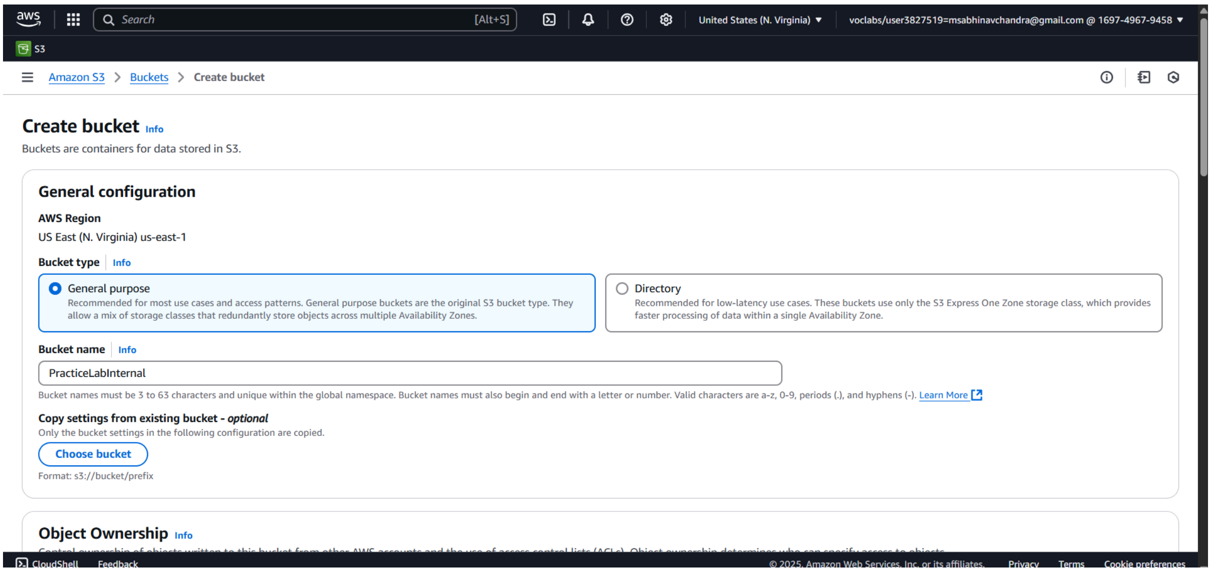
Lambda

Step 1 : We need to open 3 Services parallelly in 3 different tabs •
 S3 Bucket •Dynamo DB •
Lambda Function


aws
Search [Alt+S]
United States (N. Virginia) ▾
voclabs/user3827519=msabhinavchandra@gmail.com @ 1697-4967-9458 ▾
S3
Amazon S3 > Buckets > Create bucket
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type Info

● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

PracticeLabInternal

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn More

Copy settings from existing bucket - *optional*

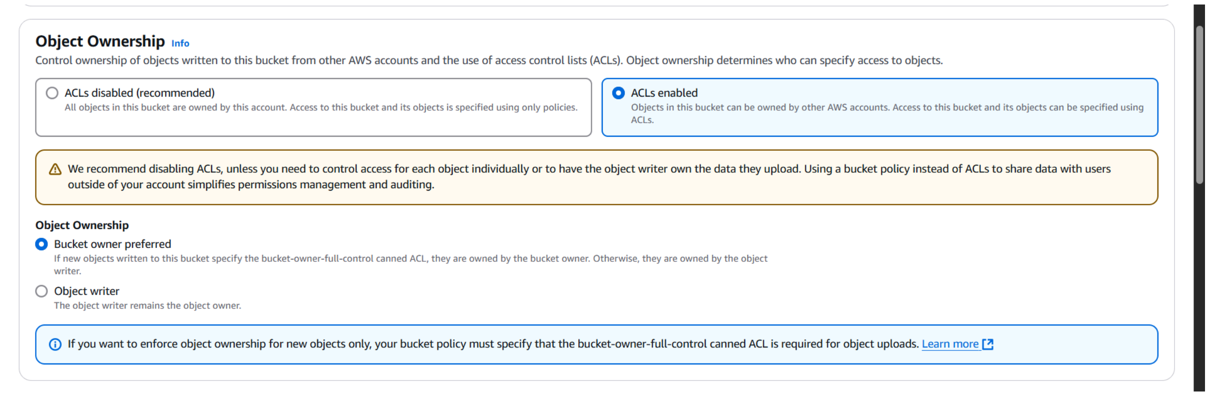
Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

