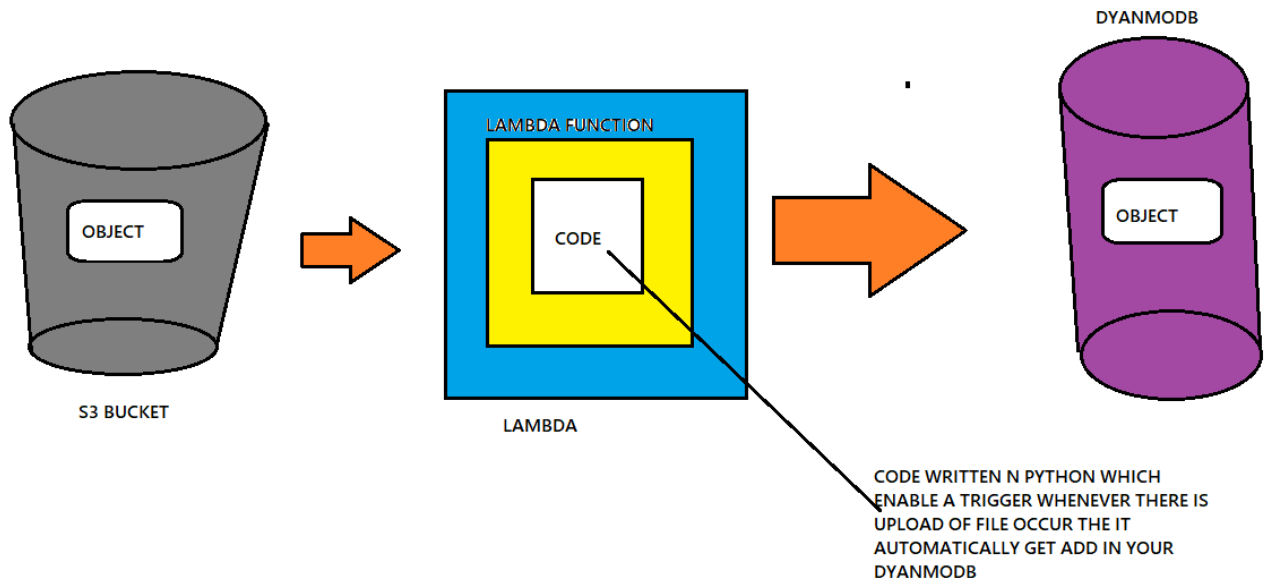


AIM-CONCEPT OF LAMBDA

ARCHITECTURE:-



STEPS-1

CREATE A LAMDA FUNCTION FIRST

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.

myFunctionName

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)

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

Basic information

Function name

Enter a name that describes the purpose of your function.

FUNCTION1

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)

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

Python 3.9



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

☐ 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**

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.

[View the role11 role](#) on the IAM console.

YOU NEED TO CREATE ROLE AND GIVE FULL ACCESS OF S3 BUCKET AND DYANMODB

▼ **Advanced settings**

☐ **Enable Code signing** [Info](#)
Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.

☐ **Enable function URL** [Info](#)
Use function URLs to assign HTTP(S) endpoints to your Lambda function.

☐ **Enable tags** [Info](#)
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources, track your AWS costs, and enforce attribute-based access control.

☐ **Enable VPC** [Info](#)
Connect your function to a VPC to access private resources during invocation.

Cancel **Create function**

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app consisting of a Lambda function with a full URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#)

[Start tutorial](#)

aws Services Search [Alt+S]

EC2 VPC S3 AWS Auto Scaling Simple Queue Service Simple Notification Service Key Management Service CloudTrail Amazon EventBridge

☑ **Successfully created the function FUNCTION1.** You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Diagram **Template**

FUNCTION1

Layers (0)

+ Add trigger + Add destination

Description

-

Last modified
7 seconds ago

Function ARN
arn:aws:lambda:us-east-1:905418179079:function:FUNCTION1

Function URL [Info](#)

