LAMBDA 👍

AWS Lambda

Definition

AWS Lambda is a service that lets you run code without having to manage servers. You just write your code, and AWS Lambda runs it for you whenever it's needed.

Explanation

Imagine you have a magic box that can do your homework for you. You tell the box what you need it to do (like solving math problems), and whenever you need help, you just call the box, and it does the work instantly.

AWS Lambda works in a similar way:

1. **You Write the Code**: You write instructions for the magic box (Lambda function). For example, you might tell it to resize a picture or calculate a number.
2. **AWS Takes Care of the Rest**: You don't need to worry about where the box is or how it works. AWS handles all the complicated parts like providing the electricity (computing power) and storage.
3. **It's Fast and On-Demand**: Whenever you need the code to run, you just call it, and it runs immediately. You only pay for the time it takes to do the work.

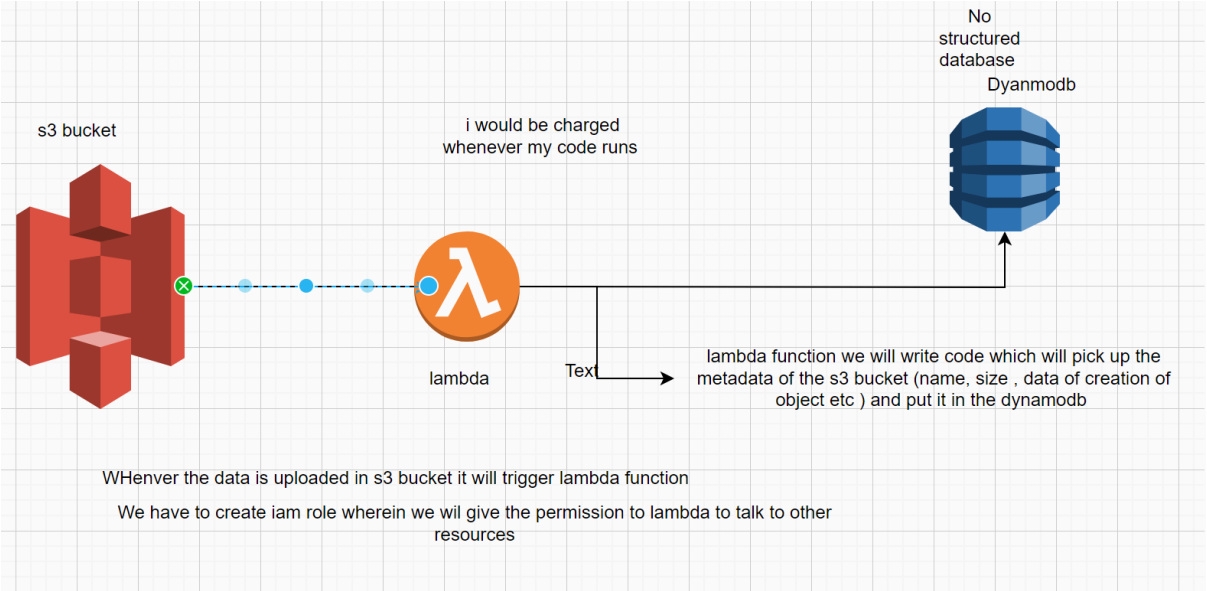
Example

Let's say you have a website where people can upload pictures. You want every picture to be resized to fit nicely on the webpage.

- **You Write a Lambda Function**: You write code that resizes pictures.
- **Someone Uploads a Picture**: When a picture is uploaded to your website, AWS Lambda automatically runs your resizing code.
- **Picture is Resized**: The picture is resized and ready to be displayed on your website.

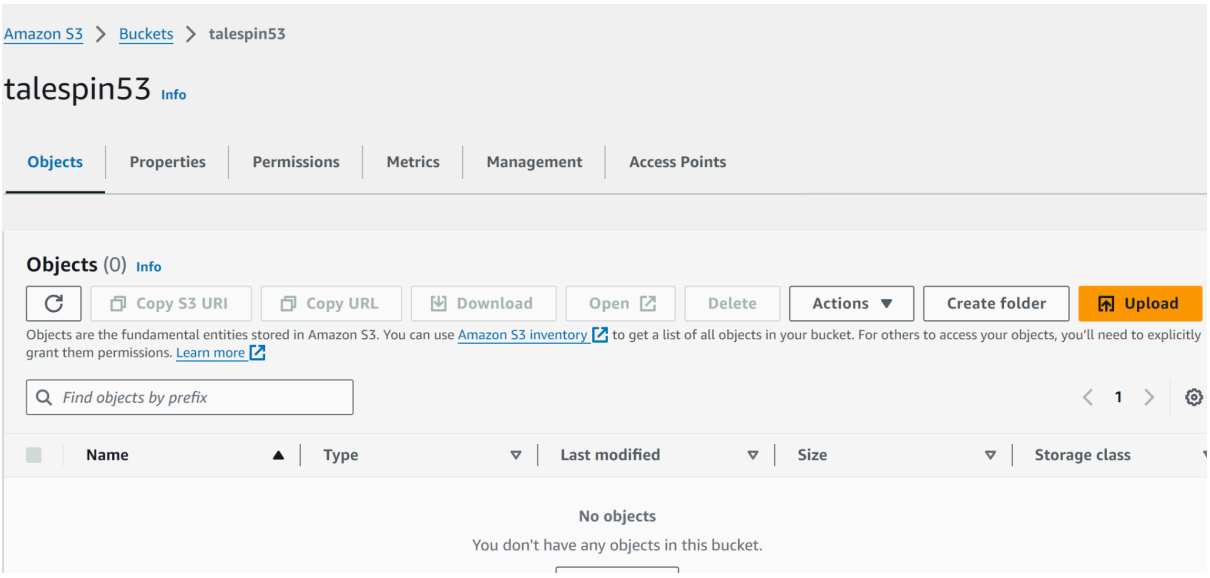
So, AWS Lambda is like having a magical helper that does specific tasks for you whenever you need them, and you only have to tell it what to do without worrying about the details of how it gets done.

