

*Laboratorio #278*

# PROTECCIÓN DE DATOS MEDIANTE CIFRADO

Hecho con amor por:

Ignacio Suárez - Sebastian Aguilera

Joel Umpierrez - Gonza Rondeau

Sabrina Magnani - Agustín Rodríguez



# Objetivos

- Crear una clave de encriptación de AWS KMS
- Instalar la CLI de AWS Encryption
- Cifrar datos de texto simple y descifrarlo

# Tarea 1: Crear una clave de AWS KMS

- Crear una clave de encriptación de AWS KMS
- Con AWS KMS, puede crear y administrar claves criptográficas y controlar su uso a lo largo de una amplia variedad de servicios de AWS y en sus aplicaciones.

Key Management Service (KMS)

Security, Identity & Compliance

AWS managed keys

Customer managed keys

Custom key stores

# AWS Key Management Service

# Accedemos a AWS Key Management Service para crear y configurar una nueva clave de acceso

Configure key

Key type

Symmetric

Asymmetric

Key usage

Encrypt and decrypt

Generate and verify MAC

Add labels

Alias

MyKMSKey

Description - optional

Llave usada para encriptar y desencriptar datos de archivos

Define key administrative permissions

Key administrators (14)

	Name	Path	Type
<input type="checkbox"/>	AWSServiceRol...	/aws-service-ro...	Role
<input type="checkbox"/>	AWSServiceRol...	/aws-service-ro...	Role
<input type="checkbox"/>	AWSServiceRol...	/aws-service-ro...	Role
<input type="checkbox"/>	AWSServiceRol...	/aws-service-ro...	Role
<input type="checkbox"/>	AWSServiceRol...	/aws-service-ro...	Role
<input type="checkbox"/>	AWSServiceRol...	/aws-service-ro...	Role
<input type="checkbox"/>	c110983a2632...	/	Role
<input type="checkbox"/>	EMR_AutoScali...	/	Role
<input type="checkbox"/>	EMR_DefaultRole	/	Role
<input type="checkbox"/>	EMR_EC2_Defa...	/	Role

Define key administrative permissions

Key administrators (1/14)

vocla

1 matches

	Name	Path	Type
<input checked="" type="checkbox"/>	voclabs	/	Role

Key deletion

☒ Allow key administrators to delete this key.

Define key usage permissions

Key users (1/14)

vocla

1 matches

	Name	Path	Type
<input checked="" type="checkbox"/>	voclabs	/	Role

Success


Your AWS KMS key was created with alias MyKMSKey and key ID f2a74ab5-9519-4ae9-b56f-519028337deb.

Finalmente copiamos el ARN de la clave creada para usarlo luego

## General configuration

Alias

✔ ARN copied

 `arn:aws:kms:us-west-2:730335271440:key/f2a74ab5-9519-4ae9-b56f-519028337deb`

# Tarea 2: Configurar la instancia del servidor de archivos

Configuramos las credenciales del servidor de archivos para después poder cifrar y descifrar los datos.

Instalamos la CLI de cifrado de AWS (**aws-encryption-cli**) y así ejecutar los comandos.

discos compactos ~  
configurar aws

vi ~/.aws/credenciales

gato ~/.aws/credentials

pip3 instala aws-encryption-sdk-cli  
exportar RUTA = \$RUTA :/home/ssm-user/.local/bin

**Instances (1/1)** Info

Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive) All states

Instance state = running Clear filters

Name	Instance ID	Instance state	Instance type
File Server	i-06acfa292e9866445	Running	t2.micro

**i-06acfa292e9866445 (File Server)**

Details Status and alarms New Monitoring Security Networking

▼ Instance summary Info

Instance ID	Public IPv4 address
i-06acfa292e9866445 (File Server)	35.88.76.232   open address
Private IPv4 addresses	IPv6 address
10.0.2.201	-
Instance state	Public IPv4 DNS
Running	ec2-35-88-76-232.us-west-2.compute.amazonaws.com

EC2 > Instances > i-06acfa292e9866445 > Connect to instance

**Connect to instance** Info

Connect to your instance i-06acfa292e9866445 (File Server) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Session Manager usage:

- Connect to your instance without SSH keys, a bastion host, or opening any inbound ports.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager Preferences page.

