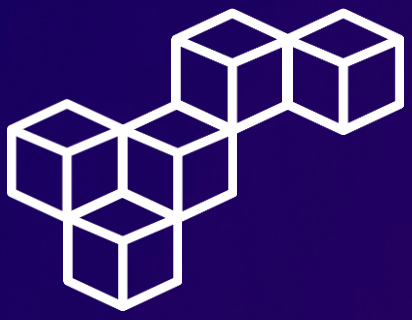
AWS re:Start

LAB

Data Protection Using Encryption

WEEK 4

aws re/start



Overview

Cryptography is the conversion of communicated information into secret code that keeps the information confidential and private. Functions include authentication, data integrity, and nonrepudiation. The central function of cryptography is encryption, which transforms data into an unreadable form.

Encryption ensures privacy by keeping the information hidden from people who the information is not intended for. Decryption, the opposite of encryption, transforms encrypted data back into data; it won't make any sense until it has been properly decrypted.

In this lab, you will connect to a file server that is hosted on an Amazon Elastic Compute Cloud (Amazon EC2) instance. You will configure the AWS Encryption command line interface (CLI) on the instance. You will create an encryption key by using the AWS Key Management Service (AWS KMS). The key will be used to encrypt and decrypt data. Next, you will create multiple text files that are unencrypted by default. You will then use the AWS KMS key to encrypt the files and view them while they are encrypted. You will finish the lab by decrypting the same files and viewing the contents.

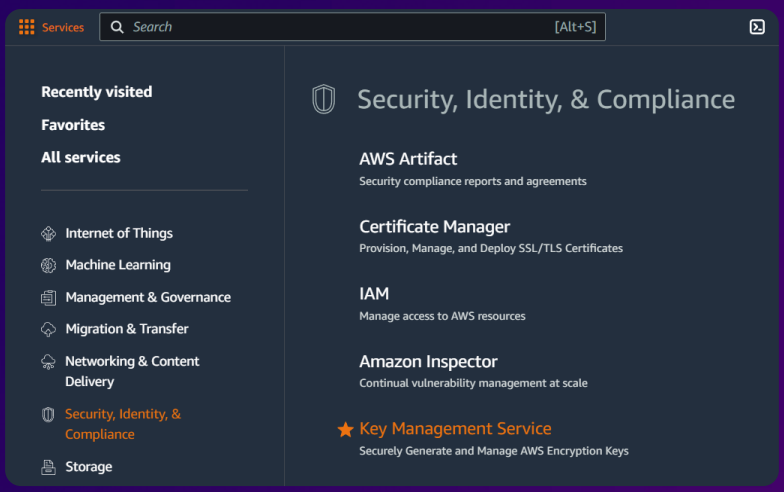
The lab environment has one preconfigured EC2 instance named File Server. An AWS Identity and Access Management (IAM) role is attached, which allows you to connect to the instance by using the AWS Systems Manager Session Manager.

Task 1

Create an AWS KMS key

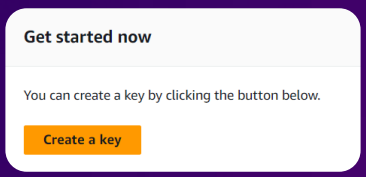
Step 1: Access the Key Management Service

Open the AWS Management Console, and select Key Management Service.



Step 2: Create a key

In the Key Management Service, select Create a key.



