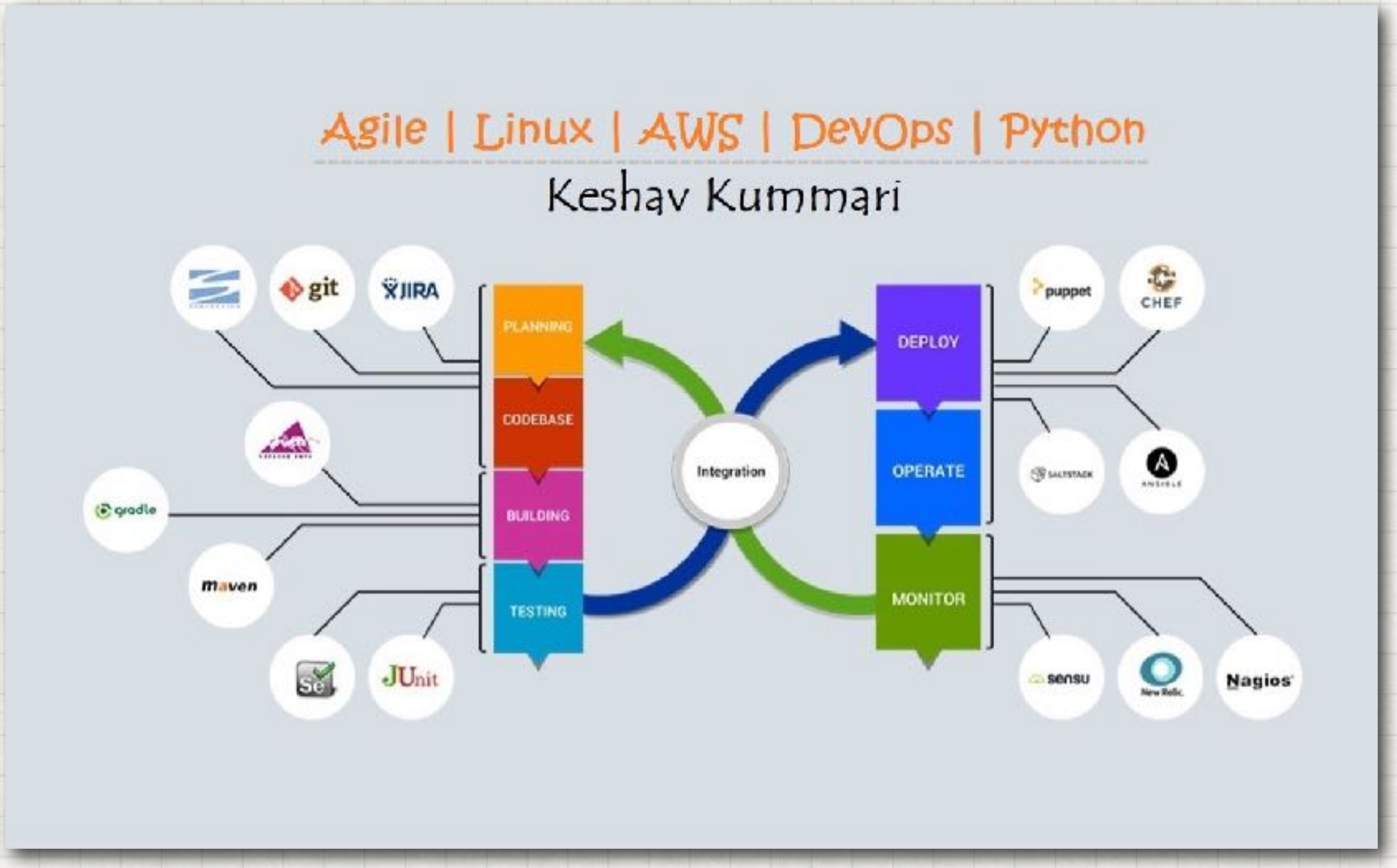
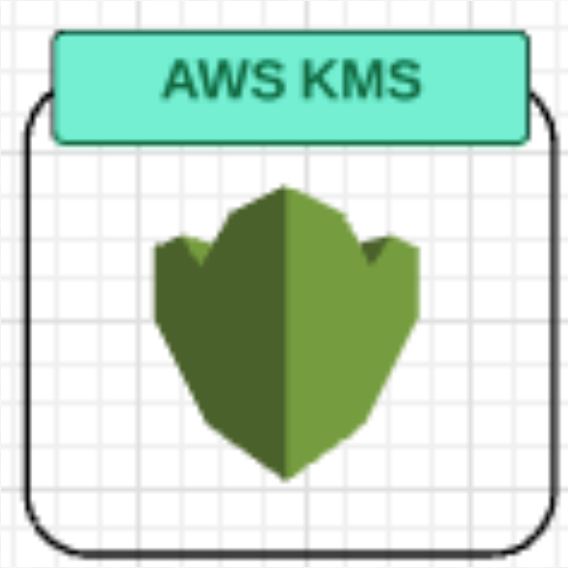
AWS

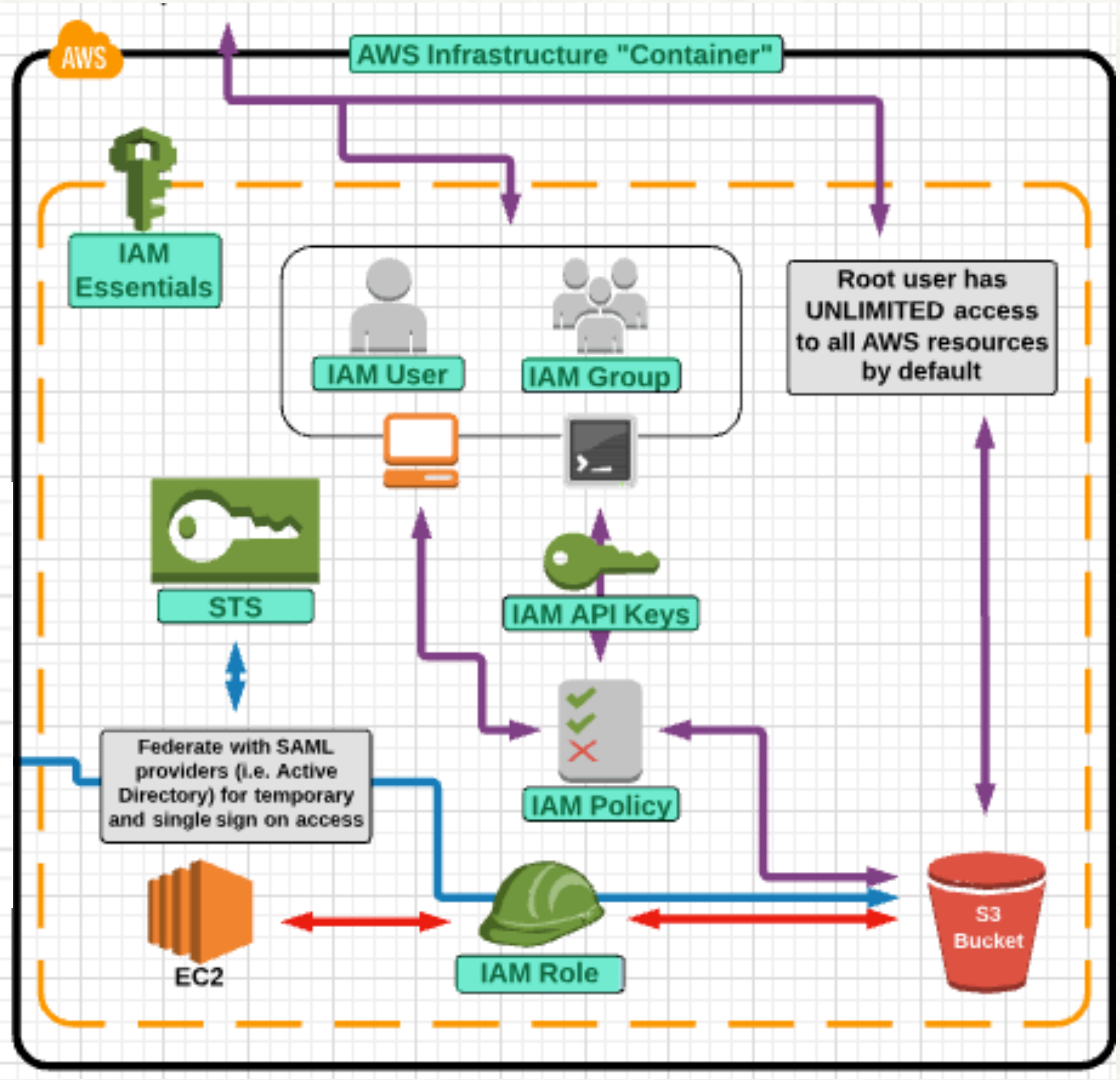
KESHAV KUMMARI



AWS CLOUD ARCHITECTURE



KMS


AWS
AWS Infrastructure "Container"
IAM Essentials
IAM User
IAM Group
Root user has UNLIMITED access to all AWS resources by default
STS
IAM API Keys
Federate with SAML providers (i.e. Active Directory) for temporary and single sign on access
IAM Policy
EC2
IAM Role
S3 Bucket

AWS

AWS - KMS (KEY MANAGEMENT SERVICE)

- KMS is a managed service to Create and Control the encryption keys used to encrypt your data.

- AWS Key Management Service (**KMS**) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data, and uses FIPS 140-2 validated hardware security modules to protect the security of your keys.

- AWS Key Management Service is integrated with most other AWS services to help you protect the data you store with these services.

- AWS Key Management Service is also integrated with AWS CloudTrail to provide you with logs of all key usage to help meet your regulatory and compliance needs.

KMS CONCEPTS

1. Customer Master Keys(CMK's):

2. Typically, CMKs generate, encrypt data keys that you use outside of AWS KMS to encrypt your data

3. There are 2 types of CMKs:

1. Customer-Managed : CMK's you create, enable/disable, rotate, and which manage the policies that allow access to use the CMK

2. AWS-Managed : CMK's that are created, managed, and used by AWS services integrated with KMS(These CMK's are named like: aws/service-name i.e. aws/s3)

