[ENGLISH]

Hello Cloud Gurus and welcome to this lesson

on Securely Storing Secrets Using AWS Secrets Manager.

And we'll begin with an introduction to Secrets Manager.

We'll take a look at storing database secrets

using Secrets Manager,

and the other type of secrets

that you can store with Secrets Manager.

Next we're gonna have a demo.

So we're gonna create a secret of our own

for an RDS database.

We're gonna compare Secrets Manager

to Systems Manager Parameter Store

and finish up with some exam tips.

Now, Secrets Manager allows you to protect

and store your secrets for AWS services,

IT resources, and applications.

And it allows you to centrally manage your secrets

that are used to access resources

both inside and outside of AWS.

You can also automate the rotation of your secrets

without having to deploy any additional code to do so.

And it also allows us to secure our secrets

using fine grain permissions like resource policies

and encrypt our secrets using AWS, KMS,

or Key Management Service.

Now Secrets Manager allows us to protect

several different types of secrets.

For instance, you can store secrets

for your RDS database, your Redshift Cluster,

DocumentDB database,

other databases,

as well as API keys that you might be storing

inside or outside of AWS.

So now let's talk about the kind of database secrets

that can be stored within Secrets Manager

and with RDS for MySQL, PostgreSQL,

Oracle, MariaDB, and SQL Server,

all of these allow you to automate the management

of database secrets using Secrets Manager.

And the kind of information stored in these secrets

is gonna be things like the database username and password,

the server address that you use to access the database,

as well as the database name and port.

Secrets are encrypted using an AWS KMS key.

So it uses a customer master key

and the customer master key is a logical representation

of a master key.

It holds the key material that's gonna be used

for encrypting your data.

And with a customer master key

you can encrypt up to four kilobytes of data.

It's best practice to configure the automatic rotation

of secrets.

For instance, you can configure your secrets

to rotate every 30, 60, 90 or a custom number of days

with 365 days being the maximum.

Secret rotation is performed by Lambda.

And when you first create a secret,

you'll get the option to either create a new Lambda function

or you can use an existing one

if one exists in your account.

And it's this Lambda function

that's going to perform the rotation.

So if it's a database password that you are storing

the function will go in and update the password for you

automatically based on the rotation schedule

that you define.

But we are gonna see all of this in action in our demo.

And we'll begin by creating an RDS database

and we're gonna use PostgreSQL.

Next, we'll create a secret

and we're gonna store our database password as our secret.

And then finally

we'll be able to view our secret information

in Secrets Manager.

So from the console, first of all, I want you to search

for RDS,

and then select create database.

Select Easy Create, Database Engine is gonna be PostgreSQL.

Under database instant size select Free Tier.

The master username is gonna be Postgres

and we need to create a master password.

So type a master password and confirm it.

And you need to remember the password

because this is the password

that we're gonna be storing in Secrets Manager.

Once you've done that, scroll down to the bottom

and create database

and we'll just need to wait a few minutes

for it to complete.

And then once it's done, we'll be good to continue.

Now that our RDS database has been created,

let's go ahead and store our database password

in Secrets Manager.

So search for Secrets Manager and select that.

Store a new secret.

The secret type is gonna be credentials for an RDS database.

And then down here you're gonna type in the credentials

that we want to store as a secret.

So that's gonna be our Postgres username and password.

And hopefully you've remembered what your password was.

By default, it's gonna encrypt your password

using a KMS key that is managed by Secrets Manager

or you can create a KMS key that you manage

and use that encryption key instead.

Next, select your database,

and it should have appeared down here.

There it is.

Click next.

And we need to give our secret a descriptive name

so that we can easily find it later.

And I'll call it RDS hyphen Prod PostgreSQL.

And I'm gonna use that for my description as well.

I'll go ahead and add a tag of, created by and then my name.

So we know who created the secret.

Down here we've got the option

to edit our resource permissions.

And let's say that you want to give access

to a specific Lambda function or EC two instance.

Then you can simply paste your resource policy in here.

Under replication

this is where you can replicate your secret

to another region.

And I'm gonna select US East to Ohio.

And once again, by default it's gonna encrypt your secret

in the replication destination as well.

So then hit next.

And this is where I have the option

to add automatic rotation.

So select automatic rotation.

Under rotation schedule,

you can create your own rotation schedule

using a cron expression if you select this option.

But the easiest way is to use

the schedule expression builder.

And we'll enable rotation on a schedule of 60 days.

So update my days to 60.