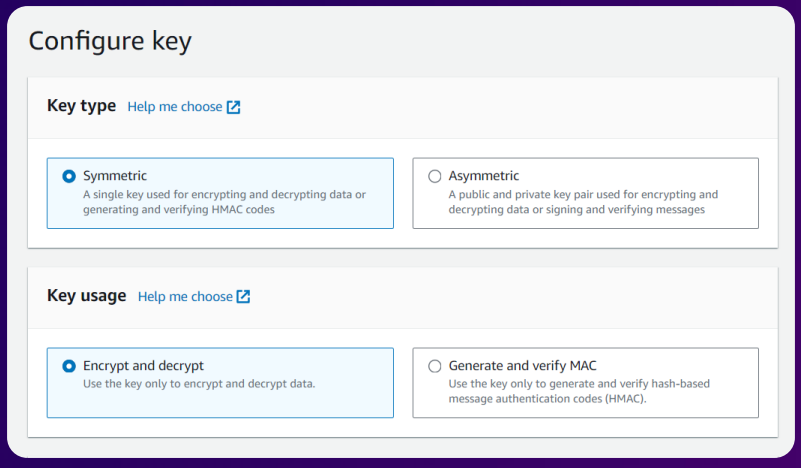
aws re/start

Task 1

Create an AWS KMS key

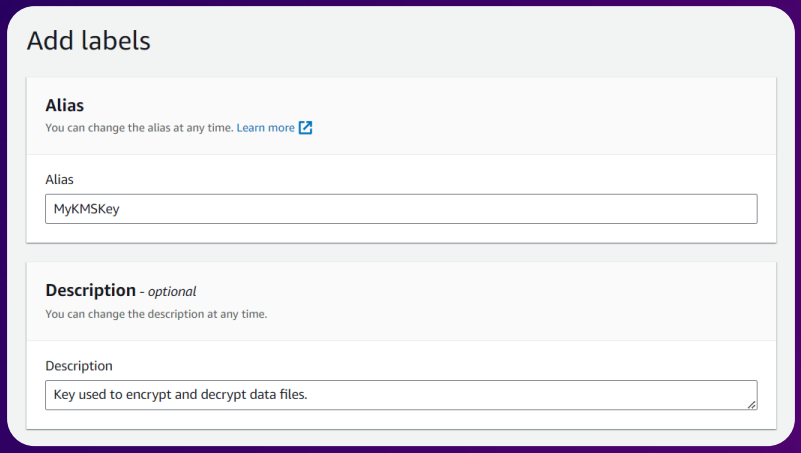
Step 3: Configure key

On the **Configure key** page, for Key type, choose Symmetric, and for Key usage, choose Encrypt and decrypt.



Step 4: Add labels

On the **Add labels** page, configure the Alias MyKMSKey and the Description Key used to encrypt and decrypt data files.



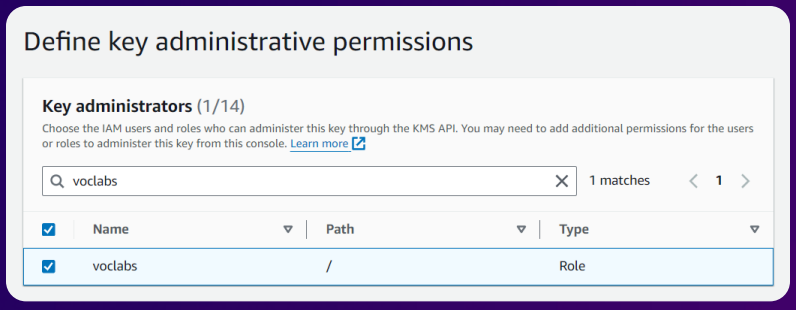
aws re/start

Task 1

Create an AWS KMS key

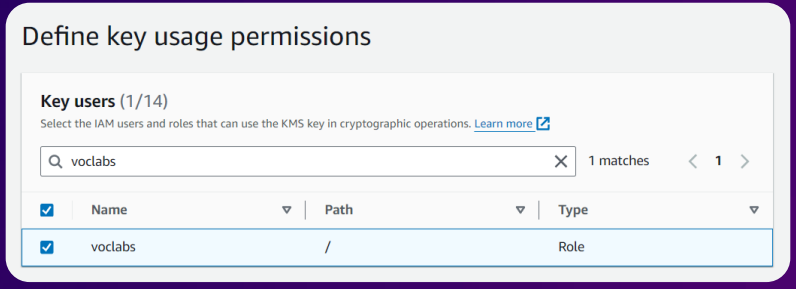
Step 5: Define key administrative permissions

On the **Define key administrative permissions** page, in the **Key administrators** section, select voclabs.



Step 6: Define key usage permissions

On the **Define key usage permissions** page, in the **Key users** section, select voclabs.