4. Data Keys:

1. Encryption keys for encrypting large amounts of data encryption keys

2. AWS CMKs can generate, encrypt, and decrypt data keys

3. KMS does not manage or store your data keys - you must use and manage them inside your application

4. KMS can not use data keys to encrypt data for you

5. Envelope Encryption :

1. Plain text data is encrypted with a data key

2. Data keys are encrypted with a key encryption key(KEK)

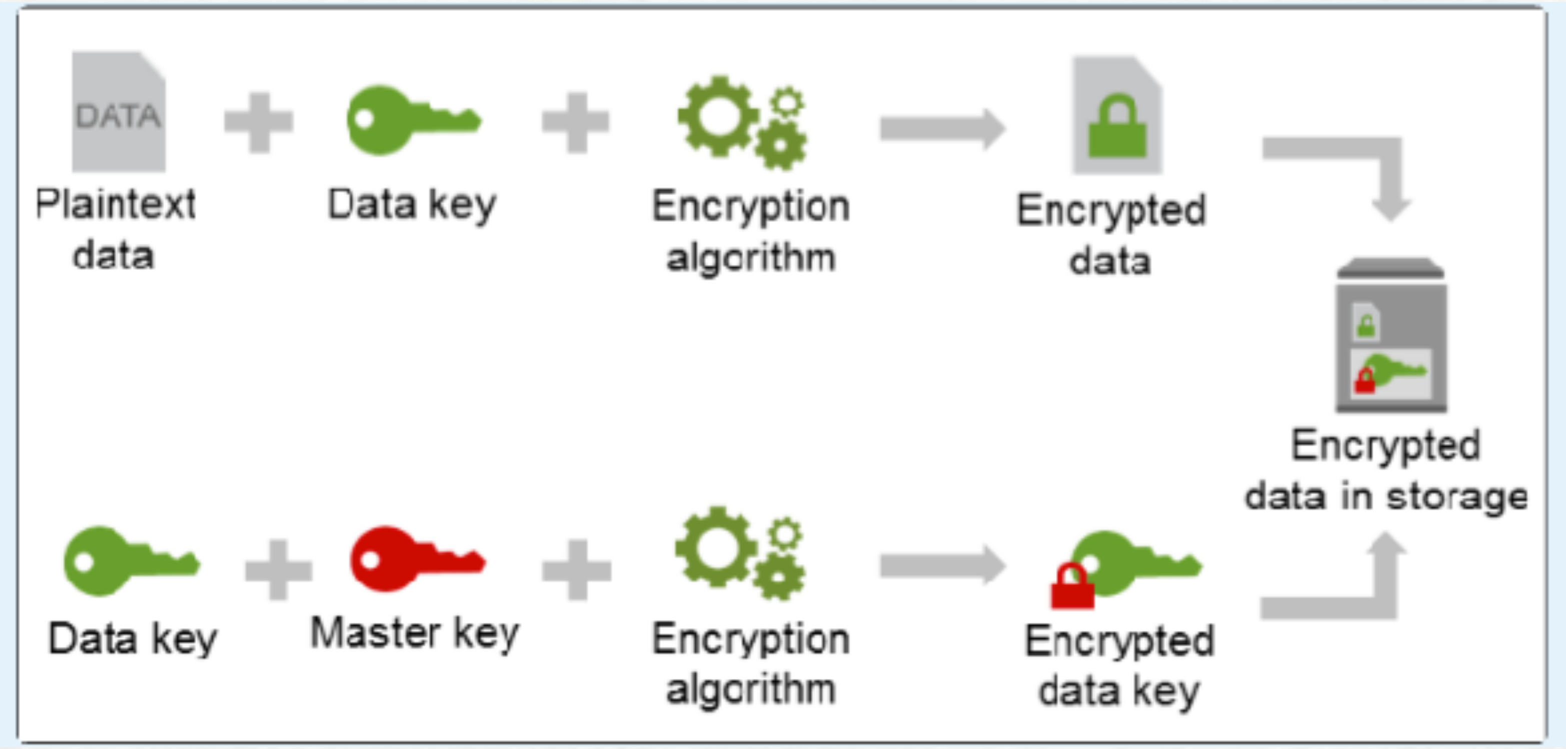
3. A KEK may be encrypted by another KEK, but eventually there is a master key(THe KMS CMK in this case) that decrypts one or more keys

KMS API ACTIONS

1. Encrypt - Encrypt plain text using a CMK

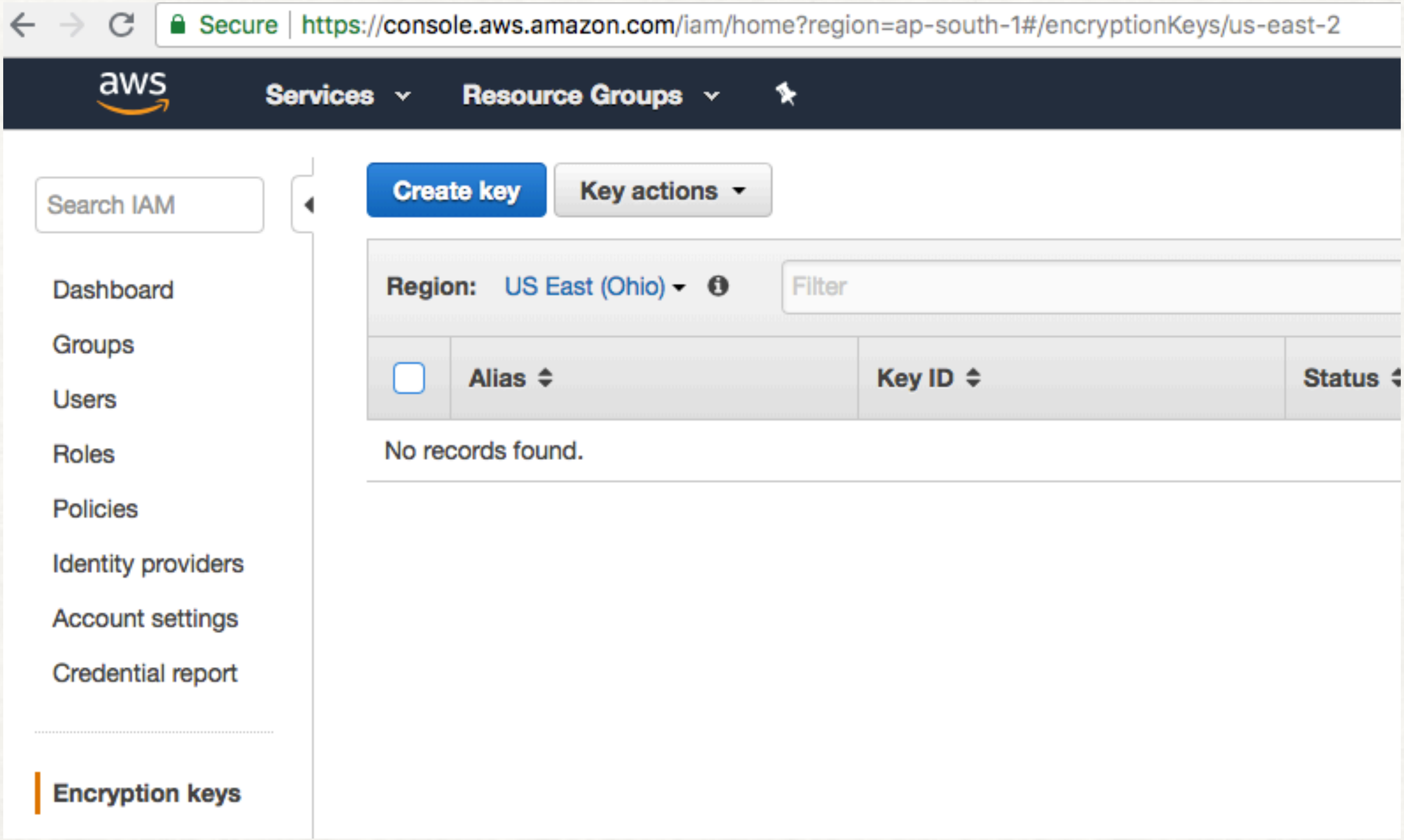
2. GenerateDataKey - Uses a CMK to return a plain text and cipher text version of a data encryption key

3. Decrypt - Decrypts ciphertext that was encrypted with the Encrypt, GenerateDataKey or GenerateDataKeyWithoutPlaintext API actions.



KMS

GO TO AWS MGMT CONSOLE >> IAM >> ENCRYPTION KEYS



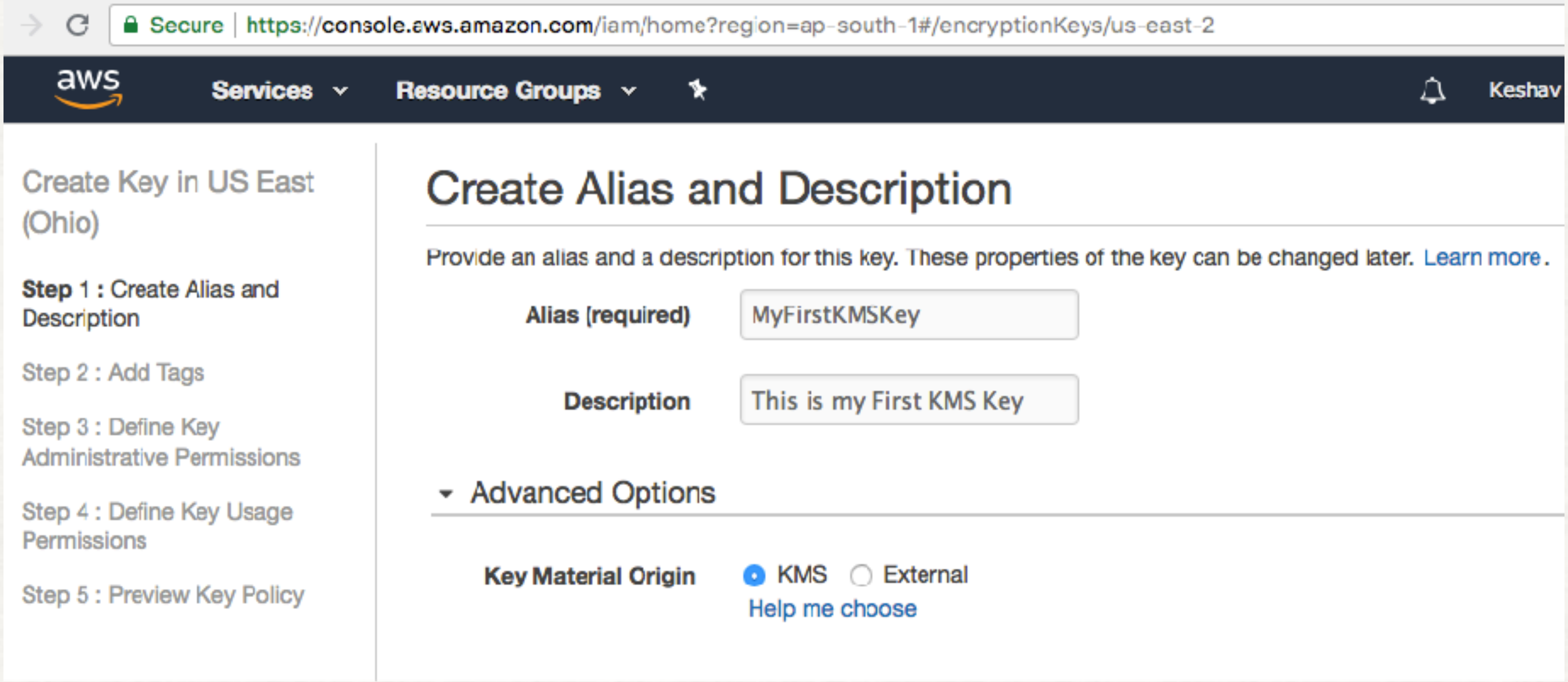
Region: US East (Ohio)

□	Alias	Key ID	Status

No records found.

Select any region & click on Create Key

FILL THE DETAILS & CONTINUE



Secure | https://console.aws.amazon.com/iam/home?region=ap-south-1#/encryptionKeys/us-east-2
aws
Services
Resource Groups
Keshav

Create Key in US East (Ohio)

Step 1 : Create Alias and Description

Step 2 : Add Tags

Step 3 : Define Key Administrative Permissions

Step 4 : Define Key Usage Permissions

Step 5 : Preview Key Policy

Create Alias and Description

Provide an alias and a description for this key. These properties of the key can be changed later. Learn more.

Alias (required) MyFirstKMSKey

Description This is my First KMS Key

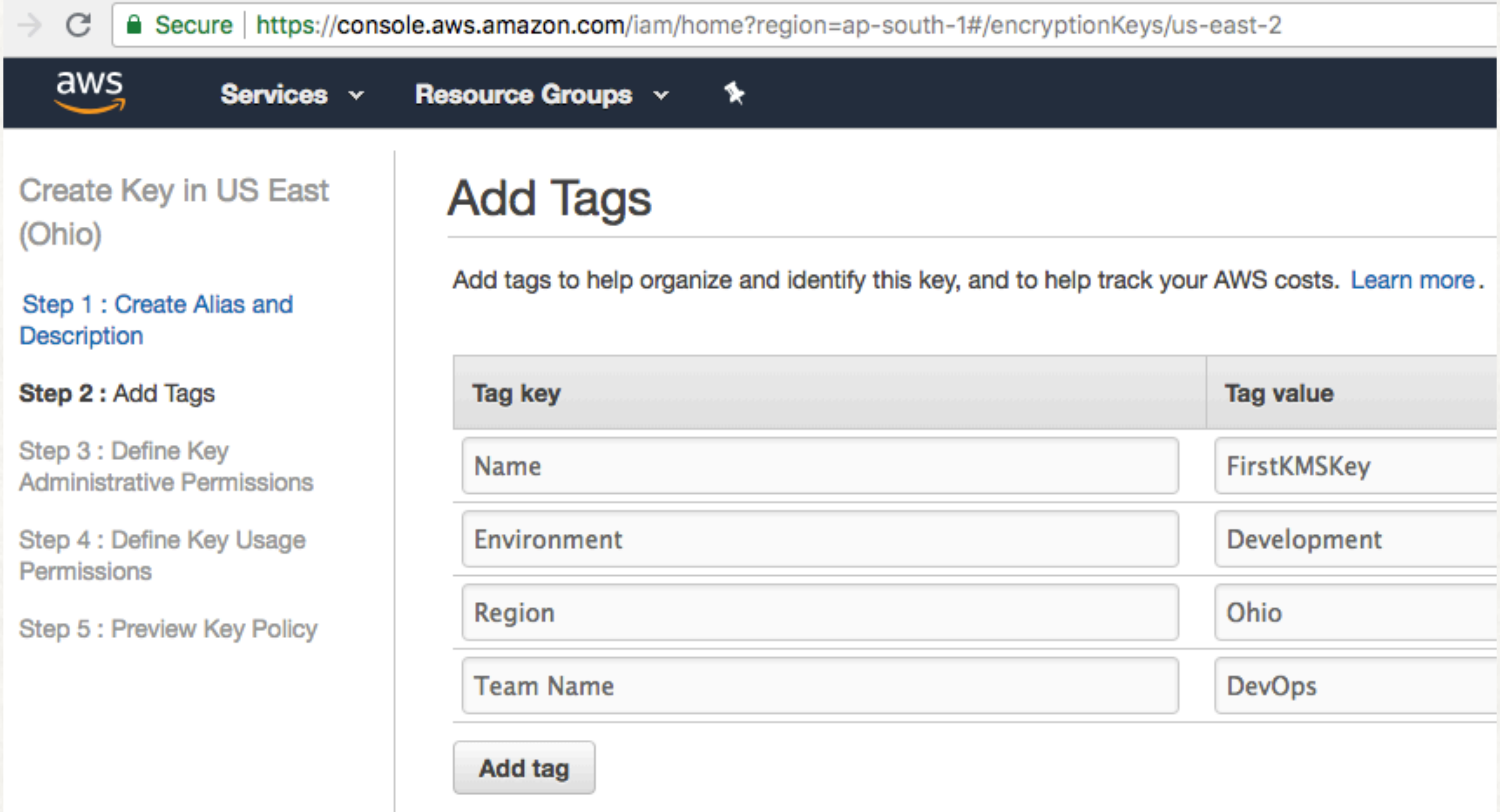
Advanced Options

Key Material Origin ● KMS ○ External

Help me choose

KMS

TAGS



https://console.aws.amazon.com/iam/home?region=ap-south-1#/encryptionKeys/us-east-2

Create Key in US East (Ohio)

Step 1 : Create Alias and Description

Step 2 : Add Tags

Step 3 : Define Key Administrative Permissions

Step 4 : Define Key Usage Permissions

Step 5 : Preview Key Policy

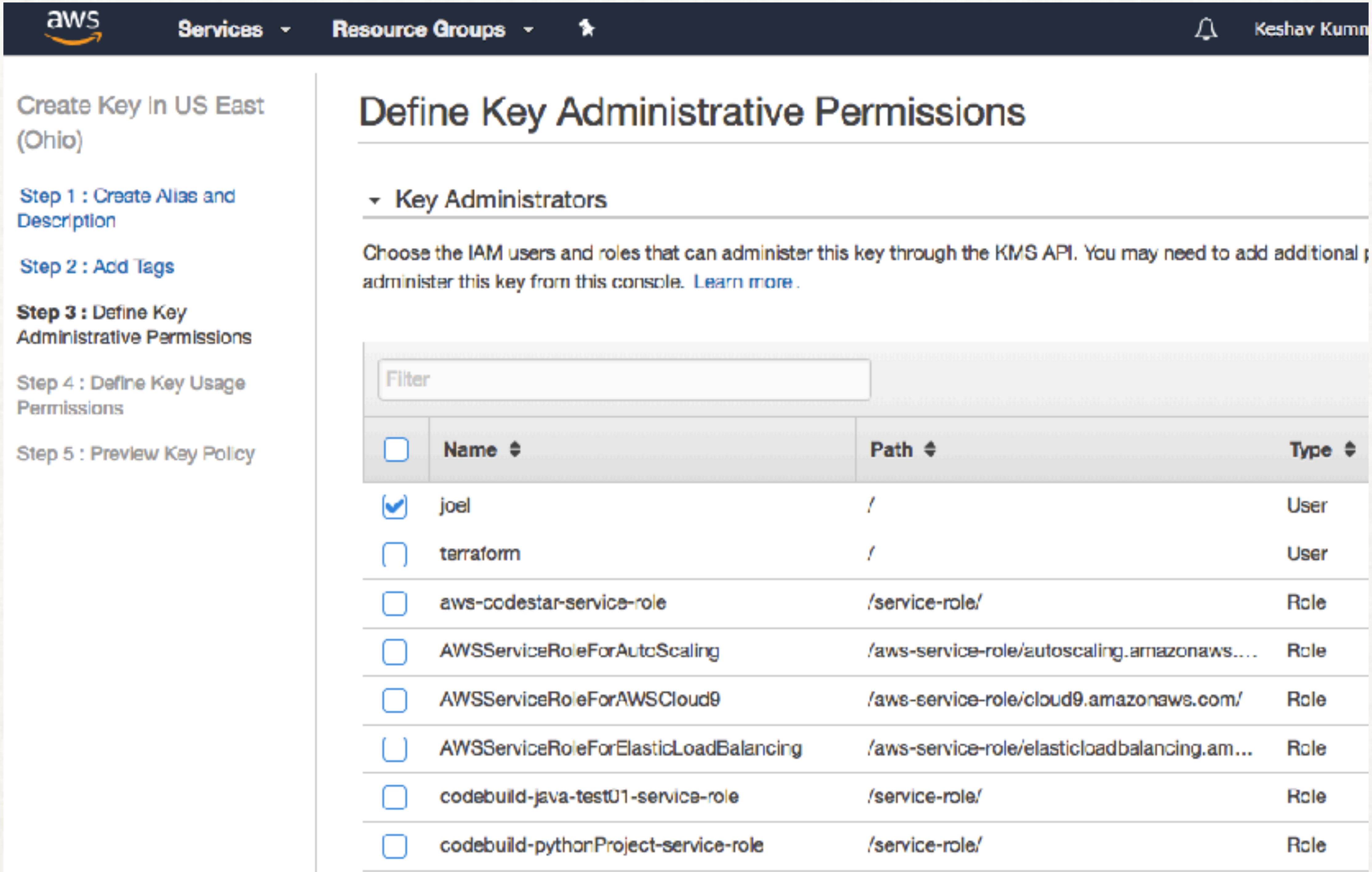
Add Tags

Add tags to help organize and identify this key, and to help track your AWS costs. Learn more.

Tag key	Tag value
Name	FirstKMSKey
Environment	Development
Region	Ohio
Team Name	DevOps

KMS

ADD KEY ADMIN



Create Key in US East (Ohio)

Step 1 : Create Alias and Description

Step 2 : Add Tags

Step 3 : Define Key Administrative Permissions

Step 4 : Define Key Usage Permissions

Step 5 : Preview Key Policy

Define Key Administrative Permissions

Key Administrators

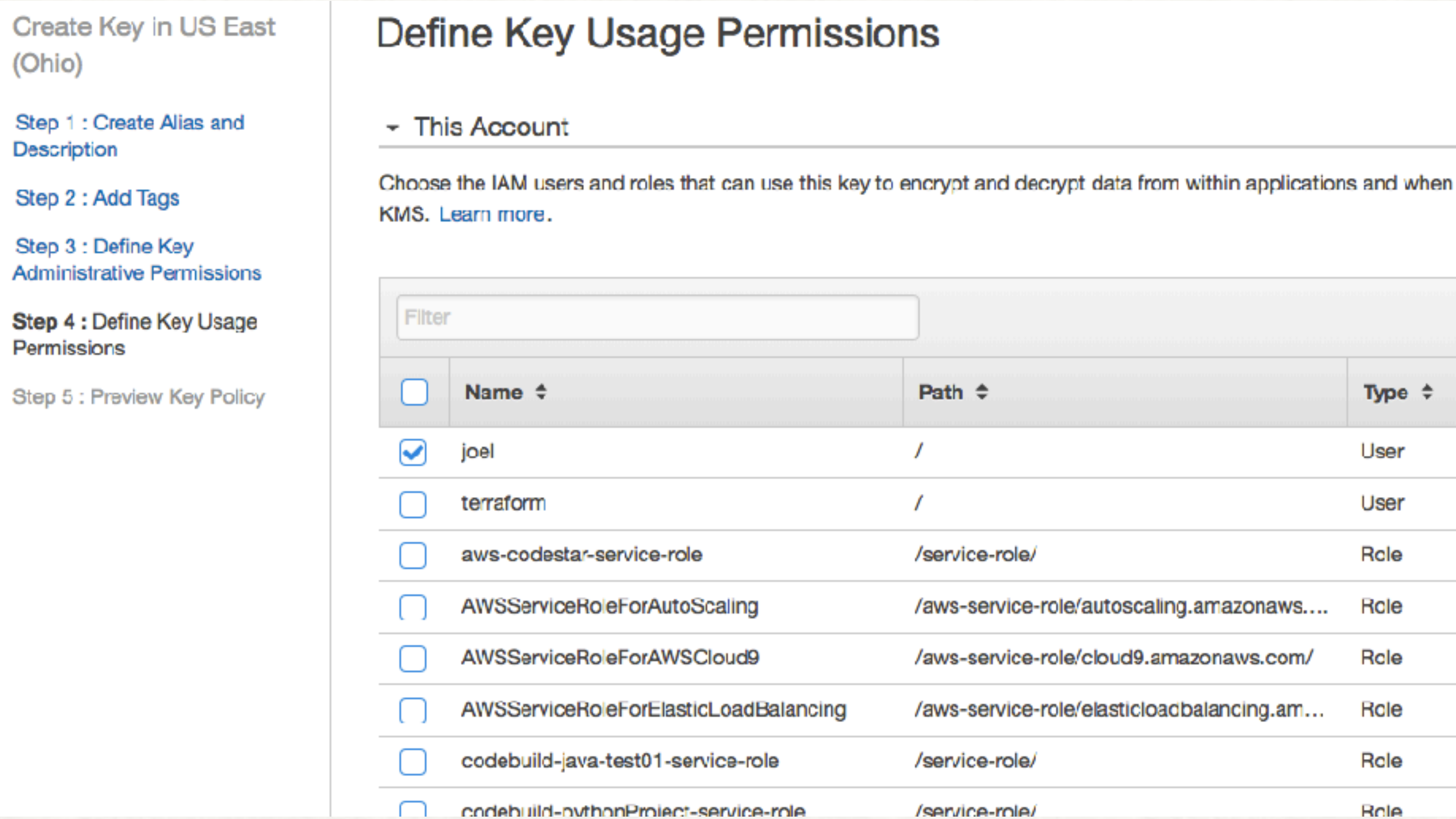
Choose the IAM users and roles that can administer this key through the KMS API. You may need to add additional p administer this key from this console. Learn more.