There's also an option to set a duration window.

Let's say you've got a four hour window

in which you would like to rotate the secret.

You can set that in here, and you can also select

whether you want the secret to rotate

immediately after it's been stored.

Scrolling down, if you remember,

rotation is handled by a Lambda function

and Secrets Manager can automatically create

a new Lambda function for you

or you could use one in your account if you already had one.

But we are gonna create a new one.

We need to give it a name

and we don't need to use any separate credentials.

And then hit next.

On this page we can review all of our settings.

And then down here at the bottom

we can see that AWS has provided some sample code

in various different programming languages

to help us access our secret later on.

So this is to help you retrieve the secret

when you are in your application.

So that's pretty cool that they've provided that

and they've provided it

in lots of different programming languages.

So now let's go ahead and select store.

And as we can see

our secret is currently in the process of being created.

It does take a couple of minutes to complete everything

including getting the rotation started.

After it's finished, we can see

that replication has succeeded

and secret rotation is enabled.

So now we can select view details

and we can see all the information about our secret.

So the encryption key that was used, our secret name,

it's Amazon Resource Name, or ARN, our tags,

here's our secret value.

And we can retrieve our secret value

by clicking this button and it will show us the details

of the secret that we stored.

And there is my very insecure database password.

It's a good job with storing it in Secrets Manager.

Scrolling down, you can also see that rotation is enabled

and our rotation schedule is 60 days.

And then down here at the bottom, we've got our replication.

And finally you can access the sample code

down here as well.

So that is everything that I wanted to show you

on Secrets Manager.

And you will also need to understand the differences

between AWS Secrets Manager

and Systems Manager Parameter Store,

including when to use each one.

So Secrets Manager

is great for storing database credentials,

API keys,

and it can be used to automatically rotate your keys.

And it's a great service that is gonna help you meet

any security and compliance requirements

in relation to your database credentials or API keys.

Systems Manager Parameter Store on the other hand

has a wider set of use cases.

And as systems manager generally

is used to perform maintenance updates

on your EC two instances and other AWS resources

Parameter Store is designed to be used to

store configuration variables and license keys.

So finally onto my Secrets Manager exam tips.

Just remember that Secrets Manager is used to

centrally manage our secrets,

rotate them without the need for code deployments,

and secure our secrets using fine-grain permissions,

and encrypt them using AWS KMS.

Keep in mind the various different types

of secret that you can store in Secrets Manager.

So RDS database secrets as we demonstrated,

as well as secrets for your Redshift Cluster,

DocumentDB database,

other databases and API keys as well.

And then lastly make sure you understand

when to use Secrets Manager

and when to use Systems Manager Parameter Store.

And the way that I try to remember it

is that Secrets Manager is for anything that's secret

or confidential, whereas Parameter Store is all

about configuration parameters.

So that's it for this lesson.

Any questions, please let me know.

And if you're ready to move on

please join us in the next lesson.

Thank you.

[SPANISH]

Hola Cloud Gurus y bienvenidos a esta lección.

sobre el almacenamiento seguro de secretos mediante AWS Secrets Manager.

Y comenzaremos con una introducción a Secrets Manager.

Echaremos un vistazo al almacenamiento de secretos de bases de datos.

usando el administrador de secretos,

y el otro tipo de secretos

que puede almacenar con Secrets Manager.

A continuación vamos a tener una demostración.

Así que vamos a crear un secreto propio

para una base de datos RDS.

Vamos a comparar Secrets Manager

al almacén de parámetros de Systems Manager

y termine con algunos consejos para el examen.

Ahora, Secrets Manager le permite proteger

y almacene sus secretos para los servicios de AWS,

Recursos y aplicaciones de TI.

Y te permite gestionar de forma centralizada tus secretos

que se utilizan para acceder a los recursos

tanto dentro como fuera de AWS.

También puedes automatizar la rotación de tus secretos

sin tener que implementar ningún código adicional para hacerlo.

Y también nos permite asegurar nuestros secretos.

usar permisos de grano fino como políticas de recursos

y encriptar nuestros secretos usando AWS, KMS,

o Servicio de gestión de claves.

Ahora Secrets Manager nos permite proteger

varios tipos diferentes de secretos.

Por ejemplo, puede almacenar secretos

para su base de datos RDS, su Redshift Cluster,

DocumentDB ,

otras bases de datos,

así como las claves API que podría estar almacenando

dentro o fuera de AWS.