Object Ownership Info

Keep it general purpose and bucket name give it unique



Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

○ ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

● ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

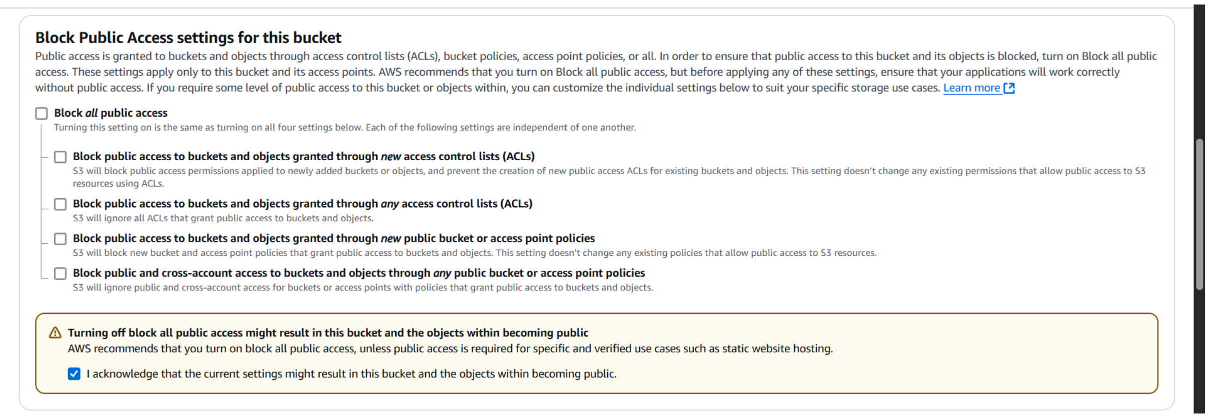
Object Ownership

● Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

○ Object writer
The object writer remains the object owner.

ⓘ If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. Learn more

Enable ACLs



Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

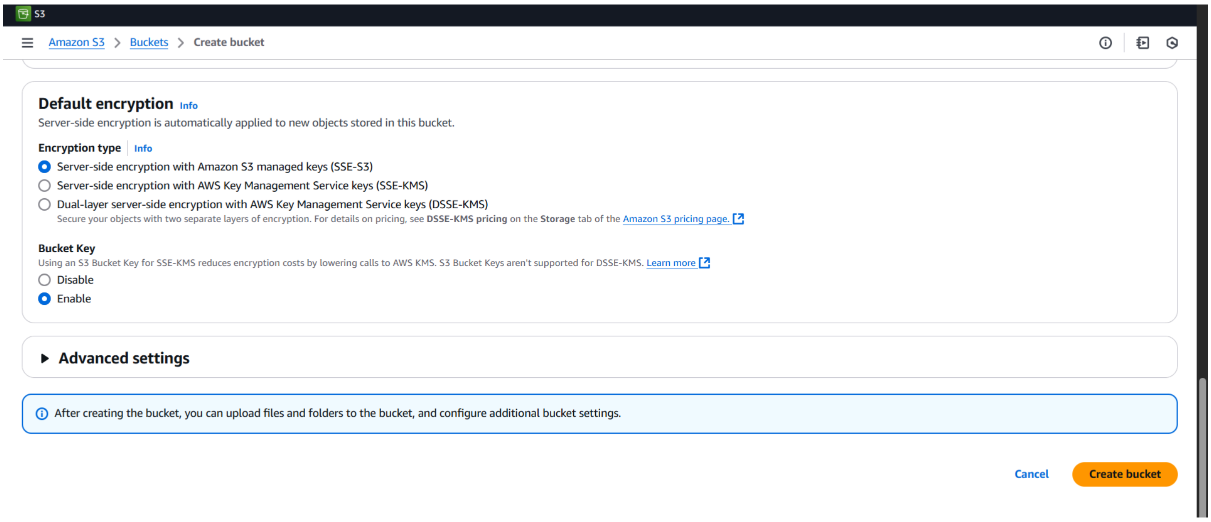
□ Block *all* public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- □ Block public access to buckets and objects granted through *new* access control lists (ACLs)
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- □ Block public access to buckets and objects granted through *any* access control lists (ACLs)
  S3 will ignore all ACLs that grant public access to buckets and objects.
- □ Block public access to buckets and objects granted through *new* public bucket or access point policies
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- □ Block public and cross-account access to buckets and objects through *any* public bucket or access point policies
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☑ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Step 4:Enable ACLs and uncheck the Block public access checkbox and proceed to accept the conditions by checking the “I acknowledge” checkbox.



Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info

● Server-side encryption with Amazon S3 managed keys (SSE-S3)
○ Server-side encryption with AWS Key Management Service keys (SSE-KMS)
○ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on the **Storage** tab of the Amazon S3 pricing page.

Bucket Key

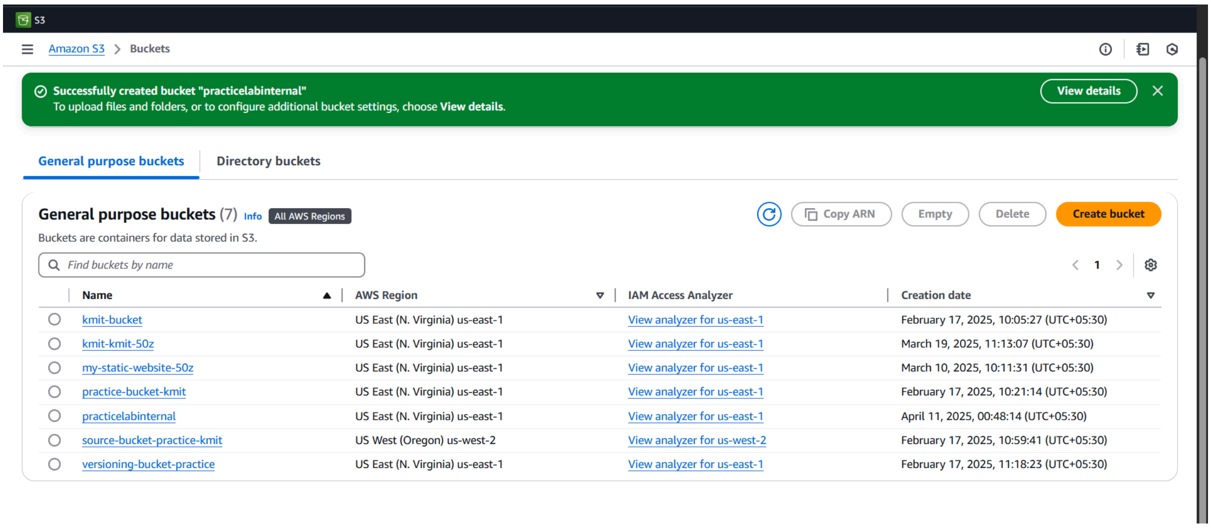
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more