Learn how to use cases in

Create app

In this tutorial you will learn how to:

- Build a simple web app
- Invoke your function through its function URL

STEP2- CREATE A S3 BUCKET

[Amazon S3](#) > [Buckets](#) > Create bucket

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 - New**

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](#)

MAKE SURE YOUR AZ SHOULD BE SAME

EC2VPCS3AWS Auto ScalingSimple Queue ServiceSimple Notification ServiceKey Management ServiceCloudTrailAmazon EventBridgeRDSIAM

☑ Successfully created bucket "bickykfbucket"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

View details

Account snapshot
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

General purpose bucketsDirectory buckets

General purpose buckets (1) Info
Buckets are containers for data stored in S3.
Find buckets by name
< 1 > ⚙

RefreshCopy ARNEmptyDeleteCreate bucket

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 - New
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
bickykfbucket
Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

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
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.


☐ 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.

☐ **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.

Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

EC2 VPC S3 AWS Auto Scaling Simple Queue Service Simple Notification Service Key Management Service CloudTrail Amazon EventBridge RDS IAM

☑ **Successfully created bucket "bickykfbucket"**
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

Account snapshot [View Storage Lens dashboard](#)
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[General purpose buckets](#) | Directory buckets

General purpose buckets (1) [Info](#)
Buckets are containers for data stored in S3.

[Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

< 1 > ⚙

Now go to lamda and add trigger and add these bucket

DiagramTemplate

FUNCTION1

Layers (0)

+ Add trigger

+ Add destination

Description

-

Last modified

4 minutes ago

Function ARN

`arn:aws:lambda:us-east-1:123456789079:function:FUNCTION1`

Function URL

[Info](#)

-

[Lambda](#) > Add trigger

Add trigger

Trigger configuration [Info](#)

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 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.

Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

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.

Cancel

Add

InfoTutorials

Learn how to implement use cases in AWS Lambda

Create a simple app

In this tutorial you will learn how to:

- Build a simple application consisting of a Lambda function with an S3 trigger that invokes a webpage
- Invoke your application through its API

[Learn more](#)

Start tutorial

Step 3- create a DYANMODB TABLE

The screenshot shows the Amazon DynamoDB console interface. At the top, there's a dark blue header with the text 'Database' and 'Amazon DynamoDB'. Below this, it says 'A fast and flexible NoSQL database service for any scale'. To the right, there's a 'Get started' section with a 'Create table' button. Below that, there's a 'Pricing' section. At the bottom, there's a navigation bar with various AWS services like VPC, S3, AWS Auto Scaling, Simple Queue Service, Simple Notification Service, Key Management Service, CloudTrail, Amazon EventBridge, and RDS.

Database

Amazon DynamoDB

A fast and flexible NoSQL database service for any scale

DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.

Get started

Create a new table to start exploring DynamoDB.

Create table

Pricing

How it works

[DynamoDB](#) > [Tables](#) > Create table

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.