Así que ahora hablemos del tipo de secretos de bases de datos

que se pueden almacenar dentro de Secrets Manager

y con RDS para MySQL, PostgreSQL ,

Oracle, MariaDB y SQL Server,

todo esto le permite automatizar la gestión

de secretos de base de datos usando Secrets Manager.

Y el tipo de información almacenada en estos secretos

es van a ser cosas como el nombre de usuario y la contraseña de la base de datos,

la dirección del servidor que utiliza para acceder a la base de datos,

así como el nombre de la base de datos y el puerto.

Los secretos se cifran mediante una clave de AWS KMS.

Entonces usa una clave maestra de cliente

y la clave maestra del cliente es una representación lógica

de una llave maestra.

Contiene el material clave que se utilizará .

para cifrar sus datos.

Y con una clave maestra de cliente

puede cifrar hasta cuatro kilobytes de datos.

Es una buena práctica configurar la rotación automática

de secretos

Por ejemplo, puede configurar sus secretos

para rotar cada 30, 60, 90 o un número personalizado de días

siendo 365 días el máximo.

Lambda realiza la rotación secreta.

Y cuando creas un secreto por primera vez,

obtendrá la opción de crear una nueva función Lambda

o puedes usar uno existente

si existe uno en su cuenta.

Y es esta función Lambda

que va a realizar la rotación.

Entonces, si es una contraseña de base de datos que está almacenando

la función entrará y actualizará la contraseña por usted

automáticamente basado en el programa de rotación

que tu definas.

Pero vamos a ver todo esto en acción en nuestra demostración.

Y comenzaremos creando una base de datos RDS

y vamos a usar PostgreSQL .

A continuación, crearemos un secreto.

y almacenaremos la contraseña de nuestra base de datos como nuestro secreto.

y finalmente

podremos ver nuestra información secreta

en Administrador de secretos.

Así que desde la consola, antes que nada, quiero que busques

para RDS,

y luego seleccione crear base de datos.

Seleccione Easy Create, el motor de base de datos será PostgreSQL .

En el tamaño instantáneo de la base de datos, seleccione Nivel gratuito.

de usuario maestro será Postgres

y necesitamos crear una contraseña maestra.

Así que escriba una contraseña maestra y confírmela.

Y necesitas recordar la contraseña.

porque esta es la contraseña

que vamos a almacenar en Secrets Manager.

Una vez que hayas hecho eso, desplázate hacia abajo hasta la parte inferior

y crear base de datos

y solo tendremos que esperar unos minutos

para que se complete.

Y luego, una vez que esté hecho, estaremos bien para continuar.

Ahora que se ha creado nuestra base de datos RDS,

sigamos adelante y almacenemos la contraseña de nuestra base de datos

en Administrador de secretos.

Así que busca Secrets Manager y selecciónalo.

Guarda un nuevo secreto.

El tipo secreto serán las credenciales para una base de datos RDS.

Y luego aquí abajo vas a escribir las credenciales

que queremos guardar como un secreto.

Ese será nuestro nombre de usuario y contraseña de Postgres .

Y espero que hayas recordado cuál era tu contraseña.

Por defecto, va a cifrar su contraseña

utilizando una clave KMS administrada por Secrets Manager

o puede crear una clave KMS que usted administre

y use esa clave de cifrado en su lugar.

A continuación, seleccione su base de datos,

y debería haber aparecido aquí abajo.

Ahí está.

Haga clic en Siguiente.

Y necesitamos darle a nuestro secreto un nombre descriptivo

para que podamos encontrarlo fácilmente más tarde.

Y lo llamaré RDS guión Prod PostgreSQL .

Y voy a usar eso para mi descripción también.

Continuaré y agregaré una etiqueta de, creado por y luego mi nombre.

Entonces sabemos quién creó el secreto.

Aquí abajo tenemos la opción

para editar nuestros permisos de recursos.

Y digamos que quieres dar acceso

a una función Lambda específica o una instancia EC two.

Entonces simplemente puede pegar su política de recursos aquí.

Bajo replicación

aquí es donde puedes replicar tu secreto

a otra región.

Y voy a seleccionar EE.UU. Este a Ohio.

Y una vez más, por defecto cifrará tu secreto

en el destino de replicación también.

Entonces pulsa siguiente.

Y aquí es donde tengo la opción.

para agregar la rotación automática.

Así que seleccione la rotación automática.

Bajo el cronograma de rotación,

puedes crear tu propio horario de rotación

utilizando una expresión cron si selecciona esta opción.