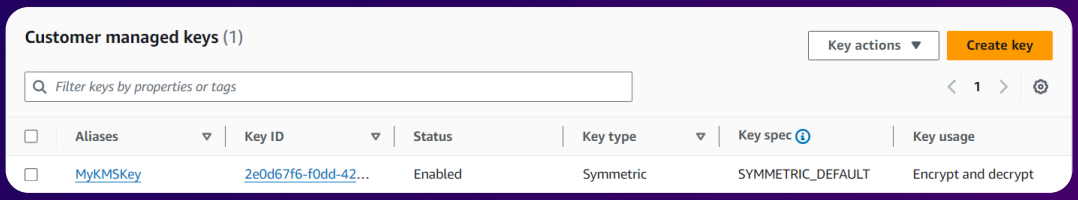
aws re/start

Task 1

Create an AWS KMS key

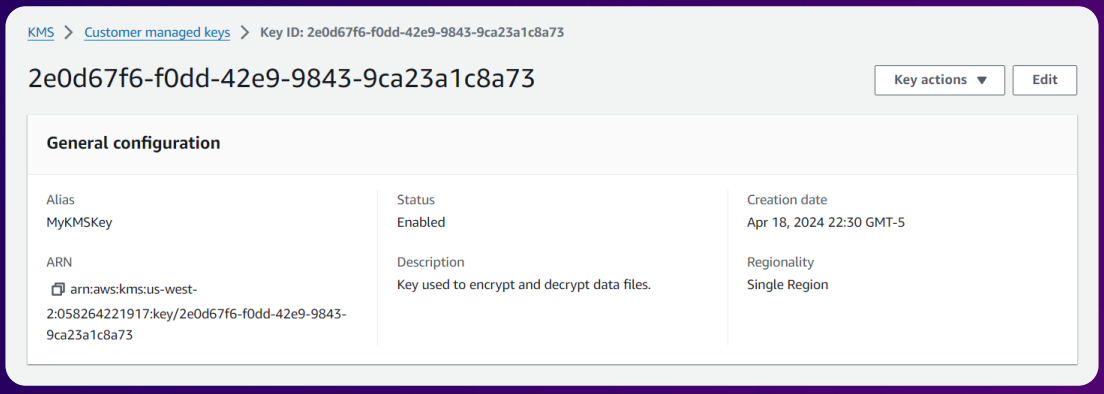
Step 7: Review Customer managed keys

Review the newly created customer managed key MyKMSKey.



Step 8: Copy the Amazon Resource Name

Copy the ARN (Amazon Resource Name) of the customer managed key MyKMSKey.



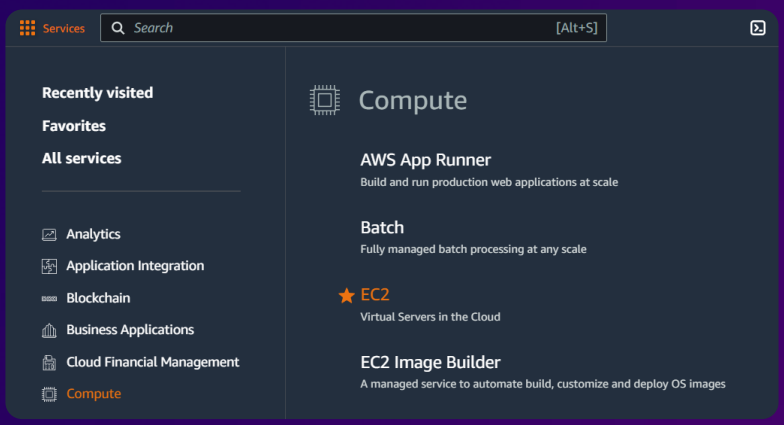
aws re/start

Task 2

Configure the File Server instance

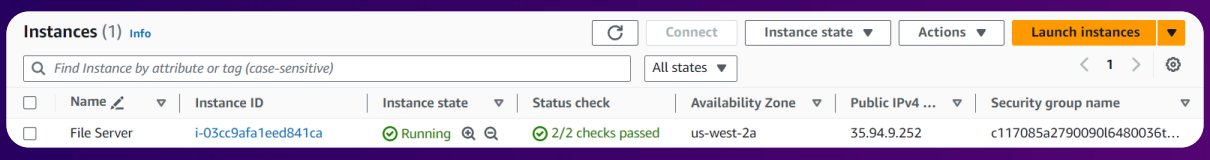
Step 1: Access the EC2 Management Console

Open the AWS Management Console, and select EC2.



