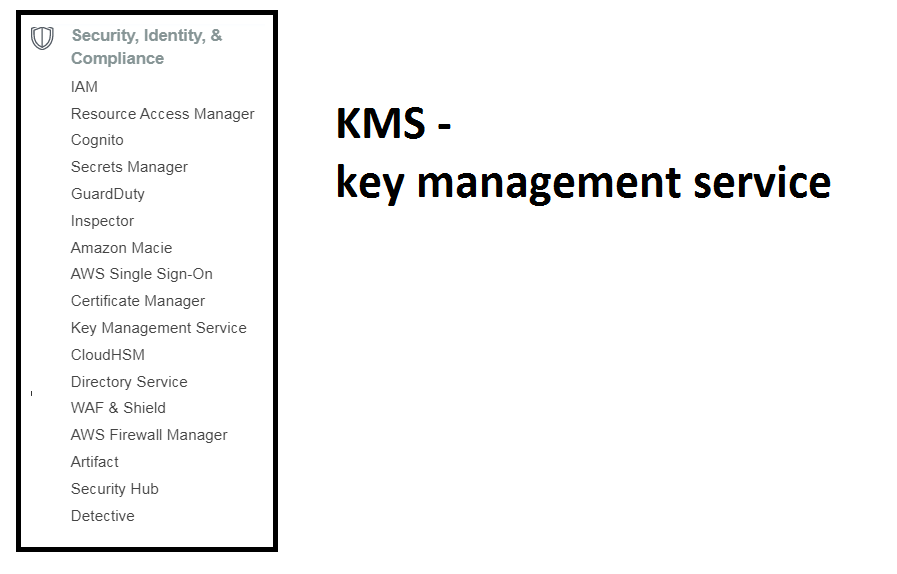
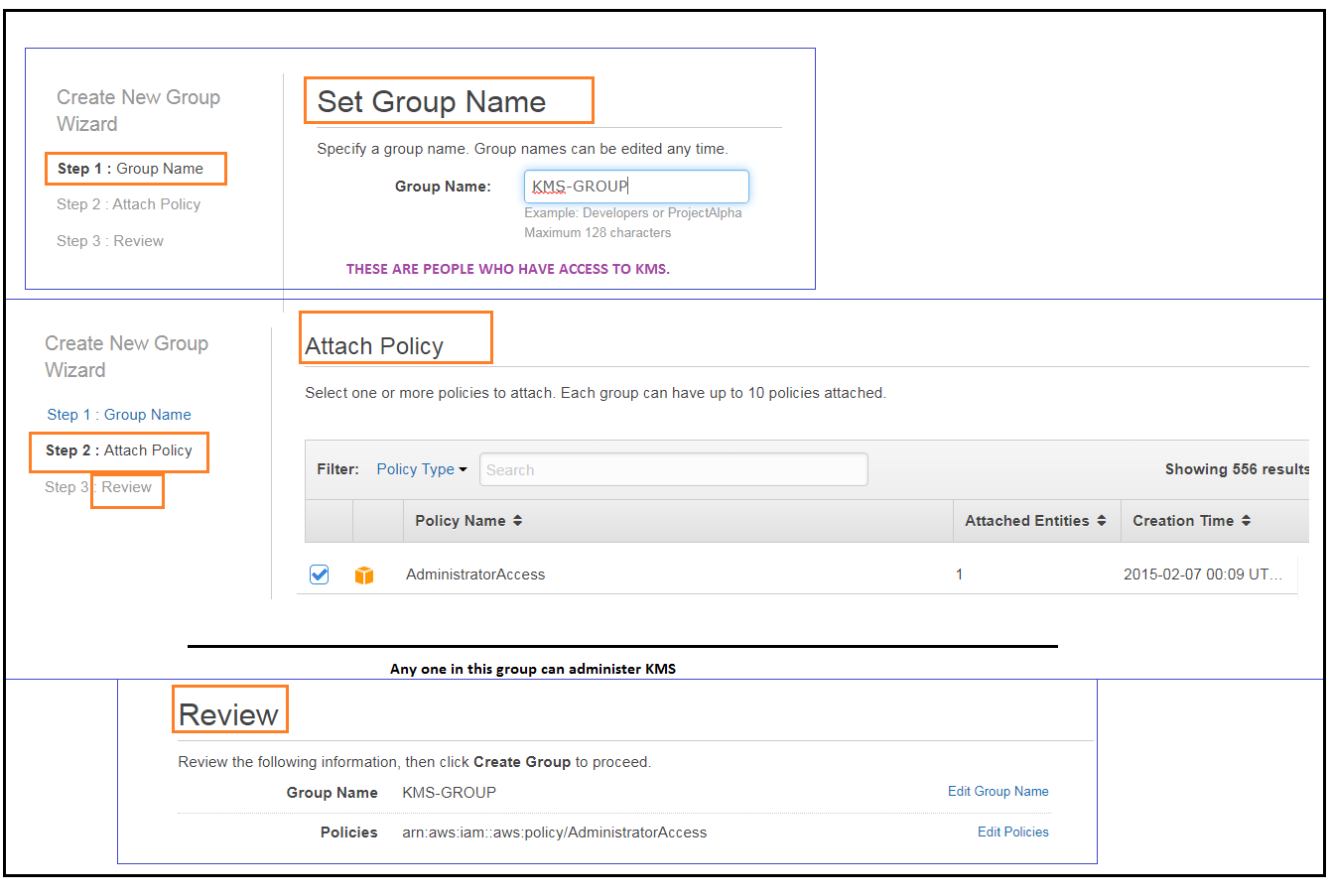