○ Disable
● Enable

▸ **Advanced settings**

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Create the bucket



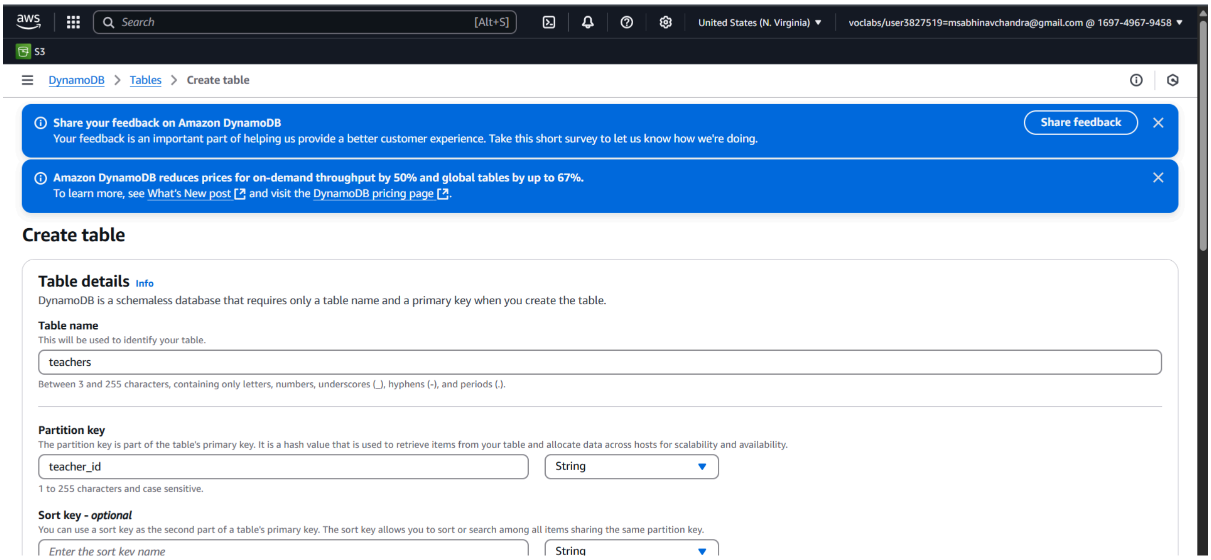
Successfully created bucket "practicelabinternal"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

General purpose buckets (7) Info All AWS Regions

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
kmit-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	February 17, 2025, 10:05:27 (UTC+05:30)
kmit-kmit-50z	US East (N. Virginia) us-east-1	View analyzer for us-east-1	March 19, 2025, 11:13:07 (UTC+05:30)
my-static-website-50z	US East (N. Virginia) us-east-1	View analyzer for us-east-1	March 10, 2025, 10:11:31 (UTC+05:30)
practice-bucket-kmit	US East (N. Virginia) us-east-1	View analyzer for us-east-1	February 17, 2025, 10:21:14 (UTC+05:30)
practicelabinternal	US East (N. Virginia) us-east-1	View analyzer for us-east-1	April 11, 2025, 00:48:14 (UTC+05:30)
source-bucket-practice-kmit	US West (Oregon) us-west-2	View analyzer for us-west-2	February 17, 2025, 10:59:41 (UTC+05:30)
versioning-bucket-practice	US East (N. Virginia) us-east-1	View analyzer for us-east-1	February 17, 2025, 11:18:23 (UTC+05:30)

Bucker created successfully
Now create dynamoDB trigger


aws
Search
[Alt+S]
United States (N. Virginia) ▾
voclabs/user3827519=msabhinavchandra@gmail.com @ 1697-4967-9458 ▾
S3
DynamoDB > Tables > Create table
Share feedback

Share your feedback on Amazon DynamoDB
Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

Amazon DynamoDB reduces prices for on-demand throughput by 50% and global tables by up to 67%.
To learn more, see What's New post and visit the DynamoDB pricing page.

Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.
teachers
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

