[ENGLISH]

Hello, Cloud Gurus, and welcome to this lesson

which is part three of the EC2 section summary,

beginning with Secrets Manager and Parameter Store.

So remember that Secrets Manager

is a secure way of storing secrets

so that they can be accessed

by applications and AWS resources.

It's a great place to store database credentials

as well as API keys.

And the real differentiator is that it supports

automatic rotation of your database passwords and API keys,

whereas Parameter Store is for wider use cases.

For instance, configuration variables

that are not necessarily secrets,

parameters that you define, or for license keys.

And if you did use it

to store database connection information

or database credentials,

then just remember it does not support

database password rotation.

So you cannot use Parameter Store

to rotate your database credentials.

MemoryDB for Redis is an in-memory database.

With ultra fast performance,

you get microsecond read

and single-digit millisecond write capability

and it is massively scalable to over 100 terabytes.

Use cases include high-performance,

large-scale microservices applications,

or online gaming with millions of users

sharing digital assets.

And if you have to choose between MemoryDB and ElastiCache,

then just be aware that MemoryDB can be used

to store your whole data set in memory,

no database required,

so it's a replacement for a traditional database,

whereas ElastiCache is an in-memory cache

for databases like RDS databases,

so it's an add-on to your existing database.

Moving on to RDS Proxy,

remember that RDS Proxy is used to scale your applications

by pooling and sharing database connections

to assist with application scalability

and database efficiency.

RDS Proxy manages the connection,

so your application is pointed towards RDS Proxy

and the RDS database receives information

from the application through RDS Proxy.

And the great thing about RDS Proxy

is that it is serverless and scales automatically

so it scales automatically to your workload

through pooling and sharing database connections.

It preserves application connections during failover,

detect failover, and route requests

to the standby database quickly.

It's deployable over multi-AZ

for protection against infrastructure failure,

and it enables up to 66% faster failover times.

EC2 Image Builder automates the process of creating

and maintaining AMI and container images.

It's a four-step process

whereby you select a base operating system image,

you customize by adding software,

then test the image

and distribute it to your chosen regions.

So the process looks like this.

First of all, you provide the base OS image,

for instance, the Amazon Linux 2 AMI.

Next, you define the software that you want to install,

for instance, .NET, Node.js, Python, et cetera.

And you can also add the latest security updates

and the latest kernel patches, et cetera.

The third step is to run tests on the new image.

For instance, you can test

whether the image boots correctly.

And then finally, distribute the image

to the regions of your choice.

And by default, it's going to be distributed

to the region that you are operating in.

AMI is a regional,

and if you would like to use an existing AMI

in a different region, then you'll need to create a copy

and specify a destination region for the copy.

You can apply encryption during the copying process.

However, you cannot remove encryption

during the copying process.

So if you're making a copy of an encrypted AMI,

then the copy is going to be encrypted.

Well, that is it for this lesson.

Any questions, let me know.

Otherwise, please join me in the next one, thank you.

[SPANISH]

Hola, Cloud Gurus, y bienvenidos a esta lección.

que es la tercera parte del resumen de la sección EC2,

comenzando con Secrets Manager y Parameter Store.

Así que recuerda que Secrets Manager

es una forma segura de almacenar secretos

para que puedan ser accedidos

por aplicaciones y recursos de AWS.

Es un gran lugar para almacenar las credenciales de la base de datos.

así como claves API.

Y el verdadero diferenciador es que admite

automática de las contraseñas de su base de datos y claves API,

mientras que Parameter Store es para casos de uso más amplios.

Por ejemplo, las variables de configuración

que no son necesariamente secretos,

parámetros que defina o para claves de licencia.

Y si lo usaste

para almacenar información de conexión a la base de datos

o credenciales de base de datos,

entonces solo recuerda que no es compatible

de la base de datos .

Entonces no puedes usar la tienda de parámetros

para rotar las credenciales de su base de datos.

MemoryDB para Redis es una base de datos en memoria.

Con un rendimiento ultrarrápido ,

obtienes una lectura de microsegundos

y capacidad de escritura de milisegundos de un solo dígito

y es enormemente escalable a más de 100 terabytes.

Los casos de uso incluyen alto rendimiento,

Gran escala aplicaciones de microservicios ,

o juegos en línea con millones de usuarios

compartir activos digitales.

Y si tiene que elegir entre MemoryDB y ElastiCache ,

entonces solo tenga en cuenta que se puede usar MemoryDB

para almacenar todo su conjunto de datos en la memoria,

no se requiere base de datos,

por lo que es un reemplazo para una base de datos tradicional,

mientras ElastiCache es un caché en memoria

para bases de datos como bases de datos RDS,

por lo que es un complemento a su base de datos existente.

Pasando a RDS Proxy,

recuerda que RDS Proxy se utiliza para escalar tus aplicaciones

agrupando y compartiendo conexiones de base de datos

para ayudar con la escalabilidad de la aplicación

y la eficiencia de la base de datos.

RDS Proxy gestiona la conexión,

por lo que su aplicación apunta hacia RDS Proxy

y la base de datos RDS recibe información

desde la aplicación a través de RDS Proxy.

Y lo mejor de RDS Proxy

es que es serverless y escala automáticamente

por lo que se adapta automáticamente a su carga de trabajo

mediante la agrupación y el intercambio de conexiones de bases de datos.

Conserva las conexiones de la aplicación durante la conmutación por error,

detectar conmutación por error y enrutar solicitudes

rápidamente a la base de datos en espera.

Es desplegable sobre multi-AZ

para la protección contra fallas en la infraestructura,

y permite tiempos de conmutación por error hasta un 66 % más rápidos.

EC2 Image Builder automatiza el proceso de creación

y el mantenimiento de imágenes AMI y de contenedores.

Es un proceso de cuatro pasos.

mediante el cual selecciona una imagen base del sistema operativo,

usted personaliza agregando software,

luego prueba la imagen

y distribúyalo a las regiones que elija.

Así que el proceso se ve así.

En primer lugar, proporciona la imagen base del sistema operativo,

por ejemplo, la AMI de Amazon Linux 2.

A continuación, define el software que desea instalar,

por ejemplo, .NET, Node.js, Python, etcétera.

Y también puedes agregar las últimas actualizaciones de seguridad

y los últimos parches del kernel, etcétera.

El tercer paso es ejecutar pruebas en la nueva imagen.

Por ejemplo, puede probar

si la imagen arranca correctamente.

Y finalmente, distribuya la imagen

a las regiones de su elección.

Y por defecto, se distribuirá

a la región en la que está operando.

AMI es un regional,

y si desea utilizar una AMI existente

en una región diferente, deberá crear una copia

y especifique una región de destino para la copia.

Puede aplicar el cifrado durante el proceso de copia.

Sin embargo, no puede eliminar el cifrado.

durante el proceso de copia.

Entonces, si está haciendo una copia de una AMI encriptada,

entonces la copia se cifrará.

Bueno, eso es todo por esta lección.

Cualquier pregunta, hágamelo saber.

De lo contrario, acompáñenme en el próximo, gracias.