CASE STUDY

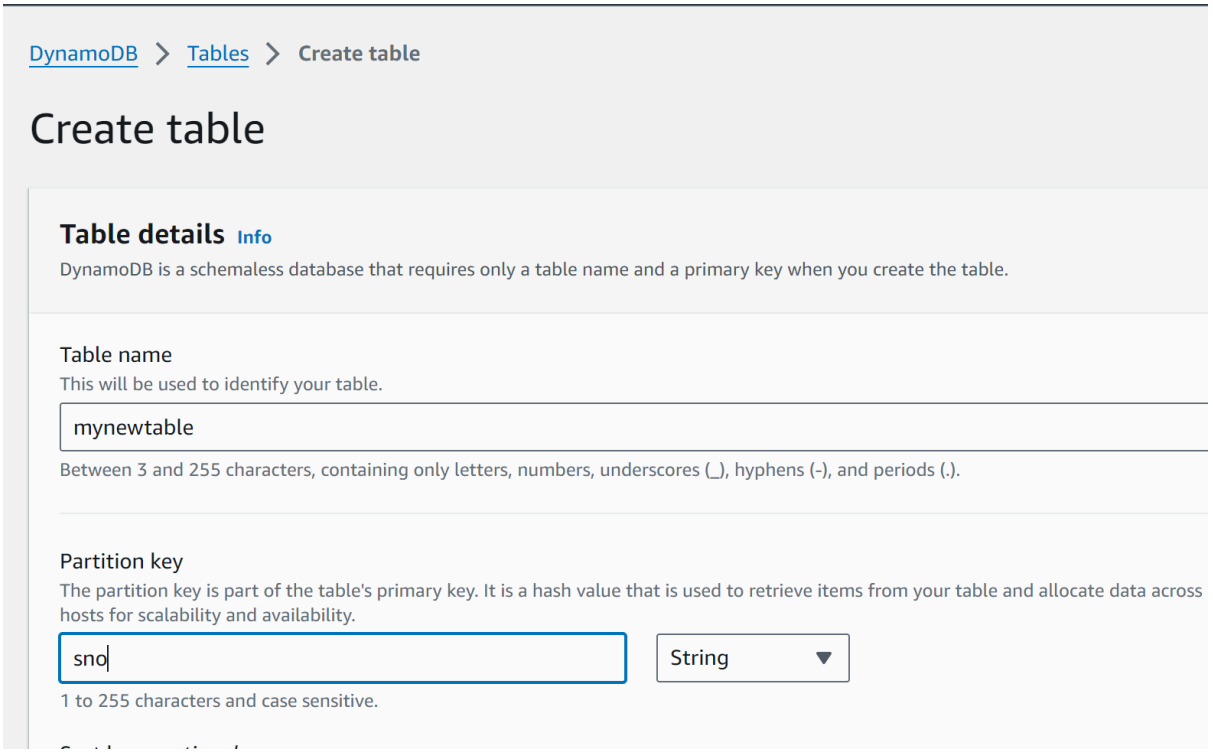


We will first create s3 bucket

Create bucket (bucket name are globally unique so keep some unique name and create)

