WHIZLABS

Home / AWS / Guided Lab / Encryption and Decryption Using KMS

Encryption and Decryption Using KMS

Level: Intermediate

Amazon EC2 AWS Key Management Service Amazon Web Services IAM

0h 26m 19s left

15 MIN

End Lab

Open Console

Validation

Lab Credentials

User Name

Whiz_User_80425.59548925

Password

1f29ab18-5134-442c-9736-ff6f49a80f40

Access Key

AKIAY6U3B7RFXCSTNCVW

Secret Key

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

Lab Resources

No Lab Resources Found

Support Documents

1. FAQs and Troubleshooting

Need help?

How to use Hands on Lab

Troubleshooting Lab

FAQs

Submit Feedback | Share

Lab Overview | Lab Steps | Lab Validation

Cloud Administrator

Security, Compute

Lab Steps

Task 1: Sign in to AWS Management Console

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
    - Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
    - Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

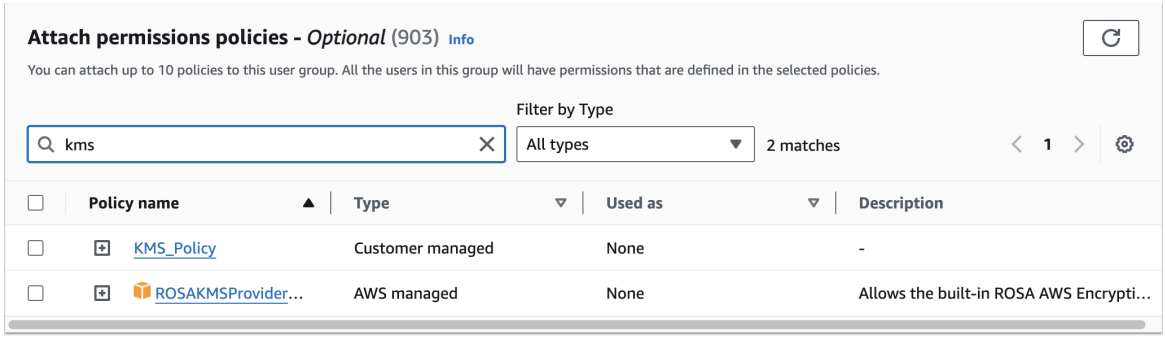
Task 2: Create a User group for KMS users and attach a Policy to the Group

1. Make sure to choose **N.Virginia** region in the AWS Management console dashboard, which is present in the top right corner.
2. Navigate to the **Services** menu at the top, click on **IAM** in the **Security, Idenitity, & Compliance** section.

3. In the IAM section, click on **User groups**.

4. Click on **Create group**

    - User group name : Enter **KMSGroup**
    - **Attach permissions policies:** For the Policy name type **KMS** and select **KMS_Policy**



5. Now, Click on **Create Group** button.

6. We have successfully created a new group for our KMS lab.

Task 3: Create 2 Users for managing the KMS

In this task, We are going to add 2 users to the group we created.

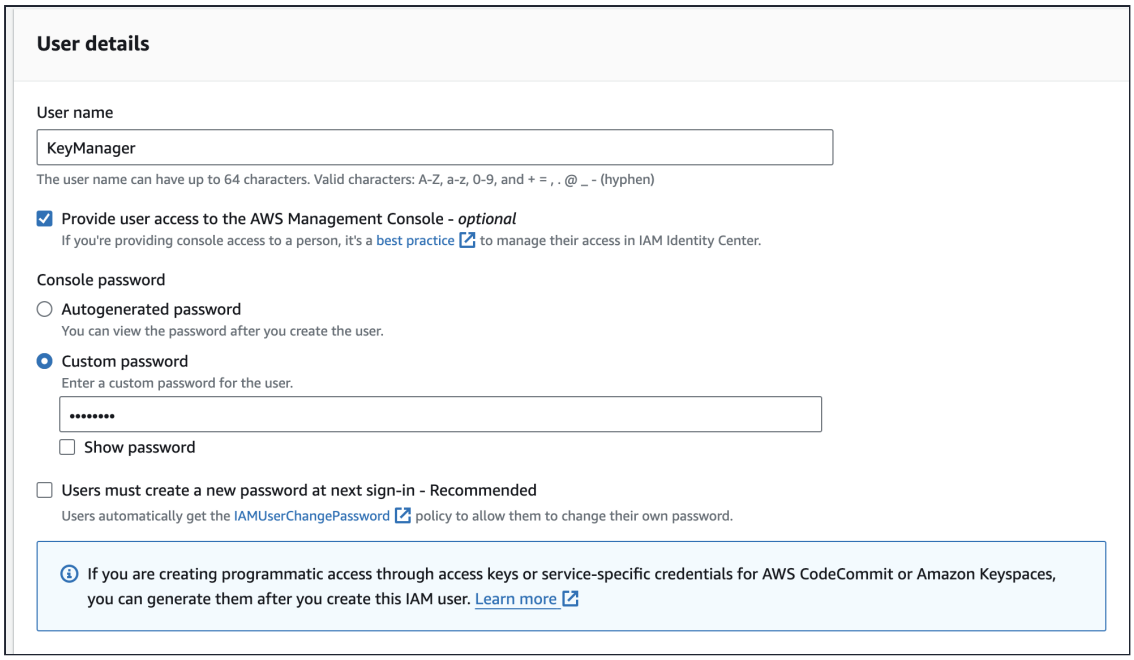
1. Click on **Users** on the left side of the IAM dashboard. Click on **Create User.**