Filter

□	Name	Path	Type
☑	joel	/	User
□	terraform	/	User
□	aws-codestar-service-role	/service-role/	Role
□	AWSServiceRoleForAutoScaling	/aws-service-role/autoscaling.amazonaws....	Role
□	AWSServiceRoleForAWSCloud9	/aws-service-role/cloud9.amazonaws.com/	Role
□	AWSServiceRoleForElasticLoadBalancing	/aws-service-role/elasticloadbalancing.am...	Role
□	codebuild-java-test01-service-role	/service-role/	Role
□	codebuild-pythonProject-service-role	/service-role/	Role

Click on Next

PERMISSIONS



Create Key in US East (Ohio)

Step 1 : Create Alias and Description
Step 2 : Add Tags
Step 3 : Define Key Administrative Permissions
Step 4 : Define Key Usage Permissions
Step 5 : Preview Key Policy

Define Key Usage Permissions

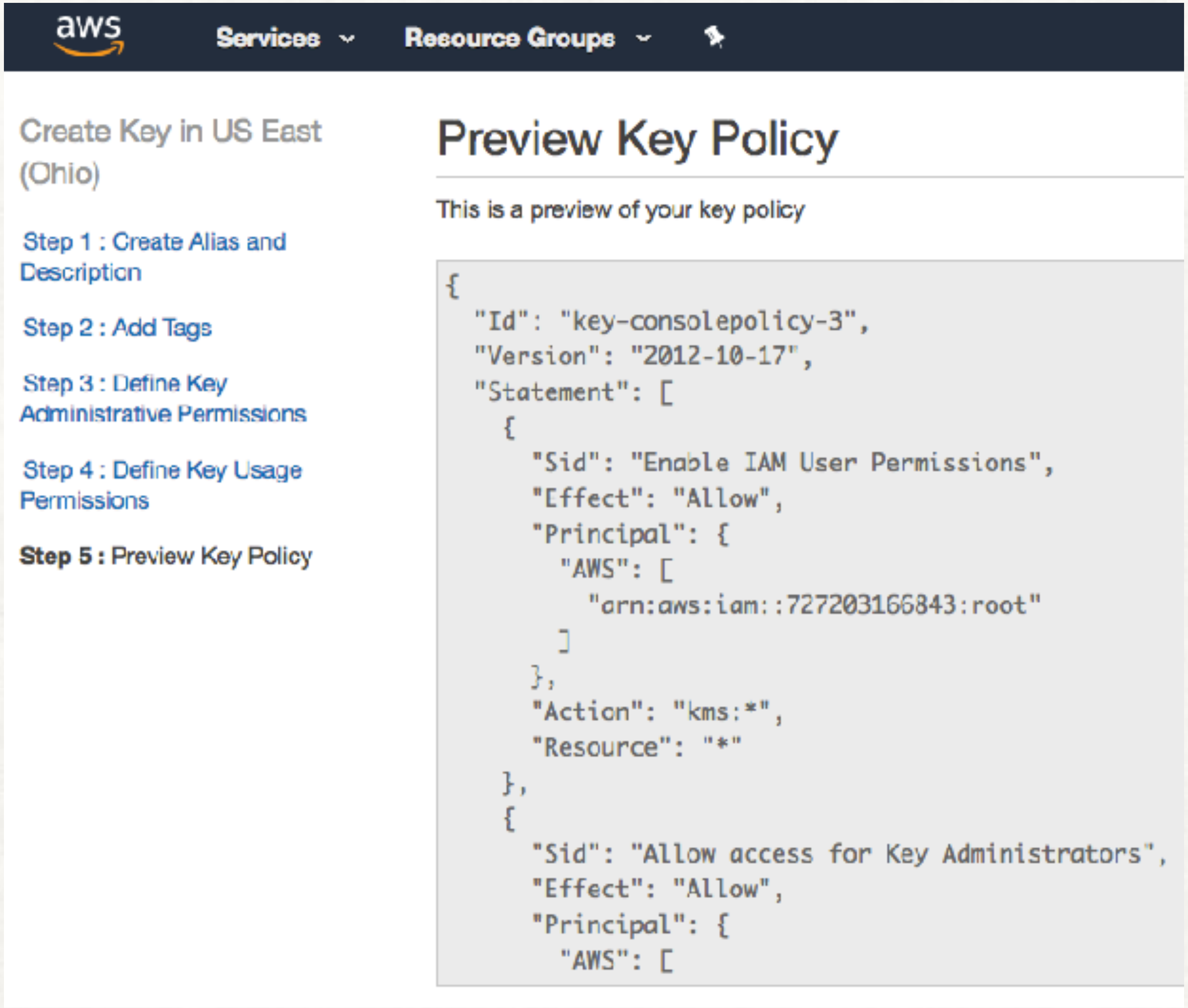
This Account

Choose the IAM users and roles that can use this key to encrypt and decrypt data from within applications and when KMS. Learn more.

□	Name	Path	Type
☑	joel	/	User
□	terraform	/	User
□	aws-codestar-service-role	/service-role/	Role
□	AWSServiceRoleForAutoScaling	/aws-service-role/autoscaling.amazonaws....	Role
□	AWSServiceRoleForAWSCloud9	/aws-service-role/cloud9.amazonaws.com/	Role
□	AWSServiceRoleForElasticLoadBalancing	/aws-service-role/elasticloadbalancing.am...	Role
□	codebuild-java-test01-service-role	/service-role/	Role
□	codebuild-pythonProject-service-role	/service-role/	Role

Click on Next

KEY POLICY



Create Key in US East (Ohio)

Step 1 : Create Alias and Description

Step 2 : Add Tags

Step 3 : Define Key Administrative Permissions

Step 4 : Define Key Usage Permissions

Step 5 : Preview Key Policy

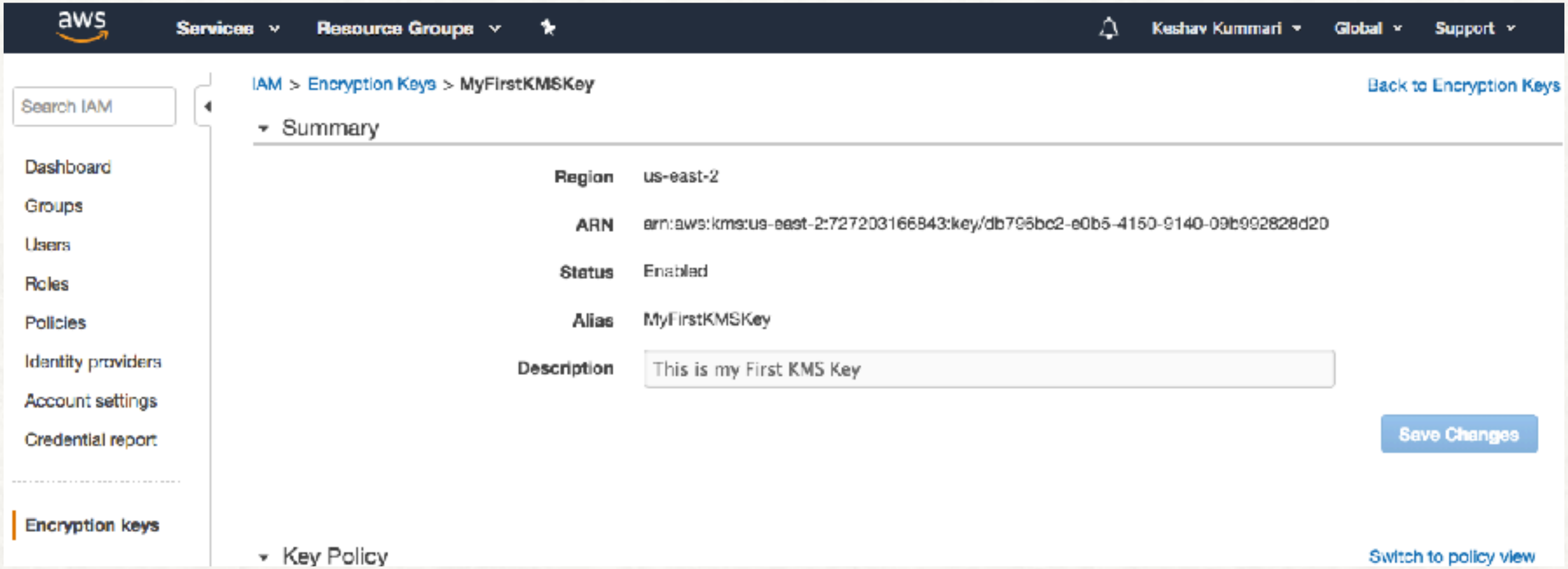
Preview Key Policy

This is a preview of your key policy

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::727203166843:root"
        ]
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
```

Click on Finish

SUMMARY OF KMS KEY



IAM > Encryption Keys > MyFirstKMSKey

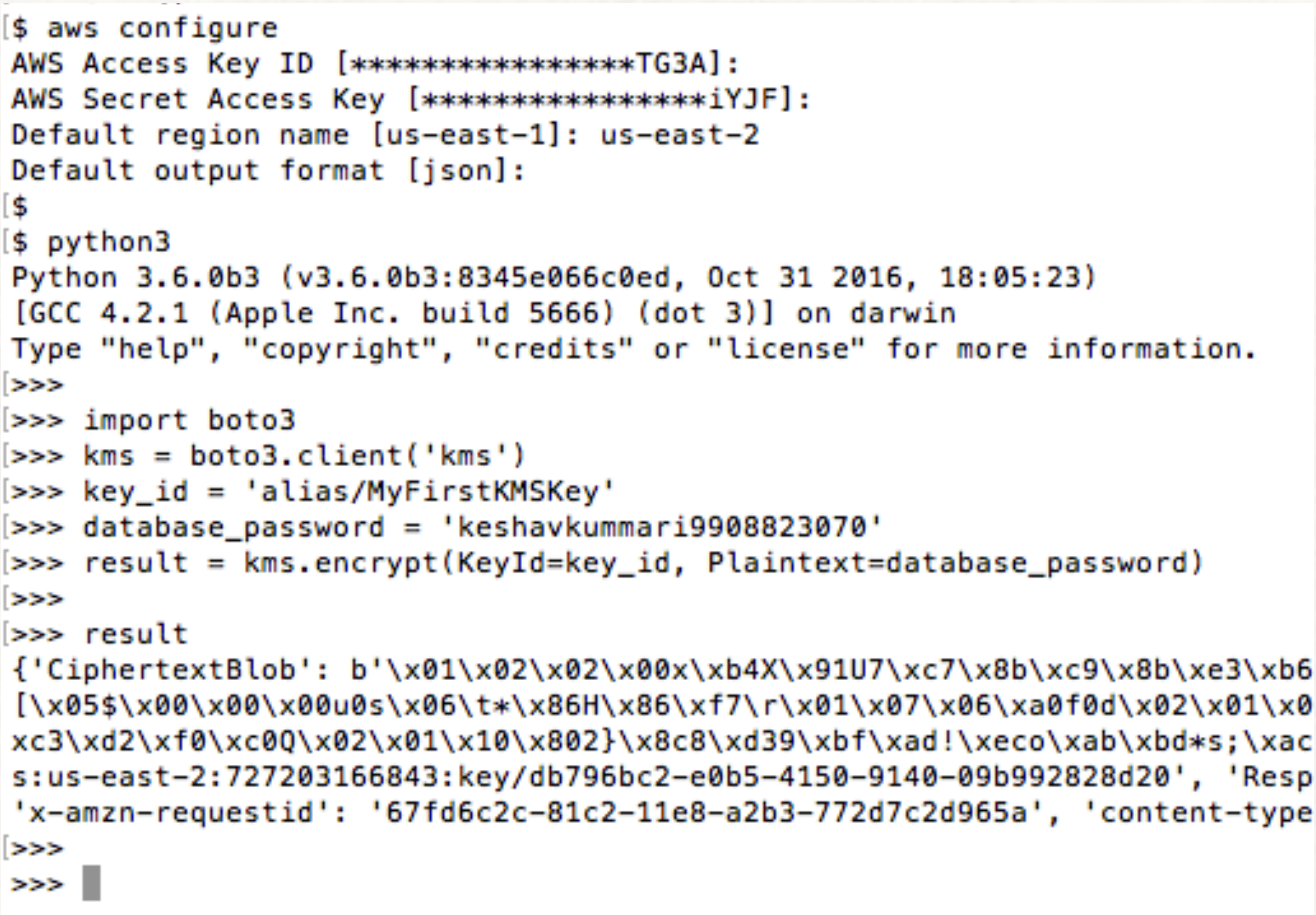
Summary

Region	us-east-2
ARN	arn:aws:kms:us-east-2:727203166843:key/db796bc2-e0b5-4150-9140-09b992828d20
Status	Enabled
Alias	MyFirstKMSKey
Description	This is my First KMS Key

Key Policy

KMS

EXECUTE BELOW COMMANDS FROM LAPTOP



