

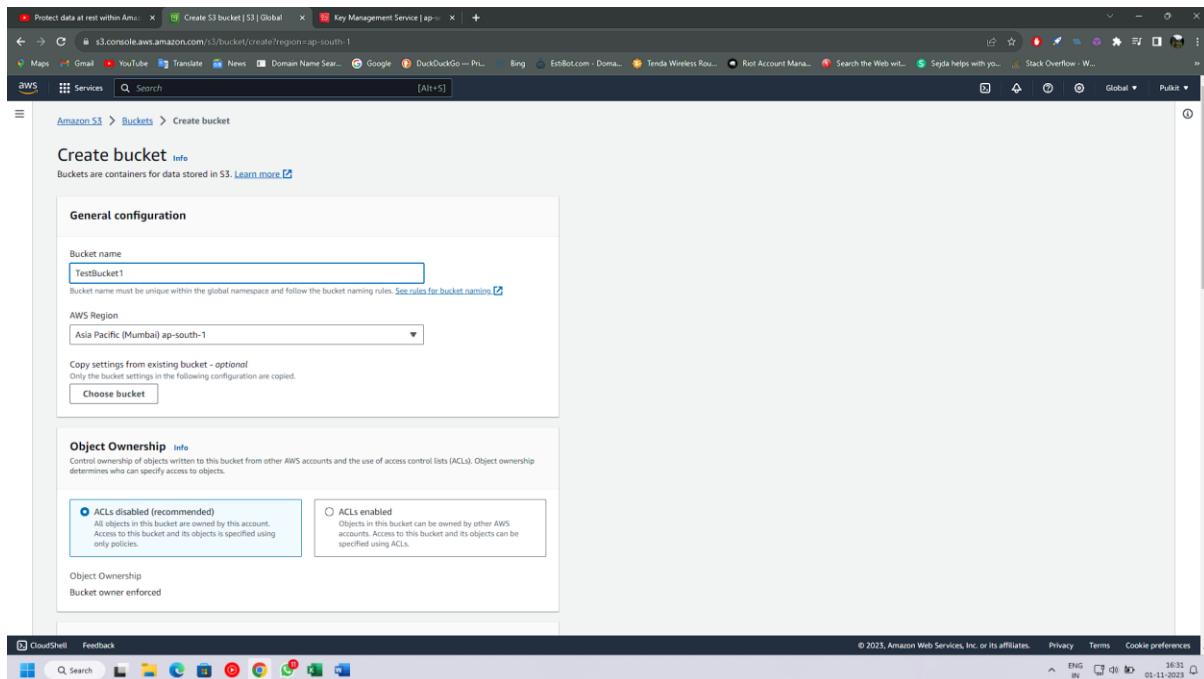
😊 Protect data at rest within Amazon S3 data centres by using AWS KMS

STEP 1:

1. Log in to Console
2. Then go to IAM and Create a Root user (We can use existing user as well) and give it a permission for S3 read only.
3. Then log into the Root user using a different browser.

STEP 2:

1. Create a S3 bucket in main account and keep the settings to default.



2. Go to Root account and open S3 in it too. We'll see the same bucket that we created in main account.
3. Now go back to main account and upload a file to S3 bucket from our system. And we'll see the file in the bucket.

Amazon S3 > Buckets > testbucket-120

testbucket-120 [Info](#)

Objects Properties Permissions Metrics Management Access Points

Objects (1)

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](#)

C Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
Build a CRUD Serverless API with AWS Lambda.pdf	pdf	November 1, 2023, 16:36:22 (UTC+05:30)	5.2 MB	Standard

4. And if we go to the Root user we'll find the same file in it. If we want to open it we can see the content in the file present.

testbucket-120 - S3 bucket | S3 | + https://s3.console.aws.amazon.com/s3/buckets/testbucket-120?region=ap-south-1&tab=objects