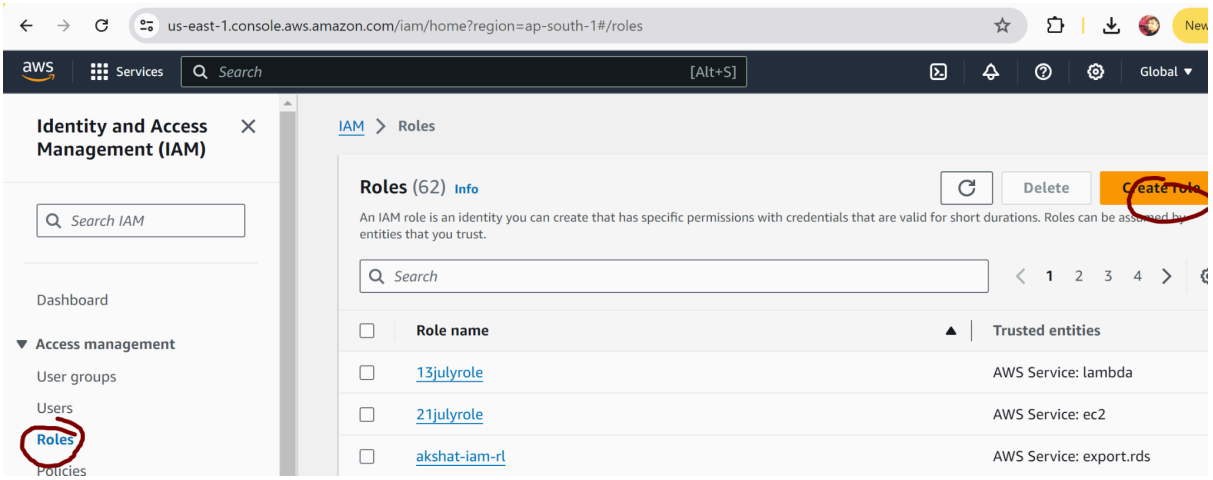


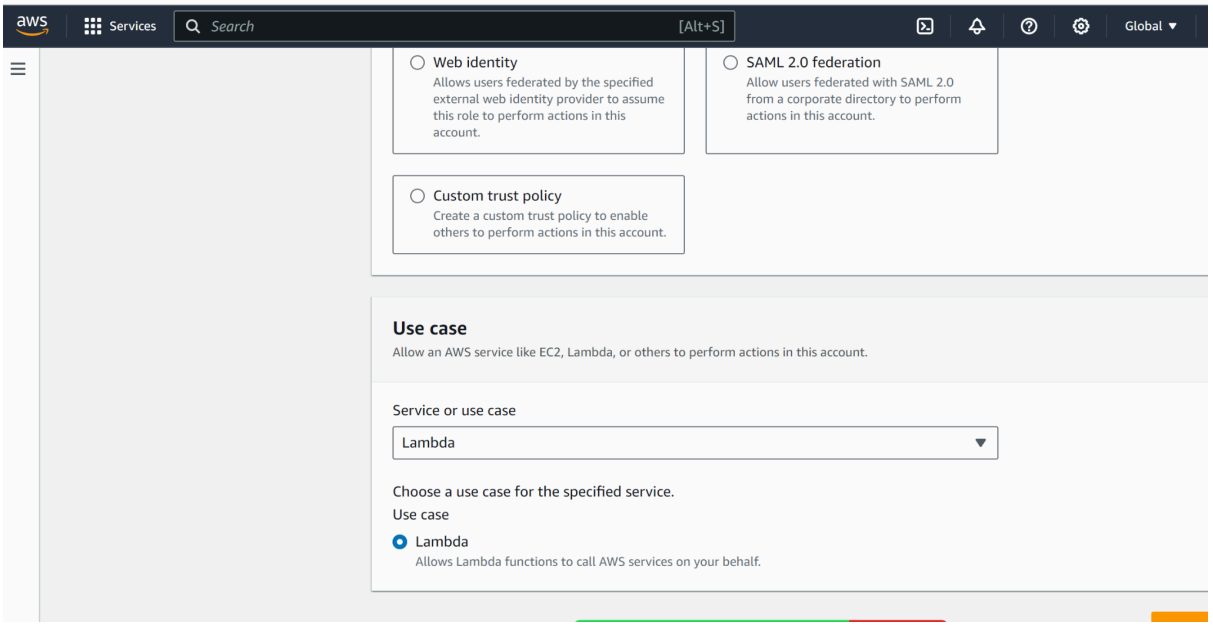
create a database dynamdb

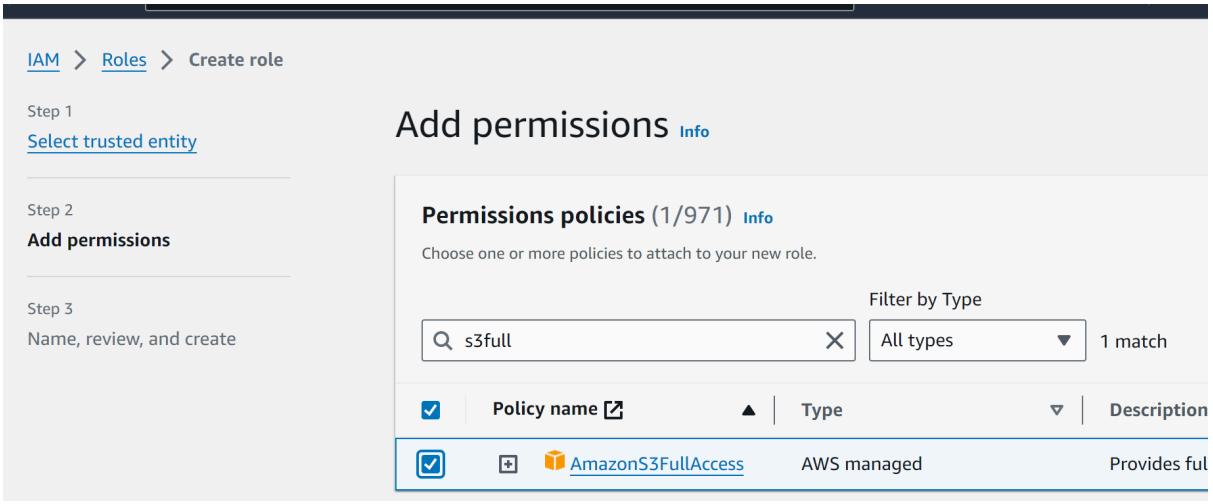


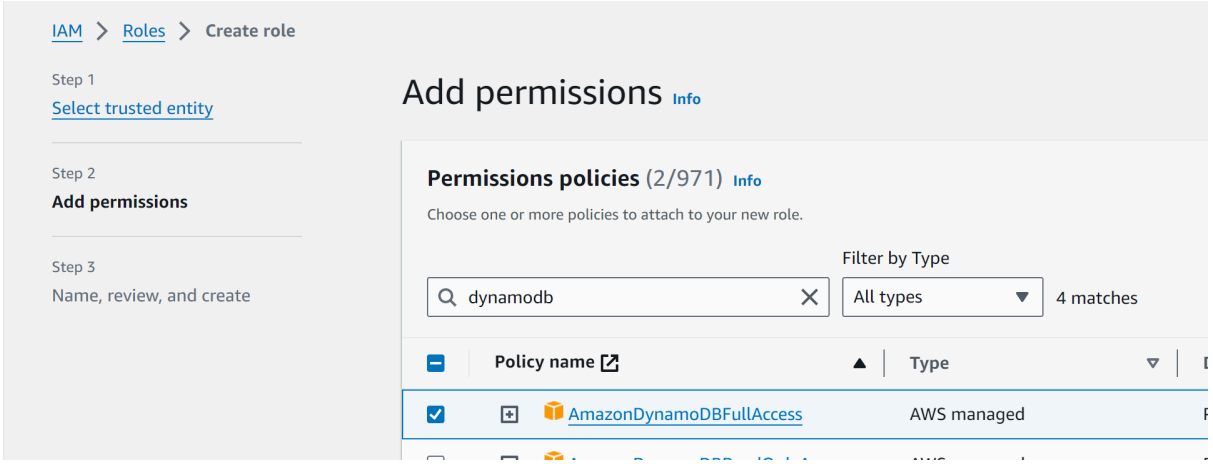
Create table

we will now create a iam role where we will give the permission to lambda to access other resources (s3 and dynamodb) on our behalf




aws
Services
Search
[Alt+S]
Global
Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.
Service or use case
