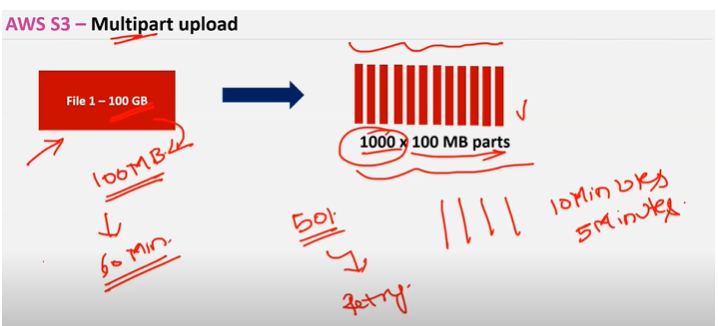
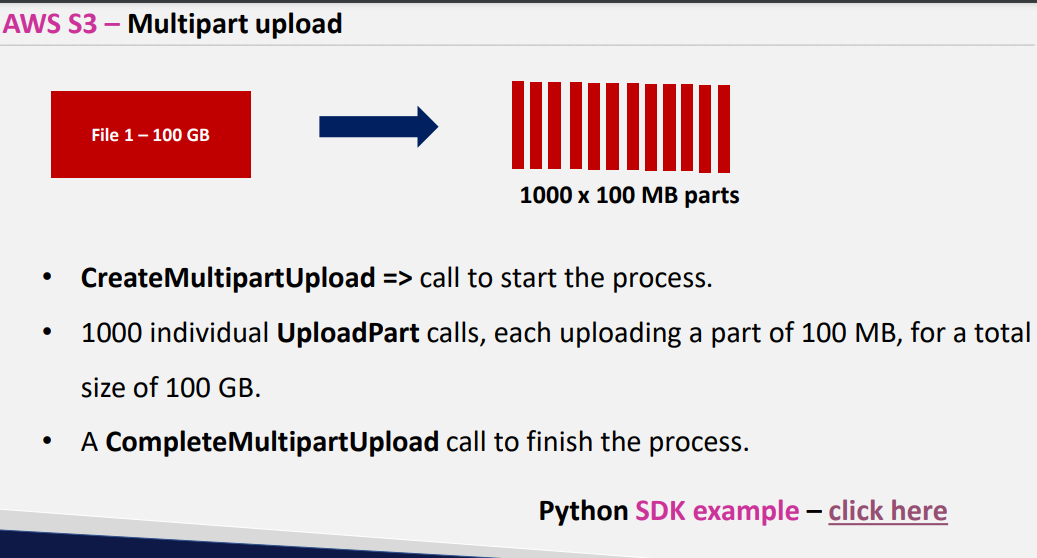
**AWS S3 Multipart upload**

* **In S3 if we have small size files we can upload it easily but we have large multiple files its difficult to upload at once**
* **If the file size is more than 100 mb its considered as large files**
* **And you should not upload the large files directly**
* **Those large files should be divided in to multiple parts for so that it can upload easily in short time**
* **So if have 1000mb file and when we try to upload at once and it got stucked in the middle again we need to upload from the strting**
* **So in multipart upload as we are dividing the files into multiple parts if we upload it and only one part got failed to upload then will retry that part to be uploaded again**
* ****
* **For multipart upload we need to use Api’s, cli, or Sdk**
* ****
* **For s3 when you say createmultipartUpload it gives you some unique id**
* **And you will give some number to multiple parts**
* **After uploading multiparts you will tell CompleteMultipartupload to finish the parts**
* **And s3 consolidated all the multiparts in to one file as soon as you call CompleteMultiPartUpload**
* [**http://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparsion.html**](http://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparsion.html) **- comparision tool between s3 direct upload and transfer acceleration**
* **In s3 if you upload directly it need to pass the data between different routers**

**It might take more time to upload u r files**

**Hop1 – Hop2 -Hop3- Hop4-Hop5……..Hop n**

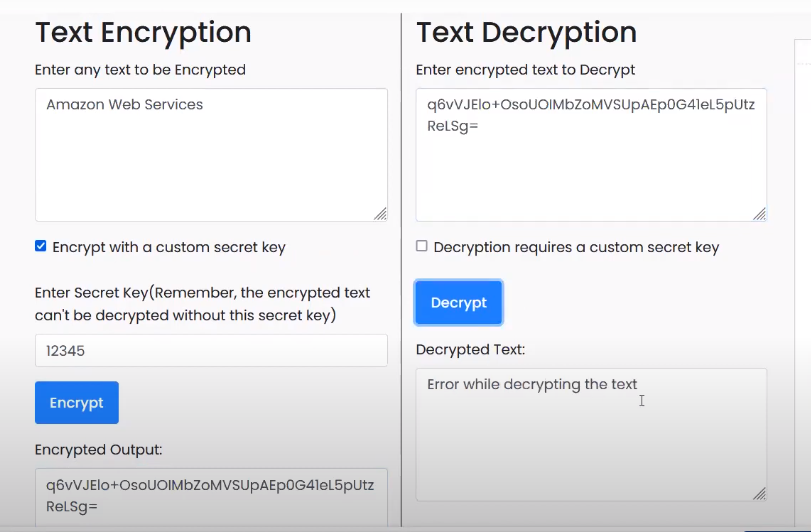
* **S3 transfer acceleration is created by aws and there will less number of routers in which u need to pass the data**

