Task - Work on S3 managed keys / aws customer-managed keys / encrypting files before uploading to S3 using customer-managed aws keys and also using S3 managed keys.

AWS Key Management Service (KMS)

AWS Key Management Service (KMS) makes it easy for us to create and manage cryptographic keys and control their use across a wide range of AWS services and in your applications. AWS KMS is a secure and resilient service that uses hardware security modules that have been validated under FIPS 140-2, or are in the process of being validated, to protect your keys. AWS KMS is integrated with AWS CloudTrail to provide us with logs of all key usage to help meet your regulatory and compliance needs.

In simple word, it can easily create and control the key used to encrypt or digitally sign our data

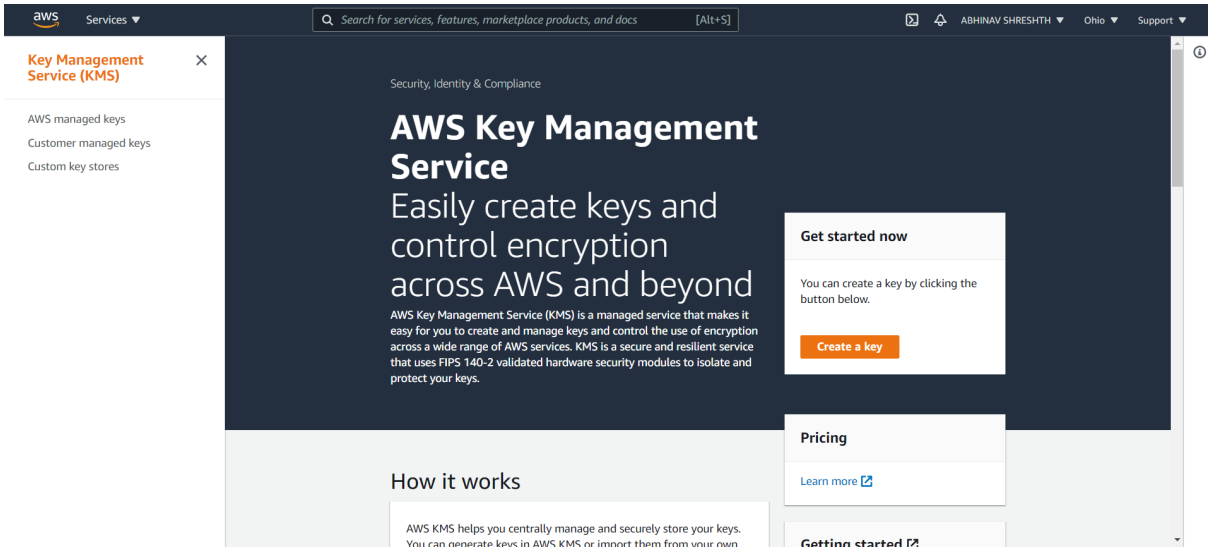
Benefits:

- Fully managed
  We control access to your encrypted data by defining permissions to use keys while AWS KMS enforces your permissions and handles the durability and physical security of your keys.

- Centralized key management:
  AWS KMS presents a single control point to manage keys and define policies consistently across integrated AWS services and your own applications. We can easily create, import, rotate, delete, and manage permissions on keys from the AWS Management Console or by using the AWS SDK or CLI.

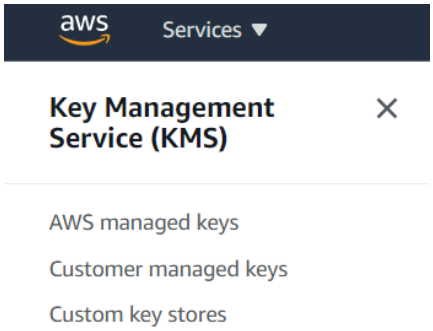
- Digitally sign data
  AWS KMS enables us to perform digital signing operations using asymmetric key pairs to ensure the integrity of your data. Recipients of digitally signed data can verify the signatures whether they have an AWS account or not.

This is the first view of AWS KMS



Here we can see in aws kms there is two methods -

1. Customer managed keys
2. Aws managed keys

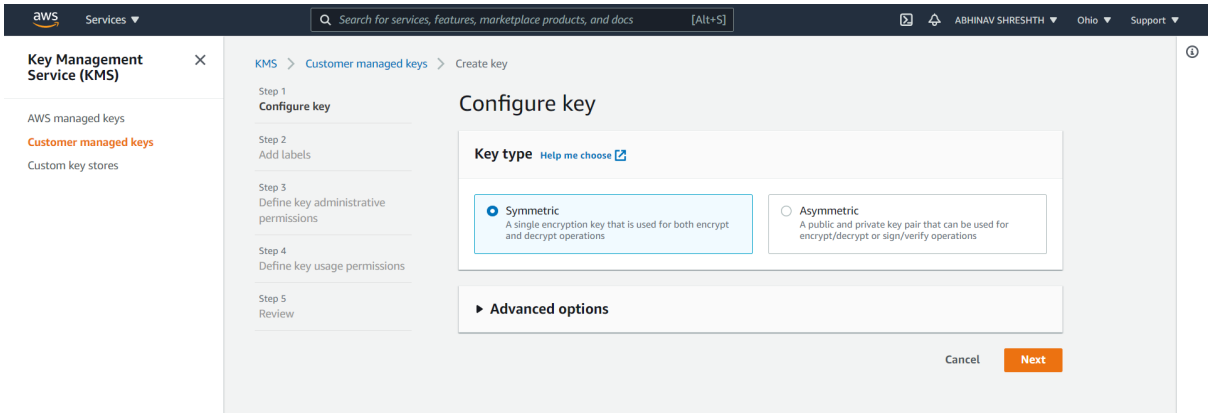


Difference between Aws managed keys and customer-managed keys

	AWS-managed CMK	**Customer-managed CMK**
Creation	AWS generated on the customer's behalf	Customer-generated
Rotation	Once every three years automatically	Once a year automatically through opt-in or on-demand manually
Deletion	Can't be deleted	Can be deleted
Scope of use	Limited to a specific AWS service	Controlled via KMS/IAM policy
Key Access Policy	AWS managed	Customer managed
User AccessManagement	IAM policy	IAM policy

Encryption is the process of taking a message and scrambling it's contents so that only certain people can look at your message.

There are two types of encryption: symmetric and asymmetric encryption.



Symametic Encryption: according to Wikipedia, Symmetric encryption is a type of encryption where only one key (a secret key) is used to both encrypt and decrypt electronic information. The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process.

The main advantage of symmetric encryption over asymmetric encryption is that it is fast and efficient for large amounts of data; the disadvantage is the need to keep the key secret - this can be especially challenging where encryption and decryption take place in different locations.

Asymmetric Encryption: according to google Asymmetric encryption is a type of encryption that uses two separates yet mathematically related keys to encrypt and decrypt data

Master key

A key created by AWS KMS can only be used within the AWS KMS service. The master key is commonly used to encrypt data keys so that the encrypted key can be securely stored by your service. However, AWS KMS master keys can also be used to encrypt or decrypt arbitrary chunks of data that are no greater than 4 KiB. Master keys are categorized as either customer-managed keys or AWS managed keys. Customer managed keys are created by a customer for use by a service or application. AWS managed keys are the default keys used by AWS services that support encryption.

Data key

A symmetric key generated by AWS KMS for your service. Inside of your service or custom application, the data key is used to encrypt or decrypt data. It can be considered a resource by a service or

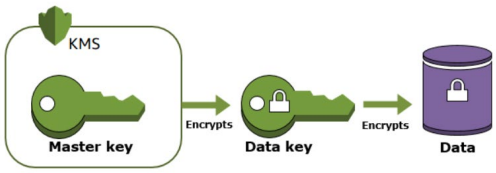
application, or it can simply be metadata associated with the encrypted data.

Envelope Encryption

When you encrypt your data, your data is protected, but you have to protect your encryption key. One strategy is to encrypt it. Envelope encryption is the practice of encrypting plaintext data with a data key, and then encrypting the data key under another key.

You can even encrypt the data encryption key under another encryption key, and encrypt that encryption key another encryption key. But, eventually, one key must remain in plaintext so you can decrypt the keys and your data. This top level plaintext key encryption key is known as the master key

And AWS KMS helps you to protect your master keys by storing and managing them securely. Master keys stored in AWS KMS, known as customer master keys (CMKs), never leave the AWS KMS FIPS validated hardware security modules unencrypted. To use an AWS KMS CMK, you must call AWS KMS.



Key policy

When you create a CMK, you determine who can use and manage that CMK. These permissions are contained in a document called the key policy. You can use the key policy to add, remove, or change permissions at any time for a customer-managed CMK. But you cannot edit the key policy for an AWS managed CMK. For more information, see Using key policies in AWS KMS.

DOC – 5 | 29/05/21 – BY ABHINAV SHRESHTH

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] ABHINAV SHRESHTH ▼ Ohio ▼ Support ▼

Key Management Service (KMS)

- AWS managed keys
- Customer managed keys
- Custom key stores

KMS > Customer managed keys > Create key

Step 1 **Configure key**
Step 2 Add labels
Step 3 Define key administrative permissions
Step 4 Define key usage permissions
Step 5 Review

Configure key

Key type Help me choose

- Symmetric — A single encryption key that is used for both encrypt and decrypt operations
- Asymmetric — A public and private key pair that can be used for encrypt/decrypt or sign/verify operations

Advanced options

Key material origin
Help me choose

- KMS
- External
- Custom key store (CloudHSM)

You can import a symmetric 256-bit key from your key management infrastructure into KMS and use it like any other customer master key.