Lenovo Computer Science... Maps Gmail YouTube Translate News Domain Name Search Google DarkDarkGo — Private Bing EstBot.com - Doma... WHOIS search results Free Whois Lookup... Whois API | Whois... Search the Web mit...

aws Services Search [Alt+S]

Amazon S3

Amazon S3 > Buckets > testbucket-120

testbucket-120 [Info](#)

Objects Properties Permissions Metrics Management Access Points

Objects (1)

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](#)

C Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
Build a CRUD Serverless API with AWS Lambda.pdf	pdf	November 1, 2023, 16:36:22 (UTC+05:30)	5.2 MB	Standard

Screenshot of the AWS S3 console showing the properties of the file "Build a CRUD Serverless API with AWS Lambda.pdf".

Object overview

- Owner: 313947572c23742f9d25bf47eadfaa2ecde5bc5d81c620e510bc76962e5df997
- AWS Region: Asia Pacific (Mumbai) ap-south-1
- Last modified: November 1, 2023, 16:36:22 (UTC+05:30)
- Size: 5.2 MB
- Type: pdf
- Key: Build a CRUD Serverless API with AWS Lambda.pdf
- S3 URI: <https://testbucket-120.s3.ap-south-1.amazonaws.com/Build+a+CRUD+Serverless+API+with+AWS+Lambda.pdf>
- Amazon Resource Name (ARN): arn:aws:s3:::testbucket-120/Build a CRUD Serverless API with AWS Lambda.pdf
- Entity tag (ETag): cc12d3ddaaad394f657ff75a222f1b44c
- Object URL: <https://testbucket-120.s3.ap-south-1.amazonaws.com/Build+a+CRUD+Serverless+API+with+AWS+Lambda.pdf>

Object management overview

The following bucket properties and object management configurations impact the behavior of this object.

Bucket properties	Management configurations
Bucket Versioning	Replication status

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 16:39 01-11-2023

Screenshot of a PDF document titled "Build a CRUD Serverless API with AWS Lambda, API Gateway and a DynamoDB from Scratch".

Build a CRUD Serverless API with AWS Lambda, API Gateway and a DynamoDB from Scratch

The above photo represent the block diagram of what is going to happen during this process build.

Create DynamoDB service

- Log in to AWS and navigate to the **DynamoDB** service.
- Then click **Create table**.
- Choose a primary key (I used **studentId**), give your table a name (I called mine **students-data**), and then click **Create**.
- The primary key you choose must uniquely identify each item

ENG IN 16:40 01-11-2023

5. And if we go back to Root user we can do the same in it too.

Object overview

Owner: 313947572c23742f9d5bf47eadfaa2ecde5bc5d81c620e310bc76962e5df997

AWS Region: Asia Pacific (Mumbai) ap-south-1

Last modified: November 1, 2023, 16:36:22 (UTC-05:30)

Size: 5.2 MB

Type: pdf

Key: Build a CRUD Serverless API with AWS Lambda.pdf

SS URI: s3://testbucket-120/Build a CRUD Serverless API with AWS Lambda.pdf

Amazon Resource Name (ARN): arnaws3::testbucket-120/Build a CRUD Serverless API with AWS Lambda.pdf

Entity tag (Etag): cc12d3ddaa394f657ff75a222f1b44c

Object URL: https://testbucket-120.s3.ap-south-1.amazonaws.com/Build+a+CRUD+Serverless+API+with+AWS+Lambda.pdf

Build a CRUD Serverless API with AWS Lambda, API Gateway and a DynamoDB from Scratch

The above photo represent the block diagram of what is going to happen during this process build.