**Hop1 -Hop2**

**Maximum you will have 2 routers from source to Destination**

* **Imp concept examination perspective transfer acceleration**

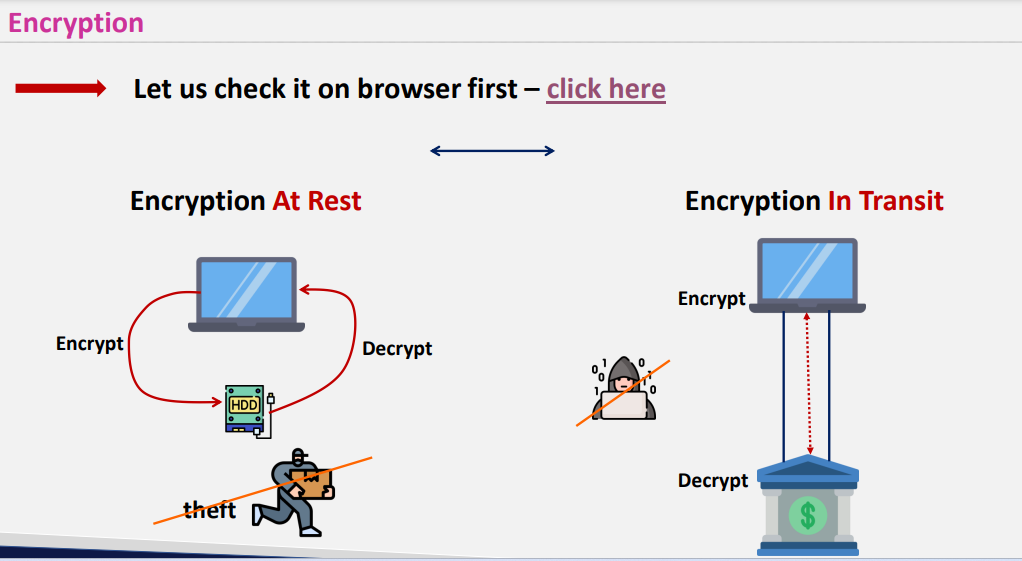


****

**Encrypt using a secret key for security purpose**

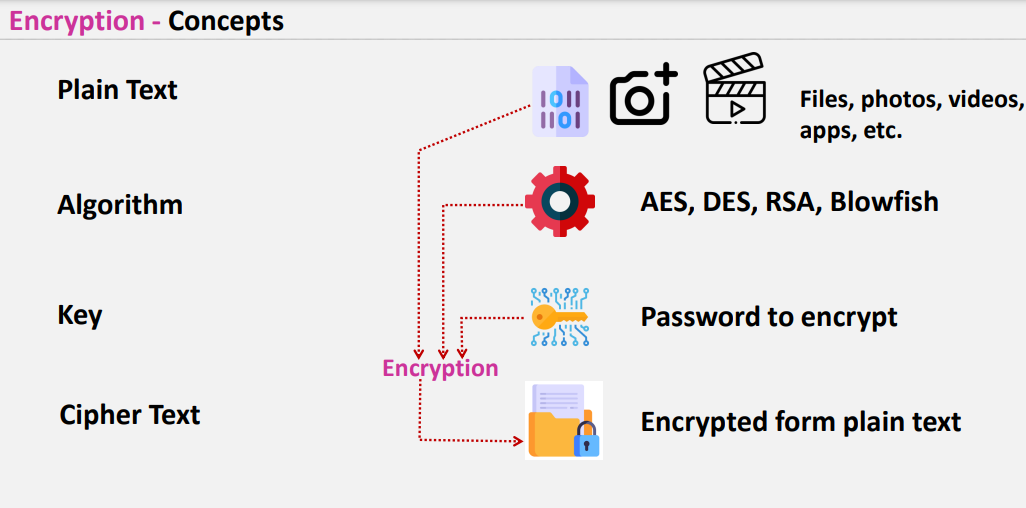
* **Nobody can decrypt the data until they have a secret key**

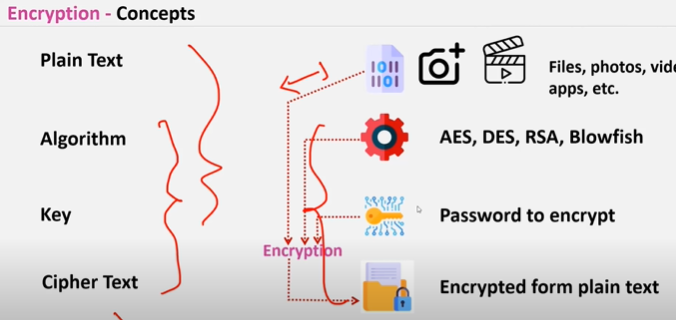
[**https://www.devglan.com/online-tools/text-encryption-decryption**](https://www.devglan.com/online-tools/text-encryption-decryption)



