[ENGLISH]

Hello Cloud Gurus, and welcome to this lesson

on increasing scalability using RDS Proxy.

We'll begin with how RDS Proxy works.

We'll take a look at RDS Proxy fault tolerance,

other use cases, and my exam tips as well.

So how does RDS Proxy work?

Well, if you think of a typical application

that is connecting to an RDS database,

the database receives information

from the application through a direct connection.

Well, when we use RDS Proxy,

your application is pointed towards RDS Proxy.

RDS Proxy sits between the user application

and the database, and the way it works is

that it pulls and shares database connections to assist

with application stability and database efficiency,

and then the database receives information

from the application through RDS Proxy.

So when we talk about connection pooling,

this is simply the process of having a pool

of active connections to the backend database,

and by maintaining this active connection,

the application is going to run more efficiently.

Now, in terms of fault tolerance, RDS Proxy is serverless

and its scales automatically.

So it's gonna scale with your workload

and provide connection pooling,

and sharing the database connections,

and because it scales automatically,

it's gonna stop the database from being overloaded

by pooling already established database connections instead

of creating new ones each time,

and it's particularly useful for serverless applications,

which could be growing all the time.

It also preserves application connections during failover.

So if your primary database fails, your application

connection is going to be preserved

during the failover process.

So it's gonna detect failover and route the requests

to your standby database really quickly.

And it's deployable over multiple availability zones

for protection from infrastructure failure,

and with RDS Proxy, you get up to 66% faster failover times

compared with not using it.

Pretty impressive, eh?

Other use cases include applications

with unpredictable workloads, applications that open

and close their database connections infrequently,

that require availability through transient failures,

and applications which have open but idle connections,

and as you can imagine, all of these types of application

are going to benefit from having a pool

of active connections available to use.

So for the exam, just remember that RDS Proxy sits

between your client applications and your RDS database.

It performs connection pooling, so it maintains a pool

of active connections to the backend database,

and that's gonna help you with application scalability

and database efficiency as well.

It's serverless and it scales automatically

to your workload.

Preserves application connections, detects failover,

and route requests to your standby.

It's deployable over multiple AZs

for protection against infrastructure failure,

and with RDS Proxy,

you get up to 66% faster failover times.

So that is it for this lesson.

Any questions, let me know.

Otherwise, I will see you in the next one.

Thank you.

[SPANISH]

Hola Cloud Gurus, y bienvenidos a esta lección.

sobre cómo aumentar la escalabilidad con RDS Proxy.

Comenzaremos con cómo funciona RDS Proxy.

Echaremos un vistazo a la tolerancia a fallas de RDS Proxy,

otros casos de uso, y también mis consejos para el examen.

Entonces, ¿cómo funciona el proxy RDS?

Bueno, si piensas en una aplicación típica

que se conecta a una base de datos RDS,

la base de datos recibe información

desde la aplicación a través de una conexión directa.

Bueno, cuando usamos RDS Proxy,

su aplicación apunta hacia RDS Proxy.

RDS Proxy se encuentra entre la aplicación de usuario

y la base de datos, y la forma en que funciona es

que extrae y comparte conexiones de base de datos para ayudar

con la estabilidad de la aplicación y la eficiencia de la base de datos,

y luego la base de datos recibe información

desde la aplicación a través de RDS Proxy.

Entonces, cuando hablamos de agrupación de conexiones,

esto es simplemente el proceso de tener una piscina

de conexiones activas a la base de datos backend,

y al mantener esta conexión activa,

la aplicación se ejecutará de manera más eficiente.

Ahora, en términos de tolerancia a fallas, RDS Proxy no tiene servidor

y sus escalas automáticamente.

Así que escalará con su carga de trabajo

y proporcionar agrupación de conexiones,

y compartir las conexiones de la base de datos,

y debido a que escala automáticamente,

es va a evitar que la base de datos se sobrecargue

agrupando conexiones de base de datos ya establecidas en su lugar

de crear nuevos cada vez,

y es particularmente útil para aplicaciones sin servidor ,

que podría estar creciendo todo el tiempo.

También conserva las conexiones de las aplicaciones durante la conmutación por error.

Entonces, si su base de datos primaria falla, su aplicación

la conexión va a ser preservada

durante el proceso de conmutación por error.

Así que detectará la conmutación por error y enrutará las solicitudes.

a su base de datos en espera realmente rápido.

Y se puede implementar en varias zonas de disponibilidad

para la protección contra fallas en la infraestructura,

y con RDS Proxy, obtiene tiempos de conmutación por error hasta un 66 % más rápidos

en comparación con no usarlo.

Bastante impresionante, ¿eh?

Otros casos de uso incluyen aplicaciones

con cargas de trabajo impredecibles, aplicaciones que se abren

y cerrar sus conexiones de base de datos con poca frecuencia,

que requieren disponibilidad a través de fallas transitorias,

y aplicaciones que tienen conexiones abiertas pero inactivas,

y como puedes imaginar, todos estos tipos de aplicaciones

se van a beneficiar de tener una piscina

de conexiones activas disponibles para usar.

Entonces, para el examen, recuerde que RDS Proxy se sienta

entre sus aplicaciones cliente y su base de datos RDS.

Realiza la agrupación de conexiones, por lo que mantiene un grupo

de conexiones activas a la base de datos backend,

y eso te ayudará con la escalabilidad de la aplicación

y la eficiencia de la base de datos también.

tiene servidor y escala automáticamente

a su carga de trabajo.

Conserva las conexiones de las aplicaciones, detecta la conmutación por error,

y enruta las solicitudes a tu standby.

Se puede implementar en varias zonas de disponibilidad

para la protección contra fallas en la infraestructura,

y con proxy RDS,

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

Eso es todo por esta lección.

Cualquier pregunta, hágamelo saber.

De lo contrario, te veré en la próxima.

Gracias.