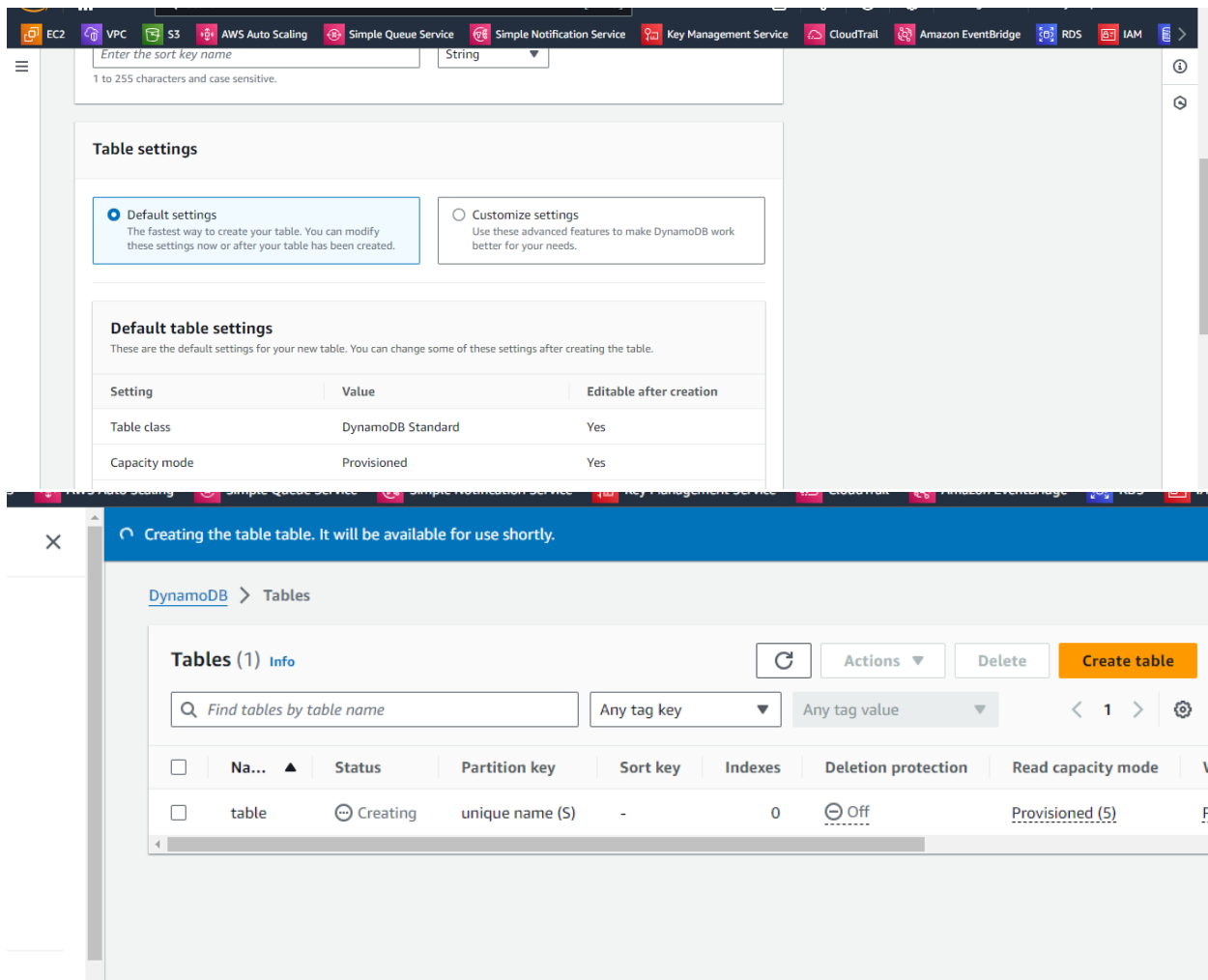
Table name
This will be used to identify your table.

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.

1 to 255 characters and case sensitive.

Sort key - *optional*



STEP4- NOW GO TO LAMDA AND ADD ONE code TO GET REQUIRE OUTPUT I.E
GETTING OBJECT UPDATE FROM S3 BUCKET TO DYANMO DB

Python 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']
dynamo_table = dynamodb.Table('Table1')
dynamo_table.put_item(
    Item={'unique': str(uuid4()),
          'Bucket': bucket_name,
          'Object': object_key,
          'Size': size,
          'Event': event_name,
          'EventTime': event_time