Create DynamoDB service

1. Log in to AWS and navigate to the **DynamoDB** service.
2. Then click **Create table**.
3. Choose a primary key (I used **studentId**), give your table a name (I called mine **students-data**), and then click **Create**.
4. The primary key you choose must uniquely identify each item in the table so that no two objects can share the same key. This

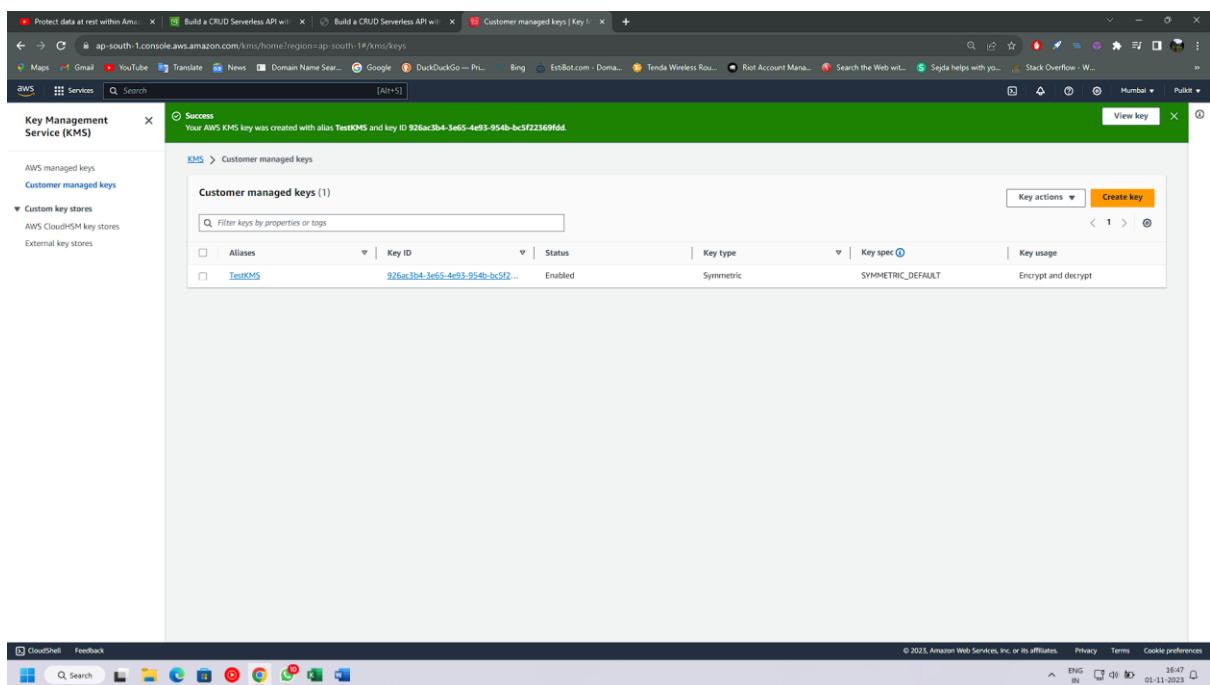
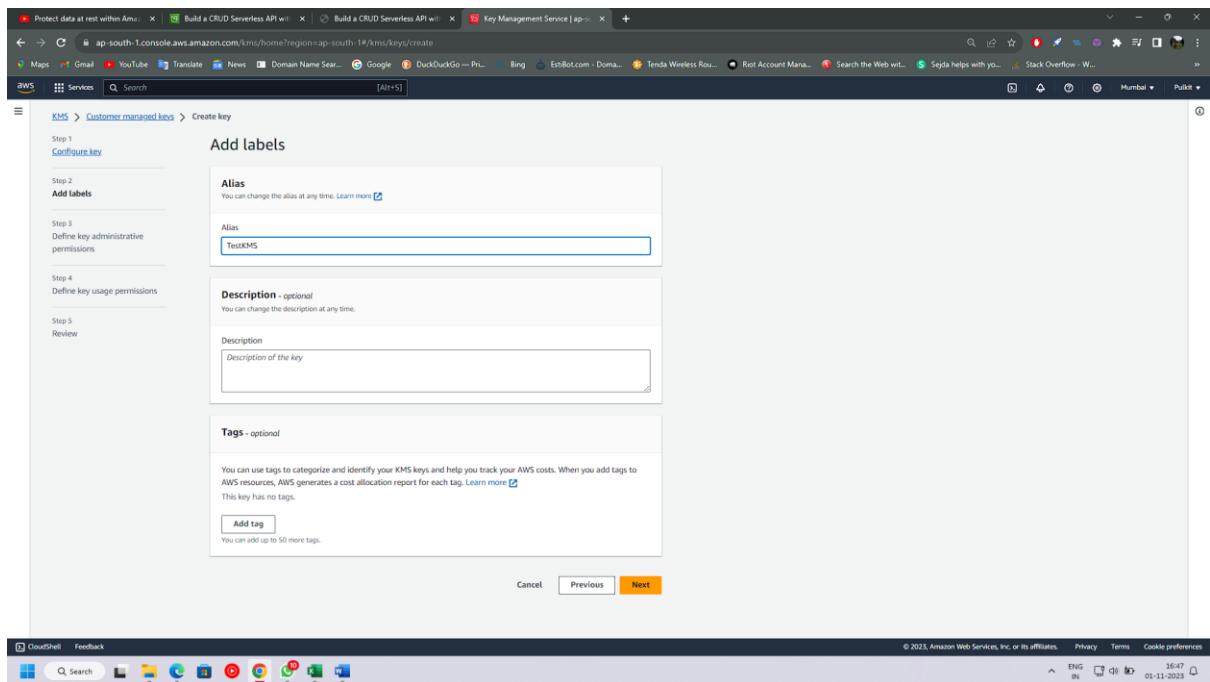
STEP 3:

1. Now go back to main user and search Key Management Service (KMS) and create a key to secure our file in S3.

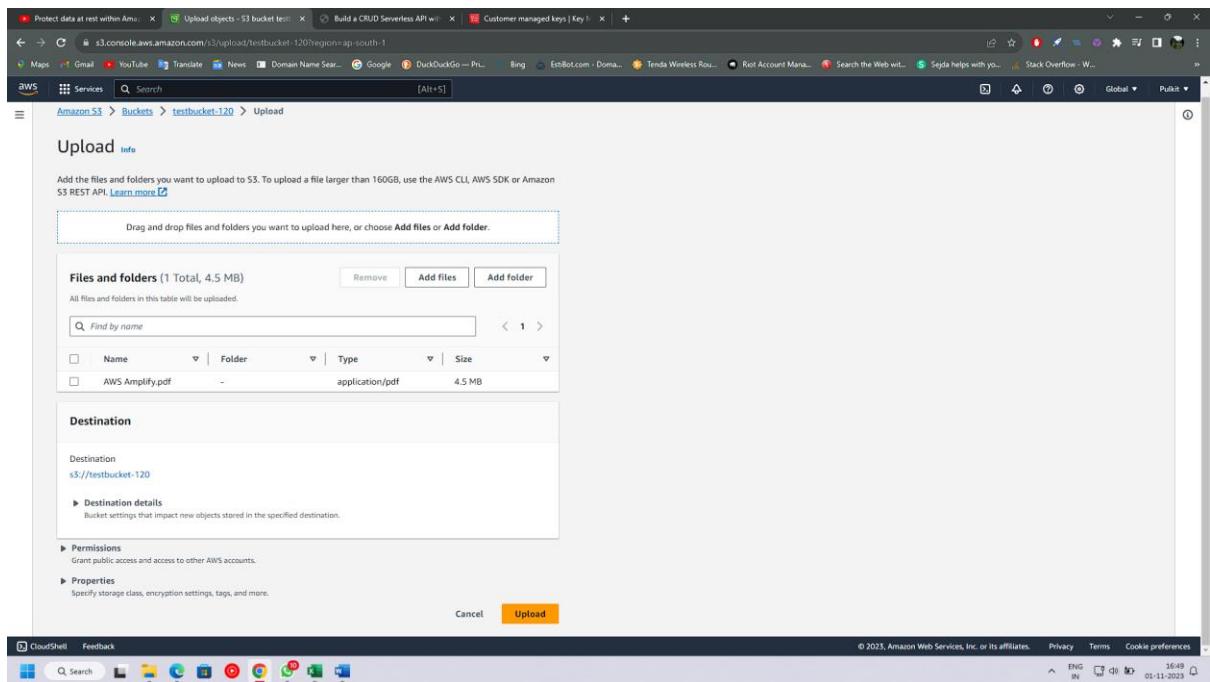
The screenshot shows the AWS KMS homepage. On the left, there's a sidebar with navigation links: AWS managed keys, Customer managed keys, Custom key stores, AWS CloudHSM key stores, and External key stores. The main content area has a heading "AWS Key Management Service" and sub-headings "Easily create keys and control encryption across AWS and beyond". Below this, there's a section titled "How it works" with a detailed description of its central management and secure storage capabilities. Another section, "Benefits and features", lists "Fully managed" and "Centralized key management". To the right, there's a "Get started now" box with a "Create a key" button, a "Pricing" section, and a "Getting started" section with links to "What is AWS Key Management Service?", "Getting started with KMS", and "Working with KMS Keys". At the bottom, there's a "More resources" section with links to "Documentation", "FAQ", and "KMS forum". The footer contains standard AWS links and a timestamp.

