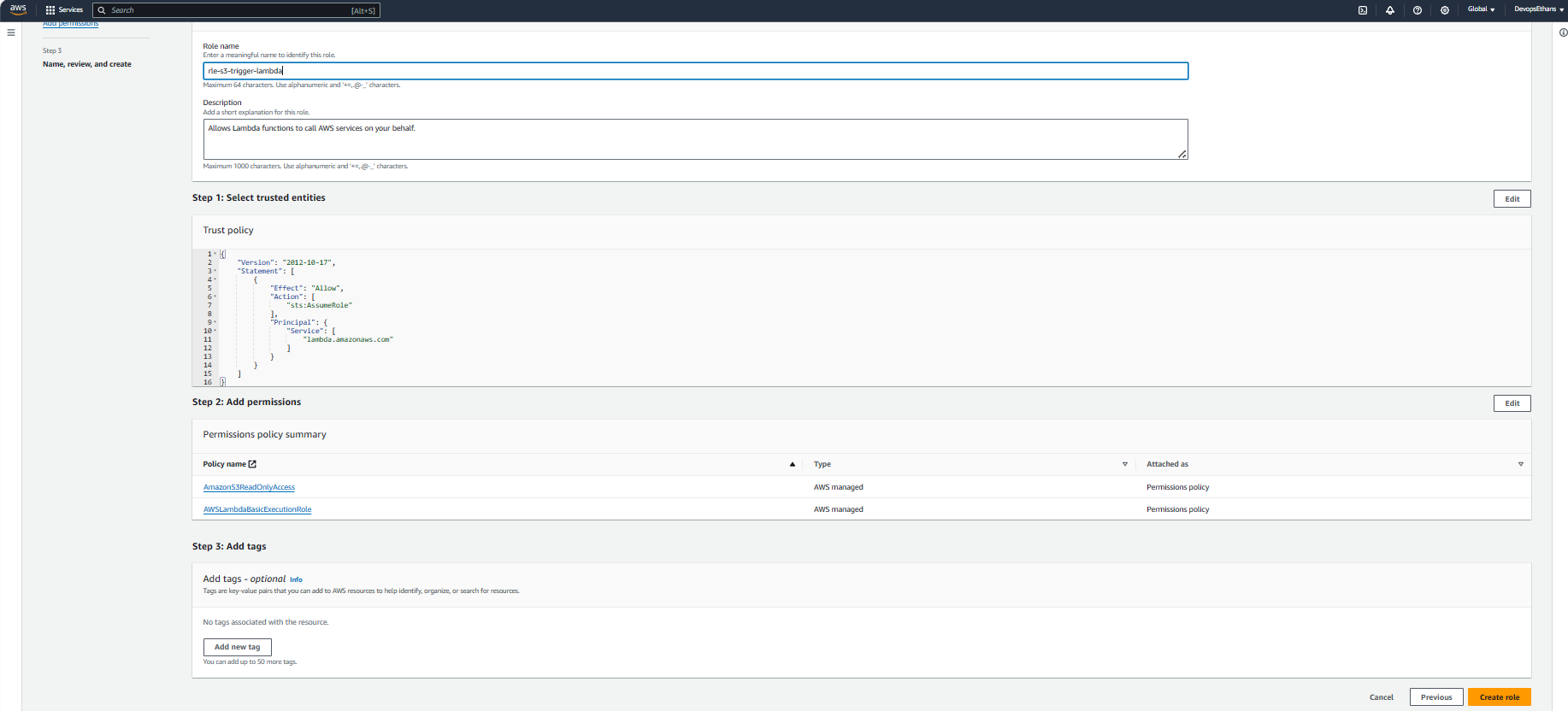
Select permissions

1. AmazonS3ReadOnlyAccess
2. AWSLambdaBasicExecutionRole



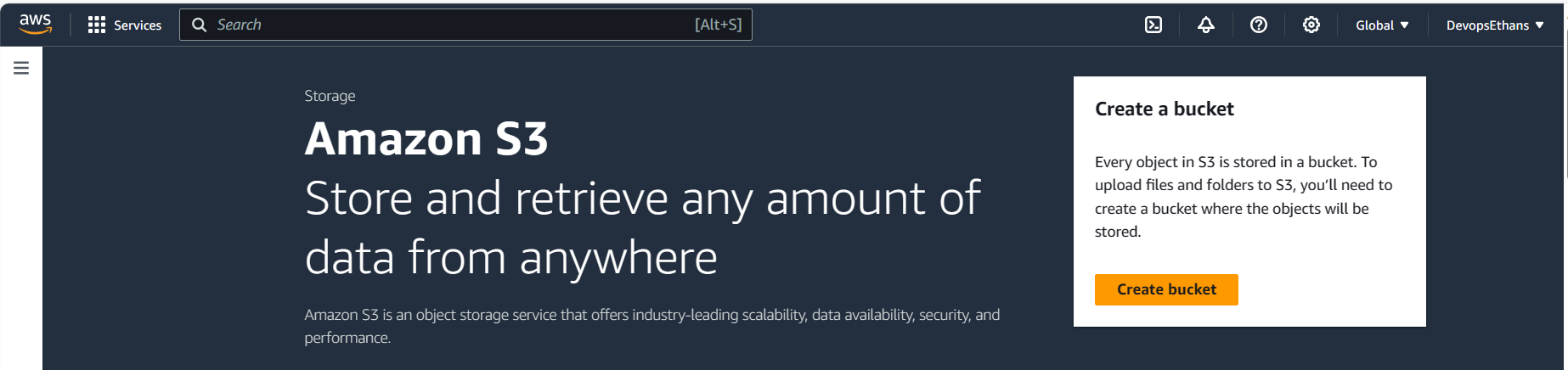