```
[$ aws configure
AWS Access Key ID [****************TG3A]:
AWS Secret Access Key [****************iYJF]:
Default region name [us-east-1]: us-east-2
Default output format [json]:
[$
[$ python3
Python 3.6.0b3 (v3.6.0b3:8345e066c0ed, Oct 31 2016, 18:05:23)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>>
[>>> import boto3
[>>> kms = boto3.client('kms')
[>>> key_id = 'alias/MyFirstKMSKey'
[>>> database_password = 'keshavkummari9908823070'
[>>> result = kms.encrypt(KeyId=key_id, Plaintext=database_password)
[>>>
[>>> result
{'CiphertextBlob': b'\x01\x02\x02\x00x\xb4X\x91U7\xc7\x8b\xc9\x8b\xe3\xb6
[\x05$\x00\x00\x00u0s\x06\t*\x86H\x86\xf7\r\x01\x07\x06\xa0f0d\x02\x01\x0
xc3\xd2\xf0\xc0Q\x02\x01\x10\x802}\x8c8\xd39\xbf\xad!\xeco\xab\xbd*s;\xac
s:us-east-2:727203166843:key/db796bc2-e0b5-4150-9140-09b992828d20', 'Resp
'x-amzn-requestid': '67fd6c2c-81c2-11e8-a2b3-772d7c2d965a', 'content-type
[>>>
>>> 
```

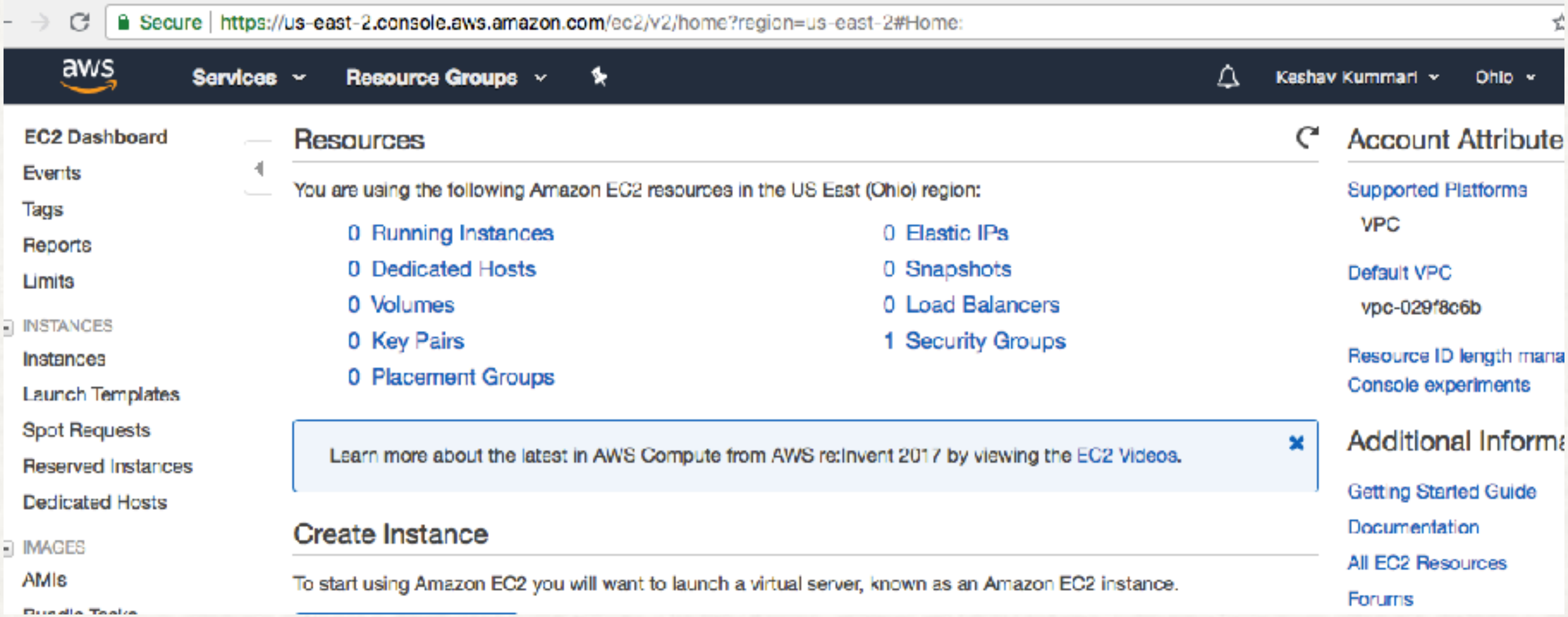
AWS

GENERATE KMS KEY TO CONNECT DB ETC.. FROM APPLICATIONS

```
>>> import boto3

>>> kms = boto3.client('kms')

>>> key_id = 'alias/MyFirstKMSKey'

>>> database_password = 'keshavkummari9908823070'

>>> result = kms.encrypt(KeyId=key_id, Plaintext=database_password)

>>> result

>>> encrypted_password = result['CiphertextBlob']

>>> encrypted_password

>>> decrypt_result = kms.decrypt(CiphertextBlob=encrypted_password)

>>> decrypt_result
```

STEP BY STEP PRACTICAL EXAMPLE

- Go to >> EC2 Dashboard >> Region >> Ohio
- Note: Because I’ve created KMS keys in Ohio Region



Secure | https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Home:

Resources

You are using the following Amazon EC2 resources in the US East (Ohio) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
0 Key Pairs	1 Security Groups
0 Placement Groups	

Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the EC2 Videos.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Account Attribute

Supported Platforms
VPC

Default VPC
vpc-029f8c6b

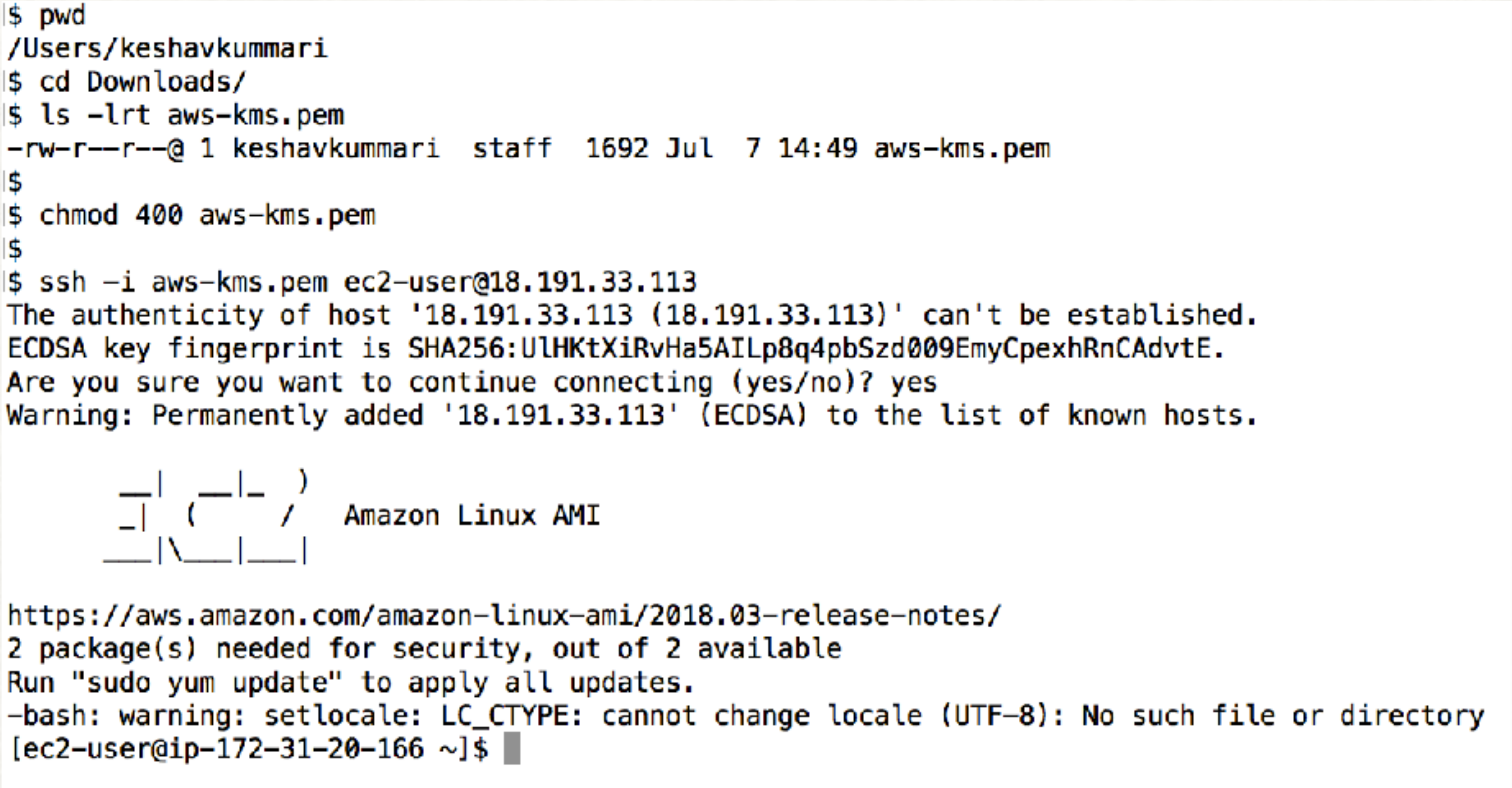
Resource ID length mana
Console experiments

Additional Informa

Getting Started Guide
Documentation
All EC2 Resources
Forums

AWS KMS Example

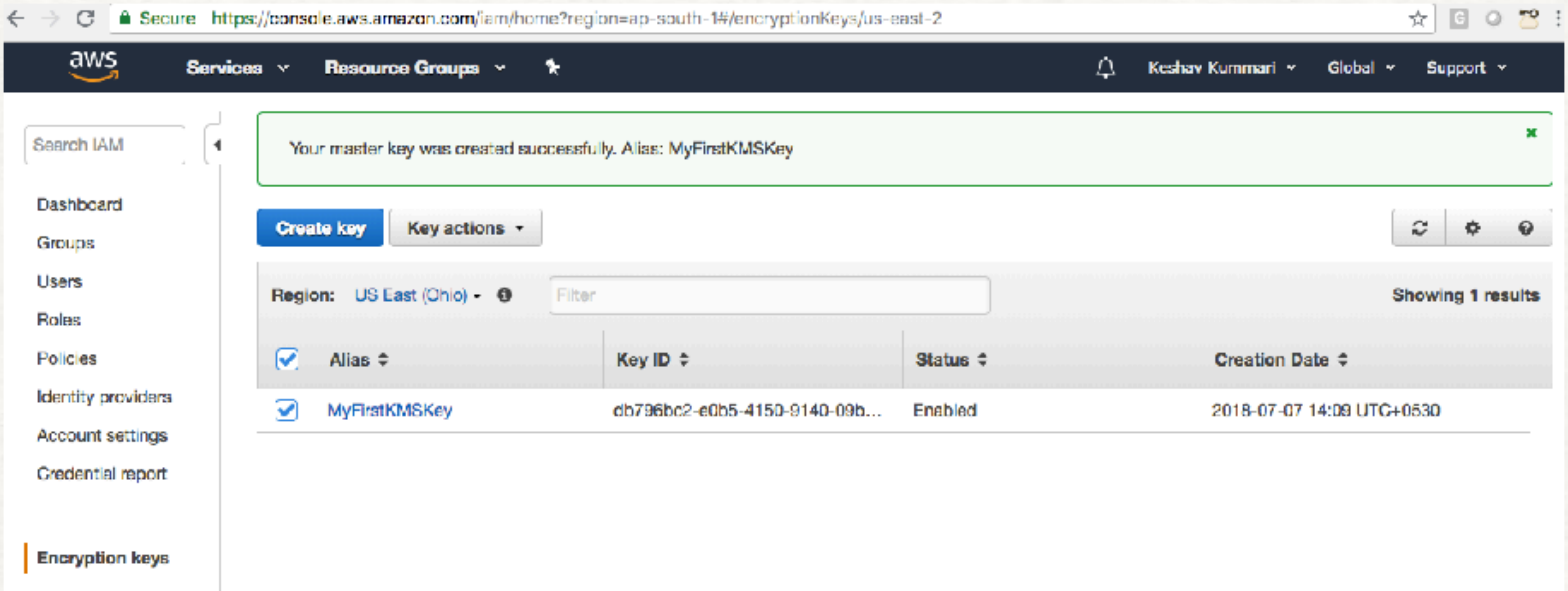
CONNECT EC2 INSTANCE FROM LAPTOP



