[ENGLISH]

Hello,

Cloud Gurus and welcome to this lecture where we're going to take a look at KMS

API calls,

and there are a few different API calls that you'll need to be aware of for the

exam and we're going to go through each one in turn and get our

hands dirty using the API. Now, in order to do that,

we're going to create an EC2 instance,

and we will log into our EC2 instance and configure the AWS

CLI using the credentials of the key user we

created in the previous demo and then we're going to create

a plain text file and use KMS to perform

encrypt and decrypt operations on our file.

We'll use the CMK that we created in the previous demo and then we're also

going to enable key rotation as well.

So if you're ready to get started, please join me in the console.

So here I am in the console and from our previous lecture,

you should still have,

your CMK and the user that we created previously and I'm

just going to head to KMS and show you the one I mean.

So you should have,

you'll CMK called my CMK and if you haven't done that already,

then go back and complete our previous lecture, creating your CMK.

So the next thing we're going to do is launch our EC2 instance.

So head to services, EC2,

and scroll down to launch instance,

and we're going to use the Amazon Linux to AMI and just make sure you're in

the same region as where you created your CMK. So for me,

that's going to be US East 1 North Virginia.

So select the Amazon Linux to AMI

we'll stick with the T2.micro. So hit next,

make sure you've got auto assigned public IP set to enable.

Hit Next, next,

next, we've got our security group here,

which is going to allow us to SSH to our EC2 instance on port 22.

So we're going to stick with the default for that hit review and launch

launch, and we'll create a new key pair,

and I'm just going to call mine North Virginia Key pair,

download the key pair and then finally hit launch instances.

So let's take a look and see if our instance is ready yet.

So there we are, it's up and running now,

and you might need to hit refresh just to refresh that view.

So I'm going to grab my public IP and copy that to my clipboard and head to a

terminal window on my local machine. So here I am in the downloads directory,

I'm just going to find my.PEM file and there it is and

as I'm working on a Mac,

I'm going to need to change the permissions on this .PEM file before I can

actually use it.

So now I should be able to log into my EC2 instance.

I'm going to paste my public

IP. So here I am on my EC2 instance, I'm just going to clear the screen.

So the first thing I'm going to do is create the plain text file that we are

going to encrypt.

So this is just going to create a simple text file with the text, hello,

cloud gurus.

So here's our text file and if we open the text file,

here is the contents of the file.

So this is all plain text file that we're going to encrypt,

but in order to do that, and in order to start using the KMS API,

we're going to need to configure the AWS command line interface and we're going

to do that using the credentials of the user that we created in our previous

lecture. So to do that,

we'll run AWS configure

and the first thing it's going to ask is for the access key of our user.

So head back to the console,

come to services, identity access management

users, select my key user,

go to security credentials,

and we're actually going to delete the existing access key

and we'll create a new one.

So the first thing we need is our access key ID.

So I'm just going to copy that and paste that in my terminal window.

It's now asking for the secret access key. So head back to the console

and we'll copy or secret access key enter the default

region. Name is going to be US East 1

and for output format, I'm just going to set it to text.

So now we've got everything set up to run our encryption commands,

and I've got a few different commands that I want to show you and these commands

are supplied as resources,

and you can find them in the resources section of the course.

So here is our first command,

which we're going to use to encrypt our file and with all of these commands that

we need to provide the key ID for our CMK.

So I'm going to head to the console and grab my key ID, close that down,

come to services, KMS,

and here is my key ID. So I'm just going to copy that.

Come back to my text file and I'm going to add my key ID in there?

In fact, I'm going to do it for all the commands where I need the key ID.

So beginning with the first command,

I'm going to copy that head back to my terminal window

and paste.

So this is the command that we use to encrypt our file.

So it's AWS KMS encrypt. We're providing our key ID.

This is our plain text file,

and we need to provide it in this format so that KMS knows where to find our

file and then this part of the command takes the encrypted

output, which is in ciphertext,

and then it decodes it using base 64.

So it turns it into a binary data file and the name of our encrypted

file is going to be encrypted, secret.txt and don't worry,

you don't need to memorize all of the syntax of these commands.

The main thing you need to remember is the name of the API call.