**Encrypt at Rest at stored place ex s3 , laptop etc Transfering one place to another**

**Ex: Https secured and http not secure**

****

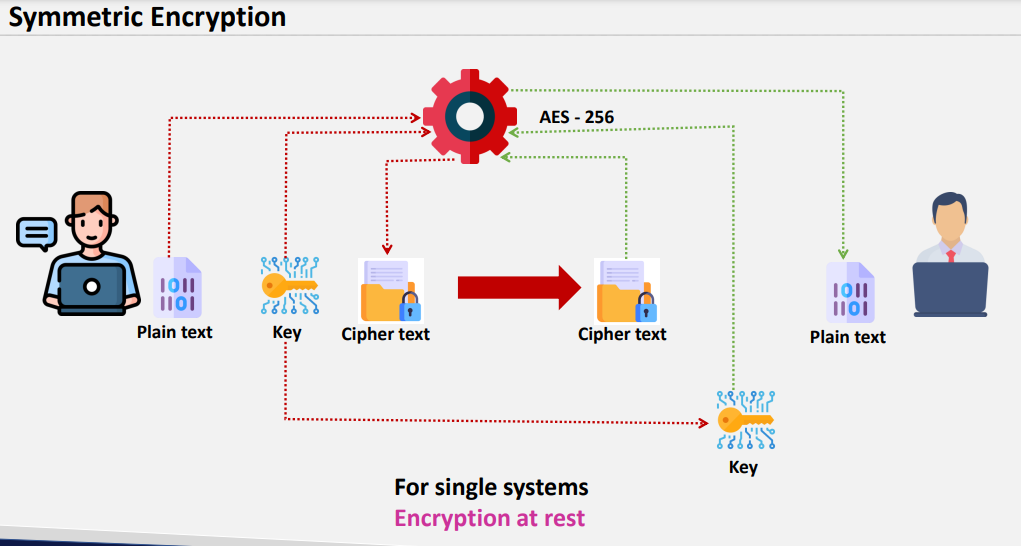
****

**For encryption we use Alogrithm key and response will be cipher text**

**For decryption we use cipher text key algorithm and will get plain text**

**Encryption is of two types**

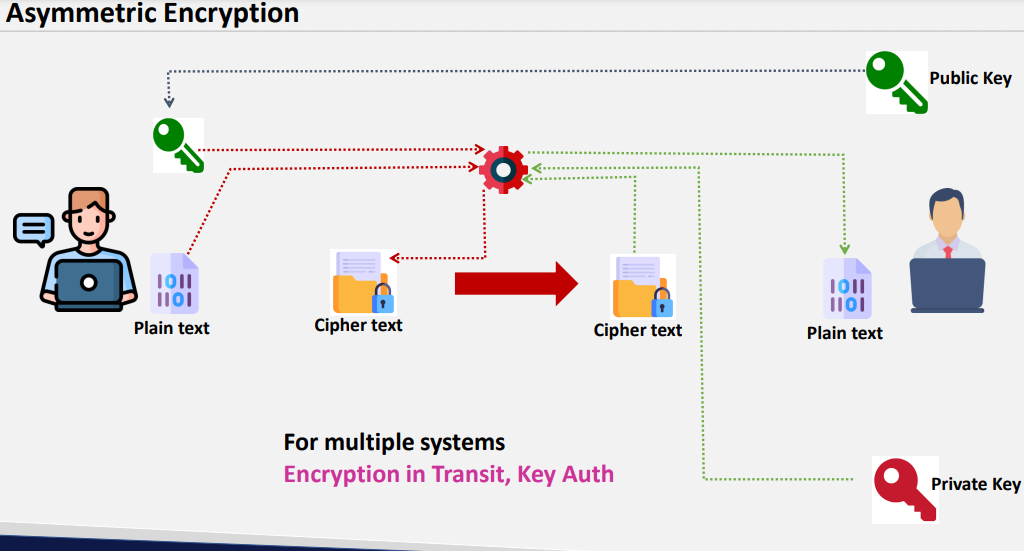
1. **Symmetric encryption**
2. **Asymmetric encryption**

****

**Symmetric encryption wont be used in between two servers or two persons**

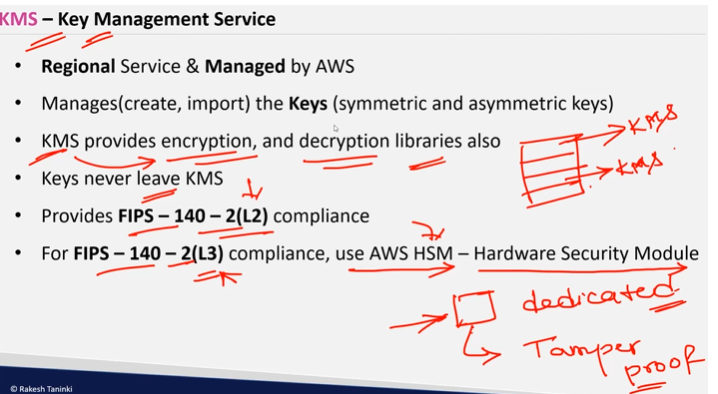
**Becz if you are giving key separately to user 2 to decrypt the data and that key has been caught by another 3 rd person you entire files can decrypted by him**

* **Symmetric encryption should be done at rest**
* **For disk storage we use symmetric encryption**