Cancel Connect

En la lista de Instancias , seleccionamos nuestra instancia para Conectarnos mediante el Session Manager

```
sh-4.2$ cd ~
sh-4.2$ aws configure
AWS Access Key ID [None]: 1
AWS Secret Access Key [None]: 1
Default region name [None]: us-west-2
Default output format [None]:
```

Accedemos a editar con vim las credenciales de AWS

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



Borramos las credenciales anteriores y pegamos el bloque de código que copiamos anteriormente de Vocareum, para luego guardar el archivo con :wq

```
[default]
aws_access_key_id=ASIA2UC273IIH63EZUEC
aws_secret_access_key=nwUP2A16Wr/oMhPQIdyjKd6hYjhecKShroa38vsa
aws_session_token=IQoJb3JpZ2luX2VjEEcaCXVzLXdlc3QtMiJHMEUCIQCFdjUn6UQH482+Gsl2yTSZx5hB3JocuYICJj9lcode8
gIgO3bTPF0hfScCeHcGDFU9GtZoLqW2Mnzt/+4UaiSFGJIqwwIIsP/////////ARAAGgw3MzAzMzUyNzE0NDAiDEP5dVOrnch7SRoN
+yqXAvdx9pW6Y0j6IMFPvrdFk2lvX4x/VxrTrnpWLa0PXzi6HcI4jbQyj6IwICOrz2fP0Y+00iAVLZ3aiuQmDi1Rpe59pvhBDVLHNB
+OMXinw6fKJYA42P6PxcNFeuMANDw6XA8AxOeT1bqTet1V1nTMPL/4eYpSsqOQRf/bcc0lagxaWO3hv5ad4Acl+AHXLZZfij0bjOIxz
4CaddnPE5WNE7g6+jwhAsqyWjycaYe7SPuy+7d8K/s /bc6I5CQCpMJ/2TTwOLyWQjXWjp0S70fWXffYnzuUIJ0y14N3plt/18lavv/
+9kI2yolbkMn3juXDaLwiXKS17rTDayMtUN6snSMu4fuv5lAmuy5et987kYo7eSRg2kJjjCD65SyBjqdAdjpF7 XnasaY2AV879OgsO
n/D2ZiYUi4en0H0UTC8KMSvqyGuoixm4XEIf7NQyhfoEWQdtjy4VY+AcL+HIPX5Qj5HhadN/KSVBpjTLxbK1/gVvkPkPzeLrZFlni06
ASWJ/GoBNITK9mh2ypFyJ4wfgx KS170RyNlAqBKFH31sten8XKeW5KYjoqlidd2zORCNRdiIfz8/A9alBupCos=
```

Procedemos a instalar la CLI de AWS Encryption y establecemos la ruta de exportación

```
sh-4.2$ pip3 install aws-encryption-sdk-cli
```

```
Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: wrapt, pycparser, cffi, cryptography, typing-extensions, zipp, importlib
-metadata, attrs, jmespath, urllib3, six, python-dateutil, botocore, s3transfer, boto3, aws-encryption-
sdk, base64io, aws-encryption-sdk-cli
WARNING: The script aws-encryption-cli is installed in '/home/ssm-user/.local/bin' which is not on PA
TH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-scri
pt-location.
Successfully installed attrs-23.2.0 aws-encryption-sdk-3.2.0 aws-encryption-sdk-cli-4.1.0 base64io-1.0.
3 boto3-1.33.13 botocore-1.33.13 cffi-1.15.1 cryptography-42.0.7 importlib-metadata-6.7.0 jmespath-1.0.
1 pycparser-2.21 python-dateutil-2.9.0.post0 s3transfer-0.8.2 six-1.16.0 typing-extensions-4.7.1 urllib
3-1.26.18 wrapt-1.16.0 zipp-3.15.0
sh-4.2$ export PATH=$PATH:/home/ssm-user/.local/bin
sh-4.2$
```



# Tarea 3: Cifrar y descifrar los datos

- En esta tarea, crearemos un archivo de texto con información confidencial ficticia.
- Usaremos el cifrado para asegurar los contenidos del archivo.
- Descifraremos los datos y veremos los contenidos del archivo.

Con el comando `touch` crearemos tres archivos, estos son los archivos con información confidencial ficticia

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

Con *echo*, insertamos la información en el archivo “secret1.txt”

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

Podemos presenciar que nuestro archivo está en plenas condiciones para ser encriptado

```
sh-4.2$ mkdir output
sh-4.2$ keyArn=arn:aws:kms:us-west-2:730335271440:key/f2a74ab5-9519-4ae9-b56f-519028337deb
```

Crearemos un directorio “output” donde guardaremos las llaves de ARN

Usaremos el comando `keyArn` para guardar en una variable nuestras llaves



```
sh-4.2$ echo $? sh-4.2$ ls output
0                secret1.txt.encrypted
```



**Amazon  
Linux**



# Conclusiones

- Creamos correctamente claves de cifrado de AWS KMS.
- Instalamos con éxito la CLI de AWS Encryption.
- Realizamos adecuadamente el cifrado de un texto.
- Realizamos con éxito el descifrado de un texto cifrado.

¡Muchas gracias!