2. Enter User name as **KeyManager**.

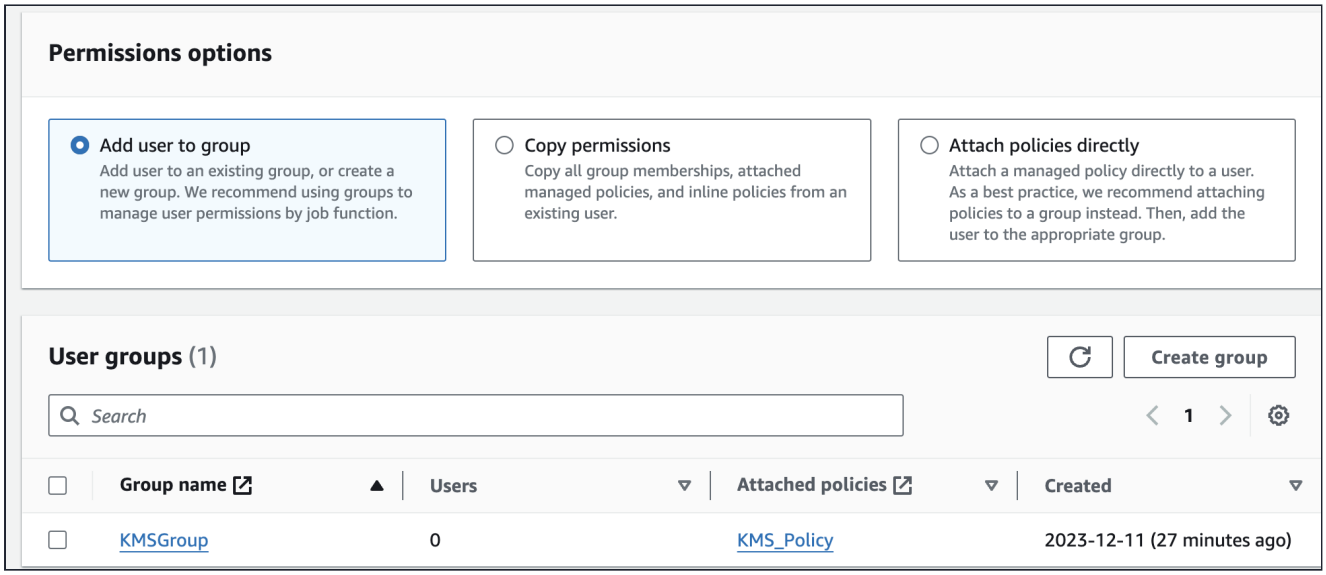
3. Check **Provide user access to the AWS Management Console** checkbox

4. Click on the Custom password and give the password you like to give it to the user.

5. Uncheck the **Users must create a new password at next sign-in**. Click on **Next** to provide permission to our user.

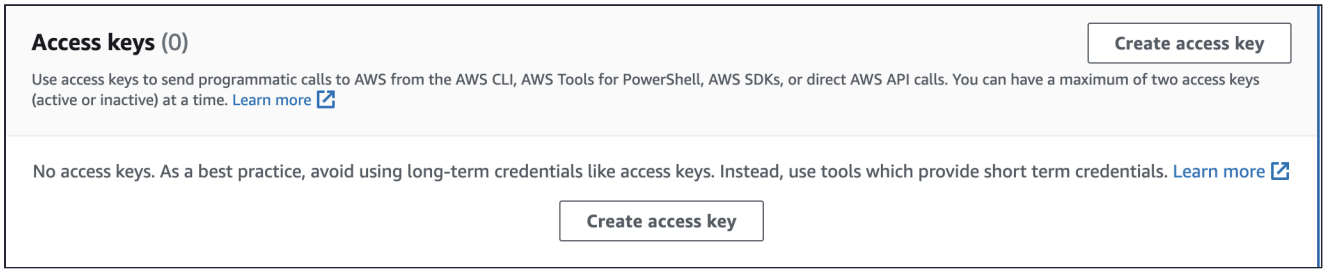


6. For permission, select **Add User to group** and select the **KMSGroup** which we created and click on Next button.
7. In the review section, if all the settings are as per the requirement, click on **create user**.
8. We have successfully created our **KeyManager.** Now similarly we're going to create a new user and this will be the person who does the decryption.