Lambda
Choose a use case for the specified service.
Use case
Lambda
Allows Lambda functions to call AWS services on your behalf.

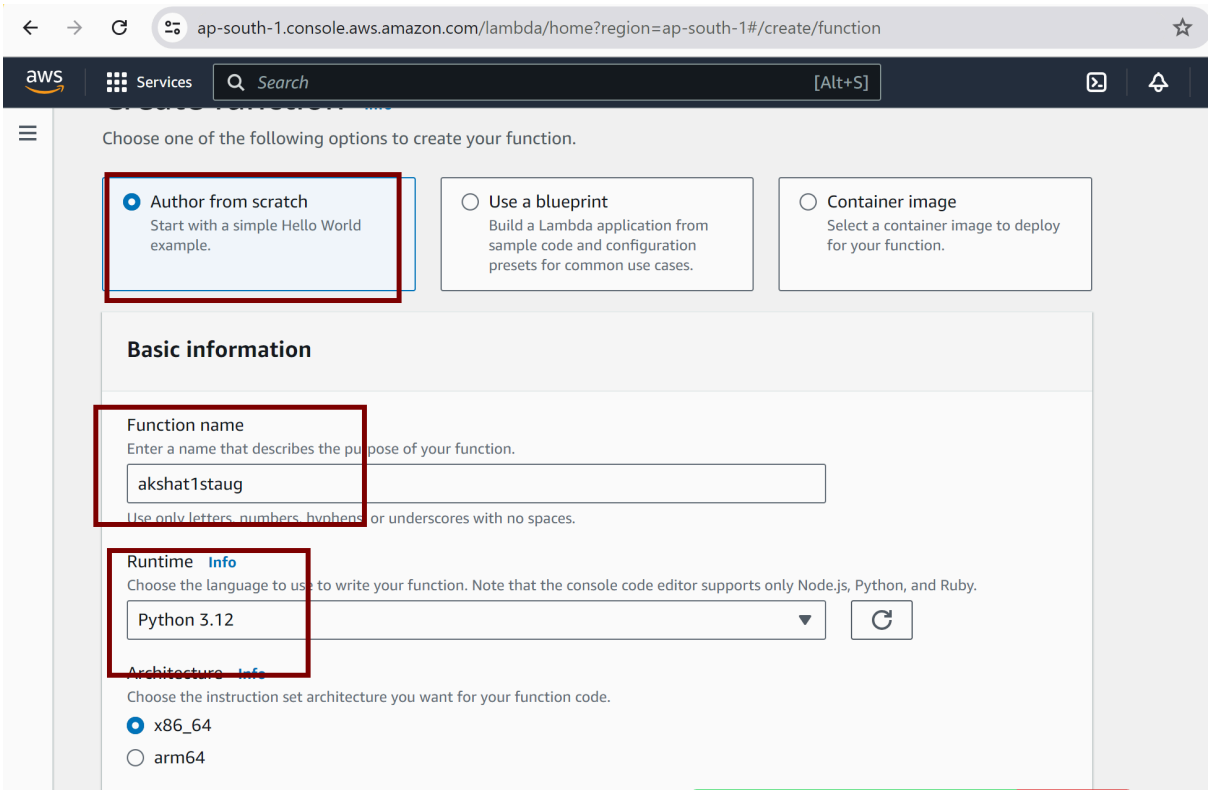

IAM
Roles
Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Add permissions
Info
Permissions policies (1/971)
Info
Choose one or more policies to attach to your new role.
Filter by Type
s3full
All types
1 match
Policy name
Type
Description
AmazonS3FullAccess
AWS managed
Provides ful

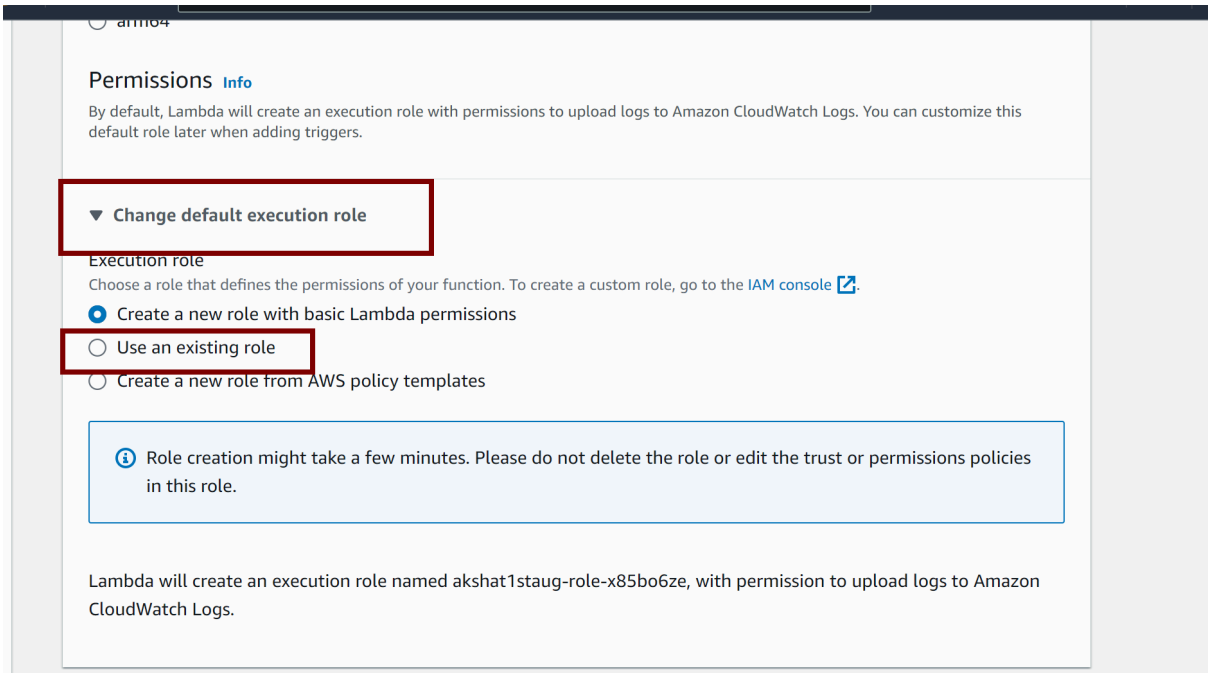

IAM
Roles
Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Add permissions
Info
Permissions policies (2/971)
Info
Choose one or more policies to attach to your new role.
Filter by Type
dynamodb
All types
4 matches
Policy name
Type
AmazonDynamoDBFullAccess
AWS managed

Create role

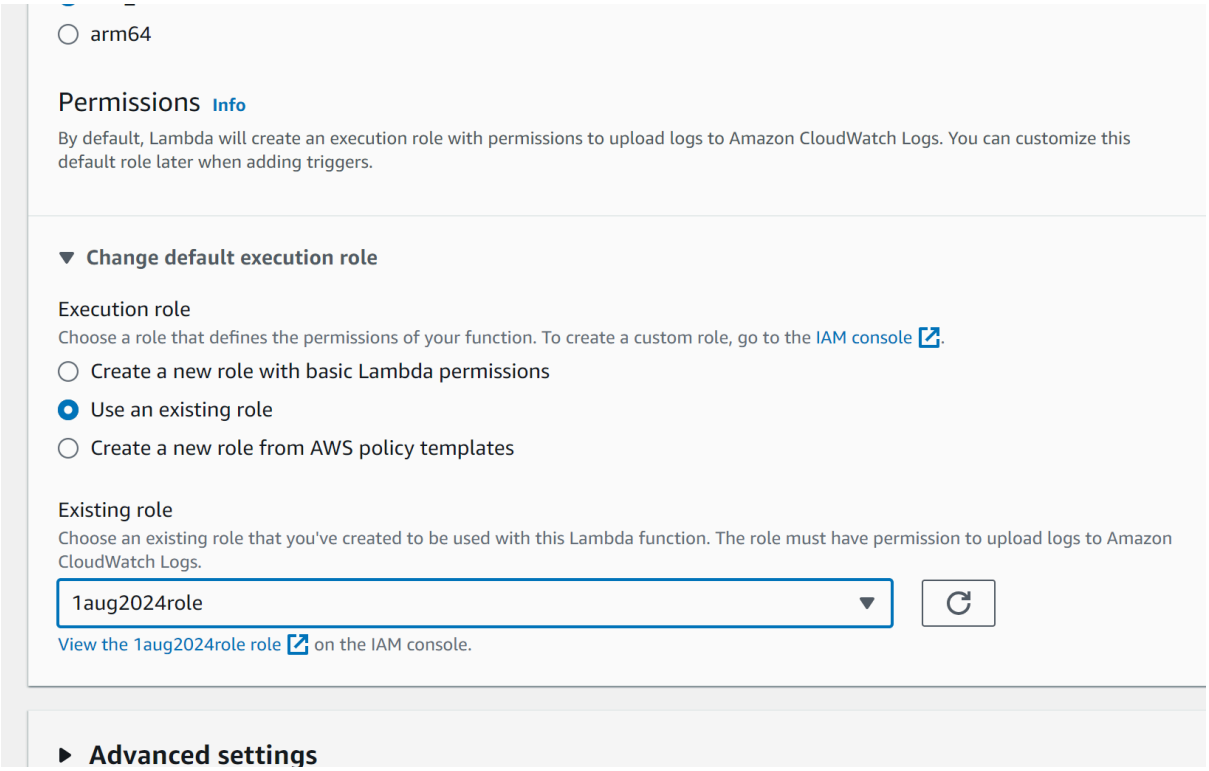
now we will create lambda function and to trigger whenever there any event happening the s3
(like uploading any object or deletion of object)

Go to lambda -> create function

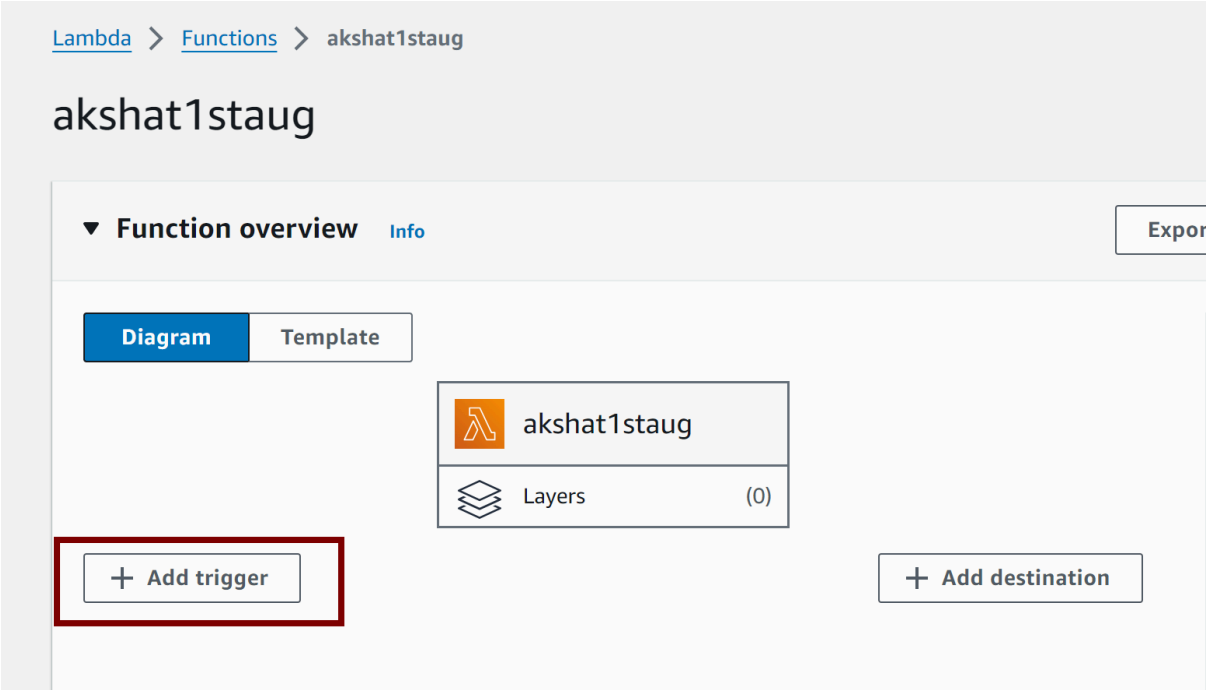


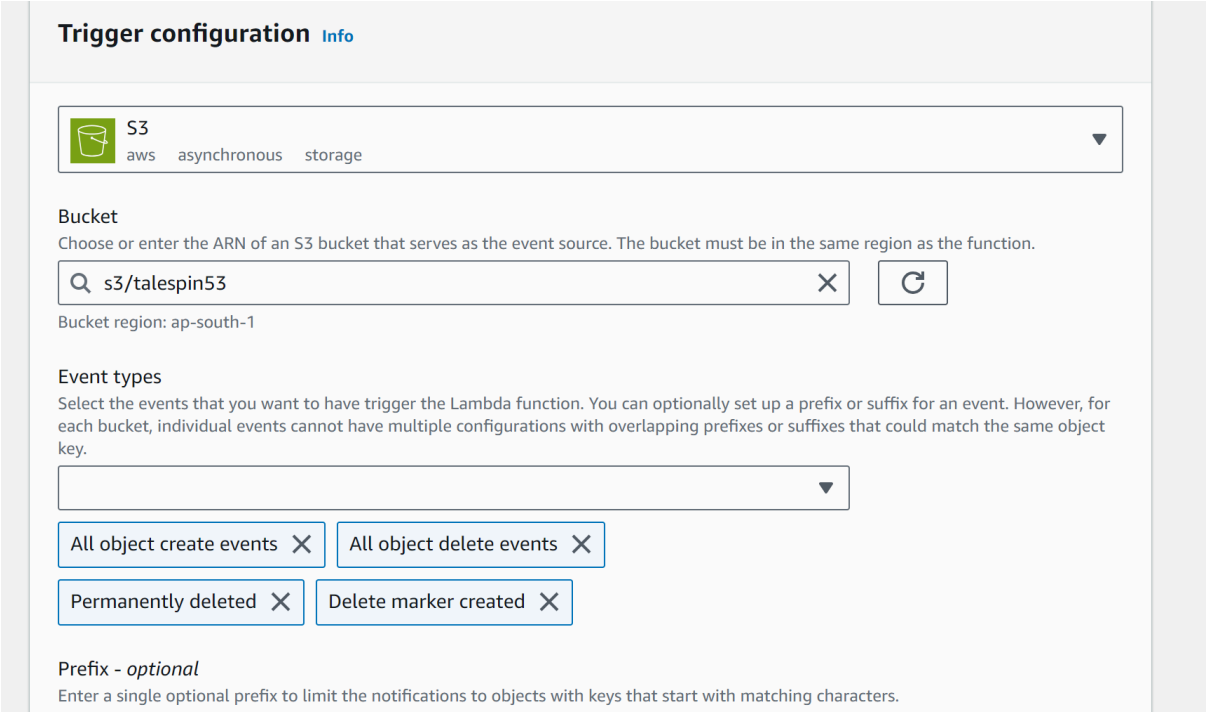


We will attach the role which we created previously

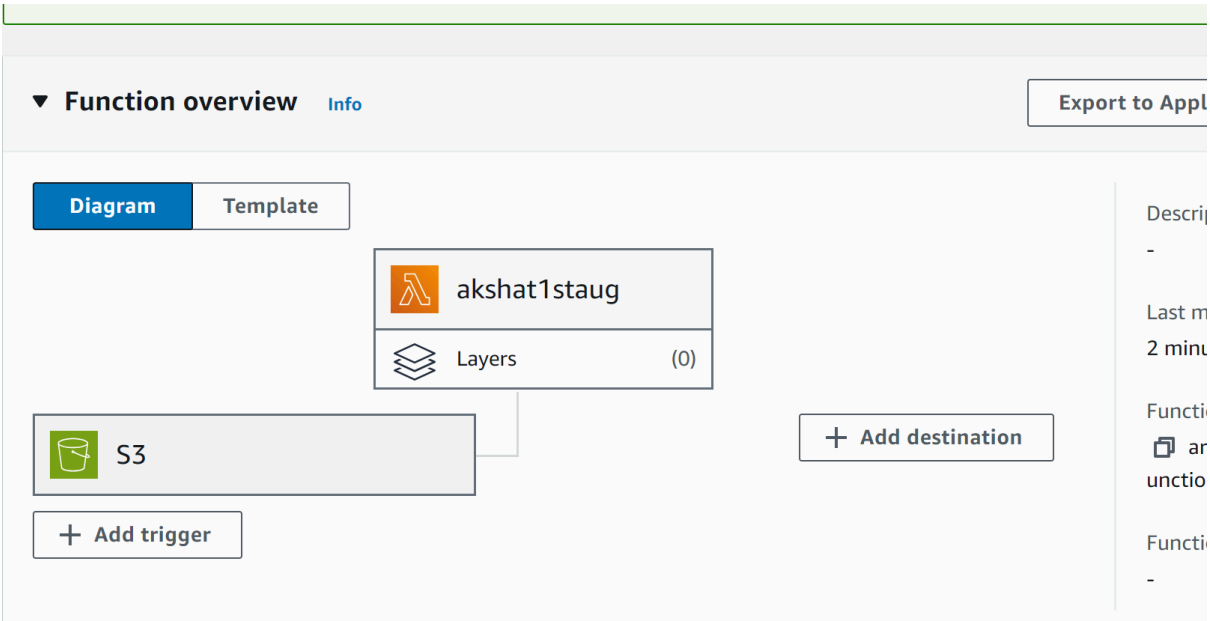


Create function





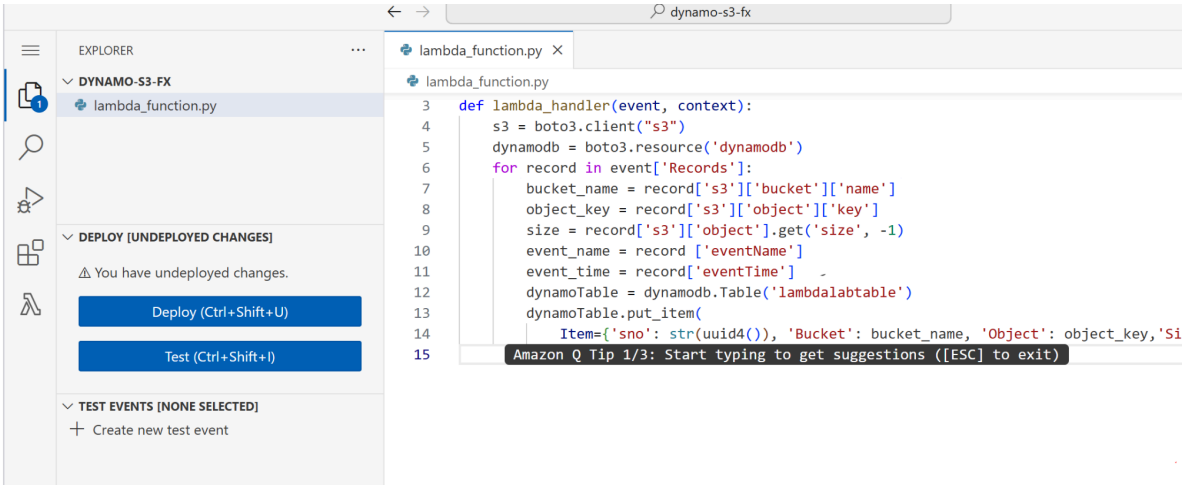
(in event types select all delete events also)



Scroll down and click on code

```
import boto3
from uuid import uuid4
def lambda_handler(event, context):
    s3 = boto3.client("s3")
    dynamodb = boto3.resource('dynamodb')
    for record in event['Records']:
```

```
        bucket_name = record['s3']['bucket']['name']
        object_key = record['s3']['object']['key']
        size = record['s3']['object'].get('size', -1)
        event_name = record ['eventName']
        event_time = record['eventTime']
        dynamoTable = dynamodb.Table('mynewtable')
        dynamoTable.put_item(
            Item={'sno': str(uuid4()), 'Bucket': bucket_name, 'Object': object_key,'Size': size,
'Event': event_name, 'EventTime': event_time})
```

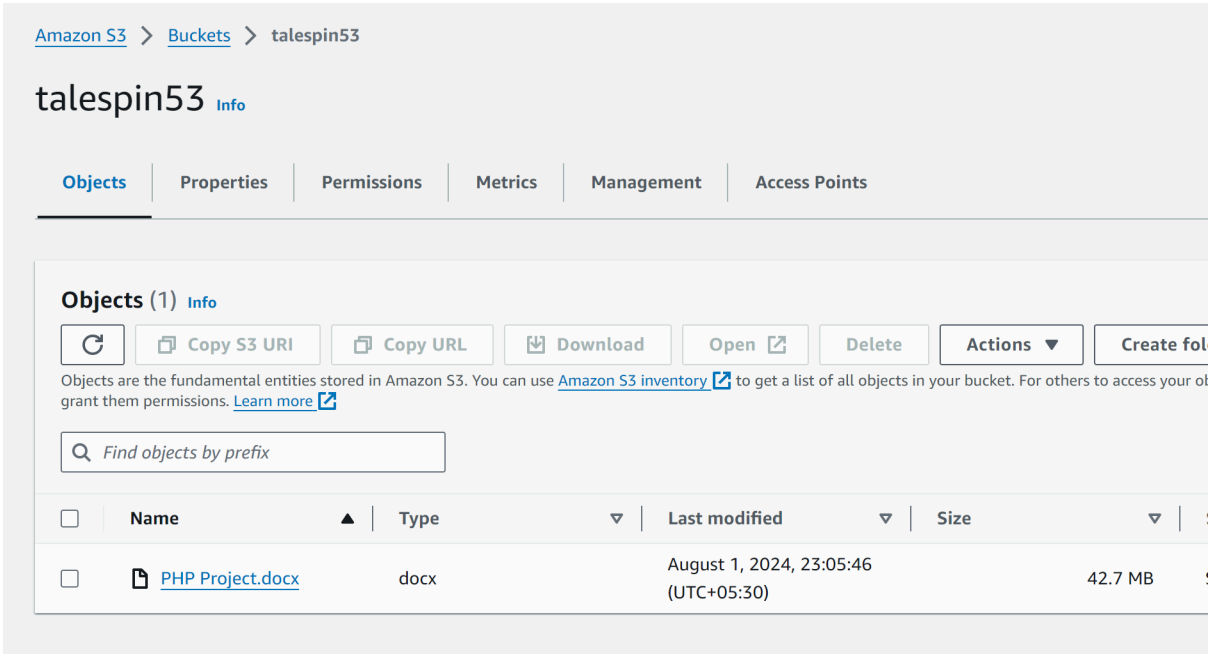


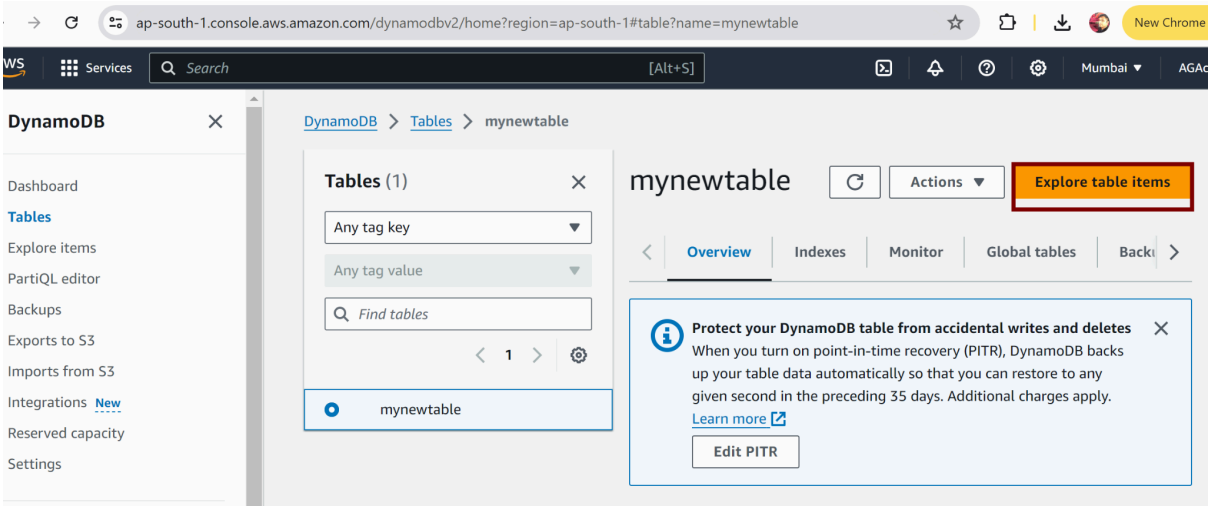
Click on deploy

Or press ctrl shift u to deploy

now lets check if we update anything in s3 whether the dynamodb is getting updated or not

Go to s3 -> click on your bucket
Click on upload button



You can upload anything

Now lets go and check dynamodb