☑ I understand the security, availability, and durability implications of using an imported key.

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S]

Key Management Service (KMS)

- AWS managed keys
- Customer managed keys
- Custom key stores

KMS > Customer managed keys > Create key

Step 1 Configure key
Step 2 **Add labels**
Step 3 Define key administrative permissions
Step 4 Define key usage permissions
Step 5 Review

Add labels

Alias
You can change the alias at any time. Learn more

Alias
cmk1-ext

Description - *optional*
You can change the description at any time.

Description - *optional*
cmk1-ext description

Tags - *optional*

You can use tags to categorize and identify your CMKs and help you track your AWS costs. When you add tags to AWS resources, AWS generates a cost allocation report for each tag. Learn more

This key has no tags.

Add tag

You can add up to 50 more tags.

Cancel Previous Next

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] ABHINAV SHRESHTH ▼ Ohio ▼

Key Management Service (KMS)

AWS managed keys
Customer managed keys
Custom key stores

KMS > Customer managed keys > Create key

Step 1
Configure key

Step 2
Add labels

Step 3
Define key administrative permissions

Step 4
Define key usage permissions

Step 5
Review

Define key administrative permissions

Key administrators

Choose the IAM users and roles who can administer this key through the KMS API. You may need to add additional permissions for the users or roles to administer this key from this console. Learn more

1

Name	Path	Type
AWSServiceRoleForEMRCleanup	/aws-service-role/elasticmapreduce.amazonaws.com/	Role
AWSServiceRoleForSupport	/aws-service-role/support.amazonaws.com/	Role
AWSServiceRoleForTrustedAdvisor	/aws-service-role/trustedadvisor.amazonaws.com/	Role
EMR_DefaultRole	/	Role
EMR_EC2_DefaultRole	/	Role
EMR_Notebooks_DefaultRole	/	Role

Key deletion

Allow key administrators to delete this key.

Cancel Previous Next

Feedback English (US) ▼ Privacy Policy Terms of Use

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] ABHINAV SHRESHTH ▼ Ohio ▼

Key Management Service (KMS)

AWS managed keys
Customer managed keys
Custom key stores

Step 2
Add labels

Step 3
Define key administrative permissions

Step 4
Define key usage permissions

Step 5
Review

This account

Select the IAM users and roles that can use the CMK in cryptographic operations. Learn more

1

Name	Path	Type
AWSServiceRoleForEMRCleanup	/aws-service-role/elasticmapreduce.amazonaws.com/	Role
AWSServiceRoleForSupport	/aws-service-role/support.amazonaws.com/	Role
AWSServiceRoleForTrustedAdvisor	/aws-service-role/trustedadvisor.amazonaws.com/	Role
EMR_DefaultRole	/	Role
EMR_EC2_DefaultRole	/	Role
EMR_Notebooks_DefaultRole	/	Role

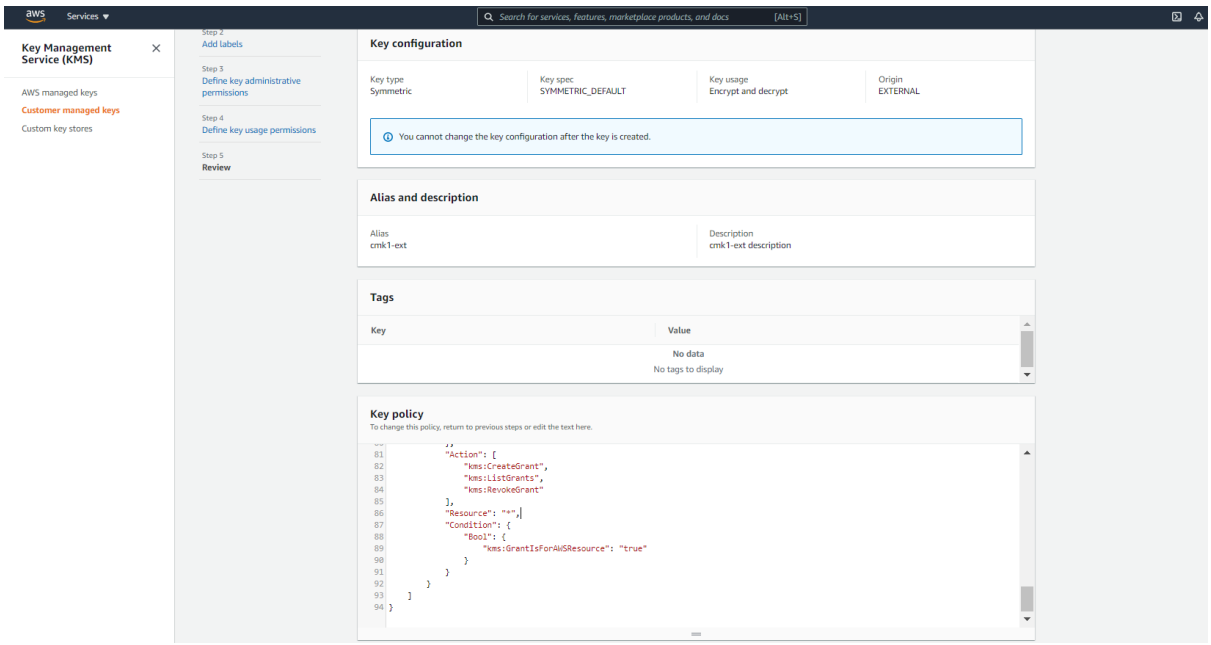
Other AWS accounts

Specify the AWS accounts that can use this key. Administrators of the accounts you specify are responsible for managing the permissions that allow their IAM users and roles to use this key. Learn more

Add another AWS account

Cancel Previous Next

Feedback English (US) ▼ Privacy Policy Terms of Use



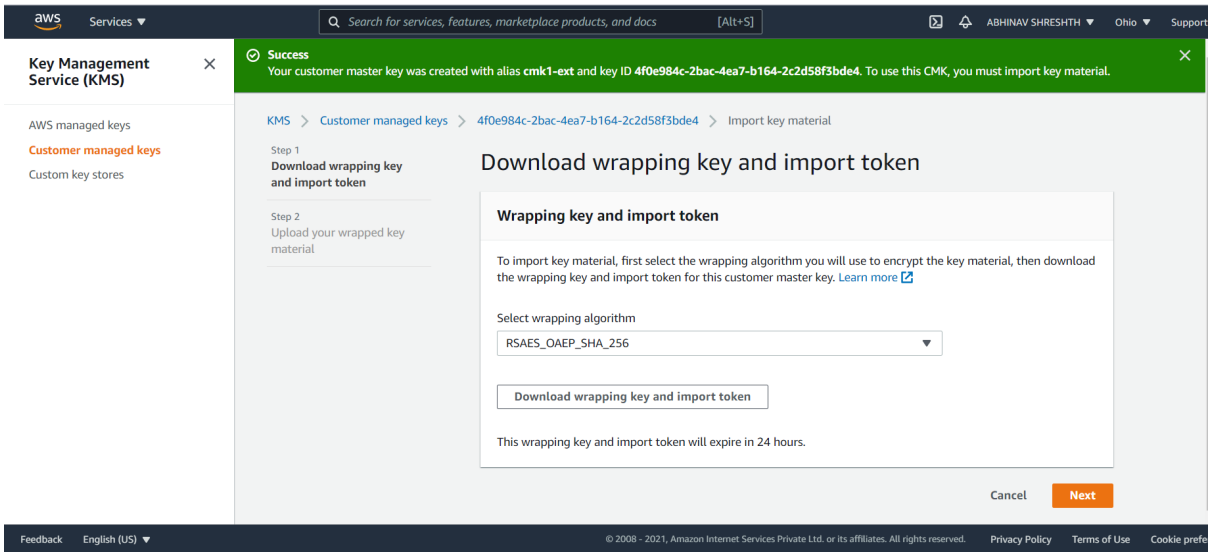

```
{
    "Id": "key-consolepolicy-3",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::431440931115:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow access for Key Administrators",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::431440931115:role/aws-service-
role/elasticmapreduce.amazonaws.com/AWSServiceRoleForEMRCleanup",
                    "arn:aws:iam::431440931115:role/aws-service-
role/support.amazonaws.com/AWSServiceRoleForSupport",
```