    })

```

CHANGE TABLE NAME , UNIQUE = PARTITION KEY , AND BUCKET NAME

```
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)
```

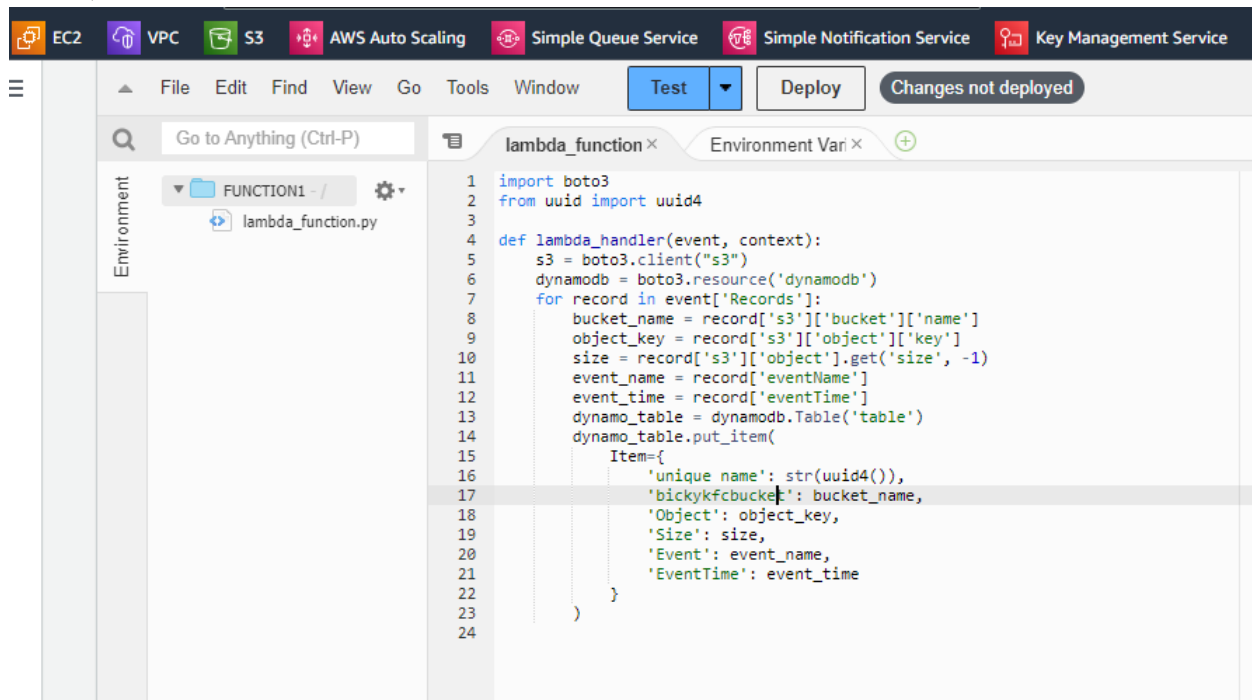
```
        event_name = record['eventName']
```