The screenshot shows the "Configure key" step of the AWS KMS key creation wizard. The left sidebar shows steps: Step 1 (Configure key), Step 2 (Add labels), Step 3 (Define key administrative permissions), Step 4 (Define key usage permissions), and Step 5 (Review). The main form is titled "Configure key" and has two main sections: "Key type" and "Key usage". Under "Key type", "Symmetric" is selected, described as "A single key used for encrypting and decrypting data or generating and verifying HMAC codes". Under "Key usage", "Encrypt and decrypt" is selected, described as "Use the key only to encrypt and decrypt data". There's also an option for "Generate and verify MAC". At the bottom, there are "Advanced options" and "Cancel" and "Next" buttons. The footer is identical to the previous screenshot.

2. Just give it a name then keep everything default and create a KMS.



3. Afterwards go back to S3 and upload another file.
4. Once we've uploaded the file we'll see Properties option at the last click on it.



5. Then scroll down a little and we'll see a option named Server-Side Encryption and on it choose the option below. (And choose the KMS which we created earlier)
6. Then click on upload.

Server-side encryption [Info](#)

Server-side encryption protects data at rest.

Server-side encryption

Do not specify an encryption key
The bucket settings for default encryption are used to encrypt objects when storing them in Amazon S3.

Specify an encryption key
The specified encryption key is used to encrypt objects before storing them in Amazon S3.

Encryption settings [Info](#)

Use bucket settings for default encryption

Override bucket settings for default encryption

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](#).

AWS KMS key [Info](#)

Choose from your AWS KMS keys

Enter AWS KMS key ARN

Available AWS KMS keys

arn:aws:kms:ap-south-1:878893308172:key/926...	▲		Create a KMS key
arn:aws:kms:ap-south-1:878893308172:key/926ac3b4-3e65-4e93-954b-bc5f22369fdd	TestKMS		ed in this bucket ettings from the bucket default encryption Learn more
arn:aws:kms:ap-south-1:878893308172:key/926ac3b4-3e65-4e93-954b-bc5f22369fdd	aws/s3		

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

7. Now if we open it from our main account it'll get open and it is readable.

The screenshot shows the AWS S3 console with the following details:

Object overview

- Owner: 313947572c23742f9d25bf47eadfaa2ecde5bc5d81c620e310bc76962e5df997
- AWS Region: Asia Pacific (Mumbai) ap-south-1
- Last modified: November 1, 2023, 16:53:16 (UTC+05:30)
- Size: 4.5 MB
- Type: pdf
- Key: AWS Amplify.pdf

S3 URI: s3://testbucket-120/AWS Amplify.pdf

Amazon Resource Name (ARN): arn:aws:s3:::testbucket-120/AWS Amplify.pdf

Entity tag (Etag): 187ae8b8dd1c7f08f722756c50ab7b01

Object URL: https://testbucket-120.s3.ap-south-1.amazonaws.com/AWS+Amplify.pdf

Object management overview

The following bucket properties and object management configurations impact the behavior of this object.