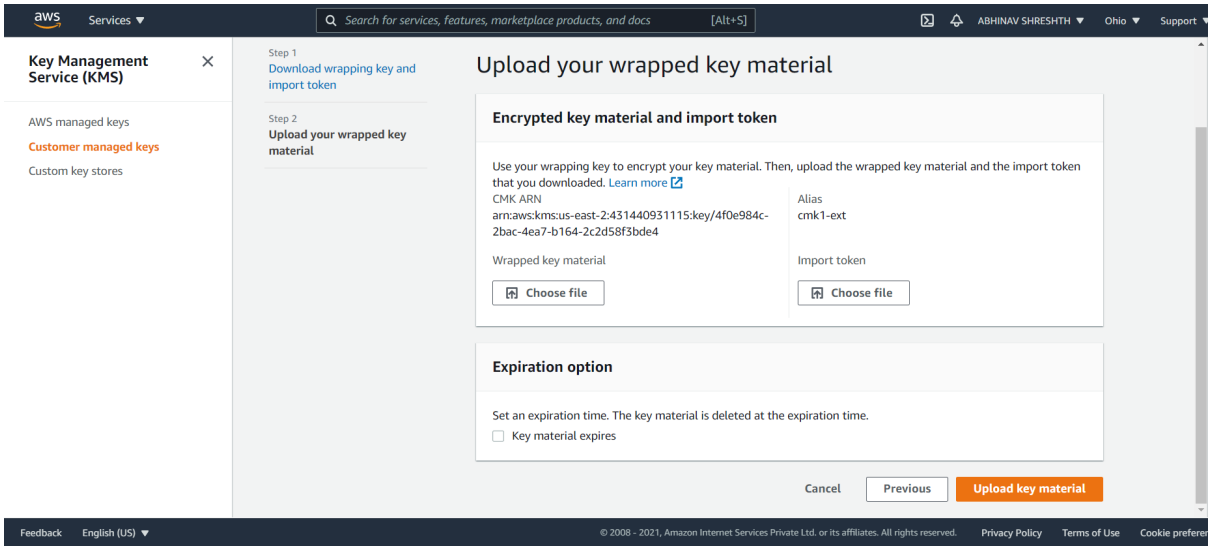
```
                    "arn:aws:iam::431440931115:role/aws-service-
role/trustedadvisor.amazonaws.com/AWSServiceRoleForTrustedAdvisor",

"arn:aws:iam::431440931115:role/EMR_DefaultRole",

"arn:aws:iam::431440931115:role/EMR_EC2_DefaultRole",

"arn:aws:iam::431440931115:role/EMR_Notebooks_DefaultRole"
                ]
            },
            "Action": [
                "kms:Create*",
                "kms:Describe*",
                "kms:Enable*",
                "kms:List*",
                "kms:Put*",
                "kms:Update*",
                "kms:Revoke*",
                "kms:Disable*",
                "kms:Get*",
                "kms:Delete*",
                "kms:ImportKeyMaterial",
                "kms:TagResource",
                "kms:UntagResource",
                "kms:ScheduleKeyDeletion",
                "kms:CancelKeyDeletion"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
```

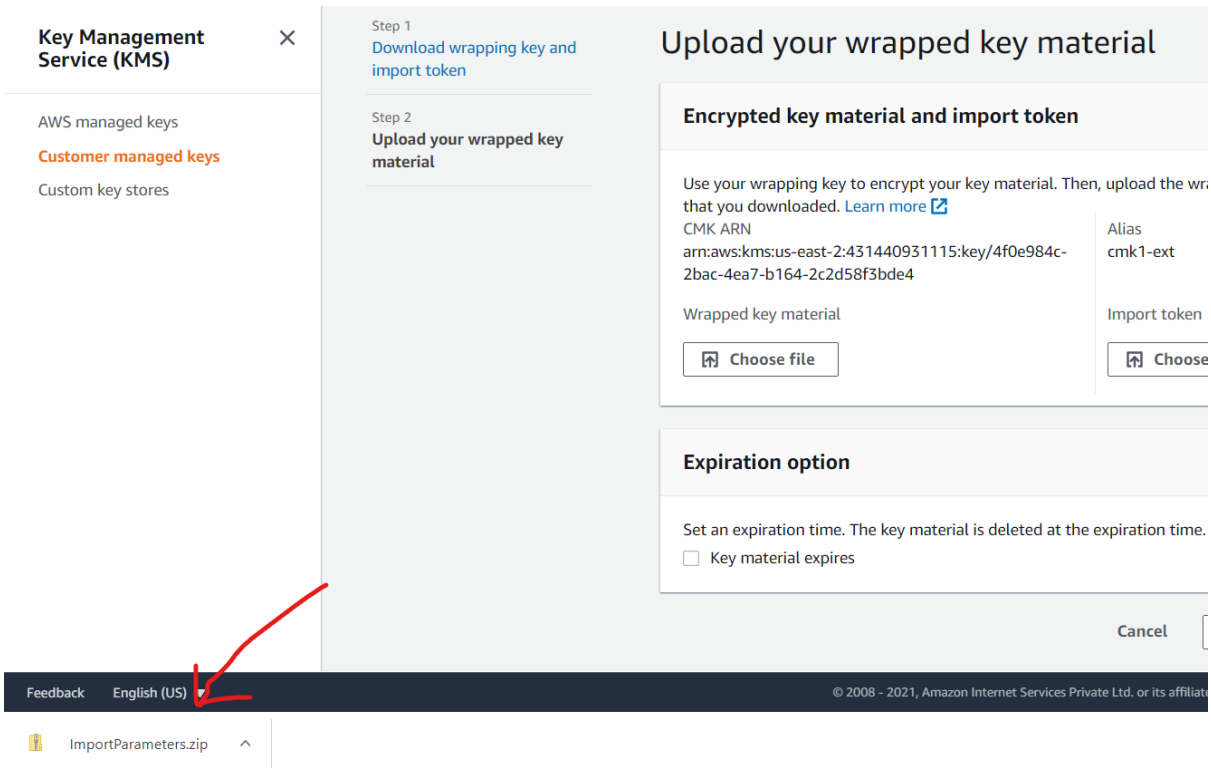
```
                    "arn:aws:iam::431440931115:role/aws-service-
role/elasticmapreduce.amazonaws.com/AWSServiceRoleForEMRCleanup",
                    "arn:aws:iam::431440931115:role/aws-service-
role/support.amazonaws.com/AWSServiceRoleForSupport",
                    "arn:aws:iam::431440931115:role/aws-service-
role/trustedadvisor.amazonaws.com/AWSServiceRoleForTrustedAdvisor",

"arn:aws:iam::431440931115:role/EMR_DefaultRole",

"arn:aws:iam::431440931115:role/EMR_EC2_DefaultRole",

"arn:aws:iam::431440931115:role/EMR_Notebooks_DefaultRole"
                ]
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow attachment of persistent resources",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::431440931115:role/aws-service-
role/elasticmapreduce.amazonaws.com/AWSServiceRoleForEMRCleanup",
                    "arn:aws:iam::431440931115:role/aws-service-
role/support.amazonaws.com/AWSServiceRoleForSupport",
                    "arn:aws:iam::431440931115:role/aws-service-
role/trustedadvisor.amazonaws.com/AWSServiceRoleForTrustedAdvisor",

"arn:aws:iam::431440931115:role/EMR_DefaultRole",
```

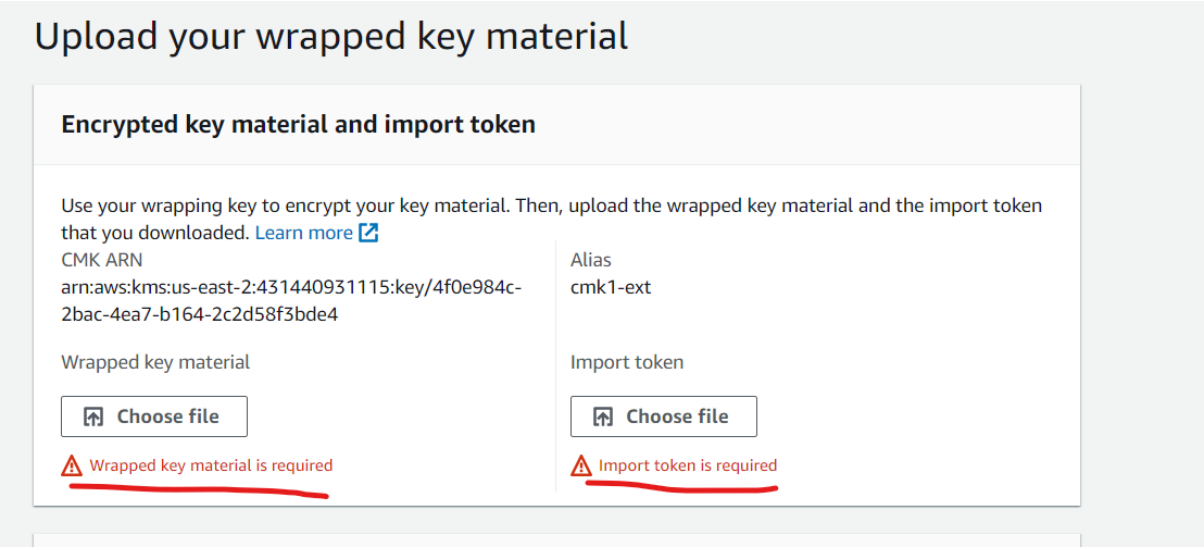
```
"arn:aws:iam::431440931115:role/EMR_EC2_DefaultRole",

"arn:aws:iam::431440931115:role/EMR_Notebooks_DefaultRole"
                ]
            },
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        }
    ]
}
```

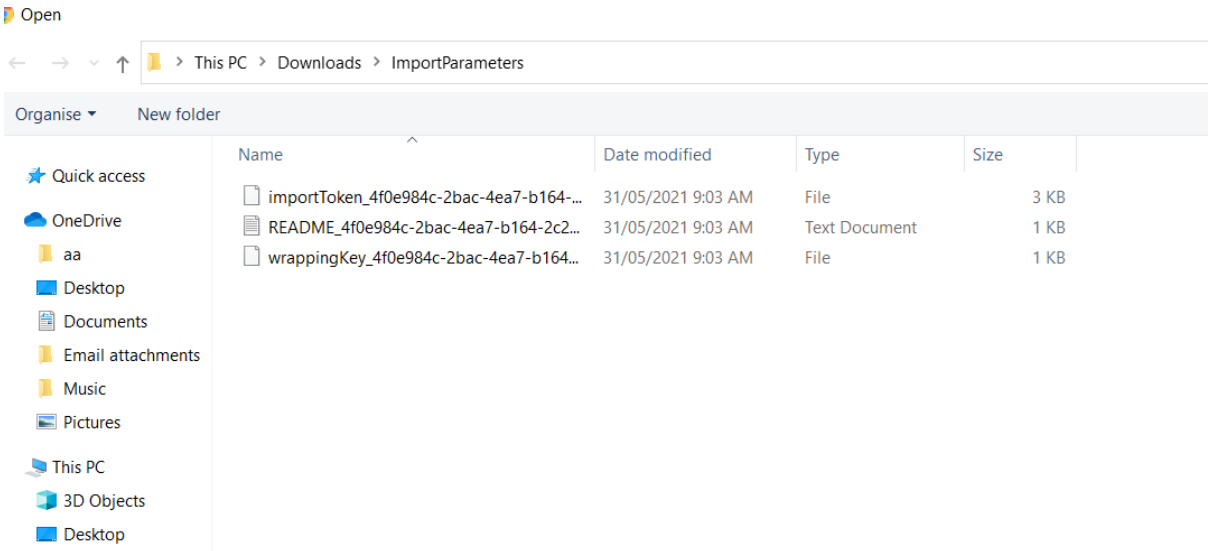
This (above) is the key policy




aws
Services
Search for services, features, marketplace products, and docs
[Alt+S]
ABHINAV SHRESHTH
Ohio
Support
Key Management Service (KMS)
AWS managed keys
Customer managed keys
Custom key stores
Step 1
Download wrapping key and import token
Step 2
Upload your wrapped key material
Upload your wrapped key material
Encrypted key material and import token
Use your wrapping key to encrypt your key material. Then, upload the wrapped key material and the import token that you downloaded. Learn more
CMK ARN
arn:aws:kms:us-east-2:431440931115:key/4f0e984c-2bac-4ea7-b164-2c2d58f3bde4
Alias
cmk1-ext
Wrapped key material
Choose file
Import token
Choose file
Expiration option
Set an expiration time. The key material is deleted at the expiration time.
Key material expires
Cancel
Previous
Upload key material
Feedback
English (US)
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

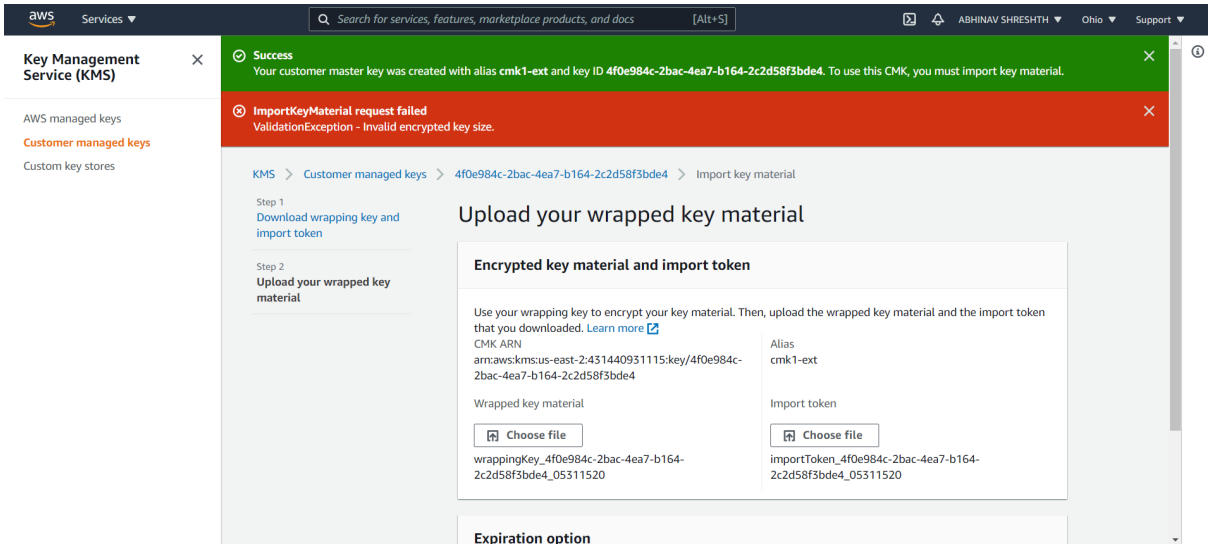

Key Management Service (KMS)
AWS managed keys
Customer managed keys
Custom key stores
Step 1
Download wrapping key and import token
Step 2
Upload your wrapped key material
Upload your wrapped key material
Encrypted key material and import token
Use your wrapping key to encrypt your key material. Then, upload the wr
that you downloaded. Learn more
CMK ARN
arn:aws:kms:us-east-2:431440931115:key/4f0e984c-2bac-4ea7-b164-2c2d58f3bde4
Alias
cmk1-ext
Wrapped key material
Choose file
Import token
Choose
Expiration option
Set an expiration time. The key material is deleted at the expiration time.
Key material expires
Cancel
Feedback
English (US)
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliat
ImportParameters.zip





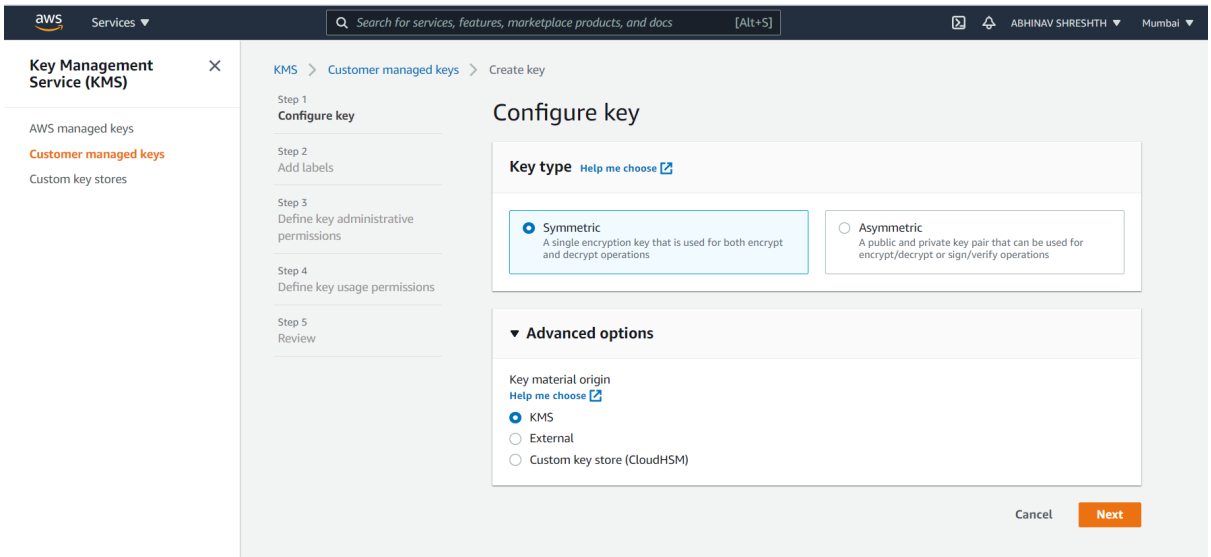
And I uploaded it

Issue - Got stuck

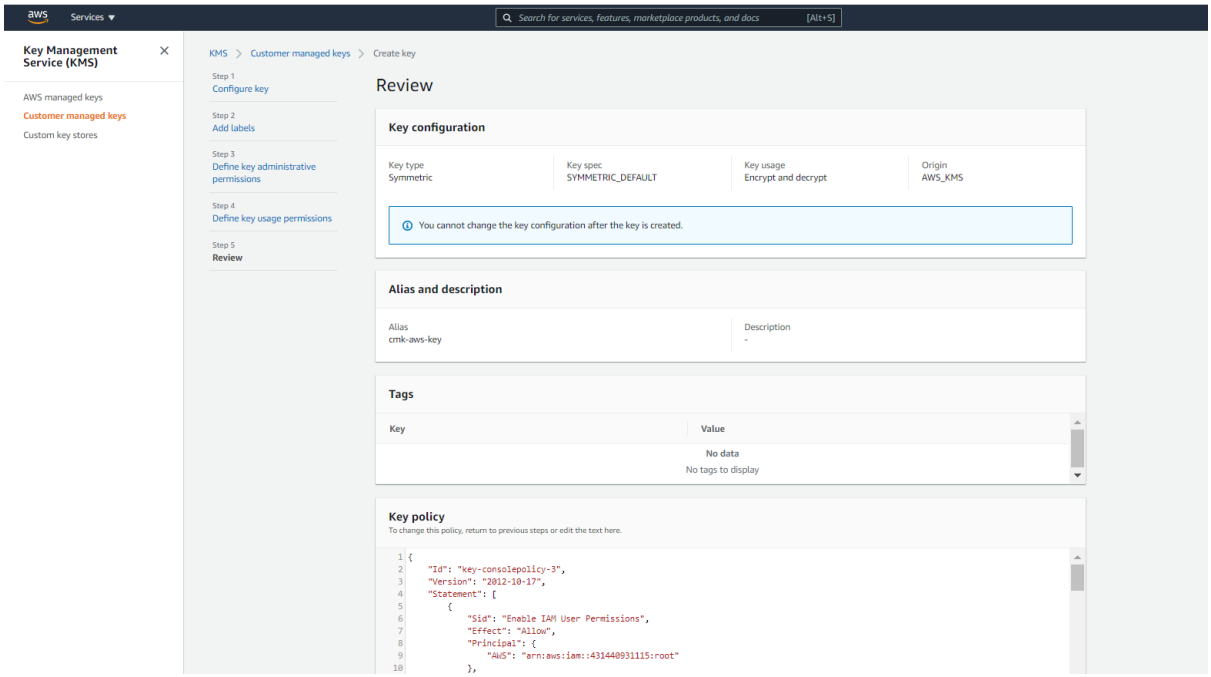


Rolling back –

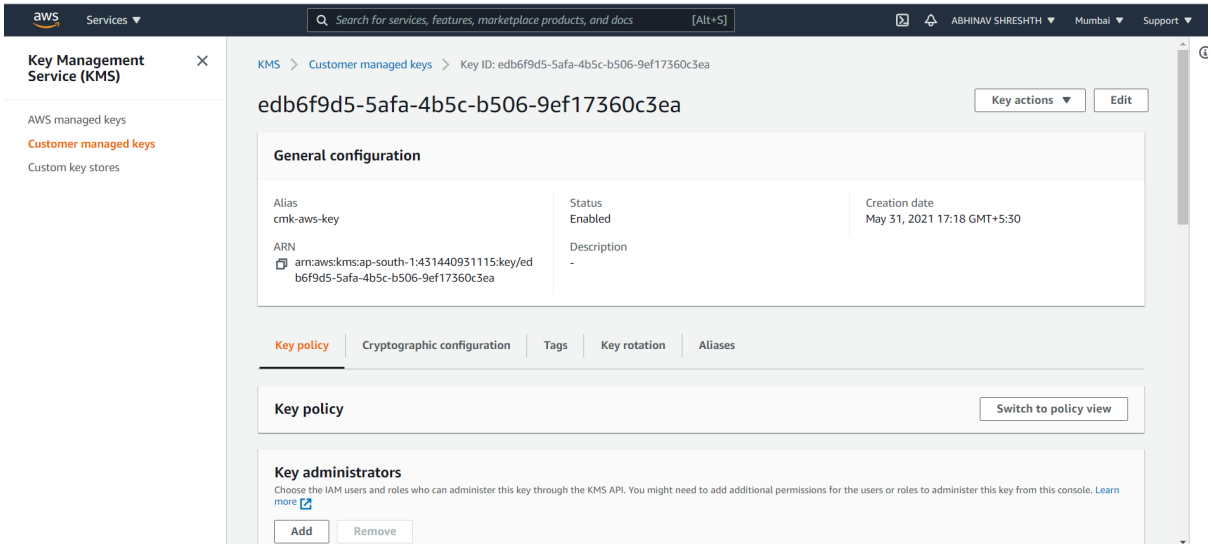
Encrypting using Customer managed keys –

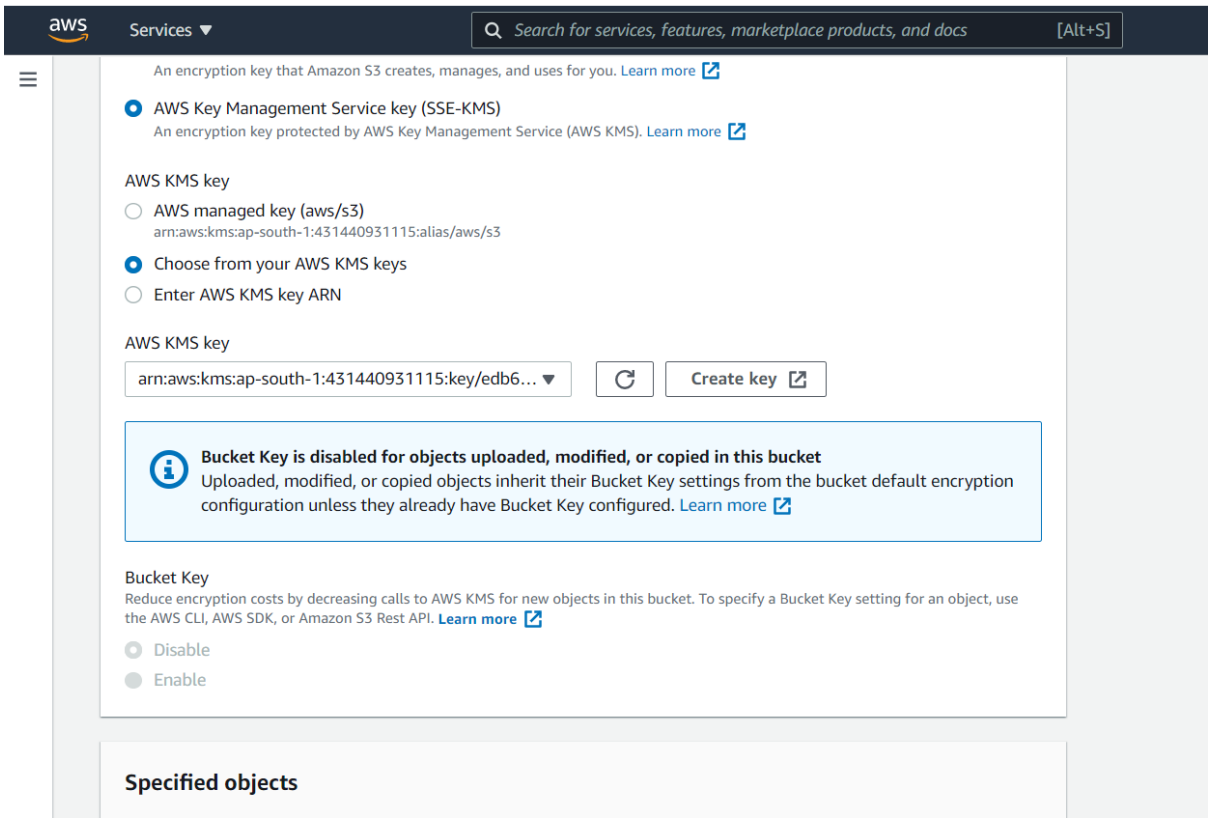


After some steps –

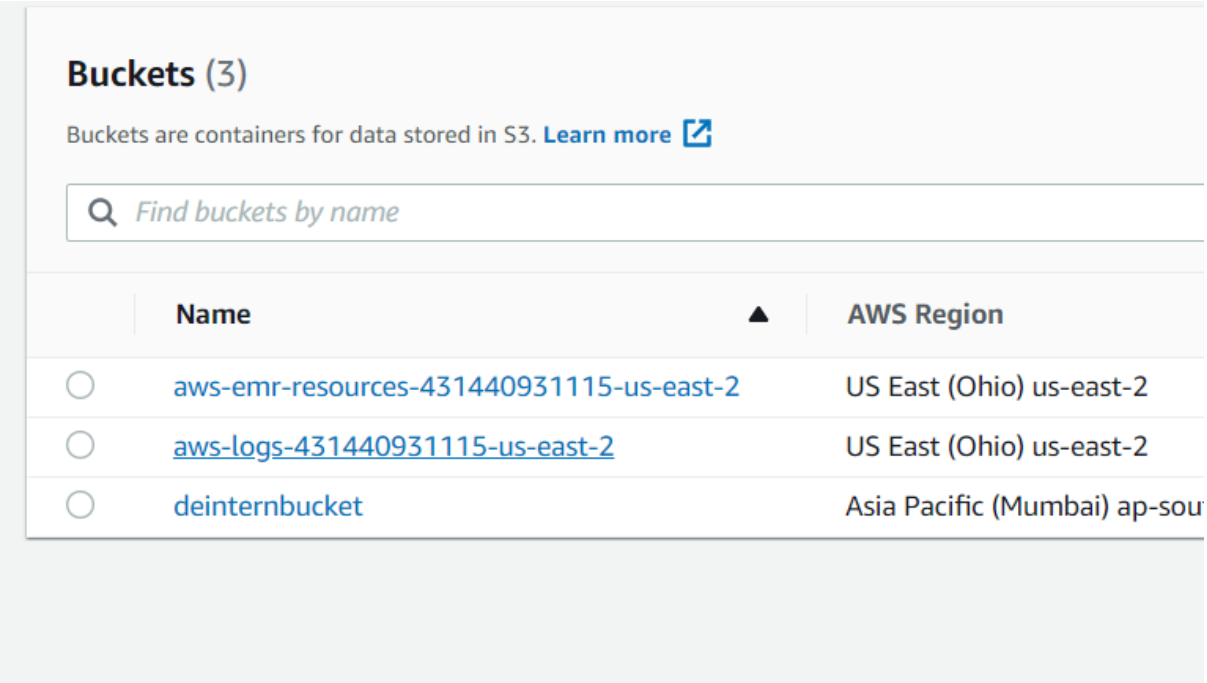


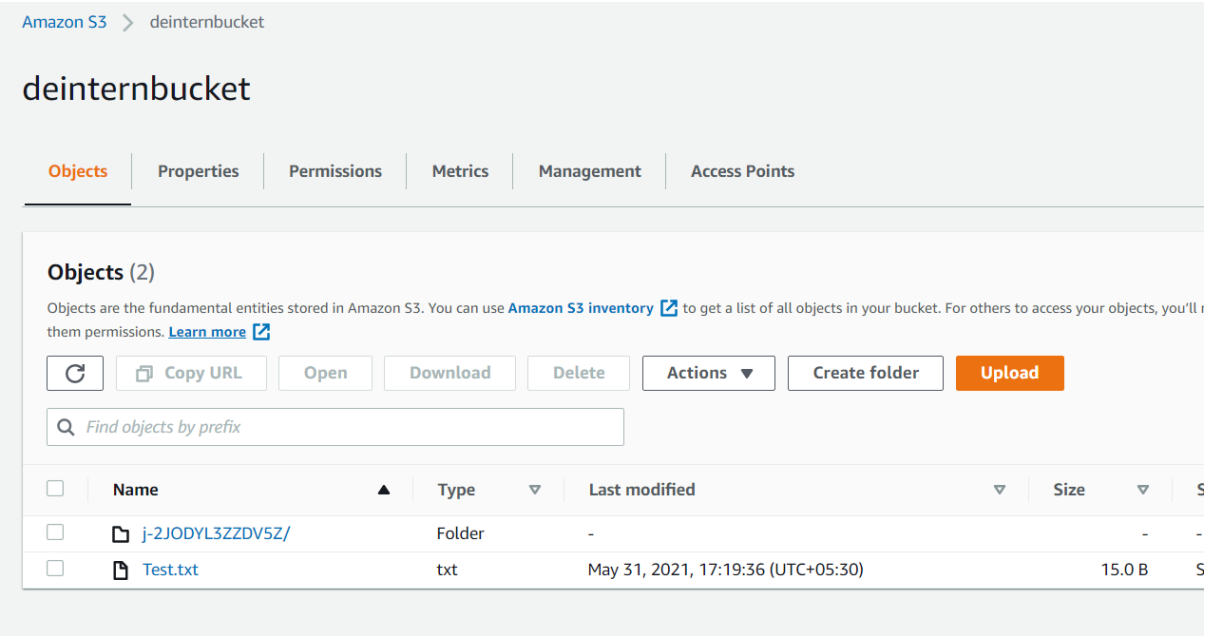
After all the procedure



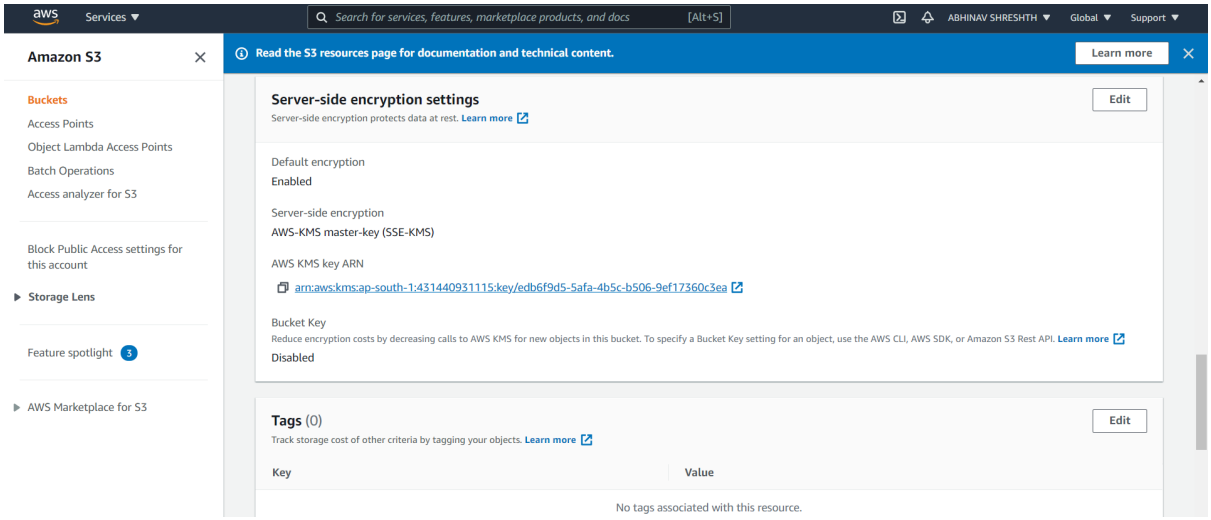


Final document –

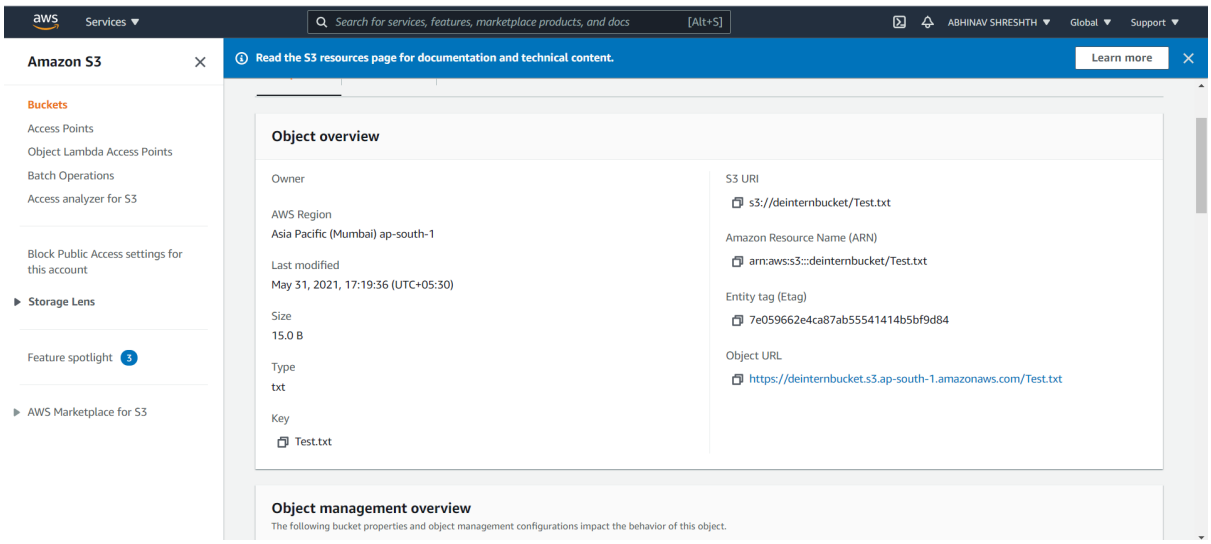




Here I am encrypting test.txt



As we can see this file is encrypted



Check this link –

https://deinternbucket.s3.ap-south-1.amazonaws.com/Test.txt

access denied

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>YF3ASZ8JWBVJ8F4P</RequestId>
  <HostId>gnZACdoJUtoVIT50y91jqbTRhfcormBDaHSLvE0BfjTx8xuQk5d23ulBAGbRchDWS6JR/pztDjY=</HostId>
</Error>
```


Encrypting using AWS-managed keys –

DOC – 5 | 29/05/21 – BY ABHINAV SHRESHTH

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] ABHINAV SHRESHTH ▼ Global ▼

Amazon S3

- Buckets
- Access Points
- Object Lambda Access Points
- Batch Operations
- Access analyzer for S3
- Block Public Access settings for this account
- Storage Lens
  - Dashboards
  - AWS Organizations settings
- Feature spotlight 3
- AWS Marketplace for S3

Amazon S3 > deinternbucket > Upload

Upload

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 folders**.

Files and folders (1 Total, 46.0 B)

All files and folders in this table will be uploaded.

Remove | Add files | Add folder

Find by name

Name	Folder	Type	Size
test2.txt	-	text/plain	46.0 B

Destination

Destination
s3://deinternbucket

Destination details

Feedback English (US) ▼ © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Coo

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] ABHINAV SHRESHTH ▼ Global ▼ Support ▼

Amazon S3

- Buckets
- Access Points
- Object Lambda Access Points
- Batch Operations
- Access analyzer for S3
- Block Public Access settings for this account
- Storage Lens
  - Dashboards
  - AWS Organizations settings
- Feature spotlight 3
- AWS Marketplace for S3

Server-side encryption settings

Server-side encryption protects data at rest. Learn more

Server-side encryption
- Do not specify an encryption key
- Specify an encryption key

Encryption key type
To upload an object with a customer-provided encryption key (SSE-C), use the AWS CLI, AWS SDK, or Amazon S3 REST API.
- Amazon S3 key (SSE-S3)
  An encryption key that Amazon S3 creates, manages, and uses for you. Learn more
- AWS Key Management Service key (SSE-KMS)
  An encryption key protected by AWS Key Management Service (AWS KMS). Learn more

AWS KMS key
- AWS managed key (aws/s3)
  arn:aws:kms:ap-south-1:431440931115:alias/aws/s3
- Choose from your AWS KMS keys
- Enter AWS KMS key ARN

Bucket Key is disabled for objects uploaded, modified, or copied in this bucket
Uploaded, modified, or copied objects inherit their Bucket Key settings from the bucket default encryption configuration unless they already have Bucket Key configured. Learn more

Bucket Key
Reduce encryption costs by decreasing calls to AWS KMS for new objects in this bucket. To specify a Bucket Key setting for an object, use

Feedback English (US) ▼ © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] ABHINAV SHRESHT

Upload succeeded
View details below.

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://deinternbucket	1 file, 46.0 B (100.00%)	0 files, 0 B (0%)

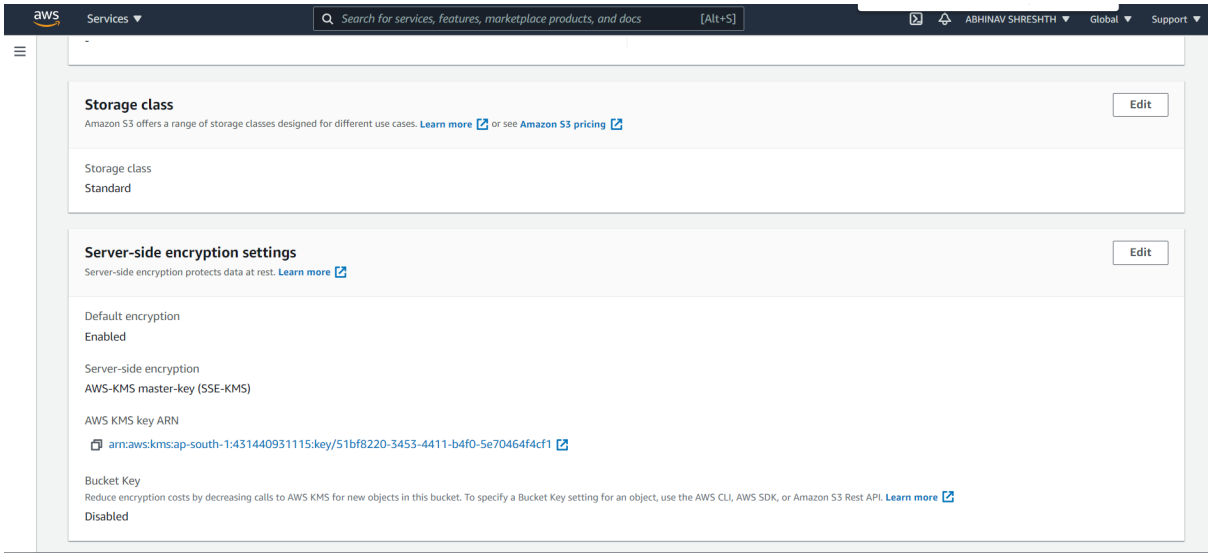
Files and folders | Configuration

Files and folders (1 Total, 46.0 B)

Find by name

Name	Folder	Type	Size	Status	Erro
test2.txt	-	text/plain	46.0 B	Succeeded	-

In properties of test2.txt



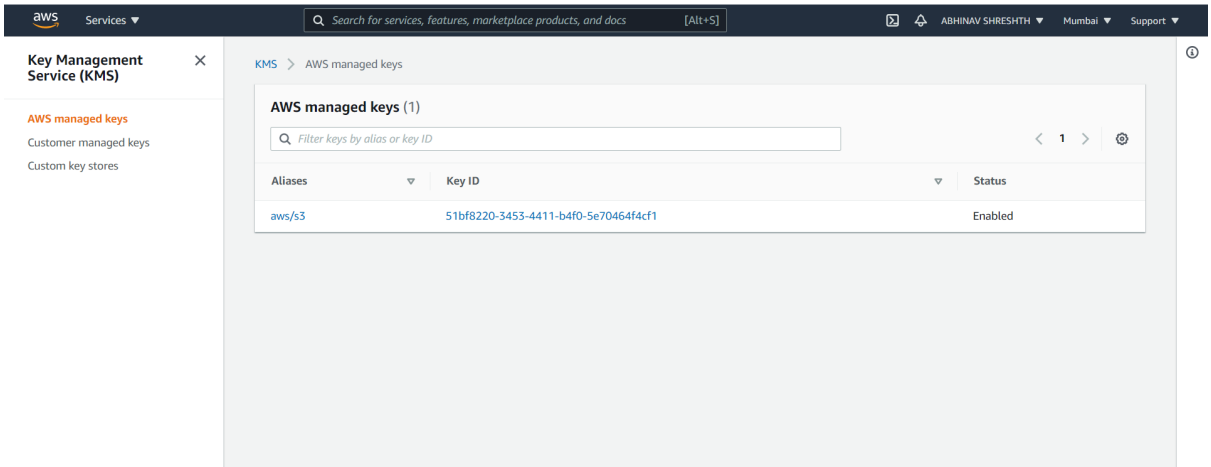
Now checking the url https://deinternbucket.s3.ap-south-1.amazonaws.com/test2.txt

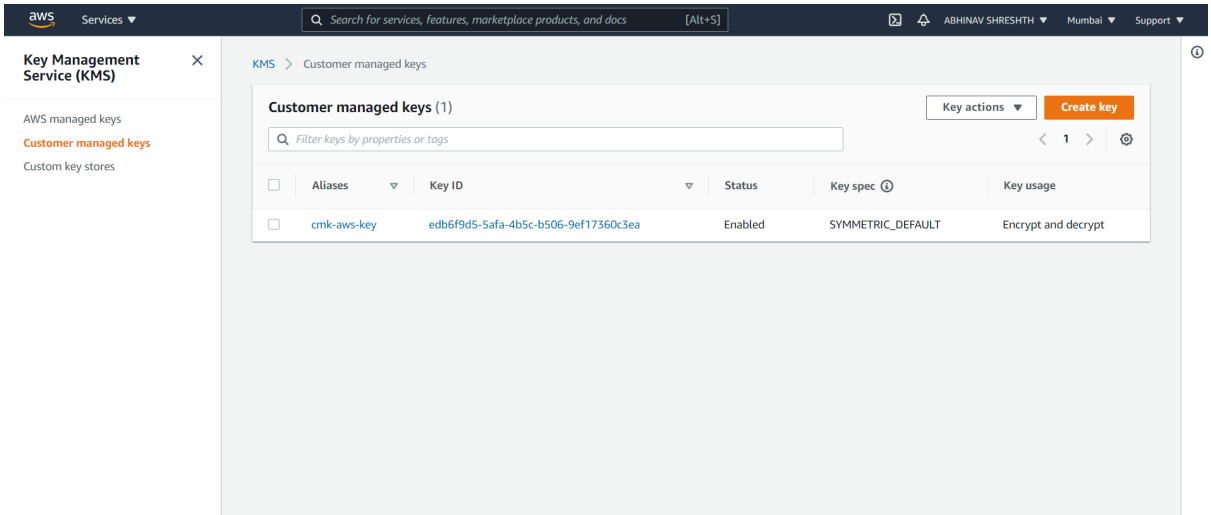
Successfully encrypted

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>0QXGZJQDTPW0E63T</RequestId>
  <HostId>xOsDe9gP27vrYW0y8L3isl2VvWgnu6OG+xyWC2PgzeYw/dwNBpz9GBjcpmbTpEwq5Phnuf8bxvY=</HostId>
</Error>
```

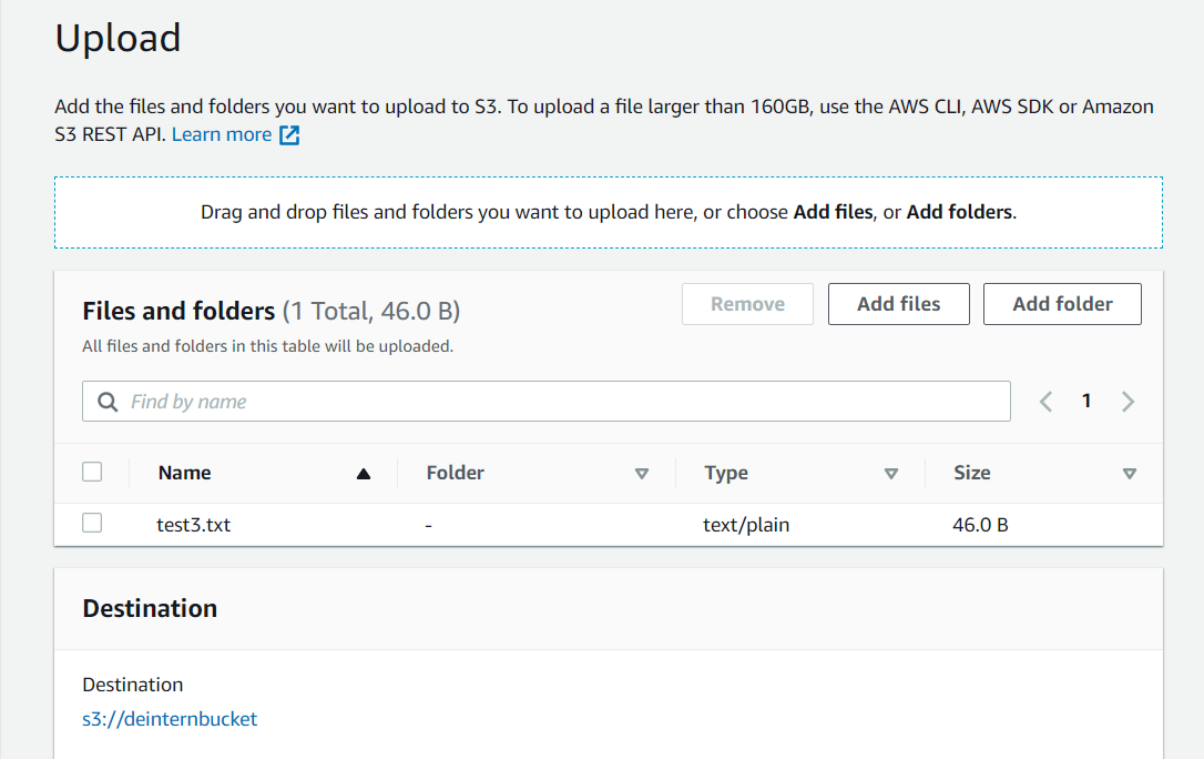

Now switching back to key management service (KMS)

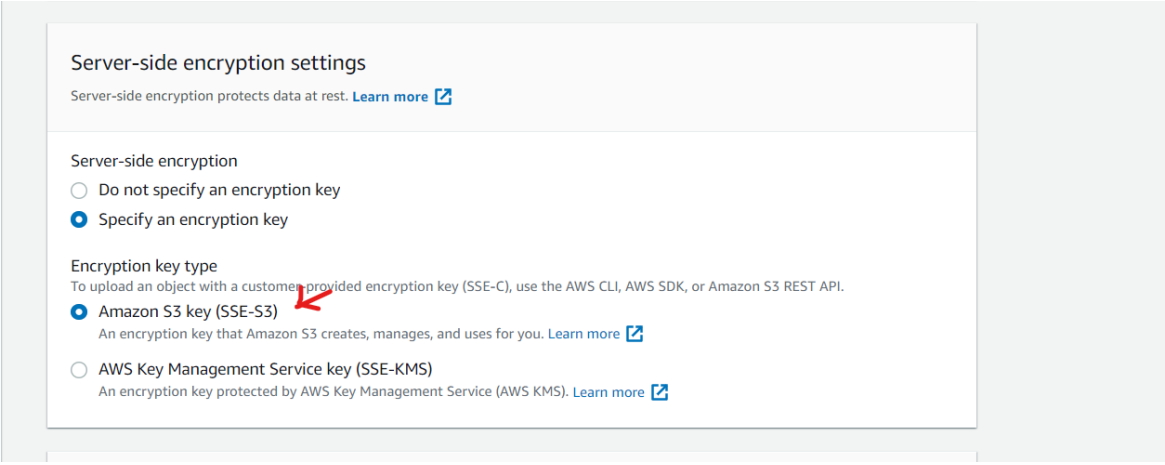




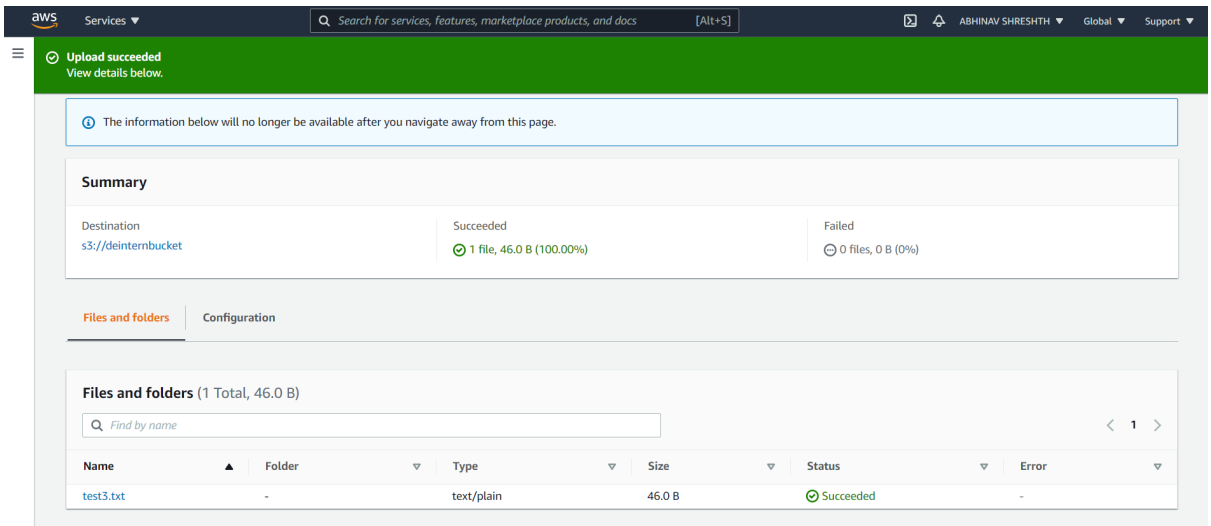
As we can see customer manged keys and AWS managed keys entry are present.

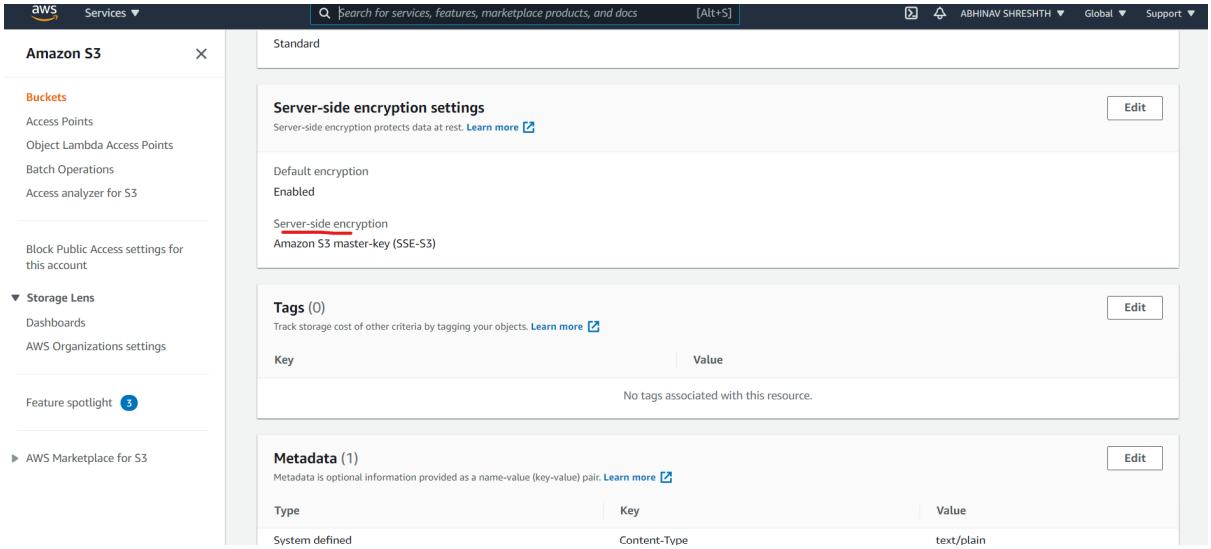
Now trying uploading test3.txt



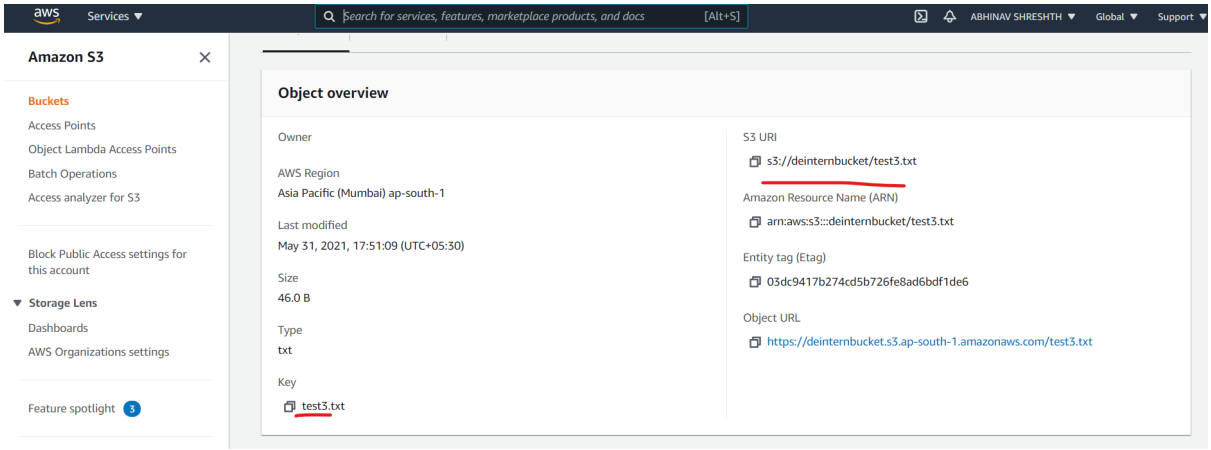


Uploading successful of test3.txt





Now checking the file test3.txt is successfully encrypted or not



https://deinternbucket.s3.ap-south-1.amazonaws.com/test3.txt

now this file is also got encrypted

++