Step 2: Review running instances

Navigate to the **Instances** section. The running File Server EC2 instance is listed.



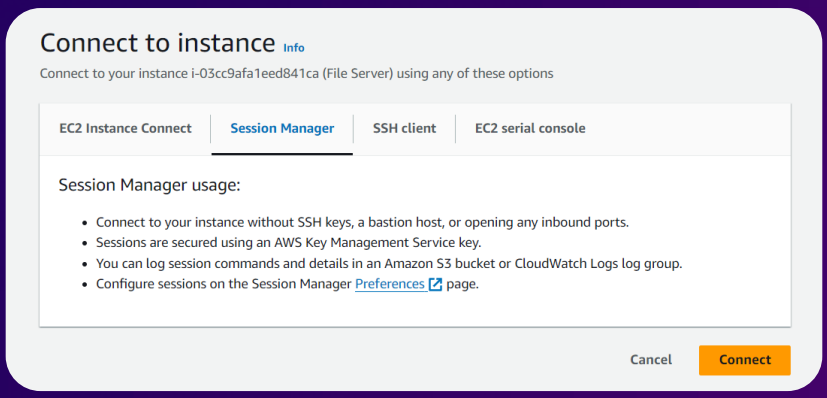
aws re/start

Task 2

Configure the File Server instance

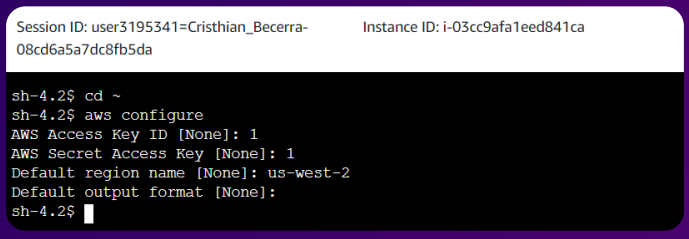
Step 3: Connect to the File Server instance

Connect to the File Server instance using Session Manager.



Step 4: Create the AWS configuration file

To create the AWS configuration file, run the following commands.



aws re/start

Task 2

Configure the File Server instance

Step 5: Open the AWS credentials file

Open the AWS credentials file using the Vim text editor.

```
sh-4.2$ vi ~/.aws/credentials
sh-4.2$
```

Step 6: Edit the AWS credentials file

Paste the aws_access_key_id, aws_secret_access_key, and aws_session_token credentials obtained from the AWS Details section. Then, save and close the Vim text editor.

```
[default]
aws_access_key_id=ASIAQ3EGRPDO2BVDMP7R
aws_secret_access_key=DUoKH65GWufH6uMSRFJbV0LAhLKFo6XAabxbI5Aj
aws_session_token=IQoJb3JpZ2luX2VjEMT//////////wEaCXVzLXdlc3QtMiJHMEUCIQCNHjt9mKyapfq
zegwz3Y5x5S1fceRLLQuL8263g95sYQIgJDbVaZ0vIvPFRKXvuWiGJFwG6liHZZcK37j+UzwSsPUqtAII/P//
////////ARAAGgwwNTgyNjQyMjE5MTciDFv1XH7zkBhXHTR/MCqIAnh7/3B374o8+CGCMAS3HkLI2G0U20dXQ
HEWbqpQnrMu96E2stwDl7Awp87LMkOGCKOYMiue6vtvRgZrws3OctbjRnm4ZEhAP2Mntd4ZKyjn5r1RFMjq9b
ES94bLoBp5UW2N7gInPg6tZ0NLCp0PJ1fz+jn2nGC3u4XXuLbWahAU+5yknqj0yjLSj43fFP9vuxdpDFdEIRj
nzMxo6shtPLw1Q8Uu6hjGSXRPSRuC2jBHrRrSuym4lMlwtEdCmwwo7NObIUOaPIsuAIizDTB8FRLHDwVbzRkE
6dyw5KpLTmeMNgM3JiXqWSKIX6fehMF5+hPyG5oPxXmA64wWnvMnQMSUUUOH38fLUDDAxYexBjqdAc+vy7SkK
a0veLHgJS9TkdFrINlT/eynleQZqcQudefdYO9dLsKY+in4n+3sPXVYtnRd4HJPKschZgl4lXzE71CsAlMkRL
4YS4xPx9483RnycUB8AsSc//tQRrfj7DWAUpTLel6nBYijzbJZRXc8WLzXBHbTfnDgWQmvOlRqrPNxulCfjLo
ZSRFM4NmPv4rqAllEZSJmSD+Zlpj18Wc=
~
~
~
-- INSERT --                                          4,799         All
```

Task 2

Configure the File Server instance

Step 7: Review the AWS credentials file

View the updated contents of the AWS credentials file.