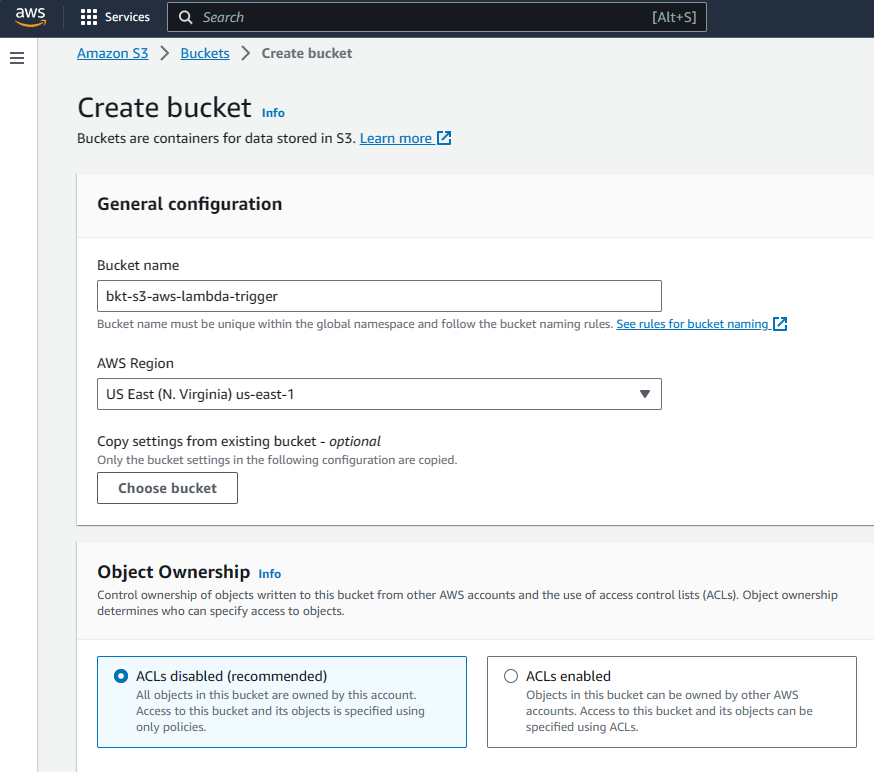
Create Role

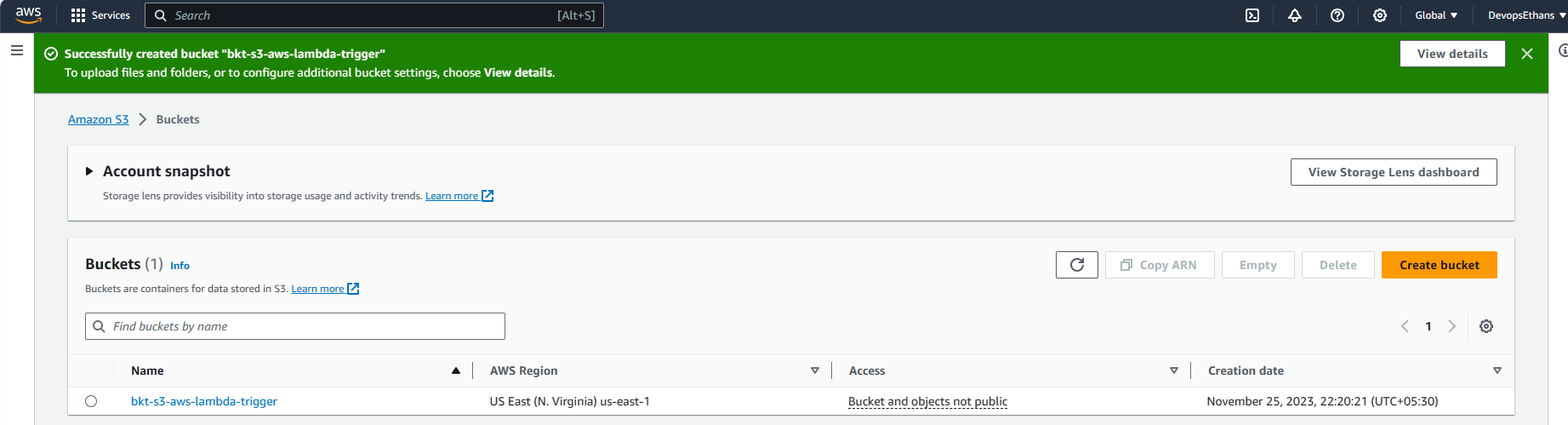


2. Create an AWS S3 Bucket.

Create S3 bucket