```
$ pwd
/Users/keshavkummari
$ cd Downloads/
$ ls -lrt aws-kms.pem
-rw-r--r--@ 1 keshavkummari  staff  1692 Jul  7 14:49 aws-kms.pem
$
$ chmod 400 aws-kms.pem
$
$ ssh -i aws-kms.pem ec2-user@18.191.33.113
The authenticity of host '18.191.33.113 (18.191.33.113)' can't be established.
ECDSA key fingerprint is SHA256:UlHKtXiRvHa5AILp8q4pbSzd009EmyCpexhRnCAdvtE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '18.191.33.113' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-20-166 ~]$ 
```

SSH

KMS KEY DETAILS



Your master key was created successfully. Alias: MyFirstKMSKey

Region: US East (Ohio)

Showing 1 results

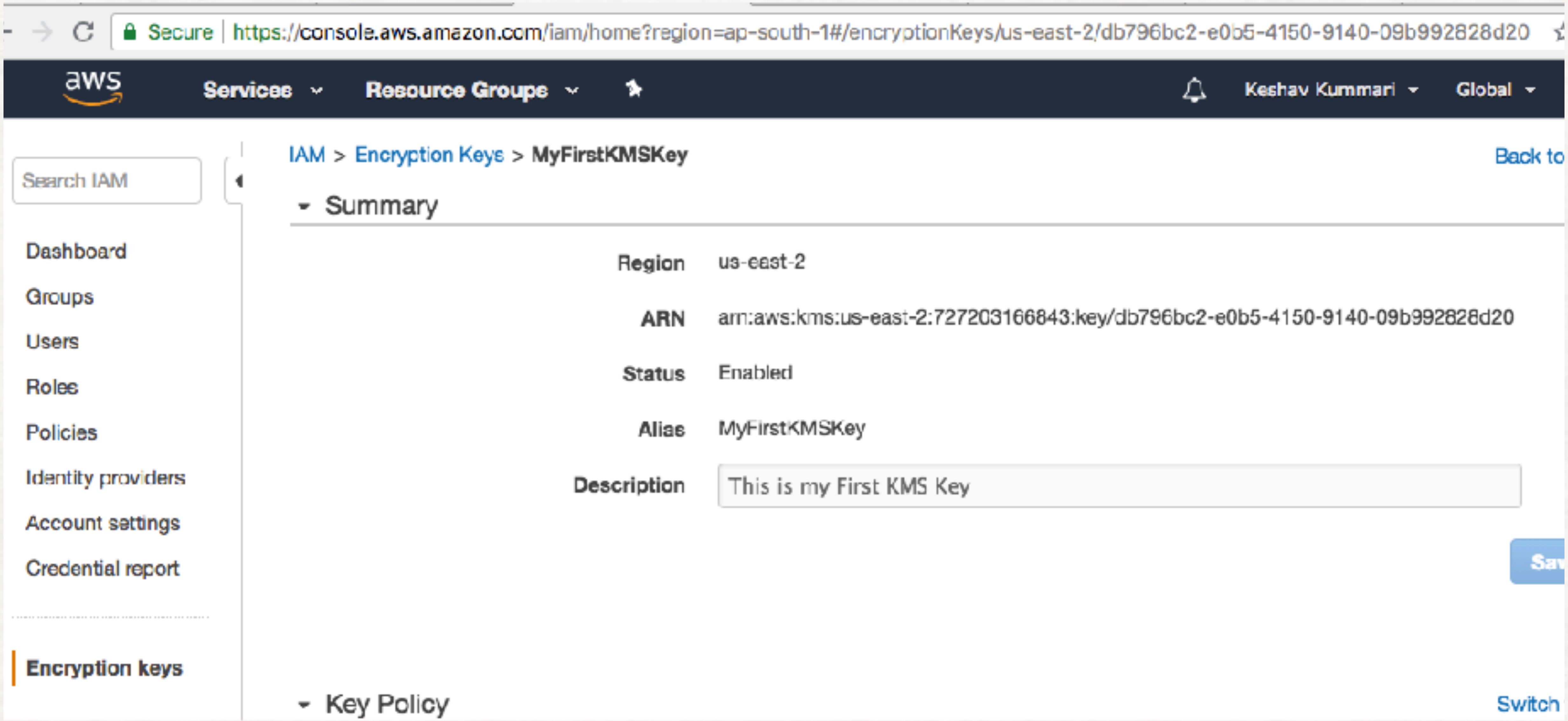
☑	Alias	Key ID	Status	Creation Date
☑	MyFirstKMSKey	db796bc2-e0b5-4150-9140-09b...	Enabled	2018-07-07 14:09 UTC+0530

KMS

KMS COMMANDS

```
# aws kms encrypt --key-id YOURKEYIDHERE --plaintext fileb://
secret.txt --output text --query CiphertextBlob | base64 --decode >
encryptedsecret.txt

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

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

# aws kms enable-key-rotation --key-id YOURKEYIDHERE
```

GO TO >> AWS MGMT CONSOLE >> IAM >> KMS KEY



Secure | https://console.aws.amazon.com/iam/home?region=ap-south-1#/encryptionKeys/us-east-2/db796bc2-e0b5-4150-9140-09b992828d20

IAM > Encryption Keys > MyFirstKMSKey

Summary

Region us-east-2

ARN arn:aws:kms:us-east-2:727203166843:key/db796bc2-e0b5-4150-9140-09b992828d20

Status Enabled

Alias MyFirstKMSKey

Description This is my First KMS Key

Key Policy

Copy the KMS Key

KMS COMMANDS

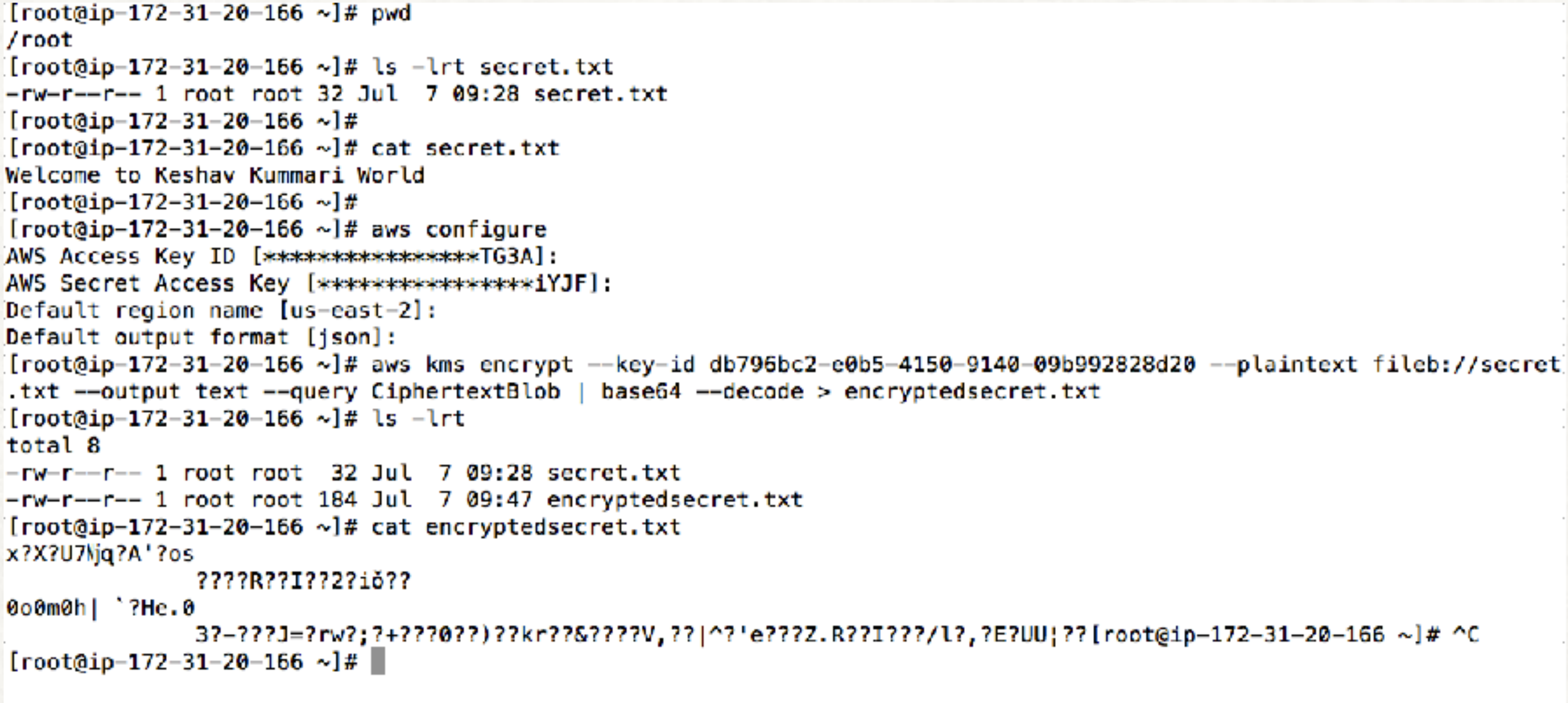
```
# aws kms encrypt --key-id db796bc2-e0b5-4150-9140-09b992828d20 --plaintext fileb://secret.txt --output text --query CiphertextBlob | base64 --decode > encryptedsecret.txt
```

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