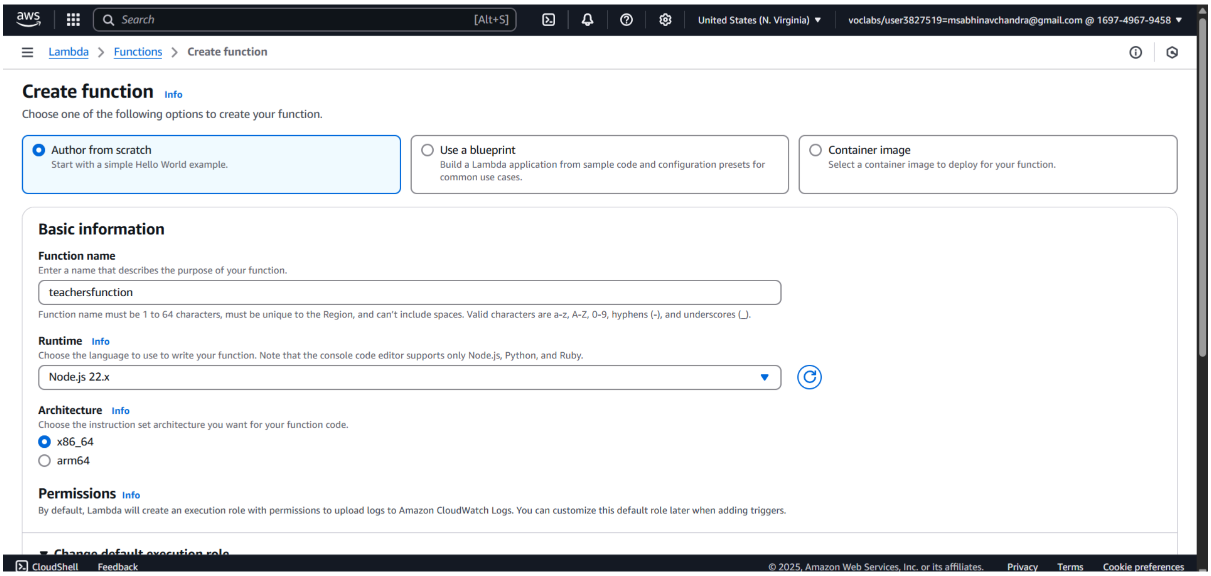
Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
teacher_id String ▾
1 to 255 characters and case sensitive.

Sort key - *optional*
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
Enter the sort key name String ▾

Enter the name of the table and Enter the partition key . Make sure that the table name is same as the name mentioned in the python script and the partition key is same as mentioned in the json file that is to be uploaded into the S3 bucket .

And just click on create table

Now go to lambda and create the function


aws
Search [Alt+S]
United States (N. Virginia) ▼
voclabs/user3827519=msabhinavchandra@gmail.com @ 1697-4967-9458 ▼
Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

● Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.
teachersfunction
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

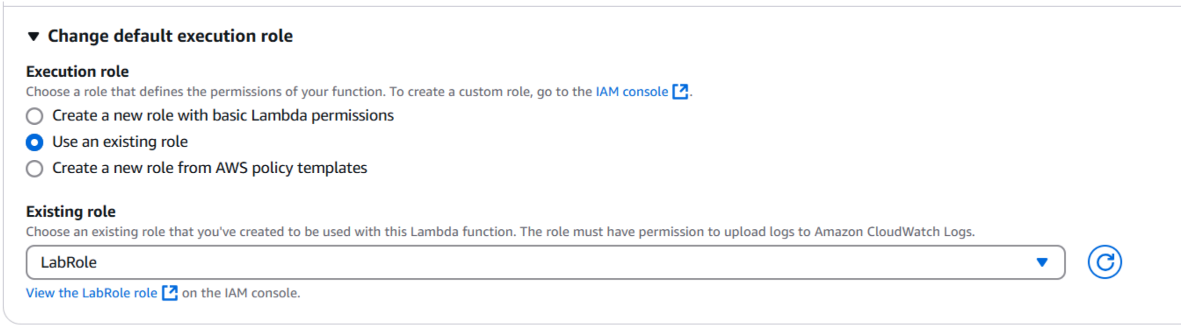
Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Node.js 22.x

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Enter a name and click on Change default execution role , choose Use an existing role and proceed to choose LabRole .

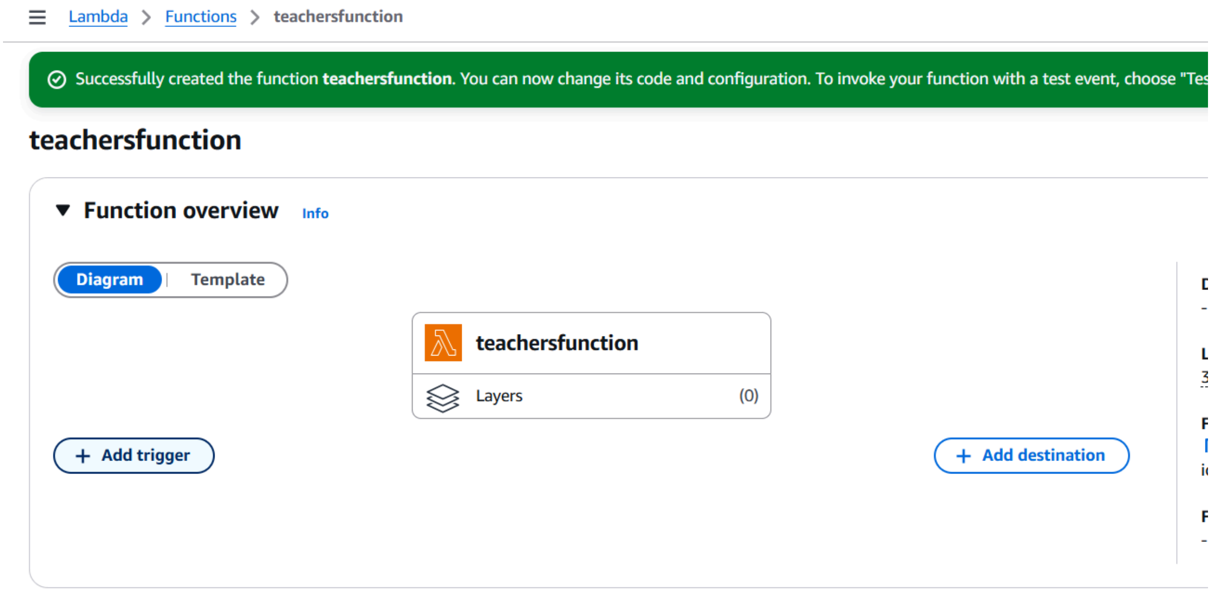


▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
○ Create a new role with basic Lambda permissions
● Use an existing role
○ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
LabRole
View the LabRole role on the IAM console.