Pero la forma más fácil es usar

el generador de expresiones de programación.

Y habilitaremos la rotación en un cronograma de 60 días.

Así que actualice mis días a 60.

También hay una opción para establecer una ventana de duración.

Digamos que tienes una ventana de cuatro horas

en el que le gustaría rotar el secreto.

Puede establecer eso aquí, y también puede seleccionar

si quieres que el secreto gire

inmediatamente después de haber sido almacenado.

Desplazándose hacia abajo, si recuerdas,

la rotación es manejada por una función Lambda

y Secrets Manager puede crear automáticamente

una nueva función Lambda para usted

o podría usar uno en su cuenta si ya tenía uno.

Pero vamos a crear uno nuevo.

Tenemos que darle un nombre

y no necesitamos usar ninguna credencial separada.

Y luego pulsa siguiente.

En esta página podemos revisar todas nuestras configuraciones.

Y luego aquí abajo en la parte inferior

podemos ver que AWS ha proporcionado un código de muestra

en varios lenguajes de programación diferentes

para ayudarnos a acceder a nuestro secreto más adelante.

Así que esto es para ayudarte a recuperar el secreto.

cuando estés en tu aplicación.

Así que es genial que hayan proporcionado eso

y lo han proporcionado

en muchos lenguajes de programación diferentes.

Así que ahora sigamos adelante y seleccionemos la tienda.

Y como podemos ver

nuestro secreto está actualmente en proceso de creación.

Se tarda un par de minutos en completar todo.

incluyendo el inicio de la rotación.

Después de que haya terminado, podemos ver

que la replicación ha tenido éxito

y la rotación secreta está habilitada.

Entonces ahora podemos seleccionar ver detalles

y podremos ver toda la información sobre nuestro secreto.

Entonces, la clave de encriptación que se usó, nuestro nombre secreto,

es el Nombre de recurso de Amazon, o ARN, nuestras etiquetas,

aquí está nuestro valor secreto.

Y podemos recuperar nuestro valor secreto

haciendo clic en este botón y nos mostrará los detalles

del secreto que guardamos.

Y está mi contraseña de base de datos muy insegura.

Es un buen trabajo almacenarlo en Secrets Manager.

Desplazándose hacia abajo, también puede ver que la rotación está habilitada

y nuestro horario de rotación es de 60 días.

Y luego aquí abajo, tenemos nuestra réplica.

Y finalmente puedes acceder al código de muestra.

abajo también.

Así que eso es todo lo que quería mostrarte.

en Administrador de secretos.

Y también necesitarás entender las diferencias.

entre el administrador de secretos de AWS

y almacén de parámetros del administrador de sistemas,

incluyendo cuándo usar cada uno.

Administrador de secretos

es excelente para almacenar credenciales de bases de datos,

claves API,

y se puede usar para rotar automáticamente sus llaves.

Y es un gran servicio que te ayudará a cumplir

cualquier requisito de seguridad y cumplimiento

en relación con sus credenciales de base de datos o claves API.

Systems Manager Parameter Store por otro lado

tiene un conjunto más amplio de casos de uso.

Y como administrador de sistemas en general

se utiliza para realizar actualizaciones de mantenimiento

en sus dos instancias de EC y otros recursos de AWS

El almacén de parámetros está diseñado para ser utilizado para

almacenar variables de configuración y claves de licencia.

Así que, finalmente, a mis consejos para el examen Secrets Manager.

Solo recuerda que Secrets Manager se utiliza para

de forma centralizada nuestros secretos,

rotarlos sin necesidad de implementaciones de código,

y asegure nuestros secretos usando permisos detallados,

y cifrarlos con AWS KMS.

Tenga en cuenta los diferentes tipos

de secreto que puede almacenar en Secrets Manager.

Entonces, los secretos de la base de datos RDS como demostramos,

así como secretos para su Redshift Cluster,

DocumentDB ,

otras bases de datos y claves API también.

Y luego, por último, asegúrese de entender

Cuándo usar Secrets Manager

y cuándo utilizar el Almacén de parámetros de Administrador de sistemas.

Y la forma en que trato de recordarlo

es que Secrets Manager es para cualquier cosa que sea secreta

o confidencial, mientras que Parameter Store es todo

sobre los parámetros de configuración.

Eso es todo por esta lección.

Para cualquier duda, por favor hágamelo saber.

Y si estás listo para seguir adelante

por favor únase a nosotros en la próxima lección.

Gracias.