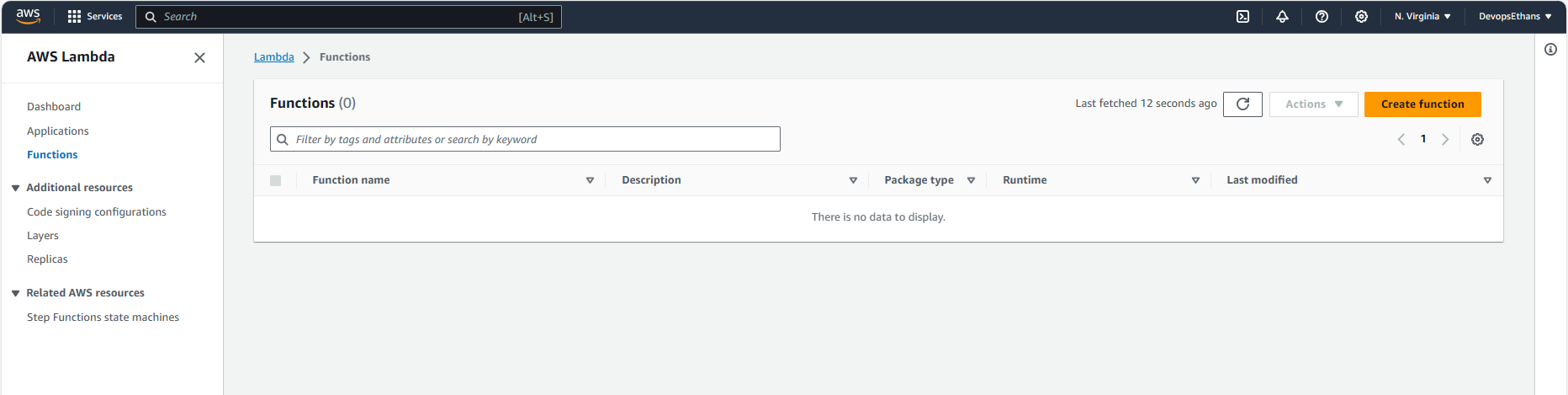




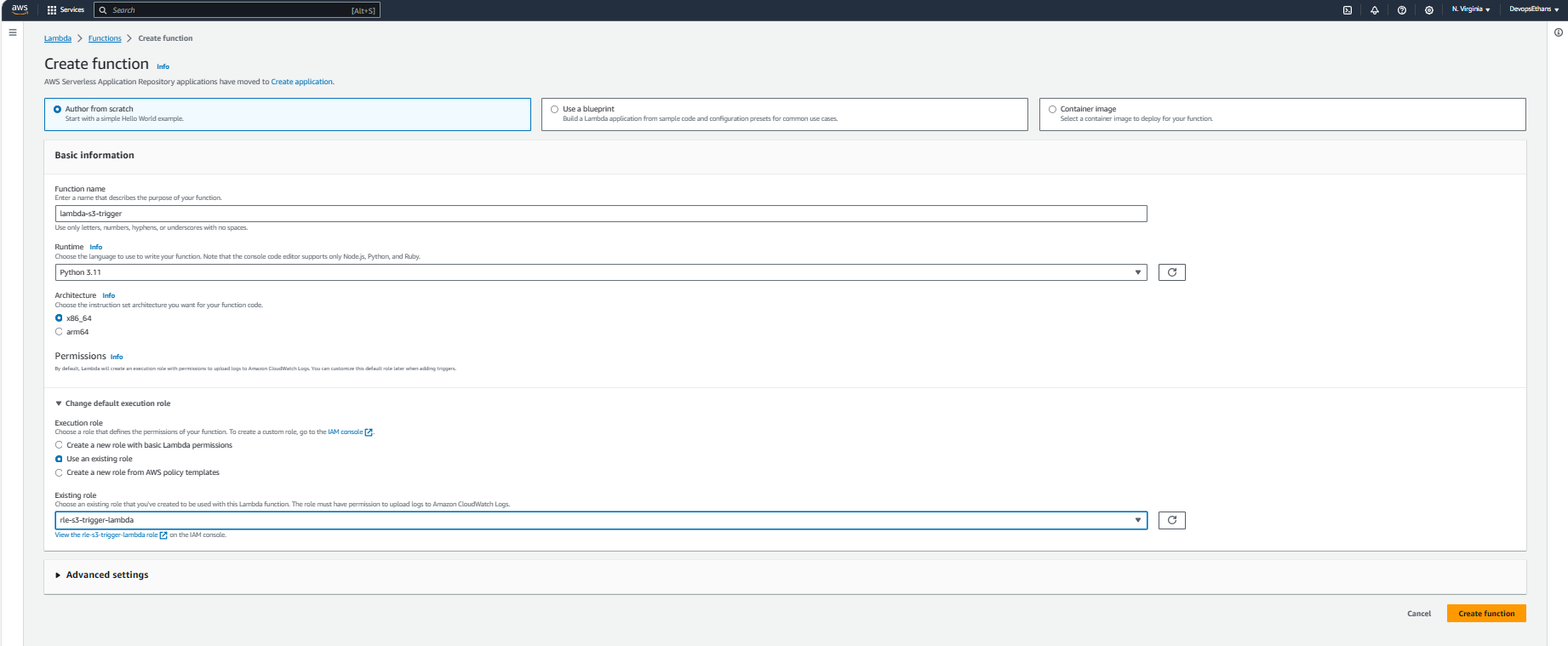


3. Create the AWS Lambda function with S3 triggers enabled.