Bucket properties

Management configurations

Replication status

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 16:53 01-11-2023

The screenshot shows a presentation slide titled "AWS Amplify". The slide content includes:

Agenda - AWS Amplify

- AWS Amplify is a complete solution that lets frontend web and mobile developers easily build, ship, and host full-stack applications on AWS, with the flexibility to leverage the breadth of AWS services as use cases evolve. No cloud expertise needed.
- Developer framework and web hosting service to deploy and build mobile and web application on AWS

Components

- UI components: There are a lot of prebuilt UI components available to use like login forms etc. Good about this is all the UI components are customizable, so you can change them according to your needs.
- Libraries: helps integrate and interact with AWS services. These will help to add features to the application like push notification services, authentication, etc.
- CLI: command-line interface to manage your AWS managed backends by configuring AWS services through CLI commands.

8. Now if we go to our Root Account we'll see the same in it too.

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Objects (2)

Actions

Create folder

Upload

Name	Type	Last modified	Size	Storage class
AWS Amplify.pdf	pdf	November 1, 2023, 16:53:16 (UTC+05:30)	4.5 MB	Standard
Build a CRUD Serverless API with AWS Lambda.pdf	pdf	November 1, 2023, 16:36:22 (UTC+05:30)	5.2 MB	Standard

9. But if we try to open this file then we'll get this message (Access Denied) which means that our file is secured.

```
<Error>
<Code>AccessDenied</Code>
<Message>The action 'Decrypt' is not authorized to perform the requested operation: kms:Decrypt on resource: arn:aws:kms:ap-south-1:878893308172:key/926ac304-3e65-4e93-9540-bc5f22369fdd because no identity-based policy allows the kms:Decrypt action message.
<RequestId>C90F2F2ZJNAP90B83H</RequestId>
<HostId>Pq4tVQz1byYv0d1uXVC1AhdzUtrgP0</HostId>
<RequestId>C90F2F2ZJNAP90B83H</RequestId>
<HostId>Pq4tVQz1byYv0d1uXVC1AhdzUtrgP0</HostId>
```

10. This is how we secure our file.

LAST STEP:

1. If we want to give permission to the root user to access that file which we've secured.
2. Then we need to go back to KMS and select the user which we created.

The screenshot shows the AWS KMS Customer managed keys page. A success message at the top states: "Success Your AWS KMS key was created with alias TestKMS and key ID 926ac3b4-3e65-4e93-954b-bc5ff22369fd." Below this, a table lists the key: "TestKMS" with Key ID "926ac3b4-3e65-4e93-954b-bc5ff22369fd", Status "Enabled", Key type "Symmetric", Key spec "SYMMETRIC_DEFAULT", and Key usage "Encrypt and decrypt".

3. Open it and scroll to down to a option which says Key User and select the root user and add it.

The screenshot shows the AWS KMS Key policy page. In the "Key users" section, a modal dialog titled "Add key users" is open. It lists "Key users (1/30)" and shows one item: "pukit_IAM" selected with a checked checkbox. Other items listed include "pukit_IAM2", "AmazonS...", "AmazonS...", "aws-elasti...", "AWS-Quic...", "AWS-Quic...", "AWServi...", "AWServi...", and "AWServi...". At the bottom of the modal are "Cancel" and "Add" buttons.

4. Now if we go back to Root User and open that same file again then we'll see that we're able to read the contents of that file.