And just click on create function

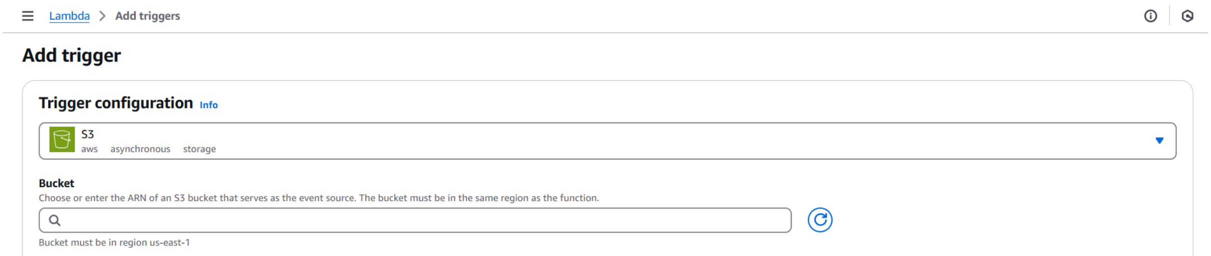


Successfully created the function **teachersfunction**. You can now change its code and configuration. To invoke your function with a test event, choose "Tes

teachersfunction

▼ **Function overview** Info

Click on add trigger



Add trigger

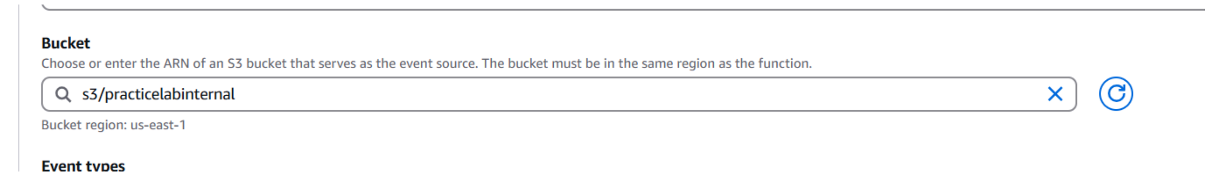
Trigger configuration

S3
aws asynchronous storage

Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

Bucket must be in region us-east-1

Trigger configuration S3



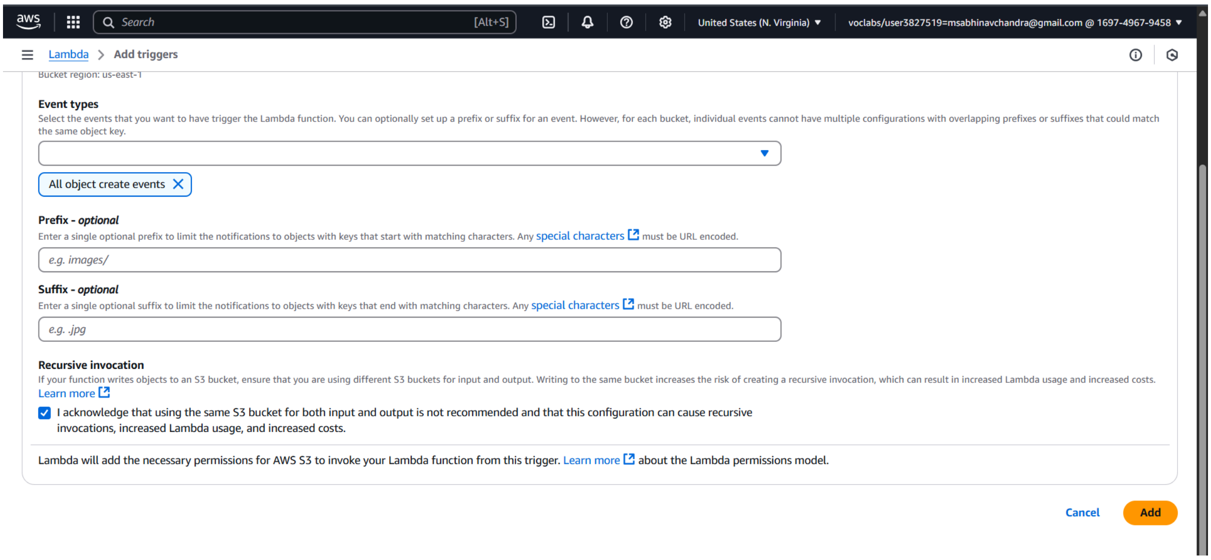
Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

s3/practicelabinternal

Bucket region: us-east-1

Event types

Choose S3 as AWS service and choose the bucket we have created .


aws
Search [Alt+S]
United States (N. Virginia) ▼
voclabs/user3827519=msabhinavchandra@gmail.com @ 1697-4967-9458 ▼
Lambda > Add triggers
Cancel
Add

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.