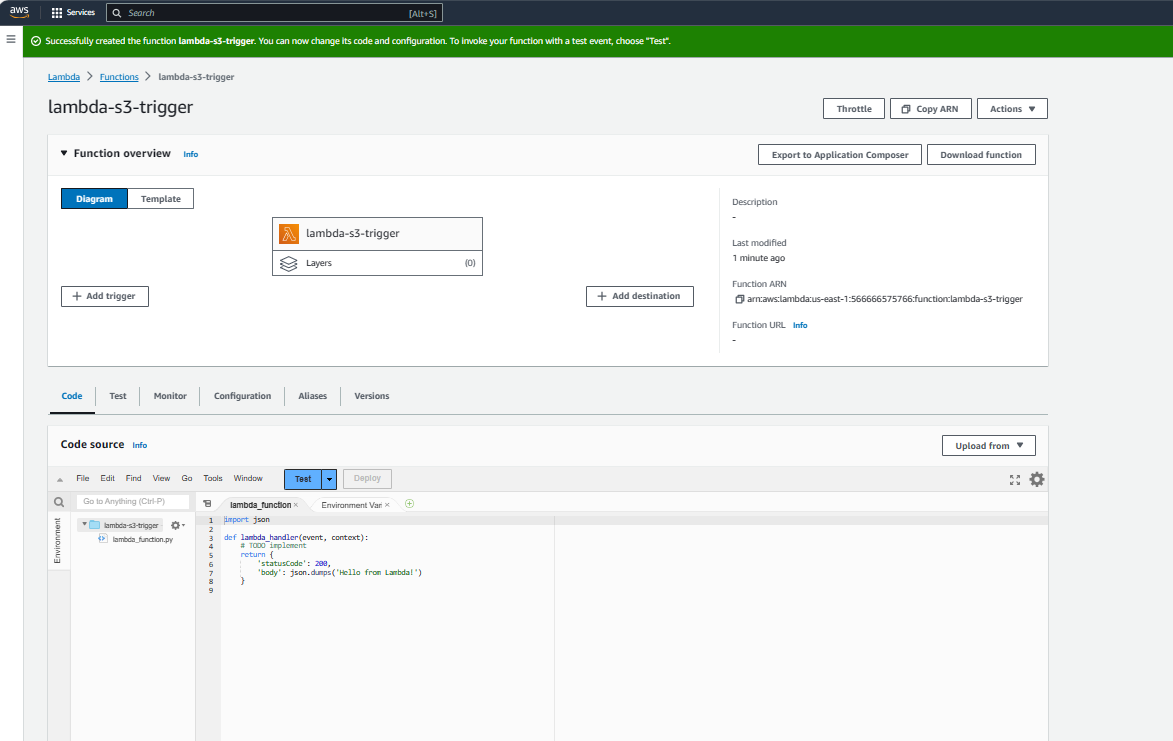
— Login to AWS management console and navigate to AWS Lambda.  
— Navigate to Lambda function & click on Create Function.