```
        event_time = record['eventTime']
```

```

dynamo_table = dynamodb.Table('table')
dynamo_table.put_item(
    Item={
        'unique name': str(uuid4()),
        'bickykfcbucket': bucket_name,
        'Object': object_key,
        'Size': size,
        'Event': event_name,
        'EventTime': event_time
    }
)

```



First save these and deploy it

Now add ANY OBJECT IN S3 BUCKET

The screenshot displays the AWS S3 console interface. At the top, a navigation bar includes icons for various AWS services: EC2, VPC, S3, AWS Auto Scaling, Simple Queue Service, Simple Notification Service, and Key Management Service. The main content area is titled 'Upload' and provides instructions on how to upload files, including a link to 'Learn more' about the S3 REST API. A dashed box indicates where to drag and drop files. Below this, a section titled 'Files and folders (1 Total, 1.6 MB)' shows a table with one file: 'AUTO SCALING.pdf'. A search bar and pagination controls are also present. A green banner at the bottom of the upload section states 'Upload succeeded' and 'View details below.' Below this banner, a summary table shows the upload details: Destination 's3://bickykfcbucket', Succeeded status with '1 file, 1.6 MB (100.00%)', and Failed status with '0 files, 0 B (0%)'. The 'Files and folders' tab is selected, showing a detailed table of the uploaded file.

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK 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, 1.6 MB) Remove Add files Add folder

All files and folders in this table will be uploaded.

< 1 >

<input type="checkbox"/>	Name	Folder	Type
<input type="checkbox"/>	AUTO SCALING.pdf	-	application/pdf

Destination [Info](#)

Upload succeeded
View details below.

Destination	Succeeded	Failed
s3://bickykfcbucket	✔ 1 file, 1.6 MB (100.00%)	✖ 0 files, 0 B (0%)

Files and folders | Configuration

Files and folders (1 Total, 1.6 MB)

< 1 >

Name	Folder	Type	Size	Status	Error
AUTO SCALI...	-	application/...	1.6 MB	✔ Succeeded	-

EC2VPCS3AWS Auto ScalingSimple Queue ServiceSimple Notification ServiceKey Management ServiceCloudTrailAmazon EventBridge

Uploading

Total remaining: 1 file: 1.0 MB(100.00%)

Estimated time remaining: calculating...

Transfer rate: 0 B/s

Files and folders

Configuration

Files and folders (1 Total, 1.0 MB)

Find by name

Name	Folder	Type	Size	Status	Error
BICKY TASK ...	-	application/...	1.0 MB	Pending	-

awsServicesSearch[Alt+S]

EC2VPCS3AWS Auto ScalingSimple Queue ServiceSimple Notification ServiceKey Management ServiceCloudTrailAmazon EventBridge

Upload succeeded

View details below.

Destination

s3://bickykfbucket

Succeeded

1 file, 1.0 MB (100.00%)

Failed

0 files, 0 B (0%)

Files and folders

Configuration

Files and folders (1 Total, 1.0 MB)

Find by name

Name	Folder	Type	Size	Status	Error
BICKY TASK ...	-	application/...	1.0 MB	Succeeded	-

NOW CHECK DYANMO>TABLE>EXPLORE ITEM DB YOUR OBJECT WILL UODATE THERE

EC2VPCSS3AWS Auto ScalingSimple Queue ServiceSimple Notification ServiceKey Management ServiceCloudTrailAmazon EventBridgeRDS

bickykfcbucketInfo

ObjectsPropertiesPermissionsMetricsManagementAccess Points

Objects (2) Info

Copy S3 URICopy URLOpenDownloadDeleteActionsCreate folderUpload

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](#)

Find objects by prefix

< 1 > ⚙

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	AUTO SCALING.pdf	pdf	March 20, 2024, 02:45:52 (UTC+05:30)	1.6 MB	Standard
<input type="checkbox"/>	BICKY TASK 5.pdf	pdf	March 20, 2024, 02:41:04 (UTC+05:30)	1.0 MB	Standard

Services

Search

[Alt+S]

VPCSS3AWS Auto ScalingSimple Queue ServiceSimple Notification ServiceKey Management ServiceCloudTrailAmazon EventBridgeRDSIAMDynamoDBLambda

DB

Find tables by table name

< 1 > ⚙

table

ScanQuery

Select a table or index

Table - table

Select attribute projection

All attributes

Filters

RunReset

Completed. Read capacity units consumed: 0.5

Items returned (2)

< 1 > ⚙

<input type="checkbox"/>	unique name (String)	bickykfcbucket	bucket	Event	EventTime	Object
<input type="checkbox"/>	da5c6148-c3b7-44af-96...	bickykfc buc...	bickykfc buc...	ObjectCreat...	2024-03-1...	BICKY+TAS...
<input type="checkbox"/>	bb6063e0-2813-49e7-9...	bickykfcbucket	bickykfc buc...	ObjectCreat...	2024-03-1...	AUTO+SCA...

Items returned (2)

< 1 > ⚙

<input type="checkbox"/>	unique name (String)	bickykfcbucket	bucket	Event	EventTime	Object
<input type="checkbox"/>	da5c6148-c3b7-44af-96...	bickykfc buc...	bickykfc buc...	ObjectCreat...	2024-03-1...	BICKY+TAS...
<input type="checkbox"/>	bb6063e0-2813-49e7-9...	bickykfcbucket	bickykfc buc...	ObjectCreat...	2024-03-1...	AUTO+SCA...