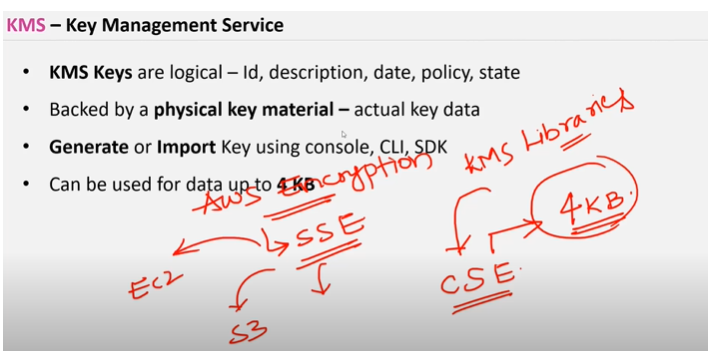
**We use asymmetric encryption when we have to transfer in multiple systems**

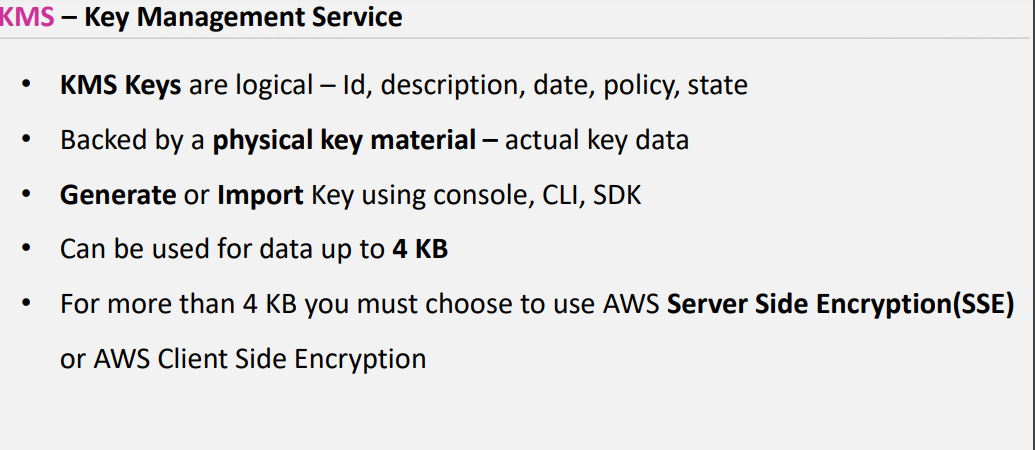
**KMS: Keymanagement service provided by aws for keys security**

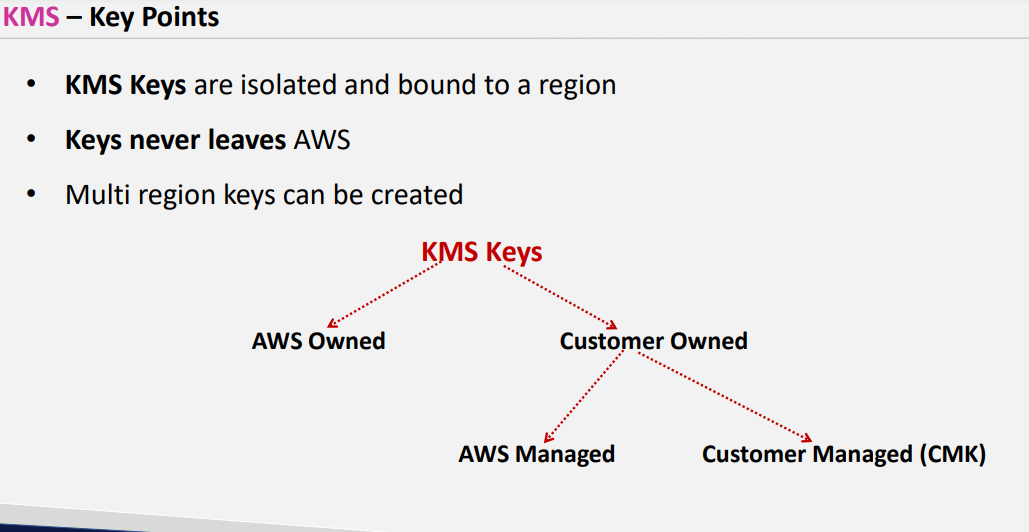
* **What ever keys is generated in Kms will be hidden for security pupose and it will be with in KMS SERVICE**
* ****

**Extra layer of security will be there in Hardware security model as it has tamper proof and also its dedicated host only your keys will be stored in that disk**