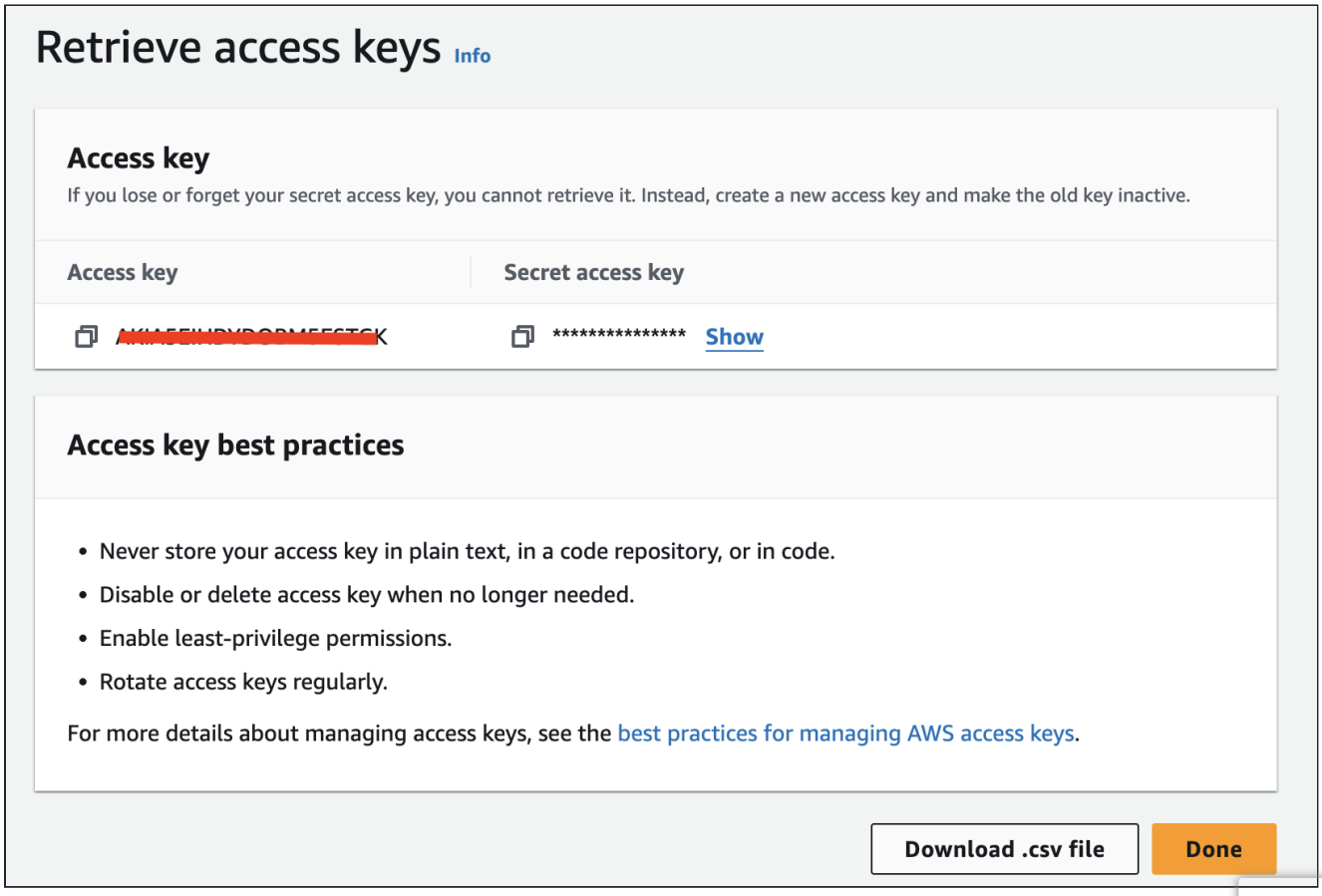


9. Click on **Create user.**
10. Enter User name as **KeyEncryption**.
11. Check **Provide user access to the AWS Management Console** checkbox
12. Click on the Custom password and give the password you like to give it to the user.
13. Uncheck the **Users must create a new password at next sign-in**. Click on **Next** to provide permission to our user.

14. Click on **Next:Permissions** to add permissions to the user.
15. For permission, select **Add User to group** and select the **KMSGroup** which we created and click on Next button.
16. In the review section, if all the settings are as per the requirement, click on **create user**.
17. Now, to get the **access** and **secret key**, click on **KeyEncryption** user and go to **Security credentials** tab.
18. Scroll down and click on **Create access key** button.