So in this case, it's AWS KMS encrypt. If you're happy with that,

just hit enter.

So that should have encrypted our file and created a new file called encrypted

at secret.txt.

So let's type LS and there is our encrypted file

and if we type file and the name of our encrypted file,

it shows it as a data file rather than an ASCII text file.

So let's see if we can open our file and read it. So as expected,

you can't read it. So this is a binary encrypted file, and we cannot read it.

So now let's try decrypting our file.

So I'm going to grab the second command,

come back to my terminal window and paste that.

So this command just runs AWS KMS decrypt,

and we provide the name of the ciphertext that we want it to decrypt and

in this case, it's going to be all encrypted, secret.txt,

and then this part of the command extract, the decrypted data,

and it performs a base of 64 decoding and then saves the

decoded file to a new file called decrypted secret.txt.

So let's just run that command and if we type LS,

we should see a third file has appeared called decrypted secret.txt

and if I run a file on decrypted at secret.txt we can see that it's an ASCII

text document and if we run Katz,

we can open the file and read it. So we've now decrypted or encrypted file.

Now the third command I want to show you is reencrypt.

So I'm going to grab my third commands, make sure we get the whole thing,

head back to my terminal and paste that command.

So reencrypt takes an encrypted file and in this case,

it's going to be encrypted secrets.txt and it decrypts it,

but without saving a plain text version anywhere.

So it's going to decrypt it and keep it in memory and then it's going to

reencrypt it and save that newly encrypted version to a new file.

And the new file is going to be called new encryption.txt.

So I'm going to hit enter,

and now we should have a fourth file and there it is.

So you might be wondering why you'd ever want to use this reencrypt command and

actually it's a really useful command. If, for example,

you ever want to encrypt something using a different CMK.

We could actually specify a different CMK ID here,

and we could reencrypt our encrypted file using a different CMK.

So say for example, you wanted to manually rotate or refresh your CMK,

or even if your CMK had become compromised for whatever reason and you want to

reencrypt everything using a different CMK.

So that's when you would use reencrypt and the next thing I wanted to show you

is how to enable automatic key rotation and you can do that using this command.

It's a really simple one.

AWS KMS enable key rotation and you provide the key ID.

So I'm just going to hit enter,

and this will enable key rotation and AWS will rotate your CMK on an

annual basis and if we want to check that it's worked, you just type AWS,

KMS, get key rotation status,

and then you provide your key ID and it should show you a key

rotation status for true and

then the last thing I wanted to show you is how to generate a data key.

So for encrypting anything over four kilobytes, you're going to need a data key.

So this is the command that we can use to do that.

So it's just AWS KMS generate data key.

We provide the key ID of our CMK,

and the type of key that we want to create and in this case,

it's going to be AES-256 bit data key.

So hit enter, and that has created our data command. So this

command returns a plain text version,

and also a cipher text version of the data key and a use case for

this kind of command might be, if, for example,

you had a Lambda function or an application which needed to run some encryption

and decryption operations on large amounts of data.

Then you could generate a data key and then use that key to encrypt your data

but really that's out of scope for this exam.

You don't need to know how to do that.

You just need to know the generated data key command and roughly when you would

use it.

So you would use it to generate a data key if you wanted to encrypt anything

over and above four kilobytes.

So those are the main KMS commands that I wanted to show you and I've included a

link to the AWS CLI documentation for KMS in case you are

interested in reading a little bit more about these commands,

but these are the main ones that you're to need to understand for the exam.

So let's take a look at my exam tips for the KMF API calls.

So the first command you're going to need to remember is the encrypt command and

we use AWS KMS encrypt to encrypt a plain text file

into ciphertext using a customer master key.

We use AWS KMS decrypt to decrypt ciphertext that was

encrypted using a customer master key.

We use AWS KMS reencrypt to decrypt ciphertext,

and then reencrypt it entirely within KMS and this is something that you

might want to use. If you want to change the CMK that you're using,

or you want to manually rotate the CMK. So it decrypts it,

and it doesn't save it as plain text.

It just goes straight into reencrypt tying it with the new CMK.

We've also got AWS KMS enabled key rotation,

and this enables automatic key rotation once per year.

So every 365 days and then finally,

we've got AWS KMS generate data key,

which uses the CMK to generate a data key,

which we're going to use it to encrypt any data over and above four kilobytes.

So that is the end of this lecture. If you have any questions,

please let me know. Otherwise, feel free to move on to the next lecture.

Thank you.

[SPANISH]

Hola,

Cloud Gurus y bienvenidos a esta conferencia donde vamos a echar un vistazo a KMS

llamadas API,

y hay algunas llamadas API diferentes que deberá tener en cuenta para el

examen y vamos a revisar cada uno por turno y obtener nuestro

manos sucias usando la API. Ahora, para hacer eso,

vamos a crear una instancia EC2,

e iniciaremos sesión en nuestra instancia EC2 y configuraremos el AWS

CLI utilizando las credenciales del usuario clave que

creado en la demostración anterior y luego vamos a crear

un archivo de texto sin formato y use KMS para realizar

cifrar y descifrar operaciones en nuestro archivo.

Usaremos la CMK que creamos en la demostración anterior y luego también

va a habilitar la rotación de claves también.

Entonces, si está listo para comenzar, únase a mí en la consola.

Así que aquí estoy en la consola y de nuestra conferencia anterior,

deberías tener,

tu CMK y el usuario que creamos anteriormente y estoy

solo voy a dirigirme a KMS y mostrarte a qué me refiero.

Entonces deberías tener,

llamará a mi CMK y si aún no lo ha hecho,

luego regrese y complete nuestra lección anterior, creando su CMK.

Entonces, lo siguiente que haremos es lanzar nuestra instancia EC2.

Así que dirígete a servicios, EC2,

y desplácese hacia abajo para iniciar la instancia,

y vamos a usar Amazon Linux para AMI y solo asegurarnos de que esté en

la misma región donde creó su CMK. Así que para mí,

eso va a ser EE.UU. Este 1 Norte de Virginia.

Así que seleccione Amazon Linux para AMI

nos quedaremos con el T2.micro. Así que pulsa siguiente,

asegúrese de tener una IP pública asignada automáticamente configurada para habilitar.

Presiona Siguiente , siguiente,

siguiente , tenemos nuestro grupo de seguridad aquí,

lo que nos permitirá usar SSH en nuestra instancia EC2 en el puerto 22.

Así que nos quedaremos con el valor predeterminado para esa revisión y lanzamiento exitosos.

launch , y crearemos un nuevo par de claves,

y solo voy a llamar a mi par de claves de Virginia del Norte,

descargue el par de claves y finalmente presione instancias de lanzamiento.

Así que echemos un vistazo y veamos si nuestra instancia ya está lista.

Así que ahí estamos, está funcionando ahora,

y es posible que deba presionar actualizar solo para actualizar esa vista.

Así que tomaré mi IP pública, la copiaré en mi portapapeles y me dirigiré a un

de terminal en mi máquina local. Así que aquí estoy en el directorio de descargas,

Voy a buscar mi archivo .PEM y ahí está y

como estoy trabajando en una Mac,

Voy a necesitar cambiar los permisos en este archivo .PEM antes de poder

realmente usarlo.

Así que ahora debería poder iniciar sesión en mi instancia EC2.

voy a pegar mi publico

IP. Así que aquí estoy en mi instancia EC2, solo voy a borrar la pantalla.

Así que lo primero que voy a hacer es crear el archivo de texto sin formato que estamos

va a cifrar.

Así que esto solo va a crear un archivo de texto simple con el texto, hola,

de la nube .

Así que aquí está nuestro archivo de texto y si abrimos el archivo de texto,

aquí está el contenido del archivo.

Este es un archivo de texto sin formato que vamos a cifrar,

pero para hacer eso, y para comenzar a usar la API de KMS,

vamos a necesitar configurar la interfaz de línea de comandos de AWS y vamos

para hacerlo usando las credenciales del usuario que creamos en nuestro anterior

conferencia \_ Así que para hacer eso,

ejecutaremos la configuración de AWS

y lo primero que nos va a pedir es la clave de acceso de nuestro usuario.

Así que regresa a la consola,

ven a servicios, administración de acceso a la identidad

usuarios , seleccione mi usuario clave,

ir a las credenciales de seguridad,

y en realidad vamos a eliminar la clave de acceso existente

y crearemos uno nuevo.

Entonces, lo primero que necesitamos es nuestra ID de clave de acceso.

Así que voy a copiar eso y pegarlo en la ventana de mi terminal.

Ahora está pidiendo la clave de acceso secreta. Así que regresa a la consola.

y copiaremos o clave de acceso secreta ingrese el valor predeterminado

región \_ El nombre será EE.UU. Este 1

y para el formato de salida, solo voy a configurarlo en texto.

Así que ahora tenemos todo configurado para ejecutar nuestros comandos de encriptación,

y tengo algunos comandos diferentes que quiero mostrarles y estos comandos

se suministran como recursos,

y puedes encontrarlos en la sección de recursos del curso.

Así que aquí está nuestro primer comando,

que vamos a usar para encriptar nuestro archivo y con todos estos comandos que

necesitamos proporcionar el ID de clave para nuestra CMK.

Así que me dirigiré a la consola y tomaré mi ID de clave, cerraré eso,

ven a servicios, KMS,

y aquí está mi clave de identificación. Así que solo voy a copiar eso.

¿Volver a mi archivo de texto y voy a agregar mi ID de clave allí?

De hecho, lo haré para todos los comandos en los que necesito el ID de la clave.

Así que comenzando con el primer comando,

Voy a copiar esa cabeza de vuelta a mi ventana de terminal.

y pegar

Este es el comando que usamos para cifrar nuestro archivo.

Entonces es cifrado de AWS KMS. Estamos proporcionando nuestra clave de identificación.

Este es nuestro archivo de texto sin formato,

y debemos proporcionarlo en este formato para que KMS sepa dónde encontrar nuestro

archivo y luego esta parte del comando toma el cifrado

salida , que está en texto cifrado ,

y luego lo decodifica usando base 64.

Entonces lo convierte en un archivo de datos binarios y el nombre de nuestro cifrado

el archivo va a ser encriptado, secret.txt y no te preocupes,

no necesita memorizar toda la sintaxis de estos comandos.

Lo principal que debe recordar es el nombre de la llamada a la API.

Entonces, en este caso, es cifrado de AWS KMS. Si estás contento con eso,

solo presiona enter.

Entonces eso debería haber encriptado nuestro archivo y creado un nuevo archivo llamado encriptado

en secreto.txt.

Así que escribamos LS y ahí está nuestro archivo encriptado

y si escribimos archivo y el nombre de nuestro archivo encriptado,

lo muestra como un archivo de datos en lugar de un archivo de texto ASCII.

Así que veamos si podemos abrir nuestro archivo y leerlo. Entonces, como era de esperar,

puedes leerlo. Este es un archivo cifrado binario y no podemos leerlo.

Así que ahora intentemos descifrar nuestro archivo.

Así que voy a agarrar el segundo comando,

vuelve a la ventana de mi terminal y pega eso.

Así que este comando solo ejecuta el descifrado de AWS KMS,

y proporcionamos el nombre del texto cifrado que queremos que descifre y

en este caso, todo estará encriptado, secret.txt,

y luego esta parte del extracto del comando, los datos descifrados,

y realiza una base de decodificación 64 y luego guarda el

decodificado a un nuevo archivo llamado decrypted secret.txt.

Así que ejecutemos ese comando y si escribimos LS,

deberíamos ver que ha aparecido un tercer archivo llamado decrypted secret.txt

y si ejecuto un archivo descifrado en secret.txt podemos ver que es un ASCII

de texto y si ejecutamos Katz,

podemos abrir el archivo y leerlo. Así que ahora hemos descifrado o cifrado el archivo.

Ahora, el tercer comando que quiero mostrarles es volver a cifrar .

Así que voy a agarrar mi tercer comando, asegurarme de que lo entendamos todo,

regresa a mi terminal y pega ese comando.

Así que volver a cifrar toma un archivo cifrado y, en este caso,

va a estar cifrado secrets.txt y lo descifra,

pero sin guardar una versión de texto sin formato en ningún lado.

Así que lo va a descifrar y guardar en la memoria y luego va a

vuelva a cifrarlo y guarde esa versión recién cifrada en un nuevo archivo.