All object create events ✕

Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.

e.g. images/

Suffix - *optional*
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.

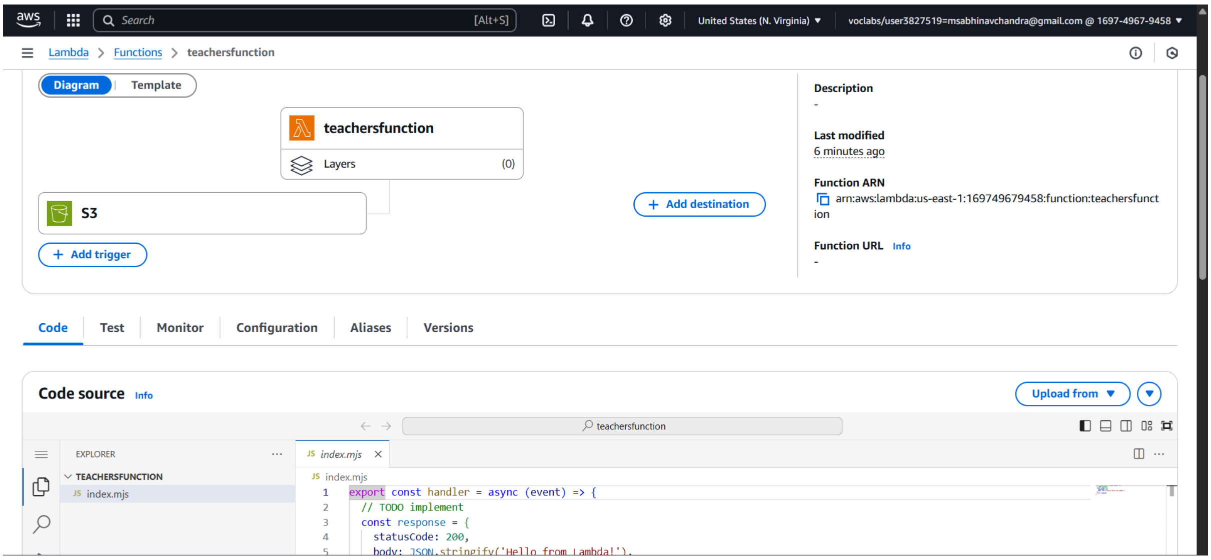
e.g. .jpg

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.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.

Click on Add to add Source Trigger , after clicking on i acknowledge



Description
-

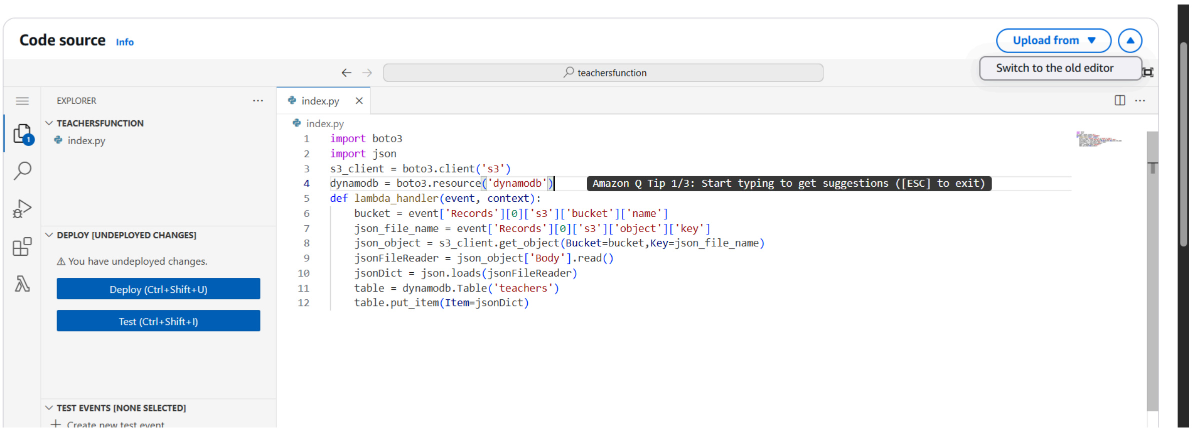
Last modified
6 minutes ago

Function ARN
arn:aws:lambda:us-east-1:169749679458:function:teachersfunction

Function URL Info
-

```
export const handler = async (event) => {
  // TODO implement
  const response = {
    statusCode: 200,
    body: JSON.stringify('Hello from Lambda!'),
```