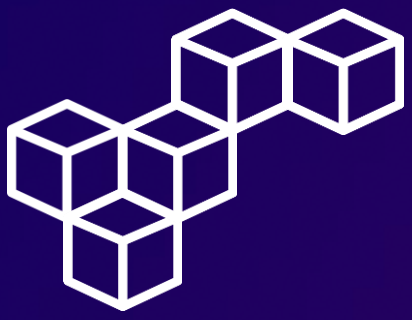
```
sh-4.2$ cat ~/.aws/credentials
[default]
aws_access_key_id=ASIAQ3EGRPDO2BVDMP7R
aws_secret_access_key=DUoKH65GWufH6uMSRFJbV0LAhLKFo6XAabxbI5Aj
aws_session_token=IQoJb3JpZ2luX2VjEMT//////////wEaCXVzLXdlc3QtMiJHMEUCIQCNHjt9mKyapfq
zegwz3Y5x5S1fceRLLQuL8263g95sYQIgJDbVaZ0vIvPFRKXvuWiGJFwG6liHZZcK37j+UzwSsPUqtAII/P//
////////ARAAGgwwNTgyNjQyMjE5MTciDFvlXH7zkBhXHTR/MCqIAnh7/3B374o8+CGCMAS3HkLI2G0U20dXQ
HEWbqpQnrMu96E2stwD17Awp87LMkOGCKOYMiue6vtvRgZrws3OctbjRnm4ZEhAP2Mntd4ZKyjn5r1RFMjq9b
ES94bLoBp5UWZN7gInPg6tZ0NLCp0PJ1fz+jn2nGC3u4XXuLbWahAU+5yknqj0yjLSj43fFP9vuxdpDFdEIRj
nzMxo6shtPLw1Q8Uu6hjGSXRPSRuC2jBHrRrSuym4lM1wtEdCmwwo7NObIUOaPIsuAIizDTB8FRLHDwVbzRkE
6dyw5KpLTmeMNgM3JiXqWSKIX6fehMF5+hPyG5oPxXmA64wWnvMnQMSUUUOH38fLUDDAxYexBjqdAc+vy7SkK
a0veLHgJS9TkdFrINIT/eynleQZqcQudefdYO9dLsKY+in4n+3sPXVYtnRd4HJPKschZg14lXzE71CsAlMkRL
4YS4xPx9483RnycUB8AsSc//tQRrfj7DWAUpTLel6nBYijzbJZRXc8WLzXBHbTfnDgWQmvOlRqrPNxulCfjLo
ZSRFM4NmPv4rqAllEZSJmSD+Zlpj18Wc=
sh-4.2$
```

Step 8: Install the AWS Encryption CLI

To install the AWS Encryption CLI and set your path, run the following commands.

```
sh-4.2$ pip3 install aws-encryption-sdk-cli
Defaulting to user installation because normal site-packages is not writeable
Collecting aws-encryption-sdk-cli
  Downloading aws_encryption_sdk_cli-4.1.0-py2.py3-none-any.whl (44 kB)
     |████████████████████████████████| 44 kB 2.8 MB/s
```

```
sh-4.2$ export PATH=$PATH:/home/ssm-user/.local/bin
sh-4.2$
```



Task 3

Encrypt and decrypt data

Step 1: Create a mock file

Create a text file with mock sensitive data in it.

```
sh-4.2$ touch secret1.txt secret2.txt secret3.txt
sh-4.2$ echo 'TOP SECRET 1!!!' > secret1.txt
sh-4.2$ cat secret1.txt
TOP SECRET 1!!!
sh-4.2$
```