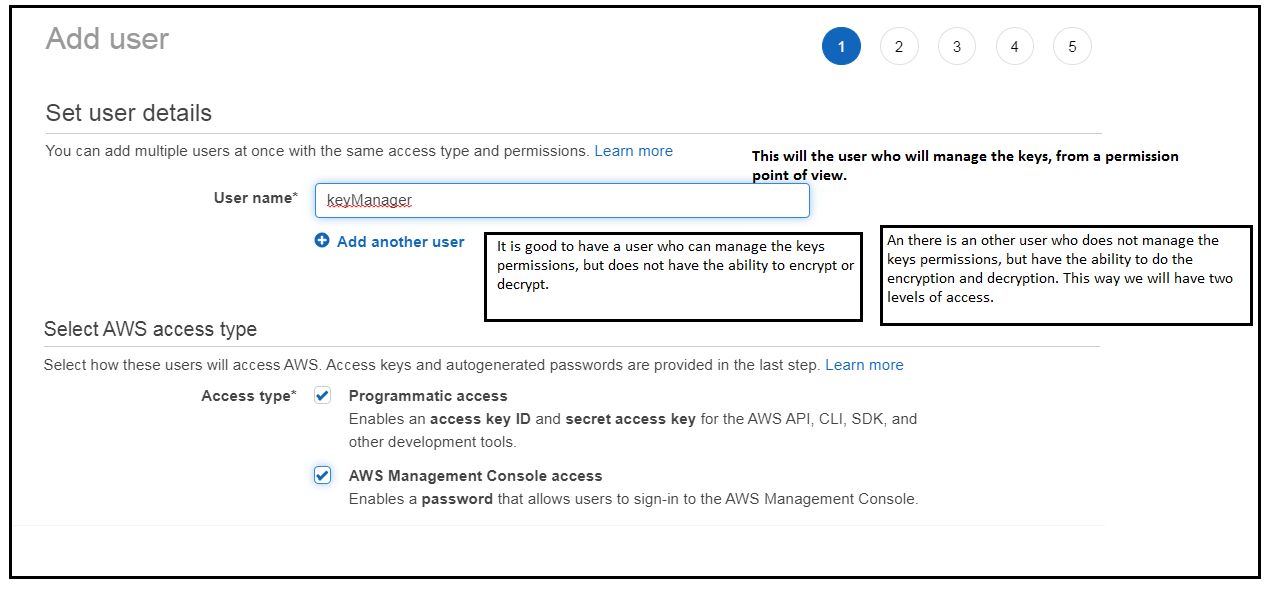
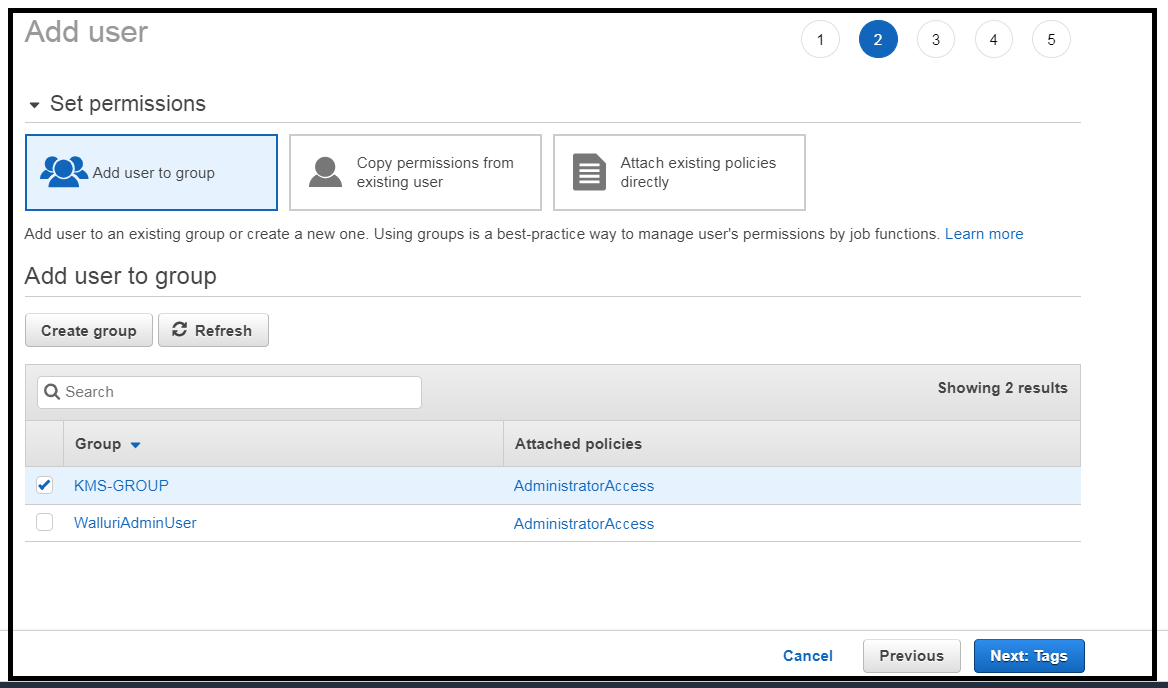
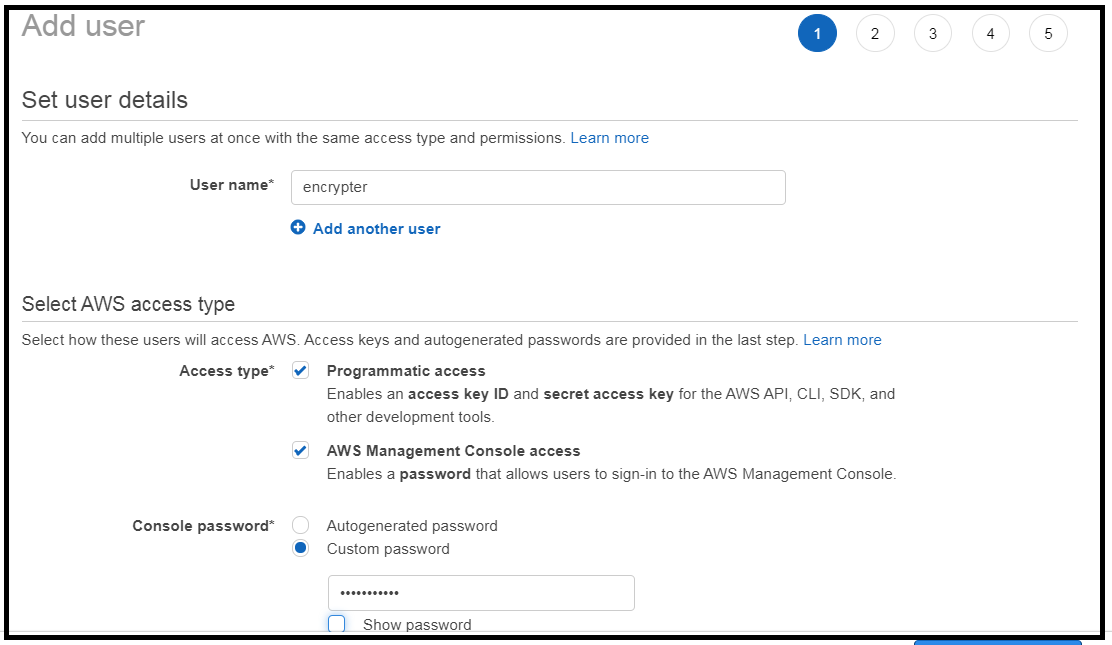