  
— Select Author from scratch and enter the basic information as mentioned below.

* **Function Name:** lambda-s3-trigger
* **Runtime:** choose run time as per the python version.
* **Architecture:** **x86\_64**
* **Change default execution role:** Select the role which was created in the Step1

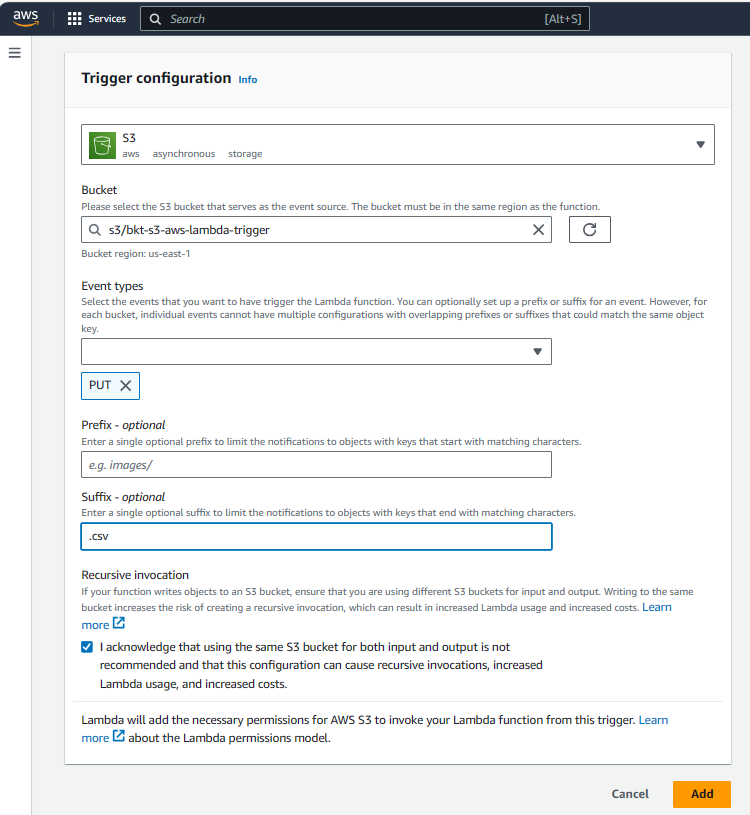


The lambda function has been created, now lets the Add the trigger to the lambda function, so that when ever the specific files have been uploaded then this lambda function will be called automatically.



— In the Add trigger window, select S3 option from the dropdown and enter the basic details mentioned below and click Add button.

* **Bucket:** select the bucket which was created in the Step 2
* **Event types:** Select only Put option, since we want to trigger the lambda function only when the files have been uploaded into the bucket.
* **Suffix:** Enter “ .csv “, so that the lambda function will be triggered only when then the file type of CSV have been uploaded.
* **Recursive invocation:**acknowledge this option. Read the description provided by AWS very carefully & try not to implement the recursive process.



4. Update the Lambda Code with python script to read the data

import json

import os

import boto3

import csv

def lambda\_handler(event, context):

for record in event['Records']:

bucket = record['s3']['bucket']['name']

file\_key = record['s3']['object']['key']

print(bucket, file\_key)

s3 = boto3.client('s3')

csvfile = s3.get\_object(Bucket=bucket, Key=file\_key)

csvcontent = csvfile['Body'].read().split(b'\n')