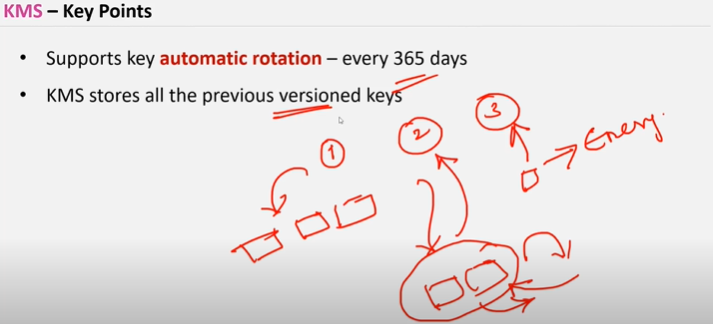
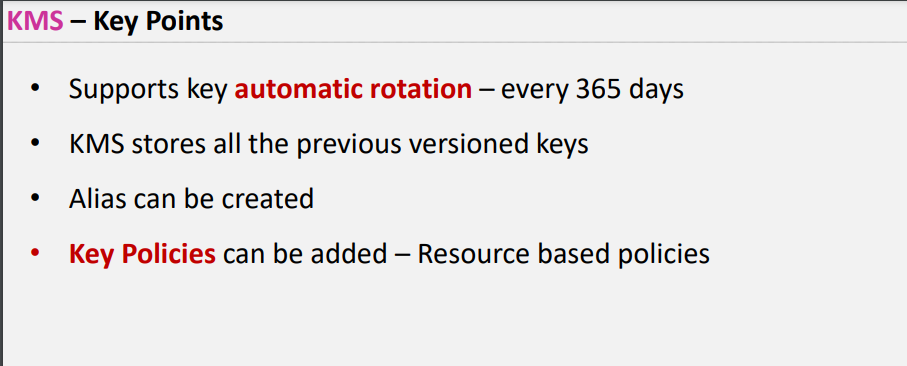
* **In KMS your keys will be stored in shared model**
* **In HSM even data center gets destroyed due to some flood issues your hsm wont be destroyed as its tamper proof**
* **Extra charges will be there in HSM**
* **Encryption in kms is of two types**
* **1. SSE(Server side encryption ) which is managed by AWS**
* **2.CSE (Client side encryption) where you use kms libraries for encryption and here you can create encryption upto 4kb data**
* **In SSE there is no limitation**
* **S3 data, Ec2 data everything is encrypted by aws and there is no limitation for encryption**

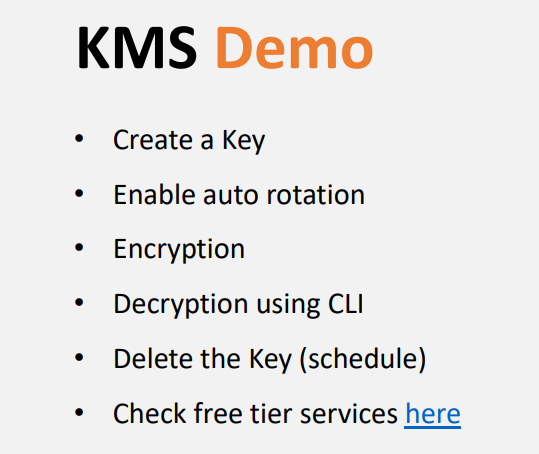
****

****

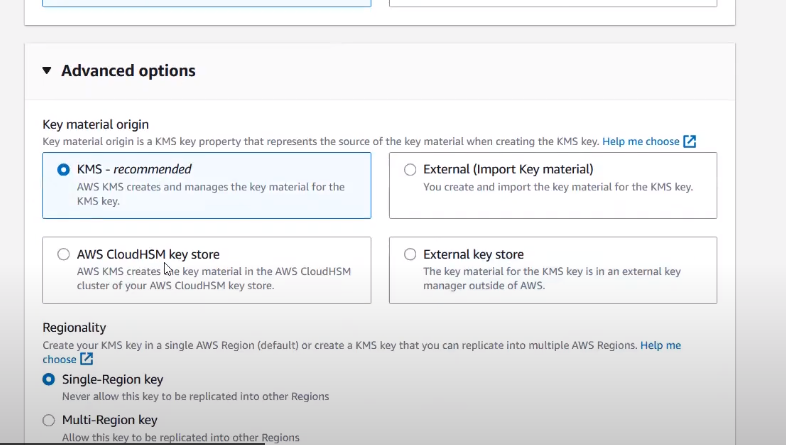
****

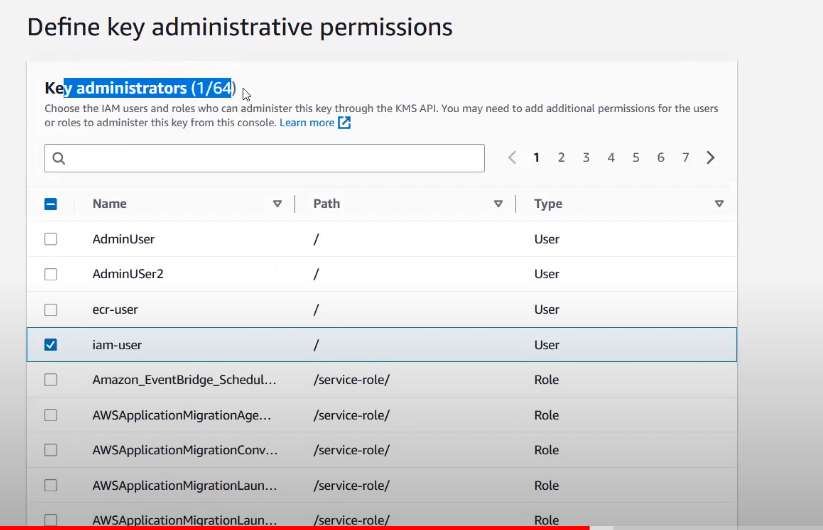
**Customer owned keys are which ever is created in kMS**

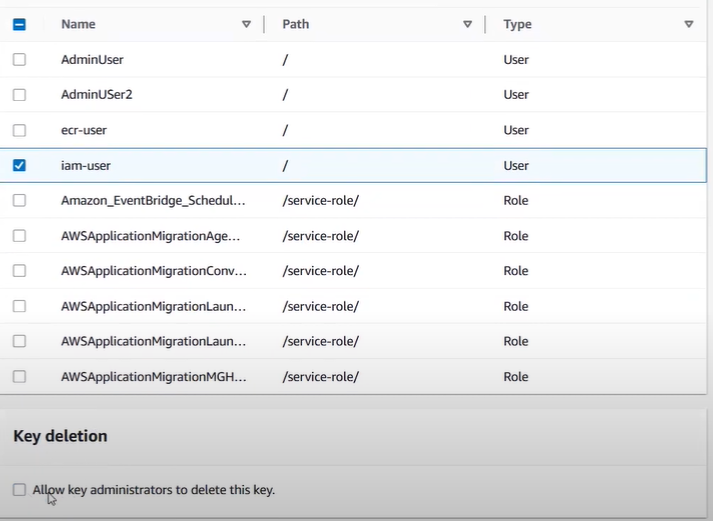
* **Imp KMS provides keys automated rotation for every 365 days**
* **Old files will also accessible as aws stores old key as well**
* **And newly generated files will be accessible based on new key which got automated**
* ****
* ****

****

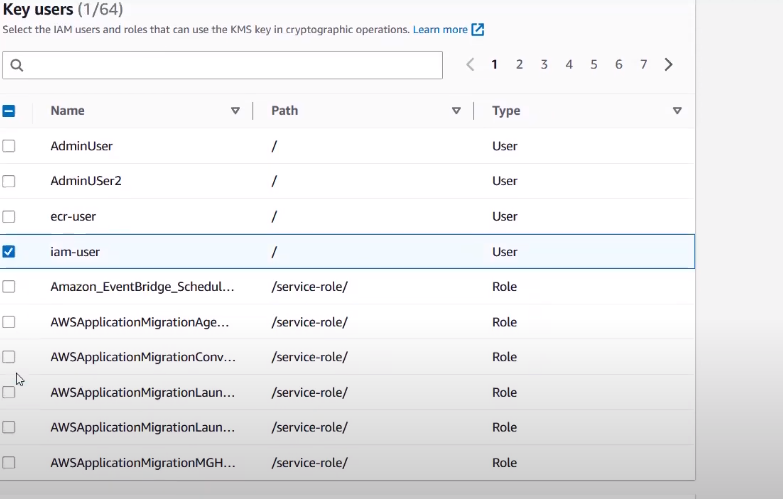
[**https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=\*all&awsf.Free%20Tier%20Categories=\*all&all-free-tier.q=key&all-free-tier.q\_operator=AND**](https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all&all-free-tier.q=key&all-free-tier.q_operator=AND)

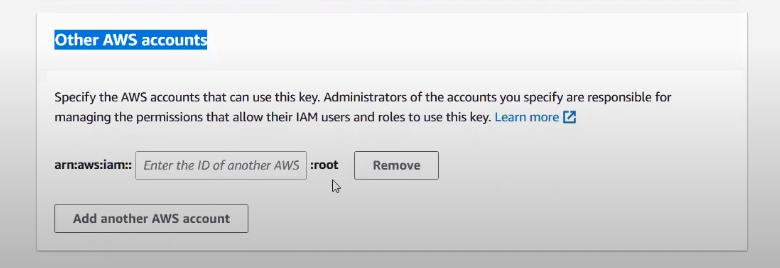
****

****

****

**You can also give permissions for this particular key to delete or not**

****

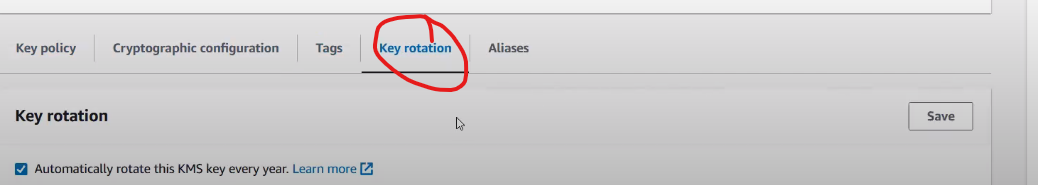
****

**Even if you have all policies disable and mentioned other act details to access s3 cross act**

**If you enable kms without mention other act id in kms section other act cannot be decrypt the data**

****

**You can rotate the keys**

****

**In cloud shell type echo (used to print)**

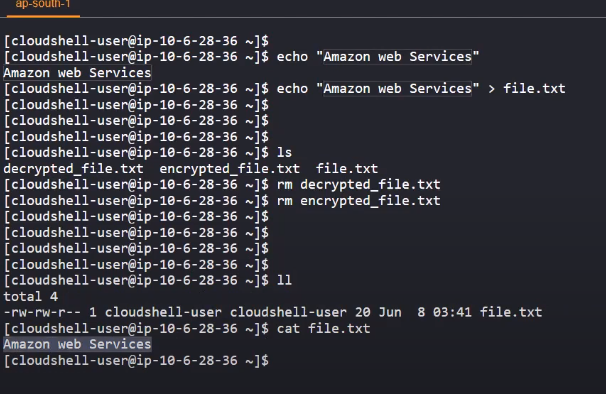
**Echo “Amazon web services” >file.txt**

**Ls (to list the files)**

**Rm file name (to remove that particular file)**

**Cat to print the data from that file**

**Echo to print the data into the file**

****

**s**

**aws kms encrypt --key-id alias/xxx --plaintext fileb://file.txt --output text --query CiphertextBlob | base64 --decode > encrypted\_file.txt -------to encrpt**

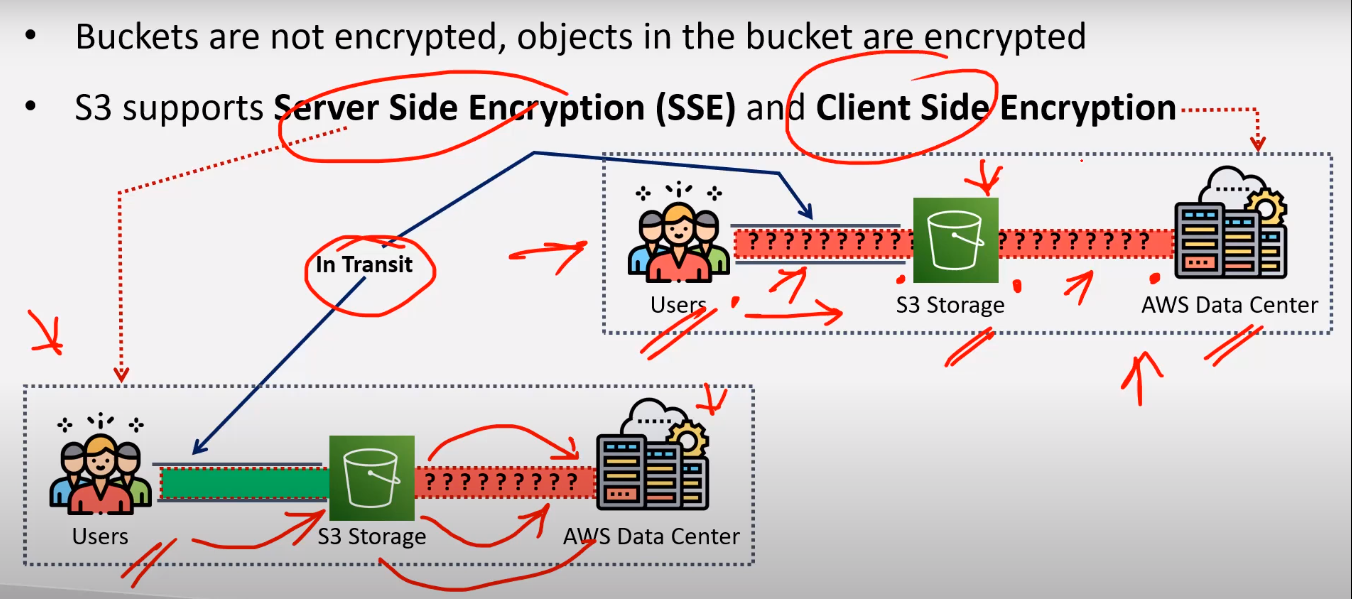
**ll (long list)**

**cat encrypted file.txt**

**aws kms decrypt --ciphertext-blob fileb://encrypted\_file.txt --output text --query Plaintext | base64 --decode > decrypted\_file.txt ------ to decrypt**

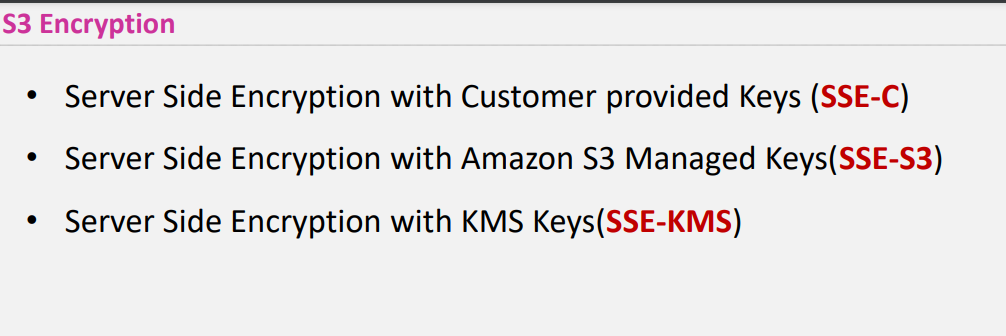
**cat decrypted file.txt**

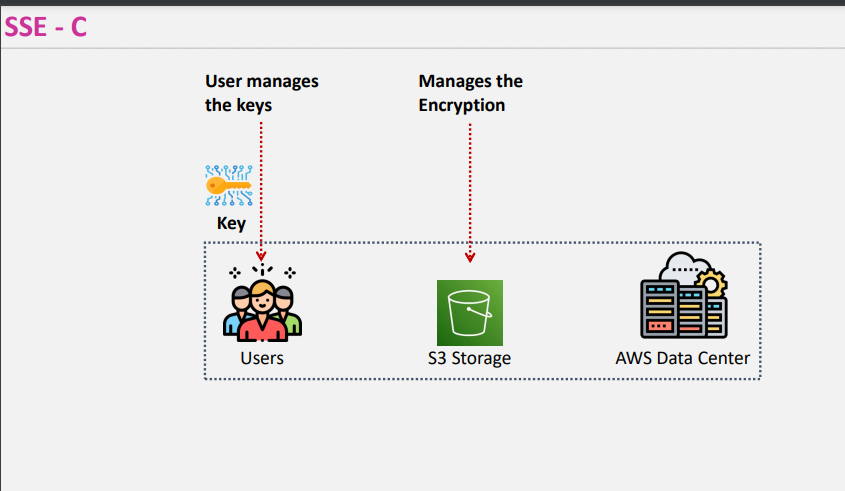
* **In real time we use symmetric encryption**
* **By default aws will encrypt the data but that can be accessible by all the users**
* **If you want to access the keys for only that particular u neeed to go with KMS**

****

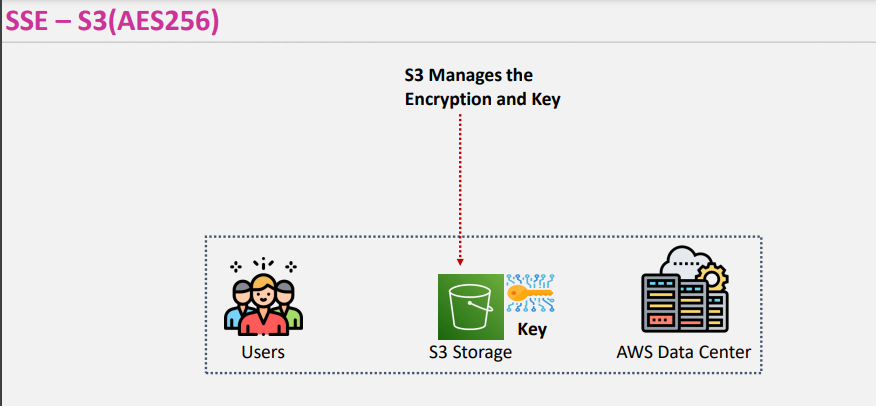
**In client side encryption u will upload a encrypted data**

* **In SSE you will upload a plain text in s3 and from s3-> data center data will be encryted**

****

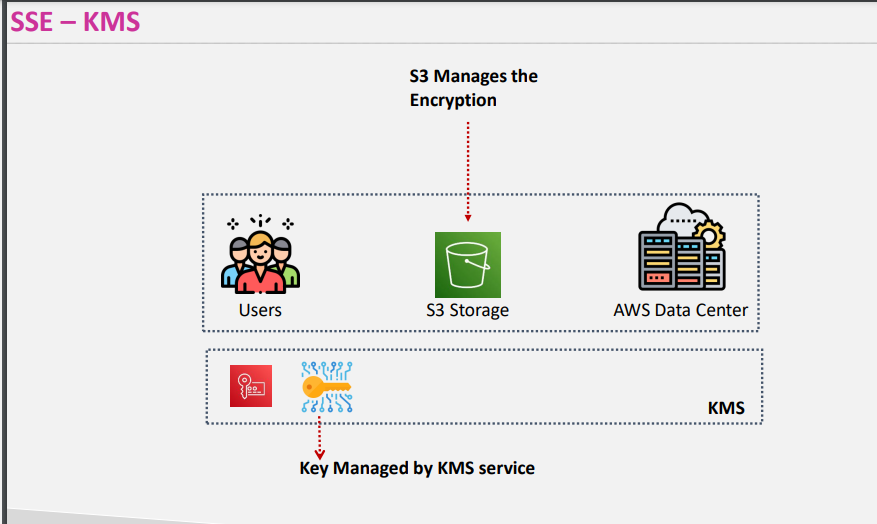
****

**Customer managed In SSE (C) Customer will manage the keys and create the keys and while uploading the file to s3 he passes the keys along with file to s3 then his data will be encrypted**

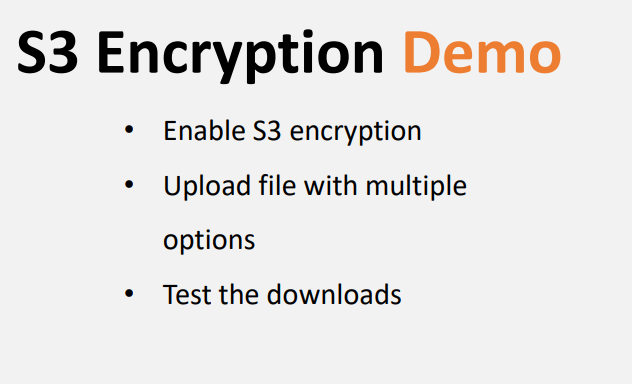
****

**AWS managed**

**Here S3 only create the keys and does encryption process users simply upload the plain text to s3**

****

**S3 managed the encryption keys will be managed by kms**

****

**By default every s3 keys buckets are managed by aws (SSE -S3)**

**SO FOR THOSE KEYS U DON’T have access**

**3 files has been uploaded**

1. **Without encryption**
2. **File with aws s3 encryption**
3. **File with kms encryption**
4. **Try accessing through Iam user**

**In keyscheduling you can schedule the data**