Step 2: Save the KMS ARN in a variable

Create a directory to output the encrypted file. Then, save the ARN of the AWS KMS key you previously copied in the $keyArn variable.

```
sh-4.2$ mkdir output
sh-4.2$ keyArn=arn:aws:kms:us-west-2:058264221917:key/2e0d67f6-f0dd-42e9-9843-9ca23a1
c8a73
sh-4.2$
```

aws re/start

Task 3

Encrypt and decrypt data

Step 3: Encrypt the file

To encrypt the secret1.txt file, run the following command.

```
sh-4.2$ aws-encryption-cli --encrypt \
>                     --input secret1.txt \
>                     --wrapping-keys key=$keyArn \
>                     --metadata-output ~/metadata \
>                     --encryption-context purpose=test \
>                     --commitment-policy require-encrypt-require-decrypt \
>                     --output ~/output/.
sh-4.2$
```

Step 4: Review the encrypted file

Determine whether the command succeeded and view the contents of the newly encrypted file.

```
sh-4.2$ echo $?
0
sh-4.2$ ls output
secret1.txt.encrypted
sh-4.2$ cd output
sh-4.2$ cat secret1.txt.encrypted
x[illegible]aaws-crypto-public-keyDA6ASNF1mW7TKa//vsx7DMq/6vJQ8aNJlzVIQ
VxdbLZCMJS1DNAa/ikSIrBDknZfgQ==purposetestaws-kmsKarn:aws:kms:us-west-2:058264221917
:key/2e0d67f6-f0dd-42e9-9843-9ca23a1c8a73[illegible]
                                        H
                                                  3xD[illegible]
0oOm0h  [illegible]e.0                                                    ~0|  *[illegible]
                4!J>~[illegible] kV[illegible]( [illegible]1
                                                        , KI[illegible]pC[illegible]`[illegible].[illegible]H[illegible]
D=[illegible] i[illegible]
}Iφ[illegible]  [illegible]  [illegible]dPg0e1[illegible]/[illegible]9[illegible]xX@[illegible]vl`~[illegible]
[illegible]E[illegible]   M[illegible]=S[illegible]:[illegible]:[illegible]VT[illegible]       ;sh-4.2$
```

aws re/start

Task 3

Encrypt and decrypt data

Step 5: Decrypt the file

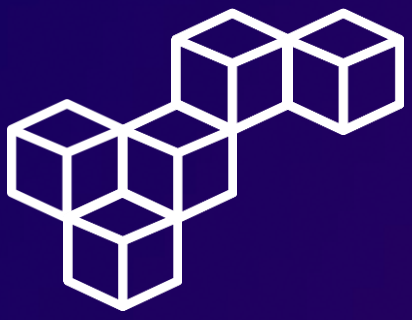
To decrypt the secret1.txt.encrypted file, run the following command.

```
sh-4.2$ aws-encryption-cli --decrypt \
>                     --input secret1.txt.encrypted \
>                     --wrapping-keys key=$keyArn \
>                     --commitment-policy require-encrypt-require-decrypt \
>                     --encryption-context purpose=test \
>                     --metadata-output ~/metadata \
>                     --max-encrypted-data-keys 1 \
>                     --buffer \
>                     --output .
sh-4.2$
```

Step 6: Review the decrypted file

View the contents of the decrypted file.

```
sh-4.2$ ls
secret1.txt.encrypted  secret1.txt.encrypted.decrypted
sh-4.2$ cat secret1.txt.encrypted.decrypted
TOP SECRET 1!!!
sh-4.2$
```



Conclusions

Key Management Service
AWS Key Management Service is essential for managing and controlling access to encryption keys securely in AWS.

Connecting to an instance using Session Manager
Connecting to an instance using Session Manager simplifies remote access management to EC2 instances without the need to open ports in network security.

aws configure
The AWS configuration file is key to setting regional configuration and default credentials for the AWS CLI.

~/.aws/credentials
The AWS credentials file securely stores access keys and session tokens for authenticating requests to AWS.

Encryption
Encryption is crucial for protecting sensitive data during storage or transmission, ensuring its confidentiality and integrity.

Decryption
Decryption is the process of reversing encryption to retrieve original data, important for securely accessing encrypted data.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru