

[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



End Lab

Open Console

Validation

Lab Credentials

User Name ⓘ

Whiz_User_80425.59548925



Password ⓘ

1f29ab18-5134-442c-9736-ff6f49a80f40



Access Key ⓘ

AKIAY6U3B7RFXCSTNCVW



Secret Key ⓘ

/BGuTXUD15FJTSPi2Ph0NvFoAD+1apB9d8nF4Nue






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**.

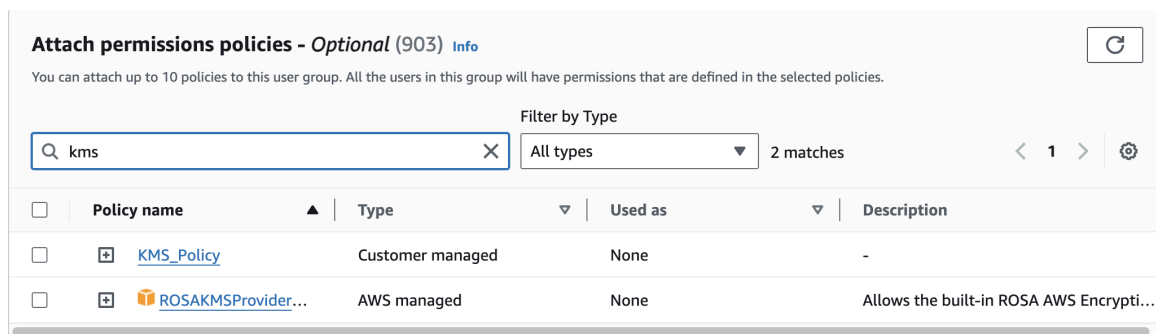
Task 2: Create a User group for KMS users and attach a Policy to the Group

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, Identity, & Compliance** section.

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

4. Click on **Create group**

- User group name : Enter **KMSGGroup**
- **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.

User details

User name

KeyManager

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☒ Provide user access to the AWS Management Console - *optional*
 If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Console password

☐ Autogenerated password
 You can view the password after you create the user.

☒ Custom password
 Enter a custom password for the user.

.....

☐ Show password

☐ Users must create a new password at next sign-in - Recommended
 Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

ⓘ If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

- For permission, select **Add User to group** and select the **KMSGGroup** which we created and click on Next button.
- In the review section, if all the settings are as per the requirement, click on **create user**.
- 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.

Permissions options

☒ Add user to group
 Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions
 Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ Attach policies directly
 Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1)

Q Search

< 1 >

⚙

<input type="checkbox"/>	Group name ↗	Users	Attached policies ↗	Created
<input type="checkbox"/>	KMSGGroup	0	KMS_Policy	2023-12-11 (27 minutes ago)

- Click on **Create user**.
- Enter User name as **KeyEncryption**.
- Check **Provide user access to the AWS Management Console** checkbox
- Click on the Custom password and give the password you like to give it to the user.
- 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 **KMSGGroup** 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.

Access keys (0) Create access key

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)



No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

Create access key

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.**

Retrieve access keys [Info](#)

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
 AKIAEFNDYDORMEESTCK	 ***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Download .csv file

Done

Task 4 : Creating a KMS Key

1. Navigate to the **Services** menu at the top, click on **KMS** in the **Security, Identity, & 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.

Key type [Help me choose](#)

☒ **Symmetric**
A single key used for encrypting and decrypting data or generating and verifying HMAC codes

☐ **Asymmetric**
A public and private key pair used for encrypting and decrypting data or signing and verifying messages

Key usage [Help me choose](#)

☒ **Encrypt and decrypt**
Use the key only to encrypt and decrypt data.

☐ **Generate and verify MAC**
Use the key only to generate and verify hash-based message authentication codes (HMAC).

► Advanced options

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**.

Define key administrative permissions

Key administrators (1/38)
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 2 3 4 >

<input type="checkbox"/>	Name	Path	Type
<input type="checkbox"/>	KeyEncryption	/	User
<input checked="" type="checkbox"/>	KeyManager	/	User
<input type="checkbox"/>	organizationuser	/	User

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.

Customer managed keys (1)						
<input type="text"/> Filter keys by properties or tags					Key actions ▾	Create key
<div> <div>< 1 ></div> <div>⚙️</div> </div>						
<input type="checkbox"/>	Aliases ▾	Key ID ▾	Status	Key type ▾	Key spec ⓘ	Key usage
<input type="checkbox"/>	Admin	db3cf41e-4e56-...	Enabled	Symmetric	SYMMETRIC_DEF...	Encrypt and decr...

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.

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Li

SUS

🔍

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

Free tier eligible ▾

ami-0230bd60aa48260c6 (64-bit (x86)) / ami-04c97e62cb19d53f1 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

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**

Instances (1)
[Info](#)

Connect

Instance state ▾

Actions ▾

Launch instances

<

1

>

<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾
<input type="checkbox"/>	MyEC2Server	i-0b9de61d59e67e1c0	Running	t2.micro	Initializing	No alarms	us-east-1d

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.

```
'      #  
~\    #####_   Amazon Linux 2023  
~~~\   #####_\n~~~~ \   ####|\n~~~~  \###|  
~~~~  \|  
~~~~  \|/_____  
~~~~  v~' '->  
~~~~  
~~~~_.  
~~~~/_/  
~~~~/_m/' -
```

[ec2-user@ip-172-31-17-131 ~]\$

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]: AKIAWXSZTITYQR7JA3D
AWS Secret Access Key [None]: y6O0oyCBzdSixF+smnC865EBqJ6hYWNIG56Kv48a
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**.

Customer managed keys (1)					
<input type="text" value="Filter keys by properties or tags"/> Key actions ▾ Create key					
<div> <div> <input type="checkbox"/> </div> <div>Aliases ▾</div> </div> <div> <div> <input type="checkbox"/> </div> <div>Key ID ▾</div> </div> <div>Status</div> <div>Key type ▾</div> <div>Key spec ⓘ</div> <div>Kr</div>					
-		ec34f04a-5a37-4612-8f50-daa29ffe...	Enabled	Symmetric	SYMMETRIC_DEF...

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=`0He.000{0D0o0J0p0u00t000 00e60w0u *0H00
000!0Uu00C0o040c0B>0Z0E00k%P0w03PB0D
_[ec2-user@ip-172-31-17-131 ~]$ 000ix>40n007
[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 ec3af04a-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  
  
ewogICAgIkNpcGhlcnRleHRCbG9iIjogIkFRSUNBSGP2R3hkSOE4SGZqbFAvZUQzSHBsQTnMSjky  
cE45N3EwUlNiOTlLnm5EUWRRRkYxTlBsRed5OE8wNG9Rd3puaWJQbEFBQUFkeKlXQmdrcWhraUc5  
dzBCQndhZ2FEQm1BZ0VBTudFRONTcUdTswizRFFFSEFUQWVCZ2xnaGtnQlpRTUVBUzR3RVFRtUt1  
RXVDV1lBNepTL1NSQURBZ0VRZ0RTVEN2L1I5Mi9QVHZ6NXNXbw1SMHVDanJWWGp5Skhyelo3aDZq  
SW9pslJ3ZGwyOHVndGwzL2FLbnVhanZEzkVidEtGM1ZZiIiwKICAgICJTB3VyY2VLZXlJZCI6ICJh  
cm46YXdzOmttczplcyllYXN0LTE6OTAyNDk0OTI5MTE2OmtleS9lyzm0Zja0YS01YTM3LTQ2MTIt  
OGY1MC1kYWeyOWZmZmUlNzkiLAogICAgIktleUlkIjogImFybpbhd3M6a2l2OnVzLWVhc3QtMTto5  
MDI0OTQ5MjkxMTY5a2V5L2VjMzRmMDRhLTVhMzc2NDYxMi04ZjUwLWRhYTl5ZmZmZTU3OSIsCiAg  
ICAiU291cmNlRW5jcmlwdGlwbkFsZ29yaXR0bSI6ICJTWUlnNRVSSUNfREVGVQVVMVCisCiAgICAI  
RGVzdGluYXRpb25FbmNyeXB0aW9uQWxnbn3JpdGhtIjogIlNZTU1VFVJJQ19ERUZBVUxUIgp9Cg==  
[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.

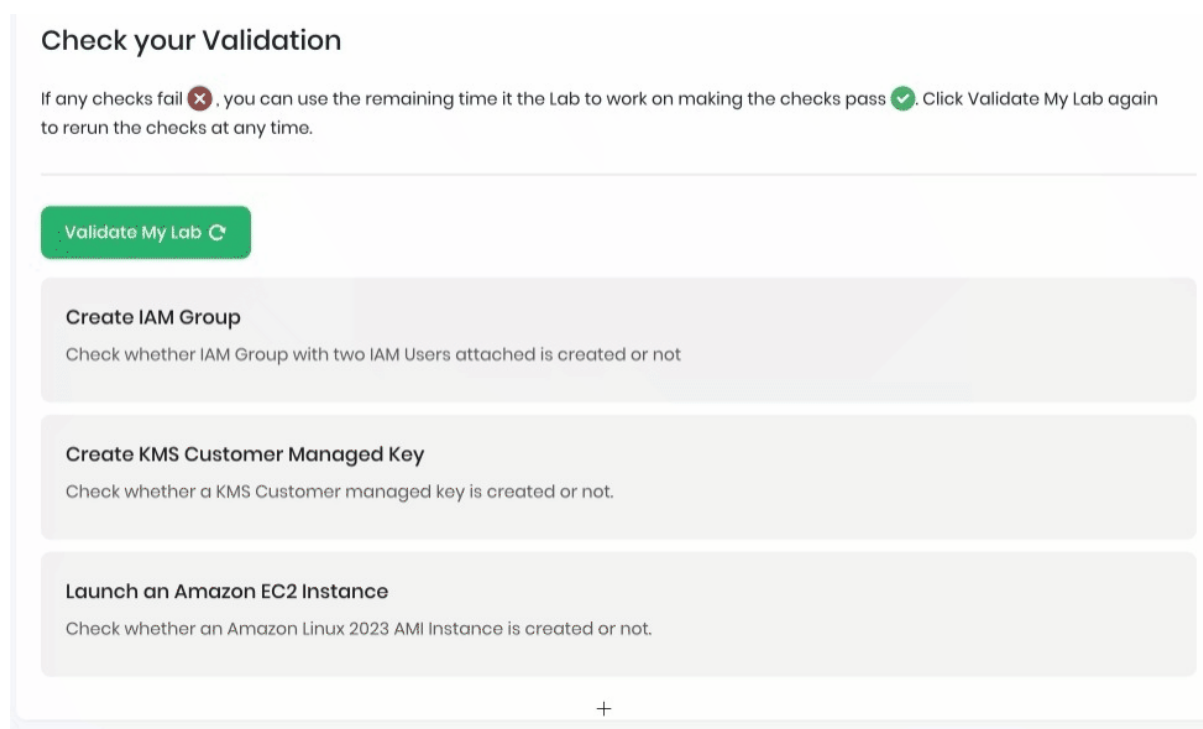
```
aws kms enable-key-rotation --key-id a8188009-1ac3-4201-ab1d-63c6e2914ce9
```


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.

[About Us](#) [Subscription](#) [Instructions and Guidelines](#) [FAQ's](#) [Contact Us](#)



© 2024, Whizlabs Software Pvt. Ltd.