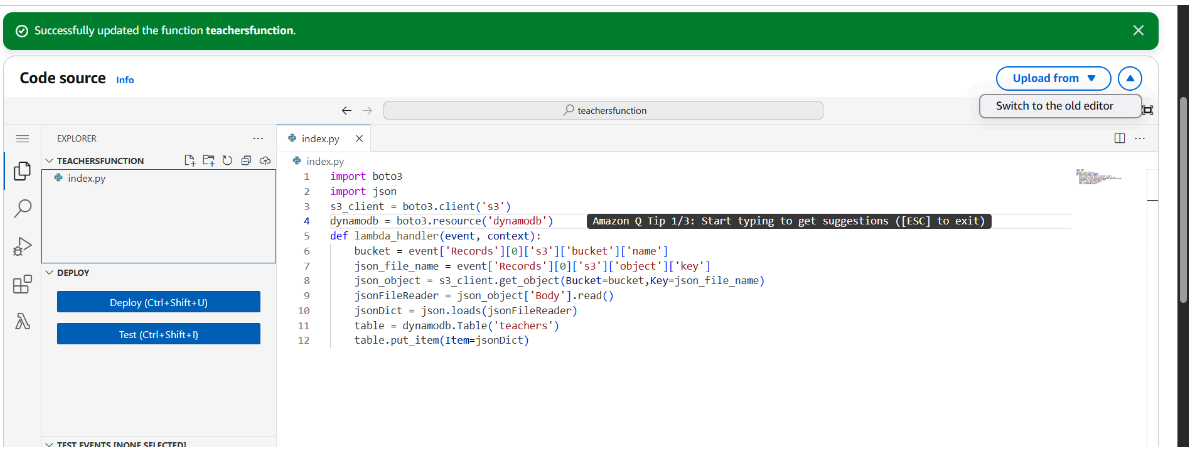
Once Created procced to Code option in the same window and paste to following code



index.py

```
import boto3
import json
s3_client = boto3.client('s3')
dynamodb = boto3.resource('dynamodb')     Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)
def lambda_handler(event, context):
    bucket = event['Records'][0]['s3']['bucket']['name']
    json_file_name = event['Records'][0]['s3']['object']['key']
    json_object = s3_client.get_object(Bucket=bucket,Key=json_file_name)
    jsonFileReader = json_object['Body'].read()
    jsonDict = json.loads(jsonFileReader)
    table = dynamodb.Table('teachers')
    table.put_item(Item=jsonDict)
```

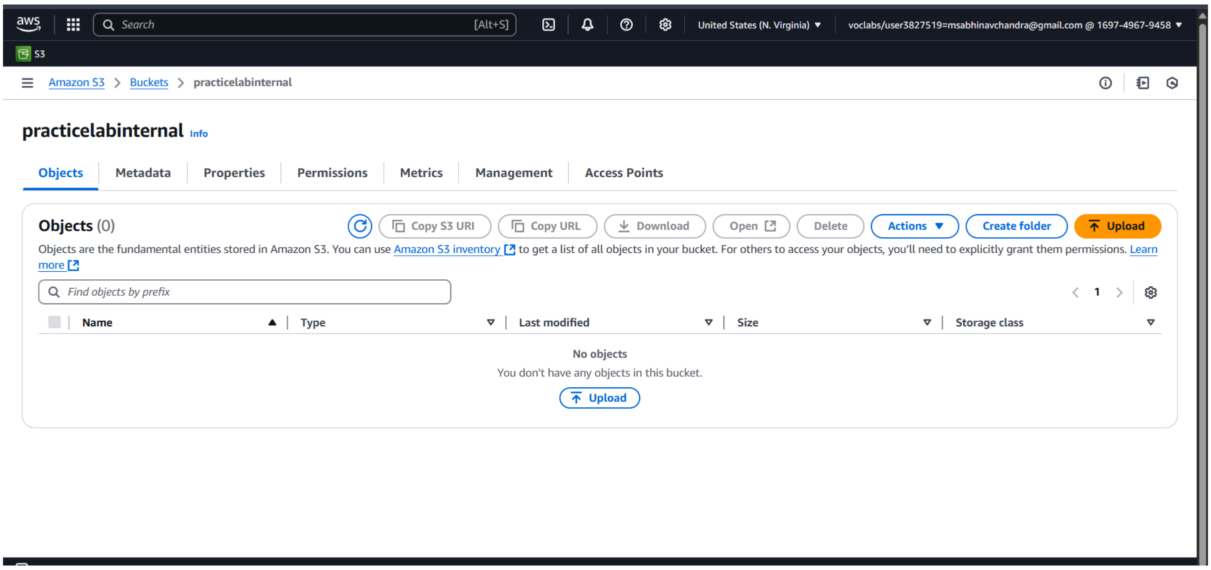
Paste the code as it is, change the dynamo table with the name of your table, and make sure u have the primary key teachers id u created for the table when your uploading this json to your s3 bucker the dynamoDB table and the s3 should be in proper sync


Successfully updated the function teachersfunction.

index.py

```
import boto3
import json
s3_client = boto3.client('s3')
dynamodb = boto3.resource('dynamodb')     Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)
def lambda_handler(event, context):
    bucket = event['Records'][0]['s3']['bucket']['name']
    json_file_name = event['Records'][0]['s3']['object']['key']
    json_object = s3_client.get_object(Bucket=bucket,Key=json_file_name)
    jsonFileReader = json_object['Body'].read()
    jsonDict = json.loads(jsonFileReader)
    table = dynamodb.Table('teachers')
    table.put_item(Item=jsonDict)
```

Click on deploy later on.