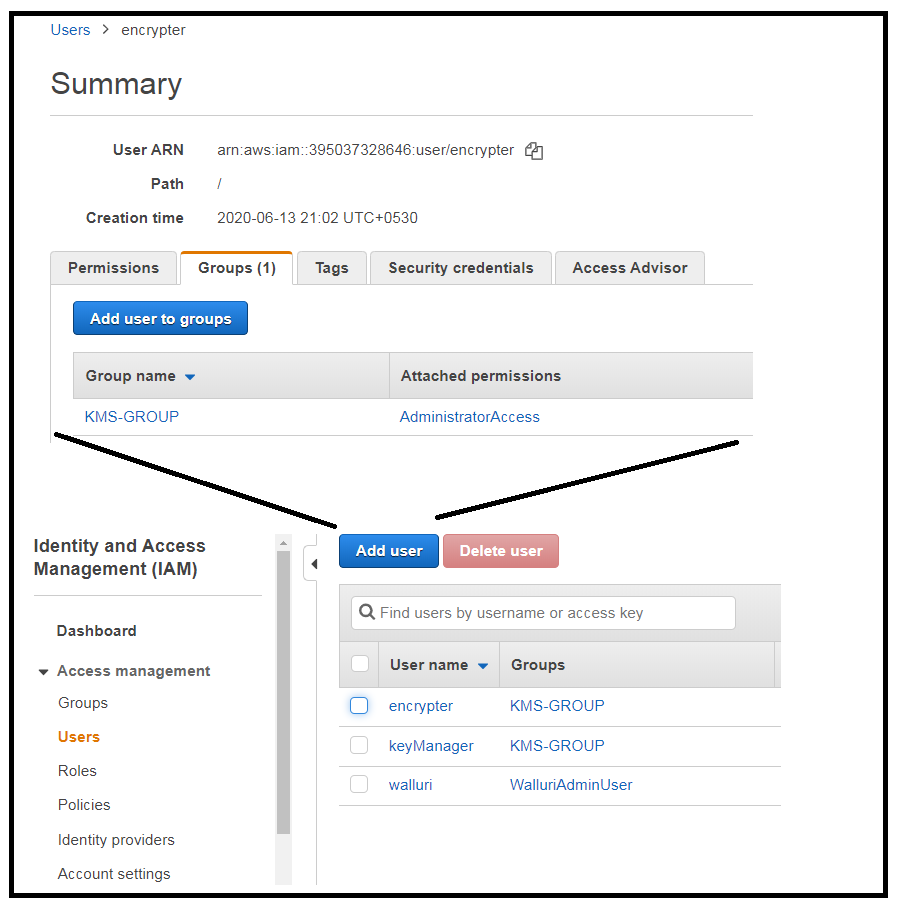
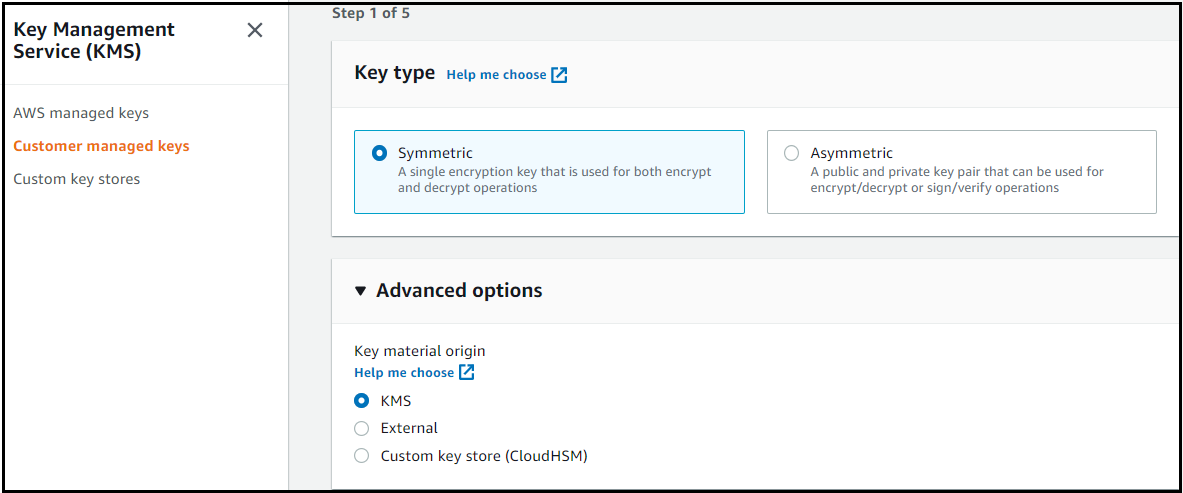
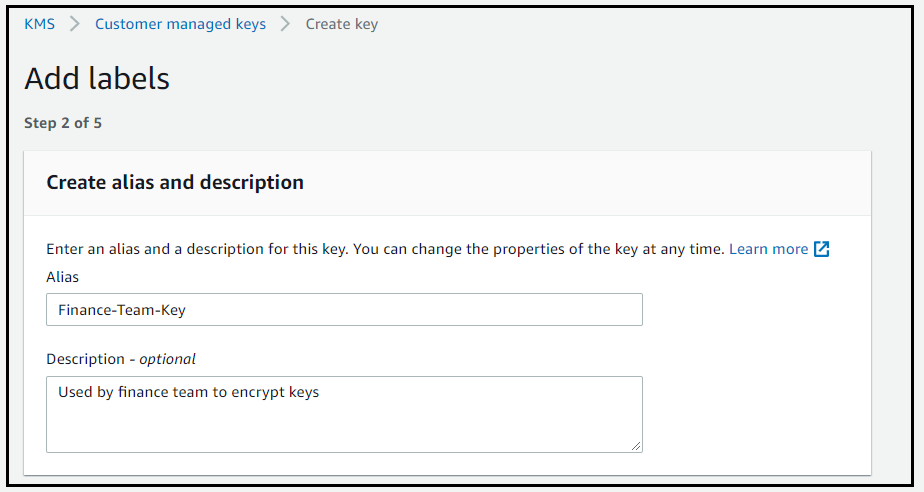
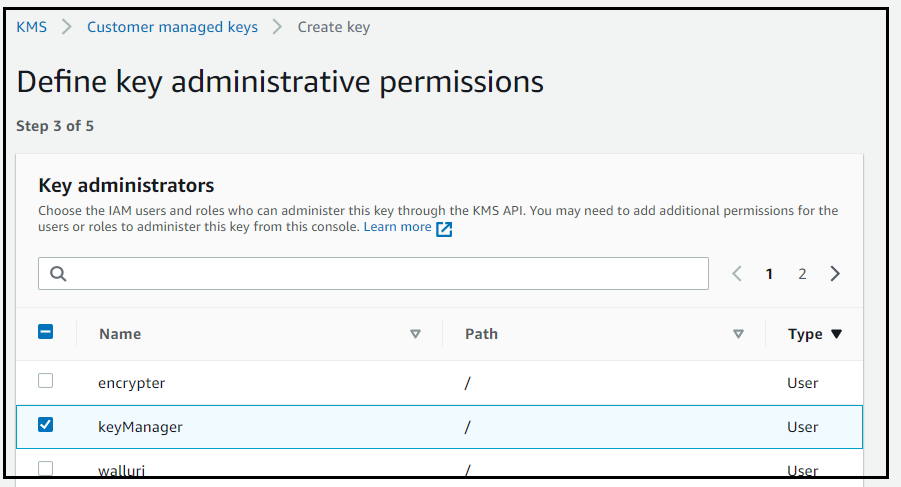
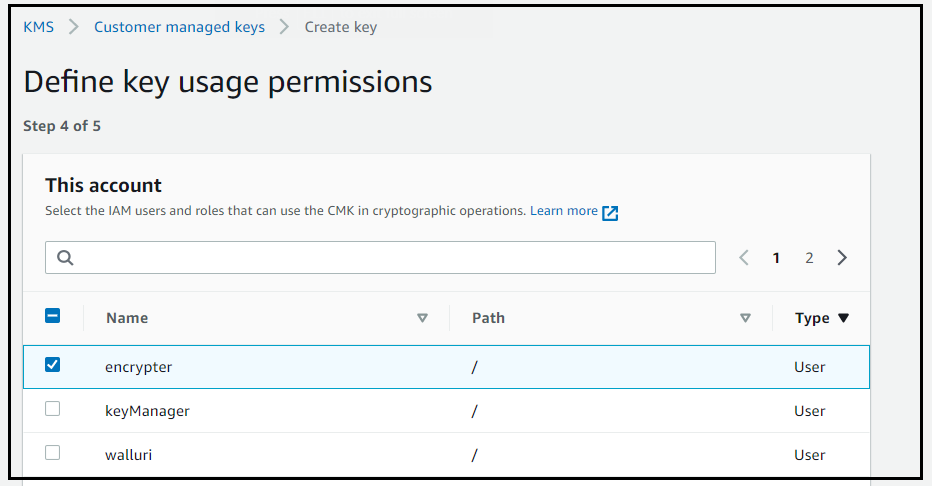
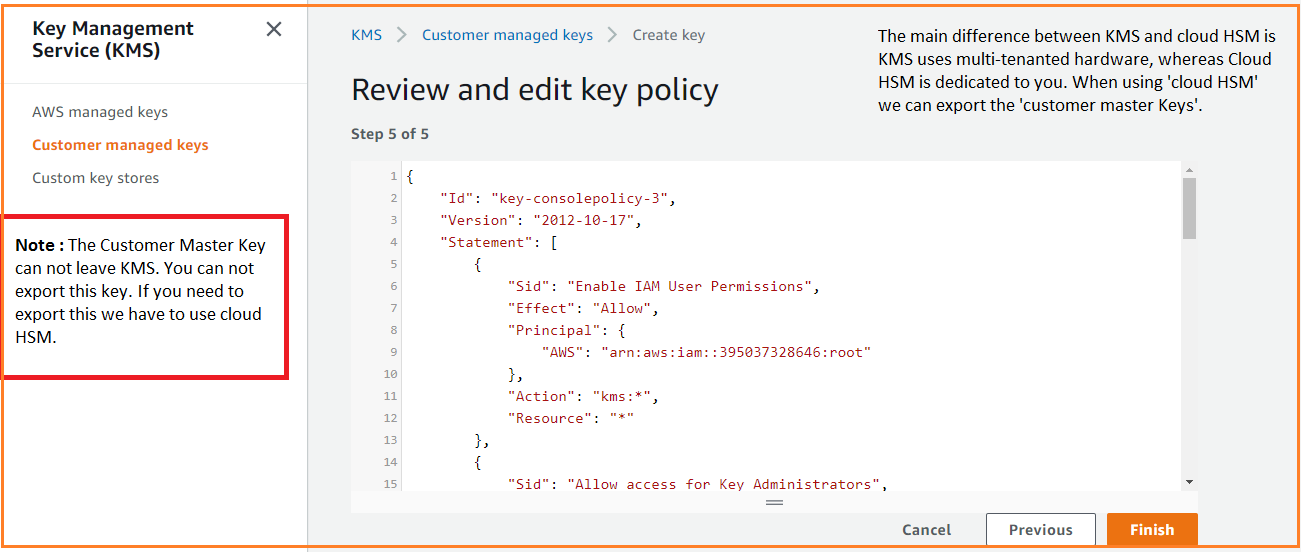
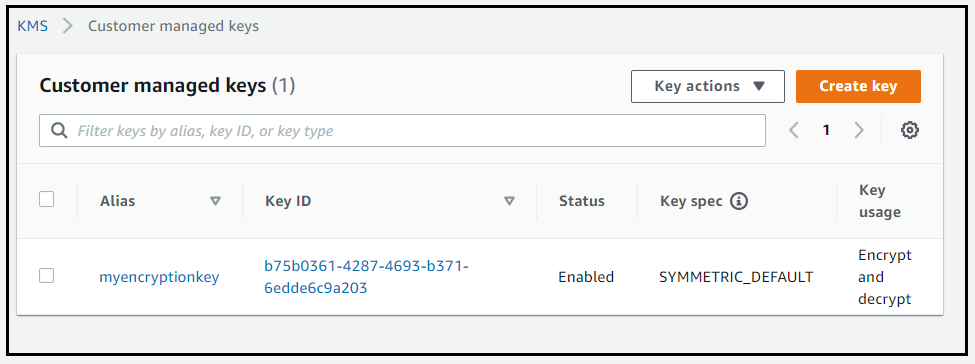
1.   
  
2. KMS   
– It is a managed service that make it easy for us to create and control the ‘encryption keys’ used to encrypt your data.  
– The Customer master keys that you create in AWS KMS are protected by HSM.  
– AWS KMS ensures the durability and physical security of our keys.  
– KMS is integrated with other AWS services like S3, RedShift, EBS , Elastic Transcoder, WorkMail, Amazon RDS and other services to encrypt your data with encryption keys that you manage.  
– KMs logs all the usage of keys to AWS Cloud Trail to find out who has accessed aws data or services.  
  
2.1   
LAB PART -1  
Create a group – Anyone in this group can administer KMS.  
  
Create a Key Manager user.   
  
Add the key manager to the group [The group which can administer KMS.]  
  
  
Create another user who can use the encryption keys.   
[In the end note down the *access key id* and *secret access key* of this user, As this is the user who is going to encrypt]  
  
Add this user also to the same group.  
  
  
LAB PART -2  
Lets create a key.  
  
  
  
  
  
  
  
  
  
  
  
  


LAB PART -3  
Making KMS API calls.  
  
Before we proceed further, Create an EC2 instance and run aws configure and login with the access key id and secret access key as that of the ‘encrypter’ - user.  
  
//encrypt the file  
aws kms encrypt --key-id AKIAVX6QPIUDCNN25FXJ --plaintext fileb://plaintext1.txt --output text --query CiphertextBlob | base64 --decode > encryptedsecret.txt

//decrypt the file

aws kms decrypt --ciphertext-blob fileb://encryptedsecret.txt --output text --query Plaintext | base64 --decode > decryptedsecret.txt

//re-encrypt the encrypted file without creating the plain text file in the process.

aws kms re-encrypt --destination-key-id AKIAVX6QPIUDCNN25FXJ --ciphertext-blob fileb://encryptedsecret.txt | base64 > newencryption.txt

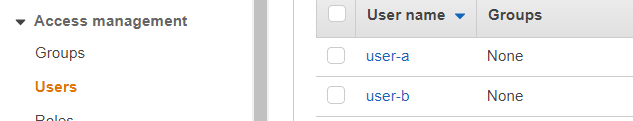
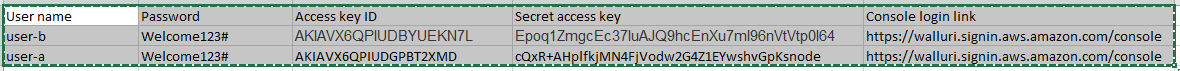
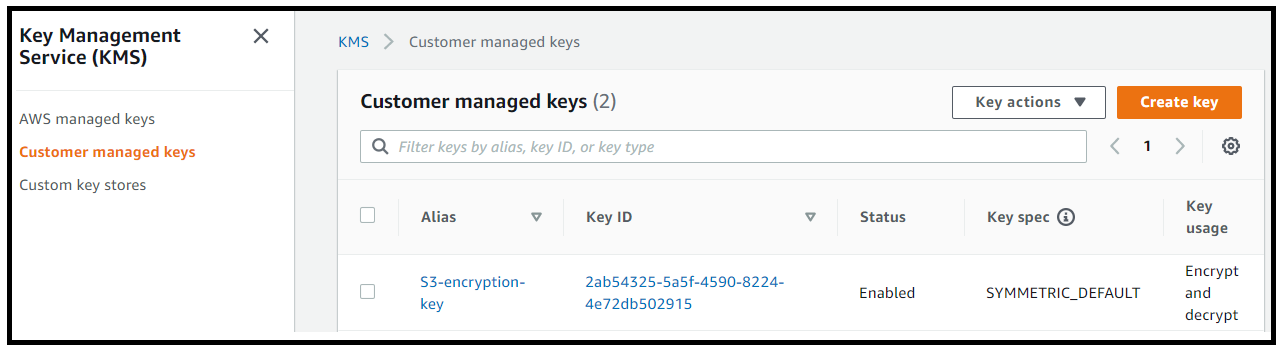
// You need the right permissions to rotate the key

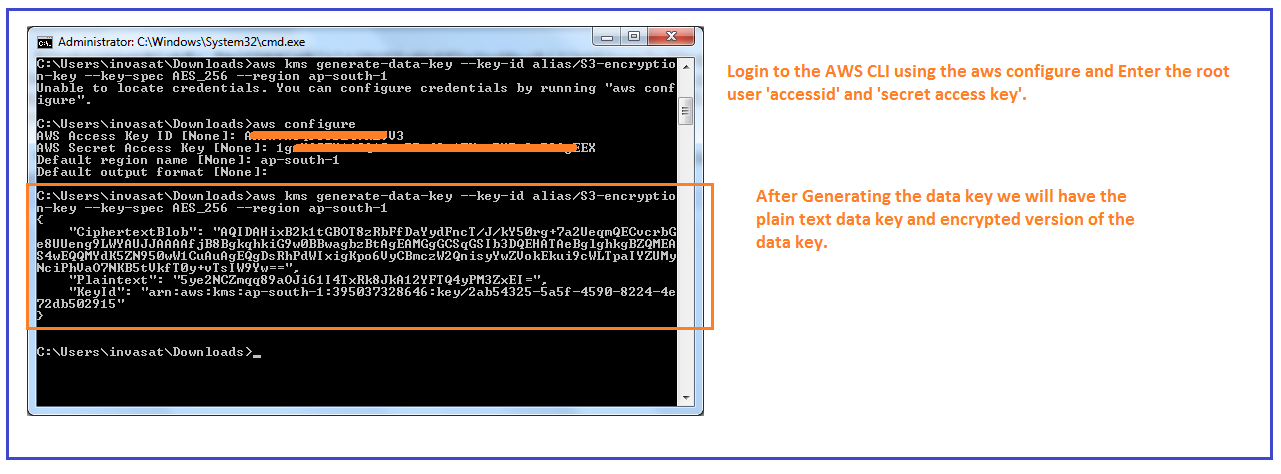
aws kms enable-key-rotation --key-id AKIAVX6QPIUDCNN25FXJ

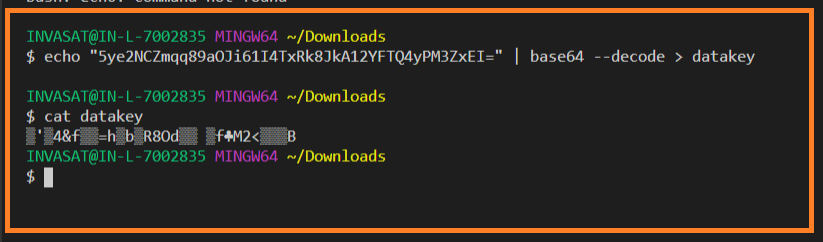
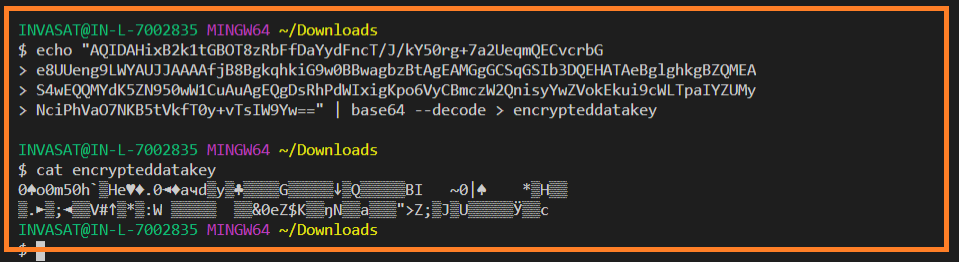
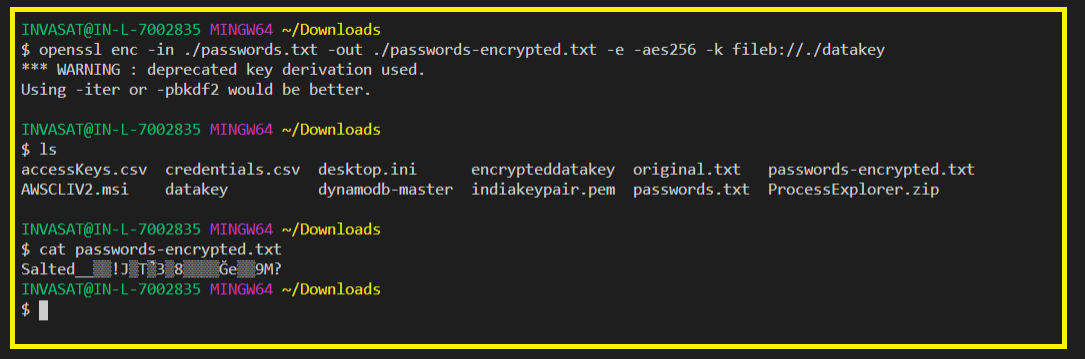
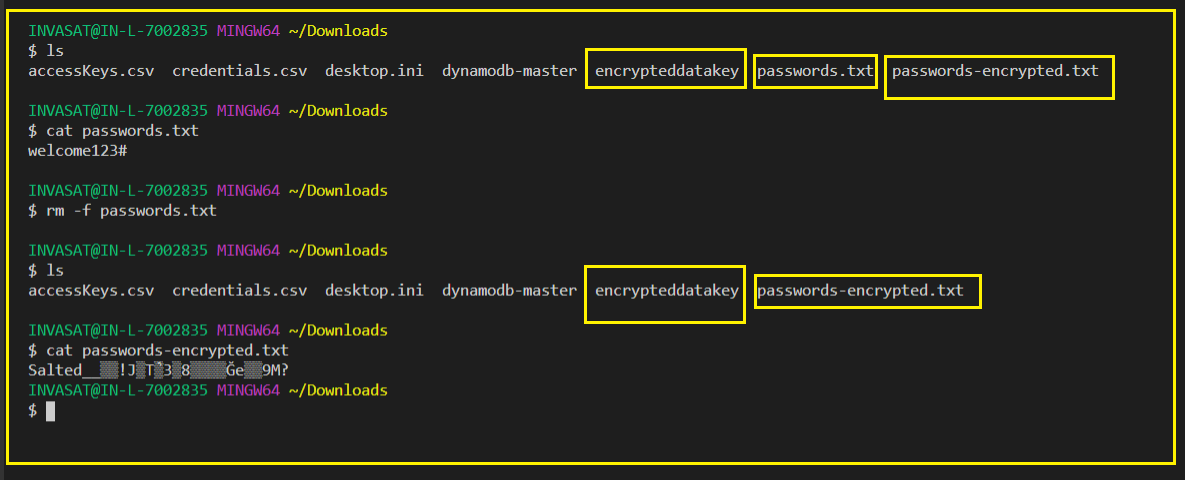
Note :

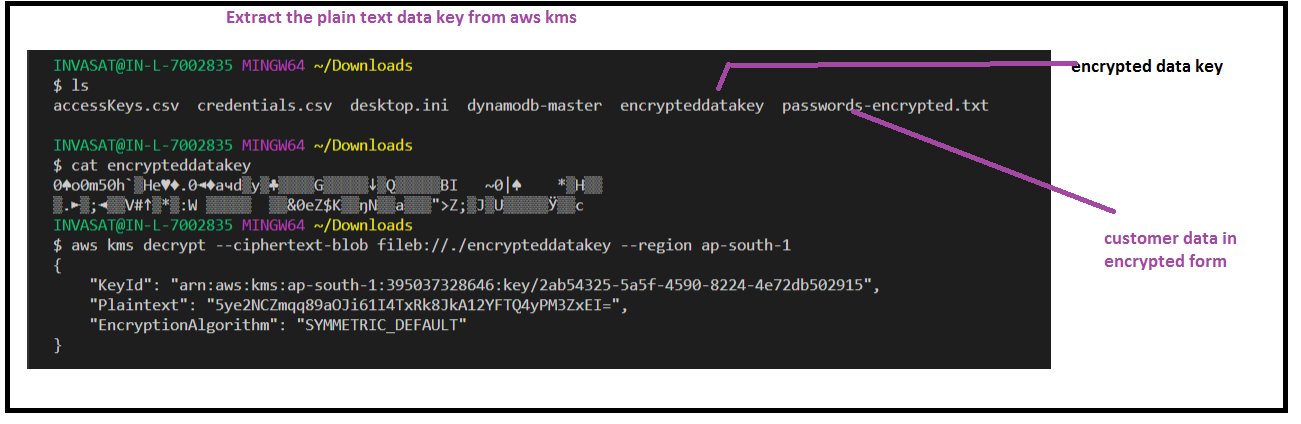
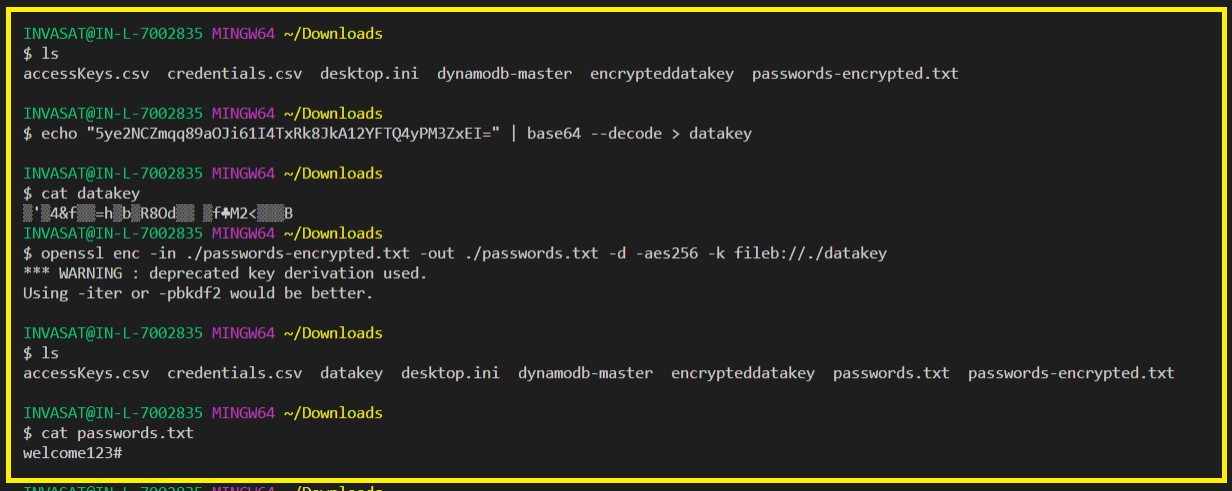
AKIAVX6QPIUDCNN25FXJ = encrypter key id

3. ENVELOPE ENCRYPTION  
  
- It is the process of encrypting the envelope key  
- Envelope Key / Data keys : It is the key used to encrypt you data.  
- CMK : Customer master Key – This key used by AWS to encrypt the envelope key. This can encrypt text which is under 4Kilo Bytes only.  
- Usage : We use the KMS service and the master key to decrypt our data-key, We use this key to decrypt our data.  
- AWS KMs helps you to protect your master key by storing and managing them securely.  
- The CMK never leave the AWS KMS validated HSM’s, unencrypted.  
- Note : the CMK can be AWS managed or Customer Managed.

4. LAB : ENCRYPTION AND DECRYPTION USING CLI.  
We use AWS KMS just for storing the encryption keys.  
  
4.1 Create two users user-a and user-b.  
  
  
  
  
  
4.2 Create a CMK(Customer Master Key) with Administrative and Key usage permissions to user-a.  
  
  
4.3 As CMK can not encrypt more than 4Kilo Bytes lets create a data-key. [AWS cli must already be installed]  
Command to generate Data key :  
aws kms generate-data-key --key-id alias/S3-encryption-key --key-spec AES\_256 --region ap-south-1



Note : The ‘plain text – Data key ‘and the ‘cipher text Data-key‘ are base 64 encoded by default.  
If we want to encrypt our data we need to decode these keys first.  
  
4.4 Decode the ‘plain text – Data key’  
  
  
Decode the encrypted version of the data key.  
  
  
4.5 Now we have the ‘encrypted data key’ and the ‘data key’ to encrypt our data.  
We shall use the datakey to encrypt our data + we will remove/delete the datakey.  
If the datakey is found by anyone they can easily decrypt our data.  
So along with our data we will store the ‘encrypted data key’.  
Encrypt our data/file (passwords.txt) with datakey.  
  
  
4.6 Let us delete the data key + Let us remove our un-encrypted data files i.e passwords.txt  


4.7 Decrypting data with encrypted data key  
How to decrypt your data in encrypted form back to normal text with ‘encrypted data key’.  
AWS KMS keeps a reference between the ‘encrypted data key’ and ‘plain text data key’.  
  
  
  
  
  
5. AWS Encryption SDK.  
Aws provides 3 types of encryption SDK’s.  
- DynamoDB Encryption client : When working directly with DynamoDB.  
- S3 Encryption client :   
- AWS Encryption SDK : When you want to encrypt data in general, be it the browser / EC2 instance or the micro service.