Y el nuevo archivo se llamará nuevo cifrado.txt.

Así que voy a presionar enter,

y ahora deberíamos tener un cuarto archivo y ahí está.

Quizás se pregunte por qué querría usar este comando de recifrado y

en realidad es un comando realmente útil. Si, por ejemplo,

desea cifrar algo con una CMK diferente.

De hecho, podríamos especificar un ID de CMK diferente aquí,

y podríamos volver a cifrar nuestro archivo cifrado utilizando una CMK diferente.

Digamos, por ejemplo, que desea rotar o actualizar manualmente su CMK,

o incluso si su CMK se ha visto comprometida por cualquier motivo y desea

vuelva a cifrar todo usando una CMK diferente.

Ahí es cuando usarías volver a cifrar y lo siguiente que quería mostrarte

es cómo habilitar la rotación automática de claves y puede hacerlo usando este comando.

Es realmente simple.

AWS KMS habilita la rotación de claves y usted proporciona el ID de la clave.

Así que solo voy a presionar enter,

y esto habilitará la rotación de claves y AWS rotará su CMK en un

anualmente y si queremos comprobar que ha funcionado, simplemente escriba AWS,

KMS, obtener el estado de rotación de claves,

y luego proporciona su ID de clave y debería mostrarle una clave

de rotación para verdadero y

entonces lo último que quería mostrarte es cómo generar una clave de datos.

Entonces, para encriptar algo de más de cuatro kilobytes, necesitará una clave de datos.

Así que este es el comando que podemos usar para hacer eso.

Entonces, es solo que AWS KMS genera la clave de datos.

Proporcionamos el ID de clave de nuestra CMK,

y el tipo de clave que queremos crear y en este caso,

será una clave de datos AES de 256 bits.

Así que presione enter, y eso ha creado nuestro comando de datos. Así que esto

comando devuelve una versión de texto sin formato,

y también una versión de texto cifrado de la clave de datos y un caso de uso para

este tipo de comando podría ser, si, por ejemplo,

tenía una función Lambda o una aplicación que necesitaba ejecutar alguna encriptación

y operaciones de descifrado en grandes cantidades de datos.

Entonces podría generar una clave de datos y luego usar esa clave para cifrar sus datos

pero realmente eso está fuera del alcance de este examen.

No necesitas saber cómo hacer eso.

Solo necesita conocer el comando clave de datos generado y aproximadamente cuándo lo haría.

úsalo \_

Entonces lo usaría para generar una clave de datos si quisiera cifrar algo

más de cuatro kilobytes.

Esos son los principales comandos de KMS que quería mostrarles y he incluido un

enlace a la documentación de AWS CLI para KMS en caso de que esté

interesado en leer un poco más sobre estos comandos,

pero estos son los principales que debe comprender para el examen.

Así que echemos un vistazo a mis consejos de examen para las llamadas a la API de KMF.

Entonces, el primer comando que necesitará recordar es el comando cifrar y

usamos el cifrado AWS KMS para cifrar un archivo de texto sin formato

en texto cifrado utilizando una clave maestra de cliente.

Usamos el descifrado de AWS KMS para descifrar el texto cifrado que fue

cifrado mediante una clave maestra de cliente.

Usamos el recifrado de AWS KMS para descifrar el texto cifrado ,

y luego volver a cifrarlo por completo dentro de KMS y esto es algo que

podría querer usar. Si desea cambiar la CMK que está utilizando,

o desea rotar manualmente la CMK. Entonces lo descifra,

y no lo guarda como texto sin formato.

Simplemente va directamente a volver a cifrar vinculándolo con la nueva CMK.

También tenemos la rotación de claves habilitada para AWS KMS,

y esto permite la rotación automática de claves una vez al año.

Entonces cada 365 días y finalmente,

tenemos AWS KMS generar clave de datos,

que utiliza la CMK para generar una clave de datos,

que vamos a utilizar para cifrar cualquier dato por encima de los cuatro kilobytes.

Así que ese es el final de esta conferencia. Si tienes alguna pregunta,

por favor hágamelo saber De lo contrario, siéntase libre de pasar a la siguiente lección.

Gracias.