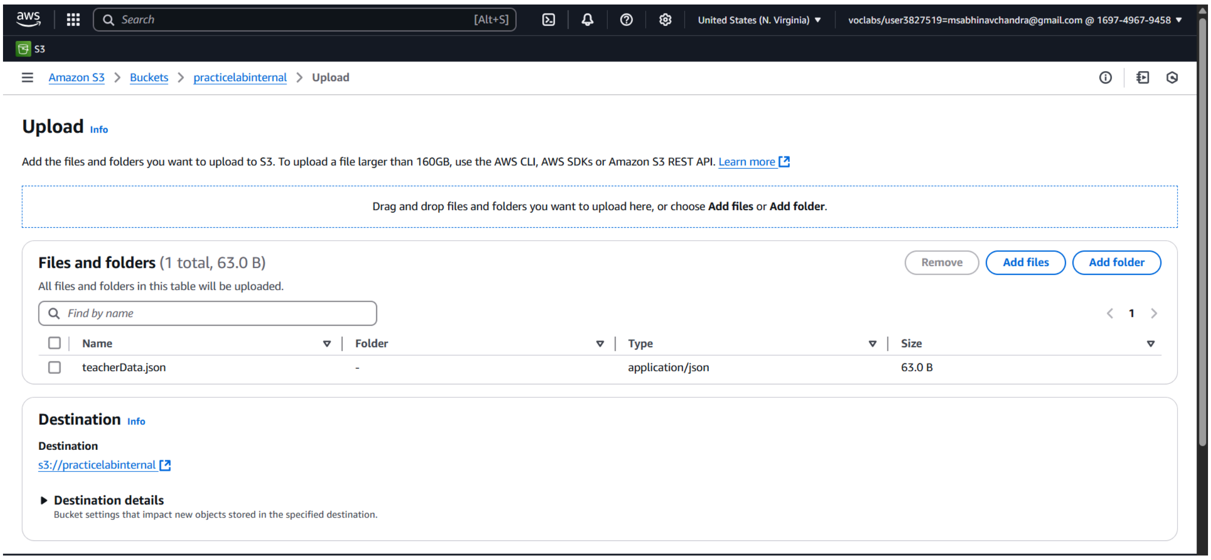
Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

No objects
You don't have any objects in this bucket.

Now Go back to S3 Bucket we have created and upload the Json file into it and Click on Upload.
Create a json data which includes that key primary or partition key which exists in the dynamoDB,
And upload it eg:
{

```
  "teacher_id": "T12345",
  "name": "John Doe"
}
```



aws
Search [Alt+S]
United States (N. Virginia) ▾
voclabs/user3827519=msabhinavchandra@gmail.com @ 1697-4967-9458 ▾
S3
Amazon S3 > Buckets > practicelabinternal > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 total, 63.0 B)

Remove
Add files
Add folder

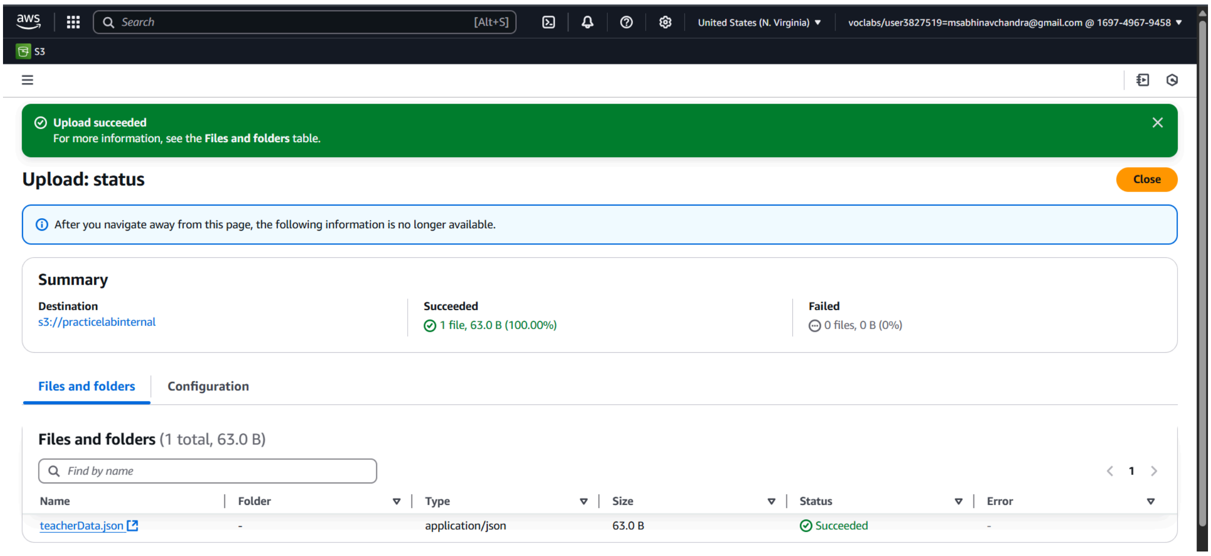
All files and folders in this table will be uploaded.

□	Name	Folder	Type	Size
□	teacherData.json	-	application/json	63.0 B

Destination Info

Destination
s3://practicelabinternal

▸ **Destination details**
Bucket settings that impact new objects stored in the specified destination.



aws
Search [Alt+S]
United States (N. Virginia) ▾
voclabs/user3827519=msabhinavchandra@gmail.com @ 1697-4967-9458 ▾
S3

Upload succeeded
For more information, see the **Files and folders** table.

Upload: status

After you navigate away from this page, the following information is no longer available.

Summary

Destination	Succeeded	Failed
s3://practicelabinternal	✓ 1 file, 63.0 B (100.00%)	0 files, 0 B (0%)

Files and folders
Configuration

Files and folders (1 total, 63.0 B)

Name	Folder	Type	Size	Status	Error
teacherData.json	-	application/json	63.0 B	✓ Succeeded	-

Once we click upload the lambda function is triggered and the result is directed to the Dynamo DB where we have our employees table with emp_id as the partition key.