```
# aws kms re-encrypt --destination-key-id db796bc2-e0b5-4150-9140-09b992828d20 --ciphertext-blob fileb://encryptedsecret.txt | base64 > newencryption.txt
```

```
# aws kms enable-key-rotation --key-id db796bc2-e0b5-4150-9140-09b992828d20
```

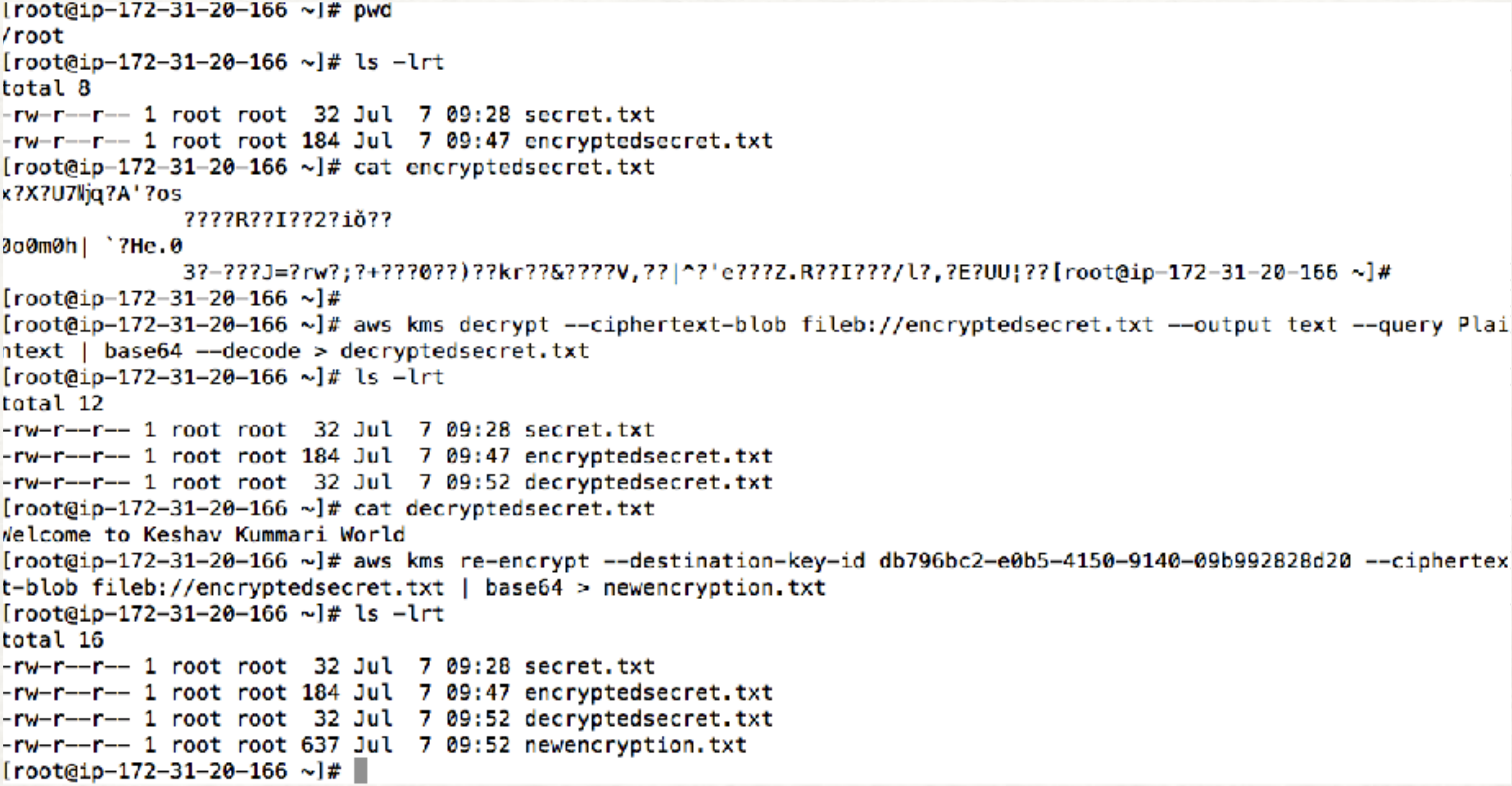
AWS - ENCRYPT THE FILE USING KMS KEYS



```
[root@ip-172-31-20-166 ~]# pwd
/root
[root@ip-172-31-20-166 ~]# ls -lrt secret.txt
-rw-r--r-- 1 root root 32 Jul  7 09:28 secret.txt
[root@ip-172-31-20-166 ~]#
[root@ip-172-31-20-166 ~]# cat secret.txt
Welcome to Keshav Kummari World
[root@ip-172-31-20-166 ~]#
[root@ip-172-31-20-166 ~]# aws configure
AWS Access Key ID [****************TG3A]:
AWS Secret Access Key [****************iYJF]:
Default region name [us-east-2]:
Default output format [json]:
[root@ip-172-31-20-166 ~]# aws kms encrypt --key-id db796bc2-e0b5-4150-9140-09b992828d20 --plaintext fileb://secret
.txt --output text --query CiphertextBlob | base64 --decode > encryptedsecret.txt
[root@ip-172-31-20-166 ~]# ls -lrt
total 8
-rw-r--r-- 1 root root  32 Jul  7 09:28 secret.txt
-rw-r--r-- 1 root root 184 Jul  7 09:47 encryptedsecret.txt
[root@ip-172-31-20-166 ~]# cat encryptedsecret.txt
x?X?U7Njq?A'?os
                ????R??I??2?iŏ??
0o0m0h| `?Hc.0
                3?-???J=?rw?;?+???0??)??kr??&????V,??|^?'e???Z.R??I???/l?,?E?UU¦??[root@ip-172-31-20-166 ~]# ^C
[root@ip-172-31-20-166 ~]# █
```

File has been encrypted

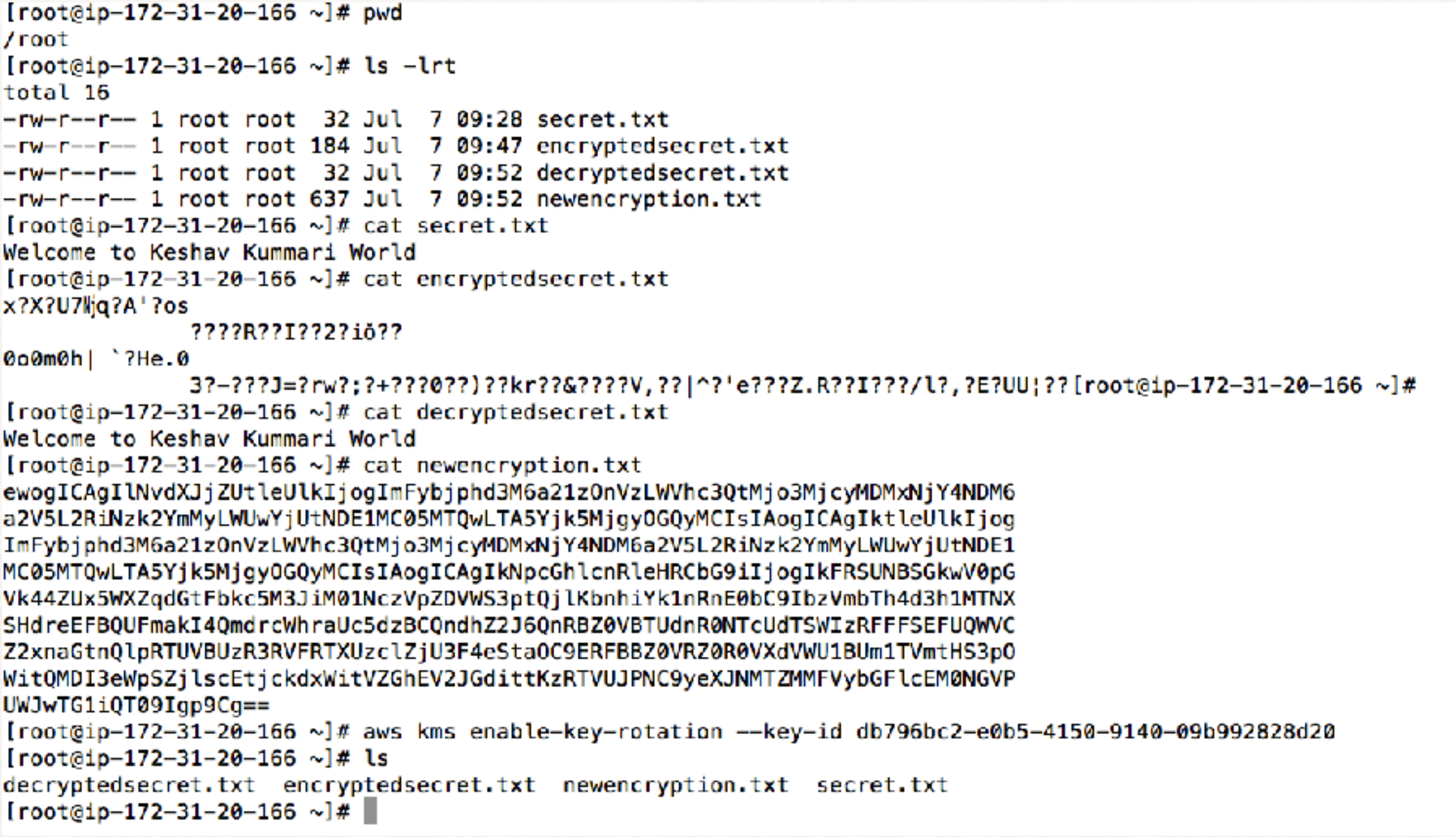
ENCRYPT OR DECRYPT USING KMS KEYS



```
[root@ip-172-31-20-166 ~]# pwd
/root
[root@ip-172-31-20-166 ~]# ls -lrt
total 8
-rw-r--r-- 1 root root  32 Jul  7 09:28 secret.txt
-rw-r--r-- 1 root root 184 Jul  7 09:47 encryptedsecret.txt
[root@ip-172-31-20-166 ~]# cat encryptedsecret.txt
x?X?U7Wjq?A'?os
              ????R??I??2?iŏ??
0o0m0h| `?Hc.0
              3?-???J=?rw?;?+???0??)??kr??&????V,??|^?'c???Z.R??I???/l?,?E?UU¦??[root@ip-172-31-20-166 ~]#
[root@ip-172-31-20-166 ~]#
[root@ip-172-31-20-166 ~]# aws kms decrypt --ciphertext-blob fileb://encryptedsecret.txt --output text --query Plai
ntext | base64 --decode > decryptedsecret.txt
[root@ip-172-31-20-166 ~]# ls -lrt
total 12
-rw-r--r-- 1 root root  32 Jul  7 09:28 secret.txt
-rw-r--r-- 1 root root 184 Jul  7 09:47 encryptedsecret.txt
-rw-r--r-- 1 root root  32 Jul  7 09:52 decryptedsecret.txt
[root@ip-172-31-20-166 ~]# cat decryptedsecret.txt
Welcome to Keshav Kummari World
[root@ip-172-31-20-166 ~]# aws kms re-encrypt --destination-key-id db796bc2-e0b5-4150-9140-09b992828d20 --ciphertex
t-blob fileb://encryptedsecret.txt | base64 > newencryption.txt
[root@ip-172-31-20-166 ~]# ls -lrt
total 16
-rw-r--r-- 1 root root  32 Jul  7 09:28 secret.txt
-rw-r--r-- 1 root root 184 Jul  7 09:47 encryptedsecret.txt
-rw-r--r-- 1 root root  32 Jul  7 09:52 decryptedsecret.txt
-rw-r--r-- 1 root root 637 Jul  7 09:52 newencryption.txt
[root@ip-172-31-20-166 ~]# █
```

KMS

AWS - KMS SUMMARY



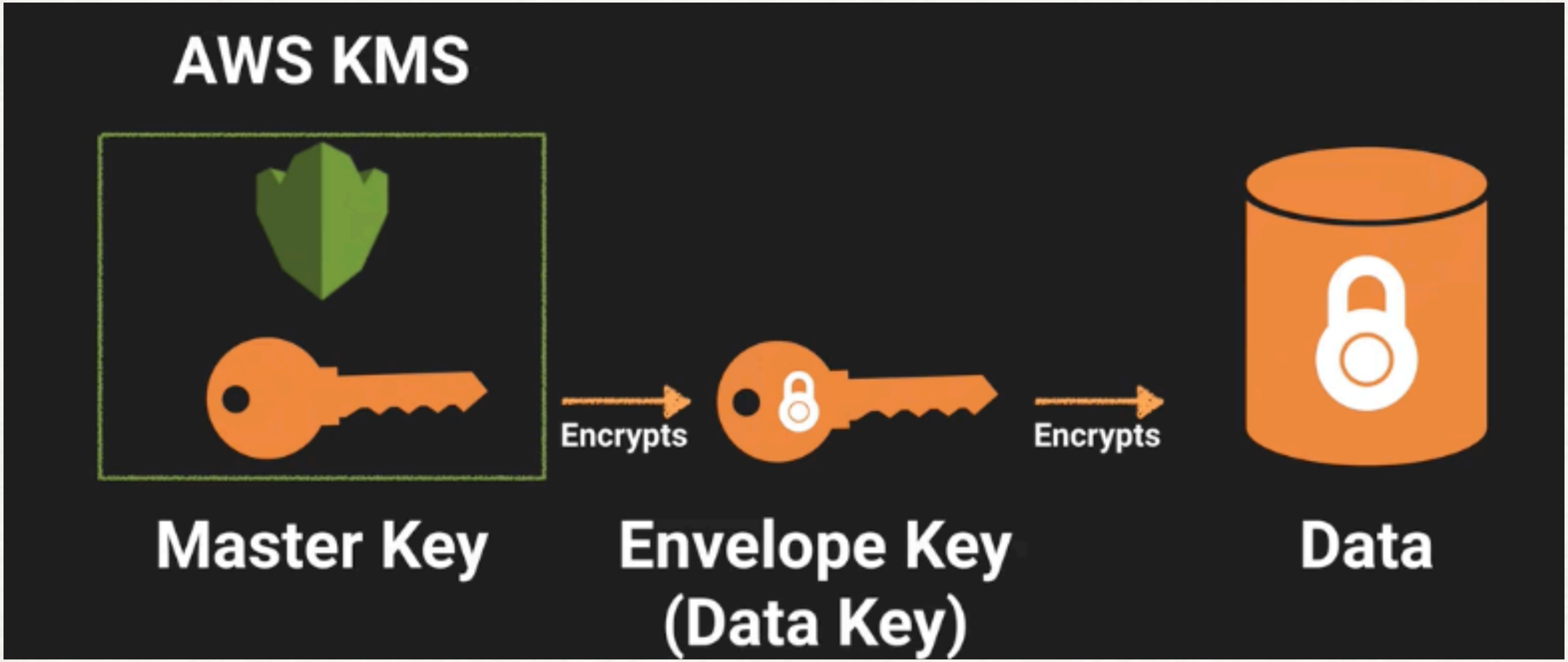
```
[root@ip-172-31-20-166 ~]# pwd
/root
[root@ip-172-31-20-166 ~]# ls -lrt
total 16
-rw-r--r-- 1 root root  32 Jul  7 09:28 secret.txt
-rw-r--r-- 1 root root 184 Jul  7 09:47 encryptedsecret.txt
-rw-r--r-- 1 root root  32 Jul  7 09:52 decryptedsecret.txt
-rw-r--r-- 1 root root 637 Jul  7 09:52 newencryption.txt
[root@ip-172-31-20-166 ~]# cat secret.txt
Welcome to Keshav Kummari World
[root@ip-172-31-20-166 ~]# cat encryptedsecret.txt
x?X?U7Wjq?A'?os
              ????R??I??2?iö??
0o0m0h| `?He.0
              3?-???J=?rw?;?+???0??)??kr??&????V,??|^?'e???Z.R??I???/l?,?E?UU¦??[root@ip-172-31-20-166 ~]#
[root@ip-172-31-20-166 ~]# cat decryptedsecret.txt
Welcome to Keshav Kummari World
[root@ip-172-31-20-166 ~]# cat newencryption.txt
ewogICAgIlNvdXJjZUtleUlkIjogImFybjphd3M6a21zOnVzLWVhc3QtMjo3MjcyMDMxNjY4NDM6
a2V5L2RiNzk2YmMyLWUwYjUtNDE1MC05MTQwLTA5Yjk5MjgyOGQyMCIsIAogICAgIktleUlkIjog
ImFybjphd3M6a21zOnVzLWVhc3QtMjo3MjcyMDMxNjY4NDM6a2V5L2RiNzk2YmMyLWUwYjUtNDE1
MC05MTQwLTA5Yjk5MjgyOGQyMCIsIAogICAgIkNpcGhlcnRleHRCbG9iIjogIkFRSUNBSGkwV0pG
Vk44ZUx5WXZqdGtFbkc5M3JiM01NczVpZDVWS3ptQjlKbnhiYk1nRnE0bC9IbzVmbTh4d3h1MTNX
SHdreEFBQUFmakI4QmdrcWhraUc5dzBCQndhZ2J6QnRBZ0VBTUdnR0NTcUdTSWIzRFFFSEFUQWVC
Z2xnaGtnQlpRTUVBUzR3RVFRTXUzclZjU3F4eStaOC9ERFBBZ0VRZ0R0VXdVWU1BUm1TVmtHS3pO
WitQMDI3eWpSZjlscEtjckdxWitVZGhEV2JGdittKzRTVUJPNC9yeXJNMTZMMFVybGFlcEM0NGVP
UWJwTG1iQT09Igp9Cg==
[root@ip-172-31-20-166 ~]# aws kms enable-key-rotation --key-id db796bc2-e0b5-4150-9140-09b992828d20
[root@ip-172-31-20-166 ~]# ls
decryptedsecret.txt  encryptedsecret.txt  newencryption.txt  secret.txt
[root@ip-172-31-20-166 ~]# 
```

KMS COMMANDS

KMS API CALLS

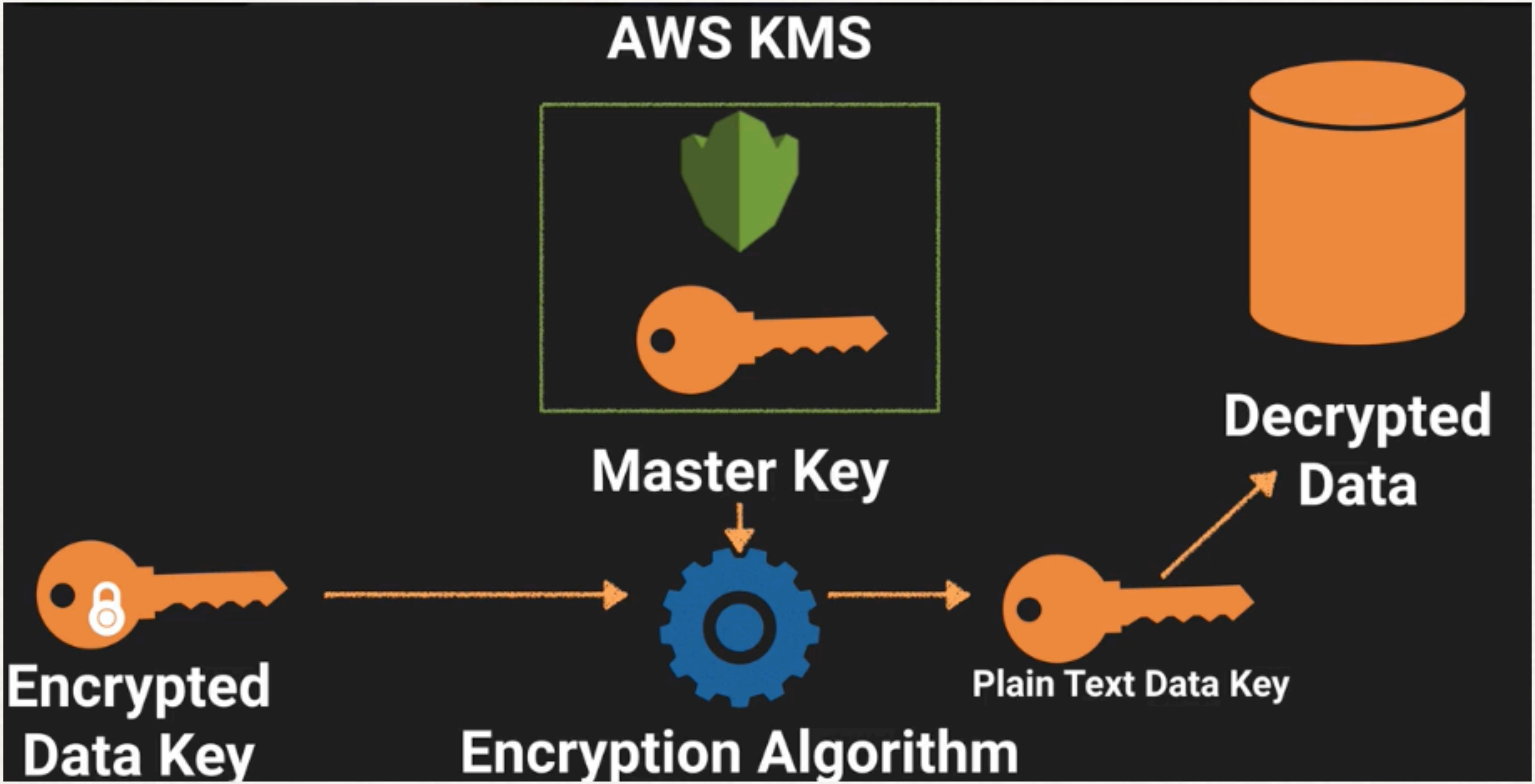
```
# aws kms encrypt

# aws kms decrypt

# aws kms re-encrypt

# aws kms enable-key-rotation
```

ENVELOPE ENCRYPTION



KMS - ENVELOPE

KMS - ENVELOPE DECRYPTION



KMS Envelope decrypt

KMS SUMMARY

- AWS KMS is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data.
- AWS KMS is integrated with other AWS Services including,
- EBS
- S3
- Redshift
- Elastic Transcoder
- WorkMail
- RDS

And others to make it simple to encrypt your data with encryption keys that you manage.

- **The Customer Master Key:**
- CMK

alias

creation date

desciption

key state

key material(Either Customer provided or AWS Provided)

- Can Never be exported
- **Setup a Customer Master Key:**
- Create Alias and Description
- Choose material option
- Define Key Administrative permissions

IAM Users/Roles that can administer(but not use) the key through the KMS API

- Define Key Usage Permissions

IAM users/roles that can use the key to encrypt and decrypt data

Thank you!