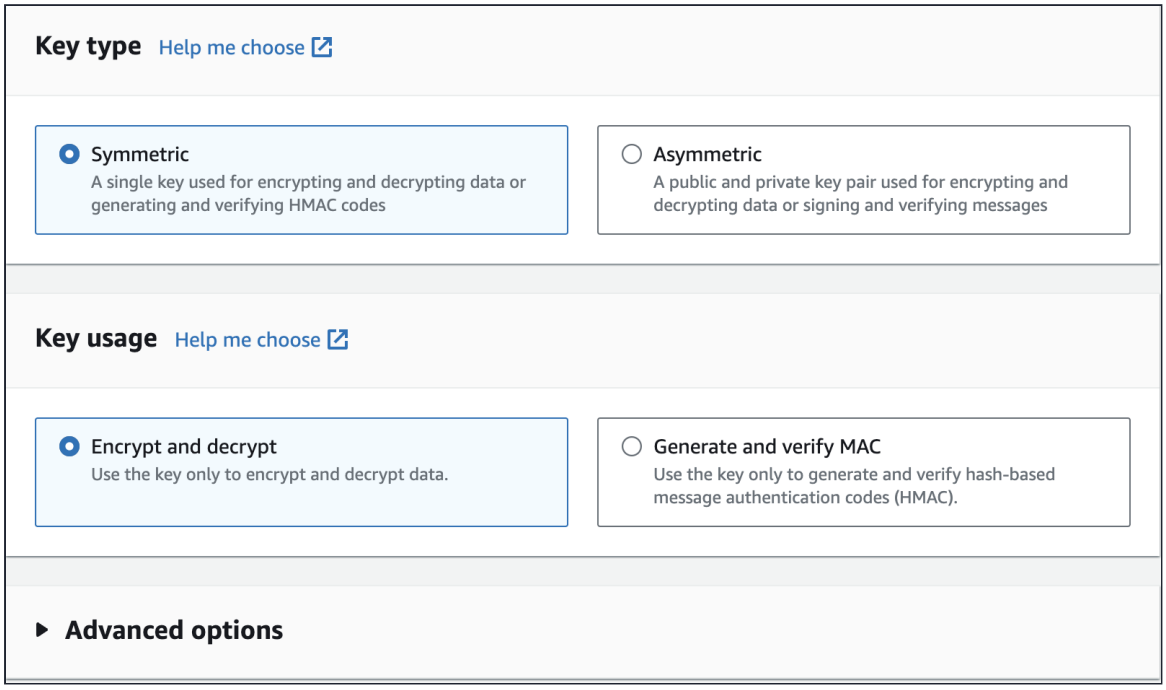


19. Select Use case as **Command Line Interface (CLI)**, check the **confirmation** box and click on **Next** button.
20. Click on create access key and **don't forget to download the secret access key of the user as it will be required to connect with our EC2 instance for encryption.**

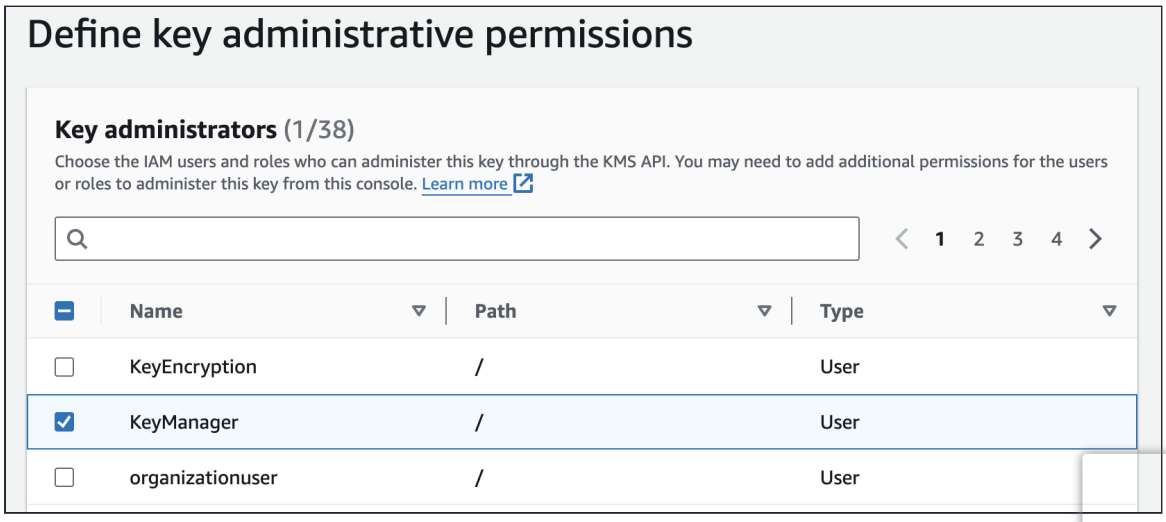


Task 4 : Creating a KMS Key

1. Navigate to the **Services** menu at the top, click on **KMS** in the **Security, Idenitity, & Compliance** section
2. Click on the **Create a key** button.
3. Basically we have two types of key management, **Symmetric** and **Asymmetric**. In this lab, we are going to use **Symmetric,** as we are going to use a single key for both encrypt and decrypt operations. Choose the key type as **Symmetric** for key material and click on next.



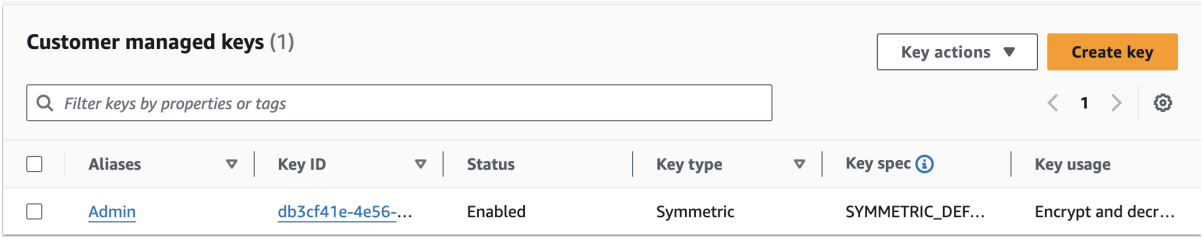
4. Enter Alias as **Admin** and click on **Next** button.
5. To **Define key administrative permissions** In this step, we need to specify the user who'll be managing the keys or an administrator for managing the keys. In our lab we have already created a user to manage the key i.e **KeyManager.** We are going to assign a key manager for administrative task. Click on **Next.**



6. To **Define key usage permissions.** In this step, we need to define the user who'll be responsible for encryption and decryption of the files. Select **KeyEncryption** and click on **Next**

7. Once you click on Next you'll be moved to the review section. Review the key policy that we have created and if everything is fine, just click on **Finish**.

8. We have successfully created the KMS key.



9. Now that we have created the KMS and User policies, move to **service** section and choose **EC2** under Compute section.

Task 5 : Launching an EC2 Instance

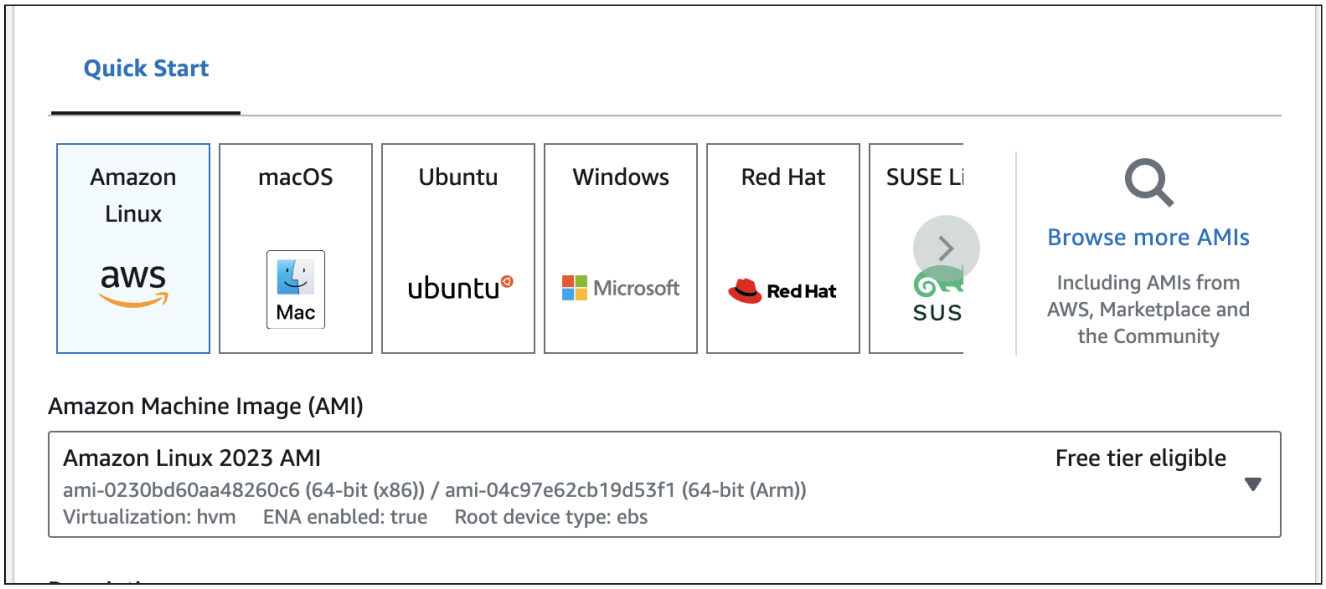
1. Make sure you are in **N.Virginia** Region.

2. Navigate to the **Services** menu at the top, click on **EC2** in the **Compute** section.

3. Click on **Launch Instance**

4. Name : Enter **MyEC2Server**

5. For AMI Select **Amazon Linux** in the quickstart menu.

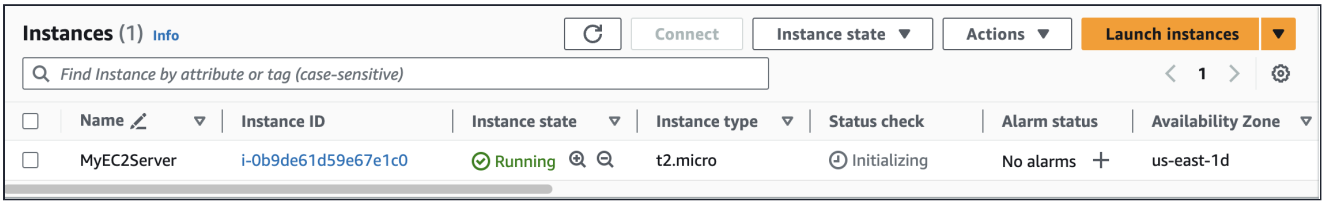


6. For **Instance Type**: Select **t2.micro**

7. For **Key pair(login)**: Select **Create a new key pair** Button

- Key pair name: **WhizKey**
- Key pair type: **RSA**
- Private key file format: **.pem**

8. Keep all the settings as default and click on **Launch instance**
9. **Launch Status:** Your instance is now launching, Click on the instance ID and wait for complete initialization of instance till status change to **Running**

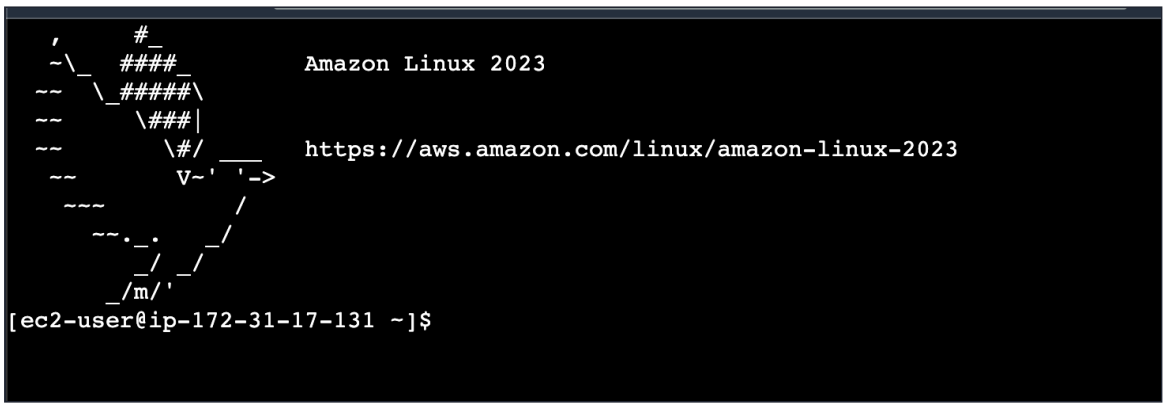


Task 6: SSH into the EC2 Instance

- Please follow the steps in **SSH into EC2 Instance.**

Task 7 : Perform KMS Encryption and Decryption

1. Once you click on connect you get a terminal which is our EC2-user login on EC2-instance.



2. No we need to create a file with the name **secret.txt** , Execute the command

```
echo “Welcome to Whizlab” > secret.txt
```

3. Now that we have created a file **secret.txt** we need to execute

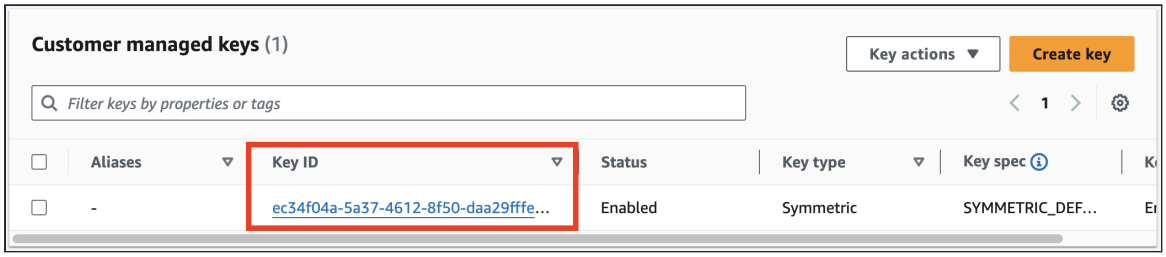
```
aws configure
```

4. Enter the access key and secret access of the user **KeyEncryption** and the default region is **us-east-1**.

```
        _/m/'
[ec2-user@ip-172-31-17-131 ~]$ echo “Welcome to Whizlab” > secret.txt
[ec2-user@ip-172-31-17-131 ~]$ aws configure

AWS Access Key ID [None]: AKIAWXBSZTITYQR7JA3D
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
[ec2-user@ip-172-31-17-131 ~]$
```

5. Once aws configure is complete, we need to execute the command for encryption. But before that we require key id for encryption and decryption. Navigate to KMS and Copy the **key id**.



6. Make the changes in all commands with the KMS key id.

```
aws kms encrypt --key-id a8188009-1ac3-4201-ab1d-63c6e2914ce9 --
plaintext fileb://secret.txt --output text --query CiphertextBlob |
base64 --decode > encryptedsecret.txt
```

7. We have successfully encrypted our text file . To view the statement execute

```
cat encryptedsecret.txt
```

```
[ec2-user@ip-172-31-17-131 ~]$ cat encryptedsecret.txt

0h0f0ax=`?He.0??{?D?o?J?p?u??t??? ??@6?w0u      *?H??
              ???!?Uu??C?o?4?c?B>?Z?ε??k%P?w?3PB?D
_[ec2-user@ip-172-31-17-131 ~]$                  ???ix>4?n??7
[ec2-user@ip-172-31-17-131 ~]$
```

8. We are going to decrypt the encrypted file to view the data.

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

9. We have successfully encrypted our text file . To view the statement execute

```
cat decryptedsecret.txt
```

```
[ec2-user@ip-172-31-17-131 ~]$ cat decryptedsecret.txt

"Welcome to Whizlab"
[ec2-user@ip-172-31-17-131 ~]$
```

10. Now we need to re-encrypt the existing file so execute the command.

```
aws kms re-encrypt --destination-key-id a8188009-1ac3-4201-ab1d-63c6e2914ce9 --ciphertext-blob fileb://encryptedsecret.txt | base64 > newencryption.txt
```

11. You can check the created files by using command

```
ls -lrt
```

```
[ec2-user@ip-172-31-17-131 ~]$ aws kms re-encrypt --destination-key-id ec34f04a-5a37-4612-8f50-daa29fffe579 --ciphertext-blob fileb://encryptedsecret.txt |
base64 > newencryption.txt
[ec2-user@ip-172-31-17-131 ~]$ ls -lrt
total 16
-rw-r--r--. 1 ec2-user ec2-user  25 Dec 11 11:16 secret.txt
-rw-r--r--. 1 ec2-user ec2-user 177 Dec 11 11:20 encryptedsecret.txt
-rw-r--r--. 1 ec2-user ec2-user  25 Dec 11 11:22 decryptedsecret.txt
-rw-r--r--. 1 ec2-user ec2-user 770 Dec 11 11:24 newencryption.txt
[ec2-user@ip-172-31-17-131 ~]$
```

12. We have successfully encrypted our text file . To view the statement execute

```
cat newencryption.txt
```

```
[ec2-user@ip-172-31-17-131 ~]$
[ec2-user@ip-172-31-17-131 ~]$ cat newencryption.txt

ewogICAgIkNpcGhlcnRleHRCbG9iIjogIkFRSUNBSGp2R3hkS0E4SGZqbFAvZUQzSHBsQTNMSjky
cE45N3EwU1NiOTlLNm5EUWRRRkYxTlBsREd5OE8wNG9Rd3puaWJQbEFBQUFkekIxQmdrcWhraUc5
dzBCQndhZ2FEQm1BZ0VBTUdFR0NTcUdTSWIzRFFFSEFUQWVCZ2xnaGtnQlpRTUVBUzR3RVFRTUtl
RXVDV1lBNEpTL1NSQURBZ0VRZ0RTVEN2L1I5Mi9QVHZ6NXNXbW1sMHVDanJWWGp5Skhyelo3aDZq
SW9pSlJ3ZGwyOHVndGwzL2FLbnVhanZEZkVidEtGM1ZZIiwKICAgICJTb3VyY2VLZXlJZCI6ICJh
cm46YXdzOmttczp1cy1lYXN0LTE6OTAyNDk0OTI5MTE2OmtleS9lYzM0ZjA0YS01YTM3LTQ2MTIt
OGY1MC1kYWEyOWZmZmU1NzkiLAogICAgIktleUlkIjogImFybjphd3M6a21zOnVzLWVhc3QtMTo5
MDI0OTQ5MjkxMTY6a2V5L2VjMzRmMDRhLTVhMzctNDYxMi04ZjUwLWRhYTI5ZmZmZTU3OSIsCiAg
ICAiU291cmNlRW5jcnlwdGlvbkFsZ29yaXRobSI6ICJTWU1NRVRSSUNfREVGQVVMVCIsCiAgICAi
RGVzdGluYXRpb25FbmNyeXB0aW9uQWxnb3JpdGhtIjogIlNZTU1FVFJJQ19ERUZBVUxUIgp9Cg==
[ec2-user@ip-172-31-17-131 ~]$
```

13. We have successfully executed the re-encrypt statement.

14. We need to enable the key rotation of KMS so that they can be periodically changed or in response to a potential leak or compromise.

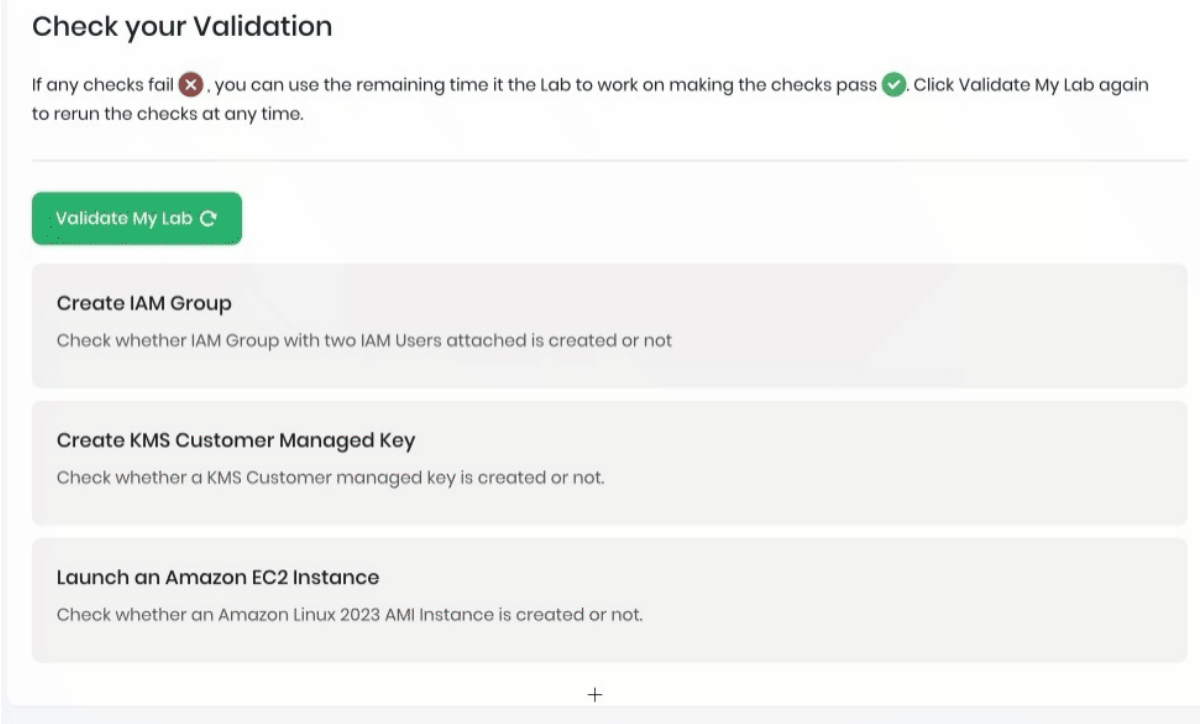
```
aws kms enable-key-rotation --key-id a8188009-1ac3-4201-ab1d-63c6e2914ce9
```

Do You Know?

KMS enforces access control policies to ensure that only authorized individuals or systems can use or manage cryptographic keys. This helps prevent unauthorized access to sensitive information.

Task 8 : Validation of the Lab

1. Once the lab steps are completed, please click on the **Validation** button on the left side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :



Completion and Conclusion

1. You have successfully created a group for KMS users and attached a policy to the group.
2. You have successfully created 2 users for managing the KMS.
3. You have successfully created a KMS Key.
4. You have successfully launched an EC2 Instance and connected to SSH using browser.
5. You have successfully configured KMS.

6. You have got familiar with Encryption, decryption, re-encryption and key rotation of KMS by executing the commands.

End Lab

1. Sign out of AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps, click on **End Lab** from your Whizlabs lab console and wait till the process gets completed.